Support Vector Machines

Los Alamos Technical Report: LAUR 00-579

Don Hush and Clint Scovel*

Computer Research Group, CIC-3
Los Alamos National Laboratory,
Los Alamos, NM, 87545
(dhush@lanl.gov and jcs@lanl.gov)

January, 2000

In this report, we discuss Vapnik's support vector machines [13] for separable and non-separable data. We discuss implementation issues, generalization performance, and how they are remarkably different from existing classifier design methodologies.

1. The optimal hyperplane as a quadratic programming problem

Consider a finite set S of vectors

$$(x_1, y_1), (x_2, y_2), ..., (x_k, y_k),$$

from the space $R^n \times \{-1, 1\}$. Consider a unit vector $\phi \in R^n$. We say that the data is separable by the hyperplane

$$x \cdot \phi = c$$

if

$$x_i \cdot \phi > c, \qquad y_i = 1$$

$$x_i \cdot \phi < c, \qquad y_i = -1.$$

We define two functions of the data S and any vector ϕ:

$$c_1(\phi) = \min_{y_i = 1} x_i \cdot \phi$$

$$c_{-1}(\phi) = \max_{y_i = -1} x_i \cdot \phi.$$

Define the margin of the hyperplane

$$x \cdot \phi = c$$

to be

$$\rho(\phi, c) = \min \{c_1(\phi) - c, c - c_{-1}(\phi)\}$$

A hyperplane is said to be optimal if it maximizes the margin over all hyperplanes. Namely, it solves the optimization problem

* partially supported by The Los Alamos DOE Program in Applied Mathematical Sciences

$$\max \rho(\phi, c) \tag{1}$$
$$|\phi| = 1. \tag{2}$$

For a fixed ϕ, $\rho(\phi, c)$ is maximized when $c = \frac{c_1 + c_{-1}}{2}$ and then $\rho = \frac{c_1(\phi) - c_2(\phi)}{2}$. Consequently, we can say that a hyperplane (ϕ_0, c_0) is optimal if

$$c_0 = \frac{c_1(\phi_0) + c_{-1}(\phi_0)}{2}$$

where ϕ_0 maximizes

$$\rho(\phi) = \frac{c_1(\phi) - c_{-1}(\phi)}{2}.$$

We thus redefine the margin

$$\rho(\phi) = \frac{c_1(\phi) - c_{-1}(\phi)}{2}$$

and the optimization problem

$$\max \rho(\phi) \tag{3}$$
$$|\phi| = 1. \tag{4}$$

We begin by proving the first important fact:

Theorem 1 *(Vapnik, 1998) The optimal hyperplane is unique.*

Proof. Existence of the maximum of ρ on $|\phi| \leq 1$ follows from the continuity of ρ. We now show that the maximum must be achieved on the boundary $|\phi| = 1$. Suppose this was not the case and the maximum was achieved at the point ϕ_0 with $|\phi_0| < 1$. Since ρ is positively homogeneous, the point

$$\phi^* = \frac{\phi_0}{|\phi_0|}$$

has the larger margin

$$\rho(\phi^*) = \frac{\rho(\phi_0)}{|\phi_0|},$$

giving a contradiction. Consequently, the maximum is achieved only at the boundary $|\phi| = 1$.

Since a maximization over a set of convex functions is convex, and concave functions are just negatives of convex ones, the fact that ρ is a minimization $-$ a maximization over linear functions(which are both convex and concave) implies that ρ is concave.

We now show that a maximum can only occur at one point on the boundary. Suppose to the contrary, that the maximum occurs at two distinct points on the boundary. Then since ρ is a concave function, the maximum must also be realized on the line segment connecting the two points contradicting the fact that the maximum may only be obtained on the boundary.

$\square$

To efficiently compute optimal hyperplanes we form an equivalent optimization problem:

$$\min \frac{1}{2}|\psi|^2 \tag{5}$$

$$x_i \cdot \psi + b \geq 1, \qquad y_i = 1 \tag{6}$$

$$x_i \cdot \psi + b \leq -1, \qquad y_i = -1. \tag{7}$$

Theorem 2 *(Vapnik, 1998) The vector ψ_0 that solves the above quadratic programming problem is related to the optimal hyperplane vector ϕ_0 by*

