# Optimizing Kernel Alignment over Combinations of Kernels

**Jaz Kandola** JAZ@CS.RHUL.AC.UK
**John Shawe-Taylor** JOHN@CS.RHUL.AC.UK
Department of Computer Science, Royal Holloway, University of London, UK

**Nello Cristianini** NELLO@CS.BERKELEY.EDU
BIOwulf Technologies, Berkeley and University of California, Berkeley, USA

## Abstract

Alignment has recently been proposed as a method for measuring the degree of agreement between a kernel and a learning task (Cristianini et al., 2001). Previous approaches to optimizing kernel alignment have required the eigendecomposition of the kernel matrix which can be computationally prohibitive especially for large kernel matrices. In this paper we propose a general method for optimizing alignment over a linear combination of kernels. We apply the approach to give both transductive and inductive algorithms based on the Incomplete Cholesky factorization of the kernel matrix. The Incomplete Cholesky factorization is equivalent to performing a Gram-Schmidt orthogonalization of the training points in the feature space. The alignment optimization method adapts the feature space to increase its training set alignment. Regularization is required to ensure this alignment is also retained for the test set. Both theoretical and experimental evidence is given to show that improving the alignment leads to a reduction in generalization error of standard classifiers.

## 1. Introduction

Kernel-based learning methods (Cristianini & Shawe-Taylor, 2000) are based around the notion of a "kernel matrix" or Gram matrix, that can informally be regarded as a pairwise similarity matrix between all pairs of points in the dataset. Of course it is necessary to define a notion of similarity, and kernel methods use the inner product between two points in a suitable feature space, information that can often be obtained with little computational cost even for very high dimensional spaces. The resulting matrix is symmetric and positive semi-definite (its eigenvalues are always non-negative reals) and consequently can always be written as $K = \sum_i \lambda_i v_i v_i'$ where $v_i$ and $\lambda_i \geq 0$ are the eigenvectors and eigenvalues of $K$.

All the information needed by the learning machine, both coming from the data and coming from the similarity measure, is contained in the Gram matrix. Its properties reflect the relative positions of the points in the feature space. For example, it is obvious that a kernel matrix $K = I$, where $I$ is the identity, would correspond to having all points orthogonal to each other in the feature space, and hence there would be no useful notion of similarity (every point is similar to every other point in the same way). Any split of the data would be as good as another, and there would be no clear way to assign a new point to a given class.

For classification problems, as those considered in (Cristianini et. al., 2001), if one already knew a priori the specific classification target function to be learned $y(x)$, the optimal kernel function would be $K_{ij} = (y(x_i), y(x_j))$. If the labels vector is denoted by $y$, the corresponding kernel matrix is $K = yy'$ and has rank 1. The structure of the paper is as follows. In section 2 we give a formal definition of alignment. Section 3 studies the properties of positive semi-definite matrices and introduces a novel characterisation of kernels. It further develops a general algorithm for optimizing the alignment over linear combinations of kernels. This paper then applies these techniques to develop new alignment optimisation algorithms based on the Gram-Schmidt orthogonalization procedure for transduction and induction in Section 4. Section 5 presents results that show, optimizing the alignment of the projection of the data into a 1-dimensional subspace is equivalent to performing Ridge Regression (RR). Furthermore, the alignment of the full kernel matrix lower bounds its projected value. Together these results show that optimising

the alignment followed by a Ridge Regression optimization gives a well founded model selection strategy. Experimental results are presented in section 6. The approach adopted in this paper is closely related to that presented by (Lanckriet et al., 2002). Here we optimize the alignment, and subsequently its projection in a ridge regression style algorithm. (Lanckriet et al., 2002) however consider an alternative optimization approach based on optimizing the margin using semi-definite programming (SDP).

## 2. Kernel Alignment

By measuring the similarity of this kernel with the kernel at hand - on the training set - one can assess the degree of fitness. The measure of similarity that we propose is referred to as *kernel alignment*.

