SVM-Optimization and Steepest-Descent Line Search *

Nikolas List and Hans Ulrich Simon
Fakultät für Mathematik, Ruhr-Universität Bochum, D-44780 Bochum, Germany
{nikolas.list,hans.simon}@rub.de

Abstract

We consider (a subclass of) convex quadratic optimization problems and analyze decomposition algorithms that perform, at least approximately, steepest-descent exact line search. We show that these algorithms, when implemented properly, are within ϵ of optimality after $O(\log 1/\epsilon)$ iterations for strictly convex cost functions, and after $O(1/\epsilon)$ iterations in the general case.[1] Our analysis is general enough to cover the algorithms that are used in software packages like SVMTorch and (first or second order) LibSVM. To the best of our knowledge, this is the first paper coming up with a convergence rate for these algorithms without introducing unnecessarily restrictive assumptions.

1 Introduction

The term "SVM-optimization" refers to the kind of convex quadratic optimization problems that typically arise when a learning problem (e.g., classification or regression problem) is solved according to the Support Vector method [16]. Among the most widely used algorithms for SVM-optimization are decomposition algorithms. They proceed iteratively and solve a lower-dimensional subproblem in every iteration. They can be viewed as "exact line search" algorithms provided that the dimension of the subproblems is not greater than necessary. In this paper, we are particularly interested in algorithms performing, at least approximately, steepest-descent exact line search. For example, the algorithms in the software packages SVMTorch [4], and (first or second order) LibSVM [1, 6] fall in this category. Despite of their wide usage in practice, they are still not well understood from a theoretical point of view. In particular, they seem to converge faster (or under more general conditions) to optimum than one would expect from the convergence rates that have been proved so far. The main purpose of our paper is to narrow this gap.

The result that comes closest to our results is found in [3] and states that, loosely speaking, approximate steepest-descent exact line search algorithms for C-SV Classification are within ϵ of optimality after $O(\log 1/\epsilon)$ iterations (= linear convergence) provided that the cost-function is strictly convex and that some additional non-degeneracy conditions are valid. However, the dependence of the convergence rate on other problem parameters, hidden in the big "O" notation, is left completely unclear in [3]. We improve on this result in several respects:

- In the case of a strictly convex function, we do not need any additional assumption to verify linear convergence.
- Our analysis covers the general case (where the convex cost function need not necessarily be *strictly* convex) and shows that $O(1/\epsilon)$ iterations are sufficient for being within ϵ of optimality.
- We reveal the dependence on the other problem parameters explicitly.
- We prove the results in a more abstract setting so that they hold for a subclass of Convex Quadratic Optimization which covers various SVM-optimization problems (e.g., C-SV Classification, ε-SV Regression, ν-SV Classification, ν-SV Regression, 1-class SVM).

The main obstacle to a satisfactory analysis of the Steepest-Descent heuristics is that going downhill quite steeply (in the landscape formed by the cost function) does not necessarily lead to a significant cost-reduction because we might be infinitesimally close to a facet of the polyhedron containing

*This work was supported by the Deutsche Forschungsgemeinschaft Grant SI 498/8-1.

[1] See the precise bounds in the paper for the dependence on various other problem parameters.

the feasible solutions.[2] We find a surprising way around this obstacle which might be interesting in its own right. See Section 3 for details.

2 Definitions, Notations and Facts

In this section we fix some notation, and we briefly call into mind some definitions and facts from the theory of convex quadratic optimization.

Throughout the paper, we use the short-hand $[\ell] = \{1, \ldots, \ell\}$. The all-ones vector (all-zeroes) is denoted as $\vec{e}$ (as $\vec{0}$, resp.). The vector with 1 in position i and zeroes elsewhere is denoted as $\vec{e_i}$. A_i denotes the i-th column vector of a matrix A. Let Q be a symmetric positive semidefinite matrix. As usual, $\|y\|_Q = (y^\top Qy)^{1/2}$ denotes the seminorm induced by Q. The largest eigenvalue of Q is denoted as $\lambda_{max}(Q)$, and the smallest-one as $\lambda_{min}(Q)$. For a symmetric positive definite matrix Q, $\kappa(Q) = \lambda_{max}(Q)/\lambda_{min}(Q)$ denotes the condition number.

An instance of (Convex) Quadratic Optimization is given by a quadratic convex cost function and linear equality- and inequality-constraints. We are particularly interested in "Boxed Quadratic Optimization" with instances of the form

$$\min_{x \in \mathbb{R}^\ell} f(x) \text{ s.t. } Ax = b \text{ and } l \le x \le u \ , \qquad (1)$$

where the following holds:

- The cost function is of the form

  $$f(x) = \frac{1}{2} x^\top Qx - w^\top x$$

  for some symmetric positive semidefinite matrix $Q \in \mathbb{R}^{\ell \times \ell}$, and some vector $w \in \mathbb{R}^\ell$.

- $A \in \mathbb{R}^{r \times \ell}$, $b \in \mathbb{R}^r$, and $l, u \in \mathbb{R}^\ell$. We may assume that the number of equality constraints, r, coincides with the rank of matrix A. The inequalities of the form $l \le x \le u$ (understood componentwise) are called "box constraints" (for the obvious reason).

Example 1: Consider, for example, (the dual of) the C-SV Classification problem:

$$\min \frac{1}{2} x^\top Qx - \vec{e}^\top x \text{ s.t. } y^\top x = 0, \vec{0} \le x \le C\vec{e}$$

Here Q is a kernel-matrix and $y \in \{-1, 1\}^\ell$ is a vector of classification labels. Note that we may *normalize* this instance by substituting $y_i x_i$ for x_i. The normalized instance is obviously still of the form (1) and has exactly one equality constraint, namely $\vec{e}^\top x = 0$.

There are several other examples for SVM-optimization problems that can be written in the form (1) so that $l = \vec{0}$ and u has the form $\beta\vec{e}$ for some scalar β. Specifically, $\beta = C$ for C-SV Classification, $\beta = 1/\ell$ for ν-SV Classification, $\beta = C/\ell$ for ε-SV Regression and ν-SV Regression, and $\beta = 1/(\nu\ell)$ for 1-class SVM. See [15] for the meaning of the parameters C and ν.

An algorithm for Boxed Quadratic Optimization has to cope with arbitrary instances (f, A, b, l, u). In what follows, we assume that an instance (f, A, b, l, u) has been fixed.

A point $x \in \mathbb{R}^\ell$ such that $Ax = b$ and $l \le x \le u$ is called a *feasible solution.* A *feasible direction* $d \in \mathbb{R}^\ell$ for a feasible solution x is a non-zero vector $d \in \mathbb{R}^\ell$ such that $Ad = \vec{0}$ and, for every $i = 1, \ldots, \ell$,

$$x_i = u_i \Rightarrow d_i \le 0 \text{ and } x_i = l_i \Rightarrow d_i \ge 0 \ . \qquad (2)$$

(It follows that $x + \theta d$ is still feasible for every sufficiently small $\theta > 0$). $F(x)$ denotes the set of feasible directions for x. Vector d is called a *profitable direction* for x if

$$\nabla f(x)^\top d < 0 \ . \qquad (3)$$

(It follows that $x + \theta d$ has smaller cost than x for every sufficiently small $\theta > 0$). Vector d is called *q-sparse* if it has at most q non-zero components.

