

Numerical Optimization

Summer 2022

Prof. Dr. Volker Schulz
Trier University

Recommended textbooks:

- [Bec14] Amir Beck. Introduction to Nonlinear Optimization - Theory, Algorithms, and Applications with MATLAB, volume 19 of MOS-SIAM Series on Optimization. SIAM, 2014. <http://bookstore.siam.org/mo19/>.
- [BV04] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University Press, 2004. <https://web.stanford.edu/~boyd/cvxbook/>.
- [NW06] J. Nocedal and S Wright. Numerical optimization. Springer, 2006.

1 Introduction

The goal of optimization is to make the optimal choice from among a number of alternative possibilities.

- The “number” is large, often infinite.
- Need a mathematical definition of “best”: objective function.
- Need a mathematical way to define the “choices”: variables.
- Need a way to describe restrictions on choices: constraints.

Variables, Objective, Constraints are the key ingredients of all optimization problems. The formulation of an application as a mathematical optimization problem is often nontrivial. It requires the interaction with applications experts, “domain scientists.”

The field of Optimization conventionally encompasses

- analyzing fundamental properties of the mathematical formulation;
- devising algorithms to solve optimization formulations, and analyzing the mathematical properties of these algorithms (convergence, complexity, efficiency);
- also implementation and testing of the algorithms.

Definition 1.1. (Notation)

a) Let X be a set and $f : X \rightarrow \mathbb{R}$ a function defined on X . We define the notation:

$$\hat{x} \in X \text{ solves } \min_{x \in X} f(x) \Leftrightarrow \hat{x} \in X \text{ solves } \min_x f(x) \text{ such that } x \in X$$

$$\Leftrightarrow \hat{x} = \arg \min_{x \in X} f(x)$$

$$\Leftrightarrow f(\hat{x}) \leq f(x), \forall x \in X$$

b) If $X \subset \mathbb{R}^n$ is described by the equation $c(x) = 0$ with $c : \mathbb{R}^n \rightarrow \mathbb{R}^{n_e}$ ($n_e \leq n$) and inequalities $d_i(x) \leq 0 \ \forall i = 1, \dots, n_i$ in the form $X = \{x | c(x) = 0, d_i(x) \leq 0 \ \forall i = 1, \dots, n_i\}$, then we write

$$\begin{aligned} \hat{x} \text{ solves } & \min_x f(x) \\ \text{s.t. } & c(x) = 0 \\ & d_i(x) \leq 0 \ \forall i = 1, \dots, n_i \end{aligned} \Leftrightarrow f(\hat{x}) \leq f(x) \quad \forall x \in X$$

Definition 1.2. (local solution)

\hat{x} is called a (strictly) local solution of the problem $\min_{x \in X} f(x)$, if there exists an open neighborhood U with

$$\begin{aligned} \hat{x} \in U \text{ and } f(\hat{x}) &\leq f(x) \quad \forall x \in U \\ (f(\hat{x}) < f(x)) \quad &\forall x \in U \setminus \{\hat{x}\} \end{aligned}$$

Definition 1.3. (*derivative vs. gradient*)

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is differentiable at the point $a \in \mathbb{R}^n$, if there exists a linear mapping $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and a function $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$, which is continuous at a and there holds

$$f(x) = f(a) + T(x - a) + r(x)\|x - a\|, \quad r(a) = 0$$

The mapping T is then called the **derivative** of f at the point a and we use the notation $T =: Df(a) =: \frac{\partial f}{\partial x}(a)$. The linear mapping T can be represented as a matrix $\mathbb{R}^{m \times n}$. If this representation is done with respect to the unit bases in \mathbb{R}^n and \mathbb{R}^m , it is called the **Jacobian**. In the special case $m = 1$, the Jacobian is just a row vector, i.e.,

$$\frac{\partial f}{\partial x}(a) = \left[\frac{\partial f}{\partial x_1}(a) \dots \frac{\partial f}{\partial x_n}(a) \right]$$

If we pick a specific scalar product in $(\cdot, \cdot)_A$, which is represented by a symmetric and positive definite matrix $A \in \mathbb{R}^{n \times n}$ such that $(x, y)_A = x^\top A y$ for all $x, y \in \mathbb{R}^n$, there exists a unique vector $g \in \mathbb{R}^n$, which represents the derivative (due to the Riesz representation theorem) in the following fashion

$$\frac{\partial f}{\partial x}(a)v = (g, v)_A, \quad \forall v \in \mathbb{R}^n$$

This makes g a Riesz representation of $\frac{\partial f}{\partial x}(a)$. We call this vector g the **gradient** of f and denote it as

$$\nabla f := g$$

If the scalar product chosen is not explicitly specified, the standard scalar product $(x, y)_2 := x^\top y$ is assumed.

Remarks:

- The definition of the derivative can be easily carried over to Banach spaces. Then, it is more precisely called the Fréchet derivative.
- If the Banach space carries a scalar product and is complete in the related norm (i.e., it is a Hilbert space), then we define analogously the gradient of a scalar valued function.
- As additional insight, we need there the fact that the derivative of a scalar valued function is an element of the dual space, where the Riesz representation theorem can be used to define the gradient.

Examples:



Attention: the gradient is only defined, after a scalar product is chosen and the gradient depends on the scalar product (in contrast to the derivative).

Remark: This definition of the gradient is completely consistent with the definition of the differential operator with the same name in the context of partial differential equations:

$$\text{grad} = \nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)^\top$$

which is thought in the standard scalar product of \mathbb{R}^3 .

We recall some concepts from manifolds:

Definition 1.4. Consider a function $g \in C^1(X, \mathbb{R}^m)$, where $X \subset \mathbb{R}^n$ and $m < n$. The point $x \in X$ is called a regular point, if the derivative $Dg(x)$ is surjective, or equivalently has full rank.

Remark: The property of $Dg(x)$ being surjective is generalizable to general Banach spaces, whereas derivatives in function spaces do not have finite rank. The fact that surjectivity of a matrix is equivalent to full rank, if $m < n$, is checked in the homework (\rightarrow exercise).

Definition 1.5. We consider the set $M := \{x \in X \mid g(x) = 0\}$ with assumptions as in definition 1.4. If $a \in M$ is a regular point, then M is a differentiable manifold in a local neighbourhood around a and we define the tangent space in a as

$$T_a M := \{v \in X \mid v = \dot{\gamma}(0), \gamma : (-\varepsilon, \varepsilon) \rightarrow M \text{ differentiable curve with } \gamma(0) = a\}$$

Here, we can give a more operational characterization of the tangent space:

Theorem 1.6.

$$T_a M = \ker (Dg(a))$$

Proof:

$$N(a) = \{x \mid f(x) = f(a)\}$$

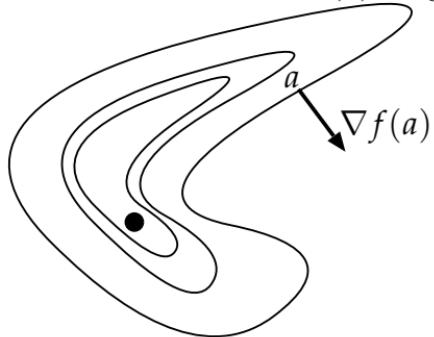


Figure 1: Visualization of the gradient

Now, we can derive the following important geometric insight:

Theorem 1.7. *We choose a scalar product $(\cdot, \cdot)_A$ and define with this the gradient $\nabla f(a)$ of a function $f \in C^1(X, \mathbb{R})$ at the point $a \in X$. Then, this gradient is orthogonal (with respect to the same scalar product) to the level surface*

$$N(a) := \{x \in X \mid f(x) = f(a)\}$$

Proof: The proposition means more accurately $\nabla f(a) \perp T_a N(a)$.

□

Remark: Both, theorem and proof can be verbatim carried over to general Hilbert spaces.

Now, second derivatives:

We recall the inductive definition

$$\begin{aligned} D^2f(a)(h_1, h_2) &:= D(Df(a)h_1)h_2 = \\ &= D\left(\frac{d}{dt_1}\Big|_{t_1=0} f(a + t_1 h_1)\right)h_2 = \\ &= \frac{d}{dt_2}\Big|_{t_2=0} \frac{d}{dt_1}\Big|_{t_1=0} f(a + t_1 h_1 + t_2 h_2) \end{aligned}$$

from which we get immediately symmetry with respect to h_1 and h_2 .

Definition 1.8. (Hessian matrix) For $f \in C^2(X, \mathbb{R})$, we define the (symmetric) Hessian matrix $\text{Hess } f$ by

$$(\text{Hess } f(a)v, w) := D^2f(a)(v, w) \quad \forall v, w \in \mathbb{R}^n$$

We denote also $\nabla^2 f := \text{Hess } f : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Remark: This definition is identical in Hilbert space, where we have a (symmetric) Hessian operator rather than a Hessian matrix.

Examples:

2 Optimality conditions

2.1 Unbounded optimization

In general, we assume now: $f \in C^2$ and investigate the problem class

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

Theorem 2.1. *(necessary condition) Let \hat{x} be a solution of the unbounded minimization problem above. Then*

- a) $\nabla f(\hat{x}) = 0$ (nec. cond. of 1st order)
- b) $\nabla^2 f(\hat{x})$ is positiv semidefinit (nec. cond. of 2nd order), i.e.

$$(v, \nabla^2 f(\hat{x}) v) \geq 0 \quad \forall v \in \mathbb{R}^n$$

Proof:

Theorem 2.2. (*sufficient conditions*)

Let $f \in C^2$ and $\hat{x} \in X \subset \mathbb{R}^n$ may satisfy the conditions

a) $\nabla f(\hat{x}) = 0$

b) $\nabla^2 f(\hat{x})$ ist coercive, i.e., $\exists \alpha > 0$ with $(v, \nabla^2 f(\hat{x})v) \geq \alpha \cdot \|v\|^2$, $\forall v \in X$

Then, \hat{x} is a strictly local minimum.

Proof: We show: for x sufficiently close to \hat{x} that we have

$$f(x) \geq f(\hat{x}) + \frac{\alpha}{4} \|x - \hat{x}\|^2 \quad (\Rightarrow \hat{x} \text{ is strictly local minimum})$$

Because $f \in C^2$ we can find a neighbourhood $U \ni \hat{x}$ such that

$$\|\nabla^2 f(\hat{x}) - \nabla^2 f(x)\| \leq \frac{\alpha}{2}, \quad \forall x \in U$$

Define $h := x - \hat{x}$. Then, we obtain for $\varphi(t) := f(\hat{x} + th)$ the fact

$$\begin{aligned}f(x) - f(\hat{x}) &= \varphi(1) - \varphi(0) \stackrel{\text{Taylor}}{=} \varphi'(0) + \frac{1}{2}\varphi''(\tau), \quad \tau \in [0, 1] \\&\stackrel{a)}{=} 0 + \frac{1}{2}(h, \nabla^2 f(\hat{x} + \tau h)h) \\&= \frac{1}{2}(h, \nabla^2 f(\hat{x})h) - \frac{1}{2}(h, [\nabla^2 f(\hat{x}) - \nabla^2 f(\hat{x} + \tau h)]h) \\&\geq \frac{\alpha}{2}\|h\|^2 - \frac{1}{2}\frac{\alpha}{2}\|h\|^2 \geq \frac{\alpha}{4}\|h\|^2\end{aligned}$$

□

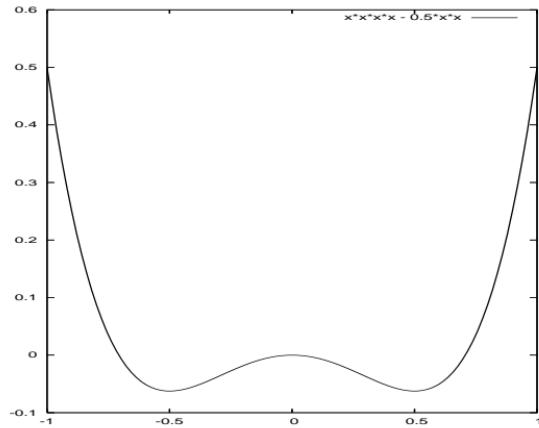
Remark: Both, theorem and proof are valid verbatim also in general Hilbert spaces. In the finite dimensional space \mathbb{R}^n , we observe that coercivity is equivalent to positive definiteness with $\alpha := \min \sigma(\nabla^2 f(\hat{x})) > 0$.

Examples:

a) $f(x) = x^2$

b) $f(x) = x^4$

Consider small perturbation $\tilde{f}(x; \varepsilon) := x^4 - 2\varepsilon x^2$



Theorem 2.3 (Stability). Let $X := \mathbb{R}^n$ and $E := \mathbb{R}^m$. Furthermore, we assume that $f : X \times E \rightarrow \mathbb{R}$ is twice differentiable. Additionally, we assume that the sufficient optimality conditions are satisfied at $x^0 \in X$ for the problem $\min_x f(x, 0)$. Then, there is a neighbourhood U with $0 \in U \subset E$, such that

$$\hat{x}(\varepsilon) := \arg \min_x f(x, \varepsilon)$$

exists uniquely for all $\varepsilon \in U$. Furthermore $\hat{x}(\varepsilon)$ is differentiable in U with derivative

$$D_\varepsilon \hat{x}(\varepsilon) = -D_x^2 f(\hat{x}(\varepsilon), \varepsilon)^{-1} D_\varepsilon D_x f(\hat{x}(\varepsilon), \varepsilon)$$

or in short notation

$$\hat{x}_\varepsilon = -f_{xx}^{-1} f_{x\varepsilon}$$

Proof: The proof for the locally unique existence is hard to find in literature, which is the reason, why it is printed in the script. Here only the formula is shown.

Remarks:

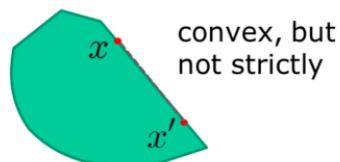
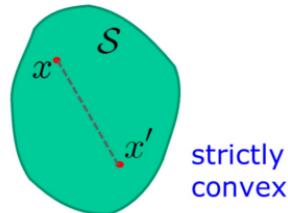
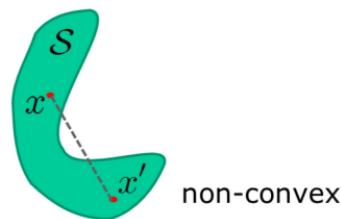
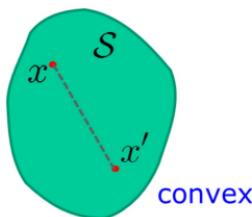
- The theorem is also valid in function spaces. The proof has to be appropriately modified relying then on the spectral theorem rather than the matrix of eigen vectors.
- Summarizing, the theorem means that sufficient conditions guarantee stability of the optimal solution. Numerical solvers need this stability severely, which is the reason, why we will often require the validity of sufficient conditions below.
- The formula for the derivative of $\hat{x}(\varepsilon)$ is of high importance in online optimization. There, one wants to adapt an expensive offline solution to small perturbations, avoiding a complete and again expensive recomputation. The expression $\hat{x}_\varepsilon \cdot \varepsilon$ gives an increment leading to a point, which is the correct solution ($\hat{x}(\varepsilon) \doteq \hat{x}_0 + \hat{x}_\varepsilon \cdot \varepsilon$) in first order approximation and needs only the information \hat{x}_ε , which can be computed before the exact value of ε is known.

2.2 Convex optimization

Definition 2.4 (Convex set). *For $x_1, x_2 \in \mathbb{R}^n$, we define the set*

$$[x_1, x_2] := \{\lambda x_1 + (1 - \lambda)x_2 \mid \lambda \in [0, 1] \subset \mathbb{R}\}$$

A set $S \subset \mathbb{R}^n$ is called convex, if for all $x_1, x_2 \in S$ holds $[x_1, x_2] \subset S$. The set is called strictly convex, if for all $x_1, x_2 \in S$ holds $[x_1, x_2] \setminus \{x_1, x_2\} \subset \text{int}(S)$, where $\text{int}(S)$ means the interior of S .

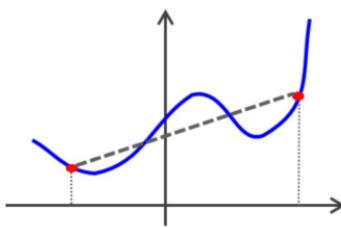


Definition 2.5 (Convex function). Let $S \subset \mathbb{R}^n$ be a convex subset. We say that the function $f : S \rightarrow \mathbb{R}$ is convex, if

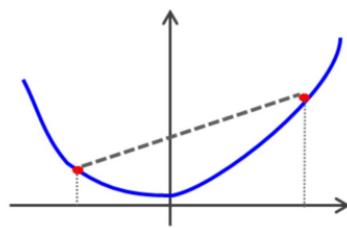
$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2), \forall x_1, x_2 \in S, \forall \lambda \in [0, 1]$$

the function f is called strictly convex, if

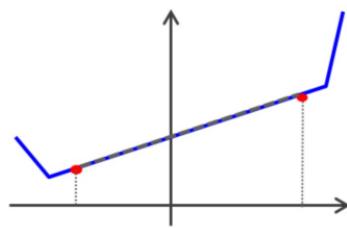
$$f(\lambda x_1 + (1 - \lambda)x_2) < \lambda f(x_1) + (1 - \lambda)f(x_2), \forall x_1 \neq x_2 \in S, \forall \lambda \in (0, 1)$$



non-convex



strictly convex



convex, not strictly

Examples for convex functions are affine function, norms (exercise). It can be shown that level sets of convex functions are convex sets (exercise).

Theorem 2.6 (Jensen's inequality, 1905). *Let $f : S \rightarrow \mathbb{R}$ be a convex function defined on a convex domain $S \subset \mathbb{R}^n$. Then, the so-called Jensen's inequality holds for convex combinations of arbitrary vectors*

$$f\left(\sum_{k=1}^m \mu_k x_k\right) \leq \sum_{k=1}^m \mu_k f(x_k), \quad \mu_k \in [0, 1], \forall k \text{ and } \sum_{k=1}^m \mu_k = 1$$

Proof: By induction.

Remark: Generalization in probability theory: Let $(\Omega, \mathfrak{F}, P)$ be a probability space, X an integrable real-valued random variable and f a convex function. Then:

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$$

Definition 2.7 (Convex Hull and Extreme Points). 1. Let $E \subset \mathbb{R}^n$ be a set. The convex hull of E is defined by

$$\text{conv}(E) := \bigcap \{C \subset \mathbb{R}^n \mid C \text{ convex}, E \subset C\}.$$

2. Let $S \subset \mathbb{R}^n$ be convex. A point $x \in S$ is called extreme point of S if there do not exist $x_1, x_2 \in S$ ($x_1 \neq x_2$) and $\lambda \in (0, 1)$, such that $x = (1 - \lambda)x_1 + \lambda x_2$. We denote the set of extreme points of S by $\text{ext}(S)$.

Theorem 2.8 (Krein-Milman). Let $S \subset \mathbb{R}^n$ be a compact convex set. Then

$$S = \text{conv}(\text{ext}(S)).$$

Proof. One needs the axiom of choice. Details can be found in the original work [KM40]. \square

Theorem 2.9 (Continuity). *A convex function $f : S \rightarrow \mathbb{R}$ defined on a convex subset $S \subset \mathbb{R}^n$ is (locally Lipschitz) continuous in $\text{int}(S)$, i.e. there exist $\varepsilon > 0$ and $L > 0$ such that $B_\varepsilon^2(x_0) \subset S$ and*

$$|f(x) - f(x_0)| \leq L\|x - x_0\|$$

for all $x \in B_\varepsilon^2(x_0)$.

Proof: The proof uses Krein-Milman theorem and Jensen's inequality and can be found in the script.

□

Remarks: In addition, we have even higher regularity properties for convex functions:

- Since convex functions are locally Lipschitz, it follows from Rademacher's (1892-1969) theorem that they are almost everywhere differentiable, i.e., the points, where they are not differentiable, form a set with Lebesgue measure zero.
- Alexandrov's (1939) theorem even states that a convex function is almost everywhere twice differentiable, i.e., the set of points, where it is not twice differentiable is of Lebesgue measure zero. (cf. [Roc70, theorem 25.5])

Classical concepts in convex optimization:

Theorem 2.10 (Linear subapproximation). *Let the function $f : S \rightarrow \mathbb{R}$ be convex and differentiable on the convex set S , then the gradient ∇f gives a lower affine approximation in the following sense*

$$f(y) \geq f(x) + (\nabla f(x), (y - x)), \quad \forall x, y \in S$$

Proof:

Theorem 2.11 (Convex optimality conditions). *Let the function $f : S \rightarrow \mathbb{R}$ be convex and differentiable on the convex set S . The point \hat{x} is a minimizer of f , if and only if*

$$(\nabla f(\hat{x}), (x - \hat{x})) \geq 0, \quad \forall x \in S$$

Proof:

Definition 2.12 (Subgradient). Let the function $f : S \rightarrow \mathbb{R}$ be convex and defined on the convex set $S \subset \mathbb{R}^n$. A vector $v \in \mathbb{R}^n$ is called a subgradient of f at $x_0 \in S$, if

$$f(y) \geq f(x_0) + (v, (y - x_0)), \quad \forall y \in S$$

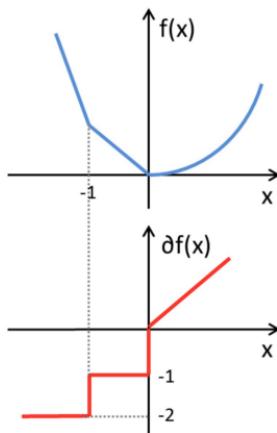
The set of all vectors v with this property is called subdifferential and notated as

$$\partial f(x_0) := \{v \in \mathbb{R}^n \mid f(y) \geq f(x_0) + (v, (y - x_0)), \forall y \in S\}$$

Remark: Note that differentiable functions yield $\partial f(x) = \{\nabla f(x)\}$.

$$f(x) = \begin{cases} -2x - 1, & x \leq -1 \\ -x, & -1 < x \leq 0 \\ x^2/2, & x > 0 \end{cases}$$

$$\partial f(x) = \begin{cases} \{-2\}, & x < -1 \\ [-2, -1], & x = -1 \\ \{-1\}, & -1 < x < 0 \\ [-1, 0], & x = 0 \\ \{x\}, & x > 0 \end{cases}$$



Theorem 2.13 (Fermat's rule). *For convex functions $f : S \rightarrow \mathbb{R}$, the following equivalence holds*

$$\hat{x} = \arg \min_{x \in S} f(x) \Leftrightarrow 0 \in \partial f(\hat{x})$$

Proof:

Remark: Thus, it is enough to look at first order conditions in convex optimization and all local minima are also global minima.

Theorem 2.14 (Monotone gradient condition). *For convex functions $f \in C^1(S, \mathbb{R})$ holds*

$$(\nabla f(x) - \nabla f(y), x - y) \geq 0 \quad \forall x, y \in S$$

Proof: use linear subapproximation twice:

□

Definition 2.15 (Strong Convexity). *A function $f \in C^1(S, \mathbb{R})$ with S a convex set is called strongly convex, if for all $x, y \in S$ holds*

$$(\nabla f(x) - \nabla f(y), x - y) \geq m \|x - y\|^2, \text{ for some } m > 0$$

The parameter m is called the modulus of convexity.

