Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues

Olvi L. Mangasarian and Edward W. Wild

Abstract—A new approach to support vector machine (SVM) classification is proposed wherein each of two data sets are proximal to one of two distinct planes that are *not parallel* to each other. Each plane is generated such that it is closest to one of the two data sets and as far as possible from the other data set. Each of the two nonparallel proximal planes is obtained by a single MATLAB command as the eigenvector corresponding to a smallest eigenvalue of a generalized eigenvalue problem. Classification by proximity to two distinct nonlinear surfaces generated by a nonlinear kernel also leads to two simple generalized eigenvalue problems. The effectiveness of the proposed method is demonstrated by tests on simple examples as well as on a number of public data sets. These examples show the advantages of the proposed approach in both computation time and test set correctness.

Index Terms—Support vector machines, proximal classification, generalized eigenvalues.

1 Introduction

Support vector machines (SVMs) [23], [4], [27] constitute the method of choice for classification problems while the generalized eigenvalue problem [22], [5] is a simple problem of classical linear algebra solvable by a single command of MATLAB [17] or Scilab [24] or by using standard linear algebra software such LAPACK [1]. In proximal support vector classification [7], [25], [6], two *parallel* planes are generated such that each plane is closest to one of two data sets to be classified and the two planes are as far apart as possible. In the present work, we drop the parallelism condition on the proximal planes and require that each plane be as close as possible to one of the data sets and as far as possible from the other data set. This formulation leads to two generalized eigenvalue problems: $Gz = \lambda Hz$ and $Lz = \lambda Mz$, where G, H, L, and M are symmetric positive semidefinite matrices. Each of the nonparallel proximal planes is generated by an eigenvector corresponding to a smallest eigenvalue of each of the generalized eigenvalue problems. Application of this method to the classical XOR problem in two dimensions where the two sets are $\{[0\ \ 0], [1\ \ 1]\}$ and $\{[1\ \ 0], [0\ \ 1]\}$ leads to an exact classification by two nonparallel proximal lines each going through the two points of each set.

Related work is the k-plane clustering of [3], where clusters are determined by proximity to various nonparallel planes based on the smallest eigenvector of a matrix generated by given data points. We also note that, in [11], the generalized eigenvalue formulation was used for protein fold recognition to determine an optimal transformation of a permutation matrix based on simultaneously minimizing within-class variation and maximizing between-class variation of various protein folds.

This work is organized as follows: In Section 2, we briefly describe the general classification problem and our proximal multiplane linear kernel formulation as a generalized eigenvalue problem. In Section 3, we extend our proximal results to a proximal multisurface nonlinear kernel formulation. In Section 4, we test our new approach and compare it with standard linear and nonlinear kernel classifiers. Section 5 concludes the paper.

A word about our notation. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$, the 2-norm of x will be denoted by $\|x\|$. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e and the identity matrix of arbitrary order will be denoted by I. The gradient of a differentiable function f on R^n is defined as the column vector of first partial derivatives: $\nabla f(x) := [\frac{\partial f(x)}{\partial x_1}, \ldots, \frac{\partial f(x)}{\partial x_n}]'$. For $A \in R^{m\times n}$ and $B \in R^{n\times k}$, a *kernel* $K(A, B)$ maps $R^{m\times n} \times R^{n\times k}$ into $R^{m\times k}$. In particular, if x and y are column vectors in R^n, then $K(x', y)$ is a real number, $K(x', A')$ is a row vector in R^m, and $K(A, A')$ is an $m \times m$ matrix. We shall make no assumptions on our kernels other than symmetry, that is, $K(x', y)' = K(y', x)$ and, in particular, we shall not assume or make use of Mercer's positive definiteness condition [27], [23]. The base of the natural logarithm will be denoted by ε. A frequently used kernel in nonlinear classification is the Gaussian kernel [27], [15] whose ijth element, $i = 1 \ldots, m,\ j = 1 \ldots, k$, is given by: $(K(A, B))_{ij} = \varepsilon^{-\mu\|A_i' - B_{\cdot j}\|^2}$, where $A \in R^{m\times n}$, $B \in R^{n\times k}$, and μ is a positive constant.