An *(exact) line search strategy* starts with an initial feasible solution $x^{(0)}$ and improves the current feasible (non-optimal) solution x iteratively as follows:

1. Select a feasible and profitable direction $d(x)$ for x.
2. Compute a minimizer $\theta' > 0$ for $f(x + \theta d(x))$ s.t. $l \le x + \theta d(x) \le u$ and proceed from x to the next feasible solution $x' := x + \theta' d(x)$.

The sequence evolving from an iterative application of a given strategy is denoted

$$\vec{X} = x^{(0)}, x^{(1)}, x^{(2)}, \ldots \ .$$

Throughout the paper x, x' are used as variables that run through the sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$, where x' is always one step-ahead, i.e., $x' = x^{(r+1)}$ for $x = x^{(r)}$. Then $d = d(x)$ is the variable that runs through the corresponding sequence of (feasible and profitable) directions. Whenever $d(x) \notin \text{kernel}(Q)$, we shall also be concerned with an (unconstrained) minimizer $\theta^* > 0$ of $f(x + \theta d(x))$. Note that, in contrast to $x' = x + \theta' d(x)$, $x^* := x + \theta^* d(x)$ does not need to satisfy the box-constraints.

By x^{opt}, we denote an optimal feasible solution. Function

$$\Delta(x) = f(x) - f(x^{opt})$$

measures how much x differs from optimality. For sake of brevity, we define $\Delta_n := \Delta(x^{(n)})$. Clearly,

[2] This is precisely why some authors [8, 11, 13, 7] were investigating alternative approaches like "Rate Certifying Pair" or "Maximum Gain".

the sequence Δ_n is strictly monotonously decreasing. A central question in this paper is how fast it converges to zero.

We close this section by noting some facts. The ones we start with are easy to obtain from the 1st-order optimality conditions for convex functions:

$$\nabla f(x')^\top (x - x') \geq 0 \quad (4)$$
$$\nabla f(x^*)^\top (x - x^*) = 0 \quad (5)$$

Taylor-expansion

$$f(x) - f(x') = \nabla f(x')^\top (x - x') + \frac{1}{2}\|x - x'\|_Q^2$$

around x' combined with (4) leads to

$$f(x) - f(x') \geq \frac{1}{2}\|x - x'\|_Q^2 \ . \quad (6)$$

Expanding around x^* and making use of (5), we obtain

$$f(x) - f(x^*) = \frac{1}{2}\|x - x^*\|_Q^2 \ . \quad (7)$$

The next lemma reveals some additional equalities for the unconstrained cost-reduction.

Lemma 1 *Assume that* $d = d(x) \notin \mathrm{kernel}(Q)$. *Then the following holds:*

$$f(x) - f(x^*) = \frac{1}{2}\left(\frac{\nabla f(x)^\top d}{\|d\|_Q}\right)^2 \quad (8)$$
$$f(x) - f(x^*) = \frac{1}{2}\inf_{y: d^\top \nabla f(y) \geq 0} \|x - y\|_Q^2 \quad (9)$$

Proof: Recall that $x^* = x + \theta^* d$. Equality (8) follows by calculating the minimizer θ^* for $f(x+\theta d)$ which happens to be $\theta^* = -\frac{\nabla f(x)^\top d}{\|d\|_Q^2}$.
In order to show (9), first note that $\nabla f(y) = Qy - w$. Consider the affine hyperplane

$$H := \{y: \ d^\top \nabla f(y) = 0\} = \{y: \ d^\top Q y = d^\top w\}$$

and the affine halfspaces H_+, H_- given by $d^\top \nabla f(y) \geq 0$ and $d^\top \nabla f(y) < 0$, respectively. According to (3), x belongs to H_-, and according to (5), x^* belongs to H. Clearly, any y satisfying $d^\top \nabla f(y) \geq 0$ belongs to H_+. Notice that the affine hyperplane H is Q-orthogonal on d. It follows that x^* is a Q-projection of x onto H (unique modulo kernel(Q)). From this discussion in combination with (7), equality (9) is evident. ■

3 A Close Look to Boxed Quadratic Optimization

In Section 3.1, we shall ignore the issue of choosing direction $d(x)$ cleverly. Instead, we shall focus on the following two questions:

1. How does a guaranteed cost-reduction of the form (10), taking place in every iteration, translate into a guaranteed convergence rate ?
2. How does a guaranteed "unconstrained cost-reduction" of the form (11), taking place in every iteration, translate into a guaranteed convergence rate ?

The first question simply amounts to solving recursions (like in the proof of Lemma 2 below). The second question is more challenging because the "witness" x^* for the unconstrained cost-reduction $f(x) - f(x^*)$ is not necessarily a feasible solution. However, we shall show that the sequence $\vec{X}$ decomposes into so-called delay-sequences and that the cumulative cost-reduction achieved during all iterations of a particular delay sequence is not much smaller than the unconstrained cost-reduction achieved in one of the iterations. The precise statement is found in Lemma 4 below.

In Section 3.2, we get back to the question of how the directions $d(x)$ should be chosen and provide a lower bound on the unconstrained cost-reduction achieved by Steepest Descent in every iteration.

The question how to control the length of delay sequences is postponed to Section 4 because the current section is reserved for results that hold for arbitrary instances of Boxed Quadratic Optimization, whereas our upper bounds on the length of delay sequences are valid for so-called "pairable" instances only.

3.1 From Stepwise Cost-Reduction to Convergence Rates

Let $0 < \alpha < 1$ and $\eta \in \{1, 2\}$ be fixed constants. We say that $\vec{X}$ (or the strategy generating $\vec{X}$) achieves a *cost-reduction of type* (α, η) if, for all $x \in \vec{X}$,

$$f(x) - f(x') \geq \alpha \Delta(x)^\eta \ . \quad (10)$$

Similarly, we say that $\vec{X}$ (or the strategy generating $\vec{X}$) achieves an *unconstrained cost-reduction of type* (α, η) if, for all $x \in \vec{X}$,

$$d(x) \notin \mathrm{kernel}(Q) \Rightarrow f(x) - f(x^*) \geq \alpha \Delta(x)^\eta \ . \quad (11)$$

The cost-reduction (either unconstrained or not) is called *weak* if the term $\Delta(x)^\eta$ in (10) or in (11), respectively, is replaced by $\Delta(x')^\eta$.

Lemma 2 *1. If* $\vec{X}$ *achieves a weak cost-reduction of type* $(\alpha, 1)$, *then*

$$\Delta_n \leq \left(1 - \frac{\alpha}{1+\alpha}\right)^n \Delta_0 \quad (12)$$

so that $\Delta_n < \epsilon$ *for every*

$$n \geq \left(1 + \frac{1}{\alpha}\right) \ln \frac{\Delta_0}{\epsilon} \ .$$

2. *If $\vec{X}$ achieves a weak cost-reduction of type $(\alpha, 2)$, then*
$$\Delta_n < \frac{1}{\alpha n} \cdot (\alpha \Delta_0)^{2/(n+2)} \qquad (13)$$
so that $\Delta_n < \epsilon$ for every
$$n \geq \max\left\{\frac{2}{\alpha\epsilon}, 2\log(\alpha\Delta_0)\right\} \ .$$

The proof of Lemma 2 is given in Section A.

