Reducing Large Diagonals in Kernel Matrices through Semidefinite Programming

Hsiao-Mei Lu
Department of Bioengineering
University of Illinois at Chicago
hlu7@uic.edu

Sumeet Gupta
Department of Bioengineering
University of Illinois at Chicago
sgupta15@uic.edu

Yang Dai*
Department of Bioengineering
University of Illinois at Chicago
yangdai@uic.edu

ABSTRACT

Classification problems in bioinformatics solved by Support Vector Machine (SVM) learning algorithms may result in kernel matrices with very large diagonal elements. This structural characteristic of the kernel matrix leads to overfitting. A method for reducing large diagonals based on the use of Semi-Definite Programming (SDP) is proposed. The kernel matrix generated by SDP is guaranteed to be positive semidefinite. The preliminary experimental performance of the new kernel is encouraging.

Keywords

Support Vector Machine, Semidefinite Programming, Large Diagonals, Kernel Methods

1. INTRODUCTION

Kernel methods are a class of state-of-art learning algorithms that give favorable performance in comparison to traditional learning methods. The support vector machine is a well known example [5; 11; 14]. A chief building block of these methods is an entity known as the kernel. A nonlinear function maps the input points into a higher dimensional feature space so that a linear classifier can be trained in that space. The kernel trick provides an efficient way to construct such a linear classifier by the use of an inner product between mapped points in the feature space. Accordingly, one does not need to actually consider the mapped points explicitly. This non-dependence of the kernel on the dimensionality of the input space and the flexibility of using any kernel make the kernel method one of the most popular approaches for classification in bioinformatics applications [2; 8; 15].

However, when data are represented as sparse vectors, the kernel matrix based on dot products usually has a so-called large diagonals, i.e., the diagonal elements of the matrix are much larger than the off-diagonal terms. Kernel matrices from certain sophisticated kernels, e.g. the string kernel and the motif kernel considered for protein sequence encoding, also possess this structural characteristic [2; 8; 12]. To encode a protein sequence as a vector, the kernel map is designed to generate a sparse vector, where each component corresponds to a "word" or a "motif" in the given sequence. Whenever the related "word"or "motif" occurs in the sequence, the component is set to one or another appropriate specified number.

To achieve better prediction in the presence of such a characteristic in the kernel matrix, Schöelkopf *et al.* [12] proposed a method based on the application of functional calculus to the kernel matrix in order to reduce the dynamic range of the matrix. Generally, this mapping will not guarantee that the new matrix will remain positive definite, one of the properties required by a kernel matrix. This issue was resolved by the introduction of an empirical kernel map [12]. This approach leads to a kernel matrix which is the Gram matrix of the modified kernel matrix. This kernel matrix can then be used with a standard SVM for training. Preliminary computational results have shown that the method can improve performance on kernels that naturally generate matrices with large diagonals [12]. However, the precise role and the choice of the function applied to reduce the dynamic range have yet to be understood.

In this paper, we propose a new reduction method of diagonals based on semidefinite programming [10]. The main features of this method are that the generated kernel is still positive definite and that the similarity measure for a pair of points in the feature space remains unchanged. We applied this kernel with SVM for the classification of two datasets of gene expression values generated from the microarray experiments for cancerous and normal tissues. The experiment shows the effectiveness of this new method for the reduction of diagonal elements.

2. SUBPOLYNOMIAL KERNEL METHODS FOR LARGE DIAGONALS IN KERNEL MATRICES

Support vector machines are supervised approximators that can be considered as an approximate implementation of the structure minimization principle suggested by Vapnik [5; 14]. When used for classification, SVMs map the input space into a higher dimensional feature space that

*The correspondence should be directed to Yang Dai. Mailing address : Department of Bioengineering (MC063), University of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL, USA.

separates a given set of binary labeled training data with an optimal hyperplane. The optimal hyperplane found by the SVM learning algorithm is the one that maximizes the separating margin between the binary classes of the training data. The motivation for mapping the data into a higher dimensional feature space is that linear decision boundaries constructed in the high dimensional feature space correspond to nonlinear decision boundaries in the input space.