2 The Multiplane Linear Kernel Classifier

We consider the problem of classifying m points in the n-dimensional real space R^n, represented by the $m_1 \times n$

- *O.L. Mangasarian is with the Computer Sciences Department, University of Wisconsin, Madison, WI 53706, and the Department of Mathematics, University of California at San Diego, La Jolla, CA 92093. E-mail: olvi@cs.wisc.edu.*
- *E.W. Wild is with the Computer Sciences Department, University of Wisconsin, Madison, WI 53706. E-mail: wildt@cs.wisc.edu.*

Manuscript received 28 Oct. 2004; revised 21 Mar. 2005; accepted 6 Apr. 2005; published online 13 Oct. 2005.
Recommended for acceptance by S.K. Pal.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0586-1004.

matrix A belonging to class 1 and the $m_2 \times n$ matrix B belonging to class 2, with $m_1 + m_2 = m$. For this problem, a standard support vector machine with a linear classifier [15], [23] is given by a plane midway between two *parallel* bounding planes that bound two disjoint halfspaces each containing points mostly of class 1 or 2. In another somewhat less standard approach, the proximal support vector classification [7], [25], [6], two *parallel* planes are generated such that each plane is closest to one of two data sets to be classified and such that the two planes are as far apart as possible. The classifying plane is again midway between the parallel proximal planes. Thus, PSVM [7] classifies points on the basis of proximity to two *parallel* planes: $x'w = \gamma + 1$ and $x'w = \gamma - 1$ that are determined by the unconstrained minimization of a quadratic function in the $n + 1$ variables $w \in R^n, \gamma \in R$, for some $\nu > 0$ [7] as follows:

$$\min_{(w,\gamma)\in R^{n+1}} \frac{\nu}{2}\left(\|e-(Aw-e\gamma)\|^2 + \|e+(Bw-e\gamma)\|^2\right) + \frac{1}{2}\left\|\begin{bmatrix} w \\ \gamma \end{bmatrix}\right\|^2. \tag{1}$$

In the present work, we drop the parallelism condition on the proximal planes and require that each plane be as close as possible to one of the data sets and as far as possible from the other one. Thus, we are seeking two planes in R^n:

$$x'w^1 - \gamma^1 = 0,\ x'w^2 - \gamma^2 = 0, \tag{2}$$

where the first plane is closest to the points of class 1 and furthest from the points in class 2, while the second plane is closest to the points in class 2 and furthest from the points in class 1. To obtain the first plane of (2), we minimize the sum of the squares of two-norm distances between each of the points of class 1 to the plane divided by the squares of two-norm distances between each of the points of class 2 to the plane. This leads to the following optimization problem:

$$\min_{(w,\gamma)\neq 0} \frac{\|Aw-e\gamma\|^2 / \|\begin{bmatrix} w \\ \gamma \end{bmatrix}\|^2}{\|Bw-e\gamma\|^2 / \|\begin{bmatrix} w \\ \gamma \end{bmatrix}\|^2}, \tag{3}$$

where $\|\cdot\|$ denotes the two-norm and it is implicitly assumed that $(w,\gamma) \neq 0 \Longrightarrow Bw - e\gamma \neq 0$. This assumption will be made explicit below. We note that the numerator of the minimization problem (3) is the sum of squares of two-norm distances in the (w,γ)-space of points in class 1 to the plane $x'w - \gamma = 0$, while the denominator of (3) is the sum of squares of two-norm distances in the (w,γ)-space of points in class 2 to the same plane [14]. Simplifying (3) gives:

$$\min_{(w,\gamma)\neq 0} \frac{\|Aw-e\gamma\|^2}{\|Bw-e\gamma\|^2}. \tag{4}$$

We now introduce a Tikhonov regularization term [26] that is often used to regularize least squares and mathematical programming problems [16], [13], [6], [25] that reduces the norm of the problem variables (w,γ) that determine the proximal planes (2). Thus, for a nonnegative parameter δ, we regularize our problem (4) as follows:

$$\min_{(w,\gamma)\neq 0} \frac{\|Aw-e\gamma\|^2 + \delta\|\begin{bmatrix} w \\ \gamma \end{bmatrix}\|^2}{\|Bw-e\gamma\|^2}. \tag{5}$$

