Two-Stage Learning Kernel Algorithms

Corinna Cortes   CORINNA@GOOGLE.COM

Google Research, 76 Ninth Avenue, New York, NY 10011.

Mehryar Mohri   MOHRI@CIMS.NYU.EDU

Courant Institute of Mathematical Sciences and Google Research, 251 Mercer Street, New York, NY 10012.

Afshin Rostamizadeh   ROSTAMI@CS.NYU.EDU

Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012.

Abstract

This paper examines two-stage techniques for learning kernels based on a notion of alignment. It presents a number of novel theoretical, algorithmic, and empirical results for alignment-based techniques. Our results build on previous work by Cristianini et al. (2001), but we adopt a different definition of kernel alignment and significantly extend that work in several directions: we give a novel and simple concentration bound for alignment between kernel matrices; show the existence of good predictors for kernels with high alignment, both for classification and for regression; give algorithms for learning a maximum alignment kernel by showing that the problem can be reduced to a simple QP; and report the results of extensive experiments with this alignment-based method in classification and regression tasks, which show an improvement both over the uniform combination of kernels and over other state-of-the-art learning kernel methods.

1. Introduction

Kernel-based algorithms have been used with great success in a variety of machine learning applications (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). But, the choice of the kernel, which is crucial to the success of these algorithms, has been traditionally entirely left to the user. Rather than requesting the user to select a specific kernel, learning kernel algorithms require the user only to specify a family of kernels. This family of kernels can be used by a learning algorithm to form a

combined kernel and derive an accurate predictor. This is a problem that has attracted a lot of attention recently, both from the theoretical point of view and from the algorithmic, optimization, and application perspective.

Different kernel families have been studied in the past, including hyperkernels (Ong et al., 2005), Gaussian kernel families (Micchelli & Pontil, 2005), or non-linear families (Bach, 2008; Cortes et al., 2009b). Here, we consider more specifically a convex combination of a finite number of kernels, as in much of the previous work in this area.

On the theoretical side, a number of favorable guarantees have been derived for learning kernels with convex combinations (Srebro & Ben-David, 2006; Cortes et al., 2009a), including a recent result of Cortes et al. (2010) which gives a margin bound for L1 regularization with only a logarithmic dependency on p, the number of kernels p: $R(h) \leq \widehat{R}_\rho(h) + O(\sqrt{(R^2/\rho^2)(\log p)/m})$. Here, R denotes the radius of the sphere containing the data, ρ the margin, and m the sample size.

In contrast, the results obtained for learning kernels in applications have been in general rather disappointing. In particular, achieving a performance superior to that of the uniform combination of kernels, the simplest approach requiring no additional learning, has proven to be surprisingly difficult (Cortes, 2009). Most of the techniques used in these applications for learning kernels are based on the same natural *one-stage method*, which consists of minimizing an objective function both with respect to the kernel combination parameters and the hypothesis chosen, as formulated by Lanckriet et al. (2004).

This paper explores a *two-stage* technique and algorithm for learning kernels. The first stage of this technique consists of *learning* a kernel K that is a convex combination of p kernels. The second stage consists of using K with a standard kernel-based learning algorithm such as support vector machines (SVMs) (Cortes & Vapnik, 1995) for

classification, or KRR for regression, to select a prediction hypothesis. With this two-stage method we obtain better performance than with the one-stage methods on several datasets.

Note that an alternative two-stage technique consists of first learning a prediction hypothesis h_k using each kernel K_k, and then learning the best linear combination of these hypotheses. But, such ensemble-based techniques make use of a richer hypothesis space than the one used by learning kernel algorithms such as (Lanckriet et al., 2004).

Different methods can be used to determine the convex combination parameters defining K from the training sample. A measure of similarity between the base kernels K_k, $k \in [1, p]$, and the target kernel K_Y derived from the labels can be used to determine these parameters. This can be done by using either the individual similarity of each kernel K_k with K_Y, or globally, from the similarity between convex combinations of the base kernels and K_Y. The similarities we consider are based on the natural notion of *kernel alignment* introduced by Cristianini et al. (2001), though our definition differs from the original one. We note that other measures of similarity could be used in this context. In particular, the notion of similarity suggested by Balcan & Blum (2006) could be used if it could be computed from finite samples.

We present a number of novel theoretical, algorithmic, and empirical results for the alignment-based two-stage techniques. Our results build on previous work by Cristianini et al. (2001; 2002); Kandola et al. (2002a), but we significantly extend that work in several directions.

