Contents

1 Introduction

There exists a sound and board theory of classical nonlinear optimization. However, this theory puts strong differentiability requirements on the given problem. Requirements that cannot always be fulfilled in practice. Examples for such nondifferentiable applications reach from problems in physics and mechanical engineering [3] over optimal control problems up to data analysis [2] and machine learning [50]. Other possible fields of applications are risk management and financial calculations [37, 54]. Additionally the problem class of bilevel programs can yield nonsmooth objective functions as shown in [43] and [36]. There is hence a need for nonsmooth optimization algorithms.

A lot of the underlying theory was developed in the 1970's also driven by the "First World Conference on Nonsmooth Optimization" taking place in 1977 [34]. These days, there exists a well understood theoretical framework of nonsmooth analysis to create the basis for practical algorithms [47].

The most popular methods to tackle nonsmooth problems at the moment are bundle methods [13]. First developed only for convex functions [27] these methods were soon extended to cope also with nonconvex objective functions [33]. Some time later the algorithms were again enhanced to deal with inexact information of the function value, the subgradient or both. Some natural applications for these cases are derivative free optimization and stochastic simulations [13].

The basic idea of bundle methods is to model the original problem by a simpler function, often some sort of stabilized cutting plane model, that is minimized as a subproblem of the algorithm [16, chapter XV]. The computed iterate is tested for sufficient descent and depending on the result is either taken as the new iterate or the model is enhanced.

There exist different types of bundle methods, a widely used one being the proximal bundle method. In this thesis two types bundle methods are worked with. One is of the proximal type and one uses a variable stabilization term that makes it possible to make use of curvature information in order to accelerate the convergence speed. The development of the algorithm

The first half of this work puts particular attention on the theoretical concepts to use bundle methods with nonconvex and inexact objectives and how to incorporate the curvature information into the method.

In the second half of the thesis the usability of bundle algorithms for bilevel programs is explored. Bilevel problems consist of an upper level problem constrained by an additional

optimization problem, the lower level. These problems occur in a variety of applications such as game theory (see [4, section 2.1] for a variety of applications). Here the bilevel problem is derived from the hyper-parameter optimization for support vector machines. In the application both nonconvexity of the objective function and inexactness in the function value and subgradient calculation are addressed.

The remainder of the thesis is organized as follows: After a short introduction of the most important definitions and results from nonsmooth analysis in section 2 a basic bundle algorithm for exact convex functions is stated in order to introduce the important concepts of this method in section 3. A survey of different methods to tackle inexactness and nonconvex objective functions i then presented in section 4. Section 5 reviews the proximal bundle algorithm for nonconvex inexact functions presented in [13] and contains some closer analysis of the method. In section 6 a variable metric variant of that algorithm is developed using the nonsmooth second model suggested in [42] and [40]. This method makes it possible to incorporate second order information into the algorithm in order to speed up convergence. The two methods are compared on different academic examples. At last the nonconvex inexact bundle method is used on the application of parameter optimization in support vector classification.

This thesis is written with the acedemic 'we'.

2 Preliminaries

When it comes to nonsmooth objective functions the derivative based framework of nonlinear optimization methods does not work any more. Meanwhile there exists a well understood theory of 'subdifferential calculus' that gives similar results in the nondifferentiable case. The most important definitions and results of this theory together with some remarks on notation are stated in this section.

2.1 Notation

Let x denote a column vector. The transpose of x is denoted by $x^\top$. The scalar product is written $\langle \cdot, \cdot \rangle$. In this thesis generally the euclidean norm is used and denoted by $\| \cdot \|$. In section 6 additionally a norm is used that is induced by a symmetric matrix. Here we use the notation $\|x\|_A^2 = \langle x, Ax \rangle$. Inequalities written for vectors $x^1 \leq x^2$, $x^1, x^2 \in \mathbb{R}^n$ are to be read component wise. With 0 we denote the zero vector of appropriate size. The identity matrix of appropriate size is written as $\mathbb{I}$.

As we work with numerical methods in this thesis occur a lot of sequences of various dimensions. For vectors iteration indices are indicated by a superscript x^k whereas the components are indicated by subscripts $x = (x_1, x_2, ..., x_n)^\top$. Sequences of numbers and matrices a indexed with subscripts. For (sub-)sequences where k comes from an index set $K \subset \mathbb{N}$ we write $\{x^k\}_{k \in K}$. If k is in the natural numbers this notation is shortened to $\{x^k\}$. We denote the open ball around x with radius r with $B_r(x)$. The subset relation is denoted by $A \subset B$. It is to be read in the sense that A is a subset of B or that $A = B$.

2.2 Nonsmooth Analysis

Throughout this thesis we consider different optimization problems of the form

$$\min_x f(x) \quad \text{s.t.} \quad x \in X \subset \mathbb{R}^n$$

where f is a possibly nonsmooth function.

Nonsmooth functions have kinks where a unique gradient cannot be defined. It is however possible to define a set of tangents to the graph called subdifferential. The subdifferential was first defined for convex functions.

Definition 2.1 ([17, Definition 1.2.1, p. 241]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function.

The *subdifferential* of f at $x \in \mathbb{R}^n$ is the set

$$\partial f(x) := \{g \in \mathbb{R}^n \mid f(y) - f(x) \geq \langle g, y - x\rangle \quad \forall y \in \mathbb{R}^n\}.$$

The subdifferential is a set valued mapping. It is closed and convex. If f is differentiable, its subdifferential is single valued and coincides with its gradient $\partial f(x) = \nabla f(x)$ [46].

It is also possible to define a subdifferential for nonconvex functions. This is the subdifferential we will work with in this thesis most of the time.

Definition 2.2 (c.f. [3, p. 25, 27]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz (and not necessarily convex). The *subdifferential* of f at $x \in \mathbb{R}^n$ is the set

$$\partial f(x) := \left\{ g \in \mathbb{R}^n \;\middle|\; \limsup_{y \to x,\ h \searrow 0} \frac{f(y + hv) - f(y)}{h} \quad \forall v \in \mathbb{R}^n \right\}.$$

All convex functions are locally Lipschitz [17, Theorem 3.1.1, p. 16] so the above definition holds also for convex functions. In fact if the function is convex the subdifferential from definition 2.2 is equivalent to the one from definition 2.1 [3, Proposition 2.2.7, p. 36]. Due to this equivalence we call elements from both subdifferentials subgradients.

Remark: It is important to observe that subgradient inequality

$$f(y) - f(x) \geq \langle g, y - x\rangle \quad \forall y \in \mathbb{R}^n \tag{2.1}$$

only holds in the convex case.

There is also a sum rule for the subdifferential.

Proposition 2.3 ([3, Proposition 2.3.3, p. 38]) Let $F(x) = \sum_i f_i(x)$ be a finite sum of nondifferentiable functions. Then it holds

$$\partial F(x) \subset \sum_i \partial f_i(x).$$

Analogous to the $\mathcal{C}^1$-case some first order optimality conditions can be stated. For nondifferentiable functions a *stationary point* x of the function f is characterized by [3, p. 38]

$$0 \in \partial f(x).$$

If the function f is convex, then every stationary point is a minimum.

A drawback of the subdifferential is that it does not indicate how near the evaluated point is to a stationary point or minimum of a function. This can only be seen if the evaluated point is already stationary.

This issue is addressed by the *ε-subdifferential*. It gathers all information in a small neighborhood of the point x.

For convex functions an *ε-subgradient* of $f(x)$ is defined as a vector $g \in \mathbb{R}^n$ satisfying the inequality

$$f(y) - f(x) \geq \langle g, y - x \rangle - \varepsilon \quad \forall y \in \mathbb{R}^n.$$

The ε-subdifferential is then the set

$$\partial_\varepsilon f(x) := \{g \in \mathbb{R}^n \mid g \text{ is an } \varepsilon\text{-subgradient of } f(x)\}.$$

For nonconvex functions the subdifferential that is used in this thesis is the *Fréchet ε-subdifferential*.

Definition 2.4 (c.f. [18, p. 73]) The Fréchet ε-subdifferential of $f(x)$ is

$$\partial_{[\varepsilon]} f(x) := \left\{ g \in \mathbb{R}^n \;\middle|\; \liminf_{\|h\| \to 0} \frac{f(x+h) - f(x) - \langle g, h \rangle}{\|h\|} \geq -\varepsilon \right\}.$$

For $\varepsilon = 0$ this is called *Fréchet subdifferential*. For convex functions the Fréchet ε-subdifferential and the ε-subdifferential are *not* the same.

In the course of this thesis we sometimes derive stronger results for a smaller class of nonsmooth functions. Those functions are called lower-$\mathcal{C}^2$ functions and can be defined as follows.

Definition 2.5 (c.f. Definition 10.29, p. 447 in [47]) A function $f : \mathbb{R}^n \to \mathbb{R}$ is *lower-$\mathcal{C}^2$*, if on some neighborhood Ω of each $\bar{x} \in \mathbb{R}^n$ there exists a representation

$$f(x) = \max_{t \in T} f_t(x)$$

where the functions f_t are of class $\mathcal{C}^2$ on Ω and $f_t(x)$ and all its first and second partial derivatives depend continuously on both x and $(x, t) \in \Omega \times T$. The index set T is a compact space.

Lower-$\mathcal{C}^2$ functions are locally Lipschitz continuous [47, Theorem 10.31, p. 448].

3 A Basic Bundle Method

When bundle methods were first introduced in 1975 by Lemaréchal and Wolfe they were developed to minimize a convex (possibly nonsmooth) function f for which at least one subgradient at any point x can be computed [34]. To provide an easier understanding of the proximal bundle method from [13] presented in section 5 and stress the most important ideas of how to deal with nonconvexity and inexactness in bundle methods first a basic bundle method is shown here.

Bundle methods can be interpreted in two different ways: From the dual point of view one tries to approximate the ε-subdifferential to finally ensure first order optimality conditions. The primal point of view interprets the bundle method as a stabilized form of the cutting plane method where the objective function is modeled by tangent hyperplanes [12]. We focus here on the primal approach.

3.1 Derivation of the Bundle Method

This section gives a short summery of the derivations and results of chapter XV in [16] where a primal bundle method is derived as a stabilized version of the cutting plane method. If not otherwise indicated the results in this section are therefore taken from chapter XV in [16].

The optimization problem considered in this section is

$$\min_x f(x) \quad \text{s.t.} \quad x \in X \tag{3.1}$$

where f is a convex but possibly nondifferentiable function and $X \subset \mathbb{R}^n$ is a closed and convex set.

3.1.1 A Stabilized Cutting Plane Method

The geometric idea of the *cutting plane method* is to build a piecewise linear model of the objective function f that can be minimized more easily than the original objective function. This model is built from a *bundle* of information that is gathered in the previous iterations. In the k'th iteration, the bundle consists of the previous iterates x^j, the respective function values $f(x^j)$ and a subgradient at each point $g^j \in \partial f(x^j)$ for all indices j in the index set J_k. From each of these triples one can construct a linear function

$$l_j(x) := f(x^j) + \left\langle g^j, x - x^j \right\rangle$$

where $f(x^j) = l_j(x^j)$ and due to convexity $f(x) \geq l_j(x),\ x \in X$.
The objective function f can then be approximated by the piecewise linear function

$$m_k(x) := \max_{j \in J_k} l_j(x). \tag{3.2}$$

This function is therefore also called *model function*. Instead of working with the original objective, a new iterate x^{k+1} is found by solving the subproblem

$$\min_x m_k(x) \quad \text{s.t.} \quad x \in X.$$

Picture of function and cutting plane approximation of it

This subproblem should of course be easier to solve than the original task. A question that depends a lot on the structure of X. If $X = \mathbb{R}^n$ or a polyhedron, the problem can be solved easily. Still there are some major drawbacks to the idea. For example if $X = \mathbb{R}^n$ the solution of the subproblem in the first iteration is always $-\infty$. In general we can say that the subproblem does not necessarily have a solution. To tackle this problem a regularization term is introduced to the subproblem. It then reads

$$\min \tilde{m}_k(x) = m_k(x) + \frac{1}{2t_k}\|x - x^k\|^2 \quad \text{s.t.} \quad x \in X,\ t_k > 0. \tag{3.3}$$

This new subproblem is strongly convex and therefore always has a unique solution.

The regularization term can be motivated and interpreted in many different ways (c.f. [16, chapter XV]). From different possible regularization terms the most popular in bundle methods is the penalty-like regularization used here.

The second major step towards the bundle algorithm is the introduction of a so called *stability center* or *serious point* $\hat{x}^k$. It is the iterate that yields the "best" approximation of the optimal point up to the k'th iteration (not necessarily the lowest function value though). The updating technique for $\hat{x}^k$ is crucial for the convergence of the method: If the next iterate yields a decrease of f that is "large enough", namely larger than a fraction of the decrease suggested by the model function for this iterate, the stability center is moved to that iterate. If this is not the case, the stability center remains unchanged.

In practice this is implemented as follows: First define the *model decrease* δ_k^M which is

the decrease of the model for the new iterate x^{k+1} compared to the function value at the current stability center $\hat{x}^k$

$$\delta_k^M := f(\hat{x}^k) - m_k(x^{k+1}) \geq 0. \tag{3.4}$$

If the actual decrease of the objective function is larger than a fraction of the model decrease

$$f(\hat{x}^k) - f(x^{k+1}) \geq m\delta_k^M, \quad m \in (0,1)$$

set the stability center to $\hat{x}^{k+1} = x^{k+1}$. This is called a *serious* or *descent step*. If this is not the case a *null step* is executed and the serious iterate $\hat{x}^{k+1} = \hat{x}^k$ remains the same.

Beside the model decrease other forms of decrease measures and variations of these are possible. Some are presented in [16] and [62].

3.1.2 Subproblem Reformulations

The subproblem to be solved to find the next iterate can be rewritten as a smooth optimization problem. For convenience we first rewrite the affine functions l_j with respect to the stability center $\hat{x}^k$:

$$\begin{aligned} l_j(x) &= f(x^j) + \left\langle g^j, x - x^j \right\rangle \\ &= f(\hat{x}^k) + \left\langle g^j, x - \hat{x}^k \right\rangle - \left(f(\hat{x}^k) - f(x^j) + \left\langle g^j, x^j - \hat{x}^k \right\rangle \right) \\ &= f(\hat{x}^k) + \left\langle g^j, x - \hat{x}^k \right\rangle - e_j^k \end{aligned}$$

where

$$e_j^k := f(\hat{x}^k) - f(x^j) + \left\langle g^j, x^j - \hat{x}^k \right\rangle \geq 0 \quad \forall j \in J_k$$

is the *linearization error*. Due to convexity of f it is nonnegative. This property is essential for the convergence theory and will also be of interest when moving on to the case of nonconvex and inexact objective functions.

Subproblem (3.3) can now be written as

$$\min_{\hat{x}^k+d\in X} \tilde{m}_k(\hat{x}^k+d) = f(\hat{x}^k) + \max_{j\in J_k}\left\{\left\langle g^j, d\right\rangle - e_j^k\right\} + \frac{1}{2t_k}\|d\|^2 \tag{3.5}$$

$$\Leftrightarrow \quad \min_{\substack{\hat{x}^k+d\in X,\\ \xi\in\mathbb{R}}} \xi + \frac{1}{2t_k}\|d\|^2 \quad \text{s.t.} \quad \left\langle g^j, d\right\rangle - e_j^k - \xi \le 0, \quad j\in J_k \tag{3.6}$$

where $d := x - \hat{x}^k$ and the constant term $f(\hat{x}^k)$ was discarded for the sake of simplicity. If X is a polyhedron this is a convex quadratic optimization problem that can be solved using standard methods of nonlinear optimization. It should however be observed that the matrix of the quadratic part is only positive semidefinite because it does not have full rank.

The pair (ξ_k, d^k) solves (3.6) if and only if

$$\begin{aligned} &d^k \text{ solves the original subproblem (3.5) and} \\ &\xi_k = \max_{j\in J_k} {g^j}^\top d^k - e_j^k = m_k(\hat{x}^k + d^k) - f(\hat{x}^k). \end{aligned} \tag{3.7}$$

The new iterate is given by $x^{k+1} = \hat{x}^k + d^k$.

3.2 The Prox-Operator

The constraint $\hat{x}^k + d \in X$ can also be incorporated directly in the objective function by using the *indicator function*

$$\mathtt{i}_X(x) := \begin{cases} 0, & \text{if } x \in X \\ +\infty, & \text{if } x \notin X \end{cases}. \tag{3.8}$$

This function is convex if and only if the set X is convex [47, p. 40].

Remark: The indicator function is actually an extended-real-valued function, meaning that it allows the function value $+\infty$. Introducing it into the subproblem means that the objective function of the subproblem also becomes an extended-real-valued function. As this is does not have any impact on the convergence theory we omit to introduce the concept of extended-real-valued functions here.

Subproblem (3.3) then reads with respect to the serious point $\hat{x}^k$

$$\min_{x\in\mathbb{R}^n} m_k(x) + \mathtt{i}_X(x) + \frac{1}{2t_k}\|x - \hat{x}^k\|^2. \tag{3.9}$$

The subproblem is now written as the *Moreau-Yosida regularization* of $\check{f}(x) := m_k(x) + \mathtt{i}_X(x)$. The emerging mapping is also known as *proximal point mapping* [12] or *prox-operator*

$$prox_{t,\check{f}}(x) := \arg\min_{y\in\mathbb{R}^n} \left\{ \check{f}(y) + \frac{1}{2t}\|x - y\|^2 \right\}, \quad t > 0. \tag{3.10}$$

This special form of the subproblems gives the primal bundle method its name, *proximal bundle method*. The above mapping also plays a key role when the method is generalized to nonconvex objective functions and inexact information.

3.3 Aggregation and Stopping Condition

We look again at a slightly different formulation of the bundle subproblem

$$\begin{aligned} \min_{\substack{d\in\mathbb{R}^n,\\ \xi\in\mathbb{R}}} \quad & \xi + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2t_k}\|d\|^2 \\ \text{s.t.} \quad & \left\langle g^j, d \right\rangle - e_j^k - \xi \le 0, \quad j \in J_k. \end{aligned}$$

As the objective function is still convex (X is a convex set) the following Karush-Kuhn-Tucker (KKT) conditions have to be valid for the minimizer $\left(\xi_k, d^k\right)$ of the above subproblem [17] assuming a constraint qualification holds if the constraint set X makes it necessary [52].

There exist a subgradient $\nu^k \in \partial \mathtt{i}_X(\hat{x}^k + d^k)$ and Lagrangian multipliers $\alpha_j,\ j \in J^k$ such that

$$0 = \nu^k + \frac{1}{t_k} d^k + \sum_{j\in J^k} \alpha_j g^j, \tag{3.11}$$

$$\sum_{j\in J_k} \alpha_j = 1, \tag{3.12}$$

$$\alpha_j \ge 0,\ j \in J^k, \tag{3.13}$$

$$\left\langle g^j, d^k \right\rangle - e_j^k - \xi_k \leq 0 \text{ and} \tag{3.14}$$

$$\sum_{j \in J^k} \alpha_j \left(\left\langle g^j, d^k \right\rangle - e_j^k - \xi_k \right) = 0. \tag{3.15}$$

From condition (3.11) follows that

$$d^k = -t_k \left(G^k + \nu^k \right) \tag{3.16}$$

with the *aggregate subgradient*

$$G^k := \sum_{j \in J^k} \alpha_j g^j \quad \in \partial m_k(x^{k+1}). \tag{3.17}$$

The fact that G^k belongs to the subdifferential of the k'th model m_k at the point $\hat{x}^k + d^k$ follows from noting that

$$0 \in \partial m_k(\hat{x}^k + d^k) + \partial \mathtt{i}_X(\hat{x}^k + d^k) + \frac{1}{2t_k} d^k$$

is the optimality condition derived from formulation (3.9) by the sum rule for subdifferentials and comparing the different components with the ones derived in (3.11).

Rewriting condition (3.15) yields the *aggregate error*

$$E_k := \sum_{j \in J^k} \alpha_j e_j^k = \left\langle G^k, d^k \right\rangle + f(\hat{x}^k) - m_k(x^{k+1}). \tag{3.18}$$

Here relation (3.7) was used to replace ξ_k.

The aggregate subgradient and error are used to formulate an implementable stopping condition for the bundle algorithm. The motivation behind that becomes clear with the following lemma.

Lemma 3.1 ([9, Theorem 6.68, p. 387]) *Let $X = \mathbb{R}^n$. Let $\varepsilon > 0$, $\hat{x}^k \in \mathbb{R}^n$ and $g^j \in \partial f(x^j)$ for $j \in J^k$. Then the set*

$$\mathcal{G}_\varepsilon^k := \left\{ \sum_{j \in J^k} \alpha_j g^j \;\middle|\; \sum_{j \in J^k} \alpha_j e_j \leq \varepsilon, \sum_{j \in J^k} \alpha_j = 1, \alpha_j \geq 0, j \in J^k \right\}$$

is a subset of the ε-subdifferential of $f(\hat{x}^k)$

$$\mathcal{G}_\varepsilon^k \subset \partial_\varepsilon f(\hat{x}^k).$$

This means that in the unconstrained case $G^k \in \partial_{E_k} f(\hat{x}^k)$. So driving $\|G^k\|$ and E_k to zero results in some approximate ε-optimality of the objective function. In the constrained case the stopping condition is written as

$$\delta_k = E^k + t_k \|G^k + \nu^k\|^2 \leq \texttt{tol},$$

for a fixed tolerance $\texttt{tol} > 0$.

The decrease measure δ_k is also taken for the decrease test. The relation

$$\begin{aligned}
\delta_k &= E^k + t_k \|G^k + \nu^k\|^2 \\
&= E^k - \left\langle G^k, d^k \right\rangle - \left\langle \nu^k, d^k \right\rangle \\
&= f(\hat{x}^k) - m_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle,
\end{aligned}$$

where (3.17) and (3.18) were used, shows that the new δ_k is only a small variation of the model decrease δ_k^M. If the iterate x^{k+1} does not lie on the boundary of the constraint set X, the vector ν^k is equal to zero and the expression simplifies to the one stated in (3.4).

