Convergence Rates for SVM-Decomposition Algorithms

Dissertation

zur Erlangung des Grades eines Doktor der Naturwissenschaften
der Fakultät für Mathematik
der Ruhr-Universität Bochum

vorgelegt von
Nikolas List

unter der Betreuung von
Hans Ulrich Simon

Bochum
Juli 2008
The following thesis is concerned with the analysis of a special class of optimization algorithms for support vector machines. The support vector machine (SVM) introduced by Boser et al. [1992] is a state of the art learning algorithm, which is successfully used in many supervised classification and regression tasks. It induces a high dimensional convex optimization problem which is typically solved by iteratively solving smaller subproblems. Such algorithms are known as decomposition algorithms for SVM optimization [Platt, 1999, Joachims, 1999, Hush and Scovel, 2003, Fan et al., 2005]. It will be shown that most of them can be seen as exact line search algorithms, where the choice of a suitable descent direction corresponds to the selection of a minimal working set.


This thesis will be mainly concerned with concrete convergence bounds for such algorithms. For conservative selection we will give run time guarantees of $O(\max\{\ell^2/\varepsilon, \ell/\varepsilon^2\})$ for typical SVM optimization problems, where $\ell$ is the number of training samples and $\varepsilon$ is the accuracy we request for the associated primal classifier. This bound is comparable to well known con-

The second result will close a gap in the analysis of decomposition algorithms for SVMs: Although it is well known, that maximum violating pair selection algorithms asymptotically converge to an optimal solution of the dual problem induced by many SVM optimization problems [Keerthi and Gilbert, 2002, Lin, 2001b, List, 2004], no convergence rates could be established. In this thesis we will show that one can guarantee a run time of $O(\ell^4/\varepsilon^2)$ for typical SVM optimization problems in the above mentioned software packages. Even more, the proof of this result gives evidence that this bound is not tight on typical instances of SVM optimization and the run time may scale in $O(\ell M^3/\varepsilon^2)$, where the term $M$ scales in the order between $\#sv$ and $\#nsv$. $\#sv$ and $\#nsv$ here denote the number of bounded and non bounded support vectors of the optimal classifier.

The third topic of this thesis is a rigorous analysis of standard stopping criteria with respect to the primal objective, the empirical risk minimization task. As first pointed out by Hush et al. [2005] the convergence guarantees for SVM algorithms have often only focused on the quality of the dual solution neglecting the primal optimality with respect to the empirical risk function. We will use the results from List et al. [2007] to ensure that all stopping criteria implemented in the presented algorithms lead to approximately optimal primal solutions. In particular, the given analysis allows for tuning the stopping criteria of standard dual algorithms directly to the need of primal accuracy.
Danksagung


Ebenso danke ich allen, die mich erträgen und ermuntert haben, als der Zweifel nagte – ganz besonders Anita.
# Contents

**Introduction**

1 Prerequisites from Convex Optimization

1.1 Notational Conventions

1.2 Abstract Duality

1.2.1 Introduction

1.2.2 The Main Idea of Duality

1.2.3 The Dual Problem and Its Properties

1.2.4 When Does the Dual Approach Succeed?

1.2.5 Quadratic Constrained Optimization

1.3 Optimization Theory

1.3.1 Notational Conventions

1.3.2 Convergence Schemes

1.3.3 Descent Line Search and Gradient Methods

2 Prerequisites from Learning Theory

2.1 Introduction

2.1.1 Basics of Statistical Learning Theory

2.1.2 Large Margin Classifiers

2.1.3 The Kernel Trick
## Contents

2.2 Support Vector Machines – Problem Formulations  
2.2.1 Abstract Regularized Risk Minimization  
2.2.2 C-Support Vector Classification  
2.2.3 ν-Support Vector Classification  
2.2.4 ε-Support Vector Regression  
2.2.5 ν-Support Vector Regression  
2.2.6 Other Variants of SVM  

3 Decomposition Techniques for Quadratic Programming  
3.1 Introduction  
3.1.1 Decomposition Algorithms  
3.2 Aggressive Decomposition Algorithms  
3.2.1 General Selection Strategies  
3.2.2 Pairable Problems  
3.2.3 Convergence Rates  
3.2.4 Bounded Locking Sequences for Pairable Problems  
3.3 Conservative Decomposition Algorithms  
3.3.1 Convergence Rates  
3.3.2 General Conservative Selection Strategies  
3.3.3 Conservative Selection for Pairable Problems  

4 Duality Gaps and Stopping Criteria  
4.1 Introduction  
4.2 Abstract Duality Gaps  
4.2.1 Forward Gap and Dual Optimality  
4.2.2 Relation between the Gaps  
4.3 Stopping Heuristics for SVM Optimization  
4.3.1 Optimality Bounds for SVM Optimization  
4.3.2 Using σ-gaps as Stopping Criteria  

5 SVMs and Optimization  
5.1 Decomposition Algorithms for SVM Optimization  
5.2 Convergence Bounds for SVM Optimization  
5.2.1 Conservative Decomposition Algorithms  
5.2.2 Aggressive Decomposition Algorithms  
5.3 Interpretation of the Bounds  
5.3.1 Conservative Decomposition  
5.3.2 Aggressive Decomposition
Introduction

In a world where huge data sets are available cheaper to more people than ever before we are confronted with the need of classifying and structuring these data piles in a sensible manner. The sheer amount of data in modern science and economy usually makes this an insurmountable task for human beings. However, in the last decades, as the amount of data available was constantly growing, there emerged many ideas how computers could act as servants in such tasks.

Today’s life is unimaginable without computer aided sorting, classifying and structuring of given data. These applications are often based on the ability of computer algorithms to act as learning machines drawing generalizations from given data on their own. The design and implementation of such learning machines is an intellectually challenging task and has raised many interesting mathematical problems.

In this thesis we will be concerned with one specific class of algorithms – called decomposition algorithms – for one special learning machine – known as support vector machine. Nevertheless, we will see, even in such a restricted area, there exist numerous questions between the poles of heuristically motivated algorithms, performing quite well in practice, and mathematically provable guarantees for accuracy and run time.

Support Vector Machines and Optimization

The support vector machine (SVM) introduced by Boser et al. [1992] is a training algorithm, which is based on separating data points in a high dimensional linear space by hyperplanes. The learning algorithm itself requires the solution of a convex optimization problem trying to compute a hyperplane which separates a given sample set “as good as possible”. The primal objective, motivated by insights from statistical learning theory, is usually
not tackled directly but in the form of its Lagrangian dual optimization problem.

This dual problem is as well a quadratic optimization problem whose dimension is given by the number of the training samples \( \ell \). However, in usual applications the size of the training data is far too large to even store the quadratic matrix of the dual objective in memory. In addition this matrix is usually not sparse and standard optimization techniques, as interior point solvers, are not applicable to such problems.

On the other hand, the typical solution of the SVM optimization problem is sparse, i.e. only few variables of a dual optimal solution differ from zero. Those differing variables are called support vectors. In fact the generalization performance of this kind of learning machine can be related to the fraction of the variables contributing to the dual solution (see for example [Schölkopf and Smola, 2002, Proposition 7.4]).

A simple but powerful approach would therefore be to restrict the training to the set of support vectors only. Unfortunately they are not accessible in advance. In addition results by Steinwart [2004] show that the number of support vectors grows linearly with the size of the data depending on the Bayes risk. However, the fact that this set is smaller than the dimension of the problem itself has inspired an iterative scheme for support vector optimization which we will introduce in the following.

### Decomposition Algorithms

The first training algorithm for large scale SVM problems has been proposed by Osuna et al. [1997]. They introduced a technique based on partitioning the training sample into a working set and its complement, a fixed set. The main idea was to successively interchange violating variables contained in the fixed set by zero variables in the working set, iteratively improving the objective by solving the sub-problem on the working set only. At the end of the algorithm, all support vectors should fit into the working set and the problem could then be solved in the last iteration. As the index set \( \{1, \ldots, \ell\} \) is decomposed into a working set and its complement, such algorithms are called decomposition algorithms in the context of SVM optimization.

This decomposition idea of Osuna et al. [1997] was quite influential and has been developed in the years following Platt [1999], Joachims [1999], Chang and Lin [2001a], Shevade et al. [2000], Keerthi et al. [2001], Vishwanathan et al. [2003]. However, the most successful algorithms, which today still form the basis of many standard software packages like libSVM [Chang and Lin, 2001b], SVMLight [Joachims, 1999], SVMTorch [Collobert and Bengio, 2001], and Shark [Igel et al., 2008] implement an important variant of this algorithm: Although they stick to the decomposition into working set and fixed set, they do choose radically smaller working set sizes.
An extreme variant where only two indexes are chosen as working set is the sequential minimal optimization (SMO) algorithm proposed by Platt [1999]. Obviously, one cannot hope that finally a problem containing all support vectors is solved. Instead, modern decomposition methods are iterative optimization schemes: in any iteration they update the variables contained in the chosen working set by the optimal solution of the sub-problem induced by this set while keeping all other variables fixed.

Obviously, the selection of the working set is crucial for the convergence of the method. The most popular way to choose such indexes is selecting the ones which exhibit the clearest violation of the Karush-Kuhn-Tucker optimality conditions of convex optimization, so called maximum violating pairs (MVP). Such a selection strategy was first proposed in a technical distinct manner by Keerthi et al. [2001] but has in fact been invented by Platt [1999] and Joachims [1999]. That such selection schemes asymptotically converge to an optimal solution has first been shown by Keerthi and Gilbert [2002] for the special case of SMO and later by Lin [2001b]. These proofs have been given for the special case of $C$-support vector classification, but have later been generalized to more general settings by List and Simon [2004] and List [2004]. Note that there exist different proofs for the asymptotic convergence of such strategies for $C$-SVM by Takahashi and Nishi [2006] and Lucidi et al. [2007]. Yet, up to now there do not exist any convergence rates for this widespread selection technique.

Before discussing the convergence properties of decomposition algorithms in more detail, let us mention various other approaches to SVM optimization. The first type of training algorithms is based on a slight change of the primal objective Mangasarian and Musicant [1999, 2001a], which transforms the standard $C$-SVM into a simpler dual problem. Second, there have been applications of active set methods to the standard dual formulation by Mangasarian and Musicant [2001b] and Scheinberg [2006]. Recently competitive tools emerged from the application of cutting plane algorithms Tsotschantaridis et al. [2005] Joachims et al. to appear and stochastic gradient projection algorithms Shalev-Shwartz et al. [2007] to large scale SVM optimization problems. In this work we will, however, concentrate on the decomposition techniques described above, which are still implemented in the most popular software packages for SVM optimization.

Convergence Rates and Working Set Selection

The working set selection based on violating pairs has been motivated by the optimality conditions for the primal-dual problem of SVM. By construction they ensure that, if no violating pair can be found, an optimal solution is reached. Nevertheless, they are mere heuristics for selecting a working set. To be more concrete: The selection of an MVP can be viewed as a re-
stricted solution of Zoutendijk’s feasible direction problem [Zoutendijk, 1960]. Nonetheless, this approach does not ensure that the selected directions lead to a considerable progress in the next iteration, as we may easily hit the border of the constraint region. This has lead to convergence analysis for slightly modified selection strategies which take the distance to the border into account. Chang et al. [2000] have been the first to give such an analysis for C-SVMs. A general account has later been given by List and Simon [2004]. A practically applicable variant is the selection of so called certifying pairs [Hush and Scovel, 2003, Simon, 2004], which guarantees a certain advance along at least one direction contained in the induced subspace. These techniques have been extended and refined by List and Simon [2005, 2007] and Hush et al. [2006]. All those approaches share the property that they are conservative in so far as they prefer less steep directions if it can be ensured that one covers a long enough distance in the feasibility region. Note that second order selection heuristics like maximum gain selection proposed by Glasmachers and Igel [2006] can be viewed as conservative selection strategies as well.

The earlier mentioned MVP selection strategies do not care of stepwise guarantees. They aggressively choose the subset containing the steepest possible directions regardless of the distance to the borders. Nevertheless, there are results giving convergence rates for such algorithms. Note however that these results are very restrictive. For example, arguments by Lin [2001a] and Chen et al. [2006] guarantee linear advance to optimality only after a finite but unknown number of iterations. Furthermore this result is restricted to strictly positive definite matrices under some non-degeneracy conditions for the optimal solution. Recently Glasmachers [2008] could give a generalization for this bound which does not depend on these restrictive assumptions. Still, it only restricts the number of “free steps” which do not hit the border. For both results it is therefore unclear how meaningful they are for standard SVM optimization. Though, in typical SVM problems many iterations are not free steps, especially in the beginning. Furthermore, steps hitting the border occur throughout the optimization procedure, which challenges especially the first result as it relies on the fact, that during the phase where linear convergence is proven the algorithm encounters only free steps.

Remarkably, aggressive strategies outperform the standard conservative methods for typical instances of SVM optimization despite their lack of convergence guarantees. Even more, we conjecture that aggressive selection is advantageous, as it tends to move more indexes to the border of the feasibility region. In this sense aggressive strategies are possibly better tailored to SVM problems, where we expect many variables to be zero anyway. The results presented in this thesis possibly shed light on this discrepancy.

Let us finally mention one more popular selection heuristics which exhibit advantages over the classical MVP and certifying pair strategies: second-order heuristics. This selection strategy, which caused a significant speedup
for SVM algorithms, is based on the idea not only to consider a first order approximation of the function but consider as well second order information during the working set selection. Yet, a pairwise selection algorithm should not scan all $\ell^2$ possible pairs. This led to various heuristics on how to choose the subset of pairs one considers (see the work of Fan et al. [2005], Hush et al. [2006] and Glasmachers and Igel [2006] for details). Note that one still chooses small subspaces which may be heavily correlated. In fact, all these algorithms suffer as well from a badly conditioned quadratic term. Let us stress one special selection heuristics proposed by Hush et al. [2006]. The authors combine the choice of pairs they scan with a selection strategy induced by the conservative selection of rate certifying pairs. The resulting algorithms have been the first competitive decomposition algorithms, for which a convergence rate could be given. Those results will be seen to be special cases of the more general theory we will present in this thesis. In addition we will present convergence guarantees for popular aggressive selection strategies as MVP selection and the second order heuristics of Fan et al. [2005].

**Stopping Criteria**

In algorithms for optimization problems it is not only important to select strategies leading to convergent series of solutions, but as well to have sensitive stopping criteria. This has been a largely neglected topic in SVM optimization. Especially convergence arguments for conservative selection strategies typically ignored the optimality of associated primal solutions, which has first been mentioned by Hush et al. [2006].

In SVM optimization algorithms the stopping criteria implemented are tightly connected to the selection strategy. To be precise, MVP algorithms stop if the value of the maximum violating pair is smaller than some predefined accuracy. It has been known that this leads to approximately optimal primal solutions (see for example the presentation by Schölkopf and Smola [2002, Chapter 10.1]). Nonetheless, it is unclear whether such stopping criteria are of the same magnitude as the current distance to optimality.

Likewise for certifying pair algorithms: They stop whenever for the first time the promise of a maximal certificate is smaller than a predefined accuracy. In this case the situation has even been worse, as rate certificates have only been motivated on the dual problem itself. That means that they readily guarantee proximity to a dual optimum but do not ensure that associated primal points are approximately optimal as well.

Hush et al. [2006] gave a partial solution to this problem by the use of approximate Karush-Kuhn-Tucker conditions [Hush et al., 2005]. There is, however, a different possibility, which has been explored by List et al. [2007]. This solution is based on the fact that one can show that certain
optimality criteria do not change while dualizing a problem. We will describe such techniques in this thesis and will apply them to the stopping criteria implemented in standard software packages. This shows that these criteria are of this type and therefore one can derive primal optimality guarantees from them. In addition we will show that they are at most a magnitude larger than the achieved optimality. We are therefore able to give algorithms with guaranteed runtime and accuracy.

As standard SVM optimization algorithms using MVP selection strategies belong to this class, they produce approximately optimal solution to the primal empirical risk minimization problem in finite time. This, and the fact that we are able to give over all worst case guarantees on the run time of such algorithms, is one of the main results of this thesis.

Outline of the Thesis

The thesis is loosely divided in three parts. The first part consists of Chapters 1 and 2 where we will remind the reader of the basic theory necessary to understand the main results. During these chapters we will especially turn our attention to abstract duality theory (Section 1.2), which will later play an important role in the derivation of general optimality gaps. In addition we will focus on the exact derivation of many SVM problems from an abstract empirical risk minimization problem (Section 2.2.1) to which we will later apply our convergence results.

The second part will be devoted to the presentation of the main results: In Chapter 3 we will give a general account on decomposition algorithms for quadratic optimization problems. Note that we will, contrary to the usual approach, view decomposition algorithms as descent line search algorithms. Although this might seem a restriction we will see (compare Section 3.1.1 and Remarks 3.2.16, 3.2.51 and 3.3.6) that almost any decomposition algorithm used in practice is covered by our analysis. Furthermore, the restriction to one-dimensional subproblems has the advantage that we will be able to control the behavior of aggressive pairwise selection strategies (see Section 3.2, especially Section 3.2.4). The second part of Chapter 3 is devoted to describe the convergence guarantees for conservative selections.

In Chapter 4 we will use the theory of abstract duality to derive general optimality criteria (see Section 4.2). Their application to the special case of SVM problems in Section 4.3 reveals that usual stopping heuristics are exactly of this type.

Finally, combining the results from Chapter 3 and Chapter 4, we will be able to derive general convergence and run time guarantees for aggressive decomposition algorithms for SVM problems. These results are presented and discussed in Chapter 5.
Chapter 1

Prerequisites from Convex Optimization

This chapter will give a brief introduction to convex analysis and optimization theory. This introduction is by no means complete. The interested reader is referred to the standard textbooks for example by Rockafellar [1970], Hiriart-Urruty and Lemaréchal [1993a,b], Bertsekas [1995] and Boyd [2004].

1.1 Notational Conventions

Throughout the thesis \( \mathbb{N} := \{0, 1, 2, \ldots\} \) will denote the natural numbers. \( \mathbb{R} \) will denote the reals and \( \mathbb{R}^+ := \{t \geq 0\} \) the positive reals. For any \( t \in \mathbb{R} \) we will use the convention to write shortly \( [t]^+ \) instead of \( \max\{0, t\} \).

We will consider \( x \in \mathbb{R}^\ell \) to be a column vector and consequently \( x^\top \) is a row vector. We will write shortly \( x^\top y = \langle x, y \rangle \) for the standard scalar product on \( \mathbb{R}^\ell \). \( e \in \mathbb{R}^\ell \) will denote the all ones vector and \( e_i \in \mathbb{R}^\ell \) the standard basis vector, which has 1 for its \( i \)-th component and 0 otherwise. The dimension of these vectors will be clear from the context. The inequality \( x \leq y \) for \( x, y \in \mathbb{R}^\ell \) is short for the component wise inequalities \( x_i \leq y_i \) for all \( i \in \{1, \ldots, \ell\} \). Given a subset \( I \subset \{1, \ldots, \ell\} \) and a vector \( x \in \mathbb{R}^\ell \) we will often write \( x_I \) for the vector consisting only of entries \( x_i \), where \( i \in I \).

Given a square matrix \( Q \in \mathbb{R}^{\ell \times \ell} \), we will denote its minimal eigenvalue by \( \lambda_{\text{min}}(Q) \) and its maximal eigenvalue by \( \lambda_{\text{max}}(Q) \). If \( \lambda_{\text{min}}(Q) > 0 \) the ratio is denoted by \( \kappa(Q) := \frac{\lambda_{\text{max}}(Q)}{\lambda_{\text{min}}(Q)} \) and is called the condition (number) of \( Q \). We will often consider the \( q \times q \) principal sub matrices \( Q_{q \times q} \) of \( Q \) and denote their maximal eigenvalue by \( \lambda_{\text{max},q}(Q) := \lambda_{\text{max}}(Q_{q \times q}) \). The relative
Chapter 1. Prerequisites from Convex Optimization

A convex function is called \emph{proper} if for all \( x \in V \) we have \( \partial f \) is constant or tends to \(-\infty\) as \( t \to 0 \). A direction \( d \in V \) is called \emph{direction of recession} if it is contained in \( C \) and all \( t \) \( > 0 \) have \( f(x + t(y - x)) \leq f(x) + t(f(x) - f(y)) \) for all \( x, y \in V \) and \( t \in [0, 1] \).

A function \( f : V \to \mathbb{R} \) is called \emph{convex function} iff \( \text{epi} f \) is a convex in \( V \times \mathbb{R} \). This is equivalent to

\[
\text{epi} f := \{(x, t) \mid f(x) \leq t \} \subseteq V \times \mathbb{R}.
\]

A convex function is called \emph{proper} iff for all \( x \in V \) we have \( f(x) > -\infty \) and \( f(x) < +\infty \) for at least one \( x \in V \). The set of points where \( f \) is finite will be called \emph{domain} of \( f \). It is denoted by \( \text{dom} f := \{ x \mid f(x) \in \mathbb{R} \} \). As for convex sets, we can study the behavior of convex functions “at infinity”. Consider therefore the function \( f_\infty' : V \to \mathbb{R} \), defined as follows:

\[
f_\infty'(d) := \sup_{t > 0} \frac{f(x_0 + td) - f(x_0)}{t}.
\]

Note that this definition is independent of \( x_0 \in \text{dom} f \) and \( f_\infty' \) is convex as well (see for example [Hiriart-Urruty and Lemaréchal [1993a, Proposition 3.2.2]]. This function is called the \emph{recession function} of \( f \) or sometimes asymptotic function. Then the set of directions \( C^\infty(f) := \{ d \mid f'_\infty(d) \leq 0 \} \), where \( f \) is constant or tends to \(-\infty \) is called recession cone of \( f \). The set \( C^0(f) := \{ d \mid f'_\infty(d) = 0 \} \) is called the \emph{constancy space} of \( f \).

The concept of differentiability can be extended to convex functions: \( x^* \in V^* \) is called \emph{sub-gradient} of \( f \) at \( x \in V \) iff for all \( y \in V \)

\[
f(y) \geq f(x) + x^*(x - y).
\]

The set of all subgradients of \( f \) at some point \( x \in V \) is called \emph{sub-differential} of \( f \) at \( x \) and is denoted by \( \partial f(x) \subseteq V^* \). Note that for finite vector spaces we have \( V \simeq V^* \) and we can as well consider \( \partial f(x) \) as a subset of \( V \).
1.2 Abstract Duality

The next section will give a brief introduction to abstract duality, which will be a main tool in the general analysis of certain stopping criteria used in SVM optimization (see Chapter [1]). We will describe the main ideas in detail but will omit all proofs. The interested reader is referred to the book by Hiriart-Urruty and Lemaréchal [1993b, Chapter XII] for a deeper introduction and the proof details.

1.2.1 Introduction

Let us consider the following abstract optimization problem:

Problem 1.2.1: Let $U$ be a nonempty set and let $\varphi : U \to \mathbb{R}$ and $c_i : U \to \mathbb{R}$, $i = 1, \ldots, m$ be real valued functions. Let $c : U \to \mathbb{R}^m$ denote the function with components $c_i$. We will denote any maximization problem $P(\varphi, U, c)$

$$\sup_{u \in U} \varphi(u) \text{ s.t. } u \in U, c(u) \leq 0. \quad (1.2.1)$$

an instance of Abstract Constrained Optimization (ACO).

Remark 1.2.2: So far we did not assume anything about the structure of these problems. We shall see later that it will be necessary to request such structural constraints on problems $P(\varphi, U, c)$ for the duality approach to be useful. Nonetheless we will develop the theory in this chapter as general as possible.

Remark 1.2.3: Note, that we might as well define the class ACO as Abstract Constrained Minimization, simply by considering the objective $\varphi' := -\varphi$. We will therefore use the term Abstract Constrained Optimization interchangeably for maximization as well as minimization problems depending on which of them is more convenient.

Note that the following class of problems are natural sub-problems of this general account:

Problem 1.2.4: Let $V$ be a linear space and $f : V \to \mathbb{R}$ and $c_i : V \to \mathbb{R}$, $i = 1, \ldots, m$ be convex functions. We will denote any optimization problem $P(\varphi, U, c)$

$$\sup_{x \in V} -f(x) \text{ s.t. } x \in V, c(x) \leq 0. \quad (1.2.2)$$

an instance of Convex Constrained Optimization (CCO).

Problem 1.2.5 (Quadratic Constrained Optimization): Let $U = \mathcal{H}$ be a Hilbert space, $w \in \mathcal{H}$ and $Q : \mathcal{H} \to \mathcal{H}$ be a nonnegative self-adjoint operator, such that $Q : \ker(Q)^\perp \to \ker(Q)^\perp$ has a continuous inverse $Q^{-1}$ and define

$$\varphi(u) := -\frac{1}{2} \langle Qu, u \rangle + \langle w, u \rangle.$$
In addition consider a linear mapping \( A : \mathcal{H} \to \mathbb{R}^m \) and \( d \in \mathbb{R}^m \). Then any optimization problem \( \mathcal{P}_{QP}(\varphi, U, A, d) \)

\[
\sup_{u \in \mathcal{H}} \varphi(u) \quad \text{s.t.} \quad Au - d \leq 0
\]

will be called an instance of quadratic constrained optimization (QCO).

Let us return to the general setting and consider an instance \( \mathcal{P}(\varphi, U, c) \) of ACO the set \( \mathcal{R}(\mathcal{P}) := \{ u \in U \mid c(u) \leq 0 \} \) is called feasibility region of \( \mathcal{P} \) and each \( u \in \mathcal{R}(\mathcal{P}) \) is called a (primal) feasible point. With any instance \( \mathcal{P} \) we will associate the following function which will be a main tool in the theory to develop:

**Definition 1.2.6 (Lagrangian function):** Given an instance \( \mathcal{P}(\varphi, U, c) \) of ACO we call the function \( L : U \times \mathbb{R}^m \to \mathbb{R} \), where

\[
L(u, \lambda) := \varphi(u) - \lambda^\top c(u)
\]

the Lagrangian associated with \( \mathcal{P} \). For any given \( \lambda \in (\mathbb{R}^+)^m \) this Lagrangian can be seen as a disturbance of the objective \( \varphi \), punishing any violation of an inequality constraint \( c_i(u) > 0 \) by \( \lambda_i c_i(u) \) and rewarding each strictly fulfilled constraint \( c_i(u) < 0 \) by \( \lambda_i c_i(u) \). We will call such \( \lambda \) dual parameters. Based on this idea, for any fixed \( \lambda \in (\mathbb{R}^+)^m \) one can associate with any instance \( \mathcal{P}(\varphi, U, c) \) of ACO the unconstrained optimization problem:

\[
\sup_{u \in U} L(u, \lambda) \quad \text{s.t.} \quad u \in U.
\]

We will call \( (1.2.4) \) the Lagrange problem of \( \mathcal{P}(\varphi, U, c) \) associated with the dual parameter \( \lambda \) and denote the possibly empty set of its maximizers by

\[
U_{\lambda} := \arg \max_{u \in U} L(u, \lambda).
\]

### 1.2.2 The Main Idea of Duality

To illustrate the main idea behind the duality approach we have to assume that the Lagrangian problem is “substantially” simpler then the original problem \( \mathcal{P}(\varphi, U, c) \). In this case let us imagine to have a black box at hand, which internally computes an optimal solution \( u_{\lambda} \in U_{\lambda} \) of \( (1.2.4) \) and will output the corresponding optimal value \( L(u_{\lambda}, \lambda) \) and its constrained value \( c(u_{\lambda}) \), given an arbitrary \( \lambda \in (\mathbb{R}^+)^m \), such that \( U_{\lambda} \neq \emptyset \). Obviously the output of the black box is ambiguous, as the mapping \( \lambda \mapsto c(u_{\lambda}) \) is not well defined and depends on the internally chosen \( u_{\lambda} \).

\[\text{Note that although it is customary to define the Lagrangian to be } \infty \text{ when } \lambda \notin (\mathbb{R}^+)^m \text{ the definition } L(\cdot, \lambda) \text{ will be convenient when applying the subdifferential calculus.}\]
The following fundamental theorem and the subsequent corollary show, that in fact the optimality of such a $u_\lambda \in U_\lambda$ with respect to the primal problem $P(\varphi, U, c)$ can be decided by $c(u_\lambda)$ and $\lambda$ alone: Namely the feasibility of $u_\lambda$, i.e. $c(u_\lambda) \leq 0$, and the so called complementary slackness.

**Theorem 1.2.7 (Everett):** Given an instance $P(\varphi, U, c)$ and a fixed $\lambda \in (\mathbb{R}^+)^m$. Assume that there exists a $u_\lambda \in U_\lambda$, then $u_\lambda$ is as well an optimal solution of

$$\sup \varphi(u) \quad \text{s.t. } u \in U, \ c_i(u) \leq c_i(u_\lambda) \text{ for all } i : \lambda_i > 0 . \quad (1.2.5)$$

In addition, given a dual parameter $\lambda \in (\mathbb{R}^+)^m$ such that there exists a $u_\lambda \in U_\lambda$ such that $c(u_\lambda) \leq 0$ and $\lambda^\top c(u_\lambda) = 0$, then $u_\lambda$ is an optimal solution of $P(\varphi, U, c)$.

A generic dual algorithm can therefore be summarized as follows: Iteratively scan the dual-space, using the information of the black box, until a $\lambda \in (\mathbb{R}^+)^m$ is found, such that there exists a $u \in U_\lambda$, which is feasible and fulfills the complementary slackness condition from Theorem 1.2.7. If such a $(u, \lambda)$ is found, we output $u$ as optimal solution of $P(\varphi, U, c)$. We will flesh out this abstract motivation in the following.

### 1.2.3 The Dual Problem and Its Properties

This aim can be formalized as follows::

**Problem 1.2.8:** Given an instance $P(\varphi, U, c)$ of ACO we will call the following problem

$$\text{Find } \lambda \in (\mathbb{R}^+)^m \quad \text{s.t. } \exists u \in U : \begin{cases} u \in U_\lambda \\ c(u) \leq 0, \lambda^\top c(u) = 0 \end{cases} \quad (1.2.6)$$

the Dual Approach to $P(\varphi, U, c)$.

**Remark 1.2.9:** If Problem 1.2.6 is feasible we say that the dual approach succeeds. Note however that this does not necessarily hold: It may well be that for all $\lambda \in (\mathbb{R}^+)^m$ we have $R(P(\varphi, U, c)) \cap U_\lambda = \emptyset$ and the dual approach fails from the very beginning. A simple counter-example is given by the one dimensional problem

$$\max u \quad \text{s.t. } 2u - 1 \leq 0, u \in \{0, 1\} .$$

where $U_\lambda \neq \emptyset$ iff $\lambda = \frac{1}{2}$. In this case we have $U_\lambda = \{0, 1\}$ but $c(1) = 1 > 0$ and $\frac{1}{2}c(0) = -\frac{1}{2} \neq 0$.

Let us now introduce a main tool in duality: a mapping of $\lambda$ to the optimal values of the associated Lagrangian, which is known as the dual function.
Chapter 1. Prerequisites from Convex Optimization

Definition 1.2.10 (Dual function): Given an instance $\mathcal{P}(\varphi, U, c)$ of ACO. Then the following function $\psi : \mathbb{R}^m \to \mathbb{R}$, defined by

$$
\psi(\lambda) := \sup_{u \in U} L(u, \lambda)
$$

is the dual function associated to $\mathcal{P}$.

The next lemma shows that without any assumptions on $\mathcal{P}(\varphi, U, c)$ the dual function has some remarkable properties.

Lemma 1.2.11: The dual $\psi$ is a proper convex function and for $u \in U_\lambda$ we have $-c(u) \in \partial \psi(\lambda)$.

A fundamental property of the dual function is the fact that it is an upper bound for any value of the primal function $\varphi$ as the following theorem states:

Theorem 1.2.12 (Weak Duality): For all $\lambda \in (\mathbb{R}^+)^m$ and any feasible $u \in U$

$$
\psi(\lambda) \geq \varphi(u).
$$

The main guide to solve the dual approach (1.2.6) will therefore be the following optimization problem:

Problem 1.2.13 (Dual Problem): Given an instance $\mathcal{P}(\varphi, U, c)$ of ACO, we will call the optimization problem $\mathcal{P}(\psi) := \mathcal{P}(\psi, \mathbb{R}^m, -\text{Id}_{\mathbb{R}^m})$

$$
\inf \psi(\lambda) \quad \text{s.t.} \quad \lambda \in \mathbb{R}^m, \; \lambda \geq 0
$$

the associated Dual Problem. From Lemma 1.2.11 we see that $\mathcal{P}(\psi)$ is an instance of Constrained Convex Optimization (CCO) and therefore of ACO.

Note that the dual is an instance of CCO and therefore considerably simpler than $\mathcal{P}(\varphi, U, c)$ – still under the assumption that the Lagrangian problem (1.2.4) is “simple”. In addition, the solutions of the dual problem $\mathcal{P}(\psi)$ are tightly connected with the success of the dual approach itself.

Theorem 1.2.14: The following two statements characterize the connection between the dual approach (1.2.6) and the dual problem $\mathcal{P}(\psi)$:

- Any solution of the dual approach (1.2.6) is a solution of the dual problem $\mathcal{P}(\psi)$. Conversely, if the dual approach has a solution at all, then any optimal solution $\hat{\lambda}$ of $\mathcal{P}(\psi)$ is a solution of (1.2.6).

- Suppose the dual approach (1.2.6) is feasible, then for any dual optimal $\hat{\lambda}$ the primal solutions are given by $u \in U_{\hat{\lambda}}$ such that $c(u) \leq 0$ and $\hat{\lambda}^\top c(u) = 0$. 

This means: If ever the dual approach works, the optimal solutions of $P(\psi)$ readily deliver the required solutions of the primal problem $P(\varphi, U, c)$. The following section will discuss the conditions under which the dual approach succeeds.

### 1.2.4 When Does the Dual Approach Succeed?

In the following we will denote with

$$C_\lambda := \{c(u) \mid u \in U_\lambda\}$$

the constraint values of all optimal values of the Lagrangian problem (1.2.4) for a given $\lambda$. From Lemma [1.2.11] and the fact that $\partial \psi(\lambda)$ is a closed convex set we conclude

$$-\text{co} \, C_\lambda \subseteq \partial \psi(\lambda), \quad (1.2.9)$$

where $-\text{co} \, U$ denotes the convex hull of a set $U \subset V$. The reverse inclusion will prove to be extremely useful so we recall the following definition from [Hiriart-Urruty and Lemaréchal 1993b, Def. XII.2.3.1]:

**Definition 1.2.15 (Filling Property):** We say the filling property holds for $\lambda$, iff

$$-\text{co} \, C_\lambda = \partial \psi(\lambda). \quad (1.2.10)$$

If in addition $C_\lambda$ is convex we say that the strict filling property holds for $\lambda$.

To illustrate the importance of (1.2.10) we end this section by the following theorem.

**Theorem 1.2.16:** Assume the filling property holds for a given $\lambda \geq 0$. Then $\lambda$ is an optimal dual solution iff there exist $s \leq m + 1$ feasible primal points $u_1, \ldots, u_s \in U_\lambda$ and $\alpha_1, \ldots, \alpha_s \geq 0$ such that $\sum_{r=1}^s \alpha_r = 1$,

$$\sum_{r=1}^s \alpha_r c(u_r) \leq 0 \quad \text{and} \quad \lambda_i \sum_{r=1}^s \alpha_r c_i(u_r) = 0 \quad \text{for all } i = 1, \ldots, m.$$

Moreover if the strict filling property holds for an optimal $\lambda$, then the duality gap $\psi^* - \varphi^*$ is 0 and the solutions of the primal problem are given by the feasible $u \in U_\lambda$, for which the complementary slackness condition $\lambda^\top c(u) = 0$ holds.

**Remark 1.2.17:** Obviously this is a rather abstract criterion for the duality approach to be successful. Let us stress however that we will exactly build on this abstract property to prove the relation of the primal dual gaps we are interested in (see Chapter [4]). To summarize the forthcoming results: We will show that for any dual point $\lambda$ where strict filling holds certain primal and dual optimal guarantees coincide. From this point of view Theorem 1.2.16 is a special case of Theorem [4.2.8].
Let us finally give some remarks on conditions for the filling property to hold. It is therefore crucial to note, that the dual function is defined as a supremum of a family of linear functions \( L(u, \cdot) \) \( u \in U \). In addition note, that the (unique) “sub-gradient” of any of this functions is given by \(-c(u)\) for \( u \in U \), which is exactly the constraint value at \( u \in U \). Consequently the question whether the filling property holds or not is exactly the following one: When is the sub-gradient of \( \sup_u L(u, \cdot) \) at \( \lambda \) completely determined by \( \bigcup_{u \in U} \partial L(u, \cdot) = \bigcup_{u \in U} -c(u) \). A general account on this question has been given by Solov’ev [1998]. We will however abandon this topic and solely concentrate on one special class of optimization problems: quadratic constrained optimization.

### 1.2.5 Quadratic Constrained Optimization

As we will see the optimization problems induced by support vector machine optimization are all special instances of QCO (see Problem 1.2.5). Let us therefore elaborate the dual approach more detailed for such problems.

For any instance \( P_{QP}(\varphi, U, A, d) \) the associated Lagrangian is given by

\[
L(u, \lambda) = -\frac{1}{2} \langle Qu, u \rangle + \langle w, u \rangle - \lambda^\top (Au - d)
\]

For the dual criterion function, we can easily derive the following properties:

**Lemma 1.2.18:** Given an instance \( P_{QP} \) of QCO. Then the dual criterion function \( \psi : \mathbb{R}^m \to \mathbb{R} \) is given by:

\[
\psi(\lambda) = \begin{cases} 
\frac{1}{2} \langle Q^{-1}(w - A^*\lambda), w - A^*\lambda \rangle + \lambda^\top d & (w - A^*\lambda) \in \text{img} \ Q \\
\infty & \text{otherwise}
\end{cases}
\]

Therefore

\[
\text{dom} \ \psi = \{ \lambda \in \mathbb{R}^m \mid w - A^*\lambda \in \text{img} \ Q \}
\]

and for any \( \lambda \in \text{dom} \ \psi \)

\[
U_\lambda = Q^{-1}(w - A^*\lambda) + \text{kern} \ Q.
\]

The subdifferential of \( \psi \) is given by

\[
\partial \psi(\lambda) = AQ^{-1}(A^*\lambda - w) + d + \{ \mu \mid A^*\mu \in \text{img} \ Q \}^\perp,
\]

and therefore the unique gradient on \( \text{dom} \ \psi \subseteq \mathbb{R}^m \) is

\[
\nabla \psi(\lambda) := Q^{-1}(w - A^*\lambda) + d \in \mathbb{R}^m.
\]
Proof. As \( \varphi : U \to \mathbb{R} \) is a continuously differentiable function, we can calculate the optimum of the Lagrangian problem (1.2.4) by solving

\[
0 = \partial_u L(u, \lambda) = -Qu + w - A^*\lambda \in U^*.
\]

This is obviously not possible if \( w - A^*\lambda \) is not orthogonal to \( \text{kern} Q \) and therefore \( \psi(\lambda) = \infty \) in this case and the first part of (1.2.11) follows. Now suppose that \( w - A^*\lambda \in (\text{kern} Q)^\perp = \text{img} Q \). We then can solve for \( u \) and get

\[
u_\lambda = Q^{-1}(w - A^*\lambda),
\]

which proves (1.2.12) and simply plugging \( u_\lambda \) in \( L(u, b) \) gives \( \psi(\lambda) \) in the second case of (1.2.11). As therefore \( \text{dom} \psi := \{ \lambda \mid w - A^*\lambda \in \text{img} Q \} \), we conclude that

\[
\partial \psi(\lambda) = AQ^{-1}(A^*\lambda - w) + d + \{ \mu \mid A^*\mu \in \text{img} Q \}^\perp
\]

and consequently \( \nabla \psi(\lambda) \) as in (1.2.14) is the unique (sub-)gradient on \( \text{dom} \psi \).

\[\square\]

**Problem 1.2.19:** Given an instance \( \mathcal{P}_{QP} \) of QCO, then the dual problem \( \mathcal{P}_{QP}(\psi) \) is given by

\[
\inf \psi(\lambda) \quad \text{s.t.} \quad (w - A^*\lambda) \in (\text{kern} Q)^\perp, \lambda \geq 0.
\]

(1.2.15)

The next lemma, which includes the linear programming case, shows that the strict filling property holds for all instances of (1.2.15):

**Theorem 1.2.20:** Given an instance \( \mathcal{P}_{QP} \) of QCO, then for any \( \lambda \in \text{dom} \psi \) the strict filling property (1.2.10) holds.

Proof. For \( \lambda \in \text{dom} \psi \) we obtain

\[
-C_\lambda := \{ -c(u) \mid u \in U_\lambda \}
= \{ d - Au \mid u \in Q^{-1}(w - A^*\lambda) + \text{kern} Q \}
= d + AQ^{-1}(A^*\lambda - w) + A\text{kern} Q.
\]

From (1.2.13) we know

\[
\partial \psi(\lambda) = AQ^{-1}(A^*\lambda - w) + d + \{ \mu \mid A^*\mu \in \text{img} Q \}^\perp
\]

and according to Lemma 1.2.11 \( -C_\lambda \subseteq \partial \psi(\lambda) \) it suffices to show that \( (A\text{kern} Q)^\perp \subseteq \{ \mu \mid A^*\mu \in \text{img} Q \} \). To that end suppose that \( \mu \perp A\text{kern} Q \). Then we have \( \langle A^*\mu, z \rangle = \langle \mu, Az \rangle = 0 \) for all \( z \in \text{kern} Q \) which implies \( A^*\mu \in \text{img} Q \).
1.3 Optimization Theory

In the following we will shortly introduce in some basic facts from optimization theory. Their exhibition is especially tailored to the needs in this work. For a more general introduction to convex optimization see the books of Bertsekas [1995] and Boyd [2004] and Hiriart-Urruty and Lemaréchal [1993a,b].

1.3.1 Notational Conventions

All instances we will consider will be instances of CCO. Let therefore recall the general one given in Problem 1.2.4 and introduce some notational conventions we will use throughout this thesis:

$$\inf_{x \in \mathbb{R}^\ell} f(x) \quad \text{s.t.} \quad x \in c_i(x) \leq 0, i = 1, \ldots, m,$$

where $f$ and the $c_i$ are assumed to be convex. If there exists at least one $x$ such that $c_i(x) \leq 0$ for all $i = 1, \ldots, m$ and $f$, and the $c_i$ have no direction of recession in common the optimal value exists and is finite [Rockafellar, 1970, 27.3.3]. We will denote it with $\hat{f} := \text{val P}$ and will write $\hat{x}$ for any $x$ with $f(x) = \hat{f}$. $\Delta(x) := f(x) - \hat{f} \geq 0$ measures the distance to optimality for an arbitrary feasible $x \in \mathbb{R}(\mathcal{P})$. We will often be concerned with iterative optimization schemes, which solve some simpler problem leading to a sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$. For any iteration $n$ we will use the notation $\Delta^{(n)} := \Delta(x^{(n)})$.

1.3.2 Convergence Schemes

One of the main questions for all such iterative optimization techniques is whether they converge towards an optimal point $\hat{x}$ in the sense that $\Delta^{(n)} \leq \varepsilon$ for some predefined accuracy $\varepsilon > 0$. Obviously a minimal requirement is given by requesting that $f(x^{(n+1)}) < f(x^{(n)})$. In this case we talk of a descent method.

Let us recall two common convergence types for descent optimization methods, which will be used at many places throughout this work.

