Neural Process Lett (2016) 44:491–517
DOI 10.1007/s11063-015-9471-0

A Study on Multi-Scale Kernel Optimisation via Centered Kernel-Target Alignment

M. Pérez-Ortiz[1] · P. A. Gutiérrez[2] · J. Sánchez-Monedero[2] · C. Hervás-Martínez[2]

Published online: 18 September 2015

Abstract Kernel mapping is one of the most widespread approaches to intrinsically deriving nonlinear classifiers. With the aim of better suiting a given dataset, different kernels have been proposed and different bounds and methodologies have been studied to optimise them. We focus on the optimisation of a multi-scale kernel, where a different width is chosen for each feature. This idea has been barely studied in the literature, although it has been shown to achieve better performance in the presence of heterogeneous attributes. The large number of parameters in multi-scale kernels makes it computationally unaffordable to optimise them by applying traditional cross-validation. Instead, an analytical measure known as centered kernel-target alignment (CKTA) can be used to align the kernel to the so-called ideal kernel matrix. This paper analyses and compares this and other alternatives, providing a review of the literature in kernel optimisation and some insights into the usefulness of multi-scale kernel optimisation via CKTA. When applied to the binary support vector machine paradigm (SVM), the results using 24 datasets show that CKTA with a multi-scale kernel leads to the

This paper is a very significant extension of a preliminar conference version [31] including much additional material: a comprehensive review of kernel model selection methods, a more detailed description of the method considered and a wider experimental section, comparing other multi-scale algorithms, and increasing the number of benchmark datasets. Besides, some hints about when multi-scale kernels are useful and how to initialise them are provided.

✉ M. Pérez-Ortiz
i82perom@uco.es

P. A. Gutiérrez
pagutierrez@uco.es

J. Sánchez-Monedero
jsanchezm@uco.es

C. Hervás-Martínez
chervas@uco.es

[1] Department of Mathematics and Engineering, Universidad Loyola Andalucía, Third Building, 14004 Córdoba, Spain

[2] Department of Computer Science and Numerical Analysis, University of Córdoba, Rabanales Campus, Albert Einstein Building 3rd Floor, 14071 Córdoba, Spain

construction of a well-defined feature space and simpler SVM models, provides an implicit filtering of non-informative features and achieves robust and comparable performance to other methods even when using random initialisations. Finally, we derive some considerations about when a multi-scale approach could be, in general, useful and propose a distance-based initialisation technique for the gradient-ascent method, which shows promising results.

Keywords Kernel-target alignment · Kernel methods · Multi-scale kernel · Parameter selection · Support vector machines · Cross-validation

1 Introduction

The crucial ingredient of kernel methodologies is undoubtedly the application of the so-called *kernel trick* [42], a procedure that maps the data into a higher-dimensional, or even infinite, feature space $\mathcal{H}$ via some mapping Φ. The kernel function implicitly determines the feature space $\mathcal{H}$ in such a way that a poor choice of this function can lead to significantly impaired performance. These choices are related to the definition of a metric between input patterns that fosters correct classification. This optimisation is often performed using a grid-search or cross-validation procedure over a previously defined search space.

Some authors suggest the use of the multi-scale kernel [6] (also known as a multi-parametric, anisotropic or ellipsoidal kernel), where a different kernel parameter is chosen for each feature. The general motivation for the use of multi-scale kernels is that, in real-world applications, the attributes can present very different nature, which hampers the performance of spherical kernels (i.e., with the same kernel width for each attribute) [17,24]. However, the number of parameters (as many as the number of features) makes the computational cost prohibitive when considering a cross-validation technique.

Ideally, we would like to find the kernel that minimises the true risk of a specific classifier for a specific dataset. Unfortunately, this quantity is not accessible; therefore, different estimates or bounds have been developed based on both analytical and experimental knowledge. In most of these cases, a large amount of computation time is needed because the bounds or the algorithms require training the learning machine several times and might even require solving an additional optimisation problem. Moreover, some of the bounds are not differentiable, which means that they must be smoothed to use a gradient descent method [6], which can result in a loose solution.

To overcome these handicaps, a differentiable and simpler approach has been proposed, which is known as kernel-target alignment (KTA) [7,10]. KTA is independent of the learning algorithm, and thus avoids the expensive computational training of the classifier. Essentially, KTA aims to find a kernel function k in a restricted family of kernels such that the induced Gram matrix presents the smallest distance to the ideal kernel matrix, which preserves perfectly the entire training label structure (represented in this case by similarities between patterns). Centred KTA (CKTA) [7] is an extension of KTA that has recently been shown to correlate better with peformance and to avoid some data distribution issues.

The first objective of this paper is to provide an analysis of the literature in kernel optimisation to find the most appropriate method for the multi-scale kernel. As a result of this analysis, several advantages of CKTA have been identified over the rest of the methods: algorithm independence, data distribution independence and simple optimisation. Therefore, this paper considers CKTA to select the multiple parameters of multi-scale kernels (multi-scale CKTA, MSCKTA). The measure is optimised by a gradient ascent procedure in which the free parameters are the different kernel widths of each feature, which, as we will show, leads inherently to the filtering of non-informative features. To the best of the authors' knowledge, this idea has been considered only in [21,24]. In the former one, non-centered KTA is used to optimise a multi-scale version of a special type of kernel for the analysis of biological sequence data, i.e., oligo kernels. In the latter, non-centered KTA is also tested to compare spherical and multi-scale kernels with different optimisation techniques. In the case of [21], although it is not clear that a multi-scale kernel may be in general useful, the author argues that KTA is clearly the best suited method for model selection in high-dimensional search spaces. The experiments performed in this paper include a more general experimental setup with 24 benchmark datasets and statistical comparisons to other uni and multi-scale methodologies, comprising an extensive experimental analysis that has not been performed until now in the context of multi-scale kernels. Moreover, we also propose a novel deterministic distance-based strategy for initialising the coefficient vector for the gradient-ascent algorithm, which is compared to random and fixed initialisations. The results suggest that MSCKTA is a competitive technique that provides binary SVM with a higher flexibility to address heterogeneous real datasets and a better determined feature space that results in simpler SVM models (in terms of the number of support vectors) at a reasonable computational complexity. This additional computational complexity when compared to uni-scale methods is the price to pay to obtain more accurate and simpler models. These conclusions are reinforced by graphically analysing those datasets in which the performance is significantly improved by MSCKTA, thus providing some hints about when the method should be applied. Furthermore, as said, the methodology naturally spans a feature filter which could be beneficial for model interpretation purposes.

The rest of the paper is organized as follows: Sect. 2 shows the literature in kernel optimization for completeness and analyses what methods are better suited for multi-scale kernels; Sect. 3 presents the MSCKTA optimization method; Sect. 4 describes the experimental study and analyses the results obtained; and Sect. 5 outlines some conclusions and future work.

2 Related Research

This section establishes the terminology and notation that will be used throughout this study and briefly reviews the methodologies in the state-of-the-art. The goal in binary classification is to assign an input vector $\mathbf{x}$ to one of $\{\mathcal{C}_{+1}, \mathcal{C}_{-1}\}$ classes (this label will be designated as y, where $y \in \mathcal{Y} = \{\mathcal{C}_{+1}, \mathcal{C}_{-1}\}$), when considering an input space $\mathcal{X} \in \mathbb{R}^d$, where d is the data dimensionality. The training data are assumed to be generated from an i.i.d. $D = \{\mathbf{x}_i, y_i\}_{i=1}^N \in \mathcal{X} \times \mathcal{Y}$ from an unknown distribution $P(\mathbf{x}, y)$. Therefore, the objective in this type of problem is to find a prediction function $f : \mathcal{X} \rightarrow \mathcal{Y},\ f \in \mathcal{F}$ that minimises the expected loss or risk [42].

The methods presented in this paper will be applied to the binary SVM paradigm [3,8]. This algorithm depends on several parameters: the cost parameter C, that controls the trade-off between margin maximisation and error minimisation, and kernel parameters, that appear in the non-linear mapping into the feature space.

The methods that we will present consider a specific kernel function (the Gaussian kernel) and its optimisation. This selection of hyperparameters is crucial because it can drastically degrade or improve the performance. For the case of multi-scale or ellipsoidal Gaussian kernels, the optimisation involves adjusting a vector of parameters. This paper will address precisely this type of problem. In this sense, these parameters can be adjusted following two strategies: algorithm-dependent methods (which require explicit training of the kernel machine) and algorithm-independent (which do not consider any concrete learning algorithm).

2.1 Algorithm-Dependent Estimators for Model Selection

The results obtained by the methods in this subsection are all dependent on the kernel machines considered, therefore the solution for a kernel method would not be equally valid for a different kernel machine. The most widely used approach is the cross-validation method (CV). Although CV is a reliable estimator, it presents an important computational load because it implies the execution of the algorithm on every possible value of the parameter vector. As a step forward, previous research [27] has presented a gradient-based methodology that uses a smooth estimation of the validation function with respect to the SVM parameters.

Leave-one out (LOO) validation is also widespread in the literature because it provides an almost unbiased estimate of the error on the test data. The computational cost in this case is even higher than for the CV. Because of this, different strategies have been considered to provide a bound for the error. These strategies are focused on the specific case of SVMs and allow the optimisation of the kernel parameters. Some of them include the span of support vectors [41] or the Jaakkola–Haussler bound [26] or the Opper–Winther bound [30]. We will focus our study in the radius margin bound and the span of support vectors.

These bounds are related to the concept of Empirical Risk Minimisation (ERM). Related to this concept, a bound on the risk R of any function $f \in \mathcal{F}$ of VC dimension h and especially the one minimising the empirical risk R_{emp} was derived (the radius margin bound). Radius margin bound (RMB) was conceived to obtain an upper bound on the number of errors of the LOO procedure. The number of scientific contributions that use this bound is very significant [6,9,13,14,18]. Nonetheless, ERM is considered to be an ill-posed problem (i.e., a slight change in the training set can entail a large change in the function); thus, several studies have focused on restricting the class of functions by imposing a regularisation constraint [15,20].

Based on the concept of RMB, Vapnik and Chapelle [41] also developed the span-rule to approximate the LOO error, which not only provides a good functional for SVM hyperparameter selection but also reflects the error better. However, this bound is very expensive to compute.

Another branch of the parameter estimation techniques (which will later be used in comparisons named to as Evidence maximisation, EVID) is based on the use of Bayesian methods [38,39] to tune the hyperparameters by maximising the so-called evidence and obtaining predictive class probabilities.

2.2 Algorithm-Independent Estimators for Model Selection

This subsection explores kernel optimisation techniques that do not depend on the learning machine itself. This concept avoids the computational cost of training the algorithm and results in a solution that could be plugged into different learning machines. To accomplish these goals, different analytic concepts are considered, such as the ideal kernel or the inter-cluster separability in the feature space induced by the kernel function.

