A General Convergence Theorem for the Decomposition Method *

Niko List and Hans Ulrich Simon

Fakultät für Mathematik, Ruhr-Universität Bochum, 44780 Bochum, Germany
Niko.List@gmx.de, simon@lmi.rub.de

Abstract. The decomposition method is currently one of the major methods for solving the convex quadratic optimization problems being associated with support vector machines. Although there exist some versions of the method that are known to converge to an optimal solution, the general convergence properties of the method are not yet fully understood. In this paper, we present a variant of the decomposition method that basically converges for any convex quadratic optimization problem provided that the policy for working set selection satisfies three abstract conditions. We furthermore design a concrete policy that meets these requirements.

1 Introduction

Support vector machines (SVMs) introduced by Vapnik and co-workers [4, 25] are a promising technique for classification, function approximation, and other key problems in statistical learning theory. In this paper, we mainly discuss the optimization problems that are induced by SVMs, which are special cases of convex quadratic optimization.[1]

Example 1. Two popular variants of SVMs lead to the optimization problems given by (1) and (2), respectively:

$$\min_x \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m Q_{i,j}x_ix_j - \sum_{i=1}^m x_i \text{ s.t. } \sum_{i=1}^m y_ix_i = 0\ ,\ \forall i = 1,\ldots,m : 0 \le x_i \le C \tag{1}$$

$$\min_x \frac{1}{2}\sum_{i=1}^m\sum_{j=1}^m Q_{i,j}x_ix_j \text{ s.t. } \sum_{i=1}^m y_ix_i = 0,\ \sum_{i=1}^m x_i \ge \nu\ ,\ \forall i = 1,\ldots,m : 0 \le x_i \le \frac{1}{m} \tag{2}$$

* This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. This work was furthermore supported by the Deutsche Forschungsgemeinschaft Grant SI 498/7-1.

[1] The reader interested in more background information about SVMs is referred to [25, 6, 23].

Here, $Q \in \mathbb{R}^{m \times m}$ is a positive (semi-)definite matrix, $y \in \{-1, 1\}^m$, and x is a vector of m real variables. C and ν are real constants. The first problem is related to one of the classical SVM models; the second-one is related to the so-called ν-SVM introduced by Schölkopf, Smola, Williamson, and Bartlett [24].

The difficulty of solving problems of this kind is the density of Q whose entries are typically non-zero. Thus, a prohibitive amount of memory is required to store the matrix and traditional optimization algorithms (such as Newton, for example) cannot be directly applied. Several authors have proposed (different variants of) a decomposition method to overcome this difficulty [20, 11, 21, 22, 5, 13, 17, 14, 12, 18, 19, 15, 9, 16, 10]. This method keeps track of a current feasible solution which is iteratively improved. In each iteration the variable indices are split into a "working set" $I \subseteq \{1, \ldots, m\}$ and its complement $J = \{1, \ldots, m\} \setminus I$. Then, the subproblem with variables x_i, $i \in I$, is solved, thereby leaving the values for the remaining variables x_j, $j \in J$, unchanged. The success of the method depends in a quite sensitive manner on the policy for the selection of the working set I (whose size is typically bounded by a small constant). Ideally, the selection procedure should be computationally efficient and, at the same time, effective in the sense that the resulting sequence of feasible solutions converges (with high speed) to an optimal limit point. Clearly, these goals are conflicting in general and trade-offs are to be expected. At the time being, it seems fair to say that the issue of convergence is not fully understood (although some of the papers mentioned above certainly shed some light on this question).

We briefly note that also the random sampling technique applied in [2, 1] (and being based on the Simple Sampling Lemma by Gärtner and Welzl [7]) can be viewed as a kind of decomposition method. Here, the working sets (= samples) are probabilistically selected according to a dynamic weighting scheme. The general idea is to update the weights in such a fashion that the support vectors not yet included in the sample become more and more likely to be chosen. At some point the sample will contain enough support vectors such that the solution obtained in the next iteration will be globally optimal. The efficiency of this technique seems to depend strongly on a parameter k that can be rigorously defined in mathematical terms but is unknown in practice. Parameter k is certainly bounded by m but might be much smaller under lucky circumstances. The sample size grows quadratically in k and in the dimension n of the feature space. If k and n are much smaller than m, the random sampling technique seems to produce nice results. We briefly point to the main differences between the random sampling technique and other work on the decomposition method (including ours):

- random selection of the working set
- dependence of the performance on an unknown parameter k
- comparably large working sets (samples)
- very few iterations on the average to optimum if k is small

We close the introduction by explaining the main difference between this paper and earlier work on the decomposition method. It seems that all existing papers concerned with the decomposition method perform a kind of non-uniform analysis in the sense that the results very much depend on the concrete instance of convex quadratic optimization that is induced by the specific SVM under consideration. Given the practical importance of SVM problems, this is certainly justified and may occasionally lead to methods with nice properties (concerning efficiency of working set selection and speed of convergence). On the long run, however, it bears the danger that any new variant of a SVM must be analyzed from scratch because the generality (if any) of the arguments being used so far is too much left in the dark. In this paper, we pursue the goal to establish convergence in a quite general setting. We present a variant of the decomposition method that converges for basically any convex quadratic optimization problem provided that the policy for working set selection satisfies three abstract conditions. We furthermore design a concrete policy that meets these requirements. We admittedly ignore computational issues. The analysis of the trade-off between computational efficiency, speed of convergence, and degree of generality is left as object of future research.