Given a training set of m input vectors $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ with known class labels $\{y_1, ..., y_m\} \in \{+1, -1\}$, a new point $\boldsymbol{x}$ is assigned a label by the SVM according to the decision function

$$f(\boldsymbol{x}) = \text{sign}(\sum_{i=1}^{m} y_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b), \tag{1}$$

where $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$ is the kernel function that implicitly defines the feature space. $\phi(\boldsymbol{x})$ is a nonlinear function from input space to feature space, $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors, and α_i $(i = 1, ..., m)$ are coefficients determined by the SVM. A kernel has to belong to the class of positive definite kernels [3]:

$$\sum_{i,j} a_i a_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \tag{2}$$

for all $a_i, a_j \in \boldsymbol{R}$ and all $\boldsymbol{x}_i, \boldsymbol{x}_j$ $(i, j = 1, ..., m)$. The matrix $K_{ij} = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ $(i, j = 1, ..., m)$ is called the Gram matrix. In some cases, the dot product of two different vectors takes a value which is much smaller than the dot product of a vector with itself. That is, given the training inputs $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$, we have

$$k(\boldsymbol{x}_i, \boldsymbol{x}_i) >> |k(\boldsymbol{x}_i, \boldsymbol{x}_j)| \quad \text{for} \quad i \neq j \quad (i, j = 1, ..., m). \tag{3}$$

In this case, the associated Gram matrix is said to have large diagonal elements.

In practice it has been observed that the SVMs with this kind of kernel matrix do not perform well [12]. To deal with the problem of larger diagonal elements, Schöelkopf *et al.* [12] proposed a nonlinear transformation to reduce the dynamic range of the elements of the matrix. They used a so-called subpolynomial kernel, defined as

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{sign}(\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle) |\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle|^p, \quad (0 < p < 1), \tag{4}$$

from the given kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$. However, this method will lead to a matrix which may no longer be positive semidefinite. A procedure was then proposed by Schöelkopf *et al.* [12] to resolve this issue based on the use of the empirical kernel map [13] :

$$\Phi_m(\boldsymbol{x}) := (k(\boldsymbol{x}, \boldsymbol{x}_1)), ..., k(\boldsymbol{x}, \boldsymbol{x}_m))^T. \tag{5}$$

and to train the following SVM :

$$\begin{array}{ll} \min. & \|\boldsymbol{a}\|^2 \\ \text{s.t.} & y_i(\langle \Phi_m(\boldsymbol{x}_i), \boldsymbol{a} \rangle + b) \geq 1 \quad (i = 1, ..., m), \end{array} \tag{6}$$

where $\boldsymbol{a} \in \boldsymbol{R}^m$. This SVM operates in an m-dimensional feature space with the standard SVM regularizor $\|\boldsymbol{a}\|^2$. The SVM is trained simply by an SVM with the kernel

$$k_m(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi_m(\boldsymbol{x}), \Phi_m(\boldsymbol{x}') \rangle, \tag{7}$$

and

$$K_m = KK^T, \tag{8}$$

where K denotes the Gram matrix of the original kernel. Equation (8) shows that when employing the empirical kernel map, it is not necessary to use a positive definite kernel, since KK^T is always positive definite for any kernel matrix. This will be useful for the problem of kernel matrices with large diagonal elements, since the subpolynomial kernel can be considered as K.