We discuss the original definitions of kernel alignment by these authors and adopt a related but different definition (Section 2). We give a novel concentration bound showing that the difference between the alignment of two kernel matrices and the alignment of the corresponding kernel functions can be bounded by a term in $O(1/\sqrt{m})$ (Section 3). Our result is simpler and directly bounds the difference between the relevant quantities, unlike previous work. We also show the existence of good predictors for kernels with high alignment, both for classification and for regression. These results correct a technical problem in classification and extend to regression the bounds of Cristianini et al. (2001). In Section 4, we also give an algorithm for learning a maximum alignment kernel. We prove that the mixture coefficients can be obtained efficiently by solving a simple quadratic program (QP) in the case of a convex combination, and even give a closed-form solution for them in the case of an arbitrary linear combination. Finally, in Section 5, we report the results of extensive experiments with this alignment-based method 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 in all datasets. We also observe a strong correlation between the alignment achieved and performance.

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. Centering kernels

Let D be the distribution according to which training and test points are drawn. Centering a feature mapping $\Phi \colon \mathcal{X} \to H$ consists of replacing 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

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

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. 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.

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 then centered by replacing it with $\Phi(x_i) - \overline{\Phi}$, with $\overline{\Phi} = \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i)$, and the kernel matrix $\mathbf{K}$ associated to K and the sample S is centered by replacing it with $\mathbf{K}_c$ defined for all $i, j \in [1, m]$ by

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

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, $\frac{1}{m^2} \sum_{i,j=1}^{m} [\mathbf{K}_c]_{ij} = 0$.

2.2. Kernel alignment

We define the alignment of two kernel functions as follows.

Definition 1. *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*

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

In the absence of ambiguity, to abbreviate the notation, we often omit the variables over which an expectation is taken. Since $|\mathrm{E}[K_c K_c']| \leq \sqrt{\mathrm{E}[K_c^2]\,\mathrm{E}[K_c'^2]}$ 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_c and K_c' are PDS kernels. We denote by $\langle \cdot, \cdot \rangle_F$ the Frobenius product and by $\| \cdot \|_F$ the Frobenius norm.

Lemma 1. *For any two PDS kernels Q and Q', $\mathrm{E}[QQ'] \geq 0$.*

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

$$\begin{aligned}
&\mathop{\mathrm{E}}_{x,x'}[Q(x,x')Q'(x,x')] \\
&= \mathop{\mathrm{E}}_{x,x'}[\Psi(x)^\top \Psi(x') \Psi'(x')^\top \Psi'(x)] \\
&= \mathop{\mathrm{E}}_{x,x'}\big[\mathrm{Tr}[\Psi(x)^\top \Psi(x') \Psi'(x')^\top \Psi'(x)]\big] \\
&= \langle \mathop{\mathrm{E}}_{x}[\Psi(x)\Psi'(x)^\top], \mathop{\mathrm{E}}_{x'}[\Psi(x')\Psi'(x')^\top] \rangle_F = \|\mathbf{U}\|_F^2,
\end{aligned}$$

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

The following similarly defines 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 2. *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*

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

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

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

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

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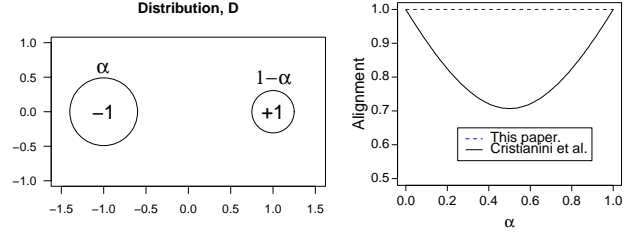

Figure 1. Alignment values computed for two different definitions of alignment: $A = [\frac{1+(1-2\alpha)^2}{2}]^{\frac{1}{2}}$ in black, $\rho = 1$ in blue. In this simple two-dimensional example, a fraction α of the points are at $(-1, 0)$ and have the label -1. The remaining points are at $(1, 0)$ and have the label $+1$.

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 y', and examine the following simple example in dimension two ($\mathcal{X} = \mathbb{R}^2$), where $K(x, x') = x \cdot x' + 1$ and 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 $+1$.

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 is never equal to one except for $\alpha = 0$ or $\alpha = 1$ and, even 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 frequently seen in experiments. Our empirical results in several tasks (not included due to lack of space) show that $\widehat{A}$ measured on the test set does not correlate well with the performance achieved. Instances of this problem have also been noticed by Meila (2003) and Pothin & Richard (2008) who have suggested various (input) data translation methods, and by Cristianini et al. (2002) who observed an issue for unbalanced data sets. The definitions we are adopting are general and require centering for both kernels K and K'.

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:

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

However, centering the kernel 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 of the kernel does not imply centering in feature space.

3. Theoretical results

This section establishes several important properties of the alignments ρ and its empirical estimate $\widehat{\rho}$: we give a concentration bound of the form $|\rho - \widehat{\rho}| \leq O(1/\sqrt{m})$, and show the existence of good prediction hypotheses both for classification and regression, in the presence of high alignment.

3.1. Concentration bound

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. give a bound on $|A' - \widehat{A}|$, where $A' \neq A$ can be defined from 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. The proof is given in a longer version of this paper.