2 Definitions, Notations and Basic Facts

For a matrix $A \in \mathbb{R}^{k \times m}$, $A_i \in \mathbb{R}^k$ denotes the i'th column. $A^\top \in \mathbb{R}^{m \times k}$ denotes the transpose of A. Vectors are considered as column vectors such that the transpose of a vector is a row vector. The "all-zeroes" vector is denoted as $\mathbf{0}$, where its dimension will always become clear from the context. For two vectors $w, x \in \mathbb{R}^m$, $w^\top x = \sum_{i=1}^m w_i x_i$ denotes the standard scalar product. $\|x\| := \left(\sum_{i=1}^m x_i^2\right)^{1/2}$ denotes the Euclidean norm of x. We often consider complementary sets $I \subseteq \{1, \ldots, m\}$, $J = \{1, \ldots, m\} \setminus I$ of indices. The notation A_I refers to the submatrix of A consisting of all column A_i such that $i \in I$. The equation $A = [A_I, A_J]$ means that A decomposes into submatrices A_I, A_J (although, strictly speaking, the equation holds only after the columns of $[A_I, A_J]$ are permuted such that they are ordered as in A). A similar convention is applied to vectors such that equations like $Ax = b$ can be expanded to

$$[A_I, A_J] \begin{pmatrix} x_I \\ x_J \end{pmatrix} = b \ .$$

Similarly, a matrix $Q \in \mathbb{R}^{m \times m}$ decomposes into four blocks $Q_{I,I}, Q_{I,J}, Q_{J,I}, Q_{J,J}$ such that an expression like $x^\top Qx$ can be expanded to

$$x^\top Qx = x_I^\top Q_{I,I} x_I + x_I^\top Q_{I,J} x_J + x_J^\top Q_{J,I} x_I + x_J^\top Q_{J,J} x_J \ .$$

If Q is symmetric (in particular, if Q is positive (semi-)definite), then $x_I^\top Q_{I,J} x_J = x_J^\top Q_{J,I} x_I$.

Let $\mathcal{P}$ denote an optimization problem that is given by a cost function $f(x)$ and a collection of constraints, where x denotes a collection of real-valued variables. As usual, a feasible solution for $\mathcal{P}$ is an assigment of values to the variables that satisfies all constraints. The feasibility region (consisting of all feasible solutions for $\mathcal{P}$) is denoted as $R(\mathcal{P})$. The smallest possible cost of a feasible solution is then given by

$$\mathrm{opt}(\mathcal{P}) = \min_{x \in R(\mathcal{P})} f(x) \enspace .$$

Writing "min" instead of "inf" is justified because we will deal only with problems $\mathcal{P}$ whose feasibility region is compact. In the remainder of the paper, we assume some familiarity with mathematical programming and matrix theory.

2.1 Convex Quadratic Programming subject to Box Constraints

Throughout this paper,

$$f(x) = \frac{1}{2} x^\top Q x + w^\top x = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} Q_{i,j} x_i x_j + \sum_{i=1}^{m} w_i x_i \tag{3}$$

denotes a convex cost function, where $Q \in \mathbb{R}^{m \times m}$ is a positive semi-definite matrix over the reals with the additional (somewhat technical) property that, for each $I \subseteq \{1, \ldots, m\}$ of size at most q, the submatrix $Q_{I,I}$ of Q is positive definite. Here, q denotes a (typically small) constant (which will later bound from above the size of the working set). Note that the technical condition for Q is satisfied if Q itself is positive definite. As the structure of the cost function has become clear by now, we move on and define our basic optimization problem $\mathcal{P}$:

$$\min_x f(x) \text{ s.t. } Ax = b, l \leq x \leq r \tag{4}$$

Here, $A \in \mathbb{R}^{k \times m}$, $b \in \mathbb{R}^k$, $l, r \in \mathbb{R}^m$, and $l \leq x \leq r$ is the short-notation for the "box constraints"

$$\forall i = 1, \ldots, m : l_i \leq x_i \leq r_i \enspace .$$

A few comments are in place:

- Any bounded[2] optimization problem with cost function $f(x)$ and linear equality- and inequality-constraints can be brought into the form (4) because we may convert the linear inequalities into linear equations by introducing non-negative slack variables. By the compactness of the feasibility region, we may also put a suitable upper bound on each slack variable such that the remaining linear inequalities take the form of box constraints.
- The technical assumption that we have put on matrix Q is slightly more restrictive than just assuming it is positive semi-definite. As far as the decomposition method and SVM applications are concerned, this assumption if often satisfied.[3]

[2] Here, "bounded" means that the feasibility region is compact (or can be made compact without changing the smallest possible cost).

