Math. Program., Ser. B 101: 185–204 (2004)

Digital Object Identifier (DOI) 10.1007/s10107-004-0541-8

Michael C. Ferris · Todd S. Munson

Semismooth support vector machines*

Received: November 29, 2000 / Accepted: December 8, 2003
Published online: 21 July 2004 – © Springer-Verlag 2004

Abstract. Support vector machines can be posed as quadratic programming problems in a variety of ways. This paper investigates a formulation using the two-norm for the misclassification error that leads to a positive definite quadratic program with a single equality constraint under a duality construction. The quadratic term is a small rank update to a diagonal matrix with positive entries. The optimality conditions of the quadratic program are reformulated as a semismooth system of equations using the Fischer-Burmeister function and a damped Newton method is applied to solve the resulting problem. The algorithm is shown to converge from any starting point with a Q-quadratic rate of convergence. At each iteration, the Sherman-Morrison-Woodbury update formula is used to solve the key linear system. Results for a large problem with 60 million observations are presented demonstrating the scalability of the proposed method on a personal computer. Significant computational savings are realized as the inactive variables are identified and exploited during the solution process. Further results on a small problem separated by a nonlinear surface are given showing the gains in performance that can be made from restarting the algorithm as the data evolves.

1. Introduction

The support vector machine is used to construct a (linear or nonlinear) surface that partitions measurements taken from representative subsets of known populations. The surface is then used to assign unknown observations to the populations, where the accuracy of the assignment is determined by cross-validation statistics [33]. Since the classifications for the input data are given, this technique is an example of a supervised learning process from the machine learning community. In this paper, only the two-population case is considered. An example application is when the two populations represent malignant and benign tumors [7, 22, 23], where historical data is used to define a surface that can later be used to classify a tumor found in a new patient. Several models for the calculation of an "optimal" partitioning surface exist. In this paper, a soft-margin support vector machine is used that leads to a strongly convex quadratic program with simple bounds on the variables and a single equality constraint.

The main goal of this paper is to present an algorithm for solving the resulting optimization problem that converges from any starting point and, near a solution, has a

M.C. Ferris: Computer Sciences Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, USA. e-mail: ferris@cs.wisc.edu

T.S. Munson: Mathematics and Computer Science Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA. e-mail: tmunson@mcs.anl.gov

* This work partially supported by NSF grant number CCR-9972372; AFOSR grant number F49620-01-1-0040; the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38; and Microsoft Corporation.

quadratic rate of convergence. For this purpose, a semismooth method [5] for processing the first-order optimality conditions for the problem is prescribed. The complementarity reformulation of the problem is equivalent to the original quadratic program and involves just one more variable (the multiplier on the equality constraint). All elements of the generalized Jacobian are shown to be nonsingular, and the sequence produced by the algorithm contains an accumulation point. Standard theory can then be applied to show that the algorithm actually converges to a solution at a Q-quadratic rate.

Other algorithms have been proposed for solving this class of problem, including an interior-point method in [8, 10] and an active-set method in [21]. For related formulations [8] there are many different techniques, see for example [2, 4, 14, 20, 28, 32, 35] and the references contained therein. The proposed semismooth algorithm implicitly combines ideas from both of these methods. It solves linear systems of equations of similar form to the interior-point method and therefore reuses the extensive computational technology that has been developed in that field. Also, an "active" set implicitly defined by the algorithm can be exploited in the linear algebra calculations, leading to substantial computational savings. Further, the amount of storage required for the semismooth method is smaller than that of the interior-point method.

Another key property of the semismooth method not shared by the interior-point implementations is the ability to restart from any point. Thus, as new data becomes available, the proposed method can update the current solution, as opposed to solving the problem from scratch. This algorithm feature is important when dealing with datasets that evolve.

Apart from these properties, the implementation of the method scales well to large sample populations. The target sample population size is between 1 million and 60 million observations. Note that 60 million observations corresponds to the current population of Britain, a random sampling of 20% of the current population of the United States, or 1% of the current population of the world. Therefore, the 60 million observation problem is a reasonable test problem. To achieve scalability, the Sherman-Morrison-Woodbury update formula is used to calculate the direction, and asynchronous I/O is used to retrieve the observation data from disk. The resulting code uses a small number of vectors with memory and disk requirements suitable for a personal computer.

The paper is organized as follows. Section 2 derives the support vector machine formulation used for the subsequent analysis and testing. The problem is posed as a mixed complementarity problem. Specifically, let $\mathcal{L}$ and $\mathcal{E}$ be a partition of the indices $\{1, 2, \ldots, n\}$, implicitly corresponding to lower bounded and free variables, and let $F : \Re^n \to \Re^n$ be a given function. Let $m = \mathbf{card}\,(\mathcal{L})$ and $c = \mathbf{card}\,(\mathcal{E})$. The mixed complementarity problem is to find an $z^*_{\mathcal{L}} \in \Re^m$ and $z^*_{\mathcal{E}} \in \Re^c$ such that

$$\begin{array}{c} 0 \le F_{\mathcal{L}}(z_{\mathcal{L}}, z_{\mathcal{E}}) \;\perp\; z_{\mathcal{L}} \ge 0 \\ F_{\mathcal{E}}(z_{\mathcal{L}}, z_{\mathcal{E}}) = 0, \end{array} \tag{1}$$

where $\perp$ is defined componentwise as $0 \le a \;\perp\; b \ge 0$ if and only if $a \ge 0$, $b \ge 0$, and $ab = 0$. This problem is the standard nonlinear complementarity problem when $c = 0$ and a square system of nonlinear equations when $m = 0$. See [9] for definitions of general mixed complementarity problems and applications. An existing procedure for constructing nonlinear separating surfaces is also outlined. This procedure results

in quadratic optimization problems of the form that the proposed method can solve efficiently.

Section 3 details a damped Newton method for semismooth equations [5, 25] for solving such complementarity problems. The method uses the Fischer-Burmeister function [11] to reformulate the complementarity conditions as a system of semismooth equations. The basic theory for these methods is given, with appropriate citations to the literature. A proof that the method converges from any starting point when applied to the support vector machine formulation is provided. In particular, the Newton direction at any arbitrary point is calculated by using applications of the Sherman-Morrison-Woodbury update formula [27].

Section 4 discusses an implementation of the method using out-of-core computations. Results for a large test problem containing 60 million points are presented and compared with the interior-point method results given in [8]. Representative results for constructing a nonlinear separating surface are given, along with details on restarting the method from a given solution when the number of observations in the data changes.

2. Support vector machines

The linear support vector machine attempts to separate two finite point sets with a hyperplane such that the separation margin is maximized. Consider two populations $\mathcal{P}_+$ and $\mathcal{P}_-$ that have been sampled, and let $P_+ \subseteq \mathcal{P}_+$ and $P_- \subseteq \mathcal{P}_-$ denote finite sample sets with $P \equiv P_+ \cup P_-$ representing the entire set of sampled elements. Let $m = \mathbf{card}\,(P)$ denote the size of the total population. Associated with each $p \in P$ is a vector $a(p) \in \Re^f$ that measures f features for the particular element. Furthermore, let $A(P) \in \Re^{m \times f}$ denote the matrix formed by the measured observations for each $p \in P$, and define $A_+ := A(P_+)$ and $A_- := A(P_-)$.

