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.

Contents

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.

We'll discuss mostly the second topic within the framework of finite dimensional continuous optimization with special emphasis on data science aspects, referring to the first for the necessary foundations (e.g. recognizing solutions).

Definition 1.1. *(Notation)*

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

$$\widehat{x} \in X \quad \textit{solves} \quad \min_{x \in X} f(x) \qquad :\Leftrightarrow \widehat{x} \in X \quad \textit{solves} \quad \min_{x} f(x) \ \textit{such that } x \in X$$

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

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

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

$$\widehat{x} \ \textit{solves} \quad \begin{array}{ll} \min\limits_{x} & f(x) \\ \textit{s.t.} & c(x) = 0 \\ & d_i(x) \geq 0 \ \forall i = 1, \ldots, n_i \end{array} \qquad :\Leftrightarrow \quad f(\widehat{x}) \leq f(x) \quad \forall x \in X$$

We need a local solution concept in the following and take the usual topology of $\mathbb{R}^n$. If the constraints c are formed by ordinary or partial differential equations, we need more abstract function spaces (Banach or Hilbert space) instead of the finite dimensional space $\mathbb{R}^n$. The concepts to be discussed below can mostly easily carried over to those function spaces and generalizations will be mentioned. However, the focus of this lecture is on finite dimensional nonlinear programming in $\mathbb{R}^n$.

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

$$\hat{x} \in U \text{ und } f(\hat{x}) \leq f(x) \quad \forall x \in U$$
$$\left(f(\hat{x}) < f(x) \quad \forall x \in U \backslash \{\hat{x}\} \right)$$

We need also a clear concept of a gradient. We do not use this notion synonymously with the term derivative. Both is defined and compared in the next definition.

Definition 1.3. *(derivative vs. gradient)*
A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable at the point $a \in \mathbb{R}^n$, if there exists a linear mapping $T : \mathbb{R}^n \to \mathbb{R}^m$ and a function $r : \mathbb{R}^n \to \mathbb{R}^m$, which is continuous at a and there holds

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

*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.,

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

If we pick a specific scalar product in $(.,.)_A$, which is represented by a symmetric and positiv 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

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

*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 explicitely 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:

a) $X = \mathbb{R}^n, (x,y) = x^\top y, \quad f : X \to \mathbb{R}$

$$\Rightarrow \mathcal{M}_{\{e_1\}}^{\{e_1,\dots,e_n\}}\big(Df(a)\big) = \left[\frac{\partial f}{\partial x_1}(a), \dots, \frac{\partial f}{\partial x_n}(a)\right] \text{ and } Df(a)v = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}(a) \cdot v_i$$

$$\Rightarrow \nabla_x f(a) = \left[\frac{\partial f}{\partial x_1}(a), \dots, \frac{\partial f}{\partial x_n}(a)\right]^\top = \mathcal{M}_{\{e_1\}}^{\{e_1,\dots,e_n\}}\big(Df(a)\big)^\top \qquad (\textit{column vector !})$$

b) $X = \mathbb{R}^n, (x,y) = x^\top A y \quad , A$ symm., pos.def.

$$Df(a)x = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_1} \end{pmatrix}^\top x = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}^\top A^{-1}Ax = \left[A^{-1}\begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}\right]^\top Ax =$$

$$= \left(A^{-1}\begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}, x\right)$$

$$\Rightarrow \nabla_x f(a) = A^{-1}\mathcal{M}_{e_1}^{e_1,\dots,e_n}\big(Df(a)\big)^\top$$

lecture 1, part 1

c) $X = L^2\big([0,1]\big)$ with $(x,y) := \int_0^1 x(t) \cdot y(t)dt$. For $x \in X$ we define

$$f(x) = \int_0^1 x(t)dt$$

Derivative = ?

$$
\begin{aligned}
Df(a)h &= \tfrac{d}{d\varepsilon}\Big|_{\varepsilon=0} f(a+\varepsilon\cdot h) = \tfrac{d}{d\varepsilon}\Big|_{\varepsilon=0} \int_0^1 \big[a(t)+\varepsilon h(t)\big]dt = \\
&= \int_0^1 h(t)dt = (\mathbf{1},h)\ \text{, with } \mathbf{1}\equiv 1 \\
\Rightarrow Df(a) &= (\mathbf{1},.) \\
&\text{and } \nabla_x f(a) = \mathbf{1}
\end{aligned}
$$

d) $X = L^2\big([0,1]\big)$, $f(x) = \int_0^1 x(t)\cdot\omega(t)dt$ and $\omega : [0,1] \to \mathbb{R}_+$ weight function (density) such that $(x,y) := \int_0^1 x(t)\cdot y(t)\cdot\omega(t)dt$

Then

$$
\begin{aligned}
Df(a)h &= \int_0^1 h(t)dt = \int_0^1 \frac{1}{\omega(t)} h(t)\omega(t)dt = \left(\frac{1}{\omega},h\right) \\
\Rightarrow \nabla_x f(a) &= \tfrac{1}{\omega}
\end{aligned}
$$

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 surjectiv, 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 ($\to$ exercise).

Definition 1.5. *We consider the set* $M := \{x \in X \,|\, g(x) = 0\}$ *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*

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

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

Theorem 1.6.

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

Proof: Take any curve γ in M through a. Then yields

$$g\big(\gamma(t)\big) = 0 \quad \forall t \in (-\varepsilon, \varepsilon)$$

$$\Rightarrow 0 = \frac{d}{dt}\Big|_{t=0} g\big(\gamma(t)\big) = Dg(a)\dot{\gamma}(0) \Rightarrow \dot{\gamma}(0) = v \in \ker Dg(a)$$

On the other hand, we can find a curve γ for every $v \in \ker Dg(a)$ with $\gamma(0) = a$ and $\dot{\gamma}(0) = v$, such that $\gamma(t) \in M$ for all $t \in (-\varepsilon, \varepsilon)$. This fact is a standard result from differential geometry, which can be found in any introductory text and whose detailed exposition is out of the scope of this course. □

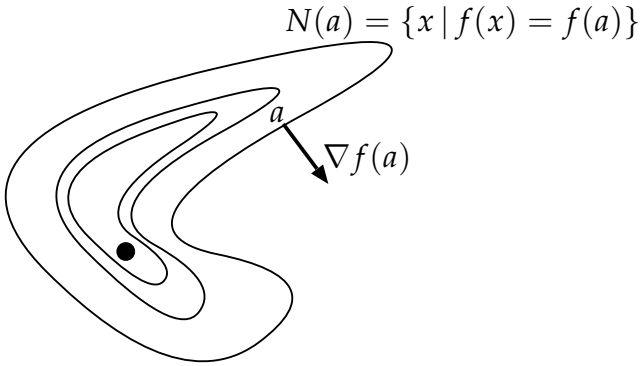


Figure 1: Visualization of the gradient

Now, we can derive the following important geometric insight:

Theorem 1.7. *We choose a scalar product* $(.,.)_A$ *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*

$$N(a) := \{x \in X \,\big|\, f(x) = f(a)\}$$

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

If $v \in T_a N(a)$, i.e., according to theorem 1.6 $v \in \ker Df(a)$, we observe

$$\Rightarrow 0 = Df(a)v = (\nabla f(a), v) \qquad \text{(per definition of the gradient)}$$

□

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

Now, second derivatives:

We recall the inductive definition

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

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 $Hess\ f$ by*

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

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

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

Examples:

a) $X = \mathbb{R}^n$, $(x, y) = x^\top y$, $f : X \to \mathbb{R}$

$$D^2 f(a)(v, w) = v^\top \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} w$$

$$\Rightarrow Hess\ f(a) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} = \frac{\partial^2 f}{\partial x^2}$$

b) Same as above, but $(x,y) = x^\top Ay$

$$\begin{aligned} \Rightarrow D^2 f(a)(v,w) &= v^\top \tfrac{\partial^2}{\partial x^2} f(a) w = v^\top \tfrac{\partial^2}{\partial x^2} f(a) A^{-1} A w = \\ &= \left(\left[A^{-1} \tfrac{\partial^2}{\partial x^2} f(a) \right] v, w \right) \end{aligned}$$

$$\Rightarrow Hess\ f(a) = A^{-1} \tfrac{\partial^2}{\partial x^2} f(a)$$

c) $X = L^2([0,1]) \quad (x,y) = \int_0^1 x(t)y(t)dt$

$$f(x) = \tfrac{1}{2} \int_0^1 x(t)^2 dt$$

$$D^2 f(a)(v,w) = \tfrac{d}{d\varepsilon_1}\Big|_{\varepsilon_1=0} \tfrac{d}{d\varepsilon_2}\Big|_{\varepsilon_2=0} \tfrac{1}{2} \int_0^1 \big(a(t) + \varepsilon_1 v(t) + \varepsilon_2 w(t)\big)^2 dt$$

$$= \int_0^1 v(t)w(t)dt$$

$$\Rightarrow Hess f(a) = id_{L^2([0,1])}$$

d) Analogously, we obtain in case d)

$$\begin{aligned} Hess f(a) = \frac{1}{\omega} \cdot id_{L^2([0,1])} : L^2([0,1]) &\to L^2([0,1]) \\ v &\mapsto \frac{v(.)}{\omega(.)} : [0,1] \to \mathbb{R} \\ & \qquad t \mapsto \tfrac{v(t)}{\omega(t)} \end{aligned}$$

lecture 1, part 2

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.*

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

Proof: We define for a general vector $v \in \mathbb{R}^n$

$$\varphi : [-\varepsilon, \varepsilon] \to \mathbb{R}$$
$$t \mapsto f(\hat{x} + t \cdot v)$$

Since $f(\hat{x})$ minimal for f , also $\varphi(0)$ is minimum of φ .

1D analysis $\Rightarrow$ $\varphi'(0) = 0$
$\varphi''(0) \geq 0$

Thus, $0 = \varphi'(0) = \frac{d}{dt}\Big|_{t=0} f(\hat{x} + tv) = Df(\hat{x})v$

$\Rightarrow Df(\hat{x})$ is the zero mapping $\Rightarrow \nabla f(\hat{x}) = 0$

And $0 \leq \varphi''(0) = \frac{d^2}{dt^2}\Big|_{t=0} f(\hat{x} + tv) = D^2 f(\hat{x})(v, v)$

$= \left(v, \nabla^2 f(\hat{x}) v\right) \quad \forall v \in X$ □

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* $\left(v, \nabla^2 f(\hat{x}) v\right) \geq \alpha \cdot \|v\|^2 \quad , \forall v \in X$

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

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

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

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

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

□

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 $\alpha := \min \sigma\big(\nabla^2 f(\hat{x})\big) > 0$.

Examples:

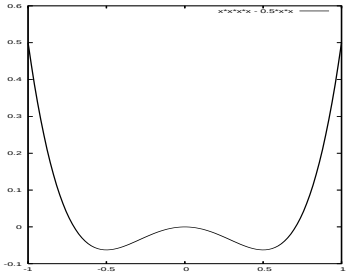
a) $f(x) = x^2$
$\Rightarrow \nabla f(x) = 2x \quad , \quad \nabla^2 f(x) = 2$
obviously, the sufficient condition is satisfied at $\hat{x} = 0$, $\Rightarrow$ Minimum

b) $f(x) = x^4$
$\Rightarrow \nabla f(x) = 4x^3 \quad , \quad \nabla^2 f(x) = 12x^2$
obviously, $\hat{x} = 0$ is strictly global minimum, however $\nabla^2 f(0) = 0$, i.e. sufficient condition of 2nd order is not satisfied.

Is there any problem?

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

has two local minima
at $x = \pm\sqrt{\varepsilon}$
local maximum at $x = 0$



$\Rightarrow$ completely different behaviour due to only small perturbation. I.e., stability of the solution under small perturbations does not hold, which is a significant problem for numerical computations. Therefore, we would like to know in advance, whether the problem that we try to solve is stable or not.

lecture 2, part 1

Theorem 2.3 (Stability). *Let $X := \mathbb{R}^n$ and $E := \mathbb{R}^m$. Furthermore, we assume that $f : X \times E \to \mathbb{R}$ 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, though important, is not relevant for the examination.*] By usage of the (necessary) condition

$$\nabla_x f(x,\varepsilon) = 0$$

and by usage of the implicit function theorem, we can find a function a local neighbourhood U and a function $\varphi : \varepsilon \mapsto \varphi(\varepsilon)$ such that

$$\nabla_x f(\varphi(\varepsilon),\varepsilon) = 0$$

This is an immediate consequence that $Hess_x f(x^0,0)$ is positive definite and therefore $D_x \nabla_x f(x^0,0)$ invertible. The function φ is continuously differentiable in U and formal differentiation yields the formula in the theorem.

Thus, the mapping

$$\varepsilon \mapsto H^\varepsilon := Hess_x f(\hat{x}(\varepsilon), \varepsilon)$$

is continuous (and differentiable) in U. We use this continuity to derive coercivity of H^ε for sufficiently small ε and thus, conclude that $\hat{x}(\varepsilon)$ is locally unique solution for sufficiently small ε.

Let $\alpha^\varepsilon := \min \sigma(H^\varepsilon)$ be an eigen value with eigen vector v, and $\alpha := \min \sigma(H^0)$. Then yields (for w.l.o.g.[1] $\alpha^\varepsilon \leq \alpha$, otherwise we obtain coercivity already from $\alpha^\varepsilon > \alpha$)

$$(H^\varepsilon - H^0)v = (\alpha^\varepsilon I - H^0)v \quad \Rightarrow \quad (\alpha^\varepsilon I - H^0)^{-1}(H^\varepsilon - H^0)v = v$$

Now, we obtain the matrix $Q \in O(n)$ of eigen vectors of H^0, such that $H^0 = Q\Lambda Q^\top$ with Λ diagonal matrix, the estimate

$$\begin{aligned} 1 &\leq \|(\alpha^\varepsilon I - H^0)^{-1}(H^\varepsilon - H^0)\| \leq \|(\alpha^\varepsilon I - H^0)^{-1}\|_2 \|H^\varepsilon - H^0\|_2 \\ &= \|Q(\alpha^\varepsilon I - \Lambda)^{-1}Q^\top\|_2 \|H^\varepsilon - H^0\|_2 \leq \|H^\varepsilon - H^0\|_2 \cdot \max_{\lambda \in \sigma(H^0)} |\alpha^\varepsilon - \lambda|^{-1} \\ \Rightarrow |\alpha^\varepsilon - \alpha| &\leq \|H^\varepsilon - H^0\|_2 . \end{aligned}$$

If we choose ϵ small enough that we obtain $\|H^\varepsilon - H^0\|_2 < \alpha/2$ (which is possible because of continuity of H^ε in ε), we conclude

$$\alpha - \alpha^\varepsilon \leq \frac{\alpha}{2} \Rightarrow \alpha^\varepsilon \geq \frac{\alpha}{2}$$

Therefore H^ε stays coercive for all such ε. □

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 severly, 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 $\hat{x}_\varepsilon$, which can be computed before the exact value of ε is known.

[1] without loss of generality

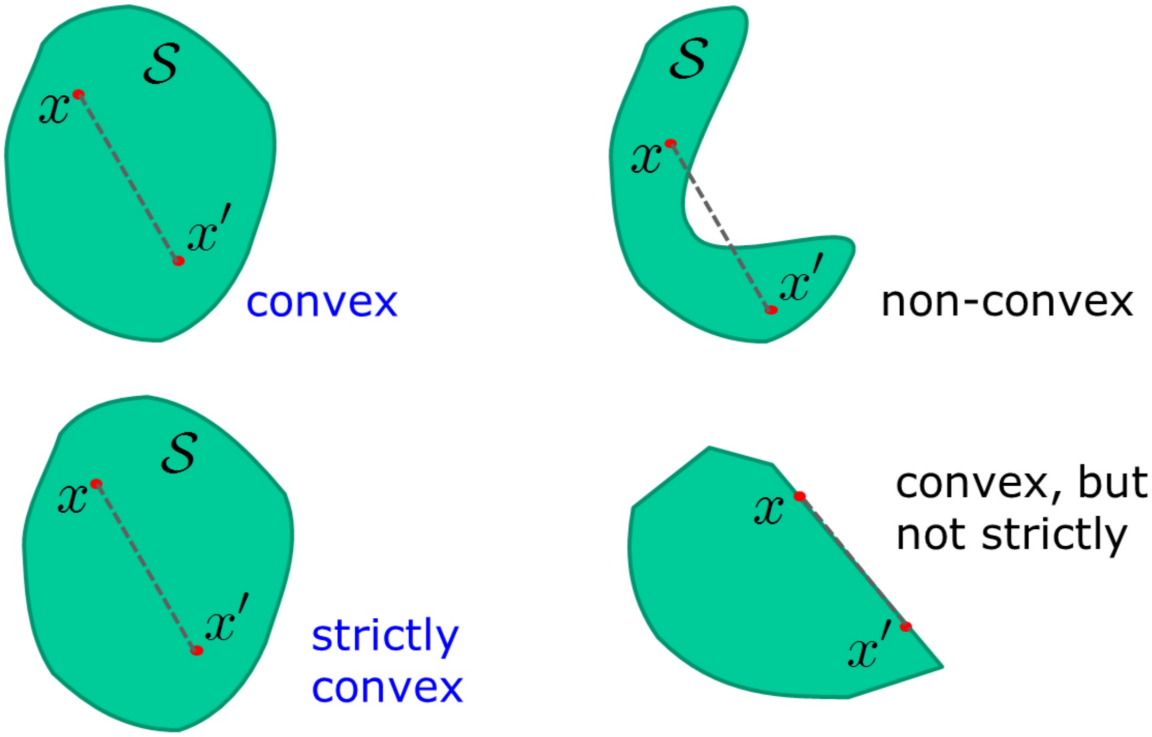
2.2 Convex optimization

There are several problem classes which are convex by nature. Convexity makes things somewhat easier, which is the reason, why less regularity (e.g., only Lipschitz continuity of the objective, instead of C^2) of the problem can be afforded, when the overall problem is assumed being convex. Furthermore, first order conditions are already sufficient for optimality. That is the reason, why convexity is often assumed somewhat ad hoc, even in cases, when it is not clear, whether this assumption is justified.

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

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

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 $[x_1, x_2] \setminus \{x_1, x_2\} \subset int(S)$, where $int(S)$ means the interior of S.

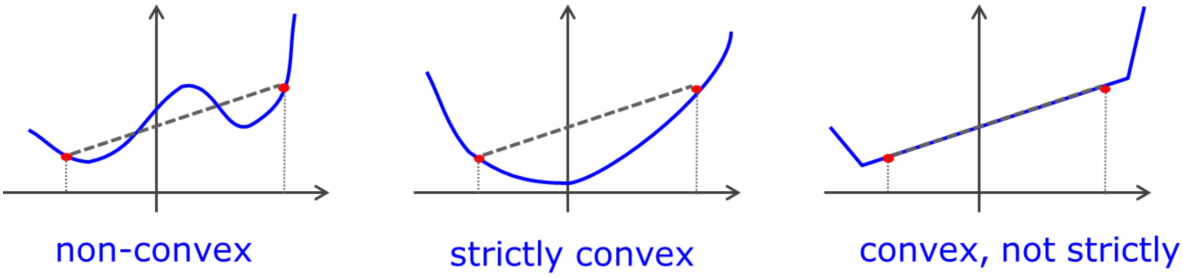


Definition 2.5 (Convex function)**.** *Let $S \subset \mathbb{R}^n$ be a convex subset. We say that the function $f : S \to \mathbb{R}$ is convex, if*

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

the function f is called strictly convex, if

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



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 $f : S \to \mathbb{R}$ 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*

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

Proof: By induction. If $\mu_m = 1$, the proposition is correct anyway. Therefore, we assume now $\mu_m < 1$. Then, we observe

$$\begin{aligned} f\left(\sum_{k=1}^{m} \mu_k x_k\right) &= f\left((1-\mu_m)\sum_{k=1}^{m-1} \frac{\mu_k}{1-\mu_m} x_k + \mu_m x_m\right) \\ &\leq (1-\mu_m) f\left(\sum_{k=1}^{m-1} \frac{\mu_k}{1-\mu_m} x_k\right) + \mu_m f(x_m) \\ &\overset{\text{(induction)}}{\leq} (1-\mu_m)\left(\sum_{k=1}^{m-1} \frac{\mu_k}{1-\mu_m} f(x_k)\right) + \mu_m f(x_m) \\ &= \sum_{k=1}^{m} \mu_k f(x_k) \end{aligned}$$

□

Remark: We have shown Jensen's inequality only in the case of finitely many terms. However, it comes in many variants, which are nevertheless always called Jensen's inequality. Here, we state the variant in probability theory, as well: Let $(\Omega, \mathfrak{F}, P)$ be a probability space, X an integrable real-valued random variable and f a convex function. Then:

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

where $\mathbb{E}$ means expected value. Theorem 2.6 states this already for discrete probability measures. The full non-discrete case follows from discrete approximations in the limit.

lecture 2, part 2

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*

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

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 $\operatorname{ext}(S)$.