[3] For some kernels like, for example, the RBF-kernel, it is certainly true; for other kernels it typically satisfied provided that q is sufficiently small. See also the discussion of this point in [17].

In order to illustrate the first comment, we convert problem (2) in a problem with box constraints by introducing the slack variable ξ:

$$\min_{x,\xi} \frac{1}{2} x^\top Q x \text{ s.t. } y^\top x = 0, \ \sum_{i=1}^m x_i - \xi = \nu, 0 \le \xi \le 1 \ , \ \forall i = 1, \dots, m : 0 \le x_i \le \frac{1}{m} \tag{5}$$

The optimal solutions for $\mathcal{P}$ can be characterized in terms of the gradient $\nabla f(x) = Qx + w$ as follows:

Lemma 1. *Let $\mathcal{P}$ denote the optimization problem that is induced by $A \in \mathbb{R}^{k \times m}$, $b \in \mathbb{R}^k$, and $l, r \in \mathbb{R}^m$ as described in (4) and let U denote the linear subspace of $\mathbb{R}^m$ that is spanned by the rows of matrix A. Then, x is optimal for $\mathcal{P}$ iff there exists $u \in U$ such that*

$$x_i \neq r_i \Rightarrow \nabla f(x)_i - u_i \ge 0 \text{ and } x_i \neq l_i \Rightarrow u_i - \nabla f(x)_i \ge 0 \ . \tag{6}$$

holds for $i = 1, \dots, m$.

Proof. It is well-known that x is optimal for $\mathcal{P}$ iff it satisfies the Karush-Kuhn-Tucker conditions. The latters are easily seen to be equivalent to the existence of $\beta \in \mathbb{R}^k$ such that the following holds for $i = 1, \dots, m$:

$$x_i \neq r_i \Rightarrow \nabla f(x)_i - A_i^\top \beta \ge 0 \text{ and } x_i \neq l_i \Rightarrow A_i^\top \beta - \nabla f(x)_i \ge 0 \ .$$

(Recall the convention that $A_i \in \mathbb{R}^k$ denotes the i'th column of A.) The lemma now follows from the observation that $A^\top \beta$ ranges over U when β ranges over $\mathbb{R}^k$. □

With each $x \in \mathbb{R}^m$, we associate the function

$$C(x) := \inf_{u \in U} \sum_{i=1}^m (x_i - l_i) \max\{0, \nabla f(x)_i - u_i\} + (r_i - x_i) \max\{0, u_i - \nabla f(x)_i\} \ , \tag{7}$$

whose properties are summarized in

Lemma 2. *$C(x)$ is a continuous function on $R(\mathcal{P})$. Moreover, for $x \in R(\mathcal{P})$, $C(x) \ge 0$ with equality iff x is optimal for $\mathcal{P}$.*

Proof. We first show that $C(x)$ is continuous. Obviously function

$$C(x, u) := \sum_{i=1}^m (x_i - l_i) \max\{0, \nabla f(x)_i - u_i\} + (r_i - x_i) \max\{0, u_i - \nabla f(x)_i\}$$

is continuous in x and u. Moreover, $C(x) = \inf_{u \in U} C(x, u)$. With each constant $B > 0$, we associate the compact region $U(B) := U \cap \{u \in \mathbb{R}^m | \ \|u\| \le B\}$. It is not hard to see that there exists a constant $B > 0$ such that $C(x) =$

$\inf_{u \in U(B)} C(x,u)$ holds for each $x \in R(\mathcal{P})$. By compactness, $C(x,u)$ is uniformly continuous on $R(\mathcal{P}) \times U(B)$. Thus, for all $x, x' \in R(\mathcal{P})$ and each $\epsilon > 0$, there exists $\delta > 0$ such that

$$\forall u \in U(B) : \|x' - x\| \leq \delta \Rightarrow |C(x', u) - C(x, u)| \leq \epsilon \ .$$

Since the latter statement implies that $|C(x') - C(x)| \leq \epsilon$, we may conclude that $C(x)$ is continuous.
If x is a feasible solution for $\mathcal{P}$, then $l \leq x \leq r$, which clearly implies that $C(x) \geq 0$. Furthermore, $C(x) = 0$ iff there exists $u \in U$ such that (6) is satisfied. According to Lemma 1, this is true iff x optimally solves $\mathcal{P}$. □

The method of feasible directions by Zoutendijk [26] allows for another characterization of the optimal solutions for $\mathcal{P}$. To this end, we associate the following optimization problem $\mathcal{D}[x]$ with each $x \in R(\mathcal{P})$:

$$\min_d \nabla f(x)^\top d \text{ s.t. } Ad = \mathbf{0}, \forall i = 1, \ldots, m : -1 \leq d_i \leq 1 \wedge \begin{cases} x_i = l_i \Rightarrow d_i \geq 0 \\ x_i = r_i \Rightarrow d_i \leq 0 \end{cases} \tag{8}$$

Intuitively, $\nabla f(x)^\top d < 0$ indicates that we can reduce the cost of the current solution x for $\mathcal{P}$ by moving it in direction d. More formally, the following holds:

Lemma 3 ([26, 3]). $\mathrm{opt}(\mathcal{D}[x]) \leq 0$ *with equality iff x is optimal for $\mathcal{P}$.*

2.2 Subproblems Induced by Working Sets

With a set $I \subseteq \{1, \ldots, m\}$, we will always associate its complement $J = \overline{I}$. Furthermore, we use the short-notation $R_I = \{x_I | \ x \in R\}$ for each $R \subseteq \mathbb{R}^m$.

