SSVM: A Smooth Support Vector Machine for Classification

Yuh-Jye Lee and O. L. Mangasarian
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706
yuh-jye@cs.wisc.edu, olvi@cs.wisc.edu

Abstract

Smoothing methods, extensively used for solving important mathematical programming problems and applications, are applied here to generate and solve an unconstrained smooth reformulation of the support vector machine for pattern classification using a completely arbitrary kernel. We term such reformulation a smooth support vector machine (SSVM). A fast Newton-Armijo algorithm for solving the SSVM converges globally and quadratically. Numerical results and comparisons are given to demonstrate the effectiveness and speed of the algorithm. On six publicly available datasets, tenfold cross validation correctness of SSVM was the highest compared with four other methods as well as the fastest. On larger problems, SSVM was comparable or faster than SVM^{light} [17], SOR [23] and SMO [27]. SSVM can also generate a highly nonlinear separating surface such as a checkerboard.

1 Introduction

Smoothing methods have been extensively used for solving important mathematical programming problems [8, 9, 7, 10, 11, 16, 29, 13]. In this paper, we introduce a new formulation of the support vector machine with linear (5) and nonlinear kernel (24) [30, 5, 6, 21] for pattern classification. We begin with the linear case which can be converted to an unconstrained optimization problem (7). Since the objective function of this unconstrained optimization problem is not twice differentiable, we employ a smoothing technique to introduce the smooth support vector machine (SSVM) (9). The smooth support vector machine has important mathematical properties such as strong convexity and infinitely often differentiability. Based on these properties, we can prove that when the smoothing parameter α in the SSVM approaches infinity, the unique solution of the SSVM converges to the unique solution of the original optimization problem (5). We also prescribe a Newton-Armijo algorithm (*i.e.* a Newton method with a stepsize determined by the simple Armijo rule which eventually becomes redundant) to solve the SSVM and show that this algorithm globally and quadratically converges to the unique solution of the SSVM. To construct a nonlinear classifier, a nonlinear kernel [30, 21] is used to obtain the nonlinear support vector machine (24) and its corresponding smooth version (25) with a completely arbitrary kernel. The smooth formulation (25) with a nonlinear kernel retains the strong convexity and twice differentiability and thus we can apply Newton-Armijo algorithm to solve it.

We briefly outline the contents of the paper now. In Section 2 we state the pattern classification problem and derive the smooth unconstrained support vector machine (9) from a constrained optimization formulation (5). Theorem 2.2 shows that the unique solution of our smooth approximation problem (9) will approach the unique solution of the original unconstrained optimization problem (7) as the smoothing parameter α approaches infinity. A Newton-Armijo Algorithm 3.1 for solving the SSVM converges globally and quadratically as shown in Theorem 3.2. In Section 4 we extend the SSVM to construct a nonlinear classifier by using a nonlinear kernel. All the Newton-Armijo convergence results apply to this nonlinear kernel formulation. Numerical tests and comparisons are given in Section 5. For moderate sized datasets, SSVM gave the highest tenfold cross validation correctness on six publicly available datasets as well as the fastest times when compared with four other methods given in [2, 4]. To demonstrate SSVM's capability in solving larger problems we compared SSVM with successive overrelaxation (SOR) algorithm [23], sequential minimal optimization (SMO) algorithm [27] and SVM^{light} [17] on the Irvine Machine Learning Database Repository Adult dataset [26]. It turns out that SSVM with a linear kernel is very efficient for this large dataset. By using a nonlinear kernel, SSVM can obtain very sharp separation for the highly nonlinear checkerboard pattern of [18] as depicted in Figures 4 and 5. To make this paper self-contained, we state and prove a Global Quadratic Convergence Theorem for the Newton-Armijo method in the Appendix which is employed in Theorem 3.2.

A word about our notation and background material. All vectors will be column vectors unless transposed to a row vector by a prime superscript $'$. For a vector x in the n-dimensional real space R^n, the plus function x_+ is defined as $(x_+)_i = \max\{0, x_i\}$, $i = 1, \ldots, n$. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$ and the p-norm of x will be denoted by $\|x\|_p$. For a matrix $A \in R^{m\times n}$, A_i is the ith row of A which is a row vector in R^n. A column vector of ones of arbitrary dimension will be denoted by e. We shall employ the MATLAB "dot" notation [25] to signify application of a function to all components of a matrix or a vector. For example if $A \in R^{m\times n}$, then $A^2_{\bullet} \in R^{m\times n}$ will denote the matrix obtained by squaring each element of A. For $A \in R^{m\times n}$ and $B \in R^{n\times l}$, the kernel $K(A, B)$ maps $R^{m\times n} \times R^{n\times l}$ into $R^{m\times l}$. In particular, if x and y are column vectors in R^n then, $K(x', y)$ is a real number, $K(x', A')$ is a row vector in R^m and $K(A, A')$ is an $m \times m$ matrix. If f is a real valued function defined on the n-dimensional real space R^n, the gradient of f at x is denoted by $\nabla f(x)$ which is a row vector in R^n and the $n \times n$ Hessian matrix of second partial derivatives of f at x is denoted by $\nabla^2 f(x)$. The level set of f is defined as $L_\mu(f) = \{x | f(x) \le \mu\}$ for a given real number μ. The base of the natural logarithm will be denoted by ε.

2 The Smooth Support Vector Machine (SSVM)

We consider the problem of classifying m points in the n-dimensional real space R^n, represented by the $m \times n$ matrix A, according to membership of each point A_i in the classes 1 or -1 as specified by a given $m \times m$ diagonal matrix D with ones or minus ones along its diagonal. For this problem the standard support vector machine with a linear kernel AA' [30, 12] is given by the following for some $\nu > 0$:

$$\begin{array}{rrcl} \displaystyle\min_{(w,\gamma,y)\in R^{n+1+m}} & \nu e'y + \frac{1}{2}w'w & & \\ \text{s.t.} & D(Aw - e\gamma) + y & \ge & e \\ & y & \ge & 0. \end{array} \tag{1}$$

