Journal of Machine Learning Research 13 (2012) 795-828 Submitted 1/11; Revised 10/11; Published 3/12

Algorithms for Learning Kernels Based on Centered Alignment

Corinna Cortes CORTES@GOOGLE.COM
Google Research
76 Ninth Avenue
New York, NY 10011

Mehryar Mohri MOHRI@CIMS.NYU.EDU
Courant Institute and Google Research
251 Mercer Street
New York, NY 10012

Afshin Rostamizadeh[*] ROSTAMI@GOOGLE.COM
Google Research
76 Ninth Avenue
New York, NY 10011

Editor: Francis Bach

Abstract

This paper presents new and effective algorithms for learning kernels. In particular, as shown by our empirical results, these algorithms consistently outperform the so-called uniform combination solution that has proven to be difficult to improve upon in the past, as well as other algorithms for learning kernels based on convex combinations of base kernels in both classification and regression. Our algorithms are based on the notion of centered alignment which is used as a similarity measure between kernels or kernel matrices. We present a number of novel algorithmic, theoretical, and empirical results for learning kernels based on our notion of centered alignment. In particular, we describe efficient algorithms for learning a maximum alignment kernel by showing that the problem can be reduced to a simple QP and discuss a one-stage algorithm for learning both a kernel and a hypothesis based on that kernel using an alignment-based regularization. Our theoretical results include a novel concentration bound for centered alignment between kernel matrices, the proof of the existence of effective predictors for kernels with high alignment, both for classification and for regression, and the proof of stability-based generalization bounds for a broad family of algorithms for learning kernels based on centered alignment. We also report the results of experiments with our centered alignment-based algorithms in both classification and regression.

Keywords: kernel methods, learning kernels, feature selection

1. Introduction

One of the key steps in the design of learning algorithms is the choice of the features. This choice is typically left to the user and represents his prior knowledge, but it is critical: a poor choice makes learning challenging while a better choice makes it more likely to be successful. The general objective of this work is to define effective methods that partially relieve the user from the requirement of specifying the features.

[*] A significant amount of the presented work was completed while AR was a graduate student at the Courant Institute of Mathematical Sciences and a postdoctoral scholar at the University of Califorinia at Berkeley.

For kernel-based algorithms the features are provided intrinsically via the choice of a positive-definite symmetric kernel function (Boser et al., 1992; Cortes and Vapnik, 1995; Vapnik, 1998). To limit the risk of a poor choice of kernel, in the last decade or so, a number of publications have investigated the idea of *learning the kernel* from data (Cristianini et al., 2001; Chapelle et al., 2002; Bousquet and Herrmann, 2002; Lanckriet et al., 2004; Jebara, 2004; Argyriou et al., 2005; Micchelli and Pontil, 2005; Lewis et al., 2006; Argyriou et al., 2006; Kim et al., 2006; Cortes et al., 2008; Sonnenburg et al., 2006; Srebro and Ben-David, 2006; Zien and Ong, 2007; Cortes et al., 2009a, 2010a,b). This reduces the requirement from the user to only specifying a family of kernels rather than a specific kernel. The task of selecting (or learning) a kernel out of that family is then reserved to the learning algorithm which, as for standard kernel-based methods, must also use the data to choose a hypothesis in the reproducing kernel Hilbert space (RKHS) associated to the kernel selected.

Different kernel families have been studied in the past, but the most widely used one has been that of convex combinations of a finite set of base kernels. However, while different learning kernel algorithms have been introduced in that case, including those of Lanckriet et al. (2004), to our knowledge, in the past, none has succeeded in consistently and significantly outperforming the *uniform combination* solution, in binary classification or regression tasks. The uniform solution consists of simply learning a hypothesis out of the RKHS associated to a uniform combination of the base kernels. This disappointing performance of learning kernel algorithms has been pointed out in different instances, including by many participants at different NIPS workshops organized on the theme in 2008 and 2009, as well as in a survey talk (Cortes, 2009) and tutorial (Cortes et al., 2011b). The empirical results we report further confirm this observation. Other kernel families have been considered in the literature, including hyperkernels (Ong et al., 2005), Gaussian kernel families (Micchelli and Pontil, 2005), or non-linear families (Bach, 2008; Cortes et al., 2009b; Varma and Babu, 2009). However, the performance reported for these other families does not seem to be consistently superior to that of the uniform combination either.

In contrast, on the theoretical side, favorable guarantees have been derived for learning kernels. For general kernel families, learning bounds based on covering numbers were given by Srebro and Ben-David (2006). Stronger margin-based generalization guarantees based on an analysis of the Rademacher complexity, with only a square-root logarithmic dependency on the number of base kernels were given by Cortes et al. (2010b) for convex combinations of kernels with an L_1 constraint. The dependency of theses bounds, as well as others given for L_q constraints, were shown to be optimal with respect to the number of kernels. These L_1 bounds generalize those presented in Koltchinskii and Yuan (2008) in the context of ensembles of kernel machines. The learning guarantees suggest that learning kernel algorithms even with a relatively large number of base kernels could achieve a good performance.

This paper presents new algorithms for learning kernels whose performance is more consistent with expectations based on these theoretical guarantees. In particular, as can be seen by our experimental results, several of the algorithms we describe consistently outperform the uniform combination solution. They also surpass in performance the algorithm of Lanckriet et al. (2004) in classification and improve upon that of Cortes et al. (2009a) in regression. Thus, this can be viewed as the first series of algorithmic solutions for learning kernels in classification and regression with consistent performance improvements.

Our learning kernel algorithms are based on the notion of *centered alignment* which is a similarity measure between kernels or kernel matrices. This can be used to measure the similarity of each base kernel with the target kernel K_Y derived from the output labels. Our definition of centered alignment is close to the uncentered kernel alignment originally introduced by Cristianini et al. (2001). This closeness is only superficial however: as we shall see both from the analysis of several cases and from experimental results, in contrast with our notion of alignment, the uncentered kernel alignment of Cristianini et al. (2001) does not correlate well with performance and thus, in general, cannot be used effectively for learning kernels. We note that other kernel optimization criteria similar to centered alignment, but without the key normalization have been used by some authors (Kim et al., 2006; Gretton et al., 2005). Both the centering and the normalization are critical components of our definition.

We present a number of novel algorithmic, theoretical, and empirical results for learning kernels based on our notion of centered alignment. In Section 2, we introduce and analyze the properties of centered alignment between kernel functions and kernel matrices, and discuss its benefits. In particular, the importance of the centering is justified theoretically and validated empirically. We then describe several algorithms based on the notion of centered alignment in Section 3.

We present two algorithms that each work in two subsequent stages (Sections 3.1 and 3.2): the first stage consists of *learning* a kernel K that is a non-negative linear combination of p base kernels; the second stage combines this kernel with a standard kernel-based learning algorithm such as support vector machines (SVMs) (Cortes and Vapnik, 1995) for classification, or kernel ridge regression (KRR) for regression (Saunders et al., 1998), to select a prediction hypothesis. These two algorithms differ in the way centered alignment is used to learn K. The simplest and most straightforward to implement algorithm selects the weight of each base kernel matrix independently, only from the centered alignment of that matrix with the target kernel matrix. The other more accurate algorithm instead determines these weights jointly by measuring the centered alignment of a convex combination of base kernel matrices with the target one. We show that this more accurate algorithm is very efficient by proving that the base kernel weights can be obtained by solving a simple quadratic program (QP). We also give a closed-form expression for the weights in the case of a linear, but not necessarily convex, combination. Note that an alternative two-stage technique consists of first learning a prediction hypothesis using each base kernel and then learning the best linear combination of these hypotheses. But, as pointed out in Section 3.3, in general, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms. In addition, we present and analyze an algorithm that uses centered alignment to both select a convex combination kernel and a hypothesis based on that kernel, these two tasks being performed in a single stage by solving a single optimization problem (Section 3.4).

We also present an extensive theoretical analysis of the notion of centered alignment and algorithms based on that notion. We prove a concentration bound for the notion of centered alignment showing that the centered alignment of two kernel matrices is sharply concentrated around the centered alignment of the corresponding kernel functions, the difference being bounded by a term in $O(1/\sqrt{m})$ for samples of size m (Section 4.1). Our result is simpler and directly bounds the difference between these two relevant quantities, unlike previous work by Cristianini et al. (2001) (for uncentered alignments). We also show the existence of good predictors for kernels with high centered alignment, both for classification and for regression (Section 4.2). This result justifies the search for good learning kernel algorithms based on the notion of centered alignment. We note that the proofs given for similar results in classification for uncentered alignments by Cristianini et al. (2001, 2002) are erroneous. We also present stability-based generalization bounds for two-stage learning kernel algorithms based on centered alignment when the second stage is kernel ridge regression (Section 4.3). We further study the application of these bounds in the case of our alignment maximization algorithm and initiate a detailed analysis of the stability of this algorithm (Appendix B).

Finally, in Section 5, we report the results of experiments with our centered alignment-based algorithms both in classification and regression, and compare our results with L_1- and L_2-regularized learning kernel algorithms (Lanckriet et al., 2004; Cortes et al., 2009a), as well as with the uniform kernel combination method. The results show an improvement both over the uniform combination and over the one-stage kernel learning algorithms. They also demonstrate a strong correlation between the centered alignment achieved and the performance of the algorithm.[1]

2. Alignment Definitions

The notion of kernel alignment was first introduced by Cristianini et al. (2001). Our definition of kernel alignment is different and is based on the notion of centering in the feature space. Thus, we start with the definition of centering and the analysis of its relevant properties.

2.1 Centered Kernel Functions

Let D be the distribution according to which training and test points are drawn. A feature mapping $\Phi\colon \mathcal{X} \to H$ is centered by subtracting from it its expectation, that is forming it by $\Phi - \mathrm{E}_x[\Phi]$, where E_x denotes the expected value of Φ when x is drawn according to the distribution D. Centering a positive definite symmetric (PDS) kernel function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ consists of centering any feature mapping Φ associated to K. Thus, the centered kernel K_c associated to K is defined for all $x, x' \in \mathcal{X}$ by

$$\begin{aligned} K_c(x,x') &= (\Phi(x) - \mathop{\mathrm{E}}_{x}[\Phi])^\top (\Phi(x') - \mathop{\mathrm{E}}_{x'}[\Phi]) \\ &= K(x,x') - \mathop{\mathrm{E}}_{x}[K(x,x')] - \mathop{\mathrm{E}}_{x'}[K(x,x')] + \mathop{\mathrm{E}}_{x,x'}[K(x,x')]. \end{aligned}$$

This also shows that the definition does not depend on the choice of the feature mapping associated to K. Since $K_c(x,x')$ is defined as an inner product, K_c is also a PDS kernel.[2] Note also that for a centered kernel K_c, $\mathrm{E}_{x,x'}[K_c(x,x')] = 0$, that is, centering the feature mapping implies centering the kernel function.

2.2 Centered Kernel Matrices

Similar definitions can be given for a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D: a feature vector $\Phi(x_i)$ with $i \in [1,m]$ is centered by subtracting from it its empirical expectation, that is forming it with $\Phi(x_i) - \overline{\Phi}$, where $\overline{\Phi} = \frac{1}{m}\sum_{i=1}^m \Phi(x_i)$. The kernel matrix $\mathbf{K}$ associated to K and the sample

[1] This is an extended version of Cortes et al. (2010a) with much additional material, including additional empirical evidence supporting the importance of centered alignment, the description and discussion of a single-stage algorithm for learning kernels based on centered alignment, an analysis of unnormalized centered alignment and the proof of the existence of good predictors for large values of centered alignment, generalization bounds for two-stage learning kernel algorithms based on centered alignment, and an experimental investigation of the single-stage algorithm.

[2] For convenience, we use a matrix notation for feature vectors and use $\Phi(x)^\top \Phi(x')$ to denote the inner product between two feature vectors and similarly $\Phi(x)\Phi(x')^\top$ for the outer product, including in the case where the dimension of the feature space is infinite, in which case we are using infinite matrices.

S is centered by replacing it with $\mathbf{K}_c$ defined for all $i, j \in [1, m]$ by

$$[\mathbf{K}_c]_{ij} = \mathbf{K}_{ij} - \frac{1}{m}\sum_{i=1}^{m}\mathbf{K}_{ij} - \frac{1}{m}\sum_{j=1}^{m}\mathbf{K}_{ij} + \frac{1}{m^2}\sum_{i,j=1}^{m}\mathbf{K}_{ij}. \tag{1}$$

Let $\mathbf{\Phi} = [\Phi(x_1), \ldots, \Phi(x_m)]^\top$ and $\overline{\mathbf{\Phi}} = [\overline{\Phi}, \ldots, \overline{\Phi}]^\top$. Then, it is not hard to verify that $\mathbf{K}_c = (\mathbf{\Phi} - \overline{\mathbf{\Phi}})(\mathbf{\Phi} - \overline{\mathbf{\Phi}})^\top$, which shows that $\mathbf{K}_c$ is a positive semi-definite (PSD) matrix. Also, as with the kernel function, $\frac{1}{m^2}\sum_{i,j=1}^{m}[\mathbf{K}_c]_{ij} = 0$. Let $\langle \cdot, \cdot \rangle_F$ denote the Frobenius product and $\|\cdot\|_F$ the Frobenius norm defined by

$$\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}, \langle \mathbf{A}, \mathbf{B}\rangle_F = \mathrm{Tr}[\mathbf{A}^\top\mathbf{B}] \text{ and } \|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A}\rangle_F}.$$

Then, the following basic properties hold for centering kernel matrices.

Lemma 1 *Let* $\mathbf{1} \in \mathbb{R}^{m\times 1}$ *denote the vector with all entries equal to one, and* $\mathbf{I}$ *the identity matrix.*

1. *For any kernel matrix* $\mathbf{K} \in \mathbb{R}^{m\times m}$*, the centered kernel matrix* $\mathbf{K}_c$ *can be expressed as follows*

$$\mathbf{K}_c = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right].$$

2. *For any two kernel matrices* $\mathbf{K}$ *and* $\mathbf{K}'$*,*

$$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F = \langle \mathbf{K}, \mathbf{K}'_c\rangle_F = \langle \mathbf{K}_c, \mathbf{K}'\rangle_F.$$

Proof The first statement can be shown straightforwardly from the definition of $\mathbf{K}_c$ (Equation (1)). The second statement follows from

$$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F = \mathrm{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}'\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\right],$$

the fact that $[\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]^2 = [\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top]$, and the trace property $\mathrm{Tr}[\mathbf{AB}] = \mathrm{Tr}[\mathbf{BA}]$, valid for all matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$. ■

We shall use these properties in the proofs of the results presented in Section 4.

2.3 Centered Kernel Alignment

In the following sections, in the absence of ambiguity, to abbreviate the notation, we often omit the variables over which an expectation is taken. We define the alignment of two kernel functions as follows.

Definition 2 (Kernel function alignment) *Let* K *and* K' *be two kernel functions defined over* $\mathcal{X} \times \mathcal{X}$ *such that* $0 < \mathrm{E}[K_c^2] < +\infty$ *and* $0 < \mathrm{E}[K_c'^2] < +\infty$*. Then, the* alignment *between* K *and* K' *is defined by*

$$\rho(K, K') = \frac{\mathrm{E}[K_c K'_c]}{\sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[K_c'^2]}}\ .$$

Since $|\,\mathrm{E}[K_c K'_c]| \le \sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[K_c'^2]}$ by the Cauchy-Schwarz inequality, we have $\rho(K, K') \in [-1, 1]$. The following lemma shows more precisely that $\rho(K, K') \in [0, 1]$ when K and K' are PDS kernels.

