A Coordinate Gradient Descent Method for Linearly Constrained Smooth Optimization and Support Vector Machines Training [1]

Paul Tseng
Department of Mathematics
University of Washington
Seattle, WA 98195, U.S.A.
E-mail: tseng@math.washington.edu

Sangwoon Yun
Department of Mathematics
University of Washington
Seattle, WA 98195, U.S.A.
E-mail: sangwoon@math.washington.edu

March 18, 2007 (2nd revision: October 7, 2008)

Abstract: Support vector machines (SVMs) training may be posed as a large quadratic program (QP) with bound constraints and a single linear equality constraint. We propose a (block) coordinate gradient descent method for solving this problem and, more generally, linearly constrained smooth optimization. Our method is closely related to decomposition methods currently popular for SVM training. We establish global convergence and, under a local error bound assumption (which is satisfied by the SVM QP), linear rate of convergence for our method when the coordinate block is chosen by a Gauss-Southwell-type rule to ensure sufficient descent. We show that, for the SVM QP with n variables, this rule can be implemented in $O(n)$ operations using Rockafellar's notion of conformal realization. Thus, for SVM training, our method requires only $O(n)$ operations per iteration and, in contrast to existing decomposition methods, achieves linear convergence without additional assumptions. We report our numerical experience with the method on some large SVM QP arising from two-class data classification. Our experience suggests that the method can be efficient for SVM training with nonlinear kernel.

Key words. Support vector machine, quadratic program, continuous quadratic knapsack problem, linear constraints, conformal realization, coordinate gradient descent, global convergence, linear convergence rate, error bound

[1]This research is supported by the National Science Foundation, Grant No. DMS-0511283.

1 Introduction

Support vector machines (SVMs), invented by Vapnik [45], have been much used for classification and regression, including text categorization, image recognition, hand-written digit recognition, and bioinformatics; see [7] and references therein. The problem of training a SVM may be expressed via duality as a convex quadratic program (QP) with bound constraints plus one equality constraint:

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & f(x) = \frac{1}{2} x^T Q x - e^T x \\ \text{s.t.} & a^T x = 0, \\ & 0 \le x \le Ce, \end{array} \tag{1}$$

where $a \in \Re^n$, $0 < C \le \infty$, $e \in \Re^n$ is the vector of all ones, and $Q \in \Re^{n \times n}$ is a symmetric positive semidefinite matrix with entries of the form

$$Q_{ij} = a_i a_j K(z_i, z_j), \tag{2}$$

with $K : \Re^p \times \Re^p \to \Re$ ("kernel function"), and $z_i \in \Re^p$ ("ith data point"), $i \in \mathcal{N} \stackrel{\text{def}}{=} \{1, \ldots, n\}$. (Here, "s.t." is short for "subject to".) Popular choices of K are the linear kernel $K(z_i, z_j) = z_i^T z_j$ (for which $Q = Z^T Z$, with $Z = [a_1 z_1 \ \cdots \ a_n z_n]$, and so rank$Q \le p$) and the radial basis function (rbf) kernel $K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|^2)$ where γ is a constant. Often p ("number of features") is not large ($4 \le p \le 300$), n is large ($n \ge 5000$), and Q is fully dense and even indefinite; see Section 7 for more discussions.

The density and huge size of Q pose computational challenges in solving (1). Interior-point methods cannot be directly applied, except in the case of linear kernel where Q has low rank or Q is the sum of a low-rank matrix and a positive multiple of the identity matrix; see [9, 10]. For nonlinear kernel, Fine and Scheinberg [11, Section 4] proposed approximating Q by a low-rank incomplete Cholesky factorization with symmetric permutations. Recently, Scheinberg [40] reported good numerical experience with an active-set method for SVM problems with positive semidefinite Q and, in particular, when the rbf kernel is used. It uses rank-one update of a Cholesky factorization of the reduced Hessian to resolve subproblems. Earlier, Osuna et al. [33] proposed a column-generation approach which solves a sequence of subproblems obtained from (1) by fixing some components of x at the bounds. They reported solving problems with up to $n = 100,000$ data points in 200 hours on a Sun Sparc 20. The SVM code in [39] is based on this approach. Motivated by this approach, decomposition methods based on iterative block-coordinate descent were subsequently developed and have become popular for solving (1), beginning with the work of Joachims [16], Platt [35], and others, and implemented in SVM codes such as SVM^{light} [16] and LIBSVM [5]. At each iteration of such a method, called *sequential minimal optimization* (SMO) method by Platt, a small index subset $\mathcal{J} \subseteq \mathcal{N}$ is chosen and the objective function of (1) is minimized with respect to the coordinates x_j, $j \in \mathcal{J}$, subject to the constraints and with the other coordinates held fixed at their current value. This minimization needs only those entries of Q indexed by $\mathcal{J}$, which can be quickly generated using (2) and, in the case of $|\mathcal{J}| = 2$, has a closed form solution.[2] (We need $|\mathcal{J}| \geq 2$ to satisfy the equality constraint $a^T x = 0$.) Such a method is simple and easy to implement, and for suitable choices of the index set $\mathcal{J}$, called *working set*, has good convergence properties in theory and in practice. The rows of Q indexed by $\mathcal{J}$ can be cached when updating $\nabla f(x)$ at each iteration, so it need not be recomputed from (2) and thus reduces CPU time. Although block-coordinate descent has been well studied for bound constrained optimization (see [2, 31, 44] and references therein), its use for linearly constrained optimization has been little studied prior to SVM.

A good choice of the working set $\mathcal{J}$ is crucial for speed and robustness. Platt [35] chose $\mathcal{J}$ heuristically with $|\mathcal{J}| = 2$. Subsequently, more systematic choices of $\mathcal{J}$ have been proposed and issues such as computational complexity, global convergence, asymptotic convergence rate, and numerical performance were studied by Joachims [16], Chang, Hsu and Lin [4], Keerthi et al. [18, 20], Lin [22, 23, 24], Hush and Scovel [14], List and Simon [26, 27], Simon [42], Fan et al. [8], Palagi and Sciandrone [34], Chen et al. [6], Hush et al. [15], Glamachers and Igel [13], Lucidi et al. [28]; see Section 6.2 for a detailed comparison.

Recently, the authors [44] proposed a block-coordinate gradient descent (abbreviated as CGD) method for minimizing the sum of a smooth function and a block-separable convex function. This method was shown to have good convergence properties in theory and in practice. In this paper, we extend this method to solve the SVM QP (1) and, more generally, a linearly constrained smooth optimization problem:

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & f(x) \\ \text{s.t.} & x \in X \stackrel{\text{def}}{=} \{x \mid Ax = b,\ l \leq x \leq u\}, \end{array} \tag{3}$$

where $f : \Re^n \to \Re$ is smooth (i.e., continuously differentiable), $A \in \Re^{m \times n}$, $b \in \Re^m$, and $l \leq u$ (possibly with $-\infty$ or ∞ components). SVM corresponds to $m = 1$ while ν-SVM corresponds to $m = 2$ [6, 41].[3] At each iteration of our CGD method, a quadratic approximation of f is minimized with respect to a subset of coordinates x_j, $j \in \mathcal{J}$, to generate a feasible descent direction, and an inexact line search on f along this direction is made to update the iterate. For convergence, we propose choosing $\mathcal{J}$ analogously to the Gauss-Southwell-q rule in [44]; see (9). We show that each cluster point of the iterates generated by this method is a stationary point of (3); see Theorem 4.1. Moreover, if a local error bound on the distance to the set of stationary points $\bar{X}$ of (3) holds and the isocost surfaces of f restricted to $\bar{X}$ are properly separated, then the iterates generated by our method converge at least linearly to a stationary point of (3); see Theorem 5.1. To our knowledge, this is the first globally convergent block-coordinate update method for general linearly constrained smooth optimization. It has the advantage of simple iterations, and is suited for large scale problems with n large and m small. When specialized to the SVM QP (1), our method is similar to the modified SMO method of Chen et al. [6] and our choice of $\mathcal{J}$ may be viewed as an approximate second-order version of the working set proposed by Chang, Hsu and Lin [4] (also see (40)), whereby a separable quadratic term is added to the objective and $\mathcal{J}$ is

[2] $|\mathcal{J}|$ denotes the cardinality of $\mathcal{J}$.

[3] ν-SVM replaces "$-e^T x$" in the objective function of (1) by the constraint "$e^T x = \nu$".

chosen as an approximate minimum (i.e., its objective value is within a constant factor of the minimum value). For $m = 1$ and $\ell = 2$, such $\mathcal{J}$ can be found in $O(n)$ operations by solving a continuous quadratic knapsack problem and then finding a conformal realization [37, Section 10B] of the solution; see Section 6. Moreover, the local error bound holds for (1) always, even if Q is indefinite; see Proposition 5.1. Thus, for SVM, our method is implementable in $O(n)$ operations per iteration and achieves linear convergence without assuming strict complementarity or Q is positive definite as in previous analyses of decomposition methods [6, 8, 23]. We report in Section 7 our numerical experience with the CGD method on large SVM QP. Our experience suggests that the method can be competitive with a state-of-the-art SVM code when a nonlinear kernel is used. We give conclusions and discuss extensions in Section 8.

During the writing of this paper, Lin et al. [25] independently proposed a decomposition method for solving the special case of (3) with $m = 1$. This method uses a similar line search as our method but generates the descent direction differently, using linear approximations of f instead of quadratic approximations and using working sets $\mathcal{J}$ with $|\mathcal{J}| = 2$ and x_j being "sufficiently free" for some $j \in \mathcal{J}$. Global convergence to stationary points is shown assuming such x_j is uniformly bounded away from its bounds, and improvement over LIBSVM on test problems using the rbf kernel is reported.

In our notation, $\Re^n$ denotes the space of n-dimensional real column vectors, ^{T} denotes transpose. For any $x \in \Re^n$, x_j denotes the jth component of x, and $\|x\| = \sqrt{x^T x}$. For any nonempty $\mathcal{J} \subseteq \mathcal{N} = \{1, \ldots, n\}$, $|\mathcal{J}|$ denotes the cardinality of $\mathcal{J}$. For any symmetric matrices $H, D \in \Re^{n \times n}$, we write $H \succeq D$ (respectively, $H \succ D$) to mean that $H - D$ is positive semidefinite (respectively, positive definite). $H_{\mathcal{J}\mathcal{J}} = [H_{ij}]_{i,j \in \mathcal{J}}$ denotes the principal submatrix of H indexed by $\mathcal{J}$. $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ denote the minimum and maximum eigenvalues of H. We denote by I the identity matrix and by 0 the matrix of zero entries. Unless otherwise specified, $\{x^k\}$ denotes the sequence $x^0, x^1, \ldots$.

2 A Coordinate Gradient Descent Method

In our method, we use $\nabla f(x)$ to build a quadratic approximation of f at x and apply coordinate descent to generate an improving feasible direction d at x. More precisely, we choose a nonempty subset $\mathcal{J} \subseteq \mathcal{N}$ and a symmetric matrix $H \in \Re^{n \times n}$ (approximating the Hessian $\nabla^2 f(x)$), and move x along the direction $d = d_H(x; \mathcal{J})$, where

$$d_H(x; \mathcal{J}) \stackrel{\text{def}}{=} \arg\min_{d \in \Re^n} \left\{ \nabla f(x)^T d + \frac{1}{2} d^T H d \;\middle|\; x + d \in X,\; d_j = 0 \; \forall j \notin \mathcal{J} \right\}. \tag{4}$$

Here $d_H(x; \mathcal{J})$ depends on H through $H_{\mathcal{J}\mathcal{J}}$ only. To ensure that $d_H(x; \mathcal{J})$ is well defined, we assume that $H_{\mathcal{J}\mathcal{J}}$ is positive definite on $\text{Null}(A_{\mathcal{J}})$ (the null space of $A_{\mathcal{J}}$) or, equivalently, $B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} \succ 0$, where $A_{\mathcal{J}}$ denotes the submatrix of A comprising columns indexed by $\mathcal{J}$ and $B_{\mathcal{J}}$ is a matrix whose columns form an orthonormal basis for $\text{Null}(A_{\mathcal{J}})$. For (3), we can choose H such that $H_{\mathcal{J}\mathcal{J}} = Q_{\mathcal{J}\mathcal{J}}$ if $B_{\mathcal{J}}^T Q_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} \succ 0$ and otherwise $H_{\mathcal{J}\mathcal{J}} = Q_{\mathcal{J}\mathcal{J}} + \rho I$ with $\rho > 0$ such that $B_{\mathcal{J}}^T Q_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} + \rho I \succ 0$; see [6, 8] for a similar perturbation technique.

We have the following lemma, analogous to [44, Lemma 2.1], showing that d is a descent direction at x whenever $d \neq 0$. We include its proof for completeness.