Theorem 2.16. *The strong convexity of definition 2.15 is equivalent to (for all $x, y \in S$)*

$$f(y) \geq f(x) + (\nabla f(x), y - x) + \frac{m}{2} \|x - y\|^2, \text{ for the same } m > 0$$

If f is even twice differentiable, strong convexity is equivalent to coercivity of the Hessian for all $x \in S$, i.e.,

$$(\text{Hess } f(x)v, v) \geq m \|v\|^2, \forall v \in \mathbb{R}^n$$

Proof: Idea: show that the function $g(x) := f(x) - \frac{m}{2} \|x - y\|^2$ is convex and use monotone gradient condition (Th. 2.14) for g , rest exercise. The idea for the Hessian is for $z := x + \lambda(y - x)$:

$$\begin{aligned} (\nabla f(z) - \nabla f(x), z - x) &\geq m \|z - y\|^2 \\ &\Rightarrow \left(\frac{\nabla f(x + \lambda(y - x)) - \nabla f(x)}{\lambda}, \lambda(y - x) \right) \geq \frac{m}{\lambda} \|\lambda(y - x)\|^2 \\ &\Rightarrow \left(\frac{\nabla f(x + \lambda(y - x)) - \nabla f(x)}{\lambda}, y - x \right) \geq m \|x - y\|^2 \end{aligned}$$

and then $\lim_{\lambda \rightarrow 0}$. The other direction is shown by a Taylor series argument. □

2.3 Optimization with equality constraints

We consider optimization problems of the following type:

$$\begin{aligned} & \min_x f(x) \\ \text{s.t. } & c(x) = 0 \end{aligned}$$

We recall from linear Algebra:

Lemma 2.17. *The linear operator $C \in \mathbb{R}^{m \times n}$, satisfies*

$$\ker(C)^\perp = \text{im}(C^{\text{ad}})$$

Proof:

Idea: $\ker(C) = \text{im}(C^{\text{ad}})^\perp$. Rest: Exercise.

□

Remark: The Lemma holds also if $C \in \text{Hom}(X, Y)$ for X, Y Hilbert spaces, with exactly the same arguments.

Theorem 2.18. (*Necessary optimality conditions of first and second order*)

Consider the optimization problem

$$\begin{aligned} \min f(x) \quad & f \in C^2(\mathbb{R}^n, \mathbb{R}) \\ \text{s.t. } c(x) = 0 \quad & c \in C^2(\mathbb{R}^n, \mathbb{R}^m); \end{aligned}$$

We assume that \hat{x} is a regular point in the sense of definition 1.4 w.r.t. c and that it is a local solution of the optimization problem. Then, there are adjoint variables $\lambda \in \mathbb{R}^m$, such that the Lagrangian function

$$\mathcal{L}(x, \lambda) := f(x) + (\lambda, c(x))_{\mathbb{R}^m}$$

satisfies

- a) $\nabla_x \mathcal{L}(\hat{x}, \lambda) = 0 \quad (\text{Karush-Kuhn-Tucker condition})$
- b) $(v, \text{Hess}_x \mathcal{L}(\hat{x}, \lambda)v) \geq 0 \quad \forall v \in \ker Dc(\hat{x})$

Proof:

Remark: The theorem can be generalized to the case $c \in C^2(X, Y)$, where X, Y are Hilbert spaces (with the same arguments) or even Banach spaces (c.f., Zeidler: Applied functional analysis, main principles and their application, Springer 1995).

Theorem 2.19 (Sufficient condition). *We consider the optimization problem formulated in theorem 2.18. If \hat{x} is a regular point with $c(\hat{x}) = 0$ and if we have an $\alpha > 0$ such that:*

- a) $\nabla \mathcal{L}(\hat{x}, \lambda) = 0 \quad \text{for a } \lambda \in \mathbb{R}^m$
- b) $(v, \text{Hess}_x \mathcal{L}(\hat{x}, \lambda)v) \geq \alpha \|v\|^2, \forall v \in \ker Dc(\hat{x})$

Then \hat{x} is strict local minimum.

Proof: by contradiction: We assume \hat{x} is not a strict local minimum.

\Rightarrow Within each neighborhood $U \exists \{x_k\} \in M := \{x \in X | c(x) = 0\} \cap U, x_k \neq \hat{x}$ and $x_k \rightarrow \hat{x}$, but $f(x_k) \leq f(\hat{x})$

$$\text{Define } s_k := \frac{x_k - \hat{x}}{\|x_k - \hat{x}\|}, \quad \delta_k := \|x_k - \hat{x}\|$$

The series $\{s_k\}$ stays within the unit ball and is thus bounded. In $X = \mathbb{R}^n$, the unit ball is compact (Note: this argument is only valid in finite dimensions!). Therefore, there is a convergent subseries $s_{k'} \rightarrow \hat{s}$ with $\|\hat{s}\| = 1$.

Now, we proof that \hat{s} does not satisfy proposition b).

(1) $\hat{s} \in \ker Dc(\hat{x})$, since we conclude with the Taylor series

$$(2) \quad \begin{aligned} 0 &= c_i(x_{k'}) - c_i(\hat{x}) = Dc_i(\hat{x})\delta_{k'} s_{k'} + \frac{\delta_{k'}^2}{2} D^2 c_i(\tilde{x}_{k,i})(s_{k'}, s_{k'}) \\ 0 &= Dc_i(\hat{x})s_{k'} + \frac{\delta_{k'}^2}{2} D^2 c_i(\tilde{x}_{k,i})(s_{k'}, s_{k'}) \\ \downarrow &\qquad\qquad\qquad \downarrow \\ Dc_i(\hat{x})\hat{s} &\qquad\qquad\qquad 0 \qquad\qquad\qquad \forall i = 1, \dots, \dim(Y) \end{aligned}$$

For $x_{k'}$ we have

$$0 \geq f(x_{k'}) - f(\hat{x}) = (\nabla f(\hat{x}), \delta_{k'} s_{k'}) + \frac{\delta_{k'}^2}{2} (s_{k'}, \text{Hess } f(\tilde{x}_{k'}) s_{k'})$$

we multiply (2) with λ_i and obtain after summation:

$$\begin{aligned} 0 &\geq \delta_{k'} \left(\underbrace{\nabla f(\hat{x}) + \sum_{i=1}^n \lambda_i \nabla c_i(\hat{x}), s_k}_{= \nabla \mathcal{L}(\hat{x}, \lambda) = 0} \right) + \frac{\delta_{k'}^2}{2} \left(\underbrace{s_{k'}, [\text{Hess } f(\tilde{x}_{k'}) + \sum_{i=1}^n \lambda_i \text{Hess } c_i(\tilde{x}_{k',i})] s_{k'}}_{\downarrow (\hat{s}, \text{Hess } \mathcal{L}(\hat{x}, \lambda) \hat{s})} \right) \end{aligned}$$

$\Rightarrow \exists \hat{s}$ which contradicts b)

Remarks

- The sufficient conditions guarantee also in the equality constrained case local stability of the optimal solution with respect to small perturbations of the data.
- For a guarantee of local stability in infinite dimensional spaces, we need additionally a stronger constraint qualification than just surjectivity of $Dc(\hat{x})$. For instance, we can require $\|Dc(\hat{x})v\| \geq \beta\|v\|$, $\forall v \in \ker(Dc(\hat{x}))^\perp$ for some $\beta > 0$. (This is equivalent to the LBB condition for the Stokes flow equation, which is discussed in the lecture on numerical methods for differential equations.(LBB)

Corollary 2.20. *If the sufficient optimality conditions are satisfied at \hat{x} , then the pair $(\hat{x}, \hat{\lambda})$ is the locally unique solution of the system of equations*

$$\boxed{\begin{aligned}\nabla_x \mathcal{L}(\hat{x}, \hat{\lambda}) &= 0 \\ c(\hat{x}) &= 0\end{aligned}}$$

Proof:

Theorem 2.21. (λ as shadow price)

Assume $f \in C^1(\mathbb{R}^n, \mathbb{R})$ and $c \in C^1(\mathbb{R}^n, \mathbb{R}^m)$. For the optimization problem

$$\begin{aligned} & \min f(x) \\ \text{unter } & c(x) = \alpha \quad , \quad \alpha \in Y \end{aligned}$$

denote $\hat{x} = x(0)$ the solution for $\alpha = 0$ and $\hat{\lambda}$ the corresponding adjoint variables.

Then, $\nabla_\alpha f(x(0)) = -\hat{\lambda}$.

Proof:

Important special case: Separable Problems : $X = Z \times P$

$$\begin{aligned} & \min f(z, p) \\ \text{s.t. } & c(z, p) = 0, \frac{\partial c}{\partial z} \text{ invertible} \end{aligned}$$

Thus, we obtain the following system of equations characterizing the optimal solution:

$$\nabla_z f(z, p) + c_z^{ad}(z, p)\lambda = 0 \quad (\text{adjoint equation, uniquely solvable for } \lambda)$$

$$\nabla_p f(z, p) + c_p^{ad}(z, p)\lambda = 0 \quad (\text{optimality condition., not necessarily solvable for } p)$$

$$c(z, p) = 0 \quad (\text{state equation, uniquely solvable for } z)$$

Remark: An algorithmic challenge lies in the construction of corresponding adjoint solution iterations, which exploit primal solution algorithms. (\rightarrow automatic differentiation, AD).

For the conditions of second order, we use and define the reduced Hessian, based on an explicit representation of the null space of the constraints.

$$\Rightarrow \ker Dc(z, p) = \left\{ \begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix} \beta \mid \beta \in P \right\} \ni v$$

$$\Rightarrow (v, \text{Hess}_x \mathcal{L}(\hat{x}, \lambda)v) = \left(\begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix} \beta, \text{Hess}_x \mathcal{L}(\hat{x}, \lambda) \begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix} \beta \right) =$$

$= (\beta, B\beta)$, with

$$B := \begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix}^{ad} \text{Hess}_x \mathcal{L}(\hat{x}, \lambda) \begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix} \text{ "reduced Hessian"}$$

On the other hand, we can conceive the variable z as $z = z(p)$ via implicit function theorem and observe

$$\text{Hess}_p f(z(p), p) = B$$

2.4 Convex optimization with inequality constraints

We consider a practically important special case, which is later generalized

Theorem 2.22. *We consider the convex set*

$$S := \{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i, \text{ for all } i = 1, \dots, n\}$$

where $a, b \in \mathbb{R}^n$ with $a_i \leq b_i$ for all i . For $\lambda, \mu \in \mathbb{R}^n$, we define the Lagrangian

$$\mathcal{L}(x, \lambda, \mu) := f(x) + (\lambda, a - x) + (\mu, x - b)$$

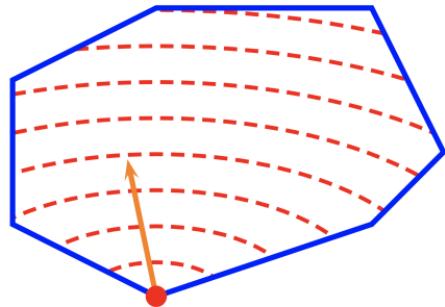
If \hat{x} is the solution to the problem

$$\begin{aligned} & \min f(x) \\ & x \in S \end{aligned}$$

with $f : S \rightarrow \mathbb{R}$ convex, then there exist $\lambda, \mu \in \mathbb{R}^n$ with $\lambda_i, \mu_i \geq 0$, for all i such that

$$\begin{aligned} & \nabla_x \mathcal{L}(\hat{x}, \lambda, \mu) = 0 \\ & \lambda_i (\hat{x}_i - a_i) = 0 = \mu_i (b_i - \hat{x}_i), \text{ for all } i = 1, \dots, n \end{aligned}$$

Remark: The exact same theorem holds pointwise almost everywhere for box constrained optimal control problems (see Tröltzsch, Borzi/Schulz).



In the general case of the problem

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & h_i(x) \leq 0, \forall i = 1, \dots, m \end{aligned}$$

we observe geometrically from theorem 2.11 in an optimal edge that

$$\nabla f(\hat{x}) = - \sum_{\text{active constraints } i} \lambda_i \nabla h_i(\hat{x})$$

with $\lambda_i \geq 0$, if the inequality constraints define a convex set. This, however, holds also in the non-convex set, as is shown in the next section.

2.5 Equality and Inequality Constraints

Consider the nonlinear programming problem

$$\begin{array}{ll}\min f(x) & f : \mathbb{R}^n \rightarrow \mathbb{R} \\ \text{s.t. } c(x) = 0 & c : \mathbb{R}^n \rightarrow \mathbb{R}^m \\ & h(x) \leq 0 \quad h : \mathbb{R}^n \rightarrow \mathbb{R}^l\end{array}$$

(component wise)

Definition 2.23. (*active set*)

- a) $S := \{x | c(x) = 0, h(x) \leq 0\}$ "**feasible region**"
- b) *for $x \in S$, we define*
 - $I(x) := \{i \in \{1, \dots, l\} | h_i(x) = 0\}$ "**active inequalities**"
 - $I^\perp(x) := \{i \in \{1, \dots, l\} | h_i(x) < 0\}$ "**inactive inequalities**"
- c) $x \in S$ *satisfies the LICQ (linear independence constraint qualification), if there holds*
 $\{\nabla c_1(x), \dots, \nabla c_m(x), \nabla h_{i_1}(x), \dots, \nabla h_{i_s}(x)\}$ *is a linear independent set,*
where $\{i_1, \dots, i_s\} = I(x)$

Remark: Note that LICQ is equivalent to surjectivity of $Dc(x)$ in the case of equality constraints only.

Theorem 2.24 (necessary optimality conditions). Let \hat{x} be a local minimum and we assume LICQ. Then, there is $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^l, \mu \geq 0$, such that for the Lagrangian

$$\mathcal{L}(x, \lambda, \mu) := f(x) + \lambda^\top c(x) + \mu^\top h(x) \text{ gilt}$$

hold

- a) $\nabla_x \mathcal{L}(\hat{x}, \lambda, \mu) = 0$ (KKT-Bed.)
- b) $\mu^\top h(\hat{x}) = 0$ (complementarity)
- c) $v^\top \text{Hess}_x \mathcal{L}(\hat{x}, \lambda, \mu) v \geq 0 \quad \forall v \in \tilde{T}(\hat{x})$

$$\tilde{T}(\hat{x}) = \{v \in \ker(Dc(\hat{x})) \mid \nabla h_j(\hat{x})^\top v = 0 \quad \text{for all } \mu_j > 0\}$$

Proof: Theorems 12.1 and 12.5 in [NW06] (with appropriate $+-$ exchanges). □

Remark: There are weaker constraint qualifications than LICQ: MFCQ (Mangasarian-Fromowitz constraint qualification), Slater condition,...

Theorem 2.25 (sufficient optimality condition). We assume that \hat{x} is feasible, LICQ and that a), b) from theorem 2.24 are satisfied and

$$v^\top \text{Hess}_{\hat{x}} \mathcal{L}(\hat{x}, \lambda, \mu) v > 0 \quad \forall v \in \tilde{T}(\hat{x}) \setminus \{0\}$$

then \hat{x} is a strict local minimum.

Proof: Theorems 12.6 in [NW06]. □

Examples:

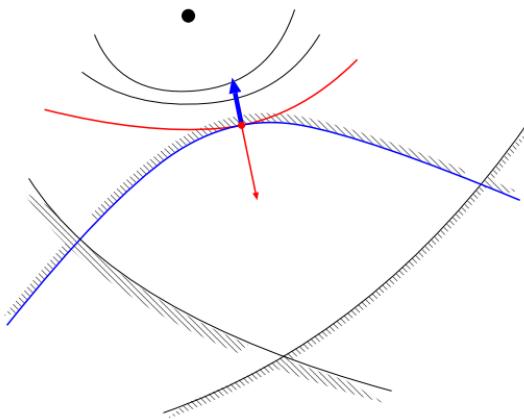
a) $\min x_1$
s.t. $x_2 \leq 0$
 $x_2 \geq x_1^2$

Unique feasible point is $x = (0, 0)$

$$\text{but } \nabla f = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \neq \mu_1 \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \mu_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \mu_1 \nabla h_1 + \mu_2 \nabla h_2$$

reason: LICQ is not satisfied!

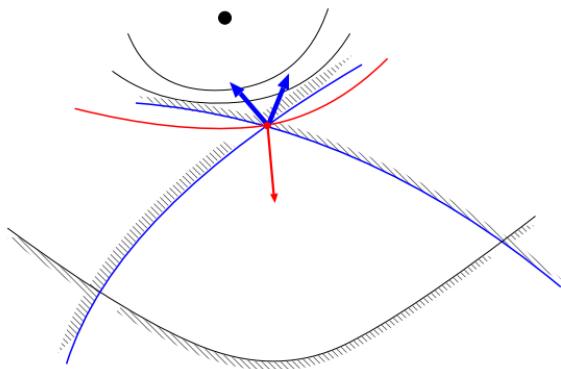
b) Example for solution on smooth boundary of feasible domain:



$$\nabla f = -\mu_1 \nabla h_1, \quad \mu_1 > 0$$

Only one inequality is active. The others do not play any role.

c) Example for solution at a corner:



$$\nabla f = -\mu_1 \nabla h_1 - \mu_2 \nabla h_2, \quad \mu_1, \mu_2 \geq 0$$

formulated differently: $-\nabla f \in \text{cone}(\nabla h_1, \nabla h_2)$

c) $\min -x^2 + 1$
s.t. $-1 \leq x \leq 2$

2.6 Exploiting duality in convex optimization for support vector machines

Famous algorithmic concept in data science, dating back to 1992 [BGV92]), which build up on the separation idea:

Theorem 2.26 (Minkowski, 1910). *Let $S_1, S_2 \subset \mathbb{R}^n$ be two nonempty convex and disjoint subsets. Then, there is a so-called separating hyperplane defined as the set $\mathcal{H}(w, b) := \{x \in \mathbb{R}^n \mid w^\top x = b\}$ with $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ such that*

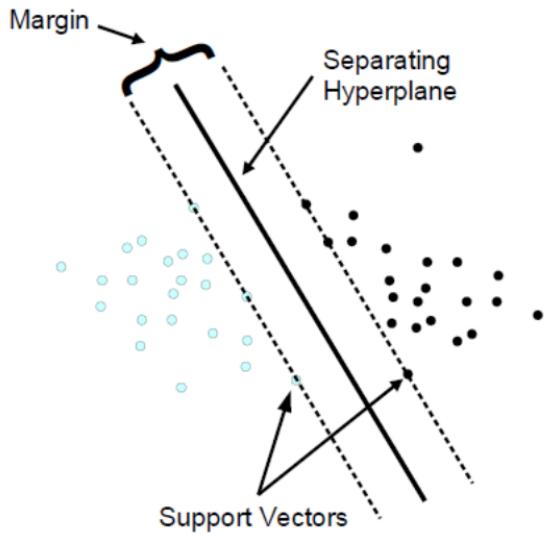
$$w^\top x \geq b, \forall x \in S_1 \text{ and } w^\top x \leq b, \forall x \in S_2$$

Proof. instead of a proof, we cite the original publication: [Min10]. □

“machine” \longleftrightarrow decision mapping d

e.g. in the form

$$d : \mathbb{R}^n \rightarrow \{\text{black class, blue class}\}$$



Data in the form of a set of pairs $D := \{(x_i, y_i) \mid x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}, i = 1, \dots, m\}$

For the moment we assume

$$\text{conv}(D^+) \cap \text{conv}(D^-) = \emptyset$$

where $D^+ := \{x \mid (x, y) \in D, y = +1\}, D^- := \{x \mid (x, y) \in D, y = -1\}$

Minkowski theorem 2.26 $\Rightarrow \exists$ separating hyperplane

$$\mathcal{H}(w, b) = \{x \in \mathbb{R}^n \mid w^\top x + b = 0\}, \quad w \in \mathbb{R}^n, b \in \mathbb{R}$$

\rightarrow search w, b such that the data lie in respective halfspaces, i.e.

$$D^+ \subset H^+ := \{x \in \mathbb{R}^n \mid w^\top x + b > 0\}, \quad D^- \subset H^- := \{x \in \mathbb{R}^n \mid w^\top x + b < 0\}$$

We observe that

$$\mathcal{H}(w, b) = \mathcal{H}(\alpha w, \alpha b), \quad \forall \alpha > 0$$

and also the respective half spaces H^+, H^- are not changed.

The support vectors are defined by:

$$x_{\min}^+ \in \operatorname{argmin}_{x \in D^+} \|x - \mathcal{H}\|, \quad x_{\min}^- \in \operatorname{argmin}_{x \in D^-} \|x - \mathcal{H}\| \quad (2.3)$$

Now, we can choose w, b such that

$$w^\top x_{\min}^+ + b = +1 \Rightarrow w^\top x_i + b \geq +1, \forall x_i \in D^+ \quad (2.4)$$

$$w^\top x_{\min}^- + b = -1 \Rightarrow w^\top x_i + b \leq -1, \forall x_i \in D^- \quad (2.5)$$

since this defines just two scalar equations for two unknowns (α, b) .

Half margin $\frac{m}{2} = \|\mu w\| = \frac{1}{\|w\|}$ to be maximized.

\Rightarrow we aim at the solution of the optimization problem

$$\min_{w,b} \frac{1}{2} \|w\|_2^2 \quad (2.6)$$

$$\text{s.t. } y_i(w^\top x_i + b) \geq +1, \forall i = 1, \dots, m \quad (2.7)$$

Important example of a convex (linear-quadratic) optimization problem.

→ Chance for exploiting the dual formulation, which is only advantageous in convex problems.

General optimization problem

$$\min_x f(x) \quad (2.8)$$

$$\text{s.t. } h(x) \leq 0, \quad h : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (2.9)$$

We assume f and $h_i, i = 1 \dots m$ convex such that the feasible region $S := \{x \mid h_i(x) \leq 0, i = 1 \dots m\}$ is convex.

Lagrangian

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^\top h(x), \quad (\lambda \geq 0)$$

Dual objective function

$$q(\lambda) := \min_x \mathcal{L}(x, \lambda)$$

Definition 2.27. *The dual problem to the convex optimization problem above is defined as the optimization problem*

$$\begin{aligned} & \max_{\lambda} q(\lambda) \\ & \text{s.t. } \lambda \geq 0. \end{aligned}$$

Theorem 2.28 (weak duality). *For any feasible \bar{x} and any $\bar{\lambda} \geq 0$, we have $q(\bar{\lambda}) \leq f(\bar{x})$.*

Proof.

$$q(\bar{\lambda}) = \min_x \{f(x) + \bar{\lambda}^\top h(x)\} \leq f(\bar{x}) + \bar{\lambda}^\top h(\bar{x}) \leq f(\bar{x})$$

since $\bar{\lambda} \geq 0$ and $h(\bar{x}) \leq 0$. □

Strong duality (i.e. $f(\hat{x}) - q(\hat{\lambda}) = 0$ for an optimal pair $(\hat{x}, \hat{\lambda})$) needs deeper discussion and only plays a role in linear programming.