A possible geometric interpretation of the formulation (5) is that the first equation of (2) is obtained as a closest plane to the data set represented by A with distances to points of A normalized by the sum of the distances to the points of B. By making the definitions:

$$\begin{aligned} G &:= [A \quad -e]'[A \quad -e] + \delta I, \\ H &:= [B \quad -e]'[B \quad -e], z := \begin{bmatrix} w \\ \gamma \end{bmatrix}, \end{aligned} \tag{6}$$

the optimization problem (4) becomes:

$$\min_{z\neq 0} r(z) := \frac{z'Gz}{z'Hz}, \tag{7}$$

where G and H are symmetric matrices in $R^{(n+1)\times(n+1)}$. The objective function of (7) is known as the *Rayleigh quotient* [22, p. 357] and has some very useful properties which we now cite.

Theorem 2.1 [22, Theorem 15.9.2] (Rayleigh Quotient Properties). *Let G and H be arbitrary symmetric matrices in $R^{(n+1)\times(n+1)}$. When H is positive definite, the Rayleigh quotient of (7) enjoys the following properties:*

1. *(**Boundedness**) The Rayleigh quotient ranges over the interval $[\lambda_1, \lambda_{n+1}]$ as z ranges over the unit sphere, where λ_1 and λ_{n+1} are the minimum and maximum eigenvalues of the generalized eigenvalue problem:*
$$Gz = \lambda Hz, \quad z \neq 0. \tag{8}$$
2. *(**Stationarity**)*
$$\nabla r(z) = 2\frac{(Gz - r(z)Hz)}{z'Hz} = 0. \tag{9}$$
Thus, $r(z)$ is stationary at and only at the eigenvectors of the generalized eigenvalue problem (8).

We note the following consequence of this theorem: Under the rather unrestrictive assumption that the columns of the matrix $[B \quad -e]$ are linearly independent, the global minimum of problem (7) is achieved at an eigenvector of the generalized eigenvalue problem (8) corresponding to a smallest eigenvalue λ_1. If we denote this eigenvector by z^1, then $[w^{1\prime} \quad \gamma^1]' = z^1$ determines the plane $w^{1\prime}x - \gamma^1 = 0$ of (2) which is closest to all the points of data set 1 and furthest away from the points of data set 2.

By an entirely similar argument, we define an analogous minimization problem to (5) for determining (w^2, γ^2) for the plane $x'w^2 - \gamma^2 = 0$ of (5) which is closest to the points of set 2 and furthest from set 1 as follows:

$$\min_{(w,\gamma)\neq 0} \frac{\|Bw-e\gamma\|^2 + \delta\|\begin{bmatrix} w \\ \gamma \end{bmatrix}\|^2}{\|Aw-e\gamma\|^2}. \tag{10}$$

By defining:

$$L := [B \quad -e]'[B \quad -e] + \delta I, M := [A \quad -e]'[A \quad -e], \tag{11}$$

and z as in (6), the optimization problem (10) becomes:

$$\min_{z\neq 0} s(z) := \frac{z'Lz}{z'Mz}, \tag{12}$$

where L and M are again symmetric matrices in $R^{(n+1)\times(n+1)}$. The minimum of (12) is achieved at an eigenvector corresponding to a smallest eigenvalue of the generalized eigenvalue problem:

$$Lz = \lambda Mz,\ z \neq 0. \tag{13}$$

We can now state the following theorem.

Theorem 2.2 (Proximal Multiplane Classification). *Let $A \in R^{m_1\times n}$ represent the data set of class 1 and $B \in R^{m_2\times n}$ represent the data set of class 2. Define G, H, L, M, and z as in (6) and (11). Assume that $[A \quad -e]$ and $[B \quad -e]$ have linearly independent columns. Then, the proximal planes (2) are obtained by the two MATLAB [17] commands: $eig(G, H)$ and $eig(L, M)$, each of which generates $n+1$ eigenvalues and eigenvectors of the generalized eigenvalue problems (8) and (13). The proximal planes (2) are obtained by:*

$$\begin{bmatrix} w^1 \\ \gamma^1 \end{bmatrix} = z^1,\ \begin{bmatrix} w^2 \\ \gamma^2 \end{bmatrix} = z^2, \tag{14}$$

where z^1 is an eigenvector of the generalized eigenvalue problem (8) corresponding to a smallest eigenvalue and z^2 is an eigenvector of the generalized eigenvalue problem (13) corresponding to a smallest eigenvalue.