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

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

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

Theorem 2.9 (Continuity). *A convex function $f : S \to \mathbb{R}$ defined on a convex subset $S \subset \mathbb{R}^n$ is (locally Lipschitz) continuous in $int(S)$, 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, though important, is not relevant for the examination.*] This proof is rephrased from the proof of theorem 7.36 in [Bec14]. Since $x_0 \in int(S)$, it follows that there exists $\varepsilon > 0$ such that $B_\varepsilon^\infty(x_0) \subset S$. Next we show that f is upper bounded over $B_\varepsilon^\infty(x_0)$.
Let $v_1, \ldots, v_{2^n}$ be the 2^n extreme points of $B_\varepsilon^\infty(x_0)$. The vectors v_i are of the form

$$v_i = x_0 + \varepsilon w_i,$$

where $w_1, \ldots, w_{2^n}$ are the vectors in $\{-1,1\}^n$. Then by the Krein-Milman theorem 2.8 for any $x \in B_\varepsilon^\infty(x_0)$ there exists a $\lambda \in \mathbb{R}^{2^n}$, $\sum_{i=1}^{2^n} \lambda_i = 1$, $\lambda_i \geq 0$ for $i = 1, \ldots, 2^n$, such that $x = \sum_{i=1}^{2^n} \lambda_i v_i$. Hence, by Jensen's inequality,

$$f(x) = f\left(\sum_{i=1}^{2^n} \lambda_i v_i\right) \leq \sum_{i=1}^{2^n} \lambda_i f(v_i) \leq M,$$

where $M = \max_{i=1,\dots,2^n} f(v_i)$. Since $B_\varepsilon^2(x_0) \subset B_\varepsilon^\infty(x_0)$ we conclude that $f(x) \leq M$ on $B_\varepsilon^2(x_0)$.
Let $x \in B_\varepsilon^2(x_0)$ and assume without loss of generality that $x \neq x_0$. Define

$$z = x_0 + \frac{1}{\alpha}(x - x_0),$$

where $\alpha = \frac{1}{\varepsilon}\|x - x_0\|$. Then obviously $\alpha \leq 1$ and $z \in B_\varepsilon^2(x_0)$, and in particular $f(z) \leq M$.
In addition,

$$x = \alpha z + (1 - \alpha)x_0.$$

Consequently, by Jensen's inequality we have

$$\begin{aligned} f(x) &\leq \alpha f(z) + (1 - \alpha) f(x_0) \\ &\leq f(x_0) + \alpha(M - f(x_0)) \\ &= f(x_0) + \frac{M - f(x_0)}{\varepsilon}\|x - x_0\|. \end{aligned}$$

We can therefore deduce that $f(x) - f(x_0) \leq L\|x - x_0\|$, where $L = \frac{M - f(x_0)}{\varepsilon}$. To prove the result, we need to show that $f(x) - f(x_0) \geq -L\|x - x_0\|$. For that, define

$$u = x_0 + \frac{1}{\alpha}(x_0 - x).$$

Clearly we have $\|u - x_0\| = \varepsilon$ and hence $f(u) \leq M$. In addition, $x = x_0 + \alpha(x_0 - u)$. Therefore,

$$f(x) = f(x_0 + \alpha(x_0 - u)) \geq f(x_0) + \alpha(f(x_0) - f(u)). \tag{2.1}$$

The latter inequality is valid since

$$x_0 = \frac{1}{1 - \alpha}(x_0 + \alpha(x_0 - u)) + \frac{\alpha}{1 + \alpha}u,$$

and hence, by Jensen's inequality

$$f(x_0) \leq \frac{1}{1 + \alpha} f(x_0 + \alpha(x_0 - u)) + \frac{\alpha}{1 + \alpha} f(u),$$

which is the same as in inequality (2.1) (after some rearrangement of terms). Now, continuing (2.1),

$$\begin{aligned} f(x) &\geq f(x_0) + \alpha(f(x_0) - f(u)) \\ &\geq f(x_0) - \alpha(M - f(x_0)) \\ &= f(x_0) - \frac{M - f(x_0)}{\varepsilon}\|x_0 - x\| \\ &= f(x_0) - L\|x - x_0\|, \end{aligned}$$

and the desired result is established. □

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])

There has been developed a rich theory (classical authors: Clark, Rockafellar) for convex functions, which are nondifferentiable. In order to get a glimpse on that, we discuss the notion of subdifferentials. In order to understand that, we need the following characterization of differentiable convex functions.

Theorem 2.10 (Linear subapproximation)**.** *Let the function $f : S \to \mathbb{R}$ 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: Consider $z := \lambda y + (1 - \lambda)x$ for $\lambda \in [0, 1]$. Then

$$f(z) = f(\lambda y + (1 - \lambda)x) \leq \lambda f(y) + (1 - \lambda) f(x)$$

and therefore

$$f(z) - f(x) \leq \lambda (f(y) - f(x))$$

From the definition of the gradient, we conclude now

$$(\nabla f(x), (y - x)) = \lim_{\lambda \to 0} \frac{f(x + \lambda(y - x)) - f(x)}{\lambda} = \lim_{\lambda \to 0} \frac{f(z) - f(x)}{\lambda} \leq f(y) - f(x)$$

□

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

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

Proof: 1) First assume that $\hat{x}$ is a minimizer. Consider $z := \hat{x} + \lambda(x - \hat{x})$ for $\lambda \in [0, 1]$. Then

$$f(z) \geq f(\hat{x}) \text{ , since } \hat{x} \text{ is minimizer}$$

and therefore

$$\frac{1}{\lambda}\left(f(\hat{x}+\lambda(x-\hat{x}))-f(\hat{x})\right)\geq 0$$

With the limit $\lim_{\lambda\to 0}$ we conclude

$$(\nabla f(\hat{x}),(x-\hat{x})\geq 0$$

for arbitrary $x\in S$.

2) Now assume for $\hat{x}\in S$ that there holds

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

Then, the linear subapproximation from theorem 2.10 gives us

$$f(x)\geq f(\hat{x})+(\nabla f(\hat{x}),(x-\hat{x})\geq f(\hat{x})$$

for arbitrary $x\in S$, which means that $\hat{x}$ is a minimizer. □

Definition 2.12 (Subgradient)**.** *Let the function $f:S\to\mathbb{R}$ 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*

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

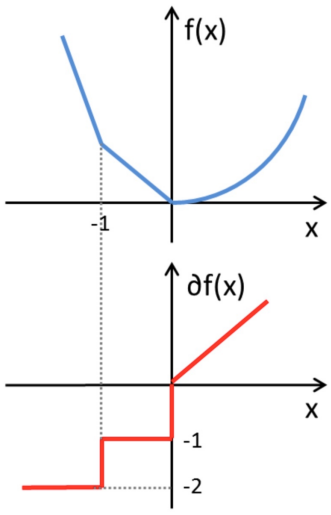
The set of all vectors v with this property is called subdifferential and notated as

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

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 $f : S \to \mathbb{R}$, the following equivalence holds*

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

Proof: $\hat{x}$ minimizes f if and only if

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

which by definition is equivalent to $0 \in \partial f(\hat{x})$. □

Remark: Thus, it is enough to look at first order conditions in convex optimization and all local minima are also global minima, which is the major aspect, why convex optimization is intensly investigated.

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

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

Proof: use linear subapproximation twice:

$$f(y) \geq f(x) + (\nabla f(x), (y - x))$$
$$f(x) \geq f(y) + (\nabla f(y), (x - y))$$

substract the second from the first and obtain the proposition □

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*

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

The parameter m is called the modulus of convexity.

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

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

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

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

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

lecture 3, part 1

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*

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

Proof:

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

□

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

This characterization of the orthogonal space to the constraints naturally leads to adjoint variables or Lagrange multipliers for constrained optimization problems.

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

Consider the optimization problem

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

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

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

satisfies

a) $\nabla_x \mathscr{L}(\hat{x}, \lambda) = 0$ *(Karush-Kuhn-Tucker condition)*

b) $\big(v, Hess_x \mathscr{L}(\hat{x}, \lambda) v\big) \geq 0 \quad \forall v \in \ker Dc(\hat{x})$

Proof: $\hat{x}$ is local solution, i.e.

$$f(\hat{x}) \leq f(x) \,, \forall x \in M := \{x \in X \big| c(x) = 0\} \cap U$$

Consider a curve $\gamma : (-\varepsilon, \varepsilon) \to M$ with $\gamma(0) = \hat{x}$

$\Rightarrow \varphi(t) := f\big(\gamma(t)\big)$ has a local minimum at $t = 0$

$\Rightarrow 0 = \varphi'(t) = Df\big(\gamma(0)\big)\dot{\gamma}(0) = Df(\hat{x})\dot{\gamma}(0) = \big(\nabla f(\hat{x}), \dot{\gamma}(0)\big)$

γ arbitrary $\Rightarrow \nabla f(\hat{x}) \in (T_{\hat{x}} M)^{\perp} = \big[\ker\big(Dc(\hat{x})\big)\big]^{\perp} = \operatorname{im}\big(Dc(\hat{x})^{ad}\big)$

$\Rightarrow \exists \tilde{\lambda} \in \mathbb{R}^m$ with $\nabla f(\hat{x}) = Dc(\hat{x})^{ad} \tilde{\lambda}$

The definition $\lambda := -\tilde{\lambda}$ yields $0 = \nabla f(\hat{x}) + Dc(\hat{x})^{ad} \lambda = \nabla_x \mathscr{L}(\hat{x}, \lambda)$

For curves γ we define $v := \dot{\gamma}(0)$ and $L(x) := \big(\lambda, c(x)\big)$

and observe $0 \leq \varphi''(0) = \frac{d^2}{dt^2}\Big|_{t=0} f\big(\varphi(t)\big) = (v, Hess_x f(\hat{x})\, v) + \big(\nabla f, \ddot{\gamma}(0)\big)$ and

$0 = \frac{d^2}{dt^2}\Big|_{t=0} L\big(\gamma(t)\big) = \big(v, Hess_x\, L(\hat{x}) v\big) + \big(\nabla L, \ddot{\gamma}(0)\big)$.

Since $0 = \nabla_x \mathscr{L}(\hat{x}) = \nabla_x f(\hat{x}) + \nabla_x L(\hat{x}, \lambda)$ we conclude

$$0 \leq \big(v, Hess_x \mathscr{L}(\hat{x}, \lambda) v\big) \,, \quad \forall v \in \ker Dc$$

□

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., [Zei95]). The challenge lies here in the proof of twice differentiability of c, which is often nontrivial or even impossible.

lecture 3, part 2

Now, we formulate sufficient optimality conditions in the equality constrained case.

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) $\nabla \mathscr{L}(\hat{x}, \lambda) = 0 \quad$ *for a* $\lambda \in \mathbb{R}^m$

b) $\left(v, Hess_x \mathscr{L}(\hat{x}, \lambda) v\right) \geq \alpha \|v\|^2, \ \forall v \in \ \ker Dc(\hat{x})$

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 $x_k \to \hat{x}$, 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 $s_{k'} \to \hat{s}$ 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

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

For $x_{k'}$ we have

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

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

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

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

lecture 4, part 1

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.

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

$$\boxed{\begin{array}{c} \nabla_x \mathscr{L}(\widehat{x}, \widehat{\lambda}) = 0 \\ c(\widehat{x}) = 0 \end{array}}$$

Proof: Theorem 2.19 shows that the solution $\hat{x}$ and its adjoint vector satisfy the conditions proposed. On the other hand, an exercise shows that the matrix

$$\frac{\partial}{\partial(x,\lambda)} \begin{pmatrix} \nabla_x \mathscr{L}(\widehat{x}, \widehat{\lambda}) \\ c(\widehat{x}) \end{pmatrix} = \begin{bmatrix} Hess_x\, \mathscr{L}(\widehat{x}, \widehat{\lambda}) & Dc(\hat{x})^{ad} \\ Dc(\hat{x}) & 0 \end{bmatrix}$$

is invertible. Thus, the proposition follows from the theorem of inverse functions. $\square$

At the first glance, adjoint variables seem quite abstract. The following theorem gives a specific interpretation in the form of shadow prices.

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

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

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: The necessary optimality conditions of first order in $\hat{x}$ give

$$D_x f(\hat{x})v + (\hat{\lambda}, Dc(\hat{x})v) = 0\,, \quad \forall v \in \mathbb{R}^n$$

We observe

$$\alpha = c\big(x(\alpha)\big) = 0, \Rightarrow id = \frac{d}{d\alpha}\bigg|_{\alpha=0} c\big(x(\alpha)\big) = D_x c(\hat{x})\frac{dx(\alpha)}{d\alpha}\bigg|_{\alpha=0}$$

$$\Rightarrow \frac{df\big(x(\alpha)\big)}{d\alpha}\bigg|_{\alpha=0} w = D_x f(\hat{x})\frac{dx(\alpha)}{d\alpha}\bigg|_{\alpha=0} w = \Big(-\hat{\lambda}, D_x c(\hat{x})\frac{dx(\alpha)}{d\alpha}\bigg|_{\alpha=0} w\Big) = (-\hat{\lambda}, w)\,, \forall w \in \mathbb{R}^m$$

$$\Rightarrow \nabla_\alpha f(x(0)) = -\hat{\lambda}$$ □

If we consider f as costs to be minimized, then $(-\hat{\lambda}, \alpha)$ is a first order approximation of the change of the costs, if the constraints are changed by the amount of α. This is the meaning of the term “shadow price”.

2.3.1 Important special case: Separable Problems

$\longrightarrow$ separability frame work : $X = Z \times P$

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

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

$$\begin{aligned} \nabla_z f(z,p) + c_z^{ad}(z,p)\lambda = 0 \quad &\text{(adjoint equation, uniquely solvable for } \lambda) \\ \nabla_p f(z,p) + c_p^{ad}(z,p)\lambda = 0 \quad &\text{(optimality condition., not necessarily solvable for } p) \\ c(z,p) = 0 \quad &\text{(state equation, uniquely solvable for } z) \end{aligned}$$

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.

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

$$\Rightarrow (v, Hess_x \mathscr{L}(\hat{x},\lambda)v) =$$

$$= \left(\begin{bmatrix} -c_z^{-1}c_p \\ I \end{bmatrix} \beta, Hess_x \mathscr{L}(\hat{x},\lambda) \begin{bmatrix} -c_z^{-1}c_p \\ I \end{bmatrix} \beta \right) =$$

$$= (\beta, B\beta), \text{ with}$$

$$B := \begin{bmatrix} -c_z^{-1}c_p \\ I \end{bmatrix}^{ad} Hess_x \mathscr{L}(\hat{x},\lambda) \begin{bmatrix} -c_z^{-1}c_p \\ I \end{bmatrix}$$

"reduced Hessian matrix"

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

$$Hess_p f\big(z(p),p\big) = B$$

lecture 4, part 2

2.4 Convex optimization with inequality constraints

First, we recall theorem 2.11 for necessary and sufficient optimality conditions in the convex case: Let the function $f : S \to \mathbb{R}$ be convex and differentiable on the convex set S. The point $\hat{x}$ is a minimizer of f, if and and only if

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

Now we consider a practically important special case, which is later generalized

Theorem 2.22. *We consider the convex set*

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

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

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

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

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

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

$$\nabla_x \mathscr{L}(\hat{x},\lambda,\mu) = 0$$
$$\lambda_i\,(\hat{x}_i - a_i) = 0 = \mu_i\,(b_i - \hat{x}_i)\,, \text{ for all } i = 1,\ldots,n$$

Proof. The convexity of the set S is obvious. Therefore, we can apply Theorem 2.11. Before that, we define

$$\lambda_i := (\nabla f(\hat{x})_i)^+ , \qquad \mu_i := (\nabla f(\hat{x})_i)^- , \ \forall i$$

where we define for $r \in \mathbb{R}$

$$r^+ := (|r| + r)/2 , \qquad r^- := (|r| - r)/2$$

From that definition of λ, μ we obtain immediately

$$\nabla_x \mathscr{L}(\hat{x}, \lambda, \mu) = \nabla f(\hat{x}) - \lambda + \mu = 0$$

From Theorem 2.11, we obtain now

$$(\lambda - \mu, x - \hat{x}) \geq 0 , \ \forall x \in S \tag{2.2}$$

We define

$$I^+ := \{i \in \{1, \dots n\} | \, \nabla f(\hat{x})_i > 0\}$$

Obviously, we have $\mu|_{I^+} = 0$. If $\lambda_i > 0$ on I^+ and $\hat{x}_i > a_i$, we obtain a contradiction to (2.2) and therefore also

$$\lambda_i \, (\hat{x}_i - a_i) = 0 \ \forall i \in I^+$$

On $\{1, \dots, n\} \setminus I^+$, we obviously observe $\lambda_i = 0$ and therefore

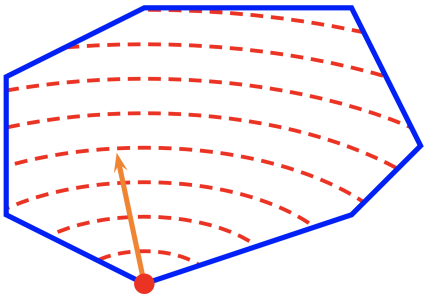
$$\lambda_i \, (\hat{x}_i - a_i) = 0 \ \forall i \in \{1, \dots, n\} \setminus I^+$$

Analogously, we show

$$\mu_i \, (b_i - \hat{x}_i) = 0 \ \forall i \in \{1, \dots, n\}$$

□

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 pogramming problem

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

Definition 2.23. *(active set)*

a) $S := \{x | c(x) = 0, h(x) \leq 0\}$ *"feasible region""*

b) for $x \in S$*, we define*
$I(x) := \{i \in \{1, \ldots, l\} \big| h_i(x) = 0\}$ *"active inequalities"*
$I^{\perp}(x) := \{i \in \{1, \ldots, l\} \big| h_i(x) < 0\}$ *"inactive inequalities"*

c) $x \in S$ *satisfies the LICQ (linear independence constraint qualification), if there holds*
$\{\nabla c_1(x), \ldots, \nabla c_m(x), \nabla h_{i_1}(x), \ldots, \nabla h_{i_s}(x)\}$ *is a linear independent set, where* $\{i_1, \ldots, i_s\} = I(x)$

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*

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

hold

a) $\nabla_x \mathscr{L}(\hat{x}, \lambda, \mu) = 0$ *(KKT-Bed.)*

b) $\mu^\top h(\hat{x}) = 0$ *(complementarity)*

c) $v^\top Hess_x \mathscr{L}(\hat{x}, \lambda, \mu) v \geq 0 \quad \forall v \in \tilde{T}(\hat{x})$

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

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*

$$v^\top Hess_x \mathscr{L}(\hat{x}, \lambda, \mu) v > 0 \quad \forall v \in \tilde{T}(\hat{x}) \backslash \{0\}$$

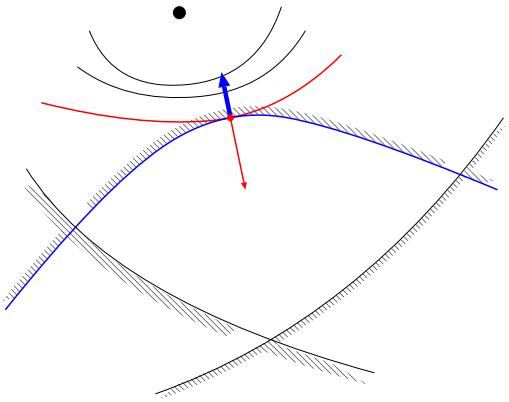
then $\hat{x}$ *is a strict local minimum.*

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 $x = (0, 0)$
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!