**Lemma 1.3.1** (Linear Convergence): Given an $\varepsilon > 0$ and an $\alpha > 0$. Any descent method, such that for all $n \geq 0$

$$f(x^{(n+1)}) - f(x^{(n)}) \geq \alpha \Delta^{(n)},$$

and therefore

$$\Delta^{(n+1)} \leq \Delta^{(n)} - \alpha \Delta^{(n)}$$

is within $\varepsilon$ of optimality after at most

$$n := \max \left\{ 0, \left[ \frac{1}{\alpha} \ln \frac{\Delta^{(0)}}{\varepsilon} \right]\right\}.$$
iterations. We call such a procedure linear convergent.

Proof. Consider the following chain of inequalities
\[ \Delta(n) \leq (1 - \alpha)^n \Delta(0) \leq e^{-\alpha n} \Delta(n) \leq \varepsilon. \]
Solving for \( n \) proves the claim. \( \square \)

The second scheme, which will recur during our work is the following:

**Lemma 1.3.2 (Quadratic Convergence):** Given an \( \varepsilon > 0 \) and an \( \alpha > 0 \). Any descent method, such that for all \( n \geq 0 \)
\[ f(x^{(n+1)}) - f(x^{(n)}) \geq \alpha \left( \Delta^{(n)} \right)^2, \]
and therefore
\[ \Delta^{(n+1)} \leq \Delta^{(n)} - \alpha \left( \Delta^{(n)} \right)^2, \]
is within \( \varepsilon \) of optimality after at most
\[ n := \left\lceil \frac{1}{\alpha \varepsilon} \right\rceil \]
iterations. We call such a procedure quadratic convergent.

Proof. The following proof is due to Dunn [1979]. Making the pessimistic assumption \( \Delta^{(n)} > 0 \) we can consider the reciprocals \( \frac{1}{\Delta^{(n)}} \). Then
\[ \frac{1}{\Delta^{(n+1)}} - \frac{1}{\Delta^{(n)}} \geq \frac{1}{(1 - \alpha \Delta^{(n)}) \Delta^{(n)}} - \frac{1}{\Delta^{(n)}} = \frac{\alpha}{1 - \alpha \Delta^{(n)}} \geq \alpha, \]
where the last inequality is implied by the fact, \( 0 \leq 1 - \alpha \Delta^{(n)} \leq 1 \) as otherwise \( \Delta^{(n+1)} \leq (1 - \alpha \Delta^{(n)}) \alpha_n < 0 \). Then
\[ \frac{1}{\Delta^{(n)}} = \frac{1}{\Delta^{(0)}} + \sum_{i=0}^{n-1} \left( \frac{1}{\Delta^{(i+1)}} - \frac{1}{\Delta^{(i)}} \right) \geq n \alpha, \]
and therefore
\[ \Delta^{(n)} \leq \frac{1}{n \alpha} \]
which implies the claim. \( \square \)

With this schemes in mind we will later try to design iterative methods for support vector machine optimization, which guarantee an advance in each iteration, as used in Lemma [1.3.1] and Lemma [1.3.2]. In the next section we will present a classic strategy for the unconstrained case, whose adaption to the constrained setting has often been used in SVM optimization.
1.3.3 Descent Line Search and Gradient Methods

In the following we will present a classic descent method for unconstrained optimization: Descent line search as given in Algorithm 1.1 (compare [Boyd, 2004, Chapter 9.2]). Throughout this work we will only consider exact line search procedures, where \( t \) is chosen to minimize \( f(x + t \cdot d) \). Therefore the only crucial point remaining is the choice of \( d \), such that \( \nabla f(x) \top d < 0 \).

Algorithm 1.1: Descent Line Search Algorithm

\[
\begin{align*}
\text{Input:} & \quad f, \quad \varepsilon > 0 \\
\text{Output:} & \quad x \text{ such that } \Delta(x) \leq \varepsilon \\
\text{while } & \Delta(x^{(n)}) > \varepsilon \text{ do} \\
1 & \quad \text{Select } d \in \mathbb{R}^\ell, \text{ where } \nabla f(x) \top d < 0; \\
2 & \quad \text{Find optimal solution } \hat{t} \text{ of } f(x + t d) ; \\
3 & \quad \text{Update } x^{(n+1)} \leftarrow x^{(n)} + \hat{t} \cdot d; \\
4 & \quad n \leftarrow n + 1; \\
5 & \quad \text{return } x^{(n)}; \\
6 & \quad \text{end while}
\end{align*}
\]

One of the most popular variants of choosing \( d \) is given by choosing the so-called steepest descent direction (see for example [Boyd, 2004, Chapter 9.4] for details). The main idea is that \( d \) is chosen to maximize

\[

d = \arg \max \left\{ -\nabla f(x) \top d \mid \|d\| \leq 1 \right\}, \tag{1.3.1}
\]

where \( \|\cdot\| \) is an arbitrary norm on \( \mathbb{R}^\ell \). This method subsumes many other well known optimization methods: In the case of the Euclidean norm \( \|d\|_2 = \sqrt{d \top d} \) the choice of \( d \) coincides with the well-known gradient-descent methods choosing \( d = -\nabla f(x) \), and in the case of the 1-norm \( \|d\|_1 = \max |d_i| \) it coincides with the steepest gradient-method \( d = \max -\nabla f(x)_i \) (see [Boyd, 2004, Section 9.4.2-3]).

Let us shortly discuss the convergence of such methods for the quadratic function

\[
f(x) := \frac{1}{2} x \top Q x - w \top x,
\]

where \( Q \in \mathbb{R}^{\ell \times \ell} \) is a symmetric strictly positive definite matrix and \( w \in \mathbb{R}^\ell \) an arbitrary vector. First note, that \( f \) is strictly convex and the unique optimal point \( \hat{x} \) is given by \( \nabla f(\hat{x}) = 0 \), which is achieved for \( \hat{x} = Q^{-1} w \). Simple calculation reveals (for details see Lemma 3.1.14) that the optimal step length is given by

\[
\hat{t} := \frac{\nabla f(x) \top d}{d \top Q d}.
\]

The standard guarantee for the unconstrained advance is given in the following Lemma (compare to Lemma 3.2.3).
Lemma 1.3.3: Given an arbitrary \( x \in \mathbb{R}^\ell \), then if \( d \) is chosen according to (1.3.1), where \( \| \cdot \| = \| \cdot \|_2 \) is the Euclidean norm, then
\[
f(x) - f(x + \hat{t} \cdot d) \geq \frac{1}{\kappa(Q)} \Delta(x),
\]
where \( \kappa(Q) \) is the condition of the matrix \( Q \) given by
\[
\kappa(Q) := \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)}.
\]

Proof. See for example [Boyd, 2004, Chapter 9.2].

With the help of Lemma 1.3.3 we conclude

Corollary 1.3.4: For any strictly convex quadratic function
\[
f(x) = \frac{1}{2} x^\top Q x - w^\top x
\]
Algorithm 1.1, using exact line-search along the gradient \(-\nabla f(x^{(n)})\) in each iteration is within \( \varepsilon > 0 \) of optimality, after
\[
n := \left\lceil \kappa(Q) \ln \frac{\Delta^{(0)}}{\varepsilon} \right\rceil
\]
iterations.

Let us give a few remarks

Remark 1.3.5: 1) Convergence rates for other bounds can be derived from this guarantee (see [Boyd, 2004] for details). Let us highlight the steepest gradient descent based on the 1-norm \( \| \cdot \|_1 \). Using the fact that for any \( d \in \mathbb{R}^\ell \) we have \( \|d\|_2 \leq \|d\|_1 \leq \sqrt{\ell} \|d\|_1 \) we can derive
\[
f(x) - f(x + \hat{t} \cdot d) \geq \frac{1}{\ell \kappa(Q)} \Delta(x),
\]
and therefore this method converges in at most
\[
n := \left\lceil \ell \kappa(Q) \ln \frac{\Delta^{(0)}}{\varepsilon} \right\rceil
\]
iterations.

2) These bounds are tight up to a constant factor as one can easily see for the simple two-dimensional problem
\[
\min_x \quad \frac{1}{2} \left( \frac{1+\gamma}{2} \frac{1-\gamma}{2} \frac{1-\gamma}{2} \right).
\]
for $\lambda \in (0, 1)$ which has condition number $\kappa = \lambda^{-1}$. Note that this is the standard example $\begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}$ rotated by $\pi/2$, such that the 1 norm-line search does not (accidentally) perform a conjugate gradient descent converging in two steps.

3) In case of SVM optimization algorithm we will later deal with a constrained optimization problem. Typically one cannot expect linear convergence in this case (see [Bertsekas, 1995, Section 2.2]). In addition the kernel matrix will often exhibit a non trivial kernel and is therefore not strictly positive definite. We will later see how the difficulties in this case can be circumvented.

4) It is worth to remark that the rates for SVM optimization given by Lin [2001a] and Chen et al. [2006] can be reduced to the arguments above if one interprets decomposition algorithms as 1-norm gradient descent (see Section 3.2.1). To use the bounds above Lin did extend arguments from the classical SVM convergence paper [Lin, 2001b] showing that from some iteration $n_0$ on, candidates in the interior of the feasibility region stay in the interior. This implies that for $n > n_0$ the optimization problem solved by the decomposition method can be considered unbounded. Under strict positive definiteness assumptions the linear convergence then follows from the above arguments.
Chapter 2
Prerequisites from Learning Theory

The following chapter will be devoted to a brief introduction into the basic building stones of the SVM and the introduction of the optimization problems such learning machines induce.

It is divided into two parts. In Section 2.1 we will shortly introduce the main ideas behind statistical learning theory and large margin classifiers. In Section 2.2 we will use this basic results to motivate an abstract empirical risk minimization problem, which allows us to deduce many common instances of the SVM from one basic problem (see Sections 2.2.2–2.2.6 for details).

2.1 Introduction

This section gives a short introduction to basics of learning theory. It is admittedly far too short and by no means complete. The interested reader is referred to standard textbooks like Kearns and Vazirani [1994], Vapnik [1995, 1998], Christianini and Shawe-Taylor [2003], Schölkopf and Smola [2002] and Shawe-Taylor and Christianini [2004].

2.1.1 Basics of Statistical Learning Theory

Let us introduce the standard problem setting of statistical learning theory (see for example Vapnik [1995]). We assume to have a finite set

\[ T = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\} \subset \mathcal{X} \times \mathcal{Y} \]
Chapter 2. Prerequisites from Learning Theory

of input/output pairings, which are i.i.d. drawn according to an unknown but fixed probability distribution $D$ on $\mathcal{X} \times \mathcal{Y}$. Note that we will often use the vector writing $\vec{x} \in \mathcal{X}^\ell$ and $\vec{y} \in \mathcal{Y}^\ell$. A learning algorithm then performs the following general task: Given a set of samples $T$, choose a hypothesis $h_T \in \mathcal{F}$ from a predefined set of measurable hypotheses

$$
\mathcal{F} := \{ h : \mathcal{X} \to \mathcal{Y} \mid h \text{ measurable} \},
$$

such that the following risk is small:

$$
R_D(h) := \int_{\mathcal{X} \times \mathcal{Y}} c(y, h(x)) dD(x, y).
$$

(2.1.1)

Here $c : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ denotes a measurable loss function, which is used to penalize the deviation of two given labels $y, y'$. As an example consider a typical loss function used in classification: the 0-1-loss $c_{01} : \mathcal{Y} \times \mathcal{Y} \to \{0,1\}$, which is given by $c_{01}(y, y') = 0$ iff $y = y'$. See [Schölkopf and Smola 2002, Chapter 3] for a more extensive introduction on loss functions.

Although (2.1.1) cannot be minimized directly because the distribution $D$ is unknown the value

$$
R^*_D := \inf_{h \in \mathcal{F}} R(h)
$$

often plays a crucial role. It is called the Bayes risk and gives a lower bound on the best risk we can achieve for the Distribution $D$ using the hypothesis class $\mathcal{F}$. Any hypothesis achieving this minimal risk is called Bayes optimal. The learning problem can therefore be seen as an approximation problem to the Bayes risk.

A usual approach is to minimize the empirical risk functional based on a given sample set $T$:

$$
R_T(h) := \frac{1}{2} \sum_{i=1}^{\ell} c(y_i, h(x_i)).
$$

Under the i.i.d. assumption the law of large numbers ensures that for large samples the empirical risk converges in distribution to the overall risk $R_D$. I.e.

$$
R_T(h) \xrightarrow{D} R_D(h) \quad \text{for any } h \in \mathcal{F}
$$

and the following approach seems to be a solution to the learning task: Given a sample set $T$ simply choose

$$
h_T := \arg \min_{h \in \mathcal{F}} R_T(h).
$$

This has ever since been the heart of inductive reasoning and reflects our belief that explanations which made few errors on the past observations will make few errors on the domain they are drawn from.
The problem, however, is that $h_T$ is not given a priori but in contrary optimally chosen for the given sample set $T$ and therefore biased. This effect is usually called over-fitting in the literature. To circumvent this, we have to ensure in addition that in the limit the minima coincide on the given hypothesis class $\mathcal{F}$, i.e.

$$\inf_{h \in \mathcal{F}} R_T(h) \xrightarrow{\ell \to \infty} \inf_{h \in \mathcal{F}} R_D(h) = R_D^*. $$

The following theorem, which can be considered the main theorem of statistical learning theory gives a sufficient and necessary condition that such convergence takes place.

**Theorem 2.1.1 (Vapnik and Chervonenkis [1991]):** The one sided uniform convergence in probability

$$\lim_{\ell \to \infty} P_D \left\{ T_\ell \mid \sup_{h \in \mathcal{F}} (R_D(h) - R_T(h)) \geq \varepsilon \right\} = 0 $$

for all $\varepsilon > 0$, is a necessary and sufficient condition for nontrivial consistency of empirical risk minimization.

Intuitively, this states that for large data sets the probability that a randomly chosen sample set misleads to an untypical choice of $h_T$ goes to zero. In other words, for any $\delta > 0$ there exists an $\varepsilon$ – depending on the sample size $\ell$, the complexity of the hypothesis class $\mathcal{F}$ and the chosen $\delta$ – such that with probability $1 - \delta$

$$R_D(h) \leq R_T(h) + \varepsilon(\mathcal{F}, \ell, \delta).$$

We are therefore able to bound the overall risk of the computed hypothesis $h_T$ by the sum of the empirical risk plus some complexity term $\varepsilon(\mathcal{F}, \ell, \delta)$. This ensures that for small $\varepsilon$ hypotheses with small empirical error will – with high probability – suffer small loss on the whole domain, they are “probably approximately correct”.

Vapnik and Chervonenkis [1971, 1991] have been the first giving such complexity bounds in term of the so called VC dimension. A notably result is the fact that these bounds do not depend on the distribution $D$. Such bounds are therefore called distribution-free bounds. This means in particular, if ever we find an efficiently computable strategy to derive $h_T$ from a given sample set $T$, such algorithms are PAC-learners in the sense of Valiant [1984].

1 Note that we ignore an import technical detail: if there exists a minimizer in $\hat{h} \in \mathcal{F}$ such that for all $x$ we have $h(x) \geq \hat{h}(x)$ this convergence is trivially fulfilled (see Vapnik [1995] for details).
2.1.2 Large Margin Classifiers

In this section we will be concerned with one of the simplest hypothesis classes – affine linear functions. Consider therefore the case where the input space $\mathcal{X} = \mathcal{H}$ is a linear space, equipped with a scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}} : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$, and assume $\mathcal{Y} \subseteq \mathbb{R}$. Then the class we are interested in is given as follows

$$\mathcal{F} := \{ x \mapsto \langle w, x \rangle_{\mathcal{H}} + b \mid w \in \mathcal{H}, \ b \in \mathbb{R} \}.$$

Let us give a brief introduction to the use of this hypothesis class for classification, where their use in machine learning goes back to Rosenblatt [1958]. Consider therefore that $\mathcal{Y} = \{ \pm 1 \}$. Then any function $(w, b) \in \mathcal{F}$ induces a separation of the input space into two half-spaces by $\text{sgn}(\langle w, x \rangle + b)$, which could potentially be used as classifier for the domain $\mathcal{X}$. Given a sample set $T$ and using the principle of empirical risk minimization we could then aim to find a linear separation $(w, b)$ which separates the given samples “as good as possible”. As $T$ in general need not be linear separable we typically penalize the error a hypothesis $(w, b)$ makes on a data-point $(x_i, y_i)$. In SVM learning this is typically done by the hinge-loss

$$c_\rho(y, h(x)) := \left[ \rho - y_i(\langle w, x \rangle_{\mathcal{H}} + b) \right]^+, \quad \rho > 0$$

where $\rho > 0$ is called margin. In addition, having the above basics of statistical learning theory in mind, we have to control the complexity of this hypothesis class. The next theorem, which is based on VC Theory, gives an answer:

**Theorem 2.1.2** (see Christianini and Shawe-Taylor [2003]): Consider $(w, b) \in \mathcal{F}$ such that $\|w\|_{\mathcal{H}} = 1$. Then there exists a constant $B$ such that for any probability distribution $D$ on $\mathcal{X} \times \{ \pm 1 \}$ with support in radius $R$ around the origin, with probability $1 - \delta$ over $\ell$ random samples $T$, the probability to misclassify a randomly drawn test pattern $(x, y)$ is at most

$$B \ell \left( \frac{R^2 + \|\tilde{\xi}\|^2}{\rho^2} \log^2 \ell + \log \frac{1}{\delta} \right),$$

where $\tilde{\xi} := [\rho - y_i(\langle w, x \rangle_{\mathcal{H}} + b)]^+$ is the margin slack vector for $(w, b)$ on the samples $T$.

This result implies that with growing margin parameter the generalization bound gets better. A promising approach is therefore to compute the classifier $(w, b)$ achieving maximal margin $\rho > 0$ while keeping the empirical error term $\sum_{i=1}^\ell \xi_i$ small. This leads to the concept of maximum margin classifiers. Maximizing $\rho$, while the norm $\|w\|_{\mathcal{H}}$ is fixed, corresponds to minimizing $\|w\|_{\mathcal{H}}$, while $\rho$ is fixed. Consequently the following optimization
problem (2.1.2) is a well founded candidate to compute a separating hyperplane from a sample set $T$. It has first been introduced by [Boser et al. 1992] and is known as $C$-support vector machine ($C$-SVM).

$$
\min_{(w, b)} \frac{1}{2} \|w\|_H^2 + C \sum_{i=1}^\ell \xi_i \quad \text{s.t.} \quad \xi_i \geq 0, \quad \xi_i \geq 1 - y_i (\langle w, x_i \rangle_H + b), \quad (2.1.2)
$$

where $C > 0$ is a regularization constant measuring the trade-off between the two conflicting objectives to choose a large margin and a small empirical error term.

Note that this approach can be justified from a different perspective as well: As already mentioned above there exists a trade-off between choosing our hypothesis from a rich class $F$ to allow a small empirical error term and the choice from a less complex class $F$ keeping the generalization error $\varepsilon(F, \ell, \delta)$ small. The influence of the regularization term $\frac{1}{2} \|w\|_H^2$ can be interpreted as penalty to more “complex” hyperplanes. This technique often effectively prohibits an over-fitting to the given samples $T$. The above objective therefore is a typical example of a regularized risk functional. For a further introduction to the concept of regularization the reader is referred to [Schölkopf and Smola 2002, Chapter 4].

2.1.3 The Kernel Trick

Note that one of the most important properties of large margin bounds, as given in Theorem 2.1.2, is the following: They do not depend on the dimension of the space $H$. This makes SVMs accessible for one of the most powerful techniques in machine learning: the kernel trick, which opens the possibility to embed any input space into a possibly infinite dimensional linear space in which we can then use the maximal margin classifiers strategy. Let us introduce this approach shortly. For further reading see the textbooks by [Schölkopf and Smola 2002] and [Shawe-Taylor and Christianini 2004].

A kernel is a symmetric mapping $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ such that for any $\ell \in \mathbb{N}$ the Gram evaluation matrix

$$(K_{ij})_{1 \leq i,j \leq n} = (k(x_i, x_j))_{1 \leq i,j \leq n} \in \mathbb{R}^{\ell \times \ell}$$

is positive semi-definite for any choice of $x_1, \ldots, x_\ell \in \mathcal{X}$. This functions will be used to define a notion of similarity on the input space $\mathcal{X}$. Typical kernel functions on $\mathcal{X} = \mathbb{R}^n$ are for example Gaussian radial basis kernel $\exp(-\sigma \|x-x'\|_2^2)$, where $\sigma > 0$ denotes the width of the kernel or polynomial kernels like $(x^\top x')^d$. Note that there are various other possibilities of kernel functions even for unstructured domains (see especially the book of [Shawe-Taylor and Christianini 2004] for an overview).

Given such a kernel one can consider the pre-Hilbert space

$$H_0 := \text{span} \{ k(x, \cdot) \mid x \in \mathcal{X} \},$$

where $H_0$ is the linear span of the set of functions $k(x, \cdot)$ for $x \in \mathcal{X}$. This construction is known as the representer theorem, which is a fundamental property of SVMs.
which consists of all finite linear combinations of the evaluation functions
\( k(x, \cdot) : \mathcal{X} \rightarrow \mathbb{R} \), where \( x \in \mathcal{X} \) (for details consult for example the book of Schölkopf and Smola [2002, Chapter 2.2]). For the analysis of the learning performance it is often convenient to consider the topological closure of this space \( \mathcal{H} := \overline{\mathcal{H}_0} \), which is a Hilbert space. If this space has the reproducing property, i.e. for all \( h \in \mathcal{H} \) we have

\[
h(x) = \langle h, k(x, \cdot) \rangle_{\mathcal{H}}
\]

we will call \( \mathcal{H} \) reproducing kernel Hilbert space (RKHS) to the kernel \( k \).

Note that this RKHS is unique in the sense, that the properties of the RKHS completely determine the kernel.

The mapping \( \Phi : \mathcal{X} \rightarrow \mathcal{H} \) such that \( x \mapsto k(x, \cdot) \) is the associated feature map. Note that one can as well motivate kernels the other way round, by considering \( \Phi \) as a mapping from \( \mathcal{X} \) to a linear space \( \mathcal{H} \) and defining \( k(x, x') := \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \), which ensures that \( k \) is a kernel in the above sense.

The key idea in the application of kernel methods in machine learning is the following: Any learning algorithm solely depending on the inner product of some linear space can be applied to any domain where we can define suitable kernels on. This is especially true for the large margin classifiers and SVM presented above. Furthermore this mapping does not impose a computational burden, as evaluating the inner product in the RKHS can be computed simply by evaluating the kernel function.

2.2 Support Vector Machines – Problem Formulations

The combination of the simple hyperplane classifier combined with the kernel trick made the SVM one of the most successful machine learning algorithms. Although the most common formulation of SVM is the \( C \)-SVM given in (2.1.2) there exist a number of different formulations adapted to different tasks and with different theoretical properties.

One of the main drawbacks in the analysis of SVM optimization algorithms has therefore been the fact that they did mostly concentrate on the most popular one – the \( C \)-SVM. In this thesis we aim however to give a broader approach, which subsumes many of the SVM variants available in practice.

To this end we will present a general setting from which we can derive all problems we will be interested in. This admittedly hardens the understanding of any single problem but it allows us to flesh out the common properties of all SVM optimization problems we will consider. For more detailed introductions we recommend the books of Christianini and Shawe-Taylor [2003] or Schölkopf and Smola [2002].
2.2.1 Abstract Regularized Risk Minimization

Given the input space $\mathcal{X}$ and a positive semi-definite kernel. We will then consider the reproducing kernel Hilbert space $\mathcal{H}$ of $k$. In all considered instances of SVM the main aim is to find an affine linear function $h(x) + b$, where $(h, b) \in \mathcal{H} \times \mathbb{R}$, which will later be used as regression function or classifier.

This function is an optimal solution to a regularized empirical risk functional $\frac{1}{2}\|h\|^2_\mathcal{H} + R_T(h, b)$ based on a given sample set $T$. Furthermore we will imply the restriction that (as typical in SVM) the empirical risk stems from a piecewise linear loss function. With the help of slack-variables this leads to the following abstract regularized risk optimization problem:

$$\inf_{(h, b, \rho, \xi)} R(h, b, \rho, \xi) := \frac{1}{2}\|h\|^2_\mathcal{H} + \sum_{i=1}^{\bar{\ell}} \Gamma_i \xi_i - \nu \rho$$

s.t. $0 \leq \xi_i$, $\left(\begin{array}{c} A^H \ A^b \ A^\rho \end{array}\right) \left(\begin{array}{c} h \\ b \\ \rho \end{array}\right) - d \leq \xi_i$, for all $1 \leq i \leq \bar{\ell}$.

(2.2.1)

Before we start with the plan to derive all SVM problems we are interested in from this minimization problem, let us give some comments on its structure:

First we will use a set of trade-off parameters $\Gamma_i$, which we allow to be adaptable to any specific slack-variable $\xi_i$. All in all they give a trade-off between the structural risk $\frac{1}{2}\|h\|^2_\mathcal{H}$ and the empirical risk $\sum_{i=1}^{\bar{\ell}} \xi_i$. Second, mainly with $\nu$-SVM in mind, we allow for a third optimization term $-\nu \rho$, $\nu \in (0, 1]$. There will be two cases – one for classification, where a large margin is rewarded, and one in regression, where a small $\epsilon$-tube is preferred (we will set $\epsilon = -\rho$). Note that we impose no restriction on $\rho$ (see Lemma 2.2.4 and 2.2.8 for a justification). Third, by the assumption that our loss-function is piecewise affine linear, we only have linear constraints. They will be represented by the linear mappings $A^H \in \mathcal{H}^{\ell}$, where $\mathcal{H}'$ denotes the topological dual of $\mathcal{H}$, and $A^b, A^\rho \in \mathbb{R}^{\bar{\ell}}$ and the vector $d \in \mathbb{R}^{\ell}$. $\bar{\ell}$ is a multiple of $\ell$ depending on the number of non-differentiable points of our loss-function (for example in the regression we measure positive and negative deviation of each data-point and therefore have $\bar{\ell} = 2\ell$). Note that by Riesz representation theorem (see for example [Scheja and Storch 1988, Theorem 76.23]) we can identify any $h^* \in \mathcal{H}'$ by a unique $h \in \mathcal{H}$ such that $h^* = \langle h, \cdot \rangle : \mathcal{H} \to \mathbb{R}$. We will therefore in the following simple assume $A^H \in \mathcal{H}^{\ell}$.

In any case, this problem is an instance of GQO (see Problem 1.2.5). This can be seen as follows: We define

$$U := \mathcal{H} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{\bar{\ell}}$$
Chapter 2. Prerequisites from Learning Theory

such that \( u = (h, b, \rho, \xi) \) and set

\[
Q := \text{Id}_H \times 0_{R^{(2+\bar{\ell})\times (2+\bar{\ell})}} \quad \text{and} \quad w := \begin{pmatrix} 0_H \\ 0 \\ \nu \\ -\Gamma \end{pmatrix}.
\]

In addition we choose the linear mapping \( A : U \to \mathbb{R}^{2\bar{\ell}} \) as follows

\[
A = \begin{pmatrix} A^H & A^b & A^\rho & -\text{Id}_{R^{\bar{\ell}}} \\ 0 & 0 & 0 & -\text{Id}_{R^{\bar{\ell}}} \end{pmatrix}
\]

As \( \text{kern} \ Q = \{0_H\} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{\bar{\ell}} \) and the inverse of \( Q \) on \( (\text{kern} \ Q)^\perp \) is given by \( Q^{-1} = \text{Id}_H \) we see that (2.2.2) is an instance of QCO:

\[
\sup_{u \in U} \frac{1}{2} \langle Qu, u \rangle + \langle w, u \rangle \quad \text{s.t.} \quad Au - \begin{pmatrix} d \\ 0_{R^{\bar{\ell}}} \end{pmatrix} \leq 0, \quad (2.2.2)
\]

We will now simply apply the results from Section 1.2.5 to derive the properties of any dual problem of our abstract regularized risk optimization problem. Let us therefore denote with

\[
\lambda = \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{2\bar{\ell}},
\]

the dual variables. From Lemma 1.2.18 we see that the domain of the dual objective \( \psi \) is given by

\[
\text{dom} \ \psi = \{ \lambda \mid w - A^* \lambda \in H \times \{0_{R^{2+\bar{\ell}}}\} \}
\]

which in this case is equivalent to

\[
\text{dom} \ \psi = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \mid \begin{array}{c} -A^b^\top x = 0, \\ \nu + A^\rho^\top = 0, \\ \text{Id} x + \text{Id} y = \Gamma \end{array} \right\}. \quad (2.2.3)
\]

In addition we have for any \( \lambda \in \text{dom} \ \psi \)

\[
U_{\lambda} = U_x := \left\{ x^\top A^H \right\} \times \mathbb{R}^{2+\bar{\ell}} \quad (2.2.4)
\]

and we conclude

\[
\psi(\lambda) = \frac{1}{2} x^\top K x + d^\top x,
\]

where we define

\[
K_{ij} = \langle A^H_i, A^H_j \rangle_H. \quad (2.2.5)
\]
This implies in particular that $K \in \mathbb{R}^{\ell \times \ell}$ is symmetric and positive semi-definite.

Obviously $y$ does not influence the objective. For any dual feasible $(\frac{z}{y}) \geq 0$ we can therefore replace $y_i := \Gamma_i - x_i$ if we bound each $x_i \leq \Gamma_i$. Consequently any dual of an instance of the above introduced regularized risk minimization problem (2.2.1) is of the following type:

**Problem 2.2.1:** Given a set of samples $T \subseteq X \times Y$, a positive semi-definite kernel $k : X^2 \to \mathbb{R}$, a piecewise affine linear loss function and the parameters $\Gamma \geq 0$ and $\nu > 0$. Then the following optimization problem

$$
\inf_x f(x) = \frac{1}{2} x^\top K x + d^\top x \\
\text{s.t.} \quad \begin{pmatrix} -A^b^\top \\ A^\rho^\top \end{pmatrix} x = \begin{pmatrix} 0 \\ \nu \end{pmatrix}, \quad 0 \leq x \leq \Gamma
$$

where $K \in \mathbb{R}^{\ell \times \ell}$ is defined in (2.2.5), is the dual of problem (2.2.2).

For any feasible $x$ of (2.2.6) define $h := x^\top A^H$. Then any primal solution $(h, b, \rho, \xi) \in U_x, b, \rho \in \mathbb{R}$ and $\xi \in \mathbb{R}^\ell$, is a minimizer of the Lagrangian problem (1.2.4) of (2.2.1) at the dual solution $(x, \Gamma - x)$.

A few remarks are in place:

**Remark 2.2.2:**

1) Note that $\{0 \leq x \leq \Gamma\}$ is a compact subset of $\mathbb{R}^\ell$ and therefore the minimal value of (2.2.6) is finite and attained. By duality theory (see Section 1.2; especially Theorem 1.2.16 and 1.2.20) we can as well conclude that therefore any instance of our abstract regularized risk minimization problem (2.2.1) attains an optimal solution as well.

2) Problem 2.2.1 is a special finite instance of QCO optimization, where we have a finite domain and the equality constraints are given as so-called box-constraints. This form has been used in generalizations of SVM algorithms earlier (see List and Simon, 2004, 2007, List, 2007) and will as well be used in this work (see Chapter 3, Problem 3.1.1).

3) We will later see that for many instances of SVM optimization the two vectors $A^b$ and $A^\rho$ consist of $\pm 1$ entries. The equality constraint matrix therefore exhibits a quite simple structure which we will exploit for our convergence proofs.

4) From the last remark in Problem 2.2.1 we see that any optimal function $\hat{h}$ can be written as a finite combination of the $A^H_1, \ldots, A^H_\ell$. As these $A^H_i$ usually consist of kernel evaluations on the data points, this result is a special case of the well known representer theorem (cf. Schölkopf and Smola, 2002, Theorem 4.2).
For completeness we will now state any problem instance for which we will give convergence bounds during this work. Note that all following instances of SVM are well known and have often been derived by standard Lagrangian techniques as used in our arguments as well (see for example [Schölkopf and Smola, 2002, Christianini and Shawe-Taylor [2003, Chen et al. [2005])].

### 2.2.2 C-Support Vector Classification

Let us start with the well known C-SVM [Boser et al., 1992]. The trade-off between structural and empirical risk in this case is driven by a constant $C > 0$, which leads to the following primal problem:

$$
\inf_{(h, b, \xi)} R_C(h, b, \xi) := \frac{1}{2} \|h\|_H^2 + C \sum_{i=1}^\ell \xi_i
$$

subject to:

$$
0 \leq \xi_i, \quad 1 - y_i(h(x_i) + b) \leq \xi_i \text{ for all } 1 \leq i \leq \ell.
$$

This can be cast into a general convex quadratic optimization problem of the form (2.2.2), where $\ell = \ell$. By setting $\Gamma_i = C$, $\nu = 0$,

$$
A^H = (-y_i \Phi(x_i))_{i=1,...,\ell}, \quad A^b = -y, \quad d = -e,
$$

leading to the well known dual representation

$$
\inf_x f_C(x) = \frac{1}{2} x^\top K' x - e^\top x
$$

subject to:

$$
\vec{y}^\top x = 0, \quad 0 \leq x \leq C \cdot e, \quad x \geq 0,
$$

where

$$
K' = \begin{pmatrix}
y_1 y_1 k(x_1, x_1) & \cdots & y_1 y_\ell k(x_1, x_\ell) \\
\vdots & \ddots & \vdots \\
y_\ell y_1 k(x_\ell, x_1) & \cdots & y_\ell y_\ell k(x_\ell, x_\ell)
\end{pmatrix}
$$

is the kernel gram matrix of $k$ on the data samples $x_1, \ldots, x_\ell$ multiplied with the label values $y_i y_j$. Following [Simon 2004] we simply normalize (2.2.8) by setting $x'_i := y_i x_i$ for each $i = 1, \ldots, \ell$. Then solving any instance of the C-support vector classification regularized risk minimization problem (2.2.7) can be transformed to an instance of the following class of dual optimization problems:

**Problem 2.2.3 (C-SVC):** Given a set of $\ell$ samples $T \subseteq X \times \{\pm 1\}$, a positive semi-definite kernel $k : X^2 \to \mathbb{R}$ and a $C > 0$. Then the following optimization problem

$$
\min_{x \in \mathbb{R}^\ell} f_C(x) := \frac{1}{2} x^\top K x - \vec{y}^\top x \quad \text{s.t.} \quad e^\top x = 0, \quad l_i \leq x_i \leq u_i,
$$

(2.2.9)
where \( K = \left( k(x_i, x_j) \right)_{1 \leq i, j \leq \ell} \) and
\[
l_i := \begin{cases} 0 & y_i = 1 \\ -C & y_i = -1 \end{cases} \quad \text{and} \quad u_i := \begin{cases} C & y_i = 1 \\ 0 & y_i = -1 \end{cases}
\]
is an instance of \( C \)-support vector classification (\( C \)-SVC) and will be denoted by \( P_C = P_C(k, \tilde{x}, \tilde{y}, C) \).

For any feasible \( x \in \mathcal{P}(P_C) \) we can consider the following associated primal solution \( (h, b, \xi) \in U_x \) given by
\[
h \colon= \sum_{i=1}^{\ell} x_i \Phi(x_i), \quad b \in \mathbb{R} \quad \text{and} \quad \xi \in \mathbb{R}^{\ell},
\]
where we will use \( x \mapsto \text{sgn}(h(x) + b) \) as classification function. We will later motivate a special choice of \( b \) (see Corollary 4.3.2 and especially Theorem 4.3.7).

### 2.2.3 \( \nu \)-Support Vector Classification

We will now pursue the same line of arguments for the \( \nu \)-SVM introduced by Schölkopf et al. [2000]. The \( \nu \)-SVM is a variant of SVM, which instead of using the trade-off parameter \( C \) lets the machine itself adapt the margin \( \rho \) to the given data depending on a new parameter \( \nu \in (0, 1] \). As the authors showed, the parameter \( \nu \) in this case gives an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors of the optimal solution (see [Schölkopf et al., 2000] for details). This ensures that the new parameter \( \nu \) is more meaningful than the old parameter \( C \). The proposed primal optimization problem of \( \nu \)-support vector classification is given as follows:

\[
\inf_{(h, b, \rho, \xi)} R_\nu(h, b, \rho, \xi) := \frac{1}{2} \|h\|^2 + \frac{1}{\ell} \sum_{i=1}^{\ell} \xi_i - \nu \rho \\
\text{s.t.} \quad 0 \leq \xi_i, \quad \rho - y_i(h(x_i) + b) \leq \xi_i, \quad \text{for all } 1 \leq i \leq \ell \\
0 \leq \rho
\]

Let us first show that the restriction \( \rho \geq 0 \) is not needed:

**Lemma 2.2.4:** Given \( (h, b, \rho, \xi) \in \mathcal{H} \times \mathbb{R}^{2+\ell} \) such that \( 0 \leq \xi_i \) and \( \rho - y_i(h(x_i) + b) \leq \xi_i \), for all \( i \in \{1, \ldots, \ell\} \), but \( \rho < 0 \). Then there exists a \( t \in (0, 1] \) such that \( (h', b', 0, \xi') = (th, tb, 0, \xi - t\rho) \) is a feasible point of (2.2.10) such that
\[
R_\nu(h', b', 0, \xi') < R_\nu(h, b, \rho, \xi).
\]

**Proof.** Let us first show that \( (h', b', 0, \xi') \) is feasible for all choices of \( t \in (0, 1] \). As \( \rho < 0 \) we have \( \xi'_i \geq \xi_i \geq 0 \) for all \( i \in \{1, \ldots, \ell\} \). To prove the second inequality, note that \( \text{sgn}(\rho - y_i(h(x_i) + b)) = \text{sgn}(t\rho - y_i(h'(x_i) + b')) \). For any \( \xi_i \geq 0 \) we therefore have
\[
t\rho - y_i(h'(x_i) + b') \leq \xi_i,
\]
which ensures \(0 - y_i(h'(x_i) + b') \leq \xi_i'\) for all \(i \in \{1, \ldots, \ell\}\).

Now consider the continuous auxiliary function

\[
g(t) := R_\nu(h', b', 0, \xi') - R_\nu(h, b, \rho, \xi) = \frac{(t^2 - 1)}{2} \|h\|_H^2 - \left(\frac{t}{\ell} - \nu\right) \rho.
\]

We easily see that \(g(0) = -\frac{1}{2} \|h\|_H^2 + \nu \rho < 0\). Therefore there exists a \(t > 0\) such that \(g(t) < 0\) and consequently \((h', b', 0, \xi')\) achieves an even better value of \(R_\nu\) than \((h, b, \rho, \xi)\).

**Remark 2.2.5:** Note that this fact has earlier been reflected in the possibility to transform the dual inequality constraint \(e^\top x \geq \nu\) to an equality constraint \(e^\top x = \nu\) (see below). This has first been observed by [Chen et al. 2005](#). The proof given here however is more informative and allows us to view 2.2.10 as a special instance of (2.2.1).

**Lemma 2.2.4** implies that we can ignore the inequality constraint \(\rho \geq 0\) and therefore 2.2.10 is an instance of (2.2.2) if we set \(\Gamma_i = \frac{1}{\ell}, d = 0, A^H = (-y_i \Phi(x_i))_{i=1,\ldots,\ell}, A^b = -\bar{y}\) and \(A^\rho = e\).

We conclude that solving an instance of the \(\nu\)-support vector classification regularized risk minimization problem 2.2.10 can be transformed to an instance of the following class of dual optimization problems:

**Problem 2.2.6 (\(\nu\)-SVC):** Given a set of \(\ell\) samples \(T \subseteq \mathcal{X} \times \{-1, 1\}\), a positive semi-definite kernel \(k : \mathcal{X}^2 \to \mathbb{R}\) and a \(\nu \in (0,1]\). Then the following optimization problem

\[
\min_{x \in \mathbb{R}^\ell} f_\nu(x) = \frac{1}{2} x^\top K' x \quad \text{s.t.} \quad e^\top x = \nu, \quad \bar{y}^\top x = 0, \quad 0 \leq x_i \leq \frac{1}{\ell}, \quad (2.2.11)
\]

where \(K' = (y_i y_j k(x_i, x_j))_{1 \leq i, j \leq \ell}\) is an instance of \(\nu\)-support vector classification (\(\nu\)-SVC) and will be denoted by \(\mathcal{P}_\nu = \mathcal{P}_\nu(k, \bar{x}, \bar{y}, \nu)\).

For any feasible \(x \in \mathcal{R}(\mathcal{P}_\nu)\) we can consider the following associated primal solution \((h, b, \rho, \xi) \in U_\mathcal{P}\) given by \(h := \sum_{i=1}^\ell y_i x_i \Phi(x_i), b, \rho \in \mathbb{R}\) and \(\xi \in \mathbb{R}^\ell\), where we will use \(x \mapsto \text{sgn}(h(x) + b)\) as classifier (see Theorem 4.3.7 how to choose \(b\)).

### 2.2.4 \(\epsilon\)-Support Vector Regression

We will now present an approach to solve regression problems with the help of C-SVM (cf. [Vapnik 1995](#)). The key idea is to measure the error a regression estimator suffers by a so called \(\epsilon\)-insensitive loss function as follows:

\[
|y - (h(x) + b)|_\epsilon := \max\{0, |y - (h(x) + b)| - \epsilon\},
\]
which does not count deviations which are lying in a tube of $2\epsilon$ around the data points. Given again a trade-off parameter $C > 0$ between structural and empirical risk and a width $\epsilon > 0$ of the tube, this leads to the following primal optimization problem:

$$\inf \left\{ R_C(h, b, \xi) := \frac{1}{2} \|h\|^2_H + C \sum_{i=1}^{2\ell} \xi_i \right\}$$

$$\text{s.t. } 0 \leq \xi_i, 1 \leq i \leq 2\ell$$

$$(h(x_i) + b) - y_i - \epsilon \leq \xi_i, 1 \leq i \leq \ell$$

$$y_i - (h(x_i) + b) - \epsilon \leq \xi_i, 1 \leq i \leq \ell.$$  

This can be cast into a general convex quadratic optimization problem of the form (2.2.2), where $\bar{\ell} = 2\ell$, $\Gamma_i = C\ell$, $\nu = 0$,

$$A^H = \begin{pmatrix} \Phi(x_1) \\ \vdots \\ \Phi(x_\ell) \\ -\Phi(x_1) \\ \vdots \\ -\Phi(x_\ell) \end{pmatrix}^\top, \quad A^b = \begin{pmatrix} e \\ -e \end{pmatrix} \quad \text{and} \quad d = \begin{pmatrix} \bar{y} \\ -\bar{y} \end{pmatrix} + \epsilon \cdot e.$$

With the same techniques as above, we see that solving an instance of (2.2.12) is equivalent to solving an instance of the following class of problems:

**Problem 2.2.7 ($\varepsilon$-SVR):** Given a set of $\ell$ samples $T \subseteq \mathcal{X} \times \mathbb{R}$, a positive definite kernel $k : \mathcal{X}^2 \to \mathbb{R}$, a $C > 0$ and an $\epsilon > 0$. Then the following optimization problem

$$\min_{x \in \mathbb{R}^{2\ell}} f_{C\varepsilon}(x) := \frac{1}{2} x^\top \begin{pmatrix} K & -K \\ -K & K \end{pmatrix} x + \begin{pmatrix} \bar{y} + \epsilon \cdot e \\ -\bar{y} + \epsilon \cdot e \end{pmatrix}^\top x$$

$$\text{s.t. } (-e^\top \ -e^\top) x = 0, \quad 0 \leq x_i \leq C \ell,$$

where $K = (k(x_i, x_j))_{1 \leq i,j \leq \ell}$ is an instance of $\varepsilon$-support vector regression ($\varepsilon$-SVR) and will be denoted by $\mathcal{P}_{\varepsilon} = \mathcal{P}_{\varepsilon}(k, \bar{x}, \bar{y}, C, \epsilon)$.