For the model update the following two conditions are assumed to be fulfilled in consecutive null steps:

$$m_{k+1}(\hat{x}^k + d) \geq f(\hat{x}^{k+1}) - e_{k+1}^{k+1} + \left\langle g^{k+1}, d \right\rangle \quad \forall d \in \mathbb{R}^n \text{ and} \tag{3.19}$$

$$m_{k+1}(\hat{x}^k + d) \geq a_k(\hat{x}^k + d) \quad \forall d \in \mathbb{R}^n. \tag{3.20}$$

The first condition means that the newly computed information is always put into the bundle. The second one is important when updating the bundle index set J^k. It holds trivially if no or only inactive information j with $\alpha_j = 0$ is removed [13]. It is also always satisfied if the aggregate linearization a_k itself is added to the bundle. In this case active information can be removed without violating the condition. This is the key idea of Kiwiel's aggregation technique and ensures that the set $\{j \in J^k \mid \alpha_j > 0\}$ can be

bounded.

An issue of bundle methods is that in spite of the possibility to delete inactive information the bundle can still become very large. Kiwiel therefore proposed a totally different use of the aggregate objects in [21]. The aggregate subgradient can be used to build the *aggregate linearization*

$$a_k(\hat{x}^k + d) := m_k(x^{k+1}) + \langle G^k, d - d^k \rangle.$$

This function can be used to avoid memory overflow as it compresses the information of all bundle elements into one affine plane. Adding the function a_k to the cutting plane model preserves the assumptions (3.19) and (3.20) put on the model and can therefore be used instead of or in combination with the usual cutting planes.

This can however impair the speed of convergence if the bundle is kept too small and provides hence less information about the objective function [6, p. 654].

3.4 The Algorithm

We have now all the ingredients so that the following basic bundle algorithm can be stated:

Algorithm 3.1: Basic Bundle Method

Select a descent parameter $m \in (0, 1)$ and a stopping tolerance $\texttt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute $f(x^1)$ and g^1. Set the initial index set $J_1 := \{1\}$ and the initial stability center to $\hat{x}^1 := x^1$. Set $f(\hat{x}^1) = f(x^1)$ and select $t_1 > 0$.
For $k = 1, 2, 3 \ldots$

1. Calculate
$$d^k = \arg\min_{d \in \mathbb{R}^n} m_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2t_k}\|d\|^2$$
and the corresponding Lagrange multipliers $\alpha_j^k,\ j \in J_k$.

2. Set
$$G^k = \sum_{j \in J_k} \alpha_j^k g_j^k,$$
$$E_k = \sum_{j \in J_k} \alpha_j^k e_j^k \text{ and}$$

$$\delta_k = E_k + \frac{1}{t_k} d_k^2.$$

If $\delta_k \leq \texttt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.
4. Compute $f(x^{k+1}),\ g^{k+1}$.
   If $f(x^{k+1}) \leq f(\hat{x}^k) - m\delta_k \quad \rightarrow$ serious step:
   Set $\hat{x}^{k+1} = x^{k+1},\ f(\hat{x}^{k+1}) = f(x^{k+1})$ and select a suitable $t_{k+1} > 0$.
   Otherwise $\quad \rightarrow$ nullstep:
   Set $\hat{x}^{k+1} = \hat{x}^k,\ f(\hat{x}^{k+1}) = f(\hat{x}^k)$ and choose $t_{k+1} > 0$ in a suitable way.
5. Select the new bundle index set J_{k+1}, calculate e_j^{k+1} for all $j \in J_{k+1}$ and update the model m_{k+1}.

In steps 4 and 5 of the algorithm it is not specified how to update the parameter t_k, the index set J^k and the model m_k. For the convergence proof it is only necessary that $\liminf_{k\to\infty} t_k > 0$ and that conditions (3.19) and (3.20) are fulfilled.

In practice the choice of t_k can be realized by taking

$$t_{k+1} = \kappa_+ t_k, \quad \kappa_+ > 1 \tag{3.21}$$

at every serious step and

$$t_{k+1} = \max\{\kappa_- t_k, t_{min}\}, \quad \kappa_- < 1 \text{ and } t_{min} > 0 \tag{3.22}$$

at every null step. The idea behind this management of t_k is taken from the trust region method: If the computed iterate was good, the model is assumed to be reliable in a larger area around this serious iterate so bigger step sizes are allowed. If a null step was taken, the model seems to be too inaccurate far from the current serious point. Then smaller step sizes are used. A more sophisticated version of this kind of step size management is also used by Noll et al. in [42] and [40]. The trust region idea was very much exploited by Schramm and Zowe in [49]. In the case $X = \mathbb{R}^n$ the sequence $\{\hat{x}^k\}$ can be unbounded. In this case bounding $t_k \leq t_{max} < \infty$ for all k preserves the convergence proof [16, Theorem 3.2.2, p. 308].

In general it can be shown that if f posses global minima and the basic bundle algorithm generates the sequence $\{\hat{x}^k\}$, this sequence converges to a minimizer of problem (3.1) (c.f

[16]).

4 Variations of the Bundle Method

After their discovery in 1975 bundle methods soon became very successful. Only a few years later they were generalized to be used also with nonconvex objective functions. Early works that contain fundamental ideas still used for these algorithms are [33] and [20]. It then took over 25 years that bundle methods were again generalized to the use of inexact information, first works on this subject being [15, 22] and [51].

This section of the thesis shortly presents the key ideas of those two kinds of generalizations and different types of bundle methods that realize them. This is first done for the case of convex objective functions with inexact function values and/or subgradient information and then for nonconvex objective functions.

4.1 Convex Bundle Methods with Inexact Information

We focus here on *convex* bundle methods with inexact information. The reason for this is that there is a fundamental difference in treating inexactness between methods that assume convex and those that assume nonconvex objective functions. When dealing with nonconvex objective functions inexactness is treated as some additional nonconvexity therefore no additional strategies are used to cope with the noise. This is not possible if the convexity property is to be exploited for better convergence results. A throughout study on this subject including a synthetic convergence theory is done in [62]. Here the most important aspects of that paper are reviewed.

4.1.1 Different Types of Inexactness

Throughout this section we consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{4.1}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ is a finite convex function. The function values and one subgradient at each point x are given by an inexact oracle. It is reasonable to define different kinds of inexactness and further assumptions can be put on the noise to achieve stronger convergence results. However, generally inexact information for convex objective functions is defined in the following way:

$$f_x := f(x) - \sigma_x, \quad \sigma_x \leq \bar{\sigma} \text{ and} \tag{4.2}$$

$$g_x \in \mathbb{R}^n \text{ such that } f(\cdot) \geq f_x + \langle g_x, \cdot - x \rangle - \theta_x, \quad \theta_x \leq \bar{\theta}. \tag{4.3}$$

From this follows because of

$$f(\cdot) \geq f(x) + \langle g_x, \cdot - x \rangle - (\sigma_x + \theta_x) \tag{4.4}$$

that g_x is an ε-subgradient of $f(x)$ with $\varepsilon := \sigma_x + \theta_x \geq 0$ independently of the signs of the errors.

Different convergence results for the applied bundle methods are possible depending on if the bounds $\bar{\sigma}$ and $\bar{\theta}$ are unknown, known or even controllable.

In case of controllability of $\bar{\sigma}$ and $\bar{\theta}$ it may be possible to drive them to zero as the iterations increase such that $\lim_{k\to\infty} \sigma_k = 0$ and $\lim_{k\to\infty} \theta_k = 0$. We talk then of *asymptotically vanishing errors*. This case is important because it allows convergence to the exact minimum of the problem even if function values and subgradients are erroneous. In the case of $\bar{\theta} = 0$ it even suffices to show that the errors are only asymptotically exact for descent steps [23]. This observation was the motivation for the partly inexact bundle methods presented in [23] and [62]. The idea is to calculate a value of the objective function with a demanded accuracy (which is finally going to be exact) only if a certain target descent γ_x is reached. This approach can save a lot of (unnecessary) computational effort while still enabling convergence to the exact minimum (c.f. [62]).

In view of good convergence properties oracles that only underestimate the true function, so called *lower oracles*, are also very interesting. Lower oracles provide f_x and g_x such that $f_x \leq f(x)$ and $f(\cdot) \geq f_x + \langle g_x, \cdot - x \rangle$. That means the cutting plane model is always minorizing the true function as it is the case for exact information. In this case if the value to approximate the optimal function value is chosen properly, it is not necessary to include any new steps into the method to cope with the inexactness, such as noise attenuation [62, Corollary 5.2, p. 256].

4.1.2 Noise Attenuation

In the case of inexact information, especially if the inexact function value can overestimate the real one, it is possible that the aggregate linearization error E_k becomes very small (or

even negative) even though the current iterate is far from the minimum of the objective function. To tackle this problem the authors propose a procedure called *noise attenuation* that was developed in [15] and [22]. The basic idea is to allow bigger step sizes t_k whenever the algorithm comes in the situation described above. This ensures that either some significant descent towards the real minimum can be done or shows that the point where the algorithm is stuck is actually such a minimum. Noise attenuation is triggered when E_k or respectively the descent measure δ_k that is used for the descent test is negative. A more detailed description is given in [62].

4.1.3 Convergence Results

Depending on the kind of error many slightly different convergence results can be proven for bundle methods that handle convex objective functions with inexact information. In case of the general error defined in (4.2) and (4.3) it can be shown that for bounded sequences $\{\hat{x}^k\}$ every accumulation point $\bar{x}$ of an infinite series of serious steps or the last serious iterate before an infinite tail of null steps is a $\bar{\sigma}$-solution of the problem meaning that

$$f(\bar{x}) \leq f^* + \bar{\sigma}$$

with f^* being an exact solution of problem (4.1).

Generally for asymptotically vanishing errors it is possible to construct bundle methods very similar to the basic bundle method that converge to the exact minimum of the problem. For more detailed results refer to [62].

4.2 Nonconvex Bundle Methods with Exact Information

In the nonconvex case the optimization problem is the following:

$$\min_{x \in \mathbb{R}^n} f(x). \tag{4.5}$$

This time $f : \mathbb{R}^n \to \mathbb{R}$ is a finite, locally Lipschitz function. It is neither expected to be convex nor differentiable.

In the case of inexactness in convex bundle methods, where a lot of different assumptions can be put on the errors to reach different convergence results, the strategy to cope

with these errors remains very much the same. In contrast to this in case of nonconvex objective functions the set of functions to be studied is rather uniform still there exist very different approaches to tackle the problem. As the nonnegativity property of the linearization errors e_j^k is crucial for the convergence proof of convex bundle methods an early idea was forcing the errors to be so by different downshifting strategies. A very common one is using the *subgradient locality measure* [21, 33]. Here the linearization error is essentially replaced by the nonnegative number

$$\tilde{e}_j^k := \max_{j \in J_k} \{|e_j^k|, \gamma \|\hat{x}^k - x^j\|^2\}$$

or a variation of this expression.

The expression gradient locality measure comes from the dual point of view, where the aggregate linearization error provides a measure for the distance of the calculated ε-subgradient to the objective function.

Methods that use downshifting for building the model function are often endowed with a line search to provide sufficient decrease of the objective function. For the linesearch to terminate finitely usually semismoothness of the objective function is needed.

4.2.1 Proximity Control

Instead of using line search it is also possible to do *proximity control*. This means that the step size parameter t_k is managed in a smart way to ensure the right amount of decrease in the objective function. This method is very helpful in the case of nonconvex objective functions with inexact information as it is predominantly considered in this thesis.

As inexactness can be seen as a kind of slight nonconvexity one could be tempted to think that nonconvex bundle methods are destined to be extended to the inexact case. Indeed, the two existing algorithms [13] and [40] that deal with both nonconvexity and inexactness are both extensions of a nonsmooth bundle method. This is however seldom possible for algorithms that employ a line search because for functions with inexact information convergence of this subroutine cannot be proven.

To this end proximity control seems to be a very promising strategy. It is used in many different variations in [1, 31, 39, 41, 42] and [49].

4.2.2 Other Concepts

In the beginning bundle methods were mostly explored from the dual point of view. Newer concepts focus also on the primal version of the method. This invokes for example having different model functions for the subproblem.

In [7] and [8] the difference function

$$h(d) := f(x^j + d) - f(x^j), \quad j \in J_k$$

is approximated to find a descent direction of f. The negative linearization errors are addressed by using two different bundles. One contains the indices with nonnegative linearization errors and one contains the other ones. From these two bundles two cutting plane approximations can be constructed which provide the bases for the calculation of new iterates.

In [42] Noll et al. follow an approach of approximating a local model of the objective function. The model can be seen as a nonsmooth generalization of the Taylor expansion and looks the following:

$$\Phi(y, x) = \phi(y, x) + \frac{1}{2}(y - x)^\top Q(x)(y - x).$$

The so called *first order model* $\phi(., x)$ is convex but possibly nonsmooth and can be approximated by cutting planes. The *second order part* is quadratic but not necessarily convex. The algorithm proceeds similarly to a general bundle algorithm. Instead of a line search it uses proximity control to ensure convergence.

Generally for all of these methods convergence to a stationary point is established under the assumptions of a locally Lipschitz objective function and bounded level sets $\{x \in \mathbb{R}^n \mid f(x) \leq f(\hat{x}^1)\}$. If the method uses a line search additionally semismoothness of the objective function is needed.

In [40] the second order approach of [42] is extended to functions with inexact information. As far as we know this is the only other bundle method that can deal with nonconvexity and inexactness in both the function value and subgradient. In this method a lower-$\mathcal{C}^1$ objective function and some assumptions on the form of inexactness are needed to prove convergence.

The above algorithm inspires the variable metric variation of the method used by Hare et al. in [13] that is presented in section 6 of this thesis.

5 Proximal Bundle Method for Nonconvex Functions with Inexact Information

This section focuses on the proximal bundle method presented by Hare et al. in [13]. The idea is to extend the basic bundle algorithm for nonconvex functions with both inexact function and subgradient information. The key idea of the algorithm is the one already developed by Hare and Sagastizábal in [12]: When dealing with nonconvex functions a very critical difference to the convex case is that the linearization errors are not necessarily nonnegative any more. To tackle this problem the errors are manipulated to enforce nonnegativity. In this case this is done my modeling not the objective function directly but a convexified version of it.

5.1 Derivation of the Method

Throughout this section we consider the optimization problem

$$\min_x f(x) \quad \text{s.t.} \quad x \in X. \tag{5.1}$$

The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz and (subdifferentially) regular. $X \subset \mathbb{R}^n$ is assumed to be a convex compact set.

Definition 5.1 [47, Theorem 7.25, p. 260] $f : \mathbb{R}^n \to \mathbb{R}$ is called *subdifferentially regular* at $\bar{x} \in \mathbb{R}^n$ if the epigraph

$$epi(f) := \{(x, \alpha) \in \mathbb{R}^n \times \mathbb{R} \mid \alpha \geq f(x)\}$$

is Clarke regular at $\bar{x}, f(\bar{x})$.

Clarke regularity basicly means, that epigraph of a function f does not have any 'inward' corners. An exact definition is given in [47, Definition 6.4, p. 199].

5.1.1 Inexactness

It is assumed that both the function value as well as one element of the subdifferential can be provided in an inexact form. For the function value inexactness is defined straight forwardly: If

$$\|f_x - f(x)\| \leq \sigma_x,$$

then f_x approximates the value $f(x)$ within σ_x. This is slightly different from the definition in (4.2). In the convex case it follows from (4.4) that $\bar{\sigma} \geq \sigma_x \geq -\theta_x \geq -\bar{\theta}$ and therefore $f_x \in [f(x) - \bar{\theta}, f(x) + \bar{\sigma}]$ for the overall error bounds $\bar{\sigma}$ and $\bar{\theta}$.

As the 'normal' ε-subdifferential is not defined for nonconvex functions we adopt the notation used in [40] and interpret inexactness in the following way: $g \in \mathbb{R}^n$ approximates a subgradient of $\partial f(x)$ within $\theta \geq 0$ if

$$g_x \in \partial f(x) + B_\theta(0) =: \partial_{[\theta]} f(x)$$

where $\partial f(x)$ is the Clarke subdifferential of f.

The given definition of the inexactness can be motivated by the relation

$$g_x \in \partial_{[\theta]} f(x) \Leftrightarrow g_x \in \partial(f + \theta \| \cdot - x \|)(x)$$

noticed in [55]. It means that the approximation of the subgradient of $f(x)$ is an exact subgradient of a small perturbation of f at x. The set $\partial_{[\varepsilon]} f(x)$ is also known as the Fréchet ε-subdifferential of $f(x)$.

Remark: For convex objective functions this approximate subdifferential does *not* equal the usual convex ε-subdifferential. The two can however be related via

$$\partial_\theta f(x) \subset \partial_{[\theta']} f(x)$$

for a suitable θ'. Generally an explicit relation between θ and θ' is hard to find [40, p. 558].

Like in the paper it is assumed that the errors are bounded although the bound does not have to be known:

$$|\sigma_j| \leq \bar{\sigma}, \bar{\sigma} > 0 \quad \text{and} \quad 0 \leq \theta_j \leq \bar{\theta} \quad \forall j \in J^k \text{ and } \forall k.$$

For ease of notation we write from now on f_j and g_j instead of f_{x^j} and g_{x^j} for the approximation of the function value and subgradient at the j'th iterate in the bundle J^k. The approximation at the k'th stability center reads $\hat{f}_k$.

Remark: Before the exact subgradient was denoted with g_j. Here the same notation is used for the approximate subgradient. We leave this double notation for the sake of

readability and remark that for the remainder of this thesis g_j denotes the approximate subgradient. In the few situations, where the exact gradient is needed, it is marked clearly.

5.1.2 Nonconvexity

A main issue both nonconvexity and inexactness entail is that the linearization errors e_j^k are not necessarily nonnegative any more. So based on the results in [61] not the objective function but a convexified version of it is modeled as the objective function of the subproblem.

As already pointed out in section 3.2 the bundle subproblem can be formulated by means of the prox-operator (3.10).

The idea is to use the relation

$$prox_{T=\frac{1}{\eta}+t,f}(x) = prox_{t,f+\eta/2\|\cdot-x\|^2}(x).$$

This means, that the proximal point of the function f for the parameter $T = \frac{1}{\eta} + t$, $\eta, t > 0$, is the same as the one of the convexified function

$$\tilde{f}(y) = f(y) + \frac{\eta}{2}\|y - x\|^2 \tag{5.2}$$

with respect to the parameter t [12]. The parameter η is therefore also called the *convexification parameter* and t the *prox-parameter*.

The main difference of the method in [13] to the basic bundle algorithm 3.1 is that the function that is modeled by the cutting plane model is no longer the original objective function f but the convexified version $\tilde{f}$. This results in the following changes:

In addition to downshifting the linear functions forming the model they have a tilted slope. This is because instead of subgradients of the original objective f subgradients of the function $\tilde{f}$ are taken. We call them *augmented subgradients*. At the iterate x^j such a subgradient is given by

$$s_j^k := g^j + \eta_k \left(x^j - \hat{x}^k\right).$$

Downshifting is done in a way that keeps the linearization error nonnegative. The *aug-*

mented linearization error is therefore defined as

$$0 \leq c_j^k := e_j^k + b_j^k, \quad \text{with} \quad \begin{cases} e_j^k := \hat{f}_k - f_j - \left\langle g^j, \hat{x}^k - x^j \right\rangle \\ b_j^k := \frac{\eta_k}{2} \|x^j - \hat{x}^k\|^2 \end{cases}$$

and

$$\eta_k \geq \max \left\{ \max_{j \in J_k, x^j \neq \hat{x}^k} \frac{-2e_j^k}{\|x^j - \hat{x}^k\|^2}, 0 \right\} + \gamma.$$

The parameter $\gamma \geq 0$ is a safeguarding parameter to keep the calculations numerically stable.

The new model function can therefore be written as

$$M_k(\hat{x}^k + d) := \hat{f}_k + \max_{j \in J_k} \left\{ \left\langle s_j^k, d \right\rangle - c_j^k \right\}. \tag{5.3}$$

At the proximal center $\hat{x}^k$ holds $M_k(\hat{x}^k) = \hat{f}_k$ for all k by the fact that then $d = 0$ and $c_j^k = 0$.

5.1.3 Aggregate Objects

The definitions of the *augmented aggregate subgradient* S^k, *error* C_k and *linearization* A_k follow straightforwardly from the KKT-conditions. Again $\alpha_j^k, j \in J^k$ denote the Lagrangian multipliers of the subproblem.

$$S^k := \sum_{j \in J_k} \alpha_j^k s_j^k, \tag{5.4}$$

$$C_k := \sum_{j \in J_k} \alpha_j^k c_j^k \text{ and} \tag{5.5}$$

$$A_k(\hat{x}^k + d) := M_k(x^{k+1}) + \left\langle S^k, d - d^k \right\rangle. \tag{5.6}$$

The model decrease is

$$\delta^k := C_k + t_k \|S^k + \nu^k\|^2 = C_k + \frac{1}{t_k} \|d^k\|^2, \tag{5.7}$$

which contains the normal vector

$$\nu^k \in \partial \mathtt{i}_X(x^{k+1}) \tag{5.8}$$

of the constraint set X.

The second formulation in (5.7) follows from the relation $d^k = -t_k(S^k + \nu^k)$ which itself comes from the KKT-conditions.

By the same argumentation as for (3.18) the KKT conditions also reveal another useful characterization of the augmented aggregate linearization error:

$$C_k = \hat{f}_k - M_k(x^{k+1}) + \left\langle S^k, d^k \right\rangle. \tag{5.9}$$

As the model function M_k is convex even for nonconvex objective functions it is still minorized by the aggregate linearization. It holds

$$A_k(\hat{x}^k + d) \leq M_k(\hat{x}^k + d) \quad \forall d \in \mathbb{R}^n. \tag{5.10}$$

The update of t_k can be done in the same way as described in (3.21) and (3.22) for the basic bundle method. Similarly the methods to update the bundle index set J^k stay valid. The update conditions (3.19) and (3.20) for the model are now written with respect to the augmented aggregate linearization and the approximate function value $\hat{f}_{k+1}$.

$$M_{k+1}(\hat{x}^k + d) \geq \hat{f}_{k+1} - c_{k+1}^{k+1} + \left\langle s^{k+1}, d \right\rangle, \quad \forall d \in \mathbb{R}^n \text{ and} \tag{5.11}$$

$$M_{k+1}(\hat{x}^k + d) \geq A_k(\hat{x}^k + d), \quad \forall d \in \mathbb{R}^n. \tag{5.12}$$

A bundle algorithm that deals with nonconvexity and inexact function and subgradient information can now be stated.

