* * * Artificial Intelligence and Machine Learning * * *

Optimization Techniques

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

- **I** Machine Learning Introduction
- **II** Optimization Techniques
- **III** Bayesian Decision Theory
- **IV** Non-parametric Density Estimation
- **V** Probabilistic Graphical Models
- **VI** Linear Regression
- **VII** Logistic Regression
- **VIII** Deep Learning
- **IX** Evaluation
- **X** Decision Trees
- **XI** Support Vector Machines
- **XII** Clustering
- **XIII** Principal Component Analysis
- **XIV** Reinforcement Learning
- **XV** Advanced Regression

Agenda for this Unit

1. Introduction and Motivation
2. Important Concepts and Definitions
3. Unconstrained and constrained Optimization
4. Numerical Optimization
5. Wrap-Up

Section:

Introduction and Motivation

What Learning is all about

A motivational Example

Learning equals Optimization

- In every machine learning problem, we have:
  1. An **objective function** we want to optimize
  2. **Data** we want to learn from
  3. **Parameters** which need to be learned
  4. Assumptions about the problem and the data
- We would like to have general solutions to the problem of learning
- Different algorithms embody different objective functions and assumptions

Every machine learning problem is an optimization problem!

Example: Linear Regression

- Suppose we want to fit an **affine-linear** function to a dataset with N data points:

$$h_{\theta}(x) = \theta_1 x + \theta_0$$

- To do so, we have to find parameters $\theta_0 \in \mathbb{R}$ (y-intercept) and $\theta_1 \in \mathbb{R}$ (slope) such that the function is a good fit
- **Question: How to measure the goodness of fit?**
- **Answer:** Use a **cost function** (objective function) which encodes the desired behavior, i. e. prefer functions which are close to the data points

Example: Linear Regression (Ctd.)

- In linear regression we usually use the **squared error** cost function

$$\mathfrak{J}(\theta_0, \theta_1) := \sum_{n=1}^{N} \left(\theta_1 x_n + \theta_0 - y_n\right)^2 \tag{1}$$

- Choose θ_0, θ_1 so as to minimize the cost function

$$\theta_0^\star, \theta_1^\star = \arg\min_{\theta_0, \theta_1} \mathfrak{J}(\theta_0, \theta_1)$$

It is the task of the optimization algorithm to find the best solution!

Section:

Important Concepts and Definitions

Partial Derivatives and Gradients
HESSE Matrix
TAYLOR Expansion
Convex Functions and convex Sets

Partial Derivatives of Functions $\mathbb{R}^D \to \mathbb{R}$

Definition: Let $S \subset \mathbb{R}^D$ be an open set and $f : S \to \mathbb{R}$. We say that f is **partially differentiable** in $\boldsymbol{x_0} \in S$ with respect to the i-th input variable x_i, if the limit

$$\lim_{h \to 0} \frac{f\left(\boldsymbol{x_0} + h\boldsymbol{e}^i\right) - f\left(\boldsymbol{x_0}\right)}{h} \tag{2}$$

exists.

- $\boldsymbol{e}^i$ is the i-th canonical unit vector
- The limit (2) – if existent – is denoted by $\frac{\partial}{\partial x_i} f(\boldsymbol{x_0})$

Gradients ∇

- The **gradient** of the function $f : S \rightarrow \mathbb{R}$ is obtained by stacking all D partial derviatives of f into a column vector:

$$\nabla f(\boldsymbol{x_0}) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(\boldsymbol{x_0}) \\ \vdots \\ \frac{\partial}{\partial x_D} f(\boldsymbol{x_0}) \end{pmatrix} \tag{3}$$

- We will often use **transposition** to save space: $\left(\frac{\partial}{\partial x_1} f(\boldsymbol{x_0}), \ldots, \frac{\partial}{\partial x_D} f(\boldsymbol{x_0})\right)^\top$
- The symbol ∇ is pronounced *'nabla'* or *'gradient of'* *(it is **not** a capital delta Δ!)*

Computation of the Gradient

How can we compute the gradient?

- Luckily, we can apply the same rules of differentiation we know from one-dimensional calculus:
  - **Chain rule**,
  - **product rule**,
  - **Quotient rule**,
  - etc.
- Differentiate with respect to one variable and hold the others fixed

Example: Computation of the Gradient

- Let the function $f : \mathbb{R}^2 \to \mathbb{R}, (x_1, x_2) \mapsto \log(x_1 x_2) + \sin(x_2)$ be given
- Partial derivatives:

$$\frac{\partial}{\partial x_1} f(x_1, x_2) = \frac{1}{x_1} \qquad \text{and} \qquad \frac{\partial}{\partial x_2} f(x_1, x_2) = \frac{1}{x_2} + \cos(x_2)$$

- Thus, the gradient is:

$$\nabla f(x_1, x_2) = \begin{pmatrix} \frac{1}{x_1} \\ \frac{1}{x_2} + \cos(x_2) \end{pmatrix}$$

Direction of steepest Ascent

Lemma: The gradient ∇f points into the direction of steepest **ascent**.

Proof: Let $\boldsymbol{v} \in \mathbb{R}^D$ be a unit vector, i. e. $\|\boldsymbol{v}\| = 1$. The directional derivative of f in the direction of $\boldsymbol{v}$ is given by the scalar product $\boldsymbol{v}^\top \nabla f$. We aim to find the vector $\boldsymbol{v}$ which maximizes the directional derivative. This vector will then point into the direction of steepest ascent. We can rewrite the scalar product using its geometrical definition and receive $\boldsymbol{v}^\top \nabla f = \|\boldsymbol{v}\| \cdot \|\nabla f\| \cdot \cos(\alpha)$.