We say that $x^{(n)}, \ldots, x^{(m-1)}$ is a *delay sequence of length $m-n$ within $\vec{X}$* if, for all $s, t = n, \ldots, m-1$, the following holds:
$$\nabla f(x^{(s)})^\top (x^{(t+1)} - x^{(t)}) < 0 \ . \qquad (14)$$
A delay sequence $x^{(n)}, \ldots, x^{(m-1)}$ such that $x^{(m-1)}$ is not yet optimal is said to be *maximal* if one of the following conditions is valid:

1. There exists an $s \in \{n, \ldots, m-1\}$ such that
$$\nabla f(x^{(s)})^\top (x^{(m+1)} - x^{(m)}) \geq 0 \ . \qquad (15)$$
2. There exists a $t \in \{n, \ldots, m-1\}$ such that
$$\nabla f(x^{(m)})^\top (x^{(t+1)} - x^{(t)}) \geq 0 \ . \qquad (16)$$

Lemma 3 *Assume that $x^{(n)}, \ldots, x^{(m-1)}$ is a maximal delay sequence so that one of the conditions (15), (16) is valid. Then (15) implies that $d(x^{(m)}) \notin$ kernel(Q). Similarly (16) implies that $d(x^{(t)}) \notin$ kernel(Q).*

Proof: Note that $\nabla f(x)^\top d = -w^\top d$ for any $d \in$ kernel(Q). Thus, a direction belonging to the kernel of Q is either profitable for every x or it is unprofitable for every x. Condition (15) implies that direction $d(x^{(m)})$, which is clearly profitable for $x^{(m)}$, is unprofitable for $x^{(s)}$ so that it cannot belong to kernel(Q). Similarly, Condition (16) implies that direction $d(x^{(t)})$, which is clearly profitable for $x^{(t)}$, is unprofitable for $x^{(m)}$ so that it cannot belong to kernel(Q) either. ■

The sequence $\vec{X} = x^{(1)}, x^{(2)}, \ldots,$ decomposes into maximal delay sequences (and possibly a final delay sequence that ends in an optimal feasible solution) in the obvious fashion. $\vec{X}$ is said to have a *delay bounded by L* if none of the delay sequences in the decomposition of $\vec{X}$ has a length exceeding L.

Lemma 4 *For every maximal delay sequence*
$$x^{(n)}, \ldots, x^{(m-1)}$$
within $\vec{X}$, there exists $x \in \{x^{(n)}, \ldots, x^{(m)}\}$ such that
$$f(x^{(n)}) - f(x^{(m)}) > \frac{1}{2}(f(x) - f(x^*)) \ . \qquad (17)$$
Moreover, if $\vec{X}$ satisfies (11), then
$$f(x^{(n)}) - f(x^{(m)}) > \frac{\alpha}{2}\Delta(x)^\eta \geq \frac{\alpha}{2}\Delta_m^\eta \ . \qquad (18)$$

Proof: Since $x^{(n)}, \ldots, x^{(m-1)}$ is a maximal delay sequence, Condition (14) and one of the conditions (15), (16) must be satisfied. Let us first assume that condition (15) holds. Observe that
$$f(x^{(n)}) - f(x^{(m)}) \geq$$
$$(f(x^{(s)}) - f(x^{(m-1)})) + (f(x^{(m-1)}) - f(x^{(m)})) \ .$$
According to (6),
$$f(x^{(m-1)}) - f(x^{(m)}) \geq \frac{1}{2}\|x^{(m-1)} - x^{(m)}\|_Q^2 \ .$$
Taylor-expansion around $x^{(m-1)}$ shows that $f(x^{(s)}) - f(x^{(m-1)})$ equals
$$\nabla f(x^{(m-1)})^\top (x^{(s)} - x^{(m-1)}) + \frac{1}{2}\|x^{(s)} - x^{(m-1)}\|_Q^2 \ .$$
The first term written as a telescope-sum looks like
$$\sum_{t=s}^{m-2} \nabla f(x^{(m-1)})^\top (x^{(t)} - x^{(t+1)})$$
and is strictly positive according to (14). We may now conclude that $f(x^{(n)}) - f(x^{(m)})$ is greater than
$$\frac{1}{2}\left((\|x^{(s)} - x^{(m-1)}\|_Q^2 + \|x^{(m-1)} - x^{(m)}\|_Q^2\right) \ .$$
An application of the triangle inequality,
$$\|x^{(s)} - x^{(m-1)}\|_Q + \|x^{(m-1)} - x^{(m)}\|_Q \geq \|x^{(s)} - x^{(m)}\|_Q \ ,$$
and of some calculus yields
$$f(x^{(n)}) - f(x^{(m)}) > \frac{1}{4}\|x^{(s)} - x^{(m)}\|_Q^2 \ .$$
Let $d := x^{(m+1)} - x^{(m)}$. Condition (15) reads as $d^\top \nabla f(x^{(s)}) \geq 0$, whereas $d^\top \nabla f(x^{(m)}) < 0$. We conclude from (9) that $\frac{1}{2}\|x^{(s)} - x^{(m)}\|_Q^2$ is an upper bound on the unconstrained cost-reduction at $x^{(m)}$. Our discussion shows that (17) holds for $x = x^{(m)}$. Note that $d(x^{(m)}) \notin$ kernel(Q) according to Lemma 3. Thus, if $\vec{X}$ satisfies (11), we clearly obtain (18) . Finally note: if condition (16) is assumed instead of (15), a similar reasoning applies with $d := x^{(t+1)} - x^{(t)}$ and $x^{(t)}$ in the role of x. ■

Lemma 2 applied to the sequence Δ_{nL} yields the following result:

Corollary 5 *1. If $\vec{X}$ has a delay bounded by L and achieves an unconstrained cost-reduction of type $(\alpha, 1)$, then*
$$\Delta_{nL} \leq \left(1 - \frac{\alpha/2}{1 + \alpha/2}\right)^n \Delta_0$$
so that $\Delta_n \leq \epsilon$ for every
$$n \geq \left(\frac{2}{\alpha} + 1\right) L \ln\frac{\Delta_0}{\epsilon} \ .$$

2. *If $\vec{X}$ has a delay bounded by L and achieves an unconstrained cost-reduction of type $(\alpha, 2)$, then*

$$\Delta_{nL} \le \frac{2}{\alpha n} \cdot \left(\frac{\alpha\Delta_0}{2}\right)^{2/(n+2)}$$

so that $\Delta_n \le \epsilon$ for

$$n = \max\left\{\frac{4L}{\alpha\epsilon}, 2L\log\left(\frac{\alpha\Delta_0}{2}\right)\right\} .$$

3.2 Unconstrained Cost Reduction Achieved by Steepest Descent

Let x be a feasible non-optimal solution, and let $\|\cdot\|$ be a vector norm on $\mathbb{R}^\ell$. We say that d^* is a *steepest-descent direction for x w.r.t.* $\|\cdot\|$ if d^* is a maximizer of $-\nabla f(x)^\top d$ s.t. $\|d\| \le 1$ and to feasibility of d. In this case, $-\nabla f(x)^\top d^* > 0$, i.e., a steepest-descent direction for x is profitable for x. Similarly, d is called a *τ-approximate steepest-descent direction for x w.r.t.* $\|\cdot\|$ if d is feasible for x, $\|d\| \le 1$ and $-\nabla f(x)^\top d \ge -\tau\nabla f(x)^\top d^*$.