Algorithm 5.1: Nonconvex Proximal Bundle Method with Inexact Information

---

Select parameters $m \in (0,1), \gamma > 0$ and a stopping tolerance $\mathtt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute f_1 and g^1. Set the initial index set $J_1 := \{1\}$ and the initial prox-center to $\hat{x}^1 := x^1$. Set $\hat{f}_1 = f_1$ and select $t_1 > 0$.
For $k = 1, 2, 3, \ldots$

1. Calculate
$$d^k = \arg\min_{d \in \mathbb{R}^n} \left\{ M_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2t_k}\|d\|^2 \right\}.$$

2. Set
$$G^k = \sum_{j \in J_k} \alpha_j^k s_j^k$$
$$C_k = \sum_{j \in J_k} \alpha_j^k c_j^k \text{ and}$$
$$\delta_k = C_k + \frac{1}{t_k}\|d^k\|^2.$$
If $\delta_k \leq \mathtt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.

4. Compute f^{k+1}, g^{k+1}.
If $f^{k+1} \leq \hat{f}^k - m\delta_k \quad \rightarrow$ serious step:
Set $\hat{x}^{k+1} = x^{k+1}$, $\hat{f}^{k+1} = f^{k+1}$ and select $t_{k+1} > 0$.
Otherwise $\quad \rightarrow$ nullstep:
Set $\hat{x}^{k+1} = \hat{x}^k$, $\hat{f}^{k+1} = \hat{f}^k$ and choose $0 < t_{k+1} \leq t_k$.

5. Select new bundle index set J_{k+1}, calculate
$$\eta_{k+1} = \max\left\{ \max_{j \in J_{k+1}, x^j \neq \hat{x}^{k+1}} \frac{-2e_j^{k+1}}{|x^j - \hat{x}^{k+1}|^2}, 0 \right\} + \gamma$$
and c_j^{k+1} for all $j \in J^{k+1}$. Update the model M_k.

5.2 On Different Convergence Results

In terms of usability of the described algorithm it is interesting to see if stronger convergence results are possible if additional assumptions are put on the objective function. This is investigated in the following section.

5.2.1 The Constraint Set

The constraint set X ensures the boundedness of the sequence $\{\hat{x}^k\}$. This is not necessary if the objective function is assumed to have bounded level sets $\{x \in \mathbb{R}^n \mid f(x) \leq f(\hat{x}^1)\}$, an assumption commonly used when optimizing nonconvex functions. As the objective

function is assumed to be continuous bounded level sets are compact. Additionally the descent test ensures that $f(\hat{x}^{k+1}) \leq f(\hat{x}^k)$ for all k. The proof holds therefore in the same way as with the set X.

In [62] another stopping criterion is proposed that ensures convergence even for unbounded sequences $\{\hat{x}^k\}$.

5.2.2 Exact Information and Vanishing Errors

As the presented algorithm was originally designed for nonconvex objective functions where function values as well as subgradients are available in an exact manner, all convergence results stay the same with the error bounds $\bar{\sigma} = \bar{\theta} = 0$. As already indicated previously this is the case because inexactness can be seen as a kind of nonconvexity and no additional concepts had to be added to the method when generalizing it to the inexact setting.

If we additionally require the objective function to be lower-$\mathcal{C}^2$ it can be proven that the sequence $\{\eta_k\}$ is bounded [12, Lemma 3, p. 2454]. This is not possible in the case of inexact information even for convex objective functions (see example in [13, p. 22]).

For asymptotically vanishing errors, meaning $\lim_{k\to\infty} \sigma_k = 0$ and $\lim_{k\to\infty} \theta_k = 0$ the convergence theory holds equally well with error bounds $\bar{\sigma} = \bar{\theta} = 0$ in [13, Lemma 5, p. 11]. Still it is difficult if not impossible to show that the sequence $\{\eta_k\}$ is bounded without further assumptions. Under the assumption that f is lower-$\mathcal{C}^2$ and some continuity bounds on the errors

$$\frac{|\sigma_j - \hat{\sigma}_k|}{\|x^j - \hat{x}^k\|^2} \leq L_\sigma, \qquad \frac{\theta_j}{\|x^j - \hat{x}^k\|} \leq L_\theta \quad \forall k \text{ and } \forall j \in J_k$$

boundedness of the sequence $\{\eta_k\}$ can be shown. The question remains however if those assumptions are possible to be assured in practice.

5.2.3 Convex Objective Functions

An obvious gain when working with convex objective functions is that the approximate stationarity condition of [13, Lemma 5 (iii), p. 12] is then an approximate optimality condition. If one takes the error definitions (4.2) and (4.3) that are available in the convex case and assumes $X = \mathbb{R}^n$, statement (22) in [13, p. 12] therefore means that

$$0 \in \partial_{\bar{\sigma}+\bar{\theta}} f(\bar{x}).$$

Thus $\bar{x}$ is $(\bar{\sigma} + \bar{\theta})$-optimal.

This follows from the definition of S^k in (5.4) and local Lipschitz continuity of the ε-subdifferential [47, Proposition 12.68, p. 573].

To conclude this section we can say: At the moment there exist two fundamentally different approaches to tackle inexactness in various bundle methods depending on if the method is developed for convex or nonconvex objective functions. In the nonconvex case inexactness is only considered in the paper by Hare, Sagastizàbal and Sodolov [13] presented above and Noll [40]. In these cases the inexactness can be seen as an additional nonconvexity. In practice this means that the algorithm can be taken from the nonconvex case with no or only minor changes. This includes that all results of the exact case remain true as soon as function and subgradient are evaluated in an exact way. In case of convex objective functions with inexact information stronger convergence results are possible. However to be able to exploit convexity in order to achieve those results the algorithms look different from those designed for nonconvex objective functions and are generally not able to deal with such functions.

6 Variable Metric Bundle Method

A way to extend the proximal bundle method is to use an arbitrary metric $\frac{1}{2}\langle d, W_k d\rangle$ with a symmetric and positive definite matrix W_k instead of the Euclidean metric for the stabilization term $\frac{1}{2t_k}\|d\|^2$. Methods doing so are called *variable metric bundle methods*. This section combines the method of Hare et al. presented in section 5 with the second order model function used by Noll in [40] to a metric bundle method suitable for nonconvex functions with noise.

The section starts by explaining the ideas from [40] used to extend the method presented above. It then gives a convergence proof for the developed method and concludes with an explicit strategy how to update the metric during the steps of the algorithm.

Throughout this section we still consider the optimization problem (5.1). We also keep the names and definitions of the objects used in section 5.

6.1 Main Ingredients to the Method

As already mentioned in section 3 the stabilization term can be interpreted in many different ways. In the context of this section we can understand it as a pretty rough approximation of the curvature of the objective function. In this thesis we work with locally Lipschitz continuous functions. Rademacher's theorem [14, Theorem 3.1, p. 18] states that on any open set $U \subset \mathbb{R}^n$ such a function is differentiable at almost every point in U. The set of points where the function is nondifferentiable is of zero Lebesgue measure. This means that between those points curvature information can be present and can be used to speed up convergence.

6.1.1 Variable Metric Bundle Methods

Variable metric bundle methods use an approach that can be motivated by the thoughts stated above. Instead of using the Euclidean norm for the stabilization term $\frac{1}{2t_k}\|d\|^2$ the metric is derived from a symmetric and positive definite matrix W_k. As the name of the method suggests, this matrix can vary over the iterations of the algorithm. The subproblem in the k'th iteration therefore reads

$$\min_{\hat{x}^k + d \in \mathbb{R}^n} M_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2}\langle d, W_k d\rangle .$$

As explained in [28, chapter XV.4] like (3.9) this is a Moreau-Yosida regularization of the objective function (on the constraint set), so this subproblem is still strictly convex and has a unique solution. It is however harder to solve especially if the matrices W_k are no diagonal matrices [32, p. 594]. In the unconstrained case or for a very simple constraint set the subproblem can be solved by calculating a quasi Newton step. Such a method is presented by Lemaréchal and Sagastizábal in [29] for convex functions. Lukšan and Vlček use an algorithm in those lines in [60] which is adapted to a limited memory setting by Haarala et al. in [11].

A challenging question is how to update the matrices W_k. It is important that the updating strategy preserves positive definiteness of the matrices and that the matrices stay bounded. The updates that are used most often are the symmetric rank 1 formula (SR1 update) and the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update. These updates make it possible to assure the required conditions with only little extra effort even in the nonconvex case. Concrete instances of the updates are given in [60] and [28].

6.1.2 Noll's Second Order Model

In [42] Noll et al. present a proximal bundle method for nonconvex objective functions. An important ingredient to the method is that not the objective function itself is approximated in the subproblem but a quadratic model of it:

$$\Phi(x, \hat{x}) = \phi(x, \hat{x}) + \frac{1}{2} \langle x - \hat{x}, Q(\hat{x})(x - \hat{x}) \rangle \tag{6.1}$$

The first order model $\phi(\cdot, \hat{x})$ is convex and possibly nonsmooth. The second order part $\frac{1}{2} \langle \cdot - \hat{x}, Q(\hat{x})(\cdot - \hat{x}) \rangle$ is quadratic but not necessarily convex.

As the first order part of this model is convex it can be approximated by a cutting plane model just like the objective function in usual convex bundle methods. The subproblem emerging from this approach is

$$\min_{\hat{x}^k + d} m_k(\hat{x}^k + d) + \frac{1}{2} \left\langle d, Q(\hat{x}^k) d \right\rangle + \frac{1}{2t_k} \|d\|^2$$

where m_k is the usual cutting plane model (3.2) for the convex, nonsmooth function ϕ.

The matrix $Q(\hat{x}^k)$ itself does not have to be positive definite. In fact the only conditions put on this matrix are that it is symmetric and that all eigenvalues are uniformly bounded. We adopt the notation in [40] and write

$$Q(\hat{x}^k) := Q_k = Q_k^\top \quad \text{and} \quad -q\mathbb{I} \preccurlyeq Q_k \preccurlyeq q\mathbb{I} \text{ for a } q > 0.$$

The notation $A \preccurlyeq B$ with $A, B \in \mathbb{R}^{n\times n}$ means that the matrix $(B - A)$ is positive semidefinite. The symbol $\prec$ means analogously that $(B - A)$ is positive definite.

As the matrix Q_k is symmetric it can also be pulled into the stabilization term. This means the k'th bundle subproblem can also be written as

$$\min_{\hat{x}^k+d\in X} m_k(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle. \tag{6.2}$$

During the algorithm it is assured that $W_k = Q_k + \frac{1}{t_k}\mathbb{I}$ is positive definite, so this is a variable metric subproblem.

Instead of the first order model $\phi(\cdot, \hat{x})$ the convexified objective function (5.2) is used. In the subproblem this function is approximated by the augmented cutting plane model M_k given in (5.3). The final subproblem of this variable metric bundle algorithm is then

$$\min_{\hat{x}^k+d\in X} M_k(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle. \tag{6.3}$$

The decomposition of the stabilization term into a curvature approximation and a proximal term makes is easier to reach two goals at the same time:

One the one hand, curvature of the objective can be approximated only under the conditions of the boundedness and symmetry of Q_k. No positive definiteness has to be ensured for convergence. On the other hand the proximal term can be used in the trust region inspired way to make a line search obsolete. As already mentioned in section 4 this is an advantage especially when working with inexact functions where a line search is not useable.

comment on line search and curve search in [28, 29, 60]?

6.1.3 The Descent Measure

Due to the different formulation of subproblem (6.3) the descent measure δ_k has to be adapted in the variable metric bundle method. In the same way as for (3.16) from the optimality condition

$$0 \in \partial M_k(x^{k+1}) + \partial \mathtt{i}_X(x^{k+1}) + \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k$$

it follows that

$$S^k + \nu^k = -\left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k, \tag{6.4}$$

S^k and ν^k being the augmented aggregate subgradient and outer normal defined in (5.4) and (5.8) respectively.

From this the model decrease (5.7) can be calculated using (5.6), (5.9) and (6.4):

$$\begin{aligned}
\delta_k &:= \hat{f}_k - M_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\
&= \hat{f}_k - A_k(x^{k+1}) - \left\langle \nu^k, d^k \right\rangle \\
&= C_k - \left\langle S^k + \nu^k, d^k \right\rangle \\
&= C_k + \left\langle d^k, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \quad \forall k \in \mathbb{N}.
\end{aligned} \tag{6.5}$$

The new δ_k is used in the same way as in algorithm 5.1 for the descent test and stopping conditions.

Because the changes in the algorithm concern only the stabilization and the decrease measure δ_k all other relations that were obtained for the different parts of the model M_k in section 5 are still valid.

6.2 The Variable Metric Bundle Algorithm

The variable metric bundle algorithm can now be stated as a variation of algorithm 5.1.

Algorithm 6.1: Nonconvex Variable Metric Bundle Method with Inexact Information

---

Select parameters $m \in (0,1), \gamma > 0, q > 0, 0 < t_{min} < \frac{1}{q}$ and a stopping tolerance $\mathtt{tol} \geq 0$. Choose a starting point $x^1 \in \mathbb{R}^n$ and compute f_1 and g^1. Set the initial metric matrix $Q_1 = \mathbb{I}$, the initial index set $J_1 := \{1\}$ and the initial prox-center to $\hat{x}^1 := x^1$. Set $\hat{f}_1 = f_1$ and select $t_1 > 0$.
For $k = 1, 2, 3, \ldots$

1. Calculate

$$d^k = \arg\min_{d\in\mathbb{R}^n} \left\{ M_k(\hat{x}^k + d) + \mathtt{i}_X(\hat{x}^k + d) + \frac{1}{2}\left\langle d, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d\right\rangle \right\}.$$

2. Set

$$\begin{aligned} G^k &= \sum_{j\in J_k} \alpha_j^k s_j^k, \\ C_k &= \sum_{j\in J_k} \alpha_j^k c_j^k \text{ and} \\ \delta_k &= C_k + \left\langle d^k, \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle. \end{aligned}$$

   If $\delta_k \leq \mathtt{tol} \rightarrow$ STOP.

3. Set $x^{k+1} = \hat{x}^k + d^k$.

4. Compute f^{k+1}, g^{k+1}.
   If $f^{k+1} \leq \hat{f}^k - m\delta_k \quad \rightarrow$ serious step:
   Set $\hat{x}^{k+1} = x^{k+1}, \hat{f}^{k+1} = f^{k+1}$ and calculate a symmetric matrix Q_{k+1} with $-q\mathbb{I} \preccurlyeq Q_{k+1} \preccurlyeq q\mathbb{I}$.
   Adjust t_{k+1} such that $Q_{k+1} + \frac{1}{t_{k+1}}\mathbb{I} \succ 0$ and $t_{k+1} > t_{min}$.
   Otherwise $\quad \rightarrow$ nullstep:
   Set $\hat{x}^{k+1} = \hat{x}^k, \hat{f}^{k+1} = \hat{f}^k$ and choose $t_{min} < t_{k+1} \leq t_k$.

5. Select the new bundle index set J_{k+1}. Calculate

$$\eta_{k+1} = \max\left\{ \max_{j\in J_{k+1}, x^j\neq\hat{x}^{k+1}} \frac{-2e_j^{k+1}}{|x^j - \hat{x}^{k+1}|^2}, 0 \right\} + \gamma$$

   and c_j^{k+1} for all $j \in J^{k+1}$. Update the model M^{k+1}.

6.3 Convergence Analysis

In this section the convergence properties of the new method are analyzed. We do this the same way it is done by Hare et al. in [13].

In the paper all convergence properties are first stated in [13, Lemma 5]. It is then shown that all sequences generated by the method meet the requirements of this lemma which we repeat here for convenience.

Lemma 6.1 ([13, Lemma 5]) *Suppose that the cardinality of the set* $\{j \in J^k \mid \alpha_j^k > 0\}$ *is uniformly bounded in* k.

(i) If $C^k \to 0$ *as* $k \to \infty$, *then*

$$\sum_{j \in J^k} \alpha_j^k \|x^j - \hat{x}^k\| \to 0 \text{ as } k \to \infty.$$

(ii) If additionally for some subset $\tilde{K} \subset \{1, 2, \dots\}$,

$$\hat{x}^k \to \bar{x}, S^k \to \bar{S} \text{ as } K \ni k \to \infty, \text{ with } \{\eta_k \mid k \in \tilde{K}\} \text{ bounded,}$$

then we also have

$$\bar{S} \in \partial f(\bar{x}) + B_{\bar{\theta}}(0).$$

(iii) If in addition $S^k + \nu^k \to 0$ *as* $\tilde{K} \in k \to \infty$, *then* $\bar{x}$ *satisfies the approximate stationarity condition*

$$0 \in (\partial f(\bar{x}) + \partial \mathtt{i}_X(\bar{x})) + B_{\bar{\theta}}(0). \tag{6.6}$$

(iv) Finally if f *is also lower-*$\mathcal{C}^1$, *then for each* $\varepsilon > 0$ *there exists* $\rho > 0$ *such that*

$$f(y) \geq f(\bar{x}) - (\bar{\theta} + \varepsilon)\|y - \bar{x}\| - 2\bar{\sigma}, \quad \text{for all } y \in X \cap B_\rho(\bar{x}). \tag{6.7}$$

As neither the stabilization nor the descent test are involved in the proof of lemma 6.1 it is the same as in [13].

We prove now that also the variable metric version of the algorithm fulfills all requirements of lemma 6.1. The proof is divided into two parts. The first case covers the case of infinitely many serious steps, the second one considers infinitely many null steps after a finite number of serious steps.

For both proofs the equivalence of norms is used between the Euclidean norm and the norm $\|\cdot\|_{Q_k + \frac{1}{t_k}\mathbb{I}}$. We show here shortly, that the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ can be used to define a scalar product which induces the norm $\|\cdot\|_{Q_k + \frac{1}{t_k}\mathbb{I}}$.

In order to do this, it is first shown, that the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is bounded.

Proposition 6.2 The matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is bounded in the sense that $\left\|Q_k + \frac{1}{t_k}\mathbb{I}\right\|_2 < \infty$ for all k and the spectral norm $\|\cdot\|_2$ [48, Example 5.6.6].

This means also that the following relation holds for all vectors $x \in \mathbb{R}^n$:

$$\left\| \left(Q_k + \frac{1}{t_k}\mathbb{I} \right) x \right\| \leq \left| q + \frac{1}{t_{min}} \right| \|x\| < \infty, \quad \forall k, \tag{6.8}$$

where $q > 0$ is the bound on the eigenvalues of Q_k and t_{min} the lower bound for the step size.

Proof:

Algorithm 6.1 ensures that $-q\mathbb{I} \preccurlyeq Q_k \preccurlyeq q\mathbb{I}$ for some preset constant $q > 0$. This means, that the matrices $Q_k + q\mathbb{I}$ and $q\mathbb{I} - Q_k$ are positive semidefinite yielding that all their eigenvalues are nonnegative.

In section 8.1.1 in the appendix it is shown that the eigenvalues of a matrix of the form $A + b\mathbb{I}$ for $A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}$ are $\tilde{\lambda}_i := \lambda_i^A + b$, with λ_i^A, $i = 1, ..., n$ being the eigenvalues of the matrix A. This means that the following holds for all eigenvalues λ_i^k, $i = 1, ..., n$, of Q_k for all k:

$$\lambda_i^k + q \geq 0 \text{ and } q - \lambda_i^k \geq 0 \quad \Rightarrow \quad |\lambda_i^k| \leq q, \quad i = 1, ..., n, \quad \forall k.$$

The step size t_k is bounded below by t_{min} for all k. This yields for the spectral norm that

$$\|Q_k + \frac{1}{t_k}\mathbb{I}\|_2 = |\lambda_{max}^k + \frac{1}{t_k}| \leq |q + \frac{1}{t_{min}}| < \infty, \quad \forall k. \tag{6.9}$$

Here λ_{max}^k denotes the eigenvalue of Q_k that fulfills $\lambda_{max}^k = \arg\max_{i\in\{1,...,n\}} |\lambda_i^k + \frac{1}{t_k}|$.

The spectral norm is induced by the Euclidean norm thus it is compatible with it. Therefore with relation (6.9) it holds

$$\left\| Q_k + \frac{1}{t_k}\mathbb{I} \right\| \leq \left\| Q_k + \frac{1}{t_k}\mathbb{I} \right\|_2 \|x\| \leq \left| q + \frac{1}{t_{min}} \right| \|x\| < \infty, \quad \forall k.$$

□

Proposition 6.3 The norm $\|\cdot\|_{Q_k + \frac{1}{t_k}\mathbb{I}}$ induced by the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is well-defined for all k.

The proof to this proposition is found in section 8.1.2 of the appendix.

The first part of the convergence proof is now stated.

Theorem 6.4 (c.f. [13, Theorem 6, p. 14]) *Let algorithm 6.1 generate an infinite number of serious steps. Then $\delta_k \to 0$ as $K \ni k \to \infty$ for the sequence of serious steps*

$K \subset \{1, 2, ...\}$. Let the sequence $\{\eta_k\}$ be bounded. As $K \ni k \to \infty$ we have $C_k \to 0$. For every accumulation point $\bar{x}$ of $\{\hat{x}^k\}$ there exists a subsequence $\tilde{K}$ and a vector $\bar{S}$ such that $S^{\tilde{k}} \to \bar{S}$ and $S^{\tilde{k}} + \nu^{\tilde{k}} \to 0$ for $\tilde{K} \ni \tilde{k} \to \infty$. In particular if the cardinality of $\{j \in J^k \mid \alpha_j^k > 0\}$ is uniformly bounded in k then the conclusions of lemma 6.1 hold.