$$\phi_0 = \frac{\psi_0}{|\psi_0|}.$$

The margin of the optimal hyperplane (ϕ_0, c_0) is

$$\rho(\phi_0) = \frac{1}{|\psi_0|}.$$

Proof. Let ψ_0 denote a solution to the quadratic programming problem. Consider

$$\phi_0 = \frac{\psi_0}{|\psi_0|}.$$

Since ψ_0 satisfies the constraints for some b, it is clear that $\rho(\psi_0) \geq 1$ so that

$$\rho(\phi_0) = \rho(\frac{\psi_0}{|\psi_0|}) \geq \frac{1}{|\psi_0|}.$$

To prove the theorem, we just need to show that there does not exist a unit vector ϕ^* such that

$$\rho(\phi^*) > \frac{1}{|\psi_0|}.$$

Suppose that such a ϕ^* exists. If we define

$$\psi^* = \frac{\phi^*}{\rho(\phi^*)}$$

then it is clear that

$$\rho(\psi^*) = 1$$

and

$$|\psi^*| = \frac{1}{\rho(\phi^*)} < |\psi_0|.$$

For $y_i = 1$,

$$x_i \cdot \psi^* \geq c_1(\psi^*) = \frac{c_1(\psi^*) - c_{-1}(\psi^*)}{2} + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2}$$

$$= \rho(\psi^*) + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2} = 1 + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2}$$

so that ψ^* satisfies the constraint

$$x_i \cdot \psi + b \geq 1, \qquad y_i = 1$$

with

$$b = -\frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2}.$$

For $y_i = -1$,

$$x_i \cdot \psi^* \leq c_{-1}(\psi^*) = \frac{c_{-1}(\psi^*) - c_1(\psi^*)}{2} + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2}$$

$$= -\rho(\psi^*) + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2} = -1 + \frac{c_1(\psi^*) + c_{-1}(\psi^*)}{2}$$

so that ψ^* satisfies the constraint

$$x_i \cdot \psi + b \leq -1, \qquad y_i = -1$$

with the same value of b.

Consequently, $|\psi^*| < |\psi_0|$ and satisfies the constraints, contradicting the assumption that ψ_0 was optimal for the problem {5,6,7}.

□

Theorem 3 *(Kuhn-Tucker) Consider the convex programming problem:*

$$\min f(x)$$

subject to the constraints

$$g_i(x) \leq 0, \qquad i = 1, .., k$$

where f and g_i, $i = 1, .., k$ are convex.

Define the Lagrangian

$$L(x, \lambda_0, \lambda) = \lambda_0 f(x) + \sum_{i=1}^{k} \lambda_i g_i(x)$$

where $\lambda = (\lambda_1, ..., \lambda_k)$. If x^* solves the convex programming problem, then there exists Lagrange multipliers λ_0^* and $\lambda^* = (\lambda_1^*, ..., \lambda_k^*)$ both not simultaneously zero such that the following three conditions hold

$$\min_x L(x, \lambda_0^*, \lambda^*) = L(x^*, \lambda_0^*, \lambda^*)$$

$$\lambda_0^* \geq 0, \qquad \lambda^* \geq 0$$

$$\lambda_i^* g_i(x^*) = 0, \qquad i = 1, .., k.$$

If $\lambda_0^* \neq 0$, then these three conditions are sufficient for x^* to be a solution of the convex programming problem. In order for $\lambda_0^* \neq 0$ it is sufficient for the Slater conditions to be satisfied. Namely, there should exist an $\acute{x}$ such that

$$g_i(\acute{x}) < 0, \qquad i = 1, ..., k.$$

If the Slater condition is satisfied, one can rewrite the Lagrangian as

$$L(x, \lambda) = f(x) + \sum_{i=1}^{k} \lambda_i g_i(x)$$

and the three conditions are equivalent to the existence of a saddle point of the Lagrangian, where (x^*, λ^*) is said to be a saddle point of L if

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$$

for all x and $\lambda \geq 0$.

Now let us return to the quadratic programming problem $\{5,6,7\}$. Rewrite the constraints to obtain a the more compact form:

$$\min \frac{1}{2}|\psi|^2 \tag{8}$$

$$y_i(x_i \cdot \psi + b) \geq 1. \tag{9}$$

For separable data, the Slater condition is satisfied, so by the Kuhn-Tucker theorem solving the quadratic programming problem is equivalent to finding a saddle point of the Lagrangian

$$L(\psi, b, \lambda) = \frac{1}{2}|\psi|^2 - \sum_{i=1}^{k} \lambda_i (y_i(x_i \cdot \psi + b) - 1),$$

where $\lambda \geq 0$. Since the Lagrangian is convex in (ψ, b) and concave in λ, we can apply von Neumann's theorem [8] to find the saddle by first minimizing L over (ψ, b) followed my maximizing over $\lambda \geq 0$.

