Polynomial-Time Decomposition Algorithms for Support Vector Machines

LOS ALAMOS
NATIONAL LABORATORY

Don Hush and Clint Scovel
Computer Research and Applications Group, CCS–3
Mail Stop B265

LANL Technical Report: LA–UR–00-3800

Report Date: July 11, 2001

Abstract

This paper studies the convergence properties of a general class of decomposition algorithms for support vector machines (SVMs). We provide a model algorithm for decomposition, and prove necessary and sufficient conditions for stepwise improvement of this algorithm. We also prove convergence in criterion value for the model algorithm, thereby establishing convergence for many existing SVM algorithms. We introduce a simple "rate certifying" condition and prove a polynomial-time bound on the rate of convergence of the model algorithm when it satisfies this condition. Although it is not clear that existing SVM algorithms satisfy this condition, we provide a version of the model algorithm that does. For this algorithm we show that when the slack multiplier C satisfies $\sqrt{1/2} \leq C \leq mL$, where m is the number of samples and L is a matrix norm, then it takes no more than $4LC^2m^4/\epsilon$ iterations to drive the criterion to within ϵ of its optimum.

Acknowledgments

We would like to thank the reviewers for their thorough and thoughtful considerations of the manuscript, in particular the reviewer that recommended a correction to our proof of theorem 7. We also gratefully acknowledge support from the DOE AMS program in applied mathematics at Los Alamos National Laboratory.

1 Introduction

The soft margin formulation in (Cortes & Vapnik, 1995) has the advantage that it provides a design criterion for support vector machines (SVMs) for both separable and nonseparable data while maintaining a convex programming problem. To maintain a computationally feasible approach across all kernels, algorithms are developed for the Wolfe Dual Quadratic Program (QP) problem whose size is independent of the dimension of the ambient space. The Gram matrix for the Wolfe Dual is $m \times m$ where m is the number of data samples. For large m the storage requirements for this matrix can be excessive, thereby preventing the application of many existing QP solvers. This barrier can be overcome by decomposing the original QP problem into smaller QP problems and employing algorithmic strategies that solve a sequence of these smaller QP problems. For the class of algorithms considered here these smaller QP problems are restrictions of the original QP problem where optimization is allowed over a subset of the data called the working set. The key is to select working sets that guarantee progress toward the original problem solution at each step. Such algorithms are commonly referred to as decomposition algorithms, and many existing SVM algorithms fall into this class (Cristianini & Shawe-Taylor, 2000; Joachims, 1998; Keerthi, Shevade, Bhattacharyya, & Murthy, 2001; Osuna, Freund, & Girosi, 1997; Platt, 1998; Vapnik, 1998). In this paper we provide a model algorithm for decomposition and prove necessary and sufficient conditions for stepwise improvement of this algorithm. These conditions require that each working set contain a *certifying pair* (defined in section 3). Computation of a certifying pair takes $O(m)$ time. We define a simple "rate certifying" condition on certifying pairs that enables the proof of a polynomial-time bound on the rate of convergence. It is not clear that the working sets chosen by existing SVM algorithms contain certifying pairs that satisfy this condition. On the other hand, we provide an $O(m \log m)$ algorithm for determining a certifying pair that does. The next section sets the stage for our development by providing a formal definition of the problem and establishing some of its basic properties.

2 Preliminaries

Let $S =< (x_1, y_1), ..., (x_m, y_m) >$ be a finite set of observations from a two-class pattern recognition problem where $x_i \in X$ and $y_i \in \{-1, 1\}$. The Support Vector Machine (SVM) maps the space of covariates X to a Hilbert space $\mathcal{H}$ of higher dimension (possible infinite), and fits an optimal linear classifier in $\mathcal{H}$. It does so by choosing a map $\Phi : X \to \mathcal{H}$ in such a way that $\Phi(x) \cdot \Phi(y) = K(x, y)$ for some known and easy to evaluate function K. Sufficient conditions for the existence of such a map are provided by Mercer's theorem (Vapnik, 1998). Let $z_i = \Phi(x_i)$ so that

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

A linear classifier in $\mathcal{H}$ is given by

$$\hat{y} = \text{sign}(\psi \cdot z + b). \tag{1}$$

In the soft margin formulation of (Cortes & Vapnik, 1995) the optimal ψ is given by

$$\psi = \sum_{i=1}^{m} \lambda_i y_i z_i \tag{2}$$

where $\lambda \in \Re^m$ optimizes the Wolfe Dual quadratic programming problem,

$$WD(S): \qquad \begin{array}{ll} \max & -\frac{1}{2}\lambda \cdot (Q\lambda) + \lambda \cdot 1 \\ \text{s.t.} & \lambda \cdot y = 0 \\ & 0 \le \lambda_i \le C, \quad i = 1, 2, ..., m \end{array} \tag{3}$$

where

$$Q_{ij} = y_i y_j (z_i \cdot z_j) = y_i y_j K(x_i, x_j). \tag{4}$$

The choice of the unspecified parameter $C > 0$ has been investigated but we do not address that here. Once λ has been determined the optimal value of b is given by

$$\tilde{v}(\lambda)^*_{low} \le -b \le \tilde{v}(\lambda)^*_{high}$$

where $\tilde{v}(\lambda)^*_{low}$ and $\tilde{v}(\lambda)^*_{high}$ are defined in section 3. This paper is concerned with the analysis of a class of algorithms for $WD(S)$ that are motivated by situations where m is so large that direct storage of Q is prohibitive.

Let $WD(S)$ denote an instance of the Wolfe Dual defined by the sample set S. Let $\Lambda(S)$ represent the set of feasible solutions for $WD(S)$,

$$\Lambda(S) = \{\lambda : (0 \le \lambda_i \le C) \cap (\lambda \cdot y = 0)\}$$

Note that $\Lambda(S)$ is both convex and compact. Denote the Wolfe Dual criterion by

$$R(\lambda) = -\frac{1}{2}\lambda \cdot (Q\lambda) + \lambda \cdot 1 \tag{5}$$

and let $\Lambda^*(S)$ represent the set of optimal solutions for $WD(S)$,

$$\Lambda^*(S) = \{\lambda : \lambda = \arg \max_{\lambda \in \Lambda(S)} R(\lambda)\}.$$

With $Z = [y_1 z_1, y_2 z_2, ..., y_m z_m]$ we can write $Q = Z^T Z$, verifying that Q is symmetric and positive semi-definite. Thus, $R(\lambda)$ is a concave function over $\Lambda(S)$ and $R^* = R(\Lambda^*(S))$ is unique. The Lagrangian for $WD(S)$ takes the form

$$L(\lambda, \mu, \alpha, \beta) = \frac{1}{2}\lambda \cdot (Q\lambda) - \lambda \cdot 1 + \mu(\lambda \cdot y) - \sum_i \alpha_i \lambda_i - \sum_i \beta_i (C - \lambda_i)$$

where $\alpha_i \ge 0, \beta_i \ge 0$.