The proof is very similar to the one stated in [13] but minor changes have to be made due to the different formulation of the nominal decrease δ_k.

Proof: Let $K \subset \{1, 2, ...\}$ denote the subsequence of serious steps. (For the sake of readability the same index k that is used in the algorithm for all steps is used here only for the serious steps.) At each of those steps we have

$$\hat{f}_{k+1} \leq \hat{f}_k - m\delta_k, \quad k \in K \tag{6.10}$$

where $m,\ \delta_k > 0$. As $\hat{f}_k$ is only updated in serious steps it follows that the sequence $\{\hat{f}_k\}$ is nonincreasing. Since the sequence $\{\hat{x}^k\}$ lies in the compact set X and f is continuous the sequence $\{f(\hat{x}^k)\}$ is bounded. With $|\sigma_k| < \bar{\sigma}$ also the sequence $\{f(\hat{x}^k) + \sigma_k\} = \{\hat{f}_k\}$ is bounded. Considering also the fact that $\{\hat{f}_k\}$ is nonincreasing one can conclude that it converges.
From (6.10) follows that

$$0 \leq m \sum_{k=1}^{l} \delta_k \leq \sum_{k=1}^{l} \left(\hat{f}_k - \hat{f}_{k+1} \right),$$

so letting $l \to \infty$,

$$0 \leq m \sum_{k=1}^{\infty} \delta_k \leq \hat{f}_1 - \underbrace{\lim_{k \to \infty} \hat{f}_k}_{\neq \pm\infty}.$$

This yields

$$\sum_{k=1}^{\infty} \delta_k = \sum_{k=1}^{\infty} \left(C^k + \left\langle d^k, \left(Q_k + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle \right) < \infty.$$

Hence, $\delta_k \to 0$ as $k \to \infty$. All quantities above are nonnegative due to positive definiteness of $Q_k + \frac{1}{t_k}\mathbb{I}$ and $C_k \geq 0$ so it also holds that

$$C_k \to 0 \quad \text{and} \quad \left\langle d^k, \left(Q_k + \frac{1}{t_k} \mathbb{I} \right) d^k \right\rangle \to 0.$$

Finally we need to show that for any accumulation point $\bar{x}$ of the sequence $\{\hat{x}^{\tilde{k}}\}$ holds $S^{\tilde{k}} \to \bar{S}$ and $S^{\tilde{k}} + \nu^{\tilde{k}} \to 0$ for $\tilde{K} \ni \tilde{k} \to \infty$ and the suitable subsequence $\tilde{K} \subset K$. Let $K' \subset K$ denote the subset such that the sequence $\{\hat{x}^{k'}\}$ converges to its accumulation point $\bar{x}$ for $k' \in K'$. As we only consider the subsequence of serious steps it follows from $\{\hat{x}^{k'}\}_{k' \in K'} \to \bar{x}$ that $d^{k'} = \hat{x}^{k'+1} - \hat{x}^{k'} \to 0$ for $K' \ni k' \to \infty$. The step size t_k is bounded below by $t_{min} > 0$ and because the eigenvalues of Q_k are bounded the expression

$$S^{k'} + \nu^{k'} = \left(Q_{k'} + \frac{1}{t_{k'}}\mathbb{I}\right) d^{k'} \to 0 \quad \text{for} \quad K' \ni k' \to \infty$$

because

$$\|S^{k'} + \nu^{k'}\| = \left\|\left(Q_{k'} + \frac{1}{t_{k'}}\mathbb{I}\right) d^{k'}\right\| \leq \left|q + \frac{1}{t_{min}}\right| \underbrace{\|d^{k'}\|}_{\to 0}.$$

Here the last inequality follows from (6.8). The implication $S^{\tilde{k}} \to \bar{S}$ for $\tilde{k} \in \tilde{K}$ follows from local Lipschitz continuity of f. By Rademacher's theorem this property yields that on any open set Ω the function f is differentiable except on a set Ω_{nd} of zero Lebesgue measure. As f is also Lipschitz continuous on any closed set containing such an open set Ω, the gradient of f on Ω is bounded. Let $X \subset \Omega$. By theorem 2.5.1 in [3, p. 63] the subdifferential of f at any point $x^k \in \Omega$ is the convex hull of the limits of gradients ∇f of f on the set $\Omega \setminus \Omega_{\text{nd}}$

$$\partial f(x^k) = \text{conv}\{\lim \nabla f(y) \mid y \to x^k, y \notin \Omega_{\text{nd}}\}.$$

This means that also all subgradients on Ω are bounded. As the subgradient error is assumed to be bounded by $\bar{\theta}$ the set of approximate subgradients $\{g^j, j \in J^k\}$ contained in the bundle is bounded as well. From this follows that also the augmented subgradients $s_j^k = g^j + \eta_k(x^j - \hat{x}^k)$ are bounded because η^k is bounded by assumption and $x^j, \hat{x}^k \in X$. Defining $s := \max_{k,j} \|s_j^k\|$ this yields that

$$\|S^k\| = \left\| \sum_{j \in J^k} \alpha_j^k s_j^k \right\| \leq \sum_{j \in J^k} \|\alpha_j^k\| \underbrace{\|s_j^k\|}_{\leq s \in \mathbb{R}} \leq s \underbrace{\sum_{j \in J^k} \alpha_j^k}_{=1} < \infty \quad \forall k.$$

It follows that the sequence S^k is bounded. By the Bolzano-Weierstrass theorem [25, p. 51] every bounded sequence has a convergent subsequence. Let $\tilde{K} \subset \hat{K}'$ denote the index set of this converging subsequence and $\bar{S}$ the corresponding accumulation point. Then

finally $S^{\tilde{k}} \to \bar{S}$ for $\tilde{K} \ni \tilde{k} \to \infty$.

□

For the case that only finitely many serious steps are executed we need the following result:

Whenever x^{k+1} is as declared a null step, a simple calculation shows that the relation

$$-c_{k+1}^{k+1} + \left\langle s_{k+1}^{k+1}, x^{k+1} - \hat{x}^k \right\rangle = f_{k+1} - \hat{f}_k + \underbrace{\frac{\eta_{k+1}}{2}\|x^{k+1} - \hat{x}^k\|^2}_{\geq 0} > -m\delta_k \tag{6.11}$$

holds. The exact derivation of (6.11) is also given in [13, p. 16].

Another relation that is used a few times throughout the proof is the estimate

$$\left\langle \nu^k, d^k \right\rangle \geq 0. \tag{6.12}$$

It follows from the subgradient inequality for the convex function $\mathtt{i}_X$ at the point x^{k+1}. As $\nu^k \in \partial \mathtt{i}_X(x^{k+1})$ it holds $\mathtt{i}_X(y) - \mathtt{i}_X(x^{k+1}) \geq \left\langle \nu^k, y - x^{k+1} \right\rangle$ for all $y \in X$ and as $d^k = x^{k+1} - \hat{x}^k$ and $x^{k+1}, \hat{x}^k \in X$ it follows

$$0 = \underbrace{\mathtt{i}_X(\hat{x}^k)}_{=0} - \underbrace{\mathtt{i}_X(x^{k+1})}_{=0} \geq \left\langle \nu^k, \hat{x}^k - x^{k+1} \right\rangle = -\left\langle \nu^k, d^k \right\rangle$$

yielding inequality (6.12) above.

Theorem 6.5 (c.f. [13, Theorem 7, p. 16]) *Let a finite number of serious iterates be followed by infinite null steps. Let the sequence $\{\eta_k\}$ be bounded. Then $\{x^k\} \to \hat{x}$, $\delta_k \to 0$, $C_k \to 0$, $S^k + \nu^k \to 0$ and there exist $K \subset \{1, 2, ...\}$ and $\bar{S}$ such that $S^k \to \bar{S}$ as $K \ni k \to \infty$.*
In particular if the cardinality of $\{j \in J^k \mid \alpha_j^k > 0\}$ is uniformly bounded in k then the conclusions of lemma 6.1 hold for $\bar{x} = \hat{x}$.

Proof: Let k be large enough such that $k \geq \bar{k}$, where $\bar{k}$ is the iterate of the last serious step. Let $\hat{x} := \hat{x}^{\bar{k}}$ and $\hat{f} := \hat{f}_{\bar{k}}$ be fixed. The matrix Q_k is also fixed and denoted as $Q := Q_{\bar{k}}$. Define the optimal value of the k'th subproblem (6.3) for $k > \bar{k}$ by

$$\Psi_k := M_k(x^{k+1}) + \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle. \tag{6.13}$$

It is first shown that the sequence $\{\Psi_k\}$ is bounded above. From definition (5.6) and relation (5.10) follows

$$A_k(\hat{x}) = M_k(x^{k+1}) - \left\langle S^k, d^k \right\rangle \leq M_k(\hat{x}).$$

Using (6.4) for the third equality and (6.12) in the first inequality one obtains

$$\begin{aligned}
\Psi_k + \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle &= A_k(\hat{x}) + \left\langle S^k, d^k \right\rangle + \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \\
&= A_k(\hat{x}) + \left\langle S^k + \langle Q + \frac{1}{t_k}\mathbb{I}, d^k\rangle, d^k \right\rangle \\
&= A_k(\hat{x}) - \left\langle \nu^k, d^k \right\rangle \\
&\leq A_k(\hat{x}) \\
&\leq M_k(\hat{x}) \\
&= \hat{f}.
\end{aligned}$$

By boundedness of d^k and boundedness and positive definiteness $Q + \frac{1}{t_k}\mathbb{I}$ this yields that $\Psi_k \leq \Psi_k + \frac{1}{2}\|d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \leq \hat{f}$, so the sequence $\{\Psi_k\}$ is bounded above. In the next step it is shown that $\{\Psi_k\}$ is increasing. By noting that $x^{k+2} = \hat{x} + d^{k+1}$, as the proximal center does not change in the null step case, we obtain

$$
\begin{aligned}
\Psi_{k+1} &= M_{k+1}(x^{k+2}) + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_{k+1}})\mathbb{I}\right) d^{k+1}\right\rangle \\
&\geq A_k(\hat{x} + d^{k+1}) + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= M_k(x^{k+1}) + \left\langle S^k, d^{k+1} - d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= M_k(x^{k+1}) + \left\langle -\left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k - \nu^k, d^{k+1} - d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&= \Psi_k - \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&\quad - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right)\right\rangle - \left\langle \nu^k, d^{k+1} - d^k\right\rangle \\
&= \Psi_k + \frac{1}{2}\left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\rangle + \frac{1}{2}\left\langle d^{k+1}, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle \\
&\quad - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^{k+1}\right\rangle - \left\langle \nu^k, x^{k+2} - x^{k+1}\right\rangle \\
&\geq \Psi_k + \frac{1}{2}\left\langle \left(d^{k+1} - d^k\right), \left(Q + \frac{1}{t_k}\mathbb{I}\right)\left(d^{k+1} - d^k\right)\right\rangle \\
&= \Psi_k + \frac{1}{2}\underbrace{\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}}}_{\geq 0}.
\end{aligned}
\tag{6.14}
$$

Here the first inequality comes from (5.10) and the fact that $t_{k+1} \leq t_k$ for null steps. The second equality follows from (5.6), the fourth equality by (6.4) and (6.13) and the last inequality holds by the subgradient inequality for $\nu^k \in \mathtt{i}_X(x^{k+1})$ and the fact that $x^{k+1}, x^{k+2} \in X$.

Looking again at (6.14) and taking into account that $1/t_k \geq 1/t_{\bar{k}}$ in the null step case we have

$$
\begin{aligned}
\Psi_{k+1} - \Psi_k &\geq \frac{1}{2}\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \\
&\geq \frac{1}{2}\|d^{k+1} - d^k\|^2_{Q+\frac{1}{t_{\bar{k}}}\mathbb{I}} \geq 0.
\end{aligned}
$$

This means that the sequence $\{\Psi_k\}$ is increasing and bounded from above. Thus the sequence is convergent.

This yields

$$|\Psi_{k+1} - \Psi_k| \to 0 \quad \Rightarrow \quad \|d^{k+1} - d^k\| \to 0 \text{ for } k \to \infty \tag{6.15}$$

due to the equivalence of norms.

By the last line in (6.5) and the fact that $\hat{f} = M_k(\hat{x})$ for all $k > \bar{k}$ we have

$$\begin{aligned}
\hat{f} &= M_k(\hat{x}) + \delta_k - C_k - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \\
&= M_k(\hat{x}) - \hat{f} + M_k(x^{k+1}) + \delta_k - \left\langle S^k, d^k \right\rangle - \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle \\
&= \delta_k + M_k(\hat{x} + d^k) + \left\langle \nu^k, d^k \right\rangle \\
&\geq \delta_k + M_k(\hat{x} + d^k),
\end{aligned}$$

where the second equality is by (5.9), the third holds because of relation (6.4) and the last inequality is given by (6.12). Therefore

$$\delta^{k+1} \leq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}). \tag{6.16}$$

By assumption (5.11) on the model, written for $d = d^{k+1}$,

$$-\hat{f}_{k+1} + c_{k+1}^{k+1} - \left\langle s_{k+1}^{k+1}, d^{k+1} \right\rangle \geq -M_{k+1}(\hat{x} + d^{k+1}).$$

In the null step case it holds $\hat{f}_{k+1} = \hat{f}$ so combining condition (6.11) and the inequality above, one obtains that

$$m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \geq \hat{f} - M_{k+1}(\hat{x} + d^{k+1}).$$

In combination with (6.16) this yields

$$0 \leq \delta_{k+1} \leq m\delta_k + \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \leq m\delta_k + \left| \left\langle s_{k+1}^{k+1}, d^k - d^{k+1} \right\rangle \right|. \tag{6.17}$$

For the next step lemma 3 and the corollary below it from [44, p. 45] are used. They state that for

$$u_{k+1} \leq q u_k + a_k, \quad q < 1, \quad a_k \geq 0, \quad a_k \to 0 \text{ and } u_k \geq 0$$

it holds $u_k \to 0$.

Taking the first and the last part of inequality (6.17) we can identify $u_k = \delta_k \geq 0$, $q = m \in (0,1)$ and $a_k = \left|\left\langle s_{k+1}^{k+1}, d^k - d^{k+1}\right\rangle\right| \geq 0$. To show that $a_k \to 0$ recall (6.15) and that the augmented subgradient s_{k+1}^{k+1} is bounded due to local Lipschitz continuity of f and boundedness of $\{\eta_k\}$ by the same argumentation as in the case of infinitely many serious steps.

The lemma then gives that

$$\lim_{k\to\infty} \delta_k = \lim_{k\to\infty} C_k + \left\langle d^k, \left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k \right\rangle = 0.$$

By the fact that $C_k \geq 0$ for all k and positive definiteness of $Q + \frac{1}{t_k}\mathbb{I}$ it follows that all summands above are nonnegative and hence $C_k \to 0$ as $k \to \infty$. As the matrix $Q + \frac{1}{t_k}\mathbb{I}$ is bounded due to $t_k > t_{min} > 0$ and the bounded eigenvalues of Q and because in null steps $t_k \leq t_{\bar{k}}$,

we have

$$\|d^k\|^2_{Q+\frac{1}{t_k}\mathbb{I}} \geq \|d^k\|^2_{Q+\frac{1}{t_{\bar{k}}}\mathbb{I}} \geq c\|d^k\|^2 \to 0.$$

for a constant $c \in \mathbb{R}$ by the equivalence of norms.

This means that $d^k \to 0$ for $k \to \infty$ and therefore $\lim_{k\to\infty} x^k = \hat{x}$. It also follows that $\|S^k + \nu^k\| \to 0$ as $k \to \infty$ because of

$$\|S^k + \nu^k\| = \left\|\left(Q + \frac{1}{t_k}\mathbb{I}\right) d^k\right\| \leq \left|q + \frac{1}{t_{min}}\right| \underbrace{\|d^k\|}_{\to 0} \to 0.$$

By the same arguments as in the proof of theorem 6.4 the local Lipschitz property of the objective function f and boundedness of the subgradient errors θ_k yield boundedness of the sequence S^k. Passing to some subsequence $K \subset \{1, 2, ...\}$ if necessary we can therefore conclude that the sequence $\{S^k\}_{k\in K}$ converges to some $\bar{S}$ and as $\hat{x}^k = \bar{x}$ for all $k \geq \bar{k}$ all requirements of lemma 6.1 are fulfilled.

□

Remark: In case the matrix Q_k is also updated in null steps the proof still holds as long

as the assumptions on boundedness of Q_k and especially positive definiteness of $Q_k + \frac{1}{t_k}\mathbb{I}$ even in the limit $k = \infty$ are still valid.

Remark: All results deduced in section 5.2 are still valid for this algorithm as they do not depend on the kind of stabilization used.

6.4 Updating the Metric

In [42] and [40] it is not specified how the matrices Q_k are chosen. For convergence it is necessary that Q_k is symmetric and its eigenvalues are bounded. Here we present some possibilities to update the metric matrix Q_k such that it fulfills both conditions.

Most of the presented updates are based on the BFGS-update formula (named after Broyden, Goldfarb, Fletcher and Shanno)

$$\tilde{Q}_{k+1} = Q_k + \frac{y^k {y^k}^\top}{\langle y^k, d^k \rangle} - \frac{Q_k d^k (Q_k d^k)^\top}{\langle d^k, Q_k d^k \rangle}. \tag{6.18}$$

Usually y^k is defined as the difference of the last two gradients of f. To adapt the formula to the nondifferentiable case the difference $y^k := g^{k+1} - g^k$ of two (approximate) subgradients of f is taken instead as proposed in [11]. The starting matrix is $Q_1 = \mathbb{I}$.

By definition the BFGS update is symmetric. To assure boundedness of the matrix Q_{k+1} the updates can be manipulated in the following ways:

6.4.1 Scaling of the Whole Matrix

A simple way to keep the absolute value all of eigenvalues of the constructed matrix Q_k below some threshold $0 < q < \infty$ is to scale the whole matrix down as soon as the absolute value of one eigenvalue is larger than this number. To do this define $\lambda_{max} := \max\{|\lambda_i^k| \mid \lambda_i^k \text{ is eigenvalue of } \tilde{Q}_k\}$. If $\lambda_{max}^k > q$, set $Q_k = \frac{q}{\lambda_{max}^k}\tilde{Q}_k$, where $\tilde{Q}_k$ is the matrix coming from the BFGS update. This way the absolute value of all eigenvalues is always smaller or equal to q. An advantage of this method is besides its simplicity that by scaling the whole matrix the ratio of the eigenvalues of Q_k is preserved. Scaling of Q_k corresponds to shrinking the whole quadratic function and in this way also the 'ratio of curvature' at different points of the graph stays the same.

6.4.2 Adaptive Scaling of Single Eigenvalues

The second method is motivated by some properties of lower-$\mathcal{C}^2$ functions. This function class is very suitable to be used with the presented bundle algorithm and there exist some practical applications that entail such functions (c.f. [12] and [13]).

Lower-$\mathcal{C}^2$ functions can locally be written as the maximum over finitely many $\mathcal{C}^2$ functions.

For the following motivation let us consider a lower-$\mathcal{C}^2$ function $f : \mathbb{R} \to \mathbb{R}$ on an open set $\Omega \subset \mathbb{R}$ that can be written as

$$f(x) = \max_{t \in T} f_t(x) \tag{6.19}$$

for a finite index set T fulfilling the conditions of Definition 2.5.

Consider from now on the function f on the set Ω. At points where the maximum in (6.19) is only attained by a single function f_t the function f is twice continuously differentiable and hence provides curvature information. At points where the are more than one function attaining the same value this does not have to be the case. At those points f can be nondifferentiable.

Consider a nondifferentiable point and denote it as $x_{\text{kink}} \in \Omega$. As f is locally Lipschitz continuous Rademacher's theorem yields that the points in Ω where the function f is nondifferentiable are a set of zero Lebesgue measure. This means that there exists an open interval with length $2r, r > 0$, around the point x_{kink} such that $(x_{\text{kink}} \pm r) \in \Omega$ and the function f is differentiable on the two open intervals $(x_{\text{kink}} - r, x_{\text{kink}})$ and $(x_{\text{kink}}, x_{\text{kink}} + r)$. Assume, that f is is *directionally differentiable* in x_{kink} and that all the directional derivatives in that point are finite. The directional derivative with respect to the direction $d \in \mathbb{R}^n$ is defined as [46, p. 213]

$$f'(x, d) = \lim_{h \searrow 0} \frac{f(x + hd) - f(x)}{h}.$$

The function f has for example finite directional derivatives in x_{kink}, if it is convex on Ω [53, p. 144].

In the one dimensional case this means that we can denote the *right derivative* $f(x_{\text{kink}}, 1) := a$ and the *left derivative* $-f(x_{\text{kink}}, -1) := b$. As f was assumed to be nondifferentiable in x_{kink} it holds that $a \neq b$ [46, p. 213].

To get a hint on what we can expect from the 'curvature' at that kink the following

quotient is examined:

$$\lim_{h \searrow 0} \frac{\overbrace{f'(x_{\text{kink}} - h, 1)}^{\to a} + \overbrace{f'(x_{\text{kink}} + h, -1)}^{\to b}}{h} = \pm\infty.$$

This holds by continuity of the of the derivatives on both sides of the kink and the directional derivatives being the respective limits at the kink. The sign depends on the signs of a and b.

In more dimensions this is the same for the components of the metric matrix Q_k corresponding to the direction where the kink occurs. This supports also to the intuitive thought that at a kink the slope changes 'infinitely fast'. Numerically the BFGS-update (6.18) can result in very large values for the entries of Q_k corresponding to points near the kink.

On the other hand due to the local Lipschitz property the slope of the objective function is always finite on closed sets. This means that there exists an interval $(x_{\text{kink}} - r', x_{\text{kink}} + r')$, $r' \in \mathbb{R}$, where the function f behaves similar to the scaled modulus $a|\cdot|$, $a \in \mathbb{R}$, in the direction perpendicular to the kink. Therefore in this neighborhood almost no curvature is present.