The notion of ideal kernel has been extensively described and studied [10], where KTA was first proposed. This study was followed by a large amount of scientific contributions related to this estimator [7,11,23,24,33]. KTA arises from the definition of an ideal kernel matrix that perfectly maintains the labelling structure [10]. KTA focuses supporting the information that is inherent to the data to perform the optimal mapping to the feature space (regardless of the algorithm to be employed).[1]

In [16], the notion of ideal kernel was studied by using three different measures of similarity among the matrices (KTA, the Frobenius distance and the correlation). These measures are applied to the optimisation of a spherical kernel on two different datasets. The results of comparing the traditional CV and these three methods show that the performance is similar, but KTA requires lower computational cost than the others.

The concept of distance metric learning has also been used for this purpose [28], by searching for a suitable linear map in the feature space, which computationally leads to a local-optima-free quadratic programming problem for the SVM case. In [43], the inter-cluster distances in the feature space are used to choose the kernel parameters, which involves much less computation time than training the corresponding SVM classifiers.

2.3 Multi-scale Case

Multi-scale kernels have been mainly used with evolutionary algorithms [17,19,32] or gradient-based methods for specific applications [6,24,36]. The main problem with evolutionary approaches is the high computational cost and the necessity of tuning a large number of parameters associated to the algorithm.

With concern for the applications, in [6], an experiment of the multi-scale case with the radius margin bound is performed for handwritten digit recognition. The authors consider this experiment to be as a sanity check experiment which demonstrates the feasibility of choosing multiple kernel parameters for a SVM without leading to overfitting. This approach has been considered in the experimental part of the paper (RMB and MSRMB methods). In [24], the concept of KTA (non-centered) is used to derive a method for optimising multiple hyperparameters of oligo kernels to analyse biological sequence data. Our method extends this idea by considering more robust centered KTA and general purpose Gaussian kernels, and providing extensive experiments and analysis of the potential advantages of this procedure. In [36], a gradient-based optimisation of the radius margin bound was used for the diagnosis of diffuse lung diseases. Although the performances of the SVM classifiers with spherical and multi-scale kernel in the paper do not differ significantly, the authors argue that in the absence of prior knowledge, multi-scale kernels should be preferred. A multi-scale experiment is also performed in [16]; however it achieved worse results than the spherical version at a much higher computational cost. The authors argue that this time increase could be due to the formulation of the optimisation problem, which requires the inversion of a matrix for each update of one of the hyperparameters. In our approach, the optimisation methodology is free of this computational requirement.

The case of multi-scale kernels is also studied in [21] where an evolutionary technique using the validation error is considered [22]. The author argues that this method does not achieve satisfactory performance and leads to over-fitting in contrast to the KTA measure.

3 Multi-scale Centered Kernel-Target Alignment (MSCKTA)

This section introduces the method used in this paper to optimise the parameters of multi-scale kernels. The method combines the concept of centered KTA (CKTA) with respect to the ideal

[1] The KTA measure will be formally defined in Sect. 3.1.

kernel and a gradient ascent methodology. We also include a discussion of the advantages of the method and present a distance-based technique to initialise the gradient-ascent technique.

Some attempts have been made to establish learning bounds for the Gaussian kernel with several parameters and the combination of kernels when considering large margin classifiers [29]. These studies suggest that the interaction between the margin and the complexity measure of the kernel class is multiplicative, thus discouraging the development of techniques for the optimisation of more complex and general kernels. However, recent developments have shown that this interaction is additive [40], rather than multiplicative, yielding then stronger bounds. Therefore, the number of patterns needed to obtain the same estimation error with the same probability for a multi-scale kernel compared to a spherical one grows slowly (and directly depends on the number of features). More specifically, the bound on the required sample size is $\widetilde{\mathcal{O}}(d_\phi + ||\mathbf{w}||/2)$ [40], where $\mathbf{w}$ is the SVM hyperplane and $\widetilde{\mathcal{O}}$ hides logarithmic factors in its argument, the sample size and the allowed failure probability. Note that for the spherical kernel the pseudodimension is $d_\phi = 1$ and for the multi-scale case $d_\phi = d$.

In this paper, the family of kernels is restricted to the well-known Gaussian family, which is parametrised by a d-square matrix of hyperparameters $\mathbf{Q}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{Q}(\mathbf{x}_i - \mathbf{x}_j)\right). \tag{1}$$

For the conventional Gaussian kernel (known as spherical or uni-scale), a single hyperparameter α is used (i.e., $\mathbf{Q} = \alpha^{-2}\mathbf{I}_d$, and $\mathbf{I}_d$ is the identity matrix of size d, and $\alpha > 0$), assuming that the variables are independent. However, one hyperparameter per feature (muti-scale or ellipsoidal Gaussian kernel) can also be used by setting $\mathbf{Q} = diag(\boldsymbol{\alpha}^{-2}) = diag([\alpha_1^{-2}, \ldots, \alpha_d^{-2}])$, with $\alpha_p > 0$ for all p in $\{1, \ldots, d\}$. KTA can be used to obtain the best values for α (the uni-scale method) or $\boldsymbol{\alpha}$ (the multi-scale method). Hereafter, these hyperparameters will be called kernel widths.

3.1 Ideal Kernel

Because kernel functions allow access to the feature space only via input samples, the pairwise inner products between the elements of a finite input set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ are the only information that is available on the geometry of the feature space. This information is embedded in the kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, where k is the kernel function. Most often, kernel algorithms work with this matrix rather than the kernel function itself. Gram matrices contain information about the similarity among the patterns; thus, the idealised kernel matrix $\mathbf{K}^*$ derived using an ideal kernel function k^* [10] will submit the following structure:

$$k^*(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} +1 & \text{if } y_i = y_j, \\ -1 & \text{otherwise,} \end{cases} \tag{2}$$

where y_i is the target of pattern $\mathbf{x}_i$. In other words, $\mathbf{K}^* = \mathbf{y}\mathbf{y}^{\mathrm{T}}$. $\mathbf{K}^*$ will provide information about which patterns should be considered to be similar when performing a learning task. Note that the ideal kernel can be defined only on the training patterns in practice.

Therefore, the problem of finding an optimal set of hyperparameters $\boldsymbol{\alpha}$ is changed to the problem of finding a good approximation $\mathbf{K}_{\boldsymbol{\alpha}}$ (i.e., computed for hyperparameters $\boldsymbol{\alpha}$) for the ideal kernel matrix $\mathbf{K}^*$, given a family $\mathcal{Q}$ of kernels (see Fig. 1). This way of formulating the problem allows us to separate kernel optimisation from kernel machine learning and to reduce the increase in the computational cost of learning more complex kernels (such as multi-scale ones), given that the kernel machine will be unaffected by this higher complexity.

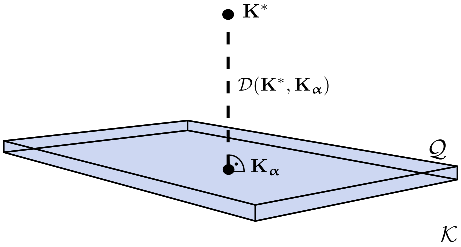

Fig. 1 The most appropriate kernel for learning is $\mathbf{K}_{\boldsymbol{\alpha}}$ (the one nearest the ideal one, $\mathbf{K}^*$, according to some measure of similarity $\mathcal{D}$, being $\mathcal{K}$ the set of positive definite kernels)

In terms of mathematical geometry, for the ideal problem presented in Fig. 1, the kernel matrix that is closest to $\mathbf{K}^*$ can be found by maximising the angle between $\mathbf{K}_{\boldsymbol{\alpha}}$ and $\mathbf{K}^*$.

3.2 Notions of Kernel-Target Alignment (KTA) and Centered KTA

Previous studies have noted several issues in KTA for different pattern distributions [10,34]. A recent study [7] presented a solution to this problem both empirically and theoretically using *centered kernel matrices*, a method that is based on centering the patterns in the feature space and that correlates better with the performance than the original definition of KTA [10]. In fact, the study in [7] shows that non-centered alignment could be even negatively correlated with the accuracy in some cases However, the centered notion of alignment shows good correlation along all datasets.

Centering a positive definite kernel function k consists on centering any feature mapping associated to k, not depending on the mapping chosen. Any kernel matrix $\mathbf{K}$ can be centered by subtracting its empirical expectation:

$$\mathbf{K}_\mathrm{c} = (\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}})^\top (\mathbf{Z} - \mathbf{Z}\mathbf{1}_{\frac{1}{N}}) = \mathbf{K} - \mathbf{K}\mathbf{1}_{\frac{1}{N}} - \mathbf{1}_{\frac{1}{N}}\mathbf{K} + \mathbf{1}_{\frac{1}{N}}\mathbf{K}\mathbf{1}_{\frac{1}{N}}, \qquad (3)$$

where $\mathbf{Z} = \left[\Phi(\mathbf{x}_1) \cdots \Phi(\mathbf{x}_n)\right]$, $\Phi(\cdot)$ is the mapping from the input space to the feature space, and $\mathbf{1}_{\frac{1}{N}}$ is a matrix with all elements equal to $\frac{1}{N}$. $\mathbf{K}_\mathrm{c}$ will also be a positive semi-definite kernel matrix that satisfies $k(\mathbf{x}, \mathbf{x}) \geq 0, \forall\, \mathbf{x} \in \mathcal{X}$ and symmetry.

Let us suppose an ideal kernel matrix $\mathbf{K}^*$ and a real kernel matrix $\mathbf{K}_{\boldsymbol{\alpha}}$ computed for some kernel parameters $\boldsymbol{\alpha}$. The Frobenius inner product between them ($\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^* \rangle_\mathrm{F} = \sum_{i,j=1}^{N} k(\mathbf{x}_i, \mathbf{x}_j) \cdot k^*(\mathbf{x}_i, \mathbf{x}_j)$, where N is the number of patterns) provides information about how *'well'* the patterns are classified in their category. Indeed, in this case, the product could be rewritten as the following equation [see Eq. (2)]:

$$\left\langle \mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^* \right\rangle_\mathrm{F} = \sum_{y_i = y_j} k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j), \qquad (4)$$

where $\sum_{y_i = y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ is related to the within-class distance, and $\sum_{y_i \neq y_j} k(\mathbf{x}_i, \mathbf{x}_j)$ to the between-class distance.

The notion of centered alignment between two kernel matrices $\mathbf{K}_{\boldsymbol{\alpha}} \in \mathbb{R}^{N \times N}$ and $\mathbf{K}^* \in \mathbb{R}^{N \times N}$ such that $||\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}||_\mathrm{F} \neq 0$ and $||\mathbf{K}_\mathrm{c}^*||_\mathrm{F} \neq 0$ [7,10] is defined as:

$$\hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) = \frac{\left\langle \mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}, \mathbf{K}_\mathrm{c}^* \right\rangle_\mathrm{F}}{\sqrt{\left\langle \mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}, \mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}} \right\rangle_\mathrm{F} \left\langle \mathbf{K}_\mathrm{c}^*, \mathbf{K}_\mathrm{c}^* \right\rangle_\mathrm{F}}}, \qquad (5)$$

and this quantity is totally maximised when a kernel can reflect the discriminant properties of the dataset that are used to define the ideal kernel (i.e., $\beta \mathbf{K}_{\boldsymbol{\alpha}} = \mathbf{K}^*$, where β is a scalar).