Minimization with respect to (ψ, b) determines the equations

$$\frac{\partial L}{\partial \psi} = \psi - \sum_{i=1}^{k} \lambda_i y_i x_i = 0,$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{k} \lambda_i y_i = 0.$$

Consequently, for the vector ψ that defines the optimal hyperplane

$$\psi = \sum_{i=1}^{k} \lambda_i y_i x_i,$$

$$\sum_{i=1}^{k} \lambda_i y_i = 0.$$

Substituting into the Lagrangian we obtain

$$W(\lambda) = \sum_{i=1}^{k} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{k} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j).$$

Upon maximization we obtain a $\lambda^* \geq 0$ such that the optimal vector ψ^* can be written

$$\psi^* = \sum_{i=1}^{k} \lambda_i^* y_i x_i.$$

In addition, the equation

$$y_i(x_i \cdot \psi^* + b^*) - 1 = 0$$

implies the vector x_i is one of the closest to the optimal hyperplane(See Figure 1 where those closest are enlarged circles and crosses). Such vectors are called support vectors.

Since the Kuhn-Tucker conditions

$$\lambda_i^*(y_i(x_i \cdot \psi^* + b^*) - 1) = 0, i = 1, .., k$$

are satisfied, they imply that a non-zero value of λ_i^* corresponds to a support vector x_i.

Since

$$\psi^* = \sum_{i=1}^{k} \lambda_i^* y_i x_i = 0.$$

is a linear combination of support vectors, the function describing the separation hyperplane

$$f(x) = x \cdot \psi^* + b^*$$

has the form

$$f(x) = \sum_{i=1}^{k} \lambda_i^* y_i (x_i \cdot x) + b^*$$

where the only nontrivial part of the sum is over the support vectors.

It will be important later in that both the function

$$f(x) = \sum_{i=1}^{k} \lambda_i^* y_i (x_i \cdot x) + b^*$$

and the objective function

$$W(\lambda) = \sum_{i=1}^{k} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{k} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j).$$

do not depend explicitly on the dimensionality of the vector x and x_i but only depend upon the inner product of such vectors.

2. Complexity properties of optimal hyperplanes

Now we describe some of the consequences of choosing an optimal hyperplane for classification. The representation of a hyperplane

$$x \cdot \psi + b = 0$$

by the tuple (ψ, b) is called canonical(with respect to the data S) if

$$\inf_{x \in S} |x \cdot \psi + b| = 1.$$

We state, without proof, the main theorem concerning optimal hyperplanes. The first proof of this theorem was obtained by Hush and Scovel [6].

Theorem 4 *(Vapnik) Suppose that the data S lies in a ball of radius D. A set of hyperplanes whose canonical representatives satisfy*

$$|\psi| \leq A$$

has its VC dimension bounded by

$$\min([D^2A^2], n) + 1$$

where [] denotes the integral part.

Since the pairs (ψ, b) are canonical, they satisfy the constraints

$$y_i(x_i \cdot \psi + b) \geq 1,$$

for the partition determined by

$$y_i = sign(x_i \cdot \psi + b).$$

Consequently, for this labelling, the margin of the hyperplane is

$$\rho(\frac{\psi_0}{|\psi_0|}) \geq \frac{1}{|\psi_0|} \geq \frac{1}{A}.$$

Consequently, this theorem can be formulated with respect the the margin but requires some terminology.