Lemma 2.1 *For any $x \in X$, nonempty $\mathcal{J} \subseteq \mathcal{N}$ and symmetric $H \in \Re^{n\times n}$ with $B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} \succ 0$, let $d = d_H(x;\mathcal{J})$ and $g = \nabla f(x)$. Then*

$$g^T d \leq -d^T H d \leq -\lambda_{\min}(B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}})\|d\|^2. \tag{5}$$

Proof. For any $\alpha \in (0,1)$, we have from (4) and the convexity of the set X that

$$g^T d + \frac{1}{2} d^T H d \quad \leq \quad g^T(\alpha d) + \frac{1}{2}(\alpha d)^T H (\alpha d) \quad = \quad \alpha g^T d + \frac{1}{2}\alpha^2 d^T H d.$$

Rearranging terms yields

$$(1-\alpha) g^T d + \frac{1}{2}(1-\alpha^2) d^T H d \leq 0.$$

Since $1-\alpha^2 = (1-\alpha)(1+\alpha)$, dividing both sides by $1-\alpha > 0$ and then taking $\alpha \uparrow 1$ prove the first inequality in (5). Since $d_{\mathcal{J}} \in \mathrm{Null}(A_{\mathcal{J}})$ so that $d_{\mathcal{J}} = B_{\mathcal{J}} y$ for some vector y, we have

$$d^T H d = y^T B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} y \geq \|y\|^2 \lambda_{\min}(B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}}) = \|d\|^2 \lambda_{\min}(B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}}),$$

where the second equality uses $B_{\mathcal{J}}^T B_{\mathcal{J}} = I$. This proves the second inequality in (5). ■

We next choose a stepsize $\alpha > 0$ so that $x' = x + \alpha d$ achieves sufficient descent, and re-iterate. We now describe formally the block-coordinate gradient descent (abbreviated as CGD) method.

CGD method:

Choose $x^0 \in X$. For $k = 0, 1, 2, ...$, generate x^{k+1} from x^k according to the iteration:

1. Choose a nonempty $\mathcal{J}^k \subseteq \mathcal{N}$ and a symmetric $H^k \in \Re^{n\times n}$ with $B_{\mathcal{J}^k}^T H_{\mathcal{J}^k\mathcal{J}^k}^k B_{\mathcal{J}^k} \succ 0$.

2. Solve (4) with $x = x^k$, $\mathcal{J} = \mathcal{J}^k$, $H = H^k$ to obtain $d^k = d_{H^k}(x^k;\mathcal{J}^k)$.

3. Set $x^{k+1} = x^k + \alpha^k d^k$, with $\alpha^k > 0$.

For the SVM QP (1), $x^0 = 0$ is a popular choice. In general, x^0 can be found by solving, say,

$$\min_{x\in\Re^n} \left\{ \|Ax - b\|^2 \;\middle|\; l \leq x \leq u \right\}$$

using the CGD method, starting at l or u. Notice that $B_{\mathcal{J}^k}$ is not needed if $H^k \succ 0$. Various stepsize rules for smooth optimization [2, 12, 32] can be adapted to choose α^k. The following adaptation of the Armijo rule [2, page 225], based on Lemma 2.1 and [44, Section 2], is simple and seems effective in theory and practice.

Armijo rule:

Choose $\alpha^k_{\text{init}} > 0$ and let α^k be the largest element of $\{\alpha^k_{\text{init}}\beta^j\}_{j=0,1,\ldots}$ satisfying

$$f(x^k + \alpha^k d^k) \le f(x^k) + \sigma\alpha^k\Delta^k \quad \text{and} \quad x^k + \alpha^k d^k \in X, \tag{6}$$

where $0 < \beta < 1$, $0 < \sigma < 1$, $0 \le \theta < 1$, and

$$\Delta^k \stackrel{\text{def}}{=} \nabla f(x^k)^T d^k + \theta {d^k}^T H^k d^k. \tag{7}$$

Since $B^T_{\mathcal{J}^k} H^k_{\mathcal{J}^k\mathcal{J}^k} B_{\mathcal{J}^k} \succ 0$ and $0 \le \theta < 1$, we see from Lemma 2.1 that

$$f(x^k + \alpha d^k) = f(x^k) + \alpha\nabla f(x^k)^T d^k + o(\alpha) \le f(x^k) + \alpha\Delta^k + o(\alpha) \quad \forall\alpha \in (0, 1],$$

and $\Delta^k \le (\theta - 1){d^k}^T H^k d^k < 0$ whenever $d^k \ne 0$. Since $0 < \sigma < 1$, this shows that α^k given by the Armijo rule is well defined and positive. This rule, like that for sequential quadratic programming methods [2, 12, 32], requires only function evaluations. And, by choosing α^k_{init} based on the previous stepsize α^{k-1}, the number of function evaluations can be kept small in practice. Notice that Δ^k increases with θ. Thus, larger stepsizes will be accepted if we choose either σ near 0 or θ near 1. The minimization rule or the limited minimization rule [2, Section 2.2.1] (also see (23), (24)) can be used instead of the Armijo rule if the minimization is relatively inexpensive, such as for a QP.

For theoretical and practical efficiency, the working set $\mathcal{J}^k$ must be chosen judiciously so to ensure global convergence while balancing between convergence speed and the computational cost per iteration. Let us denote the optimal value of the direction subproblem (4) by

$$q_H(x; \mathcal{J}) \stackrel{\text{def}}{=} \left\{\nabla f(x)^T d + \frac{1}{2} d^T H d\right\}_{d = d_H(x;\mathcal{J})}. \tag{8}$$

Intuitively, $q_H(x; \mathcal{J})$ is the predicted descent when x is moved along the direction $d_H(x; \mathcal{J})$. We will choose the working set $\mathcal{J}^k$ to achieve sufficient predicted descent, i.e.,

$$q_{D^k}(x^k; \mathcal{J}^k) \le \upsilon\, q_{D^k}(x^k; \mathcal{N}), \tag{9}$$

where $D^k \succ 0$ (typically diagonal) and $0 < \upsilon \le 1$. (In fact, it suffices that $B^T_{\mathcal{N}} D^k B_{\mathcal{N}} \succ 0$ for our analysis.) This working set choice is motivated by the *Gauss-Southwell-q* rule in [44], which has good convergence properties in theory and in practice. Specifically, (9) is identical to [44, Equation (16)] while (8) is different from but analogous to [44, Equation (15)]. We will discuss in Section 6 how to efficiently find a "small" working set $\mathcal{J}^k$ that satisfies (9) for some υ. Small $\mathcal{J}^k$ has the advantages that (i) d^k is easier to compute, (ii) H^k

can be chosen to better approximate $\nabla^2 f(x^k)$, and (iii) $\nabla f(x^{k+1})$ may be updated cheaply. Specifically, if f is quadratic or has the partially separable form

$$f(x) = h(Ex) + q^T x,$$

where $h : \Re^p \to (-\infty, \infty]$ is block-separable with $O(1)$ size blocks, $q \in \Re^n$, and each column of $E \in \Re^{p\times n}$ has $O(1)$ nonzeros, then $\nabla f(x^{k+1})$ can be updated from $\nabla f(x^k)$ in $O(|\mathcal{J}^k|n)$ operations; see the last paragraph of Section 7 for further discussions.

For the SVM QP (1), one choice of $\mathcal{J}^k$ that satisfies (9) with $\upsilon = 1/(n-\ell+1)$ is

$$\mathcal{J}^k \in \arg\min_{\mathcal{J}':|\mathcal{J}'|\le \ell} \left\{ \begin{array}{ll} \min_d & \nabla f(x^k)^T d + \frac{1}{2} d^T \mathrm{diag}(Q) d \\ \text{s.t.} & a^T d = 0, \\ & 0 \le x_j^k + d_j \le C, \ j \in \mathcal{J}', \\ & d_j = 0, \ j \notin \mathcal{J}', \end{array} \right\} \tag{10}$$

where $\ell \in \{\mathrm{rank}(A)+1, \ldots, n\}$; see Proposition 6.1. However, no fast way to find such $\mathcal{J}^k$ is known.

3 Technical preliminaries

In this section we study properties of the search direction $d_H(x;\mathcal{J})$ and the corresponding predicted descent $q_H(x;\mathcal{J})$. These will be useful for analyzing the global convergence and asymptotic convergence rate of the CGD method.

We say that an $x \in X$ is a *a stationary point* of f over X if $\nabla f(x)^T(y-x) \ge 0$ for all $y \in X$. This is equivalent to $d_D(x;\mathcal{N}) = 0$ for any $D \succ 0$; see [2, pages 229, 230].

The next lemma shows that $\|d_H(x;\mathcal{J})\|$ changes not too fast with the quadratic coefficients H. It will be used to prove Theorems 4.1 and 5.1. Recall that $B_{\mathcal{J}}$ is a matrix whose columns form an orthonormal basis for $\mathrm{Null}(A_{\mathcal{J}})$.

Lemma 3.1 *Fix any $x \in X$, nonempty $\mathcal{J} \subseteq \mathcal{N}$, and symmetric matrices $H, \tilde{H} \in \Re^{n\times n}$ satisfying $C \succ 0$ and $\tilde{C} \succ 0$, where $C = B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}}$ and $\tilde{C} = B_{\mathcal{J}}^T \tilde{H}_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}}$. Let $d = d_H(x;\mathcal{J})$ and $\tilde{d} = d_{\tilde{H}}(x;\mathcal{J})$. Then*

$$\|\tilde{d}\| \le \frac{1 + \lambda_{\max}(Q) + \sqrt{1 - 2\lambda_{\min}(Q) + \lambda_{\max}(Q)^2}}{2} \frac{\lambda_{\max}(C)}{\lambda_{\min}(\tilde{C})} \|d\|, \tag{11}$$

where $Q = C^{-1/2}\tilde{C}C^{-1/2}$.

Proof. Since $d_j = \tilde{d}_j = 0$ for all $j \notin \mathcal{J}$, it suffices to prove the lemma for the case of $\mathcal{J} = \mathcal{N}$. Let $g = \nabla f(x)$. By the definition of d and $\tilde{d}$ and [38, Theorem 8.15],

$$d \in \arg\min_u \{(g + Hd)^T u \mid x + u \in X\},$$
$$\tilde{d} \in \arg\min_u \{(g + \tilde{H}\tilde{d})^T u \mid x + u \in X\}.$$

Thus

$$\begin{aligned}(g+Hd)^T d &\leq (g+Hd)^T \tilde{d},\\ (g+\tilde{H}\tilde{d})^T \tilde{d} &\leq (g+\tilde{H}\tilde{d})^T d.\end{aligned}$$

Adding the above two inequalities and rearranging terms yield

$$d^T H d - d^T (H+\tilde{H})\tilde{d} + \tilde{d}^T \tilde{H}\tilde{d} \leq 0.$$

Since $d, \tilde{d} \in \text{Null}(A)$, we have $d = B_{\mathcal{N}} y$ and $\tilde{d} = B_{\mathcal{N}} \tilde{y}$ for some vectors $y, \tilde{y}$. Substituting these into the above inequality and using the definitions of $C, \tilde{C}$ yield

$$y^T C y - y^T (C+\tilde{C})\tilde{y} + \tilde{y}^T \tilde{C}\tilde{y} \leq 0.$$

Then proceeding as in the proof of [44, Lemma 3.2] and using $\|d\| = \|y\|$, $\|\tilde{d}\| = \|\tilde{y}\|$ (since $B_{\mathcal{N}}^T B_{\mathcal{N}} = I$), we obtain (11). ∎

The next lemma gives a sufficient condition for the stepsize to satisfy the Armijo descent condition (6). This lemma will be used to prove Theorem 4.1(d). Its proof is similar to that of [44, Lemma 3.4(b)] and is included for completeness.

Lemma 3.2 *Suppose f satisfies*

$$\|\nabla f(y) - \nabla f(z)\| \leq L\|y-z\| \quad \forall y,\, z \in X, \tag{12}$$

for some $L \geq 0$. Fix any $x \in X$, nonempty $\mathcal{J} \subseteq \mathcal{N}$, and symmetric matrix $H \in \Re^{n\times n}$ satisfying $B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} \succeq \underline{\lambda} I$ with $\underline{\lambda} > 0$. Then, for any $\sigma \in (0,1)$, $\theta \in [0,1)$, and $0 \leq \alpha \leq 2\underline{\lambda}(1-\sigma+\sigma\theta)/L$ with $x + \alpha d \in X$, we have

$$f(x+\alpha d) - f(x) \leq \sigma\alpha(g^T d + \theta d^T H d), \tag{13}$$

where $d = d_H(x;\mathcal{J})$ and $g = \nabla f(x)$.