Since $\boldsymbol{v}$ has length 1, the equation simplifies to $\boldsymbol{v}^\top \nabla f = \|\nabla f\| \cdot \cos(\alpha)$. Only the angle α can be altered to maximize this quantity (since the gradient is given). The cosine reaches a maximum value for $\alpha = 0$. This means ∇f is parallel to $\boldsymbol{v}$, i. e. ∇f points into the direction of steepest ascent. ■

Direction of steepest Descent

The following observation will be especially useful when minimizing functions!

Lemma: The negative gradient $-\nabla f$ points into the direction of steepest **descent**.

Proof: Analogous to the proof above. ■

HESSE Matrix ∇^2

The HESSE matrix $\nabla^2 f$ (also referred to as **Hessian**) contains all second-order partial derivatives of f:

$$\nabla^2 f(\boldsymbol{x_0}) := \begin{pmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(\boldsymbol{x_0}) & \frac{\partial^2}{\partial x_1 \partial x_2} f(\boldsymbol{x_0}) & \dots & \frac{\partial^2}{\partial x_1 \partial x_D} f(\boldsymbol{x_0}) \\ \frac{\partial^2}{\partial x_2 \partial x_1} f(\boldsymbol{x_0}) & \frac{\partial^2}{\partial x_2 \partial x_2} f(\boldsymbol{x_0}) & \dots & \frac{\partial^2}{\partial x_2 \partial x_D} f(\boldsymbol{x_0}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_D \partial x_1} f(\boldsymbol{x_0}) & \frac{\partial^2}{\partial x_D \partial x_2} f(\boldsymbol{x_0}) & \dots & \frac{\partial^2}{\partial x_D \partial x_D} f(\boldsymbol{x_0}) \end{pmatrix} \in \mathbb{R}^{D \times D} \tag{4}$$

Schwarz's Theorem

Schwarz's theorem: If f has continuous second-order derivatives, then

$$\frac{\partial^2}{\partial x_i \partial x_j} f(\boldsymbol{x_0}) = \frac{\partial^2}{\partial x_j \partial x_i} f(\boldsymbol{x_0}) \qquad \forall i, j = 1, 2, \ldots, D$$

- In other words: If f has continuous second-order derivatives, then the order in which the derivatives are taken is not important
- **Corollary:** If f has continuous second-order derivatives, then (4) is **symmetric**

Example: Computation of the HESSE Matrix

- Consider again the function

$$f : \mathbb{R}^2 \to \mathbb{R}, (x_1, x_2) \mapsto \log(x_1 x_2) + \sin(x_2)$$

- The partial **first-order** derivatives are given by:

$$\frac{\partial}{\partial x_1} f(x_1, x_2) = \frac{1}{x_1} \quad \text{and} \quad \frac{\partial}{\partial x_2} f(x_1, x_2) = \frac{1}{x_2} + \cos(x_2)$$

- We compute the partial **second-order** derivatives on the next slide by taking the derivatives of the first-order derivatives by x_1 and x_2

Example: Computation of the HESSE Matrix (Ctd.)

$$\frac{\partial^2}{\partial x_1 \partial x_1} f(x_1, x_2) = -\frac{1}{x_1^2}$$

$$\frac{\partial^2}{\partial x_1 \partial x_2} f(x_1, x_2) = 0$$

$$\frac{\partial^2}{\partial x_2 \partial x_1} f(x_1, x_2) = 0$$

$$\frac{\partial^2}{\partial x_2 \partial x_2} f(x_1, x_2) = -\frac{1}{x_2^2} - \sin(x_2)$$

Example: Computation of the HESSE Matrix (Ctd.)

- Therefore, we have:

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} -\frac{1}{x_1^2} & 0 \\ 0 & -\frac{1}{x_2^2} - \sin(x_2) \end{pmatrix}$$

- Please note that SCHWARZ's theorem holds true in this case as f has continuous second-order derivatives, i. e. $\nabla^2 f(x_1, x_2)$ is a symmetric matrix

Taylor Expansion for Functions $\mathbb{R} \to \mathbb{R}$

Lemma: Let $f : \mathbb{R} \to \mathbb{R}$ be n-times differentiable at $x_0 \in \mathbb{R}$. We can approximate f using a **Taylor polynomial** of degree n given by

[Expansion in x_0]

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \tag{5}$$

[Expansion in x]

$$f(x + \delta_x) \approx f(x) + f'(x)\delta_x + \frac{f''(x)}{2!}\delta_x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\delta_x^n \tag{6}$$