Lemma 6 *Assume that Q is positive definite so that $\|\cdot\|_Q$ is a vector norm. Then the following holds. If $d(x)$ is a steepest-descent direction for x w.r.t. $\|\cdot\|_Q$, then $f(x) - f(x^*) \ge \Delta(x)$. More generally, if $d(x)$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_Q$, then $f(x) - f(x^*) \ge \tau^2 \cdot \Delta(x)$*

Proof: According to (8), a steepest-descent direction for x w.r.t. $\|\cdot\|_Q$ leads to the the largest possible unconstrained cost-reduction at x, which is certainly not smaller than $\Delta(x) = f(x) - f(x^{opt})$. Similarly, a τ-approximate steepest-descent direction misses the largest possible unconstrained cost-reduction at most by a factor of τ^2. ■

Recall that, for any two vector norms $\|\cdot\|_A, \|\cdot\|_B$, there exist constants $0 < c < C$ such that

$$c\|\cdot\|_A \le \|\cdot\|_B \le C\|\cdot\|_A \ . \tag{19}$$

With this notation, the following holds:

Lemma 7 *If $\hat{d}_B$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_B$, then $c \cdot \hat{d}_B$ is a $\frac{c\tau}{C}$-approximate steepest-descent direction w.r.t. $\|\cdot\|_A$.*

Proof: For $T = A, B$, let d_T be the direction that maximizes $-\nabla f(x)^\top d$ subject to $\|d\|_T \le 1$. Note that $\|\frac{d_A}{C}\|_B \le \|d_A\|_A \le 1$ and $\|cd_B\|_A \le \|d_B\|_B \le 1$. It follows that

$$-\nabla f(x)^\top \frac{d_A}{C} \le -\nabla f(x)^\top d_B \le -\frac{1}{c\tau}\nabla f(x)^\top (c\hat{d}_B) \ ,$$

which completes the proof. ■

Corollary 8 *Assume that Q is positive definite. Then the following holds. If $d(x)$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_1$, then*

$$f(x) - f(x^*) \ge \frac{\tau^2}{\ell\kappa(Q)} \cdot \Delta(x) \ . \tag{20}$$

Proof: From

$$\|d\|_Q^2 \le \lambda_{max}(Q)\|d\|_2^2 \le \lambda_{max}\|d\|_1^2 \tag{21}$$

and

$$\|d\|_Q^2 \ge \lambda_{min}(Q)\|d\|_2^2 \ge \frac{\lambda_{min}(Q)}{\ell}\|d\|_1^2,$$

we conclude that

$$\frac{1}{\sqrt{\lambda_{max}(Q)}}\|d\|_Q \le \|d\|_1 \le \sqrt{\frac{\ell}{\lambda_{min}(Q)}}\|d\|_Q \ .$$

According to Lemma 7, $\frac{d(x)}{\sqrt{\lambda_{max}(Q)}}$ is a $\tau\sqrt{\frac{\lambda_{min}(Q)}{\lambda_{max}(Q)\ell}}$-approximate steepest-descent-direction w.r.t. $\|\cdot\|_Q$. Now, (20) immediately follows from Lemma 6. ■

Corollary 9 *If Q is positive definite and, for every $x \in \vec{X}$, $d(x)$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_1$, then $\vec{X}$ achieves an unconstrained cost-reduction of type $(\alpha, 1)$ for*

$$\alpha = \frac{\tau^2}{\ell\kappa(Q)}. \tag{22}$$

We move on and consider the general case of a semidefinite matrix Q. Since $\|\cdot\|_Q$ is a seminorm, we have in general no relation of the form (19) between $\|\cdot\|_Q$ and $\|\cdot\|_1$. However, one still gets results in the spirit of Corollaries 8 and 9:

Theorem 10 *If $d(x)$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_1$, then, for every optimal feasible solution x^{opt},*

$$f(x) - f(x^*) \ge \frac{\tau^2}{2\|x^{opt} - x\|_1^2\lambda_{max}(Q)} \cdot \Delta(x)^2 \ . \tag{23}$$

Proof: Let $d = d(x)$. By our assumption, $\|d\|_1 \le 1$. From (21) we conclude that $\|d\|_Q^2 \le \lambda_{max}(Q)$. Recall that x^* is the point of minimum cost on the ray starting from x in direction d. A straightforward calculation shows that, for every $\theta > 0$, the following holds:

$$\begin{aligned} f(x) - f(x^*) &\ge -\theta\nabla f(x)^\top d - \frac{1}{2}\theta^2\|d\|_Q^2 \\ &\ge -\theta\nabla f(x)^\top d - \frac{1}{2}\theta^2\lambda_{max}(Q) \end{aligned}$$

It easily follows that

$$f(x) - f(x^*) \ge \frac{(\nabla f(x)^\top d)^2}{2\lambda_{max}(Q)} \ .$$

Consider the direction $d^{opt} := x^{opt} - x$. Taylor-expansion of $f(x^{opt})$ around x yields

$$-\nabla f(x)^\top d^{opt} = f(x) - f(x^{opt}) + \frac{1}{2}\|d^{opt}\|_Q^2 \geq \Delta(x) \ .$$

Since d is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_1$, we get

$$-\nabla f(x)^\top d \geq -\tau \nabla f(x)^\top \frac{d^{opt}}{\|d^{opt}\|_1} \geq \frac{\tau \Delta(x)}{\|x^{opt} - x\|_1} \ . \tag{24}$$

Putting everything together, we arrive at (23). ■

Since $\|x^{opt} - x\|_1 \leq \|u - l\|_1$ for every x, we obtain the following

Corollary 11 *If, for every $x \in \vec{X}$, $d(x)$ is a τ-approximate steepest-descent direction for x w.r.t. $\|\cdot\|_1$, then $\vec{X}$ achieves an unconstrained cost-reduction of type $(\alpha, 2)$ for $\alpha = \frac{\tau^2}{2\|u-l\|_1^2 \lambda_{max}(Q)}$.*

Note that, in practice, the upper bound $\|u - l\|_1$ on $\|x^{opt} - x\|_1$ might be overly pessimistic. For example, if s denotes the number of support vectors and if $x^{(0)} = \vec{0}$, then $\|x^{opt} - x^{(0)}\|_1$ is bounded from above by the sum of the s largest side lengths of the box spanned by l and u (which is, in general, a much better bound than $\|u - l\|_1$).

4 Steepest Descent Strategies for SVM-Optimization

The SMO-algorithm by Platt [14] implemented with the Violating-Pair selection rule from [9] solves C-SV Classification by means of steepest-descent line search. We refer to this implementation of SMO by the short-hand MVP (= Maximum Violating Pair) in what follows. In Section 4.1, we show that MVP (and related algorithms including some second order versions of MVP) have an ℓ-bounded (resp. 2ℓ-bounded) delay.

In Section 4.2, these results are generalized to pairable instances of Boxed Quadratic Optimization so that various SVM-optimization problems can be addressed.