First, assume that the two point sets are disjoint, that is, the intersection of their convex hulls (denoted here by co) is empty:

$$(co\ \cup_{p\in P_+} a(p)) \cap (co\ \cup_{p\in P_-} a(p)) = \emptyset.$$

In this case, one can select $w \in \Re^f$ and $\gamma \in \Re$ such that $A_+ w > \gamma$ and $A_- w < \gamma$. The hyperplane $\{a \in \Re^f \mid a^T w = \gamma\}$ strictly separates the two point sets; and the separation margin [2, 32, 35], the minimum distance from the hyperplane to the convex hulls of the point sets, is $\frac{2}{\|w\|_2^2}$. Therefore, an optimization problem to maximize the separation margin would be

$$\begin{array}{ll} \max_{w,\gamma} & \frac{2}{\|w\|_2^2} \\ \text{subject to} & A_+ w > \gamma \\ & A_- w < \gamma. \end{array}$$

However, maximizing $\frac{2}{\|w\|_2^2}$ is the same as minimizing $\frac{1}{2}\|w\|_2^2$, and the strict inequalities can be removed by normalizing the system [18]. Therefore, the following convex quadratic optimization problem is obtained:

$$\begin{array}{ll} \min_{w,\gamma} & \frac{1}{2}\|w\|_2^2 \\ \text{subject to} & A_+ w - \gamma e \geq e \\ & A_- w - \gamma e \leq -e, \end{array} \tag{2}$$

where e is a vector of all ones of appropriate dimension. The constraints can be more succinctly written by defining an "indicator" function, $d(p)$, as follows:

$$d(p) := \begin{cases} 1 \text{ if } p \in P_+ \\ -1 \text{ if } p \in P_- \end{cases}$$

with D denoting the diagonal matrix formed from $d(p)$ for all $p \in P$. Then, the constraints of (2) can be rewritten simply as

$$D(Aw - \gamma e) \geq e.$$

Unfortunately, the underlying assumption above that the two point sets are disjoint is typically not satisfied, and (2) is infeasible. In this case, a surface is constructed that minimizes the error in satisfying the inequalities, termed the misclassification error in the machine learning community [16]. The resulting optimization problem in this case becomes

$$\begin{array}{c} \min_{w,\gamma,y} \quad \frac{1}{2}\|y\|_2^2 \\ D(Aw - \gamma e) + y \geq e, \ \ y \geq 0, \end{array} \tag{3}$$

where the two norm of the misclassification error is minimized. Other norms can be used for the misclassification error, which lead to other problem formulations. When the two norm is used the constraint $y \geq 0$ is unnecessary and is therefore dropped from the subsequent discussion.

The two problems, (2) and (3), are combined by introducing a parameter $\nu > 0$ that weights the two competing goals, maximizing the separation margin and minimizing the misclassification error. The resulting optimization problem, termed a soft-margin support vector machine, is

$$\begin{array}{c} \min_{w,\gamma,y} \quad \frac{1}{2}\|w\|_2^2 + \frac{\nu}{2}\|y\|_2^2 \\ D(Aw - \gamma e) + y \geq e, \end{array} \tag{4}$$

which is a convex quadratic program that is feasible with the objective bounded below by zero. Hence, (4) has a solution, (w^*, γ^*, y^*). The support vectors are the points where $D(Aw^* - \gamma^* e) \leq e$, that is, the misclassified points and the points on the bounding hyperplanes.

The Wolfe dual [15] of (4) is the strongly convex quadratic program

$$\begin{array}{c} \min_x \ \frac{1}{2\nu} x^T x + \frac{1}{2} x^T DAA^T Dx - e^T x \\ e^T Dx = 0, \ \ x \geq 0, \end{array} \tag{5}$$

which has a unique solution, x^*. The quadratic term consists of a rank-f update to a positive definite matrix, an observation that will be exploited in the algorithm development and resultant linear algebra calculations. The solution (w, γ, y) of (4) is recovered from a solution x of (5) by setting $w = A^T Dx$, γ to the optimal multiplier of $e^T Dx = 0$, and $y = \max(e - D(Aw - \gamma e), 0)$.

The final step in the problem derivation is to write first-order necessary and sufficient optimality conditions for (5), which form the mixed linear complementarity problem:

$$\begin{array}{c} 0 \leq \left(\frac{1}{\nu} I + DAA^T D\right) x - De\gamma - e \ \perp \ x \geq 0 \\ e^T Dx = 0. \end{array} \tag{6}$$

Following the notation of the introduction, we have that $n = m + 1$ and

$$z = \begin{bmatrix} x \\ \gamma \end{bmatrix}, \quad F(z) = \begin{bmatrix} \frac{1}{\nu} I + DAA^T D & -De \\ e^T D & 0 \end{bmatrix} z + \begin{bmatrix} -e \\ 0 \end{bmatrix}.$$

The solution to this linear complementarity problem is unique, as proven in the following theorem.

Theorem 1. *Let* **card** $(P_+) > 0$ *and* **card** $(P_-) > 0$. *The mixed complementarity problem (6) has a unique solution.*

Proof. Since (5) is a strongly convex quadratic program that is feasible and bounded below, it has a unique solution. Let x^* denote this solution. Furthermore, there must exist a γ^* such that (x^*, γ^*) is a solution to (6). The remainder of this proof shows that γ^* is unique.

Assume that $x^* = 0$. Therefore, any γ^* solving (6) must satisfy $-De\gamma^* - e \geq 0$. Recalling the definition of D and using the fact that **card** $(P_+) > 0$ and **card** $(P_-) > 0$ by assumption, the contradiction that $\gamma^* \leq -1$ and $\gamma^* \geq 1$ is obtained. Therefore, $x^* \neq 0$.

Since x^* solves (5) and $x^* \neq 0$, there must be an i such that $x_i^* > 0$ because x^* is feasible for (5). Therefore, any γ^* solving (6) must satisfy

$$\left[\left(\frac{1}{\nu} I + DAA^T D \right) x^* - De\gamma^* - e \right]_i = 0.$$

Hence, γ^* is uniquely determined by this equation, and the proof is complete. □

The solution of the complementarity problem leads directly to a discriminant function $f(a) = a^T w - \gamma = a^T A^T Dx - \gamma$. Given a new data point a, the sign of $f(a)$ determines whether the new point is assigned to $\mathcal{P}_+$ or $\mathcal{P}_-$, respectively. Generalization ability measures how well such a discriminant function correctly classifies the new data. The reason for maximizing the separation margin is to improve the generalization ability [17] of the computed separating surface.

A linear surface is not always sufficient to separate the two populations. Instead, following Mangasarian [19], one can use a nonlinear discriminant induced by a kernel function K:

$$f(a) = K(a^T, A^T)Dx - \gamma.$$

The function K maps $\Re^{k \times f} \times \Re^{f \times m}$ into $\Re^{k \times m}$. The optimal values of x and γ can be found from a generalization of (6), namely,

$$\begin{gathered} 0 \leq \left(\tfrac{1}{\nu} I + DK(A, A^T)D \right) x - De\gamma - e \ \perp \ x \geq 0 \\ e^T Dx = 0. \end{gathered}$$

The resulting problem is structurally the same as the linear support vector problem and hence can be solved by using the proposed semismooth method. However, the matrix $K(A, A^T)$ is a dense $m \times m$ matrix for all popular choices of kernel function [19], and thus the computational efficiencies outlined in Section 4 are no longer applicable.