There exists another dual, called the **Wolfe dual**, which is computationally more attractive in some instances:

$$\max_{x,\lambda} \mathcal{L}(x, \lambda) \tag{2.10}$$

$$\text{s.t. } \nabla_x \mathcal{L}(x, \lambda) = 0, \quad \lambda \geq 0 \tag{2.11}$$

Theorem 2.29. *Suppose in the primal problem (2.8,2.9) that f and $c_i, i = 1, \dots, m$ are convex functions on \mathbb{R}^n that are differentiable at \hat{x} . If the pair $(\hat{x}, \hat{\lambda})$ solves the KKT conditions for problem (2.8,2.9), where LICQ holds, then $(\hat{x}, \hat{\lambda})$ solves also problem (2.10, 2.11).*

Proof.

□

Wolfe dual of the problem (2.6, 2.7).

Lagrangian:

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^m \lambda_i(1 - y_i(w^\top x_i + b))$$

resulting (Wolfe) dual optimization problem

$$\max_{w, b, \lambda} \mathcal{L}(w, b, \lambda)$$

$$\text{s.t. } 0 = \nabla_w \mathcal{L}(w, b, \lambda) = w - \sum_{i=1}^m \lambda_i y_i x_i$$

$$0 = -\nabla_b \mathcal{L}(w, b, \lambda) = \sum_{i=1}^m \lambda_i y_i$$

The Wolfe dual problem can be simplified to

$$\begin{aligned} \max_{\lambda} & \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j y_i y_j x_i^\top x_j \\ \text{s.t. } & \sum_{i=1}^m \lambda_i y_i = 0 \\ & \lambda_i \geq 0, \quad \forall i = 1, \dots, m \end{aligned}$$

Proof:

The support vector decision machine is now evaluated for any $x \in \mathbb{R}^n$ as

$$d(x) = w^\top x + b = \sum_{\substack{i=1 \\ \lambda_i \neq 0}}^m \lambda_i y_i x_i^\top x + b \begin{cases} \geq 0 \Rightarrow x \in H^+ \\ \leq 0 \Rightarrow x \in H^- \end{cases}$$

Numerical solution methods are based on ideas from coordinate descent, SMO (sequential minimization optimization) or projected gradient methods, which are discussed below. Specific examples are LIBSVM (`mlpy.LibSvm` in Python, based on SMO which is similar to ADMM) and LIBLINEAR (based on coordinate descent). In the sequel, we discuss SVM variants enlarging the domain of applicability tremendously.

Soft margin hyperplane:

We allow small deviations from the margin in the of the modified optimization problem

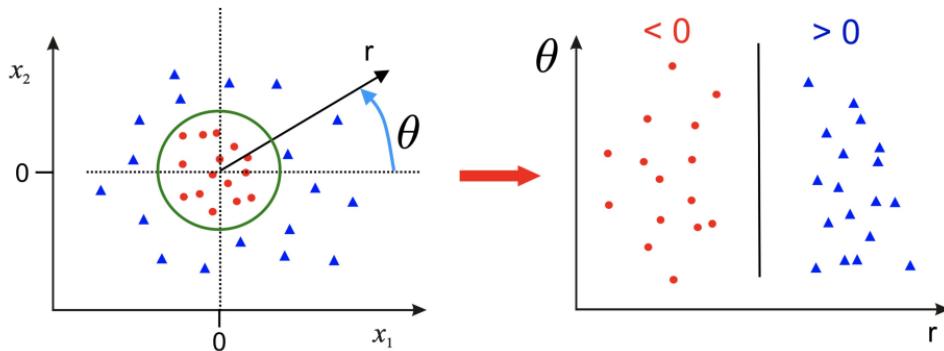
$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t. } \quad & y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, m \\ & \xi_i \geq 0, \quad \forall i = 1, \dots, m \end{aligned}$$

where $C > 0$ is a constant to be chosen by the user. The resulting dual problem (exercises) is

$$\begin{aligned} \max_{\lambda} \quad & \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j y_i y_j x_i^\top x_j \\ \text{s.t.} \quad & \sum_{i=1}^m \lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq C, \quad \forall i = 1, \dots, m \end{aligned}$$

Kernel trick for not linearly separable data:

Consider the following data situation, which is not linearly separable. However, using applying



the mapping to polar coordinates

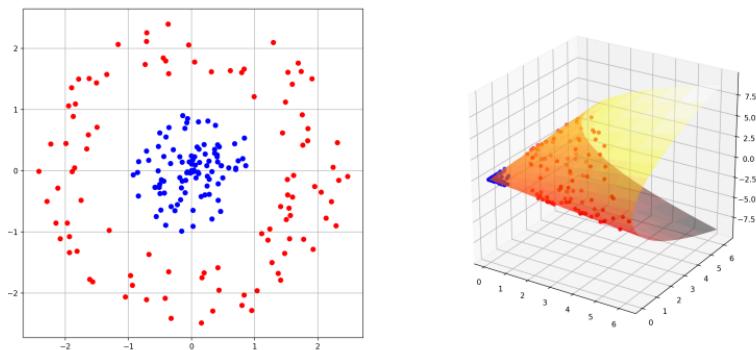
$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$
$$x = re^{i\theta} \mapsto (r, \theta)^\top$$

to the data results in data, which are linearly separable.

Also going to a higher dimension by applying the mapping

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$
$$x = \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top \quad (2.12)$$

to the data results in data in the following form which are again linearly separable.



In general, we consider a so-called feature space \mathcal{F} (vector space) and a feature map

$$\begin{aligned}\Phi : \mathbb{R}^n &\rightarrow \mathcal{F} \\ x &\mapsto \Phi(x)\end{aligned}$$

where we hope that the transformed data are linearly separable. Then, in all optimization problem formulations above, x_i is replaced everywhere by $\Phi(x_i)$. Resulting in the following dual optimization problem

$$\begin{aligned}\max_{\lambda} \quad & \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j \Phi(x_i)^\top \Phi(x_j) \\ \text{s.t.} \quad & \sum_{i=1}^m \lambda_i y_i = 0 \\ & 0 \leq \lambda_i \leq C, \quad \forall i = 1, \dots, m\end{aligned}$$

with classifier decision function

$$d(x) = \operatorname{sgn}(w^\top \Phi(x) + b) = \operatorname{sgn} \left(\sum_{\substack{i=1 \\ \lambda_i \neq 0}}^m \lambda_i y_i \Phi(x_i)^\top \Phi(x) + b \right)$$

we observe that Φ always appears in scalar products.

Therefore, all we need is the kernel mapping

$$\begin{aligned} k : \mathbb{R}^n \times \mathbb{R}^n &\rightarrow \mathbb{R} \\ (x, y) &\mapsto \Phi(x)^\top \Phi(y) \end{aligned}$$

rather than Φ itself. Thus, the optimization problem and the classifier are reformulated again

$$\begin{aligned} \max_{\lambda} \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j k(x_i, x_j) \\ \text{s.t. } \sum_{i=1}^m \lambda_i y_i = 0 \\ 0 \leq \lambda_i \leq C, \quad \forall i = 1, \dots, m \end{aligned}$$

with classifier decision function

$$d(x) = \operatorname{sgn} \left(\sum_{\substack{i=1 \\ \lambda_i \neq 0}}^m \lambda_i y_i k(x_i, x) + b \right)$$

Typically, one of the following standard kernels are used with appropriate kernel parameters

$$\text{linear: } k(x, y) = x^\top y$$

$$\text{polynomial: } k(x, y) = (\theta + x^\top y)^\ell, \text{ for some } \ell \in \mathbb{N}_+$$

$$\text{sigmoid: } k(x, y) = \tanh(\theta + x^\top y)$$

$$\text{Gaussian: } k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \text{ for some } \sigma > 0$$

SVM based on the Gaussian kernel is also called “radial basis function SVM”. We observe that the above kernel trick leads to an algorithmic complexity which depends only on $\#D$ and not on the dimension of the data, n . Note that everywhere in those kernels can additional linear parameters pop up.

Standard implementations of SVM are well-known as **LIBSVM** and **LIBLINEAR**.

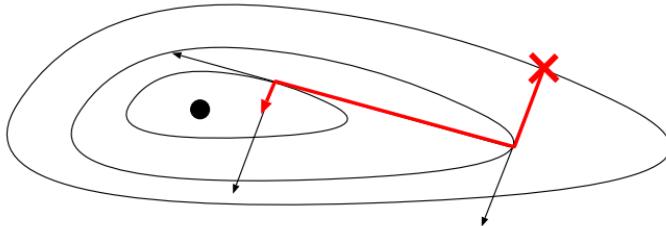
Remark: Note that the kernel from the feature map (2.12) is just $k(x, y) = (x^\top y)^2$. In general, this raises the question, whether a corresponding feature map can be found for every kernel. This is indeed the case, provided the matrix $(k(x_i, x_j))_{ij}$ is symmetric and positive definite, which is called the “Mercer condition” [Abe12].

3 Algorithms for unconstrained optimization problems

$\min f(x), f : X \rightarrow \mathbb{R}, \quad X \text{ Eucl. vector space} \quad \text{mostly: } X = \mathbb{R}^n$

where should we go, in order to reach the valley?
(only shortsighted local information available !)

3.1 Steepest descent method (Cauchy 1817)



Method of type $x^{k+1} = x^k + t^k d^k$

d^k : search direction
 t^k : step size

Definition 3.1. For $f : X \rightarrow \mathbb{R}$, differentiable, we call $d \in X$ a descent direction at the point x , if

$$(\nabla f(x), d) < 0$$

Lemma 3.2. d descent direction $\Rightarrow \exists \alpha_0 > 0$ with $f(x + \alpha d) < f(x)$, $\forall \alpha \in (0, \alpha_0) < 0$

Theorem 3.3. Let $\nabla f(x) \neq 0$. Among all d with $\|d\| = 1$, the direction $d^g := -\frac{\nabla f(x)}{\|\nabla f(x)\|}$ gives the steepest descent, i.e. solves

$$\begin{aligned} \min_d Df(x)d &= (\nabla f(x), d) \\ \text{s.t. } \|d\|^2 &= 1 \end{aligned}$$

Remark: Interpretation: d^g minimizes

$$\begin{array}{ll} \min_d & f(x) + (\nabla f(x), d) \leftarrow \text{"linear Approximation of } f\text{"} \\ \text{unter} & \|d\| = 1 \end{array}$$

Algorithm (steepest descent)

x^0 start, $\varepsilon > 0$ stopping threshold

for $k = 0, 1, \dots$, do

- (1) compute $f(x^k), \nabla f(x^k)$
- (2) if $\|\nabla f(x^k)\| \leq \varepsilon \rightarrow \text{STOP}$, solution reached
- (3) else

$$d^k := -\frac{\nabla f(x)}{\|\nabla f(x)\|} \quad \text{resp.} \quad d^k := -\nabla f(x) \text{ (because of line search)}$$

- (4) compute $t^k := \arg \min_t f(x^k + td^k), t \in (0, \infty)$
- (5) update $x^{k+1} := x^k + t^k d^k$

Theorem 3.4. (global convergence without stopping step (2))

Assume that the level set $N(x^0) := \{x \mid f(x) \leq f(x_0)\}$ is compact.

Then, either $\nabla f(x^{\bar{k}}) = 0$ for a \bar{k} , or there is at least one accumulation point x^* of the sequence $\{x^k\}$, and at this point yields $\nabla f(x^*) = 0$

Lemma 3.5. (*Zig Zagging*). *For iterates of the steepest descent algorithm (with exact line search) holds*

$$\nabla_x f(x^{k+1}) \perp \nabla_x f(x^k), \quad \forall k$$

Proof:

Remark: Steepest descent algorithms can be applied also in Hilbert spaces, convergence is shown only for quadratic functionals.

(eg. Wouk: A course of applied functional analysis, Wiley 1979 , chap. 11.2)

For the analysis of the convergence speed, we need the following

Lemma 3.6 (Inequality of Kantorovich). *If $Q \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. Then, there holds for all $x \in \mathbb{R}^n$:*

$$\frac{(x^\top x)^2}{(x^\top Qx)(x^\top Q^{-1}x)} \geq 4 \frac{\lambda_{\min}\lambda_{\max}}{(\lambda_{\min} + \lambda_{\max})^2} \geq \frac{\lambda_{\min}}{\lambda_{\max}}$$

where $\lambda_{\min}, \lambda_{\max}$ are the smallest and largest eigenvalue of Q .

Proof: see Homepage

□

Theorem 3.7. Consider a quadratic problem of the form

$$\min f(x) := \frac{1}{2}x^\top Qx + b^\top x, \quad b \in \mathbb{R}^n, Q \in \mathbb{R}^{n \times n} \text{ symm. p. d.}$$

For this problem. the steepest descent method with exact line search converges linearly:

$$\|x^{k+1} - \hat{x}\|_Q \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right) \|x^k - \hat{x}\|_Q$$

where $\lambda_{\max / \min} = \text{maximal/minimal eigenvalue of } Q$ and $\|y\|_Q := \sqrt{y^\top Qy}$

Proof: W.l.o.g. $b = 0$

(otherwise: variable substitution $y := x + Q^{-1}b$, use then: $x = y - Q^{-1}b$)

We know from the exercises

$$f(x^{k+1}) = \underbrace{\left(1 - \frac{(g_k^\top g_k)^2}{(g_k^\top Q g_k)(g_k^\top Q^{-1} g_k)}\right)}_{(*)} f(x^k)$$

with $g_k := \nabla f(x^k) = Qx^k$

The inequality of Kantorovich (see above) gives then:

$$(*) \geq 4 \frac{\lambda_{\min} \cdot \lambda_{\max}}{(\lambda_{\min} + \lambda_{\max})^2}. \text{ Thus}$$

$$\Rightarrow f(x^{k+1}) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 f(x^k)$$

$$\begin{array}{ccc} \| & & \| \\ \frac{1}{2} x_{k+1}^\top Q x_{k+1} & & \frac{1}{2} x_k^\top Q x_k \\ \| & & \| \\ \frac{1}{2} \|x^{k+1} - \hat{x}\|_Q^2 & & \frac{1}{2} \|x^k - \hat{x}\|_Q^2 \end{array}$$

□

Remarks:

- (1) The Taylor series gives the insight that the theorem holds also asymptotically in the vicinity of solutions of nonlinear problems. Then, Q has to be replaced by $Q := \text{Hess}_x f(\hat{x})$.
- (2) The local metric influences gradients and the spectrum of the Hessian, thus also the convergence rate: Let A (symm. p.d.) be the local metric with $(x, y) := x^\top A y \Rightarrow \nabla f = A^{-1}(f')^\top$ and $\text{Hess} f = A^{-1}[\partial^2 f / \partial x^2]$ is relevant for the convergence rate.
 \Rightarrow optimal choice of the metric $A \approx [\partial^2 f / \partial x^2]$ and then $\lambda_{\min} \approx \lambda_{\max}$ giving a small convergence rate !

Recall definitions of order of convergence:

quadratic: $\|x^{k+1} - \hat{x}\| \leq C \|x^k - \hat{x}\|^2$ for some $C < \infty$

superlinear: $\|x^{k+1} - \hat{x}\| \leq \alpha_k \|x^k - \hat{x}\|$ for $\alpha_k \downarrow 0$

linear: $\|x^{k+1} - \hat{x}\| \leq \alpha \|x^k - \hat{x}\|$ for some $\alpha_k < 1$

sublinear: $\|x^k - \hat{x}\| \leq \frac{C}{k^\beta}$, often $\beta \in \{2, 1, \frac{1}{2}\}$

Remark: Note that the convergence orders are in \mathbb{R}^n independent from the chosen norm—with the exception of linear independence. In [KS88] linear convergence with a slightly worse convergence rate than for the norm $\|\cdot\|_Q$ is shown also for the Euclidean norm.

Often, line search requires more effort than expected. Then, one could try constant step size.

Theorem 3.8. *Consider for the quadratic program of theorem 3.7 the steepest descent method $x^{k+1} = x^k - \tau \cdot \nabla f(x^k)$ with constant step size $\tau > 0$. Then, the iteration converges, provided $\tau < 2/\lambda_{\max}$*

Remark: It is also well-known in this context that the optimal convergence rate $\rho_{opt} = (\lambda_{\min} - \lambda_{\max}) / (\lambda_{\min} + \lambda_{\max})$ is achieved with the constant step size $\tau = 2 / (\lambda_{\min} + \lambda_{\max}) < 2 / \lambda_{\max}$. erhalten.

Now, we investigate **inexact** line search:

Requirements: we want to have $f(x^{k+1}) < f(x^k)$, but also not too large step size, in order to avoid erratic oscillations.

→ There are various strategies:

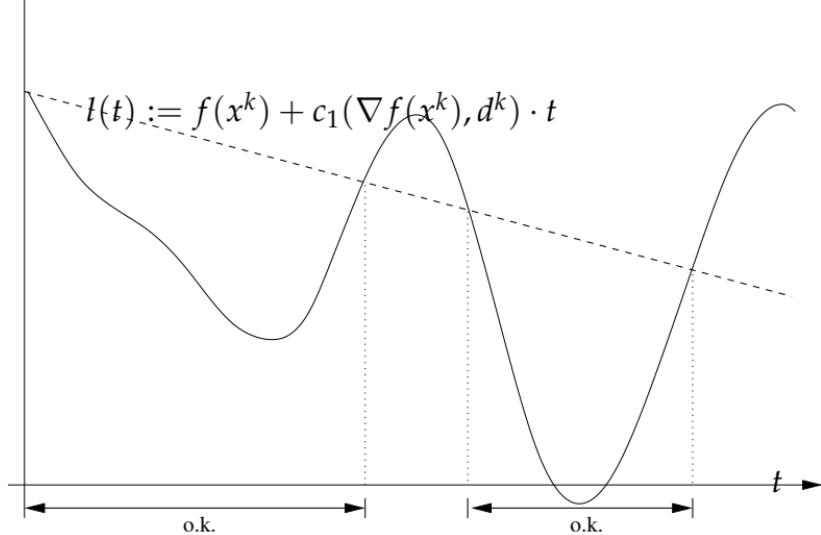
- Armijo
- Goldstein
- Wolfe conditions

$$(a) f(x^k + t^k d^k) \leq f(x^k) + c_1 t^k (\nabla f(x^k), d^k)$$

"minimum decrease"

$c_1 > 0$ independent of k , $c_1 \approx 10^{-4}$

$$\phi(t) := f(x^k + t d^k)$$

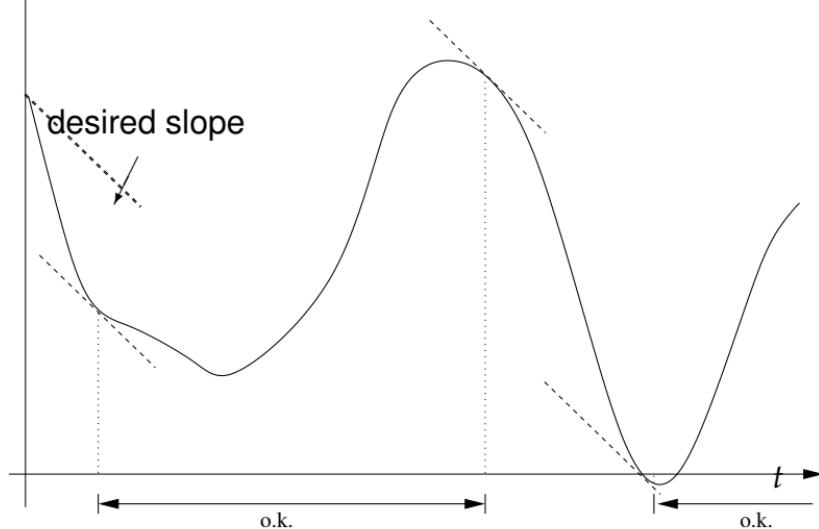


$$(b) (\nabla f(x^k + t^k d^k), d^k) \geq c_2 (\nabla f(x^k), d^k)$$

"curvature condition"

with $0 < c_1 < c_2 < 1$, $c_2 \approx 0,9$

$$\phi(t) := f(x^k + td^k)$$



Theorem 3.9. Consider the angle between the descent direction d^k and the steepest descent direction $-\nabla f(x^k)$, defined by ϑ_k , where

$$\cos \vartheta_k := -(\nabla f(x^k), d^k) / (\|\nabla f(x^k)\| \|d^k\|)$$

t^k is assumed to satisfy the Wolfe conditions (a, b) and $f \in C^1$ and ∇f Lipschitz.
Then,

$$\sum_{k \geq 0} \cos^2 \vartheta_k \|\nabla f(x^k)\|^2 < \infty$$

Proof: Nocedal/Wright Theorem 3.2

□

Corollary 3.10. If we guarantee in the setting above that $\cos \vartheta_k \geq c_3 > 0$, c_3 independent of k , then holds

$$\lim_{k \rightarrow \infty} \|\nabla f(x^k)\| = 0 \quad (\text{globale Konv.})$$

Proof: obvious ✓

backtracking line-search

Choose a fixed $\bar{t} > 0$, $\rho, c_1 \in (0, 1)$, set $t \leftarrow \bar{t}$

repeat

$$t \leftarrow \rho \cdot t$$

until $f(x^k + td^k) \leq f(x^k) + c_1 t (\nabla f(x^k), d^k)$

→ Observation: condition (a) “minimum decrease” is satisfied.

(b) does not have to be guaranteed explicitly, if \bar{t} is chosen large enough. → step size will not become too small.

3.2 Momentum methods

Steepest descent gives a linear convergence rate $\rho = \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)$. Can we get a better convergence rate, but still use only information of first order? YES -the key is momentum. The following methods can be classified as “momentum methods”:

1. heavy-ball method (Polyak, 1987)
2. Nesterov’s accelerated gradient method (Nesterov, 1983)
3. Adam, Adaptive Momentum (Kingma/Ba 2014)

4. conjugate gradient method (Hestenes/Stiefel, 1952)

Method 4 (CG) is discussed in detail in the subsequent section. Methods 1 and 2 originate from discretizations of the dynamic systems

$$\ddot{x}(t) + g(t)\dot{x}(t) + \nabla f(x(t)) = 0, \quad \lim_{x \rightarrow \infty} x(t) = \hat{x}$$

with Runge-Kutta discretizations lead to the iterations

1. heavy-ball method: $x^{k+1} = x^k - \alpha \nabla f(x^k) + \beta(x^k - x^{k-1})$

2. Nesterov's accelerated gradient method:

$$x^{k+1} = x^k - \alpha^k \nabla f(x^k + \beta_k(x^k - x^{k-1})) + \beta_k(x^k - x^{k-1}), \quad \alpha_k \approx 1/L, \beta_k \approx 1$$

3. Adam (Adaptive Momentum):

$$x_i^{k+1} = x_i^k - \alpha \frac{\nu_k}{\sqrt{s_k + \epsilon}} \nabla f(x^k)_i, \quad i = 1, \dots, n$$

where $\nu_k = \beta_1 \nu_{k-1} - (1 - \beta_1) \nabla f(x^k)_i, \quad \beta_1 \approx 0.9$

$$s_k = \beta_2 s_{k-1} - (1 - \beta_2) \nabla f(x^k)_i^2, \quad \beta_2 \approx 0.99, \quad \epsilon \approx 10^{-10}$$

Both methods have slightly faster convergence properties than steepest descent and maybe advantageous in particular cases. However, theory is only developed for convex problems. More details on the relation to dynamic systems can be found in [ZMSJ18] and on <https://distill.pub/2017/momentum/>

3.3 CG - conjugate gradient method (Hestenes/Stiefel 1952)

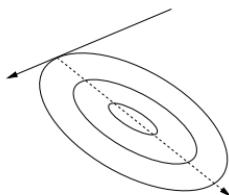
linear quadratic objective

$$\begin{aligned}\min f(x) &:= \frac{1}{2}(x, Qx) + (b, x) \\ Q &: X \rightarrow X \text{ symm. p.d. Operator} \\ b &\in X \text{ Vector}\end{aligned}$$

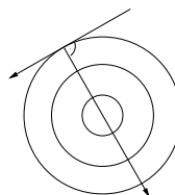
→ Observation: Setting is also useful for the sol. of linear equations with p.d. operator Q .