For each $I \subseteq \{1, \ldots, m\}$ and each $x_J \in R(\mathcal{P})_J$, we denote by $\mathcal{P}_{I,x_J}$ the problem that results from $\mathcal{P}$ by leaving x_J unchanged and choosing x_I such as to minimize $f(x)$ subject to the constraints in $\mathcal{P}$. More formally, for cost function

$$f_{I,x_J}(x_I) = \frac{1}{2} x_I^\top Q_{I,I} x_I + (Q_{I,J} x_J + w_I)^\top x_I$$

(with gradient $\nabla f_{I,x_J}(x_I) = Q_{I,I} x_I + Q_{I,J} x_J + w_I$), problem $\mathcal{P}_{I,x_J}$ reads as follows:

$$\min_{x_I} f_{I,x_J}(x_I) \text{ s.t. } A_I x_I = b - A_J x_J, \ l_I \leq x_I \leq r_I$$

Note that this problem is of type (4) with $x_I, Q_{I,I}, Q_{I,J} x_J + w_I, A_I, b - A_J x_J, l_I, r_I$ substituted for x, Q, w, A, b, l, r, respectively. Note furthermore that x_I is a feasible solution for $\mathcal{P}_{I,x_J}$ iff $x \in \mathcal{P}$, i.e., iff x_I extends x_J to a feasible solution $x \in \mathcal{P}$. Recall that, according to our notational conventions, its feasibility region is written as $R(\mathcal{P}_{I,x_J})$.

2.3 The Decomposition Method

Let q be a constant that bounds from above the size of the working set. Let $(C_I(x))_{I \subseteq \{1,\dots,m\}}$ be a family of functions from $R(\mathcal{P})$ to $\mathbb{R}_0^+$. With each such family, we associate the following method for solving $\mathcal{P}$:

(1) Let x^0 be a feasible solution for $\mathcal{P}$ (arbitrarily chosen) and $s := 0$.
(2) Construct a working set $I^s \subseteq \{1,\dots,m\}$ that maximizes $c(I) := C_I(x^s)$ subject to $|I| \leq q$. If $c(I^s) = 0$, then return x^s and stop; otherwise set $J^s := \{1,\dots,m\} \setminus I^s$, find an optimal solution $x_{I^s}^{s+1}$ for $\mathcal{P}_{I^s, x_{J^s}^s}$, set

$$x_{J^s}^{s+1} := x_{J^s}^s,\ x^{s+1} := \begin{pmatrix} x_I^{s+1} \\ x_J^{s+1} \end{pmatrix},\ s := s+1$$

and goto (2).

We refer to this algorithm as the *decomposition method induced by* $(C_I(x))$. We will show in section 3 that it converges to an optimal solution for $\mathcal{P}$ if the following conditions hold:

(C1) For each $I \subseteq \{1,\dots,m\}$ such that $|I| \leq q$, $C_I(x)$ is continuous on $R(\mathcal{P})$.

(C2) If $|I| \leq q$ and x_I is an optimal solution for $\mathcal{P}_{I,x_J}$, then $C_I\begin{pmatrix} x_I \\ x_J \end{pmatrix} = 0$.

(C3) If x is not an optimal solution for $\mathcal{P}$, then there exists an $I \subseteq \{1,\dots,m\}$ such that $|I| \leq q$ and $C_I(x) > 0$.

If these conditions are satisfied, we call the family $(C_I(x))$ a *q-sparse witness of suboptimality.* In section 4, we will present such a family of functions provided that $q \geq k+1$.

A few comments are in place here. x^{s+1} is always a feasible solution for $\mathcal{P}$. Moreover, $x_{I^s}^{s+1}$ is (by construction) an optimal solution for $\mathcal{P}_{I^s,x_J^s}$. Thus,

$$C_{I^s}(x^{s+1}) = 0 \tag{9}$$

according to (C2). If x^s is (accidentally) an optimal solution for $\mathcal{P}$, then it is (à-fortiori) an optimal solution for each subproblem $\mathcal{P}_{I,x_J}$ and, again according to (C2), the decomposition method will reach the stop-condition and return x^s. If x^s is not optimal for $\mathcal{P}$, then (C3) makes sure that there exists a working set I of size at most q such that $C_I(x^s) > 0$. Thus, the working set I^s actually constructed by the decomposition method satisfies

$$C_{I^s}(x^s) > 0\ , \tag{10}$$

and the method cannot become stuck at a suboptimal solution.