**Definition 1 Alignment** *The (empirical) alignment of a kernel $k_1$ with a kernel $k_2$ with respect to the sample $S$ is the quantity*

$$A(S, k_1, k_2) = \frac{\langle K_1, K_2 \rangle_F}{\sqrt{\langle K_1, K_1 \rangle_F \langle K_2, K_2 \rangle_F}},$$

*where $K_i$ is the kernel matrix for the sample $S$ using kernel $k_i$.*

This can also be viewed as the cosine of the angle between two bi-dimensional vectors $K_1$ and $K_2$, representing the Gram matrices. If we consider $K_2 = yy'$, where $y$ is the vector of outputs for the sample, then

$$A(S, K, yy') = \frac{\langle K, yy' \rangle_F}{\sqrt{\langle K, K \rangle_F \langle yy', yy' \rangle_F}} = \frac{y'Ky}{m\|K\|_F} \quad (1)$$

The alignment has several convenient properties (Cristianini et al., 2001). It can be efficiently computed before any training of the kernel machine takes place, and based only on training data information; it is sharply concentrated around its expected value, and hence its empirical value is stable with respect to different splits of the data; and importantly, it has been shown that if the kernel is very aligned with the target information, then there exists a separation of the data with a low bound on the generalization error. All these observations together mean that it is possible to measure and optimize this quantity based on training set information, and achieve better generalization performance on the test set in a transductive setting.

In (Cristianini et al., 2001) the parameterized class of kernels determined by the equation

$$\hat{K} = \sum_i \alpha_i v_i v_i' \quad (2)$$

were considered, where $v_i$ are the eigenvectors of the matrix $K$ and the optimization problem was solved for finding the optimal $\alpha$, that is the parameters that maximize the alignment of the combined kernel with the available labels. Given $K = \sum_i \alpha_i v_i v_i'$, the alignment is optimized by choosing

$$\alpha_i \propto \langle v_i, y \rangle^2 \quad (3)$$

giving an overall alignment of

$$A = \sqrt{\frac{\sum_i \langle v_i, y \rangle^4}{\langle yy', yy' \rangle_F}}. \quad (4)$$

A transductive algorithm can be designed to take advantage of this, by optimizing alignment with part of the dataset, and in doing so it will adapt the Gram matrix also for the unlabeled part.

**Data** : Construct kernel matrix (K), and $yy'$
$[V, D] = eigendecomp(K)$;
**for** *maximum number of runs* **do**
  Split data into training ($I$) and test set ($J$);
  **for** *$n$ = 1:length of Kernel* **do**
    $T = V(:, n) \cdot V(:, n)'$;
    $a(n) = (V(I, n)' \cdot y(I))^2 / (V(I, n)' \cdot V(I, n))^2$;
    $G = G + a(n) \cdot T$;
  **endfor**
  Compute alignment for K and G;
  Train SVM & Parzen windows with K and G;
**endfor**

Algorithm 1: A Transductive Alignment Algorithm

## 3. Properties of Positive Definite Matrices

Positive definite matrices are characterized by the following well-known proposition. We include a proof for completeness.

**Proposition 2** *Let $K$ be a symmetric matrix. Then $K$ is positive semi-definite if and only if*

$$\langle K, G \rangle_F \geq 0,$$

*for all positive semi-definite matrices $G$.*

**Proof**: Let $K = \sum_{i=1}^m \lambda_i v_i v_i'$ be the eigenvalue decomposition of $K$ and $G = \sum_{j=1}^m \mu_j u_j u_j'$ that of a general $G$. We have

$$\begin{aligned} \langle K, G \rangle_F &= \sum_{ij} \lambda_i \mu_j \langle v_i v_i', u_j u_j' \rangle_F \\ &= \sum_{ij} \lambda_i \mu_j (v_i' u_j)^2 \geq 0, \end{aligned}$$

if $\lambda_i, \mu_j \geq 0$. Conversely, if $\lambda_i < 0$ for some $i$, choose $u_i = v_i$, $i = 1, \ldots, m$, and $\mu_j = 0$, for $j \neq i$, $\mu_i = 1$. It follows that $\langle K, G \rangle_F < 0$. ■

Hence, positive semi-definite matrices form a cone of bi-dimensional vectors. The proposition also indicates an alternative characterization of positive semi-definiteness.