Here w is the normal to the bounding planes:

$$\begin{array}{ccccc} x'w & - & \gamma & = & +1 \\ x'w & - & \gamma & = & -1, \end{array} \tag{2}$$

and γ determines their location relative to the origin. The first plane above bounds the class 1 points and the second plane bounds the class -1 points when the two classes are strictly linearly separable, that is when the slack variable $y = 0$. The linear separating surface is the plane

$$x'w = \gamma, \tag{3}$$

midway between the bounding planes (2). See Figure 1. If the classes are linearly inseparable then the two planes bound the two classes with a "soft margin" determined by a nonnegative slack variable y, that is:

$$\begin{array}{ccccccccl} x'w & - & \gamma & + & y_i & \geq & +1, & \text{for } x' = A_i \text{ and } D_{ii} = +1, \\ x'w & - & \gamma & - & y_i & \leq & -1, & \text{for } x' = A_i \text{ and } D_{ii} = -1. \end{array} \tag{4}$$

The 1-norm of the slack variable y is minimized with weight ν in (1). The quadratic term in (1), which is twice the reciprocal of the square of the 2-norm distance $\frac{2}{\|w\|_2}$ between the two bounding planes of (2) in the n-dimensional space of $w \in R^n$ for a *fixed* γ, maximizes that distance, often called the "margin". Figure 1 depicts the points represented by A, the bounding planes (2) with margin $\frac{2}{\|w\|_2}$, and the separating plane (3) which separates $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.

In our smooth approach, the square of 2-norm of the slack variable y is minimized with weight $\frac{\nu}{2}$ instead of the 1-norm of y as in (1). In addition the distance between the planes (2) is measured in the $(n+1)$-dimensional space of $(w, \gamma) \in R^{n+1}$, that is $\frac{2}{\|(w,\gamma)\|_2}$. Measuring the margin in this $(n+1)$-dimensional space instead of R^n induces strong convexity and has little or no effect on the problem as was shown in [23]. Thus using twice the reciprocal squared of the margin instead, yields our modified SVM problem as follows:

$$\begin{array}{ll} \min\limits_{w,\gamma,y} & \frac{\nu}{2}y'y + \frac{1}{2}(w'w + \gamma^2) \\ \text{s.t.} & D(Aw - e\gamma) + y \;\geq\; e \\ & \qquad\qquad\qquad\; y \;\geq\; 0. \end{array} \tag{5}$$

At a solution of problem (5), y is given by

$$y = (e - D(Aw - e\gamma))_+, \tag{6}$$

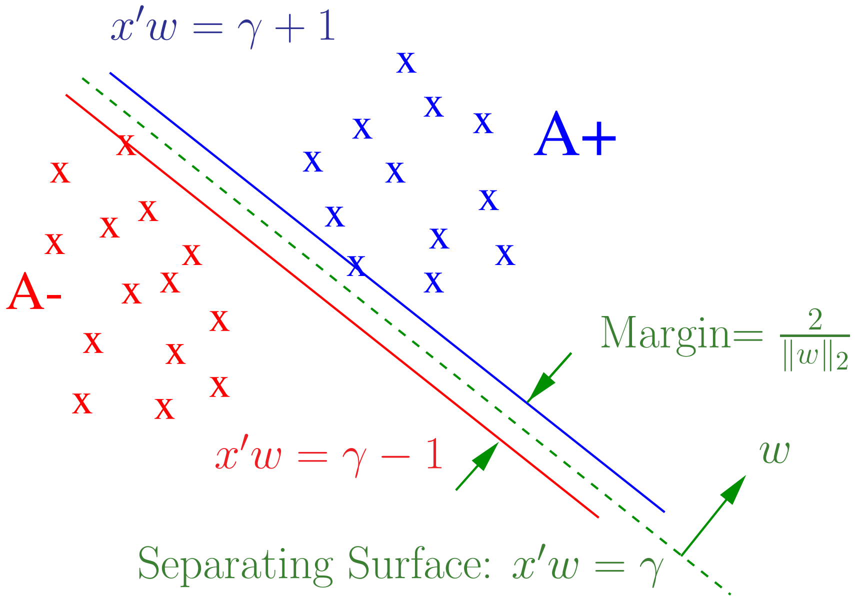

Figure 1: **The bounding planes (2) with margin $\frac{2}{\|w\|_2}$, and the plane (3) separating $A+$, the points represented by rows of A with $D_{ii} = +1$, from $A-$, the points represented by rows of A with $D_{ii} = -1$.**

where, as defined earlier, $(\cdot)_+$ replaces negative components of a vector by zeros. Thus, we can replace y in (5) by $(e - D(Aw - e\gamma))_+$ and convert the SVM problem (5) into an equivalent SVM which is an unconstrained optimization problem as follows:

$$\min_{w,\gamma} \quad \tfrac{\nu}{2}\|(e - D(Aw - e\gamma))_+\|_2^2 + \tfrac{1}{2}(w'w + \gamma^2). \tag{7}$$

This problem is a strongly convex minimization problem without any constraints. It is easy to show that it has a unique solution. However, the objective function in (7) is not twice differentiable which precludes the use of a fast Newton method. We thus apply the smoothing techniques of [8, 9] and replace x_+ by a very accurate smooth approximation (see Lemma 2.1 below) that is given by $p(x, \alpha)$, the integral of the sigmoid function $\frac{1}{1+\varepsilon^{-\alpha x}}$ of neural networks [19], that is

$$p(x, \alpha) = x + \frac{1}{\alpha}\log(1 + \varepsilon^{-\alpha x}), \;\; \alpha > 0. \tag{8}$$

This p function with a smoothing parameter α is used here to replace the plus function of (7) to obtain a smooth support vector machine **(SSVM)** :

$$\min_{(w,\gamma)\in R^{n+1}} \Phi_\alpha(w,\gamma) \;:=\; \min_{(w,\gamma)\in R^{n+1}} \frac{\nu}{2}\|p(e-D(Aw-e\gamma),\alpha)\|_2^2 + \frac{1}{2}(w'w+\gamma^2). \tag{9}$$

We will now show that the solution of problem (5) is obtained by solving problem (9) with α approaching infinity. We take advantage of the twice differentiable property of the objective function of (9) to utilize a quadratically convergent algorithm for solving the smooth support vector machine (9).

We begin with a simple lemma that bounds the square difference between the plus function $(x)_+$ and its smooth approximation $p(x,\alpha)$.

