UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 05-45
Statistics and Econometrics Series 08
July 2005

Departamento de Estadística
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34) 91 624-98-49

ON THE COMBINATION OF KERNELS FOR SUPPORT VECTOR CLASSIFIERS

I. Martín de Diego, Muñoz, A. and Moguerza, J.M.*

Abstract

The problem of combining different sources of information arises in several situations, for instance, the classification of data with asymmetric similarity matrices or the construction of an optimal classifier from a collection of kernels. Often, each source of information can be expressed as a kernel (similarity) matrix and, therefore, a collection of kernels is available. In this paper we propose a new class of methods in order to produce, for classification purposes, an unique and optimal kernel. Then, the constructed kernel is used to train a Support Vector Machine (SVM). The key ideas within the kernel construction are two: the quantification, relative to the classification labels, of the difference of information among the kernels; and the extension of the concept of linear combination of kernels to the concept of functional (matrix) combination of kernels. The proposed methods have been successfully evaluated and compared with other powerful classifiers and kernel combination techniques on a variety of artificial and real classification problems.

Keywords: Kernel methods, Combination of kernels, Similarity-based classification, Support Vector Machines.

* Martín de Diego, Universidad Carlos III de Madrid, c/ Madrid 126, 28903 Getafe, Spain; Muñoz, Universidad Carlos III de Madrid, c/ Madrid 126, 28903 Getafe, Spain; Moguerza, Universidad Rey Juan Carlos, c/ Tulipán s/n, 28933 Móstoles, Spain.

On the Combination of Kernels for Support Vector Classifiers

Isaac Martín de Diego [a,*], Alberto Muñoz [a]
Javier M. Moguerza [b]

[a] *University Carlos III de Madrid, c/ Madrid 126, 28903 Getafe, Spain*
[b] *University Rey Juan Carlos, c/ Tulipán s/n, 28933 Móstoles, Spain*

Abstract

The problem of combining different sources of information arises in several situations, for instance, the classification of data with asymmetric similarity matrices or the construction of an optimal classifier from a collection of kernels. Often, each source of information can be expressed as a kernel (similarity) matrix and, therefore, a collection of kernels is available. In this paper we propose a new class of methods in order to produce, for classification purposes, an unique and optimal kernel. Then, the constructed kernel is used to train a Support Vector Machine (SVM). The key ideas within the kernel construction are two: the quantification, relative to the classification labels, of the difference of information among the kernels; and the extension of the concept of linear combination of kernels to the concept of functional (matrix) combination of kernels. The proposed methods have been successfully evaluated and compared with other powerful classifiers and kernel combination techniques on a variety of artificial and real classification problems.

Key words: Kernel methods, Combination of kernels, Similarity-based classification, Support Vector Machines.

1 Introduction

Support Vector Machines (SVMs) have proven to be a successful tool for the solution of a wide range of classification problems since their introduction

* Corresponding author. tel: 00 34 916249579, fax: 00 34 916249849
Email addresses: isaac.martin@uc3m.es (Isaac Martín de Diego), alberto.munoz@uc3m.es (Alberto Muñoz), j.moguerza@escet.urjc.es (Javier M. Moguerza).

in [4]. The method uses as a primary source of information a kernel matrix $K(i,j)$, where K is Mercer's kernel and i,j represent data points in the sample. K is a positive-definite symmetric matrix. SVM classifiers construct a maximun margin hyperplane in a feature space nonlinearly related to an input space. By the representer theorem (see for instance [26]), SVM classifiers always take the form $f(x) = \sum_i \alpha_i K(x,i)$. The approximation and generalization capacity of the SVM is determined by the choice of the kernel K [6]. A common way to obtain SVM kernels is to consider a linear differential operator D, and choose K as the Green's function for the operator D^*D, where D^* is the adjoint operator of D [25]. It is easy to show that $\|f\|^2 = \|Df\|^2_{L_2}$ [15]. Therefore, we are imposing smoothing conditions on the solution f. However, it is hard to know in advance which particular smoothing conditions to impose for a given data set. Fortunately, kernels are straightforwardly related to similarity (or equivalently distance) measures, and this information is actually available in many data analysis problems. In addition, working with kernels avoids the need to explicitly work with Euclidean coordinates. This is particularly useful for data sets involving strings, trees, microarrays or text data sets, for instance. Nevertheless, using a single kernel may be not enough to solve accurately the problem under consideration. This happens, for instance, when dealing with text mining problems, where analysis results may vary depending on the document similarity measure chosen [14]. Thus, information provided by a single similarity measure (kernel) may be not enough for classification purposes, and the combination of kernels appears as an interesting alternative to the choice of the 'best' kernel.

The specific literature on the combination of kernels is rather in its beginnings. A natural approach is to consider linear combinations of kernels. This is the approach followed in [18], and it is based on the solution of a semi-definite programming (SDP) problem to calculate the coefficients of the linear combination. The solution of this kind of optimization problems is computationally very expensive [30]. Another difficulty regarding this method is the overfitting due to lack of capacity control. A different approach is proposed in [2]. The method, called MARK, builds a classifier (not the specific kernel matrix) by a boosting type algorithm.

In this paper we describe several methods to build a kernel matrix from a collection of kernels for classification purposes. As a previous step, we derive a natural method to calculate a kernel matrix from an asymmetric similarity matrix. We show that this method is equivalent to a particular method for the combination of two kernels. Then, we provide two general schemes for combining the available kernels. The first framework is based on the quantification of the difference of information among the kernels. The second framework is based on the concept of functional (matrix) combination of kernels, which extends the concept of linear combination of kernels.

The paper is organized as follows. A very intuitive method for the combination of kernels in an asymmetrical classification problem is discussed in section 2. Section 3 describes the proposed methods for combining kernels using the difference of information among kernels. Section 4 describes the proposed methods based on the functional combination of kernels. The experimental setup and results on artificial and real data sets are described in section 5. Section 6 concludes.

2 Building Kernels from Asymmetric Similarities

In order to motivate the problem at hand, in this section we will consider the classification of data whose similarity matrix is asymmetric. In this case, each triangular part of the matrix provides a different source of information. Examples of such matrices arise when considering citations among journals or authors, sociometric data [31], or word association strengths [20]. In the first case, suppose a paper (Web page) i cites (links to) a paper (Web page) j, but the opposite is not true. In the second example, a child i may select another child j to sit next in their classroom, but not reciprocally. In the third case, word i may appear in documents where word j occurs, but not conversely. To be more specific, let S be an asymmetric $l \times l$ similarity matrix (corresponding to a data set of l individuals), that is, $s_{ij} \neq s_{ji}$. Usually, symmetrization is achieved by defining the elements in the kernel matrix as $K^*(i,j) = \dfrac{s_{ij} + s_{ji}}{2}$, that is $K^* = \frac{1}{2}(S + S^T)$. In this way, the same importance is given to s_{ij} and s_{ji}, the two sources of information. This kernel is related to the decomposition of a matrix into its symmetric and skew-symmetric parts:

$$S = \frac{1}{2}(S + S^T) + \frac{1}{2}(S - S^T)\,. \tag{1}$$

In particular, the skew-symmetric part $\frac{1}{2}(S - S^T)$ is being ignored, which implies a loss of information.

In order to take this information into account, the problem can be treated in an alternative way. From a geometric point of view, the solution of a binary classification problem is given by a hyperplane or some type of decision surface. If it is possible to solve a classification problem in this way, then the following topologic assumption must be true: Given a single datum, points in a sufficiently small neighborhood should belong to the same class (excluding points lying on the decision surface). As a consequence, if we are going to classify a data set relying on a given proximity matrix, points close each other using such proximities should in general belong to the same class.

Therefore, we should construct a similarity matrix K^* such that $K^*(i,j)$

should be large for i and j in the same class, and small for i and j in different classes. Since we have two (possibly non-equivalent) sources of information, s_{ij} and s_{ji}, we should define $K^*(i,j)$ as a function $f(s_{ij}, s_{ji})$ that conforms to the preceding rule. We will adopt a simple and intuitive choice:

$$K^*(i,j) = \begin{cases} \max\ (s_{ij}, s_{ji}), & \text{if } i \text{ and } j \text{ belong to the same class,} \\ \min\ (s_{ij}, s_{ji}), & \text{if } i \text{ and } j \text{ belong to different classes.} \end{cases} \tag{2}$$

In this way, if i and j are in the same class, it is guaranteed that $K^*(i,j)$ will be the largest possible, according to the available information. If i and j belong to different classes, we can expect a low similarity between them, and this is achieved by the choice $K^*(i,j) = \min(s_{ij}, s_{ji})$. Hence, the method tends to move closer those points belonging to the same class, and tends to separate points belonging to different classes. This kernel matrix K^* is symmetric, and $K^* = S$ when S is symmetric. Note that this kernel makes sense only for classification tasks, since we need the class labels to build it. In order to make this matrix utile for most classification techniques (including SVMs), K^* should be a positive-definite symmetric matrix.

