Journal of the Operations Research
Society of Japan
2003, Vol. 46, No. 4, 395-408

EXTRACTING FEATURE SUBSPACE FOR KERNEL BASED LINEAR PROGRAMMING SUPPORT VECTOR MACHINES

Yasutoshi Yajima
Tokyo Institute of Technology

Hiroko Ohi
Hitachi, Ltd.

Masao Mori
Keio University

(Received January 15, 2002; Revised June 24, 2003)

Abstract We propose linear programming formulations of support vector machines (SVM). Unlike standard SVMs which use quadratic programs, our approach explores a fairly small dimensional subspace of a feature space to construct the nonlinear discriminator. This allows us to obtain the discriminator by solving a smaller sized linear program. We demonstrate that an orthonormal basis of the subspace can be implicitly treated by eigenvectors of the Gram matrix defined by the associated kernel function. When the number of given data points is very large, we construct a subspace by random sampling of data points. Numerical experiments indicate that the subspace generated by less than 2% of the entire training data points achieves reasonable performance for a fairly large instance with 60000 data points.

Keywords: Support vector machine, kernel function, nonlinear discriminator, feature space, data mining, linear programming

1. Introduction

Recently, support vector machines (SVM) have been studied extensively for classification problems in a variety of areas, such as text categorization [11, 14] and image recognition [8, 22, 23]. In classification problems, a number of labeled data points, called a training dataset, $x_1, x_2, \ldots, x_M$, are given as a set of N dimensional vectors. We assume that a binary class label $y_i \in \{+1, -1\}$ is assigned to each point x_i. SVM generates a linear discriminate function $f(x) = w^T x - b$ with a normal vector $w \in R^N$ and a real number b, which maps a vector $x \in R^N$ to a class $y \in \{+1, -1\}$ as follows:

$$y = \begin{cases} +1 & \text{if } f(x) > 0, \\ -1 & \text{if } f(x) < 0. \end{cases}$$

Statistical learning theory [26] indicates that a discriminate function which balances accuracy and capacity must be generated in order to reduce generalization error. To this end, SVM employs convex quadratic programs [15, 21]. The convex quadratic program also plays an important role to generate *nonlinear* discriminate functions. In order to obtain a nonlinear discriminate function, we consider a nonlinear function ϕ which maps the data point x into a higher, often infinite dimensional *feature space* $\mathcal{F}$, in which linear discriminate functions are developed. The problem of finding a separating hyperplane with the largest margin reduces to a quadratic program. The Wolfe dual [17] formulation enables us to formulate the quadratic optimization problems only by the dot product $\langle \phi(x), \phi(x') \rangle$ in $\mathcal{F}$, which can be computed directly from the original data points, x and x', by the *kernel functions*. The crucial point is that the quadratic optimization problems can be formulated

without involving the explicit calculations of the mapped image $\phi(x) \in \mathcal{F}$. For instance, polynomial kernels

$$\mathcal{K}(x, x') = \left(x^T x'\right)^d,$$

with an integer parameter d, RBF kernels

$$\mathcal{K}(x, x') = \exp\left(-\|x - x'\|^2/\sigma^2\right),$$

with a real parameter $\sigma \in R$, and sigmoid kernels

$$\mathcal{K}(x, x') = \tanh\left(\kappa\, x^T x' - \Theta\right),$$

with real parameters $\kappa, \Theta \in R$, are commonly used kernel functions. Mercer's Theorem [27] implies that these functions give rise to nonlinear maps $\phi(\cdot)$ for which $\mathcal{K}(x, x') = \langle \phi(x), \phi(x') \rangle$ holds. These kernel-based nonlinear discriminate functions have been successfully applied to a number of real-world problems [8, 14, 23].

In the present paper, we propose linear programming approaches for generating kernel-based nonlinear discriminate functions. There are several papers which develop variants of SVMs by solving linear programming problems, including Multisurface Method [20] and Robust Linear Programming (RLP) [2]. One benefit of these approaches is that a linear program is easier to solve compared with a quadratic program. In addition, efficient and robust optimization packages are available. Therefore, the linear programming approach has potential of handling real-world massive datasets. To our knowledge, however, little attention has been given to linear programming approaches, particularly with respect to kernel-based nonlinear discriminate functions. Mangasarian and Musicant [19], Mangasarian [18], and Weston and Watkins [28] have considered nonlinear cases by solving very large linear programs. The number of variables and the number of constraints of these linear programs are roughly proportional to those of training data points, and this can cause computational difficulties in handling huge datasets.

In contrast to these methods, we utilize smaller sized linear programs to generate nonlinear discriminate functions. Although the original data points are mapped into the higher dimensional feature space $\mathcal{F}$, a fairly small dimensional subspace can be important in discrimination. We will demonstrate that an orthonormal basis of the subspace of $\mathcal{F}$ can be implicitly treated by eigenvectors of the Gram matrix defined by the associated kernel function, and that discriminate functions in the subspace are successfully constructed by solving linear programming problems. For the case in which the number of given data points is very large, we develop a technique to compute a subspace of $\mathcal{F}$ by random sampling of data points. Our numerical experiments indicate that the subspace generated by less than 2% of the training data points achieves reasonable performance for a huge dataset with 60000 training data points.

