KERNEL/FEATURE SELECTION FOR SUPPORT VECTOR MACHINES APPLIED TO MATERIALS DESIGN [1]

Shawn Martin, Michael Kirby, Rick Miranda

Department of Mathematics
Colorado State University
Fort Collins, CO 80523-1874

Abstract: Support Vector Machines are classifiers with architectures determined by kernel functions. In these proceedings we propose a method for selecting the best SVM kernel for a given classification problem. Our method searches for the best kernel by remapping the data via a kernel variant of the classical Gram-Schmidt orthonormalization procedure then using Fisher's linear discriminant on the remapped data. By specializing to the Veronese kernel we can also perform feature selection with this method. We perform both feature and kernel selection on a materials design problem.

Keywords: Support Vector Machines, Kernel Selection, Gram-Schmidt, Fisher's Discriminant, Materials Design

1. INTRODUCTION

Support Vector Machines (Vapnik, 1998; Burges, 1998) are classifiers (Duda and Hart, 1973; Schürmann, 1996) designed around an optimal separating hyperplane. This hyperplane is known as the *maximal margin hyperplane* and is illustrated in Figure 1.

To handle nonlinearly separable data, SVMs use nonlinear maps $\Phi : \mathbb{R}^n \rightarrow F$ to preprocess the data, where F is a Hilbert space with inner product $(\bullet, \bullet)$. This idea is best illustrated using the Veronese map (Shafarevich, 1994) as shown in Figure 2.

Since F may be high-dimensional (even infinite-dimensional) this preprocessing by Φ is accomplished via a kernel function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ which satisfies

$$\kappa(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}), \Phi(\mathbf{y})).$$

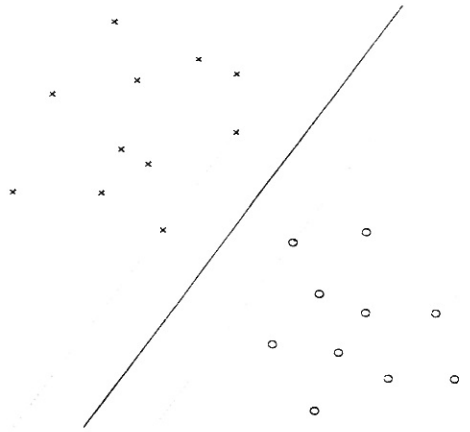

Fig. 1. Here we show a linear Support Vector Machine in the case of separable two class data. The two classes are depicted by x's and o's, with the maximal margin hyperplane shown as a solid line separating them. The two dotted lines run through the *support vectors* and are separated by a distance equal to the *margin*.

[1] This work supported by research grant DOD-USAF Office of Scientific Research F49620-99-1-0034.

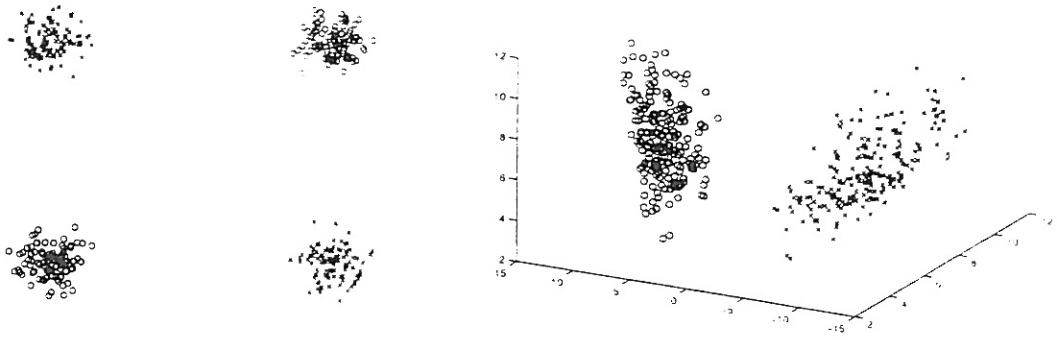

Fig. 2. Here we use the Veronese map $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ to illustrate the advantage of nonlinear preprocessing. Specifically, our nonlinearly separable data (left) becomes linearly separable (right) after remapping by Φ. We additionally observe that Φ has an associated kernel $\kappa(\mathbf{x}, \mathbf{y}) = (\mathbf{x}, \mathbf{y})^2$.