Lemma 3 *For any two PDS kernels K and K',* $\mathrm{E}[KK'] \geq 0$*.*

Proof Let Φ be a feature mapping associated to K and Φ' a feature mapping associated to K'. By definition of Φ and Φ', and using the properties of the trace, we can write:

$$\begin{aligned}\mathop{\mathrm{E}}_{x,x'}[K(x,x')K'(x,x')] &= \mathop{\mathrm{E}}_{x,x'}[\Phi(x)^\top\Phi(x')\Phi'(x')^\top\Phi'(x)]\\ &= \mathop{\mathrm{E}}_{x,x'}\left[\mathrm{Tr}[\Phi(x)^\top\Phi(x')\Phi'(x')^\top\Phi'(x)]\right]\\ &= \langle \mathop{\mathrm{E}}_{x}[\Phi(x)\Phi'(x)^\top], \mathop{\mathrm{E}}_{x'}[\Phi(x')\Phi'(x')^\top]\rangle_F = \|\mathbf{U}\|_F^2 \geq 0,\end{aligned}$$

where $\mathbf{U} = \mathrm{E}_x[\Phi(x)\Phi'(x)^\top]$. ■

The lemma applies in particular to any two centered kernels K_c and K'_c which, as previously shown, are PDS kernels if K and K' are PDS. Thus, for any two PDS kernels K and K', the following holds:

$$\mathrm{E}[K_cK'_c] \geq 0.$$

We can define similarly the alignment between two kernel matrices $\mathbf{K}$ and $\mathbf{K}'$ based on a finite sample $S = (x_1, \ldots, x_m)$ drawn according to D.

Definition 4 (Kernel matrix alignment) *Let* $\mathbf{K} \in \mathbb{R}^{m\times m}$ *and* $\mathbf{K}' \in \mathbb{R}^{m\times m}$ *be two kernel matrices such that* $\|\mathbf{K}_c\|_F \neq 0$ *and* $\|\mathbf{K}'_c\|_F \neq 0$*. Then, the* alignment *between* $\mathbf{K}$ *and* $\mathbf{K}'$ *is defined by*

$$\widehat{\rho}(\mathbf{K},\mathbf{K}') = \frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{\|\mathbf{K}_c\|_F\|\mathbf{K}'_c\|_F}\ .$$

Here too, by the Cauchy-Schwarz inequality, $\widehat{\rho}(\mathbf{K},\mathbf{K}') \in [-1,1]$ and in fact $\widehat{\rho}(\mathbf{K},\mathbf{K}') \geq 0$ since the Frobenius product of any two positive semi-definite matrices $\mathbf{K}$ and $\mathbf{K}'$ is non-negative. Indeed, for such matrices, there exist matrices $\mathbf{U}$ and $\mathbf{V}$ such that $\mathbf{K} = \mathbf{U}\mathbf{U}^\top$ and $\mathbf{K}' = \mathbf{V}\mathbf{V}^\top$. The statement follows from

$$\langle \mathbf{K},\mathbf{K}'\rangle_F = \mathrm{Tr}(\mathbf{U}\mathbf{U}^\top\mathbf{V}\mathbf{V}^\top) = \mathrm{Tr}\left((\mathbf{U}^\top\mathbf{V})^\top(\mathbf{U}^\top\mathbf{V})\right) = \|\mathbf{U}^\top\mathbf{V}\|_F^2 \geq 0. \quad (2)$$

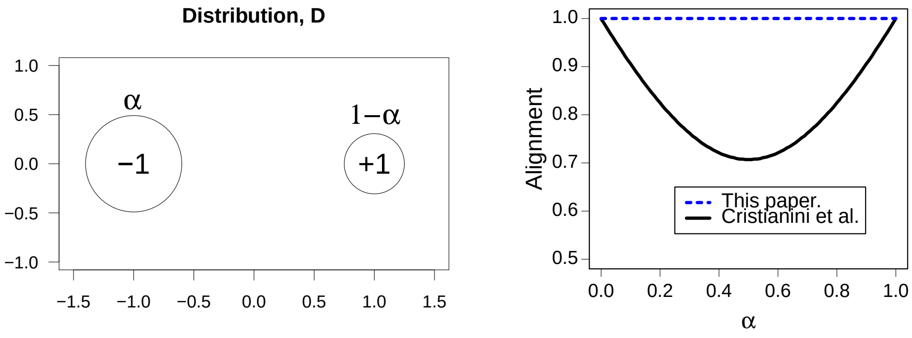
This applies in particular to the kernel matrices of the PDS kernels K_c and K'_c:

$$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F \geq 0.$$

Our definitions of alignment between kernel functions or between kernel matrices differ from those originally given by Cristianini et al. (2001, 2002):

$$A = \frac{\mathrm{E}[KK']}{\sqrt{\mathrm{E}[K^2]\,\mathrm{E}[K'^2]}}, \quad \widehat{A} = \frac{\langle \mathbf{K},\mathbf{K}'\rangle_F}{\|\mathbf{K}\|_F\|\mathbf{K}'\|_F},$$

which are thus in terms of K and K' instead of K_c and K'_c and similarly for matrices. This may appear to be a technicality, but it is in fact a critical difference. Without that centering, the definition of alignment does not correlate well with performance. To see this, consider the standard case where K' is the target label kernel, that is $K'(x,x') = yy'$, with y the label of x and y' the label of x', and examine the following simple example in dimension two ($\mathcal{X} = \mathbb{R}^2$), where $K(x,x') = x\cdot x' + 1$ and



(a) (b)

Figure 1: (a) Representation of the distribution D. In this simple two-dimensional example, a fraction α of the points are at $(-1,0)$ and have the label -1. The remaining points are at $(1,0)$ and have the label $+1$. (b) Alignment values computed for two different definitions of alignment. The solid line in black plots the definition of alignment computed according to Cristianini et al. (2001) $A = (\alpha^2 + (1-\alpha)^2)^{1/2}$, while our definition of centered alignment results in the straight dotted blue line $\rho = 1$.

	KINEMATICS (REGR.)	IONOSPHERE (REGR.)	GERMAN (CLASS.)	SPAMBASE (CLASS.)	SPLICE (CLASS.)
$\widehat{\rho}$	0.9624	0.9979	0.9439	0.9918	0.9515
$\widehat{A}$	0.8627	0.9841	0.9390	0.9889	-0.4484

Table 1: The correlations of the alignment values and error-rates of various kernels. The top row reports the correlation of the accuracy of the base kernels used in Section 5 with the centered alignments $\widehat{\rho}$, the bottom row the correlation with the non-centered alignment $\widehat{A}$.

where the distribution D is defined by a fraction $\alpha \in [0,1]$ of all points being at $(-1,0)$ and labeled with -1, and the remaining points at $(1,0)$ with label $+1$, as shown in Figure 1.

Clearly, for any value of $\alpha \in [0,1]$, the problem is separable, for example with the simple vertical line going through the origin, and one would expect the alignment to be 1. However, the alignment A calculated using the definition of the distribution D admits a different expression. Using

$$\begin{aligned}
&\mathrm{E}[K'^2] = 1, \\
&\mathrm{E}[K^2] = \alpha^2 \cdot 4 + (1-\alpha)^2 \cdot 4 + 2\alpha(1-\alpha) \cdot 0 = 4\big(\alpha^2 + (1-\alpha)^2\big), \\
&\mathrm{E}[KK'] = \alpha^2 \cdot 2 + (1-\alpha)^2 \cdot 2 + 2\alpha(1-\alpha) \cdot 0 = 2\big(\alpha^2 + (1-\alpha)^2\big),
\end{aligned}$$

gives $A = (\alpha^2 + (1-\alpha)^2)^{1/2}$. Thus, A is never equal to one except for $\alpha = 0$ or $\alpha = 1$ and for the balanced case where $\alpha = 1/2$, its value is $A = 1/\sqrt{2} \approx .707 < 1$. In contrast, with our definition, $\rho(K,K') = 1$ for all $\alpha \in [0,1]$ (see Figure 1).

This mismatch between A (or $\widehat{A}$) and the performance values can also be seen in real world data sets. Instances of this problem have been noticed by Meila (2003) and Pothin and Richard

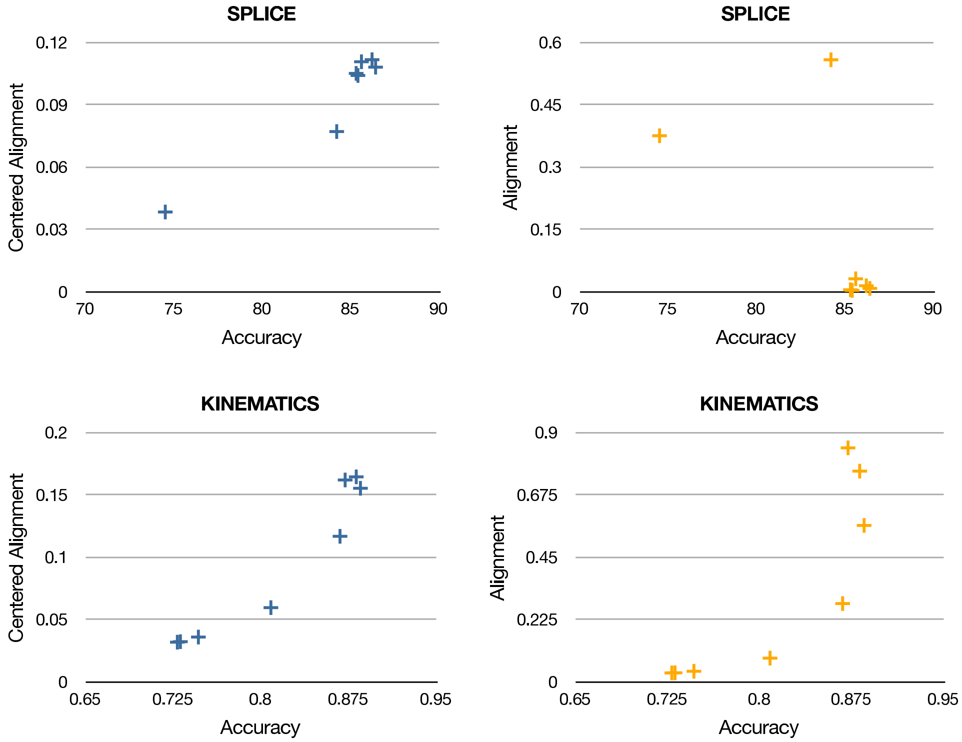


Figure 2: Detailed view of the splice and kinematics experiments presented in Table 1. Both the centered (plots in blue on left) and non-centered alignment (plots in orange on right) are plotted as a function of the accuracy (for the regression problem in the kinematics task "accuracy" is 1 - RMSE). It is apparent from these plots that the non-centered alignment can be misleading when evaluating the quality of a kernel.

(2008) who have suggested various (input) data translation methods, and by Cristianini et al. (2002) who observed an issue for unbalanced data sets. Table 1, as well as Figure 2, give a series of empirical results in several classification and regression tasks based on data sets taken from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`) and Delve data sets (`http://www.cs.toronto.edu/~delve/data/datasets.html`). The table and the figure illustrate the fact that the quantity $\widehat{A}$ measured with respect to several different kernels does not always correlate well with the performance achieved by each kernel. In fact, for the splice classification task, the non-centered alignment is negatively correlated with the accuracy, while a large positive correlation is expected of a good quality measure. The centered notion of alignment $\widehat{\rho}$ however, shows good correlation along all data sets and is always better correlated than $\widehat{A}$.

The notion of alignment seeks to capture the correlation between the random variables $K(x,x')$ and $K'(x,x')$ and one could think it natural, as for the standard correlation coefficients, to consider the following definition:

$$\rho'(K,K') = \frac{\mathrm{E}[(K-\mathrm{E}[K])(K'-\mathrm{E}[K'])]}{\sqrt{\mathrm{E}[(K-\mathrm{E}[K])^2]\,\mathrm{E}[(K'-\mathrm{E}[K'])^2]}}\ .$$

However, centering the kernel values, as opposed to centering the feature values, is not directly relevant to linear predictions in feature space, while our definition of alignment ρ is precisely related to that. Also, as already shown in Section 2.1, centering in the feature space implies the centering of the kernel values, since $\mathrm{E}[K_c] = 0$ and $\frac{1}{m^2}\sum_{i,j=1}^{m}[\mathbf{K}_c]_{ij} = 0$ for any kernel K and kernel matrix $\mathbf{K}$. Conversely, however, centering the kernel does not imply centering in feature space. For example, consider any kernel where all the row marginals are not all equal.

3. Algorithms

This section discusses several learning kernel algorithms based on the notion of centered alignment. In all cases, the family of kernels considered is that of non-negative combinations of p base kernels K_k, $k \in [1, p]$. Thus, the final hypothesis learned belongs to the reproducing kernel Hilbert space (RKHS) $\mathbb{H}_{K_{\boldsymbol{\mu}}}$ associated to a kernel of the form $K_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k K_k$, with $\boldsymbol{\mu} \geq 0$, which guarantees that $K_{\boldsymbol{\mu}}$ is PDS, and $\|\boldsymbol{\mu}\| = \Lambda \geq 0$, for some regularization parameter Λ.

We first describe and analyze two algorithms that both work in two stages: in the first stage, these algorithms determine the mixture weights $\boldsymbol{\mu}$. In the second stage, they train a standard kernel-based algorithm, for example, SVMs for classification, or KRR for regression, in combination with the kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$ associated to $K_{\boldsymbol{\mu}}$, to learn a hypothesis $h \in \mathbb{H}_{K_{\boldsymbol{\mu}}}$. Thus, these *two-stage algorithms* differ only by their first stage, which determines $K_{\boldsymbol{\mu}}$. We describe first in Section 3.1 a simple algorithm that determines each mixture weight μ_k independently, (`align`), then, in Section 3.2, an algorithm that determines the weights μ_ks jointly (`alignf`) by selecting $\boldsymbol{\mu}$ to maximize the alignment with the target kernel. We briefly discuss in Section 3.3 the relationship of such two-stage learning algorithms with algorithms based on ensemble techniques, which also consist of two stages. Finally, we introduce and analyze a *single-stage alignment-based algorithm* which learns $\boldsymbol{\mu}$ and the hypothesis $h \in \mathbb{H}_{K_{\boldsymbol{\mu}}}$ simultaneously in Section 3.4.

3.1 Independent Alignment-based Algorithm (`align`)

This is a simple but efficient method which consists of using the training sample to independently compute the alignment between each kernel matrix $\mathbf{K}_k$ and the target kernel matrix $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$, based on the labels $\mathbf{y}$, and to choose each mixture weight μ_k proportional to that alignment. Thus, the resulting kernel matrix is defined by:

$$\mathbf{K}_{\boldsymbol{\mu}} \propto \sum_{k=1}^{p} \widehat{\rho}(\mathbf{K}_k, \mathbf{K}_Y)\mathbf{K}_k = \frac{1}{\|\mathbf{K}_Y\|_F} \sum_{k=1}^{p} \frac{\langle \mathbf{K}_k, \mathbf{K}_Y \rangle_F}{\|\mathbf{K}_k\|_F} \mathbf{K}_k. \tag{3}$$

When the base kernel matrices $\mathbf{K}_k$ have been normalized with respect to the Frobenius norm, the independent alignment-based algorithm can also be viewed as the solution of a joint maximization of the unnormalized alignment defined as follows, with a L_2-norm constraint on the norm of $\boldsymbol{\mu}$.