The present paper is organized as follows. In Section 2, we briefly review a few variants of SVMs by linear programming including RLP, 1-norm and ∞-norm formulations. Section 3 is devoted to the description of a method for extracting implicitly the orthonormal basis of the subspace of the nonlinear feature space $\mathcal{F}$ defined by kernels. We also develop linear programming formulations to generate discriminate functions in this subspace. In Section 4, we present our computational experiments, and concluding remarks are presented in Section 5.

2. Preliminaries

2.1. Linear discrimination

Suppose that we have M data points expressed as vectors of N dimensional real space R^N. We assume that these points are represented by an $M \times N$ matrix A, where the j-th row vector A_j of the matrix A corresponds to the j-th data point. Each point A_j is associated with either the class $+1$ or class -1, which is denoted by $y_j \in \{+1, -1\}$. Let $y = (y_1, y_2, \ldots, y_M)^T$ be an M dimensional vector.

SVMs are used to find a linear discriminate function $f(x) = w^T x - b$ with a normal vector $w \in R^N$ and a real number $b \in R$ such that

$$\begin{array}{lll} A_j w - b \geq 1 & \text{if} & y_j = 1, \\ A_j w - b \leq -1 & \text{if} & y_j = -1. \end{array} \tag{2.1}$$

This condition is equivalently written as follows:

$$Y\,(Aw - be) \geq e, \tag{2.2}$$

where $Y \in R^{M \times M}$ is a diagonal matrix, the diagonal element of which is the vector y, and e is a vector of all ones, the dimension of which is given by context.

The set of points satisfying conditions (2.2) are called *linearly separable*. In general, however, the normal vector w and the real number b satisfying (2.2) may not always exist. Then, introducing a nonnegative error vector $\xi = (\xi_1, \xi_2, \ldots, \xi_M)^T \in R^M$, one can relax conditions (2.2) to

$$Y\,(Aw - be) + \xi \geq e, \quad \xi \geq 0. \tag{2.3}$$

Bennet and Mangasarian [2] considered Robust Linear Programming (RLP) for obtaining the function $f(x) = w^T x - b$ by minimizing the total amount of errors as follows:

$$\text{RLP} \left| \begin{array}{ll} \text{Minimize} & p^T \xi \\ \text{Subject to} & Y\,(Aw - be) + \xi \geq e, \quad \xi \geq 0, \end{array} \right. \tag{2.4}$$

where p is a positive weight vector representing misclassification cost.

Based on the idea of structural risk minimization in statistical learning theory, Vapnik [26, 27] showed that generalization ability can be improved by controlling the complexity of the discriminate function, which is characterized by the norm of w. Standard SVMs solve the following quadratic program:

$$\left| \begin{array}{ll} \text{Minimize} & \dfrac{1}{2}\|w\|_2^2 + C_0\, e^T \xi \\ \text{Subject to} & Y\,(Aw - be) + \xi \geq e, \quad \xi \geq 0. \end{array} \right. \tag{2.5}$$

The associated Wolfe dual [17] formulation [9, 27] is

$$\left| \begin{array}{ll} \text{Maximize} & -\dfrac{1}{2}\alpha^T Y Q Y \alpha + e^T \alpha \\ \text{Subject to} & y^T \alpha = 0, \\ & 0 \leq \alpha \leq C_0 e, \end{array} \right. \tag{2.6}$$

where Q is an $M \times M$ symmetric matrix defined as $Q = AA^T$, and C_0 is a positive parameter controlling the weight between the error term and the norm $\|w\|$ which is called the *capacity term*.

Several variants of the problem (2.5) have been studied [3, 4, 18, 19, 28] by choosing norms other than the 2-norm in (2.5). In the present paper, we use the formulation based on the 1-norm capacity term i.e., the problem for minimizing $C\|w\|_1+e^T\xi$ over (2.3). This problem is formulated as the following linear program:

$$\left|\begin{array}{ll} \text{Minimize} & C\,e^Ts+e^T\xi \\ \text{Subject to} & -s\le w\le s, \\ & Y(Aw-be)+\xi\ge e, \quad \xi\ge 0, \end{array}\right. \tag{2.7}$$

where $w\in R^N, b\in R, s\in R^N$ and $\xi\in R^M$ are variables, and C is a positive parameter. Thus, the dual of this problem is

$$\left|\begin{array}{ll} \text{Maximize} & e^T\alpha \\ \text{Subject to} & -Ce\le A^TY\alpha\le Ce, \\ & y^T\alpha=0, \\ & 0\le\alpha\le e, \end{array}\right.$$

where $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_M)^T$ is a vector of dual variables.