3. REDUCING LARGE DIAGONALS THROUGH SEMIDEFINITE PROGRAMMING

In this section we propose a method using a different approach to reduce the large diagonals in the kernel matrix. Our formulation is based on the use of Semidefinite Programming (SDP), which is a generalization of linear programming, see e.g. [10]. SDP deals with the optimization of convex functions over the convex cone of semidefinite matrices, or subsets of those cones. The general form of a SDP is given as follows:

$$\begin{array}{ll} \min. & \boldsymbol{c}^T \boldsymbol{z} \\ \text{s.t.} & F(\boldsymbol{z}) = F_0 + z_1 F_1 + \cdots + z_p F_p \succeq 0, \\ & A\boldsymbol{z} = \boldsymbol{b}, \end{array}$$

where $\boldsymbol{c}, \boldsymbol{z} \in \boldsymbol{R}^p$, $\boldsymbol{b} \in \boldsymbol{R}^s$, $A \in \boldsymbol{R}^{s \times p}$, $F_i = F_i^T \in \boldsymbol{R}^{m \times m}$ $(i = 0, 1, ..., p)$ are given and $F(\boldsymbol{z})$ is restricted to be contained in the positive semidefinite cone. We denote this condition as $F(\boldsymbol{z}) \succeq 0$. In a recent paper of Lanckriet *et al.* [7], SDP was used to learn an optimal kernel matrix from both the training and testing data. The learning is transductive, i.e., using the labeled part of the data to learn an "optimal" embedding also for the testing part. From the embedding, the reduced similarity, defined as the inner products between testing points is learned.

In this study, we wish to specify a convex cost function of the SDP that will enable us to learn from the original kernel matrix a new matrix whose diagonals are no longer large. The motivation for the formulation is to subtract a certain amount from each diagonal element of the kernel matrix to reduce the diagonal dominance, while keeping the new matrix positive semidefinite. The amount of the subtraction can be controlled by using different objective functions in the SDP. Here, we consider the following simplest form.

Let K be the original kernel matrix obtained from the training data. Assume that $\{1, ..., m\}$ is divided into p subsets of approximately equal size. Let

$$K^D = diag(K_{11}, ..., K_{mm})$$

be a diagonal matrix where $K^D_{ii} = K_{ii}$. Let $\boldsymbol{c} = \{-1, ..., -1\}^T \in \boldsymbol{R}^p$ $(1 \leq p \leq m)$ and $F_0 = K$. There are two ways to construct constraints in order to reduce the diagonal elements. First, let $-F_i$ be a diagonal matrix with only 1's at the positions corresponding to the ith block :

$$
-F_i = \begin{array}{c} \overbrace{\qquad\qquad\qquad\qquad}^{i\text{th block}} \\ \begin{pmatrix} 0 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 0 & & & & & & \\ & & & 1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 1 & & & \\ & & & & & & 0 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 0 \end{pmatrix} \end{array}
$$

$$(i = 1, ..., p).$$

Then $-(F_1 + \cdots + F_p)$ is the unit matrix.

Second, let

$$
-F_1 = \begin{pmatrix} K_{11} & & & & & & \\ & \ddots & & & & & \\ & & K_{i_1 i_1} & & & & \\ & & & 0 & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix}, \ldots,
$$

$$
-F_p = \begin{pmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ & & & 0 & & & \\ & & & & K_{i_{p-1}+1, i_{p-1}+1} & & \\ & & & & & \ddots & \\ & & & & & & k_{mm} \end{pmatrix}.
$$

Obviously, $-K^D = F_1 + \cdots + F_p$. In both cases, the constraint set $A\boldsymbol{z} \leq \boldsymbol{b}$ does not exist.

Based on the above two definitions of F_i $(i = 1, ..., p)$, two SDP modified kernel matrices can be obtained by solving the SDPs. Since all the off-diagonal elements remain unchanged in the modified kernels, the original kernel function can be used in the decision function (1) for the testing. The optimal number of blocks should be the one that generates a kernel matrix giving the best accuracy. In practice it can be determined empirically through the procedure of cross-validation.

Since the subpolynomial kernel changes all entries in the original kernel matrix, the similarity of a pair of points in the mapped feature space is completely altered. It is hard to measure how suitable the new kernel is for the specific application domain. In contrast, the SDP modified kernels only change the diagonal elements in the original kernel matrix, the similarity for every off-diagonal element remains unchanged.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We employed the method of SDP kernel modification described in the previous section in conjunction with an SVM for classification. The evaluation of the method was conducted on the following two data sets.