Idea: Lemma 3.5 shows $\nabla f^{k+1} \perp \nabla f^k$

→ choose metric so that orthogonal directions bring you quickly to the solution !



quick directions not orthogonal



quick directions orthogonal

Variable transformation $x \leftrightarrow y = Q^{\frac{1}{2}}x$ ($x = Q^{-\frac{1}{2}}y$)

$$\rightarrow (x, Qx) = (Q^{-\frac{1}{2}}y, QQ^{-\frac{1}{2}}y) = (y, y)$$

corresponds to choice of the new metric $(y, y)_Q = (y, Qy)$

Goal: Find successively vectors $p^k \in X$, which are orthogonal with respect to $(\cdot, \cdot)_Q$, i.e.,

$$(p^k, p^j)_Q = 0, \quad \forall i \neq j$$

such vectors are called "conjugate w.r.t. Q ". Successive minimization in these directions should lead to the solution quickly

First attempt: "dead certain" (but inefficient) algorithm in \mathbb{R}^n

start with $k = 1$, p_1 arbitrary

- ① For $\{p^1, \dots, p^{k-1}\}$ (already conjugate) find conjugate direction p^k
$$span \{p^1, \dots, p^k\} = \{P_k \omega : \omega \in \mathbb{R}^k\},$$
 where $P_k := [p^1 | \dots | p^k]$
- ② $\arg \min_{\omega} f(x^k + P_k \omega) =: \omega^k$
- ③ update $x^{k+1} := x^k + P_k \omega^k$
goto ①

Observations:

- Algorithm stops after n iterations, since then $\text{span} \{ p^1, \dots, p^n \} = \mathbb{R}^n$
- In step ② there holds

$$\left. \begin{array}{l} \min \frac{1}{2} x^\top Qx + b^\top x \\ \text{s.t. } x = x_k + P_k \omega \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \min_{\omega} \frac{1}{2} \omega^\top P_k^\top Q P_k \omega + g_k^\top P_k \omega \\ \text{with } g_k = \nabla f(x^k) = Qx^k + b \end{array} \right.$$

$$\text{solution } \omega = -(P_k^\top Q P_k)^{-1} P_k^\top g_k$$

$$\Rightarrow x^{k+1} = x^k - P_k (P_k^\top Q P_k)^{-1} P_k^\top g_k$$

Theorem 3.11. *The “dead certain algorithm” has the following properties*

a) $g_{k+1}^\top p_i = 0 \quad \forall i = 1, \dots, k$

b) $x^{k+1} = x^k + \alpha^k p_k$ with $\alpha^k = -\frac{g_k^\top p_k}{p_k^\top Q p_k}$

Proof:

Question: How to obtain the conjugate directions ?

Ansatz : $p_k = -g_k + \sum_{j=1}^{k-1} \beta_{kj} p_j \quad k \geq 2$ after several computational steps, we obtain
 $p_1 = -g_1$

$$\beta_{ki} = 0, \forall i < k-1$$

$$\beta_{k-1} := \beta_{k,k-1} = \frac{(g_k - g_{k-1})^\top g_k}{(g_k - g_{k-1})^\top p_{k-1}} \text{ and thus}$$

$$p_k = -g_k + \beta_{k-1} p_{k-1}$$

This is due to Hestenes and Stiefel, we observe easily:

$$(g_k - g_{k-1})^\top g_k = \|g_k\|_2^2 \quad \text{and} \quad (g_k - g_{k-1})^\top (-g_{k-1} + \beta_{k-2} p_{k-2}) = g_{k-1}^\top g_{k-1} = \|g_{k-1}\|_2^2$$

Thus, β_{k-1} can be written as

$$\beta_{k-1} = \frac{(g_k - g_{k-1})^\top g_k}{\|g_{k-1}\|_2^2} = \frac{\|g_k\|_2^2}{\|g_{k-1}\|_2^2}$$

↑ ↑

"Polak-Ribiere"

"Fletcher-Reeves"

→ makes a difference in nonlinear usage !

Result: 3 term recursion

Start: $\beta_1 = 0, p_1 = 0, x^1 = 0, \quad r_1^\swarrow = Qx^1 + b$ ('residual' = grad.)

$$\begin{array}{lcl} p_k & = & -r_k + \beta_{k-1} p_{k-1} \\ \alpha_k & = & \|r_k\|_2^2 / (p_k^\top Q p_k) \\ x^{k+1} & = & x^k + \alpha_k p_k \\ r_{k+1} & = & r_k + \alpha_k Q p_k \\ \beta_k & = & \|r_{k+1}\|_2^2 / \|r_k\|_2^2 \end{array}$$

$\nwarrow \qquad \leftarrow \qquad \swarrow$
3 terms / vectors !

Properties of the cg iteration:

- * By construction, termination after at most n steps

- * CG belongs to the class of Krylov subspace methods (GMRES, biCG, BiCGstab, QMR, GMRES, MINRES,...). Conceptual comparison of GMRES versus CG for the problem: solve $Qx + b = 0$, with Q symmetric and p.d.

GMRES: Within the Krylov space $\mathcal{K}_k(Q, b)$ the following problem is solved

$$\begin{aligned} \min \frac{1}{2} \|Qx + b\|_2^2 \\ x \in \mathcal{K}_k(Q, b) = \text{span}\{b, Qb, Q^2b, \dots, Q^{k-1}b\} \end{aligned}$$

where the so-called Arnoldi process (aka Gram-Schmidt for $\{b, Qb, Q^2b, \dots, Q^{k-1}b\}$) an Orthonormalbasis in the Euclidean scalar product is produced.

CG: Within the Krylov space $\mathcal{K}_k(Q, b)$ the following (different) problem is solved

$$\begin{aligned} \min \frac{1}{2} x^\top Qx + b^\top x \\ x \in \mathcal{K}_k(Q, b) = \text{span}\{b, Qb, Q^2b, \dots, Q^{k-1}b\} \end{aligned}$$

where the so-called Lanczos process (also aka Gram-Schmidt for $\{b, Qb, Q^2b, \dots, Q^{k-1}b\}$) is used to construct an orthonormal basis in the Q -scalar product.

Note that GMRES cannot be reduced to a 3 term recursion, but is usable for general linear systems of equations.

- * Thus one can show that the CG iteration stops even after $s < n$ steps, if Q has only s different eigenvalues

- * CG uses only matrix vector products
- * CG is typically used an iterative solver for high dimensional ($n \gg 1$) problems and gives good approximations already after a few iterations, if $\sigma(Q)$ has only few clusters

Otherwise convergence rate $\|x^k - \hat{x}\| \leq 2 \left(\frac{\sqrt{\lambda_{\max}(Q)} - \sqrt{\lambda_{\min}(Q)}}{\sqrt{\lambda_{\max}(Q)} + \sqrt{\lambda_{\min}(Q)}} \right)^k \|x^0 - \hat{x}\|$, proof via Tschebychev polynomials and Krylov subspace methods

- * CG generalization to Hilbert spaces obvious. convergence analysis, e.g. Wouk: applied FA
- * There are nonlinear variants ($r_k = \nabla f(x^k)$), which are different for different choices of β_{k-1} (Hestenes-Stiefel, Fletcher-Reeves, Polak-Ribiere).

Later in trust region methods, we'll exploit the following facts:

- * $f(x^k)$ is monotonously decreasing
 - * $\|x^k\|_2$ is monotonously increasing, if $x^0 = 0$
- $\left. \begin{array}{l} \\ \end{array} \right\}$ Steihaug 1983

Preconditioning The convergence rate of CG depends on the eigenvalue distribution of the system matrix. Thus, we may transform the linear system $Qx + b = 0$ equivalently such that the new linear system has a “nicer” eigenvalue distribution. We observe

$$Qx + b = 0 \Leftrightarrow E^{-1}Q(E^{-1})^\top y + E^{-1}b = 0$$

if E is an invertible matrix and $y = Ex$. Note for the resulting system matrix $E^{-1}Q(E^{-1})^\top$:

$$\sigma(E^{-1}Q(E^{-1})^\top) = \sigma(E^{-1}(E^{-1})^\top Q)$$

Thus the preconditioning matrix $W := EE^\top$ is symmetric and should be chosen somewhat close to Q . If we apply the standard CG method to the reformulated problem $E^{-1}Q(E^{-1})^\top y + E^{-1}b = 0$, the final algorithm only involves solutions with W (i.e., the factor E is not explicitly visible, only implicitly in so far, as the matrix W has to be symmetric and positive definite)

PCG algorithm:

Start: $\beta_1 = 0, p_1 = 0_1, x^1 = 0, r_1 = Qx^1 + b, z_1 = W^{-1}r_1$

$$\begin{aligned} p_k &= -z_k + \beta_{k-1}p_{k-1} \\ \alpha_k &= z_k^\top r_k / (p_k^\top Q p_k) \\ x^{k+1} &= x^k + \alpha_k p_k \\ r_{k+1} &= r_k + \alpha_k Q p_k \\ z_{k+1} &= W^{-1}r_{k+1} \\ \beta_k &= z_{k+1}^\top r_{k+1} / z_k^\top r_k \end{aligned}$$

appropriate preconditioners W ?

- Jacobi
- Gauß-Seidel (symm.)
- ILU
- Incomplete Cholesky
- multigrid

3.4 Newton method / quasi-Newton method

Necessary condition for $\min_{x \in X} f(x), \quad X = \mathbb{R}^n$

is $\nabla f(\hat{x}) = 0$, \hat{x} solution

We can apply standard Newton method (known from “Elements of math.” or “introduction to numerical mathematics”) to this nonlinear equation. We need to analyze $D\nabla f$.

Lemma 3.12. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable, then, there holds for the derivative of the gradient:*

$$D\nabla f(x) = \text{Hess } f(x)$$

Proof:

Newton method: algorithm:

Assumption: $\text{Hess } f(x^k)$ p.d. $\forall x^k$ iterates

Start at $x^0, k = 0$

solve $\text{Hess } f(x^k)d^k + \nabla f(x^k) = 0$ (Taylor)

compute $t^k := \arg \min_{0 \leq t} f(x^k + td^k)$ (approximately)

update $x^{k+1} := x^k + t^k d^k$, $k := k + 1$

Iteration until $\|d^k\| < \varepsilon$ (recall: $\|x^k - \hat{x}\| \doteq \|d^k\|$ for x^k close to \hat{x})

Observations:

- (1) • since $\text{Hess } f(x^k)$ p.d., it can be considered a varying metric, such that

$$d^k = -\text{Hess } f(x^k)^{-1} \nabla f(x^k) \text{ is also descent direction}$$

- Theorem 3.9 and Corollary 3.10 give us convergence exact/approximate line search satisfying Wolfe conditions, if the spectra satisfy $\sigma(\text{Hess } f(x^k)) \subset [m, L]$ with $m, L > 0$ independent of k .

(2) Close to the solution, we get quadratic convergence, i.e.

$$\|x^k - \hat{x}\| \leq C \|x^k - \hat{x}\|^2, C < \infty,$$

if $t^k = 1$ is chosen (Theorem of Newton-Kantorovich)

→ linesearch should be restricted to the interval $t \in (0, 1)$

($\bar{t} = 1$ in backtracking line-search)

What happens, if $\text{Hess } f(x^k)$ is not p.d. ?

- $\text{Hess } f(x^k)^{-1}$ may not exist !
- $(-\text{Hess } f(x^k)^{-1} \nabla f(x^k))$ may not be a descent direction !
- the whole line search concept is in question.

Remedies: 2 fundamentally different strategies:

- ① Construct $H^k \approx \text{Hess } f(x^k)$ with H^k p.d. and us this instead of exact Hessian.
- ② trust region method

ad ① $Hessf(x^k) \in \mathbb{R}^{n \times n}$

Since $Hessf(x^k)$ symmetric, the LU decomposition gives $Hessf(x^k) = L \cdot R$, with $R = DL^\top$ and $D = \begin{bmatrix} r_{11} & & 0 \\ & \ddots & \\ 0 & & r_{nn} \end{bmatrix}$. The decomposition $Hessf(x^k) = LDL^\top$ is called Cholesky decomposition (if $r_{ii} > 0$)

If $Hessf(x^k)$ indefinite, the diagonal entries r_{ii} are no longer all > 0 , but we can compute a positive perturbation

$r_{ii} + d_{ii} > \varepsilon \rightarrow$ modified Cholesky decomposition (N/W S. 145 ff)

ad ② Trust region method:

Idea:

- * Instead of first computing the step direction and afterwards the step length (as in line search methods), we first determine the step length (the trust region) and compute the optimal step afterwards.
- * We construct a quadratic model for the function f at the position x^k :

$$m^k(x^k + s) = f(x^k) + (\nabla f(x^k), s) + \frac{1}{2}(s, B_k s) \text{ with } B_k = Hessf(x^k) \text{ for example}$$

* We solve in each iteration

$$(TR) \begin{bmatrix} \min_s m^k(x^k + s) \\ \text{s.t. } \|s\| \leq \delta_k, \delta_k > 0 \end{bmatrix}$$

with update $x^{k+1} = x^k + s$

We have to solve two subproblems: the TR subproblem and the determination of the size of the trust region δ_k . First, the TR subproblem:

Theorem 3.13. *Let \hat{s} be the unique solution of the trust region subproblem*

$$\begin{array}{ll} \min_s & \frac{1}{2}(s, Bs) + (g, s) \\ \text{s.t.} & \|s\| \leq \delta \end{array}$$

Then, there is $\lambda \geq 0$ with

- (a) $(B + \lambda \cdot id)\hat{s} = -g$
- (b) $\lambda(\delta - \|\hat{s}\|) = 0$
- (c) $B + \lambda id$ is positive definite
- (d) λ is monotonously increasing for decreasing δ , if $(B + \lambda id)$ pos. def.

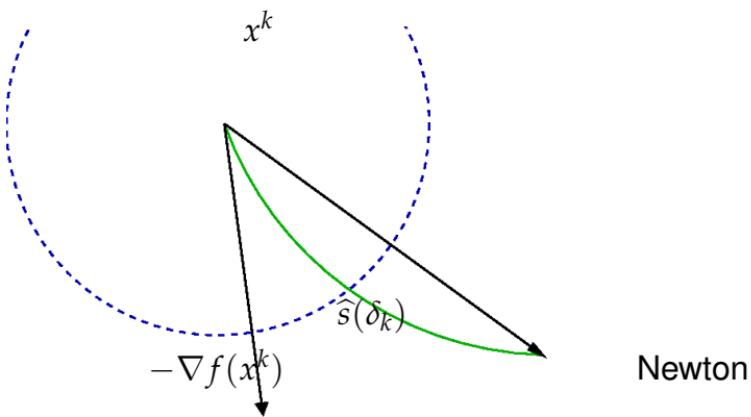
Remarks:

(1) \hat{s} is obviously descent direction in the sense of definition 3.1, since

$$\hat{s} = -(B + \lambda id)^{-1}g \text{ with } (B + \lambda id)^{-1} \text{ pos. def.}$$

(2) For $\delta \searrow 0$ we have $\lambda \nearrow \infty$ and $\hat{s}(\delta) \rightarrow -\frac{1}{\lambda}g \Rightarrow \lambda id$ dominates in $B + \lambda id$, which means that the TR step is parallel to the steepest descent.

TR produces step directions between Newton step ($B = -(Hess f)^{-1}\nabla f$) and steepest descent.



- (3) in general, equation $\|(B + \lambda id)^{-1}\| = \delta$ is difficult to solve for λ
→ several approximation strategies exist

dog-leg , double-dogleg step etc.

Simplest strategy:

Steihaug trick (1983)

- (1) CG iteration for $Bs = -\nabla f$ (resp. $\min_s \frac{1}{2}(s, Bs) + (\nabla f, s)$)
- (2) STOP , as soon as $\|s^i\| \geq \delta$ (a)
→ step $\Delta x = \alpha s^{i-1} + \beta s^i$ with $\alpha + \beta = 1$ and $\|\Delta x\| = \delta$
or $(s^i, Bs^i) \leq 0$ (s^i direction of negative curvature) (b)
→ s^i is descent direction and we use $\Delta x = \alpha \cdot s^i$ with $\|\Delta x\| = \delta$

Choice of the trust region δ^k

→ corresponding to the approximation quality of the model

$$\rho_k = \frac{f(x^k) - f(x^k + s^k)}{m_k(0) - m_k(s^k)} = \frac{\text{actual decrease}}{\text{predicted decrease}}$$

$\rho_k \approx 1 \Rightarrow$ o.k., δ_k can be enlarged

ρ_k small $\Rightarrow \delta_k$ should shrink

Model TR adaptation:

Choose $\bar{\delta} > 0, \delta_0 \in (0, \bar{\delta}), \eta \in [0, \frac{1}{4})$

↖ acceptance threshold

for $k = 0, 1, 2, \dots$

 determine s^k (approximatively, e.g., with Steihaug CG)

 evaluate ρ_k

 if $\rho_k < \frac{1}{4}$

$$\delta_{k+1} = \frac{1}{4} \|s_k\|$$

 else if $\rho_k > \frac{3}{4}$ and $\|s^k\| = \delta_k$

$$\delta_{k+1} = \min(2\delta_k, \bar{\delta})$$

 else

$$\delta_{k+1} = \delta_k$$

 if $\rho_k > \eta$

$$x^{k+1} = x^k + s^k$$

 else

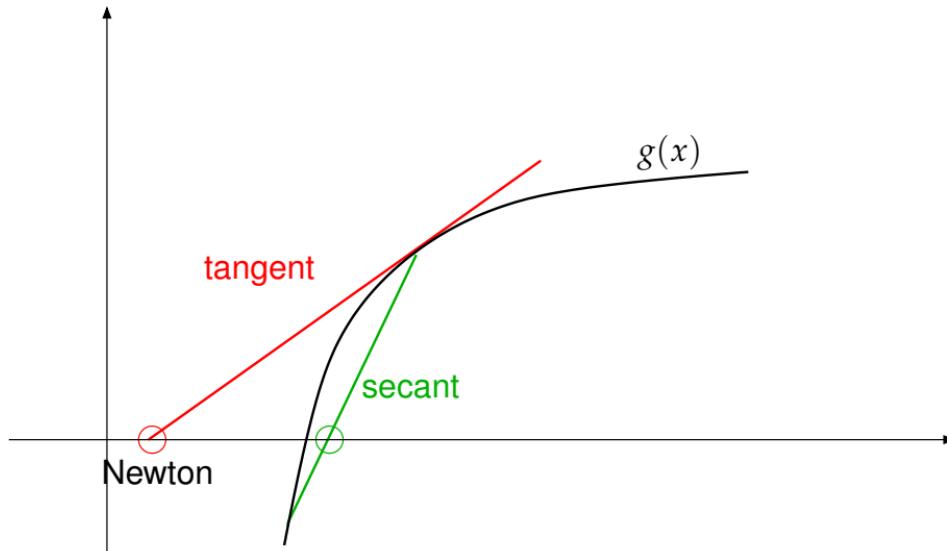
$$x^{k+1} = x^k$$

Properties of the algorithm:

- global convergence
- if TR inactive in the vicinity of the solution
 - local quadratic convergence, if $B_k = \text{Hess}f_k$

Quasi-Newton / secant method

Goal: approximate Hessian without explicit second derivatives.



Broyden formula

$$\begin{aligned} B_{k+1} \underbrace{(x^{k+1} - x^k)}_{p^k} &= \nabla f(x^{k+1}) - \nabla f(x^k) \quad (\text{SC}) \\ B_{k+1} p &= B_k p \quad \forall p \in (x^{k+1} - x^k)^\perp \quad \text{"no change"} \end{aligned}$$

$\Rightarrow n + n(n-1) = n^2$ conditions

\Rightarrow exercise B_{k+1} uniquely determined

$$\underbrace{B_{k+1}}_{U^B} = B_k + \frac{(q^k - B_k p^k)(p^k)^\top}{p^{k\top} p^k}$$

("dyadic product")

Lemma 3.14. (*minimal property*)

Let $\mathcal{M} := \{B : Bp^k = q^k\}$. The Broyden formula uniquely solves the minimization problem

$$\min_{B \in \mathcal{M}} \|B - B_k\|_F = \left(\sum_{i,j} (b_{ij} - b_{ij}^k)^2 \right)^{\frac{1}{2}}$$

\nwarrow Frobenius norm

Remark: Obviously also

$$\|U^B - B_k\|_2 \leq \|B - B_k\|_2$$

but: minimum for $\|\cdot\|_2$ may not be unique !

(Convergence) properties

Quasi Newton: Newton with Hessian updates

- globale convergence together with line search or trust region
- locally superlineare convergence (i.e., in the vicinity of the solution)
(superlinear: $\|x^{k+1} - \hat{x}\| \leq \alpha_k \|x^k - \hat{x}\|$, where $\alpha_k \rightarrow 0$ ($k \rightarrow \infty$))

Dennis-Moré characterization for proof of superlinear convergence

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(\hat{x})) \Delta x^k\|}{\|\Delta x^k\|} = 0$$

where \hat{x} minimum.

(cf. local contraction theorem from Numerik I

$$\kappa_k := \frac{\|B_k^{-1} (B_k - \nabla^2 f(x^k)) \Delta x^k\|}{\|\Delta x^k\|} \xrightarrow{(k \rightarrow \infty)} 0 \Leftrightarrow \text{superlinear convergence}$$

- Dennis-Moré characterization is satisfied for Broyden formulas (cf. Nocedal/Wright).
- Disadvantage: Formula not symmetric !
 - but useful in general root finding problems.

SR1: Broydens symmetric rank-1-update

$$\underbrace{B_{k+1}}_{U^{SR1}} = B_k + \frac{(q^k - B_k p^k)(q^k - B_k p^k)^\top}{(q^k - B_k p^k)^\top p^k}$$

Obviously $U^{SR1} \in \mathcal{M}$, i.e., satisfies (SC), and is unique in the class $B_{k+1} = B_k + \alpha v v^\top$ (exercise).

- convergence properties within quasi-Newton like Broyden
- Disadvantage: SR1 is not always pos. def., only if $(q^k - B_k p^k)^\top p^k > 0$, i.e. $q^k^\top p^k > p^k^\top B_k p^k$.