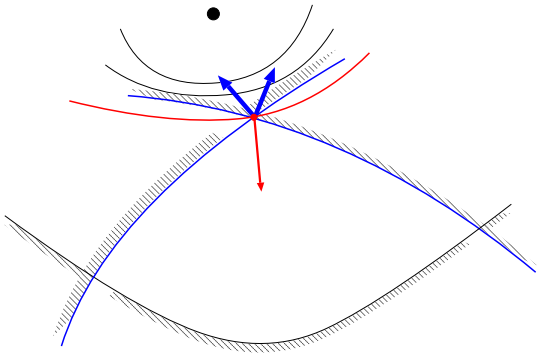
b) Example for solution on smooth boundary of feasible domain:



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

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

c) Example for solution at a corner:



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

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

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

solution candidates $x_1 = -1$
$x_2 = 2$ both satisfied sufficient condition

global solution $x_2 = 2$

lecture 4, part 3

2.6 Expoiting duality in convex optimization for support vector machines

Now, we study a famous example from data science, where duality plays a key role: Support Vector Machines (SVM). This concept (dating back to 1992 [BGV92]) aims at classification of data vectors, often separating data into two classes. A central geometric observation is the following seperation theorem.

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 $\mathcal{H}(w,b) := \{x \in \mathbb{R}^n \,|\, w^\top x = b\}$ 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]. □

Now let us look at the following illustration of two point sets. Here, we observe two classes of points, based on which we want to construct a mapping d for deciding for any point in $\mathbb{R}^n$ the membership to the appropriate point class in the following way:

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

In this context, this kind of decision mapping is called a “machine”.

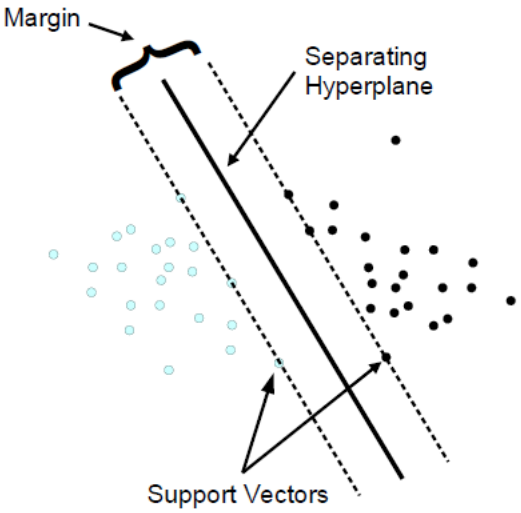
We deduce from theorem 2.26 that we can find a hyperplane, which separates the convex hulls of both point sets. Actually, there exist many of them. One can argue that the “optimal” hyperplane is defined as the one with maximum distance to both convex halls, i.e., the largest margin without any point. Those points sitting exactly at the maximal margin are called “support vectors”, thus giving the name of the decision mapping.

We consider data in the form of a set of pairs $D := \{(x_i, y_i) \,|\, x_i \in \mathbb{R}^n, y_i \in \{-1,+1\}, i = 1,\ldots,m\}$ meaning that we have several data vectors, x_i, and each belongs to exactly one of two classes with labels -1 or $+1$, instead of colors or similar. For the moment, we assume for the convex hulls

$$\begin{aligned}&\operatorname{conv}(D^+) \cap \operatorname{conv}(D^-) = \emptyset \\ &\qquad \text{where } D^+ := \{x \,|\, (x,y) \in D, y = +1\}, D^- := \{x \,|\, (x,y) \in D, y = -1\}\end{aligned}$$

thus ensuring (Minkowski theorem 2.26) that there exists a separating hyperplane. We want to find w, b and thus a separating hyperplane $\mathcal{H}(w,b)$ such that the data lie in respective halfspaces, i.e.

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



We observe that

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

and also the respective half spaces H^+, H^- are not changed at all by a constant positive factor. For any particular separating hyperplane $\mathcal{H}$, we can find the support vectors as the sets $\operatorname{argmin}_{x\in D^+} \|x - \mathcal{H}\|$, $\operatorname{argmin}_{x\in D^-} \|x - \mathcal{H}\|$ and take now one arbitrary sample from each set:

$$x^+_{\min} \in \operatorname*{argmin}_{x\in D^+} \|x - \mathcal{H}\|\ , \qquad x^-_{\min} \in \operatorname*{argmin}_{x\in D^-} \|x - \mathcal{H}\| \tag{2.3}$$

Now, we can choose w, b such that

$$w^\top x^+_{\min} + b = +1 \ \Rightarrow w^\top x_i + b \geq +1\,,\ \forall x_i \in D^+ \tag{2.4}$$

$$w^\top x^-_{\min} + b = -1 \ \Rightarrow w^\top x_i + b \leq -1\,,\ \forall x_i \in D^- \tag{2.5}$$

since this defines just two scalar equations for two unknowns (α, b). The plumbline from $x^-_{\min}$ to the respective hyperplane $\mathcal{H}(w,b)$ is a multiple $\mu \neq 0$ of w, which can be computed from the equation

$$w^\top (x^-_{\min} + \mu w) + b = 0 \ \Rightarrow \|\mu w\| = \frac{1}{\|w\|}$$

Thus, $\|\mu w\|$ is just half the margin and the total margin is therefore $m = 2/\|w\|$. We want to maximize the margin, which is equivalent to minimizing $\|w\|^2$. Both constraints in (2.4, 2.5) can be rephrased together as $y_i(w^\top x_i + b) \geq +1$. Summarizing, we have to solve the following linear-quadratic optimization problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 \tag{2.6}$$

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

Problem (2.6, 2.7) is an important example of a convex (linear-quadratic) optimization problem. Now, we study its so-called dual problem, which will give a computationally more accessible approach and also enables generalization options.

lecture 5, part 1

The dual problem can be derived for any optimization problem. The result is an optimization problem with respect to the adjoint variables instead of the problem variable $x \in \mathbb{R}^n$. If there are much less constraints than n, there may be more efficient solution strategies for the dual problem. However, duality shows a major impact only for convex problems. Therefore, we discuss the following problem here

$$\min_x f(x) \tag{2.8}$$

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

We assume that f and $h_i, i = 1 \ldots m$ are convex such that the feasible region $S := \{x \,|\, h_i(x) \leq 0, i = 1 \ldots m\}$ is convex. The Lagrangian for this optimization problem is

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

We define the dual objective function $q : \mathbb{R}^m \to \mathbb{R}$ as

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

This involves the finding of the global solution, which requires huge effort in general. However, if f, h are convex, also the Lagrangian is convex and therefore, any local solution is a global one, which makes this solvable with reasonable effort.

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

$$\max_\lambda q(\lambda)$$
$$\textit{s.t. } \lambda \geq 0\,.$$

The optimal value of the dual problem gives a lower bound on the primal problem, which is called weak duality.

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:

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

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

Theorem 2.29. *Suppose in the primal problem (2.8,2.9) that f and $c_i, i = 1, \ldots, m$ 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. The constraints (2.11) are satisfied because of the necessary optimality conditions and $\mathscr{L}(\hat{x}, \hat{\lambda}) = f(\hat{x})$. Therefore, we observe for any pair (x, λ) satifying (2.11)

$$\begin{aligned} \mathscr{L}(\hat{x}, \hat{\lambda}) = f(\hat{x}) &\geq f(\hat{x}) + \lambda^\top h(\hat{x}) = \mathscr{L}(\hat{x}, \lambda) \\ &\geq \mathscr{L}(x, \lambda) + \nabla_x \mathscr{L}(x, \lambda)(\hat{x} - x) = \mathscr{L}(x, \lambda) \end{aligned}$$

where we have exploited convexity of the mapping $x \mapsto \mathscr{L}(x, \lambda)$. □

Coming back to problem (2.6, 2.7), let us look at its Wolfe dual (2.10, 2.11), which is somewhat simpler and opens a computational perspective for generalization to more complicated data classification. We build the Lagrangian

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

and form the resulting (Wolfe) dual optimization problem

$$\begin{aligned} &\max_{w,b,\lambda} \mathscr{L}(w, b, \lambda) \\ &\text{s.t. } 0 = \nabla_w \mathscr{L}(w, b, \lambda) = w - \sum_{i=1}^{m} \lambda_i y_i x_i \quad \Rightarrow w = \sum_{i=1}^{m} \lambda_i y_i x_i \\ &\qquad 0 = -\nabla_b \mathscr{L}(w, b, \lambda) = \sum_{i=1}^{m} \lambda_i y_i \end{aligned}$$

Let us look at the effect of $w = \sum_{i=1}^m \lambda_i y_i x_i$ on interesting terms in the Lagrangian $\mathscr{L}$

$$\frac{1}{2}\|w\|_2^2 = \frac{1}{2} w^\top w = \frac{1}{2} \sum_{i,j=1}^m \lambda_i \lambda_j y_i y_j x_i^\top x_j$$

$$-\sum_{i=1}^m \lambda_i y_i w^\top x_i = -\sum_{i,j=1}^m \lambda_i \lambda_j y_i y_j x_i^\top x_j$$

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

Thus, the Wolfe dual problem can be simplified to

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

From its solution, we can recover w as $w = \sum_{i=1}^m \lambda_i y_i x_i$ and b for any i with $\lambda_i \neq 0$ from the then active equation $w^\top x_i + b = 1$. We observe that typically there are only very few $\lambda_i \neq 0$, thus reducing the effort for summation. The 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.

lecture 5, part 2

Soft margin hyperplane:

Here, we allow small deviations from the margin in the form that the constraint (2.7) of the primal problem is relaxed (2.6, 2.7) by an amount ξ_i. Of course, the amount of relaxation should be small, which causes also an additional term in the objective, This

results in the following modified optimization problem

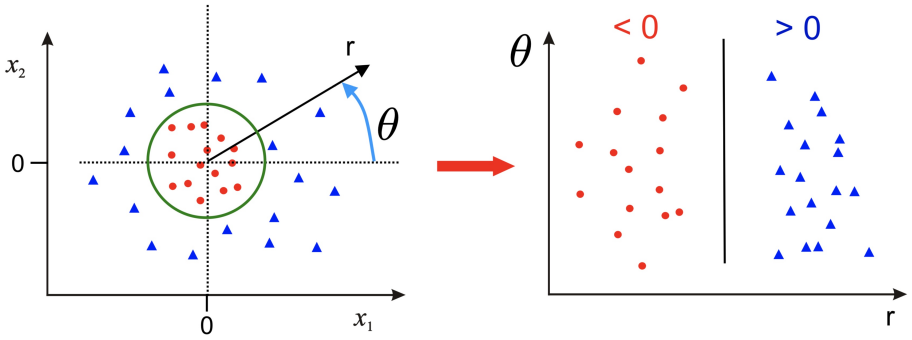
$$\begin{aligned}\min_{w,b} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i\,,\ \forall i = 1,\ldots,m \\ \xi_i \geq 0\,,\ \forall i = 1,\ldots,m\end{aligned}$$

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

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

Kernel trick for not linearly separable data:

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

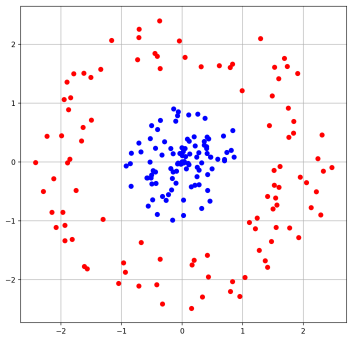


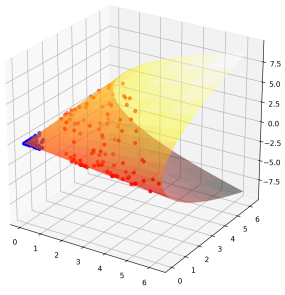
applying the mapping to polar coordinates

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

to the data results in data, which are linearly separable. Also going to a higher dimension by applying the mapping

$$\begin{aligned}\Phi : \mathbb{R}^2 \to \mathbb{R}^3 \\ x = \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2)^\top\end{aligned} \tag{2.12}$$





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

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

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

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

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

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

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

again

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

with classifier decision function

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

where typically one of the following standard kernels are used with appropriate kernel parameters

$$\begin{aligned}\text{linear:} \quad & k(x,y) = x^\top y\\ \text{polynomial:} \quad & k(x,y) = (\theta + x^\top y)^\ell \text{, for some } \ell \in \mathbb{N}_+\\ \text{sigmoid:} \quad & k(x,y) = \tanh(\theta + x^\top y)\\ \text{Gaussian:} \quad & k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma}} \text{, for some } \sigma > 0\end{aligned}$$

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.

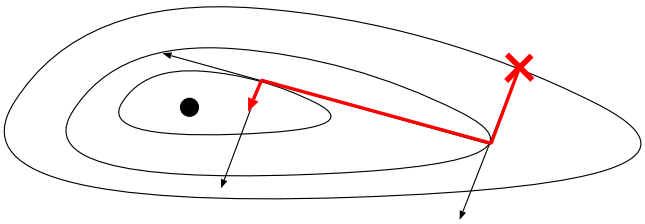
Standard implementations of SVM are well-known as LIBSVM and LIBLINEAR.

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].

lecture 5, part 3

3 Algorithms for unconstrained optimization problems

$\min\ f(x)\,, f : X \to \mathbb{R}, \quad X$ Eukl. vector space $\qquad$ 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 $f : X \to \mathbb{R}$, differentiable, we call $d \in X$ a descent direction at the point x , if*

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

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

Proof: $\left(\nabla f(x), d\right) = \frac{d}{dt}\big|_{t=0} f(x + td) = \lim_{t \to 0} \frac{1}{t}\left(f(x + td) - f(x)\right)$
$\Rightarrow$ for all $t > 0$ small enough, there must hold $f(x + td) - f(x) < 0$. □

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*

$$\min_d Df(x)d = \left(\nabla f(x), d\right)$$
$$\text{s.t. } \|d\|^2 = 1$$

Proof: Application of optimality conditions:

Lagrangefct. $\mathscr{L}(d,\lambda) = \big(\nabla_x f(x), d\big) + \lambda\big[(d,d) - 1\big]$

$$\nabla_d \mathscr{L} = \nabla_x f(x) + 2\lambda d = 0 \Rightarrow d = -\frac{1}{2\lambda}\nabla f(x)$$

Use $\|d\| = 1$ thus $\lambda = \pm\frac{1}{2}\|\nabla f(x)\|$

$Hess_d \mathscr{L}(d,\lambda) = \lambda \cdot id$ is only coercive for $\lambda > 0 \Rightarrow d^g$ is minimum. □

Remark: Interpretation: d^g minimizes

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

Algorithm (steepest descent)

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

for $k = 0, 1, \ldots,$ do
{

(1) compute $f(x^k), \nabla f(x^k)$

(2) if $\|\nabla f(x^k)\| \leq \varepsilon \rightarrow$ 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 $t^k := \arg\min\limits_t f(x^k + t d^k), t \in (0, \infty)$

(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 $\nabla f(x^*) = 0$

Proof: Consider $\{x^k\}$ as the constructed sequence, w.l.o.g. $\nabla f(x^k) \neq 0 \; \forall k$
then $f(x^{k+1}) < f(x^k) \Rightarrow$ monotone sequence and $\{x^k\} \subset N(x^0)$ comp. $\Rightarrow \exists$ convergent subsequence $\{x^{k_i}\}_{i=1,\ldots}$ with $x^{ki} \to x^* \in N(x^0) (i \to \infty)$

Assumption : $\nabla f(x^*) \neq 0 \Rightarrow \exists \tilde{t}$ with $f\big(x^* - \tilde{t}\nabla f(x^*)\big) \leq f(x^*) - \delta, \; \delta > 0$

on the other hand, the sequence $y_i := x^{k_i} - \tilde{t}\nabla f(x^{k_i})$ converges to $x^* - \tilde{t}\nabla f(x^*)$

$\Rightarrow \exists i_0 : \; f(y_i) < f(x^{k_i}) - \frac{\delta}{2} \quad \forall i \geq i_0$

↯ to line search, since then t^{k_i} not optimal □

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

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

Proof: $t^k = \arg\min f\big(x^k - t\nabla_x f(x^k)\big)$

$\Rightarrow 0 = \frac{d}{dt}\big|_{t=t_k} f\big(x^k - t\nabla f(x^k)\big) =$

$$= Df\underbrace{\big(x^k - t^k\nabla f(x^k)\big)}_{x^{k+1}}\nabla f(x^k) = \big(\nabla f(x^{k+1}), \nabla f(x^k)\big)$$ □

Remark: Steepest descent algorithms can be applied also in Hilbert spaces, convergence is shown only for quadratic functionalys.
(eg. [Wou79, 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:

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

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

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

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

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

The inequality of Kantorovich (see above) gives then:

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

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

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

□

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 replaced by $Q := Hess_x f(\widehat{x})$.

(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 $Hess f = A^{-1}[\partial^2 f/\partial x^2]$ 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 $\alpha < 1$

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

lecture 6

Remark: Note that the convergence orders are in $\mathbb{R}^n$ independent from the chosen norm—with the exception of linear independence. The reason is that, as is proved in lecture analysis, all norms in $\mathbb{R}^n$ are equivalent in the sense that one can find for any given two norms $\|.\|_\alpha, \|.\|_\beta$ numbers $0 < m \leq M < \infty$ such that

$$m\|x\|_\alpha \leq \|x\|_\beta \leq M\|x\|_\alpha\,, \quad \forall x \in \mathbb{R}^n$$

Thus, norm independence is obvious for quadratic, superlinear and sublinear convergence. However, linear convergence cannot generalized from $\|.\|_Q$ to any other norm, based on the principle of norm equivalence in $\mathbb{R}^n$ and needs special treatment. In [KS88] the following convergence rate for the steepest descent in Euclidean norm is shown:

$$\|x^{k+1} - \widehat{x}\|_2 \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max}}\right) \|x^k - \widehat{x}\|_2$$

which is, of course, also a linear convergence—with a slightly worse convergence rate, than for the norm $\|.\|_Q$

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 $\tau < 2/\lambda_{\max}$*

Proof:

$$x^{k+1} = x^k - \tau \cdot \nabla f(x^k) = x^k - \tau \cdot (Qx + b)$$

Its just the Richardson method for the linear equation $Qx + b = 0$. Thus, the theorem just restated a well-known convergence result for Richardson iteration. □

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 $\tau = 2/(\lambda_{\min} + \lambda_{\max}) < 2/\lambda_{\max}$.