Some kernels (Burges, 1998; Burges, 1999) are:

- the Veronese kernel
  $\kappa(\mathbf{x}, \mathbf{y}) = ((\mathbf{x}, \mathbf{y}) + c)^d$, $c \geq 0$, $d \in \mathbb{Z}_{>0}$.
- the radial basis function (RBF) kernel
  $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$, $\sigma \neq 0$.
- the neural network kernel
  $\kappa(\mathbf{x}, \mathbf{y}) = \tanh(a(\mathbf{x}, \mathbf{y}) + b)$, $a, b \geq 0$.

Using such kernels a Support Vector Machine has the form

$$f(\mathbf{x}) = \operatorname{sign}(\textstyle\sum_i \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \sum_j \beta_j \kappa(\mathbf{y}_j, \mathbf{x}) - b).$$

where $X = \{\mathbf{x}_i\}$, $Y = \{\mathbf{y}_j\}$ are the two classes, α_i, β_j, and b are computed via a constrained quadratic programming problem, many of the α_i, β_j are zero (nonzero α_i, β_j correspond to support vectors), and $f(X) = \{1\}, f(Y) = \{-1\}$.

In what follows we propose a method for selecting the best SVM kernel (and hence SVM architecture) for a given classification problem. Our method is based on a fast calculation of the classification ability of a given kernel. This calculation is based on a kernel variant of the classical Gram-Schmidt orthonormalization procedure (Trefethen and Bau, 1997) followed by an application of Fisher's linear discriminant (Duda and Hart, 1973). By specializing to the Veronese kernel, our method can also be used for feature selection.

In section 2 we describe our kernel variant of Gram-Schmidt; in section 3 we provide some background on Fisher's discriminant; in section 4 we combine our kernel variant of Gram-Schmidt with Fisher's discriminant to measure kernel classification ability; in section 5 we consider feature selection using the Veronese kernel; in section 6 we consider kernel selection for Support Vector Machines; in section 7 we provide symmetric versions of the SVM kernels for use in our application; and in section 8 we apply our methods to a problem in materials design.

2. KERNEL GRAM-SCHMIDT

Suppose we have data $Z = X \cup Y = \{\mathbf{z}_i\}_{i=1}^m \subset \mathbb{R}^n$. Denote $\Phi(\mathbf{z}_i)$ by $\widetilde{\mathbf{z}}_i$ and assume that $\{\widetilde{\mathbf{z}}_i\}_{i=1}^m$ is a linearly independent set of vectors in F. The goal of kernel Gram-Schmidt is to produce a matrix

$$B = \begin{pmatrix} (\widetilde{\mathbf{z}}_1, \mathbf{u}_1) & \cdots & (\widetilde{\mathbf{z}}_m, \mathbf{u}_1) \\ \vdots & \ddots & \vdots \\ (\widetilde{\mathbf{z}}_1, \mathbf{u}_m) & \cdots & (\widetilde{\mathbf{z}}_m, \mathbf{u}_m) \end{pmatrix},$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_m$ form an orthonormal basis for the subspace in F spanned by $\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_m$.

By solving for $(\widetilde{\mathbf{z}}_i, \mathbf{u}_j)$ in the classical Gram-Schmidt orthonormalization procedure (Trefethen and Bau, 1997) we obtain the formula

$$(\widetilde{\mathbf{z}}_i, \mathbf{u}_j) = \frac{1}{(\widetilde{\mathbf{z}}_j, \mathbf{u}_j)}((\widetilde{\mathbf{z}}_i, \widetilde{\mathbf{z}}_j) - (\widetilde{\mathbf{z}}_j, \mathbf{u}_{j-1})(\widetilde{\mathbf{z}}_i, \mathbf{u}_{j-1}) - (\widetilde{\mathbf{z}}_j, \mathbf{u}_{j-2})(\widetilde{\mathbf{z}}_i, \mathbf{u}_{j-2}) - \cdots - (\widetilde{\mathbf{z}}_j, \mathbf{u}_1)(\widetilde{\mathbf{z}}_i, \mathbf{u}_1)).$$