Define

$$v_i = \sum_j \lambda_j y_j K(x_j, x_i) \; - \; y_i, \qquad i = 1, 2, ..., m. \tag{6}$$

Then the Karush-Kuhn-Tucker (KKT) conditions (e.g. see (Avriel, 1976), p.96) for $WD(S)$ take the form

$$\begin{aligned}
&y_i(v_i + \mu) = \alpha_i - \beta_i \\
&\alpha_i \lambda_i = 0, \quad i = 1, 2, ..., m \\
&\beta_i(C - \lambda_i) = 0, \quad i = 1, 2, ..., m
\end{aligned}$$

where we have made use of the relation

$$(Q\lambda)_i - 1 = \sum_j \lambda_j y_i y_j z_j \cdot z_i - 1 = y_i(\sum_j \lambda_j y_j z_j \cdot z_i - y_i) = y_i(\psi \cdot z_i - y_i) = y_i v_i$$

There are three regimes for λ_i; two where it equals a bound, and one where it falls between the bounds. Combining the conditions above with these three regimes we obtain a simpler set of conditions that are equivalent to the KKT conditions

$$\begin{aligned}
&y_i(v_i + \mu) = 0, \quad 0 < \lambda_i < C \\
&y_i(v_i + \mu) \leq 0, \quad \lambda_i = C \\
&y_i(v_i + \mu) \geq 0, \quad \lambda_i = 0
\end{aligned} \tag{7}$$

It is possible to use the satisfaction of these equations as a stopping condition for optimization algorithms, but they involve μ. An alternative set of optimality conditions were introduced in (Keerthi *et al.*, 2001; Keerthi & Gilbert, 2000) that do not use μ. In the next section we present these conditions and use them to develop a simple optimality test.

3 Tests for Optimality using Certifying Pairs

We define a partition of the index set of S based upon the data

$$(v_i, y_i, \lambda_i).$$

Define

$$\begin{aligned}
&I_{low} = \{i : (\lambda_i = C, y_i = 1) \cup (\lambda_i = 0, y_i = -1)\} \\
&I_{high} = \{i : (\lambda_i = C, y_i = -1) \cup (\lambda_i = 0, y_i = 1)\} \\
&I_{int} = \{i : 0 < \lambda_i < C\}
\end{aligned} \tag{8}$$

and

$$\begin{aligned}
&V_{low} = \{v_i : i \in I_{low}\} \\
&V_{high} = \{v_i : i \in I_{high}\} \\
&V_{int} = \{v_i : i \in I_{int}\}
\end{aligned} \tag{9}$$

and let

$$v^*_{low} = \sup_{i \in I_{low}} \{v_i\} \tag{10}$$

$$v^*_{high} = \inf_{i \in I_{high}} \{v_i\} \tag{11}$$

where the *sup* and *inf* of the empty set are defined as $-\infty$ and ∞ respectively.

Definition 1. λ is *properly ordered* for S if $|V_{int}| = 0$ and

$$v^*_{low} \le v^*_{high}$$

or $|V_{int}| = 1$ and

$$v^*_{low} \le V_{int} \le v^*_{high}.$$

We now prove a result first stated by Keerthi and Gilbert (Keerthi & Gilbert, 2000).

Theorem 1. *(Keerthi and Gilbert)*

A feasible λ for the Wolfe dual problem $WD(S)$ is optimal if and only if λ is properly ordered for S.

Proof. The optimality conditions (7) can be rewritten as

$$\begin{aligned} &v_i + \mu \ge 0, \quad i \in I_{high} \\ &v_i + \mu \le 0, \quad i \in I_{low} \\ &v_i + \mu = 0, \quad i \in I_{int}. \end{aligned} \tag{12}$$

Now suppose that λ is optimal. Then equations (12) imply that

$$\begin{aligned} &|V_{int}| \le 1 \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{high}, \quad j \in I_{low} \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{high}, \quad j \in I_{int} \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{int}, \quad j \in I_{low} \end{aligned}$$

The first equation implies that $|V_{int}| = 0$ or 1 and the second equation implies that $v^*_{low} \le v^*_{high}$. When $|V_{int}| = 1$ the second and third equations imply that

$$v^*_{low} \le V_{int} \le v^*_{high}$$

and so λ is properly ordered. On the other hand, suppose λ is properly ordered. Then $|V_{int}| \le 1$. By the definitions of v^*_{low} and v^*_{high} it is clear that

$$\begin{aligned} &|V_{int}| \le 1 \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{high}, \quad j \in I_{low} \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{high}, \quad j \in I_{int} \\ &v_i - v_j = (v_i + \mu) - (v_j + \mu) \ge 0, \quad i \in I_{int}, \quad j \in I_{low} \end{aligned}$$

and we can choose $-\mu$ to be any point in $[v^*_{low}, v^*_{high}]$ when $|V_{int}| = 0$ and $-\mu = V_{int}$ when $|V_{int}| = 1$ so that the conditions (12) are satisfied. Consequently, λ is optimal if and only if it is properly ordered for S.

♦

Tests for proper ordering can be simplified if we define

$$\tilde{I}_{low} = I_{low} \cup I_{int}$$
$$\tilde{I}_{high} = I_{high} \cup I_{int}$$

and

$$\tilde{v}^*_{low} = \sup_{i \in \tilde{I}_{low}} \{v_i\} \tag{13}$$

$$\tilde{v}^*_{high} = \inf_{i \in \tilde{I}_{high}} \{v_i\} \tag{14}$$

Then λ is properly ordered for $WD(S)$ if and only if

$$\tilde{v}^*_{low} \leq \tilde{v}^*_{high}.$$

The proof of this statement follows directly from the proof of Theorem 1.

Lack of optimality can be determined by the existence of a certifying pair.

Definition 2. A *certifying pair* for $\lambda \in \Lambda(S)$ is a pair of indices i and j in the index set of S whose values (v_i, y_i, λ_i) and (v_j, y_j, λ_j) are sufficient to prove that λ is not properly ordered for S.

We note that Keerthi et. al. (Keerthi & Gilbert, 2000) refer to this as a violating pair. However, because we later define *rate certifying pair* we decided not to adopt this terminology.

Theorem 2. *λ is not properly ordered for S if and only if there exists a certifying pair. A certifying pair can be obtained by making at most one pass through the data while making two comparisons.*

Proof. Suppose that λ is not properly ordered for S. Then there exists indices $i \in \tilde{I}_{high}$ and $j \in \tilde{I}_{low}$ such that $v_i < v_j$. Choose any such pair. To determine a certifying pair make one pass through the data while keeping track of indices that represent $\tilde{v}^*_{high}$ and $\tilde{v}^*_{low}$. Stop at the first point where $\tilde{v}^*_{high} < \tilde{v}^*_{low}$.