We note that the linear independence condition is not restrictive for a great many classification problems for which $m_1 >> n$ and $m_2 >> n$. We also note that it is merely a sufficient but not a necessary condition for the above theorem to hold. Thus, in the XOR example given below, the linear independence condition is not satisfied. However, we are able to obtain a perfect two-plane classifier using Theorem 2.2 above.

Example 2.3 (Zero-Error XOR Classifier). Given the matrices:

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{15}$$

we define G, H of (6) and L, M of (11) as follows:

$$G = M = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix},\ H = L = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{bmatrix}. \tag{16}$$

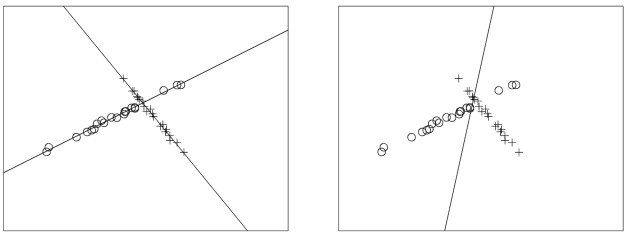
Then, the generalized eigenvalue problems (8) and (13) have the following respective minimum eigenvalues and corresponding eigenvectors:

$$\begin{array}{ll} \lambda_1 = 0, & z^{1\prime} = [1 \quad -1 \quad 0], \\ \lambda_2 = 0, & z^{2\prime} = [1 \quad 1 \quad 1]. \end{array} \tag{17}$$

These give two lines (planes) in R^2, each of which containing the two data points from one set only:

$$x_1 - x_2 = 0,\ x_1 + x_2 = 1. \tag{18}$$

We note that a standard one-norm linear SVM generates a single line classifier for the XOR example that misclassifies one point [2]. Thus, proximal separability does not imply linear separability nor is the converse true. However, it is also possible for two sets to be both proximally and linearly separable.



GEPSVM: 100% correct Linear PSVM: 80% correct

Fig. 1. The "cross planes" learned by GEPSVM and the decision boundary learned by a one-norm linear SVM together with their correctness on the training data set.

We give another simple example to visually illustrate the effectiveness of our generalized eigenvalue proximal SVM (GEPSVM). We call this example "Cross Planes" because the data is obtained by perturbing points originally lying on two intersecting planes (lines).

Example 2.4 (Cross Planes Classifier). The data consists of points that are close to one of two intersecting "cross planes" in R^2. Fig. 1 illustrates the data set and the planes found by GEPSVM. We note that training set correctness for GEPSVM is 100 percent and for PSVM is 80 percent. We note that this example, which is a perturbed generalization of the XOR example, can serve as a difficult test case for typical linear classifiers just as the XOR example does. The reason PSVM did so poorly in comparison to GEPSVM on this example is because its proximal planes have to be parallel, meaning that PSVM generates a single linear classifier plane midway between the two proximal planes, by looking at the data points in Fig. 1, it is obvious that the single plane of PSVM cannot do as well as the nonparallel planes of GEPSVM.

We turn now to multisurface nonlinear classification.

3 The Multisurface Nonlinear Kernel Classifier

To extend our results to nonlinear multisurface classifiers, we consider the following kernel-generated proximal surfaces instead of the planes (2):

$$K(x', C')u^1 - \gamma^1 = 0,\ K(x', C')u^2 - \gamma^2 = 0, \tag{19}$$

where

$$C := \begin{bmatrix} A \\ B \end{bmatrix} \tag{20}$$

and K is an arbitrary kernel as defined in Section 1. We note that the planes of (2) are a special case of (19) if we use a linear kernel $K(x', C) = x'C$ and define $w^1 = C'u^1$ and $w^2 = C'u^2$. By using the same arguments as those of Section 2, our minimization problem for generating a kernel-based nonlinear surface that is closest to one data set and furthest from the other leads to the minimization problem that generalizes (5) to the following:

$$\min_{(u,\gamma)\neq 0} \frac{\|K(A, C')u - e\gamma\|^2 + \delta\|\left[\begin{smallmatrix} u \\ \gamma \end{smallmatrix}\right]\|^2}{\|K(B, C')u - e\gamma\|^2}. \tag{21}$$

By making the definitions:

$$\begin{array}{rcl} G & := & [K(A,C') \quad -e]'[K(A,C') \quad -e] + \delta I, \\ H & := & [K(B,C') \quad -e]'[K(B,C') \quad -e], \end{array} \tag{22}$$

where G and H are now matrices in $R^{(m+1)\times(m+1)}$, the optimization problem (21) becomes:

$$\min_{z\neq 0} r(z) := \frac{z'Gz}{z'Hz}, \text{ where } z := \begin{bmatrix} u \\ \gamma \end{bmatrix}, \tag{23}$$

which is exactly the same as problem (7), but with different definitions for G and H.