To overcome this, Fine and Scheinberg [10] suggest approximating $K(A, A^T)$ using an incomplete Cholesky factorization. Essentially, they compute a lower triangular matrix $G \in \Re^{m\times k}$ with $k \ll m$ such that

$$GG^T \approx K(A, A^T).$$

The columns of G are chosen by considering only diagonal elements of $K(A, A^T)$. In particular, the largest remaining diagonal element is selected as the pivot. This approximation is shown to produce similar generalization results to the full kernel implementation. The resulting complementarity problem is identical to (6) except that A is replaced by G. Fine and Scheinberg use an interior-point method to solve the complementarity problem that exploits this structure by using a product form Cholesky factorization. The proposed algorithm described in the next section uses a semismooth method combined with the Sherman-Morrison-Woodbury update formula to exploit the structure.

3. Algorithm

A semismooth method [5, 30] based on the Fischer-Burmeister merit function [11] is prescribed to solve the linear mixed complementarity problems defined by the necessary and sufficient first-order optimality conditions found in (6). Essentially, the semismooth method reformulates this complementarity problem as a system of nonlinear, nonsmooth equations and applies a generalized Newton method to find a solution. The basic semismooth method and convergence theory are first presented, followed by a discussion of the Fischer-Burmeister function and its properties. The method is then specialized for the support vector machine problem, and convergence is proven. The proofs show how to perform the linear algebra in the implementation.

3.1. Basic semismooth method

The class of semismooth functions [24, 29, 31] are a generalized notion of continuously differentiable functions that are both Lipschitzian and directionally differentiable. To define the class of semismooth functions precisely, we need the notion of the B-subdifferential and a generalized Jacobian. Let $G : \Re^n \to \Re^n$ be a Lipschitzian function and D_G denote the set of points where G is differentiable. This definition for D_G is appropriate because, by Rademacher's theorem, G is differentiable almost everywhere.

Before proceeding, some notation used in the sequel is described. If $F : \Re^n \to \Re^n$, its Jacobian at a point is denoted by $F'(z)$ and $\nabla F(z)$ denotes the transposed Jacobian. In particular,

$$\begin{aligned}\left[F'(z)\right]_{i,j} &:= \tfrac{\partial F_i(z)}{\partial z_j}\\ \left[\nabla F(z)\right]_{i,j} &:= \tfrac{\partial F_j(z)}{\partial z_i}.\end{aligned}$$

Furthermore, $F'(z; d)$ is the directional derivative of F at z in the direction d. The following definitions can then be made.

Definition 1 (*B*-subdifferential [31]). *The B-subdifferential of G at z is*

$$\partial_B G(z) := \left\{ H \mid \exists \{z^k\} \to z, z^k \in D_G, \text{ and } \lim_{\{z^k\} \to z} G'(z^k) = H \right\}.$$

Definition 2 (Generalized Jacobian [3]). *The Clarke generalized Jacobian of G at z is*

$$\partial G(z) := co\ \partial_B G(z),$$

where co denotes the convex hull.

Definition 3 (Semismooth). *Let $G : \Re^n \to \Re^n$ be locally Lipschitzian at $z \in \Re^n$. Then G is semismooth at z if*

$$\lim_{\substack{H \in \partial G(z+td') \\ d' \to d, t \downarrow 0}} Hd' \tag{7}$$

exists for all $d \in \Re^n$. In particular, G is directionally differentiable at z with $G'(z; d)$ given by the limit in (7). If, in addition, for any $d \to 0$ and any $H \in \partial G(z + d)$,

$$Hd - G'(z; d) = O\left(\|d\|^2\right),$$

then G is said to be strongly semismooth at z. Furthermore, G is a (strongly) semismooth function if G is (strongly) semismooth for all $z \in \Re^n$.

To solve the system of equations $G(z) = 0$, where $G : \Re^n \to \Re^n$ is a semismooth function, we use a damped Newton method [30]. To this end, the merit function $g : \Re^n \to \Re$ is defined as $g(z) := \frac{1}{2} \|G(z)\|_2^2$ and an assumption is made that g is continuously differentiable. The algorithm follows.

Algorithm 2 (Damped Newton Method for Semismooth Equations)

0. *(Initialization) Let $z^0 \in \Re^n$, $\rho > 0$, $p > 2$, and $\sigma \in \left(0, \frac{1}{2}\right)$ be given. Set $k = 0$.*
1. *(Termination) If $g(z^k) = 0$, stop.*
2. *(Direction Generation) Otherwise, let $H^k \in \partial_B G(z^k)$, and calculate $d^k \in \Re^n$ solving the Newton system:*

$$H^k d^k = -G(z^k). \tag{8}$$

If either (8) is unsolvable or the descent condition

$$\nabla g(z^k)^T d^k < -\rho \|d_k\|_2^p \tag{9}$$

is not satisfied, then set $d^k = -\nabla g(z^k)$.

3. *(Linesearch) Choose $t^k = 2^{-i_k}$, where i_k is the smallest integer such that*

$$g\left(z^k + 2^{-i_k} d^k\right) \leq g(z^k) + \sigma 2^{-i_k} \nabla g(z^k)^T d^k. \tag{10}$$

4. *(Update) Let $z^{k+1} := z^k + t^k d^k$ and $k := k + 1$. Go to 2.*

The following convergence theorem, whose proof can be found in [30, 5], then holds:

Theorem 3. *Let $G : \Re^n \to \Re^n$ be semismooth for all $z \in \Re^n$ and $g(z)$ be continuously differentiable. Let $\{z^k\}$ be a sequence generated by Algorithm 2. Then any accumulation point of $\{z^k\}$ is a stationary point for $g(z)$. Furthermore, if one of these accumulation points, say z^*, solves the system $G(z) = 0$ and all $H \in \partial_B G(z^*)$ are invertible, then the following hold:*

a. For all k sufficiently large, the Newton direction calculated in (8) exists and satisfies both the descent condition (9) and linesearch rule (10) with $t^k = 1$.
b. $\{z^k\}$ converges to z^*, and the rate of convergence is Q-superlinear.
c. If, in addition, G is strongly semismooth at z^*, then the rate of convergence is Q-quadratic.

3.2. Fischer-Burmeister function

Complementarity problems such as the one in (6) can be solved by reformulating them as (square) systems of semismooth equations and applying Algorithm 2 [5]. A reformulation using the Fischer-Burmeister function [11] is used. Let $\phi : \Re^2 \to \Re$ be defined as follows:

$$\phi(a, b) := a + b - \sqrt{a^2 + b^2}.$$

This function has the NCP-property that $\phi(a, b) = 0 \Leftrightarrow 0 \leq a \perp b \geq 0$. Therefore, letting $n = m + c$, $\Phi : \Re^n \to \Re^n$ can be defined as

$$\Phi(z) := \begin{bmatrix} \phi(z_1, F_1(z)) \\ \vdots \\ \phi(z_m, F_m(z)) \\ F_{m+1}(z) \\ \vdots \\ F_n(z) \end{bmatrix}, \tag{11}$$

where there are m nonlinear complementarity constraints and c equation constraints. The properties of this function are summarized in the following theorem.