Definition 5 (Unnormalized alignment) *Let K and K' be two PDS kernels defined over $\mathcal{X} \times \mathcal{X}$ and $\mathbf{K}$ and $\mathbf{K}'$ their kernel matrices for a sample of size m. Then, the* unnormalized alignment $\rho_{\mathrm{u}}(K, K')$ *between K and K' and the* unnormalized alignment $\widehat{\rho_{\mathrm{u}}}(\mathbf{K}, \mathbf{K}')$ *between $\mathbf{K}$ and $\mathbf{K}'$ are defined by*

$$\rho_{\mathrm{u}}(K, K') = \mathop{\mathrm{E}}_{x,x'}[K_c(x, x')K'_c(x, x')] \quad \textit{and} \quad \widehat{\rho_{\mathrm{u}}}(\mathbf{K}, \mathbf{K}') = \frac{1}{m^2}\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F.$$

Since they are not normalized, the alignment values a and $\widehat{a}$ are no longer guaranteed to be in the interval $[0,1]$. However, assuming the kernel function K and labels are bounded, the unnormalized alignment between K and K_Y are bounded as well.

Lemma 6 *Let K be a PDS kernel. Assume that for all $x \in \mathcal{X}$, $K_c(x,x) \leq R^2$ and for all output label y, $|y| \leq M$. Then, the following bounds hold:*

$$0 \leq \rho_{\mathrm{u}}(K, K_Y) \leq MR^2 \quad \textit{and} \quad 0 \leq \widehat{\rho}_{\mathrm{u}}(\mathbf{K}, \mathbf{K}_Y) \leq MR^2.$$

Proof The lower bounds hold by Lemma 3 and Inequality (2). The upper bounds can be obtained straightforwardly via the application of the Cauchy-Schwarz inequality:

$$\rho_{\mathrm{u}}^2(K, K_Y) = \underset{(x,y),(x',y')}{\mathrm{E}}[K_c(x,x')yy']^2 \leq \underset{x,x'}{\mathrm{E}}[K_c^2(x,x')] \underset{y,y'}{\mathrm{E}}[yy']^2 \leq R^4M^2$$
$$\widehat{\rho}_{\mathrm{u}}(\mathbf{K}, \mathbf{K}') = \frac{1}{m^2}\langle \mathbf{K}_c, \mathbf{K}_Y\rangle_F \leq \frac{1}{m^2}\|\mathbf{K}_c\|_F\|\mathbf{K}_y\|_F \leq \frac{mR^2mM}{m^2} = R^2M,$$

where we used the identity $\langle \mathbf{K}_c, \mathbf{K}_{Y_c}\rangle_F = \langle \mathbf{K}_c, \mathbf{K}_Y\rangle_F$ from Lemma 1. ■

We will consider more generally the corresponding optimization with an L_q-norm constraint on $\boldsymbol{\mu}$ with $q > 1$:

$$\max_{\boldsymbol{\mu}} \ \widehat{\rho}_{\mathrm{u}}\Big(\sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y\Big) = \Big\langle \sum_{k=1}^{p} \mu_k \mathbf{K}_k, \mathbf{K}_Y \Big\rangle_F \tag{4}$$
$$\text{subject to:} \ \sum_{k=1}^{p} \mu_k^q \leq \Lambda.$$

An explicit constraint enforcing $\boldsymbol{\mu} \geq \mathbf{0}$ is not necessary since, as we shall see, the optimal solution found always satisfies this constraint.

Proposition 7 *Let $\boldsymbol{\mu}^*$ be the solution of the optimization problem* (4)*, then* $\mu_k^* \propto \langle \mathbf{K}_k, \mathbf{K}_Y\rangle_F^{\frac{1}{q-1}}$.

Proof The Lagrangian corresponding to the optimization (4) is defined as follows,

$$L(\boldsymbol{\mu}, \beta) = -\sum_{k=1}^{p} \mu_k \langle \mathbf{K}_k, \mathbf{K}_Y\rangle_F + \beta(\sum_{k=1}^{p} \mu_k^q - \Lambda),$$

where the dual variable β is non-negative. Differentiating with respect to μ_k and setting the result to zero gives

$$\frac{\partial L}{\partial \mu_k} = -\langle \mathbf{K}_k, \mathbf{K}_Y\rangle_F + q\beta\mu_k^{q-1} = 0 \implies \mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y\rangle_F^{\frac{1}{q-1}},$$

which concludes the proof. ■

Thus, for $q = 2$, $\mu_k \propto \langle \mathbf{K}_k, \mathbf{K}_Y\rangle_F$ is exactly the solution given by Equation (3) modulo normalization by the Frobenius norm of the base matrix. Note that for $q = 1$, the optimization becomes trivial and can be solved by simply placing all the weight on μ_k with the largest coefficient, that is the μ_k whose corresponding kernel matrix $\mathbf{K}_k$ has the largest alignment with the target kernel.

3.2 Alignment Maximization Algorithm

The independent alignment-based method ignores the correlation between the base kernel matrices. The alignment maximization method takes these correlations into account. It determines the mixture weights μ_k jointly by seeking to maximize the alignment between the convex combination kernel $\mathbf{K}_{\boldsymbol{\mu}} = \sum_{k=1}^{p} \mu_k \mathbf{K}_k$ and the target kernel $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$.

This was also suggested in the case of uncentered alignment by Cristianini et al. (2001); Kandola et al. (2002a) and later studied by Lanckriet et al. (2004) who showed that the problem can be solved as a QCQP (however, as already discussed in Section 2.1, the uncentered alignment is not well correlated with performance). In what follows, we present even more efficient algorithms for computing the weights μ_k by showing that the problem can be reduced to a simple QP. We start by examining the case of a non-convex linear combination where the components of $\boldsymbol{\mu}$ can be negative, and show that the problem admits a closed-form solution in that case. We then partially use that solution to obtain the solution of the convex combination.

3.2.1 LINEAR COMBINATION

We can assume without loss of generality that the centered base kernel matrices $\mathbf{K}_{kc}$ are independent, that is, no linear combination is equal to the zero matrix, otherwise we can select an independent subset. This condition ensures that $\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F > 0$ for arbitrary $\boldsymbol{\mu}$ and that $\widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top)$ is well defined (Definition 4). By Lemma 1, $\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_{Yc}\rangle_F = \langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_Y\rangle_F$. Thus, since $\|\mathbf{K}_{Yc}\|_F$ does not depend on $\boldsymbol{\mu}$, the alignment maximization problem $\max_{\boldsymbol{\mu}\in\mathcal{M}} \widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top)$ can be equivalently written as the following optimization problem:

$$\max_{\boldsymbol{\mu}\in\mathcal{M}} \frac{\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{y}\mathbf{y}^\top\rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F}, \tag{5}$$

where $\mathcal{M} = \{\boldsymbol{\mu} \colon \|\boldsymbol{\mu}\|_2 = 1\}$. A similar set can be defined via the L_1-norm instead of L_2. As we shall see, however, the direction of the solution $\boldsymbol{\mu}^\star$ does not change with respect to the choice of norm. Thus, the problem can be solved in the same way in both cases and subsequently scaled appropriately. Note that, by Lemma 1, $\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \mathbf{K}_{\boldsymbol{\mu}} \mathbf{U}_m$ with $\mathbf{U}_m = \mathbf{I} - \mathbf{1}\mathbf{1}^\top/m$, thus,

$$\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \Big(\sum_{k=1}^{p} \mu_k \mathbf{K}_k\Big) \mathbf{U}_m = \sum_{k=1}^{p} \mu_k \mathbf{U}_m \mathbf{K}_k \mathbf{U}_m = \sum_{k=1}^{p} \mu_k \mathbf{K}_{kc}.$$

Let

$$\mathbf{a} = (\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top\rangle_F, \ldots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top\rangle_F)^\top,$$

and let $\mathbf{M}$ denote the matrix defined by

$$\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc}\rangle_F,$$

for $k, l \in [1, p]$. Note that, in view of the non-negativity of the Frobenius product of symmetric PSD matrices shown in Section 2.3, the entries of $\mathbf{a}$ and $\mathbf{M}$ are all non-negative. Observe also that $\mathbf{M}$ is a symmetric PSD matrix since for any vector $\mathbf{X} = (x_1, \ldots, x_p)^\top \in \mathbb{R}^p$,

$$\begin{aligned}\mathbf{X}^\top\mathbf{M}\mathbf{X} &= \sum_{k,l=1}^{p} x_k x_l \mathbf{M}_{kl}\\ &= \operatorname{Tr}\Big[\sum_{k,l=1}^{p} x_k x_l \mathbf{K}_{kc}\mathbf{K}_{lc}\Big]\\ &= \operatorname{Tr}\Big[(\sum_{k=1}^{p} x_k \mathbf{K}_{kc})(\sum_{l=1}^{p} x_l \mathbf{K}_{lc})\Big] = \|\sum_{k=1}^{p} x_k \mathbf{K}_{kc}\|_F^2 > 0.\end{aligned}$$

The strict inequality follows from the fact that the base kernels are linearly independent. Since this inequality holds for any non-zero $\mathbf{X}$, it also shows that $\mathbf{M}$ is invertible.

Proposition 8 *The solution $\boldsymbol{\mu}^\star$ of the optimization problem (5) is given by $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$.*

Proof With the notation introduced, problem (5) can be rewritten as $\boldsymbol{\mu}^\star = \operatorname{argmax}_{\|\boldsymbol{\mu}\|_2=1} \frac{\boldsymbol{\mu}^\top\mathbf{a}}{\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}}$. Thus, clearly, the solution must verify $\boldsymbol{\mu}^{\star\top}\mathbf{a} \geq 0$. We will square the objective and yet not enforce this condition since, as we shall see, it will be verified by the solution we find. Therefore, we consider the problem

$$\boldsymbol{\mu}^\star = \operatorname*{argmax}_{\|\boldsymbol{\mu}\|_2=1} \frac{(\boldsymbol{\mu}^\top\mathbf{a})^2}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}} = \operatorname*{argmax}_{\|\boldsymbol{\mu}\|_2=1} \frac{\boldsymbol{\mu}^\top\mathbf{a}\mathbf{a}^\top\boldsymbol{\mu}}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}.$$

In the final equality, we recognize the general Rayleigh quotient. Let $\boldsymbol{\nu} = \mathbf{M}^{1/2}\boldsymbol{\mu}$ and $\boldsymbol{\nu}^\star = \mathbf{M}^{1/2}\boldsymbol{\mu}^\star$, then

$$\boldsymbol{\nu}^\star = \operatorname*{argmax}_{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1} \frac{\boldsymbol{\nu}^\top\big[\mathbf{M}^{-1/2}\mathbf{a}\mathbf{a}^\top\mathbf{M}^{-1/2}\big]\boldsymbol{\nu}}{\boldsymbol{\nu}^\top\boldsymbol{\nu}}.$$

Hence, the solution is

$$\boldsymbol{\nu}^\star = \operatorname*{argmax}_{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1} \frac{\big[\boldsymbol{\nu}^\top\mathbf{M}^{-1/2}\mathbf{a}\big]^2}{\|\boldsymbol{\nu}\|_2^2} = \operatorname*{argmax}_{\|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2=1} \left[\left[\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|}\right]^\top \mathbf{M}^{-1/2}\mathbf{a}\right]^2.$$

Thus, $\boldsymbol{\nu}^\star \in \operatorname{Vec}(\mathbf{M}^{-1/2}\mathbf{a})$ with $\|\mathbf{M}^{-1/2}\boldsymbol{\nu}^\star\|_2 = 1$. This yields immediately $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$, which verifies $\boldsymbol{\mu}^{\star\top}\mathbf{a} = \mathbf{a}^\top\mathbf{M}^{-1}\mathbf{a}/\|\mathbf{M}^{-1}\mathbf{a}\| \geq 0$ since $\mathbf{M}$ and $\mathbf{M}^{-1}$ are PSD. ■

3.2.2 CONVEX COMBINATION (alignf)

In view of the proof of Proposition 8, the alignment maximization problem with the set $\mathcal{M}' = \{\|\boldsymbol{\mu}\|_2 = 1 \wedge \boldsymbol{\mu} \geq \mathbf{0}\}$ can be written as

$$\boldsymbol{\mu}^* = \operatorname*{argmax}_{\boldsymbol{\mu}\in\mathcal{M}'} \frac{\boldsymbol{\mu}^\top\mathbf{a}\mathbf{a}^\top\boldsymbol{\mu}}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}. \tag{6}$$

The following proposition shows that the problem can be reduced to solving a simple QP.

Proposition 9 *Let* $\mathbf{v}^\star$ *be the solution of the following QP:*

$$\min_{\mathbf{v}\geq\mathbf{0}} \mathbf{v}^\top\mathbf{M}\mathbf{v} - 2\mathbf{v}^\top\mathbf{a}. \tag{7}$$

Then, the solution $\boldsymbol{\mu}^*$ *of the alignment maximization problem (6) is given by* $\boldsymbol{\mu}^\star = \mathbf{v}^\star/\|\mathbf{v}^\star\|$.

Proof Note that problem (7) is equivalent to the following one defined over $\boldsymbol{\mu}$ and b

$$\min_{\substack{\boldsymbol{\mu}\geq\mathbf{0},\|\boldsymbol{\mu}\|_2=1\\ b>0}} b^2\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu} - 2b\boldsymbol{\mu}^\top\mathbf{a}, \tag{8}$$

where the relation $\mathbf{v} = b\boldsymbol{\mu}$ can be used to retrieve $\mathbf{v}$. The optimal choice of b as a function of $\boldsymbol{\mu}$ can be found by setting the gradient of the objective function with respect to b to zero, giving the closed-form solution $b^* = \frac{\boldsymbol{\mu}^\top\mathbf{a}}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}$. Plugging this back into (8) results in the following optimization after straightforward simplifications:

$$\min_{\boldsymbol{\mu}\geq\mathbf{0},\|\boldsymbol{\mu}\|_2=1} -\frac{(\boldsymbol{\mu}^\top\mathbf{a})^2}{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}},$$

which is equivalent to (6). This shows that $\mathbf{v}^\star = b^*\boldsymbol{\mu}^*$ where $\boldsymbol{\mu}^*$ is the solution of (6) and concludes the proof. ■

It is not hard to see that this problem is equivalent to solving a hard margin SVM problem, thus, any SVM solver can also be used to solve it. A similar problem with the non-centered definition of alignment is treated by Kandola et al. (2002b), but their optimization solution differs from ours and requires cross-validation.

Also, note that solving this QP problem does not require a matrix inversion of $\mathbf{M}$. In fact, the assumption about the invertibility of matrix $\mathbf{M}$ is not necessary and a maximal alignment solution can be computed using the same optimization as that of Proposition 9 in the non-invertible case. The optimization problem is then not strictly convex however and the alignment solution $\boldsymbol{\mu}$ not unique.