Proposition 1. *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:*

$$\left| \frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}[K_c K'_c] \right| \leq \frac{18R^4}{m} + 24R^4 \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Theorem 1. *Under the assumptions of Proposition 1, and further assuming that the conditions of the Definitions 1-2 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:*

$$|\rho(K, K') - \widehat{\rho}(\mathbf{K}, \mathbf{K}')| \leq 18\beta \left[\frac{3}{m} + 4\sqrt{\frac{\log \frac{6}{\delta}}{2m}} \right],$$

with $\beta = \max(R^4/\mathrm{E}[K_c^2], R^4/\mathrm{E}[K_c'^2])$.

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

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

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 1 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 $\alpha = \frac{18R^4}{m} + 24R^4 \sqrt{\frac{\log \frac{6}{\delta}}{2m}}$. Using the definitions of ρ and $\widehat{\rho}$, we can write:

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

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:

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

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 $3\max(\frac{\alpha}{a}, \frac{\alpha}{a'})$. $\square$

3.2. Existence of good 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 first normalized by dividing by the standard deviation (assumed finite), thus $\mathrm{E}[y^2] = 1$. In classification, $y = \pm 1$. Let h^* denote the hypothesis defined for all $x \in \mathcal{X}$,

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

Observe that by definition of h^*, $\mathrm{E}_x[yh^*(x)] = \rho(K, K_Y)$. 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.[1]

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

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

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

$$|yh^*(x)| = |y \mathop{\mathrm{E}}_{x'}[y' K_c(x,x')]|/\sqrt{\mathrm{E}[K_c^2]}$$

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

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

$$1 - R(h^*) = \Pr[yh^*(x) \geq 0] = \mathrm{E}[\mathbf{1}_{\{yh^*(x)\geq 0\}}]$$

$$\geq \mathrm{E}[\frac{yh^*(x)}{\Gamma}\mathbf{1}_{\{yh^*(x)\geq 0\}}]$$

$$\geq \mathrm{E}[\frac{yh^*(x)}{\Gamma}] = \rho(K, K_Y)/\Gamma,$$

where $\mathbf{1}_\omega$ is the indicator variable of an event ω. $\square$

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

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

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

$$\mathop{\mathrm{E}}_x[h^{*2}(x)] = \mathop{\mathrm{E}}_x\left[\frac{\mathrm{E}_{x'}[y' K_c(x,x')]^2}{\mathrm{E}[K_c^2]}\right]$$

$$\leq \mathop{\mathrm{E}}_x\left[\frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]}\right]$$

$$= \frac{\mathrm{E}_{x'}[y'^2]\,\mathrm{E}_{x,x'}[K_c^2(x,x')]}{\mathrm{E}[K_c^2]} = \mathop{\mathrm{E}}_{x'}[y'^2] = 1.$$

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

$$\mathrm{E}[(y - h^*(x))^2] = \mathop{\mathrm{E}}_x[h^*(x)^2] + \mathop{\mathrm{E}}_x[y^2] - 2\mathop{\mathrm{E}}_x[yh^*(x)]$$

$$\leq 1 + 1 - 2\rho(K, K_Y). \quad \square$$

[1]A version of this result was presented by Cristianini et al. (2001; 2002) for the so-called Parzen window solution and non-centered kernels, but their proof implicitly relies 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 holds only if K is constant.

4. Algorithms

This section discusses two-stage algorithms for learning kernels in the form of linear combinations of p base kernels K_k, $k \in [1, p]$. In all cases, the final hypothesis learned belongs to the reproducing kernel Hilbert space associated to a kernel $K_\mu = \sum_{k=1}^p \mu_k K_k$, where the mixture weights are selected subject to the condition $\mu \geq 0$, which guarantees that K is a PDS kernel, and a condition on the norm of μ, $\|\mu\| = \Lambda > 0$, where Λ is a regularization parameter.

In the first stage, these algorithms determine the mixture weights μ. In the second stage, they train a kernel-based algorithm, e.g., SVMs for classification, or KRR for regression, in combination with the kernel $\mathbf{K}_\mu$, to learn a hypothesis h. Thus, the algorithms differ only by the first stage, where K_μ is determined, which we briefly describe.

Uniform combination (unif): this is the most straightforward method, which consists of choosing equal mixture weights, thus the kernel matrix used is $\mathbf{K}_\mu = \frac{\Lambda}{p}\sum_{k=1}^p \mathbf{K}_k$. Nevertheless, improving upon the performance of this method has been surprisingly difficult for standard (one-stage) learning kernel algorithms (Cortes, 2009).

Independent alignment-based method (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: $\mathbf{K}_\mu \propto \sum_{k=1}^p \widehat{\rho}(\mathbf{K}_k, \mathbf{K}_Y)\mathbf{K}_k$.