This problem can be formulated as the combination of two kernels. Let K_1 and K_2 be two matrices defined, respectively, from the upper and lower triangular parts of S, that is:

$$K_1(i,j) = \begin{cases} s_{ij}, & \text{if } i \leq j, \\ s_{ji}, & \text{if } i > j, \end{cases} \quad K_2(i,j) = \begin{cases} s_{ji}, & \text{if } i \leq j, \\ s_{ij}, & \text{if } i > j. \end{cases}$$

Note that K_1 and K_2 have the same diagonal elements. Equation (2) can be stated as follows:

$$K^*(i,j) = \begin{cases} \max\ (K_1(i,j), K_2(i,j)), & \text{if } i,j \text{ belong to the same class,} \\ \min\ (K_1(i,j), K_2(i,j)), & \text{otherwise.} \end{cases} \tag{3}$$

The problem of classifying data with the asymmetric similarity matrix S is thus translated to the problem of combining the two kernels K_1 and K_2, giving rise to an output kernel matrix K^*. This scheme can be expressed in a matricial form. Let y denote the vector of labels, where for simplicity $y_i \in \{-1, +1\}$ (the extension to the multilabel case is straigthforward). Consider the matrix $Y = diag(y)$, whose diagonal entries are the y_i labels. Taking into account

that

$$\begin{cases} \max(a,b) = \dfrac{1}{2}(a+b) + \dfrac{1}{2}|a-b|\,, \\ \\ \min(a,b) = \dfrac{1}{2}(a+b) - \dfrac{1}{2}|a-b|\,, \end{cases} \tag{4}$$

it is direct to show that (3) is equivalent to:

$$K^*(i,j) = \frac{1}{2}(K_1(i,j) + K_2(i,j)) + \frac{1}{2} y_i y_j |K_1(i,j) - K_2(i,j)|\,, \tag{5}$$

and the method reduces to:

$$K^* = \frac{1}{2}(K_1 + K_2) + \frac{1}{2} Y |K_1 - K_2| Y\,. \tag{6}$$

Note the similarity between this expression and the decomposition of an asymmetric matrix shown in (1). In addition to the skew-symmetric information, equation (6) takes into account the label information. In the following this method will be refered as the **'Pick-out'** method.

By analogy, in the next section we will derive kernel combinations of the form

$$K^* = \frac{1}{2}(K_1 + K_2) + \tau Y\, g(K_1 - K_2) Y\,, \tag{7}$$

where $g(K_1 - K_2)$ is a function that quantifies the difference of information between kernels K_1 and K_2, and τ is a positive constant used to control the relative importance of this difference. If K_1 and K_2 tend to produce the same classification results, then $g(K_1 - K_2)$ becomes meaningless and (7) yields $K^* \simeq K_1 \simeq K_2$. Moreover, the methods will be generalized for the combination of more than two kernels. In the following sections we will work with similarity matrices (the transformation of a kernel matrix into a similarity matrix can be made using, for instance, multidimensional scaling [5,3,23]).

3 Quantifying the difference of information among kernels: the general case

Let $K_1, K_2, \ldots, K_M$ be the set of M input kernels available (defined on a data set X), and denote by K^* the desired output combination. The extension of (7) to the combination of more than two kernels is straightforward:

$$K^* = \bar{K} + \tau Y\, g(\sum_{i<j}(K_i - K_j)) Y\,, \tag{8}$$

where $\bar{K}$ is the average of the kernels and g, as before, quantifies the difference of information among the kernels involved. To alleviate the notation, in the following let $V = g(\sum_{i<j}(K_i - K_j))$, so that equation (8) can be restated as:

$$K^* = \bar{K} + \tau Y V Y \,. \tag{9}$$

Within this framework, we will propose several definitions for V, and we will compare the different methods obtained from both the theoretical and the practical points of view.

3.1 Average Kernel Method

If we do not take into account the difference of information among the kernels, that is $\tau = 0$, then:

$$K^* = \bar{K} \,. \tag{10}$$

In this way, the present point of view provides a new interpretation for an old and intuitive method. This method will be refered in the following as **AKM** ('Average Kernel Method').

3.2 Modified Average Kernel Method (MAKM)

Our first proposal builds K^* by taking $V = 1_l 1_l^T$, where 1_l is the l-dimension vector of ones. Thus,

$$K^* = \bar{K} + \tau Y 1_l 1_l^T Y \,. \tag{11}$$

For a pair of data points (i, j) in the sample, (11) takes the form:

$$K^*(i,j) = \bar{K}(i,j) + \tau y_i y_j \,, \tag{12}$$

That is, if i and j belong to the same class, then the method adds an additional quantity τ to the mean of the kernels involved. On the other hand, if i and j belong to different classes then we substract that quantity. Since the expression $Y 1_l 1_l^T Y$ is a kernel, so is K^*. In the following, this method will be refered as **MAKM** ('Modified Average Kernel Method').

The kernel $K_o = Y 1_l 1_l^T Y$ is known as the *ideal kernel* [7]. If the classification target function were known in advance, this would be the optimal kernel function.

Given two kernels K_1 and K_2, their empirical alignment [7] is defined as:

$$A(K_1, K_2) = \frac{\langle K_1, K_2 \rangle}{\sqrt{\langle K_1, K_1 \rangle \langle K_2, K_2 \rangle}} \,, \tag{13}$$

where $\langle K_1, K_2 \rangle = \sum_{i,j=1}^{l} K_1(i,j)K_2(i,j)$ denotes the Frobenius inner product between matrices. The empirical alignment $A(K_1, K_2)$ is a similarity measure between kernels. It can be shown that if there is a high aligment between the ideal kernel K_o and a given kernel K, then the kernel K will have good generalization properties [7]. Therefore, the search for kernel combinations leading to a kernel with a high alignment with the ideal kernel is of special interest.

In the next proposition we show that the MAKM method improves the AKM method.

Proposition 1. *The empirical alignment of the MAKM kernel (K^*) with the ideal kernel is higher than the empirical alignment of the AKM kernel ($\bar{K}$) with the ideal kernel if* $\tau \geq max(-\frac{\langle \bar{K}, K_o \rangle}{l^2}, 0)$.

Proof. The empirical alignments to compare are:

$$A(\bar{K}, K_o) = \frac{\langle \bar{K}, K_o \rangle}{\sqrt{\langle K_o, K_o \rangle}\sqrt{\langle \bar{K}, \bar{K} \rangle}},$$

and

$$A(K^*, K_o) = \frac{\langle K^*, K_o \rangle}{\sqrt{\langle K_o, K_o \rangle}\sqrt{\langle K^*, K^* \rangle}}.$$

Given that,

$$\begin{aligned}
&\langle K_o, K_o \rangle = l^2 \,, \\
&\langle K^*, K_o \rangle = \langle \bar{K} + \tau K_o, K_o \rangle = \langle \bar{K}, K_o \rangle + \tau \langle K_o, K_o \rangle = \langle \bar{K}, K_o \rangle + \tau l^2 \,,
\end{aligned}$$

and

$$\langle K^*, K^* \rangle = \langle \bar{K} + \tau K_o, \bar{K} + \tau K_o \rangle = \langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, K_o \rangle + \tau^2 l^2 \,,$$

it holds that

$$\begin{aligned}
A(\bar{K}, K_o) &= \frac{\langle \bar{K}, K_o \rangle}{l\sqrt{\langle \bar{K}, \bar{K} \rangle}} \,, \\
A(K^*, K_o) &= \frac{\langle \bar{K}, K_o \rangle + \tau l^2}{l\sqrt{\langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, K_o \rangle + \tau^2 l^2}} \,.
\end{aligned}$$

Using the Cauchy-Schwarz inequality it can be shown that $|A(\bar{K}, K_o)| \leq 1$, and

$$\langle \bar{K}, K_o \rangle \leq l \sqrt{\langle \bar{K}, \bar{K} \rangle}\,.$$

Since $\tau > 0$,

$$\tau l^2 \langle \bar{K}, K_o \rangle \leq \tau l^2 [l \sqrt{\langle \bar{K}, \bar{K} \rangle}]\,,$$

and

$$l \langle \bar{K}, K_o \rangle \sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l^2 \langle \bar{K}, K_o \rangle \leq \tau l^2 [l \sqrt{\langle \bar{K}, \bar{K} \rangle}] + l \langle \bar{K}, K_o \rangle \sqrt{\langle \bar{K}, \bar{K} \rangle}\,,$$

Grouping terms,

$$\langle \bar{K}, K_o \rangle [l \sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l^2] \leq l \sqrt{\langle \bar{K}, \bar{K} \rangle} [\langle \bar{K}, K_o \rangle + \tau l^2]\,.$$

Dividing by $l^2 \sqrt{\langle \bar{K}, \bar{K} \rangle} (\sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l)$,

$$\frac{\langle \bar{K}, K_o \rangle}{l \sqrt{\langle \bar{K}, \bar{K} \rangle}} \leq \frac{\langle \bar{K}, K_o \rangle + \tau l^2}{l \sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l^2}\,,$$

and it follows that

$$A(\bar{K}, K_o) = \frac{\langle \bar{K}, K_o \rangle}{l \sqrt{\langle \bar{K}, \bar{K} \rangle}} \leq \frac{\langle \bar{K}, K_o \rangle + \tau l^2}{l [\sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l]}\,. \tag{14}$$

Given that

$$[\sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l]^2 = \langle \bar{K}, \bar{K} \rangle + 2\tau l \sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau^2 l^2 \geq \langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, K_o \rangle + \tau^2 l^2\,,$$

it holds that

$$\frac{1}{l [\sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l]} \leq \frac{1}{l \sqrt{\langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, K_o \rangle + \tau^2 l^2}}\,,$$

and using that $\tau \geq -\frac{\langle \bar{K}, K_o \rangle}{l^2}$,

$$\frac{\langle \bar{K}, K_o \rangle + \tau l^2}{l [\sqrt{\langle \bar{K}, \bar{K} \rangle} + \tau l]} \leq \frac{\langle \bar{K}, K_o \rangle + \tau l^2}{l \sqrt{\langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, K_o \rangle + \tau^2 l^2}} = A(K^*, K_o)\,. \tag{15}$$

From (14) and (15) it follows that

$$A(K^*, K_o) \geq A(K, K_o)\,.$$

□

Remark. Notice that the condition $\tau \geq max(-\frac{\langle \bar{K}, K_o \rangle}{l^2}, 0)$ is not a restriction. If $\langle \bar{K}, K_o \rangle \leq 0$, then the generalization ability of $\bar{K}$ is very low. On the other hand, if $\langle \bar{K}, K_o \rangle \geq 0$, then the positiveness of τ implies that the MAKM kernel will have an increased alignment (and a better generalization ability). Similar results are obtained in [17] by using an alernative kernel $K(i,j) = +1$ if $y_i = y_j$ and $K(i,j) = 0$ if $y_i \neq y_j$.