This method can be generalized to a set of (possibly) linearly dependent vectors $\{\widetilde{\mathbf{z}}_i\}$ to get a recursive algorithm for computing the matrix

$$B = \begin{pmatrix} (\widetilde{\mathbf{z}}_1, \mathbf{u}_1) & \cdots & (\widetilde{\mathbf{z}}_m, \mathbf{u}_1) \\ \vdots & \ddots & \vdots \\ (\widetilde{\mathbf{z}}_1, \mathbf{u}_q) & \cdots & (\widetilde{\mathbf{z}}_m, \mathbf{u}_q) \end{pmatrix},$$

where $\mathbf{u}_1, \ldots, \mathbf{u}_q$ form an orthonormal basis for the subspace in F spanned by $\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_m$.

This method, which we call *kernel Gram-Schmidt*, has several important properties. First and foremost, it is expressed entirely in terms of inner products in F so can be computed using kernels. Second, it entirely avoids actually computing the orthonormal set $\{\mathbf{u}_1, \ldots, \mathbf{u}_q\}$. This is necessary since F may be infinite dimensional. Third and last, but very important, the generated matrix B represents the data $\{\widetilde{\mathbf{z}}_i\}$ in F in exactly the same manner that it would be represented in $\mathbb{R}^q$. Thus we can apply any standard algorithm from linear algebra to B.

3. FISHER'S DISCRIMINANT

Fisher's linear discriminant uses a separating hyperplane located by optimizing Fisher's criterion function $J : \mathbb{R}^n \to \mathbb{R}$ given by

$$J(\mathbf{a}) = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2}.$$

where m_1, m_2 are the means of the (orthogonal) projections of X, Y respectively onto $\mathbf{a}$, and σ_1^2, σ_2^2 are the variances of the projections of X, Y onto $\mathbf{a}$.

Fisher's criterion function provides a useful measure of the linear separability of X and Y. By its definition, we see that larger values of J indicate better separation of X and Y. In particular, a value of one indicates a separation of the projected means by one standard deviation of the projected values of X plus one standard deviation of the projected values of Y. Similarly, if $J = 4$ the means are approximately separated by two standard deviations, et cetera.

By maximizing Fisher's criterion we can locate a good separating hyperplane for X, Y. Maximizing J is accomplished by first rewriting Fisher's criterion as

$$J(\mathbf{a}) = \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_c \mathbf{a}}.$$

where S_b is the between class scatter and S_c is the within class scatter (see Duda and Hart, 1973). By setting $\nabla J(\mathbf{a}) = \mathbf{0}$ we see that $J(\mathbf{a}^*)$ is a maximum when $\mathbf{a}^* = S_c^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$.

The resulting separating hyperplane

$$H(\mathbf{x}) = (\mathbf{x}, \mathbf{a}^*) - \frac{\sigma_2 m_1 + \sigma_1 m_2}{\sigma_1 + \sigma_2}\|\mathbf{a}^*\| = 0$$

yields a classifier known as Fisher's linear discriminant. Using Fisher's linear discriminant, a point $\mathbf{x} \in \mathbb{R}^n$ is classified as a member of X if $H(\mathbf{x})$ and $H(\mathbf{m}_1)$ have the same sign, and a member of Y otherwise.

4. KERNEL CLASSIFICATION ABILITY

Combining Fisher's discriminant with nonlinear preprocessing and computation using kernel Gram-Schmidt, we have a fast method for testing the classification ability of a given SVM kernel.

Specifically, suppose we want to evaluate the classification ability of a given kernel κ. Performing kernel Gram-Schmidt using κ on our data $Z = X \cup Y$ we obtain $B = (\Delta\ \Gamma)$, where Δ, Γ are matrices containing the implicitly mapped (via κ) data X, Y. (More precisely, the columns of Δ, Γ are the coefficients of the vectors $\Phi(X), \Phi(Y)$ projected onto an orthonormal basis for the subspace spanned by $\Phi(Z)$, where Φ is the map associated with the kernel κ.) Calculating Fisher's discriminant using Δ, Γ we obtain a separating hyperplane H^* with normal vector $\mathbf{a}^*$ and associated criterion value $J(\mathbf{a}^*)$. Finally, we calculate the percentages p and t of data and test data correctly classified using H^*. The values $J(\mathbf{a}^*), p$ and t allow us to rank different kernels. (Higher values are generally better.)