Alignment maximization algorithms (alignf): 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 $\mathbf{K}_\mu = \sum_{k=1}^p \mu_k \mathbf{K}_k$ and the target kernel $\mathbf{K}_Y = \mathbf{y}\mathbf{y}^\top$, as suggested 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. 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 also examine the case of a non-convex linear combination, where components of μ can be negative, and show that the problem then admits a closed-form solution. We start with this linear combination case and partially use that solution to obtain the solution of the convex combination.

4.1. Alignment maximization algorithm - linear combination

We can assume without loss of generality that the centered base kernel matrices $\mathbf{K}_{kc}$ are independent since oth-

erwise we can select an independent subset. This condition ensures that $\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F > 0$ for an arbitrary $\boldsymbol{\mu}$ and that $\widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{yy}^\top)$ is well defined (Definition 2). By the properties of centering, $\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_{Yc} \rangle_F = \langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{K}_Y \rangle_F$. Thus, since $\|\mathbf{K}_{Yc}\|_F$ does not depend on $\boldsymbol{\mu}$, alignment maximization can be written as the following optimization problem:

$$\max_{\boldsymbol{\mu} \in \mathcal{M}} \widehat{\rho}(\mathbf{K}_{\boldsymbol{\mu}}, \mathbf{yy}^\top) = \max_{\boldsymbol{\mu} \in \mathcal{M}} \frac{\langle \mathbf{K}_{\boldsymbol{\mu}_c}, \mathbf{yy}^\top \rangle_F}{\|\mathbf{K}_{\boldsymbol{\mu}_c}\|_F}, \qquad (1)$$

where $\mathcal{M} = \{\boldsymbol{\mu} : \|\boldsymbol{\mu}\|_2 = 1\}$. A similar set can be defined via norm-1 instead of norm-2. As we shall see, however, the problem can be solved in the same way in both cases. Note that, by definition of centering, $\mathbf{K}_{\boldsymbol{\mu}_c} = \mathbf{U}_m \mathbf{K}_{\boldsymbol{\mu}} \mathbf{U}_m$ with $\mathbf{U}_m = \mathbf{I} - \mathbf{11}^\top/m$, thus, $\mathbf{K}_{\boldsymbol{\mu}_c} = \sum_{k=1}^p \mu_k \mathbf{U}_m \mathbf{K}_k \mathbf{U}_m = \sum_{k=1}^p \mu_k \mathbf{K}_{kc}$. Let $\mathbf{a}$ denote the vector $(\langle \mathbf{K}_{1c}, \mathbf{yy}^\top \rangle_F, \ldots, \langle \mathbf{K}_{pc}, \mathbf{yy}^\top \rangle_F)^\top$ and $\mathbf{M}$ the matrix defined by $\mathbf{M}_{kl} = \langle \mathbf{K}_{kc}, \mathbf{K}_{lc} \rangle_F$, for $k, l \in [1, p]$. Note that since the base kernels are assumed independent, matrix $\mathbf{M}$ is invertible. Also, in view of the non-negativity of the Frobenius product of PSD matrices shown in Section 2.2, 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 $\mathbf{X} = (x_1, \ldots, x_m)^\top \in \mathbb{R}^m$,

$$\mathbf{X}^\top \mathbf{M} \mathbf{X} = \sum_{k,l=1}^m x_k x_l \operatorname{Tr}[\mathbf{K}_{kc} \mathbf{K}_{lc}] = \operatorname{Tr}\left[\sum_{k,l=1}^m x_k x_l \mathbf{K}_{kc} \mathbf{K}_{lc}\right]$$

$$= \operatorname{Tr}\left[\left(\sum_{k=1}^m x_k \mathbf{K}_{kc}\right)\left(\sum_{l=1}^m x_l \mathbf{K}_{lc}\right)\right] = \|\sum_{k=1}^m x_k \mathbf{K}_{kc}\|_F^2 \geq 0.$$

Proposition 2. *The solution $\boldsymbol{\mu}^\star$ of the optimization problem (1) is given by $\boldsymbol{\mu}^\star = \frac{\mathbf{M}^{-1}\mathbf{a}}{\|\mathbf{M}^{-1}\mathbf{a}\|}$.*

Proof. With the notation introduced, problem (1) 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

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

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

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

Therefore, the solution is

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

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

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. $\square$

4.2. Alignment maximization algorithm - convex combination

In view of the proof of Proposition 2, the alignment maximization problem with the set $\mathcal{M}' = \{\|\boldsymbol{\mu}\|_2 = 1 \wedge \boldsymbol{\mu} \geq \mathbf{0}\}$ can be written as

$$\boldsymbol{\mu}^* = \operatorname*{argmax}_{\boldsymbol{\mu} \in \mathcal{M}'} \frac{\boldsymbol{\mu}^\top \mathbf{aa}^\top \boldsymbol{\mu}}{\boldsymbol{\mu}^\top \mathbf{M} \boldsymbol{\mu}}. \qquad (2)$$

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

Proposition 3. *Let $\mathbf{v}^\star$ be the solution of the following QP:*

$$\min_{\mathbf{v} \geq 0} \mathbf{v}^\top \mathbf{M} \mathbf{v} - 2\mathbf{v}^\top \mathbf{a}. \qquad (3)$$

Then, the solution $\boldsymbol{\mu}^*$ of the alignment maximization problem (2) is given by $\boldsymbol{\mu}^\star = \mathbf{v}^\star/\|\mathbf{v}^\star\|$.

