On the Extensions of Kernel Alignment

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

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

Abstract

In this paper we address the problem of measuring the degree of agreement between a kernel and a learning task. The quantity that we use to capture this notion is alignment (Cristianini et al., 2001a). We motivate its theoretical properties, and derive a series of algorithms for adapting a kernel in two important machine learning problems: regression and classification with uneven datasets. We also propose a novel inductive algorithm within the framework of kernel alignment that can be used for kernel combination and kernel selection. The algorithms presented have been tested on both artificial and real-world datasets.

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 a dataset. 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 in 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 (since 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., 2001a), 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 alignment between this 'ideal' matrix and the kernel matrix is used to guide the adaptation of the kernel. The structure of this paper is as follows, in section 2 we give a formal definition of alignment. This paper is concerned with extending the notion of alignment to regression (section 3) and datasets in which the labels vector contains an uneven number of positive and negative examples (section 4) that are commonplace in many real world applications, for example text processing. Section 5 presents a novel induction algorithm that can be used for kernel target alignment. Experimental results, using both artificial and publicly available datasets, are presented in section 6.

2. Kernel Alignment

By measuring the similarity of the kernel (K_{ij}) 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* (Cristianini et al., 2001a).

Definition 1 Alignment *The (empirical) alignment of a kernel k_1 with a kernel k_2 with respect to the sam-*

ple 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 been shown to possess several convenient properties. 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, 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 (Cristianini et al., 2001a).

We defined the parameterized class of kernels determined by this equation:

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

and consider the optimization problem of finding the optimal α, that is the parameters that maximize the alignment of the combined kernel with the available labels based on an eigenvalue decomposition of the untransformed kernel. Given $K = \sum_i \alpha_i v_i v_i'$, the alignment can be written as

$$A = \frac{\langle K, yy' \rangle}{m\sqrt{\sum_{ij} \alpha_i \alpha_j \langle v_i v_i', v_j v_j' \rangle}} \quad (3)$$

From the orthonormality of the v_i and since $\langle vv', uu' \rangle = \langle v, u \rangle^2$ we can write:

$$A = \frac{\sum_i \alpha_i \langle v_i, y \rangle^2}{\sqrt{\langle yy', yy' \rangle}\sqrt{\sum_i \alpha_i^2}} \quad (4)$$

Hence we have the following optimization problem: maximize

$$W(\alpha) = \sum_i \alpha_i \langle v_i, y \rangle^2 - \lambda(\sum_i \alpha_i^2 - 1) \quad (5)$$

and hence $\alpha_i \propto \langle v_i, y \rangle^2$. This gives the overall alignment:

$$A = \sqrt{\frac{\sum_i \langle v_i, y \rangle^4}{\langle yy', yy' \rangle_F}} \quad (6)$$

A transductive algorithm can be designed to take advantage of this, by optimizing alignment with the labeled part of the dataset, and in doing so it will adapt the Gram matrix also for the unlabeled part. This algorithm is summarized by the following pseudo-code:

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

The complete eigendecomposition of the kernel matrix is an expensive computational step, and should be avoided for large kernel matrices. In a companion paper (Kandola et al., 2002), we present an approximation strategy to the full eigenvalue decomposition, based on the Gram-Schmidt decomposition. A similar approach to unsupervised learning is described by (Smola, 1998), and has also been used by (Bach & Jordan, 2001)for kernel independent components analysis (k-ICA).

Section 3 extends the theory of alignment to the case of regression providing a definition of alignment for this case together with a novel justification for why improving alignment will lead to better performance in the regression case.

Section 4 considers the case of uneven datasets as a natural extension of the classification case considered in (Cristianini et al., 2001a). Section 5 presents a novel induction algorithm that can be used for kernel target alignment, while Section 6 presents experimental results for all of the methods presented. We finish with a discussion and conclusions.

