Refining Kernels for Regression and Uneven Classification Problems

Jaz Kandola and John Shawe-Taylor
Dept. Computer Science
Royal Holloway, University of London
Surrey TW20 0EX. UK

Abstract

Kernel alignment has recently been proposed as a method for measuring the degree of agreement between a kernel and a classification learning task. In this paper we extend the notion of kernel alignment to two other common learning problems: regression and classification with uneven data. We present a modified definition of alignment together with a novel theoretical justification for why improving alignment will lead to better performance in the regression case. Experimental evidence is provided to show that improving the alignment leads to a reduction in generalization error of standard regressors and classifiers.

1 Introduction

Kernel based methods are increasingly being used for data modelling because of their conceptual simplicity and good performance on many tasks (see for example [3]). However, the kernel function is often chosen using trial-and-error heuristics, so that a quantitative measure of agreement is important from both a theoretical and practical point of view. Kernel alignment has recently been proposed as a method for measuring the degree of agreement between a kernel and a classification task [1]. This paper extends kernel alignment to two other machine learning problems: regression and classification with uneven datasets. A novel inductive kernel alignment optimization algorithm is also presented. The structure of this paper is as follows. In section 2 we give a formal definition of alignment. 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 [1]. Section 5 presents a novel inductive 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.

2 Kernel Alignment

A quantitative measure of agreement between kernels and the learning task is important from both a theoretical and a practical point of view. We introduce the concept of *alignment*, which measures the degree of agreement between a kernel and target.

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.

where we use the following definition of inner products between Gram matrices

$$\langle K_1, K_2 \rangle_F = \sum_{i,j=1}^{m} K_1(x_i, x_j) K_2(x_i, x_j) \qquad (1)$$

also known as the Frobenius inner product. If we consider a classification scenario with $K_2 = yy'$, where y is the vector of outputs (+1/-1) for the sample, then

$$A(S, K, yy') = \frac{\langle K, yy' \rangle_F}{\sqrt{\langle K, K \rangle_F \langle yy', yy' \rangle_F}} = \frac{y'Ky}{m\|K\|_F} \qquad (2)$$

The alignment has been shown to possess several convenient properties [1]. Most notably it can be efficiently computed before any training of the kernel machine takes place using an eigenvalue decomposition of

the kernel matrix, and based only on training data information. The complete eigendecomposition of the kernel matrix is an expensive computational step, and should be avoided for large kernel matrices. In [5], we present an approximation strategy to the full eigenvalue decomposition based on the Gram-Schmidt decomposition making alignment optimization tractable for large kernel matrices.

3 Kernel Alignment for Regression

The problem of regression is to approximate an unknown real-valued function from the observation of a limited sequence 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 regression function is learnt from a training set, and its performance can be measured using an independent test set. 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 non-parametric 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,

$$y_i = y_i - \bar{y} \tag{3}$$

where $\bar{y}$ represents the mean over the training set of the target values. For classification the use of alignment was motivated using two facts [1]. 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 generalization. This justified why adapting a kernel to improve its alignment with the target on the training set should result in better generalization performance. This argument cannot be applied to the regression case without modification. We will therefore now present a more complex analysis suggesting why improving the alignment for regression will improve generalization. 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 minus 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,

$$\min_w L(w) = \lambda \langle w, w \rangle + \sum_{i=1}^m \left(\langle w, \mathbf{x}_i \rangle - y_i \right)^2, \tag{4}$$

that forms the adaptable part of an upper bound on the generalization error [3].

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 (4) 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 minimizes $w'Xy$ and hence also $(w'Xy)^2$. Hence, since the result is invariant to rescaling w we can choose $\mu = 0.5$, giving

$$\text{argmax}_w w'Xy - 0.5w'XX'w,$$

which is just the negative of the Ridge Regression optimization (4) 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

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

Similarly,

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

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

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

for all i. Hence,

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

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. an unequal number of class labels in the target vector exist, are commonplace in many real world applications. 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 thereby giving rise to an uneven set of class labels. In the justification of kernel alignment in its relation to the performance of a Parzen windows estimator[1], 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:

$$y_i = \begin{cases} \frac{1}{n_+}, & \text{if } i \text{ is positive} \\ -\frac{1}{n_-}, & \text{otherwise} \end{cases} \quad (5)$$

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 generalization bound will now be for the standard Parzen window estimator with unequal weights for positive and negative examples.