3.3 The squared matrix (SM) method

Next we propose a method that generalizes (7) for the combination of more than two kernels. A natural extension is to consider $g(\sum_{i<j}(K_i - K_j)) = \sum_i \sum_{j>i}(K_i - K_j)^2$ in (8). Since $\sum_i \sum_j (K_i - K_j)^2 \propto \sum_{m=1}^{M}(K_m - \bar{K})^2$, in (9) we can take:

$$V = \sum_{m=1}^{M} (K_m - \bar{K})^2\,. \tag{16}$$

Notice that (16) is a variability measure. This suggests that expression (8) can be generalized as:

$$\bar{K} + \tau Y g(Var(K_1, K_2, \ldots, K_M))Y\,, \tag{17}$$

where $Var(K_1, K_2, \ldots, K_M)$ is a measure of the variability within the kernels. If all the kernels are equal, the previous expression (17) reduces to the average $\bar{K}$.

Using (16), the combination formula now becomes:

$$K^* = \bar{K} + \tau Y \sum_{m=1}^{M} (K_m - \bar{K})^2 Y\,, \tag{18}$$

where τ plays the same role as in the previous method. In the following, this method will be refered as **SM** ('Squared Matrix' method).

Proposition 2. *The matrix K^* in (18) obtained using the SM method is positive definite.*

Proof. Let $A_m = K_m - \bar{K}$. A_m is symmetric since K_m and $\bar{K}$ are symmetric. Then, there exists an orthogonal matrix Q_m such that $A_m = Q_m^T \Lambda_m Q_m$, where Λ_m is a diagonal matrix whose elements are the eigenvalues of A_m. Now,

$A_m^2 = A_m A_m = Q_m^T \Lambda_m Q_m Q_m^T \Lambda_m Q_m = Q_m^T \Lambda_m^2 Q_m$, that is, A_m^2 is positive semidefine. Since $Y A_m^2 Y = Y Q_m^T \Lambda_m^2 Q_m Y = (Q_m Y)^T \Lambda_m^2 (Q_m Y) = V_m^T \Lambda_m^2 V_m$, $Y A_m^2 Y$ is positive semidefine too. This means that the second term in (18) is positive semidefinite, and therefore, this method provides a matrix K^* arising from a Mercer's kernel.

□

In the next proposition we show that the SM method improves the AKM method.

Proposition 3. *The empirical alignment of the SM kernel with the ideal kernel is higher than the empirical alignment of the AKM kernel with the ideal kernel if* τ *is such that* $\tau[S(V)^2 - l^2 S(V^2)] \geq 2l^2\langle \bar{K}, YVY\rangle - 2S(V)\langle \bar{K}, K_o\rangle$ *and* $\tau \geq max(-\frac{\langle \bar{K}, K_o\rangle}{S(V)}, 0)$, *where* $S(V) = \sum_{i,j=1} V(i,j) = \sum_{i,j=1}^{l} \sum_{m=1}^{M} (K_m - \bar{K})^2$.

Proof. The empirical alignments to compare are:

$$A(\bar{K}, K_o) = \frac{\langle \bar{K}, K_o\rangle}{\sqrt{\langle K_o, K_o\rangle}\sqrt{\langle \bar{K}_o, \bar{K}\rangle}} = \frac{\langle \bar{K}, K_o\rangle}{l\sqrt{\langle \bar{K}, \bar{K}\rangle}},$$

and

$$A(K^*, K_o) = \frac{\langle K^*, K_o\rangle}{\sqrt{\langle K, K_o\rangle}\sqrt{\langle K^*, K^*\rangle}} = \frac{\langle K^*, K_o\rangle}{l\sqrt{\langle K^*, K^*\rangle}}.$$

Given that

$$\begin{aligned}
\langle K^*, K_o\rangle &= \langle \bar{K} + \tau YVY, K_o\rangle \\
&= \langle \bar{K}, K_o\rangle + \tau\langle YVY, K_o\rangle \\
&= \langle \bar{K}, K_o\rangle + \tau\langle YVY, Y 1_l 1_l^T Y\rangle \\
&= \langle \bar{K}, K_o\rangle + \tau 1_l^T V 1_l \\
&= \langle \bar{K}, K_o\rangle + \tau S(V),
\end{aligned}$$

and

$$\begin{aligned}
\langle K^*, K^*\rangle &= \langle \bar{K} + \tau YVY, \bar{K} + \tau YVY\rangle \\
&= \langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2\langle YVY, YVY\rangle \\
&= \langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2),
\end{aligned}$$

it holds that

$$A(K^*, K_o) = \frac{\langle \bar{K}, K_o\rangle + \tau S(V)}{l\sqrt{\langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)}}.$$

The proof follows as in proposition 1:

$$\begin{aligned}
\langle \bar{K}, K_o\rangle &\leq l\sqrt{\langle \bar{K}, \bar{K}\rangle}\,,\\
\langle \bar{K}, K_o\rangle \tau S(V) &\leq l\sqrt{\langle \bar{K}, \bar{K}\rangle}\tau S(V)\,,\\
l\langle \bar{K}, K_o\rangle\sqrt{\langle \bar{K}, \bar{K}\rangle} + \langle \bar{K}, K_o\rangle \tau S(V) &\leq l\sqrt{\langle \bar{K}, \bar{K}\rangle}\tau S(V) + l\langle \bar{K}, K_o\rangle\sqrt{\langle \bar{K}, \bar{K}\rangle}\,,\\
\langle \bar{K}, K_o\rangle[l\sqrt{\langle \bar{K}, \bar{K}\rangle} + \tau S(V)] &\leq l\sqrt{\langle \bar{K}, \bar{K}\rangle}[\langle \bar{K}, K_o\rangle + \tau S(V)]\,,\\
\frac{\langle \bar{K}, K_o\rangle}{l\sqrt{\langle \bar{K}, \bar{K}\rangle}} &\leq \frac{\langle \bar{K}, K_o\rangle + \tau S(V)}{l\sqrt{\langle \bar{K}, \bar{K}\rangle} + \tau S(V)}\,,
\end{aligned}$$

and

$$A(\bar{K}, K_o) = \frac{\langle \bar{K}, K_o\rangle}{l\sqrt{\langle \bar{K}, \bar{K}\rangle}} \leq \frac{\langle \bar{K}, K_o\rangle + \tau S(V)}{l[\sqrt{\langle \bar{K}, \bar{K}\rangle} + \frac{\tau}{l} S(V)]}\,. \tag{19}$$

Now, using that $\tau \geq 0$ and the first assumption in proposition 2,

$$\begin{aligned}
\tau[S(V)^2 - l^2 S(V^2)] &\geq 2l^2\langle \bar{K}, YVY\rangle - 2S(V)\langle \bar{K}, K_o\rangle\,,\\
\tau S(V)^2 + 2S(V)\langle \bar{K}, K_o\rangle &\geq 2l^2\langle \bar{K}, YVY\rangle + \tau l^2 S(V^2)\,,\\
\frac{\tau^2}{l^2} S(V)^2 + \frac{2\tau}{l^2} S(V)\langle \bar{K}, K_o\rangle &\geq 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)\,,\\
\frac{\tau^2}{l^2} S(V)^2 + \frac{2\tau}{l} S(V)\sqrt{\langle \bar{K}, \bar{K}\rangle} &\geq 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)\,,\\
\langle \bar{K}, \bar{K}\rangle + \frac{\tau^2}{l^2} S(V)^2 + \frac{2\tau}{l} S(V)\sqrt{\langle \bar{K}, \bar{K}\rangle} &\geq \langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)\,,\\
[\sqrt{\langle \bar{K}, \bar{K}\rangle} + \frac{\tau}{l} S(V)]^2 &\geq \langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)\,,
\end{aligned}$$

and it holds that

$$\frac{1}{l[\sqrt{\langle \bar{K}, \bar{K}\rangle} + \frac{\tau}{l} S(V)]} \leq \frac{1}{l\sqrt{\langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)}}\,.$$

Using that $\tau \geq -\frac{\langle \bar{K}, K_o\rangle}{S(V)}$,

$$\frac{\langle \bar{K}, K_o\rangle + \tau S(V)}{l[\sqrt{\langle \bar{K}, \bar{K}\rangle} + \frac{\tau}{l} S(V)]} \leq \frac{\langle \bar{K}, K_o\rangle + \tau S(V)}{l\sqrt{\langle \bar{K}, \bar{K}\rangle + 2\tau\langle \bar{K}, YVY\rangle + \tau^2 S(V^2)}}\,.$$

Since

$$A(K^*, K_o) = \frac{\langle \bar{K}, K_o \rangle + \tau S(V)}{l\sqrt{\langle \bar{K}, \bar{K} \rangle + 2\tau \langle \bar{K}, YVY \rangle + \tau^2 S(V^2)}}\,, \tag{20}$$

from (19) and (20) we have:

$$A(K^*, K_o) \geq A(K, K_o)\,.$$

□

3.4 The absolute value (AV) method

An alternative to (16) to measure the variability within the kernels consists in following the L_1 approach by considering $V = \sum_{m=1}^{M} |K_m - \bar{K}|$ in equation (9). Thus, the desired output K^* is built through the formula:

$$K^* = \bar{K} + \tau Y \sum_{m=1}^{M} |K_m - \bar{K}| Y\,, \tag{21}$$

where τ, as before, is a positive constant to control the relative importance given to the difference of information among kernels. Equation (21) constitutes the true generalization of equation (6). In the following, this method will be refered as **AV** ('Absolute Value' method).