We assume in the sequel that the sequence x^s evolves as described above. Note that $f(x^s)$ is decreasing with s (simply because $x_{I^s}^s$ is a feasible solution for $\mathcal{P}_{I^s,x_{J^s}^s}$). Thus, $f(x^s)$ will converge to a limit even if x^s does not converge to a limit point. However, since the feasibility region for $\mathcal{P}$ is compact, there must exist a subsequence $(x^s)_{s \in S}$ that converges to a (feasible!) limit point, say x^∞. Clearly, $f(x^\infty) = \lim_{s\to\infty} f(x^s)$. It remains to show that x^∞ is an optimal solution.

3 Analysis of the Decomposition Method

This section is devoted to the proof of convergence. The proof will proceed by assuming, for sake of contradiction, that x^∞ is not an optimal solution for $\mathcal{P}$. From condition (C3) and from a continuity argument, we will be able to conclude that x^∞ is not even optimal for subproblem $\mathcal{P}_{I^s,x^s_{J^s}}$ if $s \in S$ is chosen sufficiently large. Since x^{s+1} is an optimal solution for this subproblem (by the definition of the decomposition method), we would now be close to a contradiction if the continuity argument also applied to $s+1$. Here, however, we bomb into a difficulty since $s+1$ does not necessarly belong to S. Thus, although sequence $(x^s)_{s\in S}$ approaches x^∞ when n approaches infinity, sequence $(x^{s+1})_{s\in S}$ might perhaps behave differently? It turns out, however, that this is not the case. The main argument against this hypothetical possibility will be that the cost reduction per iteration of the decomposition method is proportional to the square of the distance between x^s and x^{s+1}. The following subsections flesh out this general idea.

3.1 Cost Reduction per Iteration

How big is the cost reduction when we pass from x^s to x^{s+1}? Here is an answer to this question:

Lemma 4. *Let $f(x)$ be the cost function of $\mathcal{P}$ as given by (3). Let*

$$\sigma := \min_{I\subseteq\{1,\dots,m\}:|I|\le q} \mathrm{eig}(Q_{I,I}) > 0 \enspace ,$$

where $\mathrm{eig}(\cdot)$ *denotes the smallest eigenvalue of a matrix.*[4] *With these notations, the following holds:*

$$f(x^{s+1}) - f(x^s) \le -\frac{\sigma}{2}\|x^{s+1} - x^s\|^2 \enspace .$$

Proof. Since f is a quadratic function of the form (3), Taylor-expansion around x^{s+1} yields

$$f(x^s) = f(x^{s+1}) + \nabla f(x^{s+1})^\top(x^s - x^{s+1}) + \frac{1}{2}(x^s - x^{s+1})^\top Q(x^s - x^{s+1}) \enspace . \tag{11}$$

Recall that x^{s+1} minimizes $f(x)$ subject to the constraints $Ax = b$, $l \le x \le r$, and $x^{s+1}_{J^s} = x^s_{J^s}$. Since these constraints define a convex region containing x^s and x^{s+1}, we may conclude that $f(x^{s+1}) = \min_{x\in L} f(x)$ where L denotes the line segment between x^s and x^{s+1}. Thus, the gradient at x^{s+1} in direction to x^s is ascending, i.e.,

$$\nabla f(x^{s+1})^\top(x^s - x^{s+1}) \ge 0 \enspace . \tag{12}$$

[4] Note that the technical property that we have put on Q in section 2.1 makes sure that σ is strictly positive.

Note furthermore that

$$(x^s - x^{s+1})^\top Q(x^s - x^{s+1}) \geq \sigma \|x^s - x^{s+1}\|^2 \tag{13}$$

is an immediate consequence of the Courant-Fischer Minimax Theorem [8]. From (11), (12), and (13), the lemma follows. $\square$

We briefly note that Lin [17] has shown a similar lemma for the special optimization problem (1). Although our lemma is more general, the proof found in [17] is much more complicated.

3.2 Facts Being Valid Asymptotically

Lemma 5. *For each $\delta > 0$, there exists $s_0 \geq 1$ such that*

$$\|x^s - x^\infty\| \leq \frac{\delta}{2} \ , \ \|x^{s+1} - x^s\| \leq \frac{\delta}{2} \ \textit{and} \ \|x^{s+1} - x^\infty\| \leq \delta \ .$$

holds for all $s \in S$ provided that $s \geq s_0$.[5]