Example: Taylor Polynomial for $\sin(x)$

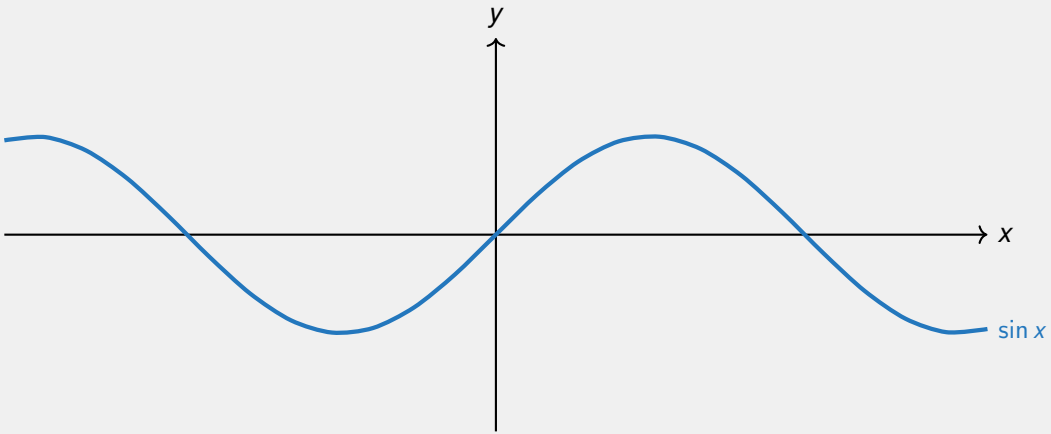
- Let us compute a degree 5 Taylor expansion of $\sin(x)$ at $x_0 = 0$:

$$\sin(x) \approx \sin(0) + x\cos(0) - \frac{x^2}{2}\sin(0) - \frac{x^3}{6}\cos(0) + \frac{x^4}{24}\sin(0) + \frac{x^5}{120}\cos(0)$$
$$= x - \frac{x^3}{6} + \frac{x^5}{120}$$

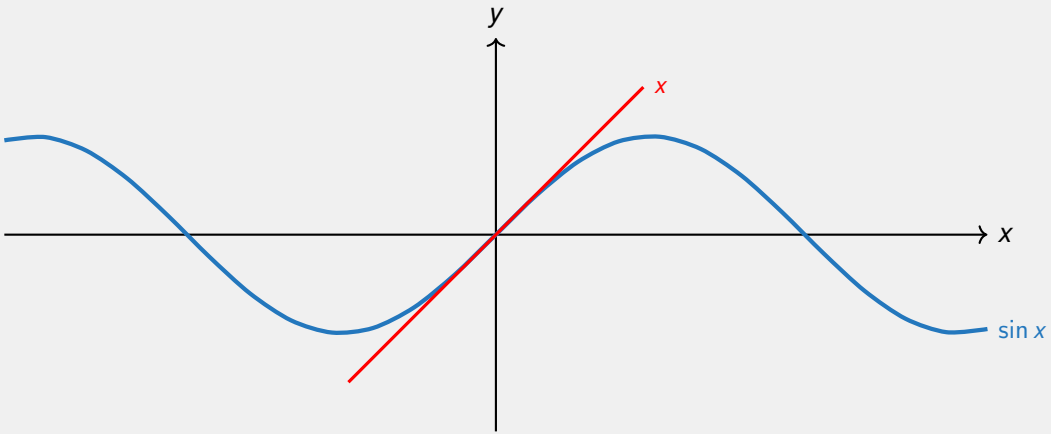
- We could continue the expansion forever because $\sin(x)$ is infinitely differentiable:

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

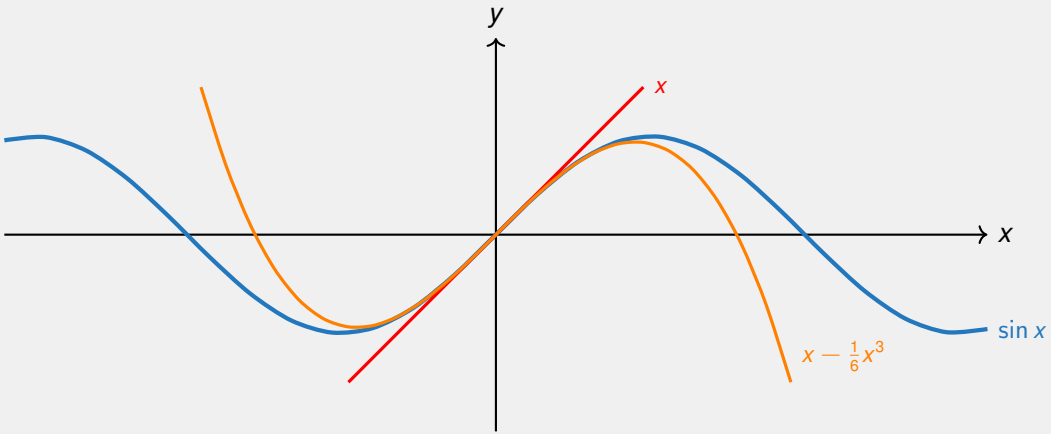
Example: Taylor Polynomial for $\sin(x)$ (Ctd.)