For any feasible $x \in \mathcal{R}(\mathcal{P}_{\varepsilon})$ we can consider the following associated primal solution $(h, b, \xi) \in U_x$ given by $h := \sum_{i=1}^{\ell}(x_i - x_{\ell+i})\Phi(x_i)$, $b \in \mathbb{R}$ and $\xi \in \mathbb{R}^{2\ell}$, where we use $x \mapsto h(x) + b$ as regression function (see Theorem 4.3.7 how to choose $b$).

### 2.2.5 $\nu$-Support Vector Regression

Consider the regression formulation of $\nu$-SVM introduced by Schölkopf et al. [2000]. Again this formulation tries to let the learning machine itself optimize
Chapter 2. Prerequisites from Learning Theory

the optimal ε-tube, instead of fixing it as before. The resulting regularized risk minimization problem is at follows:

\[
\inf_{(h, b, \epsilon, \xi)} R_{\nu\epsilon}(h, b, \epsilon, \xi) := \frac{1}{2} \|h\|^2_H + C \left( \frac{1}{\ell} \sum_{i=1}^{2\ell} \xi_i + \nu \epsilon \right)
\]

s.t. \[0 \leq \epsilon, \ 0 \leq \xi_i, \ 1 \leq i \leq 2\ell
\]
\[(h(x_i) + b) - y_i - \epsilon \leq \xi_i, \ 1 \leq i \leq \ell
\]
\[y_i - (h(x_i) + b) - \epsilon \leq \xi_i, \ 1 \leq i \leq \ell.
\]

Again note that the restriction on the ε-tube is not necessary:

**Lemma 2.2.8:** Given \((h, b, \epsilon, \xi) \in H \times \mathbb{R}^{2+2\ell}\) such that \(0 \leq \xi_i, (h(x_i) + b) - y_i \leq \xi_i \) for all \(i \in \{1, \ldots, \ell\}\) and \(y_i -(h(x_i) + b) \leq \xi_i \) for all \(i \in \{\ell+1, \ldots, 2\ell\}\) but \(\epsilon < 0\). Then there exists a feasible point \((h, b, 0, \xi')\) of (2.2.14), with

\[R_{\nu\epsilon}(h, b, 0, \xi') \leq R_{\nu\epsilon}(h, b, \epsilon, \xi).
\]

**Proof.** If \(\xi_i + \epsilon \geq 0\) and \(\xi_{\ell+i} + \epsilon \geq 0\) we define \(\xi'_i := \xi_i + \epsilon\) and \(\xi'_{\ell+i} := \xi_{\ell+i} + \epsilon\) and the feasibility for \(\ell' = 0\) follows at once and we obviously have \(\xi'_i + \xi'_{\ell+i} < \xi_i + \xi_{\ell+i} + \epsilon\).

Now note that, for any \(i \in \{1, \ldots, \ell\}\), we can conclude from the two inequalities for \(\xi_i\) and \(\xi_{\ell+i}\) that \(0 \leq (\xi_i + \epsilon) + (\xi_{\ell+i} + \epsilon)\) and therefore either \(\xi_i + \epsilon < 0\) or \(\xi_{\ell+i} + \epsilon < 0\). Without loss of generality let us assume that \(\xi_i + \epsilon < 0\) and define: \(\xi'_i := 0\) and \(\xi'_{\ell+i} := y_i - (h(x_i) + b)\). This implies

\[(h(x_i) + b) - y_i - 0 \leq \xi'_i + \epsilon < 0 = \xi'_i,
\]
\[0 \leq -\xi'_i - \epsilon \leq y_i - (h(x_i) + b) = \xi'_{\ell+i},
\]
and
\[\xi'_i + \xi'_{\ell+i} \leq \xi_i + \xi_{\ell+i} + \epsilon.
\]

We conclude that \((h, b, 0, \xi')\) is feasible and

\[R_{\nu\epsilon}(h, b, 0, \xi') = \frac{1}{2} \|h\|^2_H + C \left( \frac{1}{\ell} \sum_{i=1}^{2\ell} \xi'_i \right) \leq \frac{1}{2} \|h\|^2_H + C \left( \frac{1}{\ell} \sum_{i=1}^{2\ell} \xi_i + \nu \epsilon \right)
\]

\[\leq C \left( \frac{1}{\ell} \sum_{i=1}^{2\ell} \xi_i + \nu \epsilon \right) = R_{\nu\epsilon}(h, b, \epsilon, \xi),
\]

where the last inequality follows from \(\epsilon \leq \nu \epsilon < 0\). \(\square\)

Note that this has as well its correspondence in a possible simplification of the standard dual of (2.2.14) as for example given in [List 2007]. Yet, by Lemma 2.2.8 we see that the primal (2.2.14) can already be seen as an instance
of our abstract regularized risk minimization (2.2.1). Let us therefore use the variable substitution \( \rho = -\epsilon \) and the parameters \( \bar{\ell} = 2\ell, \Gamma_i = \ell C \),

\[
A^H = \begin{pmatrix}
\Phi(x_1) \\
\vdots \\
\Phi(x_\ell) \\
-\Phi(x_1) \\
\vdots \\
-\Phi(x_\ell)
\end{pmatrix},
A^b = \begin{pmatrix} e \\ -e \end{pmatrix},
A^\rho = \frac{1}{C}\cdot e \quad \text{and} \quad d = \begin{pmatrix} \bar{y} \\ -\bar{y} \end{pmatrix}.
\]

Again we conclude that solving an instance of the \( \nu \)-support vector regression regularized risk minimization problem (2.2.14) is equivalent to solve an instance of the following class of problems:

**Problem 2.2.9 \((\nu\text{-SVR})\):** Given a set of \( \ell \) samples \( T \subseteq X \times \mathbb{R} \), a a positive definite kernel \( k : X^2 \to \mathbb{R} \), a \( C > 0 \) and a \( \nu \in (0,1] \). Then the following optimization problem

\[
\min_{x \in \mathbb{R}^{2\ell}} f_{\nu r}(x) := \frac{1}{2}x^\top \begin{pmatrix} K & -K \\ -K & K \end{pmatrix} x + \begin{pmatrix} -\bar{y} \\ \bar{y} \end{pmatrix}^\top x
\]

\[
s.t. \quad \begin{pmatrix} -e^\top \\ e^\top \end{pmatrix}^\top x = \begin{pmatrix} 0 \\ C\nu \end{pmatrix}, \quad 0 \leq x_i \leq \frac{C}{\ell},
\]

where \( K = (k(x_i, x_j))_{1 \leq i,j \leq \ell} \) is an instance of \( \nu \)-support vector regression \((\nu\text{-SVR})\) and will be denoted by \( P_{\nu r} = P_{\nu r}(k, \bar{x}, \bar{y}, C, \nu) \).

For any feasible \( x \in \mathcal{R}(P_{\nu r}) \) we can consider the following associated primal solution \((h, b, \rho, \xi) \in U_x\) given by \( h := \sum_{i=1}^{\ell}(x_i - x_{i+1})\Phi(x_i), \rho, b \in \mathbb{R} \) and \( \xi \in \mathbb{R}^{2\ell} \), where we will use \( x \mapsto h(x) + b \) as regression function (see Theorem 4.3.7 how to choose \( b \)).

### 2.2.6 Other Variants of SVM

The generality of the above arguments allows to apply the techniques presented to other variants of SVM. For example 1D-SVMs (cf. Schölkopf and Smola, 2002, Chapter 8) or DLD-SVM by Steinwart et al., 2005. Both types of learning algorithms are design for novelty detection tasks.

Let us however point the interest in a different direction. One could think of allowing \( Y \) to be a richer label space and use the large margin techniques directly to train for example multi-class problems. This is done for example in Szedmak et al., 2006. As long a piecewise linear loss is used, we can adapt our approach to this setting. Remember therefore that there is no need to have \( A^b \in \mathbb{R}^\ell \). We are free to consider inequalities \( A^b \in \mathbb{R}^{m \times \ell} \) accounting for a piecewise loss for an \( m \) dimensional label space. Note however, that in this generalized setting we cannot expect the dual problems to have the
simple structure as we encounter in the above mentioned problem instances. We will see that this fact possibly renders simple decomposition algorithms as SMO useless.
Chapter 3

Decomposition Techniques for Quadratic Programming

The following chapter will contain the main theoretical results. We will present a general theory for decomposition techniques for quadratic programming tailored to the analysis of SVM optimization problems.

Section 3.1 will define the class of problems we will be interested in and give elementary properties. Section 3.1.1 will give a precise definition of decomposition algorithms, which will slightly differ from the usual notion. We will, however, show that both are simply two viewpoints on the same procedure.

The rest of the chapter is divided in two main parts: One is concerned with so-called aggressive selection strategies and can be found in Section 3.2. The second one with conservative selection strategies and can be found in Section 3.3. Both parts exhibit the same structure, however the most work will be done in Section 3.2 where the main concepts are introduced during the development of the theory.

Although we try to introduce the theory as general as possible, please keep in mind that the main focus of this work will be the application of the achieved results to SVM optimization. A detailed account of this task is postponed to Chapter 5.

3.1 Introduction

Remember the dual of the abstract regularized risk minimization problem (2.2.1) defined in Chapter 2. We see that any instance of (2.2.6) is an instance of the following class of problem:
Problem 3.1.1 (Boxed Quadratic Optimization): Given a convex quadratic function \( f : \mathbb{R}^\ell \to \mathbb{R} \)
\[
f(x) := \frac{1}{2} x^\top Q x - w^\top x,
\]
where \( Q \in \mathbb{R}^{\ell \times \ell} \) is symmetric and positive semi-definite and \( w \in \mathbb{R}^\ell \), an equality constraint matrix \( A \in \mathbb{R}^{k \times \ell} \), \( a, b \in \mathbb{R}^k \) and two border vectors \( l, u \in \mathbb{R}^\ell \), such that \( l < u \).

We call any problem \( \mathcal{P}_{QP}(f, A, b, l, u) = \mathcal{P}_{QP}(Q, w, A, b, l, u) \)
\[
\min_{x \in \mathbb{R}^\ell} f(x) \quad \text{s.t.} \ Ax = b, \ l \leq x \leq u \tag{3.1.1}
\]
an instance of Boxed Quadratic Optimization (BQO).

Remark 3.1.2: 1) As \( Q \) is positive semi-definite, \( f \) is obviously convex and therefore any instance of BQO is an instance of QCO. The main restriction is the fact that its dimension is finite as any finite-dimensional instance of QCO, with bounded feasibility region, can be brought in the above form, using slack-variables.

2) As the feasibility region \( \mathcal{R}(\mathcal{P}_{QP}) \) is compact the minimum value is finite and attained.

3) For simplicity we will often assume that the equality constrained matrix \( A \) is of full row rank and therefore \( k \leq \ell \). This can be achieved without loss of generality by simply deleting redundant constraints.

4) As the equality constraints \( Ax = b \) restrict the problem to an affine linear subspace of \( \mathbb{R}^\ell \) we will sometimes use the more general notation \( \mathcal{P}_{QP}(f, U, l, r) \) where \( U = \{ x \mid Ax = b \} \) for some appropriately chosen \( A \in \mathbb{R}^{k \times \ell} \) and \( b \in \mathbb{R}^k \).

Let us introduce a helpful notation:

Definition 3.1.3: Given an instance \( \mathcal{P}_{QP}(f, A, b, l, u) \) of BQO. For any \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) the set of free directions at \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) is defined as
\[
F(x) := F_{\mathcal{P}_{QP}}(x) := \{ d \in \ker A \mid x_i = u_i \Rightarrow d_i \leq 0 \land x_i = l_i \Rightarrow d_i \geq 0 \} \subseteq \mathbb{R}^\ell. \tag{3.1.2}
\]

With the help of this notation we can give the following optimality conditions for any instance \( \mathcal{P}_{QP} \) of BQO:

Lemma 3.1.4: Given an instance \( \mathcal{P}_{QP} \) of BQO. Then \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) is optimal iff for all \( d \in F(x) \) we have \( \nabla f(x)^\top d \geq 0 \).
Proof. By first order optimality condition for convex optimization [Bertsekas, 1995, Proposition 2.1.2] it is sufficient to prove that
\[ \forall x' \in \mathcal{R}(P_{QP}) : \nabla f(x')^\top (x' - x) \geq 0 \iff \forall d \in F(x) : \nabla f(x)^\top d \geq 0. \]
Obviously for any \( x' \in \mathcal{R}(P_{QP}) \) we have \( x - x' \in F(x) \) and one direction follows. On the other hand, assume that there exist a \( d \in F(x) \), such that \( \nabla f(x)^\top d < 0 \). Now note that as \( d \in \ker A \) and \( l < u \) there exists a \( t > 0 \), such that \( x' := x + td \in \mathcal{R}(P_{QP}) \), such that \( f(x') < f(x) \), which concludes the proof.

Let us give an important remark first:

Remark 3.1.5: Note that \( F(x) \) is a cone containing the origin. The dimension of the affine linear hull of this cone is at most \( \ell - k = \dim \ker A \). Note that its edges are therefore given by fixing \( \ell - k - 1 \) independent box constraints. As the constraints are aligned with the coordinate system, this corresponds exactly to \( k + 1 \) sparse directions (see Definition 3.1.6). As the cone is generated by this rays we can conclude from Lemma 3.1.4 that for any suboptimal \( x \) there exists a \( k + 1 \) sparse descent direction. Furthermore we will see, that typical decomposition algorithms walk along such rays which are parallel to the edges of the polyhedral feasibility region \( \mathcal{R}(P_{QP}) \). This circumstance has first been exploited by List and Simon [2005, 2007] in the context of SVM optimization algorithms. See Theorem 3.2.7 and Theorem 3.3.5 for details.

3.1.1 Decomposition Algorithms

In contrast to the usual notion of decomposition algorithms in SVM optimization (see the Introduction), we will present these algorithms as a special form of sparse descent line search algorithms. We will however – in accordance with tradition in SVM optimization – stick to the slightly misleading term “decomposition algorithm” and remind the reader, that we don’t have Domain decomposition algorithms in mind.

Consider the following definition:

Definition 3.1.6 (\( q \)-sparse direction): Any \( d \in \mathbb{R}^\ell \), such that
\[ |\{ i \mid d_i \neq 0 \}| \leq q \in \mathbb{N}, \]
is called a \( q \)-sparse direction.

For any instance \( P_{QP} \) of BQO, and any \( x \in \mathcal{R}(P_{QP}) \) we will call \( d \in F(x) \) a \( q \)-sparse descent direction at \( x \) iff \( \nabla f(x)^\top d < 0 \).

Remember the class of descent line search algorithms (see Algorithm 1.1). If ever we select “sparse” directions in each iteration, we will call such an algorithm decomposition algorithm. To be more precise:
Definition 3.1.7 \(\tilde{q}\)-sparse decomposition algorithm): Given a function \(\tilde{q} : BQO \rightarrow \mathbb{N}\).

Any descent line search algorithm \(A\) for BQO (see Algorithm 3.1) is called \(\tilde{q}\)-sparse decomposition algorithm for BQO iff for any instance \(P_{QP}\) from BQO we have for all \(n \geq 0\) that the selected direction \(d^{(n)}\) is \(\tilde{q}(P_{QP})\)-sparse descent direction for \(x^{(n)}\).

Algorithm 3.1: Descent line search for BQO

Input: \(P_{QP} \in BQO, \varepsilon > 0\)

Output: \(\hat{x} \in \mathbb{R} (P_{QP})\) such that \(\Delta(\hat{x}) \leq \varepsilon\)

1 while \(\Delta^{(n)} > \varepsilon\) do
2 \hspace{1em} Select \(d^{(n)} \in F(x^{(n)})\), where \(\nabla f(x^{(n)})^\top d^{(n)} < 0\);
3 \hspace{1em} \(x^{(n+1)} \leftarrow \text{opt}_{P_{QP}}(f, \{x^{(n)}\} + \mathbb{R} \cdot d^{(n)}, l, u)\);
4 \hspace{1em} \(n \leftarrow n + 1\);
5 return \(x^{(n)}\)

Let us give some remarks on this definition:

Remark 3.1.8: 1) Note that Definition 3.1.7 does not guarantee asymptotic convergence of the sequence produced by Algorithm 3.1 and therefore no finite termination is guaranteed. We will later show that for careful choice of the selection procedure and stopping criteria this finite termination can be proven (see for example Theorem 5.2.1 or Theorem 5.2.7).

2) As long as \(\Delta^{(n)} > \varepsilon\), \(x^{(n)}\) is suboptimal, Remark 3.1.5 implies that we find a descent direction \(d^{(n)} \in F(x^{(n)})\), such that \(|\{i \mid d_{i}^{(n)} \neq 0\}| \leq k + 1\) (see as well Theorem 3.2.7 and Theorem 3.3.5). This ensures that, for \(\tilde{q}(P_{QP}) = k + 1\), there exist \(\tilde{q}\)-sparse decomposition algorithms.

3) Note that if the selected \(d^{(n)}\) are sparse the inner iteration of the decomposition algorithm is substantially simpler in the following sense: To (re-) calculate the gradient for the updated \(x^{(n+1)}\) one needs to evaluate each row (column) of the matrix \(Q\) for which we have \(d_{i} \neq 0\) (see Algorithm 5.1 for the extreme case of \(q = 2\)). If the set of this indexes is small we have to evaluate fewer of these matrix rows (columns). We will later on see (Chapter 5) that the runtime of such algorithms in SVM optimization heavily depends on the sparsity of the selected directions.

4) As already described for the standard descent line search algorithms (see Section 1.3) the crucial point is the selection of \(d^{(n)}\). In the case of decomposition methods this not only dictates the convergence properties of \(A\) but as well the sparsity of such a procedure.
Connection to the Standard Notion of Decomposition Algorithms

Before we start to work in this setting let us give a few arguments that our definition of $\vec{q}$-sparse decomposition algorithms does not differ substantially from the classical notion (see Introduction).

First of all note that given an $x \in \mathbb{R}(P_{QP})$ the notion of $q$-sparse descent direction at $x$ is tightly correlated to the notion of $q$-sparse witness of suboptimality introduced by List and Simon [2004, 2007]. Let us first repeat (an adapted) definition given in List [2007]:

**Definition 3.1.9:** Given an instance $P_{QP}$ of BQO. Define

$$U_{I,z} := \{ x \mid Ax = b \land \forall i \notin I : x_i = z_i \}$$

to be the subset of $\{ Ax = b \}$, which differs from a given $z \in \mathbb{R}(P_{QP})$ only in indexes in $I$. Then let $C_I : \mathbb{R}(P_{QP}) \rightarrow \mathbb{R}^+$, $I \subseteq \{1,\ldots,\ell\}$ be a family of functions and consider the three following properties:

(C1) For each $I \subseteq \{1,\ldots,\ell\}$ such that $|I| \leq q$, $C_I(x)$ is continuous on $\mathbb{R}(P_{QP})$.

(C2) If $|I| \leq q$ and $x$ is an optimal solution of $P_{QP}(f,U_I,x,l,r)$ then $C_I(x) = 0$.

(C3) If $x$ is not an optimal solution for $\mathbb{R}(P_{QP})$, then there exists an $I \subseteq \{1,\ldots,\ell\}$ such that $|I| \leq q$ and $C_I(x) > 0$.

Any family of functions satisfying conditions (C2) and (C3) will be called a $q$-sparse witness of suboptimality. If $(C_I(x))$ fulfills (C1) in addition we call it a continuous $q$-sparse witness of suboptimality.

**Theorem 3.1.10:** Given an instance $P_{QP}$ of BQO and a fixed $q \in \{1,\ldots,\ell\}$. Then the following two statements are equivalent:

1) There exists a $q$-sparse witness of suboptimality.

2) For any suboptimal $x \in \mathbb{R}(P_{QP})$ there exists a $q$-sparse descent direction $d \in F(x)$.

**Proof.** Let us first assume that, given an instance $P_{QP}$ of BQO, there exists a $q$-sparse witness of suboptimality $C_I : \mathbb{R}(P_{QP}) \rightarrow \mathbb{R}^+$. Now, given a suboptimal $x \in \mathbb{R}(P_{QP})$ we conclude from (C3) that there exists a set $I \subseteq \{1,\ldots,\ell\}$, such that $|I| \leq q$ and $C_I(x) > 0$. With (C2) we therefore know that $x$ is not optimal for $P_{QP}(f,U_I,x,l,r)$. Now Lemma 3.1.4 implies,

---

1 Note, this definition differs slightly from the use in [List and Simon 2004], where the continuity property (C1) is crucial for the proof of asymptotic convergence. The authors therefore only considered as sparse witness what we call continuous sparse witness of suboptimality.
that there exists at least one \( d \in F_{\mathcal{P}_Q P}(f,U_I,x,l,r) \) such that \( \nabla f(x)^T d < 0 \).

As for such a \( d \) we have \( x + d \in U_I \) and therefore \( \{ i \mid d_i \neq 0 \} \subseteq I \), it follows that it is \( q \)-sparse. As \( d \in F_{\mathcal{P}_Q P}(x) \) as well, it is a \( q \)-sparse descent direction for \( \mathcal{P}_Q P \) at \( x \).

For the opposite direction assume that for any suboptimal \( x \in \mathcal{R}(\mathcal{P}_Q P) \) there exists a \( q \)-sparse descent direction and define:

\[
C_I(x) := \begin{cases} 
0 & \text{if } x \text{ is optimal for } \mathcal{P}_Q P(f,U_I,x,l,r) \\
1 & \text{otherwise}
\end{cases}
\]

Then (C2) holds by definition. To see that (C3) holds as well, simply note that given a suboptimal \( x \) and the appropriate \( q \) sparse direction \( d \in F(x) \) simply set \( I := \{ i \mid d_i \neq 0 \} \), such that by construction \( |I| \leq q \). As obviously \( d \in F_{\mathcal{P}_Q P}(f,U_I,x,l,r)(x) \), \( x \) is not optimal for \( \mathcal{P}_Q P(f,U_I,x,l,r) \) and \( C_I(x) > 0 \).

We conclude that \( C_I(x) \) as defined above is a \( q \)-sparse witness of suboptimality for \( \mathcal{P}_Q P \).

Furthermore the selection of a working set \( I \subseteq \{1,\ldots,\ell\} \) and the selection of a descent direction are equivalent in the following sense:

For any working set \( I \subseteq \{1,\ldots,\ell\} \) one can choose a direction \( d^{(I)} := x' - x \in F(x) \), where \( x' = \text{opt} \mathcal{P}_Q P(f,U_I,x,l,r) \). This selection leads to exactly the same sequence \( (x^{(n)})_{n \geq 0} \). We will later see that for certifying pair/set algorithms, which are the only class of decomposition algorithms for which convergence rates have been proven, the means of guaranteeing an advance is exactly the existence of a direction in \( U_I \), which guarantees this advance (compare Section 3.3, especially Lemma 3.3.3 and Theorem 3.3.7).

On the other hand we will see that the most selection procedures implemented in SVM optimization algorithms choose the working set \( I \) in each iteration minimal, such that optimization space \( U_I \) is of dimension 1 (see Remark 3.2.16 and 3.3.6). In fact it is a direction parallel to an edge of the feasibility region (see Remark 3.1.5). This is the main reason why our bounds, derived for the descent line search view on decomposition algorithms, hold for many existing algorithms, which have been motivated by selecting subsets of \( \{1,\ldots,\ell\} \) as working set. Note that there is one prominent exception: SVMlight, which is to our knowledge the only algorithm using a larger working set, such that \( \dim U_I > 1 \). See Remark 3.2.51 for details on such cases.

**Design Principles of Selection Strategies**

In the following we will design general selection strategies, which try to optimize three objectives:

1) Any selected direction should promise a “large advance”,

2) the selected directions should be “as sparse as possible” and last but not least,

3) the selection of the direction should be efficiently computable.

Let us discuss them briefly: First of all it is natural to request a that a selected direction leads to a sufficient advance. Possibly per iteration, at least over a longer period. The second objective has already been mentioned in Remark 3.1.8. The update during one decomposition iteration can be reduced for sparse problems (see Algorithm 5.1 for details). It is worth to note that the guarantees we will give on the advance partly scale in the sparsity of the direction as well (see for example Lemma 3.3.3). The last objective is obvious: The selection of the direction has to be done in any iteration. As an update in typical SVM algorithms takes $O(\ell)$ (see Lemma 5.1.1) we aim for linear time selection strategies as well. In fact this has been one of the main design principles in working set selection.

Although we will try to fulfill these objectives as general as possible, we will later see that for the special case of SVM optimization they specialize to well known selection strategies implemented in standard tools.

Remember the classification of SVM optimization algorithms we gave in the Introduction: The first ones have been called conservative, as they aim for giving a lower bound on advance in each iteration achievable inside the feasible region. The second ones have been called aggressive, as they try to aim for the largest improvement on the unbounded problem accepting the fact that the advance in the next iteration may be infinitesimally small by hitting the border. We will formalize these concepts in the following definition:

**Definition 3.1.11:** Given an instance $\mathcal{P}_{QP}$ of BQO. Then for any feasible $x \in \mathcal{R}(\mathcal{P}_{QP})$ and any direction $d \in F(x)$ we call

$$\delta_C^d(x) := \max \{ f(x) - f(x + td) \mid t > 0, x + td \in \mathcal{R}(\mathcal{P}_{QP}) \} \in \mathbb{R}$$

the expected conservative advance along $d$.

$$\delta_d(x) := \sup \{ f(x) - f(x + td) \mid t > 0 \} \in \mathbb{R}$$

is called expected aggressive advance along $d$.

Note that the maximum in the first case is finite and attained, as $\mathcal{R}(\mathcal{P}_{QP})$ is bounded by definition of $\mathcal{P}_{QP}$, while it may be infinite in the aggressive case. Let us state some simple properties of the defined quantities:

**Lemma 3.1.12:** Given an instance $\mathcal{P}_{QP}$ of BQO. Then for any $x \in \mathcal{R}(\mathcal{P}_{QP})$ and any $d \in F(x)$, $x$ is optimal for $\mathcal{P}_{QP}(f, \{ x \} + \mathbb{R} \cdot d, l, u)$ iff $\delta_C^d(x) = 0$. In addition, if $\delta_C^d(x) = 0$ and $\nabla f(x)^\top d < 0$, then there exist an $i \in \{1, \ldots, \ell\}$, such that $d_i \neq 0$ and $x_i \in \{l_i, u_i\}$.
Proof. The first claim follows by the definition of $\delta^C_d(x)$. The second claim follows from the application of Lemma 3.1.4 to $\mathcal{P}_{QP}(f, \{x\} + \mathbb{R} \cdot d, l, u)$. \hfill \Box

The following lemma bound the conservative advance achieved during one iteration. Similar claims (and proofs for it) have been given by Lin [2001b] and List and Simon [2004].

**Lemma 3.1.13 (Conservative advance):** Given an instance $\mathcal{P}_{QP}$ of BQO and a decomposition algorithm $A$. Then during any iteration $n$, such that $d(n)$ is the selected direction, we have have

$$\delta^C_{d(n)}(x^{(n)}) = f(x^{(n)}) - f(x^{(n+1)}) \geq \frac{1}{2} \|x^{(n+1)} - x^{(n)}\|^2_Q,$$

where (*) holds with equality iff iteration $n$ is an unconstrained step.

Proof. To prove (*) consider the Taylor expansion of $f$ around $x^{(n+1)}$:

$$f(x^{(n)}) = f(x^{(n+1)}) + \nabla f(x^{(n+1)})^T(x^{(n)} - x^{(n+1)}) + \frac{1}{2}(x^{(n)} - x^{(n+1)})^T Q (x^{(n)} - x^{(n+1)}) \geq f(x^{(n+1)}) + \frac{1}{2} \|x^{(n)} - x^{(n+1)}\|^2_Q,$$

where the last inequality follows from the fact that $x^{(n+1)}$ is the optimal solution on the feasible line segment between $x^{(n)}$ and $x^{(n+1)}$, and therefore $\nabla f(x^{(n+1)})^T(x^{(n)} - x^{(n+1)}) \geq 0$ with equality iff $x^{(n+1)}$ equals the unconstrained optimum in direction $d(n)$. The claimed equality of $\delta^C_{d(n)}(x^{(n)}) = f(x^{(n)}) - f(x^{(n+1)})$ follows from the definition of $\delta^C_{d(n)}(x^{(n)})$ and the fact, that $x^{(n+1)} \in \{x^{(n)}\} + \mathbb{R} \cdot d(n)$ is the optimal solution of the constrained optimization along $d(n)$. \hfill \Box

The next lemma shows that the value of the aggressive advance can be explicitly calculated. If $\delta_d(x)$ is finite, this is exactly the advance taken by a Newton step along $d$.

**Lemma 3.1.14 (Aggressive advance):** Given an instance $\mathcal{P}_{QP}$ of BQO. For any feasible $x \in \mathcal{R}(\mathcal{P}_{QP})$ and any optimization direction $d \in F(x)$, such that $\nabla f(x)^T d < 0$, either $d \in \text{ker} Q$ and $\delta_d(x) = \infty$, or

$$\delta_d(x) = \frac{1}{2} \left( \frac{\nabla f(x)^T d}{\|d\|_Q} \right)^2,$$

which is achieved at $x + \hat{t} d$ with

$$\hat{t} := -\frac{\nabla f(x)^T d}{\|d\|_Q^2} > 0.$$
Proof. Consider the Taylor expansion of $f(x + td)$ around $x$ and define:

$$\bar{f}(t) := f(x) - f(x + td) = -t\nabla f(x)^\top d - \frac{t^2}{2}d^\top Qd.$$  

If $d \in \text{ker} \ Q$ we have

$$\delta_d(x) = \sup_{t>0} -t\nabla f(x)^\top d = \infty$$

as by assumption $-\nabla f(x)^\top d > 0$.

If $d \notin \text{ker} \ Q$, $\bar{f}$ is strictly concave on $\mathbb{R}$ and attains its maximum at $t \in \mathbb{R}$, such that

$$\bar{f}'(t) = -\nabla f(x)^\top d - td^\top Qd = 0.$$  

Solving for $t$ gives

$$\hat{t} = -\frac{\nabla f(x)^\top d}{d^\top Qd}$$

which is strictly positive by assumption. The claimed value of $\delta_d(x)$ simply follows from plugging $\hat{t}$ in the definition of $\bar{f}(t)$. \qed

Using Definition 3.1.11 we will classify decomposition algorithms in two main classes. Consider therefore the following two definitions:

**Definition 3.1.15 (Qualified Directions):** Given an instance $P_{QP}$ of Constrained Quadratic Optimization and a fixed $\alpha \in \mathbb{R}^+$. Any direction $d \in F(x)$ is called a conservative linear $\alpha$-qualified direction, iff

$$\delta_d^C(x) \geq \alpha \Delta(x),$$

and a conservative quadratic $\alpha$-qualified direction, iff

$$\delta_d^C(x) \geq \alpha (\Delta(x))^2.$$  

Any direction $d \in F(x)$ is an aggressive linear $\alpha$-qualified direction, iff

$$\delta_d(x) \geq \alpha \Delta(x),$$

and an aggressive quadratic $\alpha$-qualified direction, iff

$$\delta_d(x) \geq \alpha (\Delta(x))^2.$$  

**Definition 3.1.16 (Qualified algorithms):** Let $\vec{\alpha} : BQO \to \mathbb{R}^+$ be a mapping assigning to each instance of BQO a positive real number.

A decomposition algorithm $A$ is called a conservative (linear/quadratic) $\vec{\alpha}$-qualified algorithm iff for any instance $P_{QP}$ it selects a conservative (linear/quadratic) $\vec{\alpha}(P_{QP})$-qualified direction in each iteration.

A decomposition algorithm $A$ is called aggressive (linear/quadratic) $\vec{\alpha}$-qualified algorithm iff for any instance $P_{QP}$ it selects an aggressive (linear/quadratic) $\vec{\alpha}(P_{QP})$-qualified direction in each iteration.
3.2 Aggressive Decomposition Algorithms

In the following section we will concentrate on aggressive decomposition algorithms. For convergence properties of conservative selection strategies see Section 3.3.

Let us give an outline of the content of this section: We will first (see Section 3.2.1) introduce a general selection strategy leading to \(k+1\)-sparse aggressive qualified optimization directions. In Section 3.2.2 we will show, that for a special class of BQO, this can be strengthened insofar as they allow for selecting 2-sparse directions. Those problems will be called pairable problems. The selection strategies for these problems will lead to SMO-style decomposition algorithms (see Chapter 5 for details). In Section 3.2.3 we will then derive convergence rates for any aggressive selection strategy, for which the length of so called “locking sequences” (see Definition 3.2.30) can be bounded. The whole section is concluded by showing that at least for the pairable problems introduced in Section 3.2.2 we can derive such a bound (see Section 3.2.4). This leads to a general convergence theorem for aggressive decomposition algorithms (see Theorem 3.2.49), which covers many decomposition algorithms for multiple variants of SVM optimization, as we will show in Chapter 5.

3.2.1 General Selection Strategies

Let us start by considering a “best selection strategy” in the aggressive selection paradigm. If we have only the advance in mind, we could think of greedily finding the direction \(d \in F(x)\) promising the best advance \(\delta_d(x)\).

Let us recall the fact that this can be done by solving a convex optimization problem with linear objective:

**Lemma 3.2.1:** Consider the following optimization problem \(P(x)\):

\[
\max \ -\nabla f(x)^\top d \quad \text{s.t. } d \in F(x), \ |d|_Q \leq 1.
\]

Then either the problem is unbounded, in which case any \(d \in \ker Q \cap F(x)\), where \(-\nabla f(x)^\top d \geq 0\) promises infinite advance, or the optimal solution \(\hat{d} \in F(x)\) is as well the optimal solution of

\[
\max \ \delta_d(x) \quad \text{s.t. } d \in F(x),
\]

i.e. for all \(d \in F(x)\) we have \(\delta_d(x) \leq \delta_{\hat{d}}(x)\). In particular we have for any optimal solution \(\hat{x}\) of \(P_{QP}\) that \(\hat{d} := \hat{x} - x \in F(x)\) and therefore

\[
\delta_d(x) \geq \delta_{\hat{d}}(x) \geq \Delta(x).
\]

**Proof.** First assume that there exists \(d \in \ker Q \cap F(x)\), where \(-\nabla f(x)^\top d = w^\top d > 0\). Obviously \(\delta_d(x) = \sup_{t \geq 0} \{-t\nabla f(x)^\top d\} = \infty\).
In the other case we can conclude that $C^\infty(f) \cap F(x) \subseteq C^0(f)$ and according to [Rockafellar 1970, Theorem 27.3] there exists an optimal solution $\hat{d} \in F(x)$. If $\hat{d} \in \text{ker}(Q)$ we can conclude that $-\nabla f(x)^\top \hat{d} = 0$. As $\hat{d}$ is optimal, this implies $-\nabla f(x)^\top d \leq 0$ for all $d \in F(x)$ and by Lemma 3.1.4 $x$ is an optimal solution. If $\|d\|_Q > 0$ the claim follows from Lemma 3.1.14.

Obviously this selection strategy has a quadratic feasibility region and is as difficult to solve as $P_{QP}$ itself. Remember in addition, that one of our main targets was to have sparse directions, to spare evaluations of the quadratic matrix $Q$. A common approach in the unconstrained case is using the steepest descent direction, which is equivalent to substitute $\|\cdot\|_Q$ by $\|\cdot\|_1$.

We will show, that this is a sensible strategy in our setting, too. Consider therefore the following central selection problem:

**Problem 3.2.2:** Given an instance $P_{QP}$ of BQO and a feasible $x \in \mathcal{R}(P_{QP})$. The following linear optimization problem $P_a(x)$ will be called aggressive gradient selection problem:

$$\max -\nabla f(x)^\top d \quad \text{s.t. } d \in F(x), \|d\|_1 \leq 1.$$ 

To give guarantees for the advance of direction selected according to $P_a(x)$ let us first assume that we have a strictly convex objective:

**Lemma 3.2.3:** Given an instance $P_{QP}$ of BQO, such that $Q$ is strictly positive definite, a fixed $\tau > 0$, and $q \in \{1, \ldots, \ell\}$. Then for any feasible $x \in \mathcal{R}(P_{QP})$ and any $q$-sparse solution $\hat{d} \in \mathcal{R}(P_a(x))$, such that $-\nabla f(x)^\top \hat{d} \geq \tau \cdot \text{val}(P_a(x))$ we conclude that

$$\delta_{\hat{d}}(x) \geq \frac{\tau^2}{\ell \kappa_q(Q)} \Delta(x),$$

where $\kappa_q(Q) = \frac{\lambda_{\text{max},q}(Q)}{\lambda_{\text{min}}(Q)}$.

**Proof.** Under the assumption that $Q$ is strictly positive definite, we know that $C^\infty(f) = \emptyset$, and we conclude from Lemma 3.2.1 that there exist an optimal free direction $\bar{d} \in F(x)$, such that $\|\bar{d}\|_Q = 1$ and $\delta_{\bar{d}} \geq \Delta(x)$. Together with Lemma 3.1.14 this implies that

$$-\nabla f(x)^\top \bar{d} \geq \sqrt{2} \Delta(x).$$

As

$$\|\bar{d}\|_1 \leq \sqrt{\ell} \|\bar{d}\|_2 \leq \sqrt{\frac{\ell}{\lambda_{\text{min}}(Q)}} \|\bar{d}\|_Q,$$
we know that \(\sqrt{\frac{\lambda_{\min}(Q)}{\ell}} \hat{d}\) is a feasible solution of \(\mathcal{P}_a(x)\) and therefore
\[
-\nabla f(x)^\top \hat{d} \geq \sqrt{\frac{2\lambda_{\min}(Q)}{\ell}} \Delta(x).
\]
As \(\hat{d}\) is \(q\)-sparse we conclude that
\[
\|\hat{d}\|_2^2 \leq \lambda_{\max,q}(Q) \|\hat{d}\|_1 \leq \lambda_{\max,q}(Q)
\]
and the claim follows from Lemma \ref{lemma:3.1.14}.

This leads to the following simple conclusion:

**Corollary 3.2.4:** Assume that \(Q\) is strictly positive definite with condition number \(\kappa(Q)\). Then, given a fixed \(\tau > 0\) and \(\bar{q} : \mathcal{BQO} \rightarrow \mathbb{N}\), any \(\bar{q}\)-sparse decomposition algorithm using \(d^{(n)} \in R(\mathcal{P}_a(x^{(n)}))\), such that
\[
-\nabla f(x^{(n)})^\top d^{(n)} \geq \tau \text{val } \mathcal{P}_a(x^{(n)}),
\]
as optimization direction in each iteration \(n\) is an aggressive linear \(\bar{\alpha}_\tau\)-qualified decomposition algorithm with
\[
\bar{\alpha}_\tau(\mathcal{P}_{QP}) := \frac{\tau^2}{\ell_{K_{\bar{q}(\mathcal{P}_{QP})}}(Q)}.
\]  

(3.2.1)

This result specifies the well-known fact that in gradient based approaches the advance per iteration heavily depends on the condition number of the quadratic function. The next lemma will however show that the strict convexity assumption is not necessary in the case, where we have a bounded feasibility region.

**Theorem 3.2.5:** Given an instance \(\mathcal{P}_{QP}\) of \(\mathcal{BQO}\) and \(q \in \{1, \ldots, \ell\}\). Then for any feasible \(x \in R(\mathcal{P}_{QP})\) and any \(q\)-sparse solution \(d \in R(\mathcal{P}_a(x))\) the following inequality holds:
\[
\delta_2(x) \geq \frac{1}{2\lambda_{\max,q}(Q)} (-\nabla f(x)^\top d)^2.
\]  

(3.2.2)

In addition, given a fixed \(\tau > 0\). Then for any \(\tau\)-optimal \(q\)-sparse solution \(\hat{d}\) of \(\mathcal{P}_a(x)\), such that \(-\nabla f(x)^\top \hat{d} \geq \tau \text{val } \mathcal{P}_a(x)\), we have
\[
\delta_2(\hat{d}) \geq \frac{\tau^2}{2s^2S_{\max}(\mathcal{P}_{QP})\lambda_{\max,q}(Q)} (\Delta(x))^2,
\]  

(3.2.3)

where \(s := |\{i \mid x_i \neq \hat{x}_i\}| \leq \ell\) is the number of coordinates \(x\) differs from an optimal solution \(\hat{x}\) and \(S_{\max}(\mathcal{P}_{QP}) := \max_{x,y \in R(\mathcal{P}_{QP})} \|x - y\|_\infty\) is the diameter of the feasibility region wrt. the infinity-norm.
Proof. According to Definition 3.1.11
\[ \delta_d(x) = \sup_{t \geq 0} -t \nabla f(x) \top d - \frac{t^2}{2} \| d \|_Q^2 \geq \sup_{t \geq 0} -t \nabla f(x) \top d - \frac{t^2}{2} L, \]
for any upper bound \( L \geq \| d \|_Q^2 \). Assuming w.l.o.g. that \( L > 0 \) the maximum is then attained at \( t = \frac{-\nabla f(x) \top d}{L} \) and we get
\[ \delta_d(x) \geq \left( \frac{-\nabla f(x) \top d}{L} \right)^2. \]
In addition \( d \) is assumed to be \( q \)-sparse which implies
\[ \| d \|_Q \leq \sqrt{\lambda_{\text{max},q}(Q)} \| d \|_2 \leq \sqrt{\lambda_{\text{max},q}(Q)} \| d \|_1 \leq \sqrt{\lambda_{\text{max},q}(Q)}. \]
Consequently \( L = \sqrt{\lambda_{\text{max},q}(Q)} \) is a suitable upper bound and (3.2.2) follows.

Under the assumption that our current solution differs in at most \( s \) coordinates from an optimal solution \( \hat{x} \), we may give the following bound on the 1-norm of \( \hat{x} - x \):
\[ \| \hat{x} - x \|_1 \leq \| \hat{x} - x \|_\infty \leq s \cdot S_{\text{max}}(P_{QP}), \quad (3.2.4) \]
and therefore \( \frac{1}{s \cdot S_{\text{max}}} (\hat{x} - x) \) is a feasible solution of \( P_a(x) \).

For any \( \tau \)-optimal \( q \)-sparse solution \( \hat{d} \) of \( P_a(x) \) we conclude from (3.2.2)
\[ \delta_d(x) \geq \left( \frac{-\nabla f(x) \top \hat{d}}{2 \sqrt{\lambda_{\text{max},q}(Q)}} \right)^2 \geq \frac{\tau^2}{2s^2 S_{\text{max}}^2(P_{QP}) \lambda_{\text{max},q}(P_{QP})} \right) \left( -\nabla f(x) \top (\hat{x} - x) \right)^2, \]
where the last inequality follows from (3.2.4) and the subsequent comment.

A Taylor expansion of \( f(\hat{x}) \) around \( x \) gives
\[ -\nabla f(x) \top (\hat{x} - x) = f(x) - f(\hat{x}) + \frac{1}{2} \| \hat{x} - x \|_Q^2 \geq \Delta(x), \]
and the claim follows.