$\hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*) \geq 0$ because the Frobenius product of any two centered positive semi-definite matrices $\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}$ and $\mathbf{K}_\mathrm{c}^*$ is non-negative. Note that this function is convex in terms of $\mathbf{K}_\alpha$ but becomes non-convex when considering the Gaussian kernel in terms of α [7].

The concentration bound for CKTA and the proof that there exists good alignment-based predictors both for regression and classification can be seen in [7], as well as a risk bound for the convergence of alignment for a finite sample (Theorem 12). Specifically, the alignment for a finite sample is bounded against the alignment expectation, and the expected risk is bounded in terms of alignment in expectation. This risk depends on the complexity of the kernel function and it could be derived by setting a bound on the $\boldsymbol{\alpha}$ parameters.

3.3 Optimisation of MSCKTA

Because of the differentiability of $\hat{\mathcal{A}}$ with respect to the kernel width vector $\boldsymbol{\alpha}$, a gradient ascent algorithm can be used to maximise the alignment between the kernel that is constructed using $\boldsymbol{\alpha}$ and the ideal kernel, as follows:

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}} \hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*). \tag{6}$$

The corresponding gradient vector is composed of partial derivatives $\nabla\hat{\mathcal{A}} = \left[\frac{\partial\hat{\mathcal{A}}}{\partial\alpha_1}, \ldots, \frac{\partial\hat{\mathcal{A}}}{\partial\alpha_d}\right]$, where d is the data dimensionality. In this work, the iRprop$^+$ algorithm is used to optimise the aforementioned centered KTA, because of its proven robustness, advantages over other related methods [25] and previous use in conjunction with KTA [24]. Each parameter α_i will be updated considering the sign of $\frac{\partial\hat{\mathcal{A}}}{\partial\alpha_i}$ but not the magnitude. Although the second partial derivatives can also be computed and used for optimisation, they could actually make the process more computationally costly due to the complexity of this second derivative formula. The alignment derivative with respect to the kernel widths $\boldsymbol{\alpha}$ (see Eq. (5)) is:

$$\frac{\partial\hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}^*)}{\partial\boldsymbol{\alpha}} = \frac{1}{||\mathbf{K}_\mathrm{c}^*||_\mathrm{F}}\left[\frac{\left\langle\frac{\partial\mathbf{K}_{\boldsymbol{\alpha}}}{\partial\boldsymbol{\alpha}}, \mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}}{||\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}||_\mathrm{F}} - \frac{\langle\mathbf{K}_{\boldsymbol{\alpha}}, \mathbf{K}_\mathrm{c}^*\rangle_\mathrm{F}\cdot\left\langle\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}, \frac{\partial\mathbf{K}_{\boldsymbol{\alpha}}}{\partial\boldsymbol{\alpha}}\right\rangle_\mathrm{F}}{||\mathbf{K}_{\boldsymbol{\alpha}_\mathrm{c}}||_\mathrm{F}^3}\right], \tag{7}$$

where $||\mathbf{A}||_F = \sqrt{\langle\mathbf{A}, \mathbf{A}\rangle_\mathrm{F}}$, $\langle\mathbf{A}, \mathbf{B}\rangle_\mathrm{F} = \mathrm{Tr}\left[\mathbf{A}^\top\mathbf{B}\right]$, and, for arbitrary matrices $\mathbf{K}_1$ and $\mathbf{K}_2$, it is satisfied that $\langle\mathbf{K}_{1_\mathrm{c}}, \mathbf{K}_{2_\mathrm{c}}\rangle_\mathrm{F} = \langle\mathbf{K}_1, \mathbf{K}_{2_\mathrm{c}}\rangle_\mathrm{F} = \langle\mathbf{K}_{1_\mathrm{c}}, \mathbf{K}_2\rangle_\mathrm{F}$ [7], which simplifies the computation. Note that the derivative for α_i is computed taking into account the other kernel parameters $\alpha_{j|j\neq i}$ because $\mathbf{K}_{\boldsymbol{\alpha}}$ is included in the formulation. The computation of the KTA takes $\mathcal{O}(N^2)$ operations per parameter α to optimise [21]. Because this optimisation does not involve any additional optimisation problem, it is very fast in practice. Therefore, the computational complexity of MSCKTA is moderated.

For the spherical Gaussian kernel, $\boldsymbol{\alpha} = \alpha\cdot\mathbf{1}$ and the derivative with respect to α can be computed as:

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial\alpha}\right) = \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\alpha^3}\cdot\exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\alpha^2}\right). \tag{8}$$

However, for the case of the multi-scale Gaussian kernel,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{z=1}^{d}\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right) = \prod_{z=1}^{d}\exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right), \tag{9}$$

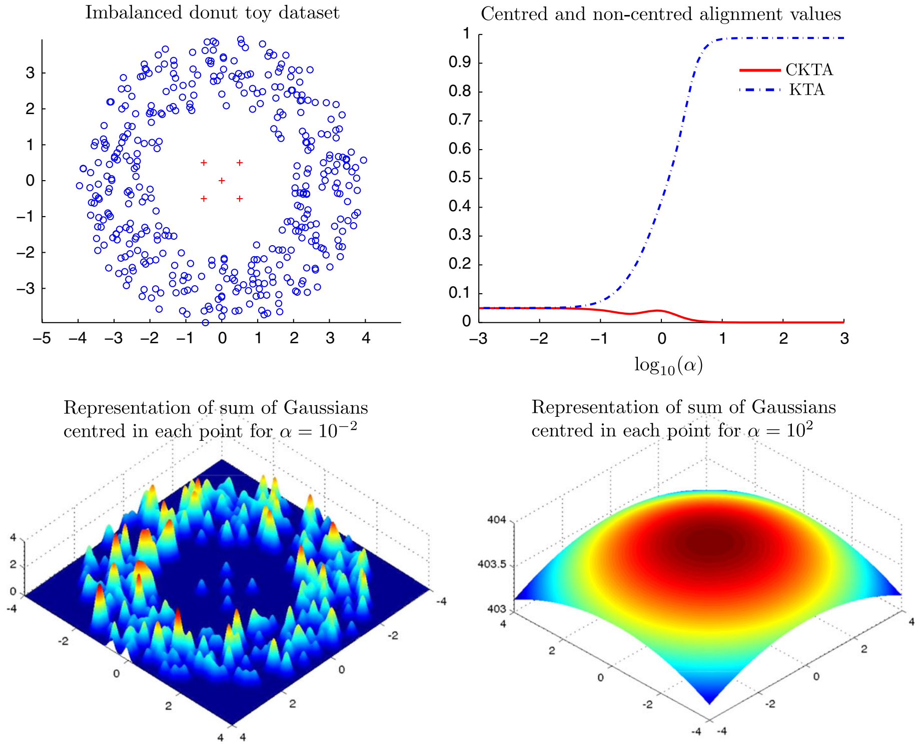

Fig. 2 Two-dimensional imbalanced toy dataset and alignment values obtained for different α values. These values of α have been optimised via CKTA (*left-bottom plot*) and KTA (*right-bottom plot*)

the derivative is the following:

$$\left(\frac{\partial k(\mathbf{x}_i, \mathbf{x}_j)}{\partial \alpha_h}\right) = \frac{(x_{ih} - x_{jh})^2}{\alpha_h^3} \cdot \prod_{z=1}^{d} \exp\left(-\frac{(x_{iz} - x_{jz})^2}{2\alpha_z^2}\right). \tag{10}$$

The specific details and pseudo-code of the iRProp+ algorithm can be checked in [25]. To avoid including positivity constraints in the optimisation problem of $\boldsymbol{\alpha}$ (note that α should vary from 0 to $+\infty$), a logarithmic scale (base 10) is used for the parametrization, which does indeed result in a more stable optimisation. In other words, we consider $\boldsymbol{\alpha} = \{10^{\alpha'_1}, \ldots, 10^{\alpha'_d}\}$ and optimise the functional with respect to $\boldsymbol{\alpha}' = \{\alpha'_1, \ldots, \alpha'_d\}$, avoiding the inclusion of any constraint for $\boldsymbol{\alpha}'$.

The results obtained for KTA and CKTA in an imbalanced toy dataset are shown in Fig. 2. In this case, it can be seen that the optimal kernel parameter (α value with maximum alignment) for KTA and CKTA are different: approximately 10^2 for KTA and 10^{-2} for CKTA. Furthermore, in the bottom part of the figure, where the two solutions are plotted, it can be seen that the kernel value obtained for CKTA is more appropriate for the discrimination of the classes (KTA tends to choose solutions that consider that all the patterns are similar to the rest by setting $\alpha \rightarrow \infty$).

Finally, Fig. 3 shows two toy datasets and the corresponding alignment optimisation surface, where it can be appreciated the necessity of the use of a multi-scale kernel. As can be seen, the optimum values are located in regions where $\alpha_1 \neq \alpha_2$.

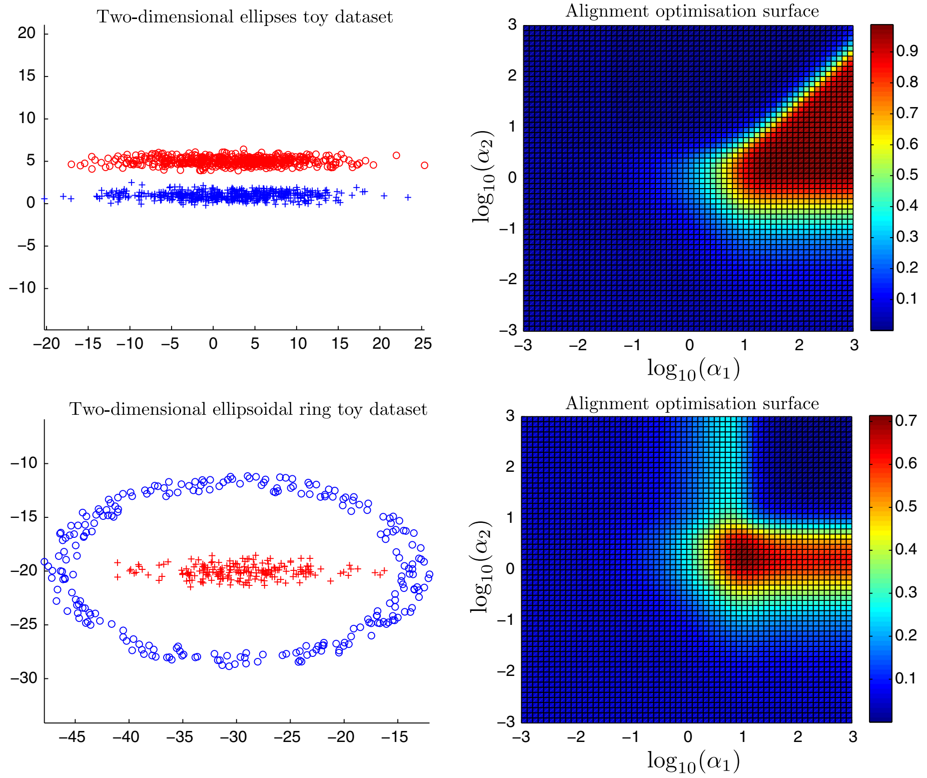

Fig. 3 Two-dimensional toy datasets presenting different class variances per feature and their alignment values when using a grid of values for α_1 and α_2

3.4 Initialisation Scheme of the Gaussian Kernel Parameters

From the KTA definition, it follows that patterns belonging to the same class should present a high similarity, as opposed to patterns belonging to different classes [43]. This idea could be exploited to obtain an initial value of the parameters $\boldsymbol{\alpha}$ of the Gaussian kernel. For example, by fitting a probability distribution to the set of within-class distances $\mathbf{d}_w$.