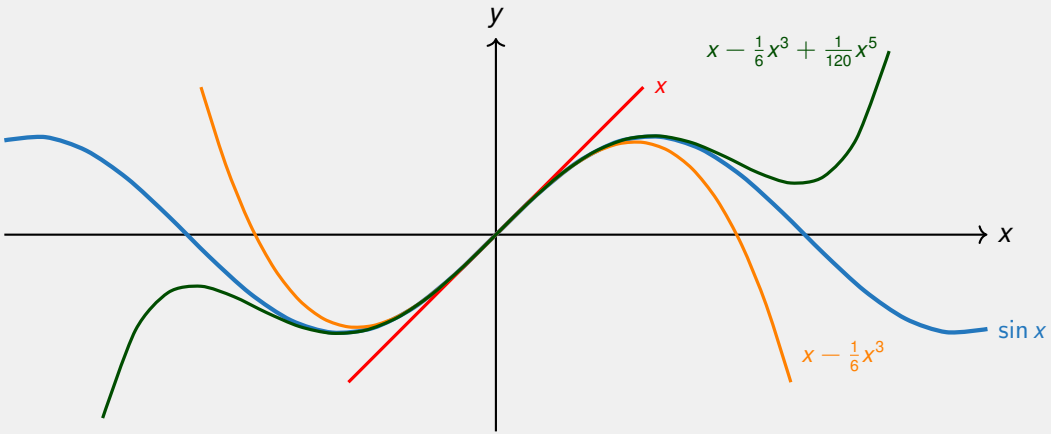
Example: Taylor Polynomial for $\sin(x)$ (Ctd.)



Example: Taylor Polynomial for $\sin(x)$ (Ctd.)



Example: Taylor Polynomial for $\sin(x)$ (Ctd.)



Taylor Expansion for Functions $\mathbb{R}^D \to \mathbb{R}$

We generalize the concept of Taylor expansions to multivariate functions:

Let $f : \mathbb{R}^D \to \mathbb{R}$ be 2-times differentiable in $\boldsymbol{x} \in \mathbb{R}^D$. We can approximate f using a **Taylor polynomial** of degree 2 given by

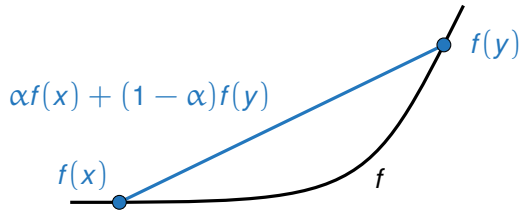
$$f(\boldsymbol{x} + \boldsymbol{\delta_x}) \approx f(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})^\top \boldsymbol{\delta_x} + \frac{1}{2}\boldsymbol{\delta_x}^\top \boldsymbol{H}(\boldsymbol{x})\boldsymbol{\delta_x} \tag{7}$$

- $\boldsymbol{g}$ is the gradient of f, and $\boldsymbol{H}$ the Hesse matrix of f
- A Taylor polynomial of degree 2 is sufficient in our case. Of course we could consider polynomials of degree $n > 2$ as well

Convex Functions

Definition: A function $f : \mathbb{R}^D \to \mathbb{R}$ is said to be **convex**, if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f)$ and $\forall \alpha \in [0, 1]$:

$$f(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) \leqslant \alpha f(\boldsymbol{x}) + (1 - \alpha) f(\boldsymbol{y}) \tag{8}$$



Strictly convex Functions

- We call a function **strictly convex**, if $\forall \boldsymbol{x}, \boldsymbol{y} \in \text{dom}(f)$ and $\forall \alpha \in (0, 1)$:

$$f(\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}) < \alpha f(\boldsymbol{x}) + (1 - \alpha) f(\boldsymbol{y})$$

- **Remark:** A function which is strictly convex is also convex *(but not vice versa!)*
- Check the definiteness of the Hesse matrix to test a function f on convexity:
  - If $\nabla^2 f(\boldsymbol{x}) \succeq 0$ (positive semi-definite), then f is convex
  - If $\nabla^2 f(\boldsymbol{x}) \succ 0$ (positive definite), then f is strictly convex

Examples: Convex Functions

Examples:

- **Convex, but not strictly convex:** (Affine-)linear functions ($\boldsymbol{a} \in \mathbb{R}^D$, $b \in \mathbb{R}$):

$$f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{x} + b$$

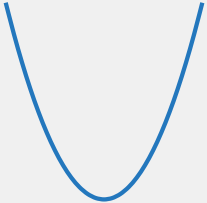
- **Strictly convex:** Quadratic functions ($\boldsymbol{A} \in \mathbb{R}^{D \times D}$, $\boldsymbol{b} \in \mathbb{R}^D$, $c \in \mathbb{R}$):

$$f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x} + c$$

- **Strictly convex:** Exponential function

Convex and Non-convex Functions

Convex function



Non-convex function

Advantages of Convexity

Question: Why are convex cost functions so appealing?

Advantages of convex functions:

- Local solutions are global optima
- Numeric optimizers do not get stuck in local optima
- Efficient implementations of optimizers are available

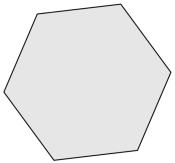
Convex Sets

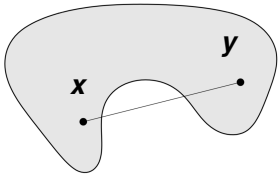
Definition: A set $C \subset \mathbb{R}^D$ is called **convex**, if $\forall \boldsymbol{x}, \boldsymbol{y} \in C$ and $\forall \alpha \in [0, 1]$:

$$\alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y} \in C \tag{9}$$

*(this is the equation of the line segment between **x** and **y**)*

convex





non-convex

Section:

Unconstrained and constrained Optimization

Unconstrained Optimization
Constrained Optimization and KKT conditions
Optimization Problems with Equality Constraints
Optimization Problems with Inequality Constraints
Duality (Primal and Dual Optimization Problem)