♦

4 A General Decomposition Algorithm

Algorithmic solutions for the Wolfe dual must consider the fact that when m is large the storage requirements for Q can be excessive. This barrier can be overcome by decomposing the original QP problem into smaller QP problems.

Suppose we partition the index set of λ into a working set W and a non-working set W^c. Note that W indexes a subset of the data. Then $\lambda = (\lambda_W, \lambda_{W^c})$ and $y = (y_W, y_{W^c})$ are partitioned accordingly and Q is partitioned as follows

$$Q = \begin{bmatrix} Q_W & Q_{WW^c} \\ Q_{W^cW} & Q_{W^c} \end{bmatrix}$$

where $Q_{WW^c} = Q^T_{W^cW}$. Then (3) can be written

$$\begin{array}{ll} \max & -\frac{1}{2}\lambda_W Q_W \lambda_W + \lambda_W \cdot (1 - Q_{WW^c}\lambda_{W^c}) - \frac{1}{2}\lambda_{W^c} Q_{W^c}\lambda_{W^c} + \lambda_{W^c}\cdot 1 \\ \text{s.t.} & \lambda_W \cdot y_W + \lambda_{W^c}\cdot y_{W^c} = 0 \\ & 0 \le \lambda_i \le C, \quad i = 1, 2, ..., m \end{array} \tag{15}$$

With λ_{W^c} 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.

Theorem 3. *Consider the subset constrained Wolfe dual problem defined as follows. Consider a feasible λ. Define a subset W of the index space of S with complement W^c. Optimize the Wolfe dual criterion with respect to $\acute{\lambda}$ subject to the constraint that $\acute{\lambda} = \lambda$ on W^c. Let λ^* denote a solution to this constrained problem. Then, $R(\lambda^*) > R(\lambda)$, if and only if W contains a certifying pair for λ.*

Proof. Since R is concave, λ is non-optimal for $WD(S)$ if and only if there is a feasible infinitesimal $\dot{\lambda}$ at λ such that

$$dR(\lambda)\cdot\dot{\lambda} > 0. \tag{16}$$

Further, the solution to the constrained Wolfe dual produces an increase in R if and only if there is a feasible constrained $\dot{\lambda}_W$ (with nontrivial components on W only) such that $dR(\lambda)\cdot\dot{\lambda}_W > 0$. Consequently, to prove the theorem it is sufficient to show that a feasible $\dot{\lambda}_W$ exists that satisfies (16) if and only if W contains a certifying pair.

The derivative of R is given by

$$dR(\lambda)\cdot\dot{\lambda} = (1 - Q\lambda)\cdot\dot{\lambda} = \sum_i (1 - y_i\psi\cdot z_i)\dot{\lambda}_i = -\sum_i y_i v_i \dot{\lambda}_i = -\sum_i d_i v_i = -d\cdot v$$

where $d_i = y_i\dot{\lambda}_i$. The feasible directions $\dot{\lambda}$ satisfy $\dot{\lambda}\cdot y = 0$, $\dot{\lambda}_i \ge 0$ when $\lambda_i = 0$, and $\dot{\lambda}_i \le 0$ when $\lambda_i = C$. In terms of d these conditions become $d\cdot 1 = 0$, $d_i \ge 0$ when $i \in I_{high}$, and $d_i \le 0$ when $i \in I_{low}$. Decompose $d = d_{high} + d_{low} + d_{int}$ and $v = v_{high} + v_{low} + v_{int}$ into their components under the subsets defined by I_{high}, I_{low}, and I_{int}. Then (16) can be written

$$d_{high}\cdot v_{high} + d_{low}\cdot v_{low} + d_{int}\cdot v_{int} < 0 \tag{17}$$

and the feasibility constraints are

$$d_{high}\cdot 1 + d_{low}\cdot 1 + d_{int}\cdot 1 = 0, \quad d_{high} \ge 0, \; d_{low} \le 0, \; d_{int} \text{ free}. \tag{18}$$

Assume that W contains a certifying pair. Then it must satisfy one of the following inequalities,

$$\begin{array}{l} v_i < v_j, \; i, j \in I_{int} \\ v_i < v_j, \; i \in I_{high}, j \in I_{int} \\ v_i < v_j, \; i \in I_{int}, j \in I_{low} \\ v_i < v_j, \; i \in I_{high}, j \in I_{low} \end{array}$$

In all four cases we can verify (17)-(18) by choosing $d_i = -d_j > 0$ for the certifying pair and $d = 0$ for all other indices so that

$$dR(\lambda) \cdot \dot{\lambda}_W = -d_i(v_i - v_j) > 0$$

The proof of "if" is finished.

Now assume that there is a feasible $\dot{\lambda}_W$ for which $dR(\lambda) \cdot \dot{\lambda}_W > 0$. Then (17)-(18) are satisfied. Let $V_{int}(W)$ ($I_{int}(W)$) be the restrictions of V_{int} (I_{int}) to the indices of W. If $|V_{int}(W)| > 1$ then any two components $i, j \in I_{int}(W)$ for which $v_i \neq v_j$ constitute a certifying pair. If $|V_{int}(W)| = 1$ let $v^* = V_{int}(W)$ and write (17) as

$$d_{high} \cdot v_{high} + d_{low} \cdot v_{low} + v^* d_{int} \cdot 1 < 0$$

Combining with (18) gives

$$d_{high} \cdot (v_{high} - v^* 1) + d_{low} \cdot (v_{low} - v^* 1) < 0, \qquad d_{high} \geq 0,\ d_{low} \leq 0$$

For this inequality to hold at least one of the two terms must be negative. To make the first term negative at least one component of $(v_{high} - v^* 1)$ must be negative. Similarly, to make the second term negative at least one component of $(v_{low} - v^* 1)$ must be positive. Either case gives a certifying pair. Finally, if $|V_{int}(W)| = 0$ then (17)-(18) becomes

$$\begin{aligned} &d_{high} \cdot v_{high} + d_{low} \cdot v_{low} < 0 \\ &d_{high} \cdot 1 = -d_{low} \cdot 1, \quad d_{high} \geq 0,\ d_{low} \leq 0 \end{aligned}$$

Without loss of generality let the components of d_{high} and d_{low} be normalized so that

$$\begin{aligned} &\textstyle\sum_{i \in I_{high}} d_i = 1 \\ &\textstyle\sum_{i \in I_{low}} -d_i = 1 \end{aligned}$$

Then $(d_{high} \cdot v_{high} + d_{low} \cdot v_{low})$ is the difference between convex combinations of $V_{high}(W)$ and convex combinations of $V_{low}(W)$. For this difference to be negative the two convex hulls must overlap. This implies a certifying pair. This finishes the "only if" part, so the proof is finished.