5 Inductive Kernel Alignment

In this section we consider an inductive algorithm for kernel alignment optimization. The transductive algorithms considered in [1] and so far 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 optimize the alignment of the training set kernel matrix to the labels using the same method described in Section 2. 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 and use the resulting feature vector to compute inner products in the transformed space. Pseudo-matlab code for this procedure is given in algorithm 1.

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

Algorithm 1: An inductive alignment algorithm

The complete eigendecomposition of the kernel matrix is an expensive computational step. In a companion paper [5], an inductive approximation strategy based on the Gram-Schmidt decomposition is presented making the optimization feasible for large kernel matrices.

6 Experiments

To demonstrate the performance of the transductive and inductive algorithms for both regression and uneven datasets a range of datasets were considered. The Medline1033 dataset commonly used in text processing [2] was used as an uneven classification problem. This dataset contains 1033 documents and 30 queries obtained from the national library of medicine. In this work we focus on query20. It contains 994 negative and 39 positive examples. 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. In order to test the performance of the alignment algorithm for regression the automobile miles per gallon (AMPG) dataset was considered. It contains the miles travelled, 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.

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 cross validation procedure was used to find the optimal value for the capacity control parameter 'C'. 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. Error results for the different algorithms are presented in the tables below together with $F1$ values. The $F1$ measure is a popular statistic used in the information retrieval community for comparing performance of algorithms on uneven data. $F1$ can be computed using $F1 = \frac{2\tilde{P}R}{\tilde{P}+R}$, where $\tilde{P}$ represents precision i.e. a measure of the proportion of selected items that the system classified correctly, and R represents recall i.e. the proportion of the target items that the system selected.

Table 1 presents the results for the Medline dataset using a support vector classifier (SVC) and Parzen window (PW) estimator using a Bag of Words kernel [4]. 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 this table it is apparent that the training alignment increases for the aligned matrix G across all data partitions. A similar affect is observed for alignment on the test set. There is also a reduction in the SVC mean generalization error, and the PW error for all of the training sets. The quoted $F1$ value derived from the SVC also increases across all data partitions for the aligned matrix.

Table 2 presents the results obtained from the inductive alignment algorithm applied to the Medline dataset. It is apparent that the training alignment increases for the adapted matrix G across all data partitions. A similar affect is observed for the test set alignment. There is also a reduction in the SVC mean generalization error and PW error for all of the training sets. Comparing the results obtained from applying the transductive and inductive alignment algorithms to the 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 test set alignment, together with improved SVC, PW and $F1$ error measures.

Table 3 represents the alignment for the training and test datasets and the associated ridge regression (RR) generalization error for the AMPG dataset. It is clear that the training alignment increases for the adapted matrix G across all data partitions. A similar affect is observed for the test alignment. There is also a reduction in the RR mean generalization error for all of the training sets. 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.

Table 1: 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 using the transductive algorithm.

	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 2: Uneven: Medline dataset: alignment values and SVC, PW and F1 error values for a Bag of Words kernel over 10 runs using the inductive algorithm.

	Train Align	Test Align	SVC Error	PW Error	F1(SVC)
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)

Table 3: Regression: AMPG dataset - alignment values and ridge regression (RR) 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}	0.514 (0.045)	0.566 (0.009)	9.12 (3.12)

Future work will assess the performance of the regression algorithm on high noise datasets.

7 Discussion & Conclusions

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 [1]. 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 with good performance. 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 not a problem. 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 [5] 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.

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

[1] N. Cristianini, A. Elisseeff, J. Shawe-Taylor, and J. Kandola. On kernel target alignment. In *Proceedings Neural Information Processing Systems 2001*, 2001.

[2] N. Cristianini, J.Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2), 2002.

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

[4] T. Joachims. Text categorization using support vector machines. In *Proceedings of European Conference on Machine Learning (ECML)*, 1998.

[5] J. Kandola, J. Shawe-Taylor, and N. Cristianini. An efficient kernel optimisation approach. In *Submitted to Proceedings Neural Information Processing Systems 2002*, 2002.