Optimization of Functions $\mathbb{R} \to \mathbb{R}$

- Let a function $f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x)$ be given
- **How do we find minimums and maximums of f?**
- **Necessary condition:** A point $x_0 \in \mathbb{R}$ is a potential candidate **(stationary point)**, if $f'(x_0) = 0$
- We then check the sign of the second-order derivative

$$f''(x_0) > 0 \qquad \Longrightarrow \text{minimum}$$

$$f''(x_0) < 0 \qquad \Longrightarrow \text{maximum}$$

Optimization of Functions $\mathbb{R}^D \to \mathbb{R}$

- Let a function $f : \mathbb{R}^D \to \mathbb{R}, \boldsymbol{x} \mapsto f(\boldsymbol{x})$ be given
- **How do we find minimums and maximums of f?**
- **Necessary condition:** A point $\boldsymbol{x_0} \in \mathbb{R}^D$ is a stationary point, if

$$\nabla f(\boldsymbol{x_0}) = \mathbf{0} \tag{10}$$

- We then check if the HESSE matrix $\nabla^2 f(\boldsymbol{x_0})$ is
  - **positive definite** *(minimum)* or
  - **negative definite** *(maximum)*

How to check the Definiteness of Matrices?

For simplicity we consider (2×2)-matrices:

Lemma: Let S be a non-empty subset of $\mathbb{R}^2$ and $f : S \to \mathbb{R}$ be a 2-times differentiable function with continuous derivatives.

We define (determinant of the HESSE matrix):

$$\Delta(\boldsymbol{x_0}) := \frac{\partial f(\boldsymbol{x_0})}{\partial x_1 \partial x_1} \cdot \frac{\partial f(\boldsymbol{x_0})}{\partial x_2 \partial x_2} - \left(\frac{\partial f(\boldsymbol{x_0})}{\partial x_1 \partial x_2} \right)^2 \tag{11}$$

Let $\boldsymbol{x_0} \in S$ be a candidate solution, i. e. $\nabla f(\boldsymbol{x_0}) = \mathbf{0}$.

How to check the Definiteness of Matrices? (Ctd.)

1. If $\Delta(\boldsymbol{x_0}) > 0$, then $\boldsymbol{x_0}$ is a $\begin{cases} \textbf{local maximum} & \text{if } \frac{\partial f(\boldsymbol{x_0})}{\partial x_1 \partial x_1} < 0. \\ \textbf{local minimum} & \text{if } \frac{\partial f(\boldsymbol{x_0})}{\partial x_1 \partial x_1} > 0. \end{cases}$
2. If $\Delta(\boldsymbol{x_0}) < 0$, then $\boldsymbol{x_0}$ is a **saddle point** (German: *Sattelpunkt*) of f.
3. If $\Delta(\boldsymbol{x_0}) = 0$, we cannot decide.

Example: Finding Minimums of Functions $\mathbb{R}^2 \to \mathbb{R}$

- Consider the function $f : \mathbb{R}^2 \to \mathbb{R}, (x_1, x_2) \mapsto x_1^2 + x_2^2$
- The gradient is given by $\nabla f(x_1, x_2) = \left(2x_1, 2x_2\right)^\top$ and therefore $\boldsymbol{x_0} = \left(0, 0\right)^\top$ the only stationary point
- The HESSE matrix is:

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

- We compute the determinant $\Delta(\boldsymbol{x_0}) = 2 \cdot 2 - 0 = 4 > 0$
- Since $\frac{\partial f(\boldsymbol{x_0})}{\partial x_1 \partial x_1} = 2 > 0$, we conclude that $\boldsymbol{x_0}$ is a local minimum

Constrained Optimization

- In many practical applications we have to consider **additional constraints** in the optimization process
- In the following we shall consider the following optimization problem (P):

$$\begin{aligned} \textit{minimize} \quad & f(\boldsymbol{x}) \\ \textit{subject to} \quad & h_k(\boldsymbol{x}) = 0 \qquad (k = 1, \ldots, K) \\ & g_\ell(\boldsymbol{x}) \leqslant 0 \qquad (\ell = 1, \ldots, L) \end{aligned}$$

Constrained Optimization (Ctd.)

- The differentiable function $f : \mathbb{R}^D \to \mathbb{R}$ is referred to as **objective function**
- The functions h_k $(k = 1, \ldots, K)$ represent K **equality constraints**
- The functions g_ℓ $(\ell = 1, \ldots, L)$ represent L **inequality constraints**
- We define the set of **feasible points**

$$\mathcal{Z}_P := \left\{ \boldsymbol{x} \in \mathbb{R}^D \mid h_k(\boldsymbol{x}) = 0 \ (k = 1, \ldots K), g_\ell(\boldsymbol{x}) \leqslant 0 \ (\ell = 1, \ldots, L) \right\}$$

- For any solution $\boldsymbol{x}^\star$ to (P) we have $\boldsymbol{x}^\star \in \mathcal{Z}_P$
- In the unconstrained case all points are feasible, i. e. we have $\mathcal{Z}_P := \mathbb{R}^D$