Summarized this means that on the one hand, the matrix Q_k should be close to zero in the components representing the directions perpendicular to the kink as soon as the iterates approach x_{kink}. But contrary to that the method that constructs Q_k can give very high values for those components.

Remark: The above considerations are only a motivation for the following practical matrix update. A more rigorous theoretical background for the update is still open.

Let again $\tilde{Q}_k$ denote the matrix coming directly form the BFGS update. The idea is now to scale only those eigenvalues of $\tilde{Q}_k$ that are especially large. To do this calculate all eigenvalues λ_i^k of $\tilde{Q}_k$. As the metric matrix $\tilde{Q}_k$ is a symmetric real matrix it is always orthogonally diagonalizable [19, Corollary 18.18 p. 282]. An eigenvalue decomposition $\tilde{Q}_k = U \cdot \tilde{D} \cdot U^\top$ is available. The diagonal matrix $\tilde{D}$ has the eigenvalues of $\tilde{Q}_k$ on its diagonal and the transformation matrix U contains the corresponding eigenvectors.

Then all eigenvalues of $\tilde{Q}_k$ that are larger than q are scaled and replaced in the matrix $\tilde{D}$. The bounded version of $\tilde{Q}_k$ is obtained by transforming the bounded matrix D back into the full matrix with help of the transformation matrix U. Let D denote the matrix with the scaled eigenvalues on the diagonal. The matrix U is not changed. This means

that $Q_k := A \cdot D \cdot A^\top$ has the same eigenvectors as the original update $\tilde{Q}_k$ but bounded eigenvalues.

The above two methods were tested in practice and the results of the algorithm are shown in section 6.5. We also compare them to a hybrid method where the first approach is used for the updates and the matrix is additionally scaled by the stepsize such that the final metric matrix is $Q_k = \frac{1}{k}\bar{Q}_k$ with $\bar{Q}_k$ being the scaled BFGS update suggested in section 6.4.1. This way the method starts out as the variable metric method but becomes more equal to the proximal bundle method 5.1 as the algorithm continues.

Remark: There appear many parameters to control the scaling of the metric update. Although these parameters were not especially tuned in this thesis it was observed that they have a considerable impact on the convergence speed of the method also depending on the objective function used. This has to be kept in mind when implementing the method in practice.

6.4.3 Other Updating Possibilities

There are certainly many other possibilities to update the metric Q_k. A third variation based on BFGS-updates is the limited memory update suggested in [38]. If the update is skipped whenever $\frac{\|d^k\|}{\|y^k\|} > \tilde{q}, \tilde{q} > 0$, the matrix Q_k stays bounded. (Remark that in general $\tilde{q} \neq q$, so it is not directly the lower bound for the step size t_{min}.) This strategy is also supported by the fact that if $\frac{\|d^k\|}{\|y^k\|} > \tilde{q}$ the change in the subgradient relative to the step size is rather small indicating that the current iterate lies within a region with only small changes in curvature. In such regions the update can be skipped. It is also possible to alter the updates presented above by a special choice of the subgradients. For example trying to compute the directional derivative or using more information by considering more subgradients of the bundle.

Another updating method is using the symmetric-rank-1 (SR1) update

$$\tilde{Q}_k = Q_{k-1} + \frac{(y^k - Q_k d^k)(y^k - Q_k d^k)^\top}{\langle y^k - Q_k d^k, d^k \rangle}.$$

Boundedness can be assured in the same way as for the BFGS update.

Finally the strategies to measure the need of scaling of the matrices to ensure boundedness are diverse. Here it could be interesting to consider the change in the matrix $\tilde{Q}_k$ relative to Q_{k-1} instead of using the absolute values of the eigenvalues as a threshold. This is however hardly possible if the eigenvalues themselves are taken as a measure as they lack

an intrinsic order. This makes it hard to find the corresponding eigenvalues λ_i^{k-1} and λ_i^k in consecutive updates in order to compute the change between them.

In higher dimensions updating the matrix Q_k can be costly. This is one of the reasons why it is not updated in null steps. Also in null steps the proximal center stays the same, so it can be assumed that not much curvature information can be gained when updating during such steps. Still updating in null steps has the advantage of making use of the additional subgradient information provided in those steps. In [11], where the metric matrix is updated also in null steps, a BFGS update is used in serious steps and the less costly SR1 update in null steps.

Remark: Bounding the eigenvalues of Q_k by $q \in \mathbb{R}$ also assures that t_k can be bounded from below without impairing positive definiteness of the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$. This can be done by setting $t_{min} = \frac{1}{q} - \varepsilon$ for a small positive constant ε as it is done in algorithm 6.

As a last remark on this topic we want to say that although in this thesis the adaption of update strategies originally developed to be used with gradients seems to work in the presented setting also with subgradients this does not always have to be the case. Although it is argued for example in [30] that locally Lipschitz functions are differentiable almost everywhere and with an adequate linesearch it is improbable to arrive at an iterate that is a nondifferentiable point of the objective function this can still happen. Especially if such a linesearch is not used like in the algorithm presented here. So despite the promising practical behavior this area is still open to research.

6.5 Numerical Tests

To compare the proximal bundle algorithm 5.1 with its variable metric variant algorithm 6.1 both are tested on some academic test functions and on a set of lower-$\mathcal{C}^2$ functions in different dimensions. The tests are done with the following parameters given in [13]: $m = 0.05$, $\gamma = 2$ and $t_0 = 0.1$. The chosen stopping tolerance is $\texttt{tol} = 10^{-6}$. If the algorithms do not meet the stopping condition after $250n$ steps for $x \in \mathbb{R}^n$, they are terminated. Contrary to [13] the stopping test is taken as given in the algorithm and the tolerance not multiplied by $1 + \hat{f}_k$. The proximity control parameters κ_- and κ_+ from (3.22) and (3.21) respectively are chosen as $\kappa_- = 0.8$ and $\kappa_+ \in \{1.2, 2\}$. When the bundle is updated at the end of each iteration additionally to the newly computed iterate the current prox-center and all elements that have corresponding Lagrange multipliers $\alpha_j^k > 10^{-15}$ are kept in the bundle.

In the metric matrix updates the threshold q is chosen 10^8. For the adaptive variant of the update eigenvalues that are larger than q are set to $q/10$.

The algorithms are abbreviated in the legends of the plots as 'Bundle Nonconv Inex' for the proximal bundle algorithm 5.1 and 'Variable Metric BFGS' and 'Variable Metric BFGS Adaptive' for the variable metric bundle algorithm 6.1 using the scaled update and the adaptive eigenvalue scaling respectively.

To test the performance for inexact function and subgradient values different types of noise are introduced. This is done by adding randomly generated elements with norm less or equal to σ_k and θ_k to the exact values $f(x^{k+1})$ and $g(x^{k+1})$ respectively.

Five different forms of noise are tested:

- N_0: No noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta_k = 0$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{f,g}}$: Constant noise, $\bar{\sigma} = \sigma_k = 0.01$ and $\bar{\theta} = \theta_k = 0.01$ for all k,
- $\mathrm{N}_\mathrm{v}^{\mathrm{f,g}}$: Vanishing noise, $\bar{\sigma} = 0.01, \sigma_k = \min\{0.01, \|x^k\|/100\}$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k,
- $\mathrm{N}_\mathrm{c}^{\mathrm{g}}$: Constant subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = \theta_k = 0.01$ for all k and
- $\mathrm{N}_\mathrm{v}^{\mathrm{g}}$: Vanishing subgradient noise, $\bar{\sigma} = \sigma_k = 0$ and $\bar{\theta} = 0.01, \theta_k = \min\{0.01, \|x^k\|/100\}$ for all k.

The exact case is used for comparison. The constant noise forms represent cases where the inexactness is outside of the optimizer's control. The vanishing noise forms represent cases where the noise can be controlled but the mechanism is considered expensive, so it is only used when approaching the minimum. The two forms of subgradient noise represent the case where the subgradient is approximated numerically.

To compare the performance of the different methods the accuracy is measured by

$$\text{accuracy} = |\log_{10}(\hat{f}_{\bar{k}})|.$$

Here $\hat{f}_{\bar{k}}$ is the current $\hat{f}_k$ when the algorithm stops.

6.5.1 Academic Test Examples

For the comparison in this section the proximal bundle method and the variable metric method with the two BFGS update rules for Q_k presented in section 6.4 are used.

To explore the benefit of the matrix Q_k the algorithms 5.1 and 6.1 are tested on a smooth

and a nonsmooth version of a badly conditioned parabola. The smooth test function is

$$p(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \langle x, Ax \rangle ,$$

where the matrix is chosen as $A = \begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix}$. The condition number of this matrix is $\kappa_A = \frac{\lambda_{max}}{\lambda_{min}} = 50$, where $\lambda_{min}, \lambda_{max}$ are the smallest and largest eigenvalue of A respectively. From smooth optimization it is known that gradient descent methods have a rather poor convergence rate for such badly conditioned matrices (c.f. chapter 7.4 in [56]). Figure 1 shows the sequences of serious iterates resulting from the two algorithms on the contour lines of the parabola. On the left the complete sequence is depicted. The plot on the right shows a detail of the left figure near the minimum of the objective. As the descent direction taken in algorithm 5.1 is an aggregate subgradient and second order information is only provided by the stabilization term $\frac{1}{t_k}\|d\|^2$ we can see a zig-zagging behavior of the sequence for the parabola in figure 1. Contrary to that the sequence of serous iterates provided by algorithm 6.1 can take advantage of the second order information provided by Q_k. It approaches the minimum almost in a straight line. The difference in behavior of the two algorithms is especially visible on the detail plot of figure 1 that shows the situation near the minimum: The proximal bundle algorithm needs a lot of steps circling around the minimum whereas the variable metric algorithm approaches the minimum directly. The resulting advantage of this behavior is the smaller number of steps needed by the variable metric algorithm.

Figure 1: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of parabola p. The right image is a detail of the plot on the left.*
Step size parameter: $\kappa_+ = 2$ *for both algorithms.*

The second test function is a nonsmooth version of the above parabola. The function is given by

$$p_n(x) : \mathbb{R}^2 \to \mathbb{R}, \quad x \mapsto \frac{1}{2} \langle x, Ax \rangle + \frac{1}{2}|x_1| + 25|x_2|.$$

Due to the kink along the x_1-axis the curvature information supplied by Q_k is less reliable than for the smooth parabola. Figure 2 shows the sequences constructed by the two algorithms. Still the sequence provided by the variable metric algorithm does less zig-zagging than the one coming from the proximal bundle algorithm. It is interesting to note, that the sequence provided by the proximal bundle algorithm is the same for both

functions. This is not the case for the sequence generated by the metric bundle algorithm because the second order information of the two objective functions is different.

Figure 2: *Sequences of serious steps constructed by the proximal bundle algorithm and the variable metric algorithm respectively on the level lines of the nonsmooth quadratic function p_n. The right image is a detail of the plot on the left.*
Step size parameter: $\kappa_+ = 2$ for both algorithms.

The bar plots in figures 3 and 4 compare the accuracy of the solution and the number of steps that is needed by the different algorithms for the various noise forms. Here the nonconvex proximal bundle algorithm is compared to both variants of the variable metric method.

To address the random nature of the noise the tests are performed 20 times and the results averaged. The number of steps is rounded to end up with integers.

Figure 3: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the parabola p under different form of noise.*
Right: Comparison of the number of steps for the three algorithms.
Step size parameters: $\kappa_+ = 1.2$ for the proximal bundle method and $\kappa_+ = 2$ for the variable metric algorithm.

In the smooth case one can see that the accuracy of the two algorithms is comparable. In the case where no noise is present and in the last case, which is the case with the least noise, the variable metric algorithm solves more accurately but for the proximal bundle algorithm the actually computed optimal value is still above the chosen tolerance of 10^{-6}. In the cases of the more involved noise the accuracy is less.

A significant difference can be seen between the needed number of steps of the different algorithms. Here the variable metric versions of the bundle method can take advantage of the curvature information and the fact that for the smooth parabola the BFGS update approximates the Hessian matrix very well. The difference in steps between the two update variants of the variable metric algorithm is neglectable and could also be present due to the random noise. This is what we expect as the scaling mechanism should not be invoked in the smooth case.

In the nonsmooth case (shown in figure 4) the accuracy of all algorithms is very similar. The difference in the number of steps is now very small as well. Only in the case of constant noise the proximal bundle algorithm performs rather badly. Here the number

Figure 4: *Left: Accuracy of the solution computed by the different versions of the variable metric bundle algorithm compared to the proximal bundle algorithm for the nonsmooth quadratic function p_n under different form of noise.*
Right: Comparison of the number of steps for the three algorithms. The left bar for the number of steps in the case of constant noise is cropped.
Step size parameters: $\kappa_+ = 1.2$ for the proximal bundle method and $\kappa_+ = 2$ for the variable metric algorithm.

of steps is extremely large (over 250) in order to gain the same accuracy as the other algorithms. The difference between the two update versions of the variable metric algorithm is still very small. It seems that the different scaling strategies have only a minor influence on the algorithm for this kind of objective function. Other tests showed that for example the choice of the step size updating parameters κ_+, κ_- have a lot more influence on the algorithm than the tested updating strategies. This can be seen in figures 9 and 10 and figures 19 to 26 in the appendix.

6.5.2 Test Examples in Higher Dimensions

For the second test, which involves testing the performance of the different algorithms in different dimensions, the Ferrier polynomials are chosen as objective functions. These nonsmooth and nonconvex functions have already been used in [12] and [13]. The polynomials are constructed in the following way:

For $i = 1, ..., n$ we define

$$h_i : \mathbb{R}^n \to \mathbb{R}, \quad h(x) = (ix_i^2 - 2x_i) + \sum_{j=1}^{n} x_j.$$

These functions are used to define

$$\begin{aligned} f_1(x) &:= \sum_{i=1}^{n} |h_i(x)|, \\ f_2(x) &:= \sum_{i=1}^{n} (h_i(x))^2, \\ f_3(x) &:= \max_{i \in \{1,...,n\}} |h_i(x)|, \end{aligned}$$

$$f_4(x) := \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}\|x\|^2 \text{ and}$$
$$f_5(x) := \sum_{i=1}^{n} |h_i(x)| + \frac{1}{2}\|x\|.$$

The graphs of the Ferrier polynomials for $x \in \mathbb{R}^2$ are shown in figure 5.

Figure 5: *Graphs of the testfunctions f_1 to f_5 for $x \in \mathbb{R}^2$*

Ferrier polynomials are nonconvex, nonsmooth (except for f_2) and lower-$\mathcal{C}^2$. They all have 0 as a global minimizer [13, p. 23]. The compact constraint set is $X = \{x \in \mathbb{R}^n | |x_i| \leq 10, i = 1, ..., n\}$.

The five test functions f_1 to f_5 are optimized for the dimensions $n = \{2, 3, ..., 15\} \cup \{20, 25, 30, 40, 50\}$. The starting value for each test problem is $x^1 = [1, \frac{1}{4}, \frac{1}{9}, ..., \frac{1}{n^2}]^\top$.

For the tests the step size of all algorithms is updated with $\kappa_+ = 1.2$, which provided better results for these specific test functions.

The accuracy measures absolute logarithmic distance to the global minimum. In case that the algorithm finds a local minimum, which is possible for nonconvex objective functions, this lowers the accuracy.

Figure 6: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of no noise.*

In the figures 6 to 8 and 12 to 18 in the appendix the achieved accuracy and the needed number of steps are shown for the proximal bundle method and two versions of the variable metric method.

Figure 6 shows the situation if no noise is present and can be seen as a benchmark for the other noise forms. It is clearly visible that the desired accuracy of 10^{-6} is not always achieved by the different algorithms. A reason for this is that the objective functions have several local minima where the algorithms can get stuck. It seems that this happens more seldom to the proximal bundle algorithm than the variable metric method.

For the Ferrier polynomials the proximal bundle algorithm needs significantly less steps than the variable metric algorithm. It can also be observed that the cases where the variable metric algorithm is stuck in a local minimum, the number of steps needed rises significantly. This is not the case for the proximal bundle algorithm.

The performance of the two variants of the two versions of the variable metric method are similar. It seems however as if the adaptive version performed slightly better in terms of the number of steps used.

Figure 7: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant noise*

In the case of constant noise the variable bundle methods perform better than the proximal version. They are more stable in the achieved accuracy and need considerably less steps than the other method.

Figure 8: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing noise*

For the other noise forms the three algorithms perform similar in terms of the accuracy but the variable bundle methods need consistently more steps. The only exception from this is case of vanishing noise. Here the proximal bundle method needs extremely many more steps than the variable metric bundle method to optimize function f_3. This shows that the performance of the different algorithms depend on both the form of noise and the specific objective function.

The plots for the higher dimensional data $x \in \mathbb{R}^n$ for $n = \{20, 25, 30, 40, 50\}$ are included in the appendix (figures 14 to 18). Also in higher dimensions the algorithms achieve a similar accuracy. The number of steps needed for convergence in generally higher, but still the proximal bundle method needs less steps in most situations. In the cases where there is noise on the function value, the algorithms almost always stop because the maximum number of steps is reached. The only exception is the smooth function f_2. Here the variable metric methods perform a lot better than the proximal bundle algorithm in terms of the number of steps, because the curvature information is more reliable.

Finally the influence of the step size updating parameter κ_+ is shown and the performance of the hybrid method. This last method, denoted by 'Variable Metric BFGS, k-scaled' in the figures, uses the scaled BFGS update for the metric matrix Q_k and then scales this matrix again by the step size. This means the final matrix is $Q_k = \frac{1}{k}\tilde{Q}_k$ if $\tilde{Q}_k$ denotes the matrix after the scaled BFGS update, lowering the influence of the metric matrix in each serious step. In this way the method starts out as the variable metric method and then behaves more and more like the proximal bundle method.

This shows that the additional curvature information can make a considerable difference

in the convergence speed, especially for matrices Q_k that model the behavior of the objective function at the kinks correctly. It is therefore an interesting but still open question if such matrix updates can be found.

The algorithms used for the comparison of the different κ_+ are endowed with the scaled BFGS update from section 6.4.1. The parameters $\kappa_+ = 1.2$ and $\kappa_+ = 2$ are compared. Here only the exact case and the case of constant noise for the lower dimensions are depicted in figure 9 and 10. The situation for the other noise forms is shown in figures 19 to 26 in the appendix.

Figure 9: *Influence of the step size updating parameter $\kappa_+ = 1.2$ and $\kappa_+ = 2$ and performance of the hybrid method in the exact case. The reached accuracy is depicted on the left, the needed number of steps on the right.*

Figure 10: *Influence of the step size updating parameter $\kappa_+ = 1.2$ and $\kappa_+ = 2$ and performance of the hybrid method for constant noise. The reached accuracy is depicted on the left, the needed number of steps on the right.*

One can see that contrary to the academic test example, where the choice $\kappa_+ = 2$ gives better results for the number of steps, here the parameter $\kappa_+ = 1.2$ performs better. In the case of constant noise the numbers of steps are similar. As expected the accuracy of the two methods is very similar, because only one parameter is changed. The number of steps shows however that parameter tuning can be useful. Here it is important to keep in mind that the optimal parameter depends on the objective function and the noise form.

The performance of the hybrid method is, as can be expected, similar to the proximal bundle method. As the scaling is quite strong the influence of the metric matrix decreases rather quickly. This means that in cases where the proximal bundle method performed better, the same is true for the hybrid method. The increase in the number of steps is only minor. Likewise in the case of constant noise (figure 10) for example, where the proximal bundle method needed a lot of steps, these steps are also needed by the hybrid method. Two advantages of the hybrid method still come into play for this noise form: The significant decrease in the number of steps starts only in higher dimensions, where the total number of steps grows larger. Here a 'slower' scaling could yield even better numbers. The other advantage is the more stable accuracy. This holds true for all dimensions.

7 Application to Model Selection for Primal SVM

Skalarprodukt anpassen, Vektoren nicht fett oder neue definition, notation, $\lambda \in \Lambda$ einfugen

7.1 Introduction

In this part of the thesis the nonconvex inexact bundle algorithm number in thesis is applied to the problem of model selection for *support vector machines* (SVMs) solving classification tasks. It relies on a bilevel formulation proposed by Kunapuli in [26] and Moore et al. in [36].

A natural application for the inexact bundle algorithm is an optimization problem where the objective function value and subgradient can only be computed by numerical approximation. This is for example the case in bilevel optimization.

A general bilevel program can be formulated as in [26, p. 20]

$$\begin{aligned} \min_{x \in X, y \in \mathbb{R}^k} \quad & F(x,y) && \text{upper level} \\ \text{s.t.} \quad & G(x,y) \leq 0 \\ & y \in \left\{ \begin{array}{ll} \arg\max\limits_{y \in Y} & f(x,y) \\ \text{s.t.} & g(x,y) \leq 0 \end{array} \right\}. && \text{lower level} \end{aligned} \tag{7.1}$$

The two objecective functions F and f map from $\mathbb{R}^n \times \mathbb{R}^k$ into $\mathbb{R}$ and the constraint functions G and g map from $\mathbb{R}^n \times \mathbb{R}^k$ into $\mathbb{R}^L$ and $\mathbb{R}^l$ respectively.

The problem consists of an *upper* or *outer level* which is the overall function to be optimized. Contrary to usual constrained optimization problems which are constrained by explicitly given equalities and inequalities a bilevel program is additionally constrained by a second optimization problem, the *lower* or *inner level* problem.

Solving bilevel problems can be divided roughly in two classes: implicit and explicit solution methods. In the explicit methods the lower level problem is usually rewritten by its KKT conditions, these are then added as constraints to the upper level problem. With this solution method the upper and lower level are solved simultaneously. For the setting of model selection for support vector machines as it is used here, this method is described in detail in [26].

The second approach is the implicit one. Here the lower level problem is solved directly in every iteration of the outer optimization algorithm and the solution is plugged into the upper level objective.