Lemma 2.1 For $x \in R$ and $|x| < \rho$: $p(x,\alpha)^2 - (x_+)^2 \le (\frac{\log 2}{\alpha})^2 + \frac{2\rho}{\alpha}\log 2$, where $p(x,\alpha)$ is the p function of (8) with smoothing parameter $\alpha > 0$.
Proof We consider two cases. For $0 < x < \rho$,

$$\begin{aligned} p(x,\alpha)^2 - (x_+)^2 &= \frac{1}{\alpha^2}\log^2(1+\varepsilon^{-\alpha x}) + \frac{2x}{\alpha}\log(1+\varepsilon^{-\alpha x}) \\ &\le (\frac{\log 2}{\alpha})^2 + \frac{2\rho}{\alpha}\log 2. \end{aligned}$$

For $-\rho < x \le 0$, $p(x,\alpha)^2$ is a monotonically increasing function, so we have

$$p(x,\alpha)^2 - (x_+)^2 = p(x,\alpha)^2 \le p(0,\alpha)^2 = (\frac{\log 2}{\alpha})^2.$$

Hence, $p(x,\alpha)^2 - (x_+)^2 \le (\frac{\log 2}{\alpha})^2 + \frac{2\rho}{\alpha}\log 2$. $\Box$

We now show that as the smoothing parameter α approaches infinity the unique solution of our smooth problem (9) approaches the unique solution of the equivalent SVM problem (7). We shall do this for a function $f(x)$ given in (10) below that subsumes the equivalent SVM function of (7) and for a function $g(x,\alpha)$ given in (11) below which subsumes the SSVM function of (9).

Theorem 2.2 Let $A \in R^{m\times n}$ and $b \in R^{m\times 1}$. Define the real valued functions $f(x)$ and $g(x,\alpha)$ in the n-dimensional real space R^n:

$$f(x) = \frac{1}{2}\|(Ax-b)_+\|_2^2 + \frac{1}{2}\|x\|_2^2 \tag{10}$$

and

$$g(x,\alpha) = \frac{1}{2}\|p((Ax-b),\alpha)\|_2^2 + \frac{1}{2}\|x\|_2^2, \tag{11}$$

with $\alpha > 0$.

(i) There exists a unique solution $\bar{x}$ of $\min\limits_{x\in R^n} f(x)$ and a unique solution $\bar{x}_\alpha$ of $\min\limits_{x\in R^n} g(x,\alpha)$.

(ii) For all $\alpha > 0$, we have the following inequality:

$$\|\bar{x}_\alpha - \bar{x}\|_2^2 \leq \frac{m}{2}((\frac{\log 2}{\alpha})^2 + 2\xi\frac{\log 2}{\alpha}), \tag{12}$$

where ξ is defined as follows:

$$\xi \;=\; \max_{1\leq i\leq m} |(A\bar{x}-b)_i|. \tag{13}$$

Thus, $\bar{x}_\alpha$ converges to $\bar{x}$ as α goes to infinity with an upper bound given by (12).

Proof (i) To show the existence of unique solutions, we know that since $x_+ \leq p(x,\alpha)$, the level sets $L_\nu(g(x,\alpha))$ and $L_\nu(f(x))$ satisfy

$$L_\nu(g(x,\alpha)) \subseteq L_\nu(f(x)) \subseteq \{x | \|x\|_2^2 \leq 2\nu\}, \tag{14}$$

for $\nu \geq 0$. Hence $L_\nu(g(x,\alpha))$ and $L_\nu(f(x))$ are compact subsets in R^n and the problems $\min\limits_{x\in R^n} f(x)$ and $\min\limits_{x\in R^n} g(x,\alpha)$ have solutions. By the strong convexity of $f(x)$ and $g(x,\alpha)$ for all $\alpha > 0$, these solutions are unique.

(ii) To establish convergence, we note that by the first order optimality condition and strong convexity of $f(x)$ and $g(x,\alpha)$ we have that

$$f(\bar{x}_\alpha) - f(\bar{x}) \geq \nabla f(\bar{x})(\bar{x}_\alpha - \bar{x}) + \frac{1}{2}\|\bar{x}_\alpha - \bar{x}\|_2^2 = \frac{1}{2}\|\bar{x}_\alpha - \bar{x}\|_2^2, \tag{15}$$

$$g(\bar{x},\alpha) - g(\bar{x}_\alpha,\alpha) \geq \nabla g(\bar{x}_\alpha,\alpha)(\bar{x} - \bar{x}_\alpha) + \frac{1}{2}\|\bar{x} - \bar{x}_\alpha\|_2^2 = \frac{1}{2}\|\bar{x} - \bar{x}_\alpha\|_2^2. \tag{16}$$

Since the p function dominates the plus function we have that $g(x,\alpha) - f(x) \geq 0$ for all $\alpha > 0$. Adding (15) and (16) and using this fact gives:

$$\begin{aligned} \|\bar{x}_\alpha - \bar{x}\|_2^2 &\leq (g(\bar{x},\alpha) - f(\bar{x})) - (g(\bar{x}_\alpha,\alpha) - f(\bar{x}_\alpha)) \\ &\leq g(\bar{x},\alpha) - f(\bar{x}) \\ &= \tfrac{1}{2}\|p((A\bar{x}-b),\alpha)\|_2^2 - \tfrac{1}{2}\|(A\bar{x}-b)_+\|_2^2. \end{aligned} \tag{17}$$

Application of Lemma 2.1 gives:

$$\|\bar{x}_\alpha - \bar{x}\|_2^2 \quad \leq \quad \tfrac{m}{2}((\tfrac{\log 2}{\alpha})^2 + 2\xi\tfrac{\log 2}{\alpha}), \tag{18}$$

where ξ is a fixed positive number defined in (13). The last term in (18) will converge to zero as α goes to infinity. Thus $\bar{x}_\alpha$ converges to $\bar{x}$ as α goes to infinity with an upper bound given by (12). $\Box$

We will now describe a Newton-Armijo algorithm for solving the smooth problem (9).

3 A Newton-Armijo Algorithm for the Smooth Support Vector Machine

By making use of the results of the previous section and taking advantage of the twice differentiability of the objective function of problem (9), we prescribe a quadratically convergent Newton algorithm with an Armijo stepsize [1, 15, 3] that makes the algorithm globally convergent.