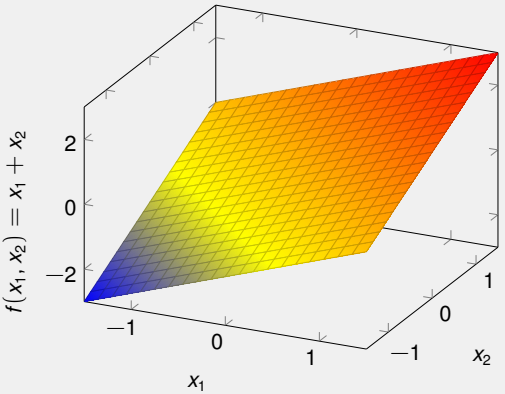
Example: Constrained Optimization

- Minimize

$$f(x_1, x_2) := x_1 + x_2$$

- Subject to the constraint

$$h(x_1, x_2) := x_1^2 + x_2^2 - 1 = 0$$



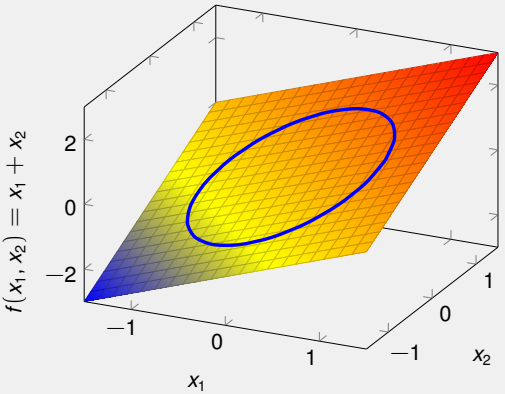
Example: Constrained Optimization

- Minimize

$$f(x_1, x_2) := x_1 + x_2$$

- Subject to the constraint

$$h(x_1, x_2) := x_1^2 + x_2^2 - 1 = 0$$



Necessary Conditions for Solutions

- For unconstrained optimization problems we search for points $\boldsymbol{x}$ which satisfy

$$\nabla f(\boldsymbol{x}) = \mathbf{0}$$

- In the following we shall introduce the so-called

KARUSH-KUHN-TUCKER (KKT) conditions

which represent first-order necessary conditions for a solution to problem (P), provided that some **regularity conditions/constraint qualifications** are satisfied

- **Please note:** In general, these conditions are not sufficient for a solution!

KKT Conditions

The following set of equalities and inequalities with the variables $(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^{D+K+L}$ are called KKT conditions for problem (P):

$$h_k(\boldsymbol{x}) = 0 \qquad (k = 1, \ldots, K) \tag{12}$$

$$g_\ell(\boldsymbol{x}) \leqslant 0 \qquad (\ell = 1, \ldots, L) \tag{13}$$

$$\nabla f(\boldsymbol{x}) + \sum_{k=1}^{K} \lambda_k \nabla h_k(\boldsymbol{x}) + \sum_{\ell=1}^{L} \mu_\ell \nabla g_\ell(\boldsymbol{x}) = \mathbf{0} \tag{14}$$

$$\mu_\ell g_\ell(\boldsymbol{x}) = 0 \qquad (\ell = 1, \ldots, L) \tag{15}$$

$$\mu_\ell \geqslant 0 \qquad (\ell = 1, \ldots, L) \tag{16}$$

KKT Conditions Remarks

- A point $(\boldsymbol{x}^\star, \boldsymbol{\lambda}^\star, \boldsymbol{\mu}^\star)$ satisfying the KKT conditions is called **KKT point**
- The conditions (12) and (13) ensure that $\boldsymbol{x}^\star \in \mathfrak{Z}_P$
- The equality (14) requires that the gradient of f be a linear combination of the gradients of the constraints, where $\boldsymbol{\lambda} \in \mathbb{R}^K$ and $\boldsymbol{\mu} \in \mathbb{R}^L$ are called **(LAGRANGE) multipliers**
- The conditions in (15) are called **complementary slackness conditions** and require for a KKT point $\boldsymbol{x}^\star$ that $\mu_\ell = 0$ or $g_\ell(\boldsymbol{x}^\star) = 0$ holds true
- **Remark:** In the unconstrained case, the KKT conditions reduce to $\nabla f(\boldsymbol{x}) = \mathbf{0}$

KKT Conditions Remarks (Ctd.)

- A local minimum $\boldsymbol{x}^\star$ of (P) is a KKT point if all constraints are (affine-)linear functions, i. e. the KKT conditions are necessary in this case
- In general, the KKT conditions are necessary conditions if the set of feasible points satisfies additional **constraint qualifications** *(not covered here)*

Lemma: Let f be a convex differentiable function with a continuous derivative and let h_k and g_ℓ be affine-linear constraints. Then $\boldsymbol{x}^\star$ is a solution to problem (P) if and only if $\boldsymbol{x}^\star$ is a KKT point for (P).

Optimization Problems

Question: What should an ideal optimization problem look like?

$$
\begin{aligned}
&\textit{minimize} && f(\boldsymbol{x}) && && \longleftarrow \text{ cost function / objective function} \\
&\textit{subject to} && h_k(\boldsymbol{x}) = 0 && (k = 1, \ldots, K) && \longleftarrow \text{ equality constraints} \\
& && g_\ell(\boldsymbol{x}) \leqslant 0 && (\ell = 1, \ldots, L) && \longleftarrow \text{ inequality constraints}
\end{aligned}
$$

Optimization Problems (Ctd.)