Proof. Recall that $(f(x^s))_{s \geq 1}$ is a monotonously decreasing sequence that approaches $f(\alpha^\infty)$ when s tends to infinity. Thus, there exists s_0' such that

$$0 < f(x^s) - f(x^{s+1}) \leq f(x^s) - f(x^\infty) \leq \frac{\sigma\delta^2}{8}$$

holds for all $s \geq s_0'$. According to Lemma 4,

$$\|x^{s+1} - x^s\| \leq \sqrt{\frac{2(f(x^s) - f(x^{s+1}))}{\sigma}} \ .$$

Thus

$$\|x^{s+1} - x^s\| \leq \frac{\delta}{2}$$

holds for all $s \geq s_0'$. Since $(x^s)_{s \in S}$ converges to x^∞, there exists s_0'' such that

$$\|x^s - x^\infty\| \leq \frac{\delta}{2}$$

holds for all $s \in S$, $s \geq s_0''$. Setting $s_0 = \max\{s_0', s_0''\}$, we obtain

$$\|x^{s+1} - x^s\| \leq \frac{\delta}{2} \text{ and } \|x^s - x^\infty\| \leq \frac{\delta}{2}$$

for all $s \in S$, $s \geq s_0$. This implies that $\|x^{s+1} - x^\infty\| \leq \delta$ and completes the proof of the lemma. $\square$

[5] Inequality $\|x^{s+1} - x^\infty\| \leq \delta$, which is immediate from the preceding two inequalities, has been included for ease of later reference.

Corollary 1. *If $(C_I(x))$ satisfies condition (C1), then, for each $\epsilon > 0$, there exists $s_0 \geq 1$ such that*

$$|C_I(x^\infty) - C_I(x^s)| < \epsilon \tag{14}$$
$$|C_I(x^\infty) - C_I(x^{s+1})| < \epsilon \tag{15}$$

holds for each working set $I \subseteq \{1, \ldots, m\}$ of size at most q and for all $s \in S$ provided that $s \geq s_0$.

Proof. The corollary easily follows from Lemma 5, the fact that there are only finitely many sets $I \subseteq \{1, \ldots, m\}$ of size at most q, and condition (C1) stating that each individual function $C_I(x)$ is continuous. □

3.3 The Main Theorem

Theorem 1. *Assume that $(C_I(x))$ satisfies conditions (C1),(C2),(C3), i.e., it is a q-sparse witness of suboptimality. Let x^s be a sequence of legal solutions for $\mathcal{P}$ that is produced by the decomposition method induced by $(C_I(x))$ and let $(x^s)_{s\in S}$ be a converging subsequence. Then, the limit point x^∞ of $(x^s)_{s\in S}$ is an optimal solution for $\mathcal{P}$.*

Proof. Assume for sake of contradiction that x^∞ is not an optimal solution for $\mathcal{P}$. According to (C3), there exists a working set $I \subseteq \{1, \ldots, m\}$ such that $|I| \leq q$ and

$$\epsilon_0 := C_I(x^\infty) > 0 \ .$$

In the sequel, we will apply Corollary 1 three times with $\epsilon = \epsilon_0/3$, respectively. Assume that $s \in S$ is sufficiently large in the sense of Corollary 1 such that, according to (14), the following holds:

$$C_I(x^s) > \frac{2\epsilon_0}{3} \ .$$

Thus, the working set I^s returned by the decomposition method in iteration $s+1$ satisfies

$$C_{I^s}(x^s) > \frac{2\epsilon_0}{3} \ .$$

Another application of (14) leads to

$$C_{I^s}(x^\infty) > \frac{\epsilon_0}{3} \ .$$

From (15), we get

$$C_{I^s}(x^{s+1}) > 0 \ .$$

Since x^{s+1} is an optimal solution for $\mathcal{P}_{I^s, x^s_J}$, we may however infer from (9) that

$$C_{I^s}(x^{s+1}) = 0 \ .$$

We arrived at a a contradiction. □

4 A Sparse Witness of Sub-optimality

In this section, we present a concrete family $(C_I(x))$ of functions that satisfies the conditions (C1),(C2),(C3) needed for our proof of convergence from section 3. We will define $C_I(x)$ such that it plays the same role for $\mathcal{P}_{I,x_J}$ that the function $C(x)$ (defined in (7)) has played for $\mathcal{P}$. More formally, let U_I denote the subspace spanned by the rows of A_I, and define $C_I(x)$ to be equal to

$$\inf_{u \in U_I} \sum_{i \in I} (x_i - l_i) \max\{0, \nabla f_{I,x_J}(x_I)_i - u_i\} + (r_i - x_i) \max\{0, u_i - \nabla f_{I,x_J}(x_I)_i\} \ .$$

In what follows, we use the notations $C_I(x)$ and $C_{I,x_J}(x_I)$ interchangeably. The former notation stresses that $C_I(x)$ is viewed as a function of all components of x, whereas the latter notation stresses the relation between this function and the subproblem $\mathcal{P}_{I,x_J}$ that is explained in Corollary 2 below.