We now consider a general linear combination of kernels

$$K(\alpha) = \sum_{k=1}^{T} \alpha_k K_k,$$

and study the problem of choosing $\alpha$ to optimism the alignment of $K(\alpha)$ to some given target vector $y$. Define $A(\alpha)$ as

$$\begin{aligned} A(\alpha) &= A(S, K(\alpha), yy') \\ &= \frac{\sum_k y' K_k y}{m\sqrt{\sum_{kl} \alpha_k \alpha_l \langle K_k, K_l \rangle_F}}. \end{aligned}$$

Hence, to maximize the alignment we maximize $A(\alpha)$ subject to the constraint $\alpha_i \geq 0$. This constraint can be weakened in some cases if we are prepared to solve a semi-definite program (Lanckriet et al., 2002). For the purposes of this paper we will see that it will be sufficient to restrict ourselves to the case $\alpha_i \geq 0$. Hence, we must solve

$$\begin{aligned} \max_{\alpha} \quad & A(\alpha) \\ \text{subject to} \quad & \alpha_i \geq 0, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \max_{\alpha} \quad & \sum_k \alpha_k y' K_k y \\ \text{subject to} \quad & \sum_{kl} \alpha_k \alpha_l \langle K_k, K_l \rangle_F = C \\ & \alpha_i \geq 0. \end{aligned}$$

Applying the Lagrange multiplier method, we obtain,

$$\max_{\alpha} \sum_k \alpha_k y' K_k y - \lambda \left( \sum_{kl} \alpha_k \alpha_l \langle K_k, K_l \rangle_F - C \right)$$
$$\text{subject to} \quad \alpha_i \geq 0.$$

Varying $C$ leads to different values of $\mu$. Since the alignment is invariant to rescaling $\alpha$, we can choose $\mu = 1$, fixing some value for the denominator and minimizing the numerator. Hence, we obtain the optimization problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_k \alpha_k y' K_k y - \sum_{kl} \alpha_k \alpha_l \langle K_k, K_l \rangle_F \\ \text{subject to} \quad & \alpha_i \geq 0. \end{aligned}$$

The second problem that arises is that if we do not constrain $\|\alpha\|$, the kernel can overfit its alignment to the training set, making its 'generalization' alignment on the test set poor. Hence, in exactly the same way as constraining the norm in for example Ridge Regression prevents overfitting, constraining $\|\alpha\|$ prevents overaligning. Including this and using the Lagrange multiplier method again, we obtain

$$\begin{aligned} \max_{\alpha} \quad & \sum_k \alpha_k y' K_k y - \sum_{kl} \alpha_k \alpha_l \langle K_k, K_l \rangle_F - \lambda \sum_k \alpha_k^2 \\ = \quad & \sum_k \alpha_k y' K_k y - \sum_{kl} \alpha_k \alpha_l \left( \langle K_k, K_l \rangle_F + \lambda \delta_{kl} \right) \\ \text{subj to} \quad & \alpha_i \geq 0. \end{aligned}$$

The resulting optimization has a very similar form to that of the Support Vector Machine optimization problem. The first linear term is a sum of positive factors of the $\alpha_k$, that is $y' K_k y \geq 0$, in place of the values 1 in the SVM case. The second part is a quadratic function, which as in the SVM case, is convex since

$$\begin{aligned} \sum_{kl} u_k u_l (\langle K_k, K_l \rangle_F &+ \lambda \delta_{kl}) \\ &= \left\langle \sum_k u_k K_k, \sum_l u_l K_l \right\rangle + \sum_k u_k u_k \\ &= \| \sum_k u_k K_k \|_F^2 + \|u\|^2 \geq 0. \end{aligned}$$

Note also that by Proposition 2 the entries in the Hessian are all positive. Hence, we can solve for $\alpha$ using the standard quadratic programming methods used in the SVM optimization. Furthermore, we can expect the $\alpha$ vector to be sparse, that is that only a subset of the kernels will be included in the optimal combination. In the next section we apply the approach to a particular combination obtained by performing a Gram-Schmidt orthogonalization in the feature space.

## 4. Gram-Schmidt Optimization

In kernel based methods, large datasets pose significant problems since the number of basis functions required for an optimal solution can equal the number of data samples (Smola & Schölkopf, 2000). A number of methods have been proposed for obtaining a low rank matrices such that the Frobenius norm is minimized (Smola & Schölkopf, 2000; Fine & Scheinberg, 2000; Williams & Seeger, 2000).