Proof. For any $\alpha \geq 0$ with $x + \alpha d \in X$, we have from the Cauchy-Schwarz inequality that

$$\begin{aligned} f(x+\alpha d) - f(x) &= \alpha g^T d + \int_0^1 (\nabla f(x+t\alpha d) - \nabla f(x))^T (\alpha d)\, dt \\ &\leq \alpha g^T d + \alpha \int_0^1 \|\nabla f(x+t\alpha d) - \nabla f(x)\| \|d\|\, dt \\ &\leq \alpha g^T d + \alpha^2 \frac{L}{2}\|d\|^2 \\ &= \alpha(g^T d + \theta d^T H d) - \alpha\theta d^T H d + \alpha^2 \frac{L}{2}\|d\|^2, \end{aligned} \tag{14}$$

where the third step uses (12) and $x + t\alpha d \in X$ when $0 \le t \le 1$. Since $\underline{\lambda}\|d\|^2 \le d^T H d$ by Lemma 2.1, if in addition $\alpha \le 2\underline{\lambda}(1-\sigma+\sigma\theta)/L$, then

$$\begin{aligned} \alpha\frac{L}{2}\|d\|^2 - \theta d^T H d &\le (1-\sigma+\sigma\theta)d^T H d - \theta d^T H d \\ &= (1-\sigma)(1-\theta)d^T H d \\ &\le -(1-\sigma)(\nabla f(x)^T d + \theta d^T H d), \end{aligned}$$

where the third step uses (5) in Lemma 2.1. This together with (14) yields (13). ■

The next lemma shows that $\nabla f(x)^T(x' - \bar{x})$ is bounded above by a weighted sum of $\|x - \bar{x}\|^2$ and $-q_D(x;\mathcal{J})$, where $x' = x + \alpha d$, $d = d_H(x;\mathcal{J})$, and $\mathcal{J}$ satisfies a condition analogous to (9). This lemma, which is new, will be needed to prove Theorem 5.1.

Lemma 3.3 *Fix any $x \in X$, nonempty $\mathcal{J} \subseteq \mathcal{N}$, and symmetric matrices $H, D \in \Re^{n\times n}$ satisfying $B_{\mathcal{J}}^T H_{\mathcal{J}\mathcal{J}} B_{\mathcal{J}} \succ 0$, $\bar{\delta} I \succeq D \succ 0$, and*

$$q_D(x;\mathcal{J}) \le \upsilon\, q_D(x;\mathcal{N}), \tag{15}$$

with $\bar{\delta} > 0$, $0 < \upsilon \le 1$. Then, for any $\bar{x} \in X$ and $\alpha \ge 0$, we have

$$g^T(x' - \bar{x}) \le \frac{\bar{\delta}}{2}\|\bar{x} - x\|^2 - \frac{1}{\upsilon}q_D(x;\mathcal{J}), \tag{16}$$

where $d = d_H(x;\mathcal{J})$, $g = \nabla f(x)$, and $x' = x + \alpha d$.

Proof. Since $\bar{x} - x$ is a feasible solution of the minimization subproblem (4) corresponding to $\mathcal{N}$ and D, we have

$$q_D(x;\mathcal{N}) \le g^T(\bar{x} - x) + \frac{1}{2}(\bar{x} - x)^T D(\bar{x} - x).$$

Since $\bar{\delta} I \succeq D \succ 0$, we have $0 \le (\bar{x} - x)^T D(\bar{x} - x) \le \bar{\delta}\|\bar{x} - x\|^2$. This together with (15) yields

$$\frac{1}{\upsilon}q_D(x;\mathcal{J}) \le g^T(\bar{x} - x) + \frac{\bar{\delta}}{2}\|\bar{x} - x\|^2.$$

Rearranging terms, we have

$$g^T(x - \bar{x}) \le \frac{\bar{\delta}}{2}\|\bar{x} - x\|^2 - \frac{1}{\upsilon}q_D(x;\mathcal{J}). \tag{17}$$

By the definition of d and Lemma 2.1, we have $g^T d \le 0$. Since $\alpha \ge 0$, this implies $\alpha g^T d \le 0$. Adding this to (17) yields (16). ■

4 Global Convergence Analysis

In this section we analyze the global convergence of the CGD method under the following reasonable assumption on our choice of H^k.

Assumption 1 $\bar{\lambda} I \succeq B_{\mathcal{J}^k}^T H_{\mathcal{J}^k \mathcal{J}^k}^k B_{\mathcal{J}^k} \succeq \underline{\lambda} I$ *for all* k, *where* $0 < \underline{\lambda} \le \bar{\lambda}$.

First, we have the following lemma relating the optimal solution and the optimal objective value of (4) when $\mathcal{J} = \mathcal{J}^k$ and $H = D^k$. This lemma will be used to prove Theorem 4.1(c).

Lemma 4.1 *For any* $x^k \in X$, *nonempty* $\mathcal{J}^k \subseteq \mathcal{N}$, *and* $\bar{\delta} I \succeq D^k \succeq \underline{\delta} I$ $(0 < \underline{\delta} \le \bar{\delta})$, $k = 0, 1, \ldots$, *if* $\{x^k\}$ *is convergent, then* $\{d_{D^k}(x^k; \mathcal{J}^k)\} \to 0$ *if and only if* $\{q_{D^k}(x^k; \mathcal{J}^k)\} \to 0$.

Proof. Let $\{x^k\}$ be a convergent sequence in X. Then $\{\nabla f(x^k)\}$ is convergent by the continuity of ∇f. If $\{d_{D^k}(x^k; \mathcal{J}^k)\} \to 0$, then (8) and the boundedness of $\{D^k\}$ imply $\{q_{D^k}(x^k; \mathcal{J}^k)\} \to 0$. Conversely, we have from (8) and (5) with $H = D^k$ that $q_{D^k}(x^k; \mathcal{J}^k) \le -\frac{1}{2} d_{D^k}(x^k; \mathcal{J}^k)^T D^k d_{D^k}(x^k; \mathcal{J}^k) \le -\frac{\delta}{2} \|d_{D^k}(x^k; \mathcal{J}^k)\|^2$ for all k. Thus if $\{q_{D^k}(x^k; \mathcal{J}^k)\} \to 0$, then $\{d_{D^k}(x^k; \mathcal{J}^k)\} \to 0$. ∎

Using Lemmas 2.1, 3.1, 3.2, and 4.1, we have the following global convergence result, under Assumption 1, for the CGD method with $\{\mathcal{J}^k\}$ chosen by the Gauss-Southwell rule (9) and $\{\alpha^k\}$ chosen by the Armijo rule (6). Its proof adapts the analysis of gradient methods for unconstrained smooth optimization [2, pages 43-45] to handle constraints and block-coordinate updating.

Theorem 4.1 *Let* $\{x^k\}$, $\{\mathcal{J}^k\}$, $\{H^k\}$, $\{d^k\}$ *be sequences generated by the CGD method under Assumption 1, where* $\{\alpha^k\}$ *is chosen by the Armijo rule with* $\inf_k \alpha^k_{\text{init}} > 0$. *Then the following results hold.*

(a) $\{f(x^k)\}$ *is nonincreasing and* Δ^k *given by* (7) *satisfies*

$$-\Delta^k \ge (1-\theta) {d^k}^T H^k d^k \ge (1-\theta)\underline{\lambda} \|d^k\|^2 \quad \forall k, \tag{18}$$

$$f(x^{k+1}) - f(x^k) \le \sigma \alpha^k \Delta^k \le 0 \quad \forall k. \tag{19}$$

(b) *If* $\{x^k\}_{\mathcal{K}}$ *is a convergent subsequence of* $\{x^k\}$, *then* $\{\alpha^k \Delta^k\} \to 0$ *and* $\{d^k\}_{\mathcal{K}} \to 0$. *If in addition* $\bar{\delta} I \succeq D^k \succeq \underline{\delta} I$ *for all* k, *where* $0 < \underline{\delta} \le \bar{\delta}$, *then* $\{d_{D^k}(x^k; \mathcal{J}^k)\}_{\mathcal{K}} \to 0$.

(c) *If* $\{\mathcal{J}^k\}$ *is chosen by* (9) *and* $\bar{\delta} I \succeq D^k \succeq \underline{\delta} I$ *for all* k, *where* $0 < \underline{\delta} \le \bar{\delta}$, *then every cluster point of* $\{x^k\}$ *is a stationary point of* (3).

(d) *If f satisfies* (12) *for some $L \geq 0$, then $\inf_k \alpha^k > 0$. If $\lim_{k\to\infty} f(x^k) > -\infty$ also, then $\{\Delta^k\} \to 0$ and $\{d^k\} \to 0$.*

Proof. (a) The first inequality in (18) follows from (7) and Lemma 2.1. The second inequality follows from $0 \leq \theta < 1$, Lemma 2.1, and $\lambda_{\min}(B_{\mathcal{J}^k}^T H^k_{\mathcal{J}^k\mathcal{J}^k} B_{\mathcal{J}^k}) \geq \bar{\lambda}$. Since $x^{k+1} = x^k + \alpha^k d^k$ and α^k is chosen by the Armijo rule (6), we have (19) and hence $\{f(x^k)\}$ is nonincreasing.

(b) Let $\{x^k\}_{\mathcal{K}}$ ($\mathcal{K} \subseteq \{0,1,\ldots\}$) be a subsequence of $\{x^k\}$ converging to some $\bar{x}$. Since f is smooth, $f(\bar{x}) = \lim_{\substack{k\to\infty \\ k\in\mathcal{K}}} f(x^k)$. Since $\{f(x^k)\}$ is nonincreasing, this implies that $\{f(x^k)\} \downarrow f(\bar{x})$. Hence, $\{f(x^k) - f(x^{k+1})\} \to 0$. Then, by (19),

$$\{\alpha^k \Delta^k\} \to 0. \tag{20}$$

Suppose that $\{d^k\}_{\mathcal{K}} \not\to 0$. By passing to a subsequence if necessary, we can assume that, for some $\delta > 0$, $\|d^k\| \geq \delta$ for all $k \in \mathcal{K}$. Then, by (18) and (20), $\{\alpha^k\}_{\mathcal{K}} \to 0$. Since $\inf_k \alpha^k_{\text{init}} > 0$, there exists some index $\bar{k} \geq 0$ such that $\alpha^k < \alpha^k_{\text{init}}$ and $\alpha^k \leq \beta$ for all $k \in \mathcal{K}$ with $k \geq \bar{k}$. Since $x^k + d^k \in X$ and X is convex, the latter implies that $x^k + (\alpha^k/\beta)d^k \in X$ for all $k \in \mathcal{K}$ with $k \geq \bar{k}$. Since α^k is chosen by the Armijo rule, this in turn implies that

$$f(x^k + (\alpha^k/\beta)d^k) - f(x^k) > \sigma(\alpha^k/\beta)\Delta^k \quad \forall k \in \mathcal{K},\ k \geq \bar{k}.$$

Using the definition of Δ^k, we can rewrite this as

$$-(1-\sigma)\Delta^k + \theta {d^k}^T H^k d^k < \frac{f(x^k + (\alpha^k/\beta)d^k) - f(x^k)}{\alpha^k/\beta} - \nabla f(x^k)^T d^k \quad \forall k \in \mathcal{K},\ k \geq \bar{k}.$$

By (18), the left-hand side is greater than or equal to $((1-\sigma)(1-\theta)+\theta)\underline{\lambda}\|d^k\|^2$, so dividing both sides by $\|d^k\|$ yields

$$((1-\sigma)(1-\theta)+\theta)\underline{\lambda}\|d^k\| < \frac{f(x^k + \hat{\alpha}^k d^k/\|d^k\|) - f(x^k)}{\hat{\alpha}^k} - \frac{\nabla f(x^k)^T d^k}{\|d^k\|} \quad \forall k \in \mathcal{K},\ k \geq \bar{k}, \tag{21}$$

where we let $\hat{\alpha}^k = \alpha^k\|d^k\|/\beta$. By (18), $-\alpha^k\Delta^k \geq (1-\theta)\underline{\lambda}\alpha^k\|d^k\|^2 \geq (1-\theta)\underline{\lambda}\alpha^k\|d^k\|\delta$ for all $k \in \mathcal{K}$, so (20) and $(1-\theta)\underline{\lambda} > 0$ imply $\{\alpha^k\|d^k\|\}_{\mathcal{K}} \to 0$ and hence $\{\hat{\alpha}^k\}_{\mathcal{K}} \to 0$. Also, since $\{d^k/\|d^k\|\}_{\mathcal{K}}$ is bounded, by passing to a subsequence if necessary, we can assume that $\{d^k/\|d^k\|\}_{\mathcal{K}} \to$ some $\bar{d}$. Taking the limit as $k \in \mathcal{K}, k \to \infty$ in the inequality (21) and using the smoothness of f, we obtain

$$0 < ((1-\sigma)(1-\theta)+\theta)\underline{\lambda}\delta \leq \nabla f(\bar{x})^T\bar{d} - \nabla f(\bar{x})^T\bar{d} = 0,$$

a clear contradiction. Thus $\{d^k\}_{\mathcal{K}} \to 0$.