♦

Theorem 3 motivates a class of algorithms of the form Algorithm A_1 below. Members from this class solve a sequence of decomposed QP problems of the form in (15) over working sets that can vary in size from 2 to $|S|$ and contain at least one certifying pair. The initialization ensures that $W(0)$ contains at least one certifying pair. The `QPSolve` routine on line 11 solves the QP problem restricted to the current working set $W(k-1)$. Line 14 chooses a certifying pair for inclusion in the next working set. The algorithm terminates when a certifying pair no longer exists. The `AnySubset` routine on line 18 chooses a subset of samples to be included with the certifying pair in the next working set. This subset is irrelevant to the issue of guaranteed improvement, but is likely to have an effect on the rate of convergence.

Algorithm A_1: General Decomposition Algorithm.

`INPUTS:` $S = \{(x_i, y_i)\}_{i=1}^m$

`OUTPUT:` λ

$I_S = \{1, 2, ..., m\}$
$\lambda_i = 0, \ i \in I_S$
$\tilde{I}_{low} = \{i : y_i = -1\}$
$\tilde{I}_{high} = \{i : y_i = 1\}$
$W(0) \leftarrow$ subset of I_S with at least one sample from each class.
$k \leftarrow 1$
loop
  $\lambda(k) \leftarrow$ `QPSolve`$(W(k-1), \lambda(k-1), S)$
  Update membership in $\tilde{I}_{low}, \tilde{I}_{high}$ for samples in $W(k-1)$
  $v_i(k) \leftarrow \sum_j y_j \lambda_j(k) K(x_j, x_i) \ - \ y_i, \quad i \in I_S$
  $P \leftarrow$ any pair i, j that satisfy $i \in \tilde{I}_{low}, j \in \tilde{I}_{high}$, and $v_i > v_j$
  if $(P = \emptyset)$ **then**
    return$(\lambda(k))$
  end if
  $W(k) \leftarrow P \cup$ `AnySubset`$(I_S \backslash P)$
  $k \leftarrow k + 1$
end loop

5 Convergence

In general, the stepwise improvement of Algorithm A_1 is not sufficient to guarantee convergence. Indeed, Keerthi and Ong (Keerthi & Ong, 2000) provide an example where each working set contains a certifying pair but Algorithm A_1 does not converge to the optimal solution. However, convergence results have been proved for some special cases, e.g. see (Keerthi & Gilbert, 2000), (Chang, Hsu, & Lin, 2000), (Lin, 2000). The convergence result in (Keerthi & Gilbert, 2000) defines λ_τ to be τ-optimal if it satisfies $\tilde{v}^*_{low} < \tilde{v}^*_{high} + \tau$ for some $\tau > 0$. It then shows that the generalized SMO (GSMO) algorithm converges to a τ-optimal solution in a finite number of steps. The GSMO algorithm is a special case of Algorithm A_1 where the `AnySubset` function returns the empty set. The analysis in (Keerthi & Gilbert, 2000) leaves open the question of accuracy with respect to the optimal solution, that is it provides no bound on $|R(\lambda_\tau) - R^*|$ or $|\lambda_\tau - \lambda^*|$.

(Chang *et al.*, 2000) give a proof of convergence for a special case of Algorithm A_1 where the working set is defined to be the indices corresponding to the nontrivial components of d in the solution to the optimization problem

$$\begin{aligned} \max \quad & dR(\lambda(k)) \cdot d \\ \text{s.t.} \quad & d \cdot y = 0 \\ & 0 \leq (\lambda(k) + d)_i \leq C, \quad i = 1, 2, ..., m \\ & |\{d_i : d_i \neq 0\}| \leq q \end{aligned}$$

where $q \geq 2$. Their proof shows that, with this choice of working set, Algorithm A_1 produces a sequence $\{\lambda(k)\}$ whose limit point is optimal for $WD(S)$. More recently (Lin, 2000) has provided a similar proof of convergence for SVM^{light} where the working set is defined by Joachims (Joachims, 1998) to be the indices corresponding to the nontrivial components of d in the solution to a slightly different optimization problem

$$\begin{aligned} \max \quad & dR(\lambda(k)) \cdot d \\ \text{s.t.} \quad & d \cdot y = 0, \quad -1 \leq d_i \leq 1, \ i = 1, 2, ..., m \\ & d_i \geq 0, \text{ if } (\lambda(k))_i = 0, \qquad d_i \leq 0, \text{ if } (\lambda(k))_i = C \\ & |\{d_i : d_i \neq 0\}| \leq q \end{aligned}$$

where $q \geq 2$.

The analysis in (Chang *et al.*, 2000) and (Lin, 2000) is asymptotic and therefore leaves open the question of finite step convergence to the optimum. In the following section we provide a finite step convergence proof for a special case of Algorithm A_1 that corresponds to "chunking".

5.1 Finite Step Convergence for Chunking

Chunking (as described in (Cristianini & Shawe-Taylor, 2000)) is a decomposition method in which each working set contains all support vectors from the current solution plus an additional set of samples that violate an "optimality condition". If the optimality condition is chosen so that the additional set always contains at least one certifying pair [1] then the resulting algorithm takes the form of Algorithm A_1 where the `AnySubset` routine returns, at a minimum, the indices for all samples with $\lambda_i > 0$. The following theorem holds for this class of chunking algorithms.

Theorem 4. *Let S be a finite set of observations containing at least one sample from each class. Consider Algorithm A_1 where the* `AnySubset` *routine returns any set that contains the indices for all samples with $\lambda_i > 0$. This algorithm converges to a solution of $WD(S)$ for finite k.*

Proof. Algorithm A_1 terminates only when there are no certifying pairs, and if it terminates then $\lambda \in \Lambda^*(S)$. We assume that `QPSolve` provides an exact solution to the constrained Wolfe dual. Then Theorem 3 guarantees that when we are not at a solution the criterion for $WD(S)$ is strictly increased from one step to the next, i.e. $R(\lambda(k+1)) > R(\lambda(k))$. Since $\lambda = 0$ on W^c, all nontrivial contribution to R is made by the working set. Thus, no working set is revisited, and since there are a finite number of working sets, and R^* is unique, termination in finite k is guaranteed. ♦

[1] This requires a slight modification to the chunking algorithm in (Cristianini & Shawe-Taylor, 2000).

We now show that with the proper choice of certifying pair we can provide polynomial-time bounds on the run time of Algorithm A_1.

5.2 Convergence Rate

In this section we give a finite step convergence result for Algorithm A_1 when each working set contains a *rate certifying pair* (defined below). We also provide bounds on the convergence rate. More specifically we give a polynomial bound on the number of iterations required to drive $|R(\lambda) - R^*|$ to within ϵ of its optimum. Note that the criterion has a strong dependence on the size of the sample set m. In general R becomes unbounded as $m \to \infty$. Consequently the development of convergence rates requires the normalization of R in terms of the number of samples. For example, in empirical risk minimization it is standard to divide the number of training errors by the number of samples to obtain the fraction of training errors. However at present we know of no natural normalization for R. Therefore to allow for the incorporation of an appropriate normalization we implicitly denote the error tolerance as a function of m through the notation ϵ_m.