Definition 1 *Let $X = \Re^n$ be the n-dimensional Euclidean space, and let H be the family of linear classifiers $c(x) = sign(h(x))$ where $h(x)$ is an affine function. Further, let H_ρ be the set of linear classifiers that dichotomize X using hyperplanes of thickness ρ. More formally, define H_ρ to be classifiers of the form*

$$c_\rho(x) = c(x), \quad D(x|h = 0) > \rho$$

where $D(x|h = 0)$ is the distance from x to the hyperplane $h = 0$. (Note that $c_\rho(x)$ is not defined *for $\{x : D(x|h = 0) \leq \rho\}$.) The* margin *of classifiers in H_ρ is defined to be ρ. Finally, let $H_{\rho+}$ be the set of linear classifiers with thickness greater than or equal to ρ, that is $H_{\rho+} = \cup_{\phi \geq \rho} H_\phi$.*

Theorem 5 *Let $S = \{x_1, x_2, ..., x_k\} \subset R^n$ denote a set of points contained within a sphere of radius D. The VC dimension of H_{ρ^+} restricted to S satisfies*

$$VCdim(H_{\rho^+}) \leq \min(\lceil \frac{D^2}{\rho^2} \rceil, n) + 1.$$

Recall that we define a support vector (x_i, y_i) to be such that the constraint

$$y_i(x_i \cdot \psi^* + b^*) - 1 \geq 0$$

is active. Namely,

$$y_i(x_i \cdot \psi^* + b^*) - 1 = 0.$$

This implies the vector x_i is one of the closest to the optimal hyperplane. In the expansion,

$$\psi^* = \sum_{i=1}^{k} \lambda_i^* y_i x_i$$

a nonzero value of the Lagrange multiplier λ^* means that the constraint must be active and consequently, the expansion of the optimal hyperplane vector ψ^* is in terms of of support vectors. Although the vector ψ^* is unique, its expansion in terms of support vectors is not. Let $\mathcal{K}$, the essential support vectors, be the set of support vectors that are in all the expansions of ψ^*. We can then prove the following theorems.

Theorem 6 *(Vapnik)*

$$|\mathcal{K}| \leq n.$$

Also,

Theorem 7 *(Vapnik)*

$$E(error) \leq \frac{E(|\mathcal{K}|)}{k+1} \leq \frac{n}{k+1}$$

3. The support vector machine

Since the VC dimension of the set of linear classifier in R^n is $n+1$, it is clear that large dimensional classification problems are difficult. However, a consequence of Theorem 5 is that even if the dimension n is large, if the data can be separated by a large margin, then the VC dimension of classifiers with that margin is bounded by

$$\min(\lceil \frac{D^2}{\rho^2} \rceil, n) + 1.$$

where ρ is the margin.

This observation gives rise to the the support vector machine as follows: Map the data to a high dimensional feature space, and in this space classify using the optimal hyperplane. If the margin happens to be large then Theorem 5 suggests that we will obtain good generalization performance. However, even if the optimal hyperplane in the high dimensional space has a good margin, the dimensionality of this space may discourage computations there. To deal with this problem, let us recall that both the hyperplane

$$f(x) = \sum_{i=1}^{k} \lambda_i^* y_i (x_i \cdot x) + b^*$$

and the objective function

$$W(\lambda) = \sum_{i=1}^{k} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{k} \lambda_i \lambda_j y_i y_j (x_i \cdot x_j).$$

do not depend explicitly on the dimensionality of the vector x and x_i but only depend upon the inner product of such vectors.

Suppose that we can map

$$\Phi : R^n \to \mathcal{H}$$

from our space of covariates R^n to some Hilbert space $\mathcal{H}$ in such a way that

$$< \Phi(x), \Phi(y) >= K(x, y)$$

for some known and easy to evaluate function K. Then finding an optimal hyperplane in $\mathcal{H}$ amounts to optimizing the objective function

$$W(\lambda) = \sum_{i=1}^{k} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{k} \lambda_i \lambda_j y_i y_j K(x_i, x_j).$$

which then defines the classifier $sign(f(x))$ where

$$f(x) = \sum_{i=1}^{k} \lambda_i^* y_i K(x_i, x) + b^*.$$

Thus, even though $\mathcal{H}$ may be of very large dimensionality, we do not need to operate there to construct an optimal hyperplane there.

Let us describe a simple example, evidently first described by Vapnik: Let $x = (x_1, x_2)$ be a two dimensional vector. The function $K(x, y) = (x \cdot y)^2$ can be represented as $K(x, y) = \Phi(x) \cdot \Phi(y)$ where

$$\Phi(x) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right)$$

since

$$K(x, y) = (x \cdot y)^2 = (x_1 y_1 + x_2 y_2)^2 = x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2 = \Phi(x) \cdot \Phi(y).$$

It is interesting to note that the maps

$$\Phi(x) = \left(x_1^2 - x_2^2, 2 x_1 x_2, x_1^2 + x_2^2\right)$$

and

$$\Phi(x) = \left(x_1^2, x_1 x_2, x_1 x_2, x_2^2\right)$$

also accomplish the same task.