Suppose that, in addition, $\bar{\delta}I \succeq D^k \succeq \underline{\delta}I$ for all k. Let $\tilde{C}^k = B_{\mathcal{J}^k}^T D^k_{\mathcal{J}^k\mathcal{J}^k} B_{\mathcal{J}^k}$ and $C^k = B_{\mathcal{J}^k}^T H^k_{\mathcal{J}^k\mathcal{J}^k} B_{\mathcal{J}^k}$. Then, for each k, $\bar{\delta}I \succeq \tilde{C}^k \succeq \underline{\delta}I$ (since $B_{\mathcal{J}^k}^T B_{\mathcal{J}^k} = I$) as well as $\bar{\lambda}I \succeq C^k \succeq \underline{\lambda}I$. Then

$$\frac{\bar{\delta}}{\underline{\lambda}}I \succeq \bar{\delta}(C^k)^{-1} \succeq (C^k)^{-1/2}\tilde{C}^k(C^k)^{-1/2} \succeq \underline{\delta}(C^k)^{-1} \succeq \frac{\underline{\delta}}{\bar{\lambda}}I,$$

so (11) in Lemma 3.1 yields

$$\|d_{D^k}(x^k;\mathcal{J}^k)\| \le \frac{1+\bar\delta/\underline\lambda+\sqrt{1-2\underline\delta/\bar\lambda+(\bar\delta/\underline\lambda)^2}}{2}\ \frac{\bar\lambda}{\underline\delta}\ \|d^k\|. \tag{22}$$

Since $\{d^k\}_\mathcal{K} \to 0$, this implies $\{d_{D^k}(x^k;\mathcal{J}^k)\}_\mathcal{K} \to 0$.

(c) Suppose that $\{\mathcal{J}^k\}$ is chosen by (9) and $\bar\delta I \succeq D^k \succeq \underline\delta I$ for all k and $\bar x$ is a cluster point of $\{x^k\}$. Let $\{x^k\}_\mathcal{K}$ be a subsequence of $\{x^k\}$ converging to $\bar x$. By (b), $\{d^k\}_\mathcal{K} \to 0$ and $\{d_{D^k}(x^k;\mathcal{J}^k)\}_\mathcal{K} \to 0$. By Lemma 4.1, $\{q_{D^k}(x^k;\mathcal{J}^k)\}_\mathcal{K} \to 0$. Since $\mathcal{J}^k$ satisfies (9), this implies that $\{q_{D^k}(x^k;\mathcal{N})\}_\mathcal{K} \to 0$. This together with Lemma 4.1 yields $\{d_{D^k}(x^k;\mathcal{N})\}_\mathcal{K} \to 0$.

By Lemma 3.1 with $\mathcal{J} = \mathcal{N}$, $H = D^k$, and $\tilde H = I$, we have

$$\|d_I(x^k;\mathcal{N})\| \le \frac{1+1/\underline\delta+\sqrt{1-2/\bar\delta+(1/\underline\delta)^2}}{2}\ \bar\delta\ \|d_{D^k}(x^k;\mathcal{N})\| \quad \forall k.$$

Hence $\{d_I(x^k;\mathcal{N})\}_\mathcal{K} \to 0$. A continuity argument then yields that $d_I(\bar x;\mathcal{N}) = 0$, so $\bar x$ is a stationary point of (3).

(d) Since α^k is chosen by the Armijo rule, either $\alpha^k = \alpha^k_{\text{init}}$ or else, by Lemma 3.2 and $x^k + d^k \in X$, $\alpha^k/\beta > \min\{1, 2\underline\lambda(1-\sigma+\sigma\theta)/L\}$. Since $\inf_k \alpha^k_{\text{init}} > 0$, this implies $\inf_k \alpha^k > 0$. If $\lim_{k\to\infty} f(x^k) > -\infty$ also, then this and (19) imply $\{\Delta^k\} \to 0$, which together with (18) imply $\{d^k\} \to 0$. ∎

Similar to the observation in [2, page 45], Theorem 4.1 readily extends to any stepsize rule that yields a larger descent than the Armijo rule at each iteration.

Corollary 4.1 *Theorem 4.1 still holds if in the CGD method the iterates are instead updated by $x^{k+1} = x^k + \tilde\alpha^k d^k$, where $\tilde\alpha^k \ge 0$ satisfies $f(x^k + \tilde\alpha^k d^k) \le f(x^k + \alpha^k d^k)$ and $x^k + \tilde\alpha^k d^k \in X$ for $k = 0, 1, \ldots$, and $\{\alpha^k\}$ is chosen by the Armijo rule with* $\inf_k \alpha^k_{\text{init}} > 0$.

Proof. It is readily seen using $f(x^{k+1}) \le f(x^k + \alpha^k d^k)$ that Theorem 4.1(a) holds. The proofs of Theorem 4.1(b)–(d) remain unchanged. ∎

For example, $\tilde\alpha^k$ may be generated by the minimization rule:

$$\tilde\alpha^k \in \arg\min_{\alpha\ge 0}\{f(x^k+\alpha d^k) \mid x^k + \alpha d^k \in X\} \tag{23}$$

or by the limited minimization rule:

$$\tilde\alpha^k \in \arg\min_{0\le\alpha\le s}\{f(x^k+\alpha d^k) \mid x^k + \alpha d^k \in X\}, \tag{24}$$

where $0 < s < \infty$. The latter stepsize rule yields a larger descent than the Armijo rule with $\alpha^k_{\text{init}} = s$. We will use the minimization rule in our numerical tests on the SVM QP; see Section 7.

5 Convergence rate analysis

In this section we analyze the asymptotic convergence rate of the CGD method under the following reasonable assumption; see [30]. In what follows, $\bar{X}$ denotes the set of stationary points of (3) and

$$\mathrm{dist}(x, \bar{X}) \stackrel{\mathrm{def}}{=} \min_{\bar{x} \in \bar{X}} \|x - \bar{x}\| \qquad \forall x \in \Re^n.$$

Assumption 2 **(a)** *$\bar{X} \neq \emptyset$ and, for any $\zeta \geq \min_{x \in X} f(x)$, there exist scalars $\tau > 0$ and $\epsilon > 0$ such that*

$$\mathrm{dist}(x, \bar{X}) \leq \tau \|d_I(x; \mathcal{N})\| \quad \textit{whenever} \quad x \in X,\ f(x) \leq \zeta,\ \|d_I(x; \mathcal{N})\| \leq \epsilon.$$

(b) *There exists a scalar $\rho > 0$ such that*

$$\|x - y\| \geq \rho \quad \textit{whenever} \quad x \in \bar{X},\ y \in \bar{X},\ f(x) \neq f(y).$$

Assumption 2 is identical to Assumptions A and B in [30]. Assumption 2(b) says that the isocost surfaces of f restricted to the solution set $\bar{X}$ are "properly separated." Assumption 2(b) holds automatically if f is a convex function. It also holds if f is quadratic and X is polyhedral [29, Lemma 3.1]. Assumption 2(a) is a local Lipschitzian error bound assumption, saying that the distance from x to $\bar{X}$ is locally in the order of the norm of the residual at x. Error bounds of this kind have been extensively studied.

Since X is polyhedral, we immediately have from [30, Theorem 2.1] the following sufficient conditions for Assumption 2(a) to hold. In particular, Assumption 2(a) and (b) hold for (1) and, more generally, any QP [29, 30].

Proposition 5.1 *Suppose that $\bar{X} \neq \emptyset$ and any of the following conditions hold.*

C1 *f is strongly convex and ∇f is Lipschitz continuous on X (i.e., (12) holds for some $L \geq 0$).*

C2 *f is quadratic.*

C3 *$f(x) = g(Ex) + q^T x$ for all $x \in \Re^n$, where $E \in \Re^{m \times n}$, $q \in \Re^n$, and g is a strongly convex differentiable function on $\Re^m$ with ∇g Lipschitz continuous on $\Re^m$.*

C4 *$f(x) = \max_{y \in Y}\{(Ex)^T y - g(y)\} + q^T x$ for all $x \in \Re^n$, where Y is a polyhedral set in $\Re^m$, $E \in \Re^{m \times n}$, $q \in \Re^n$, and g is a strongly convex differentiable function on $\Re^m$ with ∇g Lipschitz continuous on $\Re^m$.*

Then Assumption 2(a) holds.

Using Theorem 4.1 and Lemmas 2.1, 3.1, and 3.3, we have the following linear convergence result, under Assumptions 1, 2, and (12), for the CGD method with $\{\mathcal{J}^k\}$ chosen by (9) and $\{\alpha^k\}$ chosen by the Armijo rule. Its proof adapts that of [44, Theorem 5.2] to constrained problems. To our knowledge, this is the first linear convergence result for a block-coordinate update method for general linearly constrained smooth optimization. Moreover, it does not assume f is strongly convex or the stationary points satisfy strict complementarity.

Theorem 5.1 *Assume that f satisfies (12) for some $L \geq 0$ and Assumption 2. Let $\{x^k\}$, $\{H^k\}$, $\{d^k\}$ be sequences generated by the CGD method satisfying Assumption 1, where $\{\mathcal{J}^k\}$ is chosen by (9), $\bar{\delta}I \succeq D^k \succeq \underline{\delta}I$ for all k ($0 < \underline{\delta} \leq \bar{\delta}$), and $\{\alpha^k\}$ is chosen by the Armijo rule with $\sup_k \alpha^k_{\text{init}} < \infty$ and $\inf_k \alpha^k_{\text{init}} > 0$. Then either $\{f(x^k)\} \downarrow -\infty$ or $\{f(x^k)\}$ converges at least Q-linearly and $\{x^k\}$ converges at least R-linearly to a point in $\bar{X}$.*

Proof. For each $k = 0, 1, \ldots$, (7) and $d^k = d_{H^k}(x^k; \mathcal{J}^k)$ imply that

$$\begin{aligned}
\Delta^k + \left(\frac{1}{2} - \theta\right) {d^k}^T H^k d^k &= {g^k}^T d^k + \frac{1}{2}{d^k}^T H^k d^k \\
&\leq {g^k}^T \tilde{d}^k + \frac{1}{2}(\tilde{d}^k)^T H^k \tilde{d}^k \\
&= q_{D^k}(x^k; \mathcal{J}^k) + \frac{1}{2}(\tilde{d}^k)^T (H^k - D^k)\tilde{d}^k \\
&\leq q_{D^k}(x^k; \mathcal{J}^k) + \omega \|d^k\|^2, \qquad (25)
\end{aligned}$$

where we let $g^k = \nabla f(x^k)$ and $\tilde{d}^k = d_{D^k}(x^k; \mathcal{J}^k)$, and the last step uses (22) and $(\tilde{d}^k)^T(H^k - D^k)\tilde{d}^k \leq (\bar{\lambda} - \underline{\delta})\|\tilde{d}^k\|^2$. Here, $\omega \in \Re$ is a constant depending on $\bar{\lambda}, \underline{\lambda}, \bar{\delta}, \underline{\delta}$ only. Also, by (8) and Lemma 2.1 with $\mathcal{J} = \mathcal{N}$, $H = D^k$, we have

$$\begin{aligned}
q_{D^k}(x^k; \mathcal{N}) &= \left({g^k}^T d + \frac{1}{2} d^T D^k d\right)_{d = d_{D^k}(x^k;\mathcal{N})} \\
&\leq \left(-\frac{1}{2} d^T D^k d\right)_{d = d_{D^k}(x^k;\mathcal{N})} \\
&\leq -\frac{\underline{\delta}}{2}\|d_{D^k}(x^k; \mathcal{N})\|^2 \quad \forall k, \qquad (26)
\end{aligned}$$

where the last inequality uses $D^k \succeq \underline{\delta}I$.

By Theorem 4.1(a), $\{f(x^k)\}$ is nonincreasing. Thus either $\{f(x^k)\} \downarrow -\infty$ or $\lim_{k\to\infty} f(x^k) > -\infty$. Suppose the latter. Since α^k is chosen by the Armijo rule with $\inf_k \alpha^k_{\text{init}} > 0$, Theorem 4.1(d) implies $\{\Delta^k\} \to 0$ and $\{d^k\} \to 0$. Since $\{H^k\}$ is bounded by Assumption 1, we obtain from (25) that $0 \leq \lim_{k\to\infty} \inf q_{D^k}(x^k; \mathcal{J}^k)$. Then (9) and (26) yield $\{d_{D^k}(x^k; \mathcal{N})\} \to 0$.