Proposition 4. *The empirical alignment of the AV kernel with the ideal kernel is higher than the empirical alignment of the AKM kernel with the ideal kernel if* τ *is such that* $\tau[S(V)^2 - l^2 S(V^2)] \geq 2l^2\langle \bar{K}, YVY \rangle - 2S(V)\langle \bar{K}, K \rangle$ *and* $\tau \geq max(-\frac{\langle \bar{K}, K \rangle}{S(V)}, 0)$*, where* $S(V) = \sum_{i,j=1} V(i,j) = \sum_{i,j=1}^{l} \sum_{m=1}^{M} |K_m(i,j) - \bar{K}(i,j)|$*.*

Proof. Similar to the proof of proposition 3. □

Notice that the AV method does not guarantee positive definiteness of K^*. Several solutions have been proposed to face this problem [23]: A first possibility is to replace K^* by $K^* + \lambda I$, with $\lambda > 0$ large enough to make all the eigenvalues of the kernel matrix positive (a choice for λ is any value larger than the absolute value of the minimun of the eigenvalues of the kernel matrix). Another direct approach uses Multidimensional Scaling (MDS) to represent the data set in an Euclidean space: Consider the spectral decomposition $K^* = Q\Lambda Q^T$, where Λ is a diagonal matrix containing (in decreasing order) the eigenvalues of K^*, and Q is the matrix of the corresponding eigenvectors. Assume that Λ has at least p positive eigenvalues. We can consider a p-dimensional representation by taking the first p columns of Q: $K^* = Q_p \Lambda_p Q_p^T$. An alternative is

to use both positive and negative eigenvalues of K^* to represent the data set in a pseudo-Euclidean space (we will denote this decomposition by PSEUDO) [12]. The matrix K^* is then defined as $K^* = Q|\Lambda|Q^T$. The last possibility we will consider is the definition of a new kernel matrix as $K^{*T}K^*$ [27]. Notice that, in this case, the new kernel is: $K^* = Q\Lambda^2 Q^T$.

In practice, there seems not to be a universally best method to solve this problem [24]. In [13] and [22] SVM classifiers with non-positive definite kernel matrices are discussed, but further analysis is required.

3.5 Generalization of the 'Pick-out' method

Next we show how to generalize (3) for the combination of more than two kernels. The generalization is straightforward by considering:

$$K^*(i,j) = \begin{cases} \max\limits_{1\leq m\leq M} K_m(i,j), & \text{if } i \text{ and } j \text{ belong to the same class}\,, \\ \min\limits_{1\leq m\leq M} K_m(i,j), & \text{if } i \text{ and } j \text{ belong to different classes}\,. \end{cases} \tag{22}$$

In this way, if i and j are in the same class, it is guaranteed that $K^*(i,j)$ will be the largest possible according to the available information. If i and j belong to different classes, we can expect a low similarity between them, and this is achieved by the choice of the minimum kernel value. It can be shown that the AV method reduces to the Pick-out method when two kernels are being combined and $\tau = \frac{1}{2}$ in (21). This is not true in the general case $M > 2$.

3.6 The squared quantity (SQ) method

Our next proposal is a compromise between the two previous ones. In this case, the variability in (17) will be measured element by element (as in the AV method) by squaring (as in the SM method) the difference of elements $K_m(i,j) - \bar{K}(i,j)$. Therefore, in this case:

$$V(i,j) = \sum_{m=1}^{M} (K_m(i,j) - \bar{K}(i,j))^2\,, \tag{23}$$

and the element (i,j) of the output matrix K^* is:

$$K^*(i,j) = \bar{K}(i,j) + \tau y_i y_j \sum_{m=1}^{M} (K_m(i,j) - \bar{K}(i,j))^2\,, \tag{24}$$

where τ plays the same role as in the previous methods. In the following, this method will be refered as **SQ** ('Squared Quantity' method). Figure 1 shows, for $M = 2$, the effect of different values of τ in the second term of the previous expression. The straight lines correspond to the AV method. Notice that this is the upper limiting case of the SQ method. The lower limiting case is represented by the x-axis line, corresponding to the use of $\frac{1}{2}(K_1 + K_2)$. Note that the SQ curves are differentiable everywhere.

As in the previous case, positive semidefiniteness is not assured and the same comments apply.

Proposition 5. *The empirical alignment of the SQ kernel with the ideal kernel is higher than the empirical alignment of the AKM kernel with the ideal kernel if* τ *is such that* $\tau[S(V)^2 - l^2 S(V^2)] \geq 2l^2\langle \bar{K}, YVY\rangle - 2S(V)\langle \bar{K}, K\rangle$ *and* $\tau \geq max(-\frac{\langle \bar{K},K\rangle}{S(V)}, 0)$*, where* $S(V) = \sum_{i,j=1} V(i,j) = \sum_{i,j=1}^{l} \sum_{m=1}^{M} (K_m(i,j) - \bar{K}(i,j))^2$.

Proof. Similar to the proof of proposition 3. □

4 Weighting Methods

In this section, we will propose several new combination methods based on the concept of functional (matrix) combination of kernels, which extends the concept of linear combination of kernels. To motivate the approach, consider the situation in Figure 2. It is a two-class problem where the data in each class are grouped in two clusters. Suppose we have two linear kernels, K_1 and K_2, that induce two linear classifiers $f_1(x)$ and $f_2(x)$, as illustrated in the figure. We seek the best linear combination $K = \lambda_1 K_1 + \lambda_2 K_2$. The induced SVM classifier will take the form:

$$f(x) = \sum_i \alpha_i K(x, x_i) = \lambda_1 \sum_i \alpha_i K_1(x, x_i) + \lambda_2 \sum_i \alpha_i K_2(x, x_i) = \lambda_1 f_1'(x) + \lambda_2 f_2'(x)\,,$$

thus, also a linear classifier (a straight line since K_1 and K_2 are linear kernels). Hence, it is clear that do not exist constants λ_1 and λ_2 to solve the classification problem. However, if the λ_i are functions of the form $\lambda_i(x, y)$, the solution is straightforward: simply take $\lambda_1(x, y) = 1$, $\lambda_2(x, y) = 0$ for data points in the A and B clouds on the left hand side of the figure, and $\lambda_2(x, y) = 1$, $\lambda_1(x, y) = 0$ for data points in the A and B clouds on the right hand side of the figure.

In the general case, for the set of kernels $K_1, K_2, \ldots, K_M$, consider the follow-

ing (functional) weighted sum:

$$K^* = \sum_{m=1}^{M} W_m \otimes K_m \,, \tag{25}$$

where '$\otimes$' denotes the element by element product between matrices (Hadamard product), and $W_m = [w_m(i,j)]$ is a matrix whose elements are nonlinear functions $w_m(i,j)$, with i and j data points in the sample. We assume that $K_m(i,j) \in [0,1] \quad \forall \quad i,j,m$ (otherwise they can be scaled). Notice that if $w_m(i,j) = \mu_m$, where $\mu_m \,, m = 1, \ldots M$ are constants, then the method reduces to calculate a simple linear combination of matrices:

$$K^* = \sum_{m=1}^{M} \mu_m K_m \,. \tag{26}$$

As mentioned in Section 1, in [18] a method is suggested to learn the coefficients μ_m of the linear combination by solving a semi-definite programming problem. So, it is clear that the formulation used in [18] is a particular case of the formula we use. Taking $\mu_m = \frac{1}{M}$, the average of the kernels (AKM) is obtained.

Regarding our proposals, consider the (i,j) element of the matrix K^* in (25):

$$K^*(i,j) = \sum_{m=1}^{M} w_m(i,j) K_m(i,j) \,. \tag{27}$$

This is the general formula of our approximation.

Next we will show how to calculate the weighting functions $w_m(i,j)$. To this aim, we will make use of conditional class probabilities. Consider the pair (i, y_i) and an unlabeled observation j. Given the observed value j, define $P(y_i|j)$ as the probability of j being in class y_i. If i and j belong to the same class this probability should be high. Unfortunately, this probability is unknown and it has to be estimated. In our proposals we will estimate it by $P(y_i|j) = \frac{n_{ij}}{n}$, where n_{ij} is the number of the n-nearest neighbours of j belonging to class y_i. Notice that each kernel induces a different type of neighborhood. Hence, it is advisable to estimate this probability for each kernel representation, that is, for the kernel K_m we will estimate the conditional probabilities $P_m(y_i|j)$.

As in the k-nearest neighbour classifier, the appropriate size of the neighbourhood to estimate the conditional class probabilities could be determined by cross validation or using the optimal value $k = l^{\frac{4}{d+4}}$ (see [29]), where l is the number of observations, and d is the dimension of the problem. Nevertheless, we propose a dynamic alternative method: given two points i and j, we look for the first common neighbour. For each data point (i and j), the size k of the neighbourhood will be determined by the number of neighbours nearer than

the common neighbour. To be more specific, let

$$R(i,n) = \{n\text{-}nearest\ neighbours\ of\ i\}\,,$$

then

$$k = argmin_n\{R(i,n) \cap R(j,n) \neq \emptyset\}\,.$$

Obviously, the size k of the neighbourhood depends on the particular pair of points under consideration.

4.1 The kernel weighting scheme ('KWS')

To motivate our first weighting method, consider the example in Figure 3. It represents the first two coordinates obtained using multidimensional scaling for one single kernel over a training data set.

The weight $w_m(i,j)$ that should be assigned to this kernel in (27) depends on the pair of points (i,j) we are taking into account. For most pairs (i,j) the kernel suits well, but there are three points clearly surrounded by points in the other class. The four possible situations are represented in Figures 4(a) to 4(d):

(a) Two points, both in different classes, but surrounded by points in their own class. In this case the kernel is working properly. The two points under consideration belong to different classes and the similarity between them, respect to all the other similarities, is such that they are clearly in different areas of the space. We are interested in a method that assigns a high value to $w_m(i,j)$.

(b) Two points, both in different classes, and surrounded by points in the other class. In this case the kernel is clearly not working because i and j belong to different classes and they lay in the wrong area of the space. The neighbours of point i belong to the class of point j and the neighbours of point j belong to the class of point i. We are interested in a method that assigns a low value to $w_m(i,j)$.

(c) Two points, both in the same class, and surrounded by points in their own class. The kernel is working right for this pair of points, therefore, we are interested in a method that, in this situation, assigns a high value to $w_m(i,j)$.