Answer:

$$
\begin{array}{lllll}
\textit{minimize} & f(\boldsymbol{x}) & & & \longleftarrow \textbf{convex} \text{ function} \\
\textit{subject to} & h_k(\boldsymbol{x}) = 0 & (k = 1, \ldots, K) & & \longleftarrow \textbf{affine-linear} \text{ functions} \\
& g_\ell(\boldsymbol{x}) \leqslant 0 & (\ell = 1, \ldots, L) & & \longleftarrow \textbf{convex} \text{ set}
\end{array}
$$

LAGRANGE Function

- We define the **LAGRANGE function** *(also called Lagrangian)*:

$$\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := f(\boldsymbol{x}) + \sum_{k=1}^{K} \lambda_k h_k(\boldsymbol{x}) + \sum_{\ell=1}^{L} \mu_\ell g_\ell(\boldsymbol{x}) \tag{17}$$

$$= f(\boldsymbol{x}) + \boldsymbol{\lambda}^\top \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{x}) \tag{18}$$

- We can write condition (14) in terms of the LAGRANGE function:

$$\nabla_{\boldsymbol{x}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0} \tag{19}$$

Optimization Problems with Equality Constraints

- In the following we consider problems **without inequality constraints**
- Thus, problem (P) reduces to the special case

$$\begin{aligned} &\textit{minimize} && f(\boldsymbol{x}) \\ &\textit{subject to} && h_k(\boldsymbol{x}) = 0 \qquad (k = 1, \ldots, K) \end{aligned}$$

- We call this problem (P_{Eq}) as the solution is constrained exclusively by equality constraints

KKT Conditions for Problems with Equality Constraints

- The KKT conditions for problem (P_{Eq}) reduce to a system of $D + K$ equations in $D + K$ variables:

$$h_k(\boldsymbol{x}) = 0 \qquad (k = 1, \ldots, K)$$

$$\nabla f(\boldsymbol{x}) + \sum_{k=1}^{K} \lambda_k \nabla h_k(\boldsymbol{x}) = \mathbf{0}$$

- These conditions can be rewritten in terms of the LAGRANGE function (17):

$$\nabla_{\boldsymbol{x}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathbf{0} \quad \text{and} \quad \nabla_{\boldsymbol{\lambda}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \mathbf{0} \tag{20}$$

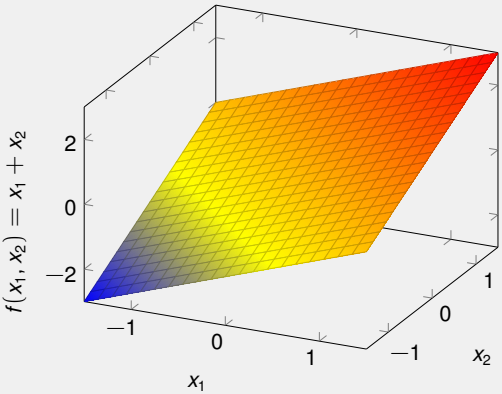
Example Revisited

- Let us revisit the example from above:
- Minimize

$$f(x_1, x_2) := x_1 + x_2$$

- Subject to the constraint

$$h(x_1, x_2) := x_1^2 + x_2^2 - 1 = 0$$



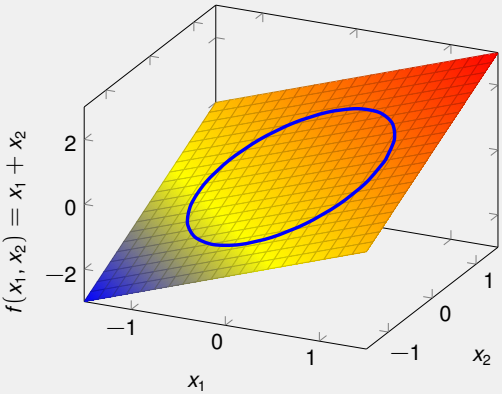
Example Revisited

- Let us revisit the example from above:
- Minimize

$$f(x_1, x_2) := x_1 + x_2$$

- Subject to the constraint

$$h(x_1, x_2) := x_1^2 + x_2^2 - 1 = 0$$



Example Revisited (Ctd.)

- **Step 1)** We set up the LAGRANGE function for the problem:

$$\mathfrak{L}(\boldsymbol{x}, \lambda) := x_1 + x_2 + \lambda(x_1^2 + x_2^2 - 1)$$

- **Step 2)** We compute the partial derivatives and set them to zero:

$$\text{(i)} \quad \partial\mathfrak{L}/\partial x_1 = 1 + 2\lambda x_1 \quad \stackrel{!}{=} 0$$

$$\text{(ii)} \quad \partial\mathfrak{L}/\partial x_2 = 1 + 2\lambda x_2 \quad \stackrel{!}{=} 0$$

$$\text{(iii)} \quad \partial\mathfrak{L}/\partial \lambda = x_1^2 + x_2^2 - 1 \quad \stackrel{!}{=} 0$$

Example Revisited (Ctd.)