Proof. Note that the objective function of problem (2) is invariant to scaling. The constraint $\|\boldsymbol{\mu}\| = 1$ only serves to enforce $0 < \|\boldsymbol{\mu}\| < +\infty$. Thus, using the same change of variable as in the proof of Proposition 2, we can instead solve the following problem from which we can retrieve the solution via normalization:

$$\boldsymbol{\nu}^\star = \operatorname*{argmax}_{\substack{0 < \|\mathbf{M}^{-1/2}\boldsymbol{\nu}\|_2 < +\infty \\ \mathbf{M}^{-1/2}\boldsymbol{\nu} \geq \mathbf{0}}} \left[\frac{\boldsymbol{\nu}}{\|\boldsymbol{\nu}\|} \cdot (\mathbf{M}^{-1/2}\mathbf{a})\right]^2.$$

Equivalently, we can solve the following problem for any finite $\lambda > 0$:

$$\max_{\substack{\mathbf{M}^{-1/2}\mathbf{u} \geq \mathbf{0} \\ \|\mathbf{u}\| = \lambda}} \left[\mathbf{u} \cdot \mathbf{M}^{-1/2}\mathbf{a}\right]^2.$$

Observe that for $\mathbf{M}^{-1/2}\mathbf{u} \geq \mathbf{0}$ the inner product is non-negative: $\mathbf{u} \cdot \mathbf{M}^{-1/2}\mathbf{a} = \mathbf{M}^{-1/2}\mathbf{u} \cdot \mathbf{a} \geq 0$, since the entries of $\mathbf{a}$ are non-negative. Furthermore, it can be written as follows:

$$\mathbf{u} \cdot \mathbf{M}^{-1/2}\mathbf{a} = -\frac{1}{2}\|\mathbf{u} - \mathbf{M}^{-1/2}\mathbf{a}\|^2 + \frac{1}{2}\|\mathbf{u}\|^2 + \frac{1}{2}\|\mathbf{M}^{-1/2}\mathbf{a}\|^2$$

$$= -\frac{1}{2}\|\mathbf{u} - \mathbf{M}^{-1/2}\mathbf{a}\|^2 + \frac{\lambda^2}{2} + \frac{1}{2}\|\mathbf{M}^{-1/2}\mathbf{a}\|^2.$$

Thus, the problem becomes equivalent to the minimization:

$$\min_{\substack{\mathbf{M}^{-1/2}\mathbf{u} \geq \mathbf{0} \\ \|\mathbf{u}\| = \lambda}} \|\mathbf{u} - \mathbf{M}^{-1/2}\mathbf{a}\|^2. \qquad (4)$$

Now, we can omit the condition on the norm of $\mathbf{u}$ since (4) holds for arbitrary finite $\lambda > 0$ and since neither $\mathbf{u} = \mathbf{0}$ or

any infinite norm $\mathbf{u}$ can be the solution even without this condition. Thus, we can now consider instead:

$$\min_{\mathbf{M}^{-1/2}\mathbf{u} \geq \mathbf{0}} \left\| \mathbf{u} - \mathbf{M}^{-1/2}\mathbf{a} \right\|^2.$$

The change of variable $\mathbf{u} = \mathbf{M}^{1/2}\mathbf{v}$ leads to: $\min_{\mathbf{v} \geq \mathbf{0}} \left\| \mathbf{M}^{1/2}\mathbf{v} - \mathbf{M}^{-1/2}\mathbf{a} \right\|^2$. This is a standard least-square regression problem with non-negativity constraints, a simple and widely studied QP for which several families of algorithms have been designed. Expanding the terms, we obtain the equivalent problem:

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

Note that this QP problem does not require a matrix inversion of $\mathbf{M}$. Also, 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.