A number of sparse greedy approximation methods have been proposed to construct a reduced representation of the kernel matrix. Smola and Scholkopf (2000) argue that if many irrelevant basis functions

are eliminated the solution on the subset of basis functions may be close to optimal. Fine and Scheinberg (2000) consider an alternative approach, based on the product form Cholesky factorization, that constructs a low rank matrix approximation to the kernel matrix. (Williams & Seeger, 2000) show that an approximation to the eigendecomposition of the Gram matrix can be computed by the Nyström method. This is achieved by carrying out an eigendecomposition on a smaller dataset and then expanding the results back up to the full dataset size.

In this work, an approximation strategy, based on the Gram-Schmidt decomposition in the feature space is considered. This algorithm is equivalent to the incomplete Cholesky decomposition of the kernel matrix used by (Bach & Jordan, 2001) for kernel ICA. The projection is built up as the span of a subset of (the projections of) a set of $k$ training examples. These are selected by performing a Gram-Schmidt orthogonalization of the training vectors in the feature space. Hence, once a vector is selected the remaining training points are transformed to become orthogonal to it. The next vector selected is the one with the largest residual norm. The whole transformation is performed in the feature space using the kernel mapping to represent the vectors obtained. The method is parametrised by the number of dimensions $T$ selected.

Figure 1 shows the pseudo-code for the Gram-Schmidt/incomplete Cholesky factorization algorithm. If we now create the vectors $v_k$, $k = 1, \ldots, T$, by setting

**Require:** A kernel $k$, training set $\{(d_1, y_1), \ldots, (d_n, y_n)\}$ and number $T$
  **for** $i = 1$ to $n$ **do**
    $norm[i] = k(d_i, d_i)$;
  **end for**
  **for** $j = 1$ to $T$ **do**
    $i_j = argmax_i(norm[i])$;
    $index[j] = i_j$;
    $size[j] = \sqrt{norm[i_j]}$;
    **for** $i = 1$ to $n$ **do**
      $feat[i,j] = \frac{\left(k(d_i, d_{i_j}) - \sum_{t=1}^{j-1} feat[i,t] * feat[i_j,t]\right)}{size[j]}$;
      $norm[i] = norm[i] - feat[i,j] * feat[i,j]$;
    **end for**
  **end for**
  return $feat[i,j]$ as the $j$-th feature of input $i$;

*Figure 1.* The Gram-Schmidt Algorithm

$$v_{ki} = \frac{feat[i,k]}{qrt\sum_{i'} feat[i',k]^2},$$

then we can express the approximate reconstruction of $K$ as

$$K \approx \sum_{k=1}^{T} d_k v_k v_k',$$

where $d_k = \sum_{i'} feat[i',k]^2$, since

$$\begin{aligned} K_{ij} &\approx \sum_{k=1}^{T} feat[i,k] feat[j,k] \\ &= \sum_{k=1}^{T} v_{ki} v_{kj} \sum_{i'} feat[i',k]^2. \end{aligned}$$

Hence, the Gram-Schmidt decomposition of $K$ returns the matrix $V$ and a diagonal matrix $D$ with diagonal entries $d_k$.

This approximation is constructed by choosing a sequence of exemplar training examples that most completely span the space and projecting the data into those directions. It therefore suggests that we explore the linear combination of kernels,

$$K(\alpha) = \sum_{k=1}^{T} \alpha_k v_k v_k',$$

and apply the methods of optimization developed in the previous section.

Note that while in combining a general set of kernels it may happen that a combination with negative coefficients is still positive semi-definite, we show that here for $K(\alpha)$ to be positive semi-definite we must have $\alpha_k \geq 0$ for all $k$. This follows from the fact that if the order in which the vectors are orthogonalized is given by $i_1, i_2, \ldots, i_T$, then the entries $v_{ki_\ell} = 0$, for $\ell < k$ and $v_{ki_k} \neq 0$ (assuming the $k$-th feature vector has non-zero residual). Hence, the matrix $V$ with the vectors $v_k$ as rows has rank $T$. Now assume $\alpha_{\bar{k}} < 0$. Since $V$ has full row rank, there is a vector $u$ such that $Vu = e_{\bar{k}}$, the $\bar{k}$-th unit vector. Now

$$u'K(\alpha)u = \sum_k \alpha_k u' v_k v_k' u = \alpha_{\bar{k}} < 0.$$

Hence, we obtain the optimally aligned linear combination by solving the optimization,

$$\begin{aligned} \max_{\alpha} \ & \sum_k \alpha_k (y' v_k)^2 - \sum_{kl} \alpha_k \alpha_l \left( (v_k' v_l)^2 + \lambda \delta_{kl} \right) \\ \text{subj to} \ & \alpha_i \geq 0. \end{aligned}$$

We refer to this optimization as *gramkernel* (returning the $\alpha$ in the variable $X$) in the description of the overall procedure given in Algorithm 2.