We note that the previous calculations are fast since applying kernel Gram-Schmidt is $O(mq^2)$ and computing Fisher's discriminant is $O(q^3)$, where generally $n < q << m$. (Recall $X, Y \subset \mathbb{R}^n$, $Z = X \cup Y = \{\mathbf{z}_i\}_{i=1}^m$, and $\mathbf{u}_1, \ldots, \mathbf{u}_q$ form an orthonormal basis for the subspace spanned by the vectors $\Phi(Z)$.)

5. FEATURE SELECTION

We can use our measure of the classification ability of a Support Vector Machine kernel for both feature and kernel selection. (By features we mean physical properties which distinguish between our two classes.) For feature selection we use the Veronese kernel $((\mathbf{x}, \mathbf{y}) + 1)^d$. We proceed as follows:

(1) For a given choice of features and a given degree d for the Veronese kernel we apply kernel Gram-Schmidt to our data X, Y to obtain a $q \times m$ matrix B. We then calculate using B
   - The vector $\mathbf{a}^* \in \mathbb{R}^q$ that maximizes Fisher's criterion $J : \mathbb{R}^q \to \mathbb{R}$ and the actual maximum $J(\mathbf{a}^*)$,
   - The percentage p of points in X, Y correctly classified using Fisher's discriminant with separating hyperplane determined by $\mathbf{a}^*$,
   - The percentage t of points in our set of test data correctly classified.

(2) We record the results of (1) for different feature combinations and degrees $d = 1, 2, \ldots, r$ of the Veronese kernel. In the materials problem, we perform this step first on a single feature, then on two features, with one being from the list of single best features. It is because kernel Gram-Schmidt and the optimization of J are computationally inexpensive that we can calculate so many combinations.

(3) We rank the effectiveness of the feature combinations using the optimal values of J and the percentages recorded in step (2). More precisely, we order the features by averaging the values $J(\mathbf{a}^*), p$, and t calculated in step (1) over the Veronese degrees $1, 2, \ldots, r$ in step (2). We denote these averages by $\overline{J(\mathbf{a}^*)}_r, \bar{p}$, and $\bar{t}$ and we use $\overline{J(\mathbf{a}^*)}_r$ to rank the features.

To see why this algorithm provides a valid ranking of features combinations consider two fixed features F_1, F_2. In this case our data X, Y lies in the plane $\mathbb{R}^2$. What does our algorithm measure? For each degree $d = 1, 2, 3, \ldots, r$ of the Veronese kernel we apply nonlinear preprocessing implicitly via kernel Gram-Schmidt to our data X, Y. Next using Fisher's criterion and Fisher's discriminant we measure the linear separability of our implicitly mapped data. By a special property of the Veronese kernel (see Martin *et al.*, 2000), we are measuring the linear separability of our data ($d = 1$), the quadratic separability of our data ($d = 2$), the cubic separability of our data ($d = 3$), et cetera. By using Fisher's discriminant in combination with the Veronese kernel, we are measuring the linear and increasingly nonlinear separability of our data for our given features F_1, F_2.

We remark that other kernels could also be used for feature selection, although they may result in different feature rankings. In addition to being more easily interpreted than the RBF and neural network kernels, we chose the Veronese kernel because it generally yields the fastest computations and is the least sensitive to the peculiarities of a given data set.

6. KERNEL SELECTION

Upon selecting the best features we perform an algorithm for kernel selection:

(4) For a given kernel and choice of kernel parameters, we calculate $\mathbf{a}^*, J(\mathbf{a}^*), p$ and t as in (1).

(5) We record the results of (4) for different kernels and kernel parameters. In the materials problem we use the kernels mentioned in section 1 along with discretizations of the corresponding kernel parameters. For the Veronese kernel we generally take d from 1 to 10, for the radial basis function (RBF) kernel $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/2\sigma^2)$ we consider $\sigma \in (0, 1]$, and for the neural net kernel $\kappa(\mathbf{x}, \mathbf{y}) = \tanh(a(\mathbf{x}, \mathbf{y}) + b)$ we let $0 \leq a, b \leq 20$.