5. Experiments

This section compares the performance of several learning kernel algorithms for classification and regression. We compare the algorithms unif, align, and alignf, from Section 4, as well as the following one-stage algorithms:

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

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} \; 2\boldsymbol{\alpha}^\top \mathbf{1} - \boldsymbol{\alpha}^\top \mathbf{Y}^\top \mathbf{K}_{\boldsymbol{\mu}} \mathbf{Y} \boldsymbol{\alpha}$$

subject to: $\boldsymbol{\mu} \geq \mathbf{0}, \mathrm{Tr}[\mathbf{K}_{\boldsymbol{\mu}}] \leq \Lambda, \boldsymbol{\alpha}^\top \mathbf{y} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}$,

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

$$\min_{\boldsymbol{\mu}} \max_{\boldsymbol{\alpha}} \; -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{K}_{\boldsymbol{\mu}} \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}$$

subject to: $\boldsymbol{\mu} \geq \mathbf{0}, \|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_2 \leq \Lambda$,

as described in Cortes et al. (2009a). 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

Table 1. Error measures (top) and alignment values (bottom) for (A) unif, (B) one-stage l2-krr or l1-svm, (C) align and (D) alignf with kernels built from linear combinations of Gaussian base kernels. The choice of γ_0, γ_1 is listed in row labeled γ, and m is the size of the dataset used. Shown with ± 1 standard deviation (in parentheses) measured by 5-fold cross-validation.

	KINEMAT.	IONOSPH.	GERMAN	SPAMBASE	SPLICE
m	1000	351	1000	1000	1000
γ	-3, 3	-3, 3	-4, 3	-12, -7	-9, -3
A	.138(.005)	.467(.085)	25.9(1.8)	18.7(2.8)	15.2(2.2)
	.158(.013)	.242(.021)	.089(.008)	.138(.031)	.122(.011)
B	.137(.005)	.457(.085)	26.0(2.6)	20.9(2.80)	15.3(2.5)
	.155(.012)	.248(.022)	.082(.003)	.099(.024)	.105(.006)
C	.125(.004)	.445(.086)	25.5(1.5)	18.6(2.6)	15.1(2.4)
	.173(.016)	.257(.024)	.089(.008)	.140(.031)	.123(.011)
D	.115(.004)	.442(.087)	24.2(1.5)	18.0(2.4)	13.9(1.3)
	.176(.017)	.273(.030)	.093(.009)	.146(.028)	.124(.011)
	REGRESSION			CLASSIFICATION	

on the performance on the validation set, while the regularization parameters $\mathbf{C}$ and λ are fixed since only the ratios $\mathbf{C}/\Lambda$ and λ/Λ matter. The $\boldsymbol{\mu}_0$ parameter is set to zero in Section 5.1, and is chosen to be uniform in Section 5.2.

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 equal to one. We test the algorithms on several datasets taken from the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/) and Delve datasets (http://www.cs.toronto.edu/~delve/data/datasets.html).

Table 1 summarizes our results. For classification, we compare against the l1-svm method and report the misclassification percentage. For regression, we compare against the l2-krr method and report RMSE. In general, we see that performance and alignment are well correlated. In all datasets, 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.

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-1 base kernels built from the eigenvectors of a single kernel matrix (Cristianini et al., 2001). In the next section, we also examine rank-1 kernels, although not gen-

Table 2. The error measures (top) and alignment values (bottom) for kernels built with rank-1 feature based kernels on four domain sentiment analysis domains. Shown with ± 1 standard deviation as measured by 5-fold cross-validation.

	BOOKS	DVD	ELEC	KITCHEN
unif	$1.442 \pm .015$	$1.438 \pm .033$	$1.342 \pm .030$	$1.356 \pm .016$
	$.029 \pm .005$	$.029 \pm .005$	$.038 \pm .002$	$.039 \pm .006$
l2-krr	$1.414 \pm .020$	$1.420 \pm .034$	$1.318 \pm .031$	$1.332 \pm .016$
	$.031 \pm .004$	$.031 \pm .005$	$.042 \pm .003$	$.044 \pm .007$
align	$1.401 \pm .035$	$1.414 \pm .017$	$1.308 \pm .033$	$1.312 \pm .012$
	$.046 \pm .006$	$.047 \pm .005$	$.065 \pm .004$	$.076 \pm .008$

REGRESSION

	BOOKS	DVD	ELEC	KITCHEN
unif	25.8 ± 1.7	24.3 ± 1.5	18.8 ± 1.4	20.1 ± 2.0
	$.030 \pm .004$	$.030 \pm .005$	$.040 \pm .002$	$.039 \pm .007$
l1-svm	28.6 ± 1.6	29.0 ± 2.2	23.8 ± 1.9	23.8 ± 2.2
	$.029 \pm .012$	$.038 \pm .011$	$.051 \pm .004$	$.060 \pm .006$
align	24.3 ± 2.0	21.4 ± 2.0	16.6 ± 1.6	17.2 ± 2.2
	$.043 \pm .003$	$.045 \pm .005$	$.063 \pm .004$	$.070 \pm .010$

CLASSIFICATION

erated from a spectral decomposition.