This leads to an obvious performance guarantee for aggressive selection, which holds without the strong convexity assumption:

**Corollary 3.2.6:** Given a fixed \( \tau > 0 \) and \( \bar{q} : BQO \to \mathbb{N} \). Any \( \bar{q} \)-sparse decomposition algorithm using \( d^{(n)} \) in \( \mathcal{R}(P_a(x^{(n)})) \), such that
\[ -\nabla f(x^{(n)}) \top d^{(n)} \geq \tau \text{val} P_a(x^{(n)}), \]
as optimization direction in each iteration \( n \) is an aggressive quadratic \( \bar{\alpha}_\tau \)-qualified decomposition algorithm with
\[ \bar{\alpha}_\tau(P_{QP}) := \frac{\tau^2}{2s^2 S_{\text{max}}^2(P_{QP}) \lambda_{\text{max},q}(P_{QP})} \lambda_{\text{max},q}(P_{Q_{\bar{q}}}) \||Q||, \quad (3.2.5)\]
Note that the bound given in Corollary 3.2.6 is not tight in general as we replaced $s$ by its upper-bound $\ell$. However especially in SVM optimization we will often be confronted with settings, where the typical $x^{(n)}$ during a decomposition algorithm only differs in few coordinates from the optimal solution and we can hope for better guarantees in this special case.

As already mentioned in Remark 3.1.5 we can find sparse direction parallel to the edges of $R(P_{QP})$, which will be proven in the next theorem. They are exactly given by the solutions of $P_{a}(x)$, which therefore leads to sparse optimization directions:

**Theorem 3.2.7:** Given an instance $P_{QP}$ of BQO and a feasible $x \in R(P_{QP})$, then there exists an optimal solution $d$ of $P_{a}(x)$, such that $d$ differs from 0 in at most $k + 1$ coordinates, where $k$ is the number of equality constraints.

This solution can be computed in linear time if we only allow instances $P_{QP}$, with constant number of equality constraints.

**Proof.** If $\text{val}_{P_{a}}(x) = 0$ obviously $d = 0$ fulfills the claim. If $\text{val}_{P_{a}}(x) > 0$, we can conclude for any optimal solution $d$, that $\|d\|_1 = \sum_{i=1}^{\ell} d_i = 1$, as otherwise $\frac{1}{\|d\|_1}d \in F(x)$ and $-\nabla f(x)^{\top} \frac{d}{\|d\|_1} > -\nabla f(x)^{\top} d$.

Solving $P_{a}(x)$ is therefore equivalent to solve the following optimization problem $P'_{a}(x)$:

$$\max \begin{pmatrix} -\nabla f(x) \\ \nabla f(x) \end{pmatrix}^{\top} \begin{pmatrix} d^+ \\ d^- \end{pmatrix} \quad \text{s.t.} \quad \sum_{i=1}^{\ell} (d_i^+ + d_i^-) = 1, \quad (3.2.6a)$$

$$\begin{pmatrix} A & -A \end{pmatrix} \begin{pmatrix} d^+ \\ d^- \end{pmatrix} = 0, \quad (3.2.6b)$$

$$d_i^+ d_i^- = 0, \quad d_i^+, \quad d_i^- \geq 0, \quad (3.2.6c)$$

$$x_i = u_i \Rightarrow d_i^+ = 0, \quad x_i = l_i \Rightarrow d_i^- = 0 \quad (3.2.6d)$$

Note that we can ignore the multiplicative constraint in (3.2.6c) because of the symmetry of the other constraints. In addition the equality constraints induced in (3.2.6d) can be ignored, if we delete the variables whose value is fixed to 0, and therefore pass to a lower dimensional problem. $P'_{a}(x)$ therefore is a LP in standard form with $k + 1$ equality constraints. A basic optimal solution therefore has only $k + 1$ non-zero entries.

Given an instance $P_{QP}$, where the number of equality constraints is fixed to a constant, we can use the technique described by Megiddo [1984] to solve the dual of $P_{a}(x)$, which in this case is an LP of fixed dimension. This shows the possibility to compute solutions in linear time.

**Corollary 3.2.8:** Let $k$ be the number of equality constraints of $P_{QP}$. Then any decomposition algorithm $A$ choosing the basic feasible solution $d^{(n)} = \ldots$
opt $P_a(x^{(n)})$ as optimization direction in each iteration $n \geq 0$ is a $k + 1$-sparse decomposition algorithm.

Note that one could use any appropriate LP solver to compute $d = \text{opt } P_a(x)$, for example simplex based solvers, which are known to be fast on most Linear Programs. It would be interesting to see how such a procedure would scale although the polynomial time convergence rate we will present might not be transferred to such a selection procedure.

3.2.2 Pairable Problems

We will now introduce a special class of problems in BQO, which is a natural superclass of many SVM optimization problems. Instances of this class have the advantage that they are solvable by 2-sparse selection strategies, i.e. only a pair of basic directions is needed to achieve progress in each iteration. The convergence analysis we will give will be mainly tailored to this special class.

Let us first present the main tool introduced by List [2004]: Given a problem instance $P(f, A, b, l, r)$ of BQO the linear equality constraint matrix induces the following equivalence relation on $\{1, \ldots, \ell\}$:

$$ i \sim j \iff \exists c_{i,j} \neq 0 : A_i = c_{i,j} A_j. $$

We will denote the equivalence classes for a given $i$ by $[i] := \{j \mid j \sim i\}$. Given a set of representatives $\{i_r \mid r = 1, \ldots, s\}$ we choose the subset

$$ \{a_r := A_{i_r} \mid r = 1, \ldots, s\} \subseteq \{A_i \mid i = 1, \ldots, \ell\} $$

whose elements represent the columns of $A$ up to scalar multiplication. For each equivalence class we define $c_i := c_{i,i_r}$ as the constants that satisfy $A_i = c_{i,i_r} a_r$, for any $i \in [i_r]$. Before we start to give the main definition of this section, we will show that various quantities derived from the constants $c_i$ do not depend on the choice of the representative $a_r$:

**Lemma 3.2.9:** Given an equality constraint matrix $A \in \mathbb{R}^{k \times \ell}$ and an equivalence class $[i_r] \subseteq \{1, \ldots, \ell\}$ as defined above. Let $i', j' \in [i_r]$ be two different representatives, such that for any $i \in [i_r]$ we have $A_i = c_i A_{i'}$ and $c'_i A_i = A_{j'}$, then for all $i, j \in [i_r]$

$$ \text{sgn } c_i = \text{sgn } c'_i \cdot \text{sgn } c'_j \quad \text{and} \quad \frac{c_i}{c_j} = \frac{c'_i}{c'_j}. $$

**Proof.** As $i' \sim j'$ we have $A_{i'} = c'_{i'} A_{j'}$ and therefore for all $i \in [i_r]$ we conclude $c'_i = c'_{i'} c_i$. The two claims follow immediately.

The following definition is crucial:
Definition 3.2.10 (Pairable Problem): Given an instance $\mathcal{P}_{QP}(f, A, b, l, r)$ of BQO, $\mathcal{P}_{QP}$ is called pairable iff any collection of pairwise linear independent columns $A_{i_1}, \ldots, A_{i_t}$ is linear independent. We will call a pairable problem normalized iff $c_i > 0$ for all $i = 1, \ldots, \ell$.

Remark 3.2.11: Note that from Lemma 3.2.9 we can conclude that the definition of normalized pairable problems does not depend on the choice of the representative.

Let us first give some equivalent formulations:

Lemma 3.2.12: Given an instance $\mathcal{P}_{QP}$ of BQO, then the following statements are equivalent

1) $\mathcal{P}_{QP}$ is pairable.

2) The number of equivalence classes $s$ induced by $A$ is equal to $\text{rank} A$.

It is immediately clear that one can transform any pairable instance $\mathcal{P}_{QP}$ into a normalized one by applying the variable substitution $x'_i = \text{sgn}(c_i)x_i$. Note that we will therefore give all arguments only for the simpler normalized case. The following Lemma 3.2.13, given without proof, will ensure that this can be done without loss of generality.

Lemma 3.2.13: Given a pairable instance $\mathcal{P}_{QP}(Q, w, A, b, l, r)$ of BQO and a set of representatives $\{i_r \mid r = 1, \ldots, s\}$, then there exists a normalized pairable instance $\mathcal{P}'_{QP}(Q', w', A', b', l', r')$, such that $\text{opt } \mathcal{P}_{QP} = \text{opt } \mathcal{P}'_{QP}$, and $x$ is an optimal solution for $\mathcal{P}_{QP}$ iff $x'$ is optimal for $\mathcal{P}'_{QP}$, where $x'_i = \text{sgn}(c_i) \cdot x_i$. $\mathcal{P}'_{QP}$ is given by $Q'_{ij} = \text{sgn}(c_i) \cdot \text{sgn}(c_j) Q_{ij}$, $w'_i = \text{sgn}(c_i) w_i$, $A'_i = \text{sgn}(c_i) A_i$ and $l'_i = \begin{cases} l_i & \text{if } c_i > 0 \\ -u_i & \text{if } c_i < 0 \end{cases}$, $u'_i = \begin{cases} u_i & \text{if } c_i > 0 \\ -l_i & \text{if } c_i < 0 \end{cases}$, for all $1 \leq i, j \leq \ell$.

Definition 3.2.14: For any normalized pairable instance $\mathcal{P}_{QP}$ of BQO we will denote by $C(\mathcal{P}_{QP})$ the ratio between the largest and smallest $c_i$ in one equivalence class, i.e.

$$C(\mathcal{P}_{QP}) := \min_{r=1,\ldots,k} \frac{C_r(\mathcal{P}_{QP})}{C_r(\mathcal{P}_{QP})} := \min_{r=1,\ldots,k} \left\{ \frac{\min_{i \in [i_r]} c_i}{\max_{i \in [i_r]} c_i} \right\} \in (0, 1].$$

The following definitions will be handy in the context of normalized pairable problems:
Definition 3.2.15: Given a normalized pairable instance \( P_{QP} \) of BQO, we define for any \( i \sim j \):

\[
d_{ij} := \frac{c_i c_j}{c_i + c_j} \left( \frac{1}{c_i} e_i - \frac{1}{c_j} e_j \right),
\]

and consequently

\[
S_{ij} := S_{d_{ij}} = \left\{ x \left| \frac{c_i c_j}{c_i + c_j} \left( \frac{1}{c_j} \nabla f(x)_j - \frac{1}{c_i} \nabla f(x)_i \right) > 0 \right. \right\} \quad \text{and}
\]

\[
N_{ij} := N_{d_{ij}} = \left\{ x \left| \frac{c_i c_j}{c_i + c_j} \left( \frac{1}{c_j} \nabla f(x)_j - \frac{1}{c_i} \nabla f(x)_i \right) \leq 0 \right. \right\} .
\]

Remark 3.2.16: 1) According to Lemma 3.2.9, Definition 3.2.14 and Definition 3.2.15 do not depend on the special choice of a representative.

2) The importance of Definition 3.2.15 will become clear during this section: In fact, we will show that in the case of normalized pairable \( P_{QP} \) the selection of a pair \( i \sim j \) is equivalent to a well defined optimization direction \( d_{ij} \). This means that SMO type decomposition algorithms, which have been motivated by selecting a working “pair” are in fact equivalent to our view on decomposition methods as descent line search algorithms (see Remark 3.3.6 and comments at end of Section 3.1.1).

We are now ready to state the most important consequence from Definition 3.2.10:

Theorem 3.2.17: Given a normalized pairable instance \( P_{QP} \) of BQO. Then for any feasible \( x \in \mathbb{R}(P_{QP}) \), the direction \( d_{ij} \in F(x) \) where \((i, j)\) is given by

\[
\arg \max \left\{ \frac{c_i \nabla f(x)_j - c_j \nabla f(x)_i}{c_i + c_j} \left| \ x_i < u_i \ and \ x_j > l_j \ and \ i \sim j \right. \right\},
\]

is an optimal solution of \( P_a(x) \). The pair \((i, j)\) is called maximum violating pair (MVP).

We will call indexes, such that \( x_i > l_i \) down candidates and indexes such that \( x_i < u_i \) up candidates

Let us postpone the proof of Theorem 3.2.17 and first focus on some intermediate claims: 1) If we restrict our problem on one equivalence class the maximum violating pair leads to an optimal solution within this class. 2) Pairable problems can be decomposed by solving the subproblems class-wise and 3) it is best to concentrate all weights in just one equivalence class.

Let us start to be more precise and define the following optimization problem:
Definition 3.2.18: Given a normalized pairable instance $P_{QP}$ of BQO. For any equivalence class $r \in \{1, \ldots, k\}$, any $0 \leq \lambda \leq 1$ and any feasible $x \in \mathcal{R}(P_{QP})$, we will then denote the following Linear Program by $P_{a}''(x, r, \lambda)$:

$$\begin{align*}
\max \left( -g \right)^\top \begin{pmatrix} d^+ \newline d^- \end{pmatrix} \quad & \text{s.t.} \quad \sum_{i \in [i_r]} (d^+_i + d^-_i) \leq \lambda, \quad (3.2.8a) \\
& (c - c) \begin{pmatrix} d^+ \\
- d^- \end{pmatrix} = 0, \quad (3.2.8b) \\
& d^+_i d^-_i = 0, \quad d^+_i \geq 0, \quad d^-_i \geq 0, \quad (3.2.8c) \\
& x_i = u_i \Rightarrow d^+_i = 0, \quad x_i = l_i \Rightarrow d^-_i = 0 \quad (3.2.8d)
\end{align*}$$

where $g = (\nabla f(x)_i)_{i \in [i_r]}$ and $c = (c_i)_{i \in [i_r]} > 0$.

Note that from Lemma 3.2.9 we can again conclude that this Definition does not depend on the choice of representatives.

The following lemma shows that the problem $P_{a}''(x, r, \lambda)$ is solvable by a pairing strategy:

Lemma 3.2.19: Given a normalized pairable instance $P_{QP}$ of BQO. Then for any feasible $x \in \mathcal{R}(P_{QP})$, any $r \in \{1, \ldots, s\}$ and any $0 \leq \lambda \leq 1$

$$\hat{d} = \lambda \cdot d_{ij},$$

where $(i, j)$ is given by

$$\arg \max \left\{ \frac{c_i \nabla f(x)_j - c_j \nabla f(x)_i}{c_i + c_j} \right\} \quad \text{where} \quad i, j \in [i_r], \quad x_i < u_i \quad \text{and} \quad x_j > l_j.$$

is an optimal solution of $P_{a}''(x, r, \lambda)$.

Proof. First of all note that if $\text{val} \ P_{a}''(x, r, \lambda) = 0$ all $\frac{1}{c_i} \nabla f(x)_i$ are balanced, as otherwise we could improve the value of $d = 0$ along the direction $\frac{1}{c_i} e_i - \frac{1}{c_j} e_j$ for some $i, j$ where $-\frac{1}{c_i} \nabla f(c)_i > -\frac{1}{c_j} \nabla f(x)_j$. In this case the maximum is always zero and the claim follows.

We can replace (3.2.8a) by $\sum_{i \in [i_r]} (d^+_i + d^-_i) = \lambda$ if $\text{val} \ P_{a}''(x, r, \lambda) > 0$ (compare to the proof of Theorem 3.2.7). In this case $P_{a}''(x, r, \lambda)$ is a linear program in standard form, with two equality constraints. That means, there exists a basic optimal solution, say $d^{\pm}$, with only two coordinates differing from zero. Let us then assume that $(i, j)$ are these two coordinates. From the fact that $c > 0$ we conclude that either $d^+_i > 0$ and $d^-_j > 0$ or $d^+_i > 0$ and $d^-_j > 0$ and

\footnote{We again ignore the equality constraints induced by the border of the box, as we can simply delete the appropriate variables and give our arguments for an even simpler problem.}
\( \hat{d}_j^+ > 0 \) as otherwise (3.2.8b) could not be fulfilled. Wlg. we can therefore assume that
\[
\hat{c}_i \hat{d}_i^+ - \hat{c}_j \hat{d}_j^+ = 0.
\]
and, by (3.2.8a), \( \hat{d}_i^+ + \hat{d}_j^- = \lambda \).

We conclude that
\[
\hat{d}_i^+ = \lambda \frac{\hat{c}_j}{\hat{c}_i + \hat{c}_j} > 0 \quad \text{and} \quad \hat{d}_j^- = \lambda \frac{\hat{c}_i}{\hat{c}_i + \hat{c}_j} > 0
\]
is the only valid choice. From (3.2.8d) it immediately follows that \( x_i < u_i \) and \( x_j > l_j \) and therefore an optimal solution can found as follows:

\[
\text{opt } P''_a(x,r,\lambda) = \max \left\{ \frac{\lambda}{\hat{c}_i + \hat{c}_j} (\hat{c}_j \nabla f(x)_j - \hat{c}_i \nabla f(x)_i) \mid x_i < u_i, x_j > l_j \right\}
\]

\[
= \max \left\{ \frac{c_i c_j}{c_i + c_j} \left( \frac{1}{c_j} \nabla f(x)_j - \frac{1}{c_i} \nabla f(x)_i \right) \mid x_i < u_i, x_j > l_j \right\},
\]
which concludes the proof.

From Lemma 3.2.19 one can draw one more simple consequence:

**Corollary 3.2.20:** Given a normalized pairable instance \( QP \) of BQO. Then for any feasible \( x \in R(QP) \), any \( r \in \{1, \ldots, k\} \) and any \( 0 \leq \lambda \leq 1 \) we have

\[
\lambda \text{opt } P''_a(x, r, 1) = \text{opt } P''_a(x, r, \lambda).
\]

It remains to show that the full selection problem \( P_a(x) \) can be solved by independently solving the selection problems \( P''_a(x, r, \lambda) \) restricted to the equivalence classes. The following lemma will show this remaining part and therefore immediately imply Theorem 3.2.17.

**Lemma 3.2.21:** Given a normalized pairable instance \( QP \) of BQO and a feasible \( x \in R(QP) \), then the following optimization problems are equivalent:

\[
P_a(x), \quad (3.2.9a)
\]

\[
\max \left\{ \sum_{r=1}^{k} \text{opt } P''_a(x, r, \lambda_r) \mid \sum_{r=1}^{k} \lambda_r \leq 1, \lambda_r \geq 0 \right\} \quad \text{and} \quad (3.2.9b)
\]

\[
\max_{r=1, \ldots, k} \text{opt } P''_a(x, r, 1). \quad (3.2.9c)
\]

**Proof.** To prove the equivalence between (3.2.9b) and (3.2.9c) simply note that for a given \( x \in R(QP) \) and an \( r \in \{1, \ldots, k\} \) the value of \( \text{opt } P''_a(x, r, 1) \) is constant. In addition, for any \( 0 \leq \lambda_r \leq 1 \) we conclude by Corollary 3.2.20 that \( \text{opt } P''_a(x, r, \lambda_r) = \lambda_r \text{opt } P''_a(x, r, 1) \). Therefore the objective of (3.2.9b)
Chapter 3. Decomposition Techniques for Quadratic Programming

is a convex combination of real numbers and the maximum is therefore attained if the weight is given to the largest of all summands, which is given by \(3.2.9c\). To prove the first equivalence we show that \(3.2.9b\) and \(P'_a(x)\) are equivalent. To this end let us recall the equality constraint \(3.2.6b\) of \(P'_a(x)\):

\[
(\mathbf{A} - \mathbf{A}) (d^+ - d^-) = \sum_{r=1}^{k} a_r \sum_{i \in [i_r]} c_i (d_i^+ - d_i^-) = 0.
\]

As by assumption the \(a_r, r = 1, \ldots, k\) are linear independent, the equality holds iff

\[
\sum_{i \in [i_r]} c_i (d_i^+ - d_i^-) = 0
\]

for all \(r = 1, \ldots, k\), which is equivalent to \(3.2.8b\). As the \([i_r]\) are pairwise disjoint the objective is separable as well and we can write (ignoring all constraints up to \(3.2.6a\) and \(3.2.8a\) respectively):

\[
\max_{\sum_i d_i \leq 1} \sum_{i=1}^{\ell} -\nabla f(x)_i (d_i^+ - d_i^-)
\]

\[
= \max_{\sum_r \lambda_r \leq 1} \sum_{r=1}^{k} \left[ \max_{\sum_i d_i \leq \lambda_r} \sum_{i \in [i_r]} -\nabla f(x)_i (d_i^+ - d_i^-) \right],
\]

which concludes the proof.

**Efficient Computation of (Approximate) Maximum Violating Pairs**

Note that the definition of maximum violating pairs given in \(3.2.7\) suggests a straightforward way to compute such a pair in \(O(\ell^2)\) steps. This however is not efficient, as one can use again the linear time algorithm of Megiddo [1984] (see Theorem 3.2.7).

In this section we will show that there exists an even simpler method with \(O(\ell)\)-runtime, if one is willing to accept a \(\tau\)-approximate solution, i.e. a pair \((i, j)\), such that for a given \(x \in \mathbb{R}^{\mathcal{P}_{QP}}\) at least

\[
\frac{c_i \nabla f(x)_j - c_j \nabla f(x)_i}{c_i + c_j} \geq \tau \text{ opt } \mathcal{P}_a(x).
\]

Together with Theorem 3.2.6 this immediately implies that the corresponding direction \(d_{ij}\) induces a 2-sparse aggressive quadratic \(\tilde{\alpha}_\tau\)-qualified decomposition algorithm.

To calculate a \(\tau = C(P_{QP})\) (see Definition 3.2.14) approximate maximal violating pair, one can consider the following easy selection strategy introduced by List [2004], which is an extension of the pairing strategies for SVM optimization (see Lin [2001b], Chang and Lin [2001a]):

\footnote{In the notation of List [2004] the scaling factor associated to each equivalence class is \(\lambda_i\) and corresponds to \(c_i = 1/\lambda_i\) in our notation. Furthermore \(\lambda_i < 0\) is possible, as the concept of normalized pairable problems has not been used by the author.}
Lemma 3.2.22: Given a normalized pairable instance $P_{QP}$ of BQO. Then for any feasible $x \in \mathcal{R}(P_{QP})$, any $r \in \{1, \ldots, k\}$, $d_{ij}$, where $(i, j)$ is given by

$$
i = \arg \max \left\{ -\frac{\nabla f(x)_i}{c_i} \mid i \in [i_r], x_i < u_i \right\},$$

$$j = \arg \min \left\{ -\frac{\nabla f(x)_j}{c_j} \mid j \in [i_r], x_j > l_j \right\}.$$  \hspace{1cm} (3.2.10)

is an approximate solution of $P''_a(x, r, 1)$, with value

$$-\nabla f(x)\top d_{ij} = \frac{c_i c_j}{c_i + c_j} \left( \frac{1}{c_j} \nabla f(x)_j - \frac{1}{c_i} \nabla f(x)_i \right) \geq C_r(P_{QP}) \text{val} P''_a(x, r, 1).$$

Proof. According to Lemma 3.2.19 we know that

$$\max_{i,j \in [i_r]} \left\{ \frac{1}{c_i + c_j} (c_i \nabla f(x)_j - c_j \nabla f(x)_i) \mid x_i < u_i, x_j > l_j \right\}$$

$$= \max_{i,j \in [i_r]} \left\{ \frac{c_i c_j}{c_i + c_j} \left( \frac{\nabla f(x)_j}{c_j} - \frac{\nabla f(x)_i}{c_i} \right) \mid x_i < u_i, x_j > l_j \right\}$$

is the optimal solution of $P''_a(x, r, 1)$. As by selection rule (3.2.10) for any $i, j \in [i_r]$ we have

$$\frac{\nabla f(x)_j}{c_j} - \frac{\nabla f(x)_i}{c_i} \geq \frac{\nabla f(x)_j}{c_j} - \frac{\nabla f(x)_i}{c_i}$$

the only difference between the optimization problems are the different factors

$$p(x, y) := \frac{xy}{x + y}.$$ 

Note that $p$ is monotonically increasing in $x$ and $y$. And therefore both factors $p(c_i, c_j)$ and $p(c_i, c_j)$ are lying in the interval

$$[p(c_{\min}, c_{\min}), p(c_{\max}, c_{\max})] = \left[ \frac{c_{\min}}{2}, \frac{c_{\max}}{2} \right],$$

where $c_{\min} = \min_{i \in [i_r]} c_i$ and $c_{\max} = \max_{i \in [i_r]} c_i$ are the smallest and largest row multiple in equivalence class $r$. The ration between $p(c_i, c_j)$ and $p(c_i, c_j)$ is therefore bounded by $c_{\min}/c_{\max} = C_r(P_{QP})$. \qed

The above Lemma 3.2.22 induces a general selection technique for normalized pairable instances of $P_{QP}$ (see Appendix A Algorithm A.1 for details). Using Lemma 3.2.21 and Corollary 3.2.6 one give aggressive guarantees for such algorithms. It is worth to note that this guarantee is directly inherited by the second order heuristic proposed by Fan et al. [2005].
Corollary 3.2.23: Given a normalized pairable instance $P_{QP}$ of BQO. Then for any $x \in R(P_{QP})$ Algorithm A.1 and Algorithm A.2 compute a pair $(i, j)$ in linear time $O(\ell)$ such that $d_{ij}$ is an aggressive quadratic $\tilde{\alpha}_2(P_{QP})$-qualified direction, where

$$\tilde{\alpha}_2(P_{QP}) = \frac{C^2(P_{QP})}{2\ell^2 S_{\max}^2(P_{QP}) \lambda_{\max, 2}(Q)}.$$ 

If $Q$ is strictly positive definite it is an aggressive linear $\tilde{\alpha}_1(P_{QP})$-qualified direction, where

$$\tilde{\alpha}_1(P_{QP}) = \frac{C^2(P_{QP})}{\ell \kappa_2(Q)}.$$ 

A decomposition algorithm using Algorithm A.1 or Algorithm A.2 as selection procedure in each iteration therefore is an aggressive quadratic $\tilde{\alpha}_2$-qualified decomposition algorithm (or linear $\tilde{\alpha}_1$-qualified if $Q$ is strictly positive definite).

Proof. It is easily seen that both Algorithms have linear run time. The claim for Algorithm A.1 directly follows from Lemma 3.2.21 and Corollary 3.2.6. To prove the claim for Algorithm A.2 note that it scans at least the approximate maximum violating pair from Lemma 3.2.22. Then given $i = \arg \max \left\{-\frac{\nabla f(x)_i}{c_i} \mid i \in [i_r], x_i < u_i \right\}$ it chooses $j$ such that

$$j = \arg \max \left\{\delta_{ij}(x) \mid j \in [i_r], x_j > l_j \right\},$$

which leads to a possible better guarantee. 

Let us finally note that the described procedure is an optimal MVP selection strategy if ever $C(P_{QP}) = 1$, which is the case in most SVM optimization algorithms we will consider (see Chapter 5 for details).

Remark 3.2.24: Obviously one can hope that Algorithm A.2 produces much better guarantees than Algorithm A.1. Although this seems to be the case in practice as the second order heuristics leads to a significant speedup in the selection process we are possibly not able to give better bounds with the methods presented here. This is due to two facts: First, the selection of the first direction is fixed by taking into account first order approximation only, which might be misleading. Second, and more important, we optimize the problem on iteratively chosen subspaces in either variant, which are possibly highly correlated.

3.2.3 Convergence Rates

Let us start the proof of a general convergence theorem for aggressive selection strategies from a simplified viewpoint: Reconsider Lemma 3.1.14 and
Definition 3.1.16 and consider only iterations, where we do not hit the border of our feasibility region. It is then easy to see that in such unconstrained steps \( n \) of an aggressive \( \alpha \)-qualified decomposition algorithm \( A \), we can achieve an advance bounded by

\[
f(x^{(n)}) - f(x^{(n+1)}) = \delta_d(x^{(n)}) \geq \tilde{\alpha}(\mathcal{P}_{QP}) \begin{cases} \Delta^{(n)} & \text{if } A \text{ is linear } \tilde{\alpha}\text{-qualified} \\ \Delta^{(n)}^2 & \text{if } A \text{ is quadratic } \tilde{\alpha}\text{-qualified} \end{cases}
\]

where \( d \in F(x^{(n)}) \) is the selected direction. This leads to a contraction by a factor \( 1 - \tilde{\alpha}(\mathcal{P}_{QP}) \) if \( A \) is linear \( \tilde{\alpha}\)-qualified and to a reduction by a factor \( 1 - \tilde{\alpha}(\mathcal{P}_{QP}) \Delta^{(n)} \) if \( A \) is quadratic \( \tilde{\alpha}\)-qualified, i.e.

\[
\Delta^{(n+1)} \leq \begin{cases} (1 - \tilde{\alpha}(\mathcal{P}_{QP})) \Delta^{(n)} & \text{if } A \text{ is linear } \tilde{\alpha}\text{-qualified} \\ (1 - \tilde{\alpha}(\mathcal{P}_{QP}) \Delta^{(n)}) \Delta^{(n)} & \text{if } A \text{ is quadratic } \tilde{\alpha}\text{-qualified} \end{cases}
\]

This however only holds for unconstrained steps, where we do not hit the border of the feasibility region before reaching the optimum in direction \( d \). All convergence arguments for aggressive selection strategies in SVM optimization (see for example [Lin 2001a, Chen et al. 2006, Glasmachers 2008]) therefore focus on these unconstrained steps.

In this work we will in contrary to previous attempts show that it may not be wrong to select directions hitting the border. In this chapter we will therefore prove a convergence theorem for aggressive selection strategies, which will be based on certain assumptions how often and how long such frustrating locks at the border of the feasibility region may occur (see Definition 3.2.30 for an exact account). We will later on show (see Section 3.2.4) that for the maximum violating pair selection strategy according to Lemma 3.2.22, which is implemented in all major software packages, these so called locking sequences can be bounded. The application to SVM optimization in Chapter 5 will then show that the usual aggressive selection strategies implemented in SVM decomposition algorithms meet these conditions.

Let us first introduce the main concepts our general convergence theorem builds on.

**Definition 3.2.25:** Given an instance \( \mathcal{P}_{QP} \) of BQO and an arbitrary direction \( d \in \mathbb{R}^\ell \), then define

\[
H^0_d := \left\{ x \mid \nabla f(x)^\top d = 0 \right\} \\
S_d := \left\{ x \mid \nabla f(x)^\top d < 0 \right\} \\
N_d := \left\{ x \mid \nabla f(x)^\top d \geq 0 \right\}
\]
Let us state two simple properties of these sets:

**Lemma 3.2.26:** For any \( d \notin \ker Q \) the set \( H^0_d \) is an affine linear subspace of dimension \( \ell - 1 \). If \( d \in \ker Q \), then either \( S_d = \mathbb{R}^\ell \) or \( N_d = \mathbb{R}^\ell \).

*Proof.* We have \( \nabla f(x)^T d = (Qx - w)^T d = x^T Qd - w^T d \). If \( d \in \ker Q \), we either have \( \nabla f(x)^T d = -w^T d \geq 0 \) and therefore \( N_d = \mathbb{R}^\ell \) or \( \nabla f(x)^T d = -w^T d < 0 \) and therefore \( S_d = \mathbb{R}^\ell \).

If \( d \notin \ker Q \), \( Qd \in \mathbb{R}^\ell \) is the normal vector to the \( \ell - 1 \) dimensional affine subspace:

\[
H^0_d = \left\{ x \mid x^T Qd = w^T d \right\}
\]

**Corollary 3.2.27:** If \( d \in F(x) \) is the selected direction during a decomposition algorithm \( A \), then \( x \in S_d \).

In addition this implies a geometric interpretation of \( \delta_d(x) \):

**Lemma 3.2.28:** Let \( d \notin \ker Q \), then for any \( x \in S_d \)

\[
\delta_d(x) = \min_{y \in N_d} \frac{1}{2} \| x - y \|^2_Q.
\]

*Proof.* According to Lemma 3.1.14 we have

\[
\hat{x} := x - \frac{\nabla f(x)^T d}{\| d \|^2_Q} \quad d \in H^0_d \subset N_d,
\]

such that \( \delta_d(x) = \frac{1}{2} \| x - \hat{x} \|^2_Q \). It therefore suffices to show that for all \( y \in N_d \)

\[
\delta_d(x) \leq \frac{1}{2} \| x - y \|^2_Q.
\]

We will first show that it is sufficient to prove that this inequality holds for all \( y \in H^0_d \). To this end given any \( y \in N_d \) we can consider the line segment \( x(t) := x + t(y - x), \quad t \in [0, 1] \). As \( -\nabla f(x(0)) < 0 \) and \( -\nabla f(x(1)) \geq 0 \) we conclude, by the intermediate value theorem, that there exists a \( t_0 \in (0, 1] \) such that \( x_0 := x(t_0) \in H^0_d \). As obviously \( \| x - y \|^2_Q = \frac{1}{t_0^2} \| x_0 - x \|^2_Q \geq \| x - x_0 \|^2_Q \) our restriction to \( H^0_d \) is justified.

Now note that for any \( y \in H^0_d \) we have \( (y - \hat{x})^T Qd = 0 \) and therefore

\[
\| y - x \|^2_Q = \| y - \hat{x} \|^2_Q + \| \hat{x} - x \|^2_Q \geq \| \hat{x} - x \|^2_Q,
\]

which concludes the proof.

As already noted, this advance is only guaranteed in any iteration where we do not hit the border. However in any iteration we can bound the advance in the function value by the distance crossed with respect to the \( Q \)-semi norm:

A crucial property for the convergence of aggressive strategies is the fact that this can be extended to an arbitrary sequence of iterations as follows:
Lemma 3.2.29 (Passive Advance): Given two arbitrary iterations \( m > n \) during a decomposition algorithm \( A \), then

\[
f \left( x^{(n)} \right) - f \left( x^{(m)} \right) \geq \frac{1}{2(m-n)} \| x^{(n)} - x^{(m)} \|_Q^2.
\]

Proof. Let us denote with \( \Delta := f \left( x^{(n)} \right) - f \left( x^{(m)} \right) \) the distance achieved from iteration \( n \) to iteration \( m \) and denote with

\[
\delta_i^2 := \frac{f \left( x^{(i)} \right) - f \left( x^{(i+1)} \right)}{\Delta}
\]

the relative advance in one step, such that \( \sum_{i=n}^{m-1} \delta_i^2 = 1 \). Then by Lemma 3.1.13 we have

\[
\Delta \delta_i^2 \geq \frac{1}{2} \| x^{(i+1)} - x^{(i)} \|_Q^2.
\]  \hspace{1cm} (3.2.11)

On the other hand

\[
\frac{1}{2} \| x^{(n)} - x^{(m)} \|_Q^2 = \sum_{i=n}^{m-1} \frac{1}{2} \| x^{(i+1)} - x^{(i)} \|_Q^2 + \sum_{n \leq i < j \leq m-1} \left\langle \left( x^{(i+1)} - x^{(i)} \right), \left( x^{(j+1)} - x^{(j)} \right) \right\rangle_Q,
\]

and therefore

\[
\Delta \sum_{i=n}^{m-1} \delta_i^2 \geq \sum_{i=n}^{m-1} \frac{1}{2} \| x^{(i+1)} - x^{(i)} \|_Q^2 - \sum_{n \leq i < j \leq m-1} \left\langle \left( x^{(i+1)} - x^{(i)} \right), \left( x^{(j+1)} - x^{(j)} \right) \right\rangle_Q,
\]  \hspace{1cm} (3.2.12)

where the last summand is 0 if \( m - n \leq 1 \) and can otherwise be bounded using the Cauchy-Schwartz inequality and (3.2.11) as follows:

\[
\sum_{n \leq i < j \leq m-1} \left\langle \left( x^{(i+1)} - x^{(i)} \right), \left( x^{(j+1)} - x^{(j)} \right) \right\rangle_Q \leq \sum_{n \leq i < j \leq m-1} \| x^{(i+1)} - x^{(i)} \|_Q \| x^{(j+1)} - x^{(j)} \|_Q \leq 2\Delta \sum_{n \leq i < j \leq m-1} \delta_i \delta_j.
\]

We can bound this value by calculating the maximum of the sum under the constraint \( \sum_{i=n}^{m-1} \delta_i^2 = 1 \). This maximum is achieved if the step-size \( \delta_i = \sqrt{\frac{1}{m-n}} \) is equal for all steps \( i = n, \ldots, m-1 \), and therefore

\[
2\Delta \sum_{n \leq i < j \leq m-1} \delta_i \delta_j \leq 2\Delta \left( \frac{m-n}{2} \right) \frac{1}{m-n} = (m-n-1)\Delta.
\]
Combining this with (3.2.11) and (3.2.12) we conclude

\[ \Delta = \Delta \sum_{i=1}^{m-1} \delta_i^2 \geq \frac{1}{2} \left\| x^{(n)} - x^{(m)} \right\|_Q^2 - \begin{cases} 0 & m-n \leq 1 \\ (m-n-1)\Delta & m-n \geq 2 \end{cases}. \]

Solving for \( \Delta \) implies

\[ \Delta \geq \frac{1}{2} \left\| x^{(n)} - x^{(m)} \right\|_Q^2 \cdot \frac{1}{1 + \max\{0, m-n-1\}} = \frac{1}{2(m-n)} \left\| x^{(n)} - x^{(m)} \right\|_Q^2. \]

This means that, if ever we can guarantee an advance in \( Q \)-distance during a decomposition algorithm, we can (with a certain loss) guarantee an advance in the objective value. This coincides with the geometric interpretation of our selection criterion \( \delta_d(x) \) (see Lemma 3.2.28), where we try to select the direction, which promises largest \( Q \)-distance. This promised advance may however not be realized during a longer period. Let us formalize such “frustrating” sequences in the following definition:

**Definition 3.2.30:** Given an instance \( \mathcal{P}_{QP} \) of BQO and a decomposition algorithm \( A \), we call any sequence of iterations \( n, \ldots, m-1 \) as locking sequence, if the following two conditions hold:

1) \( x^{(s)} \) is not optimal for all \( s = n, \ldots, m-1 \).

2) For any direction \( d \), selected in some iteration \( s_0 \in \{n, \ldots, m-1\} \), we have \( x^{(s)} \in S_d \) for all \( s \in \{n, \ldots, m-1\} \).

A sequence \( x^{(n)}, \ldots, x^{(m-1)} \) is called a maximal locking sequence starting at iteration \( n \) iff it is a locking sequence and \( x^{(n)}, \ldots, x^{(m)} \) is no locking sequences. We say \( m-n \) is the length of the maximal locking sequence starting at iteration \( n \) and denote with

\[ L_A(\mathcal{P}_{QP}) := \sup_{n,m} \left\{ m-n \mid x^{(n)}, \ldots, x^{(m-1)} \text{ is a locking sequence} \right\} \]

the longest locking sequence of \( \mathcal{P}_{QP} \) under \( A \).

Let us give some remarks on this definition:

**Remark 3.2.31:**

1) An unconstrained step coincides with a maximal locking sequence of length 1.

2) Note that \( L_A(\mathcal{P}_{QP}) \) is possibly infinite. We will later see that is crucial for the convergence proof that we can bound it from above for certain selection techniques.
3) If $L_A(\mathcal{P}_{QP})$ is finite all iterations of the decomposition algorithm $A$ on the instance $\mathcal{P}_{QP}$ can be covered by a sequence of disjoint maximal locking sequences as follows: The first maximal locking sequence starts at iteration $n_1 = 0$ and ends at a well defined iteration $n_2 - 1$. We then start the second maximal locking sequence at iteration $n_2$. This procedure can be applied recursively, such that if the $r-1$-th maximal locking sequence ends at iteration $n_r - 1$ the $r$-th maximal locking sequence starts at iteration $n_r$ for $r > 1$.

4) Consider a maximum locking sequence, which started at iteration $n$ and has not reached optimality yet. Such a sequence ends, iff either a previously selected direction $d$ finally crosses its optimality barrier, i.e. there exists an iteration $s_0 \in \{n, \ldots, m - 1\}$ such that $x^{(s_0)} \in S_d$ and $x^{(m)} \in N_d$, or a previously unselectable direction $d$ gets selected, i.e. there exist an iteration $s_1 \in \{n, \ldots, m - 1\}$ such that $x^{(s_1)} \in N_d$ and $x^{(m)} \in S_d$.

5) For any $d$, whose selection is involved in the ending of a locking sequence, we have $d \not\in \text{kern} Q$. This follows from Lemma 3.2.26 as for any $d \in \text{kern} Q$ we either have $S_d = \emptyset$ or $N_d = \emptyset$ and so none of the above mentioned conditions can occur. This subtle detail will get important (see proof of Lemma 3.2.32 and Remark 3.2.33).

We will now show that during a locking sequence we are able to guarantee a certain advance:

**Lemma 3.2.32:** Given an instance $\mathcal{P}_{QP}$ of Constrained Quadratic Programming and an aggressive $\vec{\alpha}$-qualified decomposition algorithm $A$. For any two iterations $n < m$ let us define

$$M_{n,m} := \vec{\alpha}(\mathcal{P}_{QP}) \frac{m-n}{m-n} > 0$$

Then, if $x^{(n)}, \ldots, x^{(m-1)}$ is a maximal locking sequence starting at $n$, we can bound the advance by

$$\Delta^{(m)} \leq \frac{1}{1 + M_{n,m} \Delta^{(n)}}$$

if $A$ is linear $\vec{\alpha}$-qualified.

If $A$ is quadratic $\vec{\alpha}$-qualified then

$$\Delta^{(m)} \leq \left(1 - \frac{1}{4} M_{n,m} \Delta^{(n)}\right) \Delta^{(n)}$$

if $\Delta^{(n)} \leq \frac{2}{M_{n,m}}$ and

$$\Delta^{(m)} \leq \frac{1}{2} \Delta^{(n)},$$

otherwise.
Chapter 3. Decomposition Techniques for Quadratic Programming

Proof. Take an arbitrary parameter $\lambda \in (0, 1)$, which will suitably be chosen later and consider two cases: Either (1) $\Delta^{(m)} \leq \lambda \Delta^{(n)}$ or (2) $\Delta^{(m)} > \lambda \Delta^{(n)}$. The first case would be a favorable situation, as the distance to optimality has been shrunk by a factor of $\lambda < 1$. For the second case we claim that the advance $f(x^{(n)}) - f(x^{(m)})$ made during the maximal locking sequence starting at $n$ can be bounded from below as follows:

$$f(x^{(n)}) - f(x^{(m)}) > \frac{\bar{\alpha}(P_Q P)}{m - n} \lambda^\nu (\Delta^{(n)})^\nu,$$

(3.2.13)

where $\nu = 1$ if $\mathcal{A}$ is linear $\bar{\alpha}$-qualified and $\nu = 2$ if $\mathcal{A}$ is quadratic $\bar{\alpha}$-qualified.

Let us postpone the proof of (3.2.13) and let us first draw some consequences of this inequality: Consider therefore the auxiliary function

$$g_\nu(\lambda) := 1 - \lambda^\nu M_{n,m} (\Delta^{(n)})^{\nu - 1}.$$

For any $\lambda \in (0, 1)$ we can then bound

$$\Delta^{(m)} = \Delta^{(n)} - \left( f(x^{(n)}) - f(x^{(m)}) \right) \leq \max \{ \lambda, g_\nu(\lambda) \} \Delta^{(n)}.$$  

(3.2.14)

In the case where $\mathcal{A}$ is linear $\bar{\alpha}$-qualified we will therefore choose a $\lambda$, such that $\max \{ \lambda, g_1(\lambda) \} = \max \{ \lambda, 1 - M_{n,m}\lambda \}$ is minimal, which is the case for $\lambda = g_1(\lambda)$. Solving for $\lambda$ gives $\lambda_1 = \frac{1}{1 + M_{n,m}}$ and the claim for linear $\bar{\alpha}$-qualified algorithms follows from (3.2.14).

In the case where $\mathcal{A}$ is quadratic $\bar{\alpha}$-qualified we simply choose $\lambda_2 = \frac{1}{2}$. Then $\Delta^{(n)} > \frac{2}{M_{n,m}}$ is equivalent to

$$\lambda_2 = \frac{1}{2} > 1 - \frac{1}{4} M_{n,m} \Delta^{(n)} = g_2(\lambda_2),$$

and the claims for quadratic $\bar{\alpha}$-qualified aggressive algorithms follow again from (3.2.14).