4.1 C-SV Classification Revisited

Consider a normalized instance of C-SV Classification and recall that such an instance is of the form (1) where the equality matrix A has a single row containing the all-ones vector $\vec{e}$. Thus the set $F(x)$ of feasible directions consists of all non-zero vectors $d \in \mathbb{R}^\ell$ that satisfy (2) and the equality constraint

$$\vec{e}^\top d = \sum_{i=1}^{\ell} d_i = 0 \ . \tag{25}$$

If we additionally impose the condition of 2-sparsity, d must be a scalar multiple of $\vec{e_i} - \vec{e_j}$ for some indices $1 \leq i, j \leq \ell$. A direction of this form is profitable for x iff

$$\nabla f(x)^\top d = \nabla f(x)_i - \nabla f(x)_j < 0 \ .$$

The MVP-strategy picks a 2-sparse feasible direction $d(x) = \vec{e_i} - \vec{e_j}$ which maximizes $\nabla f(x)_j - \nabla f(x)_i$. Since $\|d\|_1 = 2\|d\|_\infty$ for every 2-sparse feasible direction, we may equivalently think of picking a 2-sparse feasible direction $d \in F(x)$ that maximizes $\nabla f(x)_j - \nabla f(x)_i$ subject to $\|d\|_1 \leq 1$. We call into mind the following well-known result:

Lemma 12 *Among the maximizers d of $-\nabla f(x)^\top d$ subject to (2), (25) and $\|d\|_1 \leq 1$, there is always a 2-sparse direction. It follows that MVP picks a steepest-descent direction for x w.r.t. $\|\cdot\|_1$.*

Proof: Writing d in the form $d = d^+ - d^-$ such that $d^+, d^- \geq 0$, it is easy to rewrite the maximization problem as a linear program in standard form with two equality constraints so that the basic feasible solutions (among which there is always an optimal-one) are 2-sparse. The final conclusion about MVP is obvious. ■

Another way of saying how MVP chooses a pair (i^*, j^*) and the corresponding direction $d(x)$ is as follows:

1. Pick an index $i^* \in [\ell]$ that minimizes $\nabla f(x)_i$ s.t. $x_i < u_i$.
2. Pick an index $j^* \in [\ell]$ that maximizes $\nabla f(x)_j$ s.t. $x_j > l_j$.
3. Set $d(x) = \overrightarrow{e_{i^*}} - \overrightarrow{e_{j^*}}$.

There exist variants of MVP (e.g., the second order variant in [6]) that fit into the following scheme:

1. Pick an index $i^* \in [\ell]$ that minimizes $\nabla f(x)_i$ s.t. $x_i < u_i$.
2. Pick an index $j^* \in [\ell]$ such that $\nabla f(x)_{i^*} - \nabla f(x)_{j^*} < 0$ and $x_{j^*} > l_{j^*}$
3. Set $d(x) = \overrightarrow{e_{i^*}} - \overrightarrow{e_{j^*}}$.

In the sequel, we call such strategies "MVP-like".

Theorem 13 *Every MVP-like strategy for C-SV Classification has 2ℓ-bounded delay. Moreover, MVP has ℓ-bounded delay.*

Proof: Consider a fixed but arbitrary delay sequence $x^{(n)}, \ldots, x^{(m-1)}$. In the sequel, x denotes a variable that runs through $x^{(n)}, \ldots, x^{(m-1)}$, H_k^+ denotes the affine hyperplane given by equation $x_k = u_k$, and H_k^- denotes the affine hyperplane given by equation $x_k = l_k$. If $d(x) = \vec{e_i} - \vec{e_j}$, we say that *$x$ moves upward in dimension i* and *downward in dimension j*. If $x_i = u_i$ after an upward-move in dimension i, we say that *x hits H_i^+*. Symmetrically, if $x_j = l_j$ after a downward-move in dimension j, we say that x *hits* H_j^-. We have to show that the length $m-n$ of the delay sequence is bounded above by 2ℓ. This is immediate from the following two claims:

1. Consider an iteration that does not finish the delay sequence and let $\overrightarrow{e_i} - \overrightarrow{e_j}$ be the direction chosen in this iteration. Then, after the next move, x hits H_i^+ or x hits H_j^-.

2. During one and the same delay sequence, x cannot move downward in dimension i after it had made an upward-move in dimension i before.

As for the first claim, it suffices to show that an iteration moving x in direction $\overrightarrow{e_i} - \overrightarrow{e_j}$ so that x neither hits H_i^+ nor H_j^- finishes the delay sequence. To this end, we argue as follows. Since the chosen direction is profitable, we know that $\nabla f(x)_i - \nabla f(x)_j = \nabla f(x)^\top(\overrightarrow{e_i} - \overrightarrow{e_j}) < 0$ before the move. Since x by assumption does not hit one of the hyperplanes H_i^+, H_j^-, we know that $\nabla f(x)_i - \nabla f(x)_j = 0$ after the move. But turning a profitable direction into an unprofitable-one is something that cannot happen in one and the same delay sequence. Thus, the first claim is valid.

In order to prove the second claim, we consider the following relation on indices from $[\ell]$: $i \dot{\prec} j$ means, by definition, that direction $\overrightarrow{e_i} - \overrightarrow{e_j}$ is chosen in one of the iterations during the delay sequence, say in iteration r. Since this direction is profitable, we get

$$\nabla f(x^{(r)})_i - \nabla f(x^{(r)})_j = \nabla f(x^{(r)})^\top(\overrightarrow{e_i} - \overrightarrow{e_j}) < 0 \ ,$$

which implies that $\nabla f(x^{(r)})_i < \nabla f(x^{(r)})_j$. But, as mentioned above already, a direction that is profitable in one iteration of the delay-sequence is profitable in all iterations of the delay-sequence. Thus,

$$\forall s = n, \ldots, m-1 : \nabla f(x^{(s)})_i < \nabla f(x^{(s)})_j \quad (26)$$

Let "$\prec$" and "$\preceq$" be the transitive and the reflexive-transitive closure of "$\dot{\prec}$", respectively. Since $i \prec j$ implies (26), it is a partial ordering. For every $r = n, \ldots, m-1$, let

$$I_r = \{i : x_i^{(r)} < u_i\} \ .$$

The crucial observation is that, for all $r = n, \ldots, m-2$, the following holds:

$$\forall j \in I_{r+1}, \exists i \in I_r : i \preceq j \quad (27)$$

To see why this is true, pick an arbitrary but fixed index j from I_{r+1}. Since $j \in I_r$ would confirm (27), let us assume that $j \notin I_r$, i.e., $x_j^{(r)} = u_j$. Since $j \in I_{r+1}$ implies that $x_j^{(r+1)} < u_j$, iteration r makes x moving downward in dimension j. Thus, there exists $i \in I_r$ such that this move is in direction $\overrightarrow{e_i} - \overrightarrow{e_j}$. But this implies that $i \dot{\prec} j$. Our discussion shows that (27) is valid.