Recall the optimization problem $\mathcal{D}[x]$ from (8). Let $\mathcal{D}_{I,x_J}[x_I]$ be the optimization problem given by

$$\min_{d_I} \nabla f_{I,x_J}(x_I)^\top d_I \text{ s.t. } A_I d_I = \mathbf{0}, \forall i \in I : -1 \leq d_i \leq 1 \wedge \begin{cases} x_i = l_i \Rightarrow d_i \geq 0 \\ x_i = r_i \Rightarrow d_i \leq 0 \end{cases} .$$

Now, Lemmas 2 and 3 applied to the subproblem $\mathcal{P}_{I,x_J}$ induced by I and x_J read as follows:

Corollary 2. *1. $C_{I,x_J}(x_I)$ is a continuous function on $R(\mathcal{P}_{I,x_J})$. Moreover, $C_{I,x_J}(x_I) \geq 0$ with equality iff x_I is optimal for $\mathcal{P}_{I,x_J}$.*
2. $\mathrm{opt}(\mathcal{D}_{I,x_J}[x_I]) \leq 0$ with equality iff x_I is optimal for $\mathcal{P}_{I,x_J}$.

The first statement in Corollary 2 can clearly be strengthened:

Remark 1. $C_I(x) = C_{I,x_J}(x_I)$ viewed as a function in $x = \begin{pmatrix} x_I \\ x_J \end{pmatrix}$ is continuous on $R(\mathcal{P})$.

This already settles conditions (C1) and (C2). Condition (C3) is settled by

Lemma 6. *If x is a feasible solution for $\mathcal{P}$ that is not optimal, then there exists a working set $I \subseteq \{1, \ldots, m\}$ such that $|I| \leq k+1$ and $\mathrm{opt}(\mathcal{D}_{I,x_J}[x_I]) < 0$.*

Proof. In order to facilitate the proof, we first introduce two slight modifications of problem $\mathcal{D}[x]$. Let $\mathcal{D}'[x]$ be the problem obtained from $\mathcal{D}[x]$ by substituting the single constraint

$$\sum_{i=1}^{m} |d_i| = 1$$

for the m constraints

$$\forall i = 1, \ldots, m : -1 \leq d_i \leq 1 \ .$$

Problems $\mathcal{D}[x]$ and $\mathcal{D}'[x]$ exhibit the following relationship:

- If d is a feasible solution for $\mathcal{D}[x]$ of cost c, then $d/\sum_{i=1}^{m}|d_i|$ is a feasible solution for $\mathcal{D}'[x]$ of cost $c/\sum_{i=1}^{m}|d_i|$.

- Clearly, each feasible solution for $\mathcal{D}'[x]$ is also a feasible solution for $\mathcal{D}[x]$.

Thus, there is a feasible solution of negative cost for $\mathcal{D}[x]$ iff there is one for $\mathcal{D}'[x]$. We may therefore conclude that $\mathrm{opt}(\mathcal{D}'[x]) < 0$ iff $\mathrm{opt}(\mathcal{D}[x]) < 0$.
It will still be more convenient to consider another modification of $\mathcal{D}[x]$ that we denote as $\mathcal{D}''[x]$ in the sequel:

$$\min_{d^+,d^-} \begin{pmatrix} \nabla f(x) \\ -\nabla f(x) \end{pmatrix}^\top \begin{pmatrix} d^+ \\ d^- \end{pmatrix}$$

subject to

$$\forall i = 1,\dots,m : x_i = l_i \Rightarrow d_i^- = 0 \ \ , \ \ x_i = r_i \Rightarrow d_i^+ = 0 \tag{16}$$

$$[A, -A]\begin{pmatrix} d^+ \\ d^- \end{pmatrix} = \mathbf{0} \tag{17}$$

$$\sum_{i=1}^{m} d_i^+ + d_i^- = 1 \tag{18}$$

$$d^+, d^- \geq \mathbf{0} \tag{19}$$

$\mathcal{D}'[x]$ and $\mathcal{D}''[x]$ are easily seen to be equivalent by making use of the relation $d = d^+ - d^-$. Thus, $\mathrm{opt}(\mathcal{D}''[x]) < 0$ iff $\mathrm{opt}(\mathcal{D}'[x]) < 0$. What is the advantage of dealing with $\mathcal{D}''[x]$ instead of $\mathcal{D}[x]$? The answer is that $\mathcal{D}''[x]$ is a linear program in canonical form with very few equations. To see this note first that we need not count the equations in (16) since the variables that are set to zero there can simply be eliminated (thereby passing to a lower-dimensional problem). Thus, there are only $k+1$ equations left in (17) and (18). It follows that each basic feasible solution for $\mathcal{D}''[x]$ has has at most $k+1$ non-zero components.
We are now prepared to prove the lemma. Assume that x is a feasible but suboptimal solution of $\mathcal{P}$. We may conclude from Lemma 3 that $\mathrm{opt}(\mathcal{D}[x]) < 0$ and, therefore, $\mathrm{opt}(\mathcal{D}''[x]) < 0$. If d^+, d^- represent the optimal basic feasible solution for $\mathcal{D}''[x]$ (with at most $k+1$ non-zero components and negative cost), we obtain the feasible solution $d = d^+ - d^-$ for $\mathcal{D}'[x]$ that also has at most $k+1$ non-zero components and (the same) negative cost. Consider working set $I = \{i \in \{1,\dots,m\} |\ d_i \neq 0\}$. Clearly, d is is also a feasible solution for $\mathcal{D}_{I,x_J}[x_I]$ such that $\nabla f(x)_I^\top d_I = \nabla f(x)^\top d < 0$. Thus, $\mathrm{opt}(\mathcal{D}_{I,x_J}[x_I]) < 0$, which completes the proof. □