Let us now prove (3.2.13), which is claimed to hold under the assumption that $\Delta^{(m)} > \lambda \Delta^{(n)}$. To this end note that at the end of the maximal locking sequence starting at iteration $n$, one of the following claims holds: (1) For the direction $d_0 = d^{(m)}$, selected in iteration $m$, there exists an iteration $s_0 \in \{ n, \ldots, m \}$ such that $x^{(s_0)} \in \mathcal{N}_{d_0}$ or (2) there exists an iteration $s_1 \in \{ n, \ldots, m \}$ such that for the selected direction $d_1 = d^{(s_1)}$ we have $x^{(s_1)} \in \mathcal{S}_{d_1}$ and $x^{(m)} \in \mathcal{N}_{d_1}$. Note that neither $d_0$ nor $d_1$ are in kern $Q$ as already stated in Remark 3.2.31. In Case 1) we therefore conclude by Lemma 3.2.28 and Definition 3.1.16 that

$$\frac{1}{2} \| x^{(s_0)} - x^{(m)} \|_Q^2 \geq \delta_{d_0} (x^{(m)}) \geq \bar{\alpha}(P_Q P) (\Delta^{(m)})^\nu,$$

(3.2.15)

and alike we know in Case 2) that,

$$\frac{1}{2} \| x^{(s_1)} - x^{(m)} \|_Q^2 \geq \delta_{d_1} (x^{(s_1)}) \geq \bar{\alpha}(P_Q P) (\Delta^{(s_1)})^\nu,$$

(3.2.16)
where \( \nu = 1 \) if \( A \) is linear \( \vec{\alpha} \)-qualified and \( \nu = 2 \) if \( A \) is quadratic \( \vec{\alpha} \)-qualified.

From (3.2.15) and Lemma [3.2.29] we conclude

\[
\begin{align*}
    f(x^{(n)}) - f(x^{(m)}) &\geq f(x^{(s_0)}) - f(x^{(m)}) \\
    &\geq \frac{1}{2(m - s_0)} \|x^{(s_0)} - x^{(m)}\|_Q^2 \geq \frac{\tilde{\alpha}(\mathcal{P}_{QP})}{m - n} \left( \Delta^{(m)} \right)^\nu \\
    &> \frac{\tilde{\alpha}(\mathcal{P}_{QP})}{m - n} \lambda^\nu \left( \Delta^{(n)} \right)^\nu,
\end{align*}
\]

where the last inequality follows from the fact that we assumed \( \Delta^{(m)} > \lambda \Delta^{(n)} \). From (3.2.16) we conclude alike

\[
\begin{align*}
    f(x^{(n)}) - f(x^{(m)}) &\geq f(x^{(s_1)}) - f(x^{(m)}) \\
    &\geq \frac{1}{2(m - s_1)} \|x^{(s_1)} - x^{(m)}\|_Q^2 \geq \frac{\tilde{\alpha}(\mathcal{P}_{QP})}{m - n} \left( \Delta^{(s_1)} \right)^\nu,
\end{align*}
\]

and the claim follows, as above, from the fact that \( \Delta^{(s_1)} \geq \Delta^{(m)} > \lambda \Delta^{(n)} \).

A few comments are in place here:

**Remark 3.2.33:** One might think that using Lemma [3.1.13] as basis to derive a progress is vain hope as it may well be that a selected direction is a direction \( d \in \ker Q \) and therefore \( \|x^{(n)} - x^{(n+1)}\|_Q = 0 \). Although this is true, we derive our progress exactly by only considering directions, that end a locking sequence. They are (by definition) not directions in the kernel as already stated in Remark [3.2.33].

On the other hand one might hope for a large guaranteed advance for such directions \( d \in \ker Q \) as they promise an infinite advance \( \delta_d(x) = \infty \) for any \( x \in \mathcal{R}(\mathcal{P}_{QP}) \), where \( x \in S_d \). This however is spurious thinking as well, as they could potentially be reselected over and over again, only leading to infinitesimal advance in each iteration selected. The astonishing circumstance is the following: The directions promising the largest advance are the ones we cannot use for our analysis. The only hope to circumvent their reselection is to bound the longest locking sequence \( L_A(\mathcal{P}_{QP}) \). In Section [3.2.4] we will see that this is possible for pairable problems.

The crucial convergence theorem can now be given:

**Theorem 3.2.34:** Given an instance \( \mathcal{P}_{QP} \) of Constrained Quadratic Programming and an aggressive \( \vec{\alpha} \)-qualified decomposition algorithm \( A \), such that \( L := L_A(\mathcal{P}_{QP}) < \infty \) is finite, then \( A \) is within \( \varepsilon > 0 \) of optimality after at most

\[
    n_1 := \left\lceil \frac{L^2}{\tilde{\alpha}(\mathcal{P}_{QP}) + L} \ln \frac{\Delta^{(0)}}{\varepsilon} \right\rceil
\]
Chapter 3. Decomposition Techniques for Quadratic Programming

iterations if it is linear \( \vec{\alpha} \)-qualified and after at most
\[
n_2 := \left[ \frac{4L^2}{\alpha(P_{QP})\varepsilon} \right] + \max \left\{ 0, \left[ 2L \ln \frac{\Delta^{(0)}(P_{QP})}{2L} \right] \right\}.
\]
iterations if it is quadratic \( \vec{\alpha} \)-qualified.

**Proof.** According to Remark 3.2.31 we denote with \( n_r \) the iteration, where the \( r \)-th locking sequence starts and use the abbreviation \( \alpha := \hat{\alpha}(P_{QP}) \).

Let us first consider the case, where \( A \) is linear \( \vec{\alpha} \)-qualified. Note that for all maximal locking sequences \( r \geq 1 \) we have \( M_{n_r-1,n_r} \geq \frac{\alpha}{L} > 0 \). We now conclude from Lemma 3.2.32 that
\[
\Delta^{(n_r)} \leq \frac{1}{1 + M_{n_r-1,n_r}} \Delta^{(n_{r-1})} \leq \left( 1 - \frac{\alpha}{L + \alpha} \right) \Delta^{(n_{r-1})}
\]
and therefore by Lemma 1.3.1 we are within \( \varepsilon \) of optimality after at most
\[
r_1 = \left\lfloor \frac{L + \alpha}{\alpha} \ln \frac{\Delta^{(0)}}{\varepsilon} \right\rfloor.
\]
maximal locking sequences. As the length of each locking sequence is bounded by \( L \) the claim follows.

If \( A \) is quadratic \( \vec{\alpha} \)-qualified we will give a two phase analysis. During the first phase we assume \( \Delta^{(n_r)} > \frac{2L}{\alpha} \). The first maximal locking sequence of the second phase will therefore be the first sequence \( r \), such that \( \Delta^{(n_r)} \leq \frac{2L}{\alpha} \).

In Phase 1 we have \( \Delta^{(n_r)} > \frac{2L}{\alpha} \geq \frac{n_r-n_{r-1}}{\alpha} \) and we can conclude from Lemma 3.2.32 that we shrink during each sequence by a factor of \( \frac{1}{2} \), i.e.
\[
\Delta^{(n_r)} \leq \frac{1}{2} \Delta^{(n_{r-1})}.
\]
According to Lemma 1.3.1 Phase 1 consequently has ended after at most \( r' \) sequences where
\[
r' := \max \left\{ 0, \left[ 2\ln \frac{\Delta^{(0)}}{\alpha} \right] \right\}.
\]
For this analysis we did silently assume that \( \varepsilon < \frac{2L}{\alpha} \). If this should not be the case our procedure already ends during Phase 1, which would be to our advantage, as the whole algorithm would end after
\[
\max \left\{ 0, \left[ 2\ln \frac{\Delta^{(0)}}{\varepsilon} \right] \right\} \leq r'
\]
maximal locking sequences.
In Phase 2 the distance to optimality can be bounded as follows:

\[ \Delta^{(n_r+1)} \leq \left(1 - \frac{\alpha}{4L} \Delta^{(n_r)}\right) \Delta^{(n_r)}. \]

If \( \Delta^{(n_r)} \leq \frac{2(n_r-n_{r-1})}{\alpha} \) this follows directly from Lemma 3.2.32. If this inequality doesn’t hold, simply note that in Phase 2 we always have \( \frac{1}{2} \leq 1 - \frac{\alpha}{4L} \Delta^{(n_r)} \) and our bound is looser than the one in Lemma 3.2.32. By Lemma 1.3.2 we conclude that we are within \( \varepsilon \) of optimality after at most

\[ r'' := \left\lceil \frac{4L}{\alpha\varepsilon} \right\rceil \]

maximal locking sequences in Phase 2. The overall number of locking sequences used can therefore be bounded by

\[ r_2 = \left\lceil \frac{4L}{\alpha\varepsilon} \right\rceil + \max\left\{ 0, \left\lceil 2 \ln \frac{\Delta(0)\alpha}{2L} \right\rceil \right\}. \]

The claim again follows from the fact that the number of iterations during one sequence is bound by \( L \). \( \square \)

Obviously the above convergence rate heavily depend on the bounded length of locking sequences. There is another important part of aggressive selection strategies which is influenced by this quantity. To understand the following reasoning remember that in well known SVM algorithms the selection problem \( \mathcal{P}_a(x) \) has not only be used as selection heuristics but as well as stopping criteria. In fact most SMO type algorithms stop if the value \( -\nabla f(x)^\top d_{ij} \) of an (approximate) violating pair in the sense of Lemma 3.2.22 is smaller than a predefined accuracy \( \varepsilon > 0 \). It is well known that this implies \( \Delta(x) \leq S_{\max} \varepsilon \) (see [Schölkopf and Smola, 2002]). This corresponds to a general stopping rule depending on \( \text{val} \mathcal{P}_a(x) \), which is for example suggested by combining Theorem 4.3.7 and Lemma 4.3.8 (see Chapter 4 for details). Note however that we might well have \( \Delta(x) \leq \varepsilon \) and \( \text{val} \mathcal{P}_a(x) \gg \varepsilon \) as the following simple example shows:

**Example 3.2.35:** Consider the following instance of BQO without any equality constraints:

\[ \min_x \frac{1}{2} x^\top \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x + \begin{pmatrix} 50 \\ 50 \end{pmatrix}^\top x \quad \text{s.t.} \quad 0 \leq x \leq e. \]

It is now easy to check that the optimal solution is \( \hat{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \) which achieves a value of \( f(\hat{x}) = 0 \). Given an \( \varepsilon \in (0, 1] \), obviously any \( x \), where \( x_1, x_2 \leq \frac{\varepsilon}{101} \) achieves an optimal value \( f(x) \leq \varepsilon \). However for any \( x > 0 \) we have \( \{ d \mid d \leq 0 \} \subseteq F(x) \) and therefore

\[ \max_d \left\{ -\nabla f(x)^\top d \ \middle| \ d \in F(x), \|d\|_1 \leq 1 \right\} > 50. \]
A careful inspection of Example 3.2.35 reveals, that for any such $0 < x \leq \frac{\varepsilon}{101}$, an aggressive decomposition algorithm would still terminate two iterations later. The following theorem shows that a similar behavior can be found in general for selection strategies with bounded length of locking sequences.

**Theorem 3.2.36:** Given a normalized pairable instance $\mathcal{P}_{QP}$ of BQO where $Q \neq 0$, an aggressive $\vec{\alpha}$-qualified $\vec{q}$-sparse decomposition algorithm $A$, such that $L := L_A(\mathcal{P}_{QP})$ is finite, and a fixed $\varepsilon > 0$. If ever the algorithm reaches an iteration $n$, such that

$$\Delta^{(n)} \leq \frac{\varepsilon^2}{2\lambda_{\max,q}(Q)L},$$

where $q = q(\mathcal{P}_{QP})$, we conclude there exists an iteration $m \leq n + L + 1$, where the selected $q$-sparse direction $d^{(m)} \in \mathbb{R}^\ell$ fulfills

$$-\nabla f \left( x^{(m)} \right)^\top d^{(m)} \leq \varepsilon.$$

**Proof.** We will prove the statement by contradiction and therefore assume that in any iteration $m \in \{ n, \ldots, n + L + 1 \}$ the selected $d^{(m)}$ fulfills

$$-\nabla f \left( x^{(m)} \right)^\top d^{(m)} > \varepsilon \geq \sqrt{\frac{2\lambda_{\max,q}(Q)L\Delta^{(n)}}{\delta_d \left( x^{(m)} \right)}}.$$  \hspace{1cm} (3.2.17)

We claim that in this case $x^{(n)}, \ldots, x^{(m)}$ is a locking sequence for any $m \geq n$ which leads to a contradiction for $m > n + L$ and the claim follows.

Let us prove the above claim by induction, where the induction basis is trivially fulfilled for $m = n$. Now, for sake of contradiction, assume $x^{(n)}, \ldots, x^{(m-1)}$ is a locking sequence and the direction $d = d^{(m)}$ ends it. This implies that there exists an $n' \in \{ n, \ldots, m - 1 \}$ such that $x^{(n')} \in S_d$ and $x^{(m)} \in N_d$, or that $x^{(n')} \in N_d$ and $x^{(m)} \in S_d$. In both cases we conclude by Lemma 3.2.29 that

$$f \left( x^{(n)} \right) - f \left( x^{(m)} \right) \geq f \left( x^{(n')} \right) - f \left( x^{(m)} \right) \geq \frac{1}{2(m-n')} \left\| x^{(m)} - x^{(n')} \right\|_Q^2.$$  \hspace{1cm} (3.2.17)

Lemma 3.2.28 and Theorem 3.2.5, together with the assumption (3.2.17), then imply

$$f \left( x^{(n)} \right) - f \left( x^{(m)} \right) \geq \frac{1}{(m-n')} \delta_d \left( x^{(m)} \right) \geq \frac{1}{2\lambda_{\max,q}(Q)(m-n')} \left( -\nabla f \left( x^{(m)} \right)^\top d \right)^2 > \Delta^{(n)}.$$  

This implies $f \left( x^{(m)} \right) < \hat{f}$ which is not possible. \hfill \square
Remark 3.2.37: Despite of the fact that \( \text{val} \, P_a(x) \) is not a meaningful upper bound on \( \Delta(x) \), this theorem ensures that we can nonetheless use selected directions based on \( P_a(x) \) as stopping criteria in aggressive decomposition algorithms with bounded length of locking sequences. This is due to the fact, that aggressive strategies tend to move indexes to the border and therefore reducing the free directions. The above theorem shows that one finally reaches a point where the gradient value \( -\nabla f(x)^\top d \) of an aggressively selected direction \( d \) becomes meaningful.

We will later on see (Lemma 4.3.8) that \( \text{val} \, P_a(x) \) can as well be used as primal-dual optimality criterion and is therefore a suitable stopping criteria for SVM optimization (compare Theorem 5.2.7).

3.2.4 Bounded Locking Sequences for Pairable Problems

Remember that for pairable instances \( P_{QP} \) we could derive 2-sparse aggressive selection strategies. However one of the most important features of such a pairing strategy is the fact that it allows to bound the length of locking sequences occurring during such algorithms. This result will be presented in the following section.

In what follows \( P_{QP} \) will always denote a normalized pairable instance of BQO and we will – having Definition 3.2.15 in mind – interchangeably talk of selected directions or selected pairs, whatever is more handy. Note that the analysis given in this section is given not only for approximate MVP selection according to 3.2.10 but more general to later cover advanced SVM selection strategies as second-order heuristics of Fan et al. [2005].

Let us start by introducing a main tool in our analysis:

**Definition 3.2.38:** Given a normalized pairable instance \( P_{QP} \) of BQO. Then any feasible \( x \in \mathcal{R}(P_{QP}) \) induces an ordering on all indexes \( i \in \{1, \ldots, \ell\} \) as follows:

\[
    i \succeq j \iff \left( -\frac{1}{c_i} \nabla f(x)_i \geq -\frac{1}{c_j} \nabla f(x)_j \text{ and } i \sim j \right).
\]

In addition we will use the notation \( i \succ j \), for the related strict ordering, if ever \( i \succeq j \) and \( -\frac{1}{c_i} \nabla f(x)_i < -\frac{1}{c_j} \nabla f(x)_j \).

Given a sequence of \( (x^{(n)})_{n \geq 0} \) of feasible solutions, we will denote the relations induced by \( x^{(n)} \) with \( \succeq_n \).

**Lemma 3.2.39:** Given a normalized pairable instance \( P_{QP} \) of BQO. For any \( x \in \mathcal{R}(P_{QP}) \) the induced ordering \( \succeq \) fulfills the following properties:

1. \( i \succ j \iff x \in \mathcal{S}_{ij} \)
2. \( \delta^C_{ij}(x) = 0 \land i \succ j \Rightarrow u_i = x_i \lor l_j = x_j \)
3) $x$ is not optimal for $P_{QP} \Rightarrow \exists i > j : u_i > x_i \land l_j < x_j$

Proof. As $i > j$ is equivalent to $-\nabla f(x)^T d_{ij} > 0$ the first claim follows. By claim 1) and Lemma 3.1.12 the second claim follows directly. The last claim follows from Lemma 3.1.4 and Theorem 3.2.17.

**Definition 3.2.40 (GVP Selection):** Given a normalized pairable instance $P_{QP}$ of BQO. Any pairwise selection strategy is called Generalized Violating Pair (GVP) Selection for $P_{QP}$ iff the selected two-sparse direction $d = d_{ij}$ for a given $x \in R(P_{QP})$ fulfills the following properties:

1) $(i, j) \in \{(i, j) \mid x_i < u_i$ and $x_j > l_j\}$

2) $\forall i : x_i < u_i \Rightarrow \hat{i} \succeq_n i$ and $^n j$.

3) $\hat{i} \succ_n \hat{j}$.

A decomposition algorithm $A$ is called a GVP-decomposition algorithm iff it uses a GVP selection in any iteration $n$.

The following result is quite obvious:

**Lemma 3.2.41:** Any decomposition algorithm $A$ using a selection strategy based on (3.2.10) is a GVP-decomposition algorithm. This holds especially for Algorithm $A.1$ and Algorithm $A.2$.

In the following we will consider how such a special selection strategy influences the behavior of our algorithm during locking sequences if ever we use a GVP selection strategy. This analysis will finally lead to the following upper bound on $L_A(P_{QP})$:

**Theorem 3.2.42:** Given a normalized pairable instance of $P_{QP}$ and a GVP-decomposition algorithm $A$. Then the longest locking sequence $L_A(P_{QP})$ of $P_{QP}$ under $A$ is upper bounded by $2\ell$.

In the following let us assume that $x^{(n)}, \ldots, x^{(m-1)}$ is a locking sequence occurring during the run of the algorithm. From Definition 3.2.30 of locking sequences we see that in the setting of pairable problems for any pair $(i, j)$ selected in an arbitrary iteration during the locking sequence we have $i \succ_s j$ for all $s \in \{n, \ldots, m - 1\}$. This fact heavily restricts the selectable pairs. To formalize this, let us introduce the following sets, which clearly depend on the locking sequence $\{n, \ldots, m - 1\}$ we have at hand:

$$P_+^i(s) := \left\{ i \left| x_i^{(s)} < u_i \land \exists j : \left( x_j^{(s)} > l_j \land \forall t = n, \ldots, m - 1 : i >_t j \right) \right\} \quad \text{and}$$

$$P_0^i(s) := \left\{ j \left| x_j^{(s)} > l_j \land \exists i : \left( x_i^{(s)} < u_i \land \forall t = n, \ldots, m - 1 : i >_t j \right) \right\} .$$

$^1$ One could as well always select the minimal index, i.e. we would have $\forall j : x_j > l_j \Rightarrow j \succeq_n j$. We stay with this formulation as it will later be consistent with the second-order SMO selection strategy by Fan et al. [2005] (see Chapter 5 for details).
We call indexes in $P_{0}^{↓}(s)$ up-selectable and indexes in $P_{0}^{↑}(s)$ down-selectable in iteration $s$. Again the notion up- and down-selectable depend on the locking sequence we are currently investigating.

Given any pair $(i,j)$, selected in an iteration $s \in \{n, \ldots, m-1\}$, we know that $(i,j) \in P_{0}^{↓}(s) \times P_{0}^{↑}(s)$. The sets we will be interested in, are the sets of candidates, which may be selectable in the rest of a locking period, defined as follows:

$$P_{↓}^{↓}(s) := \bigcup_{t=s}^{m-1} P_{0}^{↓}(t) \quad \text{and} \quad P_{↑}^{↑}(s) := \bigcup_{t=s}^{m-1} P_{0}^{↑}(t).$$

In addition we will often make use of down-only-candidates which are larger than any up-candidate and up-only-candidates, which are smaller than any down candidate.

**Definition 3.2.43:** Given a normalized pairable instance $\mathcal{P}_{\text{QP}}$ of BQO and a sequence of feasible points $x_{n} \in \mathcal{R}(\mathcal{P}_{\text{QP}}), n \geq 0$. Then define:

$$L_{↓}^{↓}(n) := L_{↓}^{↓}(x^{(n)}) := \left\{ i \mid x_{i}^{(n)} = l_{i} \land \forall j : x_{j}^{(n)} > l_{j} \Rightarrow j \succ_{s} i \right\}$$

$$L_{↑}^{↑}(n) := L_{↑}^{↑}(x^{(n)}) := \left\{ j \mid x_{j}^{(n)} = u_{j} \land \forall i : x_{i}^{(n)} < u_{i} \Rightarrow j \succ_{s} i \right\}$$

The first interesting consequence drawn from this definitions is the following:

**Lemma 3.2.44:** Given an arbitrary iteration $s \in \{n, \ldots, m-1\}$. Then for any iteration $t \in \{s, \ldots, m-1\}$ the following two claims hold:

$$L_{↓}^{↓}(s) \cap \left( P_{0}^{↓}(t) \cup P_{0}^{↑}(t) \right) = \emptyset \quad \text{and} \quad L_{↑}^{↑}(s) \cap \left( P_{0}^{↓}(t) \cup P_{0}^{↑}(t) \right) = \emptyset.$$ 

**Proof.** We will first give the proof for $L_{↓}^{↓}(s)$. Let us therefore assume that $i = \arg \max \{ i \mid x_{i} < u_{i} \}$ is a maximal up candidate in iteration $s$ with respect to $\succ_{s}$ and let us prove the claim by induction over $t = s, \ldots, m-1$.

The induction basis is given as follows: We obviously have $L_{↓}^{↓}(s) \cap P_{0}^{↓}(s) = \emptyset$. To show that $L_{↓}^{↓}(s) \cap P_{0}^{↑}(s) = \emptyset$ consider an arbitrary $j \in L_{↓}^{↓}(s)$. Let us now assume that $j \in P_{0}^{↑}(s)$ is a possible down candidate. Then there exists at least one up partner $i \in \{1, \ldots, \ell\}$ such that $i \succ_{s} j$ and $x_{i}^{(s)} < u_{i}$. But as $i \succ_{s} j \succ_{s} i$, this contradicts the maximality of $i$ and we conclude that such an up-partner cannot exist for any $j \in L_{↓}^{↓}(s)$.

Now let us assume that for all $n \leq t' < t$ the claim holds. That implies especially that no $j \in L_{↓}^{↓}(s)$ has ever been selected and we therefore still have $x_{j}^{(t)} = u_{j}$. This immediately implies $L_{↓}^{↓}(s) \cap P_{0}^{↓}(t) = \emptyset$. For the second claim we can again argue as above, as any possible up partner $i \in \{1, \ldots, \ell\}$ during a locking sequence – has to fulfill $i \succ_{s} j \succ_{s} i$. This implies, that
$i \in L^1(s)$ and therefore still $x_i^{(t)} = u_i$ which contradicts the assumption that $i$ could be a valid up-partner for $j$.

The second claim for $L^1(s)$ is symmetric, where we consider the minimal down candidate $j$ of iteration $s$ with respect to $\succeq_s$.

This leads to two simple consequences:

**Corollary 3.2.45:** Given an arbitrary iteration $s \in \{n, \ldots, m - 2\}$ during a locking sequence and let $i$ be the selected up candidate. Then $i \notin P^1(s + 1)$.

**Proof.** For any $t \in \{s + 1, \ldots, m - 1\}$ a valid up-partner of $i$ would be an index $j \in \{1, \ldots, \ell\}$, which at least fulfills $j \succ_s i$. This implies $j \in L^1(s)$ as otherwise $j$ would have been selected in iteration $s$. But according to Lemma 3.2.44 any index in $L^1(s)$ will never be up-selectable and the claim follows.

**Corollary 3.2.46:** Given an arbitrary iteration $s \in \{n, \ldots, m - 2\}$ during a locking sequence and let $i$ be the selected up candidate. Then, if $x_i^{(s + 1)} = u_i$, we have $i \notin P^1(s + 1) \cup P^1(s + 1)$.

**Proof.** From Corollary 3.2.45 we know that $i$ will never be selected as down-candidate again, so it will stay on the upper border and will not get selectable again during the locking period.

We will now show that the sets of selectable candidates constantly decrease during the locking sequence:

**Lemma 3.2.47:** For any iteration $s = n, \ldots, m - 2$ during a locking sequence we have either $|P^1(s + 1)| < |P^1(s)|$ or $|P^1(s + 1)| < |P^1(s)|$.

**Proof.** By definition we have $|P^1(s)| \leq |P^1(s + 1)|$ and $|P^1(s)| \leq |P^1(s + 1)|$.

Now consider an arbitrary iteration $s \in \{n, \ldots, m - 2\}$ and the selected up-candidate $i$. If ever $x_i^{(s + 1)} = u_i$, Corollary 3.2.46 implies that $i \notin P^1(s + 1)$ and therefore $|P^1(s)| < |P^1(s + 1)|$.

Otherwise, as $x_i^{(s + 1)}$ is an optimal solution of the constrained optimization along $d_{ij}$ Lemma 3.1.12 implies $\delta_{ij}(x_i^{(s + 1)}) = 0$ and we conclude by Lemma 3.2.39 that $x_j^{(s + 1)} = l_j$ and therefore $j \notin P_0^1(s + 1)$. Now $j$ could only get to $P_0^1(t)$ again for some $t > s$, if it would be selected as up-candidate in an iteration $s_0$, where $s < s_0 < t$. But from Corollary 3.2.45 this up-selection in iteration $s_0$ implies that $j \notin P^1(s_0 + 1)$. We conclude $j \notin P^1(s + 1)$ and the claim follows.

This leads to the following consequence, which implies Theorem 3.2.42.
Lemma 3.2.48: Given a normalized pairable instance $P_{QP}$ of BQO and a GVP-decomposition algorithm $\mathcal{A}$. For any locking sequence $x^{(n)}, \ldots, x^{(m-1)}$ we can then conclude

$$m - 1 - n \leq 2 \left( \ell - |L^1(n)| - |L^1(n)| \right).$$

Proof. Assume that $m - 1 - n > 2\ell - (|L^1(n)| + |L^1(n)|)$. As $L^1(n) \cap L^1(n) = \emptyset$ Lemma 3.2.44 implies that $|P^1(n)| \leq \ell - (|L^1(n)| + |L^1(n)|)$ and $|P^1(n)| \leq \ell - (|L^1(n)| + |L^1(n)|)$ and according to Lemma 3.2.47 in each iteration $s \in \{n, \ldots, m\}$ the sum $|P^1(s)| + |P^1(s)|$ decreases. We conclude that both sets are empty after at most $2 (\ell - |L^1(n)| - |L^1(n)|)$ iterations, which contradicts the definition of locking sequence. So $m - 1 - n \leq 2 (\ell - |L^1(n)| - |L^1(n)|)$, which concludes the proof.

Putting this together we get the following main theorem, which follows directly from Theorem 3.2.34 and Theorem 3.2.42:

Theorem 3.2.49: Given a normalized pairable instance $P_{QP}$ of BQO. Then any aggressive $\vec{\alpha}$-qualified decomposition algorithm $\mathcal{A}$ using a generalized violating pair selection strategy (according to Definition 3.2.40) is within $\varepsilon > 0$ of optimality after at most

$$n_1 := \left\lceil \frac{4\ell^2}{\vec{\alpha}(P_{QP})} + 2\ell \right\rceil \ln \frac{\Delta^{(0)}}{\varepsilon}$$

iterations if it is linear $\vec{\alpha}$-qualified and after at most

$$n_2 := \left\lceil \frac{16\ell^2}{\vec{\alpha}(P_{QP})/\varepsilon} \right\rceil + \max \left\{ 0, \left\lceil 4\ell \ln \left( \frac{\Delta^{(0)} \vec{\alpha}(P_{QP})}{4\ell} \right) \right\rceil \right\}.$$

iterations if it is quadratic $\vec{\alpha}$-qualified.

Especially for the approximate MVP selection according to Lemma 3.2.22 we can derive the following bound from Corollary 3.2.23:

Theorem 3.2.50: Given a normalized pairable instance $P_{QP}$ of BQO. Then any $\vec{q}$-sparse decomposition algorithm $\mathcal{A}$, which uses Algorithm A.1 or Algorithm A.2 as selection strategy, is within $\varepsilon > 0$ of optimality after at most

$$n_1 := \left\lceil \left( \frac{4\ell^3 S_{\text{max}}^2(P_{QP})}{C^2(P_{QP})} \right) \lambda_{\text{max}, \vec{q}}(P_{QP}) + 2\ell \right\rceil \ln \frac{\Delta^{(0)}}{\varepsilon}$$

iterations if $P_{QP}$ is strictly positive definite and after at most

$$n_2 := \left\lceil \frac{32\ell^4 S_{\text{max}}^2(P_{QP})}{C^2(P_{QP})} \lambda_{\text{max}, \vec{q}}(P_{QP}) (Q) \right\rceil + \max \left\{ 0, \left\lceil 4\ell \ln \left( \frac{C^2(P_{QP}) \Delta^{(0)}}{8\ell^3 S_{\text{max}}^2(P_{QP}) \lambda_{\text{max}, \vec{q}}(P_{QP}) (Q)} \right) \right\rceil \right\}.$$

67
iterations otherwise.

**Remark 3.2.51:** As already noted in Remark 3.2.16 the selection of a pair \((i, j)\) is equivalent to the selection of a direction \(d_{ij}\). Now simply note that the general selection strategies from Definition 3.2.40 (especially Algorithm A.1 and Algorithm A.2) are generalizations of SMO-type selection strategies implemented for example in libSVM or SVMTorch. The convergence proofs for this algorithms (compare Chapter 5) will therefore be direct corollaries from the above Theorem 3.2.50. Let us note however, that there is no straightforward way to extend this result to the working set selection implemented in SVM\textsuperscript{light}, as Theorem 3.2.42 depends directly on the specific selection of a direction. That means, for SVM\textsuperscript{light} we can obviously use the fact that in each working space \(U_{I,x}\) one can find an aggressive qualifying direction, but the theory presented here does not give a bound for the length of maximal locking sequences for such types of algorithms.

Let us finally draw a simple consequence from Theorem 3.2.36 and Theorem 3.2.42 which shows that the value of a maximum violating pair can be used as stopping criterion in aggressive GVP algorithms:

**Corollary 3.2.52:** Given a normalized pairable instance \(\mathcal{P}_{QP}\) of BQO where \(Q \neq 0\), an aggressive \(\hat{\alpha}\)-qualified decomposition algorithm \(\mathcal{A}\) using a generalized violating pair selection strategy (according to Definition 3.2.40) and a fixed \(\varepsilon > 0\). If ever the algorithm reaches an iteration \(n\), such that

\[
\Delta(n) \leq \frac{\varepsilon^2}{4\lambda_{\text{max},2}(Q)(\ell - |L^{\uparrow}(n)| - |L^{\downarrow}(n)|)},
\]

we conclude there exists an iteration \(m \leq n + L + 1\), where the selected pair \(\hat{i}, \hat{j}\) fulfills

\[-\nabla f \left(x^{(m)}\right)^\top d_{\hat{i}\hat{j}} \leq \varepsilon.\]

**Proof.** A careful inspection of the proof of Theorem 3.2.36 reveals that one may replace the upper bound \(L = 2\ell\) by an upper bound on the length of locking sequences starting at \(n\). The claim then follows from Lemma 3.2.48.

### 3.3 Conservative Decomposition Algorithms

In the following section we will be concerned with conservative decomposition algorithms. Most of the results presented here can already be found in List and Simon [2005, 2007] and List [2007]. In this publications the authors did generalize the ideas from Hush and Scovel [2003], which have been the first to give SVM optimization algorithms with guaranteed runtime. Such methods have traditionally been called certifying pair/set algorithms.
We will show that almost every certifying pair/set algorithm, which has been applied to SVM optimization, is a conservative decomposition algorithm in the sense of Definition 3.1.7 and 3.1.16. We did choose to deviate from the established notation for two reasons: First we wanted to clarify the difference to aggressive selection strategies discussed in the last section and second our notation has the advantage that the convergence proof for such methods is nearly trivial and the only problem which has to be answered is the existence of sparse qualifying directions.

We will pursue this goal as follows: In Section 3.3.1 we will state the simple convergence properties of conservative decomposition algorithms. In Section 3.3.2 we will show that one can efficiently find sparse conservative qualified directions, which are shown to correspond to minimal certifying sets (see Remark 3.3.6).

### 3.3.1 Convergence Rates

In the following we will shortly give convergence rates for conservative qualified decomposition algorithms:

**Theorem 3.3.1**: Given an instance $P_{QP}$ of Constrained Quadratic Programming and a conservative $\tilde{\alpha}$-qualified algorithm $A$. Then $A$ is within $\varepsilon > 0$ of optimality after at most

$$n_1 := \max \left\{ 0, \left\lceil \frac{1}{\tilde{\alpha}(P_{QP})} \ln \frac{\Delta(0)}{\varepsilon} \right\rceil \right\}$$

iterations, if it is linear $\tilde{\alpha}$-qualified, and after at most

$$n_2 := \left\lceil \frac{1}{\tilde{\alpha}(P_{QP})\varepsilon} \right\rceil$$

iterations, if it is quadratic $\tilde{\alpha}$-qualified.

**Proof.** Simply note that for any iteration $n \geq 0$, where we choose $d^{(n)}$ as optimization direction, we have by Definition 3.1.15 and the update rule of Algorithm 3.1.1 that

$$f(x^{(n)}) - f(x^{(n+1)}) = \delta^{C}_{d^{(n)}}(x^{(n)}) \geq \tilde{\alpha}(P_{QP}) \begin{cases} \Delta^{(n)} & \text{if } A \text{ linear qualified or} \\ (\Delta^{(n)})^2 & \text{if } A \text{ quadratic qualified}. \end{cases}$$

By Lemma 1.3.1 and Lemma 1.3.2 the claim follows. \qed

Note that this convergence proof is astonishingly simple, because the definition of conservative qualifying directions has been design to make it simple.
Chapter 3. Decomposition Techniques for Quadratic Programming

The question remaining is, whether there exist such favorable directions and – if yes – whether we can compute them efficiently. Before we pursue this goal let us point out that exactly the ease of the convergence proof was the reason to introduce certifying pairs [Hush and Scovel, 2003] and their generalization certifying sets [List and Simon, 2005, 2007]. This fact is only slightly obscured by the fact that the authors did not present their decomposition algorithms as line search methods, but as sub-domain algorithms. In fact all used certifying set/pair algorithms select a one dimensional subspace, which directly corresponds to our selected direction (compare to Remark 3.3.6 and comments at end of Section 3.1.1).

3.3.2 General Conservative Selection Strategies

To motivate conservative selection strategies let us recall the “optimal” selection strategy introduced in Lemma 3.2.1, where we would select

\[
d = \arg \max \left\{ -\nabla f(x)^T d \mid d \in F(x), \|d\|_Q \leq 1 \right\}.
\]

Again we will adapt the restriction \( \|d\|_Q \leq 1 \) by a different bound leading to sparse optimization directions. In addition, we will however keep in mind that the optimal solution should be restricted to \( \mathcal{R}(\mathcal{P}_{QP}) \). To this end, given an \( x \in \mathbb{R}^\ell \) let us define the distance to the border to be

\[
\mu^+ (x)_i := r_i - x_i \in \mathbb{R} \quad \text{and} \quad \mu^- (x)_i := x_i - l_i \in \mathbb{R}. \quad (3.3.1)
\]

Note that for any \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) we immediately have \( \mu^+_i(x) \geq 0 \) and \( x_i = l_i \Leftrightarrow \mu^-_i(x) = 0 \) and \( x_i = r_i \Leftrightarrow \mu^+_i(x) = 0 \). Using this border distance let us define the following optimization problem, whose optimal value is traditionally called \( \sigma \)-gap (see for example [Hush and Scovel, 2003, List and Simon, 2005, 2007, Hush et al., 2006]: see as well Definition 4.3.3):

**Problem 3.3.2:** Given an instance \( \mathcal{P}_{QP} \) of BQO and a feasible \( x \in \mathcal{R}(\mathcal{P}_{QP}) \). The following linear optimization problem \( \mathcal{P}_\sigma(x) \) will be called \( \sigma \)-problem:

\[
\max -\nabla f(x)^T d \quad \text{s.t.} \quad Ad = 0, \quad -\mu^-(x) \leq d \leq \mu^+(x).
\]

Following [Hush and Scovel, 2003, List and Simon, 2005, 2007] we will denote its optimal value by \( \sigma(x) := \text{val} \mathcal{P}_\sigma(x) \).

Often we will use the restriction of \( \mathcal{P}_\sigma(x) \) to a subset \( I \subseteq \{1, \ldots, \ell\} \). Therefore the following linear optimization problem \( \mathcal{P}_\sigma(x|I) \) will be called restricted \( \sigma \)-problem:

\[
\max -\nabla f(x)^T_I d \quad \text{s.t.} \quad A_I d = 0, \quad -\mu^-(x)_I \leq d \leq \mu^+(x)_I.
\]

Its optimal value will be denoted by \( \sigma(x|I) := \text{val} \mathcal{P}_\sigma(x|I) \).
Note that any feasible \( d \in \mathcal{R}(\mathcal{P}_\sigma(x)) \) is a free direction at \( x \), i.e. \( \mathcal{R}(\mathcal{P}_{QP}) \subseteq \mathcal{F}(x) \). As shown in the next lemma an optimal \( d = \text{opt} \mathcal{P}_\sigma(x) \) is a conservative qualified direction.

**Lemma 3.3.3:** Given an instance \( \mathcal{P}_{QP} \) of BQO and a fixed \( \tau > 0 \) and \( q \in \{1, \ldots, \ell\} \). Then for any feasible \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) and any \( q \)-sparse solution \( \hat{d} \in \mathcal{R}(\mathcal{P}_\sigma(x)) \), such that

\[
-\nabla f(x)^\top \hat{d} \geq \tau \text{val}(x) = \tau \sigma(x),
\]

we conclude that

\[
\delta^c_d(x) \geq \begin{cases} 
\frac{\tau}{2} \Delta(x) & \text{if } \tau \Delta(x) > q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}) \\
\frac{\tau^2}{2 q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP})} (\Delta(x))^2 & \text{if } \tau \Delta(x) \leq q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}).
\end{cases}
\]

**Proof.** Note that for any optimal solution \( \hat{x} \) of \( \mathcal{P}_{QP} \) we have \( \hat{x} - x \in \mathcal{R}(\mathcal{P}_\sigma) \)

we conclude

\[
\sigma(x) \geq -\nabla f(x)^\top (\hat{x} - x) \geq f(x) - f(\hat{x}) = \Delta(x). \tag{3.3.2}
\]

By the assumption that \( \hat{d} \) differs from 0 in at most \( q \) components, we can in addition bound \( \|\hat{d}\|_Q^2 \) as follows:

\[
\|\hat{d}\|_Q^2 \leq q \lambda_{\max,q}(Q) \|\hat{d}\|_\infty^2 \leq q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}). \tag{3.3.3}
\]

Let us now consider the line segment \( \{x + t\hat{d} \mid t \in [0, 1]\} \subseteq \mathcal{R}(\mathcal{P}_{QP}) \). Using the Taylor expansion of \( f(x + t\hat{d}) \) around \( x \) we get

\[
f(x) - f(x + t\hat{d}) = -t \nabla f(x)^\top \hat{d} - \frac{t^2}{2} \|\hat{d}\|_Q^2 \\
\geq t \tau \Delta(x) - \frac{t^2}{2} q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}) := \tilde{f}(t),
\]

where the last inequality follows from (3.3.2) and (3.3.3) and the assumed \( \tau \) optimality of \( d \). As \( \delta^c_d(x) \geq \tilde{f}(t) \) for any \( t \in [0, 1] \) we can bound

\[
\delta^c_d(x) \geq \max_{0 \leq t \leq 1} \tilde{f}(t).
\]

If \( \tau \Delta(x) > q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}) \) this maximum is achieved at the border \( t = 1 \) and therefore

\[
\delta^c_d(x) \geq \tilde{f}(1) > \frac{\tau}{2} \Delta(x).
\]

If \( \tau \Delta(x) \leq q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP}) \) the maximum is achieved at

\[
t = \frac{\tau \sigma(x)}{q \lambda_{\max,q}(Q) S_{\max}^2(\mathcal{P}_{QP})}.
\]
and we conclude

\[ \delta^i_d(x) \geq \frac{\tau^2}{2q \lambda_{\text{max},q}(Q) S_{\text{max}}^2(\mathcal{P}_{QP})^2 (\Delta(x))^2} \]

\[ \square \]

The arguments given above can as well be found in the convergence proofs of List and Simon [2005, 2007]. The authors call sets

\[ I := \{ i \mid \hat{d}_i \neq 0 \} \]

corresponding to solutions \( \hat{d} \) as above, strongly \( \tau \)-certifying sets (see as well Remark 3.3.6).

A consequence of the Lemma 3.3.3 is that \( \mathcal{P}_{\sigma}(x) \) could be used as a selection procedure for conservative decomposition algorithms. However the selection by means of \( \mathcal{P}_{\sigma}(x) \) does not guarantee the selection of sparse directions. Let us therefore consider a technique to achieve sparse selection strategies in the general. As in the case of aggressive selection (cf. Problem 3.2.2) it is based on a 1-norm approximation. In this case we replace the box \( \mu^-(x) \leq d \leq \mu^+(x) \), which can be seen as a stretched 1-ball of the infinity norm \( \| \cdot \|_\infty \), by a stretched 1-ball of the 1-norm. This idea was first used by List and Simon [2005] in the context of certifying set selection strategies. It leads to the following sparse conservative gradient selection procedure:

**Problem 3.3.4:** Given an instance \( \mathcal{P}_{QP} \) of BQO and a feasible \( x \in \mathcal{R}(\mathcal{P}_{QP}) \). The following linear optimization problem \( \mathcal{P}_c(x) \) will be called conservative gradient selection problem:

\[
\begin{align*}
\max_{d^+, d^-} & \quad \begin{pmatrix} -\nabla f(x) \end{pmatrix}^\top \begin{pmatrix} d^+ \\ d^- \end{pmatrix} \\
\text{s.t.} & \quad \sum_{i: \mu_i^+(x) > 0} \frac{1}{\mu_i^+(x)} d_i^+ + \sum_{i: \mu_i^-(x) > 0} \frac{1}{\mu_i^-(x)} d_i^- \leq 1, \quad (3.3.4a) \\
& \quad (A - A) \begin{pmatrix} d^+ \\ d^- \end{pmatrix} = 0, \quad (3.3.4b) \\
& \quad d_i^+ d_i^- = 0, d_i^+/d_i^- \geq 0, \quad (3.3.4c) \\
& \quad \mu_i^+(x) = 0 \Rightarrow d_i^+ = 0, \quad \mu_i^-(x) = 0 \Rightarrow d_i^- = 0. \quad (3.3.4d)
\end{align*}
\]

The following theorem, in the context of SVM optimization first given by List and Simon [2005], will be crucial:

**Theorem 3.3.5:** Given an instance \( \mathcal{P}_{QP} \) of BQO and a feasible \( x \in \mathcal{R}(\mathcal{P}_{QP}) \). Then there always exists an optimal solution \( \hat{d} := \hat{d}^+ - \hat{d}^- \) of \( \mathcal{P}_c(x) \), such that \( \hat{d} \in \mathcal{R}(\mathcal{P}_\sigma(x)) \) and is \( k + 1 \)-sparse, where \( k \) is the rank of the equality constraint matrix \( A \), and

\[ -\nabla f(x)^\top \hat{d} \geq \frac{1}{\ell} \sigma(x). \]
This solution can be computed in linear time if we restrict the selection procedure to instances $\mathcal{P}_{QP}$, where the number of equality constraints is bounded by a constant $k_0$.