Algorithm 3.1 Newton-Armijo Algorithm for SSVM (9)
Start with any $(w^0, \gamma^0) \in R^{n+1}$. Having (w^i, γ^i), stop if the gradient of the objective function of (9) is zero, that is $\nabla\Phi_\alpha(w^i, \gamma^i) = 0$. Else compute (w^{i+1}, γ^{i+1}) as follows:

(i) **Newton Direction**: Determine direction $d^i \in R^{n+1}$ by setting equal to zero the linearization of $\nabla\Phi_\alpha(w, \gamma)$ around (w^i, γ^i) which gives $n+1$ linear equations in $n+1$ variables:

$$\nabla^2\Phi_\alpha(w^i, \gamma^i)d^i = -\nabla\Phi_\alpha(w^i, \gamma^i)'. \tag{19}$$

(ii) **Armijo Stepsize** [1]: Choose a stepsize $\lambda_i \in R$ such that:

$$(w^{i+1}, \gamma^{i+1}) = (w^i, \gamma^i) + \lambda_i d^i \tag{20}$$

where $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ such that :

$$\Phi_\alpha(w^i, \gamma^i) - \Phi_\alpha((w^i, \gamma^i) + \lambda_i d^i) \geq -\delta\lambda_i\nabla\Phi_\alpha(w^i, \gamma^i)d^i \tag{21}$$

where $\delta \in (0, \frac{1}{2})$.

Note that a key difference between our smoothing approach and that of the classical SVM [30, 12] is that we are solving here a linear system of equations (19) instead of solving a quadratic program as is the case with the classical SVM. Furthermore, we can show that our smoothing Algorithm 3.1 converges globally to the unique solution and takes a pure Newton step after a finite number of iterations. This leads to the following quadratic convergence result:

Theorem 3.2 Let $\{(w^i, \gamma^i)\}$ be a sequence generated by Algorithm 3.1 and $(\bar{w}, \bar{\gamma})$ be the unique solution of problem (9).

(i) The sequence $\{(w^i, \gamma^i)\}$ converges to the unique solution $(\bar{w}, \bar{\gamma})$ from any initial point (w^0, γ^0) in R^{n+1}.

(ii) For any initial point (w^0, γ^0), there exists an integer $\bar{i}$ such that the stepsize λ_i of Algorithm 3.1 equals 1 for $i \geq \bar{i}$ and the sequence $\{(w^i, \gamma^i)\}$ converges to $(\bar{w}, \bar{\gamma})$ quadratically.

Although this theorem can be inferred from [15, Therem 6.3.4, pp 123-125], we give its proof in the Appendix for the sake of completeness. We note here that even though the Armijo stepsize is needed to guarantee that Algorithm 3.1 is globally convergent, in most of our numerical tests Algorithm 3.1 converged from any starting point without the need for an Armijo stepsize.

4 SSVM with a Nonlinear Kernel

In Section 2 the smooth support vector machine formulation constructed a linear separating surface (3) for our classification problem. We now describe how to construct a nonlinear separating surface which is implicitly defined by a kernel function. We briefly describe now how the generalized support vector machine (GSVM) [21] generates a nonlinear separating surface by using a completely arbitrary kernel. The GSVM solves the following mathematical program for a general kernel $K(A, A')$:

$$\begin{array}{rrcl} \displaystyle\min_{u,\gamma,y} & \nu e'y + f(u) & & \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{22}$$

Here $f(u)$ is some convex function on R^m which suppresses the parameter u and ν is some positive number that weights the classification error $e'y$ versus the suppression of u. A solution of this mathematical program for u and γ leads to the nonlinear separating surface

$$K(x', A')Du = \gamma \tag{23}$$

The linear formulation (1) of Section 2 is obtained if we let $K(A, A') = AA'$, $w = A'Du$ and $f(u) = \frac{1}{2}u'DAA'Du$. We now use a different classification objective which not only suppresses the parameter u but also suppresses γ in our nonlinear formulation:

$$\begin{array}{rrcl} \displaystyle\min_{u,\gamma,y} & \multicolumn{3}{l}{\frac{\nu}{2}y'y + \frac{1}{2}(u'u + \gamma^2)} \\ \text{s.t.} & D(K(A, A')Du - e\gamma) + y & \geq & e \\ & y & \geq & 0. \end{array} \tag{24}$$

We repeat the same arguments as in Section 2 to obtain the SSVM with a nonlinear kernel $K(A, A')$:

$$\min_{u,\gamma} \quad \frac{\nu}{2}\|p(e - D(K(A, A')Du - e\gamma), \alpha)\|_2^2 + \frac{1}{2}(u'u + \gamma^2), \tag{25}$$

where $K(A, A')$ is a kernel map from $R^{m\times n} \times R^{n\times m}$ to $R^{m\times m}$. We note that this problem, which is capable of generating highly nonlinear separating surfaces, still retains the strong convexity and differentiability properties for any arbitrary kernel. All of the results of the previous sections still hold. Hence we can apply the Newton-Armijo Algorithm 3.1 directly to solve (25).

We turn our attention now to numerical testing.

5 Numerical Results and Comparisons

We demonstrate now the effectiveness and speed of the smooth support vector machine (SSVM) approach by comparing it numerically with other methods. All parameters in the SSVM algorithm were chosen for optimal performance on a tuning set, a surrogate for a testing set. For a linear kernel, we divided the comparisons into two parts, moderate sized problems and larger problems. We also tested the ability of SSVM in generating a highly nonlinear separating surface. Although we established global and quadratic convergence of our Newton algorithm under an Armijo step-size rule, in practice the Armijo stepsize was turned off in all numerical tests without any noticeable change. Furthermore, we used the limit values of the sigmoid function $\frac{1}{1+\varepsilon^{-\alpha x}}$ and the p function (8) as the smoothing parameter α goes to infinity, that is the unit step function with value $\frac{1}{2}$ at zero and the plus function $(\cdot)_+$ respectively, when we computed the Hessian matrix and the gradient in (19). This gave slightly faster computational times but made no difference in the computational results from those obtained with $\alpha = 5$. All our experiments were run on the University of Wisconsin Computer Sciences Department Ironsides cluster. This cluster of four Sun Enterprise E6000 machines, each machine consists of 16 UltraSPARC II 250 MHz processors and 2 gigabytes of RAM, resulting in a total of 64 processors and 8 gigabytes of RAM. All linear and quadratic programming formulations were solved using the CPLEX package [14] called from within MATLAB [25].