**Data** : Construct kernel matrix ($K$), and $yy'$
Perform Gram-Schmidt decomposition on $K$ to obtain the matrix $V$ and diagonal matrix $D$;
Set $\lambda$ to a constant value;
**for** *maximum number of runs* **do**
  Split data into training ($I$) and test set ($J$);
  $M = \lambda \cdot Id$;
  $b = (V(I,:)' * y(I))^2$;
  $M = M + (V(I,:)' \cdot V(I,:))^2$;
  $[nsv, X] = gramkernel(M, b)$;
  $G = V \cdot diag(X) \cdot V'$;
  Compute alignment for $K$ and $G$;
  Train SVM & Parzen windows with K and G;
**endfor**

Algorithm 2: Optimization of Alignment using Gram-Schmidt

An inductive alignment algorithm similar to that based on a complete eigenvalue decomposition (see companion paper (Kandola et al., 2002)) can also be developed by modifying the Gram-Schmidt procedure. We first apply the Gram-Schmidt optimization routine to the training data only and then use the following iteration to compute the features of a new point $d$: for $j = 1, \ldots, T$,

$$f[j] = \frac{\left(K(d, d_{i_j}) - \sum_{t=1}^{j-1} f[t] \cdot \sqrt{X} \cdot feat[i_j, t]\right)}{size[j]};$$

where $K$ denotes the kernel matrix and $\sqrt{X}$ the square root of the diagonal matrix $X$ obtained from the Gram kernel optimization routine.

## 5. Non-margin based Gram-Schmidt Optimization

In a companion paper (Kandola et al., 2002) we prove the following two theorems that relate the optimization of the projected alignment to the Ridge Regression optimization:

$$\min_w L(w) = \lambda \langle w, w \rangle + \sum_{i=1}^{m} \left(\langle w, \mathbf{x}_i \rangle - y_i\right)^2. \qquad (5)$$

**Theorem 3** *Let $X$ be a feature/example matrix expressed in a possibly kernel-defined feature space. The solution of the optimization*

$$\text{argmax}_{w:\|w\|\leq 1} A(S, X'ww'X, yy')$$

*gives the weight vector that solves the Ridge Regression problem (5) with the regularization parameter $\lambda = 0$.*

**Theorem 4** *Let $X$ be a feature/example matrix expressed in a possibly kernel-defined feature space. The solution of the optimization*

$$w_* = \text{argmax}_{w:\|w\|\leq 1} A(S, X'ww'X, yy')$$

*satisfies*

$$A(S, X'w_*w_*'X, yy') \geq A(S, X'X, yy').$$

Together the theorems show that optimizing the alignment decreases a lower bound on the objective of the Ridge Regression optimization. This suggests that optimizing the alignment will lead to better generalization performance of two norm error bound classifiers. We report experiments in the next section with Parzen windows estimators (that were proven to have good performance if the alignment is high in (Cristianini et al., 2001), Ridge Regression and standard Support Vector Machines.

## 6. Experiments

To demonstrate the performance of the algorithms presented we have used two binary classification datasets. The ionosphere dataset[1] which contains 34 inputs, a single binary output and 351 datapoints. The Wisconsin breast cancer dataset was obtained from the University of Wisconsin hospitals. It contains nine integer valued inputs, a single binary output (benign or malignant tumours) and 699 datapoints.

Three learning algorithms were implemented. A Parzen window estimator, a support vector classifier (SVC) and a kernel ridge regressor (treating the binary labels $+1$, $-1$ as real targets). A 10-fold procedure was used to find the optimal values for the capacity control parameters 'C'. The SVC was trained ten times using a range of values of 'C', and the value that gave the lowest mean error (and associated standard deviation) on the test dataset was chosen as the optimal value. Having selected the optimal value of 'C', the SVC was re-trained ten times using ten random data splits. A similar procedure was used to find the optimal Ridge Regression parameter $\lambda$. A linear kernel was used for training for both of the datasets. The $K$ matrices are before adaption, while the $G$ matrices are after optimization using the transductive and inductive alignment algorithms. The index represents the percentage of training points.