By Lemma 3.1 with $\mathcal{J} = \mathcal{N}$, $H = D^k$ and $\tilde{H} = I$, we have

$$\|d_I(x^k; \mathcal{N})\| \leq \frac{1 + 1/\underline{\delta} + \sqrt{1 - 2/\bar{\delta} + (1/\underline{\delta})^2}}{2} \bar{\delta}\, \|d_{D^k}(x^k; \mathcal{N})\| \quad \forall k. \qquad (27)$$

Hence $\{d_I(x^k;\mathcal{N})\} \to 0$. Since $\{f(x^k)\}$ is nonincreasing, so that $f(x^k) \le f(x^0)$, as well as $x^k \in X$, for all k. Then, by Assumption 2(a), there exist $\bar{k}$ and $\tau > 0$ such that

$$\|x^k - \bar{x}^k\| \le \tau \|d_I(x^k;\mathcal{N})\| \quad \forall k \ge \bar{k}, \tag{28}$$

where $\bar{x}^k \in \bar{X}$ satisfies $\|x^k - \bar{x}^k\| = \text{dist}(x^k, \bar{X})$. Since $\{d_I(x^k;\mathcal{N})\} \to 0$, this implies $\{x^k - \bar{x}^k\} \to 0$. Since $\{x^{k+1} - x^k\} = \{\alpha^k d^k\} \to 0$, this and Assumption 2(b) imply that $\{\bar{x}^k\}$ eventually settles down at some isocost surface of f, i.e., there exist an index $\hat{k} \ge \bar{k}$ and a scalar $\bar{v}$ such that

$$f(\bar{x}^k) = \bar{v} \quad \forall k \ge \hat{k}. \tag{29}$$

Fix any index $k \ge \hat{k}$. Since $\bar{x}^k$ is a stationary point of f over X, we have

$$\nabla f(\bar{x}^k)^T (x^k - \bar{x}^k) \ge 0.$$

We also have from the Mean Value Theorem that

$$f(x^k) - f(\bar{x}^k) = \nabla f(\psi^k)^T (x^k - \bar{x}^k),$$

for some ψ^k lying on the line segment joining x^k with $\bar{x}^k$. Since $x^k, \bar{x}^k$ lie in the convex set X, so does ψ^k. Combining these two relations and using (29), we obtain

$$\begin{aligned} \bar{v} - f(x^k) &\le (\nabla f(\bar{x}^k) - \nabla f(\psi^k))^T (x^k - \bar{x}^k) \\ &\le \|\nabla f(\bar{x}^k) - \nabla f(\psi^k)\| \|x^k - \bar{x}^k\| \\ &\le L\|x^k - \bar{x}^k\|^2, \end{aligned}$$

where the last inequality uses (12) and $\|\psi^k - \bar{x}^k\| \le \|x^k - \bar{x}^k\|$. This together with $\{x^k - \bar{x}^k\} \to 0$ proves that

$$\liminf_{k \to \infty} f(x^k) \ge \bar{v}. \tag{30}$$

For each index $k \ge \hat{k}$, we have from (29) that

$$\begin{aligned} f(x^{k+1}) - \bar{v} &= f(x^{k+1}) - f(\bar{x}^k) \\ &= \nabla f(\tilde{x}^k)^T (x^{k+1} - \bar{x}^k) \\ &= (\nabla f(\tilde{x}^k) - g^k)^T (x^{k+1} - \bar{x}^k) + {g^k}^T (x^{k+1} - \bar{x}^k) \\ &\le L\|\tilde{x}^k - x^k\| \|x^{k+1} - \bar{x}^k\| + \frac{\bar{\delta}}{2}\|x^k - \bar{x}^k\|^2 - \frac{1}{\upsilon} q_{D^k}(x^k;\mathcal{J}^k), \end{aligned} \tag{31}$$

where the second step uses the Mean Value Theorem with $\tilde{x}^k$ a point lying on the segment joining x^{k+1} with $\bar{x}^k$ (so that $\tilde{x}^k \in X$); the fourth step uses (12) and Lemma 3.3. Using the inequalities $\|\tilde{x}^k - x^k\| \le \|x^{k+1} - x^k\| + \|x^k - \bar{x}^k\|$, $\|x^{k+1} - \bar{x}^k\| \le \|x^{k+1} - x^k\| + \|x^k - \bar{x}^k\|$ and $\|x^{k+1} - x^k\| = \alpha^k \|d^k\|$, we see from (28), and $\sup_k \alpha^k < \infty$ (since $\sup_k \alpha^k_{\text{init}} < \infty$) that the right-hand side of (31) is bounded above by

$$C_1 \left(\|d^k\|^2 - q_{D^k}(x^k;\mathcal{J}^k) + \|d_I(x^k;\mathcal{N})\|^2 \right) \tag{32}$$

for all $k \geq \hat{k}$, where $C_1 > 0$ is some constant depending on $L, \tau, \bar{\delta}, \upsilon, \sup_k \alpha^k$ only.

By (18), we have

$$\underline{\lambda}\|d^k\|^2 \leq {d^k}^T H^k d^k \leq -\frac{1}{1-\theta}\Delta^k \quad \forall k. \tag{33}$$

By (26) and (27), we also have

$$\|d_I(x^k;\mathcal{N})\|^2 \leq \left(1 + 1/\underline{\delta} + \sqrt{1 - 2/\bar{\delta} + (1/\underline{\delta})^2}\right)^2 \frac{\bar{\delta}^2}{2\underline{\delta}}(-q_{D^k}(x^k;\mathcal{N})) \quad \forall k.$$

Thus, the quantity in (32) is bounded above by

$$C_2\left(-\Delta^k - q_{D^k}(x^k;\mathcal{J}^k) - q_{D^k}(x^k;\mathcal{N})\right) \tag{34}$$

for all $k \geq \hat{k}$, where $C_2 > 0$ is some constant depending on $L, \tau, \bar{\delta}, \underline{\delta}, \theta, \underline{\lambda}, \upsilon, \sup_k \alpha^k$ only.

Combining (25) with (33) yields

$$\begin{aligned} -q_{D^k}(x^k;\mathcal{J}^k) &\leq -\Delta^k + \left(\theta - \frac{1}{2}\right){d^k}^T H^k d^k + \omega\|d^k\|^2 \\ &\leq -\Delta^k - \max\left\{0, \theta - \frac{1}{2}\right\}\frac{1}{1-\theta}\Delta^k - \frac{\omega}{\underline{\lambda}(1-\theta)}\Delta^k. \end{aligned} \tag{35}$$

Combining (9) and (35), we see that the quantity in (34) is bounded above by

$$-C_3\Delta^k$$

all $k \geq \hat{k}$, where $C_3 > 0$ is some constant depending on $L, \tau, \bar{\delta}, \underline{\delta}, \theta, \bar{\lambda}, \underline{\lambda}, \upsilon, \sup_k \alpha^k$ only. Thus the right-hand side of (31) is bounded above by $-C_3\Delta^k$ for all $k \geq \hat{k}$. Combining this with (19), (31), and $\inf_k \alpha^k > 0$ (see Theorem 4.1(d)) yields

$$f(x^{k+1}) - \bar{\upsilon} \leq C_4(f(x^k) - f(x^{k+1})) \quad \forall k \geq \hat{k},$$

where $C_4 = C_3/(\sigma \inf_k \alpha^k)$. Upon rearranging terms and using (30), we have

$$0 \leq f(x^{k+1}) - \bar{\upsilon} \leq \frac{C_4}{1+C_4}(f(x^k) - \bar{\upsilon}) \quad \forall k \geq \hat{k},$$

so $\{f(x^k)\}$ converges to $\bar{\upsilon}$ at least Q-linearly.

Finally, by (19), (33), and $x^{k+1} - x^k = \alpha^k d^k$, we have

$$\sigma(1-\theta)\underline{\lambda}\frac{\|x^{k+1} - x^k\|^2}{\alpha^k} \leq f(x^k) - f(x^{k+1}) \quad \forall k \geq \hat{k}.$$

This implies

$$\|x^{k+1} - x^k\| \leq \sqrt{\frac{\sup_k \alpha^k}{\sigma(1-\theta)\underline{\lambda}}(f(x^k) - f(x^{k+1}))} \quad \forall k \geq \hat{k}.$$

Since $\{f(x^k) - f(x^{k+1})\} \to 0$ at least R-linearly and $\sup_k \alpha^k < \infty$, this implies that $\{x^k\}$ converges at least R-linearly. ■

Similar to Corollary 4.1, Theorem 5.1 readily extends to any stepsize rule that yields a uniformly bounded stepsize and a larger descent than the Armijo rule at each iteration. An example is the limited minimization rule (24).

Corollary 5.1 *Theorem 5.1 still holds if in the CGD method the iterates are instead updated by $x^{k+1} = x^k + \tilde{\alpha}^k d^k$, where $\tilde{\alpha}^k \geq 0$ satisfies $\sup_k \tilde{\alpha}^k < \infty$, $f(x^k + \tilde{\alpha}^k d^k) \leq f(x^k + \alpha^k d^k)$ and $x^k + \tilde{\alpha}^k d^k \in X$ for $k = 0, 1, \ldots$, and $\{\alpha^k\}$ is chosen by the Armijo rule with $\sup_k \alpha^k_{\text{init}} < \infty$ and $\inf_k \alpha^k_{\text{init}} > 0$.*

Proof. The only change to the proof of Theorem 5.1 is in proving (32) and the last paragraph, where we use $\|x^{k+1} - x^k\| = \tilde{\alpha}^k \|d^k\|$ and $\sup_k \tilde{\alpha}^k < \infty$ instead. ■

6 Working Set Selection

In the previous two sections, we showed that the CGD method with $\mathcal{J}^k$ satisfying (9) has desirable convergence properties. In this section we study how to choose the working set satisfying (9) and compare our choice with existing choices of the working set in SMO methods for the SVM QP (1).

6.1 New working set satisfying (9)

The iteration complexity of the CGD method depends on $|\mathcal{J}^k|$ and the complexity of finding $\mathcal{J}^k$. In this subsection we show that a "small" $\mathcal{J}^k$ satisfying (9), for some constant $0 < \upsilon \leq 1$, can be found "reasonably fast" when D^k is diagonal. Our approach is based on the notion of a conformal realization [36], [37, Section 10B] of $d_{D^k}(x^k, \mathcal{N})$. Specifically, for any $d \in \Re^n$, the support of d is $\text{supp}(d) \stackrel{\text{def}}{=} \{j \in \mathcal{N} \mid d_j \neq 0\}$. A $d' \in \Re^n$ is *conformal* to $d \in \Re^n$ if

$$\text{supp}(d') \subseteq \text{supp}(d), \qquad d'_j d_j \geq 0 \ \forall j \in \mathcal{N}, \tag{36}$$

i.e., the nonzero components of d' have the same signs as the corresponding components of d. A nonzero $d \in \Re^n$ is an *elementary vector* of Null(A) if $d \in \text{Null}(A)$ and there is no nonzero $d' \in \text{Null}(A)$ that is conformal to d and $\text{supp}(d') \neq \text{supp}(d)$. Each elementary vector d satisfies $|\text{supp}(d)| \leq \text{rank}(A) + 1$ (since any subset of $\text{rank}(A) + 1$ columns of A are linearly dependent) [37, Exercise 10.6].

Proposition 6.1 *For any $x \in X$, $\ell \in \{\text{rank}(A)+1,\ldots,n\}$, and diagonal $D \succ 0$, there exists a nonempty $\mathcal{J} \subseteq \mathcal{N}$ satisfying $|\mathcal{J}| \leq \ell$ and*

$$q_D(x;\mathcal{J}) \leq \frac{1}{n-\ell+1} q_D(x;\mathcal{N}). \tag{37}$$

Proof. Let $d = d_D(x;\mathcal{N})$. We divide our argument into three cases.

Case (i) $d = 0$: Then $q_D(x;\mathcal{N}) = 0$. Thus, for any nonempty $\mathcal{J} \subseteq \mathcal{N}$ with $|\mathcal{J}| \leq \ell$, we have from (8) and Lemma 2.1 with $H = D$ that $q_D(x;\mathcal{J}) \leq 0 = q_D(x;\mathcal{N})$, so (37) holds.

Case (ii) $d \neq 0$ and $|\text{supp}(d)| \leq \ell$: Then $\mathcal{J} = \text{supp}(d)$ satisfies $q_D(x;\mathcal{J}) = q_D(x;\mathcal{N})$ and hence (37), as well as $|\mathcal{J}| \leq \ell$.

Case (iii) $d \neq 0$ and $|\text{supp}(d)| > \ell$: Since $d \in \text{Null}(A)$, it has a conformal realization [36], [37, Section 10B], namely,

$$d = v^1 + \cdots + v^s,$$

for some $s \geq 1$ and some nonzero elementary vectors $v^t \in \text{Null}(A)$, $t = 1,\ldots,s$, conformal to d. Then for some $\alpha > 0$, $\text{supp}(d')$ is a proper subset of $\text{supp}(d)$ and $d' \in \text{Null}(A)$, where $d' = d - \alpha v^1$. (Note that αv^1 is an elementary vector of $\text{Null}(A)$, so that $|\text{supp}(\alpha v^1)| \leq \text{rank}(A)+1 \leq \ell$.) We repeat the above reduction step with d' in place of d. Since $|\text{supp}(d')| \leq |\text{supp}(d)| - 1$, after at most $|\text{supp}(d)| - \ell$ reduction steps, we obtain

$$d = d^1 + \cdots + d^r, \tag{38}$$

for some $r \leq |\text{supp}(d)| - \ell + 1$ and some nonzero $d^t \in \text{Null}(A)$ conformal to d with $|\text{supp}(d^t)| \leq \ell$, $t = 1,\ldots,r$. Since $|\text{supp}(d)| \leq n$, we have $r \leq n - \ell + 1$.