3. Kernel Alignment for Regression

The problem of regression is to approximate an unknown function from the observation of a limited se-

quence of (typically) noise corrupted input/output data pairs. More formally, consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, drawn from an unknown probability distribution, where $\mathbf{x}_i \in \mathbb{R}^n$ represents a set of inputs, $y_i \in \mathbb{R}$ represents a single output, and m represents the number of training examples. The empirical modelling problem is to discover an underlying mapping $\mathbf{x} \to y$ that is consistent with the dataset $\mathcal{D}$. The regression function is learnt from a training set, and its performance can be measured using an independent test set (Poggio & Girosi, 1997).

The first algorithm we give is a method to improve the alignment between a kernel and a fixed set target variables by acting on its eigenvalues. This algorithm performs transduction, and provides a nonparametric way to perform kernel selection, that does not require us to specify a family of kernel functions, but directly acts on the entries of the kernel matrix. To apply this transductive algorithm for the case of regression, the rank 1 matrix yy' needs to be modified using the following transformation,

$$y_i = y_i - \bar{y} \tag{7}$$

where $\bar{y}$ represents the mean over the training set of the target values.

For the case of classification the use of alignment was motivated using two facts. Firstly, that the alignment measure is concentrated around its expected value. This suggests that if we optimize its value on the training set, we can expect to see corresponding increases in the testing set alignment. This expectation was verified for the classification case. The proof of concentration made no special use of the fact that the labels were binary, and so the regression alignment is also concentrated provided the range of the output values is bounded (proof omitted).

The second observation for the case of classification was that if the value of the alignment is high, then a Parzen window estimator will give good generalisation. This justified why adapting a kernel to improve its alignment with the target on the training set should result in better generalisation performance. This argument cannot be applied in the regression case. We will therefore now present a more complex analysis suggesting why improving the alignment for regression will improve generalisation.

The key result will be that optimizing the alignment of a 1-dimensional linear projection of the data is equivalent to performing ridge regression, where the value of the alignment corresponds to the objective of the ridge regression optimization. Furthermore the alignment of the kernel matrix provides a lower bound for the projected alignment. Hence, optimizing the alignment of the kernel decreases the upper bound for the ridge regression objective,

$$\min_w L(w) = \lambda\langle w, w\rangle + \sum_{i=1}^m \left(\langle w, \mathbf{x}_i\rangle - y_i\right)^2, \tag{8}$$

that forms the adaptable part of an upper bound on the generalisation error (Cristianini and Shawe-Taylor, 2000).

Theorem 2 *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 (8) with the regularization parameter $\lambda = 0$.

Proof: First observe that

$$\begin{aligned}\text{argmax}_{w:\|w\|\leq 1}\, & A(S, X'ww'X, yy') \\ &= \text{argmax}_w \frac{\langle X'ww'X, yy'\rangle_F}{m\|X'ww'X\|_F} \\ &= \frac{1}{m}\text{argmax}_w \frac{(w'Xy)^2}{w'XX'w},\end{aligned}$$

where we have implicitly observed the invariance under rescaling of w. If we now consider optimizing the square root of the numerator with the denominator constrained to a fixed value and then introduce Lagrange multipliers we obtain the problem,

$$\text{argmax}_w w'Xy - \mu(w'XX'w - C).$$

Varying μ will correspond to obtaining different values for the constrained denominator. For every such (C, μ) pair the optimization minimises $w'Xy$ and hence also $(w'Xy)^2$. Hence, since the result is invariant to rescaling w we can choose $\mu = 1$, giving

$$\text{argmax}_w w'Xy - w'XX'w,$$

which is just the negative of the Ridge Regression optimization (8) for $\lambda = 0$. ■

The next step is to show that the projected alignment is lower bounded by the alignment of the whole matrix.

Theorem 3 *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

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

Proof: Without loss of generality we can take w_* lying in the space spanned by the columns of X. First consider creating an orthonormal basis of the space spanned by the columns of X,

$$w_* = w_1, w_2, \ldots, w_m.$$

We can now write

$$I = \sum_{i=1}^{m} w_i w_i',$$

where I is the perpendicular projection matrix onto the space spanned by the columns of X. Now observe that

$$\begin{aligned} y'X'Xy &= y'X'IXy \\ &= \sum_{i=1}^{m} y'X'w_i w_i' Xy \\ &= \sum_{i=1}^{m} (y'X'w_i)^2 \end{aligned}$$