Let us introduce a new vector of variables $\beta=(\beta_1,\beta_2,\ldots,\beta_M)^T$ and let $\beta_j=y_j\alpha_j,\ j=1,2,\ldots,M$. The dual problem can be equivalently written as

$$\left|\begin{array}{ll} \text{Maximize} & y^T\beta \\ \text{Subject to} & -Ce\le A^T\beta\le Ce, \\ & e^T\beta=0, \\ & 0\le Y\beta\le e, \end{array}\right.$$

which leads to the following formulation:

$$\left|\begin{array}{ll} \text{Maximize} & y^T\beta \\ \text{Subject to} & \|A^T\beta\|_\infty\le C, \\ & e^T\beta=0, \\ & 0\le Y\beta\le e. \end{array}\right. \tag{2.8}$$

Generally, one can also consider the problem with the p-norm capacity term. By using the conic duality [1], primal and dual formulations are given as follows:

$$\left|\begin{array}{ll} \text{Minimize} & C\|w\|_p+e^T\xi \\ \text{Subject to} & Y(Aw-be)+\xi\ge e, \quad \xi\ge 0, \end{array}\right. \tag{2.9}$$

and

$$\left|\begin{array}{ll} \text{Maximize} & y^T\beta \\ \text{Subject to} & \|A^T\beta\|_q\le C, \\ & e^T\beta=0, \\ & 0\le Y\beta\le e, \end{array}\right. \tag{2.10}$$

where $\|\cdot\|_q$ is the conjugate norm of $\|\cdot\|_p$ satisfying $q^{-1}+p^{-1}=1$. Here, it is worth noting that problem (2.9) with $p=\infty$, as well as the associated dual problem (2.10) with $q=1$, can also be handled as a linear programming problem. It has been demonstrated [3, 4] that a variant of the problem with the 1-norm capacity term works well, and that, in some cases, it generates better discriminators than that with the ∞-norm.

2.2. Nonlinear discrimination by kernels

Let us now briefly review the idea of generating nonlinear discriminators based on the quadratic programming formulation (2.6), which is employed in standard SVMs. We note that the matrix Q in (2.6) is a Gram matrix [12] of the set of data points $A_1, A_2, \ldots, A_M$ with respect to the usual inner product on R^N. Keeping this fact in mind, let us introduce the nonlinear function $\phi(x) : R^N \to \mathcal{F}$ which maps an N dimensional data point x into $\mathcal{F}$, and let $\mathcal{K}(x, x')$ denote the kernel function which gives the inner product of $\phi(x)$ and $\phi(x')$ in $\mathcal{F}$. Let $\mathcal{K}$ be a Gram matrix of the vectors $\phi(A_1)$, $\phi(A_2)$, $\ldots$, $\phi(A_M)$ with respect to the inner product on $\mathcal{F}$. Then, replacing Q with $\mathcal{K}$, we can formulate the problem for discriminating the points in $\mathcal{F}$ as the following quadratic programming problem:

$$\left|\begin{array}{ll} \text{Maximize} & -\dfrac{1}{2}\alpha^T Y \mathcal{K} Y \alpha + e^T \alpha \\ \text{Subject to} & y^T \alpha = 0, \\ & 0 \le \alpha \le C_0 e. \end{array}\right. \tag{2.11}$$

This problem enables us to construct the discriminate function without explicitly knowing the function $\phi(\cdot)$. We refer the reader to [9, 10, 26] and the references therein for a more detailed explanation of the associated primal form of (2.11) and kernel-based nonlinear discriminate functions.

In order to be consistent with the dual formulations described in the previous subsection, let us also introduce a vector of variables $\beta = (\beta_1, \beta_2, \ldots, \beta_M)^T$ and let $\beta_j = y_j \alpha_j / C_0$, $j = 1, 2, \ldots, M$. We then write problem (2.11) equivalently as follows:

$$\left|\begin{array}{ll} \text{Maximize} & -\dfrac{C_0^2}{2}\beta^T \mathcal{K} \beta + C_0\, y^T \beta \\ \text{Subject to} & e^T \beta = 0, \\ & 0 \le Y\beta \le e. \end{array}\right. \tag{2.12}$$

In the following, we introduce a formulation for the kernel-based nonlinear discriminate functions based on the dual form (2.10) with $q = 2$. That is, let us consider the problem with the square of the 2-norm capacity constraint defined below:

$$\left|\begin{array}{ll} \text{Maximize} & y^T \beta \\ \text{Subject to} & \|A^T \beta\|_2^2 \le C^2, \\ & e^T \beta = 0, \\ & 0 \le Y\beta \le e. \end{array}\right. \tag{2.13}$$

Note that the quadratic constraint can be written as

$$\|A^T \beta\|_2^2 = \beta^T Q \beta \le C^2. \tag{2.14}$$

Replacing Q in (2.14) with the Gram matrix $\mathcal{K}$ defined by the kernel function $\mathcal{K}(\cdot, \cdot)$, we obtain the following problem:

$$\left|\begin{array}{ll} \text{Maximize} & y^T \beta \\ \text{Subject to} & \beta^T \mathcal{K} \beta \le C^2, \\ & e^T \beta = 0, \\ & 0 \le Y\beta \le e. \end{array}\right. \tag{2.15}$$

The next lemma ensures that problems (2.15) and (2.12) are equivalent in the following sense:

Lemma 2.1 *For a suitable parameter C, problem (2.15) generates any optimal solutions of problem (2.12) with the parameter C_0.*

Proof

For any positive parameter C_0, let β^0 be an optimal solution of (2.12). It is easy to verify that β^0 is also an optimal solution of (2.15) with $C^2 = {\beta^0}^T \mathcal{K} \beta^0$. This completes the proof. □