Since $l - x \leq d \leq u - x$, (38) and d^t being conformal to d imply $l - x \leq d^t \leq u - x$, $t = 1,\ldots,r$. Since $Ad^t = 0$, this implies $x + d^t \in X$, $t = 1,\ldots,r$. Also, (8) and (38) imply that

$$\begin{aligned} q_D(x;\mathcal{N}) &= g^T d + \frac{1}{2} d^T D d \\ &= \sum_{t=1}^r g^T d^t + \frac{1}{2} \sum_{s=1}^r \sum_{t=1}^r (d^s)^T D d^t \\ &\geq \sum_{t=1}^r g^T d^t + \frac{1}{2} \sum_{t=1}^r (d^t)^T D d^t \\ &\geq r \min_{t=1,\ldots,r} \left\{ g^T d^t + \frac{1}{2} (d^t)^T D d^t \right\}, \end{aligned}$$

where $g = \nabla f(x)$ and the first inequality uses (36) and $D \succ 0$ being diagonal, so that $(d^s)^T D d^t \geq 0$ for all s,t. Thus, if we let $\bar{t}$ be an index t attaining the above minimum and let $\mathcal{J} = \text{supp}(d^{\bar{t}})$, then $|\mathcal{J}| \leq \ell$ and

$$\frac{1}{r} q_D(x;\mathcal{N}) \geq g^T d^{\bar{t}} + \frac{1}{2} (d^{\bar{t}})^T D d^{\bar{t}} \geq q_D(x;\mathcal{J}),$$

where the second inequality uses $x + d^{\bar{t}} \in X$ and $d^{\bar{t}}_j = 0$ for $j \notin \mathcal{J}$. ∎

It can be seen from its proof that Proposition 6.1 still holds if the diagonal matrix D is only positive semidefinite, provided that $q_D(x;\mathcal{N}) > -\infty$ (such as when X is bounded). Thus Proposition 6.1 may be viewed as an extension of [4, Lemma 2.3] and [27, Theorem 2, part 2] for the case of $D = 0$.

The proof of Proposition 6.1 suggests, for any $\ell \in \{\text{rank}(A) + 1, \ldots, n\}$, an $O(n - \ell)$-step reduction procedure for finding a conformal realization (38) of $d = d_D(x;\mathcal{N})$ with $r \le n - \ell + 1$ and a corresponding $\mathcal{J}$ satisfying $|\mathcal{J}| \le \ell$ and (37).

- In the case of $m = 1$ and $\ell = 2$, by scaling A and dropping zero columns if necessary, we can without loss of generality assume that $A = e^T$ (so d has at least one positive and one negative component) and by recursively subtracting α from a positive component d_i and adding α to a negative component d_j, where $\alpha = \min\{d_i, -d_j\}$, we can find such a conformal realization in $O(n)$ operations.

- In the case of $m = 2$ and $\ell = 3$, the preceding procedure can be extended, by using sorting, to find such a conformal realization in $O(n \log n)$ operations. For brevity we omit the details.

- In general, each step of the reduction procedure requires finding a nonzero $v \in \text{Null}(A)$ with $|\text{supp}(v)| \le \ell$ and conformal to a given $d \in \text{Null}(A)$ with $|\text{supp}(d)| > \ell$. This can be done in $O(m^3(n - \ell))$ operations as follows: Choose any $\mathcal{J} \subset \text{supp}(d)$ with $|\mathcal{J}| = m + 1$. Find a nonzero $w \in \text{Null}(A)$ with $w_j = 0$ for all $j \notin \mathcal{J}$. This can be done in $O(m^3)$ operations using Gaussian elimination. Then for some $\alpha \in \Re$, $\text{supp}(d')$ is a proper subset of $\text{supp}(d)$ and $d' \in \text{Null}(A)$, where $d' = d - \alpha w$. Repeat this with d' in place of d. The number of repetitions is at most $\text{supp}(d) - \ell \le n - \ell$. The overall time complexity of this reduction procedure is $O(m^3(n - \ell)^2)$ operations.

For diagonal $D \succ 0$ and $m = 1$, $d_D(x;\mathcal{N})$ can be found by solving a continuous quadratic knapsack problem in $O(n)$ operations; see [3, 21] and references therein. For diagonal $D \succ 0$ and $m > 1$, $d_D(x;\mathcal{N})$ can be found using an algorithm described by Berman, Kovoor and Pardalos [1], which reportedly requires only $O(n)$ operations for each fixed m.

By combining the above observations, we conclude that, for $m = 1$ and $\ell = 2$, a working set $\mathcal{J}$ satisfying $|\mathcal{J}| \le \ell$ and (37) can be found in $O(n)$ operations. For $m = 2$ and $\ell = 3$, such a working set $\mathcal{J}$ can be found in $O(n \log n)$ operations. For $m \ge 1$ and $\ell \in \{\text{rank}(A) + 1, \ldots, n\}$, such a working set $\mathcal{J}$ can be found in $O(n^2)$ operations, where the constant in $O(\cdot)$ depends on m. It is an open question whether such a $\mathcal{J}$ can be found in $O(n)$ operations for a fixed $m \ge 2$.

6.2 Comparison with existing working sets

In this subsection we compare (9) and (10) with existing choices of the working set $\mathcal{J}$ at an $x \in X$ in SMO methods for the SVM QP (1).

Joachims [16] proposed the first systematic way of choosing $\mathcal{J}$:

$$\mathcal{J} \in \arg\min_{\mathcal{J}':|\mathcal{J}'|\le \ell} \left\{ \begin{array}{ll} \min_d & \nabla f(x)^T d \\ \text{s.t.} & a^T d = 0, \\ & d_j \ge 0, \text{ if } x_j = 0,\ j \in \mathcal{J}', \\ & d_j \le 0, \text{ if } x_j = C,\ j \in \mathcal{J}', \\ & |d_j| \le 1,\ j \in \mathcal{J}', \\ & d_j = 0,\ j \notin \mathcal{J}', \end{array} \right\} \tag{39}$$

where $\ell \ge 2$ is an even number. Such $\mathcal{J}$ can be found from among the lowest $\ell/2$ terms from $a_j \nabla f(x)_j$, $j \in \mathcal{I}_+(x)$, and the highest $\ell/2$ terms from $a_j \nabla f(x)_j$, $j \in \mathcal{I}_-(x)$, in $O(n \min\{\ell, \log n\})$ operations using (partial) sorting, where $\mathcal{I}_+(x) \stackrel{\text{def}}{=} \{j \mid x_j < C, a_j = 1 \text{ or } x_j > 0, a_j = -1\}$ and $\mathcal{I}_-(x) \stackrel{\text{def}}{=} \{j \mid x_j < C, a_j = -1 \text{ or } x_j > 0, a_j = 1\}$. This choice is used in the SVM^{light} code, with $\ell = 10$ as the default value.

Motivated by the aforementioned work, Chang, Hsu and Lin [4] proposed an extension of the SMO method to problems with smooth objective function, in which the working set is chosen by

$$\mathcal{J} \in \arg\min_{\mathcal{J}':|\mathcal{J}'|\le \ell} \left\{ \begin{array}{ll} \min_d & \nabla f(x)^T d \\ \text{s.t.} & a^T d = 0, \\ & 0 \le x_j + d_j \le C,\ j \in \mathcal{J}', \\ & d_j = 0,\ j \notin \mathcal{J}', \end{array} \right\} \tag{40}$$

where $\ell \ge 2$. They proved global convergence for their method in that every cluster point of the generated iterates x is a stationary point. Simon [42, Section 6] showed that, in the case of $\ell = 2$, a $\mathcal{J}$ satisfying (40) can be found in $O(n)$ operations. For $\ell > 2$, such $\mathcal{J}$ can still be found in $O(n)$ operations [27], though the constant in $O(\cdot)$ depends exponentially in ℓ.

Keerthi et al. [18] proposed choosing, for a fixed tolerance $\epsilon > 0$, a working set $\mathcal{J} = \{i, j\}$ satisfying

$$i \in \mathcal{I}_+(x), \quad j \in \mathcal{I}_-(x), \quad a_i \nabla f(x)_i < a_j \nabla f(x)_j - \epsilon.$$

They proved that the SMO method with this choice of $\mathcal{J}$ terminates in a finite number of iterations with $m(x) \ge M(x) - \epsilon$, where

$$m(x) \stackrel{\text{def}}{=} \min_{j \in \mathcal{I}_+(x)} a_j \nabla f(x)_j, \qquad M(x) \stackrel{\text{def}}{=} \max_{j \in \mathcal{I}_-(x)} a_j \nabla f(x)_j.$$

(Note that a feasible point x of (1) is a global minimum if and only if $m(x) \ge M(x)$.) In [20], Keerthi et al. proposed a related choice of $\mathcal{J} = \{i, j\}$ with i and j attaining the minimum and maximum, respectively, in the above definition of $m(x)$ and $M(x)$. This choice, called "maximal violating pair" and used in LIBSVM, is equivalent to Joachim's choice (39) with $\ell = 2$.

The first convergence result for the SMO method using the working set (39) was given by Lin [22], who proved that every cluster point of the generated iterates x is a global minimum of (1), assuming $\min_{\mathcal{J}':|\mathcal{J}'|\le\ell}(\lambda_{\min}(Q_{\mathcal{J}'\mathcal{J}'})) > 0$. This assumption was later shown by Lin [24] to be unnecessary if $\ell = 2$. Under the further assumptions that Q is positive definite and strict complementarity holds at the unique global minimum, linear convergence was also proved [23]. List and Simon [26] proposed an extension of the SMO method to problems with more than one linear constraint, in which the working set $\mathcal{J}$ is obtained from maximizing a certain function of x and $\mathcal{J}$. They proved global convergence for their method under the same assumption on Q as Lin. Simon [42] later showed that the maximization subproblem is NP-complete and he proposed a polynomial-time approximation algorithm for finding $\mathcal{J}$ which retains the method's global convergence property.

Hush and Scovel [14] proposed choosing $\mathcal{J}$ to contain a "rate certifying pair", an example of which is (40) with $\ell = 2$. They proved that, for any $\epsilon > 0$, the SMO method with this choice of $\mathcal{J}$ terminates in $O(C^2n^2(f(x^{\text{init}}) - f(x^*) + n^2\Lambda)/\epsilon)$ iterations with $f(x) \le f(x^*) + \epsilon$, where x^* is a global minimum of (1) and Λ is the maximum norm of the 2×2 principal submatrices of Q. They also showed that a $\mathcal{J}$ satisfying (40) can be found in $O(n \log n)$ operations. These complexity bounds were further improved by List and Simon [27] to problems with general linear constraints, where they also showed that a $\mathcal{J}$ satisfying (40) can be found in $O(n)$ operations. Hush et al. [15] proposed a more practical choice of $\mathcal{J}$, based on those used in [20] and [42] that achieves the same complexity bounds as in [27].

Palagi and Sciandrone [34] proposed, as a generalization of (39), choosing $\mathcal{J}$ to have at most ℓ elements ($\ell \ge 2$) and to contain a maximal violating pair. They also added a proximal term $\tau\|x - x^{\text{current}}\|^2$ to the objective function of (1) when minimizing with respect to x_j, $j \in \mathcal{J}$. For this modified SMO method, they proved global convergence with no additional assumption. Chen et al. [6] then proposed a generalization of maximal violating pair by choosing $\mathcal{J} = \{i, j\}$ with $i \in \mathcal{I}_+(x)$, $j \in \mathcal{I}_-(x)$ satisfying

$$a_j \nabla f(x)_j - a_i \nabla f(x)_i \ge \phi(M(x) - m(x)), \tag{41}$$

where $\phi : [0, \infty) \to [0, \infty)$ is any strictly increasing function satisfying $\phi(0) = 0$ and $\phi(\alpha) \le \alpha$ for all $\alpha \ge 0$. Following [34], they also add a proximal term to the objective function, but only when it is not strong convex with respect to x_j, $j \in \mathcal{J}$. For this modified SMO method and allowing Q to be indefinite, they proved global convergence with no additional assumption. Linear convergence was proved for the choice $\phi(\alpha) = \upsilon\alpha$ ($0 < \upsilon \le 1$) and under the same assumption as in [23], namely, Q is positive definite and strict complementarity holds at the unique global minimum. While Q can be indefinite for certain kernel functions, the QP (1), being nonconvex, can no longer be interpreted as a Lagrangian dual problem. The working set choice (9) with $m = 1$, $|\mathcal{J}| = 2$, $D = 0$, and X in (4) replaced by its tangent cone at x is similar in spirit to (41) with $\phi(\alpha) = \upsilon\alpha$.