By reversing the roles of $K(A,C')$ and $K(B,C')$ in (21), we obtain the following minimization problem for the proximal surface $K(x',C')u^2 - \gamma^2 = 0$ of (19):

$$\min_{(u,\gamma)\neq 0} \frac{\|K(B,C')u - e\gamma\|^2 + \delta\|[{}^{u}_{\gamma}]\|^2}{\|K(A,C')u - e\gamma\|^2}. \tag{24}$$

By defining:

$$\begin{array}{rcl} L & := & [K(B,C') \quad -e]'[K(B,C') \quad -e] + \delta I, \\ M & := & [K(A,C') \quad -e]'[K(A,C') \quad -e], \end{array} \tag{25}$$

where L and M are again matrices in $R^{(m+1)\times(m+1)}$, the optimization problem (24) becomes:

$$\min_{z\neq 0} s(z) := \frac{z'Lz}{z'Mz}, \text{ where } z := \begin{bmatrix} u \\ \gamma \end{bmatrix}, \tag{26}$$

which is exactly the same as problem (12), but with different definitions for L and M.

An analogous theorem to Theorem 2.2 can now be given for solving (23) and (26).

Theorem 3.1 (Proximal Nonlinear Multisurface Classification). *Let $A \in R^{m_1\times n}$ represent the data set of class 1 and $B \in R^{m_2\times n}$ represent the data set of class 2. Define G, H, L, M, and z as in (22), (23), and (25). Assume that $[K(B,C') \quad -e]$ and $[K(A,C') \quad -e]$ have linearly independent columns. Then, the proximal surfaces (19) are obtained by the two MATLAB [17] commands: $eig(G,H)$ and $eig(L,M)$, each of which generates the $m+1$ eigenvalues and eigenvectors of the respective generalized eigenvalue problems:*

$$Gz = \lambda Hz, \quad z \neq 0, \tag{27}$$

and

$$Lz = \lambda Mz, \quad z \neq 0, \tag{28}$$

The proximal surfaces (27) are obtained by:

$$\begin{bmatrix} u^1 \\ \gamma^1 \end{bmatrix} = z^1, \ \begin{bmatrix} u^2 \\ \gamma^2 \end{bmatrix} = z^2, \tag{29}$$

where z^1 is an eigenvector of the generalized eigenvalue problem (19) corresponding to a smallest eigenvalue, and z^2 is an eigenvector of the generalized eigenvalue problem (28) corresponding to a smallest eigenvalue.

We note immediately that, if either m_1 or m_2 are large, the techniques of the reduced support vector machine classification [12] can be easily applied to reduce the dimensionality $m+1 = m_1 + m_2 + 1$ of the generalized eigenvalue problem (27) to $\bar{m}+1$ by replacing the kernels $K(A,C')$, $K(B,C')$ by the reduced kernels $K(A,\bar{C}')$, $K(B,\bar{C}')$, respectively, where $\bar{C}$ is matrix formed by taking a small random sample of the rows of C.

We turn to our numerical tests and comparisons now.

TABLE 1
Linear Kernel GEPSVM, PSVM [7], and SVM-Light [9]
10-Fold Testing Correctness and p-Values

Data Set $m \times n$	GEPSVM Correctness	PSVM Correctness p-value	SVM-Light Correctness p-value
Cross Planes 300×7	**98.0%**	55.3%* 5.24671e-07	45.7%* 1.4941e-08
NDC 300×7	86.7%	88.3% 0.244333	**89.0%** 0.241866
Cleveland Heart 297×13	81.8%	**85.2%** 0.112809	83.6% 0.485725
Cylinder Bands 540×35	71.3%	71.7% 0.930192	**76.1%** 0.229676
Pima Indians 768×8	73.6%	**75.9%** 0.274187	75.7% 0.380633
Spambase 4601×57	76.8%	**77.1%** 0.0654478	**77.1%** 0.0654478
Galaxy Bright 2462×14	**98.6%**	97.3%* 0.031226	98.3% 0.506412
Mushroom 8124×22	81.1%	80.9% 0.722754	**81.5%** 0.356003

The p-values are from a t-test comparing each algorithm to GEPSVM. Best correctness results are in bold. An asterisk (*) denotes a significant difference from GEPSVM based on p-values less than 0.05.