Remark: Broyden updates can be formulated also in Hilbert spaces (vgl. Sachs 1986, Griewank 1987 , Kelley/Sachs 1991).

Most important update formulas are of rank 2

Abbreviation $y = Bp$, where $p = x^{k+1} - x^k$, $q = \nabla f(x^{k+1}) - \nabla f(x^k)$

① *BFGS* (Broyden-Fletcher-Goldfarb-Shanno, 1970)

$$B^{BFGS} := B + \frac{qq^\top}{p^\top q} - \frac{yy^\top}{p^\top y} \quad (p^\top q > 0 \text{ assumed })$$

② *DFP* (Davidson-Fletcher-Powell)

$$B^{DFP} := B + \left(1 + \frac{p^\top y}{p^\top q}\right) \frac{qq^\top}{p^\top q} - \left(\frac{1}{p^\top q}\right) (qy^\top + yq^\top)$$

Remark:

- Broyden class:= { convex combination of *BFGS* and *DFP* }

- BFGS mostly used, since
 - it has advantages within inexact line search
 - it is numerically very robust

Lemma 3.15. Assume B pos. def. and $q^\top p > 0$. Then, both B^{BFGS} and B^{DFP} are positive definite.

Proof:

Remark: For \widehat{B} pos. def. at the solution, we obtain from secant condition and Dennis-Moré characterization

$$p^\top q \approx p^\top \widehat{B} p > 0$$

$\Rightarrow p^\top q > 0$ is satisfied in the vicinity of the solution

Convergence properties:

- Dennis-Moré characterization can be shown for *BFGS* and *DFP*
 \Rightarrow locally superlinear convergence
- B pos. def. \Rightarrow always descent direction
 \Rightarrow globalization possible with line search or trust region

Updates of the inverse

Sherman-Morrison-Woodbury formula gives updates of the inverse Hessian approximation.
Let $M_{k+1} := B_{k+1}^{-1}$ and $M_k := B_k^{-1}$

① Broyden:

$$M_{k+1}^B = M_k + \frac{(p^k - M_k q^k)(p^k)^\top M_k}{(p^k)^\top M_k q^k}, \quad \text{if } (p^k)^\top M_k q^k \neq 0$$

② SR1:

$$M_{k+1}^{SR1} = M_k + \frac{(p^k - M_k q^k)(p^k - M_k q^k)^\top}{(p^k - M_k q^k)^\top q^k}$$

\Rightarrow same formula as for B_{k+1}^{SR1} , only $p^k \leftrightarrow q^k$ swapped

Theorem 3.16. *BFGS and DFP are invers dual, i.e., with the notation*

$$B^{BFGS} =: U^{BFGS}(B, q, p), \quad B^{DFP} =: U^{DFP}(B, q, p) \text{ holds}$$

$$M_{k+1}^{BFGS} = U^{DFP}(M_k, p^k, q^k), \quad M_{k+1}^{DFP} = U^{BFGS}(M_k, p^k, q^k)$$

Proof: Exercise

Theorem 3.17. (Nazareth 1979) For quadratic objective functions, CG and quasi-Newton with BFGS update are identical.

Proof: L. Nazareth: a relationship between the BFGS and CG algorithms and its implications for new algorithms, SIAM J. Numer. Anal. 16, 794-800/1979,)

Corollary 3.18. Quasi-Newton methods (in particular BFGS) terminate at last after $\dim(X)$ steps, if the objective function is quadratic

Limited memory updates

Idea:

Use for B_{k+1} instead of

$$B_{k+1} = \underbrace{U \circ \dots \circ U}_{k+1 \text{ mal}} \circ B_0$$

only

$$B_{k+1} = \underbrace{U \circ \dots \circ U}_m \circ B_k^0$$

typically $3 \leq m \leq 20$

(Often $B_k^0 = I$, but it may also vary with iteration number k).

Remark:

- Update can be efficiently implemented in two loop (N/W)
- Advantage: only small storage necessary → often no convergence deterioration in comparison with full updates
- Choice of $M_k^0 := \gamma_k I$ with

$$\gamma_k = \frac{(p^k)^\top q^k}{(q^k)^\top q^k} \approx \frac{(q^k)^\top \text{Hess}^{-1} q^k}{(q^k)^\top q^k}$$

" scaling factor " (in current direction)

3.5 Gauß-Newton methods for nonlinear least squares

Example: Gauß-Ceres Problem

$\alpha(t, q)$: Model function for position of Ceres at time t → depends on parameters
 $q \in \mathbb{R}^n$

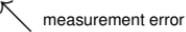
Data $(t_i, \bar{x}_i), i = 1, \dots, N$ measurements

Goal: determine unknown parameters q

1. Ansatz: choose n measurements and hope that they are by $q \in \mathbb{R}^n$ uniquely determined (direct inversion).

Problem: measurement error !

$$\bar{x}_i = \alpha(t_i, \hat{q}) + \varepsilon_i$$



2. Ansatz: (ML estimation)

Measurement error unknown:

$$\rightarrow \text{statistical assumptions: } \varepsilon_i \sim N(0, \sigma_i^2)$$

- normally distributed around expected value 0
- all measurement errors distributed independently.

Excursion: Maximum-Likelihood-Estimation

$\bar{\alpha}_i$ are understood as realizations of the random variable X_i with $X_i \sim N(\alpha(t_i, q), \sigma_i^2)$

With the density of the normal distribution, it is now possible to calculate probabilities, e.g. for $A_i \subset \mathbb{R}$

$$P(A_i) = \int_A \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(\bar{\alpha}_i - \alpha(t_i, q))^2}{2\sigma_i^2}} d\bar{\alpha}_i$$

and in independently distributed X_i for $A = A_1 \times \dots \times A_N$

$$P(A) = \int_A \underbrace{\prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^N \frac{(\bar{\alpha}_i - \alpha(t_i, q))^2}{\sigma_i^2}\right)}_{f(\alpha)} d\bar{\alpha}_1 \dots d\bar{\alpha}_N$$

Maximum-Likelihood-Principle: For given realizations, use the value q as estimation, where the common density $f(\alpha)$ becomes the maximum.

So q is determined from the optimization problem

$$\hat{q} = \arg \max_q \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^N \frac{(\bar{\alpha}_i - \alpha(t_i, q))^2}{\sigma_i^2}\right)$$

Since the logarithm is monotone, equivalency follows

$$\Leftrightarrow \hat{q} = \arg \max_q - \underbrace{\sum_{i=1}^N \log(\sqrt{2\pi}\sigma_i)}_{\text{independent of } q} - \frac{1}{2} \sum_{i=1}^N \frac{(\bar{\alpha}_i - \alpha(t_i, q))^2}{\sigma_i^2}$$

$$\Leftrightarrow \hat{q} = \arg \min_q \frac{1}{2} \sum_{i=1}^N \frac{(\bar{\alpha}_i - \alpha(t_i, q))^2}{\sigma_i^2} \rightarrow \text{least squares}$$

"ℓ₂ – estimation"

Observation: Other assumptions about the error distribution lead to other ML-estimators

rectangle distribution : $\hat{q} = \arg \min_q \max_i |\bar{\alpha}_i - \alpha(t_i, q)|$ ℓ_∞ - Sch"atzer

Laplace distribution : $\hat{q} = \arg \min_q \sum_{i=1}^N |\bar{\alpha}_i - \alpha(t_i, q)|$ ℓ_1 - estimator

→ Statistics, further investigations possible for the experimental design,
Sensitivity analysis etc.

Now: Abstract formulation of the ℓ_2 estimation problem based on a normal distribution assumption of the occurring model error:

$F : X \rightarrow Y$ with Scp $(\cdot, \cdot)_X$ bzw. $(\cdot, \cdot)_Y$
 $x \mapsto F(x)$

and F Frechet-diffb

The central output-least-squares problem is now

$$\min_x f(x) := \frac{1}{2} \|F(x)\|_Y^2$$

Let $J(x) := \frac{\partial F}{\partial x}(x)$ (Frechét-derivative)

Then we calculate: $\nabla f(x)$

$$\left. \frac{d}{dt} \right|_{t=0} f(x + th) = (F(x), J(x)h)_Y = (J(x)^{ad} F(x), h)_X$$

$$\Rightarrow \nabla f(x) = J(x)^{ad} F(x)$$

$$\text{und } \text{Hess } f(x) = J(x)^{ad} J(x) + D J(x)^{ad} F(x)$$

↑
ist in solution near 0

Lemma 3.19. $\dim(X) < \infty$ (oder $\|J(\hat{x})\| \geq c$). Then, $\text{Hess } f(\hat{x})$ p.d., if $f(\hat{x})$ is sufficiently small and $J(x)$ injective.

Proof: obvious ✓

Idea of Gauß-Newton-Method:

ignore the term $DJ(x)^{ad}f(x)$ of Newton Method

Algorithm: x^0 Start

$$B_k := J(x^k)^{ad} J(x)$$

$$\text{Solve } B_k \Delta x = -J(x^k)^{ad} F(x^k) \quad \textcircled{*}$$

Linesearch $\longrightarrow \tau$

$$\text{update } x^{k+1} = x^k + \tau \Delta x$$

Lemma 3.20. step $\textcircled{*}$ corresponds to the normal equations for linear F and is equivalent to

$$\min_{\Delta x} \frac{1}{2} \|F(x^k) + J(x^k)\Delta x\|_Y^2 \quad \text{"Newton" resp. "Linearisation under the norm"}$$

Proof:

Convergence: locally linear but with good contraction rate, if

$$F(\hat{x}) \text{ small !}$$

\Rightarrow almost quadratic only with information 1st order

Levenberg-Marquardt-Method:

$$B_k := J(x^k)^{ad} J(x) + \lambda^k I$$

$$\text{solve } B_k \Delta x = -J(x^k)^{ad} F(x^k) \quad \textcircled{*}$$

$$\text{define } x^{k+1} = x^k + \Delta x$$

So instead of line-search choose $B_k = J_k^{ad}J + \lambda^k I$ with $\lambda^k \in \mathbb{R}$

① according to trust-region method (ideal for small problems)

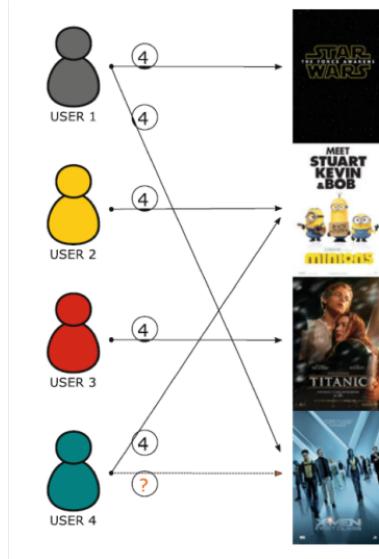
② original:

If $f(x^{k+1}) > f(x^k)$, choose $\lambda^{k+1} = 2 \cdot \lambda^k$, reject step
otherwise, accept step and choose $\lambda^{k+1} = \frac{1}{2}\lambda^k$

- background for ② : $(J^{ad} \cdot J + \lambda I)^{-1} J^{ad} F$
is descent direction for λ sufficiently large!
- heuristic method
- frequently found in the engineering literature (BFGS better at large-residual problems)

3.6 Alternating least squares — the Netflix challenge

Recommender systems like Netflix or Amazon try to recommend items based on past ratings.



	I_1	I_2	I_3	I_4
U_1	4	1	1	4
U_2	1	4	2	0
U_3	2	1	4	5
U_4	1	4	1	?

Background philosophy: Both, users and items live in some low-dimensional space describing their properties. Recommend a movie (an item) based on its proximity to the user in the latent space

Famous **Netflix Prize 2009** for the best competing recommender system.

Problem statement:

Rating matrix $R \in \mathbb{R}^{m \times n}$, where $m = \text{Number of users}$ and $n = \text{number of items}$ and, e.g., $r_{2,9} = 3$ means user 2 gives item 9 a three stars rating. The matrix R is assumed to have low rank k and is to be identified from a typically low number of available ratings $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$. The low rank is expressed by the assumption that

$$R = UV^\top, \quad U \in \mathbb{R}^{m \times k}, \quad V \in \mathbb{R}^{n \times k}$$

for some small $k \in \mathbb{N}$. The approximation problem is then expressed for given data R, Ω as

$$\boxed{\min_{U,V} \|P_\Omega(R) - P_\Omega(UV^\top)\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2)}, \quad \lambda > 0 \quad (3.1)$$

with

$$P_\Omega(A)_{ij} := \begin{cases} a_{ij} & , \text{if } (i,j) \in \Omega \\ 0 & , \text{else} \end{cases}$$

First idea: problem shares similarities with singular value decomposition (SVD). Due to SVD, there are orthogonal matrices $\tilde{U} \in O(m)$, $\tilde{V} \in O(n)$ with

$$R = \tilde{U}\Sigma\tilde{V}^\top, \quad \Sigma = \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & \vdots & & \\ & \vdots & & & \vdots & \\ & \cdots & 0 & \cdots & \cdots & 0 & \cdots \\ & & 0 & & & & \\ & & & \vdots & & & \vdots \end{bmatrix}, \quad \sigma_i > 0, \forall i$$

However, \tilde{U} , \tilde{V} from SVD satisfy the additional constraint of orthogonality, which is not required in problem (3.1). Furthermore, we do not know all entries in R (only those contained in Ω). Thus, an SVD might be a good starting point, but not the solution.

Observation: problem (3.1) is not convex due to the product UV^\top . However, for fixed U , the problem is convex in V and vice versa—they are even just quadratic problems, with standard solution approaches available.

We rewrite the objective in a way, which is more convenient. We use the following matrices:

$$W_i^U = \text{diag}(\chi_{\Omega}(i, 1), \dots, \chi_{\Omega}(i, n)) \in \mathbb{R}^{n \times n}$$

$$W_j^V = \text{diag}(\chi_{\Omega}(1, j), \dots, \chi_{\Omega}(m, j)) \in \mathbb{R}^{m \times m}$$

Therefore, the objective can be written in different forms:

$$\|P_{\Omega}(R) - P_{\Omega}(UV^\top)\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) \quad (3.2)$$

$$= \sum_{i=1}^m \left[(R_{i,\bullet} - U_{i,\bullet}V^\top) W_i^U (R_{i,\bullet} - U_{i,\bullet}V^\top)^\top + \lambda U_{i,\bullet} U_{i,\bullet}^\top \right] + \lambda \|V\|_F^2 \quad (3.3)$$

$$= \sum_{j=1}^n \left[(R_{\bullet,j} - UV_{j,\bullet}^\top)^\top W_j^V (R_{\bullet,j} - UV_{j,\bullet}^\top) + \lambda V_{j,\bullet} V_{j,\bullet}^\top \right] + \lambda \|U\|_F^2 \quad (3.4)$$

(3.5)

Thus, differentiating (3.3) with respect to U gives the following necessary condition ($0 = \partial/\partial U_{i,\bullet}$):

$$0 = (V^\top W_i^U V + \lambda I) U_{i,\bullet}^\top - W_i^U R_{i,\bullet}^\top, \quad \forall i = 1, \dots, m$$

and doing the same with (3.4) with respect to V gives

$$0 = (U^\top W_j^V U + \lambda I) V_{j,\bullet}^\top - W_j^V R_{\bullet,j}^\top, \quad \forall j = 1, \dots, n$$

Algorithmic idea: **Alternating least squares method**

For given k determine $U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}$ from the iteration

do

$$U_{k+1} := \underset{U}{\operatorname{argmin}} \|P_\Omega(R) - P_\Omega(UV_k^\top)\|_F^2 + \lambda(\|U\|_F^2 + \|V_k\|_F^2)$$

$$V_{k+1} := \underset{V}{\operatorname{argmin}} \|P_\Omega(R) - P_\Omega(U_{k+1}V^\top)\|_F^2 + \lambda(\|U_{k+1}\|_F^2 + \|V\|_F^2)$$

until convergence

This is a special case of (blockwise) coordinate descent. Another example for a coordinate descent method is the Gauß-Seidel algorithm for linear equations with symmetric and positive definite system matrix.

Theorem 3.21 (2-block coordinate descent). *We consider the optimization problem*

$$\min_{(x,y) \in \mathcal{X} \times \mathcal{Y}} f(x,y)$$

where $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is differentiable and \mathcal{X}, \mathcal{Y} are convex and closed subsets. Furthermore, we assume that the global optimization problems $\min_{x \in \mathcal{X}} f(x, \bar{y})$ and $\min_{y \in \mathcal{Y}} f(\bar{x}, y)$ are well-defined and possess a solution for any $\bar{x} \in \mathcal{X}, \bar{y} \in \mathcal{Y}$. Then, the 2-block coordinate descent algorithm

for $k = 0, 1, \dots$ **do**

$$x_{k+1} := \underset{x}{\operatorname{argmin}} f(x, y_k)$$

$$y_{k+1} := \underset{y}{\operatorname{argmin}} f(x_{k+1}, y)$$

produces a sequence $\{(x_k, y_k)\}_{k=0}^{\infty}$ with the following properties

- every limit point (\hat{x}, \hat{y}) is a stationary point for f
- if $N_0 := \{(x, y) \mid f(x, y) \leq f(x_0, y_0)\}$ is compact, then there exists at least one limit point

Proof. a) The proof outlines the proof of proposition 2.7.1 in [Ber99] □

Remarks:

- The theorem shows convergence of the alternating least squares method, once k is fixed.
- The algorithm in theorem 3.21 can be generalized to m -block iterations, which is the original formulation of proposition 2.7.1 in [Ber99]. Here it is downsized to 2-block-iterations for ease of presentation. A different approach can also be found in [GS99].
- ALS is also the de-facto standard algorithm for tensor completion as generalization of the matrix completion discussed above.
- It can be shown [JNS13] under assumptions on Ω (uniform sampling) and R (incoherence, most entries similar in magnitude) that the ALS algorithm generates U and V such that $\|R - UV^\top\|_F \leq \epsilon$ after $\mathcal{O}(\log(1/\epsilon))$ steps.
- note that the algorithm already treats optimization problems with constraints, if the constraints lead to convex sets \mathcal{X} and \mathcal{Y}
- An exemplaric implementation can be found at https://github.com/danielnee/Notebooks/blob/master/ALS/ALS_Explicit.ipynb
- ALS is surprisingly fast in particular in the first iterations and easily beats Quasi-Newton methods as justified experimentally in recent Bachelor and Master theses.

The problem of a good initialization is still left. Here, we can use the SVD in the following way:

Let us fill all entries in r_{ij} of $R = (r_{ij})_{ij}$ with zero, for which $(i, j) \notin \Omega$. Then we can start with a completely defined matrix $R \in \mathbb{R}^{m \times n}$, which possesses an SVD $R = \tilde{U}\Sigma\tilde{V}^\top$. We truncate/omit all singular values with $\sigma_i < \theta$ for a threshold $\theta > 0$. Now, we set $k := \max\{\ell \in \mathbb{N} \mid \sigma_\ell \geq \theta, \ell \leq r\}$ and use

$$U_0 := \tilde{U}_{1,\dots,k} \operatorname{diag}(\sqrt{\sigma_1}, \dots, \sqrt{\sigma_k})$$
$$V_0 := \tilde{V}_{1,\dots,k} \operatorname{diag}(\sqrt{\sigma_1}, \dots, \sqrt{\sigma_k})$$

as starting matrices for ALS, where, e.g., $\tilde{U}_{1,\dots,k}$ means the first k columns of the matrix \tilde{U} .

4 Randomized Algorithms for Artificial Neural Networks (ANN)

4.1 Interlude: essentials of back propagation

Simple first example: Consider concatenated differentiable functions $f \circ g(x)$ as:

$$f(z) \text{ , where } z = g(x) = 0$$

Thus, z is a function of x , namely $z(x) = g(x)$. Thus,

$$f \circ g(x) \equiv \mathcal{L}(x, \lambda) := f(z(x)) + \lambda^\top (z(x) - g(x)), \text{ for any } \lambda$$

Now,

$$\begin{aligned} \frac{d(f \circ g)(x)}{dx} &= \frac{\partial \mathcal{L}(x, \lambda)}{\partial x} = \frac{df}{dz} \frac{dz}{dx} + \lambda^\top \left(\frac{dz}{dx} - \frac{dg}{dx} \right) \\ &= \left(\frac{df}{dz} + \lambda^\top \right) \frac{dz}{dx} - \lambda^\top \frac{dg}{dx} \end{aligned}$$

Thus, $d(f \circ g)/dx$ can be determined in two steps

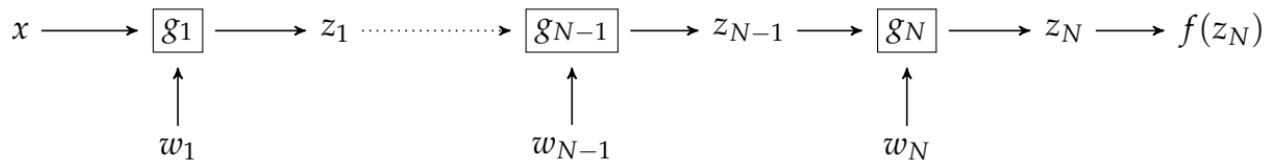
$$(1) \quad \lambda := - \left(\frac{df}{dz}(z(x)) \right)^\top$$

$$(2) \quad \frac{d(f \circ g)(x)}{dx} = -\lambda^\top \frac{dg}{dx}(x)$$

Abstract back propagation: We consider the concatenation

$$\begin{aligned} h(w_N, \dots, w_1, x) &:= f(g_N(w_N, (g_{N-1}(w_{N-1}, (\dots, g_1(w_1, x)) \dots)))) \\ &= f \circ g_N(w_N, \cdot) \circ g_{N-1}(w_{N-1}, \cdot) \circ \dots \circ g_1(w_1, x) \end{aligned}$$

illustrated by the diagramm:



Again, this can be written in the fashion

$$h(w_N, \dots, w_1, x) = f(z_N), \text{ where } z_k - g_k(w_k, z_{k-1}) = 0, k = 1, \dots, N, z_0 := x$$

Again

$$h(w_N, \dots, w_1, x) \equiv \mathcal{L}(w, x, \lambda) := f(z_N) + \sum_{k=1}^N \lambda_k^\top (z_k - g_k(w_k, z_{k-1})), \text{ for any } \lambda_k$$

Thus

$$\begin{aligned}\frac{dh}{dw_i} &= \frac{df}{dz} \frac{dz_N}{dw_i} + \sum_{k=i}^N \lambda_k^\top \frac{dz_k}{dw_i} - \sum_{k=i+1}^N \lambda_k^\top \frac{\partial g_k}{\partial z} \frac{\partial z_{k-1}}{\partial w_i} - \lambda_i^\top \frac{\partial g_i}{\partial w_i} \\ &= \frac{df}{dz} \frac{dz_N}{dw_i} + \sum_{k=i}^N \lambda_k^\top \frac{dz_k}{dw_i} - \sum_{k=i}^{N-1} \lambda_{k+1}^\top \frac{\partial g_{k+1}}{\partial z} \frac{dz_k}{dw_i} - \lambda_i^\top \frac{\partial g_i}{\partial w_i} \\ &= \left(\frac{df}{dz} + \lambda_N^\top \right) \frac{dz_N}{dw_i} + \sum_{k=i}^{N-1} \left(\lambda_k^\top - \lambda_{k+1}^\top \frac{\partial g_{k+1}}{\partial z} \right) \frac{dz_k}{dw_i} - \lambda_i^\top \frac{\partial g_i}{\partial w_i}\end{aligned}$$

Thus, if the λ 's satisfy the recursion

$$\lambda_N = - \left(\frac{df}{dz}(z_N) \right)^\top, \quad \lambda_k = \left(\frac{\partial g_{k+1}}{\partial z}(w_{k+1} z_k) \right)^\top \lambda_{k+1}, \quad k = N-1, N-2, \dots, i$$

we achieve the expression

$$\frac{dh}{dw_i}(w_N, \dots, w_1, x) = -\lambda_i^\top \frac{\partial g_i}{\partial w_i}(w_i, z_{i-1})$$

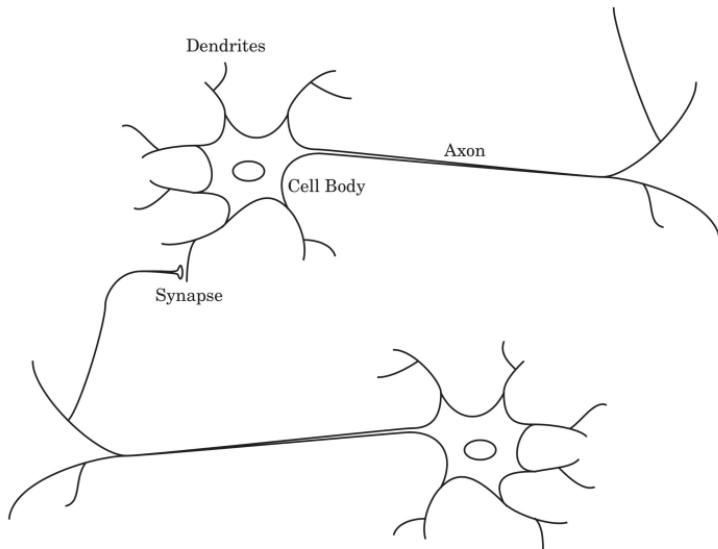
Remark: The technique presented is an example of a fairly general principle for the derivation of iterations towards complicated derivatives. This principle is used in the backward mode of automatic differentiation and in the derivation of adjoint equations for optimization problems involving differential equations.