We assume the exponential distribution $f(\mathbf{d}_w, \lambda) = \lambda \exp(-\lambda \mathbf{d}_w)$, where a close relation can be found between the λ parameter of this exponential distribution and the α parameter in the Gaussian kernel (considering now one single parameter for the kernel). The connection can be seen analysing the following equation and comparing it to the exponential distribution:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\lambda \cdot ||\mathbf{x}_i - \mathbf{x}_j||^2\right), \quad \lambda = \frac{1}{2\alpha^2}, \quad ||\mathbf{x}_i - \mathbf{x}_j||^2 \in \mathbf{d}_w \quad \text{if } y_i = y_j. \qquad (11)$$

Note that the first multiplier in the exponential distribution (i.e. λ) is not required. However, it is more realistic for real-world problems to assume a local neighbourhood-based similarity notion (e.g. for nonlinearly separable problems or multimodal ones), considering that each pattern should be similar to their k-nearest neighbours of the same class. Then, denote $\mathbf{d}_w = \{\mathbf{d}_{w+}, \mathbf{d}_{w-}\}$, where:

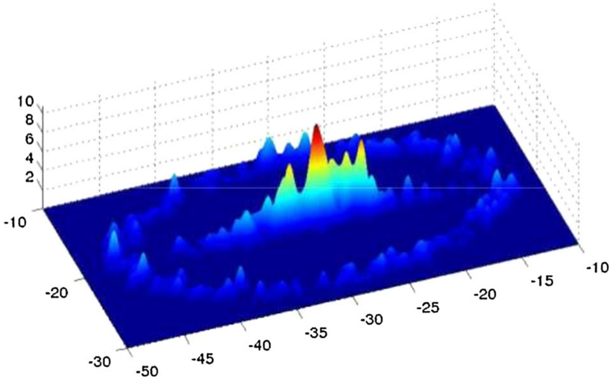

Fig. 4 Representation of the sum of multi-scale Gaussians centered in each point for the ellipsoidal ring toy dataset. The optimal parameter values obtained by the proposed initialisation scheme are $\alpha_1 = 10^{-0.53}$ and $\alpha_2 = 10^{-0.87}$ (note that the data has been standardised beforehand)

$$d_{w+}^{ij} = ||\mathbf{x}_i - \mathbf{x}_j||^2, \quad y_i, y_j = +1, \tag{12}$$

$\mathbf{x}_j$ being one of the k-nearest neighbours of $\mathbf{x}_i$ ($k = 5$ is selected for simplicity). The analogous equation is used for the negative class. Note that, for the exponential distribution, λ is estimated as the mean of $\mathbf{d}_w$. Then, the kernel parameter can be determined as $\alpha = \sqrt{\lambda/2}$. For the multi-scale case, the input features are assumed to be independent, in such a way that λ_i is computed only considering the distance of the patterns for that feature. The result obtained by means of this procedure for the ellipsoidal ring dataset in Fig. 3 can be seen in Fig. 4 where the result is $\alpha_1 = 10^{-0.53}$ and $\alpha_2 = 10^{-0.87}$ (different values per feature). The intuition behind this technique is that kernel parameters are selected depending on the data itself to construct local neighbourhoods of similar patterns.

3.5 Filtering Non-informative Features for the Construction of the Kernel Matrix

An important characteristic of multi-scale kernels is that they provide the opportunity to perform feature selection by filtering attributes with large α_z values. When the Gaussian kernel width $\alpha_z \to \infty$, the kernel matrix computed for that unique feature remains invariant and tends to a matrix of ones, which can be interpreted as feature z not being used for the kernel computation [see Eq. (9)], an omission that could be beneficial for model interpretability. In this subsection, we show that if feature z is non-informative, $\alpha_z \to \infty$ will be considered as an optimum value for the gradient ascent algorithm.

Consider the case of a variable of index z that, for all values of α_z:

$$\left(\frac{N_{y_i}}{N}|\{(x_{iz} - x_{jz})^2 \leq 2\alpha_z^2\}|_{y_i=y_j}\right) = \left(\frac{N_{y_j}}{N}|\{(x_{iz} - x_{jz})^2 \leq 2\alpha_z^2\}|_{y_i \neq y_j}\right), \tag{13}$$

where $|\{\cdot\}|$ denotes the cardinality of the set, and N_{y_i} is the number of patterns with label equal to y_i. This would mean that the number of patterns in the neigbourhood of x_{iz} (neighbourhood defined by $2\alpha_z^2$) will belong similarly to both y_i and y_j. Therefore, this variable can be said to be noisy for all values chosen for the width of the Gaussian (i.e., the similarity does not report information for the classification problem). If this holds for variable z, then:

$$\left(\sum_{y_i=y_j}(1-m_{r_i}-m_{c_j}+m)\cdot k(x_{iz},x_{jz})\right)\simeq$$
$$\left(\sum_{y_i\neq y_j}(1+m_{r_i}+m_{c_j}-m)\cdot k(x_{iz},x_{jz})\right), \tag{14}$$

where $m_{r_i}=\frac{1}{N}\sum_{z=1}^{N}y_i\cdot y_z$, $m_{c_j}=\frac{1}{N}\sum_{z=1}^{N}y_j\cdot y_z$ and $m=\frac{1}{N^2}\sum_{z,h=1}^{N}y_z\cdot y_h$. Note that both $(1-m_r-m_c+m)$ and $(1+m_r+m_c-m)$ are dependent on the label distribution and are used as weights. Under this assumption, it holds:

$$\left\langle \mathbf{K}_{\alpha_z},\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}\simeq 0,\quad \hat{\mathcal{A}}(\mathbf{K}_{\alpha_z},\mathbf{K}^*)\simeq 0, \tag{15}$$

where $\mathbf{K}_{\alpha_z}$ is the kernel matrix obtained by using only the variable z.

Recall that, for the multi-scale case, $\mathbf{K}_{\boldsymbol{\alpha}}=\mathbf{K}_{\alpha_1}\circ\ldots\circ\mathbf{K}_{\alpha_d}$ and that alignment is maximised when $\beta\mathbf{K}_{\boldsymbol{\alpha}}=\mathbf{K}^*$, where β is a scalar.

$$\beta(\mathbf{K}_{\alpha_1}\circ\ldots\circ\mathbf{K}_{\alpha_{z-1}}\circ\mathbf{K}_{\alpha_{z+1}}\circ\ldots\circ\mathbf{K}_{\alpha_d})=\mathbf{K}^*, \tag{16}$$

where $\circ$ represents the hadamard or entrywise product between matrices, i.e., for two matrices A and B of the same dimension, the hadamard product $(A\circ B)$ is another matrix with elements given by: $(A\circ B)_{i,j}=(A)_{i,j}\cdot(B)_{i,j}$.

In this way, the complete kernel matrix can be decomposed as $\mathbf{K}_{\boldsymbol{\alpha}}=\mathbf{K}^*\circ\mathbf{K}_{\alpha_z}$ with the informative features in $\mathbf{K}^*$ and the non-informative one in $\mathbf{K}_{\alpha_z}$. To analyse how the non-informative variable of index z interferes in the kernel matrix, note that:

$$\left\langle \mathbf{K}_{\alpha_z},\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}<\left\langle \mathbf{K}_{\boldsymbol{\alpha}},\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}\leq\left\langle \mathbf{K}^*,\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}, \tag{17}$$

because $\left\langle \mathbf{K}_{\alpha_z},\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}\simeq 0$ and the addition of a non-informative feature will never decrease the angle of the matrix with respect to the ideal one. Given that the maximum alignment is $\hat{\mathcal{A}}(\mathbf{K}^*,\mathbf{K}^*)=1$ and we know that $\hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}},\mathbf{K}^*)\leq\hat{\mathcal{A}}(\mathbf{K}^*,\mathbf{K}^*)$, the gradient of the alignment will converge to the best solution $\hat{\mathcal{A}}(\mathbf{K}_{\boldsymbol{\alpha}},\mathbf{K}^*)=\hat{\mathcal{A}}(\mathbf{K}^*,\mathbf{K}^*)=1$, which is true for (see Eq. (5)):

$$\left\langle (\mathbf{K}^*\circ\mathbf{K}_{\alpha_z})_\mathrm{c},\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}=\sqrt{\left\langle (\mathbf{K}^*\circ\mathbf{K}_{\alpha_z})_\mathrm{c},(\mathbf{K}^*\circ\mathbf{K}_{\alpha_z})_\mathrm{c}\right\rangle_\mathrm{F}\left\langle \mathbf{K}_\mathrm{c}^*,\mathbf{K}_\mathrm{c}^*\right\rangle_\mathrm{F}} \tag{18}$$

$$\mathrm{Tr}((\mathbf{K}^*\circ\mathbf{K}_{\alpha_z})_\mathrm{c}\cdot\mathbf{K}_\mathrm{c}^*)^2=\mathrm{Tr}((\mathbf{K}^*\circ\mathbf{K}_{\alpha_z})_\mathrm{c}^2)\cdot\mathrm{Tr}((\mathbf{K}_\mathrm{c}^*)^2), \tag{19}$$

where $\mathrm{Tr}(\mathbf{A})$ corresponds to the trace of $\mathbf{A}$. The only case that fulfils this is $\mathbf{K}_{\alpha_z}=\mathbf{1}$, and this is the case when $\alpha_z\to\infty$ (see Sect. 3.3), because all the patterns are considered to be equally similar. Therefore, from Eq. (10), $\frac{\partial\mathbf{K}_{\boldsymbol{\alpha}}}{\partial\alpha_z}\to 0$ and $\frac{\partial\hat{\mathcal{A}}}{\partial\alpha_z}\to 0$. Consequently, as the derivative is equal to zero, the case of $\alpha_z\to\infty$ will be an optimum for the gradient-based optimisation algorithm. Note that this filtering is done implicitly without including any sparsity coefficient in the optimisation. Therefore, only non-informative features are removed. However, as it is well-known, whether the gradient ascent algorithm reaches the optimum point depends on the initialisation itself.