Now, we investigate inexact line search:

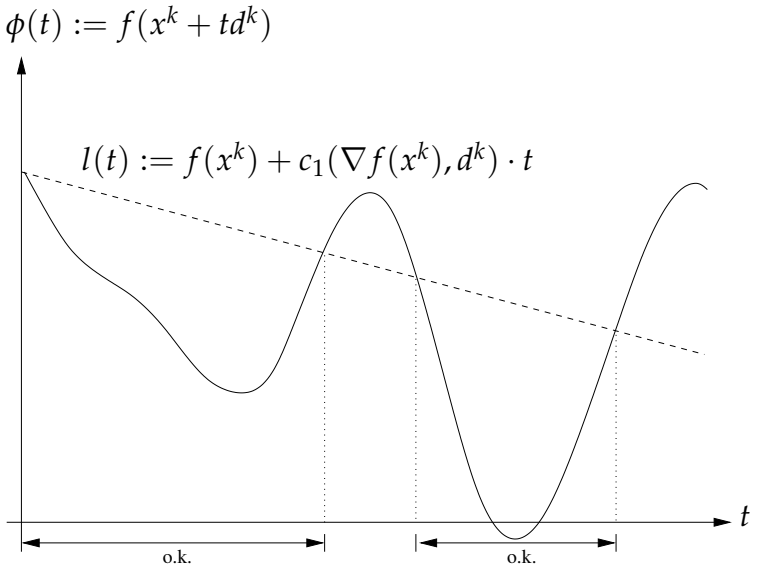
Requirements: we want to have $f(x^{k+1}) < f(x^k)$, but also not too large step size, in order to avoid erratic oscillations.

$\rightarrow$ There are various strategies:

- Armijo
- Goldstein
- Wolfe conditions

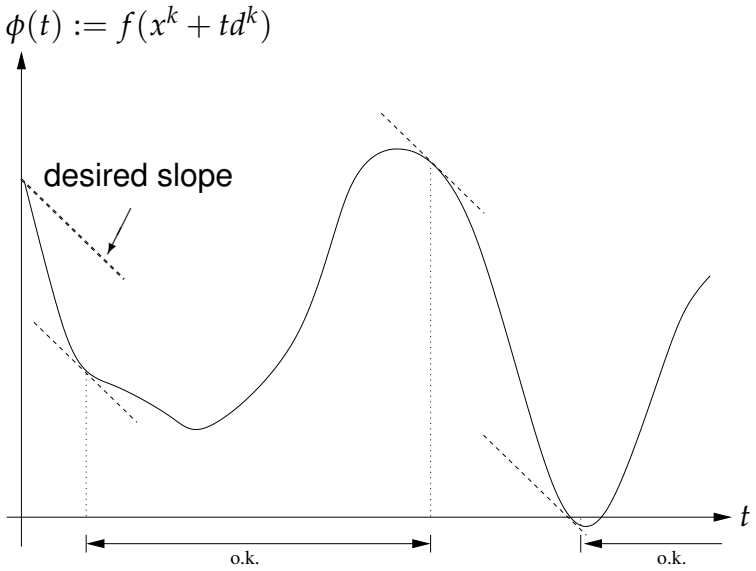
(a) $f(x^k + t^k d^k) \leq f(x^k) + c_1 t^k \big(\nabla f(x^k), d^k\big)$

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



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

"curvature condition" with $0 < c_1 < c_2 < 1$, $c_2 \approx 0,9$



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*

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

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*

$$\lim_{k\to\infty} \|\nabla f(x^k)\| = 0 \qquad \textit{(global convergence)}$$

Proof: obvious ✓

backtracking line-search

Choose a fixed $\bar{t} > 0$, ρ, $c_1 \in (0,1)$, set $t \leftarrow \bar{t}$
repeat
$\quad t \leftarrow \rho \cdot t$
until $f(x^k + td^k) \leq f(x^k) + c_1 t(\nabla f(x^k), d^k)$

$\rightarrow$ Observation: condition (a) "minimum decrease" is satisfied.
(b) does not have to be guaranteed explicitely, if $\bar{t}$ is chosen large enough.$\rightarrow$ 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

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

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:
   $x^{k+1} = x^k - \alpha^k \nabla f(x^k + \beta_k(x^k - x^{k-1})) + \beta_k(x^k - x^{k-1})$, $\alpha_k \approx 1/L, \beta_k \approx 1$
3. Adam (Adaptive Momentum):
   $x_i^{k+1} = x_i^k - \alpha\frac{\nu_k}{\sqrt{s_k+\epsilon}}\nabla f(x^k)_i, i = 1,...,n$
   where $\nu_k = \beta_1\nu_{k-1} - (1-\beta_1)\nabla f(x^k)_i$, $\beta_1 \approx 0.9$
   $s_k = \beta_2 s_{k-1} - (1-\beta_2)\nabla f(x^k)_i^2$, $\beta_2 \approx 0.99$, $\epsilon \approx 10^{-10}$

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)

At first, we consider only quadratic objectives of the form

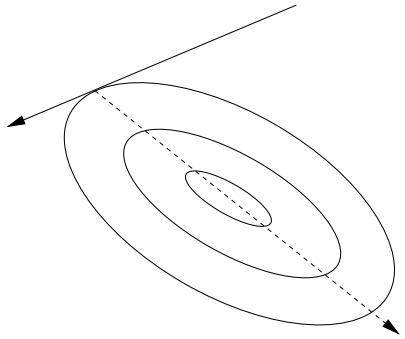
$$\begin{aligned} \min f(x) &:= \tfrac{1}{2}(x, Qx) + (b, x) \\ Q &: X \to X \text{ symm. p.d. Operator} \\ b &\in X \text{ Vector} \end{aligned}$$

obviously $\widehat{x} = -Q^{-1}b \to$, but this result is practically worthless, because we would still have to solve with Q.

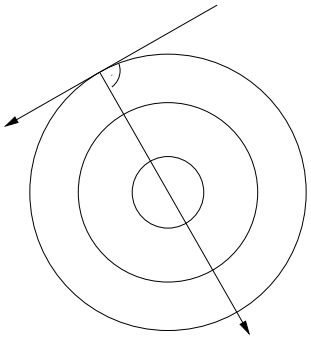
$\to$ Observation: Method is also useful for the solution of linear equations with p.d. operator Q.

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

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



quick directions not orthogonal



quick directions orthogonal

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

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

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 $(\,.,.)_Q$, 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$\{p^1, \ldots, p^{k-1}\}$(already conjugate) find conjugate direction p^k
$span\{p^1, \ldots, p^k\} = \{P_k\omega : \omega \in \mathbb{R}^k\}$, where $P_k := [p^1 | \ldots | p^k]$
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 $span\{p^1, \ldots, p^n\} = \mathbb{R}^n$
- In step ② there holds

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

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

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

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

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

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

Proof:
a)

$$\begin{aligned}[g_{k+1}^\top p_1, \ldots, g_{k+1}^\top p_k]^\top &= P_k^\top g_{k+1} = \\ &= P_k^\top(b + Qx^{k+1}) = \\ &= P_k^\top\big(\underbrace{b + Qx^k}_{=g^k} - Q(\underbrace{x^k - x^{k+1}}_{=-P_k\omega})\big) \\ &= P_k^\top g_k - P_k^\top QP_k(P_k^\top QP_k)^{-1}P_k^\top g_k = 0\end{aligned}$$

b) from a) we get $g_j^\top p_i = 0 \; \forall j > i$

Obviously: $P_k^\top g_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ p_k^\top g_k \end{pmatrix}$

Furthermore, since p_i, p_j $Q-$ conjugate $\forall i \neq j$, d.h. $p_i^\top Q p_j = 0$

$$P_k^\top Q P_k = \begin{bmatrix} p_1^\top Q p_1 & & 0 \\ & \ddots & \\ 0 & & p_k^\top Q p_k \end{bmatrix}$$

$$\Rightarrow x^{k+1} = x^k - P_k \begin{bmatrix} p_1^\top Q p_1 & & 0 \\ & \ddots & \\ 0 & & p_k^\top Q p_k \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ p_k^\top g_k \end{pmatrix}$$

$$= x^k - \frac{p_k^\top g_k}{p_k^\top Q p_k} \cdot p_k$$

□

lecture 7

Question: How to obtain the conjugate directions ?

Ansatz : $p_k = -g_k + \sum_{j=1}^{k-1} \beta_{kj} p_j \quad k \geq 2$

$p_1 = -g_1$

From theorem 3.11 a) we get for $k > i$

$$0 = g_k^\top p_i = g_k^\top \left(-g_i + \sum_{j=1}^{i-1} \beta_{ij} p_j \right) =$$

$$= -g_k^\top g_i \qquad \circledast$$

The coefficients β_{kj} are determined by the Q conjugacy of $p_1, \ldots, p_{k-1}$:

For $k > i$, there holds

$$\begin{aligned} 0 = p_i^\top Q p_k &= -p_i^\top Q g_k + \sum_{j=1}^{k-1} \beta_{kj} p_i^\top Q p_j = \\ &= -\tfrac{1}{\alpha^i} \left[Q(x_{i+1} - x_i) \right]^\top g_k + \beta_{ki} p_i^\top Q p_i \\ &= -\tfrac{1}{\alpha^i} \left[g_{i+1} - g_i \right]^\top g_k + \beta_{ki} p_i^\top Q p_i \end{aligned}$$

From $\circledast$ we get ${g_{i+1}}^\top g_k = 0 = {g_i}^\top g_k$ for all $i < k-1$

$\rightarrow$ choose $\beta_{ki} = 0$ there

$\Rightarrow$ We still only have $\beta_{k-1} := \beta_{k,k-1}$ to choose.

They are determined by the observation

$$\underbrace{(g_k - g_{k-1})^\top}_{=:y_{k-1}} p_k = \left(Q(x^k - x^{k-1})\right)^\top p_k = \alpha_{k-1} p_{k-1}^\top Q p_k = 0$$

und thus

$$0 = y_{k-1}^\top p_k \quad \overset{Ansatz}{=} \quad -y_{k-1}^\top g_k + \beta_{k-1} y_{k-1}^\top p_{k-1}$$

$$\Longrightarrow \quad \boxed{\beta_{k-1} = \frac{y_{k-1}^\top g_k}{y_{k-1}^\top p_{k-1}}}$$

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

("Hestenes-Stiefel")

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

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

Thus, β_{k-1} can be written as

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

"Polak-Ribiere" "Fletcher-Reeves"

makes a difference in nonlinear usage !

Result: 3 term recursion

Start: $\beta_1 = 0, p_1 = 0, x^1 = 0, \quad \overset{g \text{ now called } r}{r_1} = Qx^1 + b$ ('residual" = grad.)

$$\begin{array}{lcll} p_k & = & -r_k + \beta_{k-1} p_{k-1} & \\ \alpha_k & = & \|r_k\|_2^2 / (p_k^\top Q p_k) & \\ x^{k+1} & = & x^k + \alpha_k p_k & \leftarrow \quad \text{3 terms / vectors !} \\ r_{k+1} & = & r_k + \alpha_k Q p_k & \\ \beta_k & = & \|r_{k+1}\|_2^2 \ / \ \|r_k\|_2^2 & \end{array}$$

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

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

  where the so-called Arnoldi process (aka Gram-Schmidt for $\{b, Qb, Q^2b, \ldots, Q^{k-1}b\}$) 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

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

  where the so-called Lanzcos process (also aka Gram-Schmidt for $\{b, Qb, Q^2b, \ldots, Q^{k-1}$ 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. [Wou79]

* There are nonlinear variants $\big(r_k = \nabla f(x^k)\big)$, 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:

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

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

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

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 explicitely visible, only implicitely in so far, as the matrix W has to be symmetric and positive definite)

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

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

appropriate preconditioners W?

$\rightarrow$ Jacobi

$\rightarrow$ Gauß-Seidel (symm.)

$\rightarrow$ ILU

$\rightarrow$ Incomplete Cholesky

$\rightarrow$ multigrid

lecture 8

3.4 Newton method / quasi-Newton method

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

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 $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable, then, there holds for the derivative of the gradient:*

$$D\nabla f(x) = Hess f(x)$$

Proof: We analyze the action of $D\nabla f(x)$ for a vector h and its scalar product with an arbitrary vector $v \in \mathbb{R}^n$:

$$\big([D\nabla f(x)h], v\big) = \Big(\frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \nabla f(x+\varepsilon h), v\Big) =$$

$$= \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} \big(\nabla f(x+\varepsilon h), v\big) = \frac{d}{d\varepsilon}\Big|_{\varepsilon=0} Df(x+\varepsilon h)v =$$

$\nwarrow$ def. of gradient

$$= D^2 f(x)(h,v) = \big(Hess f(x)h, v\big)$$

$\nwarrow$ def. of Hessian

□

Newton method: algorithm:

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

Start at $x^0, k = 0$

solve $Hess f(x^k) d^k + \nabla f(x^k) = 0$ (Taylor)
compute $t^k := \arg\min_{0 \leq t} f(x^k + t d^k)$ (approximately)
update $x^{k+1} := x^k + t^k d^k$, $k := k + 1$

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

Observations:

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

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

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

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

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

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

$\rightarrow$ linesearch should be restricted to the intervall$t \in (0, 1)$

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

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

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

Remedies: 2 fundamentally different strategies:

① Construct $H^k \approx Hess f(x^k)$ 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

$r_{ii} + d_{ii} > \varepsilon \quad \rightarrow$ modfied Cholesky decomposition ([NW06, p. 52 ff])

lecture 9

ad ② Trust region method:

Idea:

* Instead of first computing the step direction and afterwards the step length (as in line search mathods), 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:

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

* We solve in each iteration

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

  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 $\widehat{s}$ be the unique solution of the trust region subproblem*

$$\begin{aligned} &\min_s \tfrac{1}{2}(s, Bs) + (g, s) \\ &\textit{s.t.} \quad \|s\| \le \delta \end{aligned}$$

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

(a) $(B + \lambda \cdot id)\widehat{s} = -g$

(b) $\lambda(\delta - \|\widehat{s}\|) = 0$

(c) $B + \lambda id$ *is positive definite*

(d) λ *is monotonously increasing for decreasing* δ *, if* $(B + \lambda id)$ *pos. def.*

Proof: If $\|\widehat{s}\| < \delta$, then the TR constraint is inactive, which means that muss B is positive definite.

$\Rightarrow (a), (b), (c)$ are satisfied for $\lambda = 0$ (note that λ is a scaled adjoint variable !)

Now, we assume that the TR constraint is active.

$\Rightarrow (\widehat{s}, \widehat{s}) = \delta^2 \Rightarrow (a), (b)$ satisfied via 2.24 (optimality conditions)

Ad (c):

For each arbitrary s with $\|s\|^2 = \delta^2 = \|\hat{s}\|^2$ and $s \neq \hat{s}$, we have

$$\frac{1}{2}(s, Bs) + (g, s) > \frac{1}{2}(\hat{s}, B\hat{s}) + (g, \hat{s}) + \frac{\lambda}{2}(\underbrace{(s, s) - (\hat{s}, \hat{s})}_{0})$$

Let us replace g by the knowledge $g = -(B + \lambda id)\hat{s}$. Then, it takes several equivalent transformations to result in

$$\frac{1}{2}((s - \hat{s}), (B + \lambda id)(s - \hat{s})) > 0\,, \quad \forall s : \|s\|^2 = \delta^2, s \neq \hat{s}$$

The set $\{v \,|\, v = \frac{s-\hat{s}}{\|s-\hat{s}\|}, \|s\| = \delta, s \neq \hat{s}\}$ is dense in the unit sphere (and contains a basis of the $\mathbb{R}^n$ enthält), which proofs the proposition. ✓

Ad (d): Consider $\sigma(\lambda) := \big((B + \lambda id)^{-1} g, (B + \lambda id)^{-1} g\big) = \|\widehat{s}(\lambda)\|^2$

The proposition is shown, if $\frac{d}{d\lambda}\sigma(\lambda) < 0$ holds.

$$\begin{aligned} \tfrac{d}{d\lambda}\sigma(\lambda) &= \tfrac{d}{d\lambda}\big(g, (B + \lambda id)^{-2} g\big) = \\ &= -2\big(g, (B + \lambda id)^{-3} g\big) < 0, \text{ da } (B + \lambda id)^{-3} \text{ pos.def.} \end{aligned}$$

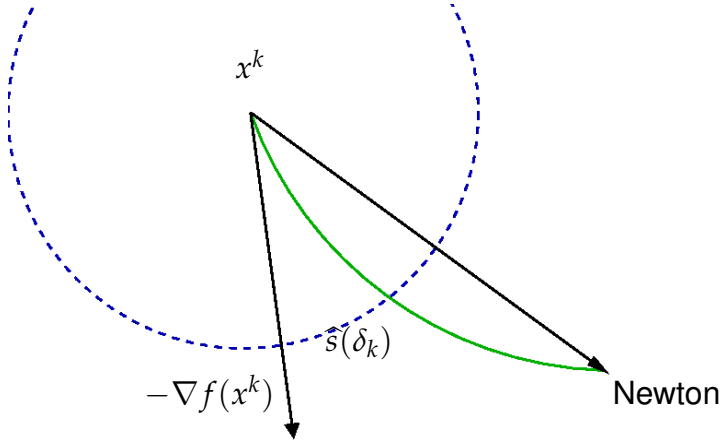
□

Remarks:

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

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

(2) Forr $\delta \searrow 0$ we have $\lambda \nearrow \infty$ and $\widehat{s}(\delta) \to -\frac{1}{\lambda} g \Rightarrow \lambda id$ dominates in $B + \lambda id$, which means that the TR step is parallel to the steepest descent.
TR produces step directions between Newton step ($B = -(Hessf)^{-1}\nabla f$) and steepest descent.



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

dog-leg , double-dogleg step etc.

Simplest strategy:

Steihaug trick (1983) [Ste83]

(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)
$\to$ 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)
$\to s^i$ is descent direction and we use $\Delta x = \alpha \cdot s^i$ with $\|\Delta x\| = \delta$

Choice of the trust region δ^k

$\to$ 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 $\Rightarrow$ δ_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 $k = 0, 1, 2...$

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

$\rightarrow$ global convergence

$\rightarrow$ if TR inactive in the vicinity of the solution
$\rightarrow$ local quadratic convergence, if $B_k = Hess f_k$

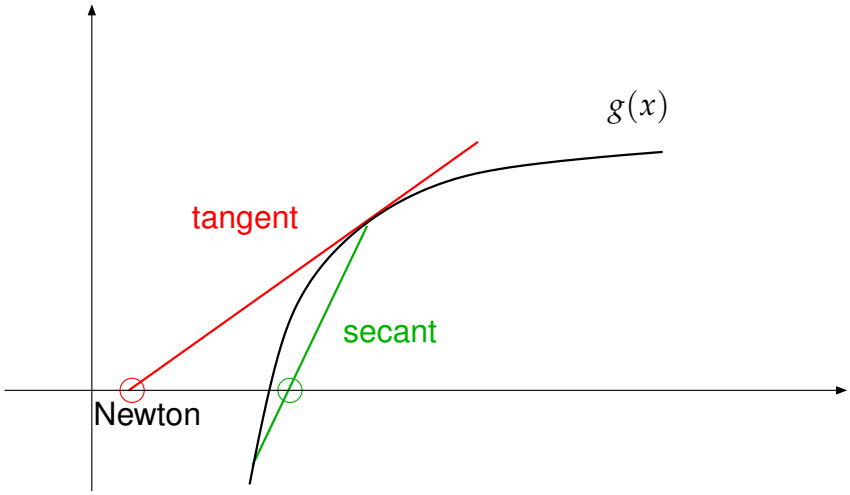
Quasi-Newton / secant method

$\rightarrow$ also "variable metric" methods

$X = \mathbb{R}^n$, Euclidian metric

Goal: approximate Hessian without explicit second derivatives.

consider in 1D the root finding problem for $0 = g(x) := f'(x)$



slope of secant as approximation of $g' = f''$:

$$m = \frac{g(x^k) - g(x^{k-1})}{x^k - x^{k-1}}$$

respectively condition for secant information

$$m(x^k - x^{k-1}) = g(x^k) - g(x^{k-1})$$

in $\mathbb{R}^n$ $\quad \boxed{B_k(x^k - x^{k-1}) = \nabla f(x^k) - \nabla f(x^{k-1})}$

" secant condition " (SC)

Problem:
B_k has n^2 entries, but (SC) determines only n.
$\Rightarrow$ further conditions necessary

Broyden formula

$$\begin{aligned} B_{k+1} \overbrace{(x^{k+1} - x^k)}^{p^k} &= \overbrace{\nabla f(x^{k+1}) - \nabla f(x^k)}^{q^k} && \text{(SC)} \\ B_{k+1}\, p &= B_k\, p \qquad \forall p \in (x^{k+1} - x^k)^\perp && \text{“no change”} \end{aligned}$$

$\Rightarrow n + n(n-1) = n^2$ conditions
$\Rightarrow_{\text{exercise}}$ 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

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

↖ *Frobenius norm*

Proof:

a) $U^B \in \mathcal{M}$ (exercise) ✓

b)

$$\begin{aligned} \|U^B - B_k\|_F &= \Big\| \tfrac{(q^k - B_k p^k)(p^k)^\top}{(p^k)^\top p^k} \Big\|_F \\ &\overset{(SC)}{=} \Big\| \tfrac{(Bp^k - B_k p^k)(p^k)^\top}{(p^k)^\top p^k} \Big\|_F \\ &= \Big\| \tfrac{(B - B_k) p^k (p^k)^\top}{(p^k)^\top p^k} \Big\|_F \\ &\overset{LA}{\leq} \|B - B_k\|_F \, \Big\| \tfrac{p^k (p^k)^\top}{(p^k)^\top p^k} \Big\|_F \overset{(*)}{=} \|B - B_k\|_F \qquad \forall B \in \mathcal{M} \end{aligned}$$

$$(*) \quad \Big\| \tfrac{p^k (p^k)^\top}{(p^k)^\top p^k} \Big\|_F = \sqrt{\mathrm{tr}\Big(\tfrac{p^k (p^k)^\top}{(p^k)^\top p^k} \tfrac{p^k (p^k)^\top}{(p^k)^\top p^k} \Big)} = \sqrt{1} = 1$$

$\Rightarrow U^B$ is smallest possible

Since $\sum_{i,j} (b_{ij} - b_{ij}^k)^2$ strictly convex and the constraint $\sum_j b_{ij} p_j = q_i$ linear, we have uniqueness of the solution.