Indeed, although this simple example is very educational it is also misleading since we can also map to infinite dimensional Hilbert spaces, just as long as we can evaluate inner products there without going to the Hilbert space. A special example of when this can be done is accomplished by utilizing Mercer's theorem.

Theorem 8 *(Mercer) Consider a compact subset C of R^n. For a symmetric, continuous, and square integrable function $K(x,y)$ over C to have an absolutely uniformly convergent representation*

$$K(x,y) = \sum_{i=1}^{\infty} \alpha_i \Psi_i(x) \Psi_i(y)$$

with $\alpha > 0$ it is necessary and sufficient that

$$\int_C \int_C K(x,y) g(x) g(y) \geq 0$$

for all $g \in L^2(C)$.

Consequently, for any kernel $K(x,y)$ which satisfies the condition of the theorem, the map $\Phi : C \to \mathcal{L}^2$ defined by

$$x \mapsto (\sqrt{\alpha_1}\Psi_1(x), \sqrt{\alpha_2}\Psi_2(x), ...)$$

satisfies

$$K(x,y) = \Phi(x) \cdot \Phi(y).$$

4. Statistical properties of support vector machines

Although Vapnik's theorem 4 is a good motivation for the support vector machine, the VC bound obtained depends upon the data and therefore VC theory cannot be applied directly to obtain performance bounds for the support vector machine. This situation is resolved by Shawe-Taylor et. al [11]. We quote here their result, without proof.

Theorem 9 *(Shawe-Taylor, Bartlett, Williamson, Anthony)*

Suppose the support of the x marginal lies in a ball of radius D. Fix δ and ρ. If we succeed in classifying the data with an optimal hyperplane with margin greater than ρ, and $b \leq D$, then with probability greater than $1-\delta$ the generalization error is bounded by

$$\frac{2}{k}\left(\frac{577D^2}{\rho^2} \log\left(\frac{8ek\rho^2}{577D^2}\right) \log 32k + \log\frac{8k}{\delta}\right).$$

5. Support vector machines for non-separable data

When the data are not separable, the previous theorems do not apply. Indeed, the margin is then negative. However, the performance bounds above have been extended by Bartlett [1] to the case of nonseparable data as follows.

Theorem 10 *(Bartlett)*
Suppose the support of the x marginal lies in a ball of radius D. There is a constant c such that if we fix δ and ρ, with probability greater than $1-\delta$ the generalization error is bounded by

$$\frac{m}{k}+\sqrt{\frac{c}{k}\left(\frac{D^2}{\rho^2}\log^2 k+\log\frac{1}{\delta}\right)},$$

where m is the number of samples with margin less than ρ.

Consequently, any algorithm which tries to minimize the number of samples with margin less than some fixed ρ could be a good candidate for a support vector machine for nonseparable data. Recall the modified optimization problem that determines the optimal hyperplane {8,9}. The optimal hyperplane is $x\cdot\psi^*+b^*=0$.

The constraints are meant to separate the data, and among those planes that separate the data, minimizing $|\psi|^2$ accomplishes separation with the greatest margin. If the data is not separable, then these constraints cannot be satisfied, so an alternative formulation is needed. Suppose we relax the constraints to

$$y_i(x_i\cdot\psi+b)\geq 1-\xi_i$$

with $\xi_i\geq 0$ but impose the penalty

$$\Theta(\xi)=\sum_{i=1}^{k}\theta(\xi_i)$$

where

$$\theta(\eta)=0,\qquad \eta=0 \tag{10}$$
$$\theta(\eta)=1,\qquad \eta>0. \tag{11}$$

This penalty counts the number of data points that are not classified correctly. However, we now have gone from the optimization problem

$$\min\frac{1}{2}|\psi|^2$$
$$y_i(x_i\cdot\psi+b)\geq 1$$

with one optimization criteria to something like

$$\min \frac{1}{2}|\psi|^2$$
$$\min \Theta(\xi)$$
$$y_i(x_i \cdot \psi + b) \geq 1 - \xi_i.$$

However, this is not a bonafide optimization problem. Indeed, there is no canonical way to determine one so we have many choices to make. A possible solution is the following:

$$\min \Theta(\xi) \tag{12}$$
$$y_i(x_i \cdot \psi + b) \geq 1 - \xi_i \tag{13}$$
$$|\psi|^2 \leq A^2 \tag{14}$$

for some predetermined A.