Theorem 4 ([1])**.** *Let $F : \Re^n \to \Re^n$ be continuously differentiable. Then the following hold:*

a. Φ is a semismooth function. If, in addition, every F_i is twice continuously differentiable with Lipschitz continuous second derivatives, then Φ is strongly semismooth everywhere.
b. $\Psi(z) := \frac{1}{2} \|\Phi(z)\|_2^2$ is continuously differentiable with $\nabla\Psi(z) = H^T \Phi(z)$ for any $H \in \partial_B \Phi(z)$.
c. $\Phi(z^*) = 0$ if and only if z^* solves the complementarity problem defined by F.
d. If z^* is a stationary point of Ψ and there exists an $H \in \partial_B \Phi(z^*)$ that is invertible, then $\Phi(z^*) = 0$, and hence z^* solves the complementarity problem defined by F.

Methods for calculating an element of the B-subdifferential are found, for example, in [1, 5]. While the B-subdifferential is used in the algorithm, an overestimate of the B-subdifferential detailed in the following theorem is used for the proofs in the sequel.

Theorem 5 ([1, 5]). *Let $F : \Re^n \to \Re^n$ be continuously differentiable. Then*

$$\partial_B \Phi(z) \subseteq \left\{ D_a + D_b F'(z) \right\},$$

where $D_a \in \Re^{n\times n}$ and $D_b \in \Re^{n\times n}$ are diagonal matrices with entries defined as follows:

a. For all $i \in \{1, \ldots, m\}$: If $\|(z_i, F_i(z))\| \neq 0$, then

$$(D_a)_{ii} = 1 - \frac{z_i}{\|(z_i, F_i(z))\|}$$
$$(D_b)_{ii} = 1 - \frac{F_i(z)}{\|(z_i, F_i(z))\|};$$

otherwise

$$((D_a)_{ii}, (D_b)_{ii}) \in \left\{ (1-\eta, 1-\rho) \in \Re^2 \;\mid\; \|(\eta, \rho)\| \leq 1 \right\}.$$

b. For all $i \in \{m+1, \ldots, n\}$:

$$(D_a)_{ii} = 0$$
$$(D_b)_{ii} = 1.$$

Furthermore, for all $i \in \{1, \ldots, n\}$, $(D_a)_{ii} \geq 0$, $(D_b)_{ii} \geq 0$, and $(D_a)_{ii} + (D_b)_{ii} > 0$. In particular, $D_a + D_b$ is positive definite.

3.3. Support vector machine specialization

At this stage, it would be nice to present standard convergence material showing that Algorithm 2 converges to a solution of (6), perhaps with a given rate, when the complementarity problem is reformulated by using the Fischer-Burmeister function. Unfortunately, such results are not available. The typical results presented in the literature either assume that the equation in (6) can be explicitly substituted out of the model (which cannot be done in this case) or assume that F is a uniform P-function (which is also not the case here). Instead of using these results, we directly prove the necessary conditions for the support vector machine problem.

The first step to show that Algorithm 2 converges when applied to (6) is to establish that for all $z \in \Re^n$, all $H \in \partial_B \Phi(z)$ are invertible. To do this, we give a method for computing the Newton direction from step 2 of Algorithm 2.

Recall that the complementarity problem (6) is solved by using the Fischer-Burmeister reformulation. Therefore, the following system of equations is solved at every iteration of Algorithm 2:

$$\begin{bmatrix} D_a + D_b(\frac{1}{\nu} I + DAA^T D) & -D_b De \\ e^T D & 0 \end{bmatrix} \begin{bmatrix} x \\ \gamma \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

for some r_1 and r_2 and diagonal matrices D_a and D_b chosen according to Theorem 5.

Proposition 1. *Suppose the mixed complementarity problem and F are defined as in (1). For all $i \in \{1, \ldots, m\}$, if $(D_b)_{ii} = 0$, then $z_i = 0$ and $F_i(z) \geq 0$.*

Proof. Let i be given with $(D_b)_{ii} = 0$. There are two cases to consider. If $\|(z_i, F_i(z))\| > 0$, then

$$0 = (D_b)_{ii} = 1 - \frac{F_i(z)}{\|(z_i, F_i(z))\|} \implies \frac{F_i(z)}{\|(z_i, F_i(z))\|} = 1$$
$$\implies F_i(z) = \|(z_i, F_i(z))\| > 0.$$

Furthermore, since $F_i(z) > 0$ and $F_i(z) = \|(z_i, F_i(z))\|$, $z_i = 0$. In the other case, $\|(z_i, F_i(z))\| = 0$, which implies $z_i = 0$ and $F_i(z) = 0$. Therefore, the conclusion of the proposition holds in both cases, and the proof is complete. □

Proposition 2. *Let* **card** $(P_+) > 0$ *and* **card** $(P_-) > 0$. *Then for the model considered in (6), $D_b \neq 0$.*

Proof. Assume $D_b = 0$. Then by Proposition 1, for all $i \in \{1, \ldots, m\}$, $z_i = 0$ and $F_i(z) \geq 0$. The definition of F implies that $-De\gamma - e \geq 0$ with D defined in Section 2. Since **card** $(P_+) > 0$ and **card** $(P_-) > 0$ by assumption, the system reduces to two inequalities, $\gamma - 1 \geq 0$ and $-\gamma - 1 \geq 0$, which implies $\gamma \geq 1$ and $\gamma \leq -1$, a contradiction. Therefore, the assumption was false, and the proposition is proved. □

Theorem 6. *Let* **card** $(P_+) > 0$, **card** $(P_-) > 0$, *and $\nu > 0$. Then for the model considered in (6) the following matrix system has a unique solution*

$$\begin{bmatrix} D_a + D_b(\frac{1}{\nu} I + DAA^T D) & -D_b De \\ e^T D & 0 \end{bmatrix} \begin{bmatrix} x \\ \gamma \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}$$

for all D_a and D_b defined in Theorem 5 and arbitrary r_1 and r_2.

Proof. Since $D_a \geq 0$ and $D_b \geq 0$ with $D_a + D_b$ positive definite by Theorem 5, it follows from $\nu > 0$ that $D_a + \frac{1}{\nu} D_b$ is also positive definite.

Let $\bar{D} := D_a + \frac{1}{\nu} D_b$. From the Sherman-Morrison-Woodbury identity, we have that

$$(\bar{D} + D_b DAA^T D)^{-1} = \bar{D}^{-1} - \bar{D}^{-1} D_b DA(I + A^T D\bar{D}^{-1} D_b DA)^{-1} A^T D\bar{D}^{-1}.$$

Note that $I + A^T D\bar{D}^{-1} D_b DA$ is a symmetric positive definite matrix and therefore invertible. Hence, $\bar{D} + D_b DAA^T D$ is invertible with the inverse defined by the Sherman-Morrison-Woodbury identity and

$$x = (\bar{D} + D_b DAA^T D)^{-1}(D_b De\gamma + r_1).$$

Substituting x out of the system leaves the following equation,

$$e^T D(\bar{D} + D_b DAA^T D)^{-1}(D_b De\gamma + r_1) = r_2,$$

which simplifies to

$$e^T D(\bar{D} + D_b DAA^T D)^{-1} D_b De\gamma = r_2 - e^T D(\bar{D} + D_b DAA^T D)^{-1} r_1.$$

A proof that $M := e^T D(\bar{D} + D_b DAA^T D)^{-1} D_b De$ is not zero is now given.