(6) We compare the kernels using the values of $J(\mathbf{a}^*), p$ and t recorded in step (4). Here we generally use the maximum values of $J(\mathbf{a}^*), p$ and t to compare the kernels, as opposed to the averages used in step (3).

7. SYMMETRIC KERNELS

Before presenting our results we provide some symmetric versions of the SVM kernels which we use in our application. These kernels are designed to exploit the component order symmetry in the materials problem.

When using the Veronese kernel, there is a natural way to enforce component order symmetry. Specifically, we preprocess the data using the elementary symmetric polynomials

$$\begin{aligned} s_1(\mathbf{x}) &= x_1 + x_2 + \cdots + x_n \\ s_2(\mathbf{x}) &= x_1x_2 + x_1x_3 + \cdots + x_1x_n \\ &\quad + x_2x_3 + \cdots + x_{n-1}x_n \\ &\vdots \\ s_n(\mathbf{x}) &= x_1x_2 \cdots x_n \end{aligned}$$

in the variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. This is accomplished by replacing our data $(x_1, \ldots, x_n) \in \mathbb{R}^n$ by the data $(s_1(\mathbf{x}), \ldots, s_n(\mathbf{x})) \in \mathbb{R}^n$. Details on this approach can be found in Martin *et al.* (2000).

We can also enforce component order symmetry for a general kernel $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. Denote by $\sigma_1, \ldots, \sigma_{n!}$ the various permutations of $(x_1, \ldots, x_n)$. In the case of $\mathbb{R}^2$, for example, we have $\sigma_1(x_1, x_2) = (x_1, x_2)$ and $\sigma_2(x_1, x_2) = (x_2, x_1)$. Then κ induces a symmetric kernel $\kappa_s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined by

$$\begin{aligned} \kappa_s(\mathbf{x}, \mathbf{y}) &= \kappa(\sigma_1(\mathbf{x}), \mathbf{y}) + \kappa(\sigma_2(\mathbf{x}), \mathbf{y}) \\ &\quad + \cdots + \kappa(\sigma_{n!}(\mathbf{x}), \mathbf{y}). \end{aligned}$$

8. APPLICATION TO MATERIALS DESIGN

Here we apply our methods to a classification problem in materials design. Specifically, a large number of chemical element combinations have been collected (by Villars, 1999) which through different processes either form or fail to form compounds. We want to use these examples to predict when other chemical element combinations will form compounds.

Our work on this problem includes feature and kernel selection on two, three, and four element combinations (1333, 4963, and 4278 examples, respectively) using a list of 88 possible features. This data was organized by Pao (1999*a*) and includes his N1 and N2 orderings (Pao, 1999*b*) in the list of features.

We organize our findings using the notation of sections 5 and 6. That is, we use $J(\mathbf{a}^*)$ to denote the optimal value of Fisher's criterion obtained using training data, p to denote the percentage of training data correctly classified using Fisher's discriminant, and t to denote the percentage of test data correctly classified. In addition, we use $\overline{J(\mathbf{a}^*)}_r, p$, and t to denote the averages of the values $J(\mathbf{a}^*), p$, and t over the Veronese degrees $1, 2, \ldots, r$.

Table 1. Top 3 Binary Features

	Name	$\overline{J(\mathbf{a}^*)}_{10}$	$\bar{p}$	$\bar{t}$
1	M13 Paos N2	3.14	85.76	83.82
2	M2 M. H t-d st. rt.	2.77	85.27	84.78
3	M6 M. Pettifor	2.69	85.03	83.77

Table 2. Top 3 Binary Feature Pairs

	Names	$\overline{J(\mathbf{a}^*)}_7$	$\bar{p}$	$\bar{t}$
1	M13 Paos N2 G2 val. elect. #	6.40	90.56	88.58
2	M4 M. H d-t st. rt. G2 val. elect. #	5.44	90.28	89.49
3	M5 M. Pettifor G2 val. elect. #	5.41	90.05	87.88

Table 3. Kernel Comparisons for Top 3 Binary Features

	F.	Best Kernel	$J(\mathbf{a}^*)$	p	t
1	M13	Veronese 3	2.42	84.77	82.80
		RBF .5	5.61	92.27	91.17
		Net 7.7, 5.2	1.35	76.22	73.41
2	M2	Veronese 3	2.18	85.22	84.68
		RBF .25	7.38	92.12	91.47
		Net 4.4, 4.7	1.18	71.94	70.81
3	M6	Veronese 3	2.10	84.44	82.51
		RBF .45	5.08	91.82	90.03
		Net 18.1, 3	1.15	76.97	75.87

These values are calculated and used to produce ranked lists of features and kernel comparisons as specified in sections 5 and 6.