This amounts to minimizing the number of points that have margin smaller than the cutoff $\frac{1}{A}$. However, the smaller the margin cutoff, the larger the number of points which will not satisfy the cutoff. On the other hand the larger the margin on the remaining points the better for generalization bounds. Indeed, one can see the balance between these two terms in the estimate of generalization error of Bartlett [1]

$$\frac{m}{k} + \sqrt{\frac{c}{k}\left(\frac{D^2}{\rho^2}\log^2 k + \log\frac{1}{\delta}\right)}.$$

where m is the number of samples with margin less than $\rho = \frac{1}{A}$.

The difficulty with this optimization problem is that this problem is close to a known to be NP-Complete problem and is suspected to be hard. Therefore, since we have already performed an adhoc modification because of non-separability, let us perform another modification to reduce the computational complexity. We wish to do so without giving up the existence of performance bounds. Consider simply changing to a new loss function

$$\theta(\eta) = \eta^p.$$

This function tends to the original as p tends to 0. However, for $p = 1$ or $p = 2$ we can solve the optimization problem in polynomial time. Unfortunately, for $p = 1$ there are no performance bounds. On the other hand, Shawe-Taylor and Cristianini [12] have performance bounds for $p = 2$.

Theorem 11 *(Shawe-Taylor, Cristianini)*

Suppose the support of the x marginal lies in a ball of radius D. Fix ρ and δ. With probability greater than $1 - \delta$, the generalization error corresponding to any hyperplane $x \cdot \psi + b = 0$ is bounded by

$$\frac{2}{k}\left(h \log\left(\frac{8ek}{h}\right) \log 32k + \log \frac{180k(21 + \log k)^2}{\delta}\right).$$

where

$$h = \frac{65[(D + \sqrt{\Theta^*})^2 + 2.25D\sqrt{\Theta^*}]}{\rho^2}$$

and

$$\Theta^* = \sum_{i=1}^{k} \left(\max\left(0, \rho - y_i(x_i \cdot \psi + b)\right)\right)^2.$$

Consequently, for this optimization problem, not only can we compute its solution but we can estimate its generalization error as a function of prescribed ρ and computed Θ. Since our optimization problem, for fixed ρ, amounts to minimization of Θ, the optimization problem amounts to optimization of the performance bound. One can also see the balance between the Θ and ρ in the function

$$h = \frac{65[(D + \sqrt{\Theta^*})^2 + 2.25D\sqrt{\Theta^*}]}{\rho^2}$$

The "optimal" balance can be determined using an iterative scheme to find the ρ that minimizes h. Such a scheme would require that we solve the QP optimization problem at each iteration.

Alternatively, we could consider a modification that optimizes ρ and Θ simultaneously, such as

$$\min\left(\frac{1}{2}|\psi|^2 + C\Theta(\xi)\right) \tag{15}$$

$$y_i(x_i \cdot \psi + b) \geq 1 - \xi_i, \tag{16}$$

$$\xi_i \geq 0. \tag{17}$$

This problem shares the same computational advantages as the previous. The constant C represents the balance between the margin cutoff and the number of training points with margin less than this cutoff. Once again we could consider iterative schemes for the determination of the constant C. Of course, the error bounds of Shawe-Taylor and Cristianini [12] still apply here.

Algorithm development for SVMs has focused on the problem {15,16,17} with $p = 1$ (i.e. $\theta(\eta) = \eta$). Although this creates a slight disconnect between the terms being optimized and those in the bound, it is believed to produce superior results in practice.

6. Algorithms for Support Vector Machines

Let

$$z_i = \Phi(x_i)$$

and

$$z_i \cdot z_j = \Phi(x_i) \cdot \Phi(x_j) = K(x_i, x_j)$$

Examples of kernels commonly used for real-valued data include

1. Polynomial
$$K(x,y) = (x \cdot y + 1)^p$$
2. Gaussian (RBF)
$$K(x,y) = e^{-\|x-y\|^2/2\sigma^2}$$
3. Sigmoid (neural network)
$$K(x,y) = \tanh\left(a_1(x \cdot y) + a_0\right)$$