4.1 Datasets

Alon's Microarray Data with Added Noise : Alon$_{100}$
Using Affymetrix oligonucleotide arrays, expression levels for 40 tumor and 22 normal colon tissues are measured for 6500 genes (features). Of these genes, the 2000 with the highest minimal intensity across the tissues are selected for classification [1]. One must distinguish between cancerous and normal tissues given the expression values of the genes. In order to obtain the larger diagonal elements in the kernel matrix, noise was added as described in [12]: $10,000$ features with values 0 were added to each tissue and a randomly chosen group of 100 out of these features was then set to be random values in $(0, 1)$. The ratio of the average off-diagonal elements to that of the diagonal elements in the linear kernel matrix of these data is 0.025 (see Table 3).

Golub's Microarray Data : Golub
The data set of Golub *et al.* [6] consists of gene expression values of samples from 27 acute lymphoblastic leukemia (ALL) and 11 acute myeloid leukemia (AML). Gene expression levels were measured using Affymetrix oligonucleotide chips containing 6817 human genes. However, the set only has 3051 genes that were filtered out through the pre-processing steps. The objective is to discriminate the two types of leukemias. The ratio of the average off-diagonal elements to that of diagonal elements of the linear kernel matrix of these data is 0.019 (see Table 3).

4.2 Experiments and Discussion

For each dataset, six types of kernels were used in the experiments : (i) linear, (ii) radial basis function kernel (RBF), (iii) polynomial kernel, (iv) subpolynomial kernel proposed in [12], (v) SDP modified kernel by the type 1 constraint (SDP-1), and (vi) SDP modified kernel by the type 2 constraint (SDP-2). The values of p in the subpolynomial kernel were changed in the range of $[0.1, 0.9]$ with an incremental size 0.1. The block sizes p in the SDP modified kernel were set as $1, 2, 4, 10, 30, 62$ for Alon$_{100}$ and $1, 2, 4, 8, 19, 38$ for Golub. SDPA (ver6.0) [9] was used for the solution of SDP. The SVM solver libsvm (ver.2.4) [4] was modified to test the performance of the SVM with our new kernels. A 10-fold cross-validation was employed in experiments. The two criteria used to evaluate the classification accuracy are given as follows:

$$\text{accuracy} = \frac{\#\text{ correctly predicted points}}{\#\text{ total number of points}},$$

and

$$\text{balanced loss} = \frac{1}{2}\frac{\#\{\hat{y}:y=1\wedge\hat{y}=-1\}}{\#\{y:y=1\}} + \frac{1}{2}\frac{\#\{y:y=-1\wedge\hat{y}=1\}}{\#\{y:y=-1\}},$$

where y denotes the true label and $\hat{y}$ presents the predicted value. The balanced loss takes into account the unequal numbers of positive and negative labeled training points.

The generalization performance of an SVM is controlled by the trade-off parameter C and parameters associated with the kernels. The optimal values of C^* and g^* in the RBF kernels ($k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-g\|\boldsymbol{x} - \boldsymbol{y}\|^2)$) which maximize the accuracy were chosen. More precisely, the values for C^* and g^* were determined as follows. For the linear, polynomial, and subpolynomial kernels, we took $C = 0.001, 0.01, 0.1$ and the values in the range of $[1, 10]$ with an incremental size 1; and took $g = 0.001, 0.01, 0.1$ and the values in $[1, 10]$ with an incremental size 1. The degree p of the polynomial kernel was set at $d = 2, 3, 4, 5$. The values of C in the SDP modified kernels were set at 1 for all tests.