Let λ^* be an optimal parameter value and $R^* = R(\lambda^*)$ denote the optimal criterion value. Let $r(\lambda) = R(\lambda) - R^*$ so that $r \leq 0$ and $r^* = 0$. Because of concavity,

$$R(\lambda^*) - R(\lambda) \leq dR(\lambda) \cdot (\lambda^* - \lambda)$$

which can be rewritten as

$$-r(\lambda) \leq dr(\lambda) \cdot (\lambda^* - \lambda).$$

If we define

$$\sigma(\lambda) = \sup_{\acute{\lambda} \in \Lambda(S)} dr(\lambda) \cdot (\acute{\lambda} - \lambda), \tag{19}$$

we obtain

$$-r(\lambda) \leq \sigma(\lambda). \tag{20}$$

Let γ denote a parameter value which differs from λ in at most two places and define

$$\acute{\sigma}(\lambda) = \sup_{\gamma \in \Lambda(S)} dr(\lambda) \cdot (\gamma - \lambda). \tag{21}$$

When $\acute{\sigma}(\lambda) \geq \alpha\sigma(\lambda)$ for some $0 < \alpha < 1$ then we can bound the distance to the optimum by $-r(\lambda) \leq \acute{\sigma}(\lambda)/\alpha$. We use this to determine a bound on the convergence rate for Algorithm A_1.

Let λ_k denote the value of the state at the k-th iteration and let γ_k denote a parameter that differs from λ_k in at most two indices. We note that in previous sections the subscripted λ_k was used for the k-th component of the vector λ and the parenthetic $\lambda(k)$ was used for the state of the algorithm at the k-th iteration. However, in the present analysis we need no components of the vector and feel the use of λ_k for the state at the k-th iteration is a better notation for this section. Let $R_k = R(\lambda_k)$, $r_k = r(\lambda_k)$, $dr_k = dr(\lambda_k)$, $\sigma_k = \sigma(\lambda_k)$, and $\acute{\sigma}_k = \acute{\sigma}(\lambda_k)$ where

$$\sigma(\lambda_k) = \sup_{\acute{\lambda} \in \Lambda(S)} dr(\lambda_k) \cdot (\acute{\lambda} - \lambda_k) \tag{22}$$

and

$$\acute{\sigma}(\lambda_k) = \sup_{\gamma_k \in \Lambda(S)} dr(\lambda_k) \cdot (\gamma_k - \lambda_k). \tag{23}$$

Definition 3. Algorithm A_1 is a *rate certifying algorithm* if there exists an α such that the certifying pair chosen on line 14 satisfies

$$\acute{\sigma}_k \geq -\alpha r_k, \qquad 0 < \alpha < 1$$

for all k. A *rate certifying pair* is a pair of indices in the index set of S for which $\acute{\sigma}_k \geq -\alpha r_k$ at iteration k of a rate certifying algorithm.

Chang, et. al. (Chang *et al.*, 2000) establish a relationship of this type for a particular choice of rate certifying pair with $\alpha = \frac{1}{m^2}$ and use it to prove asymptotic convergence. The following theorem gives a bound on the number of iterations that are sufficient to drive the criterion to within ϵ_m of its optimum for a rate certifying algorithm.

Theorem 5. *Let $\lambda(k)$ denote the sequence of states generated by Algorithm A_1. If it is a rate certifying algorithm then $R^* - R(\lambda_k) \leq \epsilon_m$ after*

$$k \geq \frac{1}{q^* \alpha}\left(\frac{BL}{\alpha \epsilon_m} - 1\right) + 1$$

iterations, where

$$q^* = \min\{\frac{1}{4C^2}, \frac{1}{2}\},$$

$$B = \max\left\{1, \frac{\alpha(R^* - R(\lambda_0))}{L}\right\},$$

and L is the maximum of the norms of the 2 by 2 matrices determined by restricting Q to 2 indices. In words, if we wish to get an accuracy of ϵ_m, then it is sufficient to perform $\frac{1}{q^*\alpha}\left(\frac{BL}{\alpha\epsilon_m} - 1\right) + 1$ iterations.

Proof. Let $\{i, j\} \subset W(k)$ denote the indices of a rate certifying pair in the working set such that $\acute{\sigma}_k \geq -\alpha r_k$.

Following (Dunn, 1979) we consider the following auxiliary equations. Let γ_k differ from λ_k in the two indices i, j.

$$\gamma_k^* = \arg \max_{\gamma_k \in \Lambda(S)} dr_k \cdot (\gamma_k - \lambda_k) \tag{24}$$

$$\bar{\lambda}_{k+1} = \lambda_k + \omega_k(\gamma_k^* - \lambda_k) \tag{25}$$

$$\omega_k = \begin{cases} \frac{\beta_k}{|\gamma_k^* - \lambda_k|^2} & , \; 0 < \frac{\beta_k}{|\gamma_k^* - \lambda_k|^2} \leq 1 \\ 1 & , \; \frac{\beta_k}{|\gamma_k^* - \lambda_k|^2} > 1 \end{cases} \tag{26}$$

$$\beta_{k+1} = (1 - \omega_k \alpha)\beta_k + \frac{\alpha \omega_k^2}{2}|\gamma_k^* - \lambda_k|^2 \tag{27}$$

where $\beta_0 = 1$.

Since $R_{k+1} - R_k = r_{k+1} - r_k$ and $R(\lambda_{k+1}) \geq R(\bar{\lambda}_{k+1})$,

$$\begin{aligned} r_{k+1} - r_k \geq \quad r(\bar{\lambda}_{k+1}) - r_k &= \omega_k dr_k \cdot (\gamma_k^* - \lambda_k) - \tfrac{\omega_k^2}{2}(\gamma_k^* - \lambda_k) \cdot Q(\gamma_k^* - \lambda_k) \\ &\geq \omega_k dr_k \cdot (\gamma_k^* - \lambda_k) - \tfrac{\omega_k^2}{2} L|\gamma_k^* - \lambda_k|^2 \end{aligned}$$

With $\acute{\sigma}_k = dr_k \cdot (\gamma_k^* - \lambda_k) \geq -\alpha r_k$ we have

$$\begin{aligned} r_{k+1} - r_k &\geq \omega_k \acute{\sigma}_k - \tfrac{\omega_k^2}{2} L|\gamma_k^* - \lambda_k|^2 \\ &\geq -\omega_k \alpha r_k - \tfrac{\omega_k^2}{2} L|\gamma_k^* - \lambda_k|^2 \end{aligned}$$

which can be written

$$r_{k+1} \geq (1 - \omega_k \alpha) r_k - \frac{\omega_k^2}{2} L|\gamma_k^* - \lambda_k|^2.$$

Define $\rho_k = -\frac{\alpha r_k}{L}$ and $B = \max(1, \rho_0)$. Then

$$\rho_{k+1} \leq (1 - \omega_k \alpha)\rho_k + \frac{\alpha \omega_k^2}{2}|\gamma_k^* - \lambda_k|^2.$$