We now further analyze the properties of the solution $\mathbf{v}$ of problem (7). Let $\widehat{\rho}_0(\boldsymbol{\mu})$ denote the partially normalized alignment maximized by (5):

$$\widehat{\rho}_0(\boldsymbol{\mu}) = \|\mathbf{y}\mathbf{y}^\top\|_F^2\,\widehat{\rho}(\boldsymbol{\mu}) = \frac{\langle\mathbf{K}_{\boldsymbol{\mu}c},\mathbf{y}\mathbf{y}^\top\rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}c}\|_F} = \frac{\boldsymbol{\mu}^\top\mathbf{a}}{\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}} = \frac{\langle\boldsymbol{\mu},\mathbf{M}^{-1}\mathbf{a}\rangle_{\mathbf{M}}}{\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}}} = \frac{\langle\boldsymbol{\mu},\mathbf{M}^{-1}\mathbf{a}\rangle_{\mathbf{M}}}{\|\boldsymbol{\mu}\|_{\mathbf{M}}}.$$

The following proposition gives a simple expression for $\widehat{\rho}_0(\boldsymbol{\mu})$.

Proposition 10 *For* $\boldsymbol{\mu} = \mathbf{v}/\|\mathbf{v}\|$, *with* $\mathbf{v}\neq 0$ *solution of the alignment maximization problem* (7), *the following identity holds:*

$$\widehat{\rho}_0(\boldsymbol{\mu}) = \|\mathbf{v}\|_{\mathbf{M}}.$$

Proof Since $\|\mathbf{v}\|_{\mathbf{M}}^2 - 2\mathbf{v}^\top\mathbf{a} = \|\mathbf{v}\|_{\mathbf{M}}^2 - 2\langle\mathbf{v},\mathbf{M}^{-1}\mathbf{a}\rangle_{\mathbf{M}} = \|\mathbf{v}-\mathbf{M}^{-1}\mathbf{a}\|_{\mathbf{M}}^2 - \|\mathbf{M}^{-1}\mathbf{a}\|_{\mathbf{M}}^2$ the optimization problem (7) can be equivalently written as

$$\min_{\mathbf{v}\geq 0} \|\mathbf{v}-\mathbf{M}^{-1}\mathbf{a}\|_{\mathbf{M}}^2.$$

This implies that the solution $\mathbf{v}$ is the $\mathbf{M}$-orthogonal projection of $\mathbf{M}^{-1}\mathbf{a}$ over the convex set $\{\mathbf{v}\colon \mathbf{v} \geq 0\}$. Therefore, $\mathbf{v} - \mathbf{M}^{-1}\mathbf{a}$ is $\mathbf{M}$-orthogonal to $\mathbf{v}$:

$$\langle \mathbf{v}, \mathbf{v} - \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}} = 0 \implies \|\mathbf{v}\|_{\mathbf{M}}^2 = \langle \mathbf{v}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}.$$

Thus,

$$\|\mathbf{v}\|_{\mathbf{M}} = \frac{\langle \mathbf{v}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}}{\|\mathbf{v}\|_{\mathbf{M}}} = \frac{\langle \boldsymbol{\mu}, \mathbf{M}^{-1}\mathbf{a} \rangle_{\mathbf{M}}}{\|\boldsymbol{\mu}\|_{\mathbf{M}}} = \rho(\boldsymbol{\mu}),$$

which concludes the proof. ■

Thus, the proposition gives a straightforward way of computing $\rho_0(\boldsymbol{\mu})$, thereby also $\rho(\boldsymbol{\mu})$, from the $\mathbf{M}$-norm of the solution vector $\mathbf{v}$ that $\boldsymbol{\mu}$ is derived from.

3.3 Relationship with Ensemble Techniques

An alternative two-stage technique for learning with multiple kernels consists of first learning a prediction hypothesis h_k using each kernel K_k, $k \in [1, p]$, and then of learning the best linear combination of these hypotheses: $h = \sum_{k=1}^p \mu_k h_k$. But, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms such as that of Lanckriet et al. (2004). For ensemble techniques, each hypothesis h_k, $k \in [1, p]$, is of the form $h_k = \sum_{i=1}^m \alpha_{ik} K_k(x_i, \cdot)$ for some $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{mk})^\top \in \mathbb{R}^m$ with different constraints $\|\boldsymbol{\alpha}_k\| \leq \Lambda_k$, $\Lambda_k \geq 0$, and the final hypothesis is of the form

$$\sum_{k=1}^p \mu_k h_k = \sum_{k=1}^p \mu_k \sum_{j=1}^m \alpha_{ik} K_k(x_i, \cdot) = \sum_{i=1}^m \sum_{k=1}^p \mu_k \alpha_{ik} K_k(x_i, \cdot).$$

In contrast, the general form of the hypothesis learned using kernel learning algorithms is

$$\sum_{i=1}^m \alpha_i K_{\boldsymbol{\mu}}(x_i, \cdot) = \sum_{i=1}^m \alpha_i \sum_{k=1}^p \mu_k K_k(x_i, \cdot) = \sum_{k=1}^p \sum_{i=1}^m \mu_k \alpha_i K_k(x_i, \cdot),$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^m$ with $\|\boldsymbol{\alpha}\| \leq \Lambda$, $\Lambda \geq 0$. When the coefficients α_{ik} can be decoupled, that is $\alpha_{ik} = \alpha_i \beta_k$ for some β_ks, the two solutions seem to have the same form but they are in fact different since in general the coefficients must obey different constraints (different Λ_ks). Furthermore, the combination weights μ_i are not required to be positive in the ensemble case. We present a more detailed theoretical and empirical comparison of the ensemble and learning kernel techniques elsewhere (Cortes et al., 2011a).

3.4 Single-stage Alignment-based Algorithm

This section analyzes an optimization based on the notion of centered alignment, which can be viewed as the single-stage counterpart of the two-stage algorithm discussed in Sections 3.1 - 3.2.

As in Sections 3.1 and 3.2, we denote by $\mathbf{a}$ the vector $(\langle \mathbf{K}_{1c}, \mathbf{y}\mathbf{y}^\top \rangle_F, \ldots, \langle \mathbf{K}_{p_c}, \mathbf{y}\mathbf{y}^\top \rangle_F)^\top$ and let $\mathbf{M} \in \mathbb{R}^{p \times p}$ be the matrix defined by $\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F$. The optimization is then defined by augmenting standard single-stage learning kernel optimizations with an alignment maximization constraint. Thus, the domain $\mathcal{M}$ of the kernel combination vector $\boldsymbol{\mu}$ is defined by:

$$\mathcal{M} = \{\boldsymbol{\mu}\colon \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \rho(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top) \geq \Omega\},$$

for non-negative parameters Λ and Ω. The alignment constraint $\rho(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{y}\mathbf{y}^\top) \geq \Omega$ can be rewritten as $\Omega\sqrt{\boldsymbol{\mu}^\top \mathbf{M}\boldsymbol{\mu}} - \boldsymbol{\mu}^\top \mathbf{a} \leq 0$, which defines a convex region. Thus, $\mathcal{M}$ is a convex subset of $\mathbb{R}^p$.

For a fixed $\boldsymbol{\mu} \in \mathcal{M}$ and corresponding kernel matrix $\mathbf{K}_{\boldsymbol{\mu}}$, let $F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ denote the objective function of the dual optimization problem minimize$_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha})$ solved by an algorithm such as SVM, KRR, or more generally any other algorithm for which $\mathcal{A}$ is a convex set and $F(\boldsymbol{\mu}, \cdot)$ a concave function for all $\boldsymbol{\mu} \in \mathcal{M}$, and $F(\cdot, \boldsymbol{\alpha})$ convex for all $\boldsymbol{\alpha} \in \mathcal{A}$. Then, the general form of a single-stage alignment-based learning kernel optimization is

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}).$$

Note that, by the convex-concave properties of F and the convexity of $\mathcal{M}$ and $\mathcal{A}$, von Neumann's minimax theorem applies:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} F(\boldsymbol{\mu}, \boldsymbol{\alpha}).$$

We now further examine this optimization problem in the specific case of the kernel ridge regression algorithm. In the case of KRR, $F(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -\boldsymbol{\alpha}^\top(\mathbf{K}_{\boldsymbol{\mu}} + \lambda\mathbf{I})\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y}$. Thus, the max-min problem can be rewritten as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\mu}\in\mathcal{M}} -\boldsymbol{\alpha}^\top(\mathbf{K}_{\boldsymbol{\mu}} + \lambda\mathbf{I})\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y}.$$

Let $\mathbf{b}_{\boldsymbol{\alpha}}$ denote the vector $(\boldsymbol{\alpha}^\top\mathbf{K}_1\boldsymbol{\alpha}, \ldots, \boldsymbol{\alpha}^\top\mathbf{K}_p\boldsymbol{\alpha})^\top$, then the problem can be rewritten as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y} - \max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\mathbf{b}_{\boldsymbol{\alpha}},$$

where $\lambda = \lambda_0 m$ in the notation of Equation (10). We first focus on analyzing only the term $-\max_{\boldsymbol{\mu}\in\mathcal{M}} \boldsymbol{\mu}^\top\mathbf{b}_{\boldsymbol{\alpha}}$. Since the last constraint in $\mathcal{M}$ is convex, standard Lagrange multiplier theory guarantees that for any Ω there exists a $\gamma \geq 0$ such that the following optimization is equivalent to the original maximization over $\boldsymbol{\mu}$.

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \; -\boldsymbol{\mu}^\top\mathbf{b}_{\boldsymbol{\alpha}} + \gamma(\Omega\sqrt{\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}} - \boldsymbol{\mu}^\top\mathbf{a}) \\ &\text{subject to } \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \gamma \geq 0. \end{aligned}$$

Note that γ is not a variable, but rather a parameter that will be hand-tuned. Now, again applying standard Lagrange multiplier theory we have that for any $(\gamma\Omega) \geq 0$ there exists an Ω' such that the following optimization is equivalent:

$$\begin{aligned} &\min \; -\boldsymbol{\mu}^\top(\gamma\mathbf{a} + \mathbf{b}_{\boldsymbol{\alpha}}) \\ &\text{subject to } \boldsymbol{\mu} \geq \mathbf{0} \wedge \|\boldsymbol{\mu}\| \leq \Lambda \wedge \gamma \geq \mathbf{0} \wedge \boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu} \leq \Omega'^2. \end{aligned}$$

Applying the Lagrange technique a final time (for any Λ there exists a $\gamma' \geq 0$ and for any Ω'^2 there exists a $\gamma'' \geq 0$) leads to

$$\begin{aligned} &\min \; -\boldsymbol{\mu}^\top(\gamma\mathbf{a} + \mathbf{b}_{\boldsymbol{\alpha}}) + \gamma'\boldsymbol{\mu}^\top\boldsymbol{\mu} + \gamma''\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu} \\ &\text{subject to } \boldsymbol{\mu} \geq \mathbf{0} \wedge \gamma, \gamma', \gamma'' \geq 0. \end{aligned}$$

This is a simple QP problem. Note that the overall problem can now be written as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A},\boldsymbol{\mu}\geq\mathbf{0}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}+2\boldsymbol{\alpha}^\top\mathbf{y}+\boldsymbol{\mu}^\top(\gamma\mathbf{a}+\mathbf{b}_{\boldsymbol{\alpha}})-\gamma'\boldsymbol{\mu}^\top\boldsymbol{\mu}-\gamma''\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu}.$$

This last problem is not convex in $(\boldsymbol{\alpha},\boldsymbol{\mu})$, but the problem is convex in each variable. In the case of kernel ridge regression, the maximization in $\boldsymbol{\alpha}$ admits a closed form solution. Plugging in that solution yields the following convex optimization problem in $\boldsymbol{\mu}$:

$$\min_{\boldsymbol{\mu}\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\lambda\mathbf{I})^{-1}\mathbf{y}-\gamma\boldsymbol{\mu}^\top\mathbf{a}+\boldsymbol{\mu}^\top(\gamma''\mathbf{M}+\gamma'\mathbf{I})\boldsymbol{\mu}.$$

Note that multiplying the objective by λ using the substitution $\boldsymbol{\mu}'=\frac{1}{\lambda}\boldsymbol{\mu}$ results in the following equivalent problem,

$$\min_{\boldsymbol{\mu}'\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}'}+\mathbf{I})^{-1}\mathbf{y}-\lambda^2\gamma\boldsymbol{\mu}'^\top\mathbf{a}+\boldsymbol{\mu}'^\top(\lambda^3\gamma''\mathbf{M}+\lambda^3\gamma'\mathbf{I})\boldsymbol{\mu}',$$

which makes clear that the trade-off parameter λ can be subsumed by the γ,γ' and γ'' parameters. This leads to the following simpler problem with a reduced number of trade-off parameters,

$$\min_{\boldsymbol{\mu}\geq\mathbf{0}} \mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\mathbf{I})^{-1}\mathbf{y}-\gamma\boldsymbol{\mu}^\top\mathbf{a}+\boldsymbol{\mu}^\top(\gamma''\mathbf{M}+\gamma'\mathbf{I})\boldsymbol{\mu}. \tag{9}$$

This is a convex optimization problem. In particular, $\boldsymbol{\mu}\mapsto\mathbf{y}^\top(\mathbf{K}_{\boldsymbol{\mu}}+\mathbf{I})^{-1}\mathbf{y}$ is a convex funtion by convexity of $f\colon\mathbf{M}\mapsto\mathbf{y}^\top\mathbf{M}^{-1}\mathbf{y}$ over the set of positive definite symmetric matrices. The convexity of f can be seen from that of its epigraph, which, by the property of the Schur complement, can be written as follows (Boyd and Vandenberghe, 2004):

$$\operatorname{epi} f=\{(\mathbf{M},t)\colon\mathbf{M}\succ\mathbf{0},\mathbf{y}^\top\mathbf{M}^{-1}\mathbf{y}\leq t\}=\left\{(\mathbf{M},t)\colon\begin{pmatrix}\mathbf{M}&\mathbf{y}\\\mathbf{y}^\top&t\end{pmatrix}\succeq\mathbf{0},\mathbf{M}\succ\mathbf{0}\right\}.$$

This defines a linear matrix inequality in $(\mathbf{M},t)$ and thus a convex set. The convex optimization (9) can be solved efficiently using a simple iterative algorithm as in Cortes et al. (2009a). In practice, the algorithm converges within 10-50 iterations.

4. Theoretical Results

This section presents a series of theoretical guarantees related to the notion of kernel alignment. Section 4.1 proves a concentration bound of the form $|\rho-\widehat{\rho}|\leq O(1/\sqrt{m})$, which relates the centered alignment ρ to its empirical estimate $\widehat{\rho}$. In Section 4.2, we prove the existence of accurate predictors in both classification and regression in the presence of a kernel K with good alignment with respect to the target kernel. Section 4.3 presents stability-based generalization bounds for the two-stage alignment maximization algorithm whose first stage was described in Section 3.2.2.

4.1 Concentration Bounds for Centered Alignment

Our concentration bound differs from that of Cristianini et al. (2001) both because our definition of alignment is different and because we give a bound directly on the quantity of interest $|\rho-\widehat{\rho}|$. Instead, Cristianini et al. (2001) give a bound on $|A'-\widehat{A}|$, where $A'\neq A$ can be related to A by replacing each Frobenius product with its expectation over samples of size m.

The following proposition gives a bound on the essential quantities appearing in the definition of the alignments.

Proposition 11 *Let* $\mathbf{K}$ *and* $\mathbf{K}'$ *denote kernel matrices associated to the kernel functions* K *and* K' *for a sample of size* m *drawn according to* D*. Assume that for any* $x \in \mathcal{X}$, $K(x,x) \leq R^2$ *and* $K'(x,x) \leq R'^2$*. Then, for any* $\delta > 0$*, with probability at least* $1-\delta$*, the following inequality holds:*

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \frac{18R^2R'^2}{m} + 24R^2R'^2\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Note that in the case $K'(x_i,x_j) = y_i y_j$, we then have $R'^2 \leq \max_i y_i^2$.