8.1 *Binary Case*

Feature selection in the binary case yields Tables 1 and 2. These tables were obtained using steps (1)-(3) in section 5, where symmetric Veronese degrees $1, \ldots, 10$ were considered in the case of single features, and degrees $1, \ldots, 7$ were considered in the case of feature pairs.

Kernel selection in the binary case yields Tables 3 and 4. These tables were obtained from steps (4)-(6) in section 6 using the symmetric versions of the Veronese, RBF, and neural network kernels. We include for the best three features and feature pairs the best of each type of kernel considered. We use notation such as Veronese 3 to denote the Veronese kernel with $d = 3$ and Net 7.7, 5.2 to denote the neural net kernel with $a = 7.7, b = 5.2$.

Of the three cases in the materials problem, the binary case offers the best opportunity for improvement in prediction rates. This may be due to the fact that the binary data is the most complete data set, i.e., that the binary data includes a higher percentage of the possible combinations than either the ternary or quaternary sets. As such, there are fewer "degrees of freedom" to use

Table 4. Kernel Comparisons for Top 3 Binary Feature Pairs

	F.P.	Best Kernel	$J(\mathbf{a}^*)$	p	t
1	M13	Veronese 3	3.97	91.00	89.88
	G2	RBF .65	12.1	95.87	93.06
		Net 14.8, 6.4	1.50	83.57	82.94
2	M4	Veronese 3	3.71	90.02	90.16
	G2	RBF .25	26.2	98.35	92.20
		Net 16.8, 7	.739	76.67	76.88
3	M5	Veronese 3	3.50	89.00	88.87
	G2	RBF .2	35.0	95.42	91.91
		Net 7, 5	1.07	85.25	80.64

Table 5. Top 3 Ternary Features

	Name	$\overline{J(\mathbf{a}^*)}_7$	$\bar{p}$	$\bar{t}$
1	M13 Paos N2	14.02	95.30	95.51
2	M2 M. H t-d st. rt.	11.24	94.98	94.86
3	M5 M. Pettifor	10.86	94.64	94.98

Table 6. Top 3 Ternary Feature Pairs

	Names	$\overline{J(\mathbf{a}^*)}_5$	$\bar{p}$	$\bar{t}$
1	M13 Paos N2 G1 group #	21.0	95.76	95.59
2	M13 Paos N2 I25 ΔH int. O	19.1	95.99	95.81
3	M13 Paos N2 E2 electroneg.	18.7	96.07	96.09

when classifying the binary data. By including more features in our classifier, we may be able to introduce the needed degrees of freedom for classifying the binary data. Here we have introduced feature pairs and already see an improvement over single features. In the future we will perform feature triple selection with the hope of further improvement.

The best feature/kernel combination is Pao's N2 ordering & valence electron number/symmetric RBF kernel with $\sigma = .65$.

8.2 *Ternary Case*

Feature selection in the ternary case yields Tables 5 and 6 again via steps (1)-(3) in section 5. In this case we used Veronese degrees $1, \ldots, 7$ and $1, \ldots, 5$ respectively.

Kernel comparisons for the top 3 ternary features are given in Table 7.

The prediction rates in the ternary case are very good. Any improvement will likely come from the actual implementation of the Support Vector Machines with the kernels selected above.

The best feature/kernel combination is Pao's N2 ordering/symmetric RBF kernel with $\sigma = .25$.