In order to evaluate how well each algorithm generalizes to future data, we performed tenfold cross-validation on each dataset [28]. To evaluate the efficacy of SSVM, we compared computational times of SSVM with robust linear program (RLP) algorithm [2], the feature selection concave minimization (FSV) algorithm, the support vector machine using the 1-norm approach ($\mathrm{SVM}_{\|\cdot\|_1}$) and the classical support vector machine ($\mathrm{SVM}_{\|\cdot\|_2^2}$) [4, 30, 12]. We ran all tests on six publicly available datasets: the Wisconsin Prognostic Breast Cancer Database [24] and four datasets from the Irvine Machine Learning Database Repository [26]. It turned out that tenfold testing correctness of the SSVM is the highest for these five methods on all datasets tested as well as the computational speed. We summarize all these results in Table 1.

We also tested SSVM on the Irvine Machine Learning Database Repository [26] Adult dataset to demonstrate the capability of SSVM in solving larger problems. Training set sizes varied from 1,605 to 32,562 training examples and each example included 123 binary attributes. We compared the results with RLP directly and SOR [23], SMO [27] and SVM^{light} [17] indirectly. The results show that SSVM not only has a very good testing set accuracy but also has low computational times. The results for SOR, SMO and SVM^{light} are quoted from [23]. We summarize these details in Table 2 and Figure 2. We also solved the Adult dataset with 1,605 training examples via a classical support vector machine approach. It took 1,695.4 seconds to solve the problem using the CPLEX quadratic programming solver which employs a barrier function method [14]. In contrast, SSVM took 1.9 seconds only to solve the same problem as indicated in Table 2. Figure 2 indicates that computational time grows almost linearly for SSVM whereas it grows at a faster rate for SOR and SMO.

To test the effectiveness of the SSVM in generating a highly nonlinear separating surface, we tested it on the checkerboard dataset of [18] depicted in Figure 3. We used the following symmetric sixth degree polynomial kernel introduced in [23] as well a Gaussian kernel in the SSVM formulation (25):

Polynomial Kernel : $((\frac{A}{\lambda} - \rho)(\frac{A}{\lambda} - \rho)' - \mu)^d_\bullet$

Gaussian Kernel : $\varepsilon^{-\mu\|A_i - A_j\|_2^2}, \; i, j = 1, 2, 3 \ldots m.$

Values of the parameters λ, ρ, μ used are given in Figures 4 and 5 as well as that of the parameter ν of the nonlinear SSVM (25). The results are shown in Figures 4 and 5 . We note that the boundaries of the checkerboard are as sharp as those of [23], obtained by a linear programming solution, and considerably sharper than those of [18], obtained by a Newton approach applied to a quadratic programming formulation.

6 Conclusion and Future Work

We have proposed a new formulation, SSVM, which is a smooth unconstrained optimization reformulation of the traditional quadratic program associated with a SVM. SSVM is solved by a very fast Newton-Armijo algorithm and has been extended to nonlinear separation surfaces by using nonlinear kernel techniques. The numerical results show that SSVM is faster than other methods and has better generalization ability. Future work includes other smooth formulations, feature selection via SSVM and smooth support vector regression. We also plan to use the row and column generation chunking algorithms of [5, 22] in conjunction with SSVM to solve extremely large classification problems which do not fit in memory, for both linear and nonlinear kernels.

Acknowledgements

The research described in this Data Mining Institute Report 99-03, September 1999, was supported by National Science Foundation Grants CCR-9729842 and CDA-9623632, by Air Force Office of Scientific Research Grant F49620-97-1-0326 and by the Microsoft Corporation.

Ten-Fold Training Correctness, % Ten-Fold Testing Correctness, % Ten-Fold Computational Time, *sec.*					
Dataset size	Method				
$m \times n$	SSVM	RLP	$\text{SVM}_{\|\cdot\|_1}$	$\text{SVM}_{\|\cdot\|_2^2}$	FSV
WPBC(24 months) 155×32	86.16 **83.47** **2.32**	85.23 67.12 6.47	74.40 71.08 6.25	81.94 82.02 12.50	88.89 81.93 22.54
WPBC(60 months) 110×32	80.20 **68.18** **1.03**	87.58 63.50 2.81	71.21 66.23 3.72	80.91 61.83 4.91	86.36 64.55 7.31
Ionosphere 351×34	94.12 **89.63** **3.69**	94.78 86.04 10.38	88.92 86.10 42.41	92.96 89.17 128.15	94.87 86.76 10.49
Cleveland 297×13	87.32 **86.13** **1.63**	86.31 83.87 12.74	85.30 84.55 18.71	72.05 72.12 67.55	87.95 85.31 21.15
Pima Indians 768×8	78.11 **78.12** **1.54**	76.48 76.16 195.21	75.52 74.47 286.59	77.92 77.07 1138.0	77.91 76.96 227.41
BUPA Liver 345×6	70.37 **70.33** **1.05**	68.98 64.34 17.26	67.83 64.03 19.94	70.57 69.86 123.24	71.21 69.81 25.10

Table 1: **Ten-fold cross-validation correctness results on six moderate sized datasets using five different methods. All linear and quadratic programming formulations were solved using the CPLEX package [14] called from within MATLAB [25]. Bold type indicates the best result.**

Dataset size (Training, Testing) n = no. of feature	Testing Correctness % Running Time *Sec*				
	Method				
	SSVM	SOR	SMO	SVM^{light}	RLP
$(1605, 30957)$	**84.27**	84.06	-	84.25	78.68
$n = 123$	1.9	**0.3**	0.4	5.4	9.9
$(2265, 30297)$	**84.57**	84.24	-	84.43	77.19
$n = 123$	2.8	1.2	**0.9**	10.8	19.12
$(3185, 29377)$	**84.63**	84.23	-	84.40	77.83
$n = 123$	3.9	**1.4**	1.8	21.0	80.1
$(4781, 27781)$	**84.55**	84.28	-	84.47	79.15
$n = 123$	6.0	**1.6**	3.6	43.2	88.6
$(6414, 26148)$	**84.60**	84.30	-	84.43	71.85
$n = 123$	8.1	**4.1**	5.5	87.6	218.8
$(11221, 21341)$	**84.79**	84.37	-	84.68	60.00
$n = 123$	**14.1**	18.8	17.0	306.6	449.2
$(16101, 16461)$	**84.96**	84.62	-	84.83	72.52
$n = 123$	**21.5**	24.8	35.3	667.2	632.6
$(22697, 9865)$	**85.35**	85.06	-	85.17	77.43
$n = 123$	**29.0**	31.3	85.7	1425.6	991.9
$(32562, 16282)$	85.02	84.96	-	**85.05**	83.25
$n = 123$	**44.5**	83.9	163.6	2184.0	1561.1