The remaining methods studied in this paper do not naturally perform any type of feature selection (i.e., a sparsity coefficient could be added to the optimisation but this step is not performed explicitly) because adding non-informative dimensions to the problem should not damage the SVM solution. This is due to the fact that the capacity control performed by the SVM method is equivalent to some form of regularisation, so that "denoising" is not necessary [37]. In the case of KTA, the optimisation performed recognises directly the variables that do not report information about the labelling or that are very noisy. KTA applied for the purpose of deciding most informative variables (i.e., to perform feature selection) has been only investigated in [34] where KTA is used to optimise a weighting variable for each feature.

4 Experimental Results

This section aims to provide an extensive empirical analysis of the use of multi-scale kernels. Firstly, the goodness of this type of kernel is analysed by plotting an approximation of the feature space that is induced by the kernel. Secondly, several approaches to uni and multi-scale kernels are tested for comparison purposes for a set of 24 binary benchmark datasets, and statistical tests are conducted to analyse whether the method previously presented improves their performance significantly. Thirdly, the feature selection performed by the methodology is analysed, and a deeper analysis of the situations in which a multi-scale approach is useful is presented. Finally, an analysis of the results for different initialisations is included.

Regarding the experimental setup, a stratified 10-fold cross-validation was applied to divide the data, using the same partitions for the methods compared. For each train split, one model is fitted with the train data and evaluated with the test data. The results are taken as the mean and standard deviation over each of the 10 test sets.

As stated before, the optimisation of the gradient-based methods is guaranteed only to find a local minimum; therefore, the quality of the solution can be sensitive to initialisation. Two different approaches are considered in this case. For the comparison with other methods, the initial point for all of the methods tested was fixed at 10^0 (as suggested by other studies [6]). As a different part of the experimental study, we also compare this fixed choice (10^0) with random initialisations and with the deterministic initialisation technique proposed in Sect. 3.4. The gradient norm stopping criterion was set at 10^{-5} and the maximum number of conjugate gradient steps at 10^2 [25].

Table 1 Characteristics for the 24 datasets tested, ordered by the number of attributes d

Dataset	N	d	Dataset	N	d
haberman (HA)	306	3	hepatitis (HE)	155	19
listeria (LI)	539	4	bands (BA)	365	19
mammographic (MA)	830	5	heart-c (HC)	302	22
monk-2 (MO)	432	6	labor (LA)	57	29
appendicitis (AP)	106	7	sick (SI)	3772	33
pima (PI)	768	8	krvskp (KR)	3196	38
glassG2 (GL)	163	9	credit-a (CR)	690	43
saheart (SA)	462	9	specftheart (SP)	267	44
breast-w (BW)	699	9	card (CA)	690	51
heartY (HY)	270	13	sonar (SO)	156	60
breast (BR)	286	15	colic (CO)	368	60
housevotes (HO)	232	16	credit-g (CG)	1000	61

All nominal variables are transformed into binary ones

Several benchmark binary datasets that have different characteristics were tested. Table 1 shows the characteristics of these datasets, where the number of patterns (N) and attributes (d) can be observed. These publicly available real classification datasets were extracted from the UCI repository [2].

4.1 Analysis of the Empirical Feature Space

This subsection explores the notion of empirical feature space to analyse the behaviour of multi-scale kernels by performing a graphical experiment. The empirical feature space is defined as an Euclidean space that preserves the dot product information of $\mathcal{H}$ contained in $\mathbf{K}$. It is possible to verify that the kernel matrix of the training images obtained by this transformation corresponds to $\mathbf{K}$, when considering the standard dot product [35,44]. An approximation of $\mathcal{H}$ can be obtained by limiting the dimensionality of the space. To do so, we have to compute the eigendecomposition of $\mathbf{K}$ and choose the r dominant eigenvalues (and their associated eigenvectors) to project the data while approximating the structure of $\mathcal{H}$. We use this method to represent the embedding space induced by CKTA and MSCKTA for several datasets (see Fig. 5). It can be appreciated from Fig. 5 that, for a multi-scale kernel (right plot of each dataset), the class separation appears to be easier (thus leading to simpler decision functions). Figure 5 also includes information of the eigenvalues of both matrices (i.e., the matrix induced by CKTA and the matrix induced by MSCKTA). This information is represented by a γ value, that corresponds to $\frac{\lambda_1+\lambda_2}{\sum_{i=1}^{N}\lambda_i}$, where λ_i is the i-th eigenvalue for a given matrix ordered in descending order. From these values, it can be observed that the normalised sum of the first two eigenvalues is higher for the kernel matrix computed by MSCKTA, indicating this that these two dimensions incorporate more information about the kernel matrix we are diagonalising. In this sense, previous studies in the literature [4] have demonstrated that, if a kernel present a higher normalised sum of the first eigenvalues than other kernel (applying kernel principal component analysis), it means that the first kernel suited the underlying problem better. In our case, because we are not applying kernel principal components analysis, but a reduction of the empirical kernel map

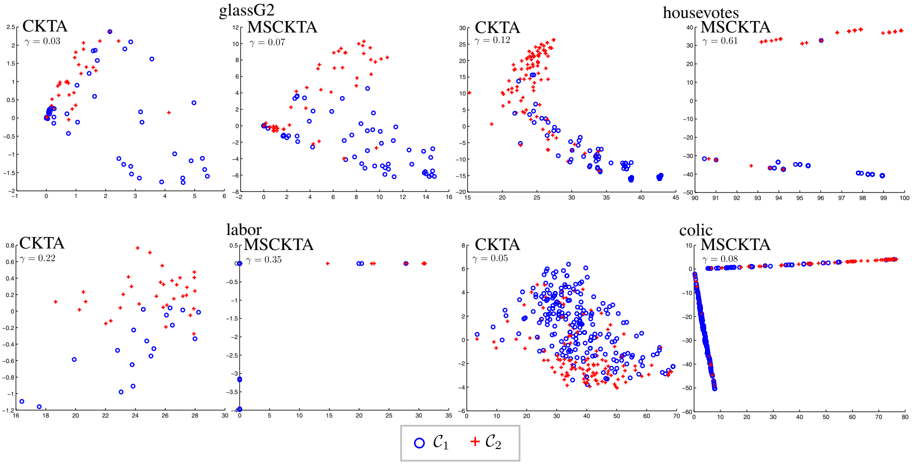

Fig. 5 Graphic showing the 2-dimensional approximation of the empirical feature space induced by CKTA (*left plot* for each dataset) and by MSCKTA (*right plot* for each dataset)

instead, this γ value does not represent the total of data variance covered, but rather the total information represented of the original kernel matrix.

4.2 Experimental Setup

The following methods were compared in the experimentation because they can be considered to be very representative methods in kernel optimisation:

- Cross-validation (CV) using a stratified nested 5-fold cross-validation on the training sets with a single kernel parameter and the C parameter of SVM selected within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$.
- CKTA for optimisating a convex combination of kernels through multiple kernel learning (AMKL) [7]. The kernels used for the optimisation are the ones associated to the kernel width values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$. Once the kernel width is adjusted, the regularisation parameter C of SVM is tuned by minimising the classification error estimated by a stratified nested 5-fold cross-validation on the training sets (with the parameter C within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$). This two stage optimisation method is also referred in the literature as second-order method [5].
- Smoothed span of support vectors (SSV) optimised using a gradient-based methodology [6]. A spherical kernel is used.
- Evidence maximisation (EVID) and its multi-scale version (MSEVID), optimised through a gradient-based methodology [39].
- Smoothed radius margin bound (RMB) and its multi-scale version (MSRMB), optimised using a gradient-based methodology [6].
- CKTA and MSCKTA, optimised using a gradient ascent methodology. Once the kernel width is adjusted, the regularisation parameter C of SVM is tuned by minimising the classification error estimated by a stratified nested 5-fold cross-validation on the training sets (with the parameter C within the values $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$), as in other studies [24].

For SSV, EVID and RMB, the optimisation of C is made together with the kernel parameter. Each benchmark dataset was appropriately standardised (note that this is a very important previous step for our method, specially if one of the objectives is to analyse the final kernel parameters). As suggested in [6], for SSV, EVID, MSEVID, RMB and MSRMB, a modified version of the Polack–Ribiere flavour of conjugate gradients was used to compute the search directions; a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria were used together with the slope ratio method to determine the initial step sizes. The first and second derivatives were used for the optimisation. For CKTA and MSCKTA, the iRprop^{+} method [25] has been selected because of its good behaviour in alignment optimisation [24]. All algorithms were tested with the $L2$ Support Vector Classification (SVC) paradigm (in order to fairly compare with [6]). All datasets, partitions and results are available (in terms of mean and standard deviation) on the website associated with this paper.[2] SSV, EVID, MSEVID, RMB and MSRMB have been tested using the publicy available Matlab code.[3]

[2] http://www.uco.es/grupos/ayrna/gbmskta.

[3] http://olivier.chapelle.cc/ams/.

4.3 Comparisons to Other Methodologies

Table 2 shows the test mean result of each method for each dataset in terms of Acc. Table 3 shows the test mean rankings (1 for the best method and 9 for the worst) and the mean test performance along all of the 24 datasets in terms of the accuracy (Acc), the number of support vectors (SVs), and the centered alignment for training ($\mathcal{A}_{tr}$) and testing sets ($\mathcal{A}_{ts}$). The number of support vectors has been reported because it was noticed that the value chosen for the cost parameter C decreases when KTA was used. This cost parameter controls the trade-off between allowing training errors and forcing rigid margins, in such a way that when $C \rightarrow \infty$ the SVM leads to the hard-margin approach. Therefore, if C is too large, we would have a high penalty for non-separable points and could store too many support vectors, which could lead to overfitting [1].

From these results, several conclusions can be drawn. First, the good performance of the MSCKTA method can be observed by analysing the mean Acc ranking, since it outperforms the other methods, especially the other multi-scale approaches (i.e., MSEVID and MSRMB). Indeed, all of the methods based on KTA (i.e., AMKL, CKTA and MSCKTA) appear to achieve acceptable results when compared to the rest of estimators. Specifically, the goodness of the gradient ascent methodology can be observed when using the multi-scale version. The poor performance of the other multi-scale approaches (compared to the uni-scale versions) could be due to two different reasons: first, the difficulty of optimising the parameters in such a high-dimensional search space (because there could be more directions to move to undesired local optima [21]), and second, the nature of the estimator because, for example, SSV and RMB are considered to be loose bounds on the generalisation errors (this problem has been previously noted in the literature [14]).

Furthermore, despite the use of a more complex kernel, it can be noted that the models obtained using MSCKTA are simpler (i.e., sparser models in terms of the number of support vectors) than the models obtained using the other kernel optimisation methods. This simplicity could result from using a more complex map, which therefore leads to a more 'ideal' transformation of the input space, using the term 'ideal' in the sense of the kernel mapping leading to a perfectly linearly separable set in the feature space.