Using the Sherman-Morrison-Woodbury identity, we have the following:

$$\begin{aligned} M &= e^T D(\bar{D}^{-1} - \bar{D}^{-1} D_b DA(I + A^T D\bar{D}^{-1} D_b DA)^{-1} A^T D\bar{D}^{-1}) D_b De \\ &= e^T D(\hat{D} - \hat{D}DA(I + A^T D\hat{D}DA)^{-1} A^T D\hat{D}) De, \end{aligned}$$

where $\hat{D} := \bar{D}^{-1} D_b$. Since $\hat{D}$ is a diagonal matrix with nonnegative diagonals, $\hat{D}$ is replaced with $\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}}$ to obtain the system

$$\begin{aligned} M &= e^T D(\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} - \hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA(I + A^T D\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA)^{-1} A^T D\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}}) De \\ &= e^T D\hat{D}^{\frac{1}{2}}(I - \hat{D}^{\frac{1}{2}} DA(I + A^T D\hat{D}^{\frac{1}{2}}\hat{D}^{\frac{1}{2}} DA)^{-1} A^T D\hat{D}^{\frac{1}{2}})\hat{D}^{\frac{1}{2}} De \\ &= e^T D\hat{D}^{\frac{1}{2}}(I + \hat{D}^{\frac{1}{2}} DAA^T D\hat{D}^{\frac{1}{2}})^{-1}\hat{D}^{\frac{1}{2}} De, \end{aligned}$$

where the last equality comes from the Sherman-Morrison-Woodbury identity.

The inner term, $(I + \hat{D}^{\frac{1}{2}} DAA^T D\hat{D}^{\frac{1}{2}})^{-1}$, is a symmetric positive definite matrix. Furthermore, since $\mathbf{card}\,(P_+) > 0$, $\mathbf{card}\,(P_-) > 0$ by assumption, it follows from Proposition 2 that $D_b \neq 0$. Hence, $\hat{D}^{\frac{1}{2}} \neq 0$ and $e^T D\hat{D}^{\frac{1}{2}} \neq 0$. Therefore, $M \neq 0$, and γ and x are uniquely determined for any r_1 and r_2. □

Hence, by Theorem 3.10, the Newton direction exists for all z and all choices of D_a and D_b. The next step is to show that the sequence generated by Algorithm 2 has an accumulation point.

Theorem 7. *Suppose that* $\mathbf{card}\,(P_+) > 0$ *and* $\mathbf{card}\,(P_-) > 0$. *Then Algorithm 2, applied to the problem (6), has an accumulation point.*

Proof. The proof is adapted from [34]. The level sets of $\Psi(z)$ are shown to be bounded, and hence by the descent properties of the algorithm, there must be an accumulation point of the iterates. To prove that the level sets of $\Psi(z)$ (where $z = (x, \gamma)$) are bounded, we need only show that $\|\Phi\|$ defined by (11) is coercive. The fact that if $(u \to -\infty)$ or $(v \to -\infty)$ or $(u \to \infty$ and $v \to \infty)$, then $\|\phi(u, v)\| \to \infty$ is used extensively in the remainder of this proof.

Suppose not; that is, suppose $\|\Phi\|$ is not coercive. Let $\{\|x^k, \gamma^k\|\} \to \infty$ be such that

$$\left\|\Phi(x^k, \gamma^k)\right\| < \infty. \tag{12}$$

Without loss, a subsequence can be taken for which

$$\frac{(x^k, \gamma^k)}{\|x^k, \gamma^k\|} \to (\bar{x}, \bar{\gamma}) \neq 0.$$

Furthermore, on this subsequence

$$\frac{F(x^k, \gamma^k)}{\|x^k, \gamma^k\|} \to \begin{bmatrix} Q\bar{x} - De\bar{\gamma} \\ e^T D\bar{x} \end{bmatrix},$$

where Q is the positive definite matrix multiplying x in (6).

Define $\bar{w}_i = F_i(\bar{x}, \bar{\gamma})$ for $i = 1, 2, \ldots, m$. If $\bar{w}_i < 0$ for some i, then $F_i(x^k, \gamma^k) \to -\infty$, resulting in $|\phi(x_i^k, F_i(x^k, \gamma^k))| \to \infty$. Hence $\left\|\Phi(x^k, \gamma^k)\right\| \to \infty$, a contradiction to (12). Similarly, whenever $\bar{x}_i < 0$. Thus, $\bar{w} \geq 0$, $\bar{x} \geq 0$.

Now, if $\bar{w}_i \bar{x}_i > 0$ for some i, then $\{x_i^k\} \to \infty$ and $\{Q_i.x^k - D_{ii}\gamma^k - 1\} \to \infty$. In this case, $|\phi(x_i^k, F_i(x^k, \gamma^k))| \to \infty$, and hence $\left\|\Phi(x^k, \gamma^k)\right\| \to \infty$, a contradiction to (12). Thus, $\bar{w}^T \bar{x} = 0$.

Furthermore, if $e^T D\bar{x} \neq 0$, it follows that $|\Phi_{m+1}(x^k, \gamma^k)| \to \infty$, also a contradiction to (12). Thus, $e^T D\bar{x} = 0$.

Note that $\bar{w}^T \bar{x} = 0$ and $e^T D\bar{x} = 0$ imply that $\bar{x}^T Q\bar{x} = 0$. Since Q is positive definite, this implies that $\bar{x} = 0$. In this case, it follows from $\bar{w} \geq 0$ that $-De\bar{\gamma} \geq 0$. Now, because **card** $(P_+) > 0$ and **card** $(P_-) > 0$, this implies that $\bar{\gamma} = 0$. However, this contradicts the fact that $(\bar{x}, \bar{\gamma}) \neq 0$. Thus, $\|\Phi\|$ is coercive, and the proof is complete. □

Note that Theorem 7 remains valid for any function ϕ that satisfies the NCP-property and the simple implications given in the first paragraph of the proof above.

Corollary 1. *Suppose that* **card** $(P_+) > 0$ *and* **card** $(P_-) > 0$. *Algorithm 2 applied to (6) converges, and the rate of convergence is Q-quadratic.*

Proof. Φ is strongly semismooth for this problem, since F is linear and Ψ is continuously differentiable. Furthermore, by Theorem 7 and Theorem 3, there is an accumulation point of the sequence generated by Algorithm 2 that is a stationary point for Ψ. Since all of the elements of the B-subdifferential are invertible by Theorem 6, this stationary point solves the system $\Phi(x) = 0$ by Theorem 4. The conclusion then follows from Theorem 3. □

4. Implementation and computational results

The main computation performed at each iteration of Algorithm 2 is to compute the Newton direction given $\Phi(x^k)$ and $H^k \in \partial_B \Phi(x^k)$. As shown in Theorem 6, the required direction generation can be calculated by using

$$\begin{aligned} x &= (\bar{D} + D_b DAA^T D)^{-1}(D_b De\gamma + r_1) \\ e^T D(\bar{D} + D_b DAA^T D)^{-1} D_b De\gamma &= r_2 - e^T D(\bar{D} + D_b DAA^T D)^{-1} r_1. \end{aligned}$$

Defining the two common components,

$$\begin{aligned} y &= (\bar{D} + D_b DAA^T D)^{-1} D_b De \\ z &= (\bar{D} + D_b DAA^T D)^{-1} r_1, \end{aligned}$$

leaves the equivalent system of equations:

$$\begin{aligned} \gamma &= \frac{r_2 - e^T Dz}{e^T Dy} \\ x &= y\gamma + z. \end{aligned}$$

Note that Theorem 6 guarantees that $e^T Dy \neq 0$. The implementation uses the Sherman-Morrison-Woodbury identity to calculate y and z simultaneously with only two passes

through the A matrix. Having y and z, one can easily construct the direction (x, γ). The major computational effort when using the Sherman-Morrison-Woodbury identity to apply $(\bar{D} + D_b D A A^T D)^{-1}$ is in calculating $(I + A^T D \bar{D}^{-1} D_b D A)^{-1}$, since this requires mf^2 floating-point operations, where m is typically very large.