3. Linear Programming Formulations for Kernel SVM

3.1. Eigenvalue decomposition of the Gram matrix

We introduce linear programming formulations for obtaining a discriminate function in the feature space, $\mathcal{F}$, characterized by the kernel function $\mathcal{K}(\cdot, \cdot)$.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{M'} > 0$ be the positive eigenvalues of the matrix $\mathcal{K}$ and $d'_1, d'_2, \ldots, d'_{M'} \in R^M$ be the associated eigenvectors, where the norm of each d_i is 1. Also, let us define

$$D = [d_1, d_2, \ldots, d_{M'}], \quad \text{where} \quad d_i = \sqrt{\lambda_i} d'_i, \quad i = 1, 2, \ldots, M'.$$

Then, the matrix $\mathcal{K}$ is decomposed as $\mathcal{K} = DD^T$, and the quadratic constraint of the problem (2.15) can be expressed as

$$\beta^T \mathcal{K} \beta = \left\| D^T \beta \right\|_2^2 \leq C^2. \tag{3.1}$$

In many practical situations, the number of data points M is very large. This implies that the number of positive eigenvalues, M', could also be large. Thus, we will introduce an approximation to this quadratic constraint. To this end, we consider the largest $S \ll M$ positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_S$, and the associated column vectors of D. Then, let us define a submatrix of D as

$$D_S = [d_1, d_2, \ldots, d_S],$$

which results in the following approximation of the quadratic part of problem (2.15):

$$\beta^T \mathcal{K} \beta \approx \beta^T D_S D_S^T \beta = \left\| D_S^T \beta \right\|_2^2.$$

The symmetric matrix $D_S D_S^T$ is the closest matrix to $\mathcal{K}$ with rank S in the sense of the Frobenius norm.

Now, returning to the linear programming scheme, and replacing the Euclidean norm constraint (3.1) with an ∞-norm constraint, we obtain the following problem which can be reduced to a linear program:

$$\left| \begin{array}{ll} \text{Maximize} & y^T \beta \\ \text{Subject to} & \left\| D_S^T \beta \right\|_\infty \leq C, \\ & e^T \beta = 0, \\ & 0 \leq Y \beta \leq e. \end{array} \right. \tag{3.2}$$

As we have seen in Section 2.1, the primal form of this problem (3.2) is:

$$\left| \begin{array}{ll} \text{Minimize} & C \left\| w_S \right\|_1 + e^T \xi \\ \text{Subject to} & Y (D_S w_S - b_S e) + \xi \geq e, \\ & \xi \geq 0, \end{array} \right. \tag{3.3}$$

where $w_S \in R^S, b_S \in R^1$ and $\xi \in R^M$ are primal variables. Let us denote an optimal solution of (3.3) as $(w_S, b_S) = (w_S^*, b_S^*)$. We then obtained an optimal linear discriminate function as

$$f(x_S) = w_S^{*T} x_S + b_S^*,$$

where x_S is an S dimensional vector of variables.

3.2. Extracting the subspace of $\mathcal{F}$

Problem (3.3) can be considered as an SVM with a 1-norm capacity term, in which each data point is given as a row vector of the matrix D_S, instead of the original data points given in rows of A. The dimension of the variable w_S is, in general, different from that of the original data points, N. In this subsection, we will introduce several propositions which link the feature space $\mathcal{F}$ and the matrix D_S. Similar discussions can also be found in Schölkopf *et al.* [24].

Let d_{jk} denote the jk elements of the matrix D_S. Associated with the k-th column vector $d_k = (\, d_{1k}\, d_{2k}\, \cdots\, d_{Mk}\,)^T$ of D_S, let us define vectors $\mathcal{V}_k,\ k = 1, 2, \ldots, S$ in $\mathcal{F}$ as follows:

$$\mathcal{V}_k = \frac{\sum_{j=1}^M d_{jk}\phi(A_j)}{\lambda_k}, \quad k = 1, 2, \ldots, S. \tag{3.4}$$

Note that since $\phi(A_j)$ is not described explicitly, each vector $\mathcal{V}_k$ can not be expressed.

The following lemma holds:

Lemma 3.2 *The set of vectors* $\{\, \mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_S \,\}$ *satisfies*

$$\langle\, \mathcal{V}_k\,,\, \mathcal{V}_{k'} \,\rangle = \begin{cases} 0 & \text{if } k \neq k', \\ 1 & o.w, \end{cases}$$

where $\langle\, \cdot\,,\, \cdot \,\rangle$ *denotes the inner product defined in* $\mathcal{F}$.

Proof

For any $k, k' = 1, 2, \ldots, S$, verifying that

$$\begin{aligned} \langle\, \mathcal{V}_k\,,\, \mathcal{V}_{k'} \,\rangle &= \frac{1}{\lambda_k \lambda_{k'}} \sum_{j=1}^M \sum_{j'=1}^M d_{jk} d_{j'k'} \langle\, \phi(A_j)\,,\, \phi(A_{j'}) \,\rangle \\ &= \frac{1}{\lambda_k \lambda_{k'}} d_k^T \mathcal{K} d_{k'} \end{aligned}$$

is straightforward. Obviously, $d_{k'}^T \mathcal{K} d_k = 0$ if $k \neq k'$, and $d_k^T \mathcal{K} d_k = \lambda_k \|d_k\|^2 = \lambda_k^2$. This completes the proof. □