Tables 1 and 2 present the results of applying the transduction and inductive alignment algorithms to the ionosphere dataset. It is apparent that the training alignment increases for the matrix G across all data

[1] available from the UCI data repository

*Table 1.* Ionosphere dataset - alignment values, SVC, Parzen window (PW) and Ridge Regression (RR) error for a linear kernel over 10 runs using transductive Gram-Schmidt.

| | Train Align | Test Align | SVC Error | PW Error | RR Error |
|---|---|---|---|---|---|
| $K_{80}$ | 0.244 (0.012) | 0.258 (0.046) | 0.320 (0.052) | 0.279 (0.042) | 0.240 (0.048) |
| $G_{80}$ | 0.342 (0.012) | 0.350 (0.089) | 0.214 (0.030) | 0.206 (0.024) | 0.207 (0.039) |
| $K_{50}$ | 0.224 (0.029) | 0.272 (0.036) | 0.297 (0.036) | 0.262 (0.033) | 0.244 (0.039) |
| $G_{50}$ | 0.360 (0.028) | 0.309 (0.034) | 0.209 (0.016) | 0.205 (0.017) | 0.203 (0.015) |
| $K_{20}$ | 0.254 (0.032) | 0.245 (0.008) | 0.310 (0.013) | 0.281 (0.020) | 0.279 (0.016) |
| $G_{20}$ | 0.316 (0.031) | 0.286 (0.016) | 0.231 (0.021) | 0.223 (0.024) | 0.222 (0.020) |

*Table 2.* Ionosphere dataset - alignment values, SVC and Parzen window (PW) and Ridge Regression (RR) error for a linear kernel over 10 runs using inductive Gram-Schmidt.

| | Train Align | Test Align | SVC Error | PW Error | RR Error |
|---|---|---|---|---|---|
| $K_{80}$ | 0.243 (0.013) | 0.261 (0.053) | 0.323 (0.047) | 0.281 (0.037) | 0.253 (0.033) |
| $G_{80}$ | 0.311 (0.013) | 0.285 (0.052) | 0.251 (0.053) | 0.230 (0.046) | 0.230 (0.040) |
| $K_{50}$ | 0.227 (0.033) | 0.270 (0.040) | 0.295 (0.034) | 0.259 (0.032) | 0.248 (0.034) |
| $G_{50}$ | 0.303 (0.040) | 0.260 (0.039) | 0.239 (0.041) | 0.223 (0.044) | 0.204 (0.031) |
| $K_{20}$ | 0.266 (0.066) | 0.243 (0.016) | 0.314 (0.017) | 0.286 (0.014) | 0.275 (0.035) |
| $G_{20}$ | 0.375 (0.064) | 0.109 (0.017) | 0.259 (0.058) | 0.260 (0.055) | 0.243 (0.066) |

partitions. A similar affect is observed for the testset alignment. There is also a reduction in the SVC, PW and RR mean generalization error over ten runs for all of data partitions considered using the transductive Gram-Schmidt algorithm. Overall, these results indicate that the optimization of the alignment on the training set increases its value by more than the sum of the standard deviations. From the concentration of the alignment (see (Cristianini et al., 2001)) this improvement is maintained in the alignment measured on the test dataset using a linear kernel.

Table 2 presents the results from the inductive alignment algorithm. It is apparent that the training alignment increases for the matrix G across all data partitions, although the increase is not as large as that for the transductive results presented in table 1. A similar affect is observed for the testset alignment, where again the alignment on the testset does not increase as much as for the transductive algorithm. These observations are entirely in accordance with the inductive algorithm. Whilst the SVC mean generalisation error for the $K$ matrices (all partitions) are similar in value to their transductive counterparts (see table 1) the SVC error for the $G$ matrices (all partions) is higher than the transductive values. This implies that the inductive algorithm when using a SVC gives worse performance. From the observation that the alignment values (for the $G$) matrix and the motivation for the inductive Gram-Schmidt (see earlier) this is expected. A similar effect is also observed for the PW and RR mean generalization errors. Overall, these results indicate that the optimization of the alignment on the training set increases its value by more than the sum of the standard deviations. From the concentration of the alignment (see (Cristianini et al., 2001)) this improvement is maintained in the alignment measured on the test dataset using a linear kernel.