Similarly,

$$\begin{aligned} \|X'X\|_F^2 &= \|X'IX\|_F^2 \\ &= \left\langle \sum_i X'w_i w_i' X, \sum_j X'w_j w_j' X \right\rangle_F \\ &= \left(\sum_{i=1}^{m} w_i' X X' w_i \right)^2 \end{aligned}$$

Taking $\theta = mA(S, X'w_* w_*' X, yy')$, we have

$$(y'X'w_i)^2 \leq \theta w_i' X X' w_i$$

for all i. Hence,

$$\begin{aligned} y'X'Xy &= \sum_{i=1}^{m} (y'X'w_i)^2 \leq \theta \sum_{i=1}^{m} w_i' X X' w_i \\ &= \theta \|X'X\|_F \end{aligned}$$

giving

$$\begin{aligned} A(S, X'X, yy') &= \frac{y'X'Xy}{m\|X'X\|_F} \leq \frac{\theta}{m} \\ &= A(S, X'w_* w_*' X, yy'), \end{aligned}$$

as required. ■

4. Kernel Alignment for Uneven Datasets

Uneven datasets, i.e. datasets that have an unequal number of class labels, are commonplace in many real world applications including text processing and machine vision. Consider the problem of document classification based on a particular query. It is not unreasonable to expect that a large number of documents do not match a particular query. In the justification of kernel alignment in its relation to the performance of a Parzen windows estimator, there was an implicit assumption that there are an equal number of positive and negative class labels as equal weights are given to positive and negative examples. Hence, to apply the Parzen window argument to uneven datasets the rank 1 matrix yy' needs to be modified using the following transformation:

$$y_i = \begin{cases} \frac{1}{n_+}, & \text{if } i \text{ is positive} \\ -\frac{1}{n_-}, & \text{otherwise} \end{cases} \quad (9)$$

where n_+ and n_- represents the number of positive and negative labels in the dataset respectively. This gives a slightly modified definition of alignment. The proof of concentration will still hold provided that the number of positive and negative examples remains $O(m)$, while the generalisation bound will now be for the standard Parzen window estimator with unequal weights for positive and negative examples.

5. Inductive Kernel Alignment

An inductive algorithm for kernel alignment can also be considered. The transductive algorithms considered in (Cristianini et al., 2001a) and for this paper have relied upon the eigenvalue decomposition of the full kernel matrix constructed from training and test data points. When reassembled a complete kernel matrix for the entire set of data is obtained. We now describe how we can implement an analogous inductive procedure.

The dataset needs to be randomly split into a training and test set and the kernel matrix constructed using the training data only. An eigenvalue decomposition of this kernel matrix can be written as: $K = V\Lambda V'$, where Λ is a diagonal matrix. The effect of this decomposition is to find the sequence of subspaces of the feature space that capture the greatest variance of the data. We now reweight those directions to optimism the alignment of the training set kernel matrix to the labels using the same method described in Section 2 and detailed in Algorithm 1. The difference is that we now project new data into the subspace of the feature space spanned by the eigenvectors using the principal axes as a coordinate system. We then rescale each coordinate using the scaling obtained in Algorithm 1 before using the resulting feature vector to compute inner products in the transformed space. Pseudo-Matlab

code for this procedure is given in Algorithm 2.

Data : Construct kernel matrix (K), and yy'
for *maximum number of runs* **do**
  Split data into training (I) and test set (J);
  [V,R] = eigendecomp(K(I));
  Threshold small eigenvalues;
  for n = *1:number of eigenvalues* **do**
    $a(n) = (V(:,n)' \cdot y(I))^2 / (V(:,n)' \cdot V(:,n))^2$;
  endfor
  $G(I,I) = V \cdot diag(a) \cdot V'$;
  $G(I,J) = V \cdot R^{-1} \cdot diag(a) \cdot V' \cdot K(I,J)$;
  $G(J,J) = K(J,I) \cdot V \cdot diag(a) \cdot R^{-2} \cdot V' \cdot K(I,J)$;
  Compute alignment for K and G;
  Train SVM & Parzen window with K and G;
endfor

Algorithm 2: An Inductive Alignment Algorithm

As was the case for the transductive algorithm, the complete eigendecomposition of the kernel matrix is an expensive computational step. In a companion paper (Kandola et al., 2002), an inductive approximation strategy based on the Gram-Schmidt decomposition is presented.