4.2 Optimization aspects of ANN

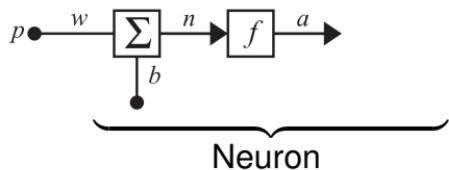
As a playful motivation, visit <https://teachablemachine.withgoogle.com>

This part of the lecture is closely related to the book [HDBJ14], from which also several graphics have been taken.

Biological inspiration:



Single input neuron:



p input signal ($\in \mathbb{R}$)

w weight ($\in \mathbb{R}$)

b bias (perturbation) ($\in \mathbb{R}$)

n neuron signal ($\in \mathbb{R}$)

f transfer function ($: \mathbb{R} \rightarrow \mathbb{R}$)

a action/output signal ($\in \mathbb{R}$)

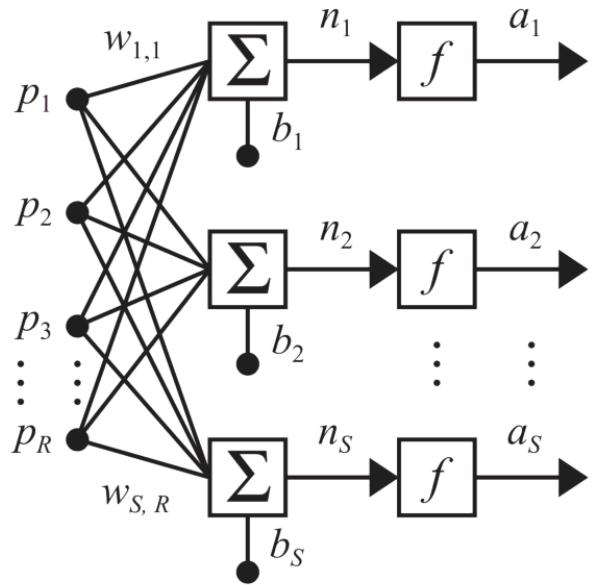
Examples of standard transfer functions are

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1 \quad \text{neuron with max } n$ $a = 0 \quad \text{all other neurons}$		compet

logistic function

ReLU: rectified linear

Multiple neurons and inputs:



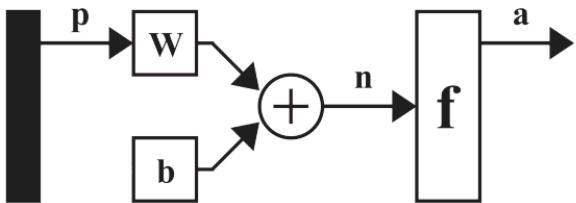
$p \in \mathbb{R}^R, n, b, a \in \mathbb{R}^S, W \in \mathbb{R}^{S \times R}:$

$$a_i = f\left(\sum_{j=1}^R w_{ij} p_j + b_i\right), i = 1, \dots, S$$

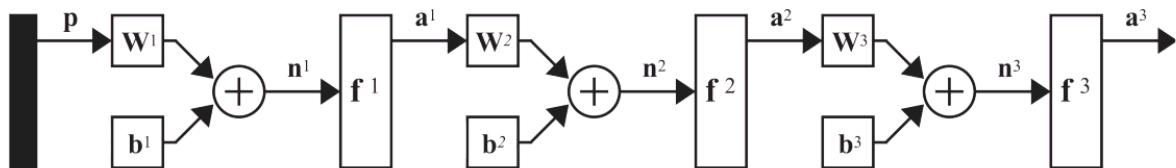
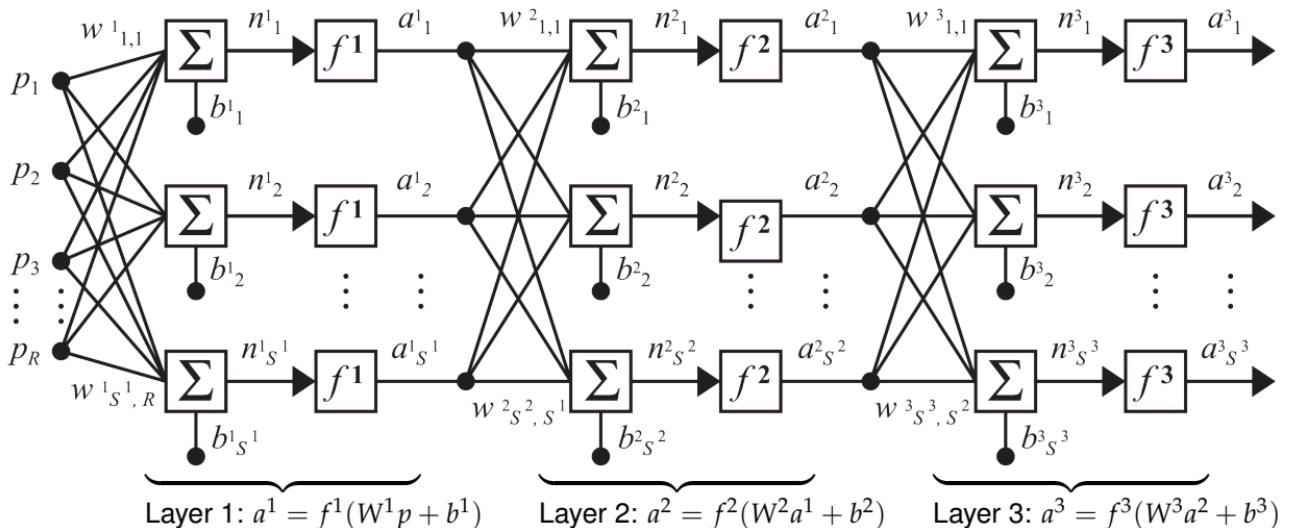
abbreviated as

$$a = f(Wp + b)$$

or with compact symbols



This is a one-layer neural network. Multi layers can be concatenated (below):



and in concatenated form

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) \quad (4.1)$$

Essentially, an ANN is a more or less intelligent way of interpolating functions.

Supervised learning:

Determine weights W^i and biases b^i such that the mapping defined by N layers of an artificial NN

$$p \xrightarrow{\text{ANN}} a^N =: \text{ANN}(p; W^1, \dots, W^N, b^1, \dots, b^N)$$

matches a training set $\{(p^i, t^i) \in \mathbb{R}^R \times \mathbb{R}^S \mid i = 1, \dots, q\}$ as close as possible, i.e.

$$\text{ANN}(p^i; W^1, \dots, W^N, b^1, \dots, b^N) \approx t^i, i = 1, \dots, q$$

If we summarize the weights W^i and biases b^i with the vector x , we achieve the loss function to be minimized

$$\min_x \ell(x) = \frac{1}{2} \sum_{i=1}^q \|\text{ANN}(p^i; x) - t^i\|_2^2 + R(x)$$

where $R(x)$ denotes a regularization function, which is sometimes appropriate.

We investigate the l_2 gradient of the core part of the objective, i.e.,

$\bar{\ell}(x) := \frac{1}{2} \sum_{i=1}^q \|\text{ANN}(p^i; x) - t^i\|_2^2$ and observe for the example $N = 3$ (equation (4.1)) and therefore specifically (not all arguments in the functions below are written explicitly)

$$\bar{\ell}(x) = \frac{1}{2} \sum_{i=1}^q \|f^3(W^3 f^2(W^2 f^1(W^1 p^i + b^1) + b^2) + b^3) - t^i\|_2^2$$

$$\frac{\partial \bar{\ell}}{\partial w_{kl}^3} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (D_k f^3) f_l^2(W^2f^1(W^1p^i + b^1) + b^2)$$

$$\frac{\partial \bar{\ell}}{\partial b^3} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (Df^3)$$

$$\frac{\partial \bar{\ell}}{\partial w_{kl}^2} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (Df^3) W^3(Df_k^2) f_l^1(W^1p^i + b^1)$$

$$\frac{\partial \bar{\ell}}{\partial b^2} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (Df^3) W^3(Df^2)$$

$$\frac{\partial \bar{\ell}}{\partial w_{kl}^1} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (Df^3) W^3(Df^2) W^2(Df_k^1) p^l$$

$$\frac{\partial \bar{\ell}}{\partial b^1} = \sum_{i=1}^q \left[f^3(W^3f^2(W^2f^1(W^1p^i + b^1) + b^2) + b^3) - t^i \right]^\top (Df^3) W^3(Df^2) W^2(Df^1)$$

Therefore, terms occurring in $\frac{\partial \bar{\ell}}{\partial (W^3, b^3)}$ are contained also in $\frac{\partial \bar{\ell}}{\partial (W^2, b^2)}$ and in $\frac{\partial \bar{\ell}}{\partial (W^1, b^1)}$ and so on, which enables a recursive computation of all those partial derivatives, which is called "backward propagation". In contrast to that, just the evaluation of the ANN is called "forward propagation". The whole computation is expressed in the following algorithm

initialize $\bar{\ell} = 0, \nabla_{W_m} \bar{\ell} = 0, \nabla_{b_m} \bar{\ell} = 0, m = 1, \dots, N$

for $i \in \{1, \dots, q\}$ **do** % (0) sum over all information available

(1) forward propagation

$$a^0 := p^i$$

for $m = 0, 1, \dots, N - 1$ **do**

$$a^{m+1} := f^{m+1}(W^{m+1}a^m + b^{m+1})$$

$$\bar{\ell} += \frac{1}{2}\|a^N - t^i\|_2^2$$

(2) backward propagation

$$s^N := (Df^N)^\top [a^N - t^i]$$

$$\nabla_{W_N} \bar{\ell} += s^N (a^{N-1})^\top$$

$$\nabla_{b_N} \bar{\ell} += s^N$$

for $m = N - 1, N - 2, \dots, 1$ **do**

$$s^m := (Df^m)^\top (W^{m+1})^\top s^{m+1}$$

$$\nabla_{W_m} \bar{\ell} += s^m (a^{m-1})^\top$$

$$\nabla_{b_m} \bar{\ell} += s^m$$

We note that the forward (1) and backward (2) loops are strictly sequential, but the **outer loop (0) can be executed in parallel**, which is the reason, why parallel architectures (in particular GPUs) are used quite heavily.

Further note that the derivative matrices Df are diagonal matrices. A steepest descent method based on this gradient information can be written in the form:

for $m = 1, \dots, N$ **do**

$$\begin{aligned} W_{k+1}^m &:= W_k^m - \alpha \nabla_{W_m} \ell \\ b_{k+1}^m &:= b_k^m - \alpha \nabla_{b_m} \ell \end{aligned}$$

Where the up-to now so-called line-search parameter α is called in this context "learning rate".

Important variant:

Stochastic gradient method: the loop (0) is not executed for the whole index set $I := \{1, \dots, q\}$ (full batch) but only for a randomly chosen subset (batch) $\tilde{I} \subset I$, where even $\#\tilde{I} = 1$ is possible and typical. The theoretical basis for this approach is discussed below.

If you want to try out neural networks, you should refer to playground.tensorflow.org

Central tool for supervised learning: stochastic gradient method for objective functions in the form of an expected value

$$\min f(x) := \frac{1}{m} \sum_{i=1}^m f_i(x) =: \mathbb{E}_i[f_i(x)]$$

Apparently applies to sufficiently smooth functions f_i

$$\nabla f(x) = \frac{1}{m} \sum_{i=1}^m \nabla f_i(x) = \mathbb{E}_i[\nabla f_i(x)]$$

Stochastic gradient methods use only a single (or the average of a few) random summands as a gradient approximation, so they can be formally written as

$$x^{k+1} = x^k - \alpha_k \nabla f_{i_k}(x^k), \quad i_k \in \{1, 2, \dots, m\} \text{ randomly}$$

Thus, this iteration defines a stochastic process. Each iterated x^k depends on any previously drawn i_{k-1}, i_{k-2}, \dots . We define the total expected value of a function φ of x^k as

$$\mathbb{E}[\varphi(x^k)] = \mathbb{E}_{i_1} \mathbb{E}_{i_2} \dots \mathbb{E}_{i_{k-1}} [\varphi(x^k)]$$

We now prove a convergence statement with arguments that are for the most part from the publication [NJLS09] (downloadable from the Uni-Trier-Network).

Theorem 4.1. In the above setting, be the objective function strictly convex in the sense that there is a $\mu > 0$

$$f(z) \geq f(x) + \nabla f(x)^\top (z - x) + \frac{1}{2}\mu\|z - x\|_2^2, \quad \forall x, z$$

and let x^* be the solution of $\min_x f(x)$. In addition, we assume that the gradient is globally constrained, that is, there is a $M < \infty$

$$\left\| \frac{1}{m} \sum_{i=1}^m \nabla f_i(x) \right\|_2 \leq M, \quad \forall x.$$

Then, the above stochastic gradient method converges with the step size sequence $\alpha_k = 1/(k\mu)$ sublinearly in the following sense:

$$\mathbb{E} \left[\|x^k - x^*\|_2^2 \right] = \mathcal{O} \left(\frac{1}{k} \right).$$

Proof:

Remarks:

- The stepsize sequence $\{\alpha_k\}_{k=1}^{\infty}$ chosen in theorem 4.1 is typical for methods with provable convergence. In [BT00], stochastic steepest descent is treated as a perturbed version of standard steepest descent. With this approach, the authors show convergence, if the step size sequence satisfies the following two conditions:

$$\sum_{k=1}^{\infty} \alpha_k = \infty, \text{ and } \sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

- Nevertheless, stochastic gradient methods are often also performed with constant step-sizes. More information about the whole range of topics can be obtained from the current review article [BCN18]. The following picture from [BCN18] compares the typical overall effort of stochastic gradient methods vs limited memory BFGS.

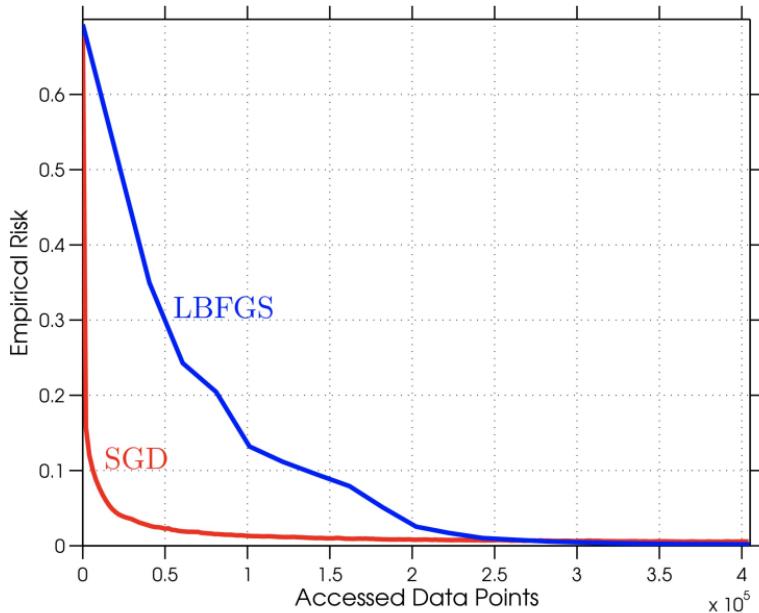
- The momentum methods mentioned in section 3.2 can be used in an obvious stochastic manner. In particular, the stochastic Adam (Adaptive Momentum) is used very often with good success, where $\nabla f(x)$ is replaced by $\nabla \tilde{f}(x) := \frac{1}{\#\tilde{I}} \sum_{j \in \tilde{I}} \nabla f_j(x^k)$ such that

$$x_i^{k+1} = x_i^k - \alpha_k \frac{\nu_k}{\sqrt{s_k + \epsilon}} \nabla \tilde{f}(x^k)_i, i = 1, \dots, n$$

$$\text{where } \nu_k = \beta_1 \nu_{k-1} - (1 - \beta_1) \nabla \tilde{f}(x^k)_i, \beta_1 \approx 0.9$$

$$s_k = \beta_2 s_{k-1} - (1 - \beta_2) \nabla \tilde{f}(x^k)_i^2, \beta_2 \approx 0.99, \epsilon \approx 10^{-10}$$

An overview can be obtained from [Rud17].



- Note further the awkward terminology in machine learning:

“learning rate η ” \iff “stepsize α ”

5 Linear quadratic programming

Example: 1. Portfolio Optimization

n investment opportunities with

expected return $r_i, i = 1, \dots, n$

common risk modeled by variance-covariance matrix $B \in \mathbb{R}^{n \times n}$

$x \in \mathbb{R}^n, \sum_{i=1}^n x_i = 1$ Portfolio , x_i = share of investment i

$$\Rightarrow \text{total profit} = \sum_{i=1}^n x_i r_i = x^\top r$$

$$\text{total risk} = x^\top B x$$

utility function: $f(x) = x^\top r - \kappa x^\top B x$

κ : Risk tolerance parameters \rightarrow customer specific

Optimal portfolio characterized by

$$\max_x x^\top r - \kappa x^\top B x$$

$$\text{s.t. } \sum_{i=1}^n x_i = 1 \quad (\text{equality constraints})$$

$$x \geq 0 \quad (\text{inequality constraints})$$

Example: 2. LASSO (least absolute shrinkage and selection operator):

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1, \quad \lambda > 0$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$.

Lack of differentiability of $\|\cdot\|_1$ as challenge. Reformulations:

5.1 QP with equality constraints

$$\begin{aligned} \min & \frac{1}{2}x^\top Bx + b^\top x \\ \text{s.t.} & Cx + c = 0 \end{aligned}$$

$$B \in \mathbb{R}^{n \times n} \text{ symmetric}$$
$$C \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^n, c \in \mathbb{R}^m$$

Recall: theory from section **2**

5.1.1 QP with positive definite quadratic form

B pos. definite on complete \mathbb{R}^n

necessary conditions:

$$\begin{array}{l} \textcircled{i} \\ \textcircled{ii} \end{array} \left[\begin{array}{cc} B & C^\top \\ C & 0 \end{array} \right] \left(\begin{array}{c} x \\ \lambda \end{array} \right) = \left(\begin{array}{c} -b \\ -c \end{array} \right)$$

B p.d. $\Rightarrow B$ invertible \rightarrow Block Gauß elimination

Application of steepest descent method to range space problem gives

$$\begin{aligned}\text{Iteration } \lambda^{k+1} &= \lambda^k - \tau(A\lambda^k - \alpha) = \\ &= \lambda^k - \tau[CB^{-1}C^\top\lambda^k + CB^{-1}b - c] = \\ &= \lambda^k - \tau C B^{-1}(\underbrace{C^\top\lambda^k + b}_{x^{k+1}}) + \tau c\end{aligned}$$

Thus, we obtain the following iteration in both variables

$$\begin{aligned}x^{k+1} &= -B^{-1}(C^\top\lambda^k + b) = x^k - B^{-1}(Bx^k + C^\top\lambda^k - b) \\ \lambda^{k+1} &= \lambda^k + \tau(Cx^{k+1} + c)\end{aligned}$$

↖ approximation gives inexact Uzawa

Definition 5.1. Iteration above is called Uzawa-Iteration (Hirofumi Uzawa, 1958, Japanese economist) and can be interpreted as a steepest descent method in range space.

Remark:

- Uzawa is mainly used in PDE Saddle point problems (Stokes)
→ is convergent, if τ is small enough

5.1.2 QP with indefinite quadratic form

B only pos.def. on $\ker(C) \setminus \{0\}$

Lemma 5.2. *For any $C \in \mathbb{R}^{m \times n}, m \leq n$ with $\text{rang}(C) = m$ there is a permutation matrix $P \in \mathbb{R}^{n \times n}$ with*

$$CP = [C_1 : C_2], \quad C_1 \in Gl(m)$$

Proof:

Lemma 5.2 allows to assume (wlog, modulo permutation) a variable splitting $x = P \begin{pmatrix} y \\ z \end{pmatrix}$ with $y \in \mathbb{R}^m, z \in \mathbb{R}^{n-m}$ and $Cx = C_1y + C_2z$

analogously $P^\top BP = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ with $B_{12} = B_{21}^\top$ and $P^\top b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

\Rightarrow necessary conditions in new variables:

$$\begin{bmatrix} B_{11} & B_{12} & C_1^\top \\ B_{21} & B_{22} & C_2^\top \\ C_1 & C_2 & 0 \end{bmatrix} \begin{pmatrix} y \\ z \\ \lambda \end{pmatrix} = \begin{pmatrix} -b_1 \\ -b_2 \\ -c \end{pmatrix} \quad (*)$$

Unique existence of solution is guaranteed, if

$$\begin{pmatrix} y \\ z \end{pmatrix}^\top \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{pmatrix} y \\ z \end{pmatrix} > 0 \quad \forall \begin{pmatrix} y \\ z \end{pmatrix} \in \ker(C) \setminus \{0\}$$

$$\ker(C) = \left\{ \begin{pmatrix} y \\ z \end{pmatrix} \mid C_1y + C_2z = 0 \right\} = \left\{ \begin{pmatrix} y \\ z \end{pmatrix} \mid y = -C_1^{-1}C_2z, z \in \mathbb{R}^{n-m} \right\} =$$

$$= \left\{ \begin{bmatrix} -C_1^{-1}C_2 \\ I \end{bmatrix} z \mid z \in \mathbb{R}^{n-m} \right\}$$

\Rightarrow necessary condition of 2nd order \Leftrightarrow

$$0 < z^\top \begin{bmatrix} -C_1^{-1}C_2 \\ I \end{bmatrix}^\top \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} -C_1^{-1}C_2 \\ I \end{bmatrix} z$$
$$= z^\top \left(\underbrace{B_{22} - C_2^\top C_1^{-\top} B_{12} - B_{21} C_1^{-1} C_2 + C_2^\top C_1^{-\top} B_{11} C_1^{-1} C_2}_{\text{"reduced Hessian" } H} \right) z \quad \forall z \in \mathbb{R}^{n-m} \setminus \{0\}$$

\Rightarrow H is nonsingular and thus invertible

How to solve the QP?