However, further inspection reveals that the number of operations is a function of the number of active elements for which $(D_b)_{ii} > 0$. By Proposition 1 the inactive elements (those with $(D_b)_{ii} = 0$) have $z_i = 0$ and $F_i(z) \geq 0$. For the support vector machine application, the active components at the solution correspond to the support vectors, and the number of support vectors is typically much smaller than m. Therefore, one would expect that near the solution most of the components of D_b would be equal to zero. Hence, as the iterations proceed, the amount of work per iteration should decrease as a result of the removal of the inactive components. This reduction in computational effort is similar to that found in active set methods, even though the algorithm does not explicitly use an active set.

To calculate H^k and $\Phi(z^k)$ is straightforward and uses two passes through the A matrix. Whenever an element is found for which $\|z_i, F_i(z)\| = 0$, the code simply sets $D_a = \frac{1}{2}$ and $D_b = \frac{1}{2}$. While, in this case, the resulting H^k may not be an element of $\partial_B \Phi(z^k)$, no difficulties were encountered on the test problems by using this definition. Note that the theory from [1] can be used to calculate an element of $\partial_B \Phi(z^k)$ in these cases by using two additional passes through the A matrix.

The main drawback of the semismooth method is in the number of function evaluations that may be needed in order to satisfy the linesearch rule. Therefore, a nonmonotone linesearch procedure [12, 13, 6] is used within the semismooth implementation to limit the number of linesearches performed. The nonmonotone procedure allows for increases in the merit function by using a reference value in the linesearch test that decreases on average. The use of such a technique affects neither the convergence nor rate of convergence results for the algorithm. For all of the tests reported, the Newton direction was always accepted, without resorting to the linesearch procedure. Furthermore, the need to use the gradient of the merit function was not encountered. As a result, the code is optimized for the case where the Newton direction is accepted.

The implementation of the semismooth algorithm for the support vector machine application uses a total of five vectors with n elements, one $f \times f$ matrix, and several vectors with f elements. Four passes through the A matrix are performed during each iteration of the algorithm. Access to the n vectors and observation matrix is provided by the low-level routines developed for the interior-point method in [8]. Access to the problem data (feature measurements) and vectors is provided by using asynchronous I/O constructs. All of the data is stored on a large disk and sequentially accessed. While one block of data is being read from the disk, work is performed on the data currently available. A buffer size of 250,000 elements is used by this particular code. The buffer size corresponds to the number of rows of the A matrix kept in-core, as well as the number of elements of the vectors kept. For the massive problem with 60 million observations studied in Section 4.1, a total of 75 MB of RAM was used, which is easily accommodated by most personal computers.

In principle, a conjugate gradient method could be used to calculate the Newton direction by solving the linear system of equations. However, the number of operations per solve is of the same order as using the Sherman-Morrison-Woodbury identity, while

many more passes through the data are required. Since accessing a large data set stored on disk is expensive, the direct method is preferred.

4.1. Interior-point comparison

A comparison with the interior-point method in [8] shows that the linear algebra performed is similar. However, the semismooth method can use the reduction in the linear algebra cost because of the active component identification, whereas the interior-point method cannot, because of the interiority condition. Furthermore, the semismooth method performs only one solve per iteration, while the (predictor-corrector) interior-point method does two. One would therefore expect to obtain better performance from the semismooth method than from the interior-point method. To test this hypothesis, we used the randomly generated test problem from [8], which has 60 million observations where each observation measures 34 features and each feature is an integer between 1 and 10. A starting point of $(x, \gamma) = 0$ was used for these tests. The log of the residual is plotted in Figure 1 for a run using the full dataset of 60 million observations. Note the observed convergence behaves in the manner predicted by the theory.

Figure 2 plots the percentage of elements per iteration where $(D_b)_{ii} > 0$. Toward the beginning of the computation, all of the elements are active, leading to full cost factor/solves. In later iterations, however, the potential support vectors are reduced to 80% of the original problem data, leading to an 80% reduction in the time to perform the factor. A zero tolerance of 10^{-10} was used during the calculation; that is, components for which $(D_b)_{ii} < 10^{-10}$ were treated as zero in the computations.

As the number of observations selected was varied between 1 million and 60 million elements, the number of iterations performed by semismooth method remained constant. In all cases, 11 function evaluations and 10 factor/solves were performed. The inf-norm of the residual at the solution was between 10^{-12} and 10^{-9} for all of these tests. In Figure 3, the number of iterations and the total time taken with the semismooth method and the interior-point method from [8] for varying problem size is compared. The same machine and setup from [8] were used for this test so that the times are comparable. In particular, the machine used was a 296 MHz SUN Ultrasparc with 2 processors and 768 MB of RAM. Throughout the testing the second processor was typically running a different user's jobs. All data was stored on a locally mounted disk with 18 gigabytes of storage space to prevent overhead resulting from network communication and disk contention with nightly backups.

Even though the total number of iterations taken by the semismooth method is larger than that of the interior-point method, a reduction in time of over 35% is observed on the 60 million observation problem. This reduction comes primarily from three sources. First, the amount of I/O required per iteration is less for the semismooth method than for the interior-point method. Second, each iteration of the semismooth method requires one solve instead of two for the predictor-corrector interior-point code. Third, the work involved in factorization is reduced by the (implicit) active set nature of the semismooth method.

Note that [8] shows that the interior-point method significantly outperforms SVM-Torch [4], a method from the machine learning community, on the chosen dataset. The

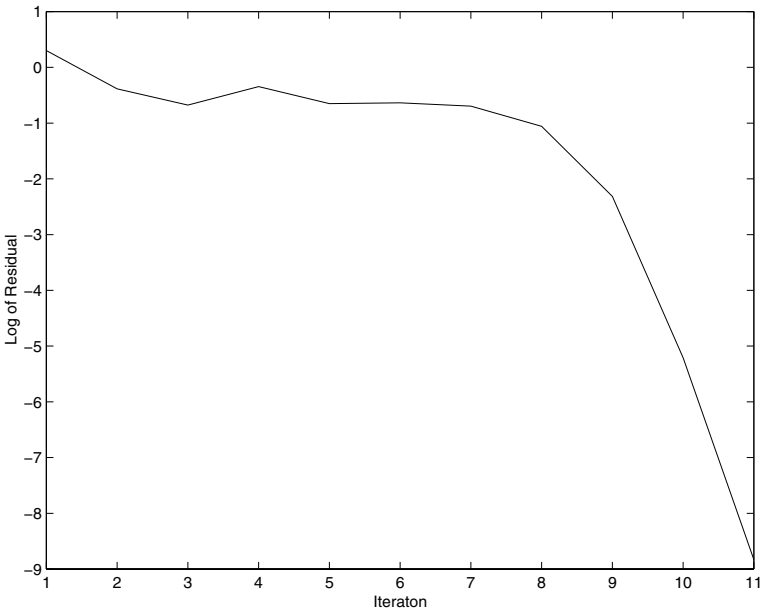


Fig. 1. Log of the residual per iteration on example problem with 60 million observations and 34 features.