Proof. Recalling the arguments from Theorem 3.2.7 we see that $\mathcal{P}_c(x)$ is a LP in standard-form with $k + 1$ equality-constraints and therefore there exists a basic optimal solution $\hat{d}^+, \hat{d}^-$ with at most $k + 1$ non-zero entries.

To prove the second claim simply note that for any feasible point $d \in \mathcal{R}(\mathcal{P}_e)$ we can split $d = d^+ - d^-$, such that $d^+, d^- \geq 0$ and $d_i^+ d_i^- = 0$. We conclude that $\frac{1}{\ell} d = \frac{1}{\ell} d^+ - \frac{1}{\ell} d^-$ is a feasible solution of $\mathcal{P}_c(x)$ and the claim follows. On the other hand, if $d^+, d^-$ is a feasible solution of $\mathcal{P}_c(x)$ we can conclude that $d^+ - d^- \in \mathcal{R}(\mathcal{P}_e(x))$.

Again the linear time bound can be achieved by using the linear time LP algorithm of Megiddo [1984].

Remark 3.3.6: Note that the optimization problem $\mathcal{P}_c(x)$ has first been proposed by List and Simon [2005, 2007], where the authors did use its optimal feasible solution $\hat{d}$ as guideline to select a working set $I$. They proposed to select the smallest affine linear subspace spanned by $e_i, i \in I$ containing the direction $\hat{d}$. This is exactly achieved by choosing $I = \{i \mid \hat{d}_i \neq 0\}$ and we conclude $U_{I,x} = \{x + t \hat{d} \mid t \in \mathbb{R}\}$. Therefore the certifying set decomposition algorithm coincides with our direction based decomposition algorithm (compare to Remark 3.2.16 and comments at end of Section 3.1.1.

This leads to the following convergence rate for general conservative decomposition algorithm:

**Theorem 3.3.7:** Given an instance $\mathcal{P}_{QP}$ of BQO and a predefined accuracy $\varepsilon > 0$. Any decomposition algorithm $\mathcal{A}$ using a basic feasible solution $d^{(n)} = \text{opt} \mathcal{P}_c(x^{(n)})$ in any iteration $n \geq 0$ is within $\varepsilon$ of optimality after at most

$$n = \left\lceil \frac{2(k + 1)\ell^2 \lambda_{\max,k+1}(Q) S_{\max}^2(\mathcal{P}_{QP})}{\varepsilon} \right\rceil + \max \left\{ 0, \left\lceil 2\ell \ln \left( \frac{\Delta^{(0)}}{\ell(k + 1)\lambda_{\max,k+1}(Q) S_{\max}^2(\mathcal{P}_{QP})} \right) \right\rceil \right\}$$

iterations.

Proof. The claim follows again by giving a two phase analysis. We end the first phase ends at iteration $n$, if $\frac{1}{\ell} \Delta^{(n)} \leq (k + 1)\lambda_{\max,k+1}(Q) S_{\max}^2(\mathcal{P}_{QP})$ for the first time. During this phase Lemma 3.3.3 and Theorem 3.3.5 imply that $\mathcal{A}$ is a conservative linear $\alpha_1$-qualified algorithm with $\bar{\alpha}_1(\mathcal{P}_{QP}) = \frac{\Delta^{(0)}}{\ell(k + 1)\lambda_{\max,k+1}(Q) S_{\max}^2(\mathcal{P}_{QP})}$. By Theorem 3.3.4 we conclude that we are within $\ell(k + 1)\lambda_{\max,k+1}(Q) S_{\max}^2(\mathcal{P}_{QP})$
of optimality after at most
\[ n_1 = \max \left\{ 0, \left\lceil 2\ell \ln \left( \frac{\Delta^{(0)}}{(k+1)\ell \lambda_{\max,k+1}(Q) S_{\max}^2(P_{QP})} \right) \right\rceil \right\}, \]
which therefore is an upper bound of iterations taken in Phase 1. Again, we
did silently assume that \( \varepsilon < \ell (k+1) \lambda_{\max,k+1}(Q) S_{\max}^2(P_{QP}) \), as otherwise
the algorithm would terminate in Phase 1 after
\[ n_1' = \max \left\{ 0, \left\lceil 2\ell \ln \left( \frac{\Delta^{(0)}}{\varepsilon} \right) \right\rceil \right\} \leq n_1 \]
iterations.

In Phase 2, where \( \frac{1}{2} \Delta^{(n)} \leq (k+1) \lambda_{\max,k+1}(Q) S_{\max}^2(P_{QP}) \), we again
conclude from Lemma 3.3.3 and Theorem 3.3.5 that \( \mathcal{A} \) is a conservative
quadratic \( \vec{\alpha}_2 \)-qualified algorithm with
\[ \vec{\alpha}_2(P_{QP}) = \frac{1}{2\ell^2 (k+1) \lambda_{\max,k+1}(Q) S_{\max}^2(P_{QP})} \]
and therefore (using Theorem 3.3.1) is within \( \varepsilon \) of optimality after at most
\[ n_2 := \left\lceil \frac{2\ell^2 (k+1) \lambda_{\max,k+1}(Q) S_{\max}^2(P_{QP})}{\varepsilon} \right\rceil. \]
The claim follows by considering the joint iterations of both phases. \( \square \)

Before we start to consider simpler selection strategies, let us show, that
for any \( I \subseteq \{1, \ldots, \ell\} \) the value of \( P_{\sigma}(x|I) \) is of the same magnitude as
the current distance to optimality. Similar claims have already been proven
by [Hush et al. 2006, Lemma 14] (see as well Remark 5.2.3).

**Lemma 3.3.8:** Given an instance \( P_{QP} \) of BQO and an arbitrary subset
\( I \subseteq \{1, \ldots, \ell\} \) of size \( q := |I| \). Then for any \( x \in \mathcal{R}(P_{QP}) \), we conclude
\[ \Delta(x) \geq \begin{cases} \frac{1}{2} \sigma(x|I) & \text{if } \sigma(x|I) \geq q S_{\max}^2(P_{QP}) \lambda_{\max,q}(Q) \\ 2q S_{\max}^2(P_{QP}) \lambda_{\max,q}(Q) & \text{otherwise} \end{cases} \]

**Proof.** Let \( \tilde{d} := \text{opt } P_{\sigma}(x|I) \) denote the optimal solution of \( P_{\sigma}(x|I) \), which
diffs from 0 in at most \( q \) components. As in the proof of Lemma 3.3.3 we
therefore have
\[ f(x) - f(x + td) = -t \nabla f(x)^\top \tilde{d} - \frac{t^2}{2} \| \tilde{d} \|^2_Q \]
\[ \geq t \sigma(x|I) - \frac{t^2}{2} q S_{\max}^2(P_{QP}) \lambda_{\max,q}(Q). \]
Continuing the arguments from the proof of Lemma 3.3.3, we get
\[
\delta_d^C(x) \geq \begin{cases} 
\frac{1}{2} \sigma(x|I) & \text{if } \sigma(x|I) \geq qS_{\max}(P_{QP})\lambda_{\max,q}(Q) \\
\frac{2qS_{\max}(P_{QP})\lambda_{\max,q}(Q)}{\sigma(x|I)^2} & \text{otherwise}
\end{cases}
\]

As obviously \( \Delta(x) \geq \delta_d^C(x) \) the claim follows.

**3.3.3 Conservative Selection for Pairable Problems**

In the rest of this section we will show that – as in the case of aggressive selection – \( P_{\sigma}(x) \) has a two 2-sparse 1/\( \ell \)-optimal solution if \( P_{QP} \) is pairable, which has first been observed by [List 2007]. This solution can be computed in linear time by a generalization of the certifying pair selection algorithm introduced by [Simon 2004]. As in Section 3.2.2 we will exploit the structure of pairable instances \( P_{QP} \) of BQO. The following observation on \( \sigma(x|I) \) will be crucial:

**Lemma 3.3.9:** Given an instance \( P_{QP} \) of BQO, a feasible \( x \in \mathcal{R}(P_{QP}) \) and a subset \( I \subseteq \{1, \ldots, \ell\} \). Then \( P_{\sigma}(x|I) \) is equivalent to the following minimization problem:

\[
\sigma(x|I) = \inf_{\lambda} \sum_{i \in I} \mu_i^-(x) \left[ \nabla f(x)_i - A_i^T \lambda \right]^+ + \mu_i^+(x) \left[ A_i^T \lambda - \nabla f(x)_i \right]^+
\]

**Proof.** The proof is due to List and Simon [2007]. By LP-Duality one gets

\[
\begin{align*}
\sigma(x|I) = \inf_{\alpha, \beta \in \mathbb{R}^q} \left\{ \begin{bmatrix} \alpha \\beta \end{bmatrix}^T \begin{bmatrix} \mu_i^-(x) \\ \mu_i^+(x) \end{bmatrix} \left| \begin{array}{c}
A_i^T \lambda + \alpha - \beta = \nabla f(x)_I, \\
\alpha, \beta \geq 0
\end{array} \right. \right\} \\
\end{align*}
\]

It is easy to see that the optimal choice of \( \alpha \) and \( \beta \) is as follows:

\[
\alpha_i = \max\{0, \nabla f(x)_i - A_i^T \lambda\} \quad \text{and} \quad \beta_i = \max\{0, A_i^T \lambda - \nabla f(x)_i\}.
\]

Plugging these settings into the above equation proves the claim.

We will now show that this function can be simplified for pairable problems:

**Lemma 3.3.10:** Given a normalized pairable instance \( P_{QP} \) of BQO. Then for any feasible \( x \in \mathcal{R}(P) \) the following properties hold:

1) If \( I \subseteq [i_r] \) for an \( r \in \{1, \ldots, s\} \), then

\[
\sigma(x|I) = \inf_{\nu \in \mathbb{R}} \sum_{i \in I} \left\{ c_i \mu_i^-(x) \left[ \frac{1}{c_i} \nabla f(x)_i - \nu \right]^+ \\
+ c_i \mu_i^+(x) \left[ \nu - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right\}
\]

As obviously \( \Delta(x) \geq \delta_d^C(x) \) the claim follows. \( \square \)
2) For all $I \subseteq \{1, \ldots, \ell\}$

$$\sigma(x|I) = \sum_{r=1}^{s} \sigma(x|I \cap [i_r]) .$$

**Proof.** To prove the first claim let us rewrite $\sigma(x|I)$ by replacing each $A_i$ by $c_i a_r$:

$$\inf_{\lambda \in \mathbb{R}^k} \sum_{i \in I} \left\{ \mu_i^{-}(x) \left[ \nabla f(x)_i - A_i^\top \lambda \right]^+ + \mu_i^{+}(x) \left[ A_i^\top \lambda - \nabla f(x)_i \right]^+ \right\}$$

$$= \inf_{\lambda \in \mathbb{R}^k} \sum_{i \in I} \left\{ c_i \mu_i^{-}(x) \left[ \frac{1}{c_i} \nabla f(x)_i - a_r^\top \lambda \right]^+ + c_i \mu_i^{+}(x) \left[ a_r^\top \lambda - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right\}$$

$$= \inf_{\nu \in \mathbb{R}} \sum_{i \in I} \left\{ c_i \mu_i^{-}(x) \left[ \frac{1}{c_i} \nabla f(x)_i - \nu \right]^+ + c_i \mu_i^{+}(x) \left[ \nu - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right\} .$$

The second claim is proven with a similar calculation:

$$\sigma(x|I) = \inf_{\lambda \in \mathbb{R}^k} \sum_{r=1}^{s} \sum_{i \in I \cap [i_r]} \left\{ c_i \mu_i^{-}(x) \left[ \frac{1}{c_i} \nabla f(x)_i - a_r^\top \lambda \right]^+ + c_i \mu_i^{+}(x) \left[ a_r^\top \lambda - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right\}$$

$$\geq \sum_{r=1}^{s} \inf_{\beta \in \mathbb{R}} \sum_{i \in I \cap [i_r]} \left\{ c_i \mu_i^{-}(x) \left[ \frac{1}{c_i} \nabla f(x)_i - \beta_r \right]^+ + c_i \mu_i^{+}(x) \left[ \beta_r - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right\}$$

$$= \sum_{r=1}^{s} \sigma(x|I \cap [i_r]) .$$

As $\mathcal{P}_{QP}$ is assumed to be pairable and therefore by Lemma [3.2.12] rank $A = s$. Consequently $(a_1, \ldots, a_s)^\top \in \mathbb{R}^{s \times k}$ is surjective and one can find for any $\beta \in \mathbb{R}^s$ a $\lambda \in \mathbb{R}^k$ solving all equations $a_r^\top \lambda = \beta_r$ simultaneously. We conclude that the inequality holds in the reverse direction as well and the claim follows. □

The central result for pairable instances $\mathcal{P}_{QP}$ of BQO is as follows:

**Theorem 3.3.11:** Given a normalize pairable instance $\mathcal{P}_{QP}$ of BQO. Then for any feasible $x \in \mathcal{R}(\mathcal{P})$ there exists a $2$-sparse $\hat{d} \in \mathcal{R}(\mathcal{P}_\sigma(x))$ such that

$$-\nabla f(x)^\top \hat{d} \geq \frac{1}{\ell} \sigma(x) .$$
With Lemma 3.3.3 we can conclude that \( \hat{d} \) is a conservative linear \( \vec{\alpha}_1 \)-qualified direction if \( \Delta(x) > 2\lambda_{\max,2}(Q)S_{\max}^2(P_{QP}) \), where \( \alpha_1(P_{QP}) = \frac{1}{\pi} \). If \( \Delta(x) \leq 2\lambda_{\max,2}(Q)S_{\max}^2(P_{QP}) \), it is a conservative quadratic \( \vec{\alpha}_2 \)-qualified direction, where

\[
\alpha_2(P_{QP}) = \frac{1}{2\lambda_{\max,2}(Q)S_{\max}^2(P_{QP})}.
\]

**Proof.** Let \( q_r = |[i_r]| \) be the cardinality of the equivalence class \( i_r \). Note that \( P_{c}(x|[i_r]) \) is a lower dimensional instance of the same type as \( P_c(x) \) and we can apply Theorem 3.3.5. As it has equality constraint matrix of rank one we conclude that there exists a 2-sparse optimal solution \( d_r \) of \( P_{c}(x|[i_r]) \), which achieves a \( \frac{1}{q_r} \)-fraction of \( \sigma(x|[i_r]) \), i.e.

\[
-\nabla f(x)^\top d_r \geq \frac{1}{q_r} \sigma(x|[i_r]) .
\]

Now consider for each \( r \in \{1, \ldots, s\} \) the fractions \( 0 \leq \alpha_r \leq 1 \) of \( \sigma(x) \) which can be achieved by the restricted \( \sigma \)-problem: \( \alpha_r \sigma(x) = \sigma(x|[i_r]) \). This implies

\[
-\nabla f(x)^\top d_r \geq \frac{\alpha_r}{q_r} \sigma(x) .
\]

From Lemma 3.3.10 we conclude that \( \sum_{r=1}^{s} \alpha_r = 1 \). For sake of contradiction let us then assume that for each \( r \in \{1, \ldots, s\} \) we have \( \frac{\alpha_r}{q_r} < \frac{1}{\ell} \). As \( \sum_{r=1}^{s} q_r = \ell \), this would imply \( \ell \alpha_r < q_r \) and therefore

\[
\ell = \sum_{r=1}^{s} \ell \alpha_r < \sum_{r=1}^{s} q_r = \ell .
\]

Consequently, there exists at least one \( r \) such that

\[
-\nabla f(x)^\top d_r \geq \frac{\alpha_r}{q_r} \sigma(x) \geq \frac{1}{\ell} \sigma(x) .
\]

\[\square\]

The main idea to compute such a 2-sparse direction is now to scan all 2-restricted problems \( P_{c}(x|[i,j]) \). Simon [2004, 2007] did show that this can be done in linear time. We will repeat the prove in the following:

**Lemma 3.3.12:** For any \( i, j \in \{1, \ldots, \ell\} \), \( i \neq j \), the following holds: \( \sigma(x|i,j) > 0 \) iff \( i \sim j \) and \( \mu^*_i(x) > 0, \mu^*_j(x) > 0 \) and \( \frac{1}{c_j} \nabla f(x)_j > \frac{1}{c_i} \nabla f(x)_i \). If this is the case, we have

\[
\sigma(x|i,j) = \left( \frac{1}{c_j} \nabla f(x)_j - \frac{1}{c_i} \nabla f(x)_i \right) \min \left\{ c_i \mu^*_i(x), c_j \mu^*_j(x) \right\} .
\]
Chapter 3. Decomposition Techniques for Quadratic Programming

Proof. From Lemma 3.3.10 we see that
\[ \sigma(x|i, j) = \sigma(x|i) + \sigma(x|j) = 0 \]
if \( i \neq j \). Let us therefore assume that \( i \sim j \) and \( i, j \in [i_r] \). Again with Lemma 3.3.10 we have
\[
\sigma(x|i, j) = \inf_{\nu \in \mathbb{R}} \left\{ c_i \mu_i^- (x) \left[ \frac{1}{c_i} \nabla f(x)_i - \nu \right]^+ + c_i \mu_i^+ (x) \left[ \nu - \frac{1}{c_i} \nabla f(x)_i \right]^+ \right. \\
\left. + c_j \mu_j^- (x) \left[ \frac{1}{c_j} \nabla f(x)_j - \nu \right]^+ + c_j \mu_j^+ (x) \left[ \nu - \frac{1}{c_j} \nabla f(x)_j \right]^+ \right\}
\]
From this formulation we can see that \( \sigma(x|i, j) = 0 \) if either \( \mu_i^+(x) = \mu_j^-(x) = 0 \) or \( \mu_i^-(x) = \mu_j^+(x) = 0 \) or \( \frac{1}{c_i} \nabla f(x)_i = \frac{1}{c_j} \nabla f(x)_j \). In all other cases left we can assume \text{wlg} \( \frac{1}{c_i} \nabla f(x)_i > \frac{1}{c_j} \nabla f(x)_j \). For any \( \nu \), such that \( \frac{1}{c_i} \nabla f(x)_i \leq \nu \leq \frac{1}{c_j} \nabla f(x)_j \), the \( \sigma \)-function (depending on \( \nu \)) reads
\[
\sigma(\nu|i, j) = c_j \mu_j^- (x) \left[ \frac{1}{c_j} \nabla f(x)_j - \nu \right]^+ + c_i \mu_i^+ (x) \left[ \nu - \frac{1}{c_i} \nabla f(x)_i \right]^+ .
\]
It is therefore optimal to choose either \( \nu = \frac{1}{c_j} \nabla f(x)_j \) if \( c_i \mu_i^+(x) \leq c_j \mu_j^-(x) \) or \( \nu = \frac{1}{c_i} \nabla f(x)_i \), otherwise. This proves the claim. \( \square \)

Lemma 3.3.13: Let \([i_r], r = 1, \ldots, s\) be a set of representatives, then consider for any \( r \) the set \( M_r := \{ c_i \mu_i^+(x), c_i \mu_i^-(x) \mid i \in [i_r] \} \) and define
\[
\sigma_{\mu,r}(x) := \max \left\{ \frac{1}{c_j} \nabla f(x)_j \mid j \in [i_r] : c_j \mu_j^-(x) \geq \mu \right\} \\
- \min \left\{ \frac{1}{c_i} \nabla f(x)_i \mid i \in [i_r] : c_i \mu_i^+(x) \geq \mu \right\} .
\]
Then the following relation holds for each \( r = 1, \ldots, s \)
\[
\max_{\mu \in M_r} \sigma_{\mu,r}(x) \cdot \mu = \max \left\{ \sigma(x|i, j) \mid i \sim j, \mu_i^+(x) > 0, \mu_j^-(x) > 0 \right\} .
\]
Proof. The proof can be found in the work of Simon [2004, 2007] and List [2007]. \( \square \)

Following the ideas of Simon [2004, 2007], one can see, that the maximization problem \( \mathcal{P}_\sigma(x, r) \) can be solved in linear time:

Lemma 3.3.14: Given a normalized pairable instance \( \mathcal{P}_{QP} \) of BQO. Then Algorithm 4.3 computes a 2-sparse \( 1/\ell \) optimal direction for \( \mathcal{P}_\sigma(x) \) in linear time \( O(\ell) \). This time bound holds except for the first call, where the run time is \( O(\ell \log \ell) \).
The run time in the first iteration is dominated by the sorting of the list $M = \bigcup_{r=1}^{s} M_r$ of length $2\ell$ and can be done in $O(\ell \log \ell)$. In any subsequent iteration the reinsertion can be done in linear time. The run time of the main block is given by twice scanning the sorted list $M$, which can be done in $O(\ell)$.

According to Lemma 3.3.12 and Lemma 3.3.13 the algorithm therefore computes $\max_{i,j} \sigma(x|i,j)$. Consequently Theorem 3.3.11 guarantees that the associated direction is a $1/\ell$ optimal solution of $P_\sigma(x)$.

This leads to the following convergence theorem for pairwise conservative selection:

**Theorem 3.3.15:** Given an instance $P_{QP}$ of BQO and a predefined accuracy $\varepsilon > 0$. Any decomposition algorithm $A$ using Algorithm A.3 as selection strategy is within $\varepsilon$ of optimality after at most

$$n = \left\lceil \frac{4\ell^2 \lambda_{\max,2}(Q) S_{\max}^2(P_{QP})}{\varepsilon} \right\rceil + \max \left\{ 0, \left\lceil 2\ell \ln \left( \frac{\Delta^{(0)}}{2\ell \lambda_{\max,2}(Q) S_{\max}^2(P_{QP})} \right) \right\rceil \right\}$$

iterations.

**Proof.** Using the arguments from the proof of Theorem 3.3.7 the claim follows from Theorem 3.3.11. \qed
Chapter 4

Duality Gaps and Stopping Criteria

In this chapter we will consider an alternative approach to derive stopping criteria for dual algorithms. This work has mainly been conducted to overcome problems in typically analysis of SVM optimization algorithms, which admittedly could have been circumvented by giving a careful analysis in terms of usual primal-dual gaps. The account we will however present here is far more general and gives a deeper insight in the relation between certain primal and dual optimality guarantees. The results in this chapter are mostly joint work with Don Hush, Clint Scovel and Ingo Steinwart from Los Alamos National Laboratory and have been published in [List et al., 2007].

4.1 Introduction

To motivate the coming theory let us tell a wrong tale about the dual approach. One might be tempted to think as follows: If ever the dual approach succeeds, it is much more favorable to be only concerned with the dual problem which exhibits nice convexity properties. So take your favorite algorithm and compute a dual solution \( \hat{\lambda} \). Then use Theorem 1.2.14 and can compute a primal solution \( u \in U_\lambda \) where \( \lambda^\top c(u) = 0 \) and we are done. In fact this has often been the chain of arguments for the optimality of the computed classifier in SVM optimization.

This is however spurious thinking, as in most cases the computed \( \hat{\lambda} \) is only approximately optimal in the sense that

\[
\Delta_D(\hat{\lambda}) := \psi(\lambda) - \hat{\psi} \leq \varepsilon,
\]
where $\varepsilon$ is a predefined accuracy. This guarantee is however not sufficient to derive primal optimality guarantees for any $u \in U_\hat{\lambda}$. This is illustrated by the following counterexample:

**Example 4.1.1:** Consider the following simple instance of ACO

$$\sup_{u \in \mathbb{R}} |u| \text{ s.t. } u^2 - u \leq 0.$$  

The optimal solution is given by $\hat{u} = 1$ and the primal optimality distance for each $u \in \mathbb{R}$ is obviously $\geq 1$ for all feasible $u \neq 1$.

To check that the dual approach succeeds for this simple instance, consider the Lagrangian function, given by $L(u, \lambda) = |u| - \lambda(u^2 - u)$. By simple calculation we get for any $\lambda \in \mathbb{R}$

$$U_\lambda = \begin{cases} \emptyset & \text{if } \lambda \leq 0 \\ \{1\} & \text{if } \lambda \in (0, 4) \\ \{1, \frac{1}{2}\} & \text{if } \lambda = 4 \\ \{\frac{1}{2}\} & \text{if } \lambda > 4 \end{cases}$$

In addition it is easy to check that each $\hat{\lambda} \in (0, 4]$ solves the dual approach as we can choose $u = 1 \in U_\hat{\lambda}$ where $c(1) = 0$.

However, if we consider the dual function

$$\psi(\lambda) := \sup_{u \in \mathbb{R}} L(u, \lambda) = \begin{cases} \infty & \text{if } \lambda \leq 0 \\ 1 & \text{if } \lambda \in (0, 4] \\ \frac{1}{\lambda} & \text{if } \lambda > 4 \end{cases}$$

we see that $\Delta_D(\lambda) = \psi(\lambda) - 1 \leq \varepsilon$ for each $\lambda \in (0, 4 + 4\varepsilon]$. Yet, for each approximately correct dual $\hat{\lambda} \in (4, 4 + 4\varepsilon]$ we have $U_{\hat{\lambda}} = \{\frac{1}{2}\}$ and therefore an associate primal solution $u_{\hat{\lambda}} = 1/2$ has primal optimality distance 1.

In the following sections we will show that this problem can be circumvented in a broad class of problems, if only one chooses the means of guaranteeing the approximate optimality of the dual solution $\hat{\lambda}$ in an appropriate manner. In addition we will present abstract optimality gaps, which allow us to concentrate on the simpler dual only and ignore the primal problem completely.

### 4.2 Abstract Duality Gaps

Let us reconsider the abstract optimization problem $\mathcal{P}(\varphi, U, c)$ from Chapter 1.2

$$\inf \varphi(u) \text{ s.t. } u \in U, c(u) \leq 0, \quad (4.2.1)$$
where \( U \) is a nonempty set and \( \varphi : U \to \mathbb{R} \) and \( c_i : U \to \mathbb{R}, i = 1, \ldots, m \) are real valued functions and the associated dual \( \mathcal{P}(\psi) \) is:

\[
\inf \psi(\lambda) \quad \text{s.t. } \lambda \geq 0.
\]

(4.2.2)

Throughout this chapter we will assume that the Lagrangian problem (4.2.1) is much simpler to solve than the primal problem (4.2.2) and that the dual approach (4.2.4) succeeds. We can therefore assume that \( \hat{\varphi} \) and \( \hat{\psi} \) exist and are equal (see Theorem 4.2.14). We will now mainly be concerned with the following function

\[
gap(u, \lambda) := \psi(\lambda) - \varphi(u),
\]

(4.2.3)

for arbitrary primal-dual pairs \((u, \lambda)\). The following lemma shows that it gives an upper-bound on the primal optimality of \( u \) as well as the dual optimality of \( \lambda \) for feasible pairs:

**Lemma 4.2.1:** Given a feasible primal-dual pair \((u, \lambda)\), where \( c(u) \leq 0 \) and \( \lambda \geq 0 \), then

\[
\Delta_P(u) := \hat{\varphi} - \varphi(u) \leq \gap(u, \lambda)
\]

\[
\Delta_D(u) := \psi(\lambda) - \hat{\psi} \leq \gap(u, \lambda)
\]

Proof. As by weak duality (see Theorem 4.2.12) we have \( \hat{\varphi} - \hat{\psi} \geq 0 \), which implies

\[
\Delta_P(u) + \Delta_D(\psi) \leq \psi(\lambda) - \varphi(u) = \gap(u, \lambda).
\]

For feasible \( u \) and \( \lambda \) we have \( \Delta_P(u) \geq 0 \) and \( \Delta_D(\lambda) \geq 0 \) and the claim follows.

In the following we will inspect \( \gap(u, \lambda) \) for semi-saddles. In this case we have a very simple representation:

**Lemma 4.2.2:** For all semi-saddles \((u, \lambda) \in U_\lambda \times \mathbb{R}^m\) we have

\[
gap(u, \lambda) = -\lambda \cdot c(u).
\]

Proof. We have \( \gap(u, \lambda) = \psi(\lambda) - \varphi(u) = L(u, \lambda) - \varphi(u) = -\lambda \cdot c(u) \).

Now note that for feasible \((u, \lambda) \in U_\lambda \times (\mathbb{R}^+)^m\) we have \( c(u) \leq 0 \) and \( \lambda \geq 0 \) and hence Lemma 4.2.2 shows that \( \gap(u, \lambda) = 0 \) is equivalent to the complementary slackness condition of Theorem 4.2.7. On the other hand one can use this result to bound the primal optimality for any \( u \in U_\lambda \) if ever \( \lambda \geq 0 \) is dual feasible. We will state this result in the following corollary, as we will use it later as main stopping criterion in our algorithms:
Corollary 4.2.3: Given a feasible \( \lambda \geq 0 \). Then for any primal feasible \( u \in U_\lambda \) we have:

\[
\Delta_P(u) \leq -\lambda^T c(u) \quad \text{and} \quad \Delta_D(\lambda) \leq -\lambda^T c(u).
\]

This relation can readily be used to judge the optimality of a calculated primal solution \( u \). Calculating this bound however involves the black-box to solve the associated Lagrangian optimization problem \([1.2.4]\).

In the following we will show that one can go beyond this relation and solely use the dual problem as a source of the quality for associated primal values \( u \in U_\lambda \). Consider therefore the following simple and natural optimality bounds:

**Definition 4.2.4 (Forward Gap):** The forward gap of a feasible \( u \in U \) is defined by

\[
\overrightarrow{G}(u) := \inf\{-\lambda^T c(u) \mid \lambda \geq 0, \ u \in U_\lambda\}.
\]

**Definition 4.2.5 (Backward Gap):** The backward gap of a feasible \( \lambda \) is defined by

\[
\overleftarrow{G}(\lambda) := \inf\{-\lambda^T c(u) \mid u \in U_\lambda, c(u) \leq 0\}.
\]

In what follows we will show that one can use the forward bound on the dual problem to judge the quality of primal solutions \( u \in U_\lambda \).

### 4.2.1 Forward Gap and Dual Optimality

Let us now focus on the dual problem with the tools from above in mind. To that end we consider the dual to be our new “primal” and write \([4.2.2]\) as a maximization problem by changing \( \psi \) to \(-\psi\). The corresponding Lagrangian is then

\[
L_D(\lambda, \mu) = -\psi(\lambda) + \mu^T \lambda.
\]

Since \( \psi \) is convex we observe that \( \lambda \in U_\mu := \arg \max_{\lambda \in \mathbb{R}^m} L_D(\lambda', \mu) \) if and only if \( 0 \in \partial_{\lambda'}(-L_D(\lambda, \mu)) = \partial \psi(\lambda) - \mu \) which occurs if and only if \( \mu \in \partial \psi(\lambda) \). In other words we have \( U_\mu = \{ \lambda \in \mathbb{R}^m \mid \mu \in \partial \psi(\lambda) \} \). Since this implies

\[
\{ \mu \geq 0 \mid \lambda \in U_\mu \} = \partial \psi(\lambda) \cap (\mathbb{R}^+)^m
\]

we see that the forward gap of \([4.2.2]\) can be computed by

\[
\overrightarrow{G}(\lambda) = \inf \{ \mu^T \lambda \mid \mu \in \partial \psi(\lambda), \ \mu \geq 0 \}
\]

This gap becomes trivial if \( \psi \) is differentiable at \( \lambda \) or is equivalent to solving a LP if \( \partial \psi(\lambda) \) is a polyhedra. We will see in Section \([4.3.1]\) that this forward gap of the dual is not only of theoretical interest, but is in fact used in analyzing dual SVM optimization problems.

The following two results establish important properties of \([4.2.6]\).

---

84
Lemma 4.2.6: Given a feasible \( \lambda \geq 0 \). Then \( \overline{G}(\lambda) \geq 0 \) and the minimum value \( \overline{G}(\lambda) \) in (4.2.6) is finite and attained iff \( \partial \psi(\lambda) \cap (\mathbb{R}^+)^m \neq \emptyset \).

Proof. Obviously \( \mu^T \lambda \geq 0 \) for \( \lambda, \mu \geq 0 \). If \( \partial \psi(\lambda) \cap (\mathbb{R}^+)^m = \emptyset \) the feasible region of (4.2.6) is empty and therefore \( \overline{G}(\lambda) = +\infty \).

In the second case the objective function \( \lambda^T \mu \) and the constraint set \{\( \mu \geq 0 \mid \lambda \in U_\mu \} \) have no direction of recession in common. Moreover \{\( \mu \geq 0 \mid \lambda \in U_\mu \} = \partial \psi(\lambda) \cap (\mathbb{R}^+)^m \) is closed and convex and hence we obtain the assertion by Rockafellar [1970] Theorem 27.3]. \( \square \)

Theorem 4.2.7: Given a dual feasible \( \lambda \geq 0 \). Then the following two statements hold:

1) If \( \overline{G}(\lambda) = 0 \), then \( \lambda \) is optimal for (4.2.2).

2) If \( \lambda \) is optimal for (4.2.2) and \( \text{ri}(\text{dom } \psi) \cap \text{ri}((\mathbb{R}^+)^m) \neq \emptyset \) then \( \overline{G}(\lambda) = 0 \), where \( \text{ri} M \) denotes the relative interior of a set \( M \).

Proof. The first assertion follows directly from Corollary 4.2.3. For the second suppose that \( \lambda \geq 0 \) is optimal for (4.2.2). We write (4.2.2) as an unconstrained maximization of the function \( -\psi(\lambda) - 1_{(\mathbb{R}^+)^m}(\lambda) \) where we note that for \( \lambda \geq 0 \) we have \( \partial 1_{(\mathbb{R}^+)^m}(\lambda) = \{ \mu \leq 0 \mid \lambda_i > 0 \Rightarrow \mu_i = 0 \} \). Since \( \lambda \geq 0 \) is optimal it follows that \( 0 \in \partial(\psi(\lambda) + 1_{(\mathbb{R}^+)^m}(\lambda)) \). However, by Rockafellar [1970] Thm. 23.8] the assumptions imply that \( \partial(\psi(\lambda) + 1_{(\mathbb{R}^+)^m}(\lambda)) = \partial \psi(\lambda) + \partial 1_{(\mathbb{R}^+)^m}(\lambda) \) so that we conclude that there exists a \( \mu \in \partial \psi(\lambda) \) such that \( \mu \geq 0 \) and \( \mu_i = 0 \) for all \( i \) such that \( \lambda_i > 0 \). This implies \( \overline{G}(\lambda) = 0 \). \( \square \)

4.2.2 Relation between the Gaps

Let us now return to the relation between primal and dual problem. Suppose we have a feasible dual variable \( \lambda \) for which we have an optimality guarantee given by the forward-gap on the dual, i.e. \( \Delta_D(\lambda) \leq \overline{G}(\lambda) \), and we ask for the best possible associated primal \( u \in U_\lambda \). Given the backward gap \( \overline{G}(\lambda) \) and Corollary 4.2.3 the answer is easy: The best possible primal \( u \in U_\lambda \) has an expected primal optimality distance of at most \( \Delta_P(u) = \varphi - \varphi(u) \leq \psi(\lambda) - \varphi(u) = \text{gap}(u, \lambda) \leq \overline{G}(\lambda) \). The main question therefore is, how \( \overline{G}(\lambda) \) and \( \overline{G}(\lambda) \) are related. The answers is tightly related to the answer given in Theorem 4.2.6.

Theorem 4.2.8: Let \( \lambda \geq 0 \) be a dual point for which the strict filling property holds. Then we have

\[
\overline{G}(\lambda) = \overline{G}(\lambda) .
\]

In addition, if \( \overline{G}(\lambda) \) is finite, there exists a feasible \( u' \in U_\lambda \) such that \(-\lambda^T c(u') = \overline{G}(\lambda)\). Moreover, \( u' \) is an optimal solution of

\[
\sup \{ \varphi(u) \mid u \in U_\lambda, c(u) \leq 0 \} .
\]
Chapter 4. Duality Gaps and Stopping Criteria

Proof. Since the strict filling property implies that the infima in (4.2.5) and (4.2.6) range over the same set, we obtain equality of the gaps. If \( G(\lambda) < +\infty \) Lemma 4.2.6 and the strict filling property then imply that there exists a feasible \( u' \in U_\lambda \) such that \( \overline{G}(\lambda) = -\lambda^T c(u') \). Moreover, for \( (u, \lambda) \in U_\lambda \times \mathbb{R}^m \) Lemma 4.2.2 shows \( \varphi(u) - \lambda^T c(u) = \psi(\lambda) \). Consequently, we see that for fixed \( \lambda \geq 0 \) maximizing \( \varphi \) is equivalent to minimizing \( -\lambda^T c(\cdot) \) and therefore \( \varphi(u') \) is also the maximal value \( \varphi \) attains on \( \{ u \in U_\lambda \mid c(u) \leq 0 \} \).

Let us summarize the achieved results by continuing (and correcting) the tale from the beginning of this chapter: Whenever you have a (probably difficult to solve) problem at hand, you can always use your favorite convex optimization algorithm on the dual problem if ever it only uses dual solutions for which the strict filling property holds. Then simply stop if \( \overline{G}(\lambda) \leq \varepsilon \) and you will have an approximately optimal \( \lambda \) such that there exists a \( u \in U_\lambda \) for which you know that \( \Delta P(u) \leq \varepsilon \). Moreover, if the minimization problem \( \overline{G}(\lambda) \) is simple to solve you can easily compute this \( \hat{u} \). Let us complete Example 4.1.1 with this knowledge:

**Example 4.2.9:** It is easy to check that for any \( \lambda \in \mathbb{R} \) we have

\[
\partial \psi(\lambda) = \begin{cases} 
\emptyset & \lambda < 0 \\
(-\infty, 0] & \lambda = 0 \\
\{0\} & \lambda \in (0, 4) \\
[0, \frac{1}{4}] & \lambda = 4 \\
\{\frac{1}{4}\} & \lambda > 4 
\end{cases}
\]

Obviously the strict filling property holds for all \( \lambda \notin \{0, 4\} \) and in fact, if our algorithm would have relied on \( \overline{G}(\lambda) \) as dual stopping criterion, it would have correctly predicted

\[
\overline{G}(\lambda) = \inf \{ \mu \lambda \mid \mu \geq 0, \mu \in \partial \psi(\lambda) \} > 1
\]

for all \( \lambda > 4 \).

The rest of the chapter will show how the introduced stopping criteria can be used in SVM optimization. In addition, we will make use of Theorem 4.2.8 as some of the implemented stopping criteria are genuinely dual optimality criteria, for which we will present derived primal optimality criteria.

### 4.3 Stopping Heuristics for SVM Optimization

We will now use the above theory to derive stopping criteria for the SVM optimization problems introduced in Section 2.2. The criteria we will derive will later help to judge the quality of existing SVM optimization algorithms. The application of them can be found in Chapter 5.
Let us start by giving a general account on such stopping criteria and later specialize them to our abstract regularized risk minimization problem \([2.2.1]\) which will lead to suitable stopping criteria for all decomposition algorithms presented in this work. To this end let us reconsider the forward and backward gaps for the special case of \([1.2.15]\), which are equal according to Theorem \([1.2.20]\) and \([4.2.8]\).

**Lemma 4.3.1:** Given an instance \(\mathcal{P}_{QP}\) of GQO, then the forward- and backward-gap on the dual problem \(\mathcal{P}_{QP}(\psi)\) are given by

\[
G_{QP}(\lambda) := \overrightarrow{G}(\lambda) = \overleftarrow{G}(\lambda) = \inf \left\{ \lambda^\top (\nabla \psi(\lambda) - Az) \mid z \in \ker Q, \nabla \psi(\lambda) - Az \geq 0 \right\}
\]

\[\text{(4.3.1)}\]

The quantity \(G_{QP}(\lambda)\) will be our main tool in deriving sensible primal-dual optimality guarantees. That it is suitable for our purpose directly follows from Corollary \([4.2.3]\) and the Definitions \([4.2.4]\) and \([4.2.5]\).

**Corollary 4.3.2:** Given an instance \(\mathcal{P}_{QP}\) of GQO and a dual feasible \(\lambda \geq 0\), \(\lambda \in \text{dom } \psi\) such that \(G_{QP}(\lambda) < \infty\). Let \(\hat{z}\) optimize the gap \(G_{QP}(\lambda)\) defined in \((4.3.1)\). Then \(\hat{u} := w - A^* \lambda + \hat{z}\) is a \(G_{QP}(\lambda)\)-optimal solution of the primal problem, i.e.

\[\Delta_P(\hat{u}) \leq G_{QP}(\lambda) \]

**General \(\sigma\)-gaps**

Before giving special optimality criteria for our abstract regularized risk minimization problem \([2.2.1]\), we will give a generalization of a different optimality bound, first introduced in the setting of SVM optimization by Hush and Scovel \([2003]\) and later on generalized by List and Simon \([2005, 2007]\): the \(\sigma\)-gap. This optimality gap is in fact used in any certifying pair/set algorithm which has been proposed for SVM optimization (see the work of Hush et al. \([2006]\) for an extensive account).

We will introduce this type of gap from a general perspective for any dual of general quadratic optimization as defined in Problem \([1.2.5]\). Let us give the following precise definition, which is a generalization of the \(\sigma\)-function introduced by Hush and Scovel \([2003]\):

**Definition 4.3.3:** Given an instance \(\mathcal{P}_{QP}\) of GQO and consider its dual problem \(\mathcal{P}_{QP}(\psi)\) as given in \([1.2.15]\). Then for any \(\lambda \in \text{dom } \psi\) and \(\lambda \geq 0\) we call

\[\sigma(\lambda) := \sup \left\{ \nabla \psi(\lambda)^\top (\lambda - \lambda') \mid \lambda' \in \text{dom } \psi, \lambda' \geq 0 \right\}
\]

\[\text{(4.3.2)}\]

the \(\sigma\)-function of \(\mathcal{P}(\psi)\) at \(\lambda\).
Chapter 4. Duality Gaps and Stopping Criteria

Using the fact that \( \nabla \psi(\lambda) \) is a subgradient of \( \psi \) at \( \lambda \) we can derive the following simple optimality guarantee for any \( \lambda \geq 0 \):

\[ \Delta_D(\lambda) = \psi(\lambda) - \hat{\psi} \leq \sigma(\lambda). \]

In addition \( \sigma(\lambda) = 0 \) iff \( \psi(\lambda) = \hat{\psi} \).

**Proof.** As \( \nabla \psi(\lambda) \) is a subgradient at \( \lambda \) we conclude for any \( \lambda' \in \mathbb{R}^k \):

\[ \psi(\lambda') \geq \psi(\lambda) + \nabla \psi(\lambda)^T(\lambda' - \lambda), \]

and therefore

\[
\sup \left\{ \psi(\lambda) - \psi(\lambda') \mid \lambda' \geq 0, \lambda' \in \text{dom } \psi \right\} \\
\leq \sup \left\{ \nabla \psi(\lambda)(\lambda - \lambda') \mid \lambda' \geq 0, \lambda' \in \text{dom } \psi \right\},
\]

which implies the claim.