Table 2: **Testing set correctness results on the larger Adult dataset obtained by five different methods. The SOR, SMO and SVM^{light} are from [23]. The SMO experiments were run on a 266 MHz Pentium II processor under Windows NT 4 and using Microsoft's Visual C++ 5.0 compiler. The SOR experiments were run on a 200 MHz Pentium Pro with 64 megabytes of RAM, also under Windows NT 4 and using VisualC++ 5.0. The SVM^{light} experiments were run on the same hardware as that for SOR, but under the Solaris 5.6 operating system. Bold type indicates the best result and a dash (-) indicates that the results were not available from [23].**

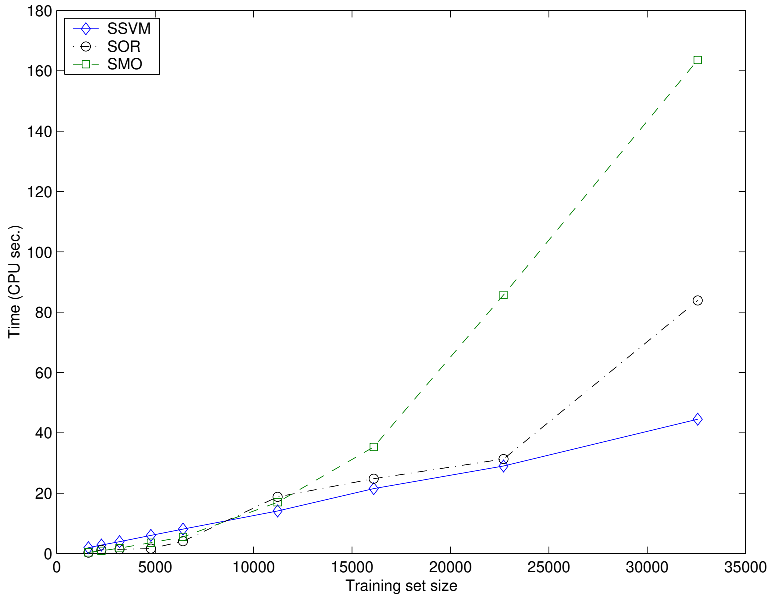

Figure 2: **Comparison of SSVM, SOR and SMO on the larger Adult dataset. Note the essentially linear time growth of SSVM.**

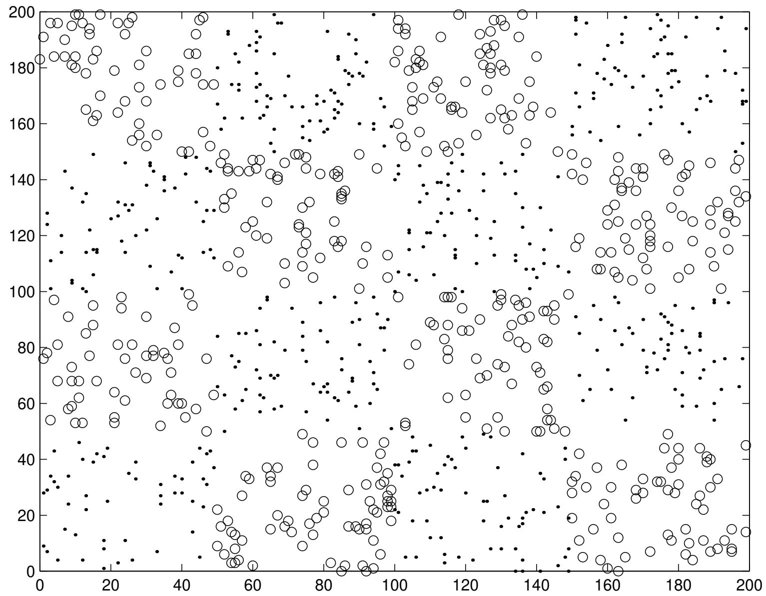

Figure 3: **The checkerboard training dataset**

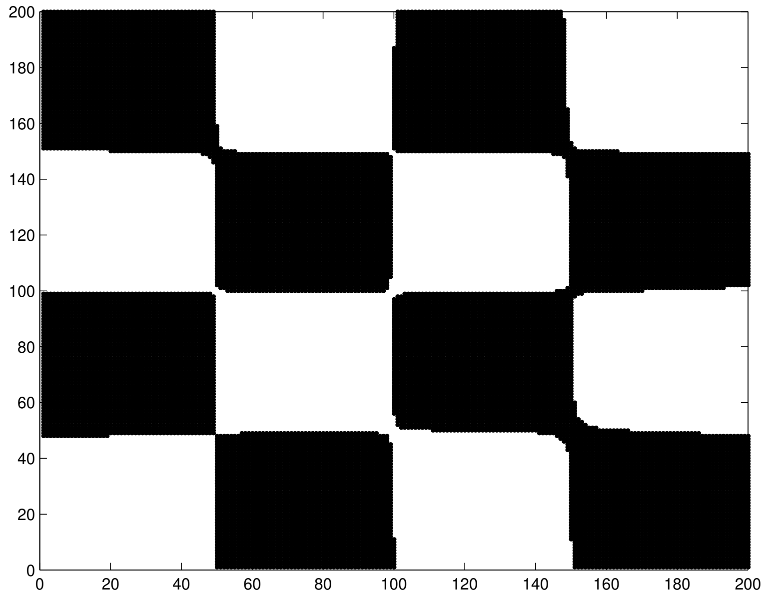

Figure 4: **Indefinite sixth degree polynomial kernel separation of the checkerboard dataset** $(\nu = 320,000, \lambda = 100, \rho = 1, d = 6, \mu = 0.5)$