(d) Two points, both in the same class, but surrounded by points in the other class. The two points under consideration belong to the same class, but their neighbourhoods belong to the other class. We are interested in a method that assings a low value to $w_m(i,j)$.

We have to define a method that integrates all the available information, namely: the kernel $K_m(i,j)$, the neighbourhood of the points i and j, and the

label information.

Let

$$\rho_m(i,j) = \frac{P_m(y_i|j) + P_m(y_j|i)}{2} = \frac{n_{ij} + n_{ji}}{2n} \,. \tag{28}$$

The weighting scheme we propose is:

$$w_m(i,j) = \gamma K_m(i,j)^{\tau(1-2\rho_m(i,j))y_i y_j} \,, \tag{29}$$

where τ is a positive constant, and γ is introduced to guarantee that the sum of the weights will be 1. In the following, this method will be refered as **KWS** ('Kernel Weighting Scheme').

This weighting scheme increases the weight of $K_m(i,j)$ if $\rho_m(i,j)$ is high and i and j belong to the same class, or if $\rho_m(i,j)$ is low and i and j belong to different classes. On the other hand, the weight $w_m(i,j)$ will be low if the points under consideration belong to the same class but $\rho_m(i,j)$ is low, or if they belong to different classes but $\rho_m(i,j)$ is high.

Notice that if $\rho_m(i,j) = 0.5$, then $w_m(i,j) = 1$, that is, no modification is made on the kernel value $K_m(i,j)$. This situation leads to the average of the kernels, the AKM method (10).

Regarding τ, it is obvious that $\tau = 0$ reduces the KWS method to the AKM method. On the other hand, high values of τ make $w_m(i,j)$ approach extreme values, that is, either close to 0 or close to 1.

Given that K^* is not necessarily a linear combination of kernels, positive definiteness of K^* is not guaranteed and the comments in the previous sections apply.

4.2 The probability weighting scheme ('ProbWS')

This method builds K^* by defining $w_m(i,j)$ in (27) as:

$$w_m(i,j) = \gamma \rho_m^\tau(i,j) \propto (P_m(y_i|j) + P_m(y_j|i))^\tau \,, \tag{30}$$

where γ is introduced to assure that $\sum_m w_m(i,j) = 1$, and τ is a positive constant. If $\tau = 0$ the method becomes to the AKM method. Within this setting, the weights quantify the relative importance of each kernel: If i and j belong to the same class (say y_i), the proportion of the nearest neighbours of j belonging to y_i should be high. Hence, the method favours the kernel whose induced neighbourhood shows the highest agreement with the data label information. In the following, this method will be refered as **ProbWS** ('Probability Weighting Scheme').

As before, positive definiteness of K^* is not guaranteed.

4.3 The exponencial and polynomial weighting scheme methods

The next two methods are influenced by the ideas in [21,8], where the variables are weighted according to their relative discrimination power. We make use of similar ideas to raise the weight of kernels with expected good classification performance and, analogously, to diminish the influence of less informative kernels.

Let

$$\bar{P}(y_i|j) = \frac{1}{M}\sum_{m=1}^{M} P_m(y_i|j)\,, \tag{31}$$

$$\bar{\rho}(i,j) = \frac{\bar{P}(y_i|j) + \bar{P}(y_j|i)}{2}\,, \tag{32}$$

and

$$r_m(i,j) = \frac{\left(\bar{\rho}(i,j) - \rho_m(i,j)\right)^2}{\rho_m(i,j)}\,, \tag{33}$$

where $r_m(i,j)$ is designed to measure the ability of the kernel m to predict $\bar{\rho}(i,j)$. The value of $r_m(i,j)$ will be inversely related to the discrimination power of K_m with respect to the whole set of kernels: The numerator in (33) approaches zero when the information conveyed by K_m tends to be similar to the information collected by the entire set of kernels.

Now, we construct $w_m(i,j)$ as a function of $r_m(i,j)$. The relative relevance of kernel K_m can be evaluated by:

$$w_m(i,j) = \gamma\, exp\left(\tau\,\frac{1}{r_m(i,j)}\right)\,. \tag{34}$$

We call this method 'Exponential Weighting Scheme' (**ExpWS**). The parameter γ assures that the sum of the weights over the number of kernels is 1. The parameter τ is used to control the influence of $r_m(i,j)$ on $w_m(i,j)$. If $\tau = 0$, this influence is ignored, and the method reduces to the AKM method. On the other hand, for large values of τ, changes in r_m will be exponentially reflected in w_m.

A different choice to quantify the relative importante of K_m is given by:

$$w_m(i,j) = \gamma\left(\frac{1}{r_m(i,j)}\right)^{\tau}\,, \tag{35}$$

where γ and τ play the same role as before. Using $\tau = 1, 2$ we have linear and quadratic weighting schemes, respectively. We will refer to this method

as 'Polynomial Weighting Scheme' (**PolyWS**).

4.4 The 'MaxMin' method

This method produces a functional combination of two kernels, namely, the maximun and the minimun of the ordered sequence of kernels, being 0 the weight assigned to the rest of the kernels:

$$K^*(i,j) = \bar{\rho}(i,j)K_{[M]}(i,j) + (1 - \bar{\rho}(i,j))K_{[1]}(i,j)\,. \tag{36}$$

If i and j belong to the same class then the conditional class probabilities $\bar{\rho}(i,j)$ will be high and the method guarantees that $K^*(i,j)$ will be large. On the other hand, if i and j belong to diferent classes the conditional class probabilities $\bar{\rho}(i,j)$ will be low and the method will produce a value close to the minimun of the kernels. In the following, this method will be refered as **MaxMin**.

If we fix $\bar{\rho}(i,j) = 1$ when i and j belong to the same class, and $\bar{\rho}(i,j) = 0$ when i and j belong to different classes, then the MaxMin method (36) reduces to the Pick-out method (22). Therefore, the 'Pick-out' method is the limiting case of the 'MaxMin' method.

4.5 The percentile methods

To end, we propose two methods whose assignation of positive weights $w_m(i,j)$ is based on the order induced by the kernels. Consider the ordered sequence:

$$\min_{1\le m\le M} K_m(i,j) = K_{[1]}(i,j) < K_{[2]}(i,j) < \cdots < K_{[M]}(i,j) = \max_{1\le m\le M} K_m(i,j)\,.$$

The two new methods build each element of K^* using, respectively, the following formulae:

$$K^*(i,j) = K_{[\bar{\rho}(i,j)M]}\,, \tag{37}$$

$$K^*(i,j) = \frac{1}{2}\left(K_{\left[\bar{P}(y_i|j)M\right]} + K_{\left[\bar{P}(y_j|i)M\right]}\right)\,. \tag{38}$$

We will denote these methods by **'Percentil-in'** method and **'Percentil-out'** method, respectively.

If the class probabilities $\bar{P}(y_i|j)$ and $\bar{P}(y_i|j)$ are high, we can expect a high similarity between i and j and both methods will guarantee a high $K^*(i,j)$. If the class probabilities $\bar{P}(y_i|j)$ and $\bar{P}(y_i|j)$ are both low, $K^*(i,j)$ will be also low.

5 Experiments

To test the performance of the proposed methods, a SVM has been trained on several real data sets using the kernel matrix K^* constructed. The value of the parameter τ has been assigned via cross-validation. For the methods based on quantifying the difference of information among kernels (MAKM, SQ, AV, and SM methods), the value of τ has been choosen taking as a reference the bounds in Propositions 1, 3, 4, and 5.

Given a non-labelled data point x, $K^*(x, i)$ has to be evaluated. We can calculate two different values for $K^*(x, i)$, the first one assumming x belongs to class $+1$ and the second assumming x belongs to class -1. For each assumption, all we have to do is to compute the distance between x and the SVM hyperplane, and to assign x to the class corresponding to the largest distance from the hyperplane.

In the following, for all the data sets, we will use 80% of the data for training and 20% for testing.

We have compared the proposed methods with the following classifiers: Multivariate Additive Regression Splines (MARS) [9], Logistic Regression (LR), Linear Discriminant Analysis (LDA), k-Nearest Neighbour classification (k-NN) and SVMs using a RBF kernel $K_c(x_i, x_j) = e^{-\|x_i - x_j\|^2/c}$, with $c = 0.5d$, where d is the data dimension (see [28] for details).

5.1 Artificial data sets

5.1.1 The two-servers data base.

The data set contains 300 data points in $\mathbb{R}^2$. There are two groups linearly separable. This data set illustrates the situation that happens when there are two groups of computers (depending on two servers) sending e-mails among them. Denote by $d_m(i, j)$, $m = 1, 2$ the time that a message takes to travel from computer i to computer j. We have defined two kernel matrices K_1 and K_2 respectively by: $K_m(i, j) = 1 - d_m(i, j)/\max\{d_m(i, j)\}$, $m = 1, 2$, where $d_m(i, j)$ denotes Euclidean distances, and we have corrupted the entries of the matrices at random: for each pair (i, j), one element of the pair $(K_1(i, j), K_2(i, j))$ has been substituted by a random number in $[0, 1]$. Therefore, some entries in K_1 and K_2 are randomly corrupted, but taking the correct information from each matrix the problem would be perfectly solvable. Thus, it is not possible to find a kernel $K^* = \lambda_1 K_1 + \lambda_2 K_2$ that modelizes the problem correctly. The difference between $K_1(i, j)$ and $K_2(i, j)$ is explained by the different ways in which the information may travel between i and j. Usually, the saturation in a

computer network implies that the quickest path is not the shortest path and, therefore, it is not sistematically true that $d_1(i,j) < d_2(i,j)$ (or the opposite). The randomness has been introduced to simulate this phenomenon.

Since we are introducing information about labels in the proposed methods, we expect that our constructed kernels will be useful for data visualization. To check this conjecture, we represent the first two coordinates obtained using MDS on the kernels. The result is shown in Figure 5, and confirms our supposition. The best graphical separation between groups is achieved by the AV and the Pick-out methods.