It follows from Lemma 3.2 that the set of vectors

$$\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_S\}$$

forms an orthonormal basis of the S dimensional subspace of $\mathcal{F}$ which will be denoted by $\mathcal{F}_S$. Moreover, for each $i = 1, 2, \ldots, M$, the projection of the vector $\phi(A_i)$ onto the subspace $\mathcal{F}_S$ can be uniquely given by

$$\langle\, \phi(A_i)\,,\, \mathcal{V}_1 \,\rangle \mathcal{V}_1 + \langle\, \phi(A_i)\,,\, \mathcal{V}_2 \,\rangle \mathcal{V}_2 + \cdots + \langle\, \phi(A_i)\,,\, \mathcal{V}_S \,\rangle \mathcal{V}_S. \tag{3.5}$$

In the following, for any point $x \in R^N$, let us denote the S dimensional coordinate vector of the projection of $\phi(x)$ onto $\mathcal{F}_S$ with respect to the basis $\mathcal{V} = \{\mathcal{V}_1,\, \mathcal{V}_2,\, \ldots,\, \mathcal{V}_S\}$ as $[x]_{\mathcal{V}}$, i.e.,

$$[x]_{\mathcal{V}} = \begin{pmatrix} \langle\, \phi(x)\,,\, \mathcal{V}_1 \,\rangle \\ \langle\, \phi(x)\,,\, \mathcal{V}_2 \,\rangle \\ \vdots \\ \langle\, \phi(x)\,,\, \mathcal{V}_S \,\rangle \end{pmatrix} \in R^S.$$

Lemma 3.3 *Let $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_S\}$ be an orthonormal basis defined in (3.4), then*

$$D_{S_j}^T = [A_j]_{\mathcal{V}}, \quad j = 1, 2, \ldots, M,$$

where D_{S_j} denotes the j-th row vector of D_S.

Proof

Due to (3.4), one can simplify the coefficient with respect to $\mathcal{V}_k$ as follows:

$$\langle\, \phi(A_i)\,,\, \mathcal{V}_k \,\rangle = \left\langle\, \phi(A_i)\,,\, \frac{\sum_{j=1}^M d_{jk}\phi(A_j)}{\lambda_k} \,\right\rangle = \frac{\sum_{j=1}^M d_{jk}\mathcal{K}\,(\,A_i\,,\,A_j\,)}{\lambda_k} = d_{ik}.$$

□

Recall that the optimal solution (w_S^*, b_S^*) of the problem (3.3) generates the discriminate function:

$$f(x_S) = w_S^{*T} x + b_S^*.$$

Note that this linear function is defined in the S dimensional subspace $\mathcal{F}_S$. Let us now consider classifying an arbitrary N dimensional data point by this function. To this end, we need to calculate the coordinate vector $[x]_{\mathcal{V}}$. As we have seen above, the k-th element of the vector $[x]_{\mathcal{V}}$ is given by the projection onto $\mathcal{V}_k$, that is $\langle\, \phi(x)\,,\, \mathcal{V}_k \,\rangle$. Substituting (3.4), we have

$$\langle\, \phi(x)\,,\, \mathcal{V}_k \,\rangle = \frac{\sum_{j=1}^M d_{jk}\mathcal{K}\,(\,x\,,\,A_j\,)}{\lambda_k}, \quad k = 1, 2, \ldots, S.$$

Note that for an arbitrary vector $x \in R^N$, each element of the vector $[x]_{\mathcal{V}}$ is explicitly calculated without knowing the vectors $\mathcal{V}_k,\ k = 1, 2, \ldots, S$. Then, one can classify the point $x \in R^N$ according to the sign of

$$[x]_{\mathcal{V}}^T\, w_S^* + b_S^*,$$

which can also be calculated explicitly.

3.3. Sampling procedure

When the number of points M is enormous, a considerable amount of computational work would be required for obtaining the largest S eigenvectors of the $M \times M$ matrix $\mathcal{K}$. In order to overcome this difficulty, we use the following sampling procedure to extract an orthonormal basis of $\mathcal{F}$ and to generate projections of the original data points.

For simplicity, let us assume that one can choose L sample points, where $L \ll M$, corresponding to the first L rows of matrix A. We also assume that, associated with the sample points, the matrix A and the Gram matrix $\mathcal{K}$ are partitioned as follows:

$$A = \begin{bmatrix} A^L \\ A' \end{bmatrix}, \quad \text{and} \quad \mathcal{K} = \begin{bmatrix} \mathcal{K}^L & \mathcal{K}'^T \\ \mathcal{K}' & \mathcal{K}'' \end{bmatrix},$$

where $A^L \in R^{L\times N}$, $A' \in R^{(M-L)\times N}$, $\mathcal{K}^L \in R^{L\times L}$, $\mathcal{K}' \in R^{(M-L)\times L}$ and $\mathcal{K}'' \in R^{(M-L)\times(M-L)}$. Then, we can perform the eigenvalue decomposition of the $L \times L$ submatrix $\mathcal{K}^L$, rather than that of $\mathcal{K}$. Now, let $\lambda_1^L \geq \lambda_2^L \geq \cdots \geq \lambda_S^L > 0$ be the largest $S(\leq L)$ positive eigenvalues of $\mathcal{K}^L$, and let $D_S^L D_S^{L^T}$ be the rank S approximation of $\mathcal{K}^L$. Let us denote the jk element of the matrix D_S^L as d_{jk}^L. Then, the basis $\mathcal{V}^L = \{\mathcal{V}_1^L, \mathcal{V}_2^L, \ldots, \mathcal{V}_S^L\}$ can be calculated as follows:

$$\mathcal{V}_k^L = \frac{\sum_{j=1}^L d_{jk}^L \phi(A_j)}{\lambda_k^L} \quad k = 1, 2, \ldots, S. \tag{3.6}$$

The same argument as the proof of Lemma 3.2 is applied to show that these vectors constitute an orthonormal basis. Thus, we can also extract the subspace of $\mathcal{F}$.