Proof The proof relies on a series of lemmas given in the Appendix. By the triangle inequality and in view of Lemma 19, the following holds:

$$\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}[K_c K'_c]\right| \leq \left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| + \frac{18R^2R'^2}{m}.$$

Now, in view of Lemma 18, the application of McDiarmid's inequality (McDiarmid, 1989) to $\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}$ gives for any $\varepsilon > 0$:

$$\Pr\left[\left|\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| > \varepsilon\right] \leq 2\exp[-2m\varepsilon^2/(24R^2R'^2)^2].$$

Setting δ to be equal to the right-hand side yields the statement of the proposition. ■

Theorem 12 *Under the assumptions of Proposition 11, and further assuming that the conditions of the Definitions 2-4 are satisfied for* $\rho(K,K')$ *and* $\widehat{\rho}(\mathbf{K},\mathbf{K}')$*, for any* $\delta > 0$*, with probability at least* $1-\delta$*, the following inequality holds:*

$$|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| \leq 18\beta\left[\frac{3}{m} + 8\sqrt{\frac{\log\frac{6}{\delta}}{2m}}\right],$$

with $\beta = \max(R^2R'^2/\mathrm{E}[K_c^2], R^2R'^2/\mathrm{E}[K_c'^2])$.

Proof To shorten the presentation, we first simplify the notation for the alignments as follows:

$$\rho(K,K') = \frac{b}{\sqrt{aa'}} \qquad \widehat{\rho}(\mathbf{K},\mathbf{K}') = \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}},$$

with $b = \mathrm{E}[K_c K'_c]$, $a = \mathrm{E}[K_c^2]$, $a' = \mathrm{E}[K_c'^2]$ and similarly, $\widehat{b} = (1/m^2)\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F$, $\widehat{a} = (1/m^2)\|\mathbf{K}_c\|^2$, and $\widehat{a}' = (1/m^2)\|\mathbf{K}'_c\|^2$. By Proposition 11 and the union bound, for any $\delta > 0$, with probability at least $1-\delta$, all three differences $a-\widehat{a}$, $a'-\widehat{a}'$, and $b-\widehat{b}$ are bounded by $\alpha = \frac{18R^2R'^2}{m} + 24R^2R'^2\sqrt{\frac{\log\frac{6}{\delta}}{2m}}$. Using the definitions of ρ and $\widehat{\rho}$, we can write:

$$\begin{aligned}
|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &= \left|\frac{b}{\sqrt{aa'}} - \frac{\widehat{b}}{\sqrt{\widehat{a}\widehat{a}'}}\right| = \left|\frac{b\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}\sqrt{aa'}}{\sqrt{aa'\widehat{a}\widehat{a}'}}\right| \\
&= \left|\frac{(b-\widehat{b})\sqrt{\widehat{a}\widehat{a}'} - \widehat{b}(\sqrt{aa'} - \sqrt{\widehat{a}\widehat{a}'})}{\sqrt{aa'\widehat{a}\widehat{a}'}}\right| \\
&= \left|\frac{(b-\widehat{b})}{\sqrt{aa'}} - \widehat{\rho}(\mathbf{K},\mathbf{K}')\frac{aa' - \widehat{a}\widehat{a}'}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})}\right|.
\end{aligned}$$

Since $\widehat{\rho}(\mathbf{K},\mathbf{K}') \in [0,1]$, it follows that

$$|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| \leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|aa' - \widehat{a}\widehat{a}'|}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})}.$$

Assume first that $\widehat{a} \leq \widehat{a}'$. Rewriting the right-hand side to make the differences $a - \widehat{a}$ and $a' - \widehat{a}'$ appear, we obtain:

$$\begin{aligned}
|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| &\leq \frac{|b-\widehat{b}|}{\sqrt{aa'}} + \frac{|(a-\widehat{a})a' + \widehat{a}(a'-\widehat{a}')|}{\sqrt{aa'}(\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'})} \\
&\leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a' + \widehat{a}}{\sqrt{aa'} + \sqrt{\widehat{a}\widehat{a}'}}\right] \leq \frac{\alpha}{\sqrt{aa'}}\left[1 + \frac{a'}{\sqrt{aa'}} + \frac{\widehat{a}}{\sqrt{\widehat{a}\widehat{a}'}}\right] \\
&\leq \frac{\alpha}{\sqrt{aa'}}\left[2 + \sqrt{\frac{a'}{a}}\right] = \left[\frac{2}{\sqrt{aa'}} + \frac{1}{a}\right]\alpha.
\end{aligned}$$

We can similarly obtain $\left[\frac{2}{\sqrt{aa'}} + \frac{1}{a'}\right]\alpha$ when $\widehat{a}' \leq \widehat{a}$. Both bounds are less than or equal to $3\max(\frac{\alpha}{a}, \frac{\alpha}{a'})$.
■

Equivalently, one can set the right hand side of the high probability statement presented in Theorem 12 equal to ϵ and solve for δ, which shows that $\Pr\left[|\rho(K,K') - \widehat{\rho}(\mathbf{K},\mathbf{K}')| > \epsilon\right] \leq O(e^{-m\epsilon^2})$.

4.2 Existence of Good Alignment-based Predictors

For classification and regression tasks, the target kernel is based on the labels and defined by $K_Y(x,x') = yy'$, where we denote by y the label of point x and y' that of x'. This section shows the existence of predictors with high accuracy both for classification and regression when the alignment $\rho(K,K_Y)$ between the kernel K and K_Y is high.

In the regression setting, we shall assume that the labels have been normalized such that $\mathrm{E}[y^2] = 1$. In classification, $y = \pm 1$ and thus $\mathrm{E}[y^2] = 1$ without any normalization. Denote by h^* the hypothesis defined for all $x \in \mathcal{X}$ by

$$h^*(x) = \frac{\mathrm{E}_{x'}[y' K_c(x,x')]}{\sqrt{\mathrm{E}[K_c^2]}}.$$

Observe that by definition of h^*, $\mathrm{E}_x[y h^*(x)] = \rho(K,K_Y)$. For any $x \in \mathcal{X}$, define $\gamma(x) = \sqrt{\frac{\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}_{x,x'}[K_c^2(x,x')]}}$ and $\Gamma = \max_x \gamma(x)$. The following result shows that the hypothesis h^* has high accuracy when the kernel alignment is high and Γ not too large.[3]

Theorem 13 (classification) *Let $R(h^*) = \Pr[y h^*(x) < 0]$ denote the error of h^* in binary classification. For any kernel K such that $0 < \mathrm{E}[K_c^2] < +\infty$, the following holds:*

$$R(h^*) \leq 1 - \rho(K,K_Y)/\Gamma.$$

[3] A version of this result was presented by Cristianini, Shawe-Taylor, Elisseeff, and Kandola (2001) and Cristianini, Kandola, Elisseeff, and Shawe-Taylor (2002) for the so-called Parzen window solution and non-centered kernels. However, both proofs are incorrect since they rely implicitly on the fact that $\max_x \left[\frac{\mathrm{E}_{x'}[K^2(x,x')]}{\mathrm{E}_{x,x'}[K^2(x,x')]}\right]^{\frac{1}{2}} = 1$, which can only hold in the trivial case where the kernel function K^2 is a constant: by definition of the maximum and expectation operators, $\max_x \left[\mathrm{E}_{x'}[K^2(x,x')]\right] \geq \mathrm{E}_x\left[\mathrm{E}_{x'}[K^2(x,x')]\right]$, with equality only in the constant case.

Proof Note that for all $x \in \mathcal{X}$,

$$|yh^*(x)| = \frac{|y \operatorname{E}_{x'}[y' K_c(x,x')]|}{\sqrt{\operatorname{E}[K_c^2]}} \leq \frac{\sqrt{\operatorname{E}_{x'}[y'^2] \operatorname{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\operatorname{E}[K_c^2]}} = \frac{\sqrt{\operatorname{E}_{x'}[K_c^2(x,x')]}}{\sqrt{\operatorname{E}[K_c^2]}} \leq \Gamma.$$

In view of this inequality, and the fact that $\operatorname{E}_x[yh^*(x)] = \rho(K, K_Y)$, we can write:

$$\begin{aligned} 1 - R(h^*) &= \Pr[yh^*(x) \geq 0] \\ &= \operatorname{E}[\mathbf{1}_{\{yh^*(x) \geq 0\}}] \\ &\geq \operatorname{E}\left[\frac{yh^*(x)}{\Gamma} \mathbf{1}_{\{yh^*(x) \geq 0\}}\right] \\ &\geq \operatorname{E}\left[\frac{yh^*(x)}{\Gamma}\right] = \rho(K, K_Y)/\Gamma, \end{aligned}$$

where $\mathbf{1}_\omega$ is the indicator function of the event ω. ■

A probabilistic version of the theorem can be straightforwardly derived by noting that by Markov's inequality, for any $\delta > 0$, with probability at least $1 - \delta$, $|\gamma(x)| \leq 1/\sqrt{\delta}$.

Theorem 14 (regression) *Let $R(h^*) = \operatorname{E}_x[(y - h^*(x))^2]$ denote the error of h^* in regression. For any kernel K such that $0 < \operatorname{E}[K_c^2] < +\infty$, the following holds:*

$$R(h^*) \leq 2(1 - \rho(K, K_Y)).$$

Proof By the Cauchy-Schwarz inequality, it follows that:

$$\begin{aligned} \operatorname*{E}_x[h^{*2}(x)] &= \operatorname*{E}_x\left[\frac{\operatorname{E}_{x'}[y' K_c(x,x')]^2}{\operatorname{E}[K_c^2]}\right] \\ &\leq \operatorname*{E}_x\left[\frac{\operatorname{E}_{x'}[y'^2] \operatorname{E}_{x'}[K_c^2(x,x')]}{\operatorname{E}[K_c^2]}\right] \\ &= \frac{\operatorname{E}_{x'}[y'^2] \operatorname{E}_{x,x'}[K_c^2(x,x')]}{\operatorname{E}[K_c^2]} = \operatorname*{E}_{x'}[y'^2] = 1. \end{aligned}$$

Using again the fact that $\operatorname{E}_x[yh^*(x)] = \rho(K, K_Y)$, the error of h^* can be bounded as follows:

$$\operatorname{E}[(y - h^*(x))^2] = \operatorname*{E}_x[h^*(x)^2] + \operatorname*{E}_x[y^2] - 2 \operatorname*{E}_x[yh^*(x)] \leq 1 + 1 - 2\rho(K, K_Y),$$

which concludes the proof. ■

The hypothesis h^* is closely related to the hypothesis h_S^* derived as follows from a finite sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$:

$$h_S(x) = \frac{\frac{1}{m} \sum_{i=1}^m y_i K_c(x, x_i)}{\sqrt{\frac{1}{m^2} \sum_{i,j=1}^m K_c(x_i, x_j)^2} \sqrt{\frac{1}{m^2} \sum_{i,j=1}^m (y_i y_j)^2}}.$$

Note in particular that $\widehat{\operatorname{E}}_x[yh_S(x)] = \widehat{\rho}(\mathbf{K}, \mathbf{K_Y})$, where we denote by $\widehat{\operatorname{E}}$ the expectation based on the empirical distribution. Using this and other results of this section, it is not hard to show that with high probability $|R(h^*) - R(h_S^*)| \leq O(1/\sqrt{m})$ both in the classification and regression settings.

For classification, the existence of a good predictor g^* based on the unnormalized alignment ρ_u (see Definition 5) can also be shown. The corresponding guarantees are simpler and do not depend on a term such as Γ. However, unlike the normalized case, the loss of the predictor g^*_S derived from a finite sample may not always be close to that of g^*. Note that in classification, for any label y, $|y| = 1$, thus, by Lemma 6, the following holds: $0 \leq \rho_{\mathrm{u}}(K, K_Y)| \leq R^2$. Let g^* be the hypothesis defined by:

$$g^*(x) = \mathop{\mathrm{E}}_{x'}[y' K_c(x, x')].$$

Since $0 \leq \rho_{\mathrm{u}}(K, K_Y)| \leq R^2$, the following theorem provides strong guarantees for g^* when the unnormalized alignment a is sufficiently large, that is close to R^2.

Theorem 15 (classification) *Let* $R(g^*) = \Pr[yg^*(x) < 0]$ *denote the error of* g^* *in binary classification. For any kernel* K *such that* $\sup_{x \in \mathcal{X}} K_c(x, x) \leq R^2$*, we have:*

$$R(g^*) \leq 1 - \rho_{\mathrm{u}}(K, K_Y)/R^2.$$

Proof Note that for all $x \in \mathcal{X}$,

$$|yg^*(x)| = |g^*(x)| = |\mathop{\mathrm{E}}_{x'}[y' K_c(x, x')]| \leq R^2.$$

Using this inequality, and the fact that $\mathrm{E}_x[yg^*(x)] = \rho_{\mathrm{u}}(K, K_Y)$, we can write:

$$\begin{aligned} 1 - R(g^*) = \Pr[yg^*(x) \geq 0] &= \mathrm{E}[\mathbf{1}_{\{yg^*(x) \geq 0\}}] \\ &\geq \mathrm{E}\left[\frac{yg^*(x)}{R^2} \mathbf{1}_{\{yh^*(x) \geq 0\}}\right] \\ &\geq \mathrm{E}\left[\frac{yg^*(x)}{R^2}\right] = \rho_{\mathrm{u}}(K, K_Y)/R^2, \end{aligned}$$

which concludes the proof. ■

4.3 Generalization Bounds for Two-stage Learning Kernel Algorithms

This section presents stability-based generalization bounds for two-stage learning kernel algorithms. The proof of a stability-based learning bound hinges on showing that the learning algorithm is *stable*, that is the pointwise loss of a learned hypothesis does not change drastically if the training sample changes only slightly. We refer the reader to Bousquet and Elisseeff (2000) for a full introduction.

We present learning bounds for the case where the second stage of the algorithm is kernel ridge regression (KRR). Similar results can be given in classification using algorithms such as SVMs in the second stage. Thus, in the first stage, the algorithms we examine select a combination weight parameter $\boldsymbol{\mu} \in \mathcal{M}_q = \{\boldsymbol{\mu} \colon \boldsymbol{\mu} \geq \mathbf{0}, \|\boldsymbol{\mu}\|_q^q = \Lambda_q\}$ which defines a kernel $K_{\boldsymbol{\mu}}$, and in the second stage use KRR to select a hypothesis from the RKHS associated to $K_{\boldsymbol{\mu}}$. While several of our results hold in general, we will be more specifically interested in the alignment maximization algorithm presented in Section 3.2.2.

Recall that for a fixed kernel function $K_{\boldsymbol{\mu}}$ with associated RKHS $\mathbb{H}_{K_{\boldsymbol{\mu}}}$ and training set $S = ((x_1,y_1),\ldots,(x_m,y_m))$, the KRR optimization problem is defined by the following constraint optimization problem:

$$\min_{h\in\mathbb{H}_{K_{\boldsymbol{\mu}}}} G(h) = \lambda_0\|h\|^2_{K_{\boldsymbol{\mu}}} + \frac{1}{m}\sum_{i=1}^{m}(h(x_i)-y_i)^2. \tag{10}$$

We first analyze the stability of two-stage algorithms and then use that to derive a stability-based generalization bound (Bousquet and Elisseeff, 2000). More precisely, we examine the pointwise difference in hypothesis values obtained on any point x when the algorithm has been trained on two data sets S and S' of size m that differ in exactly one point.