Tables 3 and 4 present the results of applying the transduction and inductive alignment algorithms to the ionosphere dataset. It is apparent that the training alignment increases for the matrix G across all data partitions. A similar affect is observed for the testset alignment. There is also a reduction in the SVC, PW and RR mean generalization error over ten runs for all of data partitions considered using the transductive Gram-Schmidt algorithm. Overall, these results indicate that the optimization of the alignment on the training set increases its value by more than the sum of the standard deviations. From the concentration of the alignment (see (Cristianini et al., 2001)) this improvement is maintained in the alignment measured on the test dataset using a linear kernel.

*Table 3.* Breast dataset - alignment values, SVC and Parzen Window (PW) and Ridge Regression (RR) error for a linear kernel over 10 runs using transductive Gram-Schmidt.

| | Train Align | Test Align | SVC Error | PW Error | RR Error |
|---|---|---|---|---|---|
| $K_{80}$ | 0.112 (0.007) | 0.137 (0.031) | 0.336 (0.024) | 0.222 (0.034) | 0.336 (0.023) |
| $G_{80}$ | 0.251 (0.006) | 0.294 (0.037) | 0.247 (0.030) | 0.131 (0.032) | 0.244 (0.032) |
| $K_{50}$ | 0.120 (0.019) | 0.115 (0.020) | 0.353 (0.017) | 0.250 (0.021) | 0.356 (0.017) |
| $G_{50}$ | 0.269 (0.019) | 0.245 (0.023) | 0.262 (0.021) | 0.139 (0.019) | 0.259 (0.021) |
| $K_{20}$ | 0.116 (0.040) | 0.117 (0.010) | 0.349 (0.008) | 0.242 (0.012) | 0.349 (0.008) |
| $G_{20}$ | 0.259 (0.039) | 0.242 (0.010) | 0.267 (0.022) | 0.146 (0.009) | 0.266 (0.021) |

*Table 4.* Breast dataset - alignment values, SVC and Parzen Window (PW) and Ridge Regression (RR) error for a linear kernel over 10 runs using inductive Gram-Schmidt.

| | Train Align | Test Align | SVC Error | PW Error | RR Error |
|---|---|---|---|---|---|
| $K_{80}$ | 0.079 (0.008) | 0.080 (0.034) | 0.239 (0.107) | 0.653 (0.045) | 0.355 (0.027) |
| $G_{80}$ | 0.311 (0.008) | 0.321 (0.031) | 0.115 (0.023) | 0.133 (0.027) | 0.167 (0.039) |
| $K_{50}$ | 0.089 (0.018) | 0.070 (0.016) | 0.240 (0.156) | 0.663 (0.021) | 0.361 (0.018) |
| $G_{50}$ | 0.312 (0.017) | 0.308 (0.016) | 0.115 (0.009) | 0.134 (0.013) | 0.204 (0.058) |
| $K_{20}$ | 0.081 (0.035) | 0.079 (0.010) | 0.188 (0.087) | 0.649 (0.011) | 0.347 (0.008) |
| $G_{20}$ | 0.328 (0.034) | 0.295 (0.009) | 0.139 (0.014) | 0.142 (0.012) | 0.254 (0.069) |

Table 4 presents the results from the inductive alignment algorithm. It is apparent that the training alignment increases for the matrix G across all data partitions, although the increase is not as large as that for the transductive results presented in table 3. A similar affect is observed for the testset alignment, where again the alignment on the testset does not increase as much as for the transductive algorithm. These observations are entirely in accordance with the inductive algorithm. Whilst the SVC mean generalisation error for the $K$ matrices (all partitions) are similar in value to their transductive counterparts (see table 1) the SVC error for the $G$ matrices (all partions) is higher than the transductive values. This implies that the inductive algorithm when using a SVC gives worse performance. This effect is in contrast to that observed for the ionosphere dataset. This observation will be investigated in future work using a range of kernel functions. Overall, these results indicate that the optimization of the alignment on the training set increases its value by more than the sum of the standard deviations. From the concentration of the alignment (see (Cristianini et al., 2001)) this improvement is maintained in the alignment measured on the test dataset using a linear kernel.