4 Numerical Testing and Comparisons

To demonstrate the performance of our approach, we report results on publicly available data sets from the UCI Repository [19] and from [21], as well as two synthetic data sets. One synthetic data set is Musicant's NDC [20] and the other is a simple extension of our "Cross Planes" example above to R^7. Table 1 shows a linear kernel comparison of GEPSVM versus PSVM [7] and SVM-Light [9]. For a linear kernel, all three algorithms have a single parameter: δ for GEPSVM, ν for PSVM, and C for SVM-Light. This parameter was selected from the values $\{10^i | i = -7, -6, \ldots, 7\}$ by using 10 percent of each training fold as a tuning set. For GEPSVM only, this tuning set was not returned to the training fold to learn the final classifier once the parameter was selected. This choice was made by observing the performance of all three classifiers on data sets not shown here. GEPSVM tended to perform better without retraining on the entire training fold, while the other two algorithms benefited from the additional data. In addition to reporting the average accuracies across the 10 folds, we performed paired t-tests [18] comparing PSVM to GEPSVM and SVM-Light to GEPSVM. The p-value for each test is the probability of the observed or a greater difference between two test set correctness values occurring, under the assumption of the null hypothesis that there is no difference between the test set correctness distributions. Thus, the smaller the p-value, the less likely that the observed difference resulted from identical test set correctness distributions. A typical threshold for p-values is 0.05. For example, the p-value of the test comparing GEPSVM and PSVM on the Galaxy Bright data set was 0.031226, which is less than 0.05, leading us

TABLE 2
Nonlinear Kernel GEPSVM, PSVM [7], and SVM-Light [9] 10-Fold Testing Correctness and p-Values

Data Set $m \times n$	GEPSVM Correctness	PSVM Correctness p-value	SVM-Light Correctness p-value
Cross Planes 300×7	**99.0%**	73.7%* 0.00025868	79.3%* 8.74044e-06
WPBC (60 mo.) 110×32	62.7%	**64.5%** 0.735302	63.6% 0.840228
BUPA Liver 345×6	63.8%	67.9% 0.190774	**69.9%** 0.119676
Votes 435×16	94.2%	94.7% 0.443332	**95.6%** 0.115748
Haberman's Survival 306×3	75.4%	**75.8%** 0.845761	71.7% 0.0571092

The p-values were calculated using a t-test comparing each algorithm to GEPSVM. Best results are in bold. An asterisk (*) denotes a significant difference from GEPSVM based on p-values less than 0.05.

to conclude that GEPSVM and PSVM have different accuracies on this data set. We note that, on the NDC and real world data sets, the performance difference between GEPSVM and the other algorithms is statistically insignificant, with the exception of Galaxy Bright, where GEPSVM is significantly better than PSVM. This indicates that allowing the proximal planes to be nonparallel allows the classifier to better represent this data set when needed.

Table 2 compares GEPSVM, PSVM, and SVM-Light using a Gaussian kernel. The kernel parameter μ was chosen from the values $\{10^i | i = -4, -3, -2, -1\}$ for all three algorithms. The parameter ν for PSVM and C for SVM-Light was selected from the set $\{10^i | i = -4, -3, \ldots, 2\}$, while the parameter δ for GEPSVM was selected from the set $\{10^i | i = -2, -1, \ldots, 4\}$. Parameter selection was done by comparing the accuracy of each combination of parameters on a tuning set consisting of a random 10 percent of each training set. As in the linear kernel comparison above, this tuning set was not returned to the training fold to retrain the classifier before evaluating on the test fold for GEPSVM, but was for PSVM and SVM-Light. We note that GEPSVM has performance that is comparable to PSVM and SVM-Light on the real world data sets and the difference between GEPSVM and the other algorithms is not statistically significant on these data sets. As expected, nonlinear GEPSVM greatly outperformed nonlinear PSVM and SVM-Light on the Cross Planes data set.