In order to compare the performance of the methods, the average results over 10 runs of each technique are shown in Table 1. We use MDS to solve the problem of building a positive definite kernel matrix. In this case we have taken into account only the two highest eigenvalues, so we work on a two-dimensional space.

The ProbWS and PolyWS method achieve the best performance (a test error of 2.0 %). Notice that SDP can not be tested here because at least two kernels matrices are needed to use it. Since we corrupt both matrices at random, they are not, in most cases, kernel matrices. The results of the MAKM and the Pick-out methods improve those obtained by using the AKM method (a test error of 6.5 % vs. 9.7 %), achieving the Pick-out method the smallest number of support vectors (4.2 % vs. 16.8 % in MAKM).

5.1.2 A false two-groups classification problem.

A natural question is whether our methods will separate any data set with arbitrary labels. They should not. To test this hypothesis we have generated a normal spherical cloud in $\mathbb{R}^2$, and assigned random labels to the data points. In this case there is no continuous classification surface able to separate the data in two classes. As expected, the classification error rates are close to 50 % for each of the proposed methods.

5.1.3 Two kernels with complementary information.

This data set consists of 400 two-dimensional points (200 per class). Each group corresponds to a normal cloud with mean vector μ_i and diagonal covariance matrix $\sigma_i^2 I$. Here $\mu_1 = (3,3)$, $\mu_2 = (5,5)$, $\sigma_1 = 0.7$ and $\sigma_2 = -0.9$. We have defined two kernels from the projections of the data set onto the coordinate axes. The point in this example is that, separately, both kernels achieve a poor result (a test error of 15 %).

Although out of the scope of this paper, we have used this data set to compare

the different approaches to solve the problem of building a positive definite matrix. Table 2 shows the results obtained with the diverse combination methods. The AKM, MAKM, and SM methods do not appear on this table because the output matrices obtained using these methods are, in fact, kernel matrices. Using $K^T K$ as kernel involves significantly less support vectors than using the other methods. On the other hand, adding a quantity to the diagonal of the eigenvalue matrix increases the percentage of support vectors. Looking at our experimental results, the use of MDS or $K^T K$ seems to be a reasonable choice (see Table 2 for the details).

Table 3 shows that the best results, in general, are obtained using our combination methods. In particular, the ExpWS and KWS methods attain the best overall results (test error of 3.8 % and 4.0 % respectively). In this case, the defined kernel matrices (being defined from projections) convey less information about the data set than the original Euclidean coordinates. The k-NN, LR, LDA, MARS and SVM methods start from the original data points coordinates (not from the previously defined kernel matrices). Nevertheless, most of our methods achieve very similar results to those obtained using these classical methods, and improve the alternative methods that use the same information, SDP and MARK-L (test error of 12.9 % and 8.4 % respectively).

5.2 Real data sets

5.2.1 Cancer Data Set

In this section we have dealt with a database from the UCI Machine Learning Repository: the Breast Cancer data set [19]. The data set consists of 683 observations with 9 features each. For this data set we have combined three kernels: a polynomial kernel $K_1(x,y) = (1 + x^T y)^2$, a RBF kernel $K_2(x,y) = exp(-||x-y||^2)$ and a linear kernel $K_3(x,y) = x^T y$. We have normalized the kernel matrices so that their entries belong to the $[0,1]$ interval: $K(x,y) = (K(x,y) - \min(K))/(\max(K) - \min(K))$.

The results, averaged over 10 runs, are shown in Table 4. All our methods improve the best individual SVM performance (achieved using a linear kernel). The MAKM method shows the best overall performance (a test error of 2.8 %). Our methods improve the alternative methods that combine kernel information, SDP and MARK-L (test error of 6.2 % and 4.2 % respectively). All our combination methods provide better results than the SVM with a single RBF kernel (a test error of 4.2 %), using usually significantly less support vectors. The lowest percentage of support vectors (2.9 %) is associated with the AV method, being the percentage of support vectors 3.1 %.

5.2.2 An alternative to parameter selection.

It is well known that the choice of kernel parameters is often critical for the good performance of SVMs. Combining kernels provides a solution that minimizes the effect of a bad parameter choice. Next we illustrate this situation using a collection of RBF kernels on the cancer data set. Let $\{K_1, \ldots, K_{12}\}$ be a set of RBF kernels with parameters c =0.1, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 respectively. We use MDS to solve the problem of building a positive definite matrices. Table 5 shows the performance of the proposed methods when combining all these kernels. Again, the results have been averaged over 10 runs. The MaxMin method, the Percentil-in method, and the Percentil-out method improve the best RBF kernel under consideration (test errors of 2.8 % for the three methods vs. 3.1 %). This results are specially significant because these three methods are parameter free, and they could be used as an alternative to the RBF parameter selection. In particular, the results provided by all the combination methods are not degraded by the inclusion of kernels with a bad generalization performance.

5.2.3 A handwritten digit recognition problem.

The experiment in this section concerns a binary classification problem: the recognition of digits '7' and '9' from the Alpaydin and Kaynak database [1]. The data set is made up by 1128 records, represented by 32 $\times$ 32 binary images. We have employed three different methods to specify features in order to describe the images. The first one is the 4×4 method: features are defined as the number of ones in each of the 64 squares of dimension 4×4 . The second method was introduced by Frey and Slate [10]: 16 attributes are derived from the image, related to the horizontal/vertical position, width, height, etc. The last method under consideration was designed by Fukushima and Imagawa [11]: features are defined as a collection of 12 different representations in a 4×4 square. This is a typical example with several different sources of information and probably complementary. We have used these representations to calculate three kernels using the Euclidean distance. When K^* is not positive defininite, $K^* + \lambda I$ has been used as a kernel, with λ equaling the absolute value of the minimun eigenvalue of K^* multiplied by 1.01.

The classification performance for all the methods is tabulated in Table 6. For each classical method we have chosen the individual representation that provides the best results, and then we check if the combination methods are able to achieve the same performance. In particular, we have taken the 4×4 representation to train the k-NN, MARS, LR and LDA methods. To train the SVM we have taken the Fukushima and Imagawa representation.

The Percentil-in method achieves the best results among the methods based on

the use of kernels (test error of 1.1 %). Furthermore, the AV, SM, Percentil-out, MaxMin, Pick-out, SQ, and KWS combinations improve the results obtained using the rest of the techniques except k-NN. The excellent performance of k-NN is unsurprising, since this method is specially efficient when working with sparse data sets in high dimensional settings.

5.2.4 A text data base

To check the methods in a high dimensional setting, we will work on a small text data base with two groups of documents. The first class is made up of 296 records from the LISA data base, with the common topic 'library science'. The second class contains 394 records on 'pattern recognition' from the INSPEC data base. There is a mild overlap between the two classes, due to records dealing with 'automatic abstracting' . We select terms that occur in at least 10 documents (obtaining 982 terms). Labels are assigned to terms by voting on the classes of documents in which these terms appear. The task is to correctly predict the class of each term. Following [20], we have defined the kernel K_1 by $K_1(i,j) = \frac{|x_i \wedge x_j|}{|x_i|} = \frac{\sum_k |\min(x_{ik}, x_{jk})|}{\sum_k |x_{ik}|}$, where $|x_i|$ measures the number of documents indexed by term i, and $|x_i \wedge x_j|$ the number of documents indexed by both i and j terms. Similarly, $K_2 = \frac{|x_i \wedge x_j|}{|x_j|}$. Therefore, $K_1(i,j)$ may be interpreted as the degree in which the topic represented by term i is a subset of the topic represented by term j. This numeric measure of subsethood is due to Kosko [16]. The task is to classify the database terms using the information provided by both kernels. Note that we are dealing with about 1000 points in 600 dimensions, and this is a near empty set. This means that it will be very easy to find a hyperplane that divides the two classes. Notwithstanding, the example is still useful to guess the relative performance of the proposed methods. Following the scheme of the preceding examples, Table 7 shows the results. In this case, we have used $K^{*T}K^*$ as kernel to solve the problem of building a positive definite matrix.

Our proposal of methods for the combination of kernels clearly outperforms the rest of the methods. In particular, the AV method achieves the best performance (test error of 0.8 % vs 1.4 % of SDP).

6 Conclusions

In this work we have proposed several techniques for the combination of kernels within the context of SVM classifiers. The proposed framework is based on the natural idea that individuals belonging to the same class should be similar. This is supported by the fact that the suggested methods compare

favorably theoretically and computationally to other well established classification techniques (and also to other techniques for the combination of kernels) in a variety of artificial and real data sets.

Within the group of new techniques proposed in this paper, there is not an overall best method, but using a score over the experiments the three best schemes are the MaxMin, Percentil-in and AV methods. Regarding the comparison with other techniques, consistently the best method in the experiments belongs to this new class of techniques.

Acknowledgements

The authors thank Prof. Robert P.W. Duin and Elżbieta Pękalska for useful discussions. This work was partially supported by Spanish grants TIC2003-05982-C05-05 (MCyT) and PPR-2003-42 (URJC).

References

[1] E. Alpaydin and C. Kaynak. *Cascading Classifiers.* Kybernetika 34 (4) 369-374, 1998.

[2] K. Bennett, M. Momma, and J. Embrechts. *MARK: A Boosting Algorithm for Heterogeneous Kernel Models.* Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[3] I. Borg and P. Groenen. *Modern Multidimensional Scaling.* Springer-Verlag, New York, 1997.

[4] C. Cortes and V. Vapnik. *Support Vector Networks.* Machine Learning, 20:273-297, 1995.

[5] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling.* Chapman & Hall, London, 1995.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[7] N. Cristianini, J. Shawe-Taylor, A. Elisseeff and J. Kandola. *On Kernel-Target Alignment.* Advances in Neural Information Processing Systems,14, Cambridge, MA: MIT Press, 2001.

[8] C. Domeniconi, J. Peng and D. Gunopulos *Adaptive Metric Nearest Neighbor Classification.* Proc. of IEEE Conf. on CVPR, 517-522, 2000.