In what follows, we denote by $\|\mathbf{K}\|_{s,t} = (\sum_{k=1}^{p}\|\mathbf{K}_k\|_s^t)^{1/t}$ the (s,t)-norm of a collection of matrices and by $\Delta\boldsymbol{\mu}$ the difference $\boldsymbol{\mu}'-\boldsymbol{\mu}$ of the combination vector $\boldsymbol{\mu}'$ and $\boldsymbol{\mu}$ returned by the first stage of the algorithm by training on S, respectively S'.

Theorem 16 (Stability of two-stage learning kernel algorithm) *Let S and S' be two samples of size m that differ in exactly one point and let h and h' be the associated hypotheses generated by a two-stage KRR learning kernel algorithm with the constraint $\boldsymbol{\mu}\in\mathcal{M}_1$. Then, for any $s,t\geq 1$ with $\frac{1}{s}+\frac{1}{r}=1$ and any $x\in\mathcal{X}$:*

$$|h'(x)-h(x)| \leq \frac{2\Lambda_1 R^2 M}{\lambda_0 m}\Big[1+\frac{\|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}_c\|_{2,t}}{2\lambda_0}\Big],$$

where M is an upper bound on the target labels and $R^2 = \sup_{\substack{k\in[1,p]\\ x\in\mathcal{X}}} K_k(x,x)$.

Proof The KRR algorithm returns the hypothesis $h(x) = \sum_{i=1}^{m}\alpha_i K_{\boldsymbol{\mu}}(x_i,x)$, where $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}} + m\lambda_0\mathbf{I})^{-1}\mathbf{y}$. Thus, this hypothesis is parametrized by the kernel weight vector $\boldsymbol{\mu}$, which defines the kernel function, and the sample S, which is used to populate the kernel matrix, and will be explicitly denoted $h_{\boldsymbol{\mu},S}$. To estimate the stability of the overall two-stage algorithm, $\Delta h_{\boldsymbol{\mu},S} = h_{\boldsymbol{\mu}',S'} - h_{\boldsymbol{\mu},S}$, we use the decomposition

$$\Delta h_{\boldsymbol{\mu},S} = (h_{\boldsymbol{\mu}',S'} - h_{\boldsymbol{\mu}',S}) + (h_{\boldsymbol{\mu}',S} - h_{\boldsymbol{\mu},S})$$

and bound each parenthesized term separately. The first parenthesized term measures the pointwise stability of KRR due to a change of a single training point with a fixed kernel. This can be bounded using Theorem 2 of Cortes et al. (2009a). Since, for all $x\in\mathcal{X}$, $K_{\boldsymbol{\mu}}(x,x) = \sum_{k=1}^{p}\mu_k K_k(x,x) \leq R^2\sum_{k=1}^{p}\mu_k \leq \Lambda_1 R^2$, using that theorem yields the following bound:

$$\forall x\in\mathcal{X},\quad |h_{\boldsymbol{\mu},S'}(x) - h_{\boldsymbol{\mu},S}(x)| \leq \frac{2\Lambda_1 R^2 M}{\lambda_0 m}.$$

The second parenthesized term measures the pointwise difference of the hypotheses due to the change of kernel from $\mathbf{K}_{\boldsymbol{\mu}'}$ to $\mathbf{K}_{\boldsymbol{\mu}}$ for a fixed training sample when using KRR. By Proposition 1 of Cortes et al. (2010c), this term can be bounded as follows:

$$\forall x\in\mathcal{X}, |h_{\boldsymbol{\mu}',S}(x) - h_{\boldsymbol{\mu},S}(x)| \leq \frac{\Lambda_1 R^2 M}{\lambda_0^2 m}\|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\|.$$

The term $\|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\|$ can be bounded using Hölder's inequality as follows:

$$\|\mathbf{K}_{\boldsymbol{\mu}'} - \mathbf{K}_{\boldsymbol{\mu}}\| = \|\sum_{k=1}^{p}(\Delta\mu_k)\mathbf{K}_k\| \leq \sum_{k=1}^{p}|\Delta\mu_k|\;\|\mathbf{K}_k\| \leq \|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}\|_{2,t},$$

which completes the proof. ■

The pointwise stability result just presented can be used directly to derive a generalization bound for two-stage learning kernel algorithms as in Bousquet and Elisseeff (2000).

For a hypothesis h, we denote by $R(h)$ its generalization error and by $\widehat{R}(h)$ its empirical error on a $S = ((x_1, y_1), \ldots, (x_m, y_m))$:

$$R(h) = \mathop{\mathrm{E}}_{x,y}[(h_S(x) - y)^2] \quad \widehat{R}(h) = \frac{1}{m}\sum_{i=1}^{m}(h_S(x_i) - y_i)^2.$$

Theorem 17 (Stability-based generalization bound) *Let h_S denote the hypothesis returned by a two-stage KRR kernel learning algorithm with the constraint $\boldsymbol{\mu} \in \mathcal{M}_1$ when trained on sample S. For any $s, t \geq 1$ with $\frac{1}{s} + \frac{1}{r} = 1$, with probability at least $1 - \delta$ over samples S of size m, the following bound holds:*

$$R(h_S) \leq \widehat{R}(h_S) + \frac{2M_1M_2}{m} + \Big(1 + \frac{16M_2}{M_1}\Big)\frac{M_1M_2}{4}\sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

with $M_1 = 2\Big[1 + \sqrt{\frac{\Lambda_1 R^2}{\lambda_0}}\Big]M$ and $M_2 = \frac{2\Lambda_1 R^2}{\lambda_0}\Big[1 + \frac{\|\Delta\boldsymbol{\mu}\|_s \|\mathbf{K}_c\|_{2,t}}{2\lambda_0}\Big]M$.

Proof Since h_S is the minimizer of the objective (10) and since $\mathbf{0}$ belongs to the hypothesis space,

$$G(h_S) \leq G(\mathbf{0}) = \frac{1}{m}\sum_{i=1}^{m}(0 - y_i)^2 \leq M^2.$$

Furthermore, since the mean squared loss is non-negative, we can write: $\lambda_0 \|h_S\|^2_{K_{\mu}} \leq G(h_S)$. Therefore, $\|h_S\|^2_{K_{\mu}} \leq \frac{M^2}{\lambda_0}$. By the reproducing property, for any $x \in \mathcal{X}$,

$$\begin{aligned}
|h_S(x)| = |\langle h_S, K_{\boldsymbol{\mu}}(x, \cdot)\rangle_{K_{\boldsymbol{\mu}}}| &\leq \|h_S\|_{K_{\boldsymbol{\mu}}}\sqrt{K_{\boldsymbol{\mu}}(x,x)} \\
&= \sqrt{\frac{M}{\lambda_0}}\sqrt{\sum_{k=1}^{p}\mu_k K_k(x,x)} \\
&\leq \sqrt{\frac{M}{\lambda_0}}\sqrt{\|\boldsymbol{\mu}\|_1 R^2} \leq RM\sqrt{\frac{\Lambda_1}{\lambda_0}}.
\end{aligned}$$

Thus, for all $x \in \mathcal{X}$ and $y \in [-M, M]$, the squared loss can be bounded as follows

$$|h_S(x) - y| \leq \Big(M + RM\sqrt{\frac{\Lambda_1}{\lambda_0}}\Big) = \frac{M_1}{2}.$$

This implies that the squared loss is M_1-Lipschitz and by Theorem 16 that the algorithm is stable with a uniform stability parameter $\beta \leq \frac{M_1M_2}{m}$ bounded as follows:

$$|(h_{S'}(x) - y)^2 - (h_S(x) - y)^2| \leq M_1|h_{S'}(x) - h_S(x)| \leq \frac{M_1M_2}{m}.$$

The application of Theorem 12 of Bousquet and Elisseeff (2000) with the bound on the loss $\frac{M_1}{2}$ and the uniform stability parameter β directly yields the statement. ■

The inequality just presented holds for all two-stage learning kernel algorithms. To determine its convergence rate, the term $\|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}_c\|_{2,t}$ must be bounded. Let $s = 1$ and $t = \infty$, and assume that the base kernels $\mathbf{K}_k$, $k \in [1, p]$, are trace-normalized as in our experiments (Section 3), then a straightforward bound can be given for this term:

$$\|\Delta\boldsymbol{\mu}\|_1\|\mathbf{K}_c\|_{2,\infty} \le (\|\boldsymbol{\mu}'\|_1 + \|\boldsymbol{\mu}\|_1) \max_{k\in[1,k]} \|\mathbf{K}_{kc}\|_2 \le \max_{k\in[1,k]} 2\Lambda_1 \operatorname{Tr}[\mathbf{K}_{kc}] \le 2\Lambda_1.$$

Thus, in the statement of Theorem 17, M_2 can be replaced with $\frac{2\Lambda_1 R^2}{\lambda_0}\left[1+\frac{\Lambda_1}{\lambda_0}\right]M$ and, for Λ_1 and λ_0 constant, the learning bound converges in $O(1/\sqrt{m})$.

The straightforward upper bound on $\|\Delta\boldsymbol{\mu}\|_s\|\mathbf{K}_c\|_{2,t}$ applies to all such two-stage learning kernel algorithms. For a specific algorithm, finer or more favorable bounds could be derived. We have initiated this study in the specific case of the alignment maximization algorithm. The result given in Proposition 21 (Appendix B) can be used to bound $\|\Delta\boldsymbol{\mu}\|_2$ and thus $\|\Delta\boldsymbol{\mu}\|_2\|\mathbf{K}_c\|_{2,2}$.

Note that in the specific case of the alignment maximization algorithm, if $\boldsymbol{\mu}^*$ is the solution obtained for the constraint $\boldsymbol{\mu} \in \mathcal{M}_2$, then it is also the alignment maximizing solution found in the set $\boldsymbol{\mu} \in \mathcal{M}_1$ with $\Lambda_1 = \|\boldsymbol{\mu}^*\|_1 \le \sqrt{p}\|\boldsymbol{\mu}\|_2 \le \sqrt{p}\Lambda_2$. This makes the dependence on p explicit in the case of the constraint $\boldsymbol{\mu} \in \mathcal{M}_2$.

5. Experiments

This section compares the performance of several learning kernel algorithms for classification and regression. We compare the alignment-based two-stage learning kernel algorithms `align` and `alignf`, as well as the single-stage algorithm presented in Section 3 with the following algorithms:

Uniform combination (`unif`*)*: this is the most straightforward method, which consists of choosing equal mixture weights, thus the kernel matrix used is,

$$\mathbf{K}_{\boldsymbol{\mu}} = \frac{\Lambda}{p}\sum_{k=1}^{p}\mathbf{K}_k.$$

Nevertheless, improving upon the performance of this method has been surprisingly difficult for standard (one-stage) learning kernel algorithms (Cortes, 2009; Cortes et al., 2011b).

Norm-1 regularized combination (`l1-svm`): this algorithm optimizes the SVM objective

$$\min_{\boldsymbol{\mu}}\max_{\boldsymbol{\alpha}}\; 2\boldsymbol{\alpha}^\top\mathbf{1} - \boldsymbol{\alpha}^\top\mathbf{Y}^\top\mathbf{K}_{\boldsymbol{\mu}}\mathbf{Y}\boldsymbol{\alpha}$$
$$\text{subject to: } \boldsymbol{\mu} \ge \mathbf{0}, \operatorname{Tr}[\mathbf{K}_{\boldsymbol{\mu}}] \le \Lambda, \boldsymbol{\alpha}^\top\mathbf{y} = 0, \mathbf{0} \le \boldsymbol{\alpha} \le \mathbf{C},$$

as described by Lanckriet et al. (2004). Here, $\mathbf{Y}$ is the diagonal matrix constructed from the labels $\mathbf{y}$ and $\mathbf{C}$ is the regularization parameter of the SVM.

Norm-2 regularized combination (`l2-krr`): this algorithm optimizes the kernel ridge regression objective

$$\min_{\boldsymbol{\mu}}\max_{\boldsymbol{\alpha}} -\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top\mathbf{K}_{\boldsymbol{\mu}}\boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top\mathbf{y}$$
$$\text{subject to: } \boldsymbol{\mu} \ge \mathbf{0}, \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_2 \le \Lambda.$$

	KINEMATICS	IONOSPHERE	GERMAN	SPAMBASE	SPLICE
SIZE	1000	351	1000	1000	1000
γ	-3, 3	-3, 3	-4, 3	-12, -7	-9, -3
`unif`	$.138 \pm .005$ $.158 \pm .013$	$.479 \pm .033$ $.246 \pm .033$	$.259 \pm .018$ $.089 \pm .008$	$.187 \pm .028$ $.138 \pm .031$	$.152 \pm .022$ $.122 \pm .011$
`1-stage`	$.137 \pm .005$ $.155 \pm .012$	$.470 \pm .032$ $.251 \pm .035$	$.260 \pm .026$ $.082 \pm .003$	$.209 \pm .028$ $.099 \pm .024$	$.153 \pm .025$ $.105 \pm .006$
`align`	$.125 \pm .004$ $.173 \pm .016$	$.456 \pm .036$ $.261 \pm .040$	$.255 \pm .015$ $.089 \pm .008$	$.186 \pm .026$ $.140 \pm .031$	$.151 \pm .024$ $.123 \pm .011$
`alignf`	$.115 \pm .004$ $.176 \pm .017$	$.444 \pm .034$ $.278 \pm .057$	$.242 \pm .015$ $.093 \pm .009$	$.180 \pm .024$ $.146 \pm .028$	$.139 \pm .013$ $.124 \pm .011$

REGRESSION CLASSIFICATION

Table 2: Error measures (top) and alignment values (bottom) for `unif`, `1-stage` (`l2-krr` or `l1-svm`), `align` and `alignf` with kernels built from linear combinations of Gaussian base kernels. The choice of γ_0, γ_1 is listed in the row labeled γ and the total size of the data set used is listed under SIZE. The results are shown with ± 1 standard deviation measured by 5-fold cross-validation. Further measures of significance are shown in Appendix C, Table 4.

The L_2 regularized method is used for regression since it is shown in Cortes et al. (2009a) to outperform the alternative L_1 regularized method in similar settings. Here, λ is the regularization parameter of KRR and $\boldsymbol{\mu}_0$ is an additional regularization parameter for the kernel selection.

In all experiments, the error measures reported are for 5-fold cross validation, where, in each trial, three folds are used for training, one used for validation, and one for testing. For the two-stage methods, the same training and validation data is used for both stages of the learning. The regularization parameter Λ is chosen via a grid search based on the performance on the validation set, while the regularization parameters $\mathbf{C}$ for `l1-svm` and λ for `l2-krr` are fixed since only the ratios $\mathbf{C}/\Lambda$ and λ/Λ are important. More explicitly, for the KRR algorithm, scaling the vector $\boldsymbol{\mu}$ by Λ results in a scaled dual solution: $\boldsymbol{\alpha} = (\mathbf{K}_{\boldsymbol{\mu}}\Lambda + \lambda\mathbf{I})^{-1}\mathbf{y} = \Lambda^{-1}(\mathbf{K}_{\boldsymbol{\mu}} + \frac{\lambda}{\Lambda}\mathbf{I})^{-1}\mathbf{y}$. In turn, we see that the primal solution $h(x) = \sum_{i=1}^{m} \Lambda^{-1}\alpha_i \Lambda K_{\boldsymbol{\mu}}(x, x_i) = \sum_{i=1}^{m} \alpha_i K_{\boldsymbol{\mu}}(x, x_i)$ is equivalent to the solution of the KRR algorithm that uses a regularization parameter equal to λ/Λ without scaling $\boldsymbol{\mu}$ and, thus, it suffices to vary only one regularization parameter. In the case of SVMs, the scale of the hypothesis does not change its sign (or the binary prediction) and thus the same property can be shown to hold. The $\boldsymbol{\mu}_0$ parameter is set to zero in our experiments.