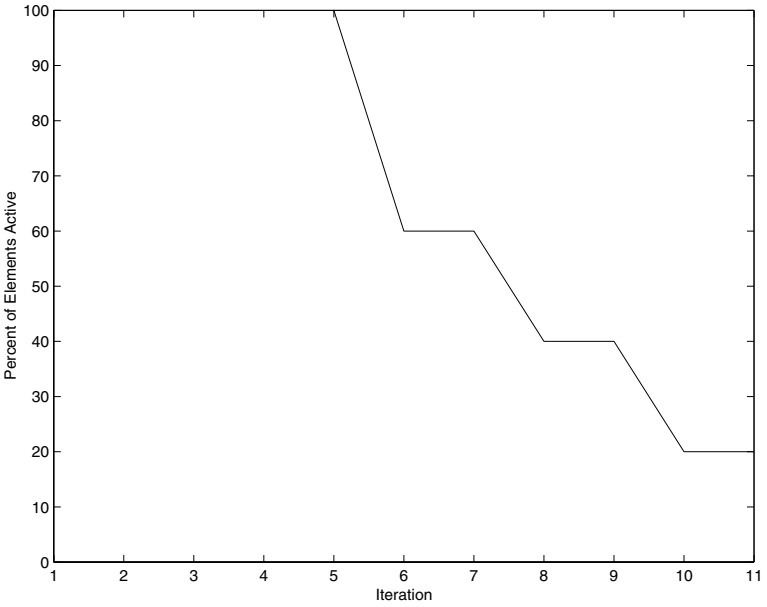
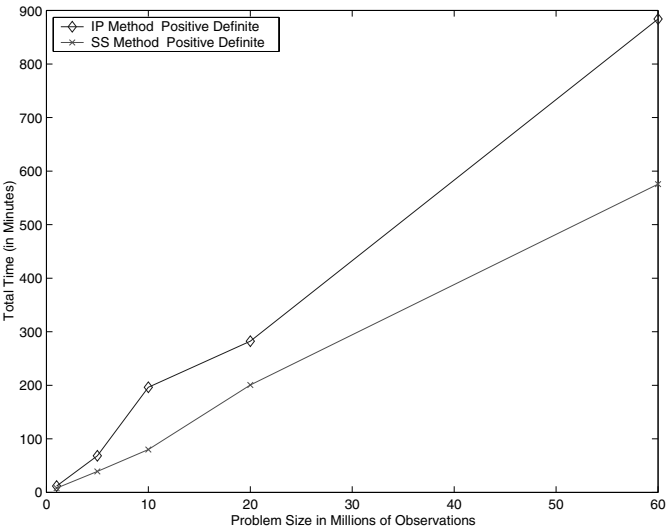
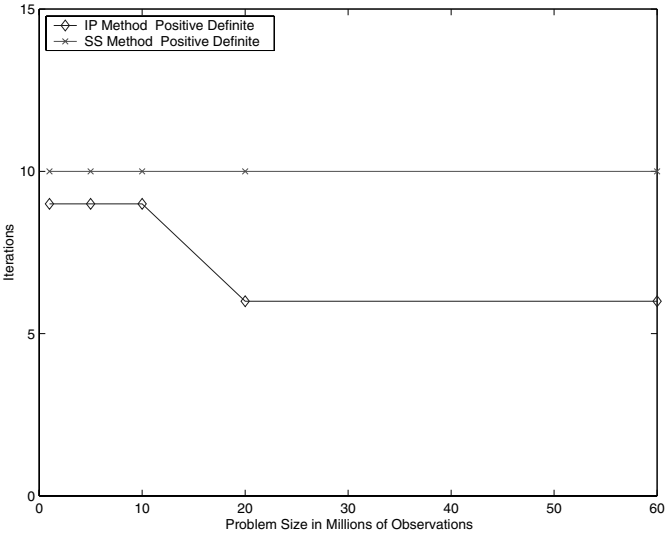


Fig. 2. Percentage of observations that are active per iteration on example problem with 60 million observations and 34 features.



(a) Total solution time



(b) Iterations

Fig. 3. Comparisons between an interior-point method and the semismooth method for varying problem size.

Table 1. Comparison of iterations and times when observation data is added.

	Interior-Point		Semismooth			
			Scratch		Restart	
Observations	Iter.	Time	Iter.	Time	Iter.	Time
3760	15	3.30	15	2.26	-	-
4177	18	4.47	16	2.53	8	1.09

same conclusion can be made with regard to the semismooth method because it outperforms the interior-point method.

4.2. Restarts

The semismooth algorithm presented is globally convergent from any starting point. This feature can be used, for example, when new observations are added to an existing dataset. Restarting the semismooth method from the solution obtained for the old dataset may be beneficial and result in a reduction in time when compared with solving the problem from scratch. In comparison, interior-point methods cannot currently be started from an arbitrary point.

The Abalone dataset from the UCI Repository [26] was used as the basis for this test. This dataset contains 10 features and 4,177 observations. The observation data was scaled and shifted so that all measurements were between -1 and 1, and some measurements have values of -1 and 1 for each feature. An approximation to a nonlinear separating surface was used for this problem. In particular, the kernel used (defined componentwise) was

$$K(a, b) := (a^T b + 1)^5$$

and corresponds to using a surface defined by a fifth degree polynomial to separate the data. The dense kernel matrix was approximated by using the technique from [10] where an incomplete Cholesky factorization of $K(A^T, A)$ is used to obtain the features for the support vector machine computation.

In order to study the impact on performance when the number of observations is increased, the number of columns in the nonlinear kernel approximation was fixed at 50. The resulting dataset fits into main memory. The test starts by solving the support vector machine problem with 90% of the data (3,760 observations). The optimization is then restarted using the full data (4,177 observations) from the solution obtained with the reduced set. The solution values x for the new observations are initialized to zero. The iterations and times for solving the problem from scratch and restarting are compared in Table 1. The results indicate that if all the data is known a priori, the best strategy is to solve the entire problem. If the dataset evolves, however, the semismooth method can be effectively restarted, leading to fewer iterations and a reduced time when compared with starting from scratch. Furthermore, all of the iterations of the restarted problem exhibit significant reductions in the linear algebra cost because of the active component identification.

Note that when using an approximation to a nonlinear kernel, new features are added when the approximation is refined. Restarting the semismooth method from the solution

obtained for the lower fidelity model is not beneficial in this case because significant changes in the optimal surface are made in order to reduce the misclassification error.

The results from [10], which uses an interior-point algorithm similar to [8] and the same nonlinear kernel approximation, indicate that the interior-point methods can be much faster than alternative methods from the machine learning community. In particular, they show significant reductions in time when compared with SMO [28] and SVM^{light} [14] on particular test problems. Since the semismooth method is typically faster than the interior-point method, a similar conclusion can be reached for the semismooth method.