The second claim simply follows from the first order optimality condition [Bertsekas 1995, Proposition 2.1.2]. \( \square \)

This lemma obviously proves that \( \sigma(\lambda) \) is a sensible dual optimality criterion. However, it has not been clear that it leads to primal optimality guarantees for \( u \in U_{\lambda} \) as defined in (1.2.12). The next lemma (first proven by List and Simon 2007 and List et al. 2007) will close this gap, by relating the \( \sigma \)-gap to the forward-/backward-gap \( G_{QP}(\lambda) \):

**Lemma 4.3.5:** Given an instance \( P \) of GQO. Then for any feasible \( \lambda \in \text{dom } \psi \) such that the forward gap on the dual problem \( G_{QP}(\lambda) < \infty \), we have

\[ \sigma(\lambda) = G_{QP}(\lambda). \]

**Proof.** Let \( P : \mathbb{R}^m \to A \ker Q \) denote the orthogonal projection onto \( A \ker Q \). Since the duality gap for linear programming is zero (cf. Hiriart-Urruty and Lemaréchal 1993b, Cor. XII.2.3.6) we have

\[
G_{QP}(\lambda) = \inf \left\{ \lambda \cdot (\nabla \psi(\lambda) - Az) \mid z \in \ker Q, \nabla \psi(\lambda) - Az \geq 0 \right\} \\
= -\sup \left\{ \lambda \cdot \eta \mid \eta \in \mathbb{R}^m, P\eta = 0, \eta \leq \nabla \psi(\lambda) \right\} + \lambda \cdot \nabla \psi(\lambda) \\
= -\inf \left\{ \mu \cdot \nabla \psi(\lambda) \mid \mu \geq 0, \nu \in \mathbb{R}^m, \mu + P\nu = \lambda \right\} + \lambda \cdot \nabla \psi(\lambda).
\]

Since \( (\lambda - \mu) = P\nu \) is equivalent \( \nu - A^*\mu \perp \ker Q \) the right hand is equivalent to the \( \sigma \)-gap defined in (4.3.2) and the claim follows. \( \square \)

Using this result, we can now give primal optimality guarantees for the \( \sigma \)-gap as well:
Corollary 4.3.6: Given an instance \( \mathcal{P} \) of GQO and a dual feasible \( \lambda \geq 0 \), \( \lambda \in \text{dom } \psi \) such that \( \sigma(\lambda) < \infty \). Let \( \hat{z} \) optimize the gap \( G_{QP}(\lambda) \) defined in (4.3.1). Then \( \hat{u} := w - A^* \lambda + \hat{z} \) is a \( \sigma(\lambda) \)-optimal solution of the primal problem, i.e.

\[
\Delta_P(\hat{u}) \leq \sigma(\lambda).
\]

4.3.1 Optimality Bounds for SVM Optimization

Let us now return to the abstract regularized risk minimization problem (2.2.1) defined in Chapter 2, which we will repeat here for convenience:

\[
\sup_{h,b,\rho,\xi} \left\{ -\frac{1}{2} \langle h, h \rangle_H + \begin{pmatrix} 0_H \\ \nu \\ -\Gamma \end{pmatrix}^\top \begin{pmatrix} h \\ b \\ \rho \end{pmatrix} - \begin{pmatrix} d \\ 0 \end{pmatrix}_{\mathbb{R}^\tilde{\ell}} \right\} \leq 0.
\]

(4.3.3)

As the gradient of the dual objective function \( \psi \) is given by

\[
\nabla \psi(x, \Gamma - x) = \begin{pmatrix} x^\top (A^H + d) \\ 0 \end{pmatrix} = \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix},
\]

the forward-backward gap (as defined in (4.3.1)) for any dual feasible \( (x, \Gamma - x) \) can be given as follows:

\[
\inf_{b,\rho,\xi} \begin{pmatrix} x \\ \Gamma - x \end{pmatrix}^\top \begin{pmatrix} \nabla f(x) - A^b b - A^\rho \rho + \xi \\ \xi \end{pmatrix} \geq 0.
\]

(4.3.4)

As for dual feasible \( x \), we have \( x \geq 0 \) and \( \Gamma - x \geq 0 \) it is best to choose \( \xi \) minimal, i.e.

\[
\xi_i = \max \left\{ 0, A_i^b b + A_i^\rho \rho - \nabla f(x)_i \right\} = \left[ A_i^b b + A_i^\rho \rho - \nabla f(x) \right]^+.
\]

Using the equality \( t = [t]^+ - [-t]^+ \) for any \( t \in \mathbb{R} \) we can rewrite the objective of (4.3.4):

\[
\begin{align*}
\begin{pmatrix} x \\ \Gamma - x \end{pmatrix}^\top \begin{pmatrix} \nabla f(x) - A^b b - A^\rho \rho + \xi \\ \xi \end{pmatrix} &= x^\top \left( \nabla f(x) - A^b b - A^\rho \rho \right) + \Gamma^\top \xi \\
&= \sum_{i=1}^{\tilde{\ell}} x_i \left[ \nabla f(x)_i - A_i^b b - A_i^\rho \rho \right]^+ + (\Gamma - x_i) \left[ A_i^b b + A_i^\rho \rho - \nabla f(x)_i \right]^+,
\end{align*}
\]
Consequently

\[ G_{SV}(x) := G_{QP}(x) \]

\[ = \inf \left\{ \sum_{i=1}^{\ell} x_i \left[ \nabla f(x)_i - A^b_i b - A^\rho_i \rho \right]^+ \right. \]

\[ + (\Gamma_i - x_i) \left[ A^b_i b + A^\rho_i \rho - \nabla f(x)_i \right]^+ \left| \begin{array}{c} \ell \left. \right| b, \rho \in \mathbb{R} \end{array} \right. \] \]

(4.3.5)

is an optimality gap for the abstract regularized risk minimization algorithm (2.2.1) as stated in the following theorem:

**Theorem 4.3.7:** Consider a dual instance \( \mathcal{P}_{QP} \) (see (2.2.6)) of the abstract regularized risk minimization problem (2.2.1) and a predefined accuracy \( \varepsilon > 0 \). Then for any dual feasible \( x \in \mathcal{R}(\mathcal{P}_{QP}) \) such that \( G_{SV}(x) \leq \varepsilon \) we conclude that \( x \) is \( \varepsilon \)-optimal, i.e. \( \Delta_D(x) \leq \varepsilon \). In addition, given \( \hat{b}, \hat{\rho} \) as minimizers of (4.3.5) and define \( h := x^\top A^H \) and \( \hat{\xi}_i := \left[ A^b_i \hat{b} + A^\rho_i \hat{\rho} - \nabla f(x)_i \right]^+ \), we conclude that \( (h, \hat{b}, \hat{\rho}, \hat{\xi}) \) is a \( \varepsilon \)-optimal solution of the primal regularized risk minimization problem

\[ R(h, \hat{b}, \hat{\rho}, \hat{\xi}) - \hat{\mathcal{R}} \leq \varepsilon. \]

**Proof.** The claim follows directly from the above calculations and Corollary 4.3.2. \( \square \)

The crucial point will now be that, if the dual (2.2.6) is a pairable instance of BQO, we can relate the optimal solution of our aggressive selection problem \( \mathcal{P}_a(x) \) to the optimality bound \( G_{SV}(x) \). In this case we can give an upper bound on \( G_{SV}(x) \) as stated in the following lemma:

**Lemma 4.3.8:** Assume that the dual (2.2.6) of an instance of abstract regularized risk minimization is a pairable instance of BQO. Then for any dual feasible \( x \), i.e. \( A^b x = 0 \), \( A^\rho x = \nu \) and \( 0 \leq x \leq \Gamma \), we have

\[ G_{SV}(x) \leq \|\Gamma\|_\infty \left( \ell - \left| L^1(x) \right| - \left| L^1(x) \right| \right) \text{val} \mathcal{P}_a(x), \]

where \( L^1(x) \) and \( L^1(x) \) as given in Definition 3.2.43.

**Proof.** Then, following the arguments of Lemma 3.3.10 we can rewrite
\[ G_{SV M}(x) = \sum_{r=1,\ldots,s} \inf_{\lambda^{(r)}} \left\{ \sum_{i \in \{r\}} c_i x_i \left[ \frac{1}{c_i} \nabla f(x)_i - \lambda^{(r)} \right]^+ \right\} \]

\[ + c_i (\Gamma_i - x_i) \left[ \lambda^{(r)} - \frac{1}{c_i} \nabla f(x)_i \right]^+ \}

\leq \|\Gamma\|_{\infty} \sum_{r=1,\ldots,s} \inf_{\lambda^{(r)}} \left\{ \sum_{i \in \{r\}} c_i \left[ \frac{1}{c_i} \nabla f(x)_i - \lambda^{(r)} \right]^+ \right\} \]

\[ + \sum_{i \in \{r\}} c_i \left[ \lambda^{(r)} - \frac{1}{c_i} \nabla f(x)_i \right]^+ \}

By reversing the proof of Lemma 3.3.9 we get for any \( r = 1, \ldots, s \) that

\[ \inf_{\lambda^{(r)}} \left\{ \sum_{i \in \{r\}} c_i \left[ \frac{1}{c_i} \nabla f(x)_i - \lambda^{(r)} \right]^+ \right\} \]

\[ = \inf \left\{ \sum_{i \in \{r\}} c_i \alpha_i^{(r)} + \sum_{i \in \{r\}} c_i \beta_i^{(r)} \right\} \]

\[ \left\{ \lambda^{(r)} + \alpha_i^{(r)} - \beta_i^{(r)} = \frac{1}{c_i} \nabla f(x)_i , \alpha_i^{(r)}, \beta_i^{(r)} \geq 0 \right\} \]

\[ = \sup \left\{ \sum_{i \in \{r\}} -\frac{1}{c_i} \nabla f(x)_i d_i' \right\} \]

\[ \left\{ \lambda^{(r)} \sum_{i \in \{r\}} d_i' = 0 , -c_i \leq d_i' \leq c_i \right\} \]

\[ x_i = 0 \Rightarrow d_i' \geq 0 , \]

\[ x_i = \Gamma_i \Rightarrow d_i' \leq 0 \}

Now note that using the variable substitution \( d_i' = c_i d_i \) and splitting \( d_i = d_i^+ - d_i^- \), where \( d_i^+, d_i^- \geq 0 \) and \( d_i^+ d_i^- = 0 \), we see that the optimization problem (4.3.7) is similar to \( P_a''(x, r, 1) \), where we did replace \( \|d\|_1 \leq 1 \) by \( \|d\|_{\infty} \leq 1 \) and we conclude that for any solution \( d \) of (4.3.7) \[ \frac{1}{\|v\|} d \] is a solution of \( P_a''(x, r, 1) \). This implies

\[ G_{SV M}(x) \leq \|\Gamma\|_{\infty} \sum_{r=1,\ldots,s} \|\{r\}\| \text{ val } P_a''(x, r, 1) \leq \|\Gamma\|_{\infty} \ell \max_{r=1,\ldots,s} \text{ val } P_a''(x, r, 1) . \]

By a careful inspection of the above formula one sees that locked up and down indexes \( L^1(x) \) and \( L^1(x) \), as given in Definition 3.2.43, never contribute to a gap – neither to (4.3.6) nor to \( P_a''(x, r, 1) \). We therefore see that the
deviation between the 1-norm and the ∞-norm of the different solutions \( d \) can (all in all) be bounded by \( \ell - |L^1(x)| - |L^1(x)| \). The claim then follows by Lemma 3.2.21.

As an immediate consequence of this bound, we see that the maximum violating pair selection according to Algorithm A.1 is not only a means of selecting aggressive descent directions (as shown in Section 3.2.2) but as well a suitable primal optimality criterion for decomposition algorithms. Let us state this fact in the following Lemma which is a generalization of the results for MVP-selection in List et al., 2007:

**Lemma 4.3.9:** Consider a dual instance \( P_{QP} \) (see \( 2.2.6 \)) of the abstract regularized risk minimization problem \( 2.2.1 \), which we assume to be pairable and a predefined accuracy \( \varepsilon > 0 \). Then for any \( x \) such that \( -\nabla f(x)^T d_{ij} \leq \varepsilon \), where \( i, j \) are the indexes selected by Algorithm A.1 we conclude that \( x \) is approximately dual optimal, i.e.

\[
\Delta_D(x) \leq \|\Gamma\|_\infty C^2(P_{QP})(\ell - L^\top(x) - L^1(x))\varepsilon.
\]

Define \( h := x^T A^H \), choose \((b, \rho)\), such that

\[-\nabla f(x)_j \leq A^H_i \rho + A^H_i b \leq -\nabla f(x)_i \]

for some \( i \in [i] = [j] \), and set \( \xi_i := [A^H_i b + A^H_i \rho - \nabla f(x)_i]^+ \). We conclude that \((h, b, \rho, \xi)\) is an approximately primal optimal solution of the primal regularized risk minimization problem, i.e.

\[
R(h, b, \rho, \xi) - \hat{R} \leq \|\Gamma\|_\infty C^2(P_{QP})(\ell - L^\top(x) - L^1(x))\varepsilon,
\]

where \( C(P_{QP}) \) is given in Definition 3.2.14.

**Proof.** According to Lemma 3.2.22 the gradient value fulfills \(-\nabla f(x)^T d_{ij} \geq C^2(P_{QP}) \text{val} P_a(x)\). As therefore \( \|\Gamma\|_\infty C^2(P_{QP})\varepsilon \geq G_{\text{SVM}}(x) \) is an upper bound on the complementary slackness for the feasible primal-dual pair \((h, b, \rho, \xi)\) and \((x, \Gamma - x)\) the claim follows with Corollary 4.2.3.

Note that such a choice of \( b, \rho \) is possible because the matrix of the representing columns is assumed to be surjective.

**Remark 4.3.10:** 1) The results presented in Lemma 4.3.9 are quite standard primal-dual arguments and do not need the above introduced theory. In a rather sketchy and less general version they can be found in the book by Schölkopf and Smola, 2002, Chapter 10.1, while in many other publications no primal optimality guarantees have been considered.
2) For the special choice of C-SVC this implies simply to choose a \( b \) between the gradient values \(-\nabla f(x)_j\) and \(-\nabla f(x)_i\), which form the most violating pair. The usual heuristic to average over all gradient values seems, at least from the viewpoint of this bound, needless effort.

3) In case we are looking for the best possible primal point we should better choose \((b, \rho)\) as minimizer of \((4.3.5)\) directly. This can be done efficiently (see Remark 4.3.13). Note that, in the case of MVP algorithms, we have to sort the indexes according to their distance to border first, which takes \(O(\ell \log \ell)\) time. This seems to be affordable as it has to be done only once at the end of the algorithm.

4) It is important to notice that Algorithm A.2 implements this stopping criteria as well. The guarantees given here are therefore transferable to decomposition algorithms using the second order selection strategy given by Fan et al. [2005].

4.3.2 Using \(\sigma\)-gaps as Stopping Criteria

Note that \((4.3.5)\) is exactly the representation of \(\sigma(x)\) for the special instance \((2.2.6)\) as given in Lemma 3.3.9 which is (in the light of Lemma 4.3.5) not very astonishing. We therefore recover a generalization of a well known optimality criterion in SVM optimization:

**Lemma 4.3.11:** Consider a dual instance \(\mathcal{P}_{QP}\) (see \((2.2.6)\)) of the abstract regularized risk minimization problem \((2.2.1)\) and a predefined accuracy \(\varepsilon > 0\). Then for any dual feasible \(x \in \mathcal{R}(\mathcal{P}_{QP})\) such that \(\sigma(x) \leq \varepsilon\) we conclude that \(x\) is \(\varepsilon\)-optimal, i.e. \(\Delta_D(x) \leq \varepsilon\). In addition, given \((\hat{b}, \hat{\rho})\) as minimizers of \((4.3.5)\) and define \(h := x^\top A^\mathcal{H}\) and \(\hat{\xi}_i := \left[A_i^\top \hat{b} + A_i^\top \hat{\rho} - \nabla f(x)_i\right]^+\), we conclude that \((h, \hat{b}, \hat{\rho}, \hat{\xi})\) is a \(\varepsilon\)-optimal solution of the primal regularized risk minimization problem

\[R(h, \hat{b}, \hat{\rho}, \hat{\xi}) - \hat{R} \leq \varepsilon.\]

**Proof.** The claim follows directly from the above calculations and Corollary 4.3.6. 

Note that therefore the \(\sigma\)-function \(\sigma(x) = \text{val} \mathcal{P}_{\sigma}(x)\) is not only a mean of selecting conservative descent directions (as shown in Section 3.3.2) but as well a suitable stopping criterion for any conservative decomposition algorithm. In particular, this means that one can use the gradient value of the selected direction directly as optimality criterion. Let us state this result as a simple corollary of Lemma 4.3.11. Note that it gives a justification of the well-known stopping rule for certifying pair SVM optimization algorithms used by Hush et al. [2006].
Corollary 4.3.12: Consider a dual instance $\mathcal{P}_{QP}$ (see (2.2.6)) of the abstract regularized risk minimization problem (2.2.1), which we assume to be pairable and a predefined accuracy $\varepsilon > 0$. Then for any $x$ such that $\sigma(x|i, j) \leq \varepsilon$, where $i, j$ are the indexes selected by Algorithm A.3, we conclude that $x$ is $\ell\varepsilon$-optimal, i.e. $\Delta_D(x) \leq \ell\varepsilon$. In addition, given $(b, \rho)$ as minimizers of (4.3.5) and define $h := x^\top A^\top$ and $\hat{\xi}_i := [A_i^b + A_i^\rho - \nabla f(x)_i]^+$, we conclude that $(h, \hat{b}, \hat{\rho}, \hat{\xi})$ is a $\ell\varepsilon$-optimal solution of the primal regularized risk minimization problem

$$R(h, \hat{b}, \hat{\rho}, \hat{\xi}) - \hat{R} \leq \ell\varepsilon.$$ 

Proof. According to Theorem 3.3.11 the selected indexes fulfill $\sigma(x|i, j) \geq \frac{1}{\ell} \sigma(x)$. As therefore $\ell\varepsilon \geq \ell\sigma(x|i, j) \geq \sigma(x)$ Lemma 4.3.11 implies the claim.

Remark 4.3.13: Note that according to Lemma 3.3.10 the solution of (4.3.5), restricted to one equivalence class, has the form of a threshold problem described by Simon [2004, 2007]. Under the assumption that the indexes are sorted according to their distance to border, this can be solved in linear time [Simon, 2004, Corollary 4.2]. We conclude that in the case of pairable dual $\mathcal{P}_{QP}$ optimal $b$ and $\rho$ are simply found by inverting the matrix $a_1, \ldots, a_s$ of the representing columns. As in SVM optimization typically $s \leq 2$ this is done in constant time.

Note that in the case of certifying pair algorithms we keep the indexes sorted anyway and no overhead is needed (see Algorithm A.3 for details).
In the following chapter we will apply the achieved results to SVM optimization. To this end, we will introduce a general decomposition algorithm in Section 2.2. This algorithm is uniformly applicable to any pairable dual instances (2.2.6) of the abstract regularized risk minimization problem (2.2.1) introduced in Chapter 2. In addition it is a natural extension of the well know SMO-style decomposition algorithms introduced by Platt [1999] (see the work of Keerthi et al. 2001 and Shevade et al. 2000 for improvements and extensions). The following Section 5.2 will be devoted to present the convergence guarantees for the problems introduced in Section 2.2. Using our general convergence results from Theorem 3.2.49 and Theorem 3.3.15 we will present concrete run time and accuracy bounds for the application of this algorithm to typical instances of SVM optimization. In Section 5.3 we conclude this chapter by discussing various aspects of the results presented so far. We will especially focus on properties of the algorithms which are not exactly reflected in the worst case bounds we will present. Namely these are the influence of the sparsity of the typical solution in SVM optimization and the influence of the box-constraints. In this last section we will be mainly concerned with the classification case (C-SVC and ν-SVC), but most remarks will hold for the other problem instances as well.

Before we start, let us give a few remarks: In the sequel we will often not only give upper bounds on the iterations a decomposition algorithm needs to solve a certain instance of SVM optimization but upper bounds on the number arithmetic operations such a procedure needs. All these upper bounds will hold for a random access machine model with unit cost for an arithmetic operation over the reals. In addition we will admittedly ignore numerical issues.
5.1 Decomposition Algorithms for SVM Optimization

Remember the definition of sparse decomposition algorithms for convex quadratic optimization problems (see Algorithm 3.1 and Definition 3.1.7). In the following we will flesh out this procedure for the special class of pairable problems according to Definition 3.2.10. As any pairable instance $\mathcal{P}_{QP}$ of BQO can be transformed into a normalized pairable instance (see Lemma 3.2.13) we will give the code only for normalized instances.

Throughout the calculation we will assume, that the following data structures and functions/methods are given:

- A function $Q : \{1, \ldots, \ell\} \times \{1, \ldots, \ell\} \to \mathbb{R}$, such that $Q(i, j)$ computes the entry $Q_{ij}$ of the positive semidefinite symmetric matrix $Q$ of problem $\mathcal{P}_{QP}$ in $E = O(1)$ steps,
- vectors $l, r, w \in \mathbb{R}^\ell$, $b \in \mathbb{R}^k$ and a matrix $A \in \mathbb{R}^{k \times \ell}$ encoding the parameters of $\mathcal{P}_{QP},$
- a vector of scalar factors $c \in \mathbb{R}^\ell$, scaling each row $A_i$ to its representative in equivalence class $[i]$ (see Section 3.2.2 for details),
- a mapping $[i_r] \ni i \mapsto r$ encoding the membership of each $i$ in an equivalence class $[i_r],$
- a vector $x \in \mathbb{R}^\ell$ of the optimization variables and
- a vector $g \in \mathbb{R}^\ell$ to save the gradient values $\nabla f(x)_1, \ldots, \nabla f(x)_\ell$ for the current solution $x.$

Throughout this exposition we will ignore two important technical details of SVM optimization algorithms: caching and shrinking. Both lead to significant speedup of SVM training but especially the influence of shrinking is challenging to quantify.

Using this data structures we will refine Algorithm 3.1 as given in Algorithm 5.1. Note, that the integral part of such an algorithm is encapsulated in the function $\text{boolean select}(x, g, i, j)$, which takes the $x$ and gradient values as input and initializes $i$ and $j$ if $x$ has not already achieved the requested $\varepsilon$-optimality. The function should return true if ever a working set $(i, j)$ has been initialized and false if optimality is reached (see Appendix A for details).

Let us start by giving a simple analysis of the main loop of Algorithm 5.1 in the following lemma:

**Lemma 5.1.1:** The main loop (without working set selection) of Algorithm 5.1 takes $\Theta(\ell)$ operations.
Algorithm 5.1: General SMO decomposition algorithm

Input: A normalized pairable instance \( P_{QP}(Q, w, A, b, l, r) \) of BQO, \( c \) scalar parameters for equivalence classes, \( \varepsilon > 0 \)

Output: \( x \)

/* Initialization */
1 for \( i = 1, \ldots, \ell \) do
2 \( x_i = 0; g_i = -w_i; \)

3 while select\((x, g, i, j)\) do /* Select WS if \( x \) suboptimal */
4 \( t = \frac{c_i g_j - c_j g_i}{c_i + c_j}; \) /* Calculate update step */
5 \( k = Q(i, i) + Q(j, j) - 2Q(i, j); \)
6 /* Clip to box */
7 if \( k = 0 \) then
8 \( t = \min\{r_i - x_i, l_j - x_j\}; \)
9 else
10 \( t = \min\{t/k, r_i - x_i, l_j - x_j\}; \)

/* Update \( x \) and gradient values */
11 \( x_i = x_i + t; x_j = x_j - t; \)
12 for \( k = 1, \ldots, \ell \) do
13 \( g_k = g_k + t(Q(j, k) - Q(i, k)); \)

14 return \( x \)

Proof. Obviously the most expensive operation is the gradient update, which can be done in by \( \Theta(\ell E) \) steps. Note, that if we assume to have a kernel cache and the working set selection evaluates at least \( Q(i, \cdot) \) and \( Q(j, \cdot) \) we spare the evaluation cost \( E \), which improves the constant.

As the gradient update needs \( \Theta(\ell) \) steps, one tries to find selection procedures which do not exceed this bound. In fact linear run time of the working set selection has been of the most important design criteria. The overall run time heavily depends on the selection and its convergence properties. We will use the results from Chapter 3 to give upper bounds on the run time of Algorithm 5.1 in the case of SVM optimization in the next section.

5.2 Convergence Bounds for SVM Optimization

Reconsider the abstract regularized risk minimization problem (2.2.1). In the following we will use Algorithm 5.1 to solve instances of the dual 2.2.6. We will however design our selection procedures in a way that the stopping criteria are sensible in the sense of Chapter 4 i.e. not only the resulting dual parameter \( x \) computed by Algorithm 5.1 is dual optimal, but also for the
associated linear functional $h := x^\top A^H$, there exist $(b, \rho, \xi)$, such that the primal variable $(h, b, \rho, \xi)$ is approximately primal optimal, i.e.

$$R(h, b, \rho, \xi) - \hat{R} \leq \varepsilon',$$

where $\varepsilon' = O(\varepsilon)$ for some predefined accuracy $\varepsilon > 0$.

To this end we will use the aggressive maximum violating pair selection strategy described in Lemma 3.2.22 and the derived second order heuristic, and the conservative certifying pair selection procedure according to Lemma 3.3.14 (see Appendix A for details).

### 5.2.1 Conservative Decomposition Algorithms

Let us first give convergence bounds for conservative algorithms. Note, that these bounds follow already from List and Simon [2005, 2007], although the authors did not give concrete run time bounds, as they didn’t analyze the influence of the stopping criteria used to guarantee primal optimality. Such an analysis is found for the first time by Hush et al. [2006], where the authors present bounds for the special case of $C$-SVC and an application of SVM to density level detection Steinwart et al. [2005] (see Remark 5.2.3 for comments on their result). Admittedly the presented results do not hold for software packages as for example Shark, which depends on maximum gain working set selection techniques Glasmachers and Igel [2006].

Throughout this section we will be concerned with the conservative decomposition algorithm, which is given by Algorithm 5.1 using the certifying pair selection strategy selectCP given in Algorithm A.3.

Let us start by giving a general convergence theorem for abstract regularized risk minimization:

**Theorem 5.2.1:** Given an instance of the abstract regularized risk minimization problem (2.2.1), such that the dual (2.2.6) is a normalized pairable instance of BQO and a predefined accuracy $\varepsilon > 0$. Assume, that $Q(i, j) := \langle A_i^b A_j^\rho \rangle$ is computable in constant time $E$, then Algorithm 5.1 using the selection procedure selectCP (Algorithm A.3), with input parameters

$$(Q, d, \left(\begin{array}{c} A^b \\ A^\rho \end{array}\right)^\top, \left(\begin{array}{c} 0 \\ 0 \end{array}\right), 0, \Gamma, c, \varepsilon, \ell)$$

stops after at most

$$O\left(\max\left\{\frac{\ell^4 U}{\varepsilon}, \frac{\ell^5 U^2}{\varepsilon^2}\right\}\right)$$

steps, where $U := \lambda_{\max,2}(Q) \|\Gamma\|_2^2$. Given the output variables $\hat{x}$ we define $\hat{h} := \hat{x}^\top A^H$. Then

$$R(\hat{h}, b, \rho, \xi) - \hat{R} \leq \varepsilon,$$

if $(b, \rho)$ are minimizers of $G_{SVM}(\hat{x})$ (4.3.5) and $\xi_i := \left[ A_i^b b + A_i^\rho \rho - \nabla f(\hat{x})_i \right]^+$. 

98
Proof. Theorem 3.3.15 implies, that
\[ \Delta_D(x) \leq \min \left\{ \frac{\varepsilon}{2\ell}, \frac{\varepsilon^2}{4U\ell^2} \right\} \]
after at most
\[ n = \left\lfloor \frac{4\ell^2 U}{\min \left\{ \frac{\varepsilon}{\ell}, \frac{\varepsilon^2}{4U\ell^2} \right\}} \right\rfloor + \max \left\{ 0, \left\lfloor 2\ell \ln \left( \frac{\Delta_D^{(0)}}{2U} \right) \right\rfloor \right\} \] iterations. Lemma 3.3.8 and Theorem 3.3.11 then imply that \( \xi \geq \sigma(x|i,j) \) and the algorithm therefore stops after at most \( n \) iterations. The overall run time follows with Lemma 5.1.1.

From Corollary 4.3.12 and the stopping criterion \( \sigma(x|i,j) \leq \xi \) we conclude, the primal optimality guarantees for \( (h,b,\rho,\xi) \).

In the following we will apply this result to the various instances of (2.2.1) introduced in Chapter 2.

Corollary 5.2.2: Given an instance \( P_C(k,\overline{x},\overline{y},C) \) of \( C \)-SVC, such that the evaluation of \( k(\cdot,\cdot) \) can be done in constant time \( E \), then Algorithm 5.1, using the selection procedure selectCP (Algorithm A.3), with input parameters
\[ \left( K, e^\top, 0, l, r, e, \frac{\varepsilon}{\ell} \right) \]
stops after at most
\[ O \left( \max \left\{ \frac{\ell^4 C^2 \lambda_{\max,2}(K)}{\varepsilon}, \frac{\ell^5 C^4 \lambda_{\max,2}(K)^2}{\varepsilon^2} \right\} \right) \]
steps. Given the output variables \( \hat{x} \) we define \( \hat{h} := \sum_{i=1}^{\ell} \hat{x}_i \Phi(x_i) \). Then
\[ R(h,b,\xi) - \hat{R} \leq \varepsilon, \]
if \( b \) is a minimizers of \( G_{\text{SVM}}(\hat{x}) \) and \( \xi_i := [b - \nabla f(\hat{x})]_i^+ \).

Proof. As the equality constraint of \( P_C \) is given by \( e^\top \) it is obviously a normalized pairable instance with \( c = e \) and one equivalence class. As the kernel matrix \( K(i,j) = k(x_i, x_j) \) can be evaluated in constant time the claim follows from Theorem 5.2.1.

Remark 5.2.3: Note that a similar bound has been given by [Hush et al. 2006, Example 1] where they derive an overall run time for \( C \)-SVC of \( O \left( \frac{\varepsilon}{\ell} \right) \). Let us briefly comment on the difference.

First of all note that they consider an instance of \( C \)-SVC which is a scaled version \( R_C = \frac{1}{\ell} R_C \) of the standard formulation we used (cf. Section 2.2.2).
In this case the term $\|\Gamma\|_\infty$ would read $\|\Gamma\|_\infty = \frac{C}{\ell}$ and $\lambda_{\max,2}(K) = \ell\lambda_{\max,2}(K)$ leads to the convergence rate

$$O\left(\max\left\{\frac{\ell^2 C^2}{\varepsilon}, \frac{\ell^4 C^4}{\varepsilon^2}\right\}\right).$$

(Note that the extra factor $1/\ell$ comes from the fact that we stop at $\varepsilon/\ell$.) Yet, we decided to chose the standard formulation of $C$-SVC and therefore the bound above is readily applicable to standard software packages, which typically use the representation given in (2.2.7). In addition, consider the universal consistency results for SVMs given by Steinwart [2002], where the author (under the assumption that $X$ is a compact subset of $\mathbb{R}^m$ and the used kernel is a Gaussian RBF kernel) proposes a choice of $C_\ell = \ell^{2-1}$ for $0 < \beta < \frac{1}{m}$. This means, given a high dimensional input space, our bound approaches the one given by Hush et al. [2006] in the sense that $\ell^4 C^2_\ell \to \ell^2$ for $m \to \infty$.

The second difference is due to the following fact: As already mentioned Hush et al. [2006] use an argument similar to Lemma 3.3.8 to justify the certifying pair stopping rule (Corollary 4.3.12). However their argument in the example given do not take into account that the dependence on $\varepsilon$ might be quadratic. Taking this into account the bound of Hush et al. [2006] would be given correctly $O(\max\{\ell^2/\varepsilon, \ell^4/\varepsilon^2\})$ which, considering the above argument, resembles the bound given in Corollary 5.2.2.

**Corollary 5.2.4:** Given an instance $P_\nu(k, \bar{x}, \bar{y}, \nu)$ of $\nu$-SVC, such that the evaluation of $k(\cdot, \cdot)$ can be done in constant time $E$, then Algorithm 5.1, using the selection procedure selectCP (Algorithm A.3), with input parameters

$$\left( K', 0, \left( \bar{y}^T \right), \left( \nu \right), 0, \frac{1}{\ell} \cdot e, e, \frac{\varepsilon}{\ell} \right)$$

stops after at most

$$O\left(\max\left\{\ell^2 \lambda_{\max,2}(K'), \ell^4 \lambda_{\max,2}(K')^2\right\}\right)$$

steps. Given the output variables $\hat{x}$ we define $\hat{h} := \sum_{i=1}^{\ell} \hat{x}_i y_i \Phi(x_i)$. Then

$$R(\hat{h}, b, \rho, \xi) - \hat{R} \leq \varepsilon,$$

if $(b, \rho)$ is a minimizers of $G_{SVM}(\hat{x})$ [4.3.5] and $\xi_i := \left[ \rho - y_i b - \nabla f(\hat{x})_i \right]^+ +$.

**Proof.** Again the equality constraints $\left( \bar{y}^T \right)$ are easily seen to induce two equivalence classes if there exits a label pair $y_i \neq y_j$ with representatives (1) and (−1), such that $c = e$. Otherwise there would be only one equivalence class and the equality constraints would be of rank one. Again we see that
Proof. Note, that we did transform
\[ K'(i,j) = y_i y_j k(x_i, x_j) \]
the problem instance is normalized pairable with \( e = e \). As the kernel matrix
\[ \kappa_{\ell}(i,j) = \gamma_i \gamma_j k(x_i, x_j) \]
can be evaluated in constant time the claim follows from Theorem [3.2.1]

**Corollary 5.2.5:** Given an instance \( \mathcal{P}_{\epsilon r}(k, \bar{x}, \bar{y}, C, \epsilon) \) of \( \epsilon \)-SVR, such that
the evaluation of \( k(\cdot, \cdot) \) can be done in constant time \( E \), then Algorithm [5.1]
using the selection procedure \( \text{selectCP} \) (Algorithm A.3), with input parameters
\[
\left( \begin{pmatrix} K & -K \\ -K & K \end{pmatrix}, \begin{pmatrix} \bar{y} + e \cdot e \\ \bar{y} - e \cdot e \end{pmatrix}, e^\top, 0, \begin{pmatrix} -\frac{c}{r} \cdot e \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{c}{r} \cdot e \end{pmatrix}, e, \bar{e} \right)
\]
stops after at most
\[
O \left( \max \left( \frac{\ell^2 C^2 \lambda_{\max,2}(K)}{\bar{e}}, \frac{\ell C^4 \lambda_{\max,2}(K)^2}{\bar{e}^2} \right) \right)
\]
steps. Given the output variables \( \hat{x} \) we define \( \hat{h} := \sum_{i=1}^{\ell} (\hat{x}_i - \hat{x}_{\ell+1}) \Phi(x_i) \).
Then
\[
R(\hat{h}, b, \xi) - \hat{R} \leq \bar{e},
\]
if \( b \) is a minimizers of \( G_{\text{SVM}}(\bar{x}) \) \((3.3.5)\) and \( \xi_i := [b - \nabla f(\bar{x})]_i^+ \), for \( i = 1, \ldots, \ell \) and \( \xi_i := [-b - \nabla f(\bar{x})]_i^+ \) for \( i = \ell + 1, \ldots, 2\ell \).

**Proof.** Note, that we did transform \( \mathcal{P}_{\epsilon r} \) by substituting \( x'_i = -x_i \) for all
\( i = 1, \ldots, \ell \). This flips the lower and upper bound of the box on these indexes and leads to a simple equality constraint \( e^\top \in \mathbb{R}^{\ell \times 1} \). Therefore the instance solved by Algorithm [5.1] is normalized pairable and as the kernel matrix \( K(i,j) = \pm k(x_i, x_j) \) can be evaluated in constant time the claim follows from Theorem [3.2.1]

**Corollary 5.2.6:** Given an instance \( \mathcal{P}_{\nu r}(k, \bar{x}, \bar{y}, C, \nu) \) of \( \nu \)-SVR, such that
the evaluation of \( k(\cdot, \cdot) \) can be done in constant time \( E \), then Algorithm [5.1]
using the selection procedure \( \text{selectCP} \) (Algorithm A.3), with input parameters
\[
\left( \begin{pmatrix} K & -K \\ -K & K \end{pmatrix}, \begin{pmatrix} \bar{y} \\ -\bar{y} \end{pmatrix}, \begin{pmatrix} -e \cdot e^\top \end{pmatrix}, \begin{pmatrix} 0 \end{pmatrix}, \begin{pmatrix} 0 \end{pmatrix}, \frac{C}{\ell} \cdot e, e, \frac{\bar{e}}{\ell} \right)
\]
stops after at most
\[
O \left( \max \left( \frac{\ell^2 C^2 \lambda_{\max,2}(K)}{\bar{e}}, \frac{\ell C^4 \lambda_{\max,2}(K)^2}{\bar{e}^2} \right) \right)
\]
steps. Given the output variables \( \hat{x} \) we define \( \hat{h} := \sum_{i=1}^{\ell} (\hat{x}_i - \hat{x}_{\ell+1}) \Phi(x_i) \).
Then
\[
R(\hat{h}, b, \rho, \xi) - \hat{R} \leq \bar{e},
\]
if \( (b, \rho) \) is a minimizers of \( G_{\text{SVM}}(\bar{x}) \) \((4.3.5)\) and \( \xi_i := [\frac{1}{\nu} \rho - \nabla f(\bar{x})]_i^+ \), for \( i = 1, \ldots, \ell \) and \( \xi_i := [\frac{1}{\nu} \rho - b - \nabla f(\bar{x})]_i^+ \) for \( i = \ell + 1, \ldots, 2\ell \).
Proof. Again the equality constraints \( \begin{pmatrix} -e^T \\ e^T \\ e^T \\ e^T \end{pmatrix} \) are easily seen to induce 2 equivalence classes with representatives \((1,1)\) and \((-1,-1)\), such that \( c = e \). As the kernel matrix \( K(i,j) = \pm k(x_i, x_j) \) can be evaluated in constant time the claim follows from Theorem 5.2.1.

5.2.2 Aggressive Decomposition Algorithms

To our knowledge there only exist two different aggressive pairing strategies in state of the art SMO-like SVM packages: The first is the well known maximal violating pair approach implemented in libSVM (version 2.7 and earlier), Torch and SVMlight for the special case \( q = 2 \). The second one the second order heuristics for violating pairs introduced by Fan et al. [2005], which is implemented in libSVM (since version 2.8.). In the following we will give convergence bounds for both of them, which hold for any pairable dual of the abstract regularized risk minimization problem \( 2.2.1 \).

Throughout this section we therefore consider the aggressive decomposition algorithms, which are given by Algorithm 5.1 using the maximum violating pair selection strategy \( \text{selectMVP} \) given in Algorithm A.1 and the derived second order strategy \( \text{select2ndMVP} \) given in Algorithm A.2.

We will now present the main result of this work, which gives run time and accuracy guarantees for aggressive pairing strategies. This rates hold for any instance of the abstract regularized risk minimization problem, which lead to pairable duals:

**Theorem 5.2.7:** Given an instance of the abstract regularized risk minimization problem \( 2.2.1 \), such that the dual \( 2.2.6 \) is a normalized pairable instance of BQO and a predefined accuracy \( \varepsilon > 0 \). Assume, that \( Q(i,j) := \langle A_i^T \nu, A_j^T \nu \rangle \) is computable in constant time \( E \), then Algorithm 5.1 using the selection procedure \( \text{selectMVP} \) or \( \text{select2ndMVP} \) (Algorithms A.1 and A.2), with input parameters

\[
\left( Q, d, \begin{pmatrix} A^b^T \\ A^\nu^T \end{pmatrix}, \begin{pmatrix} 0 \\ \nu \end{pmatrix}, 0, \Gamma, c, \varepsilon \right)
\]

stops after at most

\[
O\left( \ell^4 \kappa_2(Q) \frac{\Delta(0) \lambda_{max,2}(Q) \ell}{\varepsilon^2} \right),
\]

if \( Q \) is strictly positive definite, and after at most

\[
O\left( \ell^6 \|\Gamma\|_\infty^2 \lambda_{max,2}(Q)^2 \right)
\]
steps otherwise. Given the output variables $\hat{x}$ we define $\hat{h} := \hat{x}^\top A^H$. Then

$$R(\hat{h}, b, \rho, \xi) - \hat{R} \leq (\ell - L^1(\hat{x}) - L^1(\hat{x}))\|\Gamma\|_{\infty}\varepsilon,$$

for any choice of $b, \rho$ and $\xi$ according to Lemma 4.3.9.

Proof. Theorem 3.2.50 implies, that

$$\Delta_D(x) \leq \varepsilon^2 \ell \ln \frac{\|\Gamma\|_{\infty}^2}{2\lambda_{\text{max}, 2}(Q)}$$

after at most

$$n := \left\lfloor \frac{128\ell^2\|\Gamma\|_{\infty}^2\lambda_{\text{max}, 2}(Q)^2}{C^2(\mathcal{P}_Q)\varepsilon^2} \right\rfloor + \max \left\{ 0, \left\lceil \frac{4\ell \ln \frac{C^2(\mathcal{P}_Q)\Delta(0)}{8\ell^3\|\Gamma\|_{\infty}^2\lambda_{\text{max}, 2}(Q)}}{2\lambda_{\text{max}, 2}(Q)} \right\rceil \right\}$$

iterations. With Corollary 3.2.52 we conclude that at most $2\ell + 1$ iterations later $-\nabla f(x)^\top d_i \leq \varepsilon$ and the algorithm terminates. The overall run time follows with Lemma 5.1.1. From Lemma 4.3.9 and the stopping criterion $-\nabla f(x)^\top d_i \leq \varepsilon$ the primal optimality guarantee for $(h, b, \rho, \xi)$ follows.

For completeness we will apply the result from Theorem 5.2.7 to the various instances of (2.2.1) introduced in Chapter 2. As the following claims are similar to Corollary 5.2.2-5.2.6 we omit their elementary proofs.

Corollary 5.2.8: Given an instance $\mathcal{P}_C(k, \vec{x}, \vec{y}, C)$ of C-SVC, such that the evaluation of $k(\cdot, \cdot)$ can be done in constant time $E$, then Algorithm 5.1, using the selection procedure selectMVP or select2ndMVP (Algorithms A.1 and A.2), with input parameters

$$\left( K, -\vec{y}, e^\top, 0, l, r, e, \varepsilon \right)$$

stops after at most

$$O \left( \ell^4 K^2(K) \ln \frac{\Delta(0)}{\varepsilon^2} \lambda_{\text{max}, 2}(K) \ell \right)$$

steps, if $K$ is strictly positive definite, and after at most

$$O \left( \ell^2 C^2 \lambda_{\text{max}, 2}(K) \varepsilon \right)$$

steps otherwise. Given the output variables $\hat{x}$ we define $\hat{h} := \sum_{i=1}^\ell \hat{x}_i \Phi(x_i)$. Then

$$R_C(\hat{h}, b, \xi) - \hat{R}_C \leq (\ell - L^1(\hat{x}) - L^1(\hat{x}))C\varepsilon,$$

for any $b \in [-\nabla f(\hat{x})_j, -\nabla f(\hat{x})_j]$ and $\xi_i = [b - \nabla f(\hat{x})_i]_+$. 