5.1 General Kernel Combinations

In the first set of experiments, we consider combinations of Gaussian kernels of the form

$$\mathbf{K}_\gamma(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

with varying bandwidth parameter $\gamma \in \{2^{\gamma_0}, 2^{\gamma_0+1}, \ldots, 2^{1-\gamma_1}, 2^{\gamma_1}\}$. The values γ_0 and γ_1 are chosen such that the base kernels are sufficiently different in alignment and performance. Each base kernel is centered and normalized to have trace one. We test the algorithms on several data sets taken from the UCI Machine Learning Repository (`http://archive.ics.uci.edu/ml/`) and Delve (`http://www.cs.toronto.edu/~delve/data/datasets.html`).

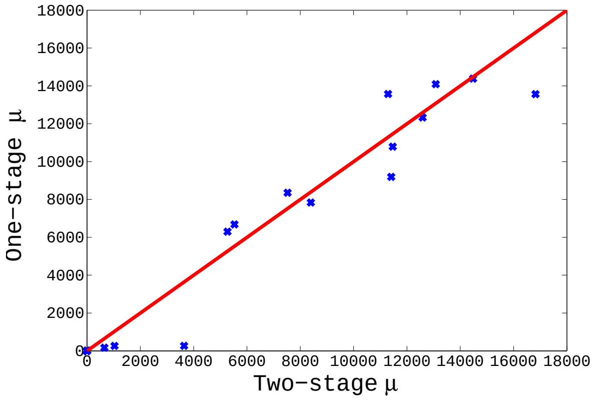


Figure 3: A scatter plot comparison of the different kernel combination weight values obtained by optimally tuned one-stage and two-stage algorithms on the kinematics data set.

Table 2 summarizes our results. For regression, we compare against the `l2-krr` method and report RMSE. For classification, we compare against the `l1-svm` method and report the misclassification percentage. In general, we see that performance and alignment are well correlated. In all data sets, we see improvement over the uniform combination as well as the one-stage kernel learning algorithms. Note that although the `align` method often increases the alignment of the final kernel, as compared to the uniform combination, the `alignf` method gives the best alignment since it directly maximizes this quantity. Nonetheless, `align` provides an inexpensive heuristic that increases the alignment and performance of the final combination kernel.

In our experiments with the one-stage KRR algorithm presented in Section 3.4, there was no significant improvement found over the two-stage `alignf` algorithm with respect to the kinematics and ionosphere data sets. In fact, for optimally cross-validated parameters γ, γ' and γ'' the solution combination weights were found to closely coincide with the `alignf` solution (see Figure 3). This would suggest the use of the two-stage algorithm over the one-stage, since there are fewer parameters to tune and the problem can be solved as a standard QP.

To the best of our knowledge, these are the first kernel combination experiments for alignment with general base kernels. Previous experiments seem to have dealt exclusively with rank-one base kernels built from the eigenvectors of a single kernel matrix (Cristianini et al., 2001). In the next section, we also examine rank-one kernels, although not generated from a spectral decomposition.

5.2 Rank-one Kernel Combinations

In this set of experiments we use the sentiment analysis data set version 1 from Blitzer et al. (2007): *books*, *dvd*, *electronics* and *kitchen*. Each domain has 2,000 examples. In the regression setting, the goal is to predict a rating between 1 and 5, while for classification the goal is to discriminate positive (ratings ≥ 4) from negative reviews (ratings ≤ 2). We use rank-one kernels based on the 4,000 most frequent bigrams. The kth base kernel, $\mathbf{K}_k$, corresponds to the kth bigram count $\mathbf{v}_k$, $\mathbf{K}_k = \mathbf{v}_k\mathbf{v}_k^\top$. Each base kernel is normalized to have trace one and the labels are centered.

The `alignf` method returns a sparse weight vector due to the constraint $\boldsymbol{\mu} \geq \mathbf{0}$. As is demonstrated by the performance of the `l1-svm` method, Table 3, and also previously observed by Cortes

	BOOKS	DVD	ELEC	KITCHEN
`unif`	1.442 ± .015	1.438 ± .033	1.342 ± .030	1.356 ± .016
	0.029 ± .005	0.029 ± .005	0.038 ± .002	0.039 ± .006
`l2-krr`	1.410 ± .024	1.423 ± .034	1.318 ± .033	1.333 ± .015
	0.036 ± .008	0.036 ± .009	0.050 ± .004	0.056 ± .005
`align`	1.401 ± .035	1.414 ± .017	1.308 ± .033	1.312 ± .012
	0.046 ± .006	0.047 ± .005	0.065 ± .004	0.076 ± .008

REGRESSION

	BOOKS	DVD	ELEC	KITCHEN
`unif`	0.258 ± .017	0.243 ± .015	0.188 ± .014	0.201 ± .020
	0.030 ± .004	0.030 ± .005	0.040 ± .002	0.039 ± .007
`l1-svm`	0.286 ± .016	0.292 ± .025	0.238 ± .019	0.236 ± .024
	0.030 ± .011	0.033 ± .014	0.051 ± .004	0.058 ± .007
`align`	0.243 ± .020	0.214 ± .020	0.166 ± .016	0.172 ± .022
	0.043 ± .003	0.045 ± .005	0.063 ± .004	0.070 ± .010

CLASSIFICATION

Table 3: The error measures (top) and alignment values (bottom) on four sentiment analysis domains using kernels learned as combinations of rank-one base kernels corresponding to individual features. The results are shown with ±1 standard deviation as measured by 5-fold cross-validation. Further measures of significance are shown in Appendix C, Table 5.

et al. (2009a), a sparse weight vector $\boldsymbol{\mu}$ does not generally offer an improvement over the uniform combination in the rank-one setting. Thus, we focus on the performance of `align` and compare it to `unif` and one-stage learning methods. Table 3 shows that `align` significantly improves both the alignment and the error percentage over `unif` and also improves somewhat over the one-stage `l2-krr` algorithm. Evidence of statistical significance is provided in Appendix C, Table 5. Note that, although the sparse weighting provided by `l1-svm` improves the alignment in certain cases, it does not improve performance.

6. Conclusion

We presented a series of novel algorithmic, theoretical, and empirical results for learning kernels based on the notion of centered alignment. Our experiments show a consistent improvement of the performance of alignment-based algorithms over previous learning kernel techniques, as well as the straightforward uniform kernel combination, which has been difficult to surpass in the past, in both classification and regression. The algorithms we described are efficient and easy to implement. All the algorithms presented in this paper are available in the open-source C++ library available at `www.openkernel.org`. They can be used in a variety of applications to improve performance. We also gave an extensive theoretical analysis which provides a number of guarantees for centered alignment-based algorithms and methods. Several of the algorithmic and theoretical results presented can be extended to other learning settings. In particular, methods based on similar ideas could be used to design learning kernel algorithms for dimensionality reduction.

The notion of centered alignment served as a key similarity measure to achieve these results. Note that we are not proving that good alignment is necessarily needed for a good classifier, but both our theory and empirical results do suggest the existence of accurate predictors with a good centered alignment. Different methods based on possibly different efficiently computable similarity measures could be used to design effective learning kernel algorithms. In particular, the notion of similarity suggested by Balcan and Blum (2006), if it could be computed from finite samples, could be used in a equivalent way.

Acknowledgments

The work of author MM was partly supported by a Google Research Award.

Appendix A. Lemmas Supporting Proof of Proposition 11

For a function f of the sample S, we denote by $\Delta(f)$ the difference $f(S') - f(S)$, where S' is a sample differing from S by just one point, say the m-th point is x_m in S and x'_m in S'. The following perturbation bound will be needed in order to apply McDiarmid's inequality.

Lemma 18 *Let $\mathbf{K}$ and $\mathbf{K}'$ denote kernel matrices associated to the kernel functions K and K' for a sample of size m according to the distribution D. Assume that for any $x \in \mathcal{X}$, $K(x,x) \leq R^2$ and $K'(x,x) \leq R'^2$. Then, the following perturbation inequality holds when changing one point of the sample:*

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F)| \leq \frac{24R^2R'^2}{m}.$$

Proof By Lemma 1, we can write:

$$\begin{aligned}
\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F &= \operatorname{Tr}\left[\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}\left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m}\right]\mathbf{K}'\right] \\
&= \operatorname{Tr}\left[\mathbf{K}\mathbf{K}' - \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\mathbf{K}' - \mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}' + \frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}\frac{\mathbf{1}\mathbf{1}^\top}{m}\mathbf{K}'\right] \\
&= \langle \mathbf{K}, \mathbf{K}' \rangle_F - \frac{\mathbf{1}^\top(\mathbf{K}\mathbf{K}' + \mathbf{K}'\mathbf{K})\mathbf{1}}{m} + \frac{(\mathbf{1}^\top\mathbf{K}\mathbf{1})(\mathbf{1}^\top\mathbf{K}'\mathbf{1})}{m^2}.
\end{aligned}$$

The perturbation of the first term is given by

$$\Delta(\langle \mathbf{K}, \mathbf{K}' \rangle_F) = \sum_{i=1}^{m} \Delta(\mathbf{K}_{im}\mathbf{K}'_{im}) + \sum_{i \neq m} \Delta(\mathbf{K}_{mi}\mathbf{K}'_{mi}).$$

By the Cauchy-Schwarz inequality, for any $i, j \in [1, m]$,

$$|\mathbf{K}_{ij}| = |K(x_i, x_j)| \leq \sqrt{K(x_i, x_i)K(x_j, x_j)} \leq R^2$$

and the product can be bound as $|\mathbf{K}_{i,j}\mathbf{K}'_{i,j}| \leq |\mathbf{K}_{i,j}||\mathbf{K}'_i j| \leq R^2 R'^2$. The difference of products is then bound as $|\Delta(\mathbf{K}_{i,j}\mathbf{K}'_{i,j})| \leq 2R^2R'^2$. Thus,

$$\frac{1}{m^2}|\Delta(\langle \mathbf{K},\mathbf{K}'\rangle_F)| \leq \frac{2m-1}{m^2}(2R^2R'^2) \leq \frac{4R^2R'^2}{m}.$$

Similarly, for the first part of the second term, we obtain

$$\begin{aligned}\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top \mathbf{K}\mathbf{K}'\mathbf{1}}{m}\right)\right| &= \left|\Delta\left(\sum_{i,j,k=1}^{m}\frac{\mathbf{K}_{ik}\mathbf{K}'_{kj}}{m^3}\right)\right| \\ &= \left|\Delta\left(\frac{\sum_{i,k=1}^{m}\mathbf{K}_{ik}\mathbf{K}'_{km} + \sum_{i,j\neq m}\mathbf{K}_{im}\mathbf{K}'_{mj}}{m^3} + \frac{\sum_{k\neq m, j\neq m}\mathbf{K}_{mk}\mathbf{K}'_{kj}}{m^3}\right)\right| \\ &\leq \frac{m^2+m(m-1)+(m-1)^2}{m^3}(2R^2R'^2) \leq \frac{3m^2-3m+1}{m^3}(2R^2R'^2) \\ &\leq \frac{6R^2R'^2}{m}.\end{aligned}$$

Similarly, we have:

$$\frac{1}{m^2}\left|\Delta\left(\frac{\mathbf{1}^\top \mathbf{K}'\mathbf{K}\mathbf{1}}{m}\right)\right| \leq \frac{6R^2R'^2}{m}.$$

The final term is bounded as follows,

$$\begin{aligned}\frac{1}{m^2}\left|\Delta\left(\frac{(\mathbf{1}^\top \mathbf{K}\mathbf{1})(\mathbf{1}^\top \mathbf{K}'\mathbf{1})}{m^2}\right)\right| &\leq \left|\Delta\left(\frac{\sum_{i,j,k}\mathbf{K}_{ij}\mathbf{K}'_{km} + \sum_{i,j,k\neq m}\mathbf{K}_{ij}\mathbf{K}'_{mk}}{m^4} + \right.\right. \\ &\qquad \left.\left.\frac{\sum_{i,j\neq m,k\neq m}\mathbf{K}_{im}\mathbf{K}'_{jk} + \sum_{i\neq m,j\neq m,k\neq m}\mathbf{K}_{mi}\mathbf{K}'_{jk}}{m^4}\right)\right| \\ &\leq \frac{m^3+m^2(m-1)+m(m-1)^2+(m-1)^3}{m^4}(2R^2R'^2) \\ &\leq \frac{8R^2R'^2}{m}.\end{aligned}$$

Combining these last four inequalities leads directly to the statement of the lemma. ∎

Because of the diagonal terms of the matrices, $\frac{1}{m^2}\langle \mathbf{K}_c,\mathbf{K}'_c\rangle_F$ is not an unbiased estimate of $\mathrm{E}[K_cK'_c]$. However, as shown by the following lemma, the estimation bias decreases at the rate $O(1/m)$.

Lemma 19 *Under the same assumptions as Lemma 18, the following bound on the difference of expectations holds:*

$$\left|\underset{x,x'}{\mathrm{E}}[K_c(x,x')K'_c(x,x')] - \underset{S}{\mathrm{E}}\left[\frac{\langle \mathbf{K}_c,\mathbf{K}'_c\rangle_F}{m^2}\right]\right| \leq \frac{18R^2R'^2}{m}.$$

Proof To simplify the notation, unless otherwise specified, the expectation is taken over x,x' drawn according to the distribution D. The key observation used in this proof is that

$$\underset{S}{\mathrm{E}}[\mathbf{K}_{ij}\mathbf{K}'_{ij}] = \underset{S}{\mathrm{E}}[K(x_i,x_j)K'(x_i,x_j)] = \mathrm{E}[KK'], \tag{11}$$

for i, j distinct. For expressions such as $\mathrm{E}_S[\mathbf{K}_{ik}\mathbf{K}'_{kj}]$ with i, j, k distinct, we obtain the following:

$$\operatorname*{E}_S[\mathbf{K}_{ik}\mathbf{K}'_{kj}] = \operatorname*{E}_S[K(x_i,x_k)K'(x_k,x_j)] = \operatorname*{E}_{x'}[\operatorname*{E}_x[K]\operatorname*{E}_x[K']]. \tag{12}$$

Let us start with the expression of $\mathrm{E}[K_cK'_c]$:

$$\mathrm{E}[K_cK'_c] = \mathrm{E}\left[\left(K - \operatorname*{E}_{x'}[K] - \operatorname*{E}_x[K] + \mathrm{E}[K]\right)\left(K' - \operatorname*{E}_{x'}[K'] - \operatorname*{E}_x[K'] + \mathrm{E}[K']\right)\right]. \tag{13}$$

After expanding this expression, applying the expectation to each of the terms, and simplifying, we obtain:

$$\mathrm{E}[K_cK'_c] = \mathrm{E}[KK'] - 2\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right] + \mathrm{E}[K]\,\mathrm{E}[K'].$$