\rightarrow Can use a block Gauß elimination as above in 5.1 ?

We cannot use the matrix $\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ as pivot, because nonsingularity is not guaranteed!

\rightarrow IDEA: shovel around the KKT matrix

linear system $\textcircled{*}$ is equivalent to

$$\left[\begin{array}{cc|c} B_{11} & C_1^\top & B_{12} \\ C_1 & 0 & C_2 \\ \hline B_{21} & C_2^\top & B_{22} \end{array} \right] \begin{pmatrix} y \\ \lambda \\ z \end{pmatrix} = \begin{pmatrix} -b_1 \\ -c \\ -b_2 \end{pmatrix} \quad (\text{swap 2nd/3rd row and column})$$

Lemma 5.3. *The matrix $\begin{bmatrix} B_{11} & C_1^\top \\ C_1 & 0 \end{bmatrix}$ is invertible, if C_1 is invertible.*

Proof:

\Rightarrow matrix can be used as pivot.

Let us perform a block Gauß elimination again: $\begin{bmatrix} B_{11} & C_1^\top \\ C_1 & 0 \end{bmatrix}$ as Pivot

Analogously

$$\begin{pmatrix} y \\ \lambda \end{pmatrix} = - \begin{bmatrix} B_{11} & C_1^\top \\ C_1 & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} B_{12} \\ C_2 \end{bmatrix} z + \begin{pmatrix} b_1 \\ c \end{pmatrix} \right) =$$

$$= - \begin{bmatrix} 0 & C_1^{-1} \\ C_1^{-\top} & -C_1^{-\top}B_{11} & C_1^{-1} \end{bmatrix} \left(\begin{bmatrix} B_{12} \\ C_2 \end{bmatrix} z + \begin{pmatrix} b_1 \\ c \end{pmatrix} \right)$$

Using this in the remaining equation gives

$$B_{21} \left(-C_1^{-1}C_2z - C_1^{-1}c \right) + C_2^\top \left(-C_1^{-\top}B_{12}z + C_1^{-\top}B_{11}C_1^{-\top}C_2z - C_1^{-\top}b_1 + C_1^{-\top}B_{11}C_1^{-1}c \right) + B_{22}z = -b_2$$

\iff

$$\underbrace{\left(B_{22} - C_2^\top C_1^{-\top}B_{12} - B_{21}C_1^{-1}C_2 + C_2^\top C_1^{-\top}B_{11}C_1^{-1}C_2 \right)}_{S \text{ Schur complement} = \text{reduced Hessian}} z = -b_2 + B_{21}C_1^{-1}c + C_2^\top C_1^{-\top}b_1 - C_2^\top C_1^{-\top}B_{11}C_1^{-1}c$$

Remarks:

- Nullspace / range space methods can be used with advantage, if already existing solution methods for systems with B resp. C_1 have to be re-used.
- Alternative: straight forward Gauß algorithm is also possible
- For large scale problems, special Krylov subspace methods in the spirit of CG are available: MINRES, SYMMLQ.

5.2 QP with inequality constraints

For ease of presentation, we consider the problem class

$$\begin{aligned} \min_x \frac{1}{2} x^\top B x + b^\top x \\ B \in \mathbb{R}^{n \times n} \text{ p.d. (symm.)} \\ \text{s.t. } Cx + c \leq 0 \end{aligned}$$

we only have to solve

$$\begin{aligned} & \min_x \frac{1}{2} x^\top Bx + b^\top x \\ \text{s.t. } & (Cx + c)_i = 0 \quad \forall i \in \hat{I} \end{aligned}$$

which is only an equality constrained problem.

Active set strategy:

→ aims at finding exchange steps starting from an initial active set guess. Active constraints are added, if trial points hit the boundary of the admissible set. Other inequalities are released from the active set, if corresponding adjoints are $\lambda_j < 0$, since this violates KKT conditions.

Detailed active set algorithm

- ① Determine feasible point x^0 with $Cx^0 + c \leq 0$ (see below)

Define I^0 by $I^0 := \{i \mid (Cx^0 + c)_i = 0\}$

- ② For the index set I^k compute improving step $x^k + d^k$ via

$$\begin{aligned} & \min_d \frac{1}{2} (x^k + d)^\top B(x^k + d) + b^\top (x^k + d) \\ & \text{s.t. } ((C(x^k + d) + c)_i = 0 \quad \forall i \in I^k) \end{aligned}$$

$$\Leftrightarrow \begin{aligned} & \min \frac{1}{2} d^\top Bd + g^\top d \quad (g = Bx^k + b) \\ & \text{s.t. } (Cd)_i = 0 \quad \forall i \in I^k \end{aligned}$$

- ③ If $d^k \neq 0$ compute

$$\alpha^k := \max\{\alpha \in [0, 1] \mid (C(x^k + \alpha d^k) + c)_i \leq 0, \forall i \notin I^k\}$$

If α hits constraint j , add j

$$I^{k+1} = I^k \cup \{j\}$$

$$x^{k+1} := x^k + \alpha^k d^k \longrightarrow ①$$

If $d^k = 0$, consider λ :

If $\lambda_i \geq 0 \quad \forall i \in I^k \longrightarrow$ ready, solution found

else $\exists \lambda_j < 0 \Rightarrow$ remove constraint j with largest $|\lambda_j|$.

$$I^{k+1} = I^k \setminus \{j\} \longrightarrow ①$$

Objective function decreases monotonically

⇒ Algorithm is not cycling ⇒ ready after finitely many steps.

Supplement to ① : typical Ansatz:

Phase I Problem of linearen Programming

Solve $\min_x \sum_{i=1}^n x_i$ This is an LP, which can be solved, e.g., by SIMPLEX method
s.t. $Cx + c \leq 0$

(→ OR I)

Remarks:

- active set strategies have polynomial complexity in worst case
- Interior point methods (similar to the primal variant treated in definition 6.10 and theorem 6.11) have only linear complexity for given approximation accuracy, In QP, so-called primal-dual interior point methods are the de facto standard and implemented, e.g., in `CVXOPT` or also in `sklearn.linear_model.Lasso`. More on interior point methods in the context of data science can be learnt from the book [SNW11]¹
- Sometimes: crossover from IPM → AS
- Another alternative mainly in the context of partial differential equations and variational inequalities is semismooth-Newton aka primal-dual active set method

¹<https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Optimization%20for%20Machine%20Learning%20%5BSra%2C%20Nowozin%20%26%20Wright%202011-09-30%5D.pdf>

6 Nonlinear constrained optimization

6.1 Penalty and barrier techniques

To the term formation:

- penalty approach:
You may enter the forbidden area, but must pay penalty
- barrier approach:
The inadmissible area must not be entered

6.1.1 Quadratic penalty function

first only equality constraints:

$$(NLP) \quad \begin{array}{ll} \min f(x) & f : X \rightarrow \mathbb{R} \\ \text{s.t. } c(x) = 0 & g : X \rightarrow Y \end{array} \quad X, Y \text{ linear spaces}$$

We consider a family of solutions $x(\mu)$ of the problem

$$x(\mu) = \arg \min_x f(x) + \frac{\mu}{2} (c(x), c(x))_Y$$

and show $x(\mu) \xrightarrow{\mu \rightarrow \infty} \hat{x} = \text{solution of NLP}$

Example:

$$\begin{array}{ll} \min & x_1^2 + 4x_1x_2 + 5x_2^2 - 10x_1 - 20x_2 \\ \text{s.t.} & 2 - x_1 - x_2 = 0 \end{array}$$

$$P(x, \mu) = f(x) + \frac{\mu}{2} (c(x), c(x))_Y$$

Algorithm: choose $\mu^0 > 0$, $k := 0$

- ① determine $x^k := \arg \min_x P(x, \mu^k)$
- ② if $\|c(x^k)\| \leq \text{TOL}$ \Rightarrow STOP, ready
- ③ choose $\mu^{k+1} > \mu^k$ e.g. $\mu^{k+1} = 10 \cdot \mu^k$

Theorem 6.1. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ continuous and $\underline{f} := \inf\{x \in X | c(x) = 0\}$. The sequence $\{\mu^k\}_{k=0}^{\infty} \subset \mathbb{R}$ is assumed to be strictly monotonically increasing with $\mu^k \rightarrow \infty$. Furthermore, for each $\mu^k > 0$ there is an $x^k \in \mathbb{R}^n$ with

$$P(x^k, \mu^k) \leq P(x, \mu^k), \quad \forall x \in \mathbb{R}^n \quad (x^k \text{ solves penalized problem globally})$$

where $P(x, \mu) := f(x) + \frac{\mu}{2} \|c(x)\|_Y^2$.

Then, there holds:

- a) $\{P(x^k, \mu^k)\}_{k=0}^{\infty}$ is monotonically increasing
- b) $\{\|c(x^k)\|\}_{k=0}^{\infty}$ is monotonically decreasing
- c) $\{f(x^k)\}_{k=0}^{\infty}$ is monotonically increasing
- d) $\lim_{k \rightarrow \infty} c(x^k) = 0$
- e) Every accumulation point of $\{x^k\}_{k=0}^{\infty}$ is a solution of

$$\min_x f(x)$$

 s.t. $c(x) = 0$

Proof:

Remark: The global solution of the penalized problem is not trivial. If only local solutions can be achieved, the accumulation points of the iterations are still KKT points (cf. N/W: Theorem 17.2)

Surprising observation:

$\mu^k c(x^k) \rightarrow \hat{\lambda}$, in this way, adjoint variables can be estimated

Theorem 6.2. $f : X \rightarrow \mathbb{R}$ diffb. and $\{x^k\}$ as above; furthermore exist $J(x) := Dc(x)$, $H(\hat{x}) := J(\hat{x})J(\hat{x})^{ad}$ is assumed invertible (e.g. J surj.)

Then there holds

a) $\lim_{k \rightarrow \infty} \mu^k c(x^k) = \hat{\lambda} = - (J(\hat{x})J(\hat{x})^{ad})^{-1} J(\hat{x}) \nabla f(\hat{x})$

b) for $(\hat{\lambda}, \hat{x})$ according to a) we have

$$\begin{aligned}\nabla f(\hat{x}) + J(\hat{x})^{ad} \hat{\lambda} &= 0 \quad (\text{nec. cond.}) \\ c(\hat{x}) &= 0\end{aligned}$$

Remark: From necessary conditions, we know

$$\begin{aligned} \nabla f(\hat{x}) &\perp T_{\hat{x}}M \text{ and } \Pi := \left(I - J(\hat{x})^{ad} (J(\hat{x})J(\hat{x})^{ad})^{-1} J(\hat{x}) \right) \\ &\text{is the orthogonal projection on } T_{\hat{x}}M \\ \Rightarrow b) \Leftrightarrow \Pi \nabla f(\hat{x}) = 0 &\Leftrightarrow \nabla f(\hat{x}) \perp T_{\hat{x}}M \end{aligned}$$

Proof:

a) $x^k \rightarrow \hat{x} \Rightarrow J(x^k) \rightarrow J(\hat{x})$. By presupposition is

$$J(\hat{x})J(\hat{x})^{ad} \quad \text{invertible} \Rightarrow \text{also}$$

$$J(x^k)J(x^k)^{ad} \quad \text{for } k \text{ sufficiently large}$$

x^k Minimum \Rightarrow

$$\nabla_x P(x^k, \mu^k) = \nabla f(x^k) + \mu^k J(x^k)^{ad} c(x^k) = 0 \quad (*)$$

$$\Rightarrow J(x^k) \nabla f(x^k) + \mu^k J(x^k) J(x^k)^{ad} c(x^k) = 0$$

$$\Rightarrow \mu^k c(x^k) = -[J(x^k) J(x^k)^{ad}]^{-1} J(x^k) \nabla f(x^k) \Rightarrow a)$$

b) obviously follows $\lambda = -[J(\hat{x}) J(\hat{x})^{ad}]^{-1} J(\hat{x}) \nabla f(\hat{x})$

With limit in (*) we obtain

$$\Rightarrow \nabla f + J(\hat{x})^{ad} \lambda = 0$$

Remark: $\hat{\lambda}$ like in a) is called a least-squares estimator of adjoint variables because of the projection properties

What about the Hessian matrix and its bad conditioning?

Theorem 6.3. $X = \mathbb{R}^n, Y = \mathbb{R}^m$ and presupposition like in theorem 6.2

$$\text{Then there holds } \text{cond}(\nabla^2 P(x^k, \mu^k)) = \frac{\lambda_{\max}^k}{\lambda_{\min}^k} \rightarrow \infty$$

Proof:

Quadratic penalty function with inequalities

$$\min_x f(x) \quad f, c_i, h_i : \mathbb{R}^n \rightarrow \mathbb{R}$$

$$\text{s.t. } c_i(x) = 0, i = 1, \dots, m$$

$$h_i(x) \leq 0, i = 1, \dots, l$$

Definition 6.4. *The quadratic penalty function with inequalities is*

$$P(x, \mu) = f(x) + \frac{\mu}{2} \|c(x)\|_2^2 + \frac{\mu}{2} \sum_{i=1}^l \max\{h_i(x), 0\}^2$$

Results

- algorithmic approach analogously
- Convergence to solution comparable
- Attention: $P(x, \mu)$ is only once differentiable !

ℓ_1 - Penalty Function

$$\begin{array}{ll} \min_x f(x) & x \in \mathbb{R}^n, c(x) \in \mathbb{R}^m \\ \text{s.t. } c(x) = 0 & \circledast \\ h_i(x) \leq 0 & \end{array}$$

Definition 6.5. ℓ_1 - *Penalty Function*

$$P_1(x, \mu) = f(x) + \mu \sum_{i=1}^m |c_i(x)| + \mu \sum_{i=1}^l \max\{h_i(x), 0\}$$

P_1 is a so-called exact penalty function

Definition 6.6. A Penalty Function $\phi(x, \mu)$ is called exact, if there is $\mu_0 > 0$ with

$$\hat{x} = \arg \min_x \phi(x, \mu) \quad \forall \mu \geq \mu_0$$

i.e. \hat{x} Solution of NLP \circledast

Theorem 6.7. With the equation-constrained problem is $P_1(x, \mu)$ for

$\mu > \max\{\widehat{|\lambda_i|}\}$ exact Penalty Function

Proof: Consider candidate $\hat{x} + \Delta x$. Then

$$P_1(\hat{x} + \Delta x, \mu) \stackrel{\text{Taylor}}{=}$$

$$f(\hat{x}) + \underbrace{\nabla_x f(\hat{x})^\top}_{=Dc(\hat{x})^\top \hat{\lambda}} \Delta x + \mu \underbrace{|c(\hat{x}) + Dc(\hat{x})\Delta x|}_{=0} + \mathcal{O}(\|\Delta x\|^2)$$

directional derivative

$$= f(\hat{x}) + \underbrace{\hat{\lambda}^\top Dc(\hat{x})\Delta x + \mu |Dc(\hat{x})\Delta x|}_{>0, \text{ since } \mu > \max\{|\hat{\lambda}_i|\}} + \mathcal{O}(\|\Delta x\|^2)$$

\Rightarrow For $\|\Delta x\|^2$ small enough yields $P_1(\hat{x} + \Delta x, \mu) > P_1(\hat{x}, \mu)$

$\Rightarrow \hat{x}$ is local minimum for P_1

□

Remarks:

- exact penalty function \Rightarrow Conditioning less problematic
- but P_1 not differentiable \rightarrow In practice, it often goes well
- Variant: Powell-Meritfunction

$$P_p(x, \mu) = f(x) + \sum_{i=1}^m \alpha_i |c_i(x)|$$

$$\text{mit } \alpha_i > |\hat{\lambda}_i| > 0$$

Augmented Lagrangian

Definition 6.8. *The Augmented Lagrangian function is defined as (here only equality conditions)*

$$P_{AL}(x, \mu) = f(x) + \hat{\lambda}^\top c(x) + \frac{\mu}{2} \|c(x)\|_2^2$$

with $\hat{\lambda}$ = adj. Variable in NLP solution

Theorem 6.9. *P_{AL} is an exact penalty function*

Proof:

Problem: $\hat{\lambda}$ usually not known.

Successive estimation: $\mathcal{L}_A(x, \lambda; \mu) := f(x) + \lambda^\top c(x) + \frac{\mu}{2} \|c(x)\|_2^2$

obviously, there holds for λ arbitrary: $\nabla_x \mathcal{L}_A(x, \lambda; \mu) = \nabla_x f(x) + Dc(x)^\top (\lambda + \mu \cdot c(x))$

Thus, $\hat{x} \rightarrow \hat{x}$ only, if $\lambda + \mu \cdot c(x) \rightarrow \hat{\lambda}$

\Rightarrow update for λ : $\lambda^{k+1} = \lambda^k + \mu^k \cdot c(x^k)$

This defines the so-called **Augmented-Lagrangian algorithm or Method of Multipliers**.

But what about $c(x) = 0$? If $\|c(x^k)\|$ too large, then make μ^k larger.

LANCELOT - sophisticated Augmented-Lagrangian-Method

① Choose $0 < \beta < 1$, $\gamma < 1$, $\mu^0, \omega^0, \eta^0, x^0, \lambda^0, \tau > 1$

② Compute x^k with

$$\|\nabla_x \mathcal{L}_A(x^k, \lambda^k)\| \leq \omega^k$$

③ If $\|c(x^k)\| \leq \gamma \cdot \eta^k$

update Lagrange variable, i.e.

$$\lambda^{k+1} = \lambda^k + \mu^k c(x^k)$$

$$\mu^{k+1} = \mu^k$$

else update penalty parameter, i.e.

$$\lambda^{k+1} = \lambda^k$$

$$\mu^{k+1} = \tau \cdot \mu^k$$

④ Tighten tolerances:

$$\omega^{k+1} = \omega^k \cdot \frac{1}{\mu^{k+1}}$$

$$\eta^{k+1} = \eta^k \cdot \left(\frac{1}{\mu^{k+1}}\right)^\beta$$

STOP iteration, if criteria in ① and ② small enough

cf. Conu / Gould / Toint: LANCELOT, 1992

Large
And
Nonlinear
Constrained
Extended
Lagrangian
Optimization
Techniques

most recent version: → GALAHAD

Eigenschaften:

- very robust
- globally convergent
- easy implementation
- less efficient than SQP methods (below)

Generalization to inequalities via slack variables with bound constraints
→ N/W Chap 17.4

Barrier methods / interior point methods

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & c(x) \leq 0 \end{aligned}$$

Definition 6.10. *Barrier function: (logarithmic)*

$$B(x, \mu) := f(x) - \mu \sum_{i=1}^m \ln(-c_i(x))$$

Strategy

determine $x^k := \arg \min_x B(x, \mu^k)$

reduce $\mu^{k+1} < \mu^k$

Theorem 6.11. Let f and g be continuous functions and assume that there exists a lower bound U with $f(x) \geq U \forall x \in M = \{x | c(x) \geq 0\}$. Let $\{\mu^k\}$ be monotonically decreasing to 0 and $M = \overline{M^\circ}$. Let the series $\{x^k\}$ be defined according to the strategy above, then, there holds

- a) $\inf f(x) \leq B(x^{k+1}, \mu^{k+1}) \leq B(x^k, \mu^k)$
- b) $\gamma(x^k) \leq \gamma(x^{k+1}), \gamma(x) = - \sum_{i=1}^m \ln(-c_i(x))$
- c) $f(x^{k+1}) \leq f(x^k)$
- d) Every accumulation point of $\{x^k\}$ solves $\min_{c(x) \leq 0} f(x)$

Proof: analogously to theorem 6.2. □

Remark: There are many details to be told and variants of interior point methods to be discussed, which we omit here for the sake of brevity.

6.2 ADMM - Alternating Direction Method of Multipliers (Glowinski/Marocco 1975 and Gabay/Mercier 1976)

Motivation: let us look at the method of multipliers for a particular problem structure:

$$\begin{aligned} & \min_{(x,z)} f(x) + g(z) \\ \text{s.t. } & Ax + Bz = c \end{aligned}$$

where $x \in \mathbb{R}^n, z \in \mathbb{R}^m, A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}, c \in \mathbb{R}^p$. We assume that $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ are closed (in the sense that all level sets N_α are closed sets), proper (i.e., there is at least one x with $f(x) < \infty$ and one z with $g(z) < \infty$), and convex functions.

We build the augmented Lagrangian

$$\mathcal{L}_A(x, z, \lambda; \mu) := f(x) + g(z) + \lambda^\top (Ax + Bz - c) + \frac{\mu}{2} \|Ax + Bz - c\|_2^2$$

The classical method of multipliers discussed above, now takes the form

$$(x^{k+1}, z^{k+1}) = \arg \min_{x,z} \mathcal{L}_A(x, z, \lambda^k; \mu)$$

$$\lambda^{k+1} = \lambda^k + \mu(Ax^{k+1} + Bz^{k+1} - c)$$

block iteration idea as in section 3.6 leads to so-called Alternating Direction Method of Multipliers (ADMM)

$$\begin{aligned}x^{k+1} &= \arg \min_x \mathcal{L}_A(x, z^k, \lambda^k; \mu) = \arg \min_x f(x) + (\lambda^k)^\top A x + \frac{\mu}{2} \|Ax + Bz^k - c\|_2^2 \\z^{k+1} &= \arg \min_z \mathcal{L}_A(x^{k+1}, z, \lambda^k; \mu) = \arg \min_z g(z) + (\lambda^k)^\top B z + \frac{\mu}{2} \|Ax^{k+1} + Bz - c\|_2^2 \\\lambda^{k+1} &= \lambda^k + \mu(Ax^{k+1} + Bz^{k+1} - c)\end{aligned}$$

We observe that the minimization problem for x and also for z are both strictly convex optimization problems.