□

Remark: Obviously also

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

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

lecture 10

The replacement of the Hessian by the Broyden update results in the following generic optimization algorithm:

Start at $x^0, k = 0$, choose initial B_0, typically $B_0 = \alpha \cdot I, \alpha > 0$

evaluate $\nabla f(x^k)$ and thus $q^{k-1} := \nabla f(x^k) - \nabla f(x^{k-1}), p^{k-1} := x^k - x^{k-1}$

update $B_k = U(B_{k-1}, p^{k-1}, q^{k-1})$

solve $B_k d^k + \nabla f(x^k) = 0$ (reminder of Taylor)

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

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

Iteration until $\|d^k\| < \varepsilon$

Here, $U(B_{k-1}, p^{k-1}, q^{k-1}) = B_{k-1} + \frac{(q^{k-1} - B_{k-1}p^{k-1})(p^{k-1})^\top}{{p^{k-1}}^\top p^{k-1}}$, i.e., the Broyden update above.
We will use this generic algorithm also for other updates, where we later only have to replace $U(B_{k-1}, p^{k-1}, q^{k-1})$ by the appropriate update formula.

(Convergence) properties Quasi Newton: Newton with Hessian updates

- global convergence together with line search or trust region. Here one has to be careful, because the Broyden search direction may not be a descent direction for f but rather for $\|\nabla f\|$. Since the Broyden update is not used in the context of optimization, we'll not go into details here.
- locally superlinear convergence (i.e., in the vicinity of the solution)
  (superlinear: $\|x^{k+1} - \hat{x}\| \le \alpha_k \|x^k - \hat{x}\|$, where $\alpha_k \to 0$ $(k \to \infty)$)
  Dennis-Moré characterization for proof of superlinear convergence

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

  where $\hat{x}$ minimum.
  $\Big($ cf. local contraction theorem for Newton method from Numerics I
  $\kappa_k := \frac{\|B_k^{-1}\left(B_k - \nabla^2 f(x^k)\right)\Delta x^k\|}{\|\Delta x^k\|} \xrightarrow[(k\to\infty)]{} 0 \Leftrightarrow$ superlinear convergence $\Big)$
- Dennis-Moré characterization is satisfied for Broyden formulas (cf. [NW06]).
- 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. $q^{k^\top} p^k > p^{k^\top} B_k p^k$.
  $\Big($ Since:

$$v^\top B_{k+1} v = \underbrace{v^\top B_k v}_{>0} + \frac{1}{\underbrace{(q^k - B_k p^k)^\top p^k}_{>0}} \cdot \big[\underbrace{(q^k - B_k p^k)^\top v}_{\geq 0}\big]^2 > 0 \quad \forall v \neq 0 \Big)$$

  $(q^k - B_k p^k)^\top p^k > 0$ cannot always be guaranteed.
  We can only hope for

$$q^{k^\top} p^k > 0,$$

  since

$$\underbrace{\nabla f(x^{k+1}) p^k}_{\substack{=0 \text{ with} \\ \text{exact line search}}} > \underbrace{\nabla f(x^k) p^k}_{\substack{<0, \text{ since } p^k \\ \text{descent direction}}}$$

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)

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

② DFP (Davidson-Fletcher-Powell)

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

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: We show the proposition only for $BFGS$

For $v \in \mathbb{R}^n \setminus \{0\}$, we see

$$\begin{aligned} v^\top B^{BFGS} v &= v^\top B v + \frac{(q^\top v)^2}{p^\top q} - \frac{(y^\top v)^2}{p^\top y} \\ &=: \quad \text{I} \quad + \quad \text{II} \quad - \quad \text{III} \end{aligned}$$

From $y = Bp$ and thus $p^\top y = p^\top Bp$ follows with $\tilde{v} := B^{\frac{1}{2}} v$ and $\tilde{p} := B^{\frac{1}{2}} p$

$$\begin{aligned} \text{I} - \text{III} &= v^\top \left(B - \frac{(Bp)(Bp)^\top}{p^\top Bp}\right) v \\ &= \tilde{v}^\top \tilde{v} - \frac{(\tilde{v}^\top \tilde{p})^2}{\tilde{p}^\top \tilde{p}} \\ &\underset{CSI}{\geq} 0 \end{aligned}$$

and $''= 0''$ only, if $\tilde{v} \parallel \tilde{p}$, i.e., $v \parallel p$, i.e., $v = \alpha p$

$$\text{II} = \frac{(q^\top v)^2}{p^\top q} \geq 0$$

For $v = \alpha p$, there holds

$$\frac{(q^\top v)^2}{p^\top q} = \frac{\alpha^2 (q^\top p)^2}{q^\top p} > 0$$

since $q^\top p > 0$ is assumed.

$\Rightarrow \quad \text{I} + \text{II} - \text{III} \qquad > 0$ $\square$

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. mit den Def.*

$$\begin{aligned} B^{BFGS} &=: U^{BFGS}(B,q,p) \\ B^{DFP} &=: U^{DFP}(B,q,p) \end{aligned}$$

gilt

$$\begin{aligned} M_{k+1}^{BFGS} &= U^{DFP}(M_k, p^k, q^k) \\ M_{k+1}^{DFP} &= U^{BFGS}(M_k, p^k, q^k) \end{aligned}$$

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 (inparticular $BFGS$) terminate at last after $\dim(X)$ steps, if the objective function is quadratic*

Limited memory updates

Idea:

Use for B_{k+1} instead of

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

only

$$B_{k+1} = \underbrace{U \circ \ldots \circ U}_{m} \circ B_k^0$$

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 (Nocedal/Wright 2006)
- Advantage: only small storage necessary $\rightarrow$ often very mild convergence deterioration in comparison with full updates. Indeed, one looses the superlinear convergence property of full BFGS. However, in many practical cases L-BFGS with $m = 3$ or $m = 5$ leads to a very good linear convergence.
- Choice of $M_k^0 := \gamma_k I$ with

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

" scaling factor " (in current direction)

lecture 11

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
$\rightarrow$ depends on parameters $q \in \mathbb{R}^n$

Data $(t_i, \bar{\alpha}_i), i = 1, \ldots, N$ 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 !

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

ε_i: measurement error

2. Ansatz: (ML estimation)
Measurement error unknown:

$\rightarrow$ 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 $X_i \sim N\big(\alpha(t_i,q),\sigma_i^2\big)$

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

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

and in independently distributed X_i for $A = A_i \times \ldots \times A_N$

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

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

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

Since the logarithm is monotone, equivalency follows

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

$$\Leftrightarrow \hat{q} = \arg\min_q \frac{1}{2}\sum_{i=1}^{N}\frac{\big(\bar{\alpha}_i-\alpha(t_i,q)\big)^2}{\sigma_i^2} \longrightarrow \text{ least squares or " } \ell_2 \text{ -estimation "}$$

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

rectangle distribution : $\hat{q} = \arg\min_q \max_i |\bar{\alpha}_i - \alpha(t_i,q)|$ $\quad \ell_\infty -$ estimator

Laplace distribution : $\hat{q} = \arg\min_q \sum_{i=1}^{N} |\bar{\alpha}_i - \alpha(t_i,q)|$ $\quad \ell_1 -$ estimator

$\rightarrow$ 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:

$F: \begin{array}{l} X \to Y \\ x \mapsto F(x) \end{array}$ with Scp $(.,.)_X$ resp. $(.,.)_Y$

and F Frechet-diffb

The central output-least-squares problem is now

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

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

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

$$\begin{array}{rcl} \frac{d}{dt}\Big|_{t=0} f(x+th) & = & \big(F(x), J(x)h\big)_Y \quad = \\ & = & \big(J(x)^{ad}F(x), h\big)_X \\ \Rightarrow \nabla f(x) & = & J(x)^{ad}F(x) \\ \text{and } Hess f(x) & = & J(x)^{ad}J(x) + DJ(x)^{ad}F(x) \\ & & \qquad\qquad\qquad\qquad \uparrow \\ & & \qquad\qquad\qquad\qquad \text{0 near solution} \end{array}$$

Lemma 3.19. *$\dim(X) < \infty$ (or $\|J(\hat{x})\| \geq c$). Then, $Hess f(\hat{x})$ 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

$$\begin{array}{l} B_k := J(x^k)^{ad}J(x) \\ \text{Solve } B_k \Delta x = -J(x^k)^{ad}F(x^k) \qquad \circledast \\ \text{Linesearch} \longrightarrow \tau \\ \text{update } x^{k+1} = x^k + \tau \Delta x \end{array}$$

Remarks: :

- Note that the Rosenbrock function is of nonlinear least squares type. Gauß-Newton behaves surprisingly well here.
- In Data Science, Gauß-Newton is sometimes called “natural gradient method”. And indeed, according to our careful discussion on the gradient in section 1, the

vector $(J(x^k)^{ad}J(x))^{-1}J(x^k)^{ad}F(x^k)$ can be considered the gradient with respect to the scalar product defined by the matrix $A = J(x^k)^{ad}J(x)$.

Lemma 3.20. *step ⊛ corresponds to the normal equations for linear F and is equivalent to*

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

Proof:

Let $\phi(\xi) := \frac{1}{2}\|F(x^k) + J(x^k)\xi\|_Y^2 \Rightarrow \nabla_\xi \phi = J(x^k)^{ad}J(x^k)\xi + J(x^k)^{ad}F(x^k) = 0$ ✓

Let $F(x)$ be affine linear in $\mathbb{R}^n$, d.h. $F(x) = Ax + b$
$\Rightarrow$ normal equations $A^\top Ax = -A^\top b$, thus exactly ⊛ □

Convergence: locally linear but with good contraction rate, if

$$F(\widehat{x}) \text{ small} \quad !$$

$\Rightarrow$ almost quadratic only with information 1st order

Levenberg-Marquardt-Method:

$$\begin{aligned} &B_k := J(x^k)^{ad}J(x) + \lambda^k I \\ &\text{solve } B_k\Delta x = -J(x^k)^{ad}F(x^k) \qquad \circledast \\ &\text{define } x^{k+1} = x^k + \Delta x \end{aligned}$$

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)

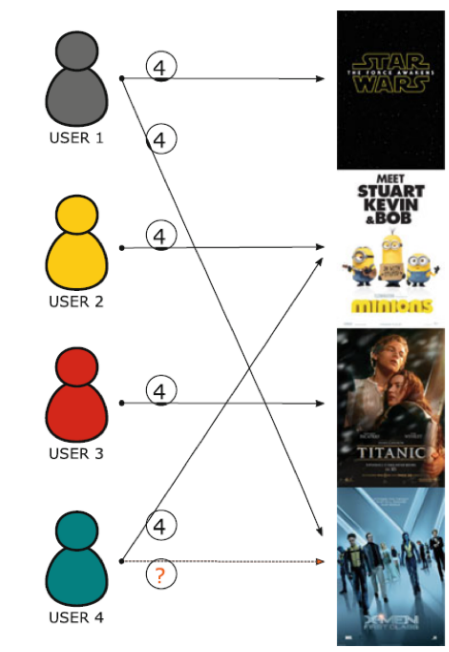
② original:
If $f(x^{k+1}) > f(x^k)$, choose $\lambda^{k+1} = 2 \cdot \lambda^k$, reject step
otherwise, accept step and choose $\lambda^{k+1} = \frac{1}{2}\lambda^k$

- 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)

lecture 12

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:

- 500,000 users
- 20,000 movies
- 100 million ratings
- Goal: to obtain lower root mean squared error (RMSE) than Netflix's existing system on 3 million held out ratings

Problem statement:

Rating matrix $R \in \mathbb{R}^{m \times n}$, where $m =$ Number of users and $n =$ 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 $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$. 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

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

with

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

The approximation problem includes already regularization terms due to the fact that the problem is severly ill-posed otherwise. Often, inequality constraints ($U, V \geq 0$) are added, which we omit for this first introduction.

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

$$R = \tilde{U}\Sigma\tilde{V}^\top, \quad \Sigma = \left[\begin{array}{ccc|ccc} \sigma_1 & & & & \vdots & \\ & \ddots & & \cdots & 0 & \cdots \\ & & \sigma_r & & \vdots & \\ \hline & \vdots & & & \vdots & \\ \cdots & 0 & \cdots & \cdots & 0 & \cdots \\ & \vdots & & & \vdots & \end{array}\right], \quad \sigma_i > 0, \forall i$$

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.

Therefor, we investigate the structure of the problem by formulating the necessary conditions. We rewrite the objective in a way, which is more convenient. We use the following matrices:

$$W_i^U = \text{diag}(\chi_\Omega(i,1), \ldots, \chi_\Omega(i,n)) \in \mathbb{R}^{n\times n}$$
$$W_j^V = \text{diag}(\chi_\Omega(1,j), \ldots, \chi_\Omega(m,j)) \in \mathbb{R}^{m\times m}$$

Therefore, the objective can be written in different forms:

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

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

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

$$\tag{3.5}$$

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

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

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

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

from which U, V can be determined row-wise by solving the linear equations defined above. Note that the necessary conditions for the rows of U are independent from each other and the same holds for the rows of V.

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

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

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

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 $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ 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 $k = 0, 1, \ldots$ ***do***

$$x_{k+1} := \operatorname*{argmin}_x f(x, y_k)$$
$$y_{k+1} := \operatorname*{argmin}_y f(x_{k+1}, y)$$

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

a) every limit point $(\hat{x}, \hat{y})$ *is a stationary point for* f

b) if $N_o := \{(x, y) \,|\, f(x, y) \leq f(x_0, y_0)\}$ *is compact, then there exists at least one limit point*

Proof. a) The proof outlines the proof of proposition 2.7.1 in [Ber99] Let $(\hat{x}, \hat{y})$ be a limit point in $\mathcal{X} \times \mathcal{Y}$ with a corresponding subsequence $\{(x_{k_j}, y_{k_j})\}_{j=0}^{\infty}$. Because of continuity of f, we observe $\lim_{j \to \infty} f(x_{k_j}, y_{k_j}) = f(\hat{x}, \hat{y})$. In a rather technical argumentation, the proof of proposition 2.7.1 in [Ber99] shows

$$(x_{k_j+1} - x_{k_j}) \to 0 \text{ , for } j \to \infty$$

and therefore $(x_{k_j+1}, y_{k_j}) \to (\hat{x}, \hat{y})$. From the definition of the iteration, we observe

$$f(x_{k_j+1}, y_{k_j}) \leq f(x, y_{k_j}) \,, \; \forall x \in \mathcal{X}$$

and thus in the limit

$$f(\hat{x}, \hat{y}) \leq f(x, \hat{y}) \,, \; \forall x \in \mathcal{X}$$

Necessary optimality conditions for optimization on convex sets of theorem 2.11 give

$$\nabla_x f(\hat{x}, \hat{y})(x - \hat{x}) \geq 0 \,, \; \forall x \in \mathcal{X}$$

Analogously, one concludes

$$\nabla_y f(\hat{x}, \hat{y})(y - \hat{y}) \geq 0 \,, \; \forall y \in \mathcal{Y}$$

and thus

$$\nabla f(\hat{x}, \hat{y}) \begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix} \geq 0 \,, \; \forall (x, y) \in \mathcal{X} \times \mathcal{Y}$$

which is stationarity.

b) since the iteration is decreasing, all iterates stay in N_o, which is assumed compact. Thus there is at least one limit point, which is stationary according to a) □

Remarks:

- Note that the proof uses the necessary condition of theorem 2.11. This theorem gives necessary and sufficient conditions for convex optimization problems. However, the proof of the necessary condition only does not rely on convexity of the objective function, but only on the convexity of the feasible set,
- 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. Mostly, random initialization is used, but 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 $k := \max\{\ell \in \mathbb{N} \,|\, \sigma_\ell \geq \theta, \ell \leq r\}$ and use

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

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

lecture 13

4 Randomized Algorithms for Artificial Neural Networks (ANN)

4.1 Interlude: essentials of back propagation

Simple first example: Consider concatenated differentiable functions $f \circ g(x)$ written in the seemingly complicated form with intermediate variable z:

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

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

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

Now,

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

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

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

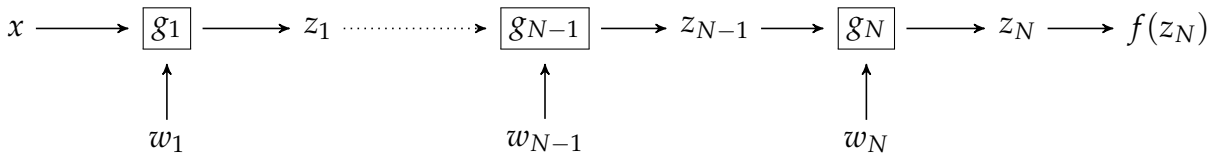
(2) $\frac{d(f \circ g)(x)}{dx} = -\lambda^\top \frac{dg}{dx}(x)$

These two steps are just a more complicated formulation of the chain rule, but this Lagrangian principle is the foundation of the backward mode of automatic differentiation and also of the back propagation in ANN, as we'll see now.

Abstract back propagation: We consider the concatenation

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

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

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

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

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

we achieve the expression

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

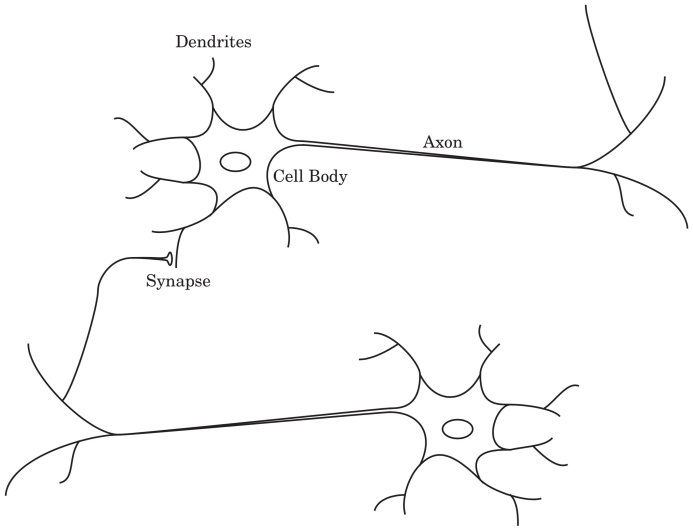
Remark: The technique presented is an example of a fairly general principle for the derivation of iterations towards complicated derivatives. This principle is used in the backward mode of automatic differentiation and in the derivation of adjoint equations for optimization problems involving differential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

4.2 Optimization aspects of ANN

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

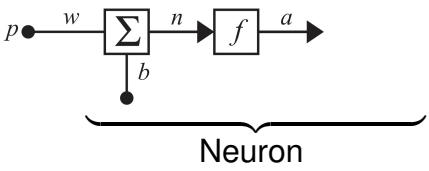
This part of the lecture is closely related to the book [HDBJ14], from which also several graphics have been taken.