Finally, when analysing the alignment results ($\mathcal{A}_{tr}$ and $\mathcal{A}_{ts}$), several statements in the literature can be validated. First, the use of the multi-scale approach leads to a far better alignment. Indeed, using this type of kernel achieves even better alignment values than a combination of kernels (AMKL). Moreover, similar alignment values were reported for CV (0.221 and 0.211) and CKTA (0.227 and 0.212), which shows the relationship between alignment optimisation (CKTA) and accuracy optimisation (CV).

Although the necessity of using an ellipsoidal or multi-scale kernel is inherent to the nature of the features of the problem, the probability that the dataset presents attributes that have very different scales is higher as the number of features grows. This hypothesis can be observed in Fig. 6, where the mean accuracies for each dataset are represented for CKTA and MSCKTA, and the datasets have been ordered according to the number of features. As observed, when the number of features is high, the differences between the methodologies grow and the importance of using multiple hyperparameters is thus demonstrated.

To analyse the value of the results, the non-parametric Friedman's test [12] (with $\alpha = 0.05$) has been applied to the mean Acc rankings, rejecting the null-hypothesis that all of the algorithms perform similarly in mean. The confidence interval was $C_0 = (0, F_{(\alpha=0.05)} = 1.99)$, and the corresponding F-value was $7.28 \notin C_0$. The Holm test for multiple comparisons was also applied (see Table 4), and the test concluded that there were statistically significant

Table 2 Mean test values obtained for all the methods and datasets tested in terms of Acc

Dataset	CV	AMKL	SSV	EVID	MSEVID	RMB	MSRMB	CKTA	MSCKTA
haberman	72.86	71.25	72.87	73.52	73.19	**74.51**	73.17	*73.86*	73.51
listeria	**92.40**	69.57	71.93	70.77	*72.54*	64.06	67.91	65.54	72.18
mammographic	82.29	82.17	81.81	81.93	82.41	81.20	*84.34*	82.41	**84.70**
monk-2	96.98	96.99	97.22	96.99	**100.00**	96.75	**100.00**	**100.00**	**100.00**
appendicitis	86.73	85.82	86.82	*87.82*	86.82	87.73	**88.64**	86.82	87.73
pima	76.69	**78.12**	74.23	73.83	74.74	*77.86*	75.91	*77.86*	77.73
glassG2	81.54	78.46	*82.13*	80.29	81.51	79.08	**85.85**	76.58	80.33
saheart	73.38	*73.39*	70.34	66.22	66.22	72.30	70.12	72.53	**74.69**
breast-w	*96.71*	96.56	96.13	**96.86**	96.56	**96.86**	96.28	96.57	*96.71*
heartY	*84.44*	83.70	83.33	82.96	72.96	83.70	72.96	**85.56**	*84.44*
breast	*71.33*	68.89	66.48	69.25	65.42	*71.33*	69.21	70.30	**71.34**
housevotes	*96.54*	*96.54*	95.24	94.82	83.54	96.12	81.54	92.63	**96.96**
hepatitis	*85.21*	85.13	79.38	82.50	79.38	85.13	79.38	**85.88**	84.54
bands	69.31	**72.31**	63.55	67.39	64.38	67.64	64.38	65.71	*69.54*
heart-c	84.40	*84.74*	**85.06**	83.75	65.55	80.23	65.55	82.76	84.40
labor	90.33	*92.33*	64.67	64.67	64.67	64.67	64.67	64.67	**94.33**
sick	96.71	96.69	96.77	96.58	*97.77*	96.74	96.77	97.30	**97.99**
krvskp	99.31	98.72	*99.34*	99.22	89.99	99.28	89.99	**99.41**	*99.34*
credit-a	84.49	85.07	**85.94**	**85.94**	68.84	81.59	68.84	84.78	*85.51*
spectfheart	80.56	**82.05**	79.42	80.94	79.42	80.16	79.42	80.19	*81.71*
card	85.80	*86.52*	**87.12**	76.96	65.51	67.97	65.51	85.94	*86.52*
sonar	75.79	77.67	75.04	55.17	55.17	55.17	55.17	**80.88**	*77.71*
colic	*83.15*	82.60	65.23	65.23	65.23	65.23	65.23	65.50	**85.03**
credit-g	77.10	*77.50*	70.10	70.10	70.10	70.10	70.10	74.10	**77.70**

The best method is in bold face and the second one in italics

Table 3 Mean test values and rankings obtained for all the methods tested and the following metrics: accuracy (Acc), number of support vectors (SVs), training alignment ($\mathcal{A}_{tr}$) and testing alignment ($\mathcal{A}_{ts}$)

Methodology	CV	AMKL	SSV	EVID	MSEVID	RMB	MSRMB	CKTA	MSCKTA
Average Acc	*84.33*	84.21	81.11	80.04	76.62	79.92	77.15	82.28	**85.21**
Average ranking	4.19	4.56	5.69	5.75	6.64	5.50	6.37	*3.94*	**2.35**
Average SVs	*292.15*	379.71	370.01	407.27	489.63	417.68	498.34	292.61	**289.62**
Average ranking	**2.08**	5.12	3.98	5.56	6.41	6.35	7.48	4.18	*3.81*
Average $\mathcal{A}_{tr}$	0.221	*0.231*	0.184	0.195	0.138	0.201	0.151	0.227	**0.379**
Average ranking	5.29	3.33	6.56	6.92	6.79	5.89	5.91	*3.29*	**1.00**
Average $\mathcal{A}_{ts}$	0.211	*0.218*	0.161	0.175	0.110	0.180	0.125	0.212	**0.352**
Average ranking	4.42	*3.58*	6.81	6.75	6.79	5.73	5.92	3.92	**1.08**

The best method is in bold face and the second one in italics

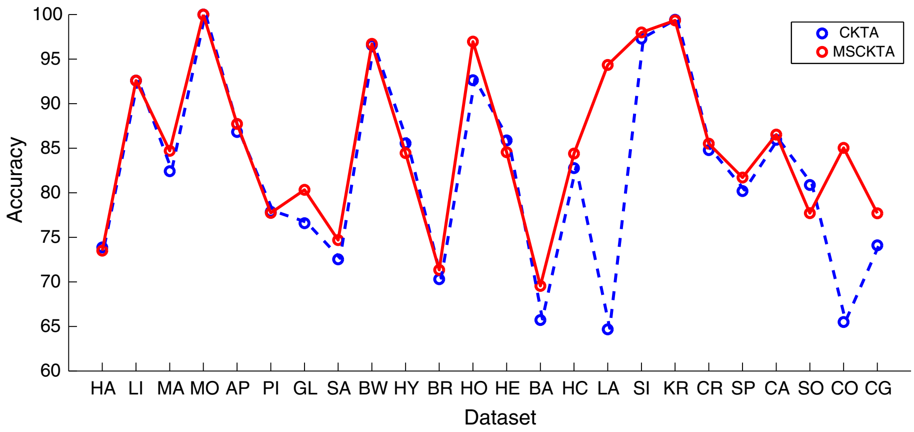

Fig. 6 Accuracy values for CKTA and MSCKTA

Table 4 Comparison in mean *Acc* ranking of the different algorithms using the Holm procedure with MSCKTA as the control algorithm

i	Algorithm	z	p value	Adjusted alpha
1	MSEVID	5.42857	0.00000*	0.00625
2	MSRMB	5.08599	0.00000*	0.00714
3	EVID	4.29542	0.00002*	0.00833
4	SSV	4.21637	0.00002*	0.01000
5	RMB	3.97920	0.00007*	0.01250
6	AMKL	2.79334	0.00522*	0.01667
7	CV	2.31900	0.02040*	0.02500
8	CKTA	2.00277	0.04520*	0.05000

* Statistically significant differences for $\alpha = 0.05$

differences in mean *Acc* ranking for $\alpha = 0.05$ when the MSCKTA was selected as the control method for all of the methods considered.

Table 5 includes the mean runtime values used to optimise all of the parameters for the SVM method for all of the optimisation methods considered. This time includes the seconds needed to adjust all the hyperparameters (by cross-validation or by gradient-descent depending on the parameter and the method), but not the time needed for training and testing the model afterwards. It can be seen that the methods based on CKTA optimising a spherical kernel are computationally efficient (AMKL and CKTA) and present a computational complexity similar to CV, resulting then in a suitable optimisation technique for kernel learning purposes. Furthermore, MSCKTA also obtains reasonable time results (note for example the case of the sonar dataset where there were 60 parameters to optimise but only took 142 secs because of the low number of patterns). Observe that, from all the multi-scale methods, MSCKTA reports an average computational time. The computational time for MSRMB is lower but at the cost of serious performance degradation (see Table 3).

4.4 Feature Selection

Not only can MSCKTA be useful in many real-world applications that present very different attributes, but it also appears to outperform uni-scale approaches (in accuracy) and obtain sparser models than other methods in the literature. Moreover, as stated above, another

Table 5 Mean runtime values (sec) to optimise the parameters with the different methods considered

Dataset	CV	AMKL	SSV	EVID	MSEVID	RMB	MSRMB	CKTA	MSCKTA
haberman	11.05	*6.30*	10.48	27.42	30.89	6.55	8.91	**5.84**	11.82
listeria	*20.28*	24.55	49.06	141.60	171.55	42.89	36.01	**6.86**	71.12
mammographic	48.83	*42.14*	127.54	392.15	287.56	82.35	126.17	**30.19**	108.69
monk-2	*16.54*	17.82	36.88	86.12	139.76	38.50	28.40	**6.49**	25.04
appendicitis	6.52	*2.45*	6.31	7.93	7.54	3.87	3.41	**1.54**	9.21
pima	*40.45*	48.22	234.89	120.15	220.94	81.76	217.29	**25.56**	132.94
glassG2	8.20	3.19	12.81	11.05	24.78	*2.79*	18.26	**2.28**	15.29
saheart	*19.02*	24.00	66.14	27.39	51.17	21.54	49.92	**11.38**	122.60
breast-w	32.86	*14.79*	68.49	173.00	533.76	73.22	24.78	**14.42**	71.61
heartY	11.15	4.62	15.94	20.90	12.63	26.36	*4.59*	**3.73**	46.25
breast	12.52	*5.71*	14.37	25.35	12.08	6.52	20.67	**4.66**	39.05
housevotes	10.13	4.24	10.72	14.14	7.58	14.72	*4.07*	**2.91**	34.71
hepatitis	8.34	5.20	3.67	6.39	6.69	6.87	**1.97**	*2.38*	40.27
bands	17.05	*5.78*	25.33	44.12	21.75	25.93	11.53	**5.66**	79.81
heart-c	14.43	6.64	23.09	29.44	22.10	30.17	*5.97*	**5.01**	115.16
labor	7.06	5.09	*1.40*	1.83	1.78	**1.35**	1.54	1.89	30.87
sick	**1385.19**	1425.50	1676.19	5987.80	37008.14	2766.37	3869.13	**1180.89**	9137.40
krvskp	1172.12	**636.69**	1238.80	7801.57	11973.69	1795.19	1821.03	*1004.08*	10010.25
credit-a	63.51	*58.81*	198.65	188.06	239.42	100.87	64.42	**36.69**	569.15
spectfheart	16.16	12.75	8.42	29.47	31.15	*7.74*	12.85	**7.43**	178.93
card	70.96	60.51	209.00	99.27	233.04	*40.22*	76.28	**37.16**	802.58
sonar	10.95	12.89	11.79	*3.75*	17.36	**1.89**	7.41	3.93	142.61
colic	28.22	11.44	17.67	15.93	66.79	**6.16**	22.24	*7.54*	360.12
credit-g	163.56	172.97	177.30	140.42	710.80	31.91	199	72.08	1952.35
Mean	133.13	*108.84*	176.87	641.47	2159.71	217.32	276.49	**103.36**	1004.49