As we have already shown, each row of the matrix D_S^L corresponds to the projection of the associated sample points, i.e., the associated row vector of A^L. Moreover, it is straightforward to verify that the projection of the unsampled point, that is the j-th row vector of A', is written as

$$\left[A'_j\right]_{\mathcal{V}^L} = \mathcal{K}'_j D_S^L \Lambda^{L^{-1}},$$

where $\mathcal{K}'_j$ corresponds to the j-th row vector of $\mathcal{K}'$ and Λ^L is an $S \times S$ diagonal matrix with the elements $\lambda_i^L \quad i = 1, 2, \ldots, S$.

Therefore, the inequalities corresponding to the norm constraints in (3.2) become

$$\left\| \begin{bmatrix} D_S^L \\ \mathcal{K}' D_S^L \Lambda^{L-1} \end{bmatrix}^T \beta \right\|_\infty \leq Ce. \tag{3.7}$$

In our sampling procedure, L sample points are used only for extracting the basis of the S dimensional subspace $\mathcal{F}_S$, whereas the entire training dataset can be used in the linear programming problem.

4. Computational Experiments

In this section, we demonstrate the performance of the proposed methods. We use a dataset of handwritten digits called MNIST[16], which has been served as a test bed for classification problems. The dataset contains 60000 images of handwritten digits for training and 10000 images for testing purposes. Each handwritten digit is given as a 28 $\times$ 28 pixel image, i.e., as a point in R^{784}.

For each class $k = 0, 1, \ldots, 9$, we generate a function $f_k(\cdot)$ discriminating the class k from the other nine classes. Then, the class of a point x is determined as $\arg\max\{\ f_k(x) \mid k = 0, 1, \ldots, 9\ \}$. This method is called the one-against-all method and is commonly used in Support Vector literatures.

In our experiments, we use the polynomial kernels

$$\mathcal{K}(x, x') = \left((x^T x')/784\right)^d, \tag{4.1}$$

and the RBF kernels

$$\mathcal{K}(x, x') = \exp\left(-\|x - x'\|^2/(784 \times \gamma)\right), \tag{4.2}$$

where d is a given positive integer and γ is a given positive real number. We randomly sample L points from the training set and generate discriminate functions in the S dimensional subspace extracted by the eigenvector decomposition described in the previous sections. We run the procedure over two randomly generated samples, and show the average performances.

We solve the linear programming formulation of type (3.2) which is expressed as follows:

$$\left|\begin{array}{ll} \text{Maximize} & y^T \beta \\ \text{Subject to} & \begin{bmatrix} D_S^L \\ \mathcal{K}' D_S^L \Lambda^{L-1} \end{bmatrix}^T \beta - z = 0, \\ & e^T \beta = 0, \\ & 0 \leq Y\beta \leq e, \\ & -Ce \leq z \leq Ce, \end{array}\right. \tag{4.3}$$

Table 1: Average error rate (%) obtained by polynomial kernels (4.2) with $d = 7, L = 3000$

	C									
S	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
100	5.46	5.44	5.42	5.40	5.43	5.42	5.43	5.42	5.40	5.42
200	3.74	3.73	3.74	3.73	3.72	3.73	3.76	3.76	3.79	3.81
400	2.78	2.72	2.73	2.72	2.76	2.72	2.73	2.73	2.75	2.73
600	2.57	2.44	2.49	2.50	2.46	2.49	2.45	2.48	2.45	2.42
800	2.43	2.33	2.18	2.20	2.19	2.22	2.21	2.19	2.27	2.28

where β and z are variable vectors of size $M = 60000$ and S, respectively. Thus, the problems have $60000 + S$ variables with upper and lower bounds, and $S + 1$ equality constraints. The experiments are conducted on an AlphaServer GS320 workstation (CPU: Alpha21264 1GHz, 4G memory) using CPLEX 7.1 [13] as a linear programming solver.

In Table 1, 2, and 3, we show the test set error rates (%). These tables show the averages over two sampling runs. Table 1 is obtained using the polynomial kernel (4.1) with $d = 7$ and $L = 3000$ sample points. The discriminate functions are generated by solving the linear programming problem (4.3) with parameter C ranging from 0.01 to 0.1. From Table 1, we observe that the performance is rather insensitive to the choice of parameter C within this range. Reasonable performance appears to be obtained with $C = 0.08$. Therefore, $C = 0.08$ was used for the rest of our experiments. Table 2 and 3 show the results obtained by the polynomial kernels (4.1) with $d = 7$ and 8, and those obtained by the RBF kernels (4.2) with $\gamma = 0.2$ and 0.4. We sample ($L =$)1000 or 3000 points out of 60000 training points to extract the subspace $\mathcal{F}_S$. The number of dimension S ranges from 100 to 800. In Table 4, we show the total CPU times in seconds for obtaining the discriminate functions for the ten classes. The values in the parentheses indicate the time spent for the eigenvalue decomposition. We solve the same problem using SVMTorch [7], which is one of the commonly used implementations [6] of the standard SVM, and the CPU time is reported in the last row of the table. In Table 5, we show the results of ten binary classifications for each class (digit) obtained using the polynomial kernels with $d = 7, L = 3000$ and $C = 0.08$. The average numbers of test errors out of 10000 test points, as well as the total error rates (in the last columns), are listed in this table. The last row of the table corresponds to the results of the standard SVM [5] with the polynomial kernel.