Combining Lemma 6 with Corollary 2, we get

Corollary 3. *If x is a feasible but non-optimal solution for $\mathcal{P}$, then there exists a working set I of size at most $k+1$ such that x is not optimal for $\mathcal{P}_{I,x_J}$ and $C_I(x) = C_{I,x_J}(x_I) > 0$.*

5 Final Remarks and Open Problems

Chang, Hsu, and Lin prove the convergence for a decomposition method that is tailored to the optimization problem (1) except that the cost function may be an arbitrary continuously differentiable function [5]. They apply techniques of "projected gradients". Although their analysis is tailored to problem (1), we would like to raise the question whether the techniques of projected gradients can be used to extend our results to a wider class of cost functions.

The function $C(x)$ defined in (7) is easily seen to bound $f(x) - f(x^\infty)$ from above. In this sense it measures (an upper bound on) the current distance from optimum. Schölkopf and Smola have proposed to select the working set I whose indices point to the (at most q) largest terms in $C(x)$ [23]. This policy for working set selection looks similar to ours (but the policies are, in general, not identical). The question whether the (somewhat simpler) policy proposed by Schölkopf and Smola makes sequence x^s converging to an optimal limit point remains open (although we cannot rule out that both policies actually coincide for the specific problems resulting from SVM applications).

The most challenging task for future research is gaining a deeper understanding of the trade-off between the following three goals:

- efficiency of working set selection
- fast convergence to optimum
- generality of the arguments

It would be nice to lift the decomposition method from SVM applications to a wider class of optimization problems without much loss of efficiency or speed of convergence.

Acknowledgments Thanks to Dietrich Braess for pointing us to a simplification in the proof of Lemma 4. Thanks to the anonymous referees for their comments and suggestions and for drawing our attention to the random sampling technique.

References

1. José L. Balcázar, Yang Dai, and Osamu Watanabe. Provably fast trainig algorithms for support vector machines. In *Proceedings of the 1st International Conference on Data Mining*, pages 43–50, 2001.
2. José L. Balcázar, Yang Dai, and Osamu Watanabe. A random sampling technique for training support vector machines. In *Proceedings of the 12th International Conference on Algorithmic Learning Theory*, pages 119–134. Springer Verlag, 2001.
3. Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms.* John Wiley & Sons, 1993.
4. Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.
5. Chih-Chung Chang, Chih-Wei Hsu, and Chih-Jen Lin. The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 11(4):248–250, 2000.

6. Nello Christianini and John Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.
7. Bernd Gärtner and Emo Welzl. A simple sampling lemma: Analysis and applications in geometric optimization. *Discrete & Computational Geometry*, 25(4):569–590, 2001.
8. Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The John Hopkins University Press, third edition, 1996.
9. Chih-Wei Hsu and Chih-Jen Lin. A simple decomposition method for support vector machines. *Machine Learning*, 46(1–3):291–314, 2002.
10. Don Hush and Clint Scovel. Polynomial-time decomposition algorithms for support vector machines. *Machine Learning*, 51:51–71, 2003.
11. Thorsten Joachims. Making large scale SVM learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods—Support Vector Learning.* MIT Press, 1998.
12. S. S. Keerthi and E. G. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46:351–360, 2002.
13. S. S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, 2000.
14. S. S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13:637–649, 2001.
15. P. Laskov. An improved decomposition algorithm for regression support vector machines. *Machine Learning*, 46:315–350, 2002.
16. S.-P. Liao, H.-T. Lin, and Chih-Jen Lin. A note on the decomposition methods for support vector regression. *Neural Computation*, 14:1267–1281, 2002.
17. Chih-Jen Lin. On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*, 12:1288–1298, 2001.
18. Chih-Jen Lin. Asymptotic convergence of an SMO algorithm without any assumptions. *IEEE Transactions on Neural Networks*, 13:248–250, 2002.
19. Chih-Jen Lin. A formal analysis of stopping criteria of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 13:1045–1052, 2002.
20. E. Osuna, R. Freund, and F.Girosi. Training support vector machines: an application to face detection. In *Proceedings of CVPS'97*, 1997.
21. J. 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.* MIT Press, 1998.
22. C. Saunders, M. O. Stitson, J. Weston, L. Bottou, Bernhard Schölkopf, and Alexander J. Smola. Support vector machine reference manual. Technical Report CSD-TR-98-03, Royal Holloway, University of London, Egham, UK, 1998.
23. Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels.* MIT Press, 2002.
24. Bernhard Schölkopf, Alexander J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.
25. Vladimir Vapnik. *Statistical Learning Theory.* Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, 1998.
26. G. Zoutendijk. *Methods of Feasible Directions.* Elsevier Publishing Company, 1960.