6. Experiments

To demonstrate the performance of the transductive alignment algorithm for regression and uneven datasets, a range of artificial and publicly available datasets were considered. Two different learning algorithms were implemented for the uneven datasets. A Parzen window estimator and a support vector classifier (SVC). In the regression case we implemented Ridge Regression (RR) as motivated by the analysis of Section 3. A 10-fold procedure was used to find the optimal value for the capacity control parameter 'C'. The SVC was trained ten times using a range of values of 'C', and the value which gave the lowest mean error (and associated standard deviation) on the test set was chosen as the optimal value. Having selected the optimal 'C' parameter, the SVC was re-trained ten times using ten random data splits.

A similar procedure was used to select the Ridge Regression parameter λ. The inductive alignment algorithm was also tested on artificial and publicly available datasets. The alignment algorithm for uneven datasets was tested on two datasets. The first of these was an artifically generated dataset consisting of 10 input variables and 100 datapoints where the inputs were drawn randomly from a Gaussian distribution with zero mean and unit variance, and a single output that consisted of an uneven number of target labels (65 positive labels and 35 negative labels). A linear kernel was used for training. The results are presented in table 1. The K matrices are before adaption, while the G matrices are after optimization using the transductive alignment algorithm. The index represents the percentage of training points.

The second dataset considered was the Medline1033 dataset commonly used in text processing (Cristianini et al., 2001b). This dataset contains 1033 documents and 30 queries obtained from the national library of medicine. In this work we focus on query20. A Bag of Words kernel was used (Joachims, 1998). Stop words and punctuation were removed from the documents and the Porter stemmer was applied to the words. The terms in the documents were weighted according to a variant of the $tfidf$ scheme. It is given by $\log(1+tf) * \log(m/df)$, where tf represents the term frequency, df is used for the document frequency and m is the total number of documents.

The results are presented in table 2. The K matrices are before adaption, while the G matrices are after optimization using the transductive alignment algorithm. The index represents the percentage of training points.

From tables 1 and 2 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 mean generalisation error, and the PW error for all of the training sets. It is interesting to note that in all cases the performance of the SVC algorithm exceeds that of the PW method. Both tables 1 and 2 also quote "F1" error values. The F1 measure is a popular statistic used in the information retrieval community for comparing performance of algorithms typically on uneven data. A detailed definition can be found in (Baeza-Yates & Ribeiro-Neto, 2001). This value is bounded between 0 and 1, where 1 represents optimal algorithm performance. For both datasets the F1 value increases across all data partions. 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., 2001a)) this improvement is maintained in the alignment measured on the test set using both a linear and Bag of Words kernel.

In order to test the performance of the alignment algorithm for regression two datasets were considered. The first of these was the an artificial dataset - the sinc function dataset which was modelled using a Gaussian kernel.

Table 1. Uneven: Toy dataset - alignment values, SVC and PW test error together with F1 values (obtained from SVC) for a linear kernel over 10 runs.

	Train Align	Test Align	SVC Error	PW Error	F1 (SVC)
K_{80}	0.009 (0.006)	0.080 (0.072)	0.567 (0.176)	0.617 (0.153)	0.696 (0.021)
G_{80}	0.085 (0.005)	0.218 (0.072)	0.450 (0.087)	0.600 (0.101)	0.720 (0.008)
K_{50}	0.027 (0.035)	0.022 (0.027)	0.493 (0.153)	0.613 (0.064)	0.587 (0.068)
G_{50}	0.104 (0.037)	0.104 (0.022)	0.460 (0.080)	0.593 (0.046)	0.605 (0.005)
K_{20}	0.062 (0.067)	0.007 (0.003)	0.546 (0.142)	0.642 (0.026)	0.756 (0.012)
G_{20}	0.187 (0.066)	0.074 (0.029)	0.533 (0.138)	0.638 (0.025)	0.759 (0.026)

Table 2. Uneven: Medline dataset - alignment values, SVC and PW test error together with F1 values (obtained from SVC) for a Bag of Words Kernel over 10 runs.