Note that restrictions must be placed on (a_1, a_0) to satisfy Mercer's condition [3]. We wish to solve the following (primal) quadratic programming (QP) problem to produce a linear classifier in $\mathcal{H}$,

$$\begin{array}{ll} \min & \frac{1}{2}\|\psi\|^2 + C\sum_{i=1}^{k}\xi_i \\ \text{s.t.} & y_i(z_i \cdot \psi + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \quad i = 1, 2, ..., k \end{array} \tag{18}$$

Cortes and Vapnik call this the "soft margin formulation" [4]. This problem has size $dim(z) + k$ variables. The size of $dim(z)$ in the SVM can make this problem too large to solve on a digital computer. Fortunately there is a dual form with more manageable size. Consider the Lagrangian of the QP above,

$$L(\psi, b, \xi, \lambda, \alpha) = \frac{1}{2}\|\psi\|^2 + C\sum_{i=1}^{k}\xi_i - \sum_i \lambda_i\left(y_i(z_i \cdot \psi + b) - 1 + \xi_i\right) - \sum_i \alpha_i\xi_i$$

Differentiating with respect to ψ, b, ξ and applying the Kuhn-Tucker optimality conditions gives

$$\psi^* = \sum_i \lambda_i y_i z_i \tag{19}$$

$$\sum_i \lambda_i y_i = 0 \tag{20}$$

$$\alpha_i + \lambda_i = C \tag{21}$$

Substituting (19-21) into the Lagrangian yields

$$L(\psi^*, b^*, \xi^*, \lambda, \alpha) = \sum_i \lambda_i - \frac{1}{2}\sum_i\sum_j \lambda_i\lambda_j y_i y_j z_i \cdot z_j$$

Maximizing L with respect to λ, α and incorporating the conditions in (20-21) gives the Wolfe Dual optimization problem

$$\begin{array}{ll} \max & -\frac{1}{2}\lambda \cdot (Q\lambda) + \lambda \cdot 1 \\ \text{s.t.} & \lambda \cdot y = 0 \\ & 0 \leq \lambda_i \leq C, \quad i = 1, 2, ..., k \end{array} \tag{22}$$

where Q is of the form

$$Q_{ij} = y_i y_j (z_i \cdot z_j) = y_i y_j K(x_i, x_j)$$

This problem has size k (independent of $dim(z)$!). Further, it can be solved using computations that live entirely in the original data space $R^n \times \{1, -1\}$ through use of the kernel. Q is positive semidefinite, making this a concave QP problem with simple constraints (one equality constraint and k box constraints). Thus, it admits a polynomial-time solution [2].

It is easy to show that Q is symmetric and positive semi-definite. If we define $Z = [y_1 z_1, y_2 z_2, ..., y_k z_k]$ then Q is equal to $Z^T Z$, and Q is symmetric by $Q^T = (Z^T Z)^T = Q$ and positive semi-definite by $u \cdot (Qu) = |Zu|^2 \geq 0$.

For small to moderate values of k the Wolfe Dual can be solved using standard algorithms for convex QP problems [5] (although care must be taken to account for the reduced rank of Q). For large k however, the storage requirements can be excessive for (most) modern day computers. For example, with $k = 50,000$ approximately 20 GBytes of storage would be required for Q. This barrier can be overcome by decomposing the original QP problem into a collection smaller problems.

Suppose we partition λ into two sets, a working set λ_W and a non-working set λ_N. Similarly y is partitioned into y_W and y_N, and Q is partitioned as follows

$$Q = \begin{bmatrix} Q_W & Q_{WN} \\ Q_{NW} & Q_N \end{bmatrix}$$

where $Q_{WN} = Q_{NW}$. Then (22) can be written

$$\begin{array}{ll} \max & -\frac{1}{2}\lambda_W Q_W \lambda_W + \lambda_W \cdot (1 - Q_{WN}\lambda_N) - \frac{1}{2}\lambda_N Q_N \lambda_N + \lambda_N \cdot 1 \\ \text{s.t.} & \lambda_W \cdot y_W + \lambda_N \cdot y_N = 0 \\ & 0 \leq \lambda_i \leq C, \quad i = 1, 2, ..., k \end{array} \tag{23}$$

With λ_N fixed this becomes a QP problem of size $dim(\lambda_W)$ with the same generic properties as the original. This motivates algorithmic strategies that solve a sequence of QP problems over different working sets. The key is to select a working set at each step that will guarantee progress toward the original problem solution.