We show by induction that $\rho_k \leq B\beta_k$ as follows.

$$\begin{aligned} &\rho_{k+1} \leq (1 - \omega_k \alpha)\rho_k + \frac{\alpha \omega_k^2}{2}|\gamma_k^* - \lambda_k|^2 \\ &\leq (1 - \omega_k \alpha) B\beta_k + \frac{\alpha \omega_k^2}{2}|\gamma_k^* - \lambda_k|^2 \\ &= B((1 - \omega_k \alpha)\beta_k + \frac{\alpha \omega_k^2}{2B}|\gamma_k^* - \lambda_k|^2) \\ &\leq B\beta_{k+1}. \end{aligned}$$

We now control β_k. Plugging the definition of ω_k in equation (26) into equation (27) for β_k we obtain

$$\begin{array}{ll} \beta_{k+1} = \beta_k - \frac{\alpha}{2|\gamma_k^* - \lambda_k|^2}\beta_k^2 & \frac{\beta_k}{|\gamma_k^* - \lambda_k|^2} \leq 1 \\ \beta_{k+1} = (1 - \alpha)\beta_k + \frac{\alpha}{2}|\gamma_k^* - \lambda_k|^2 & \frac{\beta_k}{|\gamma_k^* - \lambda_k|^2} > 1 \end{array} \tag{28}$$

In the latter case $|\gamma_k^* - \lambda_k|^2 < \beta_k$ so then

$$\beta_{k+1} \leq (1 - \frac{\alpha}{2})\beta_k.$$

Putting the two equations from (28) together we obtain

$$\beta_{k+1} \leq \beta_k - q_k \beta_k^2 \tag{29}$$

where

$$q_k = \min\{\frac{\alpha}{2|\gamma_k^* - \lambda_k|^2}, \frac{\alpha}{2\beta_k}\}$$

$$= \alpha \min\{\frac{1}{2|\gamma_k^* - \lambda_k|^2}, \frac{1}{2\beta_k}\} \geq \alpha \min\{\frac{1}{4C^2}, \frac{1}{2\beta_k}\} \geq \alpha \min\{\frac{1}{4C^2}, \frac{1}{2}\} = \alpha q^* \tag{30}$$

since $\beta_k \leq \beta_0 = 1$. Therefore, by (Dunn, 1979) equations (29) and (30) imply that

$$\beta_k \leq \frac{1}{1 + \alpha q^*(k-1)}$$

but going back through the relations $\rho_k = -\frac{\alpha r_k}{L}$ and $\rho_k \leq B\beta_k$ implies

$$-r_k \leq \frac{BL}{\alpha\left(1 + \alpha q^*(k-1)\right)}.$$

Consequently, when

$$k \geq \frac{1}{\alpha q^*}\left(\frac{BL}{\alpha \epsilon_m} - 1\right) + 1,$$

then

$$\frac{BL}{\alpha\left(1 + \alpha q^*(k-1)\right)} \leq \epsilon_m$$

and

$$-r_k \leq \epsilon_m.$$

The proof is finished. ♦

5.3 Efficient Computation of a Rate Certifying Pair

In the previous section we determined that $\acute{\sigma}_k \geq \alpha\sigma_k$ is sufficient to establish $\acute{\sigma}_k \geq -\alpha r_k$. (Chang *et al.*, 2000) show that a certifying pair always exists such that $\acute{\sigma}_k \geq \frac{1}{m^2}\sigma_k$. They do this by considering the solution to a linear programming (LP) problem (similar to the LP problem for σ_k), and then restricting this solution to two indices. In this section we show how to solve this LP to produce a rate certifying pair in $O(m \log m)$ operations.

Let $\lambda = \lambda(k)$ be the current solution and define

$$L_i = -\lambda_i, \quad i = 1, 2, ..., m$$

$$U_i = C - \lambda_i, \quad i = 1, 2, ..., m$$

Let η^* be the solution to the linear program

$$\begin{aligned} \max \quad & dR(\lambda) \cdot \eta \\ \text{s.t.} \quad & \eta \cdot y = 0 \\ & L_i \leq \eta_i \leq U_i, \quad i = 1, 2, ..., m \end{aligned} \tag{31}$$

Note that the solution to this problem and (19) are related by $\acute{\lambda}^* = \eta^* + \lambda$. As in section 3, define

$$\tilde{I}_{low}^{\eta} = \{i : (\eta_i^* = L_i, y_i = -1) \cup (\eta_i^* = U_i, y_i = 1) \cup (L_i < \eta_i^* < U_i)\}$$

$$\tilde{I}^{\eta}_{high} = \{i : (\eta^*_i = U_i, y_i = -1) \cup (\eta^*_i = L_i, y_i = 1) \cup (L_i < \eta^*_i < U_i)\}$$

$$\begin{aligned} \tilde{v}^{\eta}_{low} &= \max_{i \in \tilde{I}^{\eta}_{low}} \{v_i\} \\ &= -\infty \qquad \tilde{I}^{\eta}_{low} = \emptyset \end{aligned}$$

$$\begin{aligned} \tilde{v}^{\eta}_{high} &= \min_{i \in \tilde{I}^{\eta}_{high}} \{v_i\} \\ &= \infty \qquad \tilde{I}^{\eta}_{high} = \emptyset \end{aligned}$$

and choose

$$\mu^* \in [\tilde{v}^{\eta}_{low}, \tilde{v}^{\eta}_{high}]$$

From (Chang *et al.*, 2000) we know that the certifying pair (i, j) given by

$$i = \arg\max_p |\eta^*_p||v_p - \mu^*| \tag{32}$$

$$j = \arg \max_{p \neq i, (y_i \eta_i)(y_p \eta_p) < 0} |\eta^*_p| \tag{33}$$

is a rate certifying pair with rate $\alpha = \frac{1}{m^2}$. The following lemma establishes that this pair can be determined in a computationally efficient manner.

Lemma 1. *Given y, $\lambda = \lambda(k)$, and $v = v(\lambda(k))$ the rate certifying pair (i, j) in (32) and (33) can be computed in $O(m \log m)$ time.*

Proof. We describe an algorithm that computes this pair in $O(m \log m)$ time. Our algorithm solves the LP in (31) and then computes the two indices using (32)-(33). Once the LP is solved it is straightforward to implement (32)-(33) in $O(m)$ steps, so we describe only the LP solution.