Obviously if the inner level problem is solved numerically, the solution cannot be exact. Additionally the *solution map* $S(x) = \{y \in \mathbb{R}^k \mid y$, that solves the lower level problem, is can be nondifferentiable [43] and since elements of the solution map are plugged into the outer level objective function in the implicit approach, the outer level function then becomes nonsmooth itself.This is why the inexact bundle algorithm seems a natural choice to tackle these bilevel problems.

Moore et al. use the implicit approach in [36] for support vector regression. However they use a gradient decent method which is not guaranteed to stop at an optimal solution. In [35] he also suggests the nonconvex exact bundle algorithm of Fuduli et al. [8] for solving the bilevel regression problem. This allows for nonsmooth inner problems and can theoretically solve some of the issues of the gradient descent method. It ignores however, that the objective function values can only be calculated approximately. A fact which is not addressed in Fuduli's algorithm.

7.2 Notation

special notation only in this chapter, x, y now different. Optimization variable now λ, C. with index -> data, without variables from problem. Will be clear from context.

do this due to the standard notation in the field of SVM

7.3 Introduction to Support Vector Machines

Support vector machines are linear learning machines that were developed in the 1990's by Vapnik and co-workers. Soon they could outperform several other programs in this area [5] and the subsequent interest in SVMs lead to a very versatile application of these machines [26].

The case that is considered here is binary support vector classification using supervised learning. For a throughout introduction to this subject see also [5]. Here a summary of the most important expressions and results is given.

In classification data from a possibly high dimensional vector space $\tilde{X} \subset \mathbb{R}^n$, the *feature* or *input space* is divided into two classes. These lie in the *output domain* $\tilde{Y} = \{-1, 1\}$.

Elements from the feature space will mostly be called *data points* here. They get *labels* from the feature space. Labeled data points are called *examples*. The functional relation between the features and the class of an example is given by the usually unknown *response* or *target function* $f(x)$. Supervised learning is a kind of machine learning task where the machine is given examples of input data with associated labels, the so called *training data* (X, Y). Mathematically this can be modeled by assuming that the examples are drawn identically and independently distributed (iid) from the fixed joint distribution $P(x, y)$. This usually unknown distribution states the probability that a data point x has the label y [58, p. 988]. The overall goal is then to optimize the generalization ability, meaning the ability to predict the output for unseen data correctly [5, chapter 1.2].

7.3.1 Risk minimization

The concept of SVM's was originally inspired by the statistical learning theory developed by Vapnik. A detailed examination of the subject is given in [57]. In [59] the subject is approached from a more explaining point of view.

The idea of *risk minimization* is to find from a fixed set or class of functions the one that is the best approximation to the response function. This is done by minimizing a loss function that compares the given labels of the examples to the response of the learning machine.

As the response function is not known only the expected value of the loss can be calculated. It is given by the *risk functional*

$$R(\lambda) = \int \mathcal{L}(y, f_\lambda(x)) \mathrm{d}P(x, y). \tag{7.2}$$

Here $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R}$ is the loss function, $f_\lambda : \mathbb{R}^n \cap \mathcal{F} \to \mathbb{R},\ \lambda \in \Lambda$ the approximate response function found by the learning machine and $P(x, y)$ the joint distribution the training data is drawn from. The goal is now to find a function $f_{\bar{\lambda}}(x)$ in the chosen function space $\mathcal{F}$ that minimizes this risk functional [58, 989].

As the only given information is provided by the training set inductive principles are used to work with the *empirical risk*, rather than with the risk functional. The empirical risk only depends on the finite training set and is given by

$$R_{\mathrm{emp}}(\lambda) = \frac{1}{l} \sum_{i=1}^{l} \mathcal{L}(y_i, f_\lambda(x^i)), \tag{7.3}$$

where l is the number of data points. The law of large numbers ensures that the empirical risk converges to the risk functional as the number of data points grows to infinity. This however does not guarantee that the function $f_{\lambda,\text{emp}}$ that minimizes the empirical risk also converges towards the function $f_{\bar{\lambda}}$ that minimizes the risk functional. The theory of consistency provides necessary and sufficient conditions that solve this issue [58, p. 989].

Vapnik therefore introduced the structural risk minimization (SRM) induction principle . It ensures that the used set of functions has a structure that makes it strongly consistent [58]. Additionally it takes the complexity of the function that is used to approximate the target function into account. "The SRM principle actually suggests a tradeoff between the quality of the approximation and the complexity of the approximating function" [58, p. 994]. This reduces the risk of *overfitting*, meaning to overly fit the function to the training data with the result of poor generalization [5, chapter 1.3].

Support vector machines fulfill all conditions of the SRM principle. Due to the kernel trick that allows for nonlinear classification tasks it is also very powerful. For more detailed information on this see [26] and references therein.

7.3.2 Support Vector machines

In the case of linear binary classification one searches for a an affine hyperplane $w \in \mathbb{R}^n$ shifted by $b \in \mathbb{R}$ to separate the given data. The vector w is called weight vector and b is the bias.

Let the data be linearly separable. The function deciding how the data is classified can then be written as

$$f(x) = \text{sign}(\langle w, x\rangle - b).$$

Support vector machines aim at finding such a hyperplane that separates also unseen data optimally.

???Picture of hyperplane

One problem of this intuitive approach is that the representation of a hyperplane is not unique. If the plane described by (w, b) separates the data, there exist infinitely many hyperplanes (tw, b), $t > 0$, that separate the data in the same way. To have a unique description of a separating hyperplane the *canonical hyperplane for given data* $x \in X$ is defined by

$$f(x) = \langle w, x \rangle - b \quad \text{s.t.} \quad \min_i |\left\langle w, x^i \right\rangle - b| = 1.$$

This is always possible in the case where the data is linearly separable and means that the inverse of the norm of the weight vector is equal to the distance of the closest point $x \in X$ to the hyperplane [26, p. 10].

This gives rise to the following definition: The *margin* is the minimal Euclidean distance between a training example x^i and the separating hyperplane. A bigger margin means a lower complexity of the function [5].

A *maximal margin hyperplane* is the hyperplane that realizes the maximal possible margin for a given data set.

Proposition 7.1 ([5, Proposition 6.1]) Given a linearly separable training sample $\Omega = \{(x^i, y_i), ..., (x^l, y_l)\}$ the hyperplane (w, b) that solves the optimization problem

$$\|w\|^2 \quad \text{s.t.} \quad y_i(\langle w, x \rangle - b) \geq 1, \quad i = 1, ..., l,$$

realizes a maximal margin hyperplane.

The proof is given in [5, chapter 6.1].

Generally one cannot assume the data to be linearly separable. This is why in most applications a so called *soft margin classifier* is used. It introduces the slack variables ξ_i that measure the distance of the misclassified points to the hyperplane:

Fix $\gamma > 0$. A *margin slack variable of the example* (x^i, y_i) with respect to the hyperplane (w, b) and target margin γ is

$$\xi_i = \max(0, \gamma - y_i(\langle w, x \rangle + b))$$

If $\xi_i > \gamma$ the point is considered misclassified. One can also say that $\|\xi\|$ measures the amount by which training set "fails to have a margin of γ" [5, section 2.1.1].

For support vector machines the target margin is set to $\gamma = 1$.

This results in the following optimization problem for finding an optimal separating hyperplane (w, b):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i\left(\left\langle w, x^i\right\rangle - b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall i = 1, \ldots, l \end{aligned} \tag{7.4}$$

The first part of the objective function is the regularization, the second part the actual loss function. The parameter $C > 0$ gives a trade-off between the richness of the chosen set of functions f_λ to reduce the error on the training data and the danger of overfitting to have good generalization. It has to be chosen a priori [26].

To derive a subgradient of the bilevel problem introduced later in this section another formulation of the classification problem is used. It makes use of the *implicit bias*, meaning that the bias is not calculated separately but as part of the variable w. By adding an additional one to the end of each feature vector we can use the vectors $\tilde{x}_i := (x_i, 1)^\top$ and $\tilde{w} := (w, w_b)$ to state the following optimization problem:

$$\begin{aligned} \min_{\tilde{w},\xi} \quad & \frac{1}{2}\|\tilde{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i\left\langle \tilde{w}, \tilde{x}^i\right\rangle = y_i\left(\left\langle w, x^i\right\rangle - w_b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \\ & \forall i = 1, \ldots, l. \end{aligned} \tag{7.5}$$

Not treating the bias separately is a strategy used for example to achieve a more efficient implementation [10, section 3.2, p. 22]. In the course of this section the gain of this particular formulation is the fact that it has a unique solution compare for... where it is shown. It is however shown in [10, section 3.2, p. 22] that the solutions that are found with explicit and implicit bias are different. This results from the fact that in case of implicit bias it also enters the regularization term.

From now on we work with formulation (7.5) of the support vector classification problem. To see the derivation of the bilevel problem with the explicit bias compare for [26].

7.4 Bilevel Approach and Multiple Hyper-Parameters

The hyper-parameter C in the objective function of the classification problem has to be set beforehand. This step is part of the model selection process. To set this parameter optimally different methods can be used. A very intuitive and widely used approach is doing and *cross validation* (CV) with a grid search implementation [26, p. 30].

To prevent overfitting and get a good parameter selection, especially in case of little data, commonly T-fold cross validation is used [26, p. 30]. For this technique the training data is randomly partitioned into T subsets of equal size. One of these subsets is then left out of the training set and instead used afterwards to get an estimate of the generalization error. To use CV for model selection it has to be embedded into an optimization algorithm over the hyper-parameter space. Commonly this is done by discretizing the parameter space and for T-fold CV training T models at each grid point. The resulting models are then compared to find the best parameters in the grid. Obviously for a growing number of hyper-parameters this is very costly. An additional drawback is that the parameters are only chosen from a finite set [26, p. 30].

7.4.1 Reformulation as Bilevel Problem

A more recent approach is the formulation as a bilevel problem used in [26] and [36]. This makes it possible to optimize the hyper-parameters continuously.

Let $\Omega = \{(x^1, y_1), ..., (x^l, y_l)\} \subset \mathbb{R}^{n+1}$ be a given data set of size $l = |\Omega|$. The associated index set is denoted by $\mathcal{N}$. For classification the labels y_i are ± 1. For T-fold cross validation let $\bar{\Omega}_t$ and Ω_t be the training set and the validation set respectively within the t'th fold and $\bar{\mathcal{N}}_t$ and $\mathcal{N}_t$ the respective index sets. Furthermore let $f^t : \mathbb{R}^{n+1} \cap \mathcal{F} \to \mathbb{R}$ be the response function trained on the t'th fold and $\lambda \in \Lambda$ the hyper-parameter to be optimized. For a general machine learning problem with upper and lower loss function $\mathcal{L}_{upp}$ and $\mathcal{L}_{low}$ respectively the bilevel problem reads

$$
\begin{aligned}
\min_{\lambda, f^t} \quad & \mathcal{L}_{upp}\left(\lambda, f^1|_{\Omega_1}, ..., f^T|_{\Omega_T}\right) && \text{upper level} \\
\text{s.t.} \quad & \lambda \in \Lambda \\
& \text{for } t = 1, ..., T: \\
& f^t \in \left\{ \begin{array}{ll} \arg\min\limits_{f \in \mathcal{F}} & \mathcal{L}_{low}(\lambda, f, (x^i, y_i)_{i=1}^l \in \bar{\Omega}_t) \\ \text{s.t.} & g_{low}(\lambda, f) \leq 0 \end{array} \right\}. && \text{lower level}
\end{aligned}
\tag{7.6}
$$

In the case of support vector classification the T inner problems have the SVM formulation (7.5). This problem can also be rewritten into an unconstrained form. It is helpful when using the inexact bundle algorithm for solving the bilevel problem. For the t'th fold the resulting hyperplane is identified with the variable $\tilde{w}^t \in \mathbb{R}^{n+1}$. The inner level problem for the t'th fold can therefore be stated as

$$(\tilde{w}^t) \in \arg\min_{\tilde{w}} \left\{ \frac{1}{2}\|\tilde{w}\|^2 + C \sum_{i \in \bar{\mathcal{N}}_t} \max\left\{1 - y_i \left\langle \tilde{w}, \tilde{x}^i \right\rangle, 0\right\} \right\}. \tag{7.7}$$

For the upper level objective function there are different choices possible. All that are presented here use the implicit bias. They all can also be used with the explicit bias term.

Simply put the outer level objective should compare the different inner level solutions and pick the best one. An intuitive choice is therefore to pick the misclassification loss, that counts how many data points of the respective validation set Ω_t are misclassified when taking function f^t.

The misclassification loss can be written as

$$\mathcal{L}_{mis} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{N}_t|} \sum_{i \in \mathcal{N}_t} \left[-y_i \left\langle \tilde{w}^t, \tilde{x}^i \right\rangle \right]_\star, \tag{7.8}$$

where the step function $(\cdot)_\star$ is defined component wise for a vector as

$$(r_\star)_i = \begin{cases} 1, & \text{if } r_i > 0, \\ 0, & \text{if } r_i \leq 0 \end{cases}. \tag{7.9}$$

The drawback of this simple loss function is that it is not continuous and as such not suitable for subgradient based optimization. Therefore another loss function is used for the upper level problem - the *hinge loss* or *L1-loss*. It is an upper bound on the misclassification loss and reads

$$\mathcal{L}_{hinge} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{|\mathcal{N}_t|} \sum_{i \in \mathcal{N}_t} \max\left\{1 - y_i \left\langle \tilde{w}^t, \tilde{x}^i \right\rangle, 0\right\}. \tag{7.10}$$

It is also possible to square the max term. This results in the *L2-loss* function

$$\mathcal{L}_{hinge} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left\{1 - y_i\left\langle\tilde{w}^t, \tilde{x}^i\right\rangle, 0\right\}^2. \tag{7.11}$$

We refer to this second loss function also as *hingequad loss*.

For the bilevel problem discussed in this thesis the hingequad function is chosen as objective of the upper level problem. Hence the final resulting bilevel formulation for model selection in support vector classification is

$$\begin{aligned}
\min_{C} \quad & \mathcal{L}_{hingequad}(\tilde{w}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left\{1 - y_i\left\langle\tilde{w}^t, \tilde{x}^i\right\rangle, 0\right\}^2 \\
\text{s.t.} \quad & C > 0 \\
& \text{for } t = 1, ..., T \\
& \tilde{w}^t \in \arg\min_{\tilde{w}}\left\{\frac{1}{2}\|\tilde{w}\|^2 + C\sum_{i\in\bar{\mathcal{N}}_t}\max\left\{1 - y_i\left\langle\tilde{w}, \tilde{x}^i\right\rangle, 0\right\}\right\}.
\end{aligned} \tag{7.12}$$

7.4.2 Multiple Hyper-parameters

To examine the performance of the bilevel approach in the more dimensional case a model suggested by Moore et al. in [36] and called *multi-group* support vector classification (multiSVC). There different hyper-parameters are allowed for different subgroups of the trainings data. In section 4.3 of [36] the model is described for a regression function. In this thesis the same technique is used for classification.

The motivation behind the approach is that different groups of samples from the trainings set can have slightly different properties and should therefore have their own weighting parameters C_g. On the one hand this can improve the generalization results, on the other hand properties of the different data groups can be identified by their respective hyper-parameters. Moore et al. explain in [36, section 4.3, p. 9] that for example a large C_g signifies reliable data in the respective group whereas a smaller C_g suggests a poorer quality.

To perform multiSVC divide the trainings data into G (pairwise disjoint) groups. Define the vector of hyper-parameters $C := [C_1, ..., C_G]^\top$. For T-fold cross validation the t'th lower level problem in the constrained form can be stated as

$$
\begin{aligned}
\min_{\tilde{w},\xi} \quad & \frac{1}{2}\|\tilde{w}\|^2 + \sum_{g=1}^{G}\left(C_g \sum_{i\in\bar{\mathcal{N}}_t^g} \xi_i\right) \\
\text{s.t.} \quad & y_i \left\langle \tilde{w}, \tilde{x}^i \right\rangle \geq 1 - \xi_i \\
& \xi_i \geq 0 \\
& \forall i \in \bigcup_{g=1}^{G} \bar{\mathcal{N}}_t^g = \bar{\mathcal{N}}_t.
\end{aligned}
$$

Here $\bar{\mathcal{N}}_t^g$ is the index set that contains the indices of the data of the g'th group in the t'th fold.

The whole bilevel problem (stated with the unconstrained lower level problem) is given by

$$
\begin{aligned}
\min_{C} \quad & \mathcal{L}_{hingequad}(\tilde{w}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left\{1 - y_i\left\langle \tilde{w}^t, \tilde{x}^i\right\rangle, 0\right\}^2 \\
\text{s.t.} \quad & C = [C_1, ..., C_G]^\top > 0 \text{ and} \\
& \text{for } t = 1, ..., T \\
& \tilde{w}^t \in \arg\min_{\tilde{w}} \left\{\frac{1}{2}\|\tilde{w}\|^2 + \sum_{g=1}^{G}\left(C_g \sum_{i\in\bar{\mathcal{N}}_t^g}\max\left\{1 - y_i\left\langle \tilde{w}, \tilde{x}^i\right\rangle, 0\right\}\right)\right\}.
\end{aligned}
\tag{7.13}
$$

As the multigroup problem contains the single group bilevel problem (7.12) as a special case we consider only problem (7.13) in the following sections.

7.5 Solution with the Inexact Bundle Algorithm

more introduction
remark on missing error bounds for function value and subgradient

The bilevel problem (7.13) derived above is to be solved with the bilevel algorithm which one? more?. This requires having in every iteration an approximate value of the upper level objective function and an approximate subgradient of the upper level objective.

To see the issues arising when computing especially the subgradient of a bilevel objective let us again consider the general bilevel problem (7.1).

At the current iterate x^k (this is now no data point but the optimization variable of the upper level problem) the function value of the upper level objective function can be computed by solving the lower level problem given x^k. The resulting solution y^k is then inserted into the upper level objective and its value can be calculated.

For computing a subgradient, it is not that simple. Additionally to the variation of the upper level function F with respect to the variable x also the variation of the solution y^k with respect to x has to be considered. To do this we follow the strategy described in [43]. Refer there also for a more throughout analysis on this subject.

Consider the general bilevel problem formulation (7.1) without the explicit constraint $G(x, y) \leq 0$. Let the feasible set X be a nonempty compact set and $\tilde{A}$ and open set containing X.

To use the implicit programming approach suggested in the book, the following assumptions have to hold true:

(A1) The upper level objective F is continuously differentiable on $\tilde{A} \times \mathbb{R}^k$.

(A2) The lower level program possesses a unique solution y_x for every $x \in \tilde{A}$.

(A3) The generalized equation coming from the lower level program is strongly regular at all points (x, y_x), where $x \in \tilde{A}$ and y_x is the corresponding solution of the lower level problem.

A *(perturbed) generalized equation* is a relation of the form

$$0 \in H(x, y) + N_U(y),$$

where $H : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^k$ and $N_U(y)$ denotes the *normal cone* to the convex set $U \subset \mathbb{R}^k$ at the point $y \in \mathbb{R}^k$.

Definition 7.2 ([43, Definition 2.6, p. 18]) Let U be a convex subset of $\mathbb{R}^k$ and y in the closure of U. Then

$$N_U(y) := \{\xi \in \mathbb{R}^k \mid \langle \xi, z - y \rangle \leq 0 \quad \forall z \in U\}$$

is called normal cone to U at y.

The constraint induced by the inner level problem can be rewritten as a generalized equation via its optimality condition. Let U correspond to the feasible set of the inner level problem defined by the intersection of the set Y with the set defined by $g(\bar{x}, y) \leq 0$ for

a fixed variable $\bar{x} \in X$. Then the lower level problem can be expressed in an unconstrained way by using the indicator function defined in (3.8). If the constraint set is convex, the indicator function is a convex function and for every element of its subdifferential the subgradient inequality (6.12) holds.

This yields that for all elements ξ of the subdifferential $\partial \mathtt{i}_U(y)$ it holds

$$\begin{aligned} &\mathtt{i}_U(z) - \mathtt{i}_U(y) \geq \langle \xi, z - y \rangle \quad \forall z \in U \\ \Leftrightarrow &\langle \xi, z - y \rangle \leq 0 \quad \forall z \in U, \end{aligned}$$

because for $y, z \in U$ the indicator function is zero. This means that the subdifferential of the indicator function of the set U at the point y coincides with the normal cone to the set U at the point y and so for a constrained minimizer of the function $f(x, \cdot)$ on the set U the following optimality condition holds:

$$0 \in \partial f(x, y) + N_U(y).$$

If the subdifferential of f at the point (x, y) is single valued, this is a generalized equation.

Let us now verify the above assumptions for the hyper-parameter finding bilevel problem (7.13). First of all, the feasible set X has to be a convex compact set. In order to assure that, the constraint of the hyper-parameter C has to be adapted. Therefore choose two constants $l_C, u_C > 0$ with $l_C < u_C$ and constrain the vector C by the component wise meant inequalities

$$l_c \leq C \leq u_C.$$

Assumption (A1) is fulfilled as $\mathcal{L}_{hingequad}$ is a continuously differentiable function because of the squared max-term.

Next assumption (A2) is verified. In order to do this consider the lower level problem in its unconstrained formulation. It is the sum of a strictly convex ($\frac{1}{2}\|\tilde{w}\|^2$) and a convex function ($\sum_{g=1}^{G} \left(C_g \sum_{i \in \bar{\mathcal{N}}_t^g} \max\left\{1 - y_i \left\langle \tilde{w}, \tilde{x}^i \right\rangle, 0\right\}\right)$). This means the function is strictly convex. It has therefore a unique global minimizer $\tilde{w}^*$ for any vector of hyper-parameters C. Hence the assumption is fulfilled.

Finally to show that assumption (A3) holds for the considered, strong regularity has to be shown for the corresponding generalized equation. In order to do this the lower level

problem is considered in its constrained form.

Let $\tilde{w}^* \in \mathbb{R}^{n+1}$ denote the unique solution of the unconstrained problem. Due to the equivalence of the two problem formulations the optimal solution $(\tilde{w}^*, \xi^*)$ of the constrained problem is given by $\xi_i = 1 - y_i \left(\langle w, x^i \rangle - b\right)$.