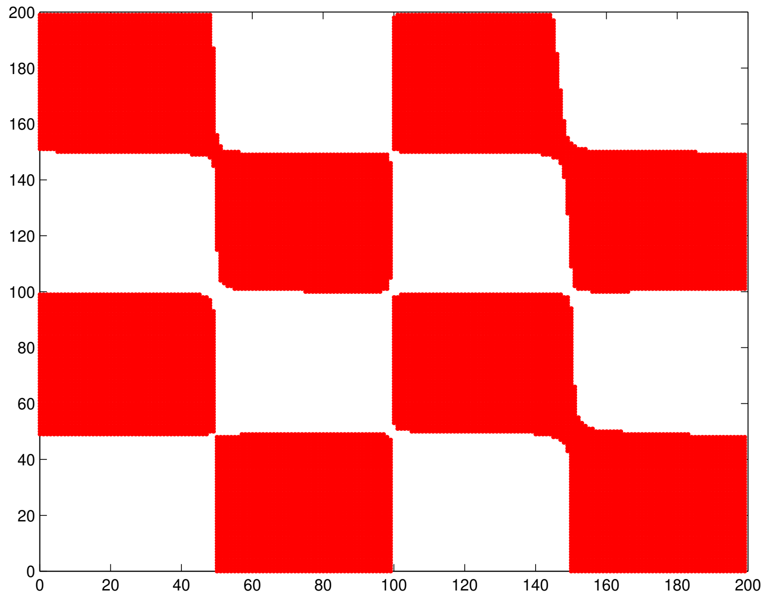

Figure 5: **Gaussian kernel separation of he checkerboard dataset** $(\nu = 10, \mu = 2)$

Appendix

We establish here global quadratic convergence for a Newton-Armijo Algorithm for solving the unconstrained minimization problem:

$$\min_{x \in R^n} f(x), \tag{26}$$

where f is a twice differentiable real valued function on R^n satisfying conditions (i) and (ii) below. This result, which can also be deduced from [15, Theorem 6.3.4, pp 123-125] is given here for completeness.

Newton-Armijo Algorithm for Unconstrained Minimization Let f be a twice differentiable real valued function on R^n. Start with any $x^0 \in R^n$. Having x^i, stop if the gradient of f at x^i is zero, that is $\nabla f(x^i) = 0$. Else compute x^{i+1} as follows:

(i) **Newton Direction**: Determine direction $d^i \in R^n$ by setting equal to zero, the linearization of $\nabla f(x)$ around x^i which gives n linear equations in n variables:

$$\nabla^2 f(x^i) d^i = -\nabla f(x^i)'. \tag{27}$$

(ii) **Armijo Stepsize** [1]: Choose a stepsize $\lambda_i \in R$ such that:

$$x^{i+1} = x^i + \lambda_i d^i \tag{28}$$

where $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ such that :

$$f(x^i) - f(x^i + \lambda_i d^i) \geq -\delta \lambda_i \nabla f(x^i) d^i \tag{29}$$

where $\delta \in (0, \frac{1}{2})$.

Global Quadratic Convergence Theorem Let f be a twice differentiable real valued function on R^n and such that :

(i) $\|\nabla^2 f(y) - \nabla^2 f(x)\|_2 \leq \kappa \|y - x\|_2$, for all $x, y \in R^n$ and some $\kappa > 0$,

(ii) $y \nabla^2 f(x) y \geq \mu \|y\|_2^2$, for all $x, y \in R^n$ and some $\mu > 0$.

Let $\{x^i\}$ be a sequence generated by the Newton-Armijo Algorithm. Then:

(a) The sequence $\{x^i\}$ converges to $\bar{x}$ such that $\nabla f(\bar{x}) = 0$, where $\bar{x}$ is the unique global minimizer of f on R^n.

(b) $\|x^{i+1}-\bar{x}\|_2 \leq \frac{\kappa}{2\mu}\|x^i-\bar{x}\|_2^2$, for $i \geq \bar{i}$ for some $\bar{i}$.

Proof (a) Since

$$f(x^i) - f(x^{i+1}) \geq \delta\lambda_i \nabla f(x^i)\nabla^2 f(x^i)^{-1}\nabla f(x^i)' \geq 0, \tag{30}$$

the sequence $\{x^i\}$ is contained in the compact level set:

$$L_{f(x^0)}(f(x)) \subseteq \{x|\ \|x-x^0\|_2 \leq \frac{2}{\mu}\|\nabla f(x^0)\|_2\}. \tag{31}$$

Hence

$$y\nabla^2 f(x^i)y \leq \nu\|y\|_2^2,\ \text{for all } i=0,1,2,\ldots,\ y\in R^n,\ \text{for some } \nu>0. \tag{32}$$

It follows that

$$-\nabla f(x^i)d^i = \nabla f(x^i)\nabla^2 f(x^i)^{-1}\nabla f(x^i)' \geq \frac{1}{\nu}\|\nabla f(x^i)\|^2, \tag{33}$$

and by Theorem 2.1, Example 2.2 (iv) of [20] we have that each accumulation point of $\{x^i\}$ is stationary. Since all accumulation points of $\{x^i\}$ are identical to the global unique minimizer of the strongly convex function f, it follows that $\{x^i\}$ converges to $\bar{x}$ such that $\nabla f(\bar{x})=0$ and $\bar{x}$ is the unique global minimizer of f on R^n.

(b) We first show that the Armijo inequality holds from a certain i_0 onward with $\lambda_i = 1$. Suppose not, then there exists a subsequence $\{x^{i_j}\}$ such that

$$f(x^{i_j}) - f(x^{i_j}+d^{i_j}) < -\delta\nabla f(x^{i_j})d^{i_j},\ d^{i_j} = -\nabla^2 f(x^{i_j})^{-1}\nabla f(x^{i_j})'. \tag{34}$$

By the twice differentiability of f:

$$f(x^{i_j}+d^{i_j}) - f(x^{i_j}) = \nabla f(x^{i_j})d^{i_j} + \frac{1}{2}d^{i_j\prime}\nabla^2 f(x^{i_j}+t^{i_j}d^{i_j})d^{i_j}, \tag{35}$$

for some $t^{i_j}\in(0,1)$. Hence

$$-\nabla f(x^{i_j})d^{i_j} - \frac{1}{2}d^{i_j\prime}\nabla^2 f(x^{i_j}+t^{i_j}d^{i_j})d^{i_j} < -\delta\nabla f(x^{i_j})d^{i_j}. \tag{36}$$

Collecting terms and substitution for d^{i_j} gives:

$$\begin{aligned} -\tfrac{1}{2}\nabla f(x^{i_j})\nabla^2 f(x^{i_j})^{-1}\nabla^2 f(x^{i_j}+t^{i_j}d^{i_j})\nabla^2 f(x^{i_j})^{-1}\nabla f(x^{i_j})' \\ < -(1-\delta)\nabla f(x^{i_j})\nabla^2 f(x^{i_j})^{-1}\nabla f(x^{i_j})'. \end{aligned} \tag{37}$$

Dividing by $\|\nabla f(x^{i_j})\|_2^2$ and letting j go to infinity gives:

$$-\frac{1}{2}\bar{q}\nabla^2 f(\bar{x})^{-1}\bar{q}' \leq -(1-\delta)\bar{q}\nabla^2 f(\bar{x})^{-1}\bar{q}',$$

or

$$(\frac{1}{2}-\delta)\bar{q}\nabla^2 f(\bar{x})^{-1}\bar{q}' \leq 0, \tag{38}$$

where $\bar{q} = \lim_{j\to\infty} \frac{\nabla f(x^{i_j})}{\|\nabla f(x^{i_j})\|_2}$.