Consider the LP in (31). Recall that $dR(\lambda)_i = -y_i v_i$. The Karush-Kuhn-Tucker conditions for the solution η are

$$\begin{aligned} &y_i v_i = \alpha_i - \beta_i + \mu y_i \\ &\alpha_i(\eta_i - L_i) = 0 \\ &\beta_i(U_i - \eta_i) = 0 \end{aligned}$$

with $\alpha_i \geq 0$, $\beta_i \geq 0$ and $\eta \cdot y = 0$. These equations can be written

$$\begin{aligned} v_i - \mu \geq 0, \quad i \in I^{\eta}_{high} \\ v_i - \mu \leq 0, \quad i \in I^{\eta}_{low} \\ v_i - \mu = 0, \quad i \in I^{\eta}_{int} \end{aligned} \tag{34}$$

where

$$\begin{aligned} I^{\eta}_{low} &= \{i : (\eta_i = U_i, y_i = 1) \cup (\eta_i = L_i, y_i = -1)\} \\ I^{\eta}_{high} &= \{i : (\eta_i = U_i, y_i = -1) \cup (\eta_i = L_i, y_i = 1)\} \\ I^{\eta}_{int} &= \{i : L_i < \eta_i < U_i\} \end{aligned}$$

To solve these equations, fix μ and determine η to satisfy

$$\begin{array}{ll} i \in I^{\eta}_{high} & v_i - \mu \geq 0 \\ i \in I^{\eta}_{low} & v_i - \mu \leq 0. \end{array} \tag{35}$$

For example, if $v_i > \mu$, then set $\eta_i = L_i$ if $y_i = 1$ and $\eta_i = U_i$ if $y_i = -1$. To determine μ we use the constraint

$$\eta \cdot y = 0.$$

Written out this becomes

$$0 = \eta \cdot y = \underbrace{\sum_{i \in I^{\eta}_{low}} y_i \eta_i}_{\geq 0} + \underbrace{\sum_{i \in I^{\eta}_{high}} y_i \eta_i}_{\leq 0} + \sum_{i \in I^{\eta}_{int}} y_i \eta_i$$

Our strategy is to choose μ so that it splits the samples into I^{η}_{low} and I^{η}_{high} in such a way that the first and second sums cancel as closely as possible. When they do not cancel exactly we shift μ so that the split occurs *on* a value v_i, thereby placing samples with this value into I^{η}_{int} and allowing us to choose their parameters η_i to satisfy the equality. More specifically we sort the values of v in increasing order and use k to index the sorted list (i.e. $v_k \leq v_{k+1}$). As μ increases from $-\infty$ to ∞, jumping over values where $\mu = v_k$, with η being determined as above, the value of $\eta \cdot y$ is monotonically increasing and must pass from negative to positive. In fact it is easy to see that $\eta \cdot y$ increases by C each time an individual sample is jumped. Suppose that this increasing function achieves the value 0 on the interval (v_k, v_{k+1}). Then we let μ be any value in this interval and since I^{η}_{int} is empty and η was chosen to satisfy (35) we have a solution. Suppose this increasing function skips the value 0 and jumps from $-a < 0$ to $b > 0$ at $\mu = v_k$ and there are a total of $M \geq 1$ samples with this value of v (i.e. $v_k = v_{k+1} = ... = v_{k+M-1}$). Then set $\mu = v_k$ and place the first $M_1 = \lfloor a/C \rfloor$ of these samples in I^{η}_{low} (the rest remain in I^{η}_{high}). If a/C is integral then this gives $\eta \cdot y = 0$ and we have a solution once again (with M of the samples satisfying (35) with equality and $I^{\eta}_{int} = \emptyset$ as before). If a/C is not integral then its remainder is used to determine η_{k+M_1}, the component of η corresponding to v_{k+M_1}. This gives $\eta \cdot y = 0$ and places this sample in I^{η}_{int}, and again we have a solution. Note that there are many solutions to these equations. This construction gives η^* and μ^*, both of which are necessary to implement (32)-(33). It takes $O(m \log m)$ steps to sort the v, followed by an additional pass through the list to initialize η, placing all samples in I^{η}_{high} and yielding $\eta(0) \cdot y$. Since $\eta \cdot y$ begins at $\eta(0) \cdot y$ and increases by C each time μ is increased past a data point, the components of η for all the points up to $k^* = \lfloor \frac{-\eta(0) \cdot y}{C} \rfloor$ are changed by C placing them in I^{η}_{low}. Then, if $\frac{-\eta(0) \cdot y}{C}$ is not integral its remainder is used to determine the component of η for the $k^* + 1$ sample which is moved to I^{η}_{int}. Updating η in this way requires at most one complete pass through the list. This completes the proof. ♦

Algorithm 5.3 computes a rate certifying pair using the method described in the proof above. In addition to the sort, this algorithm makes a total of four passes through the list. The number of computations in this procedure can sometimes be reduced. Let i, j be a rate certifying pair. Then v_i and v_j are on opposite sides of μ^*, and since i, j is also a certifying pair μ^* must lie between $\tilde{v}^*_{high}$ and $\tilde{v}^*_{low}$ (defined in (13) and (14)). This means that the sorting operation required in our search for μ can be restricted to the v_i in this interval. Since the sorting operation dominates the run time this can lead to a substantial savings when the number of samples in this interval is small.

Algorithm A_2: Rate Certifying Pair Algorithm.

`INPUTS:` y, v, and λ (at the current iteration)

`OUTPUT:` (i_1, i_2) {sample indices for a rate certifying pair}

{L is an ordered list of indices in nondecreasing order of $\{v_i\}$ so that $v_{L(l)} \leq v_{L(l+1)}$}
$L \leftarrow$ `LSort`(V)

{initially place all samples in I^η_{high} and compute $\eta(0) \cdot y$}
$EtaDotY \leftarrow 0$
for $i = 1$ to m **do**
  if $(y_i = 1)$ **then** $\eta_i \leftarrow -\lambda_i$
  else if $(y_i = -1)$ **then** $\eta_i \leftarrow C - \lambda_i$
  $EtaDotY \leftarrow EtaDotY + \eta_i y_i$
end for

{determine split point index and move samples into I^η_{low}}
$l^* \leftarrow \lfloor -EtaDotY/C \rfloor$
for $l = 1$ to l^* **do**
  $\eta_{L(l)} \leftarrow \eta_{L(l)} + y_{L(l)} \cdot C$
end for
$EtaDotY \leftarrow EtaDotY + l^* \cdot C$

{if needed, move sample into I^η_{int}}
if $(EtaDotY < 0)$ **then**
  $l^* \leftarrow l^* + 1$
  $\eta_{L(l^*)} \leftarrow \eta_{L(l^*)} - y_{L(l^*)} \cdot EtaDotY$
  $\mu \leftarrow v_{L(l^*)}$
else
  $\mu \leftarrow$ value in $[v_{L(l^*)}, v_{L(l^*+1)}]$ {if $i^* = 0$ or $i^* = m$ then use $v_{L(i^*)} = v_{L(i^*+1)}$}
end if

{determine indices for rate certifying pair}
$i_1 \leftarrow \max_{i=1,m} |\eta_i||v_i - \mu|$
$i_2 \leftarrow \max_{i=1,m \text{ and } (y_{i_1}\eta_{i_1})(y_i\eta_i)<0} |\eta_i|$