[9] J. Friedman. *Multivariate adaptative regression splines (with discussion).* Annals of Statistics, vol. 19, no. 1, 1-141, 1991.

[10] P.W. Frey and D.J. Slate. *Letter Recognition Using Holland-Style Adaptive Classifiers.* Machine Learning, 6 (2) 161-182.

[11] K. Fukushima and T. Imagawa. *Recognition and segmentation of connected characters with selective attention.* Neural Networks, 6:33-41, 1993.

[12] L. Goldfarb. *A unified approach to pattern recognition.* Pattern Recognition, 17 (1984) 575-582.

[13] B. Haasdonk. *Feature Space Interpretation of SVMs with non Positive Definite Kernels.* Internal report 1/03, Department of Pattern Recognition and Image Processing, Freiburg University, submitted to a journal, 2003.

[14] T. Joachims. *Learning to Classify Text using Support Vector Machines.* Kluwer, 2002.

[15] M.I. Jordan. *Advanced Topics in Learning & Decision Making.* Course material available at www.cs.berkeley.edu/∼jordan/courses/281B-spring01.

[16] B. Kosko. *Neural Networks and Fuzzy Systems: A Dynamical Approach to Machine Intelligence.* Prentice Hall, 1991.

[17] J.T. Kwok and I.W. Tsang *Learning with Idealized Kernels* Proc. 19th Int Conf Machine Learning, pp. 400-407, 2003.

[18] G.R.G. Lanckriet, N. Cristianini, P. Barlett, L. El Ghaoui and M.I. Jordan. *Learning the kernel matrix with semi-definite programming.* Journal of Machine Learning Research, 5, pp. 27-72, 2004.

[19] O.L. Mangasarian and W.H. Wolberg. *Cancer diagnosis via linear programming.* SIAM News, Volume 23, Numer 5,1990,1-18.

[20] A. Muñoz. *Compound key word generation from document databases using a hierarchical clustering ART model.* Intelligent Data Analysis, vol. 1, pp. 25-48, 1997.

[21] A. Muñoz and T. Villagarcía. *Unsupervised neural networks for variable selection with mixed covariates.* Analyse Multidimensionelle des Donnes, CSIA Ceresta, 217-227, 1998.

[22] C.S. Ong, X. Mary, S. Canu, and A. Smola. *Learning with Non-Positive Kernels.* Proc. ICML (2004), Springer, 639-646.

[23] E. Pekalska, P. Paclík and R.P.W. Duin. *A Generalize Kernel Approach to Dissimilarity-based Classification.* JMLR, Special Issue on Kernel Methods 2 (2) (2002) 175-211.

[24] E. Pekalska, R.P.W. Duin, S. Günter and H. Bunke. *On Not Making Dissimilarities Euclidean.* Proc. SSPR and SPR (2004), LNCS 3138, Springer, 1145-1154.

[25] T. Poggio and F. Girosi. *Networks for Approximation and Learning.* Proceedings of the IEEE, 78(10):1481-1497, 1990.

[26] B. Schölkopf, R. Herbrich, A. Smola and R. Williamson. *A Generalized Representer Theorem.* NeuroCOLT2 TR Series, NC2-TR2000-81, 2000.

[27] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K. Müller, G. Rätsch and A. Smola. *Input Space versus Feature Space in Kernel-based Methods.* IEEE Transactions on Neural Networks 10 (5) (1999) 1000-1017.

[28] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola and R.C. Williamson. *Estimating the Support of a High Dimensional Distribution.* Neural Computation, 13(7):1443-1471 , 2001.

[29] B. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London, 1986.

[30] L. Vandenberghe and S. Boyd. *Semidefinite programming.* SIAM Review, 38(1):49-95, 1996.

[31] B. Zielman and W.J. Heiser. *Models for Asymmetric Proximities.* British Journal of Mathematical and Statistical Psychology, 49:127-146, 1996.

List of Figures

Table 1
Classification errors for the two-servers database.

Method	Train error	Test error	Support vectors
AKM	6.8 %	9.7 %	50.2 %
MAKM	4.1 %	6.5 %	16.8 %
SM	4.4 %	6.3 %	22.4 %
AV	3.1 %	6.0 %	8.2 %
Pick-out	1.5 %	6.5 %	4.2 %
SQ	3.5 %	6.3 %	8.9 %
KWS	5.5 %	6.3 %	51.4 %
ProbWS	2.1 %	2.0 %	14.2 %
ExpWS	2.2 %	2.3 %	19.0 %
PolyWS	2.1 %	2.0 %	14.2 %
MaxMin	2.3 %	3.5 %	13.3 %
Percentil-in	2.3 %	3.0 %	4.9 %
Percentil-out	2.2 %	3.0 %	6.7 %
MARK-L	4.0 %	4.5 %	0.8 %

Table 2
Classification errors for the kernels with complementary information, using several methods to solve the problem of building a positive definite matrix (Train error %, Test error %, Support Vectors %). The best solution for each method is marked in bold font.

Method	Adding λI	MDS	PSEUDO	$K^T K$
AV	(6.4 , 6.5 , 35.4)	**(4.4 , 4.3 , 21.5)**	(6.4 , 6.5 , 35.4)	(6.4 , 6.5 , 35.4)
Pick-out	(0.1 , 8.1 , 60.1)	**(4.4 , 4.3 , 21.5)**	(5.9 , 12.0 , 19.4)	(5.9 , 6.9 , 3.7)
SQ	(1.8 , 5.4 , 53.3)	(5.9 , 5.6 , 33.9)	(5.4 , 5.3 , 33.4)	**(3.8 , 4.2 , 15.1)**
KWS	(2.6 , 5.7 , 49.5)	(6.0 , 5.6 , 34.0)	(5.9 , 5.6 , 33.8)	**(3.8 , 4.0 , 15.4)**
ProbWS	(6.4 , 6.5 , 35.4)	(6.4 , 6.5 , 35.4)	(6.4 , 6.5 , 35.4)	**(6.4 , 6.5 , 35.4)**
ExpWS	(6.4 , 6.5 , 35.4)	(6.4 , 6.5 , 35.4)	(4.0 , 4.0 , 34.1)	**(3.6 , 3.8 , 16.7)**
PolyWS	(6.4 , 6.5 , 35.4)	(6.4 , 6.5 , 35.4)	(4.8 , 4.8 , 36.3)	**(4.1 , 4.1 , 19.4)**
MaxMin	(0.4 , 6.4 , 45.3)	**(4.9 , 5.0 , 26.5)**	(4.4 , 5.5 , 24.8)	(4.6 , 5.8 , 9.8)
Percentil-in	(0.0 , 7.8 , 68.4)	**(4.8 , 4.9 , 24.4)**	(5.3 , 6.5 , 22.0)	(4.7 , 6.5 , 8.0)
Percentil-out	(0.5 , 5.4 , 55.0)	**(5.0 , 5.2 , 25.9)**	(6.3 , 7.5 , 25.3)	(5.1 , 6.4 , 9.9)

Table 3
Classification errors for the kernels with complementary information.

Method	Train error	Test error	Support vectors
AKM	6.4 %	6.5 %	35.4 %
MAKM	2.2 %	4.4 %	11.3 %
SM	6.4 %	5.8 %	33.2 %
AV	4.4 %	4.3 %	21.5 %
Pick-out	4.4 %	4.3 %	21.5 %
SQ	1.8 %	5.4 %	52.4 %
KWS	3.8 %	4.0 %	15.4 %
ProbWS	6.4 %	6.5 %	35.4 %
ExpWS	3.6 %	3.8 %	16.7 %
PolyWS	4.1 %	4.1 %	19.4 %
MaxMin	4.9 %	5.1 %	26.5 %
Percentil-in	4.8 %	4.9 %	24.4 %
Percentil-out	5.0 %	5.2 %	25.9 %
k**-NN**	3.4 %	4.0 %	
MARK-L	8.1 %	8.4 %	1.0 %
MARS	3.4 %	3.9 %	
LDA	7.5 %	7.8 %	
LR	7.6 %	7.7 %	
SDP	12.0 %	12.9 %	43.3 %
SVM	3.6 %	3.9 %	26.5 %

Table 4
Classification errors for the cancer data.

Method	Train error	Test error	Support vectors
K_1:**Polynomial**	0.1 %	7.8 %	8.3 %
K_2:**RBF**	0.0 %	10.8 %	65.6 %
K_3:**Linear**	2.6 %	3.7 %	7.1 %
AKM	1.3 %	3.3 %	31.1 %
MAKM	1.1 %	2.8 %	31.1 %
SM	1.2 %	3.1 %	35.3 %
AV	2.1 %	3.1 %	2.9 %
Pick-out	2.4 %	3.2 %	5.9 %
SQ	1.3 %	3.0 %	40.6 %
KWS	0.6 %	3.1 %	72.7 %
ProbWS	2.2 %	2.9 %	37.8 %
ExpWS	2.3 %	2.9 %	25.3 %
PolyWS	2.0 %	2.9 %	34.2 %
MaxMin	0.7 %	2.9 %	25.3 %
Percentil-in	1.8 %	3.4 %	59.1 %
Percentil-out	0.0 %	3.1 %	55.4 %
k-**NN**	2.7 %	3.4 %	
MARK-L	2.0 %	4.2 %	9.2 %
MARS	2.9 %	3.1 %	
LDA	3.8 %	3.9 %	
LR	13.2 %	13.0 %	
SDP	0.0 %	6.2 %	65.5 %
SVM	0.1 %	4.2 %	49.2 %

Table 5
Classification errors for the cancer data using a battery of RBF kernels.