Tables 1 and 2 present the results of the experiments. The average accuracy and balanced loss are taken as the average results from 10 runs of the 10-fold cross-validation for each fixed parameter pair, respectively. The numbers in the parentheses are the standard deviations. The best results are presented in bold face. In general, the subpolynomial and SDP modified kernels exhibit the improved generalization performance over the linear, polynomial, and RBF kernels with these two datasets. However, the SDP modified kernels show better results than the subpolynomial kernels for Golub, while the opposite is observed for Alon_{100}. In order to provide a possible explanation for this outcome, we calculated the ratios of the kernel matrices resulting from these two different methods.

Table 3 presents the ratios, i.e., the average of the off-diagonal elements over that of the diagonal elements, corresponding to the kernel matrices of the linear, the subpolynomial, and the SDP. The ratios are shown only for the matrices which generated good results for the latter two kernels. The ratios corresponding to the SDP methods are much larger for Alon_{100} compared with those of the subpolynomial method. However, the ratios for both methods are about the same for Golub.

This result can be explained as follows. Since some of the off-diagonal elements are less than 1 for Alon_{100}, as shown in K_1 in Figure 1, the subpolynomial operation actually has the effect of increasing the values for those elements. Therefore, this results in a substantial reduction of the difference between the diagonal and off-diagonal elements in the subpolynomial kernel matrix for this data set (see Figure 2).

On the other hand, the SDP methods only reduce the magnitude of the diagonal elements. This is why the ratios remain small in the SDP kernel matrices for the same dataset. As for Golub, since all elements are great than 1, as shown in K_2 in Figure 1, the subpolynomial method can not have the effect described above.

$$K_1 = \begin{pmatrix} 36.9545 & 0.71564 & 0.685017 & 1.32080 \\ 0.71563 & 34.7406 & 0.667553 & 1.58186 \\ 0.68501 & 0.66755 & 34.50790 & 0.73551 \\ 1.3208 & 1.58186 & 0.735511 & 37.1586 \end{pmatrix},$$

$$K_2 = \begin{pmatrix} 3108.16 & 159.557 & 43.0674 & 54.7271 \\ 159.557 & 3044.70 & 143.160 & 53.5596 \\ 43.0674 & 143.160 & 3072.59 & 148.006 \\ 54.7271 & 53.5596 & 148.006 & 3038.89 \end{pmatrix},$$

and

$$K_1^{RBF} = \begin{pmatrix} 1 & 0.00032 & 0.00038 & 0.00047 \\ 0.00032 & 1 & 0.00032 & 0.00040 \\ 0.00038 & 0.00032 & 1 & 0.00046 \\ 0.00047 & 0.00040 & 0.00046 & 1 \end{pmatrix}.$$

Figure 1 : The first 4×4 entries of the kernel matrices K_1 and K_2 of the linear kernel on Alon_{100} and Golub, respectively, and the first 4×4 entries of the kernel matrix of the RBF kernel K_1^{RBF} ($g = 1$)of Alon_{100}.

$$K_1^{Subpoly} = \begin{pmatrix} 1.4347 & 0.9671 & 0.9629 & 1.0282 \\ 0.9671 & 1.4259 & 0.9604 & 1.0469 \\ 0.9629 & 0.9604 & 1.4249 & 0.9697 \\ 1.0282 & 1.0469 & 0.9697 & 1.4355 \end{pmatrix},$$

$$K_1^{SDP-1} = \begin{pmatrix} 6.554 & 0.7156 & 0.685 & 1.321 \\ 0.7156 & 4.340 & 0.6676 & 1.582 \\ 0.685 & 0.6676 & 4.166 & 0.7355 \\ 1.321 & 1.582 & 0.7355 & 6.8170 \end{pmatrix},$$

$$K_1^{SDP-2} = \begin{pmatrix} 5.142 & 0.7156 & 0.6850 & 1.321 \\ 0.7156 & 4.834 & 0.6676 & 1.582 \\ 0.6850 & 0.6676 & 4.356 & 0.7355 \\ 1.321 & 1.582 & 0.7355 & 4.691 \end{pmatrix}.$$

Figure 2 : The first 4×4 entries of the subpolynomial modified kernel matrix $K_1^{Subpoly}$ with $p = 0.1$ on the linear kernel of Alon_{100}, the SDP modified kernel matrices K_1^{SDP-1} and K_1^{SDP-2} of the linear kernel on Alon_{100} by SDP-1 with $p = 30$, and SDP-2 with $p = 30$, respectively.