	Train Align	Test Align	SVC Error	PW Error	F1 (SVC)
K_{80}	0.103 (0.008)	0.096 (0.020)	0.357 (0.109)	0.963 (0.014)	0.472 (0.001)
G_{80}	0.141 (0.009)	0.110 (0.015)	0.183 (0.078)	0.916 (0.012)	0.481 (0.001)
K_{50}	0.112 (0.023)	0.089 (0.021)	0.381 (0.208)	0.964 (0.010)	0.603 (0.014)
G_{50}	0.175 (0.028)	0.094 (0.020)	0.139 (0.032)	0.956 (0.009)	0.615 (0.012)
K_{20}	0.099 (0.012)	0.093 (0.003)	0.404 (0.228)	0.962 (0.003)	0.427 (0.177)
G_{20}	0.105 (0.014)	0.100 (0.004)	0.358 (0.222)	0.957 (0.007)	0.441 (0.019)

Table 3. Regression: Sinc function dataset - alignment values (with asccociated standard deviations) and RR error for a Gaussian kernel over 10 runs.

	Train Align	Test Align	RR Error
K_{80}	0.002 (0.001)	0.039 (0.013)	0.163 (0.121)
G_{80}	0.253 (0.020)	0.191 (0.083)	0.119 (0.051)
K_{50}	0.011 (0.016)	0.007 (0.006)	0.166 (0.063)
G_{50}	0.274 (0.052)	0.141 (0.030)	0.123 (0.017)
K_{20}	0.002 (0.001)	0.008 (0.006)	0.196 (0.074)
G_{20}	0.368 (0.021)	0.086 (0.034)	0.136 (0.009)

Table 4. Regression: AMPG dataset - alignment values and SVM error for a linear kernel over 10 runs.

	Train Align	Test Align	RR Error
K_{80}	0.531 (0.015)	0.521 (0.062)	18.23 (3.19)
G_{80}	0.574 (0.013)	0.560 (0.049)	7.89 (1.18)
K_{50}	0.534 (0.055)	0.524 (0.056)	16.58 (2.35)
G_{50}	0.590 (0.054)	0.539 (0.055)	7.55 (0.69)
K_{20}	0.491 (0.026)	0.538 (0.006)	18.57 (4.35)
G_{20}	0.490 (0.045)	0.466 (0.009)	9.12 (3.12)

Table 3 represents the alignment for the training and test datasets and the associated RR generalisation error for the artificial sinc function dataset. The second dataset to be considered was the automobile miles per gallon (AMPG) dataset that contains the miles traveled, per gallon of fuel consumed, for various cars. The input variables measure six characteristics of a car; the number of cylinders (discrete), displacement, horsepower, weight, acceleration and model year (discrete). The goal is to discover a relationship between the AMPG and the cars' characteristics. After removing a small number of entries with missing values from the original dataset 353 datapoints remain.

Table 4 represents the alignment for the training and test datasets and the associated SVC generalisation error for the AMPG dataset. From table 3 it is apparent that the training alignment increases for the matrix G across all data partitions. A similar affect is observed for the test alignment. There is also a reduction in the RR mean generalisation error for all of the training sets. 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. For the AMPG dataset a similar trend is observed. Overall for both datasets there is a significant decrease in the RR errors for both datasets. Future work will assess the performance of the regression algorithm on high noise datasets.

The inductive algorithm was tested on the artificial uneven dataset and the Medline dataset both of which where described earlier.

Tables 5 and 6 present the results from the inductive alignment algorithm. From both tables 5 and 6 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 mean generalisation error for all of the training sets. From the concentration of the alignment this improvement is maintained in the alignment measured on the test set using both a linear and Bag of Words kernel. Comparing the results obtained from applying the transduction alignment algorithm to the artificial and medline datasets (see tables 1 and 2) we can note very similar behaviour of the two algorithms. There is consistent improvement in the train and testset alignment, together with improved SVC, PW and F1 error measures. The error values obtained from applying both the inductive and the transductive are as expected similar and can be be considered to show the merits of the methods discussed.