$\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F$ can be expanded and written more explicitly as follows:

$$\begin{aligned}\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F &= \langle \mathbf{K}, \mathbf{K}'\rangle_F - \frac{\mathbf{1}^\top\mathbf{K}\mathbf{K}'\mathbf{1}}{m} - \frac{\mathbf{1}^\top\mathbf{K}'\mathbf{K}\mathbf{1}}{m} + \frac{\mathbf{1}^\top\mathbf{K}'\mathbf{1}\mathbf{1}^\top\mathbf{K}\mathbf{1}}{m^2}\\ &= \sum_{i,j=1}^m \mathbf{K}_{ij}\mathbf{K}'_{ij} - \frac{1}{m}\sum_{i,j,k=1}^m (\mathbf{K}_{ik}\mathbf{K}'_{kj} + \mathbf{K}'_{ik}\mathbf{K}_{kj}) + \frac{1}{m^2}(\sum_{i,j=1}^m \mathbf{K}_{ij})(\sum_{i,j=1}^m \mathbf{K}'_{ij}).\end{aligned}$$

To take the expectation of this expression, we use the observations (11) and (12) and similar identities. Counting terms of each kind, leads to the following expression of the expectation:

$$\begin{aligned}\operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right] = &\left[\frac{m(m-1)}{m^2} - \frac{2m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[KK']\\ &+ \left[\frac{-2m(m-1)(m-2)}{m^3} + \frac{2m(m-1)(m-2)}{m^4}\right]\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right]\\ &+ \left[\frac{m(m-1)(m-2)(m-3)}{m^4}\right]\mathrm{E}[K]\,\mathrm{E}[K']\\ &+ \left[\frac{m}{m^2} - \frac{2m}{m^3} + \frac{m}{m^4}\right]\operatorname*{E}_x[K(x,x)K'(x,x)]\\ &+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x)K'(x,x')]\\ &+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x')K'(x,x)]\\ &+ \left[\frac{m(m-1)}{m^4}\right]\operatorname*{E}_x[K(x,x)]\operatorname*{E}_x[K'(x,x)]\\ &+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\operatorname*{E}_x[K(x,x)]\,\mathrm{E}[K']\\ &+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\mathrm{E}[K]\operatorname*{E}_x[K'(x,x)].\end{aligned}$$

Taking the difference with the expression of $\mathrm{E}[K_cK'_c]$ (Equation 13), using the fact that terms of form $\mathrm{E}_x[K(x,x)K'(x,x)]$ and other similar ones are all bounded by $R^2R'^2$ and collecting the terms gives

$$\begin{aligned}\left|\mathrm{E}[K_cK'_c] - \operatorname*{E}_S\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c\rangle_F}{m^2}\right]\right| \leq &\frac{3m^2-4m+2}{m^3}\mathrm{E}[KK'] - 2\frac{4m^2-5m+2}{m^3}\operatorname*{E}_x\left[\operatorname*{E}_{x'}[K]\operatorname*{E}_{x'}[K']\right]\\ &+ \frac{6m^2-11m+6}{m^3}\mathrm{E}[K]\,\mathrm{E}[K'] + \gamma,\end{aligned}$$

with $|\gamma| \leq \frac{m-1}{m^2} R^2 R'^2$. Using again the fact that the expectations are bounded by $R^2 R'^2$ yields

$$\left| \mathrm{E}[K_c K'_c] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right] \right| \leq \left[\frac{3}{m} + \frac{8}{m} + \frac{6}{m} + \frac{1}{m}\right] R^2 R'^2 \leq \frac{18}{m} R^2 R'^2,$$

and concludes the proof. ■

Appendix B. Stability Bounds for Alignment Maximization Algorithm

Lemma 20 *Let $\boldsymbol{\mu} = \mathbf{v}/\|\mathbf{v}\|$ and $\boldsymbol{\mu}' = \mathbf{v}'/\|\mathbf{v}'\|$. Then, the following identity holds for $\Delta\boldsymbol{\mu} = \boldsymbol{\mu}' - \boldsymbol{\mu}$:*

$$\Delta\boldsymbol{\mu} = \left[\frac{\Delta\mathbf{v}}{\|\mathbf{v}'\|} - \frac{(\Delta\mathbf{v})^\top (\mathbf{v} + \mathbf{v}')\mathbf{v}}{\|\mathbf{v}\|\|\mathbf{v}'\|(\|\mathbf{v}\| + \|\mathbf{v}'\|)}\right].$$

Proof By definition of $\Delta\boldsymbol{\mu}$, we can write

$$\Delta\boldsymbol{\mu} = \Delta\left(\frac{\mathbf{v}}{\|\mathbf{v}\|}\right) = \left[\frac{\mathbf{v}' - \mathbf{v}}{\|\mathbf{v}'\|} - \frac{\mathbf{v}\|\mathbf{v}'\| - \mathbf{v}\|\mathbf{v}\|}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right] = \left[\frac{\Delta\mathbf{v}}{\|\mathbf{v}'\|} - \frac{\mathbf{v}\Delta(\|\mathbf{v}\|)}{\|\mathbf{v}\|\|\mathbf{v}'\|}\right]. \tag{14}$$

Observe that:

$$\Delta(\|\mathbf{v}\|) = \frac{\Delta(\|\mathbf{v}\|^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\Delta(\sum_{i=1}^p v_i^2)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{\sum_{i=1}^p \Delta(v_i)(v_i + v'_i)}{\|\mathbf{v}\| + \|\mathbf{v}'\|} = \frac{(\Delta\mathbf{v})^\top(\mathbf{v} + \mathbf{v}')}{\|\mathbf{v}\| + \|\mathbf{v}'\|}.$$

Plugging in this expression in (14) yields the statement of the lemma. ■

Consider the minimization (7) shown by Proposition 9 to provide the solution of the alignment maximization problem for a convex combination. The matrix $\mathbf{M}$ and vector $\mathbf{a}$ are functions of the training sample S. To emphasize this dependency, we rewrite that optimization for a sample S as

$$\min_{\mathbf{v} \geq \mathbf{0}} F(S, \mathbf{v}), \tag{15}$$

where $F(S, \mathbf{v}) = \mathbf{v}^\top \mathbf{M} \mathbf{v} - 2\mathbf{v}^\top \mathbf{a} = \|\mathbf{v}\|^2_{\mathbf{M}} - 2\mathbf{v}^\top \mathbf{a}$. The following lemma provides a stability result for this optimization problem.

Proposition 21 *Let S and S' denote two samples of size m differing by only one point. Let $\mathbf{v}$ and $\mathbf{v}'$ be the solution of* (15)*, respectively, for sample S and S'. Then, the following inequality holds for $\Delta\mathbf{v} = \mathbf{v}' - \mathbf{v}$:*

$$\|\Delta\mathbf{v}\|^2_{\mathbf{M}} \leq \left[\Delta\mathbf{a} - (\Delta\mathbf{M})\mathbf{v}'\right]^\top \Delta\mathbf{v}.$$

Proof Since $C = \{\mathbf{v} \colon \mathbf{v} \geq 0\}$ is convex, for any $s \in [0, 1]$, $\mathbf{v} + s\Delta\mathbf{v}$ and $\mathbf{v}' - s\Delta\mathbf{v}$ are in C. Thus, by definition of $\mathbf{v}'$ and $\mathbf{v}$,

$$F(S, \mathbf{v}) \leq F(S, \mathbf{v} + s\Delta\mathbf{v}) \quad \text{and} \quad F(S', \mathbf{v}') \leq F(S', \mathbf{v}' - s\Delta\mathbf{v}).$$

Summing up these inequalities, we obtain

$$\begin{aligned} &\|\mathbf{v}\|^2_{\mathbf{M}} - \|\mathbf{v} + s\Delta\mathbf{v}\|^2_{\mathbf{M}} + \|\mathbf{v}'\|^2_{\mathbf{M}'} - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\mathbf{M}'} \\ &\qquad\qquad \leq 2\mathbf{v}^\top \mathbf{a} - 2(\mathbf{v} + s\Delta\mathbf{v})^\top \mathbf{a} + 2\mathbf{v}'^\top \mathbf{a}' - 2(\mathbf{v}' + s\Delta\mathbf{v})^\top \mathbf{a}' \\ &\qquad\qquad = -2[s\mathbf{a}^\top \Delta\mathbf{v} - s\mathbf{a}'^\top \Delta\mathbf{v}] = 2s(\Delta\mathbf{a})^\top \Delta\mathbf{v}. \end{aligned}$$

Kinematics

	unif	l2-krr	align	alignf
unif	–	1	1	1
l2-krr	0	–	1	1
align	0	0	–	1
alignf	0	0	0	–

Ionosphere

	unif	l2-krr	align	alignf
unif	–	1	1	1
l2-krr	0	–	1	1
align	0	0	–	1
alignf	0	0	0	–

German

	unif	l1-svm	align	alignf
unif	–	0	1	1
l1-svm	0	–	0	1
align	0	0	–	1
alignf	0	0	0	–

Spambase

	unif	l1-svm	align	alignf
unif	–	0	0	0
l1-svm	1	–	1	1
align	0	0	–	0
alignf	0	0	0	–

Splice

	unif	l1-svm	align	alignf
unif	–	0	0	1
l1-svm	0	–	0	0
align	0	0	–	0
alignf	0	0	0	–

Table 4: Significance tests for general kernel combination results presented in Table 2. An entry of 1 indicates that the algorithm listed in the column has a significantly better accuracy than the algorithm listed in the row.

The left-hand side of this inequality can be rewritten as follows after expansion and using the identity $\|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\mathbf{M}'} - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\mathbf{M}} = \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}$:

$$\begin{aligned} -\|s\Delta\mathbf{v}\|^2_{\mathbf{M}} - 2s\mathbf{v}^\top\mathbf{M}\Delta\mathbf{v} + \|\mathbf{v}'\|^2_{\mathbf{M}'} - \|\mathbf{v}'\|^2_{\mathbf{M}} - \|s\Delta\mathbf{v}\|^2_{\mathbf{M}} + 2s\mathbf{v}'^\top\mathbf{M}(\Delta\mathbf{v}) - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}} \\ = 2s(1-s)\|\Delta\mathbf{v}\|^2_{\mathbf{M}} + \|\mathbf{v}'\|^2_{\Delta\mathbf{M}} - \|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}. \end{aligned}$$

Then, expanding $\|\mathbf{v}' - s\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}}$ results in the final inequality

$$2s(1-s)\|\Delta\mathbf{v}\|^2_{\mathbf{M}} - s^2\|\Delta\mathbf{v}\|^2_{\Delta\mathbf{M}} + 2s\mathbf{v}'^\top(\Delta\mathbf{M})(\Delta\mathbf{v}) \leq 2s(\Delta\mathbf{a})^\top\Delta\mathbf{v}.$$

Dividing by s and setting $s = 0$ yields

$$\|\Delta\mathbf{v}\|^2_{\mathbf{M}} + \mathbf{v}'^\top(\Delta\mathbf{M})(\Delta\mathbf{v}) \leq (\Delta\mathbf{a})^\top\Delta\mathbf{v},$$

which concludes the proof of the lemma. ■

Appendix C. Significance Tests for Empirical Results

Tables 4 and 5 show the results of paired-sample one-sided T-tests for all pairs of algorithms compared across all data sets presented in Section 5 for both regression and classification. Each entry of the tables indicates whether the mean error of the algorithm listed in the column is significantly less than the mean error of the algorithm listed in the row at significance level $p = 0.1$. An entry

BOOKS

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

DVD

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	0
align	0	0	–

ELEC

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

KITCHEN

	unif	l2-krr	align
unif	–	1	1
l2-krr	0	–	1
align	0	0	–

REGRESSION

BOOKS

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

DVD

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

ELEC

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

KITCHEN

	unif	l1-svm	align
unif	–	0	1
l1-svm	1	–	1
align	0	0	–

CLASSIFICATION

Table 5: Significance tests for rank-one kernel combination results presented in Table 3. An entry of 1 indicates that the algorithm listed in the column has a significantly better accuracy then the algorithm listed in the row.

of 1 indicates a significant difference, while an entry of 0 indicates that the null hypothesis (that the errors are not significantly different) cannot be rejected.

Table 4 indicates that the `alignf` method offers significant improvement over `unif` in all data sets with the exception of spambase and significantly improves over the compared one-stage method in all data sets apart from splice. Table 5 indicates that the `align` method significantly improves over both the uniform and one-stage combination in all data sets apart from dvd in the regression setting, where improvement over `l2-krr` is not deemed significant.

References

Andreas Argyriou, Charles Micchelli, and Massimiliano Pontil. Learning convex combinations of continuously parameterized basic kernels. In *COLT*, 2005.

Andreas Argyriou, Raphael Hauser, Charles Micchelli, and Massimiliano Pontil. A DC-programming algorithm for kernel selection. In *ICML*, 2006.

Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, 2008.

Maria-Florina Balcan and Avrim Blum. On a theory of learning with similarity functions. In *ICML*, 2006.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*, 2007.

Bernhard Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, volume 5, 1992.

Olivier Bousquet and André Elisseeff. Algorithmic stability and generalization performance. In *NIPS*, 2000.

Olivier Bousquet and Daniel J. L. Herrmann. On the complexity of learning the kernel matrix. In *NIPS*, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3), 2002.

Corinna Cortes. Invited talk: Can learning kernels help performance? In *ICML*, 2009.

Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3), 1995.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning sequence kernels. In *MLSP*, 2008.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. L_2-regularization for learning kernels. In *UAI*, 2009a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning non-linear combinations of kernels. In *NIPS*, 2009b.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel methods. In *ICML*, 2010a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Generalization bounds for learning kernels. In *ICML*, 2010b.

Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the Impact of Kernel Approximation on Learning Accuracy. In *AISTATS*, 2010c.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Ensembles of kernel predictors. In *UAI*, 2011a.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Tutorial: Learning kernels. In *ICML*, 2011b.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. On kernel-target alignment. In *NIPS*, 2001.

Nello Cristianini, Jaz S. Kandola, André Elisseeff, and John Shawe-Taylor. On kernel target alignment. http://www.support-vector.net/papers/alignment_JMLR.ps, unpublished, 2002.

Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic learning theory*, 2005.

Tony Jebara. Multi-task feature and kernel selection for SVMs. In *ICML*, 2004.

Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. On the extensions of kernel alignment. technical report 120, Department of Computer Science, Univ. of London, UK, 2002a.

Jaz S. Kandola, John Shawe-Taylor, and Nello Cristianini. Optimizing kernel alignment over combinations of kernels. technical report 121, Dept. of CS, Univ. of London, UK, 2002b.

Seung-Jean Kim, Alessandro Magnani, and Stephen Boyd. Optimal kernel selection in kernel fisher discriminant analysis. In *ICML*, 2006.

Vladimir Koltchinskii and Ming Yuan. Sparse recovery in large ensembles of kernel machines. In *COLT*, 2008.

Gert Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5, 2004.

Darrin P. Lewis, Tony Jebara, and William Stafford Noble. Nonstationary kernel combination. In *ICML*, 2006.

Colin McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141, 1989.

Marina Meila. Data centering in feature space. In *AISTATS*, 2003.

Charles Micchelli and Massimiliano Pontil. Learning the kernel function via regularization. *JMLR*, 6, 2005.

Cheng Soon Ong, Alexander Smola, and Robert Williamson. Learning the kernel with hyperkernels. *JMLR*, 6, 2005.

Jean-Baptiste Pothin and Cédric Richard. Optimizing kernel alignment by data translation in feature space. In *ICASSP*, 2008.

Craig Saunders, A. Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

Nathan Srebro and Shai Ben-David. Learning bounds for support vector machines with learned kernels. In *COLT*, 2006.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In *ICML*, 2009.

Alexander Zien and Cheng Soon Ong. Multiclass multiple kernel learning. In *ICML*, 2007.