We see from these tables that both the RBF and polynomial kernels attain around 3% error rates when 400 eigenvectors are used, and that the error rates gradually decrease as the number of eigenvectors increases. Moreover, fairly good performance is achieved by the subspace extracted by a small fraction of the sampled points, which is less than 2% of the entire training set. Table 5 shows that the accuracy of the results obtained by our procedure is slightly less than that for the results in [5] obtained by the standard SVM with the polynomial kernels. However, our discriminate functions are obtained by linear programming, whereas the standard SVM requires quadratic programming.

5. Conclusion

In the present paper, we have proposed a linear programming approach for generating kernel-based nonlinear discriminate functions. The eigenvectors of the Gram matrix are employed to generate the orthonormal basis of the subspace of the feature space $\mathcal{F}$, and the coordinates of the projection of the nonlinearly mapped points $\phi(x) \in \mathcal{F}$ with respect

Table 2: Average error rate (%) obtained by polynomial kernels (4.1), when $C = 0.08$

	$d = 7$		$d = 8$	
S	$L = 1000$	$L = 3000$	$L = 1000$	$L = 3000$
100	5.67	5.42	5.64	5.42
200	3.99	3.76	4.06	3.75
400	3.17	2.73	3.13	2.74
600	2.56	2.48	2.67	2.39
800	2.49	2.19	2.53	2.24

Table 3: Average error rate (%) obtained by RBF kernels (4.2), when $C = 0.08$

	$\gamma = 0.2$		$\gamma = 0.4$	
S	$L = 1000$	$L = 3000$	$L = 1000$	$L = 3000$
100	5.69	5.46	5.89	5.78
200	4.03	3.84	4.09	3.95
400	3.19	2.78	3.02	2.92
600	2.68	2.43	2.63	2.48
800	2.50	2.22	2.41	2.25

Table 4: CPU time (in seconds)

	RBF kernel with $\gamma = 0.2$		polynomial kernel with $d = 7$	
S	$L = 1000$	$L = 3000$	$L = 1000$	$L = 3000$
100	857.1 (145.0)	1176.2 (485.5)	798.0 (112.1)	1074.5 (402.9)
200	2064.3 (167.0)	2506.5 (593.7)	1961.8 (133.8)	2263.7 (473.9)
400	7376.4 (311.05)	8367.1 (1017.5)	6634.4 (224.0)	7367.1 (791.4)
600	17312.2 (293.0)	18804.7 (1737.9)	15069.0 (251.8)	15746.9 (1268.2)
800	31927.2 (314.5)	33108.0 (2888.0)	27121.6 (274.9)	27763.7 (2046.6)
SVMTorch [7]	75294.9		47988.8	

Table 5: Number of errors out of 10000 test points obtained by polynomial kernels with $d = 7$ when $L = 3000, C = 0.08$

	Class (Digit)										
S	0	1	2	3	4	5	6	7	8	9	Err.(%)
100	46.5	34.5	146	136	129.5	149.5	88	116	243.5	230	5.4
200	36	25	96.5	105	96	105.5	64.5	80.5	137	180.5	3.8
400	28.5	20.5	69	78.5	65.5	69	43	72	91	135.5	2.7
600	29	21	66.5	65	57	48.5	41	70	81.5	111	2.5
800	27	20.5	64.5	62	53.5	46	38.5	61.5	72	100	2.1
SVM in [5]	17	15	34	32	30	29	30	43	47	56	1.4

to this basis are explicitly obtained. Through this procedure, we can generate a nonlinear discriminate function by solving a linear programming problem.

We have numerically demonstrated that relatively small dimensional subspace carries enough ability to discriminate between classes. In fact, for the MNIST dataset with 60000 training points, nonlinear discriminate functions of moderate accuracy can be generated in the subspace with 400 dimension. Also, in this setting, our linear programming formulation generates discriminators as efficiently as standard SVMs with the quadratic programming formulation. The proposed procedure takes advantage of the fact that the linear programming problem with a large number of variables can be solved when the number of constraints are a few hundred. Moreover, in many practical situations, parameters such as C should be adjusted by performing cross–validation procedures [25], where we need to solve optimization problems repeatedly by changing C. In such a situation, a series of several linear programming problems is efficiently optimized by applying the parametric linear program. In addition, since the proposed formulation, as well as the original QPs, has an optimal solution with many zeros, the linear program can be optimized within a smaller number of pivoting iterations. These are the advantages of the proposed method.

Although our computational results are performed using a general purpose linear programming package as an optimization engine, efficient implementation of a column generation and a chunking method would allow problems to be handled efficiently on PCs. This is a subject for our future research. In addition, the problem for generating multi-class discriminators can be handled by extending the proposed linear programming approach, which is now under development.