Let us get back to the proof of claim 2 above and assume that x is moved upward in dimension i^*, say in iteration r. We have to show that x is not moved downward in dimension i^* during one of the subsequent iterations of the same delay sequence. Note that i^* must be a minimizer of $\nabla f(x^{(r)})_i$ subject to $i \in I_r$ since the directions are chosen by an MVP-like strategy. It follows that i^* is among the minimal elements of I_r. Let us assume, for sake of contradiction, that x is moved downward in dimension i^* during some iteration $r' > r$ of the same delay sequence, say by a move in direction $\overrightarrow{e_j} - \overrightarrow{e_{i^*}}$ for some $j \in I_{r'}$. It follows that $j \dot{\prec} i^*$. According to (27) on the other hand, there must exist an index $i \in I_r$ such that $i \preceq j$. It follows that $i \prec i^*$, which contradicts to the minimality of i^* within I_r. Thus Claim 2 must be valid, and the verification of the delay-bound 2ℓ is complete.

For strategy MVP, a symmetry-argument applies and the second claim above remains valid after an exchange of the words "downward" and "upward". This immediately yields delay-bound ℓ. ■

Since MVP performs steepest descent and has ℓ-bounded delay, Corollary 5 applies to MVP. Since the resulting convergence-rate is valid not only for C-SV Classification but also for the more general problem "Pairable Boxed Quadratic Optimization", we postpone its specification to Section 4.2.

4.2 Pairable Boxed Quadratic Optimization Revisited

List [10] introduced the following notion. An instance (f, A, b, l, u) of Boxed Quadratic Optimization is called *decomposable by pairing* or simply *pairable* if any collection of pairwise linear independent columns of A is linear independent. *Pairable Boxed Quadratic Optimization* means Boxed Quadratic Optimization restricted to pairable input instances. Surprisingly many SVM Optimization problems fall in this category (e.g., all problems mentioned in Example 1).

For the remainder of this section, we assume that (f, A, b, l, u) is a fixed but arbitrary pairable instance. Let r denote the rank of A. Then, A has r linear independent columns, say $A_1, \ldots, A_r$ (after renumbering if necessary), so that the following holds. For every $i \in [\ell]$, there exists a unique $k := k(i) \in [r]$ and a unique constant c_i such that $A_i = c_i A_k$. Let us define

$$I_k := \{i \in [\ell] : \ k(i) = k\} \ .$$

We may assume that $c_i \geq 0$ for every $i \in [\ell]$ (after a suitable variable substitution if necessary).

Note that the set $F(x)$ of feasible directions consists of all non-zero vectors $d \in \mathbb{R}^\ell$ that satisfy (2) together with the equality constraints $Ad = \vec{0}$. If we additionally assume 2-sparsity of d and $\|d\|_1 = 1$, then d must be of the form

$$\overrightarrow{d_{i,j}} := \frac{1}{c_i + c_j} \left(c_j \overrightarrow{e_i} - c_i \overrightarrow{e_j}\right) \tag{28}$$

for some indices $1 \le i, j \le \ell$ such $k(i) = k(j)$. Note that

$$\begin{aligned} \nabla f(x)^\top \overrightarrow{d_{i,j}} &= \frac{1}{c_i + c_j} \left(c_j \nabla f(x)_i - c_i \nabla f(x)_j\right) \\ &= \frac{c_i c_j}{c_i + c_j} \left(\frac{\nabla f(x)_i}{c_i} - \frac{\nabla f(x)_j}{c_j}\right) \ . \end{aligned}$$

Strategy SD is defined to pick a 2-sparse and feasible direction $d(x)$ of unit L_1-norm which maximizes $-\nabla f(x)^\top d$. In other words, it picks a maximizer (i^*, j^*) of

$$\frac{c_i c_j}{c_i + c_j} \left(\frac{\nabla f(x)_j}{c_j} - \frac{\nabla f(x)_i}{c_i}\right) \tag{29}$$

s.t. $x_i < u_i$, $x_j > l_j$, and then chooses direction $d(x) := \overrightarrow{d_{i^*,j^*}}$. The following result generalizes Lemma 12:

Lemma 14 *Among the maximizers d of $-\nabla f(x)^\top d$ subject to (2), $Ad = \vec{0}$ and $\|d\|_1 = 1$, there is always a 2-sparse direction. It follows that SD picks a steepest-descent direction for x w.r.t. $\|\cdot\|_1$.*

Proof: Let $\mathcal{P}$ denote the maximization problem described in Lemma 14 and let g_* denote its optimal value. For every fixed but arbitrary $k \in [r]$, consider the subproblem $\mathcal{P}_k$ where d_i is set to 0 for every $i \notin I_k$, and let g_k denote its optimal value. The corresponding submatrix of A consists of all columns that are scalar multiples of A_k so that it essentially contains one equality constraint. A second equality constraint will result from the condition $\|d\|_1 \le 1$ (that may be replaced by $\|d\|_1 = 1$ because an optimal solution satisfies $\|d\|_1 \le 1$ without slackness). As in the proof for Lemma 12, it follows that the basic feasible solutions for $\mathcal{P}_k$ are 2-sparse. On one hand, clearly $g^* \ge g_k$ for every $k \in [r]$. On the other hand, since $\|d\|_1 = 1$, g^* is a convex combination of $g_1, \ldots, g_k$. Thus, there exists an index $k' \in [r]$ such that $g_* = g_{k'}$. Since $\mathcal{P}_{k'}$ has 2-sparse basic feasible solutions, $\mathcal{P}$ has is a 2-sparse maximizer. The final conclusion about SD is obvious. ■

In the sequel, we discuss another strategy named ASD (= Approximate Steepest Descent). It is simple to implement, and it computes a feasible and profitable direction in $O(\ell)$ steps as follows:

1. For $k = 1, \ldots, r$, choose $i'(k)$ so as to minimize $\frac{1}{c_i} \nabla f(x)_i$ s.t. $i \in I_k$ and $x_i < u_i$.

2. For $k = 1, \ldots, r$, choose $j'(k)$ so as to maximize $\frac{1}{c_j} \nabla f(x)_j$ s.t. $j \in I_k$ and $x_j > l_j$.

3. Among all $(i'(k), j'(k))$, $k = 1, \ldots, r$, choose the pair which, in the sense of (28), induces the direction vector d with the largest value of $-\nabla f(x)^\top d$.

Any strategy that, for every $k = 1, \ldots, r$, selects two indices $i' = i'(k), j' = j'(k) \in I_k$ such that the first of the above conditions for ASD holds and such that direction vector $\overrightarrow{d_{i',j'}}$ is feasible and profitable is called "ASD-like".

When applied to C-SV Classification, ASD collapses to MVP and, similarly, ASD-like strategies collapse to MVP-like strategies. This follows directly from the fact that a normalized instance of C-SV Classification has an equality matrix of the form $A = [1, \ldots, 1] \in \mathbb{R}^{1 \times \ell}$ so that $c_1 = \cdots = c_\ell = 1$ and $k(1) = \cdots = k(\ell) = 1$. Similarly, when applied to ν-SV Classification, ASD collapses to the strategy from [2]. Moreover, the following holds:

Lemma 15 *Let $c_{min}(k) := \min_{i \in I_k} c_i$, $c_{max}(k) := \max_{i \in I_k} c_i$, and*

$$\tau(A) := \min_{k=1,\ldots,r} \frac{c_{min}(k)}{c_{max}(k)} \ .$$

With this notation, it holds that ASD chooses a $\tau(A)$-approximate steepest-descent direction.