Theorem 6.12. *Consider the structured optimization problem above together with the assumptions on x, z, A, B, c, f, g . Furthermore, we assume that A and B have each rank p . Also, we assume the parameter $\mu > 0$ and that the optimization problem possesses a KKT point. Then, the ADMM-iteration above converges to a KKT point of the problem, i.e., $\lim_{k \rightarrow \infty} (x^k, z^k, \lambda^k) = (x^*, z^*, \lambda^*)$ KKT point.*

Proof. The proof requires a significant amount of convex analysis and is thus beyond the scope of this lecture. The theorem is shown in [EY15, lemma 18] for the case $B = I$, which appears frequently in data science. This result is generalized in [MXAP11] to the weaker assumptions of the theorem here. □

Remarks:

- The theorem does not depend on the particular size of $\mu > 0$. A common strategy is to choose it so that the primal residual $r^k := Ax^k + Bz^k - c$ and the so-called dual residual $s^k := A^\top B(z^k - z^{k-1})$ do not differ more than one order of magnitude in size. A more detailed strategy is discussed in [BPC⁺10, section 3.4.1].
- We do not require differentiability of f, g , only convexity, which means that the KKT conditions at the solution are

$$\begin{aligned} 0 &\in \partial f(x^*) + A^\top \lambda \\ 0 &\in \partial g(z^*) + B^\top \lambda \\ 0 &= Ax^* + Bz^* - c \end{aligned}$$

and the necessary conditions of optimality for each step may look similar. Often, the subproblems can be solved analytically, as we will see below.

- Convergence speed is discussed, e.g., in [DY12]. Depending on additional assumptions like strict convexity, one obtains sublinear convergence of type $\mathcal{O}(1/k)$ or $\mathcal{O}(1/k^2)$ or even r-linear convergence ($y^k \rightarrow y^*$ r-linearly, if there exists a sequence $\{\sigma^k\}_{k=1}^\infty$ converging (Q-)linearly to 0, such that $\|y^k - y^*\| \leq \sigma^k$).
- Simple examples show that ADMM can be very slow to converge to high accuracy.

6.2.1 ADMM applied to LASSO

Recall the LASSO problem formulation from example 2 at the beginning of section 5, which we write now in an equivalent formulation appropriate for ADMM

$$\min_{(x,z)} \frac{1}{2} \|Ax - b\|_2^2 + \rho \|z\|_1, \quad \lambda > 0$$

$$\text{s.t. } x - z = 0$$

Thus, the resulting ADMM iteration is

$$x^{k+1} = \arg \min_x \frac{1}{2} \|Ax - b\|_2^2 + (\lambda^k)^\top x + \frac{\mu}{2} \|x - z^k\|_2^2$$

$$z^{k+1} = \arg \min_z \rho \|z\|_1 - (\lambda^k)^\top z + \frac{\mu}{2} \|x^{k+1} - z\|_2^2$$

$$\lambda^{k+1} = \lambda^k + \mu(x^{k+1} - z^{k+1})$$

Both minimization subproblems can be solved analytically.

optimization problem for x :

Quadratic optimization problem with necessary (and sufficient) condition (normal equations) for the optimal \hat{x} :

$$\begin{aligned} A^\top(Ax^{k+1} - b) + \lambda^k + \mu(x^{k+1} - z^k) &= 0 \\ \Leftrightarrow x^{k+1} &= (A^\top A + \mu I)^{-1}(A^\top b + \mu z^k - \lambda^k) \end{aligned}$$

optimization problem for z :

The convex problem includes discontinuities, which can be treated by the convex analysis methodology introduced in Fermat's rule (theorem 2.13), which gives us the necessary condition

$$0 \in \partial_z \left(\rho \|z\|_1 - (\lambda^k)^\top z + \frac{\mu}{2} \|x^{k+1} - z\|_2^2 \right) = \rho \partial_z \|z\|_1 - \lambda^k - \mu(x^{k+1} - z)$$

The definition of the subgradient gives us

$$\partial_z \|z\|_1 = \begin{pmatrix} \partial_{z_1}|z_1| \\ \vdots \\ \partial_{z_n}|z_n| \end{pmatrix}$$

and

$$\partial_{z_i}|z_i| = \begin{cases} -1 & , \text{ for } z_i < 0 \\ [-1, 1] & , \text{ for } z_i = 0 \\ 1 & , \text{ for } z_i > 0 \end{cases}$$

with the definition

$$S_\kappa(a) := \begin{cases} a + \kappa & , \text{for } a < -\kappa \\ 0 & , \text{for } |a| < \kappa \\ a - \kappa & , \text{for } a > \kappa \end{cases}$$

the so-called soft threshold operator, the resulting ADMM iteration for the LASSO problem can be written as

$$\begin{aligned} x^{k+1} &= (A^\top A + \mu I)^{-1}(A^\top b + \mu z^k - \lambda^k) \\ z_i^{k+1} &= S_{\rho/\mu} \left(\frac{\lambda_i^k}{\mu} + x_i^{k+1} \right), \quad \forall i = 1, \dots, n \\ \lambda^{k+1} &= \lambda^k + \mu(x^{k+1} - z^{k+1}) \end{aligned}$$

6.2.2 ADMM for consensus optimization

Important for parallel programming and also Neural Networks. Consider the optimization problem

$$\min_x \sum_{j=1}^m f_j(x)$$

where each f_j is convex. This can be written appropriately for ADMM as

$$\begin{aligned} & \min_{x_j} \sum_{j=1}^m f_j(x_j) \\ \text{s.t. } & x_j - z = 0, \forall j \end{aligned}$$

The augmented Lagrangian is

$$\mathcal{L}_A(x_1, \dots, x_m, z, \lambda_1, \dots, \lambda_m; \mu) = \sum_{j=1}^m \left(f_j(x_j) + \lambda_j^\top (x_j - z) + \frac{\mu}{2} \|x_j - z\|_2^2 \right)$$

and the resulting ADMM iteration

$$\begin{aligned}x_j^{k+1} &= \arg \min_{x_j} f_j(x_j) + (\lambda_j^k)^\top x_j + \frac{\mu}{2} \|x_j - z^k\|_2^2, \quad \forall j = 1, \dots, m \\z_i^{k+1} &= \frac{1}{m} \sum_{j=1}^m \left(x_j^{k+1} + \frac{1}{\mu} \lambda_j^k \right) \\ \lambda_j^{k+1} &= \lambda_j^k + \mu(x_j^{k+1} - z^{k+1}), \quad \forall j = 1, \dots, m\end{aligned}$$

using $\sum_{j=1}^m \lambda_j^k = 0$, which holds in the solution anyway (and thus λ_j^0 should be initialized with $\sum_{j=1}^m \lambda_j^0 = 0$) gives

$$\begin{aligned}x_j^{k+1} &= \arg \min_{x_j} f_j(x_j) + (\lambda_j^k)^\top x_j + \frac{\mu}{2} \|x_j - \bar{x}^k\|_2^2, \quad \forall j = 1, \dots, m \\ \lambda_j^{k+1} &= \lambda_j^k + \mu(x_j^{k+1} - \bar{x}^{k+1}), \quad \forall j = 1, \dots, m\end{aligned}$$

where $\bar{x}^k = \frac{1}{m} \sum_{j=1}^m x_j^k$.

Usage in parallel processing:

in each iteration

- gather x_j^k and average to get \bar{x}^k
- scatter the average \bar{x}^k to processors
- update λ_j^k locally (in each processor, in parallel)
- update x_j locally

There is an abundance of further applications like in image denoising and inpainting, e.g. [Woh19]

6.3 SQP Methods

Sequential

Quadratic Programming

Succesive

Generalization to problems with inequalities: $\min_x f(x)$

$$\text{s.t. } c(x) = 0$$

$$g(x) \leq 0$$

Iteration: [in the vicinity of the solution]

$$x^{k+1} = x^k + \Delta x^k \quad \text{with}$$

$$\begin{aligned} & \min_{\Delta x^k} \frac{1}{2} (\Delta x^k)^\top H_k \Delta x^k + (\nabla f^k)^\top \Delta x^k \\ & \text{unter } C_k \Delta x^k + c^k = 0 \\ & \quad G_k \Delta x^k + g^k \leq 0 \end{aligned} \tag{QP}$$

Hence the name SQP !

$$H_k \approx \text{Hess}_x(f(x) + \lambda^\top c(x) + \mu^\top g(x))$$

Convergence:

Theorem 6.13. *Be \hat{x} a solution point, and assume that there hold the sufficient conditions of optimality with strict complementarity. Then the QP for (x^k, λ^k) sufficiently close to $(\hat{x}, \hat{\lambda})$ has the same active-set as (\hat{x}) .*

Idea of the proof: Perturbation approach and strict complementarity □

Corollary 6.14. *The SQP method with exact Hessian matrix has local quadratic convergence.*

Proof: Choose (x^k, λ^k) sufficiently close to $(\hat{x}, \hat{\lambda})$ for the active-set to be constant □

Obvious questions:

① construction of H_k ?

② global convergence ?

H_k can generally only be approximated

→ IDEE: Update formulas as in the unconstrained case

Broyden family

$H_{k+1} = H_k + U(H_k, p^k, q^k)$ with

$$p^k = x^{k+1} - x^k \text{ and } q^k = \nabla_x \mathcal{L}(x^{k+1}, \lambda^k, \mu^k) - \nabla_x \mathcal{L}(x^k, \lambda^k, \mu^k)$$

↑ ↑

Index not increased, since only
2nd deriv. w.r.t. x needed!

Attention $\text{Hess}_X \mathcal{L}$ need not be pos.def. - only on the nullspace of C_k

⇒ Ideally suited Broyden symmetric rank 1 with appropriate treatment of indefinite QP
(e.g. trust-region)

or BFGS Update with so-called Powell-Relaxation

If $p^{k\top} q^k \geq \alpha p^{k\top} H_k p^k$ (α e.g. 0.2)

not satisfied, define $r^k := \theta_k q^k + (1 - \theta_k) H_k p^k$

and perform BFGS update with p^k, r^k instead of p^k, q^k

where θ_k is defined by

$$\theta_k = \begin{cases} 1 & \text{if } p^{k\top} q^k \geq 0.2 p^{k\top} H_k p^k \\ (0.8 p^{k\top} h_k p^k) / (p^{k\top} H_k p^k - p^{k\top} q^k) & \text{else} \end{cases}$$

In the case $\theta_k \neq 1$ this ensures that

$$p^{k\top} r^k = 0.2 p^{k\top} H_k s^k > 0$$

Result:

- Interpolation between H_{full}^{k+1} and H^k !
- works well in combination with line search methods
- Problem: "blow-up" of the Hessian approximation (\Rightarrow restarts often necessary)
- If $\nabla_x^2 \mathcal{L}$ pos.def. (unrealistic) superlinear convergence can be shown - otherwise only linear.
(Secant condition violated in Powell relaxation!)

Several SQP variants:

Variant 1: Augmented Lagrangian SQP (Han, Kunisch)

IDEA in the equation constrained case: instead of f , we use $f + \frac{\mu}{2}c(x)^\top c(x)$ in the objective. Then the augmented Lagrangian function of the previous chapter results here as Lagrangian function:

$$\Leftrightarrow \nabla \mathcal{L}_A = 0 \text{ with } \mathcal{L}_A = f(x) + \lambda^\top c(x) + \frac{\mu}{2}c(x)^\top c(x)$$

For μ sufficiently large is $Hess_x \mathcal{L}_A$ pos.def

Make updates for $Hess_x \mathcal{L}_A$.

Variant 2: Reduced SQP methods

Consider optimization problem with separability structure

$$\begin{array}{ll} \min_{(z,p)} & f(z, p) \\ \text{s.t.} & c(z, p) = 0, \frac{\partial c}{\partial z} \text{ invertible} \end{array}$$

Knowledge: $T(z, p) := \begin{bmatrix} -c_z^{-1} c_p \\ I \end{bmatrix}$ spans nullspace of $[C_z, C_p]$

and $R(z, p) := \begin{bmatrix} -c_z^{-1} \\ 0 \end{bmatrix}$ is right inverse, i.e. $[C_z, C_p]R(z, p) = I$

both used in (II) (cf. nullspace decomposition in section 5.1.2)

$$T^\top HT\Delta p = -T^\top(\nabla_{(z,p)}f - HRc) \quad (\text{tedious calculation})$$

This results in total

$$\begin{pmatrix} \Delta z \\ \Delta p \end{pmatrix}^{SQP} = -T[T^\top HT]^{-1} \underbrace{T^\top(\nabla_{z,p}f - HRc)}_{\text{red. grad.}} - R c$$

\uparrow
reduced Hessian
 B

Term small near
solution, $c = 0$,
thus, can be omitted

$$\begin{pmatrix} \Delta z \\ \Delta p \end{pmatrix}^{RSQP} := -T[T^\top HT]^{-1}T^\top(\nabla_{z,p}f) - R c$$

tangential step | orthogonal step

Remark:

Names are correct, if $R = Q_1 L^{-1}$, $T = Q_2$ from LQ decomposition ($Q := C[Q_1 \ Q_2] = [L \ 0]$)
→ Projected SQP methods, in this form - via a QR decomposition - impracticable.

Summarizing, the RSQP method corresponds to a Newton/Quasi-Newton method for

$$\min_p f(z(p), p) \quad , \quad \text{where for } z(p) \text{ only one} \\ \text{Newton step is performed}$$

and vice versa an SQP method with special Hessian matrix approximation:

$$\begin{bmatrix} 0 & 0 & C_z^\top \\ 0 & T^\top HT & C_p^\top \\ C_z & C_p & 0 \end{bmatrix} \begin{pmatrix} \Delta z \\ \Delta p \\ -\lambda \end{pmatrix} = \begin{pmatrix} -\Delta_z f \\ -\Delta_p f \\ -c \end{pmatrix}$$

It needs "only" an approximation of $T^\top HT$.

local convergence:

- If $T^\top HT$ exactly available \Rightarrow 2-step quadratic convergence
- If Quasi-Newton updates \Rightarrow 2-step superlinear convergence

→ For this only use tangential information:

$$\begin{aligned} p^k &= \Delta p \\ q^k &= T^\top (\nabla_{(z,p)} \mathcal{L}(z^k + T\Delta p, \lambda^k) - \nabla_{(z,p)} \mathcal{L}(z^k, \lambda^k)) \end{aligned}$$

(or analytical approximation for $T^\top HT = \text{Hess}_p f(z(p), p)$)

Remark: method can also incorporate additional equality and inequality constraints in the so-called partially reduced SQP methods (Schulz '96).

Variant 3: Generalized Gauß-Newton methods for constrained least squares problems

Problem class

$$\begin{aligned} \min_x \quad & \frac{1}{2} \|F(x)\|_2^2, & F : \mathbb{R}^n \rightarrow \mathbb{R}^k \\ \text{s.t.} \quad & c(x) = 0, & c : \mathbb{R}^n \rightarrow \mathbb{R}^m \\ & g(x) \leq 0, & g : \mathbb{R}^n \rightarrow \mathbb{R}^\ell \end{aligned}$$

Analogous to the unconstrained case (see section 3.5) we use a Linearization under the norm:

Convergence:

- locally linear with convergence rate $\kappa = \mathcal{O}(\|F(\hat{x})\|)$
- theory based on particular pseudoinverse (cf. Bock 1987).

→ check out python implementation of the solver SLSQP[Kra88] in
`scipy.optimize.minimize`.

6.4 Globalization of Convergence

a) line-search technique

We need a criterion to evaluate the current step length, a so-called merit function. For this we require the following characteristics:

- achieve both an improvement in the objective function and an improvement in the constraints conditions
- Consistency: a descent with respect to the merit function should be possible at least for small steps
- it should be possible to prove convergence towards a solution.

IDEA: use already discussed penalty functions for this.

Example: Powell penalty function for

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & c(x) = 0 \end{aligned}$$

$$P_p(x, \mu) = f(x) + \sum_{i=1}^m \alpha_i |c_i(x)| \quad \alpha_i > |\hat{\lambda}_i| > 0$$

The step Δx is determined by $\min \Delta x \frac{1}{2} \Delta x^\top H \Delta x + \nabla f^\top \Delta x$
s.t. $C\Delta x + c = 0$

where $H \approx \text{Hess}_x \mathcal{L}$, but H positiv definite approximation.

Lemma 6.15 (Consistency). *For P_p and Δx as above, there holds*

$$\left. \frac{d}{d\varepsilon} \right|_{\varepsilon=0+} P_p(x + \varepsilon \Delta x, \alpha) < 0$$

Proof:

Remarks:

- This strategy is implemented in the solver option SLSQP[Kra88] to be used within `scipy.optimize.minimize`.
- The arguments of this lemma and the following theorem can also easily be transferred to the ℓ_1 -penalty function P_1 .

Theorem 6.16 (Global convergence). *We assume that the weights α_i in P_p are selected so that the following applies $\alpha_i > |\lambda_i^k|$, for all iterations k . And that the level set $N := \{x \mid P_p(x, \alpha) \leq P_p(x^0, \alpha)\}$ is compact and the Hessian matrix approximations H^k uniformly bounded (i.e. there is $\beta, \gamma > 0$ with $\beta < H^k < \gamma$, $\forall k$). Then there is at least one accumulation point of the damped SQP iteration and each such accumulation point is KKT point.*

Proof:

b) Trust region globalization for constrained problems

Aspects:

1. update of the trust region
2. Solution of the TR sub-problems

Step 1: Determine within the trust-region radius δ a a (almost) feasible substep, i.e. solve the subproblem

$$\begin{aligned} & \min \|C_z \Delta z_1 + C_p \Delta p_1 + c\|^2 \\ \text{s.t. } & \left\| \begin{pmatrix} \Delta z_1 \\ \Delta p_1 \end{pmatrix} \right\| \leq (\sigma \cdot \Delta)^2 \end{aligned}$$

where typically $\sigma = 0.8$. Determine $\delta_1 := \min\{\sigma\delta, \left\| \begin{pmatrix} \Delta z_1 \\ \Delta p_1 \end{pmatrix} \right\|\}$.

Step 2: Computation of tangential step in the direction of optimality, thus solving

$$\begin{aligned} & \min \frac{1}{2} \Delta p_2^\top B^k \Delta p_2 + (T^\top \nabla f)^\top \Delta p_2 \\ \text{s.t. } & \|T \Delta p_2\| \leq \delta^2 - \delta_1^2 \end{aligned}$$

where B_k denotes the corresponding approximation of the reduced Hessian matrix.

The total step then results from

$$\begin{pmatrix} \Delta z \\ \Delta p \end{pmatrix} = \begin{pmatrix} \Delta z_1 \\ \Delta p_1 \end{pmatrix} + T \Delta p_2$$

Remark: This brief description of globalization strategies is far from being comprehensive. Rather, it should be understood as a list of examples of the major classes line-search and trust-region. More information can be found, for example, in the textbook [NW06].

Final remarks on knowing omissions: Every lecture series can only present a selection of important topics, and I tried to select the topics in a way I feel appropriate in particular for the application field of data science. In particular, we have included the central modern tools SVM, ALS, stochastic optimization in neural networks, ADMM.

Nevertheless, there have been painful omissions, which I would at least like to suggest:

- there is much more to say about interior point methods
- optimization on manifolds is a central topic in data science
- variants of approximate SQP methods, in particular important for optimal control (see next semester)
- optimization with constraints in the form of complementarity problems like Stackelberg games and variational inequalities

Further literature:

- [Abe12] Shigeo Abe. Support Vector Machines for Pattern Classification. Springer, 2012.
- [BCN18] L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. SIAM Review, 60(2):223–311, 2018. http://users.iems.northwestern.edu/~nocedal/PDFfiles/SIAMREVIEW_optML.pdf.
- [Ber99] Dimitri P. Bertsekas. Nonlinear Programming. Athena Scientific, 1999.
- [BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.
- [BPC⁺10] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1–122, 2010.
- [BT00] Dimitri P. Bertsekas and John N. Tsitsiklis. Gradient convergence in gradient methods with errors. SIAM Journal on Optimization, 10(3):627–642, 2000.
- [DY12] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Journal of Scientific Computing, 66(3), 2012.

- [EY15] Jonathan Eckstein and Wang Yao. Understanding the convergence of the alternating direction method of multipliers: Theoretical and computational perspectives. *Pacific Journal of Optimization*, 11:619–644, 10 2015.
- [GS99] Luigi Grippo and Marco Sciandrone. Globally convergent block-coordinate techniques for unconstrained optimization. *Optimization Methods and Software*, 10(4):587–637, 1999.
- [HDBJ14] M. Hagana, H. Demuth, M. Beale, and O. De Jesus. *Neural Network Design (2nd Edition)*. Amazon ebook, 2014. <http://hagan.okstate.edu/NNDesign.pdf>.
- [JNS13] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013. <https://arxiv.org/pdf/1212.0467.pdf>.
- [KM40] M. Krein and D. Milman. On extreme points of regular convex sets. *Studia Mathematica*, 9(133-138), 1940. <https://www.impan.pl/en/publishing-house/journals-and-series/studia-mathematica/all/9/1/93490/on-extreme-points-of-regular-convex-sets>.
- [Kra88] Dieter Kraft. A software package for sequential quadratic programming. Technical Report Technical Report DFVLR-FB 88-28, Oberpfaffenhofen: Institut für Dynamik der Flugsysteme (DLR), 1988. "http://degenerateconic.com/wp-content/uploads/2018/03/DFVLR_FB_88_28.pdf".

- [KS88] A. V. Knyazev and A. L. Skorokhodov. The rate of convergence of the method of steepest descent in Euclidean norm. *U.S.S.R. Comput. Maths. Math. Phys.*, 28(5):195–196, 1988.
- [Min10] H. Minkowski. Geometrie der Zahlen. Teubner, Leipzig, 1910.
- [MXAP11] Joao F. C. Mota, Joao M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. A proof of convergence for the alternating direction method of multipliers applied to polyhedral-constrained functions. Technical Report 1112.2295, arXiv, 2011. <https://arxiv.org/abs/1112.2295v1>.
- [NJLS09] A. Nemirovski, A. Juditsky, G. Lan, and A Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. <http://pubs.siam.org/doi/pdf/10.1137/070704277>.
- [Roc70] R. Tyrrell Rockafellar. Convex analysis. Princeton Mathematical Series. Princeton University Press, 1970.
- [Rud17] Sebastian Ruder. An overview of gradient descent optimization algorithms. Technical Report 1609.04747, arXiv, 2017. <https://arxiv.org/pdf/1609.04747v2.pdf>.
- [SNW11] Suvrit Sra, Sebastian Nowozin, and Stephen Wright. Optimization for Machine Learning. MIT Press, 2011. "<https://doc.lagout.org/science/Artificial%20Intelligence/Machine%20learning/Optimization%20for%20Machine%20Learning%20%5BSra,%20Nowozin%20&%20Wright%202011-09-30%5D.pdf>".

- [Woh19] Brendt Wohlberg. Sporco: a python package for standard and convolutional sparse representations. In PROC. OF THE 16th PYTHON IN SCIENCE CONF. (SCIPY 2017), 2019. "https://conference.scipy.org/proceedings/scipy2017/pdfs/brendt_wohlberg.pdf".
- [ZMSJ18] Jingzhao Zhang, Aryan Mokhtari, Suvrit Sra, and Ali Jadbabaie. Direct Runge-Kutta discretization achieves acceleration. Technical Report 1805.00521, arXiv, 2018. <https://arxiv.org/pdf/1805.00521.pdf>.