One approach is to design the sequence of QP problems to search for a working set that contains all the support vectors. This approach is motivated by the fact that we expect the number of support vectors to be small, and that the solution to (22) can be obtained by solving a (smaller) QP problem over the support vectors alone. To see why, recall that $\lambda_i^* \neq 0$ corresponds to an "active" constraint in the primal, which in turn corresponds to a support vector. Typically $n \ll k$, and since Theorem 6 gives $|\mathcal{K}| \leq n$, it is reasonable to expect the number of support vectors to be small (this expectation is less plausible in the non-separable case). Now, if the support vectors are known the solution to (22) can be obtained by placing them in the working set and solving (23) with $\lambda_N = 0$. In the *chunking* [13] strategy the working set is initialized to a subset of the data, and an initial QP problem is solved. Then the non-support vectors from this solution are moved to the non-working set and a "chunk" (subset) of samples from $S - W$ that violate the Kuhn-Tucker optimality conditions are moved to W. This forms a new working set, and the process is repeated until all samples satisfy the Kuhn-Tucker optimality conditions. A proof of convergence for this process is given by Osuna, et.al. [9], who also propose a slightly different approach. Their *decomposition* strategy is similar to chunking except that the QP problems are always the same size [9]. The working set is initialized to a subset of the data, and an initial QP problem is solved. Then the non-support vectors from this solution are "swapped" with a subset of samples from $S - W$ that violate the Kuhn-Tucker optimality conditions. This forms a new working set, and the process is repeated until all samples satisfy the Kuhn-Tucker optimality conditions.

Note that both *chunking* and *decomposition* require the working set to be larger than the final number of support vectors, which is not known ahead of time. Alternatively we can modify the algorithm so that it allows support vectors to be swapped out of the working set. Employing swaps that remove support vectors from the working set leads to QP problems for which $\lambda_N \neq 0$ in (23), but this is easily accommodated. The key is to "swap in" samples from the non-working set that guarantee a reduction in the original criterion. steepest feasible descent direction vector [7]. One possible strategy is to select samples that correspond to the largest components of the Software that employs this method can be obtained at http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html. Platt's Sequential Minimal Optimization (SMO) algorithm employs essentially the same strategy, but restricts its working sets to size 2 [10]. The advantage of SMO is that the "size 2" QP problems have a simple closed form solution.

References

1. Bartlett, P. L., The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network, *IEEE Transactions on Information Theory* **44**(1998), 525–536.
2. M. Bellare and P. Rogaway, *The complexity of approximating a nonlinear program*, in "Complexity in numerical optimization", Ed: P.M. Pardalos, World Scientific Pub. Co., pp. 16-32, 1993.
3. Burges, C. J. C., A tutorial on Support Vector Machines for pattern recognition, *Data Mining and Knowledge Discovery* **2**(1998), 121–167.
4. Cortes, C. and Vapnik, V., Support-Vector networks, *Machine Learning* **20**(1995), 273–297.
5. P.E. Gill and W. Murray and M.H. Wright, *Practical optimization*, Academic Press, London;New York, 1981.
6. Hush, D., and Scovel, C., On the VC dimension of bounded margin classifiers, submitted to *Machine Learning*, June, 1999.
7. T. Joachims, Making large-scale SVM learning practical, *University of Dortmund Computer Science LS-8 Technical Report 24*, June, 1998.
8. von Neumann, J., Zur Theorie der Gesellsaftspiele, *Mathematische Annalen* **100**(1928),295–320.
9. E.E. Osuna, R. Freund, and F. Girosi, Support vector machines: training and applications, *MIT Technical Report AIM-1602*(1997).
10. J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in *Advances in Kernel Methods - Support Vector Learning,* B. Schlkopf, C. Burges, and A. Smola, eds., MIT Press, (1998).

11. Shawe-Taylor, J., Bartlett, P.L., Williamson, R. C., and M. Anthony, Structural Risk Minimization over Data-Dependent Hierarchies, *NeuroCOLT Technical Report* **NC-TR-96-053**(1996).
12. Shawe-Taylor, J., Cristianini, N., Robust bounds on generalization from the margin, *NeuroCOLT Technical Report* **NC2-TR-1998-029**(1998), (http://www.neurocolt.com/ abstracts/contents_1998.html)
13. Vapnik, V. N., *Statistical Learning Theory*, John Wiley and Sons, Inc., New York, 1998.

This article was processed using the LaTeX macro package with JNS style