References

[1] F. Alizadeh and S. Schmieta: Symmetric cones, potential reduction methods and word-by-word extensions. *Handbook of Semidefinite Programming* (Kluwer Academic Publishers, Boston, MA, 2000), 195–233.

[2] K.P. Bennett and O.L. Mangasarian: Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software* **1** (1992) 23–34.

[3] P.S. Bradley and O.L. Mangasarian: Feature selection via concave minimization and support vector machines. *13 th International Conference on Machine Learning* (1998), 82–90.

[4] P.S. Bradley and O.L. Mangasarian: Massive data discrimination via linear support vector machines. *Optimization Methods and Software* **13** (2000) 1–10.

[5] C.J.C. Burges and B. Schölkopf: Improving the accuracy and speed of support vector learning machines. M. Mozer, M. Jordan and T. Petsche eds.: *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 1997), 375–381.

[6] C. Campbell: Kernel methods: a survey of current techniques. *Neurocomputing* **48** (2002) 63–84.

[7] R. Collobert and S. Bengio: SVMTorch: Support Vector Machines for Large-Scale Regression Problems. *Journal of Machine Learning Research* **1** (2001) 143–160.

[8] C. Cortes and V. Vapnik: Support-Vector Networks. *Machine learning* **20** (1995) 273–297.

[9] N. Cristianini and J. Shawe-Taylor eds.: *An Introduction to Support Vector Machines and Other Kernel–Based Learning Methods* (Cambridge University Press, U.K., 2000).

[10] F. Cucker and S. Smale: On the mathematical foundations of learning. *Bull. Amer. Math. Soc. (N.S.)* **39** (2002) 1–49 (electronic).

[11] S.T. Dumais, J. Platt, D. Heckerman and M. Sahami: Inductive learning algorithms and representations for text categorization. G. Gardarin ed.: *7th International Conference on Information and Knowledge Discovery* (1998), 148–155.

[12] R.A. Horn and C.R. Johnson: *Matrix Analysis* (Cambridge University Press, Cambridge-New York, 1985).

[13] ILOG: *ILOG CPLEX7.1, user's manual* (ILOG, France, 2001).

[14] T. Joachims: Text categorization with support vector machines: learning with many relevant features. *Lecture notes in computer science* **1398** (1998) 137–142.

[15] T. Joachims: Making large-scale support vector machine learning practical. B. Schölkopf, C. Burges and A. Smola eds.:*Advances in Kernel Methods* (The MIT Press, 1999), 169–184.

[16] Y. LeCun, L.D. Jackel, L. Bottou, A. Brunot, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Müller, E. Säckinger, P. Simard and V. Vapnik: Comparison of learning algorithms for handwritten digit recognition. F. Fogelman-Soulié and P. Gallinari eds.: *Proceedings ICANN'95 – International Conference on Artificial Neural Networks* (Nanterre, France, 1995), 53–60.

[17] O.L. Mangasarian: *Nonlinear Programming* (SIAM, Philadelphia, PA, 1994).

[18] O.L. Mangasarian: Generalised support vector machines. A. Smola, P. Bartlett, B. Schölkopf and D. Schuurmans eds.: *Advances in Large Margin Classifiers* (The MIT Press, 2000), 135–146.

[19] O.L. Mangasarian and D.R. Musicant: Data discrimination via nonlinear generalized support vector machines. M.C. Ferris, O.L. Mangasarian and J.S. Pang eds.: *Complementarity : Applications, Algorithms and Extensions* (Kluwer Academic Publishers, The Netherlands, 2001), 233–251.

[20] O.L. Mangasarian, R. Setiono and W.H. Wolberg: Pattern recognition via linear programming: Theory and application to medical diagnosis. *Proceedings of the Workshop on Large-Scale Numerical Optimization* (1990), 22–31.

[21] J.C. Platt: Fast training of support vector machines using sequential minimal optimization. B. Schölkopf, C. Burges and A. Smola eds.: *Advances in Kernel Methods* (The MIT Press, 1999), 185–208.

[22] M. Pontil and A. Verri: Support vector machines for 3D object recognition. *IEEE transactions on pattern analysis and machine intelligence* **20** (1998) 637–646.

[23] B. Schölkopf, C. Burges and V. Vapnik: Extracting support data for a given task. U.M. Fayyad and R. Uthurusamy eds.: *Proceedings, First International Conference on Knowledge Discovery & Data Mining* (1995), 252–257.

[24] B. Schölkopf, A.J. Smola and K. Müeller: Kernel principal component analysis. B. Schölkopf, C. Burges and A. Smola eds.: *Advances in Kernel Methods* (The MIT Press, 1999), 327–352.

[25] M. Stone: Cross–validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society* **36** (1974) 111–147.

[26] V.N. Vapnik: *Statistical Learning Theory* (John Wiley & Sons, 1998).

[27] V.N. Vapnik: *The Nature of Statistical Learning Theory*, (Springer-Verlag, New York, 2000).

[28] J. Weston and C. Watkins: Multi-class Support Vector Machines. Technical Report CSD-TR-98-04, Royal Holloway, University of London, 1998.

Yasutoshi Yajima
Department of Industrial Engineering and Management,
Tokyo Institute of Technology,
Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan.
E-mail: yasutosi@me.titech.ac.jp