In order to investigate the noise resistance of the new algorithm with increasing noise levels, we performed the following experiments. In addition to the data Alon_{100}, three more data sets with different noise levels were generated. That is, 10, 000 features with values 0 were added to each tissue and randomly chosen groups of 50, 150, and 250 out of these features were then set to be random values in $(0, 1)$, respectively. The data files were named Alon_{50}, Alon_{150}, and Alon_{250}, respectively. Then the experiments described above were performed on these data sets. Table 4 shows the results. We could see that both the subpolynomial and SDP modified kernels demonstrated consistent improvement over the linear and RBF kernels, which presented almost unchanged performance. However, the performance of the two former ones do get worse when the noise level is increased. The predicting rates from the SDP-1 modified kernels are slightly lower than that of the subpolynimial one. This picture may change if we tune the parameter C for the SDP-1 case, where the value C was fixed at 1 during the experiment.

5. CONCLUSIONS

This paper has introduced a new SDP-based method for the reduction of large diagonal elements in kernel matrices generated from conventional kernel functions. The performance of the proposed method was empirically tested on

Table 1: Results of SVM with different kernels for Alon_{100}.

Kernel	Parameter	Ave. Bal. Loss	Ave. Accuracy%
Linear	$C^* = 2$	0.465 (0.034)	64.84 (0.01)
RBF	$C^* = 10$, $g^* = 1$	0.460 (0.039)	64.52 (0.00)
Poly	$C^* = 10$, $p = 5$	**0.460** (**0.046**)	**64.52** (**0.00**)
	$p = 4$	0.485 (0.034)	64.52 (0.00)
	$p = 3$	0.480 (0.026)	64.52 (0.00)
	$P = 2$	0.465 (0.034)	64.52 (0.01)
Sub -poly	$C^* = 3$, $p = 0.9$	0.465 (0.021)	65.32 (0.01)
	$p = 0.8$	0.391 (0.024)	69.35 (0.02)
	$p = 0.7$	0.372 (0.037)	71.45 (0.01)
	$p = 0.6$	0.316 (0.044)	75.32 (0.02)
	$p = 0.5$	0.242 (0.031)	79.52 (0.03)
	$p = 0.4$	0.222 (0.040)	81.29 (0.02)
	$p = 0.3$	0.186 (0.036)	84.84 (0.02)
	$p = 0.2$	0.174 (0.037)	85.00 (0.03)
	$p = 0.1$	**0.130** (**0.021**)	**87.58** (**0.02**)
SDP-1	$C = 1$, $p = 1$	0.455 (0.033)	65.16 (0.01)
	$p = 2$	0.429 (0.054)	65.97 (0.02)
	$p = 4$	0.232 (0.069)	78.23 (0.06)
	$p = 10$	0.181 (0.055)	81.45 (0.04)
	$p = 30$	**0.146** (**0.041**)	**83.06** (**0.02**)
	$p = 62$	0.155 (0.040)	82.74 (0.05)
SDP-2	$C = 1$, $p = 1$	0.480 (0.022)	65.00 (0.01)
	$p = 2$	0.455 (0.027)	65.16 (0.01)
	$p = 4$	0.205 (0.054)	79.03 (0.05)
	$p = 10$	0.181 (0.032)	80.32 (0.05)
	$p = 30$	**0.164** (**0.037**)	**81.77** (**0.04**)
	$p = 62$	0.171 (0.012)	80.65 (0.03)

Table 2: Results of SVM with different kernels for data Golub.