Table 7. Kernel Comparisons for Top 3 Ternary Features

	F.	Best Kernel	$J(\mathbf{a}^*)$	p	t
		Veronese 3	12.6	97.05	97.08
1	M13	RBF .25	80.1	99.37	98.79
		Net 12.6, 14.8	7.31	95.56	95.96
		Veronese 3	9.96	96.70	96.61
2	M3	RBF .3	46.3	98.92	98.24
		Net 18.2, 19	8.90	95.86	95.92
		Veronese 3	9.91	96.24	96.38
3	M5	RBF .25	56.8	99.23	98.42
		Net 11.8, 15.4	2.55	92.87	92.90

Table 8. Top 3 Quaternary Features

	Name	$\overline{J(\mathbf{a}^*)_5}$	$\bar{p}$	$\bar{t}$
1	M5 M. Pettifor	53.1	97.37	97.16
2	M13 Paos N2	51.3	97.24	97.10
3	E8 chem. pot.	40.4	97.32	97.60

Table 9. Kernel Comparisons for Top 3 Quaternary Features

	F.	Best Kernel	$J(\mathbf{a}^*)$	p	t
		Veronese 2	27.2	98.97	98.93
1	M5	RBF .45	508	99.96	99.88
		Net 15.6, 13.6	25.3	98.51	98.34
		Veronese 3	52.0	99.88	99.80
2	M13	RBF .4	656	100	100
		Net 7.6, 18.4	28.9	98.73	99.21
		Veronese 2	38.7	99.56	99.49
3	E8	RBF .55	299	99.88	99.80
		Net 4.4, 17.8	34.8	99.03	98.74

8.3 *Quaternary Case*

In the quaternary case we used the symmetric Veronese kernel with degrees $1, \ldots, 5$ for single feature selection. The results are given in Table 8.

Kernel comparisons for the quaternary case are given in Table 9.

The prediction rates for the quaternary case are so good that we didn't bother to produce results for the case of feature pairs.

The best feature/kernel combination is Pao's N2 ordering/symmetric RBF kernel with $\sigma = .4$.

9. CONCLUSIONS

This work is an extension of our previous efforts (Martin *et al.*, 2000). In particular, we were previously constrained to low dimensional/degree cases (cases in which we first ran up against the memory constraints of the computer) using the symmetric Veronese map. With the invention of kernel Gram-Schmidt, we can now handle higher dimensional problems (problems where we first run up against computational numerical precision constraints) using not only the symmetric Veronese map (via the symmetric Veronese kernel), but any map with an associated kernel.

We have applied kernel Gram-Schmidt coupled with Fisher's discriminant with success to the materials problem. In particular, we have:

(A) Confirmed the success of Pao's (1999*b*) new orderings of the periodic table in classifying the materials data.

(B) Achieved prediction rates comparable to those of Pao.

(C) Selected good features and kernels for use in training Support Vector Machines on the materials problem. In particular, we have discovered that Pao's N2 ordering with the symmetric RBF kernel (various parameters) works well in the binary, ternary and quaternary cases of the materials problem.

The main product of this work is of course (C). In future work we will implement the Support Vector machines associated with our feature/kernel selections made above. Preliminary work in this direction confirms that these optimal classifiers will achieve prediction rates even better than those obtained to date.

10. REFERENCES

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining.*

Burges, C. (1999). Geometry and invariance in kernel based methods. In: *Advances in Kernel Methods — Support Vector Learning* (B. Schölkopf, C. Burges and A. Smola, Eds.). MIT Press. Boston.

Duda, R.O. and P.E. Hart (1973). *Pattern Classification and Scene Analysis.* John Wiley & Sons, Inc.. New York.

Martin, S., M. Kirby and R. Miranda (2000). Symmetric veronese classifiers with application to materials design. *to appear in Engineering Applications of Artificial Intelligence.*

Pao, Yoh-Han (1999*a*).

Pao, Yoh-Han (1999*b*). Topological models of interacting systems and visualization (power point presentation).

Schürmann, J. (1996). *Pattern Classification: A Unified View of Statistical and Neural Approaches.* John Wiley & Sons, Inc.. New York.

Shafarevich, I.R. (1994). *Basic Algebraic Geometry 1.* Springer-Verlag. Berlin.

Trefethen, L. and D. Bau (1997). *Numerical Linear Algebra.* Society of Industrial and Applied Mathematics. Philadelphia.

Vapnik, V. (1998). *Statistical Learning Theory.* John Wiley & Sons, Inc.. New York.

Villars, P. (1999). Iterim report special project spc 98-4028 covering 15 march 1998 - 14 march 1999.