Table 3 contains a typical sample of the computation times of the three methods compared in Table 1. We report the average of times to learn the linear kernel classifier for each fold with the parameter selected by the tuning procedure described above on the Cylinder Bands data set [19]. These times were obtained on a machine running Matlab 7 on Red Hat Linux 9.0 with a Pentium III 650MHz processor and 256 megabytes of memory. Complexity of the generalized eigenvalue problem is of order n^3 [8, Section 7.7] which is similar to that of solving the system of linear equations resulting from PSVM, although the constant multiplying $O(n^3)$ for the generalized eigenvalue problem is larger. For an interior point method used for solving a two-norm SVM quadratic program, the complexity is of order $n^{3.5}$ based on a linear complementarity problem formulation of the quadratic program [10]. These facts help explain the computation times of Table 3, where PSVM is over one order of magnitude faster than GEPSVM, which is nearly two orders of magnitude faster than SVM-Light.

TABLE 3
Average Time to Learn One Linear Kernel GEPSVM, PSVM [7], and SVM-Light [9] on the Cylinder Bands Data Set [19]

GEPSVM Time (seconds)	PSVM Time (seconds)	SVM-Light Time (seconds)
0.96	**0.08**	75.4

As final remarks, we note that, for our multiplane linear kernel classifiers of Section 2, very large data sets can be handled by GEPSVM provided the input space dimension n is moderate in size, say of the order of a few hundred. This is so because the generalized eigenvalue problem (8) is in the space R^{n+1}. Thus, even for two randomly generated matrices G and H of the order of $1,000 \times 1,000$, MATLAB was able to solve the generalized eigenvalue problem (8) in less than 75 seconds on a Pentium 4 1.7Ghz machine. For our multisurface nonlinear kernel classifiers of Section 3, the reduced kernel techniques of [12] can be used to handle such data sets as discussed at the end of Section 3.

5 Conclusion and Outlook

We have proposed a novel approach to classification problems that relaxes the universal requirement that bounding or proximal planes generated by SVMs be parallel in the input space for linear kernel classifiers or in the higher dimensional feature space for nonlinear kernel classifiers. Each of our proposed nonparallel proximal planes is easily obtained using a single MATLAB command that solves the classical generalized eigenvalue problem. Classification accuracy results are comparable to those of classical support vector classification algorithms and, in some cases, they are better. Also, in our experience. the generalized eigenvalue problem can be solved more quickly than the optimization algorithm needed for SVM-Light. The simple program formulation, computational efficiency, and accuracy of GEPSVM on real world data indicate that it is an effective algorithm for classification. Analysis of the statistical properties of GEPSVM and extensions to multicategory classification are promising areas of future research.

Acknowledgments

This research supported by US National Science Foundation Grant CCR-0138308, Public Health Services Grant 5 T15 LM07359-02, by Microsoft, and by ExxonMobil. Data Mining Institute Report 04-03, June 2004. Revised September 2004 and March 2005.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide*, third ed., Philadelphia: SIAM, 1999, http://www.netlib.org/lapack/.

[2] K.P. Bennett and O.L. Mangasarian, "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets," *Optimization Methods and Software*, vol. 1, pp. 23-34, 1992.