Biological inspiration:



The human brain has roughly 10^{11} neurons, each of which has roughly 10^4 connections. Largest changes in later life are caused by weakening/strengthening of synaptic connections, which is used as a paradigm in ANN. However striking differences between the human brain and an electronic brain (computer) are present: even though biological neurons are very slow when compared to electrical circuits (10^{-3}s compared to 10^{-10}s), the brain is able to perform many tasks much faster than any conventional computer, and also, the brain takes up just 40W, where the electronic computer is not limited in power consumption.

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

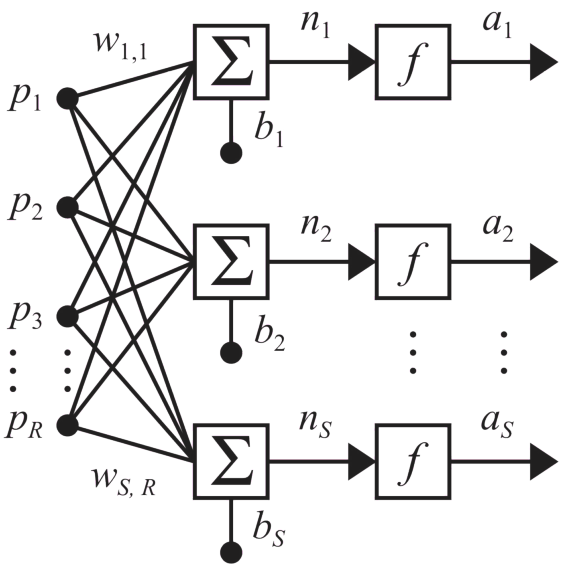
Examples of standard transfer functions are

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	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

logistic function

ReLU: rectified linear

Multiple neurons and inputs:



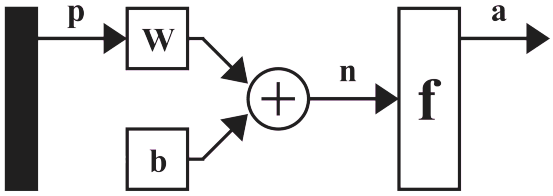
$p \in \mathbb{R}^R, n, b, a \in \mathbb{R}^S, W \in \mathbb{R}^{S \times R}$:

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

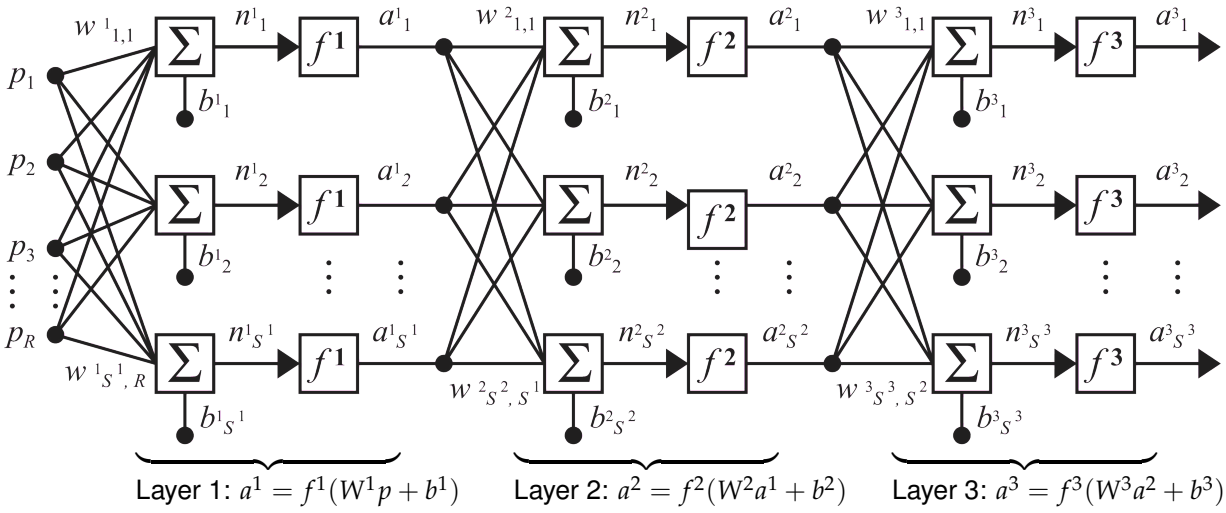
abbreviated as

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

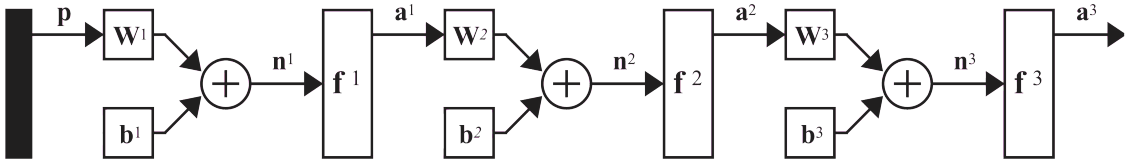
or with compact symbols



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



or with compact symbols



and in concatenated form

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

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

$$p \overset{\text{ANN}}{\mapsto} a^N =: \text{ANN}(p; W^1, \ldots, W^N, b^1, \ldots, b^N)$$

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

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

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 explicitely)

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

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

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

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

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

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

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

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

initalize $\bar{\ell} = 0, \nabla_{W_m} \bar{\ell} = 0, \nabla_{b_m} \bar{\ell} = 0\,, \quad m = 1, \ldots, N$

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

(1) forward propagation

$a^0 := p^i$

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

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

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

(2) backward propagation

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

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

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

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

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

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

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

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, tryout the usage of a GPU at GoogleColab) 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 $m = 1, \ldots, N$ **do**

$W^m_{k+1} := W^m_k - \alpha \nabla_{W_m} \ell$

$b^m_{k+1} := b^m_k - \alpha \nabla_{b_m} \ell$

Important variant:

Stochastic gradient method: the loop (0) is not executed for the whole index set $I := \{1, \ldots, q\}$ (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

lecture 14

The central tool for supervised learning is the 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

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

The step size α_k is called in the ANN terminology '" learning rate '". Thus, this iteration defines a stochastic process. Each iterated x^k depends on any previously drawn $i_{k-1}, i_{k-2}, \ldots$. We define the total expected value of a function φ of x^k as

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

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* $\mu > 0$

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

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* $M < \infty$

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

Then, the above stochastic gradient method converges with the step size sequence $\alpha_k = \theta/k$*, where* $\theta > 1/(2\mu)$ *sublinearly in the following sense:*

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

Proof: From definition 2.15 and theorem 2.16 we know that strong convexity is equivalent to

$$(x-z)^\top(\nabla f(x) - \nabla f(z)) \geq \mu\|z-x\|_2^2\,, \qquad \forall x,z\,.$$

Hence, in the case that $z = x^*$ holds, the statement

$$(x^k - x^*)^\top \nabla f(x^k) \geq \mu \|x^k - x^*\|_2^2 \,, \qquad \forall k \,.$$

The definition of the iteration results

$$\begin{aligned} \frac{1}{2} \left\| x^{k+1} - x^* \right\|_2^2 &= \frac{1}{2} \left\| x^k - \alpha_k \nabla f_{i_k}(x^k) - x^* \right\|_2^2 \\ &= \frac{1}{2} \left\| x^k - x^* \right\|_2^2 - \alpha_k (x^k - x^*)^\top \nabla f_{i_k}(x^k) + \frac{1}{2} \alpha_k^2 \left\| \nabla f_{i_k}(x^k) \right\|_2^2 \end{aligned}$$

We define $a_k := \frac{1}{2} \mathbb{E}[\left\| x^k - x^* \right\|_2^2]$ and obtain by transition to the expected value

$$a_{k+1} \leq a_k - \alpha_k \mathbb{E}[(x^k - x^*)^\top \nabla f_{i_k}(x^k)] + \frac{1}{2} \alpha_k^2 M^2$$

In the middle term, we observe that only the index of ∇f_{i_k} depends on i_k and everything else on previous indices. So

$$\begin{aligned} \mathbb{E}[(x^k - x^*)^\top \nabla f_{i_k}(x^k)] &= \mathbb{E}_{i_1} \mathbb{E}_{i_2} \dots \mathbb{E}_{i_{k-1}} \mathbb{E}_{i_k} [(x^k - x^*)^\top \nabla f_{i_k}(x^k)] \\ &= \mathbb{E}_{i_1} \mathbb{E}_{i_2} \dots \mathbb{E}_{i_{k-1}} [(x^k - x^*)^\top \mathbb{E}_{i_k} [\nabla f_{i_k}(x^k)]] \\ &= \mathbb{E}_{i_1} \mathbb{E}_{i_2} \dots \mathbb{E}_{i_{k-1}} [(x^k - x^*)^\top \nabla f(x^k)] \\ &= \mathbb{E}[(x^k - x^*)^\top \nabla f(x^k)] \\ &\geq \mathbb{E}[\mu \|x^k - x^*\|_2^2] = 2\mu a_k \end{aligned}$$

This gives us the inequality

$$a_{k+1} \leq (1 - 2\mu\alpha_k) a_k + \frac{1}{2} \alpha_k^2 M^2$$

Now [NJLS09] concludes by induction that with the choice $\alpha_k = \theta / k$ and $Q := \max\{\mathbb{E}\|x_1 - x^*\|_2^2, \frac{\theta^2 M^2}{2\mu\theta - 1}\}$ yields

$$a_k \leq \frac{Q}{2k} \,.$$

□

If we additionally assume Lipschitz continuity of ∇f, i.e., the existence of a number $L < \infty$

$$\|\nabla f(x) - \nabla f(z)\| \leq L \|x - z\| \,, \qquad \forall x, z$$

we also get convergence of the function values from the standard argument

$$\begin{aligned} f(x) - f(x^*) - \nabla f(x^*)(x - x^*) = \psi(1) - \psi(0) - \psi'(0) &= \int_0^1 \psi'(t) - \psi'(0) dt \\ &= \int_0^1 \nabla f(x^* + t(x - x^*)(x - x^*) - \nabla f(x^*)(x - x^* dt \\ &\leq \int_0^1 tL \left\| x - x^* \right\|_2^2 dt \leq \frac{L}{2} \left\| x - x^* \right\|_2^2 \end{aligned}$$

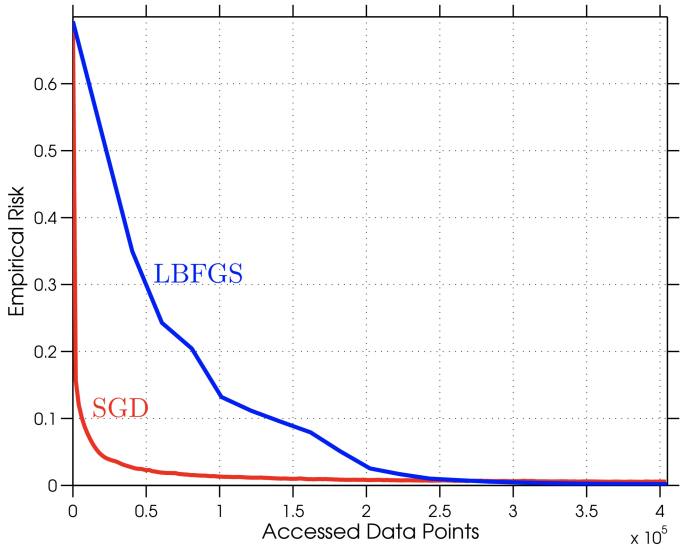
with $\psi(t) := f(x^* + t(x - x^*)$ and thus (since $\nabla f(x^*) = 0$)

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

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 stepsizes. 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 Adam (Adaptive Momentum) is used very often with good success. An overview can be obtained from [Rud17].
- Note further the awkward terminology in machine learning:
$$\text{“learning rate } \eta\text{”} \Longleftrightarrow \text{“stepsize } \alpha\text{”}$$

lecture 15


Empirical Risk
0.6
0.5
0.4
0.3
0.2
0.1
0
LBFGS
SGD
0
0.5
1
1.5
2
2.5
3
3.5
4
Accessed Data Points
x 10^5

Interlude on methods based on function evaluations only

Up to this point, we have discussed unconstrained optimization methods based on derivatives. Of course, there are also optimization methods for unconstrained optimization problems, which rely only on evaluations of the objective function f and do not require any gradients. These methods are very slow and often of heuristic nature. Here, I want to mention only the following:

- Nelder-Mead/Simplex/Poytope method: it is a deterministic method, which I personally have also used with good success, but it is very slow
- Genetic algorithms/Evolutionary algorithms: It is a heuristic approach of stochastic nature. Advocates of the methods claim that it reaches global optima rather than only local minima, even in nonconvex cases. These algorithms are biologically inspired, very slow, limited to only a small number of variables and require intensive parameter tuning.
- Simulated annealing: another stochastic approach. It can be shown that the iterations converge to the global minimum with probability one. Nevertheless, the methodology is also very slow and requires a lot of f-evaluations.
- there are more nature inspired algorithms like ant colonies, tabu search ....

For the sake of brevity, we skip these methodologies, also because they are not in the core of numerical optimization. The reader may find lots of information, when googling for the keywords mentioned above.

5 Linear quadratic programming

Example: 1. Portfolio Optimization

n Investment opportunities with

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

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

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

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

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

κ : Risk tolerance parameters $\rightarrow$ customer specific

Optimal portfolio characterized by

$$\begin{array}{lll} \max\limits_{x} & x^\top r - \kappa x^\top B x & \\ \text{s.t.} & \sum\limits_{i=1}^{n} x_i = 1 & \text{(equality constraints)} \\ & x \geq 0 & \text{(inequality constraints)} \end{array}$$

Example: 2. LASSO (least absolute shrinkage and selection operator): ℓ_2 least squares problem with ℓ_1 regularization, i.e., the following convex optimization problem

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

where $\|x\|_1 = \sum_{i=1}^n |x_i|$. If λ is the adjoint to the problem

$$\begin{aligned} \min_x & \frac{1}{2} \|Ax - b\|_2^2 \\ \text{s.t. } & \|x\|_1 \leq t \end{aligned}$$

for some $t > 0$ to be chosen, then both problems are equivalent. By using additional variables x_i^+, x_i^-, this can be rephrased as a standard QP

$$\begin{aligned} \min_x & \frac{1}{2} \left\|A(x^+ - x^-) - b\right\|_2^2 \\ \text{s.t. } & \sum_{i=1}^{n} x_i^+ + x_i^- \leq t \\ & x_i^+ \geq 0, \quad x_i^- \geq 0 \ \forall i \end{aligned}$$

Another way of formulating an equivalent LASSO problem is the problem

$$\begin{aligned} \min_{x,v} & \frac{1}{2} \|Ax - b\|_2^2 + \lambda \sum_i v_i \\ \text{s.t. } & -v_i \leq x_i \leq v_i \ \forall i \end{aligned}$$

5.1 QP with equality constraints

$$\boxed{\begin{array}{l} \min \frac{1}{2} x^\top B x + b^\top x \\ \text{s.t. } Cx + c = 0 \end{array}} \qquad \begin{array}{l} B \in \mathbb{R}^{n \times n} \text{ symmetric} \\ C \in \mathbb{R}^{m \times n} \end{array}$$

$b \in \mathbb{R}^n, c \in \mathbb{R}^m$

Recall: LICQ and uniqueness of solution $\Leftrightarrow$

C surjective (resp. full rank) and

$$v^\top B v > 0, \forall v \in \ker C \backslash \{0\}$$

5.1.1 QP with positive definite quadratic form

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

necessary conditions: $\swarrow$ "KKT Matrix

$$\begin{matrix} \text{(I)} \\ \text{(II)} \end{matrix} \begin{bmatrix} B & C^\top \\ C & 0 \end{bmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} -b \\ -c \end{pmatrix}$$

B p.d. $\Rightarrow B$ invertible $\rightarrow$ Block Gauß elimination
from (I) we conclude $x = -B^{-1}(C^\top \lambda + b)$
using this in (II) results in linear system for $\lambda (\in$ range space of C [dual space])

$$\underbrace{CB^{-1}C^\top}_{A \text{ p.d.}} \lambda = \underbrace{c - CB^{-1}b}_{\alpha} \qquad \text{"range space method"}$$

$\Rightarrow$ Block decomposition:

$$\begin{bmatrix} B & C^\top \\ C & 0 \end{bmatrix} = \begin{bmatrix} B & 0 \\ C & S \end{bmatrix} \begin{bmatrix} I & B^{-1}C^\top \\ 0 & I \end{bmatrix} \quad \text{mit } S := -CB^{-1}C^\top \quad \text{"Schur complement"}$$

range space system $\Leftrightarrow \min_\lambda \frac{1}{2}\lambda^\top A \lambda - \alpha^\top \lambda$

Application of steepest descent method gives

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

Thus, we obtain the following iteration in both variables

$$\begin{aligned} x^{k+1} &= -B^{-1}(C^\top \lambda^k + b) = x^k - B^{-1}(Bx^k + C^\top \lambda^k - b) \\ \lambda^{k+1} &= \lambda^k + \tau(Cx^{k+1} + c) \qquad \nwarrow \text{approximation gives inexact Uzawa} \end{aligned}$$

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)

$\rightarrow$ is convergent, if τ is small enough

5.1.2 QP with indefinite quadratic form

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

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

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

Proof: row rank = column rank

$\Rightarrow$ there are m linearly independent columns in C .

determine permutation P such that the linearly independent columns appear in the front

. □

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 $Cx = C_1 y + C_2 z$

analogously $P^\top B P = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$ with $B_{12} = B_{21}^\top$

$$P^\top b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$\Rightarrow$ necessary conditions in new variables: $\swarrow$ "KKT" Matrix

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

Unique existence of solution is guaranteed, if

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

$$\begin{aligned} \ker(C) &= \left\{ \begin{pmatrix} y \\ z \end{pmatrix} \,\middle|\, C_1 y + C_2 z = 0 \right\} = \\ &= \left\{ \begin{pmatrix} y \\ z \end{pmatrix} \,\middle|\, y = -C_1^{-1} C_2 z \,, z \in \mathbb{R}^{n-m} \right\} = \\ &= \left\{ \begin{bmatrix} -C_1^{-1} C_2 \\ I \end{bmatrix} z \,\middle|\, z \in \mathbb{R}^{n-m} \right\} \end{aligned}$$

$\Rightarrow$ necessary condition of 2nd order $\Leftrightarrow$

$$0 < z^\top \begin{bmatrix} -C_1^{-1} C_2 \\ I \end{bmatrix}^\top \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} -C_1^{-1} C_2 \\ I \end{bmatrix} z$$

$$= z^\top \Bigg(\underbrace{B_{22} - C_2^\top C_1^{-\top} B_{12} - B_{21} C_1^{-1} C_2 + C_2^\top C_1^{-\top} B_{11} C_1^{-1} C_2}_{\text{"reduced Hessian" } H} \Bigg) z \quad \forall z \in \mathbb{R}^{n-m} \backslash \{0\}$$

$\Rightarrow$ 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 $\circledast$ is equivalent to

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

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:

$$\begin{bmatrix} B_{11} & C_1^\top \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & C_1^{-1} \\ C_1^{-\top} & -C_1^{-\top} B_{11} C_1^{-1} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

□

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

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

Using this in the remaining equation gives

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

$$\Longleftrightarrow$$

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

$$-b_2 + B_{21}C_1^{-1}c + C_2^\top C_1^{-\top} b_1 - C_2^\top C_1^{-\top} B_{11} C_1^{-1} c$$

According to sufficient cond. of 2nd order, follows that S is pos.def., thus invertible $\Rightarrow$ Schur complement system is solvable.