The best method is in bold and the second one is in italics

Table 6 Percentage of features used for each dataset with MSCKTA and number of attributes for each dataset (a)

Dataset	d	Perc. of features	Dataset	d	Perc. of features
haberman	3	100.00 ± 0.00	hepatitis	19	68.42 ± 47.76
listeria	4	100.00 ± 0.00	bands	19	52.63 ± 51.30
mammographic	5	100.00 ± 0.00	heart-c	22	59.09 ± 50.32
monk-2	6	100.00 ± 0.00	labor	29	27.59 ± 45.49
appendicitis	7	85.71 ± 37.80	sick	33	80.65 ± 40.16
pima	8	62.50 ± 51.75	krvskp	38	44.74 ± 50.39
glassG2	9	88.89 ± 33.33	credit-a	43	57.14 ± 50.09
saheart	9	77.78 ± 44.10	specftheart	44	47.73 ± 50.53
breast-w	9	100.00 ± 0.00	card	51	55.10 ± 50.25
heartY	13	84.62 ± 37.55	sonar	60	36.67 ± 48.60
breast	15	80.00 ± 41.40	colic	60	38.33 ± 49.03
housevotes	16	50.00 ± 51.64	credit-g	61	47.46 ± 50.36

advantage is that it provides us with the opportunity to perform feature selection by filtering attributes with large α_i values. Table 6 shows the percentage of selected features (in terms of the mean and standard deviation) for all of the selected datasets. From this Table, it can be appreciated that the whole set of variables is used in most cases for datasets that have few variables, which indicates that there are no trivial variables for the classification and that MSCKTA is not performing an arbitrary selection. However, as the number of attributes grows, the proportion of attributes selected tends to decrease (note that in 6 of the datasets, the number of selected features is lower than a 50 %).

Note that the rest of algorithm-dependent estimators do not naturally perform feature selection due to the capacity control of SVM methods. However, for unregularised methods, this could be an important characteristic to consider.

4.5 Analysis of Different Initialisation Methods

As said, several random or even fixed initial points for $\boldsymbol{\alpha}$ can be considered. For simplicity, the same initial point has been used for this optimisation in some previous works [6] (the initial point considered for all members of $\boldsymbol{\alpha}$ is 10^0, because it corresponds to the standard deviation of all of the variables in the dataset[4]). For the experiments previously presented in this paper, we thus considered $\alpha_i = 10^0$ in order to fairly compare to other methodologies. However, the suitable choice of these initial points is crucial for the algorithm. In order to analyse the stability of the algorithm with respect to this choice, we compare the results obtained from different initialisations (one initialisation per training/test set was used):

- Fixed initialisation with $\alpha_i = 10^0, \quad i = 1, \ldots, d$.
- Random initialisation with $\alpha_i = 10^{r_i}, \quad i = 1, \ldots, d, r_i \in [-1, 1]$.
- Random initialisation with $\alpha_i = 10^{r_i}, \quad i = 1, \ldots, d, r_i \in [-3, 3]$.
- Deterministic distance-based initialisation proposed in Sect. 3.4.

Table 7 shows the results of these initialisation procedures for 10 datasets (using the same experimental procedure than before), where it can be seen that the proposed distance-based

[4] Note that a data standardisation procedure is applied before optimisation.

Table 7 Results obtained from the different initialisations considered

Dataset	$\alpha_i = 10^0$	$r_i \in [-1, 1]$	$r_i \in [-3, 3]$	Distance-based
mammographic	*84.70 ± 2.90*	*84.70 ± 2.90*	83.25 ± 3.70	*84.82 ± 2.73*
pima	77.73 ± 3.05	*77.86 ± 3.30*	64.85 ± 0.73	*77.87 ± 3.29*
glassG2	80.33 ± 10.68	*80.96 ± 12.41*	53.38 ± 2.74	*82.72 ± 11.02*
saheart	**74.69 ± 7.19**	70.34 ± 8.09	65.37 ± 0.31	*72.07 ± 7.55*
breast-w	**96.71 ± 2.34**	*96.42 ± 2.15*	94.71 ± 2.94	*96.42 ± 2.15*
heartY	**84.44 ± 7.15**	*80.74 ± 11.01*	55.55 ± 0.00	**84.44 ± 7.45**
breast	*71.34 ± 3.50*	70.28 ± 2.35	68.53 ± 3.65	**72.39 ± 3.69**
sick	*97.99 ± 1.20*	97.75 ± 1.01	97.69 ± 0.74	**98.06 ± 1.22**
credit-a	85.51 ± 4.21	*85.80 ± 3.79*	67.83 ± 9.03	**86.09 ± 4.28**
colic	**85.03 ± 5.82**	83.99 ± 7.11	63.87 ± 2.06	*84.48 ± 6.35*

The best method is in bold and the second one is in italics

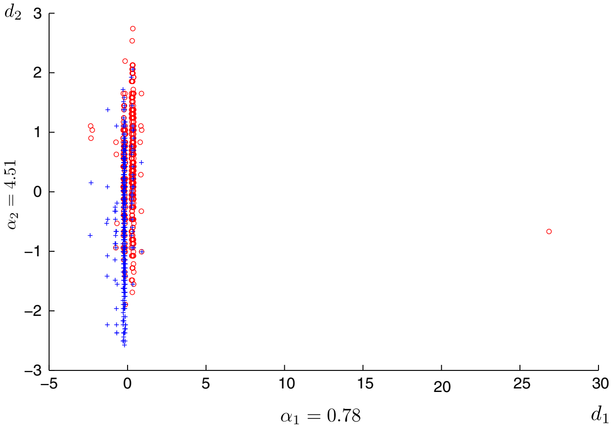

Fig. 7 Two-dimensional plot of the mammographic dataset (the first and second dimensions). In this case, the chosen kernel parameters for each data dimension vary significantly, which clarifies when a multi-scale kernel could be useful. Indeed, MSCKTA achieved better performance for this dataset (84.70 %) than CKTA (82.41 %)

strategy presents the most competitive performance (although close to the one obtained for $\alpha_i = 10^0$). From these results, it can be stated that both a random initialisation between $[-1, 1]$ or just initialising all $\alpha_i = 10^0$ result in a stable and robust optimisation performance (as opposed to initialise the random numbers between $[-3, 3]$, i.e. the cross-validation grid used). These results also show the possibility of initialising the problem in a more intelligent way, to further improve the results in those cases where the best possible performance is required. Note that the remaining methods shown in previous subsections could also benefit from this initialisation.

4.6 Graphical Analysis of the Usefulness of MSCKTA

Several advantages of MSCKTA can be identified: algorithm independence, data distribution independence, simple optimisation, inherent feature selection, sparser SVM models, easy

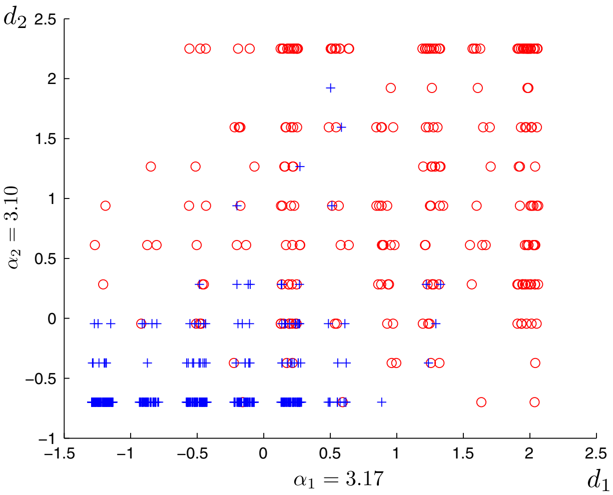

Fig. 8 Two-dimensional plot of the breast-w dataset (the first and second dimensions). Specifically, for this dataset almost the same kernel widths have been chosen for all of the dimensions. In this case, the performances of the CKTA and MSCKTA were similar (96.57 vs 96.71 %, respectively). The graphical representation shows that the patterns can be differentiated by the use of a spherical kernel

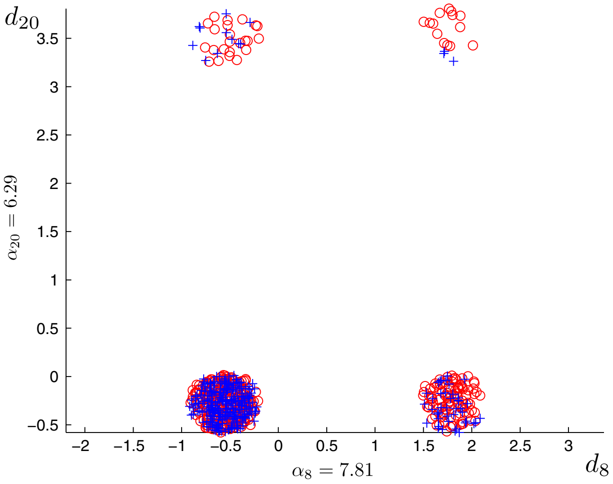

Fig. 9 Two-dimensional plot of the card dataset (8th and 20th dimensions). This figure represents the case of two dimensions used for the kernel computation, i.e., that contain useful information about the labelling structure of the data. Although these dimensions do not allow us to perfectly classify the data (note that the actual dimensionality of the dataset is 51), they give some useful discrimination knowledge about the patterns

extension to other paradigms, to different kernel functions and when only pattern similarities are available.