Method	Train error	Test error	Support vectors
Best RBF	2.3 %	3.1 %	13.6 %
Worst RBF	0.0 %	24.7 %	74.0 %
AKM	1.6 %	3.4 %	21.7 %
MAKM	1.6 %	3.3 %	21.7 %
SM	1.6 %	3.3 %	22.2 %
AV	3.1 %	3.1 %	6.5 %
Pick-out	2.5 %	3.4 %	7.7 %
SQ	3.0 %	3.1 %	10.2 %
KWS	2.9 %	3.1 %	9.3 %
ProbWS	3.0 %	3.1 %	10.5 %
ExpWS	3.0 %	3.1 %	10.5 %
PolyWS	3.0 %	3.1 %	10.5 %
MaxMin	0.1 %	2.8 %	14.2 %
Percentil-in	1.9 %	2.8 %	7.8 %
Percentil-out	0.2 %	2.8 %	19.2 %
MARK-L	0.0 %	3.6 %	18.3 %
SDP	0.0 %	3.2 %	41.5 %

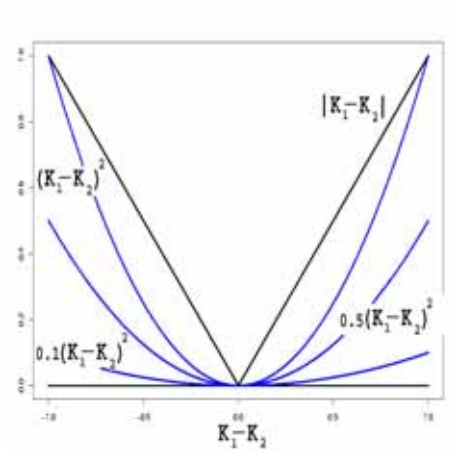

Fig. 1. Different choices of $g(K_1 - K_2)$ for different values of τ. The straight lines correspond to the AV method. The curves correspond to the SQ method.

Table 6
Classification errors for the handwritten digit data set.

Method	**Train error**	**Test error**	**Support vectors**
4×4	0.0 %	3.6 %	3.6 %
Frey-Slate	5.5 %	11.1 %	9.8 %
Fukushima	0.0 %	4.5 %	7.0 %
AKM	0.0 %	4.5 %	6.1 %
MAKM	0.0 %	4.5 %	5.9 %
SM	2.5 %	1.7 %	8.8 %
AV	0.1 %	1.4 %	48.8 %
Pick-out	0.0 %	3.1 %	38.0 %
SQ	0.0 %	3.1 %	29.4 %
KWS	0.0 %	3.1 %	32.8 %
ProbWS	0.0 %	4.5 %	6.1 %
ExpWS	0.0 %	3.6 %	67.9 %
PolyWS	0.0 %	3.6 %	68.1 %
MaxMin	0.0 %	1.9 %	34.2 %
Percentil-in	0.0 %	1.1 %	35.1 %
Percentil-out	0.0 %	1.7 %	32.8 %
k**-NN**	0.0 %	0.6 %	
MARK-L	0.0 %	4.2 %	13.0 %
MARS	0.1 %	3.9 %	
LDA	0.4 %	5.0 %	
LR	0.0 %	3.6 %	
SDP	0.0 %	3.6 %	6.2 %
SVM	0.0 %	3.6 %	60.9 %

Table 7
Classification errors for the term data base. '–' indicates non convergence of the method.

Method	Train error	Test error	Support vectors
AKM	0.0 %	1.4 %	13.4 %
MAKM	0.0 %	1.4 %	10.2 %
SM	0.0 %	1.4 %	13.2 %
AV	0.0 %	0.8 %	8.0 %
Pick-out	0.1 %	1.4 %	6.0 %
SQ	0.0 %	1.2 %	10.1 %
KWS	0.0 %	1.4 %	13.4 %
ProbWS	0.0 %	1.4 %	13.4 %
ExpWS	0.0 %	1.4 %	13.4 %
PolyWS	0.0 %	1.4 %	13.4 %
MaxMin	0.0 %	1.2 %	6.9 %
Percentil-in	0.1 %	1.1 %	5.9 %
Percentil-out	0.0 %	1.3 %	6.9 %
k**-NN**	12.8 %	14.0 %	
MARK-L	– %	– %	– %
MARS	– %	– %	– %
LDA	0.0 %	31.4 %	
LR	– %	– %	
SDP	0.0 %	1.4 %	13.4 %
SVM	23.8 %	23.9 %	63.2 %

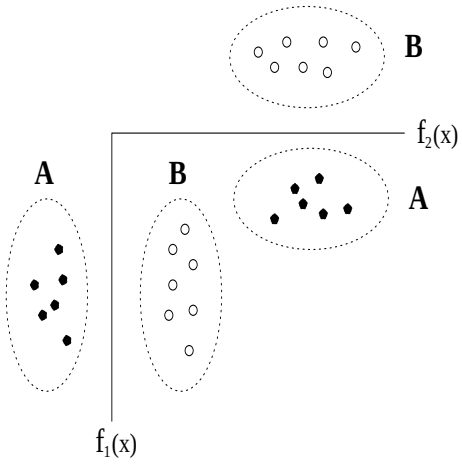

Fig. 2. Example of non-separable groups using linear combinations of kernels.

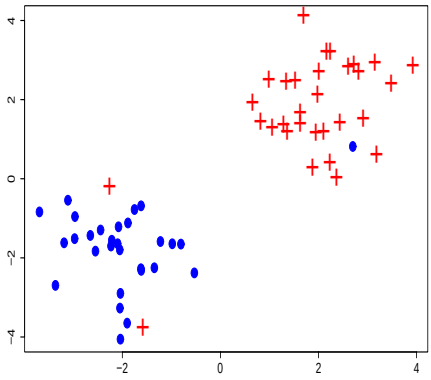

Fig. 3. MDS for a single kernel.

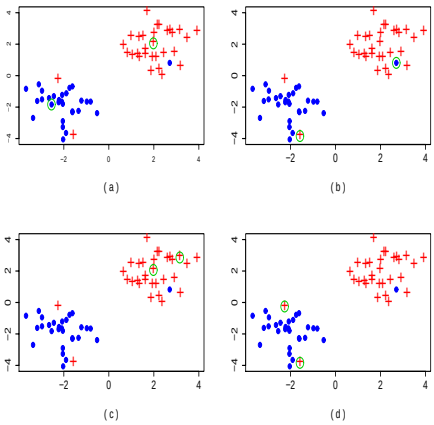

Fig. 4. Possible situations: the considered pair of points in each situation are marked
.

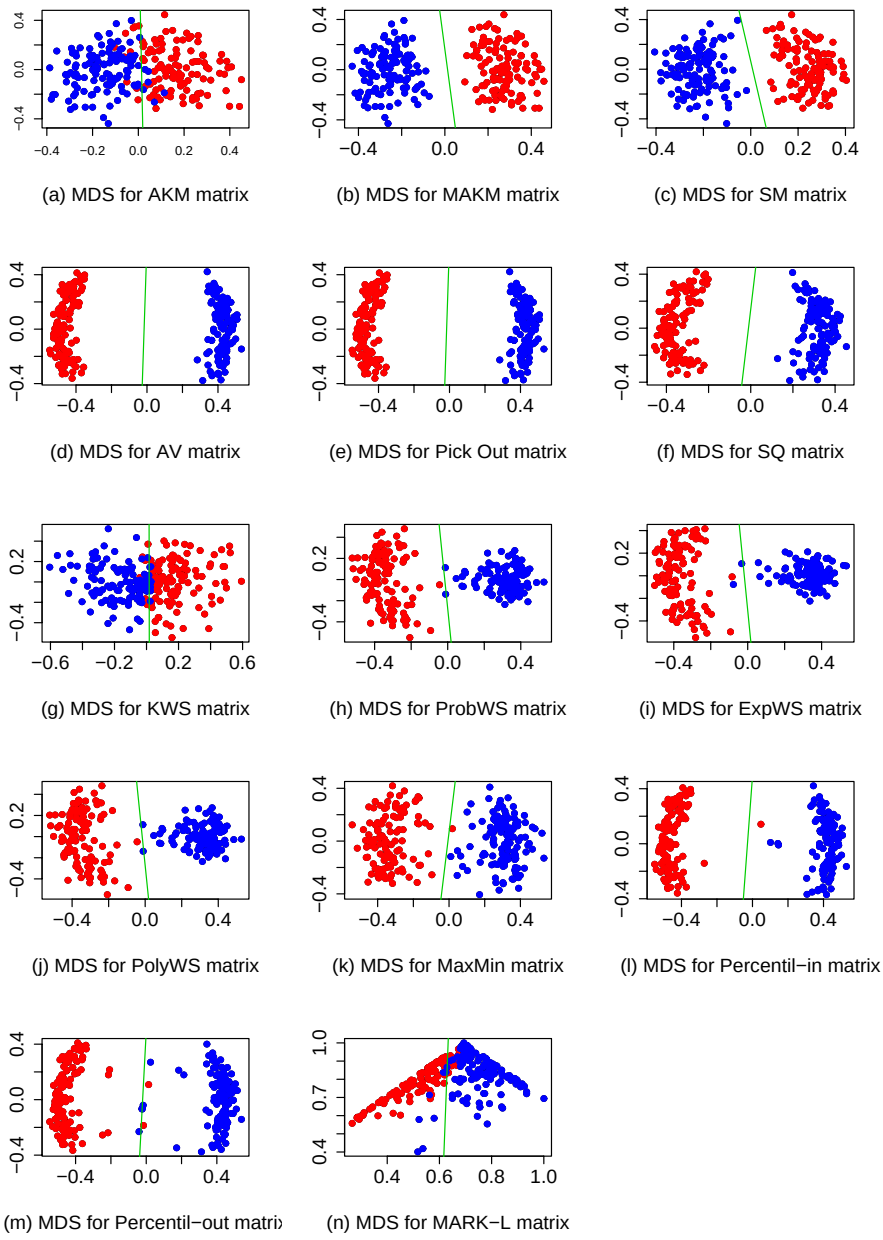

Fig. 5. Multidimensional scaling (MDS) representation of the output kernels and SVM hyperplane for the two-servers data base.