The Schur complement system is formulated for z, and z spans the nullspace of $[C_1 C_2]$ aufspannt, which is the reason, why this method is called null space method.

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.

lecture 16

5.2 QP with inequality constraints

For ease of presentation, we consider the problem class

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

Remark: Additional equality constraints can be simply added as double inequality constraints.

Recall: necessary and (in convex QP also sufficient) conditions for $\min_x \tfrac{1}{2}\, x^\top Bx + b^\top x$ s.t. $Cx + c \leq 0$

$\exists\ \underline{\lambda \geq 0}$ componentenwise with $\left(\mathscr{L} = \tfrac{1}{2}\, x^\top Bx + b^\top x + \lambda^\top (Cx + x)\right)$

$$\nabla_x \mathscr{L}(x,\lambda) = Bx + b + C^\top \lambda = 0$$
$$\lambda^\top (Cx + c) = 0 \quad \text{(complementarity)}$$

When we gave found the set of active constraints

$$\widehat{I} = \left\{ i \in \{1,\ldots,m\} \middle| (C\widehat{x} + c)_i = 0 \right\} \ (\widehat{x} \text{ solution})$$

we only have to solve

$$\begin{aligned} &\min_x \tfrac{1}{2}\, x^\top Bx + b^\top x \\ &\text{s.t. } (Cx + c)_i = 0 \ \forall i \in \widehat{I} \end{aligned}$$ which is only an equality constrained problem.

Active set strategy:

$\rightarrow$ 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

⓪ Determine feasible point x^0 with $Cx^0 + c \leq 0$ (see below)
Define I^0 by $I^0 := \{i \big| (Cx^0 + c)_i = 0\}$

① For the index set I^k compute improving step $x^k + d^k$ via

$$\min_d \tfrac{1}{2}\,(x^k+d)^\top B(x^k+d)+b^\top(x^k+d)$$
$$\text{s.t. } \big((C(x^k+d)+c\big)_i = 0\ \forall i\in I^k$$

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

② If $d^k \neq 0$ compute

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

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 $\lambda_i \geq 0\ \forall i\in I^k \ \longrightarrow$ ready, solution found
else $\exists\lambda_j < 0 \Rightarrow$ remove constraint j with largest $|\lambda_j|$.

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

Objective function decreases monotonically

$\Rightarrow$ Algorithm is not cycling $\Rightarrow$ ready after finitely many steps.

Supplement to ⓪ : typical Ansatz:

Phase I Problem of linearen Programming

Solve $\min_x \sum_{i=1}^n x_i$ s.t. $Cx + c \leq 0$ This is an LP, which can be solved, e.g., by SIMPLEX method ($\rightarrow$ 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][2]

- Sometimes: crossover from IPM $\rightarrow$ AS
- Another alternative mainly in the context of partial differential equations and variational inequalities is semismooth-Newton aka primal-dual active set method

lecture 17

[2] 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 appraoch:
  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) $\begin{array}{ll} \min f(x) & f: X \to \mathbb{R} \\ \text{s.t. } c(x) = 0 & g: X \to Y \end{array}$ X, Y linear spaces

We consider a family fo solutions $x(\mu)$ of the problem
$x(\mu) = \arg\min_x f(x) + \frac{\mu}{2}\big(c(x), c(x)\big)_Y$
and show $x(\mu) \overset{\mu\to\infty}{\longrightarrow} \widehat{x} =$ solution of NLP

Example:

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

Solution at $x = \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix}$

Penalty function: $P(x,\mu) = f(x) + \frac{\mu}{2}(2 - x_1 - x_2)^2$

$$\nabla_x P(x) = \underbrace{\begin{bmatrix} 2+\mu & 4+\mu \\ 4+\mu & 10+\mu \end{bmatrix}}_{\text{Hessian}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 10+2\mu \\ 20+2\mu \end{pmatrix} = 0$$

Solution: $x = \begin{pmatrix} \dfrac{10+\mu}{2+2\mu} \\ \\ \dfrac{6\mu + \frac{3}{2}\mu^2}{4+5\mu+\mu^2} \end{pmatrix} \overset{\mu\to\infty}{\longrightarrow} \begin{pmatrix} \frac{1}{2} \\ \\ \frac{3}{2} \end{pmatrix} = \widehat{x}$

Danger: System matrix $\begin{bmatrix} 2+\mu & 4+\mu \\ 4+\mu & 10+\mu \end{bmatrix} \overset{\mu\to\infty}{\longrightarrow} \mu \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

Condition is getting bad !

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

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

① determine $x^k := \arg\min_x P(x, \mu^k)$

② if $\|c(x^k)\| \leq$ TOL $\Rightarrow$ STOP, ready

③ choose $\mu^{k+1} > \mu^k$ e.g. $\mu^{k+1} = 10 \cdot \mu^k$

Theorem 6.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ 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 $\mu^k \to \infty$. Furthermore, for each $\mu^k > 0$ there is an $x^k \in \mathbb{R}^n$ with*

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

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) $\lim_{k\to\infty} c(x^k) = 0$

e) Every accumulation point of $\{x^k\}_{k=0}^{\infty}$ is a solution of

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

Proof:

a) x^k solves $\min_x P(x, \mu^k)$

$$\Rightarrow\ P(x^k, \mu^k) \le P(x, \mu^k),\ \forall x \in X, \text{ also for } x^{k+1}$$

$$\Rightarrow\ P(x^k, \mu^k) \le P(x^{k+1}, \mu^k) = f(x^{k+1}) + \frac{\mu^k}{2}\big(c(x^{k+1}), c(x^{k+1})\big)$$

$$\overset{\mu^{k+1} > \mu^k}{\le} f(x^{k+1}) + \frac{\mu^{k+1}}{2}\big(c(x^{k+1}), c(x^{k+1})\big)$$

$$= P(x^{k+1}, \mu^{k+1}) \qquad \checkmark$$

b) with the arguments from a) we know:
$P(x^k, \mu^k) \le P(x^{k+1}, \mu^k)$ and $P(x^{k+1}, \mu^{k+1}) \le P(x^k, \mu^{k+1})$

Summation gives:

$$P(x^k, \mu^k) + P(x^{k+1}, \mu^{k+1}) \le P(x^{k+1}, \mu^k) + P(x^k, \mu^{k+1})$$

$$\Leftrightarrow\ \mu^k \|c(x^k)\|^2 + \mu^{k+1}\|c(x^{k+1})\|^2 \le \mu^k \|c(x^{k+1})\|^2 + \mu^{k+1}\|c(x^k)\|^2$$

$$\Leftrightarrow\ \underbrace{(\mu^k - \mu^{k+1})}_{<0}\ \underbrace{(\|c(x^k)\|^2 - \|c(x^{k+1})\|^2)}_{\ge 0} \le 0$$

$$\Rightarrow\ \|c(x^k)\| \ge \|c(x^{k+1})\| \qquad \checkmark$$

c) $P(x^k, \mu^k) \le P(x^{k+1}, \mu^k)$

$$\Rightarrow\ 0 \le f(x^{k+1}) - f(x^k) + \frac{\mu^k}{2}\ \underbrace{(\|c(x^{k+1})\|^2 - \|c(x^k)\|^2)}_{\le 0}$$

$$\le f(x^{k+1}) - f(x^k) \qquad \checkmark$$

d) Obviously for each k

$$f(x^0) \le f(x^0) + \frac{\mu^k}{2}\ \|c(x^k)\|^2 \le f(x^k) + \frac{\mu^k}{2}\ \|c(x^k)\|^2$$

$$= P(x^k, \mu^k) \le \inf_{c(x)=0} P(x, \mu^k) \le \inf_{c(x)=0} f(x) =: \underline{f}$$

$$\Rightarrow\ \mu^k \|c(x^k)\|^2 \le 2 \cdot \big(\underline{f} - f(x^0)\big)\ \forall k$$

$$\Rightarrow\ \{\mu^k \|c(x^k)\|^2\} \quad \text{is bounded, but}$$

$$\mu^k \to \infty \Rightarrow \|c(x^k)\| \to 0$$

e) Let x^* be an accumulation point of $\{x^k\}$, $x^{k_j} \to x^*$ from d) we know $\lim_{j\to\infty} c(x^{k_j}) = 0 \Rightarrow x^*$ admissible

and $f(x^{k_j}) \leq f(x^{k_j}) + \frac{\mu^{k_j}}{2} \|c(x^{k_j})\|^2 \overset{\text{analogously to above}}{\leq} \inf_{c(x)=0} f(x) = \underline{f}$

and $f(x^*) = \lim_{j\to\infty} f(x^{k_j}) \left(\leq \lim_{j\to\infty} P(x^{k_j}, \mu^{k_h}) \right)$

$\leq \inf_{c(x)=0} f(x) = \underline{f}$

Since $x^* \in X$, there holds $\underline{f} = \inf_{c(x)=0} f(x) \leq f(x^*) \leq \underline{f}$ □

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. [NW06, Theorem 17.2]).

Surprising observation:

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

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

Then there holds

a) $\lim_{k\to\infty} \mu^k c(x^k) = \widehat{\lambda} = -\left(J(\widehat{x})J(\widehat{x})^{ad}\right)^{-1} J(\widehat{x})\nabla f(\widehat{x})$

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

$$\nabla f(\widehat{x}) + J(\widehat{x})^{ad}\widehat{\lambda} = 0 \quad \textit{(nec. cond.)}$$
$$c(\widehat{x}) = 0$$

Remark: From necessary conditions, we know

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

Proof:

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

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

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

x^k Minimum $\Rightarrow$

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

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

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

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

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*

Then there holds $\operatorname{cond}\left(\nabla^2 P(x^k, \mu^k)\right) = \dfrac{\lambda^k_{\max}}{\lambda^k_{\min}} \to \infty$

Proof: $x^k \to \hat{x}$

$\Rightarrow \nabla^2 f(x^k) =: A^k \to \nabla^2 f(\hat{x}) =: A$

$\nabla^2 \|c(x^k)\|^2 = J(x^k)^\top J(x^k) + DJ(x^k)^\top \underbrace{c(x^k)}_{\searrow 0}$

$\to J(\hat{x})^\top J(\hat{x}) =: J^\top J$

obviously $J^\top J$ pos. semidef. und $\lambda_{\min}(J^\top J) = 0$

$\Rightarrow \lambda_{\max}\left(\nabla^2 P(x^k, \mu^k)\right) =$

$\max_{\|z\|=1} \left[z^\top A^k z + \mu_2^k \underbrace{J_k^\top J_k}_{J^\top J \text{ mit } \lambda_{\max}(J^\top J)>0} z\right] + \mathcal{O}\left(c(x^k)\right) \longrightarrow \infty$

$\lambda_{\min}\left(\nabla^2 P(x^k, \mu^k)\right) =$

$$= \min_{\|z\|=1} (z^\top A^k z + \mu^k z J_k^\top J_k z) + \mathcal{O}\big(c(x^k)\big)$$

$$\leq \min_{\substack{\|z\|=1 \\ J_k z = 0}} (z^\top A^k z + \mu^k \underbrace{z J_k^\top J_k z}_{0}) \to \ \lambda_{\min}(A) < \infty$$

□

Quadratic penalty function with inequalities

$$\begin{aligned} &\min_x f(x) && f, c_i, h_i : \mathbb{R}^n \to \mathbb{R} \\ &\text{s.t. } c_i(x) = 0, i = 1, \ldots, m \\ &\qquad h_i(x) \leq 0, i = 1, \ldots, l \end{aligned}$$

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 tp solution comparable
- Attention: $P(x, \mu)$ is only once differentiable !

ℓ_1 - Penalty Function

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

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*

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

i.e. $\widehat{x}$ *Solution of NLP* $\circledast$

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

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

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

$$P_1(\widehat{x}+\Delta x,\mu) \overset{\text{Taylor}}{=}$$

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

directional derivative

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

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

$\Rightarrow$ $\widehat{x}$ 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

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

with $\alpha_i > |\widehat{\lambda}_i| > 0$

lecture 18

Augmented Lagrangian

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

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

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

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

Proof: necessary optimality condition for P_{AL} is

$$0 = \nabla_x P_{AL}(x,\mu) = \nabla_x f(\widehat{x}) + Dc(\widehat{x})^\top \widehat{\lambda} + \mu Dc(\widehat{x})^\top c(\widehat{x})$$
$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \| \\ 0$$

$\Rightarrow$ is satisfied by $\widehat{x}$

For μ sufficiently large $\mu > \mu_0$ is $\nabla^2 P_{AL}(\widehat{x},\mu)$ positive definite $\Rightarrow \widehat{x}$ minimum cf. proof of TH. 17.5 in [NW06]) □

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

Successive estimation:

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

obviously, there holds for λ arbitrary:

$\nabla_x \mathscr{L}_A(x,\lambda;\mu) = \nabla_x f(x) + Dc(x)^\top \big(\lambda + \mu \cdot c(x)\big)$

Thus, $x \to \widehat{x}$, if $\lambda + \mu \cdot c(x) \to \widehat{\lambda}$

$\Rightarrow$ 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

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

① Compute x^k with
$\|\nabla_x \mathscr{L}_A(x^k, \lambda^k)\| \leq \omega^k$

② 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.

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

③ 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 ①and ②small enough

cf. Conu / Gould / Toint: LANCELOT [CGT92]
Large
And
Nonlinear
Constrained
Extended
Lagrangian
Optimization
Techniques

most recent version: $\longrightarrow$ GALAHAD [GT03]

Features:

- very robust
- globally convergent
- easy implementation

- less efficient than SQP methods (below)

Generalization to inequalities via slack variables with bound constraints
$\rightarrow$ [NW06, Chap 17.4]

Barrier methods / interior point methods

$\min f(x)$
s.t. $c(x) \leq 0$

Definition 6.10. *Barrier function: (logarithmic)*

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

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/Marrocco 1975 and Gabay/Mercier 1976)

ADMM has been invented in the 1970s for the numerical solution of particular finite element problems and has recently gained significant interest in the data science community. An excellent survey with many data science applications can be found in [BPC+10], on which we base the discussion here.

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 $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ 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. (The paper [MXAP11] even discusses the case, where x and z are restricted to polygonal sets.)

We build the augmented Lagrangian

$$\mathscr{L}_A(x,z,\lambda;\mu) := f(x) + g(z) + \lambda^\top(Ax + Bz - c) + \frac{\mu}{2}\|Ax + Bz - c\|_2^2$$

The classical method of multipliers discussed above, now takes the form

$$\begin{aligned} (x^{k+1}, z^{k+1}) &= \arg\min_{x,z} \mathscr{L}_A(x,z,\lambda^k;\mu) \\ \lambda^{k+1} &= \lambda^k + \mu(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

where we fix μ here. Now, we borrow the idea of block iterations discussed in section 3.6 (similar to Gauß-Seidel) for the first step in this algorithm and change it (not equivalent!) to the so-called *Alternating Direction Method of Multipliers* (ADMM)

$$\begin{aligned} x^{k+1} &= \arg\min_x \mathscr{L}_A(x,z^k,\lambda^k;\mu) &&= \arg\min_x f(x) + (\lambda^k)^\top Ax + \frac{\mu}{2}\|Ax + Bz^k - c\|_2^2 \\ z^{k+1} &= \arg\min_z \mathscr{L}_A(x^{k+1},z,\lambda^k;\mu) &&= \arg\min_z g(z) + (\lambda^k)^\top Bz + \frac{\mu}{2}\|Ax^{k+1} + Bz - c\|_2^2 \\ \lambda^{k+1} &= \lambda^k + \mu(Ax^{k+1} + Bz^{k+1} - c) \end{aligned}$$

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., $\lim_{k\to\infty}(x^k, z^k, \lambda^k) = (x^*, z^*, \lambda^*)$ 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 well 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 ($y^k \to y^*$ r-linearly, if there exists a sequence $\{\sigma^k\}_{k=1}^\infty$ converging (Q-)linearly to 0, such that $\|y^k - y^*\| \leq \sigma^k$).
- However, simple examples show that ADMM can be very slow to converge to high accuracy. However, it is often the case that ADMM converges to modest accuracy—sufficient for many applications—within a few tens of iterations. This behavior makes ADMM similar to algorithms like the conjugate gradient method, for example, in that a few tens of iterations will often produce acceptable results of practical use.

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

$$\begin{aligned} &\min_{(x,z)} \frac{1}{2} \|Ax - b\|_2^2 + \rho \|z\|_1 \ , \quad \rho > 0 \\ &\text{s.t. } x - z = 0 \end{aligned}$$

Thus, the resulting ADMM iteration is

$$\begin{aligned} x^{k+1} &= \arg\min_x \frac{1}{2} \|Ax - b\|_2^2 + (\lambda^k)^\top x + \frac{\mu}{2} \|x - z^k\|_2^2 \\ z^{k+1} &= \arg\min_z \rho \|z\|_1 - (\lambda^k)^\top z + \frac{\mu}{2} \|x^{k+1} - z\|_2^2 \\ \lambda^{k+1} &= \lambda^k + \mu(x^{k+1} - z^{k+1}) \end{aligned}$$

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

$$\partial_{z_i}|z_i| = \begin{cases} -1 & \text{, for } z_i < 0 \\ [-1,1] & \text{, for } z_i = 0 \\ 1 & \text{, for } z_i > 0 \end{cases}$$

Now, in the case $z_i < 0$, we have

$$0 = -\rho - \lambda_i^k - \mu(x_i^{k+1} - z_i)$$

which leads to

$$z_i = x_i^{k+1} + \frac{\lambda_i^k}{\mu} + \frac{\rho}{\mu}, \text{ if } x_i^{k+1} + \frac{\lambda_i^k}{\mu} < -\frac{\rho}{\mu}$$

Analogously, if $z_i > 0$

$$z_i = x_i^{k+1} + \frac{\lambda_i^k}{\mu} - \frac{\rho}{\mu}, \text{ if } x_i^{k+1} + \frac{\lambda_i^k}{\mu} > \frac{\rho}{\mu}$$

Now, the case $z_i = 0$ means

$$0 \in \rho[-1,1] - \lambda_i^k - \mu x_i^{k+1}$$

which is equivalent to

$$x_i^{k+1} \in \frac{\rho}{\mu}[-1,1] - \frac{\lambda_i^k}{\mu} \Leftrightarrow x_i^{k+1}\frac{\lambda_i^k}{\mu} \in [\frac{-\rho}{\mu}, \frac{\rho}{\mu}]$$

Thus, in total, we get for each $i = 1, \ldots, n$

$$z_i = S_{\rho/\mu}\left(\frac{\lambda_i^k}{\mu} + x_i^{k+1}\right)$$

where

$$S_\kappa(a) := \begin{cases} a + \kappa & \text{, for } a < -\kappa \\ 0 & \text{, for } |a| < \kappa \\ a - \kappa & \text{, for } a > \kappa \end{cases}$$

is the so-called soft threshold operator. In total the resulting ADMM iteration for the LASSO problem can be written as

$$\begin{aligned} x^{k+1} &= (A^\top A + \mu I)^{-1}(A^\top b + \mu z^k - \lambda^k) \\ z_i^{k+1} &= S_{\rho/\mu}\left(\frac{\lambda_i^k}{\mu} + x_i^{k+1}\right), \ \forall i = 1, \ldots, n \\ \lambda^{k+1} &= \lambda^k + \mu(x^{k+1} - z^{k+1}) \end{aligned}$$

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

$$\mathscr{L}_A(x_1,\ldots,x_m,z,\lambda_1,\ldots,\lambda_m;\mu) = \sum_{j=1}^m \left(f_j(x) + \lambda_j^\top (x_j - z) + \frac{\mu}{2}\|x_j - z\|_2^2\right)$$

and the resulting ADMM iteration

$$\begin{aligned} x_j^{k+1} &= \arg\min_{x_j} f_j(x_j) + (\lambda_j^k)^\top x_j + \frac{\mu}{2}\|x_j - z^k\|_2^2\,,\ \forall j = 1,\ldots,m \\ z_i^{k+1} &= \frac{1}{m}\sum_{j=1}^m \left(x_j^{k+1} + \frac{1}{\mu}\lambda_j^k\right) \\ \lambda_j^{k+1} &= \lambda^k + \mu(x_j^{k+1} - z^{k+1})\,,\ \forall j = 1,\ldots,m \end{aligned}$$

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

$$\begin{aligned} x_j^{k+1} &= \arg\min_{x_j} f_j(x_j) + (\lambda_j^k)^\top x_j + \frac{\mu}{2}\|x_j - \bar{x}^k\|_2^2\,,\ \forall j = 1,\ldots,m \\ \lambda_j^{k+1} &= \lambda^k + \mu(x_j^{k+1} - \bar{x}^{k+1})\,,\ \forall j = 1,\ldots,m \end{aligned}$$

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]

lecture 19

6.3 SQP Methods

Sequential

Quadratic **P**rogramming

Succesive

We start with equality constraints

$$\begin{array}{ll} \min\limits_x f(x) & f : \mathbb{R}^n \to \mathbb{R} \\ \text{s.t. } c(x) = 0 & c : \mathbb{R}^n \to \mathbb{R}^m \end{array}$$

KKT conditions $\left(\mathscr{L}(x,\lambda) = f(x) + \lambda^\top c(x)\right)$

$$F(x,\lambda) := \begin{pmatrix} \Delta f(x) + Dc(x)^\top \lambda \\ c(x) \end{pmatrix} = 0$$

nonlinear root finding problem $\longrightarrow$ Newton

Iteration $\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \begin{pmatrix} \Delta x^k \\ \Delta\lambda^k \end{pmatrix}$ with $\quad C_k := Dc(x^k)$

$$\left[\begin{array}{cc} H_k & C_k^\top \\ C_k & 0 \end{array} \right] \left(\begin{array}{c} \Delta x^k \\ \Delta\lambda^k \end{array} \right) = \left(\begin{array}{c} -\nabla f^k - C_k^\top \lambda^k \\ -c^k \end{array} \right) \circledast$$

with $H_k = Hess_x \mathscr{L}(x^k, \lambda^k) =$ Hessian of the Lagrangian

Newton $\Rightarrow$ local quadratic convergence !