This last subsection is intended to provide a deeper analysis of the situations in which a multi-scale approach is useful. To provide this analysis, some scenarios in the benchmark datasets that were used are shown in Figs. 7, 8, 9, 10 and 11. For each figure, two of the original input dimensions have been selected and are represented together with the class labelling. Furthermore, the kernel width that is associated with each dimension is included

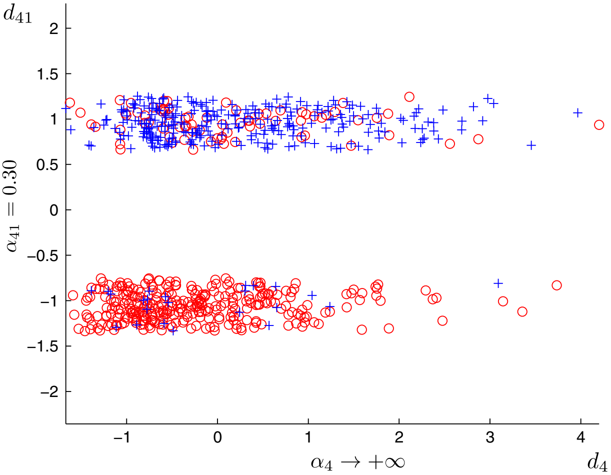

Fig. 10 Two-dimensional plot of the card dataset (4th and 41th dimensions). In this case, the plot represents one significant dimension and one that does not report useful information for classification

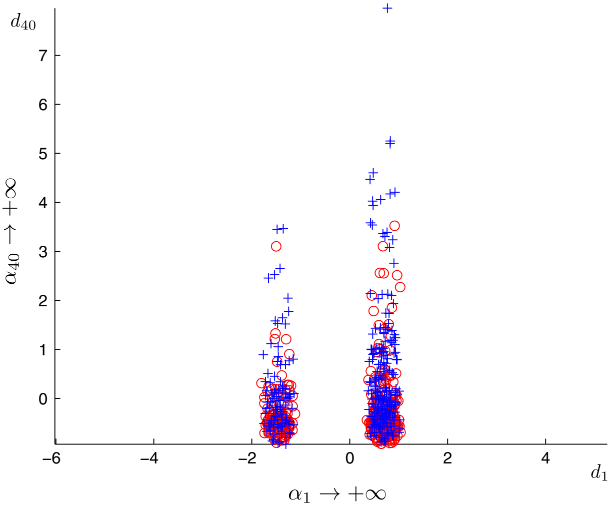

Fig. 11 Two-dimensional plot of the card dataset (first and 40th dimensions). This plot represents the case of two non-significant dimensions for the card dataset, where neither of the variables contains useful information about the labelling structure

in the corresponding axis. These figures have been altered through the use of a random jitter methodology to better visualise the number of patterns per point. It is important to note how MSCKTA assigns equal α values to features with similar class geometry and $\alpha \rightarrow \infty$ values to non-relevant features.

Figure 7 represents the case of a dataset that has two dimensions significant for classification (i.e., that have not been excluded by setting the associated kernel width to infinity); however these two dimensions present different kernel widths. Figures 8 and 9 represent the case of a dataset with two significant dimensions for classification, which also presents similar widths. Figure 10 shows the case in which one of the variables includes significant information and the other does not (i.e., the associated kernel value tends to infinity). Finally, Fig. 11 represents the case in which none of the variables contains useful information about the labelling structure. A discussion of each case is included in the different figure captions.

Although it is difficult to acquire a clear understanding about when to use multi-scale kernels, a general idea can be inferred from the previous results and figures. It is clear that multi-scale kernels should be preferred to spherical ones in cases where the computational time is not a requirement, since multi-scale kernels are more general and can lead to the same solution. Moreover, they can be helpful in the presence of heterogeneous attributes, e.g. when the class-variance of the data varies differently per attribute (see Fig. 7). Finally, multi-scale kernels can also be useful for analysing the most relevant features for the data discrimination, as seen in Figs. 10 and 11.

5 Conclusions

This paper uses the CKTA concept to optimise a multi-scale kernel using a gradient ascent algorithm. The optimisation of the kernel width is usually done by cross-validation, which is computationally unaffordable for multiple kernel widths. The results obtained show that CKTA is highly correlated with performance, and that the optimisation of a multi-scale kernel with this technique leads inherently to a better determined feature space, to feature selection, to significantly better results and to simpler models at a reasonable computational complexity. Moreover, a distance-based initilisation technique is presented which is able to further improve the results for the majority of the datasets considered. Our results encourage the development of a hybrid metaheuristic approach with the gradient ascent method to explore the whole search space and obtain better results. Another direction of future work is a study of the multi-class and regression cases to analyse whether the statements made in this paper are also valid for these learning paradigms.

Acknowledgments This work has been partially subsidized by the TIN2014-54583-C2-1-R project of the Spanish Ministerial Commission of Science and Technology (MINECO), FEDER funds and the P2011-TIC-7508 project of the "Junta de Andalucía" (Spain).

References

1. Alpaydin E (2004) Introduction to machine learning (adaptive computation and machine learning). The MIT Press, Cambridge
2. Asuncion A, Newman D (2007) UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Boser B, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proceedings of the fifth annual ACM workshop on computational learning theory. ACM Press, Pittsburgh, pp 144–152
4. Braun ML, Buhmann JM, Müller KR (2008) On relevant dimensions in kernel feature spaces. J Mach Learn Res 9:1875–1908
5. Chapelle O, Rakotomamonjy A (2008) Second order optimization of kernel parameters. In: Neural information processing systems workshop on kernel learning (NIPS)
6. Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1–3):131–159
7. Cortes C, Mohri M, Rostamizadeh A (2012) Algorithms for learning kernels based on centered alignment. J Mach Learn Res 13:795–828
8. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

9. Cristianini N, Campbell C, Shawe-Taylor J (1998) Dynamically adapting kernels in support vector machines. Adv Neural Inf Process Syst 11:204–210
10. Cristianini N, Kandola J, Elisseeff A, Shawe-Taylor J (2002) On kernel-target alignment. Adv Neural Inf Process Syst 14:367–373
11. Cuturi M (2011) Fast global alignment kernels. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 929–936
12. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
13. Do H, Kalousis A, Woznica A, Hilario M (2009) Margin and radius based multiple kernel learning. In: Proceedings of the European conference on machine learning and knowledge discovery in databases: Part I. Springer-Verlag, New York, pp 330–343
14. Duan K, Keerthi, Poo AN (2003) Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51:41–59
15. Evgeniou T, Pontil M, Poggio T (2000) Regularization networks and support vector machines. Adv Comput Math 13(1):1–50
16. Fauvel M (2012) Kernel matrix approximation for learning the kernel hyperparameters. In: IEEE international geoscience and remote sensing symposium (IGARSS), pp 5418–5421
17. Friedrichs F, Igel C (2004) Evolutionary tuning of multiple svm parameters. Neurocomputing 64:107–117 **(Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004)**
18. Gai K, Chen G, Zhang C (2010) Learning kernels with radiuses of minimum enclosing balls. In: Proceedings of the international conference on neural information processing systems (NIPS),. Curran Associates, Inc, Red Hook pp 649–657
19. Gascón-Moreno J, Ortiz-García EG, Salcedo-Sanz S, Paniagua-Tineo A, Saavedra-Moreno B, Portilla-Figueras JA (2011) Multi-parametric gaussian kernel function optimization for ε-svmr using a genetic algorithm. In: Proceedings of the 11th international conference on artificial neural networks, IWANN'11, vol 2. Springer-Verlag, New York, pp 113–120
20. Girosi F, Jones M, Poggio T (1995) Regularization theory and neural networks architectures. Neural Comput 7:219–269
21. Glasmachers T (2008) Gradient based optimization of support vector machines. Ph.D. thesis
22. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9(2):159–195
23. Howard A, Jebara T (2009) Transformation learning via kernel alignment. Proceedings of the 2009 International conference on machine learning and applications. IEEE Computer Society, Washington, pp 301–308
24. Igel C, Glasmachers T, Mersch B, Pfeifer N, Meinicke P (2007) Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. IEEE/ACM Trans Comput Biol Bioinformatics 4(2):216–226
25. Igel C, Hüsken M (2003) Empirical evaluation of the improved rprop learning algorithms. Neurocomputing 50:105–123
26. Jaakkola TS, Haussler D (1999) Probabilistic kernel regression models. In: Proceedings of the 1999 conference on ai and statistics
27. Keerthi S, Sindhwani V, Chapelle O (2007) An efficient method for gradient-based adaptation of hyper-parameters in svm models. In: Schölkopf B, Platt J, Hoffman T (eds) dvances in neural information processing systems, vol 19. MIT Press, Cambridge
28. Kwok JT, Tsang IW (2003) Learning with idealized kernels. In: Proceedings of the twentieth international conference (ICML). AAAI Press, New York, pp 400–407
29. Lanckriet GRG, Cristianini N, Bartlett PL, Ghaoui LE, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72
30. Opper M, Winther O (2000) Gaussian processes for classification: mean-field algorithms. Neural Comput 12(11):2655–2684
31. Pérez-Ortiz M, Gutiérrez PA, Sánchez-Monedero J, Hervás-Martínez C (2013) Multi-scale Support Vector Machine Optimization by Kernel Target-Alignment. In: European symposium on artificial neural networks, computational intelligence and machine learning (ESANN), pp 391–396
32. Phienthrakul T, Kijsirikul B (2005) Evolutionary strategies for multi-scale radial basis function kernels in support vector machines. In: Proceedings of the 2005 conf. on genetic and evolutionary computation, GECCO '05, pp 905–911
33. Pothin JB, Richard C (2006) A greedy algorithm for optimizing the kernel alignment and the performance of kernel machines. In: Proceedings on the European signal processing conference
34. Ramona M, Richard G, David B (2012) Multiclass feature selection with kernel gram-matrix-based criteria. IEEE Trans Neural Netw Learn Syst 23(10):1611–1623

35. Schölkopf B, Mika S, Burges CJC, Knirsch P, Müller KR, Rätsch G, Smola AJ (1999) Input space versus feature space in kernel-based methods. IEEE Trans Neural Netw 10:1000–1017
36. Shamsheyeva A, Sowmya A (2004) The anisotropic gaussian kernel for svm classification of hrct images of the lung. In: Proceedings of the 2004 intelligent sensors, sensor networks and information processing conference, 2004, pp 439 – 444
37. Smola AJ, Schölkopf B, Müller KR (1998) The connection between regularization operators and support vector kernels. Neural Netw 11(4):637–649
38. Sollich P (2000) Probabilistic methods for support vector machines. Adv Neural Inf Process Syst 12:349–355
39. Sollich P (2002) Bayesian methods for support vector machines: evidence and predictive class probabilities. Mach Learn 46:21–52
40. Srebro N, Ben-david S (2006) Learning bounds for support vector machines with learned kernels. In: In annual conference on learning theory (COLT. Springer, New York, pp. 169–183
41. Vapnik V, Chapelle O (2000) Bounds on error expectation for support vector machines. Neural Comput 12(9):2013–2036
42. Vapnik VN (1998) Statistical learning theory, 1st edn. Wiley, New York
43. Wu KP, Wang SD (2009) Choosing the kernel parameters for support vector machines by the inter-cluster distance in the feature space. Pattern Recognit 42(5):710–717
44. Xiong H, Swamy MNS, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16(2):460–474