5.2. Rank-1 kernel combinations

In this set of experiments we use the sentiment analysis dataset 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-1 kernels based on the 4,000 most frequent bigrams. The kth base kernel, $\mathbf{K}_k$, corresponds to the k-th bigram count $\mathbf{v}_k$, $\mathbf{K}_k = \mathbf{v}_k \mathbf{v}_k^\top$. Each base kernel is normalized to have trace 1 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 2) and also previously observed by Cortes et al. (2009a), a sparse weight vector $\boldsymbol{\mu}$ does not generally offer an improvement over the uniform combination in the rank-1 setting. Thus, we focus on the performance of align and compare it to unif and one-stage learning methods. Table 2 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. 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 theoretical, algorithmic, and empirical results for a two-stage learning kernel algorithm based on a notion of alignment. Our experiments show a consistent improvement of the performance over previous learning kernel techniques, as well as the straightforward uniform kernel combination, which has been difficult to surpass in the past. These improvements could suggest a better one-stage algorithm with a regularization term taking into account the alignment quality of each base kernel, a topic for future research.

References

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

Balcan, Maria-Florina and Blum, Avrim. On a theory of learning with similarity functions. In *ICML*, pp. 73–80, 2006.

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

Cortes, Corinna. Invited talk: Can learning kernels help performance? In *ICML 2009*, pp. 161, 2009.

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

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

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

Cortes, Corinna, Mohri, Mehryar, and Rostamizadeh, Afshin. Generalization bounds for learning kernels. In *ICML '10*, 2010.

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

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

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

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

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

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

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

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

Pothin, J.-B. and Richard, C. Optimizing kernel alignment by data translation in feature space. In *ICASSP*, 2008.

Schölkopf, Bernhard and Smola, Alex. *Learning with Kernels*. MIT Press: Cambridge, MA, 2002.

Shawe-Taylor, John and Cristianini, Nello. *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, 2004.

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

A. Proof of Proposition 1

The proof relies on a series of lemmas shown below.

Proof. By the triangle inequality and in view of Lemma 4, the following holds:

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

$$
\left| \frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} \right] \right| + \frac{18 R^4}{m}.
$$

Now, in view of Lemma 3, the application of McDiarmid's inequality to $\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}$ gives for any $\epsilon > 0$:

$$
\Pr\left[\left| \frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} - \mathrm{E}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2} \right] \right| > \epsilon \right] \leq
$$
$$
2 \exp[-2m\epsilon^2 / (24 R^4)^2].
$$

Setting δ to be equal to the right-hand side yields the statement of the proposition. $\qquad\square$

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

Lemma 2. *The following properties hold for centering kernel matrices:*

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

$$
\mathbf{K}_c = \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m} \right] \mathbf{K} \left[\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{m} \right]. \tag{5}
$$

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

$$
\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}, \mathbf{K}'_c \rangle_F = \langle \mathbf{K}_c, \mathbf{K}' \rangle_F. \tag{6}
$$

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

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

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

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'.

Lemma 3. *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:*

$$
\frac{1}{m^2} |\Delta(\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F)| \leq \frac{24 R^4}{m}.
$$

Proof. By Lemma 2, we can write:

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

The perturbation of the first term is given by

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

By the Cauchy-Schwarz inequality, for any $i, j \in [1, m]$, $|\mathbf{K}_{ij}| = |K(x_i, x_j)| \leq \sqrt{K(x_i, x_i)K(x_j, x_j)} \leq R^2$. Thus,

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

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

$$
\frac{1}{m^2}\left| \Delta\left(\frac{\mathbf{1}^\top\mathbf{K}\mathbf{K}'\mathbf{1}}{m} \right) \right| = \left| \Delta\left(\sum_{i,j,k=1}^{m} \frac{\mathbf{K}_{ik}\mathbf{K}'_{kj}}{m^3} \right) \right|
$$
$$
= \left| \Delta\left(\frac{\sum_{i,k=1}^{m} \mathbf{K}_{ik}\mathbf{K}'_{km} + \sum_{i,j \neq m} \mathbf{K}_{im}\mathbf{K}'_{mj}}{m^3} \right) + \right.
$$
$$
\left. \Delta\left(\frac{\sum_{k \neq m, j \neq m} \mathbf{K}_{mk}\mathbf{K}'_{kj}}{m^3} \right) \right|
$$
$$
\leq \frac{m^2 + m(m-1) + (m-1)^2}{m^3}(2R^4)
$$
$$
\leq \frac{3m^2 - 3m + 1}{m^3}(2R^4) \leq \frac{6R^4}{m}.
$$

Similarly, we have:

$$
\frac{1}{m^2}\left| \Delta\left(\frac{\mathbf{1}^\top\mathbf{K}'\mathbf{K}\mathbf{1}}{m} \right) \right| \leq \frac{6R^4}{m}, \tag{7}
$$

and it can be shown that

$$
\frac{1}{m^2}\left| \Delta\left(\frac{(\mathbf{1}^\top\mathbf{K}\mathbf{1})(\mathbf{1}^\top\mathbf{K}'\mathbf{1})}{m^2} \right) \right| \leq \frac{8R^4}{m}. \tag{8}
$$

Combining these last four inequalities leads directly to the statement of the lemma. $\qquad\square$

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 $\mathrm{E}[K_c K'_c]$. However, as shown by the following lemma, the estimation bias decreases at the rate $O(1/m)$.