7.6 Numerical Experiments

In this section algorithm 5.1 is used to solve the bilevel problems presented above for different synthetic and real world data sets.

- Problem what to optimize
- comment on nonlinear
- overfitting
- pictures that show situation

7.6.1 Selection of the Data Sets

- parameter λ against overfitting, there to make generalization better
- data sets that "allow" for high overfitting: not too big, many features

The following two real world data sets are used:

Data set	l_{train}	l_{test}	n	T
Wisconsin Breast Cancer Database	240	443	9	3
John Hopkins University Ionosphere Database	240	111	33	3

Table 1

Plots

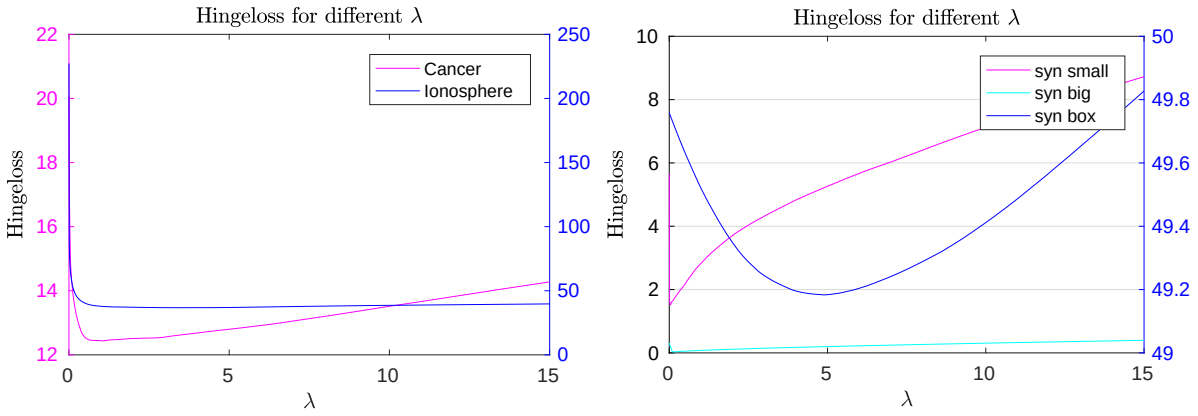

Figure 11: *Plots of the hingloss error for different λ values. The figure on the right shows the plot for the* `cancer` *(left axis) and* `ionosphere` *(right axis) data sets. The blot on the left depicts the situation for the sythetic sets* `syn small`, `syn big` *(left axis) and* `syn box` *(right axis).*

results

Data set	algorithm	0.1	1	10	100
`cancer`	Bundle	1.0975	1.0974	1.0974	1.1341
	Noll Bundle	1.0974	1.0974	1.0974	1.0974
	fminsearch	1.0974	1.0974	1.0974	-10
	fmindnb	1.0974			
`ionosphere`	Bundle	2.6081	2.3108	0.3953	4.7401
	Noll Bundle	3.5386	3.6521	10	8.2773
	fminsearch	3.5104	3.5104	3.5104	3.5104
	fminbnd	3.5104			
`syn small`	Bundle	0.0337	0.031	0.0329	0.03
	Bundle Noll	0.0323	0.0352	0.0352	0
	fminsearch	0.0346	-0.1939	-1.79	-3.5673
	fminbnd	0.0346			
`syn big`	Bundle	0.1	1	10	100
	Bundle Noll	0.0099	0.0108	0.0109	0.0086
	fminsearch	0.0114	-0.8748	-0.8748	-0.8748
	fminbnd	0.0206	plot: 0.0114		
`syn box`	Bundle	0.1	1	4.8486	4.45336
	Bundle Noll	0.1	1	4.2760	0
	fminsearch	4.8950	4.8950	4.8950	4.8950
	fminbnd	4.8950			

Table 2

times

Data set	algorithm	0.1	1	10	100
`syn big`	Bundle Noll	202	320	386	503
	fminsearch	672	637	587	741
	fminbnd	557			

Table 3

sometimes says problem not convex, then negative values. Why??? Because tries negative lambda values in course of optimization and then it gets negative
sometimes even for negative lambda in the end right solution
can I find out why bundle methods so bad?
try matlab method (smooth) with subgradient???

7.6.2 Solution of the Bilevel Program

ab hier: Theorie fehlt
!!! notation - oder in prelininaries einfugen

To solve the given bilevel problem with the above presented nonconvex inexact bundle algorithm the algorithm jumps between the two levels. Once the inner level problems are solved for a given λ - this is possible with any QP-solver as the problems are convex - the bundle algorithm takes the outcome w and b and optimizes the hyper-parameter again.

The difficulty with this approach is that the bundle algorithm needs one subgradient of the outer level objective function with respect to the parameter λ. However to compute this subgradient also one subgradient of w and b with respect to λ has to be known.

The Differentiable Case example in differentiable case
Let us first assume that the outer and inner objective functions and $w(\lambda) = \arg\min \mathcal{L}_{low}(w, \lambda)$ are sufficiently often continuously differentiable to demonstrate the procedure of calculating the needed (sub-)gradients.

Let $\mathcal{L}_{upp}(w, \lambda)$ be the objective function of the outer level problem, where the variable b was left out for the sake of simplicity. To find an optimal hyper parameter λ given the input w the gradient g_λ^{upp} of $\mathcal{L}_{upp}$ with respect to λ is needed in every iteration of the solving algorithm. In order to calculate this gradient the chain rule is used yielding

$$g_\lambda^{upp} = \left(\frac{\partial}{\partial w}\mathcal{L}_{upp}(w, \lambda)\right)^\top \frac{\partial w(\lambda)}{\partial \lambda} + \frac{\partial}{\partial \lambda}\mathcal{L}_{upp}(w, \lambda).$$

The challenge is here to find the term $\frac{\partial w(\lambda)}{\partial \lambda}$ because

$$\frac{\partial w}{\partial \lambda} \in \frac{\partial}{\partial \lambda} \arg\min_{w} \mathcal{L}_{low}(w, \lambda).$$

Assuming $\mathcal{L}_{low}$ is twice continuously differentiable at the optimal solution w^* of the lower level problem the optimality condition for any parameter $\lambda_0 > 0$

$$0 = \frac{\partial}{\partial w} \mathcal{L}_{low}(w^*, \lambda_0) \tag{7.14}$$

can be used to calculate the needed gradient in an indirect manner.

In the differentiable case the theoretical framework for the following calculations is given by the implicit function theorem.

Theorem 7.3 (c.f. [24, chapter 3.4]) *Let $F : U \times V \to Z$, $U \in \mathbb{R}^m, V, Z \in \mathbb{R}^n$, be a $\mathcal{C}^1$ mapping, $(x_0, y_0) \in U \times V$ and $F(x_0, y_0) = 0$. If the matrix $\frac{\partial}{\partial y} F(x_0, y_0)$ is invertible, there exist neighborhoods $U_0 \subset U$ of x_0 and $V_0 \subset V$ of y_0 and a continuously differentiable mapping $f : U_0 \to V_0$ with*

$$F(x, y) = 0, (x, y) \in U_0 \times V_0 \quad \Leftrightarrow \quad y = f(x), x \in U_0.$$

Identifying $x \hat{=} \lambda$, $y \hat{=} w$ and $F(x_0, y_0) \hat{=} \frac{\partial}{\partial w} \mathcal{L}_{low}(w^*, \lambda_0)$ and assuming $\frac{\partial^2}{\partial w^2} \mathcal{L}_{low}(w^*, \lambda_0)$ is invertible this theorem provides the existence of the continuously differentiable function $w(\lambda)$ whose gradient is needed.
what about the neighborhoods?

If the inner level loss function yields a linear optimality condition in w it is possible to calculate the gradient explicitly. This is for example the case for SVM loss functions with a squared one- or two-norm as given in problem (7.4). The optimality condition can then be written as the linear system

$$H(\lambda) w = h(\lambda).$$

By taking the partial derivative with respect to λ on both sides of the system one gets

$$\frac{\partial H(\lambda)}{\partial \lambda} w + H(\lambda) \frac{\partial w}{\partial \lambda} = \frac{\partial h(\lambda)}{\partial \lambda}.$$

If $H(\lambda)$ is invertible for all $\lambda \in \Lambda$ then the needed gradient is given by

$$\frac{\partial w}{\partial \lambda} = H^{-1}(\lambda)\left(\frac{\partial h(\lambda)}{\partial \lambda} - \frac{\partial H(\lambda)}{\partial \lambda} w\right).$$

The Nondifferentiable Case now for subgradients

In practice we cannot expect $\mathcal{L}_{low}$ to satisfy such strong differentiability properties. It is therefore only assumed that $\mathcal{L}_{low}$ is once continuously differentiable in w. This assures that the optimality condition of the lower level problem is an equality like in (7.14). Contrary to the exemplary calculations from above in practice the second derivative $\frac{\partial^2}{\partial w \partial \lambda}\mathcal{L}_{low}(w(\lambda), \lambda)$ however is not existent in this form, but a set of subgradients.

Notation

First the theoretical framework given to derive the results from above in the nondifferentiable case

An important result about Lipschitz functions is Rademacher's theorem which states that these functions are differentiable almost everywhere but on a set of Lebesgue measure zero[14, Theorem 3.1]. Clarke deduces from this that the subdifferential at each of the nondifferentiable points is the convex hull of the limits of the sequence gradients a these points [3, see Theorem 2.5.1].

This motivates the multidimensional definition of Clake's generalized gradient

Definition 7.4 ([3, Definition 2.6.1]) *generalized Jacobian*: $F : \mathbb{R}^n \to \mathbb{R}^m$, with locally Lipschitz component functions $F(x) = (f_1(x), ..., f_m(x))$.
Denote generalized Jacobian by $\partial F(x) = conv\,(\lim JF(x_i) | x_i \to x, x_i \notin \Omega_F)$ where Ω_F is the set of nondifferentiable points of F

(after that comes proposition with properties of ∂F)

To facilitate readability we use the following notation for the derivation of the nondifferentiable results.

The *'partial' subdifferential* of a function $f(a^*, b_0, c_0, ...)$ at the point a^* with respect to one variable a when all other variables are fixed is denoted by

$$\partial^a f(a^*, b_0, c_0, ...).$$

A subgradient of this subdifferential is written $g^a \in \partial^a f(a^*, b_0, c_0, ...)$.

Next step: show that chain rule is still valid in the nonsmooth case Chain rules for sub-

differential

Theorem 7.5 ([3, Theorem 2.6.6]) *Let $f(x) = \phi(F(x))$, with the locally Lipschitz functions $F : \mathbb{R}^n \to \mathbb{R}^m$ and $\phi : \mathbb{R}^m \to \mathbb{R}$. Then f is locally Lipschitz and it holds*

$$\partial f \subset conv\{\partial\phi(F(x))\partial F(x)\}\,.$$

If in addition ϕ is strictly differentiable at $F(x)$, then equality holds.

strictly differentiable: c.f. [47, Theorem 9.17 and 9.18] locally Lipschitz continuous and at most one subgradient at the point in question (see also comment to Definition 91 in [47])

Theorem 7.6 (c.f. [45, Theorem 7.1]) *Let $p(x) = f(F(x))$, where $F : \mathbb{R}^n \to \mathbb{R}^d$ is locally Lipschitz and $f : \mathbb{R}^d \to \mathbb{R}$ is lower semicontinuous. Assume*

$$\nexists y \in \partial^\infty f(F(\bar{x})), y \neq 0 \quad with \quad 0 \in y\partial F(\bar{x}).$$

Then for the sets

$$M(\bar{x}) := \partial f(F(\bar{x}))\partial F(\bar{x}), \quad M^\infty(\bar{x}) := \partial^\infty f(F(\bar{x}))\partial F(\bar{x}),$$

one has $\hat{\partial}p(\bar{x}) \subset M(\bar{x})$ and $\hat{\partial}^\infty p(\bar{x}) \subset M^\infty(\bar{x})$.

The main idea is to replace the inner level problem by its optimality condition

$\partial(w, b)$ means in this case that the subdifferential is taken with respect to the variables w and b.
-> theory for subdifferentials in more than one variable!!!

For convex inner level problem this replacement is equivalent to the original problem.

The difference to the approach described in [26] is that the problem is not smoothly replaced by its KKT conditions but only by this optimality condition. The weight vector w and bias b are treated as a function of λ and are optimized separately from this hyperparameter. The reformulated bilevel problem becomes:

$$
\begin{aligned}
\min_{w,\boldsymbol{b}} \quad & \mathcal{L}_{hinge}(w, \boldsymbol{b}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left(1 - y_i((w^t)^\top x - b_t), 0\right) \\
\text{subject to} \quad & \lambda > 0 \\
& \text{for } t = 1, ..., T \\
& 0 \in \partial(w, b)\mathcal{L}_{low}(\lambda, w^t, b_t)
\end{aligned}
\tag{7.15}
$$

where $\mathcal{L}_{low}$ can be the objective function of either of the two presented lower level problems.

solve the inner level problem (quadratic problem in constrained case) by some QP solver put solution into upper level problem and solve it by using bundle method
difficulty: subgradient is needed to build model of the objective function –> need subgradient $\frac{\partial\mathcal{L}}{\partial\lambda}$ –> for this need $\frac{\partial(W,b)}{\partial\lambda}$
but (w, b) not available as functions -> only values

Moore et al. [36] describe a method for getting the subgradient from the KKt-conditions of the lower level problem:

lower level problem convex -> therefore optimality conditions (some nonsmooth version -> source???) necessary and sufficient -> make "subgradient" of optimality conditions and then derive subgradient of w, b from this.
—> what are the conditions? optimality condition Lipschitz?

Say (show) that all needed components are locally Lipschitz; state theorems about differentiability almost everywhere and convex hull of gradients gives set of subgradients
introduce special notation (only for this section) and because of readability adopt "gradient writing"

Subgradients: $\mathcal{G}_{upp,\lambda}, \mathcal{G}_{upp,w}, \mathcal{G}_{upp,b}$ -> subgradients of outer objective
g_w, g_b -> subgradient of w, b

$$
finalsubgradient = \left(\mathcal{G}_{upp,w}(w, b, \lambda)\right)^\top g_w + \left(\mathcal{G}_{upp,b}(w, b, \lambda)\right)^\top g_b + \mathcal{G}_{upp,\lambda}(w, b, \lambda)
$$

subgradients $\mathcal{G}_{upp,...}$ easy to find (assumption that locally Lipschitz) -> in this application differentiable

difficulty: find g_w, g_b important: optimality condition must be a linear system in w, b ->

this is the case in this application

$$H(\lambda) \cdot (w, b)^\top = h(\lambda)$$

find subgradients of each element (from differentiation rules follows)

$$\partial_\lambda H \cdot (w, b)^\top + H \cdot (\partial_\lambda w, \partial_\lambda b)^\top = \partial_\lambda h$$

solve this for (w, b):

$$(\partial_\lambda w, \partial_\lambda b)^\top = H^{-1} \left(\partial_\lambda h - \partial_\lambda H \cdot (w, b)^\top \right)$$

matrix H has to be inverted -> in the feature space so scalable with size of data set -> still can be very costly [36]

Applied to the two bilevel classification problems derived above, the subgradients have the following form:

derivative of upper level objective: Notation: $\delta_i := 1 - y_i(w^\top x^i - b)$

$$\partial_w \mathcal{L}_{upp} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{\mathcal{N}_t} \sum_{i \in \mathcal{N}_t} \begin{cases} -y_i x^i & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.16}$$

$$\partial_b \mathcal{L}_{upp} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{\mathcal{N}_t} \sum_{i \in \mathcal{N}_t} \begin{cases} y_i & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.17}$$

here at the kink subgradient 0 is taken

for hingequad: -> here subgradient
optimality condition:

$$0 = \partial_w \mathcal{L}_{low} = \lambda w + 2 \sum_{i \in \bar{\mathcal{N}}_t} \begin{cases} (1 - y_i(w^\top x^i - b))(-y_i x^i) & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.18}$$

$$0 = \partial_b \mathcal{L}_{low} = 2 \sum_{i \in \bar{\mathcal{N}}_t} \begin{cases} (1 - y_i(w^\top x^i - b))(y_i) & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.19}$$

subgradient??? is this smooth? with respect to λ

$$0 = w + \lambda \partial_\lambda w + 2 \sum_{i \in \bar{\mathcal{N}}_t} \begin{cases} (-y_i(\partial_\lambda w^\top x^i - \partial_\lambda b))(-y_i x^i) & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.20}$$

$$0 = 2 \sum_{i \in \bar{\mathcal{N}}_t} \begin{cases} (-y_i(\partial_\lambda w^\top x^i - \partial_\lambda b))(y_i) & \text{if } \delta_i > 0 \\ 0 & \text{if } \delta_i \leq 0 \end{cases} \tag{7.21}$$

From this the needed subgradients can be calculated via:

$$2 \cdot \begin{pmatrix} \frac{\lambda}{2} + \sum_{i \in \bar{\mathcal{N}}_t} y_i^2 x^i (x^i)^\top & \sum_{i \in \bar{\mathcal{N}}_t} -y_i^2 x^i \\ \sum_{i \in \bar{\mathcal{N}}_t} -y_i^2 (x^i)^\top & \sum_{i \in \bar{\mathcal{N}}_t} y_i^2 \end{pmatrix} \cdot \begin{pmatrix} \partial_\lambda w \\ \partial_\lambda b \end{pmatrix} = \begin{pmatrix} -w \\ 0 \end{pmatrix} \tag{7.22}$$

for hinge not quad:

not as much information in the subgradient/derivative

similar calculation leads to

$$\partial_\lambda w = -\frac{w}{\lambda} \tag{7.23}$$

$$\partial_\lambda b = 0 \tag{7.24}$$

does not work with just Hingeloss because set valued optimality condition –> have to find “corresponding” equation to the chosen subgradient –> do not know how

7.6.3 Multi Group Model

Idea: different samples within the group have different have different quality –> put the samples with similar quality within one group –> give it its own λ_g such that the groups are weighted depending on their quality.

two goals: better modleing and implicit information about the groups provided by the parameter λ_g -> small λ good quality, large λ bad quality???

Examle: different people do same experiment; measure same data...

Model (Bilevel Problem)

for simplicity: all groups of same size; in all folds same number of elements of each group

!! not possible to work with λ here, have to work with C-Formulation

matrices for MATLAB calculations:

$$\frac{1}{2}\langle x, Hx\rangle + h$$
$$Ax \leq b$$

$$H = \begin{pmatrix} \mathbb{I}_{n\times n} & & \\ & 0 & \\ & & 2C \end{pmatrix} \in \mathbb{R}^{feat+1+J\max\{T-1,1\}G \times feat+1+J\max\{T-1,1\}G} \tag{7.25}$$

$$h = 0, \qquad A = \begin{pmatrix} -YX^\top & Y & -\mathbb{I}_{J\max\{T-1,1\}G \times J\max\{T-1,1\}G} \end{pmatrix}, \qquad b = -\texttt{ones}(J\max\{T-1,1\}G, 1)$$

with $C = [\underbrace{c_1, ..., c_1}_{J\max\{T-1,1\} \text{ times}}, c_2, ..., c_2, ..., c_G, ..., c_G]^\top$.