103
Note that again the arguments from Remark 5.2.3 apply.

**Corollary 5.2.9:** Given an instance $\mathcal{P}_\nu(k, \vec{x}, \vec{y}, \nu)$ of $\nu$-SVC, such that the evaluation of $k(\cdot, \cdot)$ can be done in constant time $E$ and there exist at least two indexes $i, j$ with different label $y_i \neq y_j$, then Algorithm 5.1, using the selection procedure selectMVP or select2ndMVP (Algorithms A.1 and A.2), with input parameters

\[
\left( K', 0, \left( \frac{\vec{y}^T}{\epsilon^T} \right), \left( \frac{0}{\nu} \right), 0, \frac{1}{\ell} \cdot \epsilon, \epsilon, \epsilon \right)
\]

stops after at most

\[
O \left( \ell^4 \kappa_2(K') \ln \frac{\Delta^{(0)} \lambda_{\max,2}(K') \ell}{\epsilon^2} \right)
\]

steps, if $K'$ is strictly positive definite, and after at most

\[
O \left( \frac{\ell^4 \lambda_{\max,2}(K')}{\epsilon^2} \right)
\]

steps otherwise. Given the output variables $\hat{x}$ we define $\hat{h} := \sum_{i=1}^\ell \hat{x}_i \Phi(x_i)$. Then

\[
R_{\nu}(\hat{h}, b, \rho, \xi) - \hat{R}_{\nu} \leq \frac{\ell - L^1(\hat{x}) - L^1(\hat{x})}{\ell} \epsilon \leq \epsilon,
\]

for any choice of $b, \rho, \xi$ according to Lemma 4.3.9.

**Corollary 5.2.10:** Given an instance $\mathcal{P}_{\epsilon r}(k, \vec{x}, \vec{y}, C, \epsilon)$ of $\epsilon$-SVR, such that the evaluation of $k(\cdot, \cdot)$ can be done in constant time $E$, then Algorithm 5.1, using the selection procedure selectMVP or select2ndMVP (Algorithms A.1 and A.2), with input parameters

\[
\left( \begin{pmatrix} K & -K \\ -K & K \end{pmatrix}, \begin{pmatrix} \vec{y} + \epsilon \cdot \epsilon \\ \vec{y} - \epsilon \cdot \epsilon \end{pmatrix}, \epsilon^T, 0, \begin{pmatrix} -C \cdot \epsilon \\ 0 \end{pmatrix}, \epsilon, \epsilon \right)
\]

stops after at most

\[
O \left( \ell^4 \kappa_2(K) \ln \frac{\Delta^{(0)} \lambda_{\max,2}(K) \ell}{\epsilon^2} \right)
\]

steps, if $K$ is strictly positive definite, and after at most

\[
O \left( \frac{\ell^4 C^2 \lambda_{\max,2}(K)}{\epsilon^2} \right)
\]

steps otherwise. Given the output variables $\hat{x}$ we define $\hat{h} := \sum_{i=1}^\ell (-\hat{x}_i - \hat{x}_{t+1}) \Phi(x_i)$. Then

\[
R_{\epsilon r}(\hat{h}, b, \xi) - \hat{R}_{\epsilon r} \leq C \epsilon,
\]

for any $b \in [\nabla f(\hat{x})_j, -\nabla f(\hat{x})_i]$ and $\xi_i = [b - \nabla f(\hat{x})_i]^+$. 

104
Corollary 5.2.11: Given an instance \( \mathcal{P}_{\nu r}(k, x, y, C, \nu) \) of \( \nu \)-SVR, such that the evaluation of \( k(\cdot, \cdot) \) can be done in constant time \( E \), then Algorithm 5.1, using the selection procedure selectMVP or select2ndMVP (Algorithms A.1 and A.2), with input parameters

\[
\left( \begin{pmatrix} K & -K \\ -K & K \end{pmatrix}, \begin{pmatrix} \bar{y} \\ -\bar{y} \end{pmatrix}, \begin{pmatrix} -e^T \\ e^T \end{pmatrix}, \begin{pmatrix} 0 \\ c_{\nu} \end{pmatrix} \right), 0, \frac{C}{\ell} \cdot e, e, \varepsilon
\]

stops after at most

\[
O\left( \ell^4 K_2(K) \ln \frac{\Delta^{(0)}_2(K)}{\varepsilon^2} \ell \right)
\]

steps, if \( K \) is strictly positive definite, and after at most

\[
O\left( \ell^4 C^2 \lambda_{\max, 2}(K) \varepsilon^{-2} \right)
\]

steps otherwise. Given the output variables \( \hat{x} \) we define \( \hat{h} := \sum_{i=1}^\ell (\hat{x}_i - \hat{x}_{\ell+1})\Phi(x_i) \), and \( b, \rho, \xi \). Then

\[
R_{\nu r}(\hat{h}, b, \rho, \xi) - \hat{R}_{\nu r} \leq C\varepsilon,
\]

for any choice of \( b, \rho, \xi \) according to Lemma 4.3.9.

5.3 Interpretation of the Bounds

We will discuss the results in two main sections: one will be considered with the guarantees given for conservative selection the second one with the guarantees for aggressive selection.

In both sections we will highlighting the major drawbacks of the general worst case bounds given and will then continue to give arguments why especially in SVM optimization the negative influence of them is limited.

5.3.1 Conservative Decomposition

First of all, consider the general bounds given by Theorem 5.2.1. The dependence on \( \ell \) is of the unfortunate high order of \( O(\ell^5) \). Although there is no evidence that this bound is tight we conjecture that one may construct worst case instances of BQO on which the run time of the algorithm on standard work stations exceeds any acceptable time bound.

Nevertheless, the theorem gives a polynomial time bound for any pairable instance of BQO. This is especially astonishing if one takes into account that no dependence on the condition of the matrix \( Q \) is needed. Especially we are able to apply the proposed methods to semi-definite kernel matrices,
which are easily encountered in standard learning problems (for instance for polynomial kernels or even Gaussian kernels applied to mislabeled data). This is even more remarkable as the motivation for the presented algorithms is a steepest gradient descent, which fails from the very beginning for badly conditioned matrices (cf. Remark 1.3.5).

The reason for this phenomenon is the influence of the box constraints: In any step the error we suffer from first order approximation is in a way bounded by distance we can cover in the box, which is upper bounded by \( S_{\max}^2(\mathcal{P}_{QP})\lambda_{\max}(K) \) in general. In addition we restrict ourselves to low dimensional sub spaces – the extreme variant is SMO choosing only two indexes – which leads to an improved upper bound \( S_{\max}^2(\mathcal{P}_{QP})\lambda_{\max,2}(K) \) as shown in Lemma 3.3.3.

This is the main reason why the application of Theorem 5.2.1 to instances of SVM optimization leads to competitive bounds (cf Remark 5.2.3). This can best be seen for problem instances with intrinsic “small box”, as for example \( \nu \)-SVC (see Corollary 5.2.4). In the addition note, the influence of \( \lambda_{\max,2}(Q) \) can be ignored for the Gaussian kernel, where \( k(x_i, y_j) \leq 1 \) and therefore the maximal eigenvalue on \( 2 \times 2 \) principal matrices is bounded by \( \lambda_{\max,2}(Q) \leq 2 \).

Unfortunately the rates presented her exhibit a quadratic dependence on the accuracy \( 0 < \varepsilon < 1 \). This is due the fact that the used stopping criteria depend on first order heuristics as well (see Lemma 3.3.8). In general this leads to the worst case behavior of \( O(\ell^5/\varepsilon^2) \) which admittedly is a discouraging result. Yet, the influence of the box is again to our advantage. This is due to the fact that according to Lemma 3.3.8 the quadratic dependence holds iff \( \sigma(x|i, j) > 2\|\Gamma\|_\infty^2\lambda_{\max,2}(K) \). Consequently problem instances with small box exhibit more likely the favored linear dependence. An example is again the run time bound for \( \nu \)-SVC given by Corollary 5.2.4 which reads \( O(\max\{\ell^2/\varepsilon, \ell^2/\varepsilon^2\}) \). In this case note that the run time is given by \( O(\ell^2/\varepsilon) \) if ever \( \ell > 1/\varepsilon \). This situation might well happen in large scale SVM problems.

It is worth to remark that in the second case the bound coincides with the conjectured lower bound given by [Hush et al. 2006, Theorem 12] and is therefore optimal. This corresponds to the complexity consideration given by [Bottou and Lin 2007].

5.3.2 Aggressive Decomposition

In the case of aggressive selection the achieved worst case results appear to be even more disappointing. Although they seem to work better in practice the guarantees given by Theorem 5.2.7 are even worse concerning the dependence on \( \ell \). We will see in the following that, from a broader perspective, the results presented give evidence that the aggressive methods applied to SVM optimization problems perform much better in practice.

For the beginning let us ignore the influence of the stopping criterion and
Interpretation of the Bounds

rely solely on the dual convergence guarantees given by Theorem 3.2.50. This means we will consider overall run time guarantees for an fictive algorithm miraculously stopping if \( \Delta(x) \leq \varepsilon \) and therefore ignore primal optimality criteria. The presented arguments will hold for our algorithms as well.

For any pairable instance of BQO with strictly positive kernel matrix \( K \) Theorem 3.2.50 guarantees linear convergence which would lead to a run time of

\[
O \left( \ell^4 \kappa_2(K) \ln \frac{\Delta^{(0)}}{\varepsilon} \right),
\]

steps. In the less restrictive case we have quadratic convergence leading to a run time of

\[
O \left( \frac{\ell^5 \| \Gamma \|_2 \lambda_{\max,2}(K)}{\varepsilon} \right)
\]

steps. On first sight the second bound seems to be worse, note however, that there is a trade-off between the influence of the box and the condition of the principal \( 2 \times 2 \) sub matrices of \( K \). For example consider again problem instances of \( \nu \)-SVC: the small \( 1/\ell \) box leads to fast convergence rates of

\[
O \left( \frac{\ell^3 \lambda_{\max,2}(K)}{\varepsilon} \right)
\]

This run time bound holds without any assumptions and might be faster than the one given for the restricted case depending on \( \lambda_{\min}(K) \). This corroborates the claim that fast convergence of SVM optimization problems heavily depend on the size of the box.

In any way, up to now we have to admit that the guarantees for the practically superior aggressive strategies are worse than the given guarantees for the conservative strategy. In general we will not be able to close this gap. Still, we will present evidence that the worst case bounds given in Theorem 5.2.7 and subsequent Corollaries are far from being tight.

Exploiting the Sparsity

Remember the technique of maximal locking sequences used to derive convergence rates for aggressive selection strategies (see Definition 3.2.30 and Theorem 3.2.34). The guarantee in this situation was derived from an upper bound on the number of subsequent locking sequences such algorithms could run through before reaching an approximately optimal point. The overall run time then was derived by multiplying this number with the maximal length of one locking sequence. This bound is pessimistic in two ways: First of all remember that the longer a locking sequence the smaller the guarantee it gives (Lemma 3.2.32). Second, the upper bound on the length of locking sequences heavily relied on the fact that, during such locking sequences, more and more possible directions are locked at the border. Consequently
Chapter 5. SVMs and Optimization

the worst case bound, which is the basis of our convergence results, holds exactly when each locking sequence is of maximal length and in the end all directions are free again.

We conjecture however that typical SVM algorithms run in one of those regimes, not both. To be more concrete: Either we have long locking sequences but if the sequence ends most involved indexes are fixed at the border and the problem has become substantially smaller; for example at the beginning of an optimization where many steps directly hit the upper border. Or we have many free directions but encounter only short locking sequences; for example at the end of an SVM optimization process, where the algorithm in practice takes many free steps which always end a locking sequence. The claim therefore is that we either have a few long locking sequences or many short ones – not both at the same time; this situation would substantially improve the bounds.

A more concrete hint on why the bounds may be much too conservative are the following two cases, where we did use worst case upper bounds, which are obviously not tight in many instances of SVM optimization: 1) the promised aggressive advance only depends on the dimension the current solution differs from an optimal one (see Theorem 3.2.5) and 2) the length of a maximal locking sequence for GVP selection is reduced by the number of indexes locked at the border (see Lemma 3.2.48). Let us give some comments on this observation.

The first remark is exemplified quite obvious on the usual starting point $x^{(0)} = 0$: In this case the bound from Theorem 3.2.5 states that the aggressive guarantee for MVP selection (we assume $C(P_{QP}) = 1$), given at $x^{(0)}$, is

$$\delta_d(x^{(0)}) \geq \frac{1}{2\left(\#sv\right)^2\|\Gamma\|_\infty^2\lambda_{max,2}(K)} (\Delta^{(0)})^2,$$

where $\#sv = |\{i \mid \hat{x}_i \neq 0\}|$ is the number of support vectors. Even more if we assume that the first steps of an optimization procedure hit the top border for misclassified data points. This seems to be often the case in typical SVM problem instances and implies that the bound in later iterations even scales in $(\#nsv)^2$ where $\#nsv := |\{i \mid \ell_i < \hat{x}_i < u_i\}|$ denotes the number of unbounded support vectors. At least, in many cases of SVM optimization this seems to be more realistic than the assumption that $|\{i \mid x_i^{(n)} \neq \hat{x}_i\} \sim \ell$, which determined the given bound in the aggressive case. A typical iteration during a decomposition method could be of the order

$$\delta_d(x) \geq \frac{1}{2(\#nsv)^2\|F\|_\infty^2\lambda_{max,2}(K)} (\Delta(x))^2.$$

It is worth to remark that this improved bound holds without the application of any shrinking heuristics. The mere fact that we coincide in many dimension with the optimal solution improves the aggressive guarantee.
Interpretation of the Bounds

The second remark is concerned with an upper bound on the length of locking sequences. To judge the influence of $L^\uparrow(x)$ and $L^\downarrow(x)$ let us consider the example of $C$-SVC. Given a arbitrary dual feasible $x$ consider the associated classifier $h = \sum_{j=1}^{\ell} x_j \Phi(x_j)$. We conclude

$$\nabla f(x)_i = K_i^\top x - y_i = h(x_i) - y_i.$$  

Consequently, if we can guarantee $|\hat{h}(x_i)| \gg 1$ for the optimal classifier $\hat{h} = \sum_{j=1}^{\ell} \hat{x}_j \Phi(x_j)$, we may conclude, by continuity of $\sum_{i=1}^{\ell} x_i \Phi(x_i)$, that this holds for approximately optimal classifiers as well. We can therefore hope that clearly classifiable data points are locked early in the optimization process and therefore never contribute to a locking sequence. The same holds for clear outlier. An extreme is again given at the end of the optimization process where might find the situation that

$$\ell - L^\uparrow(\hat{x}) - L^\downarrow(\hat{x}) = O(\#nsv).$$  

To prove this one would have to demand non-degeneracy condition on the optimal solution $\hat{x}$ as follows: There exists a $c > 0$ such that for any $d \notin F(\hat{x})$ we have $\nabla f(\hat{x})^\top d > c$. The larger this $c$ would be, the clearer one could separate the unbounded support vectors from the indexes at the border. It seems plausible that this would help the algorithm to “sort out” the zero vectors as well as the bounded support vectors fast and train the classifier on the important set of unbounded support vectors only. With the above condition, one could adapt the arguments given by [Lin 2001a] and [Chen et al. 2006]. This would imply, that there exists an iteration $n$ such that the algorithm converges (in the case of $\nu$-SVC) after at most

$$O \left( \frac{(\#nsv)^4}{\ell \varepsilon} \right)$$

steps. Note that this is overly too optimistic but exemplifies that the worst case bound given above is far too conservative in many cases.

It is however worth to note that, from the optimization point of view, zero variables do not differ in their influence from variables encoding outliers, which typically are at the border of the constraint region as well. This implies that the complexity of problem should be expected to scale in the number of so called free support vectors which are exactly the data points on the decision boundary. This set might be much smaller than the dimension despite the results given by [Steinwart 2004].

Unfortunately influence of the stopping criterion for MVP algorithms suffers from the quadratic dependency on $\varepsilon$ as well. In addition the stopping guarantee derived from Corollary [3.2.52] introduces a dependence on $\ell$ which worsens our bound. Keep in mind that for an optimal $\hat{x}$ the above remarks hold in particular. I.e. the influence is probably rather of the magnitude of $\#nsv$ than of $\ell$. 

109
Let us finally note that one of the main advantages of the second order heuristic proposed by Fan et al. [2005] (implemented in Algorithm A.2) might be that they enforce the described behavior. This follows from the fact that optimizing the second index according to $\delta_{ij}(x)$ they favor directions with small $Q$-norm $\|d_{ij}\|_Q^2 = Q_{ii} + Q_{jj} - 2Q_{ij}$. As in this cases the promised advance may easily be larger then $\Delta(x)$ we surely hit the border. Although we have no mean of guaranteeing that such directions stay locked it is obvious that this strategy augments the tendency to keep index at the border – with all positive consequences described above.

All in all, the above remarks suggest that aggressive strategies are well adapted to problem instances where only a few variable are lying in the interior of the box constraints. Although the provable worst case guarantees are fairly weak these results support the use of aggressive selection in SVM optimization. It is finally worth noting that one can read the bounds considering the sparsity of the optimal solution as given in Theorem 3.2.5 Lemma 3.2.48 and Corollary 3.2.52 as justification for the shrinking heuristic as well. Although, we admittedly cannot determine which variables undoubtedly can be dropped, all those results suggest that aggressive selection strategies adapt automatically to the sparsity of typical SVM optimization problems.
Summary and Outlook

In this thesis we gave run time and accuracy bounds for decomposition algorithms in support vector machine (SVM) optimization. The bounds given hold uniformly for a special class of regularized risk minimization, which is a natural super class of standard SVM problem formulations. In addition the algorithms, which have been motivated from a general perspective, coincide with algorithms implemented in well known software packages used in practical application of SVMs.

The main contribution is twofold: First of all the given analysis is, to our knowledge, the first that systematically takes the requested primal optimality of the computed classifier or regression function into account. Second, this thesis presents state of the art convergence bounds for the two main variants of selection strategies in SVM optimization. It is especially worth to remark that it is the first time that provable convergence guarantees could be given for popular maximum violating pair selection (MVP) strategies. This includes the second order heuristics implemented in one of the fastest software packages available at the time of writing – libSVM. Admittedly the presented results do not hold for maximum gain working set strategies, as they rely on past decisions of the optimizer, and the implementation of MVP selection SVM\textsuperscript{light} for working set sizes larger than two.

Summary

We did present two types of results: The first type is concerned with convergence guarantees for SVM decomposition algorithms (Chapter 3). To this
end special selection strategies are designed, which coincide in the special case with known strategies as maximum violating pair or certifying pair selection. Secondly, we gave an analysis of general optimality bounds (Chapter 4). Again we show that special cases of this general gaps are found as stopping criteria in SVM optimization.

The main results of the first type are the following: For the special class of pairable problems we could show that MVP strategies and the derived second order MVP selection leads to a linear convergence rate of

$$O \left( \frac{\ell^3 k_\mathcal{P}(\mathcal{P}_Q) (Q)}{C^2 (\mathcal{P}_Q)} \ln \frac{\Delta^{(0)}}{\varepsilon} \right)$$

for instance $\mathcal{P}_Q$ with strictly positive definite matrix $Q$ and to a quadratic rate of

$$O \left( \frac{\ell^4 S^2_{\max} (\mathcal{P}_Q) \lambda_{\max,\mathcal{P}(\mathcal{P}_Q)} (Q)}{C^2 (\mathcal{P}_Q) \varepsilon} \right)$$

otherwise (Theorem 3.2.50). For the conservative certifying pair selection (Theorem 3.3.15) we did as well derive a quadratic convergence bound of

$$O \left( \frac{\ell^2 \lambda_{\max,2} (Q) S^2_{\max} (\mathcal{P}_Q)}{\varepsilon} \right).$$

The second type of results ensure that approximately optimal dual solutions computed by decomposition algorithms are mapped to approximately optimal primal solutions (see Corollary 4.3.12 and Lemma 4.3.9).

Combining these two results (cf. Chapter 5), we could show that on typical instances of SVM optimization (i.e. $\nu$-SVC) SMO-style decomposition algorithms stop after at most

$$O \left( \max \left\{ \frac{\ell^2 \lambda_{\max,2} (K')}{\varepsilon}, \frac{\ell \lambda_{\max,2} (K')^2}{\varepsilon^2} \right\} \right)$$

steps, where $K'$ is the kernel matrix, if a certifying pair strategy is used. If an aggressive selection is used the bound depends on the strict convexity of the dual objective. A worst case bound is however given by

$$O \left( \frac{\ell^4 \lambda_{\max,2} (K')}{\varepsilon^2} \right).$$

In both cases we can guarantee accuracy for the associated primal classifier.

In addition we did argue that a realistic run time bound can be expected to be smaller, especially for the aggressive selection strategies.
Outlook

The fact that the bounds in the aggressive case seem to be far from tight and the bad dependence on the accuracy parameter $\varepsilon$ are a main drawback of the presented theory. This discrepancy between provable guarantees and the practical applicability of decomposition methods is far from understood. To close this gap further is a natural topic for future research. In addition it would be worth to explore how the results for the second order heuristics can be improved. The fact that they perform even better in practice than classical maximum violating pair strategies is in no way reflected in the bounds presented.

Yet, the analysis given here might give a hint for a realistic run time bound for qualified generalized violating pair (GVP) selections. One should aim to give an amortized analysis of the length of maximum locking sequences. Such a proof could possibly exploit the fact that the bound of the length given in this work heavily depends on indexes being locked at the border. It seems to be realistic to assume that they stay locked at least for a bounded number of iterations. Therefore we conjecture that there cannot be arbitrary many long locking sequences.

A second direction of future research has already been touched in Section 2.2.6, where we did indicate that the abstract regularized risk formulation presented in this thesis might be applicable to more general instances of SVM. However it is obvious that the results given might not be sufficient to give alike guarantees for aggressive selection strategies, if the associated dual is not pairable. Note however, that the results here might apply for conservative selection strategies in such cases.
Appendix A

Algorithms

The following algorithms are selection procedures for the SMO like decomposition algorithm presented in Chapter 5 (Algorithm 5.1). As input parameters they request the components of the problem which might change, i.e. the gradient value $\nabla f(x) \in \mathbb{R}^\ell$, the current solution $x \in \mathbb{R}^\ell$ and references to the return values $wsi, wsj$. All other parameters are assumed to be available either as global variable or as object instance variable. These include the kernel function $Q(i,j)$, the box constraints $l, u \in \mathbb{R}^\ell$, the mapping $[i_r] \ni i \mapsto r$, the scalar coefficients $c \in \mathbb{R}^\ell$, and the requested accuracy $\varepsilon > 0$.

The return value of the presented selection procedures will be true if the working pair $(i,j)$ was initialized and false if no working set is selected because $x$ is optimal according to the chosen optimality criteria.
A.1 Aggressive Selection Procedures

The following Algorithm A.1 is an implementation of the approximate maximum violating pair algorithm proposed in Lemma 3.2.22. According to Lemma 4.3.9 it uses the gradient value $-\nabla f(x)^T d_{ij}$ of the selected pair $(i, j)$ as stopping criterion.

Algorithm A.1: selectMVP

Input: $x$, $\nabla f(x)$, $wsi$, $wsj$

1. foreach $r = 1, \ldots, s$ do /* Initialization */
2. max$[r] = -\infty$;
3. maxi$[r] = -1$;
4. maxr = -1; maxVal = -\infty;
5. foreach $i = 1, \ldots, \ell$ do /* Choose maximal up candidates */
6. Choose $r$ such that $i \in [i_r]$;
7. if $(x_i < u_i)$ and $(-\nabla f(x)_i / c_i > max[r])$ then
8. maxi$[r] = i$; max$[r] = -\nabla f(x)_i / c_i$;
9. foreach $i = 1, \ldots, \ell$ do /* Choose maximal pair */
10. Choose $r$ such that $i \in [i_r]$;
11. if $(x_i > l_i)$ and $(\text{max}[r] + \nabla f(x)_i / c_i > \text{maxVal})$ then
12. max$[r] = r$; wsj = $i$; maxVal = max$[r] + \nabla f(x)_i / c_i$;
13. if maxVal < $\varepsilon$ then /* Check optimality */
14. wsi = $i[\text{max}[r]]$
15. return true
16. else
17. return false
The second aggressive selection algorithm is based on the second order heuristics presented by Fan et al. [2005] (see also Corollary 3.2.23 and subsequent remarks). This selection technique again uses the gradient value $-\nabla f(x)^T d_{ij}$ of a maximal pair in the sense of Algorithm A.1 and Lemma 3.2.22. As above it therefore is a valid stopping criteria according to Lemma 4.3.9.

**Algorithm A.2: select2ndMVP**

**Input:** $x, \nabla f(x), w_{st}, w_{sj}$

1. **foreach** $r = 1, \ldots, s$ do /* Initialization */
2.   $\text{max}[r] = -\infty$;
3.   $\text{max}[i] = -1$;
4.   $\text{maxr} = -1; \text{maxVal} = \text{maxGrad} = -\infty$;
5. **foreach** $i = 1, \ldots, \ell$ do /* Choose maximal up candidates */
6.   Choose $r$ such that $i \in [i_r]$;
7.   if $(x_i < u_i)$ and $(-\nabla f(x)_i c_i > \text{max}[r])$ then
8.     $\text{max}[r] = i; \text{max}[r] = -\nabla f(x)_i c_i$;
9. **foreach** $i = 1, \ldots, \ell$ do /* Choose maximal second order pair */
10.   Choose $r$ such that $i \in [i_r]$;
11.   if $(x_i > l_i)$ and $(\text{max}[r] + \nabla f(x)_i c_i > 0)$ then
12.     if $\text{max}[r] + \nabla f(x)_i c_i > \text{maxGrad}$ then
13.       $\text{maxGrad} = \text{max}[r] + \nabla f(x)_i c_i$;
14.       $k = K(\text{max}[r], \text{max}[r]) + K(i,i) - 2K(\text{max}[r], i)$;
15.       if $k=0$ then
16.         $\text{maxr} = r; w_{sj} = i; \text{maxVal} = \infty$;
17.       else if $(\text{max}[r] + \nabla f(x)_i c_i)/k > \text{maxVal}$ then
18.         $\text{maxr} = r; w_{sj} = i; \text{maxVal} = (\text{max}[r] + \nabla f(x)_i c_i)/k$;
19.   if $\text{maxGrad} < \varepsilon$ then /* Check optimality */
20.     $w_{si} = i[\text{maxr}]$;
21.   return true
22. else
23.   return false

Note that we ignored the threshold parameter $\tau$ [Fan et al., 2005] used to account for kernel functions which are not positive definite.
A.2 Conservative Selection Procedures

The following selection procedure will be the main conservative selection strategy. It is based on the linear time certifying pair algorithm introduced by [Simon 2004]. According to Lemma 3.3.14 it induces a conservative qualified selection. As it uses the value of the computed certifying pair, Theorem 4.3.7 implies that it is a sensible stopping criteria.

Note that this algorithm uses an additional list \( M \) of length \( 2\ell \). It contains all \((i,c,\mu^+_{i}(x))\) and \((i,c,\mu^-_{i}(x))\) sorted in decreasing order according to the border distance. They are initialized during the first selection. In any subsequent iteration we assume that \( w_{si} \) and \( w_{sj} \) contain the indexes of the last working set.

It is worth to note that the data structures \( i[r][\mu] \) and \( j[r][\mu] \) do only need \( O(\ell) \) space as each \( \mu \) belongs to exactly one equivalence class \([i_r]\).
Algorithm A.3: selectCP

Input: $x, \nabla f(x), wsi, wsj$

1 if firstrun then /* Sort list */
2 // Sort M decreasing according to $c_i\mu_i^+(x)$, $c_j\mu_j^-$;
3 else /* Reinsert changed indexes */
4 reinsert $(wsi, c_{wsi}^+)$ and $(wsj, c_{wsj}^+)$ in M;
5 reinsert $(wsi, c_{wsi}^-)$ and $(wsj, c_{wsj}^-)$ in M;
6 foreach $r = 1, \ldots, s$ do /* Initialization */
7 maxM[$r$] = $-\infty$; minM[$r$] = $\infty$
8 foreach $(k, \mu) \in M$ do
9 Choose $r$ such that $k \in [i_r]$;
10 $i[r][\mu] = j[r][\mu] = -1$;
11 foreach $(i, \mu) \in M$ do
12 Choose $r$ such that $i \in [i_r]$;
13 if $-\nabla f(x)_i > \text{maxM}[r]$ then
14 $i[r][\mu] = i$;
15 maxM[$r$] = $-\nabla f(x)_i$;
16 if $-\nabla f(x)_i < \text{minM}[r]$ then
17 $j[r][\mu] = i$;
18 minM[$r$] = $-\nabla f(x)_i$;
19 maxVal = $-\infty$;
20 foreach $(k, \mu) \in M$ do
21 Choose $r$ such that $k \in [i_r]$;
22 $i = i[r][\mu]$, $j = j[r][\mu]$;
23 if $(\nabla f(x)_j - \nabla f(x)_i)_i \mu > \text{maxVal}$ then
24 maxVal = $(\nabla f(x)_j - \nabla f(x)_i)_i \mu$;
25 $wsi = i$; $wsj = j$;
26 if $\text{maxVal} < \varepsilon$ then /* Check optimality */
27 return true
28 else
29 return false


Index

\$\epsilon\$-SVR, 27
\$\nu\$-SVC, 26
\$\nu\$-SVR, 29

ACO, 3
advance
    aggressive, 37, 38
    conservative, 37, 38
    passive, 35
asymptotic
    cone, see recession cone
    function, see recession function
backward gap, 84
box constraints, 32
BQO, 31

C-SVC, 19, 24
C-SVM, 24
CCO, 3
certifying pair, viii, 76, 119
certifying set, 73
classifier
    linear, 21
condition number, 1
constancy space, 2
constrained optimization
abstract, 3
convex, 3
quadratic, 3, 8, 31
convergence
    aggressive, 59, 67
    conservative, 69, 73, 79
    linear, 10
    quadratic, 11
convex
    function, 2
    hull, 7
    set, 2
decomposition algorithm, vi, 33, 97
\$q\$-sparse, 34
    aggressive, 40, 59, 67
    conservative, 68, 69
    qualified, 39, 59, 67, 69
descent direction, 34
descent line search, 34
descent method, 10
direction
    \$q\$-sparse, 33
descent, 35, 54
free, 32, 34
    qualified, 39, 77
domain, 2
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dual approach</td>
<td>5</td>
</tr>
<tr>
<td>function</td>
<td>6</td>
</tr>
<tr>
<td>duality</td>
<td>4</td>
</tr>
<tr>
<td>abstract</td>
<td>3</td>
</tr>
<tr>
<td>empirical risk minimization</td>
<td>17</td>
</tr>
<tr>
<td>epigraph</td>
<td>2</td>
</tr>
<tr>
<td>feasibility region</td>
<td>1</td>
</tr>
<tr>
<td>feasible point</td>
<td></td>
</tr>
<tr>
<td>primal</td>
<td>4</td>
</tr>
<tr>
<td>feature map</td>
<td>20</td>
</tr>
<tr>
<td>filling property</td>
<td>7, 85</td>
</tr>
<tr>
<td>forward gap</td>
<td>84</td>
</tr>
<tr>
<td>gap</td>
<td></td>
</tr>
<tr>
<td>forward/backward</td>
<td>84</td>
</tr>
<tr>
<td>sigma</td>
<td>87</td>
</tr>
<tr>
<td>gradient selection</td>
<td></td>
</tr>
<tr>
<td>aggressive</td>
<td>41</td>
</tr>
<tr>
<td>conservative</td>
<td>72</td>
</tr>
<tr>
<td>Hilbert space</td>
<td>3</td>
</tr>
<tr>
<td>reproducing kernel</td>
<td>20</td>
</tr>
<tr>
<td>hypothesis class</td>
<td>16</td>
</tr>
<tr>
<td>linear</td>
<td>18</td>
</tr>
<tr>
<td>kernel</td>
<td>19</td>
</tr>
<tr>
<td>kernel trick</td>
<td>19</td>
</tr>
<tr>
<td>Lagrangian</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>4</td>
</tr>
<tr>
<td>optimization problem</td>
<td>4</td>
</tr>
<tr>
<td>locking sequence</td>
<td>56, 57</td>
</tr>
<tr>
<td>bounded</td>
<td>59, 63, 64</td>
</tr>
<tr>
<td>loss</td>
<td>16</td>
</tr>
<tr>
<td>piecewise linear</td>
<td>21</td>
</tr>
<tr>
<td>maximum margin classifiers</td>
<td>18</td>
</tr>
<tr>
<td>maximum violating pair</td>
<td>viii, viii</td>
</tr>
<tr>
<td>approximate</td>
<td>51, 52</td>
</tr>
<tr>
<td>newton step</td>
<td>38</td>
</tr>
<tr>
<td>norm</td>
<td></td>
</tr>
<tr>
<td>1-norm</td>
<td>12</td>
</tr>
<tr>
<td>Euclidean</td>
<td>12</td>
</tr>
<tr>
<td>Q-(semi)-norm</td>
<td>2</td>
</tr>
<tr>
<td>optimality condition</td>
<td></td>
</tr>
<tr>
<td>BQO</td>
<td>32</td>
</tr>
<tr>
<td>over fitting</td>
<td>17</td>
</tr>
<tr>
<td>pairable</td>
<td>48, 75</td>
</tr>
<tr>
<td>normalized</td>
<td>40</td>
</tr>
<tr>
<td>problem</td>
<td>46</td>
</tr>
<tr>
<td>QCO</td>
<td>3</td>
</tr>
<tr>
<td>recession</td>
<td></td>
</tr>
<tr>
<td>cone</td>
<td>2</td>
</tr>
<tr>
<td>direction of</td>
<td>2</td>
</tr>
<tr>
<td>function</td>
<td>2</td>
</tr>
<tr>
<td>regularized risk minimization</td>
<td>21</td>
</tr>
<tr>
<td>risk</td>
<td>16</td>
</tr>
<tr>
<td>Bayes</td>
<td>16</td>
</tr>
<tr>
<td>empirical</td>
<td>16</td>
</tr>
<tr>
<td>regularized</td>
<td>19</td>
</tr>
<tr>
<td>selection</td>
<td></td>
</tr>
<tr>
<td>aggressive</td>
<td>viii, 37, 40, 42, 43</td>
</tr>
<tr>
<td></td>
<td>52, 116, 117</td>
</tr>
<tr>
<td>conservative</td>
<td>viii, 37, 70, 72, 119</td>
</tr>
<tr>
<td>pairwise</td>
<td>51</td>
</tr>
<tr>
<td>qualified</td>
<td>72</td>
</tr>
<tr>
<td>linear</td>
<td>42</td>
</tr>
<tr>
<td>quadratic</td>
<td>43, 52</td>
</tr>
<tr>
<td>second order</td>
<td>117</td>
</tr>
<tr>
<td>space</td>
<td>53</td>
</tr>
<tr>
<td>sparse</td>
<td>44, 72</td>
</tr>
<tr>
<td>strategy</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>10, 64, 70</td>
</tr>
<tr>
<td>sigma</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>70, 74</td>
</tr>
<tr>
<td>gap</td>
<td>84</td>
</tr>
<tr>
<td>problem</td>
<td>70</td>
</tr>
<tr>
<td>SMO</td>
<td>97</td>
</tr>
<tr>
<td>sparse witness of suboptimality</td>
<td>35</td>
</tr>
<tr>
<td>statistical learning theory</td>
<td>15</td>
</tr>
<tr>
<td>stopping criteria</td>
<td>13, 86, 90, 92, 106</td>
</tr>
<tr>
<td></td>
<td>109, 116, 117, 119</td>
</tr>
<tr>
<td>Term</td>
<td>Page(s)</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>sub-differential</td>
<td>2</td>
</tr>
<tr>
<td>sub-gradient</td>
<td>2</td>
</tr>
<tr>
<td>support vector</td>
<td>108</td>
</tr>
<tr>
<td>machine</td>
<td>20</td>
</tr>
<tr>
<td>classification</td>
<td>24, 25</td>
</tr>
<tr>
<td>machine</td>
<td>19</td>
</tr>
<tr>
<td>regression</td>
<td>26, 27</td>
</tr>
<tr>
<td>unbounded</td>
<td>108</td>
</tr>
<tr>
<td>support vector machine</td>
<td>V</td>
</tr>
<tr>
<td>weak duality</td>
<td>6</td>
</tr>
<tr>
<td>working set</td>
<td>36, 73</td>
</tr>
</tbody>
</table>
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>vector space</td>
</tr>
<tr>
<td>$V^*$</td>
<td>algebraical dual of vector space $V$</td>
</tr>
<tr>
<td>$\mathbb{N}$</td>
<td>positive integers containing zero</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>the real numbers</td>
</tr>
<tr>
<td>$\mathbb{R}^+$</td>
<td>the positive real numbers ${t \mid t \geq 0} \subset \mathbb{R}$</td>
</tr>
<tr>
<td>$\mathbb{R}^\infty$</td>
<td>closure of the reals $\mathbb{R} \cup {\pm \infty}$</td>
</tr>
<tr>
<td>$x_I$</td>
<td>sub vector of $X$ consisting of entries $x_i$, $i \in I$</td>
</tr>
<tr>
<td>$\kappa(Q)$</td>
<td>condition number of square matrix $Q$</td>
</tr>
<tr>
<td>$\kappa_q(Q)$</td>
<td>relative condition number of $q \times q$ principal sub matrix $Q$</td>
</tr>
<tr>
<td>$\lambda_{\text{max},q}(Q)$</td>
<td>maximal eigenvalue of $q \times q$ principal sub matrix of $Q$</td>
</tr>
<tr>
<td>$\lambda_{\text{min}}(Q), \lambda_{\text{max}}(Q)$</td>
<td>minimal/maximal eigenvalue of square matrix $Q$</td>
</tr>
<tr>
<td>$|\cdot|_1, |\cdot|<em>2, |\cdot|</em>\infty$</td>
<td>1-norm, Euclidean norm, and $\infty$-norm on $\mathbb{R}^\ell$</td>
</tr>
<tr>
<td>$|\cdot|Q$</td>
<td>(semi) norm induced by a positive (semi-)definite symmetric matrix $Q \in \mathbb{R}^\ell$</td>
</tr>
<tr>
<td>$(x,y)$</td>
<td>labeled sample from $\mathcal{X} \times \mathcal{Y}$</td>
</tr>
<tr>
<td>$D$</td>
<td>probability distribution on $\mathcal{X} \times \mathcal{Y}$</td>
</tr>
<tr>
<td>$R_D$</td>
<td>risk functional depending on distribution $D$</td>
</tr>
<tr>
<td>$R_T$</td>
<td>empirical risk based on a data set $T$</td>
</tr>
<tr>
<td>$T_\ell$</td>
<td>training sample $T_\ell \subset (\mathcal{X} \times \mathcal{Y})^\ell$ of size $\ell$</td>
</tr>
<tr>
<td>$\mathcal{F}$</td>
<td>hypothesis class</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$\mathcal{H}$</td>
<td>(reproducing kernel) Hilbert space</td>
</tr>
<tr>
<td>$\mathcal{H}'$</td>
<td>topological dual of Hilbert space $\mathcal{H}$</td>
</tr>
<tr>
<td>$\mathcal{X}$</td>
<td>input space</td>
</tr>
<tr>
<td>$\mathcal{Y}$</td>
<td>label space</td>
</tr>
<tr>
<td>$\check{x}, \check{y}$</td>
<td>vector of data $\check{x} \in \mathcal{X}^d$ and labels $\check{y} \in \mathcal{Y}^d$</td>
</tr>
<tr>
<td>$k$</td>
<td>positive semi-definite symmetric kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>$\text{co}U$</td>
<td>convex hull of a set $U \subset V$</td>
</tr>
<tr>
<td>$C^0(f)$</td>
<td>constancy space of a convex function $f$</td>
</tr>
<tr>
<td>$\text{dom} f$</td>
<td>domain of a function $F : V \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>$\text{epi} f$</td>
<td>epigraph of a function $F : V \rightarrow \mathbb{R}$</td>
</tr>
<tr>
<td>$\partial f(x)$</td>
<td>sub-differential of a convex function $f$ at $x$</td>
</tr>
<tr>
<td>$C^\infty(C), C^\infty(f)$</td>
<td>recession cone of a convex set $C$, convex function $f$</td>
</tr>
<tr>
<td>$f'_\infty$</td>
<td>recession function of a convex function $f$</td>
</tr>
<tr>
<td>$C_\lambda$</td>
<td>constraint values at points in $U_\lambda$</td>
</tr>
<tr>
<td>$U_\lambda$</td>
<td>maximizers of Lagrangian function at $\lambda$</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>optimization problem</td>
</tr>
<tr>
<td>$\mathcal{R}(\mathcal{P})$</td>
<td>feasibility region of an optimization problem $\mathcal{P}$</td>
</tr>
<tr>
<td>$\hat{f}, \hat{\varphi}, \hat{\psi}$</td>
<td>optimal value of optimization problem with objective $f, \varphi, \psi$</td>
</tr>
<tr>
<td>$\check{x}, \check{d}, \check{u}, \check{\lambda}$</td>
<td>optimal solution of various optimization problems with variables $x, d, u, \lambda$</td>
</tr>
<tr>
<td>$x^{(n)}$</td>
<td>value of $n$-th iteration</td>
</tr>
<tr>
<td>$G_{\text{QP}}(\lambda)$</td>
<td>optimality gap for $\mathcal{P}_{\text{QP}}$</td>
</tr>
<tr>
<td>$G_{\text{SV}}(\mathcal{X})$</td>
<td>optimality gap for SVM problems</td>
</tr>
<tr>
<td>$L_A(\mathcal{P}_{\text{QP}})$</td>
<td>longest locking sequence of algorithm $A$ on instance $\mathcal{P}_{\text{QP}}$</td>
</tr>
<tr>
<td>$[i_r]$</td>
<td>pairable equivalence class</td>
</tr>
<tr>
<td>$L^U(x), L^L(x)$</td>
<td>up/down locked candidates at $x$</td>
</tr>
<tr>
<td>$P^U(n), P^L(n)$</td>
<td>up/down selectable candidates at $x^{(n)}$</td>
</tr>
<tr>
<td>$\delta_d(x)$</td>
<td>aggressive advance along $d$ at $x$</td>
</tr>
<tr>
<td>$\delta_d^C(x)$</td>
<td>conservative advance along $d$ at $x$</td>
</tr>
<tr>
<td>$\hat{\alpha}$</td>
<td>mapping $\hat{\alpha} : B\text{QO} \rightarrow \mathbb{R}^+$</td>
</tr>
<tr>
<td>$\hat{q}$</td>
<td>mapping $\hat{q} : B\text{QO} \rightarrow \mathbb{N}$</td>
</tr>
<tr>
<td>$\mathcal{P}_{a}(x)$</td>
<td>aggressive gradient selection problem at $x$</td>
</tr>
<tr>
<td>$\mathcal{P}_{c}(x)$</td>
<td>conservative gradient selection problem at $x$</td>
</tr>
<tr>
<td>$\mathcal{P}<em>{\sigma}(x), \mathcal{P}</em>{\sigma}(x</td>
<td>I)$</td>
</tr>
<tr>
<td>$\sigma(x), \sigma(x</td>
<td>I)$</td>
</tr>
<tr>
<td>$G(u), G(\lambda)$</td>
<td>forward/backward gap</td>
</tr>
</tbody>
</table>