[3] P.S. Bradley and O.L. Mangasarian, "k-Plane Clustering," *J. Global Optimization,* vol. 16, pp. 23-32, 2000, ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-08.ps.
[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines.* Cambridge, Mass.: Cambridge Univ. Press, 2000.
[5] J.W. Demmel, *Applied Numerical Linear Algebra.* Philadelphia: SIAM, 1997.
[6] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization Networks and Support Vector Machines," *Advances in Computational Math.,* vol. 13, pp. 1-50, 2000.
[7] G. Fung and O.L. Mangasarian, "Proximal Support Vector Machine Classifiers," *Proc. Knowledge Discovery and Data Mining,* F. Provost and R. Srikant, eds., pp. 77-86, 2001, ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.
[8] G.H. Golub and C.F. Van Loan, *Matrix Computations,* third ed. Baltimore: The John Hopkins Univ. Press,. 1996.
[9] T. Joachims, "Making Large-Scale Support Vector Machine Learning Practical," *Advances in Kernel Methods—Support Vector Learning,* B. Schölkopf, C.J.C. Burges, and A.J. Smola, eds., pp. 169-184, Cambridge, Mass.: MIT Press, 1999.
[10] M. Kojima, S. Mizuno, T. Noma, and A. Yoshise, *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.* Berlin: Springer-Verlag, 1991.
[11] R.H. Leary, J.B. Rosen, and P. Jambeck, "An Optimal Structure-Discrminative Amino Acid Index for Protein Recognition," *Biophysical J.,* vol. 86, pp. 411-419, 2004.
[12] Y.-J. Lee and O.L. Mangasarian, "RSVM: Reduced Support Vector Machines," *Proc. First SIAM Int'l Conf. Data Mining,* Apr. 2001, ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/00-07.ps.
[13] O.L. Mangasarian, "Least Norm Solution of Non-Monotone Complementarity Problems," *Functional Analysis, Optimization and Mathematical Economics,* pp. 217-221, New York: Oxford Univ. Press, 1990.
[14] O.L. Mangasarian, "Arbitrary-Norm Separating Plane," *Operations Research Letters,* vol. 24, pp. 15-23, 1999, ftp://ftp.cs.wisc.edu/math-prog/tech-reports/97-07r.ps.
[15] O.L. Mangasarian, "Generalized Support Vector Machines," *Advances in Large Margin Classifiers,* A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds., pp. 135-146, Cambridge, Mass.: MIT Press, 2000, ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.
[16] O.L. Mangasarian and R.R. Meyer, "Nonlinear Perturbation of Linear Programs," *SIAM J. Control and Optimization,* vol. 17, no. 6, pp. 745-752, Nov. 1979.
[17] *MATLAB, User's Guide,* The MathWorks, Inc., 1994-2001, http://www.mathworks.com.
[18] T.M. Mitchell, *Machine Learning.* Boston: McGraw-Hill, 1997.
[19] P.M. Murphy and D.W. Aha, "UCI Machine Learning Repository," 1992, www.ics.uci.edu/mlearn/MLRepository.html.
[20] D.R. Musicant, "NDC: Normally Distributed Clustered Datasets," 1998, www.cs.wisc.edu/musicant/data/ndc/.
[21] S. Odewahn, E. Stockwell, R. Pennington, R. Humphreys, and W. Zumach, "Automated Star/Galaxy Discrimination with Neural Networks," *Astronomical J.,* vol. 103, no. 1, pp. 318-331, 1992.
[22] B.N. Parlett, *The Symmetric Eigenvalue Problem.* Philadelphia: SIAM, 1998.
[23] B. Schölkopf and A. Smola, *Learning with Kernels.* Cambridge, Mass.: MIT Press, 2002.
[24] Scilab, free scientific software package, 1990-2004, http://scilabsoft.inria.fr.
[25] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines.* Singapore: World Scientific Publishing Co., 2002.
[26] A.N. Tikhonov and V.Y. Arsen, *Solutions of Ill-Posed Problems.* New York: John Wiley & Sons, 1977.
[27] V.N. Vapnik, *The Nature of Statistical Learning Theory,* second ed. New York: Springer, 2000.

Olvi L. Mangasarian received the PhD degree in applied mathematics from Harvard University, and worked for eight years as a mathematician for Shell Oil Company in California before coming to the University of Wisconsin at Madison. He is now the John von Neumann Professor Emeritus of Mathematics and Computer Sciences at the University of Wisconsin at Madison and a research scientist in the Department of Mathematics at the University of California at San Diego. His main research interests are in mathematical optimization, machine learning, and data mining. He is the author of the book *Nonlinear Programming*, coeditor of four books, and an associate editor of two journals. His recent papers are available at http://www.cs.wisc.edu/~olvi and http://www.cs.wisc.edu/dmi.

Edward W. Wild received the BS degree in computer sciences from the University of Texas at Austin and the MS degree in computer sciences from the University of Wisconsin-Madison. He is now a PhD student and research assistant under Professor Mangasarian. His research interest is the application of techniques in optimization to problems in machine learning and his recent papers are available at http://www.cs.wisc.edu/~wildt.