Kernel	Parameter	Balanced Loss	Average Accuracy (%)
Linear	$C^* = 4$	0.340 (0.042)	71.05 (0.00)
RBF	$C^* = 7$, $g^* = 3$	0.375 (0.042)	71.05 (0.00)
Poly	$C^* = 6$, $p = 5$	0.360 (0.052)	71.05 (0.00)
	$P = 4$	0.365 (0.041)	71.05 (0.00)
	$P = 3$	**0.345** (**0.050**)	**71.05**(**0.00**)
	$P = 2$	0.380 (0.026)	71.05 (0.00)
Sub -poly	$C^* = 10$, $p = 0.9$	0.405 (0.044)	71.05 (0.00)
	$p = 0.8$	0.318 (0.046)	74.21 (0.02)
	$P = 0.7$	0.179 (0.034)	85.26 (0.03)
	$P = 0.6$	**0.130** (**0.055**)	**87.63** (**0.04**)
	$P = 0.5$	0.196 (0.051)	82.11 (0.04)
	$P = 0.4$	0.211 (0.042)	80.79 (0.02)
	$P = 0.3$	0.215 (0.038)	79.47 (0.04)
	$P = 0.2$	0.204 (0.066)	77.63 (0.05)
	$P = 0.1$	0.184 (0.074)	80.79 (0.08)
SDP-1	$C = 1$, $p = 1$	**0.098** (**0.04**)	88.42 (0.045)
	$p = 2$	0.145 (0.03)	86.05 (0.041)
	$p = 4$	0.138 (0.04)	**88.68** (**0.029**)
	$p = 8$	0.148 (0.02)	86.05 (0.029)
	$p = 19$	0.112 (0.04)	86.58 (0.022)
	$p = 38$	0.375 (0.03)	71.05 (0.000)
SDP-2	$C = 1$, $p = 1$	0.143 (0.044)	88.16 (0.02)
	$p = 2$	0.110 (0.042)	89.21 (0.04)
	$p = 4$	**0.103** (**0.037**)	**90.79** (**0.03**)
	$p = 8$	0.110 (0.039)	89.74 (0.04)
	$p = 19$	0.283 (0.121)	75.44 (0.10)
	$p = 38$	0.355 (0.035)	71.05 (0.00)

two biological datasets. The experiments indicate that the SDP modified kernels can produce better generalization performance for certain classes of data. Extensive experiments are required to compare further the subpolynomial and the SDP kernels in order to understand their roles. We are currently investigating the string kernel [8] and the motif kernel [2] designed for protein sequences involving the application of protein family prediction.

6. ACKNOWLEDGMENTS

Support for HL was partially provided under contracts with the Army Research Office (DAAG55-97-1-0310) and the Department of Energy at the Sandia National Laboratories. Sandia is a multiprogram laboratory operated by the Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract no. DE-AC04-94AL85000. We would also like to thank the anonymous referees for their valuable suggestions.

7. REFERENCES

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays *Proc. Natl. Acad. Sci., USA*, 96 (1999) 6745-6750.

[2] A. Ben-Hur and D. Brutlag, Remote homology detection: a motif based approach, to appear in *Proceedings of ISMB* 2003.

[3] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups (Theory of Positive definite and Related Functions),* Springer-Verlag, New York, 1984.

[4] C-C. Chang and C-J. Lin, LIBSVM : a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

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

[6] T. R. Golub *et al.*, Molecular classification of cancer : class discovery and class prediction by gene expression monitoring, *Science* 286 (1999) 531-537.

[7] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, M. Jordan, Learning the kernel matrix with semi-definite programming, in C. Sammut and A. Hoffmann (Eds.), *Proceedings of the 19th International Conference on Machine Learning*, Morgan Kaufmann, 2002.

[8] C. Leslie, E. Eskin and W. S. Noble, The spectrum kernel: A string kernel for SVM protein classification, *Proceedings of the Pacific Symposium on Biocomputing (PSB-2002). Kaua'i, Hawaii* 2002.

Table 3: The ratio of the average off-diagonal elements to that of diagonal elements in kernel matrices.