Fan et al. [8] considered a version of maximal violating pair that uses 2nd-derivative information by adding a Hessian term to the objective of (39) with $\ell = 2$:

$$\mathcal{J} \in \mathop{\arg\min}_{\mathcal{J}':|\mathcal{J}'|=2} \left\{ \begin{array}{ll} \min_d & \nabla f(x)^T d + \frac{1}{2} d^T Q d \\ \text{s.t.} & a^T d = 0, \\ & d_j \geq 0, \text{ if } x_j = 0, \ j \in \mathcal{J}', \\ & d_j \leq 0, \text{ if } x_j = C, \ j \in \mathcal{J}', \\ & d_j = 0, \ j \notin \mathcal{J}'. \end{array} \right\} \tag{42}$$

(This minimizes $f(x + d)$ over all feasible directions d at x with two nonzero components.) However, no fast way for finding such a $\mathcal{J}$ is known beyond checking all $\binom{n}{2}$ subsets of $\mathcal{N}$ of cardinality 2, which is too slow for SVM applications. Fan et al. [8] thus proposed a hybrid strategy of choosing an index i from a maximal violating pair (i.e., $i \in \mathcal{I}_+(x)$ with $a_i \nabla f(x)_i = m(x)$ or $i \in \mathcal{I}_-(x)$ with $a_i \nabla f(x)_i = M(x)$) and then further constraining $\mathcal{J}'$ in (42) to contain i. The resulting $\mathcal{J}$ can be found in $O(n)$ operations and improved practical performance. Moreover, such $\mathcal{J}$ belongs to the class of working sets studied in [6], so the convergence results in [6] for a modified SMO method can be applied. Glamachers and Igel [13] proposed a modification of this hybrid strategy whereby if the most recent working set contains an i with $(1 - \delta)C \leq x_i \delta C$ ($0 < \delta < 1/2$, e.g., $\delta = 10^{-8}$), then choose $\mathcal{J}$ by (42) with $\mathcal{J}'$ further constrained to contain i; otherwise choose $\mathcal{J}$ to be a maximal violating pair. Glamachers and Igel showed that this choice of $\mathcal{J}$ belongs to the class of working sets studied in [26], so the convergence result in [26] for the SMO method can be applied. Motivated by this work, Lucidi et al. [28] proposed choosing the working set to be a maximal violating pair $\{i, j\}$ and, if x_i, x_j are strictly between their bounds after the SMO iteration, then performing an auxiliary SMO iteration with respect to a subset $\mathcal{J}'$ of coordinates whose corresponding columns in Q are currently cached. Global convergence for this SMO method was proved under a sufficient descent condition on the auxiliary SMO iteration, which holds if either Q is positive definite or $|\mathcal{J}'| = 2$.

When applied to (1), the new working set in Subsection 6.1, like those in [8, 13], has cardinality 2 and is found using 2nd-derivative information in $O(n)$ operations. However, this working set is found by minimizing a separable quadratic approximation of f over the feasible set X and then decomposing the displacement into elementary vectors and finding the ‘best’ one, which is very different from choosing one index of the working set from a maximal violating pair and choosing the other index to minimize descent over the set of feasible directions. And unlike existing working sets, it yields linear convergence without any additional assumption on (1); see Theorem 5.1.

7 Numerical Experience on the SVM QP

In order to better understand its practical performance, we have implemented the CGD method in Fortran to solve the SVM QP (1)-(2), with the working set chosen as described in Section 6. In this case, the CGD method effectively reduces to an SMO method, so the novelty is our choice of the working set. In this section, we describe our implementation and report our numerical experience on some large two-class data classification problems. This is compared with LIBSVM (version 2.83), which chooses the working set differently, but with the same cardinality of 2.

In our tests, we use $C = 1, 10$ and the linear kernel $K(z_i, z_j) = z_i^T z_j$, the radial basis function kernel $K(z_i, z_j) = \exp(-\gamma \|z_i - z_j\|^2)$, the polynomial kernel $K(z_i, z_j) = (\gamma z_i^T z_j + s)^{deg}$, and the sigmoid kernel $K(z_i, z_j) = \tanh(\gamma z_i^T z_j + s)$ with $\gamma = 1/p$, $s = 0$, $deg = 3$ (cubic), the default setting for LIBSVM. For the sigmoid kernel, Q can be indefinite.

For the test problems, we use the two-class data classification problems from the LIBSVM data webpage `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`, for which $a \in \{-1, 1\}^n$. Due to memory limitation on our departmental Linux system, we limit n to at most 50,000 and p to at most 300. This yields the five problems shown in Table 1.

Our implementation of the CGD method has the form

$$x^{k+1} = x^k + d_Q(x^k; \mathcal{J}^k), \quad k = 0, 1, \ldots,$$

with $|\mathcal{J}^k| = 2$ always. This corresponds to the CGD method with α^k chosen by the minimization rule. (The choice of H^k is actually immaterial here.) As with SMO methods, we initialize $x^0 = 0$ and, to save time, we cache the most recently used columns of Q, up to a user-specified limit `maxCN`, when updating the gradient $\nabla f(x^k) = Qx^k - e$. In our tests, we set `maxCN`=5000 for `ijcnn1` and otherwise `maxCN`=8000. We terminate the method when $-q_D(x^k; \mathcal{N}) \leq 10^{-5}$.

We describe below how we choose the working set $\mathcal{J}^k$ for the CGD method. We fix the diagonal scaling matrix

$$D = \text{diag}\left[\max\{Q_{jj}, 10^{-5}\}\right]_{j=1,\ldots,n}.$$

(We also experimented with $D = I$, but this resulted in worse performance.) At the initial iteration and at certain subsequent iterations k, we compute $d_D(x^k, \mathcal{N})$ and $q_D(x^k; \mathcal{N})$ by using a linear-time Fortran code `k1vfo` provided to us by Krzysztof Kiwiel, as described in [21], to solve the corresponding continuous quadratic knapsack problem. Then we find a conformal realization of $d_D(x^k, \mathcal{N})$ using the linear-time reduction procedure described in Section 6. By Proposition 6.1, there exists at least one elementary vector in this realization whose support $\mathcal{J}$ satisfies

$$q_D(x^k; \mathcal{J}) \leq \frac{1}{n-1} q_D(x^k; \mathcal{N}).$$

From among all such $\mathcal{J}$, we find the best one (i.e., has the least $q_Q(x^k; \mathcal{J})$ value) and make this the working set $\mathcal{J}^k$. (We also experimented with choosing one with the least $q_D(x^k; \mathcal{J})$ value, but this resulted in worse performance.) Since the continuous quadratic knapsack problem takes significant time to solve by `k1vfo`, we in addition find from among all such $\mathcal{J}$ the second-best and third-best ones, if they exist. (In our tests, they always exist.) If the second-best one is disjoint from $\mathcal{J}^k$, we make it the next working set $\mathcal{J}^{k+1}$, and if the third-best one is disjoint from both $\mathcal{J}^k$ and $\mathcal{J}^{k+1}$, we make it the second-next working set $\mathcal{J}^{k+2}$. (In our tests, the latter case occurs about 85-90% of the time.) If the second-best one is not disjoint from $\mathcal{J}^k$ but the third-best one is, then we make the third-best one the next working set $\mathcal{J}^{k+1}$. (We can also allow them to overlap, though the updating of $\nabla f(x^k)$ becomes more complicated and might not significantly improve the performance as the overlapping case occurs only about 10-15% of the time.) This working set selection procedure is then repeated at iteration $k+3$ or $k+2$ or $k+1$, depending on the case, and so on. It is straightforward to check that the global convergence and asymptotic linear convergence properties of the CGD method, as embodied in Theorems 4.1 and 5.1, extend to this choice of the working set. We refer to this CGD method as CGD-3pair.

We report in Table 1 our numerical results, showing the number of iterations (iter), final f-value (obj), total time (cpu) in minutes. For CGD-3pair, we also show the total time taken by `k1vfo` to solve the knapsack problems (kcpu), the total time to compute/cache columns of Q and update the gradient (gcpu), and the total number of knapsack problems solved (kiter). All runs are performed on an HP DL360 workstation, running Red Hat Linux 3.5. LIBSVM and CGD-3pair are compiled using the Gnu C++ and F-77 3.2.3 compiler (g++ -Wall -O3 and g77 -O), respectively. From Table 1, we see that the total number of iterations and the final f-value for CGD-3pair are comparable (within a factor of 2) to those of LIBSVM. On the other hand, the cpu times for CGD-3pair are much higher when the linear kernel is used, due to the greater times spent in `k1vfo` and for updating the gradient. When a nonlinear kernel is used, the cpu times for CGD-3pair are comparable to those of LIBSVM.

In general, CGD-3pair is significantly slower than LIBSVM when the linear kernel is used. But when a nonlinear kernel is used, CGD-3pair is comparable to LIBSVM in speed and solution quality–except for the rbf kernel with $C = 10$, for which CGD-3pair is 1.5-2 times slower than LIBSVM. This suggests that the working set choice of Section 6 could be a viable alternative to existing choices, especially when a nonlinear kernel is used. Conceivably CGD-3pair can be further speeded up by omitting infrequently updated components from computation ("shrinkage"), as is done in LIBSVM and SVM^{light}, and by incorporating "warm start" in the knapsack problem solver `k1vfo`, i.e., using a solution of the previous knapsack problem to initialize the solution of the next knapsack problem. Recoding CGD-3pair in C++ to make use of dynamic memory allocation and pointer structure is another direction for future research, as are extensions to multi-class data classification.

For the SVM QP (1), SMO method and CGD method have the advantage that they can be implemented to use only $O(n)$ operations per iteration and the number of iterations is typically $O(n)$ or lower. By starting at $x = 0$, the gradient can be computed in $O(n)$ operations and subsequently be updated in $O(n)$ operations. In contrast, an interior-point method would need to start at an $x > 0$, so it would take $O(n^2)$ operations just to compute the gradient, and then one needs to compute a quantity of the form $y^T(\rho I + Q)^{-1}y$ $(\rho > 0)$ at each iteration to obtain the search direction d. An exception is when Q has low rank r or is the sum of a rank-r matrix with a positive multiple of the identity matrix, such as linear

SVM. Then Qx can be computed in $O(rn)$ operations and $(\rho I + Q)^{-1}y$ can be efficiently computed using low-rank updates [9, 10, 11]. In this case, it might be advantageous to use larger $\mathcal{J}^k$ in the CGD method to accelerate convergence. This is worth further exploration.

8 Conclusions and Extensions

We have proposed a block-coordinate gradient descent method for linearly constrained smooth optimization, and have established its global convergence and asymptotic linear convergence to a stationary point under mild assumptions. On the SVM QP (1), this method achieves linear convergence under no additional assumption, and is implementable in $O(n)$ operations per iteration. Our preliminary numerical experience suggests that it can be competitive with state-of-the-art SVM code on large data classification problems when a nonlinear kernel is used.

There are many directions for future research. For example, in Section 6 we mentioned that a conformal realization can be found in $O(n \log n)$ operations when $m = 2$. However, for large-scale applications such as ν-SVM, this can still be slow. Can this be improved to $O(n)$ operations? Also, in our current implementation of the CGD method, we use a diagonal D^k when finding a working set $\mathcal{J}^k$ satisfying (9). Can we use a nondiagonal D^k and still efficiently find a $\mathcal{J}^k$ satisfying (9)?

The problem (3) and that in [44] can be generalized to the following problem:

$$\begin{array}{ll} \min\limits_{x \in \Re^n} & f(x) + cP(x) \\ \text{s.t.} & Ax = b, \end{array}$$

where $c > 0$, $P : \Re^n \to (-\infty, \infty]$ is a block-separable proper convex lower semicontinuous function. In particular, the problem in [44] corresponds to the special case of $A = 0$, $b = 0$ and (3) corresponds to the special case of

$$P(x) = \begin{cases} 0 & \text{if } l \le x \le u; \\ \infty & \text{else.} \end{cases} \tag{43}$$

For example, it may be desirable to replace 0 in (43) with the 1-norm $\|x\|_1$ to seek a sparse SVM solution. Can the CGD method be extended to solve this more general problem?

One of the referees asked about applying the CGD method to the least-squares SVM [43], which has the form

$$\min_{x \in \Re^n} \frac{1}{2} x^T Q x - e^T x + \frac{1}{2C}\|x\|^2 \quad \text{s.t.} \quad a^T x = 0,$$

assuming $Q + \frac{1}{C} I \succ 0$. This problem has a much simpler structure than (1) and in particular, by using $a^T x = 0$ to eliminate one of the variables, reduces to an unconstrained strictly convex quadratic optimization problem. In [19], an SMO method is proposed and compared with a conjugate gradient method. The CGD method can also be applied to this problem, for which $d_D(x, \mathcal{N})$ can be obtained in closed form without solving a continuous quadratic knapsack problem. How the CGD method performs on this problem is a topic for future study.