Lemma 4. *Under the same assumptions as Lemma 3, the following bound on the difference of expectations holds:*

$$\left| \mathop{\mathrm{E}}_{x,x'}[K_c(x,x')K'_c(x,x')] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right] \right| \leq \frac{18R^4}{m}.$$

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

$$\mathop{\mathrm{E}}_{S}[\mathbf{K}_{ij}\mathbf{K}'_{ij}] = \mathop{\mathrm{E}}_{S}[K(x_i,x_j)K'(x_i,x_j)] = \mathrm{E}[KK'], \quad (9)$$

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:

$$\mathop{\mathrm{E}}_{S}[\mathbf{K}_{ik}\mathbf{K}'_{kj}] = \mathop{\mathrm{E}}_{S}[K(x_i,x_k)K'(x_k,x_j)] = \mathop{\mathrm{E}}_{x',x}[\mathop{\mathrm{E}}_{x}[K]\mathop{\mathrm{E}}[K']]. \quad (10)$$

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

$$\mathrm{E}[K_c K'_c] = \mathrm{E}\left[\left(K - \mathop{\mathrm{E}}_{x'}[K] - \mathop{\mathrm{E}}_{x}[K] + \mathrm{E}[K]\right)\right.$$
$$\left.\left(K' - \mathop{\mathrm{E}}_{x'}[K'] - \mathop{\mathrm{E}}_{x}[K'] + \mathrm{E}[K']\right)\right]. \quad (11)$$

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

$$\mathrm{E}[K_c K'_c] = \mathrm{E}[KK'] - 2\mathop{\mathrm{E}}_{x}\left[\mathop{\mathrm{E}}_{x'}[K]\mathop{\mathrm{E}}_{x'}[K']\right] + \mathrm{E}[K]\mathrm{E}[K'].$$

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

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

To take the expectation of this expression, we shall use the observations (9) and (10) and similar identities. Counting terms of each kind, leads to the following expression of the expectation:

$$\mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right]$$
$$= \left[\frac{m(m-1)}{m^2} - \frac{2m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[KK']$$
$$+ \left[\frac{-2m(m-1)(m-2)}{m^3} + \frac{2m(m-1)(m-2)}{m^4}\right]$$
$$\mathop{\mathrm{E}}_{x}\left[\mathop{\mathrm{E}}_{x'}[K]\mathop{\mathrm{E}}_{x'}[K']\right]$$
$$+ \left[\frac{m(m-1)(m-2)(m-3)}{m^4}\right]\mathrm{E}[K]\mathrm{E}[K']$$
$$+ \left[\frac{m}{m^2} - \frac{2m}{m^3} + \frac{m}{m^4}\right]\mathop{\mathrm{E}}_{x}[K(x,x)K'(x,x)]$$
$$+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x)K'(x,x')]$$
$$+ \left[\frac{-m(m-1)}{m^3} + \frac{2m(m-1)}{m^4}\right]\mathrm{E}[K(x,x')K'(x,x)]$$
$$+ \left[\frac{m(m-1)}{m^4}\right]\mathop{\mathrm{E}}_{x}[K(x,x)]\mathop{\mathrm{E}}_{x}[K'(x,x)]$$
$$+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\mathop{\mathrm{E}}_{x}[K(x,x)]\mathrm{E}[K']$$
$$+ \left[\frac{m(m-1)(m-2)}{m^4}\right]\mathrm{E}[K]\mathop{\mathrm{E}}_{x}[K'(x,x)].$$

Taking the difference with the expression of $\mathrm{E}[K_c K'_c]$ (Equation 11), 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^4 and collecting the terms gives

$$\left| \mathrm{E}[K_c K'_c] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right] \right| \leq \frac{3m^2 - 4m + 2}{m^3}\mathrm{E}[KK']$$
$$- 2\frac{4m^2 - 5m + 2}{m^3}\mathop{\mathrm{E}}_{x}\left[\mathop{\mathrm{E}}_{x'}[K]\mathop{\mathrm{E}}_{x'}[K']\right]$$
$$+ \frac{6m^2 - 11m + 6}{m^3}\mathrm{E}[K]\mathrm{E}[K'] + \gamma,$$

with $|\gamma| \leq \frac{m-1}{m^2}R^4$. Using again the fact that the expectations are bounded by R^4 yields

$$\left| \mathrm{E}[K_c K'_c] - \mathop{\mathrm{E}}_{S}\left[\frac{\langle \mathbf{K}_c, \mathbf{K}'_c \rangle_F}{m^2}\right] \right| \leq \left[\frac{3}{m} + \frac{8}{m} + \frac{6}{m} + \frac{1}{m}\right]R^4$$
$$\leq \frac{18}{m}R^4,$$

and concludes the proof. $\square$