Dataset	kernel	parameter	ratio
Alon_{100}	linear		0.025
	sub-poly	$p = 0.2$	0.472
	sub-poly	$p = 0.1$	0.686
	SDP-1	$p = 30$	0.103
	SDP-1	$p = 62$	0.155
	SDP-2	$p = 30$	0.113
	SDP-2	$p = 62$	0.154
Golub	linear		0.019
	sub-poly	$p = 0.7$	0.059
	sub-poly	$p = 0.6$	0.087
	SDP-1	$p = 4$	0.084
	SDP-1	$p = 8$	0.085
	SDP-2	$p = 4$	0.085
	SDP-2	$p = 8$	0.083

[9] K. Fujisawa, M. Kojima and K. Nakada, SDPA, SemiDefinite Programming Algorithm software package, avaiable at http://www.is.titech.ac.jp/~yamashi9/sdpa/, 2002.

[10] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, Philadelphia, 1993.

[11] B. Schöelkopf And A. J. Smola, *Learning with Kernels : Support Vector Machines, Regularization, Optimizations, and Beyond*, The MIT Press, 2002.

[12] B. Schöelkopf, J. Weston, E. Eskin, C. Leslie and W. S. Noble, Dealing with Large Diagonals in Kernel Matrices. (or A kernel approach for learning from almost orthogonal patterns). *Annals of the Institute of Statistical Mathematics* (or ECML'2002 and PKDD'2002).

[13] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller, A new discriminative kernel from probabilistic models, in T. G. Dietterich *et al.* eds. *Advances in Neural Information Processing Systems*, MIT Press 14 (2002) 977-984.

[14] V. Vapnik, *Statistical Learning Theory*, Johu Wiley and Sons, New York, 1998.

[15] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, C. Lemmen, A. Smola, T. Lengauer and K. R. Müller, Engineering support vector machine kernels that recognize translation initiation sites, *Proceedings of the German Conference on Bioinformatics '99. Hannover, Germany*, (1999) 37-43.

Table 4: Results of SVM with different kernels for Alon's data with different noise levels(Only the best results are shown. The parameter C in the SDP modified kernel was fixed at 1. All other C^* and p^* were determined as above).

Kernel	Parameter	Balanced Loss	Average Accuracy (%)
Alon_{50}			
Linear	$C^* = 1$	0.430 (0.037)	65.97 (0.01)
RBF	$g^* = 0.01$, $C^* = 7$	0.460 (0.032)	64.68 (0.01)
sub-poly	$C^* = 4$, $p^* = 0.1$	0.066 (0.012)	93.87 (0.01)
SDP-1	$C = 1$, $p^* = 62$	0.053 (0.022)	93.71 (0.02)
Alon_{100}			
Linear	$C^* = 3$	0.460 (0.046)	64.84 (0.01)
RBF	$g^* = 1$, $C^* = 10$	0.440 (0.061)	64.52 (0.00)
sub-poly	$C^* = 3$, $p^* = 0.1$	0.130 (0.021)	87.58 (0.02)
SDP-1	$C = 1$, $p^* = 30$	0.147 (0.041)	83.06 (0.02)
Alon_{150}			
Linear	$C^* = 10$	0.460 (0.046)	64.52 (0.00)
RBF	$g^* = 3$, $C^* = 0.01$	0.445 (0.044)	64.52 (0.00)
sub-poly	$C^* = 3$, $^*p = 0.1$	0.261 (0.025)	74.68 (0.03)
SDP-1	$C = 1$, $p^* = 62$	0.302 (0.035)	69.52 (0.03)
Alon_{250}			
Linear	$C^* = 8$	0.455 (0.050)	64.52 (0.00)
RBF	$g^* = 1$, $C^* = 2$	0.445 (0.037)	64.52 (0.00)
sub-poly	$C^* = 9$, $p^* = 0.3$	0.298 (0.046)	73.87 (0.03)
SDP-1	$C = 1$, $p^* = 2$	0.287 (0.018)	72.74 (0.03)