Proof: We have to compare the objective values $\nabla f(x)^\top d$ achieved by SD and ASD, respectively. Since both strategies choose a 2-sparse and feasible direction of unit L_1-norm, both objective values are of the form (29). Define an auxiliary function $h(u, v) := uv/(u + v)$. Let $i^*, j^* \in I_k$ denote the pair of indices chosen by SD, and let $i' = i'(k), j' = j'(k))$ be the pair of indices in I_k determined by ASD. The resulting directions are denoted as d^* and d', respectively. It suffices to show that d' is a $\tau(A)$-approximate steepest-descent direction. It follows from the definition of ASD that

$$\left(\frac{\nabla f(x)_{j'}}{c_{j'}} - \frac{\nabla f(x)_{i'}}{c_{i'}}\right) \ge \left(\frac{\nabla f(x)_{j^*}}{c_{j^*}} - \frac{\nabla f(x)_{i^*}}{c_{i^*}}\right) \ .$$

In view of (29), a simple computation now shows that

$$-\nabla f(x)^\top d' \ge \frac{h(c_{i'}, c_{j'})}{h(c_{i^*}, c_{j^*})} \left(-\nabla f(x)^\top d^*\right) \ .$$

Since $h(u, v)$ is monotonously increasing in both arguments, it follows that

$$\frac{-\nabla f(x)^\top d'}{-\nabla f(x)^\top d^*} \ge \frac{h(c_{min}, c_{min})}{h(c_{max}, c_{max})} = \frac{c_{min}(k)}{c_{max}(k)} \ge \tau(A) \ ,$$

which concludes the proof. ■

The following result is a straightforward generalization of Theorem 13:

Theorem 16 *Every ASD-like strategy for Pairable Boxed Quadratic Optimization has 2ℓ-bounded delay. Moreover, ASD has ℓ-bounded delay.*

We briefly sketch the proof of Theorem 16. A move in direction $d(x) = \overrightarrow{d_{i',j'}}$ moves x upward in dimension i' and downward in dimension j'. We have therefore a similar starting point as in the proof of Theorem 13. The main differences to the special case of MVP-like strategies are as follows. First, the full set $[\ell]$ of dimensions splits into classes $I_1, \ldots, I_r$. The indices i', j' chosen in an iteration always belong to the same class. Second (compare with (28)), x moves upward in dimension i' with speed $c_{j'}$ and downward in dimension j' with speed $c_{i'}$ (whereas, in the proof of Theorem 13, we had speed 1 in both cases). But these differences do not cause much trouble. One can show that (slight adaptations of) the two central claims within the proof for Theorem 13 are still valid (where the full set of dimensions is discussed "classwise"). We omit the details.

Since (according to Lemma 15) ASD performs $\tau(A)$-approximate steepest descent and has ℓ-bounded delay, we may now apply Corollaries 9, 11, and 5 so as to obtain the following convergence-rate:

Corollary 17 *Assume that sequence $\vec{X}$ is produced by a decomposition algorithm that applies strategy ASD. Then the following threshold n_0 makes sure that $\Delta_n \leq \epsilon$ for every $n \geq n_0$ provided that Q is positive definite:*

$$n_0 = \frac{2\ell^2\kappa(Q) + \ell}{\tau(A)^2} \cdot \ln\frac{\Delta_0}{\epsilon} \tag{30}$$

In the general case of a positive semidefinite matrix Q, threshold n_0 can be chosen as the maximum of the following two terms:

$$\frac{8\ell\|u-l\|_1^2\lambda_{max}(Q)}{\tau(A)^2\epsilon}\ ,\ \log\frac{\tau(A)^2\Delta_0}{2\|u-l\|_1^2\lambda_{max}(Q)} \tag{31}$$

See the concluding remarks below for the convergence-rates that we obtain from Corollary 17 in the special case of SVM-optimization.

We close this section by a short discussion of the following strategy named "2nd-order ASD":

1. For $k = 1, \ldots, r$, choose $i'(k) \in I_k$ according to the ASD-strategy.
2. For $k = 1, \ldots, r$, choose $j'(k)$ so as to maximize the unconstrained cost-reduction in direction $\overrightarrow{d_{i'(k),j}}$ s.t. $j \in I_k$ and $x_j > l_j$.
3. Among all $(i'(k), j'(k))$, $k = 1, \ldots, r$, choose the pair which leads to the largest unconstrained cost-reduction.

Like ASD, it is simple to implement, and it computes a feasible and profitable direction in $O(\ell)$ steps. Moreover, the following holds:

1. 2nd-order ASD is ASD-like.
2. Any unconstrained cost-reduction achieved by ASD at a given point x is, more than ever, achieved by 2nd-order ASD.

These observations imply that the convergence rate described in Corollary 17 carries over from ASD to 2nd-order ASD (except that n_0 must be twice as large because the calculation for ASD was based on delay-bound ℓ but, for 2nd-order ASD, we can guarantee delay-bound 2ℓ only). We finally would like to mention that, for the special case of C-SV and ν-SV Classification, 2nd-order ASD collapses to the second order strategy from [6].

Analysis of Stopping Criteria: Let us assume that directions $d(x)$ are always normalized w.r.t. $\|\cdot\|_1$. The following rule (slightly generalized to arbitrary pairable instances) is most often used in practice: stop and return the current feasible solution x as soon as a direction $d(x)$ is selected such that

$$\delta(x) := -\nabla f(x)^\top d(x) < \epsilon\ .$$

One can show that, under assumptions specified below, $\delta(x)$ relates to the (unknown) quantity $\Delta(x) = f(x) - f(x^{opt})$ according to

$$\Delta(x^{(n)}) \quad \leq \quad \frac{\|u-l\|_1\delta(x^{(n)})}{\tau}\ , \tag{32}$$

$$\Delta(x^{(n)}) \quad \geq \quad \frac{\delta(x^{n'})^2}{4\lambda_{max}(Q)}\ , \tag{33}$$

for some (properly chosen) $n' \in \{n, \ldots, n+L\}$. Here is a sketch of proof. Inequality (32) is valid for any τ-approximate steepest-descent strategy and follows directly from (24).[3] Inequality (33) is valid for any strategy with L-bounded delay as can be seen by the following reasoning, which makes use of an $x \in \{x^{(n)}, \ldots, x^{(n+L)}\}$ that is chosen in accordance with Lemma 4:

$$\Delta(x^{(n)}) \quad \geq \quad f(x^{(n)}) - f(x^{(n+L)}) \overset{L.\ 4}{\geq}$$

$$\frac{1}{2}(f(x) - f(x^*)) \quad \overset{(8)}{=} \quad \left(\frac{\nabla f(x)^\top d(x)}{2\|d(x)\|_Q}\right)^2$$

Now (33) follows from (21) and the definition of δ.

Concluding Remarks: For the sake of a simple presentation, we did not try to get the full "horse power" out of our upper bounds on the number of iterations. Some improvements, however, are quite straightforward. For example, let us denote the largest eigenvalue of any $q \times q$ principal submatrix of Q as $\lambda_{max,q}(Q)$, and let $\kappa_q(Q) := \lambda_{max,q}(Q)/\lambda_{min}(Q)$. With this notation, some of our results can be sharpened:

[3] In our paper, $\Delta(x)$ refers to the dual SVM-optimization problem. Using techniques from [12], one can show that the right-hand side in (32) even upper-bounds $\Delta_{primal}(x^{(n)}) + \Delta_{dual}(x^{(n)})$.