Derivatives

overall gradient:

$$\frac{\mathrm{d}\mathcal{L}_{upp}}{\mathrm{d}\lambda} = \underbrace{\frac{\partial \mathcal{L}_{upp}}{\partial(w,b)}}_{\in\mathbb{R}^{n+1}}^\top \underbrace{\frac{\partial(w,b)}{\partial\lambda}}_{\in\mathbb{R}^{n+1\times G}} \in \mathbb{R}^G$$

gradient outer level objective:

$$\frac{\partial \mathcal{L}_{upp}}{\partial w} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\delta_i\left(-y_i x^i\right) \in \mathbb{R}^n$$

$$\frac{\partial \mathcal{L}_{upp}}{\partial b} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\delta_i y_i \in \mathbb{R}$$

$$\delta_i = \begin{cases} 1 & \text{if } 1 - y_i\left(\langle w, x^i\rangle - b\right) > 0 \\ 0 & \text{else} \end{cases}$$

in MATLAB:

$$\frac{\partial \mathcal{L}_{upp}}{\partial w} = \frac{1}{T}\texttt{sum}(-\texttt{bsxfun}(\texttt{@times}, X, \texttt{delta}.*Y), 2) \in \mathbb{R}^n$$

$$\frac{\partial \mathcal{L}_{upp}}{\partial b} = \frac{1}{T}\texttt{sum}(\texttt{delta.} * Y, 1) \in \mathbb{R}$$

gradient lower level objective:
first optimality conditions $0 = \frac{\partial \mathcal{L}_{low}}{\partial w}, 0 = \frac{\partial \mathcal{L}_{low}}{\partial b}$:

$$0 = \sum_{g=1}^{G}(\lambda_g w) + 2\sum_{i \in \bar{\mathcal{N}}_t} \delta_i \left\{1 - y_i \left(\left\langle w, x^i \right\rangle - b\right)\right\}(-y_i x^i) \in \mathbb{R}^n$$

$$0 = 2\sum_{i \in \bar{\mathcal{N}}_t} \delta_i \left\{1 - y_i \left(\left\langle w, x^i \right\rangle - b\right)\right\}(y_i) \in \mathbb{R}$$

derivatives with respect to $\lambda_g, g = 1, ..., G$, then $\in \mathbb{R}^{n+1 \times G}$

write one column of the matrix $\in \mathbb{R}^{n+1} \to G$ such columns:

$$0 = w + \lambda_g \frac{\partial w}{\partial \lambda_g} + 2\sum_{i \in \bar{\mathcal{N}}_t} \delta_i \left(y_i^2 x^i (x^i)^\top \frac{\partial w}{\partial \lambda_g} - y_i^2 x^i \frac{\partial b}{\partial \lambda_g}\right) \in \mathbb{R}^n$$

$$0 = 2\sum_{i \in \bar{\mathcal{N}}_t} \delta_i \left(-y_i^2 (x^i)^\top \frac{\partial w}{\partial \lambda_g} + y_i^2 \frac{\partial b}{\partial \lambda_g}\right) \in \mathbb{R}$$

rewrite to calculate $[\partial w/\partial \lambda, \partial b/\partial \lambda]^\top \in \mathbb{R}^{n+1 \times G} \to$ have to calculate every column of this matrix on its own
solve G times this system of equations

$$2\begin{pmatrix} \frac{\lambda_g}{2}\mathbb{I} + \sum_{i \in \bar{\mathcal{N}}_t} \delta_i (y_i^2 x^i (x^i)^\top) & \sum_{i \in \bar{\mathcal{N}}_t} \delta_i (-y_i^2 x^i) \\ \sum_{i \in \bar{\mathcal{N}}_t} \delta_i (-y_i^2 (x^i)^\top) & \sum_{i \in \bar{\mathcal{N}}_t} \delta_i (y_i^2) \end{pmatrix} \begin{pmatrix} \frac{\partial w}{\partial \lambda_g} \\ \frac{\partial b}{\partial \lambda_g} \end{pmatrix} = \begin{pmatrix} -w \\ 0 \end{pmatrix}$$

7.7 Application of Outrata-theory to bilevel problem

Remark: For multigroup (optimization in more than one variable) only "C-formulation" possible, not λ-formulation.

procedure taken from Outrata et al. in [43]

Follow steps in Appendix A:

A.1 Problem

bilevel problem has to be adapted to fit the theory

$\rightarrow$ general: U_{ad} compact and convex, this means adapt constraint for the C_g; (no disadvantage compared to grid search)

$\rightarrow$ Inner level problem: introduce implicit bias
By adding a column of ones to the data matrix X the last component of $\tilde{w} := (w, w_b)^\top$ is the bias
Disadvantage: bias now also in regularization
Advantage: problem fits theory (for Hingequad)

$\rightarrow$ Outer level problem: has to be continuously differentiable
Take hingequad function
reformulation of outer level problem as constrained smooth optimization problem (for both loss functions possible) seems to be not possible as outer level problem (A.1) in [43] is unconstrained in z except for the implicit function constraint

$\Rightarrow$ Bilevel problem:

1.

$$\begin{aligned}
\min_{\tilde{w}} \quad & \mathcal{L}_{hingequad}(\tilde{w}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{|\mathcal{N}_t|}\sum_{i\in\mathcal{N}_t}\max\left\{1 - y_i\left\langle\tilde{w}^t, x^i\right\rangle, 0\right\}^2 \\
\text{s.t.} \quad & \bar{c} \le C_g \le \bar{C}, \quad \text{for} \quad \bar{c}, \bar{C} \in \mathbb{R}_+, \quad \bar{c} < \bar{C} \quad g = 1, ..., G \\
& \text{for } t = 1, ..., T \\
& \left\{\begin{array}{l} \tilde{w}^t \in \arg\min_{\tilde{w}} \left\{\frac{1}{2}\left(\|\tilde{w}\|_2^2 + \sum_{g=1}^{G} C_g \sum_{i\in\bar{\mathcal{N}}_t^g} \xi_i^2\right)\right\} \\ \text{s.t. for } i \in \bigcup_{g=1}^{G}\mathcal{N}_t^g : \quad y_i\left\langle\tilde{w}, x^i\right\rangle \ge 1 - \xi_i \end{array}\right\}
\end{aligned} \tag{7.26}$$

corresponding variables:
$x \rightarrow (C_1, ..., C_G)$
y, z (both used for the same variable in Outrata) $\rightarrow (\tilde{w}, \xi)^\top$

A.2 Assumptions

- U_{ad} compact, nonempty ✓
- $\mathcal{L}_{hingequad}$ is continuously differentiable ✓
- S is single valued on $\tilde{A}$ (open set containing U_{ad})
  this follows from theorem 4.8 in [43, p. 82] ✓
  derivation see below
- the considered GE is strongly regular (at all points (x, z) with $x \in \tilde{A}, z = S(x)$)
  this follows form theorem 5.8 in [43, p. 96] ✓
  derivation see below

Derivations

To assert the above assumptions rewrite lower level problem as given in A.3.1:

size of variables: $\tilde{w} \in \mathbb{R}^{feat+1}$, $\xi \in \mathbb{R}^N$

$$\begin{aligned} &\min_{(\tilde{w},\xi)} f(\tilde{w},\xi) := \frac{1}{2}\left\langle \begin{pmatrix}\tilde{w}\\ \xi\end{pmatrix}, \begin{pmatrix}\mathbb{I} & \\ & C\end{pmatrix}\begin{pmatrix}\tilde{w}\\ \xi\end{pmatrix}\right\rangle \\ &\text{s.t.} \begin{pmatrix} -y_1(x^1)^\top & -1 & & \\ \vdots & & \ddots & \\ -y_N(x^N)^\top & & & -1 \end{pmatrix}\begin{pmatrix}\tilde{w}\\ \xi\end{pmatrix} \leq \begin{pmatrix}-1\\ \vdots\\ -1\end{pmatrix} \\ &\quad \text{for} \quad i \in \bigcup_{g=1}^{G} \mathcal{N}_t^g \end{aligned} \tag{7.27}$$

with $C := (\underbrace{C_1, ..., C_1}_{|\mathcal{N}_t^1| \text{ times}}, ..., \underbrace{C_G, ..., C_G}_{|\mathcal{N}_t^G| \text{ times}})^\top$ and $N := |\bigcup_{g=1}^{G} \mathcal{N}_t^g|$.

Definition 7.7 ([43, Definition 4.2, p. 79]) The function F is said to be strongly monotone on Ω if there exists an $\alpha > 0$ such that

$$\langle F(v) - F(w), v - w\rangle \geq \alpha\|v - w\|^2 \quad \text{for all } v, w \in \Omega.$$

Theorem 7.8 ([43, Theorem 4.4 (ii), p. 79]) *If F is strongly monotone on Ω, then the equilibrium problem has exactly one solution.*

Check strong monotonicity for $F = \nabla f$:

$$\left\langle \begin{pmatrix} \mathbb{I} & \\ & C \end{pmatrix} (v-w), v-w \right\rangle = \sum_{i=1}^{N} \lambda_i (v_i - w_i)^2 \geq \min\{1, \lambda_{min}\} \|v-w\|^2, \quad \forall v, w \in \mathbb{R}^{feat+1+N}.$$

Here the $\lambda_i > 0$ are the eigenvalues and λ_{min} the largest eigenvalue of the positive definite matrix $\begin{pmatrix} \mathbb{I} & \\ & C \end{pmatrix}$. As the C_g are constrained away from 0 all eigenvalues are strictly larger than zero. Thus the theorem applies.

The set Ω is the constraint set of the lower level problem. It is given in the form of (4.3) in the book.

Theorem 7.9 ([43, Theorem 4.8, p. 82]) *Let F be strongly monotone on Ω given by (4.3) and $(\bar{w}, \bar{\xi})$ be the unique solution of the lower level problem. Assume that the linear independence constraint qualification holds at $(\bar{w}, \bar{\xi})$. Then the GE (4.14) (special form of the GE, provided in this case) possesses a unique solution $(\hat{y}, \hat{\lambda})$ with $\hat{y} = (\bar{w}, \bar{\xi})$.*

Theorem above essentially states that there is a unique solution of the KKT equations. Compare also to theorem 16.14 in [56].

Frage: wie ist $\bar{y} := y(\bar{x})$ in der (ESCQ) auf Seite 92 gemeint???
Ist $\bar{y}$ die optimale Lösung des lower level Problems fü den Parameter $\bar{x}$ oder ist es einfach irgendein y und es soll lediglich die Zugehörigkeit in der (ESCQ) zum Parameter $\bar{x}$ ausdrücken?

Theorem 7.10 ([43, Theorem 5.8, p. 96]) *Suppose that for some $x_0 \in \tilde{\mathcal{A}}$ the gradients $\nabla_y h^i(x_0, y_0)$, for $i = 1, ..., l$ and $\nabla_y g^j(x_0, y_0)$, for all $j \in \{1, ..., s\}$ corresponding to active inequalities at the point (x_0, y_0) are linearly independent.*
Additionally suppose that the Jacobian with respect to y of the Lagrangian $(J)_y \mathcal{L}(x_0, y_0, \mu_0, \lambda_0)$ is strictly copositive with respect to $\ker(\mathcal{J}_y H(x_0, y_0)) \cap \ker(\mathcal{J}_y \underbrace{G_{I^+}}_{\text{only active inequalities with } \lambda^j > 0} (x_0, y_0))$.
Then the GE is strongly regular.

Linear independence:

no equalitiy constraints

Jacobian of the inequality constraints: $\begin{pmatrix} -y_1(x^1)^\top & -1 & & \\ \vdots & & \ddots & \\ -y_N(x^N)^\top & & & -1 \end{pmatrix}$
linearly independent because of identity in the last part of the matrix

Lagrangian of the lower level problem:

$$\mathcal{L}(\tilde{C},(\tilde{w},\xi),\lambda) = \begin{pmatrix} \mathbb{I} & \\ & C \end{pmatrix} \begin{pmatrix} \tilde{w} \\ \xi \end{pmatrix} + (\lambda_1,...,\lambda_N) \begin{pmatrix} -y_1(x^1)^\top & -1 & \\ \vdots & & \ddots \\ -y_N(x^N)^\top & & -1 \end{pmatrix}$$

Jacobian with respect to $(\tilde{w},\xi)$:

$$\nabla_{(\tilde{w},\xi)}\mathcal{L}(\tilde{C},(\tilde{w},\xi),\lambda) = \begin{pmatrix} \mathbb{I} & \\ & C \end{pmatrix}$$

As shown before this matrix is uniformly positive definite. This means that it is also strictly copositive.

8 Appendix

8.1 Omitted Proofs

In this section the proofs that were omitted in the main part of the thesis are given.

8.1.1 Eigenvalues of the Metric Matrix

Proposition 8.1 Let $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}$ and $\mathbb{I} \in \mathbb{R}^{n\times n}$ the identity matrix. Let λ_i^A, $i = 1, ..., n$ be the eigenvalues of the matrix A. Then the eigenvalues of the matrix $A + b\mathbb{I}$ are given by $\tilde{\lambda}_i := \lambda_i^A + b$ for all $i = 1, ..., n$.

Proof: Let v^i, $i = 1, ..., n$ be the corresponding eigenvectors to the eigenvalues $\lambda^A + i$. Then it follows that for $i = 1, ..., n$

$$(A + b\mathbb{I})v^i = Av^i + bv^i = (\lambda_i^A + b)v^i.$$

This means that v^i is an eigenvector of $A+b\mathbb{I}$ to the eigenvalue λ_i^A+b for all $i = 1, ..., n$. □

8.1.2 Proof of Proposition 6.3

Proof: We show that the scalar product $\langle x, y\rangle_{Q_k+\frac{1}{t_k}\mathbb{I}} := x^\top(Q_k + \frac{1}{t_k}\mathbb{I})y$ is well-defined. This yields directly that also the norm induced by the scalar product is well-defined (see for example [19, Corollary 12.6, p.172]).

By proposition 6.2 the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ is bounded and relation (6.8) assures that the following calculations are valid for all k.

We prove now that the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$ can be used to define a scalar product.

From the rules for matrix-vector multiplication follows that

$$(x + y)^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z = x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z + y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z, \quad x, y, z \in \mathbb{R}^n$$

and

$$x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) (y + z) = x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) y + x^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) z, \quad x, y, z \in \mathbb{R}^n.$$

Thus linearity of the defined scalar product is proven.

The symmetry of $Q_k + \frac{1}{t_k}\mathbb{I}$ yields symmetry of the scalar product by

$$x^\top \underbrace{\left(Q_k + \frac{1}{t_k}\mathbb{I}\right) y}_{:=\tilde{y}} = \tilde{y}^\top x = y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right)^\top x = y^\top \left(Q_k + \frac{1}{t_k}\mathbb{I}\right) x.$$

Finally positive definiteness of the scalar product follows directly from positive definiteness of the matrix $Q_k + \frac{1}{t_k}\mathbb{I}$.

This means the scalar product $\langle \cdot, \cdot \rangle_{Q_k + \frac{1}{t_k}\mathbb{I}}$ is well defined and thus induces the norm $\| \cdot \|_{Q_k + \frac{1}{t_k}\mathbb{I}}$. □

8.2 Additional Figures

8.2.1 Variable Metric Bundle Method

The following plots show the behavior in accuracy and number of steps of the proximal bundle algorithm 5.1 and different realizations of the variable metric bundle method 6.1 when optimizing the Ferrier polynomials f_1 to f_5 in different dimensions and for different noise forms. The conditions and parameters used for the plots are described in section 6.5.2

The two plots below depict the situation for $x \in \mathbb{R}^n$ for $n = 2, 3, ..., 15$.

Figure 12: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant gradient noise*

Figure 13: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing gradient noise*

The following plots show the situation for larger dimensions $n = \{20, 25, 30, 40, 50\}$.

Figure 14: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of no noise*

Figure 15: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant noise*

Figure 16: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing noise*

Figure 17: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of constant gradient noise*

Figure 18: *Comparison of accuracy and number of steps for the proximal bundle algorithm and the variable metric bundle algorithm in the case of vanishing gradient noise*

Next come the figures that illustrate the difference in behavior for different step size updating parameter κ_+ and the performance of the hybrid method. In this method the metric matrix is scaled for boundedness of the eigenvalues and then scaled again by $1/k$.

Figure 19: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for vanishing noise.*

Figure 20: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for constant gradient noise.*

Figure 21: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for vanishing gradient noise.*

Figure 22: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for the exact case for higher* x*-dimensions. The reached accuracy is depicted on the left, the needed number of steps on the right.*

Figure 23: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for constant noise.*

Figure 24: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for vanishing noise.*

Figure 25: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for constant gradient noise.*

Figure 26: *Influence of the step size updating parameter* $\kappa_+ = 1.2$ *and* $\kappa_+ = 2$ *and performance of the hybrid method for vanishing gradient noise.*

Rechtschreibfehler, Namen, Stil überprüfen

References

[1] P. Apkarian, D. Noll, and O. Prot. A trust region spectral bundle method for nonconvex eigenvalue optimization. *SIAM J. Optim.*, 19(1):281–306, jan 2008.

[2] A. Bagirov, N. Karmitsa, and M. M. Mäkelä. *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer International Publishing Switzerland, 2014.

[3] F. H. Clarke. *Optimization and nonsmooth analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics Philadelphia, 1990.

[4] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, Sep 2007.

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

[6] W. de Oliveira and C. Sagastizábal. Bundle methods in the XXIst century: A bird's-eye view. *Pesquisa Operacional*, 34(3):647–670, dec 2014.

[7] A. Fuduli, M. Gaudioso, and G. Giallombardo. A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optim. Method. Softw.*, 19(1):89–102, 2004.

[8] A. Fuduli, M. Gaudioso, and G. Giallombardo. Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. *SIAM J. Optim*, 14(3):743–756, 2004.

[9] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer-Lehrbuch Masterclass. Springer Berlin Heidelberg, 2002.

[10] S. R. Gunn. Support vector machines for classification and regression. Technical report, Faculty of Enigneering, Science and Mathematics, School of Electronics and Computer Science, University of Southampton, 1998.

[11] N. Haarala, K. Miettinen, and M. M. Mäkelä. Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Math. Program.*, 109(1):181–205, 2007.

[12] W. Hare and C. Sagastizàbal. A redistributed proximal bundle method for nonconvex optimization. *SIAM J. Optim.*, 20(5):2442–2473, 2010.

[13] W. Hare, C. Sagastizàbal, and M. Solodov. A proximal bundle method for nonsmooth nonconvex functions with inexact information. *Computational Optimization and Applications*, 63:1–28, 2016.

[14] J. Heinonen. Lectures on Lipschitz analysis. Lectures at the 14th Jyväskylä Summer School in August 2004, 2004.

[15] M. Hintermüller. A proximal bundle method based on approximate subgradients. *Computational Optimization and Applications*, 20:245–266, 2001.

[16] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*, volume 306 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg, 1993.

[17] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*, volume 305 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg, 2 edition, 1996.

[18] A. Jofré, D. T. Luc, and M. Théra. ε-subdifferential and ε-monotonicity. *Nonlinear Analysis: Theory, Methods & Applications*, 33(1):71–90, jul 1998.

[19] V. M. Jörg Liesen. *Linear Algebra*. Springer International Publishing, 2015.

[20] K. C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization*. Springer, 1985.

[21] K. C. Kiwiel. An aggregate subgradient method for nonsmooth and nonconvex minimization. *J. Comput. Appl. Math.*, 14(3):391–400, 1986.

[22] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim*, 16(4):1007–1023, jan 2006.

[23] K. C. Kiwiel. Bundle methods for convex minimization with partially inexact oracles. Technical report, Systems Research Institute, Polish Academy of Sciences, 2010.

[24] K. Königsberger. *Analysis 2*. Springer Berlin Heidelberg, 2002.

[25] K. Königsberger. *Analysis 1*. Springer-Verlag GmbH, 2003.

[26] G. Kunapuli. *A bilevel optimization approach to machine learning*. PhD thesis, Rensselaer Polytechnic Institute Troy, New York, 2008.

[27] C. Lemaréchal. Nonsmooth optimization and descent methods. IIASA research report, International Institute for Applied Systems Analysis, 1978.

[28] C. Lemaréchal and C. Sagastizábal. *An approach to variable metric bundle methods*, pages 144–162. Springer Berlin Heidelberg, 1994.

[29] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: From conceptual to implementable forms. *Math. Program.*, 76(3):393–410, 1997.

[30] A. S. Lewis and M. L. Overton. Nonsmooth optimization via BFGS. *Submitted to SIAM Journal on Optimization*, 2008.

[31] A. S. Lewis and S. J. Wright. A proximal method for composite minimization. *Math. Program.*, 158(1-2):501–546, aug 2015.

[32] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *Journal of Optimization Theory and Applications*, 102(3):593–613, sep 1999.

[33] R. Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. In *Mathematical Programming Studies*, volume 17, pages 77–90. Springer Nature, 1982.

[34] R. Mifflin and C. Sagastizàbal. A science fiction story in nonsmooth optimization originating at IIASA. *Documenta Mathematica*, Extra Volume ISMP:291–300, 2012.

[35] G. Moore, C. Bergeron, and K. P. Bennett. Gradient-type methods for primal SVM model selection. Technical report, Rensselaer Polytechnic Institute, 2010.

[36] G. Moore, C. Bergeron, and K. P. Bennett. Model selection for primal SVM. *Machine Learning*, 85(1):175–208, 2011.

[37] Y. Nesterov and V. Shikhman. Algorithmic principle of least revenue for finding market equilibria. In B. Goldengorin, editor, *Optimization and Its Applications in Control and Data Sciences*, volume 115 of *Springer Optimization and Its Applications*, pages 381–435. Springer Nature, 2016.

[38] J. Nocedal. Updating quasi-newton matrices with limited storage. *Math. Comput.*, 35(151):773–782, 1980.

[39] D. Noll. Cutting plane oracles to minimize non-smooth non-convex functions. *Set-Valued and Variational Analysis*, 18(3-4):531–568, sep 2010.

[40] D. Noll. Bundle method for non-convex minimization with inexact subgradients and function values. In *Computational and Analytical Mathematics*, pages 555–592. Springer Nature, 2013.

[41] D. Noll and P. Apkarian. Spectral bundle method for non-convex maximum eigenvalue functions: first-order methods. *Math. Program.*, 104(2-3):701–727, jul 2005.

[42] D. Noll, O. Prot, and A. Rondepierre. A proximity control algorithm to minimize non-smooth and non-convex functions. *Pacific Journal of Optimization*, 4(3):571–604, 2012.

[43] J. Outrata, M. Kočvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*. Springer US, 1998.

[44] B. T. Polyak. *Introduction to Optimization*. Optimization Software , Inc., Publications Division, New York, 1987.

[45] R. Rockafellar. Extensions of subgradient calculus with applications to optimization. *Nonlinear Analysis: Theory, Methods & Applications*, 9(7):665–698, jul 1985.

[46] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[47] R. T. Rockafellar and R. J. B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg, 3rd edition, 2009.

[48] C. R. J. Roger A. Horn. *Matrix Analysis*. Cambridge University Press, 2012.

[49] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM J. Optim*, 2(1):121–152, feb 1992.

[50] A. J. Smola, S. V. N. Vishwanathan, and Q. V. Le. Bundle methods for machine learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1377–1384, USA, 2007. Curran Associates Inc.

[51] M. V. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119(1):151–165, 2003.

[52] M. V. Solodov. *Constraint Qualifications*. Wiley Encyclopedia of Operations Research and Management Science, 2011.

[53] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I*, volume 163 of *Die Grundlehren der mathematischen Wissenschaften*. Springer Berlin Heidelberg, 1970.

[54] C. H. Teo, A. J. Smola, S. Vishwanathan, and Q. V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

[55] J. S. Treiman. Clarke's gradients and ε-subgradients in Banach spaces. *Transactions of the American Mathematical Society*, 294(1):65–65, jan 1986.

[56] M. Ulbrich and S. Ulbrich. *Nichtlineare Optimierung*. Springer Basel AG, 2012.

[57] V. N. Vapnik. *Statistical Learning Theory*. JOHN WILEY & SONS INC, 1998.

[58] V. N. Vapnik. An overview of statistical learning theory. *IEEE Trans. Neural Networks*, 10(5):988–999, 1999.

[59] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer New York, 2013.

[60] J. Vlček and L. Lukšan. Globally convergent variable metric bundle method for nonconvex nondifferentiable unconstrained minimization. *Journal of Optimization Theory and Applications*, 111(2):407–430, 2001.

[61] C. S. Warren Hare. Computing proximal points of nonconvex functions. *Math. Program.*, 116:221–258, 2009.

[62] C. L. Welington de Oliveira, Claudia Sagastizàbal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Math. Program.*, 148:241–277, 2014.