5. Conclusions

This paper presented a formulation for the support vector machine problem and proved that the semismooth algorithm converges when applied to it. These results extend the general theory to the special case of support vector machine problems.

Significant reductions in the direction generation time can be obtained by using information related to the active components. Furthermore, the algorithm identifies these active components automatically. The number of active components is related to the number of support vectors in the model, which is typically much smaller than the number of variables. The results presented indicate this to be the case and show substantial reductions in solution time by exploiting this fact. An added benefit of the semismooth algorithm is the ability to restart when the data evolves.

A comparison of the semismooth method with an interior-point method applied to the same model demonstrated a significant decrease in the total solution time. A problem with 60 million variables on a standard workstation (using only 75 MB of RAM) was solved in around 9.5 hours. Parallel implementations of the code are also possible. However, the main benefit of this work is to show that a large machine with many processors and a huge amount of RAM is not needed to obtain reasonable performance.

Other reformulations of the support vector machine that do not contain the linear constraint can also be used. In this case, similar improvements in performance are realized when compared with the interior-point method on the same model. This formulation was not discussed here, because the theory is uninteresting. For completeness, we note that removing the linear constraint gives a different model that is simply a bound-constrained positive definite quadratic program. When the semismooth method is used, the amount of computation and number of iterations is about the same with or without the linear constraint. Other techniques have been described for this case in [20]. Furthermore, it may be possible to use some of the techniques outlined in [10] as an alternative to the Sherman-Morrison-Woodbury formula, provided issues related to symmetry can be addressed.

While the semismooth method developed in this paper outperforms the interior-point method detailed in [8], the latter method is more general in that even more formulations can be solved. A key requirement for the semismooth method is the positive definiteness of the quadratic term in the optimization problem. This assumption is not required for the interior-point method, so it can solve problems where the quadratic term is only positive semidefinite.

References

1. Billups, S.C.: Algorithms for Complementarity Problems and Generalized Equations. PhD thesis, University of Wisconsin, Madison, Wisconsin, August 1995
2. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2** (2), 121–167 (1998)
3. Clarke, F.H.: Optimization and Nonsmooth Analysis. John Wiley & Sons, New York, 1983
4. Collobert, R., Bengio, S.: SVMTorch: Support vector machines for large-scale regression problems. J. Machine Learn. Res. **1**, 143–160 (2001)
5. De Luca, T., Facchinei, F., Kanzow, C.: A semismooth equation approach to the solution of nonlinear complementarity problems. Math. Program. **75**, 407–439 (1996)
6. Ferris, M.C., Lucidi, S.: Nonmonotone stabilization methods for nonlinear equations. J. Optim. Theor. Appl. **81**, 53–71 (1994)
7. Ferris, M.C., Mangasarian, O.L.: Breast cancer diagnosis via linear programming. IEEE Comput. Sci. Eng. **2**, 70–71 (1995)
8. Ferris, M.C., Munson, T.S.: Interior point methods for massive support vector machines. SIAM J. Optim. **13**, 783–804 (2003)
9. Ferris, M.C., Pang, J.S.: Engineering and economic applications of complementarity problems. SIAM Rev. **39**, 669–713 (1997)
10. Fine, S., Scheinberg, K.: Efficient SVM training using low-rank kernel representations. J. Mach. Learn. Res. **2**, 243–264 (2001)
11. Fischer, A.: A special Newton-type optimization method. Optimization **24**, 269–284 (1992)
12. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
13. Grippo, L., Lampariello, F., Lucidi, S.: A class of nonmonotone stabilization methods in unconstrained optimization. Numer. Math. **59**, 779–805 (1991)
14. Joachims, T.: Making large-scale support vector machine learning practical. In: B. Schölkopf, C.J.C. Burges, A.J. Smola (eds.), Advances in Kernel Methods - Support Vector Learning, MIT Press, 1999, pp. 169–184
15. Mangasarian, O.L.: Nonlinear Programming. McGraw–Hill, New York, 1969, SIAM Classics in Applied Mathematics 10, SIAM, Philadelphia, 1994
16. Mangasarian, O.L.: Machine learning via polyhedral concave minimization. In: H. Fischer, B. Riedmueller, S. Schaeffler (eds.), Applied Mathematics and Parallel Computing - Festschrift for Klaus Ritter, Physica-Verlag A Springer-Verlag Company, Heidelberg, 1996, pp. 175–188
17. Mangasarian, O.L.: Mathematical programming in machine learning. In: G. Di Pillo, F. Giannessi (eds.), Nonlinear Optimization and Applications, Plenum Publishing, New York, 1996, pp. 283–295
18. Mangasarian, O.L.: Mathematical programming in data mining. Data Mining and Knowledge Discovery **1**, 183–201 (1997)
19. Mangasarian, O.L.: Generalized support vector machines. In: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, Massachusetts, 2000, pp. 135–146
20. Mangasarian, O.L., Musicant, D.R.: Lagrangian support vector machines. J. Mach. Learn. Res. **1**, 161–177 (2001)
21. Mangasarian, O.L., Musicant, D.R.: Active set support vector machine classification. In: T.K. Leen, T.G. Dietterich, V. Tresp (eds.), Advances in Neural Information Processing Systems 13, MIT Press, 2001, pp. 577–583
22. Mangasarian, O.L., Street, W.N., Wolberg, W.H.: Breast cancer diagnosis and prognosis via linear programming. Oper. Res. **43**, 570–577 (1995)
23. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. SIAM News **23**, 1 & 18 (1990)
24. Mifflin, R.: Semismooth and semiconvex functions in constrained optimization. SIAM J. Contr. Optim. **15**, 957–972 (1977)
25. Munson, T.S., Facchinei, F., Ferris, M.C., Fischer, A., Kanzow, C.: The semismooth algorithm for large scale complementarity problems. INFORMS J. Comput. **13**, 294–311 (2001)
26. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, California, http://www.ics.uci.edu/AI/ML/MLDB-Repository.html, 1992
27. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, San Diego, California, 1970
28. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. In: B. Schölkopf, C.J.C. Burges, A.J. Smola (eds.), Advances in Kernel Methods – Support Vector Learning, MIT Press, 1999, pp. 185–208

29. Qi, L.: Convergence analysis of some algorithms for solving nonsmooth equations. Math. Oper. Res. **18**, 227–244 (1993)
30. Qi, L., Sun, D. A survey of some nonsmooth equations and smoothing Newton methods. In: A. Eberhard, B. Glover, R. Hill, D. Ralph (eds.), Progress in Optimization, volume 30 of Applied Optimization, Kluwer Academic Publishers, Dordrecht, 1999, pp. 121–146
31. Qi, L., Sun, J.: A nonsmooth version of Newton's method. Math. Program. **58**, 353–368 (1993)
32. Schölkopf, B., Burges, C., Smola, A. (eds.): Advances in Kernel Methods: Support Vector Machines. MIT Press, Cambridge, Massachusetts, 1998
33. Stone, M.: Cross-validatory choice and assessment of statistical predictions. J. Royal Stat. Soc. **36**, 111–147 (1974)
34. Tseng, P.: Growth behavior of a class of merit functions for the nonlinear complementarity problem. J. Optim. Theor. Appl. **89**, 17–37 (1996)
35. Vapnik, V.N.: The Nature of Statistical Learning Theory. second edition., Springer-Verlag, New York, 2000