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. In this paper we addressed the problem of measuring the degree of agreement between a kernel and two learning tasks. We extended the notion of kernel alignment originally presented in (Cristianini et al., 2001a). Alignment for regression analysis and classification with uneven datasets was motivated and demonstrated. A novel inductive algorithm within the framework of kernel alignment that can be used for kernel combination and kernel selection was also presented. All of the algorithms were tested on artificial and publicly available datasets with good performance.

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 RR error and PW error. For the case of uneven datasets, the F1 statistic, that is used extensively in the information retrieval processing community, was used as a performance measure. For the datasets considered, there was an associated increase in F1 values for the aligned matrix G.

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 small to moderate size and as such computational cost was kept to a minimum. For larger kernel matrices, that arise typically with many real world datasets, this method would be prohibitive. In a companion paper we have proposed a faster approach based on performing Gram-Schmidt optimization in the kernel defined feature space (Kandola et al., 2002) and it would be interesting to compare the performance of this approach. The performance of the algorithms will also be evaluated on high noise datasets. These tasks are left for future work. 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., 2001a).

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.

Table 5. Toy dataset: alignment values, SVC and PW test error and F1 values (obtained from SVC) for a linear kernel over 10 runs using the inductive algorithm.

	Train Align	Test Align	SVC Error	PW Error	F1 (SVC)
K_{80}	0.006 (0.003)	0.062 (0.076)	0.565 (0.170)	0.665 (0.097)	0.556 (0.012)
G_{80}	0.198 (0.004)	0.102 (0.024)	0.535 (0.091)	0.540 (0.091)	0.750 (0.001)
K_{50}	0.011 (0.005)	0.012 (0.005)	0.564 (0.155)	0.656 (0.038)	0.629 (0.063)
G_{50}	0.249 (0.004)	0.044 (0.004)	0.522 (0.050)	0.524 (0.046)	0.681 (0.010)
K_{20}	0.091 (0.073)	0.008 (0.003)	0.496 (0.138)	0.644 (0.001)	0.658 (0.012)
G_{20}	0.375 (0.072)	0.027 (0.004)	0.491 (0.056)	0.519 (0.040)	0.714 (0.101)

Table 6. Medline dataset: alignment values and SVC error for a Bag of Words kernel over 10 runs using the inductive algorithm.

	Train Align	Test Align	SVC Error	PW Error	F1
K_{80}	0.098 (0.006)	0.109 (0.015)	0.342 (0.081)	0.960 (0.010)	0.442 (0.018)
G_{80}	0.157 (0.006)	0.153 (0.013)	0.248 (0.042)	0.251 (0.045)	0.564 (0.005)
K_{50}	0.104 (0.012)	0.093 (0.011)	0.394 (0.150)	0.964 (0.006)	0.448 (0.021)
G_{50}	0.161 (0.011)	0.129 (0.012)	0.266 (0.039)	0.269 (0.039)	0.529 (0.010)
K_{20}	0.110 (0.028)	0.097 (0.096)	0.428 (0.296)	0.963 (0.004)	0.427 (0.052)
G_{20}	0.148 (0.025)	0.129 (0.010)	0.309 (0.074)	0.337 (0.079)	0.444 (0.012)

References

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

Baeza-Yates, R., & Ribeiro-Neto, B. (2001). *Modern information retrieval*. Addison-Wesley.

Cristianini, N., Kandola, J., Eliseff, A., & Shawe-Taylor, J. (2001a). On kernel target alignment. *Submitted to Journal of Machine Learning Research*, Available from http://www.neurocolt.org.

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

Cristianini, N., Shawe-Taylor, J., & Lodhi, H. (2001b). Latent semantic kernels. *Journal of Intelligent Information Systems, 18(2/3)*, 127–152.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the European Conference on Machine Learning* (pp. 137–142).

Kandola, J., Shawe-Taylor, J., & Cristianini, N. (2002). Optimizing kernel alignment over linear combinations of kernels. *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.