$\rightarrow$ what is quadratic in this linear problem ?

We observe the equivalent formulation

$$\circledast \Leftrightarrow \left[\begin{array}{cc} H_k & C_k^\top \\ C_k & 0 \end{array} \right] \left(\begin{array}{c} \Delta x^k \\ \lambda^{k+1} - \lambda^k \end{array} \right) = \left(\begin{array}{c} -\nabla f^k - C_k^\top \lambda^k \\ -c^k \end{array} \right)$$

$$\Leftrightarrow \left[\begin{array}{cc} H_k & C_k^\top \\ C_k & 0 \end{array} \right] \left(\begin{array}{c} \nabla x^k \\ \lambda^{k+1} \end{array} \right) = \left(\begin{array}{c} -\nabla f^k \\ -c^k \end{array} \right)$$

$$\Leftrightarrow \qquad \min_{\Delta x^k} \tfrac{1}{2}(\Delta x^k)^\top H_k \Delta x^k + (\nabla f^k)^\top \Delta x^k \quad \text{s.t. } C_k \Delta x^k + c^k = 0 \qquad (QP)$$

The adjoint variable of the QP is used as a new estimate of λ.

Δx^k is possibly changed by line-search or TR.

Generalization to problems with inequalities: $\min_x f(x)$ s.t. $c(x) = 0$, $g(x) \leq 0$

Iteration: [in the vicinity of the solution]

$x^{k+1} = x^k + \Delta x^k$ with

$$\min_{\Delta x^k} \tfrac{1}{2}(\Delta x^k)^\top H_k \Delta x^k + (\nabla f^k)^\top \Delta x^k \quad \text{unter } C_k \Delta x^k + c^k = 0,\ G_k \Delta x^k + g^k \leq 0$$

(QP, compare section 5)

Hence the name SQP !

$$H_k \approx Hess_x\big(f(x) + \lambda^\top c(x) + \mu^\top g(x)\big)$$

Convergence:

Theorem 6.13. *Be $\widehat{x}$ 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 $(\widehat{x}, \widehat{\lambda})$ has the same active-set as $(\widehat{x})$.*

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 $(\widehat{x}, \widehat{\lambda})$ for the active-set to be constant □

Obvious questions:

① construction of H_k ?

② global convergence ?

H_k can generally only be approximated

$\longrightarrow$ IDEA: Update formulas as in the unconstrained case

Broyden family

$H_{k+1} = H_k + U(H_k, p^k, q^k)$ with

$p^k = x^{k+1} - x^k$ and $q^k = \nabla_x \mathscr{L}(x^{k+1}, \lambda^k, \mu^k) - \nabla_x \mathscr{L}(x^k, \lambda^k, \mu^k)$

$\uparrow \quad \uparrow$

Index not increased, since only

2nd deriv. w.r.t. x needed!

Attention $Hess_X \mathscr{L}$ need not be pos.def. - only on the nullspace of C_k

$\Rightarrow$ Ideally suited Broyden symmetric rank 1 with appropriate treatment of indefinite QP (e.g. trust-region)

or BFGS Update with so-called Powell-Relaxation or damped BFGS update [Pow78]

If $p^{k\top} q^k \geq \alpha p^{k\top} H_k p^k \quad (\alpha \text{ e.g. } 0.2)$

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 $\nabla_x^2 \mathscr{L}$ pos.def. (unrealistic) superlinear convergence can be shown - otherwise only linear.
  (Secant condition violated in Powell relaxation!)

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 (compare definition 6.8) results here as Lagrangian function:

$$\Leftrightarrow \nabla \mathscr{L}_A = 0 \text{ with } \mathscr{L}_A = f(x) + \lambda^\top c(x) + \frac{\mu}{2} c(x)^\top c(x)$$

For μ sufficiently large is $Hess_x \mathscr{L}_A$ pos.def

Make updates for $Hess_x \mathscr{L}_A$.

Variant 2: Reduced SQP methods

Consider optimization problem with separability structure (cf. section 2.3.1)

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

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. $[C_z, C_p] R(z,p) = I$

SQP step is defined by

$$\begin{matrix} \text{(I)} \\ \text{(II)} \\ \text{(III)} \end{matrix} \begin{bmatrix} H_{zz} & H_{zp} & c_z^\top \\ H_{pz} & H_{pp} & c_p^\top \\ c_z & c_p & 0 \end{bmatrix} \begin{pmatrix} \triangle z \\ \triangle p \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_z f \\ -\nabla_p f \\ -c \end{pmatrix}$$

$$\begin{array}{lll} \text{from (I)} \Rightarrow & \lambda & = -c_z^{-\top}(\nabla_z f + H_{zz}\triangle z + H_{zp}\triangle p) \\ \text{from(II)} \Rightarrow & \triangle z & = -c_z^{-1}(c + c_p \triangle p) \end{array}$$

both used in (II) (cf. nullspace decomposition in section 5.1.2)

$$T^\top HT\triangle p = -T^\top(\nabla_{(z,p)} f - HRc) \qquad \text{(tedious calculation)}$$

This results in total

$$\begin{pmatrix} \triangle z \\ \triangle p \end{pmatrix}^{SQP} = -T\left[T^\top HT\right]^{-1} \underbrace{T^\top(\nabla_{z,p} f}_{\text{red. grad.}} - HRc) - Rc$$

$\uparrow$ reduced Hessian B

Term small near solution, $c = 0$, thus, can be omitted

$$\begin{pmatrix} \triangle z \\ \triangle p \end{pmatrix}^{RSQP} := -T\left[T^\top HT\right]^{-1} T^\top(\nabla_{z,p} f) - Rc$$

tangential step | orthogonal step

Remark:
Names are correct, if $R = Q_1 L^{-1}, T = Q_2$ from LQ decomposition $\quad (Q := C[Q_1\ Q_2] = [L\ 0])$

$\rightarrow$ Projected SQP methods, in this form - via a QR decomposition - impracticable.

Summarizing, the RSQP method corresponds to a Newton/Quasi-Newton method for

$$\min_p f\big(z(p), p\big) \quad , \quad \begin{array}{l} \text{where for } z(p) \text{ only one} \\ \text{Newton step is performed} \end{array}$$

and vice versa an SQP method with special Hessian matrix approximation:

$$\begin{bmatrix} 0 & 0 & C_z^\top \\ 0 & T^\top HT & C_p^\top \\ C_z & C_p & 0 \end{bmatrix} \begin{pmatrix} \Delta z \\ \Delta p \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla_z f \\ -\nabla_p f \\ -c \end{pmatrix}$$

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

$\rightarrow$ For this only use tangential information:

$$\begin{aligned} p^k &= \Delta p \\ q^k &= T^\top \left(\nabla_{(z,p)} \mathscr{L}(z^k + T\Delta p, \lambda^k) - \nabla_{(z,p)} \mathscr{L}(z^k, \lambda^k) \right) \end{aligned}$$

$\Big($ or analytical approximation for $T^\top HT = Hess_p f\big(z(p), p\big)\Big)$

Remark: method can also incorporate additional equality and inequality constraints in the so-called partially reduced SQP methods [Sch96, Sch98].

Variant 3: Generalized Gauß-Newton methods for constrained least squares problems

Problem class

$$\begin{array}{lll} \min\limits_x & \frac{1}{2}\|F(x)\|_2^2, & F : \mathbb{R}^n \to \mathbb{R}^k \\ \text{s.t.} & c(x) = 0, & c : \mathbb{R}^n \to \mathbb{R}^m \\ & g(x) \leq 0, & g : \mathbb{R}^n \to \mathbb{R}^\ell \end{array}$$

Analogous to the unconstrained case (see section 3.5) we use a '"Linearization under the norm'":

$x^{k+1} = x^k + \Delta x^k$, where Δx^k solves

$$\begin{array}{ll} \min\limits_{\Delta x^k} & \frac{1}{2}\|F'(x^k)\Delta x^k + F(x^k)\|_2^2 \\ \text{s.t.} & c_x(x^k)\Delta x^k + c(x^k) = 0, \\ & g_x(x^k)\Delta x^k + g(x^k) \leq 0, \end{array}$$

The first line of this subproblem is equivalent to

$$\min_{\Delta x^k} \tfrac{1}{2}(\Delta x^k)^\top F'(x^k)^\top F'(x^k)\Delta x^k + \left[F'(x^k)^\top F(x^k)\right]^\top \Delta x^k$$

Therefore, the generalized Gauß-Newton method also corresponds to an SQP method with the Hessian matrix approximation:

$H_k = F'(x^k)^\top F'(x^k)$

Convergence:

- locally linear with convergence rate $\kappa = \mathcal{O}(\|F(\hat{x})\|)$
- theory based on particular pseudoinverse (cf. [Boc87, Deu04]).

lecture 20

6.4 Globalization of Convergence

The methods from the SQP family (SQP, rSQP, generalized GN) provide as such only locally convergent algorithms. Globalization strategies are to widen the region of convergence. As in the unrestricted case, there are two different approaches.

a) line-search technique (compare with section 3.1)

We need a criterion to evaluate the current step length, a so-called *merit function*. For this we require the following characteristics:

- The aim is to achieve both an improvement in the objective function and an improvement in the constraints conditions and a trade-off is desired.
- 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 (cf. section 6.1) for this.

Example: Powell penalty function for $\min f(x)$ s.t. $c(x) = 0$

$$P_p(x,\mu) = f(x) + \sum_{i=1}^{m} \alpha_i |c_i(x)| \qquad \alpha_i > |\widehat{\lambda}_i| > 0$$

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 $H \approx Hess_x \mathscr{L}$, but H positiv definite '"approximation"'.

Lemma 6.15 (Consistency)**.** *Assume that the step Δx is the result of the QP above with H positive semidefinite, and that α is chosen such that $\alpha_i > |\lambda_i|$, where λ is the adjoint vector in the QP above. Then for P_p, there holds*

$$\left.\frac{d}{d\varepsilon}\right|_{\varepsilon=0+} P_p(x + \varepsilon\Delta x, \alpha) < 0$$

Proof:

$$\begin{aligned}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0+} P_p(x+\varepsilon\Delta x,\alpha) &= \nabla_x f^\top \Delta x + \sum_{c_i>0} \alpha_i \underbrace{C_i\Delta x}_{<0} - \sum_{c_i<0} \alpha_i \underbrace{C_i\Delta x}_{>0} + \sum_{c_i=0} \alpha_i |\underbrace{C_i\Delta x}_{=0}| \\ &= \nabla_x f^\top \Delta x - \sum_{c_i\neq 0} \alpha_i |C_i\Delta x|\end{aligned}$$

From the necessary optimality conditions follows $\nabla_x f^\top \Delta x = -\Delta x^\top H \Delta x - \sum_{i=1}^m \lambda_i C_i \Delta x$ and therefore

$$\begin{aligned}\frac{d}{d\varepsilon}\Big|_{\varepsilon=0+} P_p(x+\varepsilon\Delta x,\alpha) &= -\Delta x^\top H\Delta x - \sum_{c_i>0} (\alpha_i-\lambda_i)|C_i\Delta x| - \sum_{c_i<0} (\alpha_i+\lambda_i)|C_i\Delta x| \\ &< 0\end{aligned}$$

□

Remarks:

- This strategy is implemented in the solver option `SLSQP`[Kra88] to be used within `scipy.optimize.minimize`.
- We have shown now global convergence for all the methods of SQP (SQP, RSQP, Generalized Gauß-Newton) class above in connection with line-search
- The arguments of this lemma and the following theorem can also easily be transferred to the ℓ_1-penalty function P_1.
- There is a minor step in the convergence considerations still missing: one has to show that a backtracking line-search starting with $t^0 = 1$ stays at $t^k = 1$ in the vicinity of the solution. Then, the local convergence results (quadratic, superlinear, linear) discussed above materialize also in a concrete problem. The respective results can be found, e.g., in [NW06].

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 $\beta I < H^k < \gamma I, \forall k$, with the Loewner order "<"). Then there is at least one accumulation point of the damped SQP iteration and each such accumulation point is KKT point.*

Proof: N is compact $\Rightarrow \exists$ accumulation point $\hat{x}$. Let $x^k \to \hat{x}$.

$\Rightarrow \exists$ subsequence with $H^k \to \hat{H}$ (uniform boundedness of H^k)

$\Rightarrow \exists \Delta\hat{x}$ with $\Delta x^k \to \Delta\hat{x}$.

$\hat{x}$ is a KKT point, if and only if $\Delta\hat{x} = 0$. Thus, we assume in a proof by contradiction that $\Delta\hat{x} \neq 0$.

$\Rightarrow$ according to lemma 6.15 there is $\bar{t}$ with $P_p(\hat{x} + \bar{t}\Delta\hat{x}, \alpha) = P_p(\hat{x}, \alpha) - \delta$, for a $\delta > 0$.

Furthermore, there holds $\circledast P_p(\hat{x}, \alpha) < P_p(x^{k+1}, \alpha) = P_p(x^k + t_k \Delta x^k, \alpha)$, $\forall k$.

But, since $x^k + \bar{t}\Delta x^k \longrightarrow \hat{x} + \bar{t}\Delta\hat{x}$ there is a k_0 with

$P_p(x^k + t\Delta\bar{x^k}, \alpha) < P_p(\hat{x}, \alpha) - \delta/2$, $\forall j > k_0$ $\quad$ $\lightning$ to $\circledast$. $\square$

b) Trust region globalization for constrained problems

Aspects:

1. update of the trust region
2. Solution of the TR sub-problems

ad 1. analogously to the unconstrained case (section 3.4) only with merit function instead of objective functional as criterion.

ad 2. We briefly discuss here the so-called Byrd-Omojokun strategy [MNP98] for reduced SQP methods in the separability framework (compare SQP varian 2 above and section 2.3.1), in which Pythagoras is slightly abused in the usual way:

Step 1: Determine within the trust-region radius δ a a (almost) feasible substep, i.e. solve the subproblem

$$\begin{aligned} &\min \|C_z \Delta z_1 + C_p \Delta p_1 + c\|^2 \\ &\text{s.t.} \quad \left\| \begin{pmatrix} \Delta z_1 \\ \Delta p_1 \end{pmatrix} \right\| \leq (\sigma \cdot \Delta)^2 \end{aligned}$$

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

$$\begin{aligned} &\min \tfrac{1}{2}\Delta p_2^\top B_k \Delta p_2 + (T^\top \nabla f)^\top \Delta p_2 \\ &\text{s.t.} \quad \|T\Delta p_2\| \leq \delta^2 - \delta_1^2 \end{aligned}$$

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: a very good introduction can be found in the book [Van96]. Particular interior point aspects for data science can be found in [SNW11], which I already mentioned earlier.
- Optimization on manifolds is a central topic in data science, see [SN19] as a very recent example. A thorough introduction into the field can be found in the book [AMS08b].
- Variants of approximate SQP methods, in particular important for optimal control (see next semester in lecture Optimization with Differential Equations as an optional choice)
- Optimization with constraints in the form of complementarity problems like Stackelberg games and variational inequalities, which is very interesting and a field of current research, but maybe of less importance for data science.

lecture 21

Further literature:

[Abe12] Shigeo Abe. *Support Vector Machines for Pattern Classification*. Springer, 2012.

[AMS08b] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. https://pdfs.semanticscholar.org/fd49/6952a8bea3a3056f9cb07cbe72cd52bcfe2f.pdf.

[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.

[Boc87] H.G. Bock. Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. *Bonner Mathematische Schriften 183*, Bonn, 1987.

[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.

[CGT92] A.R. Conn, N.I.M Gould, and P.L. Toint. *Lancelot, A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Springer, 1992.

[Deu04] P. Deuflhard. *Newton Methods for Nonlinear Problems, Affine Invariance and Adaptive Algorithms*. Number 35 in Springer Series in Computational Mathematics. Springer Berlin, Heidelberg, 2004.

[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.

[GT03] N.I.M. Gould and P.L. Toint. GALAHAD, A Library of Thread-Safe Fortran 90 Packages for Large-Scale Nonlinear Optimization. *ACM Transactions on Mathematical Software*, 29(4):352–372, 2003. "https://dl.acm.org/doi/pdf/10.1145/962437.962438".

[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.

[MNP98] M.Lalee, J. Nocedal, and T. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3):682–706, 1998.

[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://epubs.siam.org/doi/pdf/10.1137/070704277.

[Pow78] M.J. D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.

[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.

[Sch96] V.H. Schulz. *Reduced SQP methods for large-scale optimal control problems in DAE with application to path planning problems for satellite mounted robots*. PhD thesis, University of Heidelberg, 1996.

[Sch98] V. H. Schulz. Solving discretized optimization problems by partially reduced SQP methods. *Computing and Visualization in Science*, 1(2):83–96, 1998.

[SN19] Guang-Jing Song and Michael Kwok-Po Ng. Nonnegative low rank matrix approximation for nonnegative matrices, 2019. https://arxiv.org/abs/1912.06836.

[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".

[Ste83] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. 20:626–637, 1983.

[Van96] R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer, 1996.

[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".

[Wou79] Arthur Wouk. *A course of applied functional analysis*. Wiley, 1979.

[Zei95] Eberhard Zeidler. *Applied functional analysis, main principles and their application*. Springer, 1995.

[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.