- We may substitute $\kappa_q(Q)$ for $\kappa(Q)$ in (20) and in (22), respectively, provided that $d(x)$ is assumed as q-sparse.
- The analogous manipulations, with $\lambda_{max,q}(Q)$ substituted for $\lambda_{max}(Q)$, are possible in Theorem 10 and Corollary 11, respectively, provided that $d(x)$ is assumed as q-sparse.
- Consequently, we may substitute $\kappa_2(Q)$ for $\kappa(Q)$ in (30), and $\lambda_{max,2}(Q)$ for $\lambda_{max}(Q)$ in (31). For all SVM-optimization problems mentioned in Example 1, we may furthermore substitute 1 for $\tau(A)$ in (30) and in (31), respectively (which follows from a close inspection of the corresponding quadratic programs in [15]).

These considerations lead to the following result:

Corollary 18 *Assume that strategy ASD is applied to one of the problems C-SV Classification, ε-SV Regression, ν-SV Classification, ν-SV Regression, and 1-class SVM, respectively. The the following setting of threshold n_0 makes sure that $\Delta_n \leq \epsilon$ for every $n \geq n_0$ (or for every $n \geq 2n_0$ if 2nd-order ASD is applied):*

1. *If Q is positive definite, then let*
$$n_0 := \left(2\ell^2\kappa_2(Q) + \ell\right) \cdot \ln \frac{\Delta_0}{\epsilon} = \tilde{O}(\ell^2) \ .$$
2. *If Q is an arbitrary positive semidefinite matrix, then n_0 can be chosen as the maximum of the terms*
$$8\frac{\ell}{\epsilon}\|u-l\|_1^2\lambda_{max,2}(Q)\ ,\ \log\frac{\Delta_0}{2\|u-l\|_1^2\lambda_{max,2}(Q)} \ .$$
Moreover, the following holds (compare with Example 1):
    - *For C-SV Classification, $\|u-l\|_1 = C\ell$, which leads to $n_0 = \tilde{O}(\ell^3 C^2/\epsilon)$.*
    - *For ν-SV Classification, $\|u-l\|_1 = 1$, which leads to $n_0 = \tilde{O}(\ell/\epsilon)$.*
    - *For ε-SV Regression and for ν-SV Regression, $\|u-l\|_1 = C$, which leads to $n_0 = \tilde{O}(C^2\ell/\epsilon)$.*
    - *For 1-class SVM, $\|u-l\|_1 = 1/\nu$, which leads to $n_0 = \tilde{O}(\ell/(\epsilon\nu^2))$.*

References

[1] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a Library for Support Vector Machines*, 2001. Available from http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[2] Chih-Chung Chang and Chih-Jen Lin. Training ν- Support Vector Classifiers: Theory and Algorithms. *Neural Computation*, 10(9):2119–2147, 2001.

[3] Pai-Hsuen Chen, Rong-En Fan, and Chih-Jen Lin. A study on SMO-type decomposition methods for Support Vector Machines. *IEEE Transactions on Neural Networks*, 17(4):893–908, 2006.

[4] Ronan Collobert and Samy Bengio. SVMTorch: Support Vector Machines for large scale regression problems. *Journal of Machine Learning Research*, 6:143–160, 2001.

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

[6] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training Support vector Machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[7] Tobias Glasmachers and Christian Igel. Maximum-gain working set selection or Support Vector Machines. *Journal of Machine Learning Research*, 7:1437–1466, 2006.

[8] Don Hush and Clint Scovel. Polynomial-time decomposition algorithms for Support Vector Machines. *Machine Learning*, 51(1):51–71, 2003.

[9] S. Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649, 2001.

[10] Nikolas List. Convergence of a generalized gradient selection approach for the decomposition method. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, pages 338–349, 2004.

[11] Nikolas List. Generalized SMO-style decomposition algorithms. In *Proceedings of the 20th Annual Conference on Learning Theory*, pages 365–377, 2007.

[12] Nikolas List, Don Hush, Clint Scovel, and Ingo Steinwart. Gaps in Support Vector Optimization. In *Proceedings of the 20th Annual Conference on Learning Theory*, pages 336–348, 2007.

[13] Nikolas List and Hans Ulrich Simon. General polynomial time decomposition algorithms. *Journal of Machine Learning Research*, 8:303–321, 2007.

[14] John C. Platt. Fast training of Support Vector Machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning*, pages 185–208. MIT Press, 1998.

[15] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2002.

[16] Vladimir Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

A Proof of Lemma 2

Assume first that $\vec{X}$ achieves a weak cost-reduction of type $(\alpha, 1)$. It follows that, for every $n \geq 0$,

$$\Delta_n - \Delta_{n+1} \geq \alpha \Delta_{n+1} \ .$$

Solving for Δ_{n+1} yields

$$\Delta_{n+1} \leq \frac{1}{1+\alpha} \Delta_n = \left(1 - \frac{\alpha}{1+\alpha}\right) \Delta_n$$

from which (12) is obvious.

Now assume that $\vec{X}$ achieves a weak cost-reduction of type $(\alpha, 2)$. Consider an arbitrary but fixed n such that $\Delta_n > 0$. For every $s = 0, \ldots, n-1$, define $\rho_s = \frac{\Delta_{s+1}}{\Delta_s}$. Since $\vec{X}$ achieves a weak cost-reduction of type $(\alpha, 2)$, we get

$$\Delta_{s+1} \leq \Delta_s - \alpha \Delta_{s+1}^2 = \Delta_s \underbrace{(1 - \alpha \rho_s^2 \Delta_s)}_{\in (0,1)} \ .$$

It follows that

$$\begin{aligned} \frac{1}{\Delta_{s+1}} - \frac{1}{\Delta_s} &\geq \frac{1}{\Delta_s (1 - \alpha \rho_s^2 \Delta_s)} - \frac{1}{\Delta_s} \\ &= \frac{\alpha \rho_s^2}{1 - \alpha \rho_s^2 \Delta_s} \\ &\geq \alpha \rho_s^2 \ . \end{aligned}$$

We conclude that

$$\begin{aligned} \frac{1}{\Delta_n} &= \frac{1}{\Delta_0} + \sum_{s=0}^{n-1} \frac{1}{\Delta_{s+1}} - \frac{1}{\Delta_s} \\ &> \alpha \sum_{s=0}^{n-1} \rho_s^2 \\ &\geq \alpha n \left(\frac{\Delta_n}{\Delta_0}\right)^{2/n} \ , \end{aligned}$$

where the last inequality easily follows from

$$\rho_0 \cdots \rho_{n-1} = \frac{\Delta_n}{\Delta_0} \ .$$

Solving $\frac{1}{\Delta_n} > \alpha n \left(\frac{\Delta_n}{\Delta_0}\right)^{2/n}$ for Δ_n yields (13).

Our proof of the second part of Lemma 2 builds on a proof by Dunn [5] who solved a similar recursion dealing with a cost-reduction of type $(\alpha, 2)$ (as opposed to a *weak* cost-reduction of this type).