## 7. Discussion & Conclusions

The problem of assessing the quality of a kernel is central to the theory of kernel-machines, and deeply related to the problem of model/feature selection as a whole. Being able to quantify this property is an important step towards effective algorithms for kernel selection, combination and adaptation. Previous approaches to optimizing kernel alignment have required the full eigendecomposition of the kernel matrix which can be computationally prohibitive especially for large kernel matrices. In this paper we demonstrated a general method for optimizing alignment over a linear combination of kernels. The approach we developed has been extended to give both transductive and inductive algorithms based on the Incomplete Cholesky factorization of the kernel matrix. The method is based upon the incomplete Cholesky factorization, which as we argue in the paper is equivalent to performing a Gram-Schmidt orthogonalization of the training points in the feature space. The alignment optimization method adapts the feature space to increase its training set alignment. Regularization is required to ensure this alignment is also retained for the test set, and ensures that a sparse solution is obtained. In this paper we provided both theoretical and experimental evidence to show that improving the alignment

leads to a reduction in generalization error of standard classifiers. From the tables of results presented in section 6, the alignment increases on the training and the test datasets. There is also an associated performance increase as denoted by measures such as SVC error, RR error and PW error.

The computational cost of performing an eigenvalue decomposition on a kernel matrix can be prohibitive for large kernel matrices. The examples considered in this paper were of moderate size and as such computational cost was kept to a minimum, however there is no reason why larger datasets cannot be considered. The performance of the algorithms will also be evaluated on high noise datasets. These tasks are left for future work. In a companion paper (Kandola et al., 2002) we extended the notion of kernel alignment to two other learning problems: regression and classification with uneven datatsets. This work used a complete eigenvalue decomposition of the kernel matrix, as such the method proposed in this paper will be tested (making the appropriate modifications to the rank 1 matrix (see ((Kandola et al., 2002)) for regression and uneven datasets. Recent work by (Lanckriet et al., 2002) has also used semi-definate programming to learn the kernel matrix from a set of data. It would be interesting to compare the performance of this approach with that of kernel alignment presented here and in (Cristianini et al., 2001).

Theoretically, we should explore the connections between high alignment and good generalization in larger classes of learning machines, and its relations with the luckiness framework (Shawe-Taylor et al., 1998), and the notion of stability. More general quality measures can be designed (basically any kernel between Gram matrices could be used), so some work will be devoted to exploring some possible options. Other forms of kernel combination and adaptation will be studied with the tool of alignment maximization.

## Acknowledgments

We would like to acknowledge the financial support of EPSRC Grant No. GR/N08575, EU Project KerMIT, No. IST-2000-25341 and the Neurocolt working group No. 27150.

## References

Bach, F., & Jordan, M. (2001). *Kernel independent components analysis* (Technical Report). University of California, Berkeley.

Cristianini, N., Kandola, J., Eliseff, A., & Shawe-Taylor, J. (2001). On kernel target alignment. *Submitted to Journal of Machine Learning Research.*

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines.* Cambridge University Press.

Fine, S., & Scheinberg, K. (2000). *Efficient svm training using low-rank kernel representation* (Technical Report). IBM TJ Watson Research Center (RC 21911).

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002). On the extensions of kernel alignment. *Submitted to International Conference on Machine Learning 2002* (p. Available from http://www.neurocolt.org).

Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. (2002). Learning the kernel matrix with semidefinate programming. *Submitted to International Conference on Machine Learning 2002.*

Poggio, T., & Girosi, F. (1997). *Regularization networks and support vector machines* (Technical Report). AI Memo. M.I.T.

Shawe-Taylor, J., Bartlett, P., Williamson, B., & Anthony, M. (1998). Structural risk minimization over data dependent hierarchies. *IEEE Information Technology.*

Smola, A. (1998). *Learning with kernels.* Doctoral dissertation, GMD-First.

Smola, A., & Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. *Proc. 17th International Conf. on Machine Learning* (pp. 911–918). Morgan Kaufmann, San Francisco, CA.

Williams, C., & Seeger, M. (2000). Using the nystrom method to speed up kernel machines. *Advances in Neural Information Processing Systems 13.* MIT Press.