Return$((i_1, i_2))$

5.4 Summary of Rates

If we use Algorithm A_2 to choose a rate certifying pair then $\alpha = \frac{1}{m^2}$ and by theorem 5 Algorithm A_1 will drive the criterion to within ϵ_m of its optimum in no more than

$$\frac{m^2}{q^*}\left(\frac{BLm^2}{\epsilon_m} - 1\right) + 1$$

iterations. Further, with $\lambda_0 = 0$ we have $R^* - R(\lambda_0) \leq dR(\lambda_0)(\lambda^* - \lambda) = 1 \cdot \lambda^* \leq Cm$ so that $B = 1$ when $C \leq Lm$. Thus, when $\sqrt{1/2} \leq C \leq Lm$, $q^* = \frac{1}{4C^2}$ and neglecting lower order terms, the number of iterations simplifies to

$$\frac{4LC^2m^4}{\epsilon_m}$$

In the case where the working sets are of size two we can use this result to establish a worst case overall run time for Algorithm A_1. At each iteration we must solve a 2 by 2 QP problem, update the $v_i(k)$, and determine the next certifying pair. The time to solve the 2 by 2 QP problem is a constant, and it takes order m operations to update the $v_i(k)$. If we add $m \log m$ operations to determine the certifying pair, the worst case run time is of order

$$\frac{4LC^2m^5 \log m}{\epsilon_m}$$

Now consider our choice for ϵ_m obtained through an appropriate normalization of R (see discussion at the beginning of this section). Because R tends to increase with m, ϵ_m will be an increasing function of m. Although the form of this function is not yet known it will clearly improve the run-time bounds presented above. For example, if $\epsilon_m = m^p\epsilon$ then the order of the polynomial in these bounds is reduced by p.

6 Discussion

This paper considers a class of algorithms for support vector machines that decompose the original Wolfe Dual QP problem into a sequence of smaller QP problems defined on subsets of the data. Following the work of Keerthi et al. (Keerthi & Gilbert, 2000; Keerthi *et al.*, 2001) we provide a scalar condition that is necessary and sufficient for optimality of the QP problem. This leads naturally to the introduction of certifying pairs as a necessary and sufficient condition for stepwise improvement, and motivates the use of Algorithm A_1 as a model algorithm for this problem. By leveraging the results of Chang, et al. (Chang *et al.*, 2000) we have developed Algorithm A_2 for selecting the certifying pair in Algorithm A_1. Theorem 5 shows that the number of iterations for this instantiation of Algorithm A_1 is $O(m^4)$ and the overall run time is $O(m^5 \log m)$.

Many existing SVM algorithms are either special cases of Algorithm A_1 or can be made so through slight modification. For example, Platt's Sequential Minimal Optimization (SMO) algorithm, which chooses working sets of size two, is designed to choose a pair that give a strict increase in R at each step (Platt, 1998). The original algorithm however, contains a flaw that can lead to improper behavior (Keerthi *et al.*, 2001; Keerthi & Gilbert, 2000). This behavior can be traced to its inability to guarantee a certifying pair in each working set. By forcing each working set to contain a certifying pair the corrected algorithm not only has guaranteed convergence, but also improved performance (Keerthi *et al.*, 2001).

The SVM^{light} algorithm in (Joachims, 1998) uses a modification of Zoutendijk's method (Zoutendijk, 1970) to choose working sets of size $q \geq 2$. This choice can be shown to contain the $q/2$ largest v_i from $\tilde{I}_{low}$ and the $q/2$ smallest v_i from $\tilde{I}_{high}$, thus guaranteeing at least one certifying pair.

The chunking algorithm described in (Cristianini & Shawe-Taylor, 2000) and the decomposition algorithm of (Osuna *et al.*, 1997) both attempt to ensure improvement in R by choosing working sets that include support vectors from the current solution plus a subset of samples that violate an "optimality condition" with respect to this solution. A strict implementation of the algorithms described in these papers can lead to undesirable behavior because they cannot guarantee a certifying pair in their working sets. However, such a guarantee can be achieved with a slight modification (as we did for the chunking algorithm in section 5.1).

It is not clear that the algorithms above satisfy the rate certifying condition in Definition 3, nor that this is necessary to establish rates for them. We have described a new SVM algorithm that satisfies the rate certifying condition and has polynomial-time rates. It is not yet clear how this algorithm will compare with existing algorithms in practice. Note that Keerthi's GSMO algorithm (Keerthi *et al.*, 2001) and Jochamin's SVM^{light} algorithm (Joachims, 1998) require $O(m)$ time to determine a certifying pair while A_2 requires $O(m \log m)$ time. However, we know of no bounds on the rates of convergence for GSMO and SVM^{light} (although they seem to work well in practice), but can guarantee a polynomial convergence rate when we use A_2.

Finally we note that the polynomial-time bound on the number of iterations scales as m^4, which is unattractive. We leave open the issue of the tightness of this bound, although we suspect that it may be loose. A closely related issue is the determination of a proper normalization for R that would give rise to an explicit functional dependence of ϵ on m. This is likely to improve the rate.

References

Avriel, M. (1976). *Nonlinear Programming: Analysis and Methods* (1st edition). Prentice Hall, Englewood Cliffs, N.J.

Chang, C., Hsu, C., & Lin, C. (2000). The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, *11*(4), 1003–1008.

Cortes, C., & Vapnik, V. (1995). Support-Vector networks. *Machine Learning*, *20*, 273–297.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (1st edition). Cambridge University Press, Canbridge ; United Kingdom.

Dunn, J. (1979). Rates of convergence for conditional gradient algorithms near singular and non-singular extremals. *SIAM J. Control and Optimization*, *17*(2), 187–211.

Joachims, T. (1998). Making large-scale SVM learning practical. In Scholkopf, B., Burges, C., & Smola, A. (Eds.), *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA.

Keerthi, S., & Gilbert, E. (2000). Convergence of a generalized SMO algorithm for SVM classifier design. Control division technical report CD-00-01, Dept. of Mechanical and Production Engineering, National University of Singapore. to appear in Machine Learning.

Keerthi, S., & Ong, C. (2000). On the role of the threshold parameter in SVM training algorithms. Control division technical report CD-00-09, Dept. of Mechanical and Production Engineering, National University of Singapore.

Keerthi, S., Shevade, S., Bhattacharyya, C., & Murthy, K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, *13*, 637–649.

Lin, C.-J. (2000). On the convergence of the decomposition method for support vector machines. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. to appear in IEEE Trans. Neural Networks.

Osuna, E., Freund, R., & Girosi, F. (1997). Support vector machines: training and applications. Technical report AIM-1602, MIT.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In Scholkopf, B., Burges, C., & Smola, A. (Eds.), *Advances in Kernel Methods - Support Vector Learning*, pp. 41–64. MIT Press, Cambridge, MA.

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

Zoutendijk, G. (1970). *Methods of Feasible Directions: A study in linear and non-linear programming*. Elsevier.