Acknowledgement. We thank Krzysztof Kiwiel for providing us with `k1vfo` and help with testing and debugging. We thank two referees for their helpful suggestions.

References

[1] Berman, P., Kovoor, N., and Pardalos, P. M., Algorithms for the least distance problem, in Complexity in Numerical Optimization, P. M. Pardalos, ed., World Scientific, Singapore, 1993, 33-56.

[2] Bertsekas, D. P., Nonlinear Programming, 2nd edition, Athena Scientific, Belmont, 1999.

[3] Brucker, P., An $O(n)$ algorithm for quadratic knapsack problems, Oper. Res. Lett., 3 (1984), 163-166.

[4] Chang, C.-C., Hsu, C.-W., and Lin, C.-J., The analysis of decomposition methods for support vector machines, IEEE Trans. Neural Networks, 11 (2000), 1003-1008.

[5] Chang, C.-C. and Lin, C.-J., LIBSVM: a library for support vector machines, 2001, available from `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[6] Chen, P.-H., Fan, R.-E., and Lin, C.-J., A study on SMO-type decomposition methods for support vector machines, IEEE Trans. Neural Networks, 17 (2006), 893-908.

[7] Cristianini, N. and Shawe-Taylor, J., An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, 2000.

[8] Fan, R.-E., Chen, P.-H., and Lin, C.-J., Working set selection using second order information for training support vector machines, J. Mach. Learn. Res., 6 (2005), 1889-1918.

[9] Ferris, M. C. and Munson, T. S., Interior-point methods for massive support vector machines, SIAM J. Optim., 13 (2003), 783-804.

[10] Ferris, M. C. and Munson, T. S., Semismooth support vector machines, Math. Program., 101 (2004), 185-204.

[11] Fine, S. and Scheinberg, K., Efficient SVM training using low-rank kernel representations, J. Mach. Learn. Res., 2 (2001), 243-264.

[12] Fletcher, R., Practical Methods of Optimization, 2nd edition, John Wiley & Sons, Chichester, 1987.

[13] Glasmachers, T. and Igel, C., Maximum-gain working set selection for SVMs, J. Mach. Learn. Res., 7 (2006), 1437-1466.

[14] Hush, D. and Scovel, C., Polynomial-time decomposition algorithms for support vector machines, Mach. Learn., 51 (2003), 51-71.

[15] Hush, D., Kelly, P., Scovel, C., and Steinwart, I., QP algorithms with guaranteed accuracy and run time for support vector machines, J. Mach. Learn. Res., 7 (2006), 733-769.

[16] Joachims, T., Making large-scale SVM learning practical, in Advances in Kernel Methods - Support Vector Learning, B. Schólkopf, C. J. C. Burges, and A. J. Smola, eds., MIT Press, Cambridge, MA, 1999, 169-184.

[17] Keerthi, S. S. and Gilbert, E. G., Convergence of a generalized SMO algorithm for SVM classifier design, Mach. Learn., 46 (2002), 351-360.

[18] Keerthi, S. S. and Ong, C. J., On the role of the threshold parameter in SVM training algorithm, Technical Report CD-00-09, Department of Mathematical and Production Engineering, National University of Singapore, Singapore, 2000.

[19] Keerthi, S. S. and Shevade, S. K., SMO algorithm for least-squares SVM formulations, Neural Comput., 15 (2003), 487-507.

[20] Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., and Murthy, K. R. K., Improvements to Platt's SMO algorithm for SVM classifier design, Neural Comput., 13 (2001), 637-649.

[21] Kiwiel, K. C., On linear time algorithms for the continuous quadratic knapsack problem, J. Optim. Theory Appl., 134 (2007), 549-554.

[22] Lin, C.-J., On the convergence of the decomposition method for support vector machines, IEEE Trans. Neural Networks, 12 (2001), 1288-1298.

[23] Lin, C.-J., Linear convergence of a decomposition method for support vector machines, Technical Report, Department of Computer Science and Information Engineering, Taiwan University, Taipei, Taiwan, 2001.

[24] Lin, C.-J., Asymptotic convergence of an SMO algorithm without any assumptions, IEEE Trans. Neural Networks, 13 (2002), 248-250.

[25] Lin C.-J., Lucidi S., Palagi L., Risi A., and Sciandrone M., A decomposition algorithm model for singly linearly constrained problems subject to lower and upper bounds, Technical Report, DIS-Università di Roma "La Sapienza", Rome, January 2007; to appear in J. Optim. Theory Appl.

[26] List, N. and Simon, H. U., A general convergence theorem for the decomposition method, in Proceedings of the 17th Annual Conference on Learning Theory, 2004, 363-377.

[27] List, N. and Simon, H. U., General polynomial time decomposition algorithms, in Lecture Notes in Computer Science Volume 3559/2005, Springer, Berlin, 2005, 308-322.

[28] Lucidi, S., Palagi, L., Risi, A., and Sciandrone, M., On the convergence of hybrid decomposition methods for SVM training, Technical Report, DIS-Università di Roma "La Sapienza", Rome, July 2006; submitted to IEEE Trans. Neural Networks.

[29] Luo, Z.-Q. and Tseng, P., Error bounds and the convergence analysis of matrix splitting algorithms for the affine variational inequality problem, SIAM J. Optim., 2 (1992), 43-54.

[30] Luo, Z.-Q. and Tseng, P., Error bounds and convergence analysis of feasible descent methods: a general approach, Ann. Oper. Res., 46 (1993), 157-178.

[31] Mangasarian, O. L. and Musicant, D. R., Successive overrelaxation for support vector machines, IEEE Trans. Neural Networks, 10 (1999), 1032-1037.

[32] Nocedal, J. and Wright S. J., Numerical Optimization, Springer-Verlag, New York, 1999.

[33] Osuna, E., Freund, R., and Girosi, F., Improved training algorithm for support vector machines, Proc. IEEE NNSP '97, (1997).

[34] Palagi, L. and Sciandrone, M., On the convergence of a modified version of $\text{SVM}_{\text{light}}$ algorithm, Optim. Methods Softw., 20 (2005), 317-334.

[35] Platt, J., Sequential minimal optimization: A fast algorithm for training support vector machines, in Advances in Kernel Methods-Support Vector Learning, B. Schölkopf, C. J. C. Burges, and A. J. Smola, eds. MIT Press, Cambridge, MA, 1999, 185-208.

[36] Rockafellar, R. T., The elementary vectors of a subspace of R^N, in Combinatorial Mathematics and its Applications, Proc. of the Chapel Hill Conference 1967, R. C. Bose and T. A. Dowling, eds., Univ. North Carolina Press, Chapel Hill, NC, 1969, 104-127.

[37] Rockafellar, R. T., Network Flows and Monotropic Optimization, Wiley-Interscience, New York, 1984; republished by Athena Scientific, Belmont, MA, 1998.

[38] Rockafellar, R. T. and Wets R. J.-B., Variational Analysis, Springer-Verlag, New York, 1998.

[39] Saunders, C., Stitson, M. O., Weston, J., Bottou, L., Schölkopf., B., and Smola, A. J., Support vector machine – reference manual, Report CSD-TR-98-03, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998.

[40] Scheinberg, K., An efficient implementation of an active set method for SVM, J. Mach. Learn. Res., 7 (2006), 2237-2257.

[41] Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L., New support vector algorithms, Neural Comput., 12 (2000), 1207-1245.

[42] Simon, H. U., On the complexity of working set selection, Proceedings of the 15th International Conference on Algorithmic Learning Theory, 2004, 324-337.

[43] Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J., Least Squares Support Vector Machines, World Scientific, New Jersey, 2002.

[44] Tseng, P. and Yun S., A coordinate gradient descent method for nonsmooth separable minimization, Math. Program. B., 117 (2009), 387-423.

[45] Vapnik, V., Estimation of Dependences Based on Empirical Data, Springer-Verlag, New York, 1982.

Data set	n/p	C/kernel	**LIBSVM**	**CGD-3pair**
			iter/obj/cpu	iter/obj/cpu(kcpu/gcpu)/kiter
a7a	16100/122	1/lin	64108/-5699.253/1.3	56869/-5699.246/6.3(1.7/4.0)/21296
		10/lin	713288/-56875.57/4.6	598827/-56875.55/59.4(20.3/34.1)/228004
		1/rbf	4109/-5899.071/1.3	4481/-5899.070/1.0(0.1/0.8)/1593
		10/rbf	10385/-55195.29/1.4	16068/-55195.30/2.0(0.5/1.4)/5834
		1/poly	4149/-7720.475/1.1	4470/-7720.478/0.8(0.1/0.6)/1536
		10/poly	4153/-67778.17/1.2	4593/-67778.17/0.8(0.1/0.6)/1599
		1/sig	3941/-6095.529/1.7	4201/-6095.529/1.2(0.1/1.0)/1474
		10/sig	9942/-57878.56/1.7	10890/-57878.57/1.8(0.3/1.3)/4211
a8a	22696/123	1/lin	83019/-8062.410/2.7	95522/-8062.404/16.0(4.4/10.4)/35686
		10/lin	663752/-80514.32/10.7	782559/-80514.27/106.2(35.1/61.2)/291766
		1/rbf	5641/-8249.503/2.6	6293/-8249.504/2.1(0.2/1.6)/2222
		10/rbf	15469/-77831.16/2.7	26137/-77831.16/4.8(1.1/3.3)/9432
		1/poly	5819/-10797.56/2.2	6202/-10797.57/1.7(0.3/1.2)/2133
		10/poly	5656/-92870.58/2.1	6179/-92870.59/1.6(0.3/1.2)/2136
		1/sig	5473/-8491.386/3.2	6172/-8491.388/2.5(0.3/2.0)/2197
		10/sig	10955/-81632.40/3.3	17157/-81632.41/3.8(0.8/2.8)/6646
a9a	32561/123	1/lin	80980/-11433.38/5.7	110602/-11433.38/27.3(7.9/17.3)/40667
		10/lin	1217122/-114237.4/24.0	1287193/-114237.4/291.4(92.9/175.8)/482716
		1/rbf	7975/-11596.35/5.2	8863/-11596.35/4.3(0.5/3.3)/3110
		10/rbf	21843/-110168.5/5.4	36925/-110168.5/10.7(2.8/7.3)/13140
		1/poly	8282/-15243.50/4.5	8777/-15243.50/3.4(0.6/2.5)/3002
		10/poly	7816/-128316.3/4.0	8769/-128316.4/3.3(0.6/2.4)/3019
		1/sig	7363/-11904.90/6.5	8268/-11904.90/5.1(0.5/4.1)/2897
		10/sig	15944/-115585.1/6.4	15792/-115585.1/6.5(1.1/5.0)/5859
ijcnn1	49990/22	1/lin	16404/-8590.158/3.0	20297/-8590.155/6.5(2.2/4.0)/7870
		10/lin	155333/-85441.01/4.2	155274/-85441.00/46.9(17.9/27.1)/63668
		1/rbf	5713/-8148.187/4.6	6688/-8148.187/3.8(0.7/2.7)/2397
		10/rbf	6415/-61036.54/3.5	12180/-61036.54/4.8(1.3/3.2)/4570
		1/poly	5223/-9693.566/2.5	7156/-9693.620/3.1(0.9/2.0)/2580
		10/poly	5890/-95821.99/2.9	7987/-95822.02/3.3(1.0/2.1)/2949
		1/sig	6796/-9156.916/7.0	6856/-9156.916/5.0(0.8/3.9)/2452
		10/sig	10090/-88898.40/6.4	12420/-88898.39/6.5(1.4/4.7)/4975
w7a	24692/300	1/lin	66382/-765.4115/0.4	72444/-765.4116/8.2(2.5/5.4)/27920
		10/lin	662877/-7008.306/1.1	626005/-7008.311/75.3(20.2/52.6)/241180
		1/rbf	1550/-1372.011/0.4	1783/-1372.010/0.5(0.1/0.4)/731
		10/rbf	4139/-10422.69/0.4	4491/-10422.70/0.8(0.2/0.6)/1792
		1/poly	758/-1479.816/0.1	2297/-1479.825/0.5(0.1/0.4)/871
		10/poly	1064/-14782.40/0.2	3591/-14782.53/0.7(0.2/0.5)/1347
		1/sig	1477/-1427.453/0.4	2020/-1427.455/0.4(0.1/0.3)/796
		10/sig	2853/-11668.85/0.3	5520/-11668.86/0.9(0.2/0.6)/2205

Table 1: Comparing LIBSVM and CGD-3pair on large two-class data classification problems. Here n is the number of data points; p is the dimension of the data points (i.e., number of features); iter, obj, and cpu are, respectively, the total number of iterations, final f-value, and total time (in minutes); kiter and kcpu are, respectively, the total number of knapsack problems solved and the total time (in minutes) to solve the knapsack problems; gcpu is the total time (in minutes) to compute/cache columns of Q and update the gradient.