This contradicts the positive definiteness of $\nabla^2 f(\bar{x})$ for $\delta \in (0, \frac{1}{2})$. Hence for $i \geq \bar{i}$ for some $\bar{i}$, we have a pure Newton iteration, that is

$$x^{i+1} = x^i - \nabla^2 f(x^i)^{-1}\nabla f(x^i)',\ i \geq \bar{i}. \tag{39}$$

By part (a) above $\{x^i\}$ converges to $\bar{x}$, such that $\nabla f(\bar{x}) = 0$. We now establish the quadratic rate given by the inequality in (b). Since for $i \geq \bar{i}$:

$$\begin{array}{l} x^{i+1} - \bar{x} = x^i - \bar{x} - \nabla^2 f(x^i)^{-1}\left[\nabla f(\bar{x} + t(x^i - \bar{x}))\right]_{t=0}^{t=1} \\ = (x^i - \bar{x}) - \nabla^2 f(x^i)^{-1}\int_0^1 (\nabla^2 f(\bar{x} + t(x^i - \bar{x}) - \nabla^2 f(x^i) + \nabla^2 f(x^i))(x^i - \bar{x})dt, \end{array} \tag{40}$$

we have, upon cancelling the terms $(x^i - \bar{x})$ inside and outside the integral sign and taking norms, that :

$$\begin{array}{rcl} \|x^{i+1} - \bar{x}\|_2 \leq & \|\nabla^2 f(x^i)^{-1}\|_2 \int_0^1 \|\nabla^2 f(\bar{x} + t(x^i - \bar{x}) - \nabla^2 f(x^i)\|_2 \|x^i - \bar{x}\|_2 dt \\ \leq & \|\nabla^2 f(x^i)^{-1}\|_2 \|x^i - \bar{x}\|_2 \int_0^1 \kappa(1-t)\|x^i - \bar{x}\|_2 dt \\ = & \frac{\kappa}{2\mu}\|x^i - \bar{x}\|_2^2. \end{array} \tag{41}$$

□

References

[1] L. Armijo. Minimization of functions having Lipschitz-continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

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

[3] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, second edition, 1999.

[4] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In J. Shavlik, editor, *Machine Learning Proceedings of the Fifteenth International Conference(ICML '98)*, pages 82–90, San Francisco, California, 1998. Morgan Kaufmann. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[5] P. S. Bradley and O. L. Mangasarian. Massive data discrimination via linear support vector machines. *Optimization Methods and Software*, 13:1–10, 2000. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-03.ps.

[6] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[7] B. Chen and P. T. Harker. Smooth approximations to nonlinear complementarity problems. *SIAM Journal of Optimization*, 7:403–420, 1997.

[8] Chunhui Chen and O. L. Mangasarian. Smoothing methods for convex inequalities and linear complementarity problems. *Mathematical Programming*, 71(1):51–69, 1995.

[9] Chunhui Chen and O. L. Mangasarian. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, 5(2):97–138, 1996.

[10] X. Chen, L. Qi, and D. Sun. Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. *Mathematics of Computation*, 67:519–540, 1998.

[11] X. Chen and Y. Ye. On homotopy-smoothing methods for variational inequalities. *SIAM Journal on Control and Optimization*, 37:589–616, 1999.

[12] V. Cherkassky and F. Mulier. *Learning from Data - Concepts, Theory and Methods.* John Wiley & Sons, New York, 1998.

[13] P. W. Christensen and J.-S. Pang. Frictional contact algorithms based on semismooth newton methods. In *Reformulation: Nonsmooth, Piecewise*

Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, (editors), pages 81–116, Dordrecht, Netherlands, 1999. Kluwer Academic Publishers.

[14] CPLEX Optimization Inc., Incline Village, Nevada. *Using the CPLEX(TM) Linear Optimizer and CPLEX(TM) Mixed Integer Optimizer (Version 2.0)*, 1992.

[15] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, N.J., 1983.

[16] M. Fukushima and L. Qi. *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[17] T. Joachims. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, 1999.

[18] L. Kaufman. Solving the quadratic programming problem arising in support vector classification. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 147–167. MIT Press, 1999.

[19] O. L. Mangasarian. Mathematical programming in neural networks. *ORSA Journal on Computing*, 5(4):349–360, 1993.

[20] O. L. Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization*, 33(6):1916–1925, 1995. ftp://ftp.cs.wisc.edu/tech-reports/reports/93/tr1145.ps.

[21] O. L. Mangasarian. Generalized support vector machines. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 135–146, Cambridge, MA, 2000. MIT Press. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps.

[22] O. L. Mangasarian and David R. Musicant. Massive support vector regression. Technical Report 99-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, July

1999. To appear in: "Applications and Algorithms of Complementarity", M. C. Ferris, O. L. Mangasarian and J.-S. Pang, editors, Kluwer Academic Publishers,Boston 2000. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-02.ps.

[23] O. L. Mangasarian and David R. Musicant. Successive overrelaxation for support vector machines. *IEEE Transactions on Neural Networks*, 10:1032–1037, 1999. ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-18.ps.

[24] O. L. Mangasarian, W. N. Street, and W. H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, July-August 1995.

[25] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1992.

[26] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases, 1992. www.ics.uci.edu/∼mlearn/MLRepository.html.

[27] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press, 1999. http://www.research.microsoft.com/∼jplatt/smo.html.

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

[29] P. Tseng. Analysis of a non-interior continuation method based on chen-mangasarian smoothing functions for complementarity problems. In *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods, M. Fukushima and L. Qi, (editors)*, pages 381–404, Dordrecht, Netherlands, 1999. Kluwer Academic Publishers.

[30] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.