- **Step 3)** We solve the system of equations from step 2):
  - Assuming $\lambda \neq 0$ we can solve (i) for x_1 and (ii) for x_2
  - We obtain $x_1 = -\frac{1}{2\lambda} = x_2$
  - Plugging this result into (iii), we get

$$2\lambda^2 = 1 \quad \Longleftrightarrow \quad \lambda = \pm\frac{1}{\sqrt{2}}$$

  - Thus, the candidate solutions are:

$$\underline{\boldsymbol{x}} = \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right) \qquad \text{and} \qquad \overline{\boldsymbol{x}} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$$

Bordered HESSE Matrix

- We can use the **bordered HESSE matrix** (German: *geränderte* HESSE *Matrix*) to decide if a candidate solution is a minimum or a maximum
- In the example above the bordered HESSE matrix has the following structure:

$$\widetilde{\nabla}^2 f(\boldsymbol{x}, \lambda) := \begin{pmatrix} 0 & \frac{\partial h}{\partial x_1} & \frac{\partial h}{\partial x_2} \\ \frac{\partial h}{\partial x_1} & \frac{\partial^2 \mathfrak{L}}{\partial x_1 \partial x_1} & \frac{\partial^2 \mathfrak{L}}{\partial x_1 \partial x_2} \\ \frac{\partial h}{\partial x_2} & \frac{\partial^2 \mathfrak{L}}{\partial x_2 \partial x_1} & \frac{\partial^2 \mathfrak{L}}{\partial x_2 \partial x_2} \end{pmatrix} \tag{21}$$

Bordered HESSE Matrix (Ctd.)

The sign of the determinant tells us if we have a local minimum or maximum:

Lemma:

- $\det\left(\widetilde{\nabla}^2 f(\boldsymbol{x}, \lambda)\right) > 0 \implies \boldsymbol{x}$ is a **local maximum**
- $\det\left(\widetilde{\nabla}^2 f(\boldsymbol{x}, \lambda)\right) < 0 \implies \boldsymbol{x}$ is a **local minimum**
- $\det\left(\widetilde{\nabla}^2 f(\boldsymbol{x}, \lambda)\right) = 0 \implies$ **We cannot decide**

Attention: We have to check the sign of the determinant of the bordered HESSE matrix, not its definiteness!

Example Revisited (Ctd.)

- **Step 4)** We decide for minimum or maximum:
  - We compute the bordered HESSE matrix according to (21):

$$\widetilde{\nabla}^2 f(\boldsymbol{x}, \lambda) = \begin{pmatrix} 0 & 2x_1 & 2x_2 \\ 2x_1 & 2\lambda & 0 \\ 2x_2 & 0 & 2\lambda \end{pmatrix}$$

  - Now we plug the two candidates $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ into the bordered HESSE matrix and compute the determinants

Example Revisited (Ctd.)

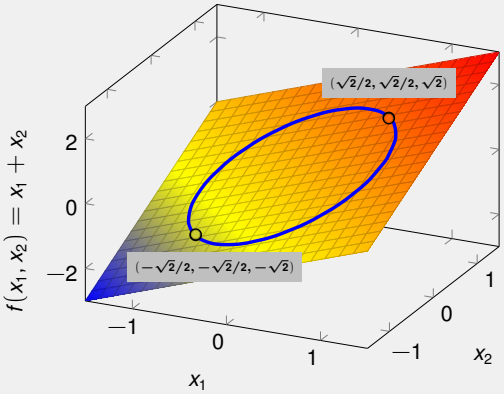
For $\underline{\boldsymbol{x}}$ we get a **minimum**:

$$\det\left(\widetilde{\nabla}^2 f(\underline{\boldsymbol{x}}, 1/\sqrt{2})\right) = \det\begin{pmatrix} 0 & -\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & 2/\sqrt{2} & 0 \\ -\sqrt{2} & 0 & 2/\sqrt{2} \end{pmatrix} = -4\sqrt{2} \quad < 0$$

For $\overline{\boldsymbol{x}}$ we get a **maximum**:

$$\det\left(\widetilde{\nabla}^2 f(\overline{\boldsymbol{x}}, -1/\sqrt{2})\right) = \det\begin{pmatrix} 0 & \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -2/\sqrt{2} & 0 \\ \sqrt{2} & 0 & -2/\sqrt{2} \end{pmatrix} = 4\sqrt{2} \quad > 0$$

Solution to the Example



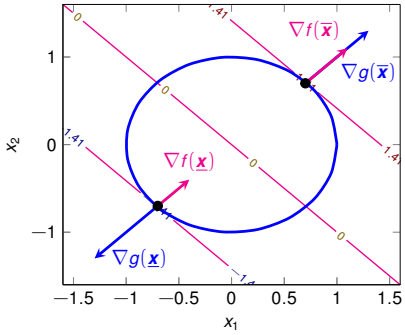
Geometry of LAGRANGE Optimization

- This is a **contour plot** of f
- Both points, $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$, fulfill

$$\nabla f(\boldsymbol{x}) = \lambda \nabla h(\boldsymbol{x}) \tag{22}$$

$$h(\boldsymbol{x}) = 0 \tag{23}$$

- ∇f and ∇h are **collinear**
- λ is a stretching/compression factor
- This leads to the same system of equations we have solved above!

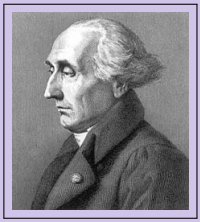


Portrait: JOSEPH-LOUIS LAGRANGE

JOSEPH-LOUIS LAGRANGE (1736 – 1813) was an Italian mathematician, physicist and astronomer. He made significant contributions to the fields of analysis, number theory, and both classical and celestial mechanics.

LAGRANGE was one of the creators of the calculus of variations, deriving the EULER–LAGRANGE equations for extrema of functionals. He extended the method **to include possible constraints**, arriving at the method of **LAGRANGE multipliers**. LAGRANGE invented the method of solving differential equations known as variation of constants and applied differential calculus to the theory of probabilities.

(Wikipedia)



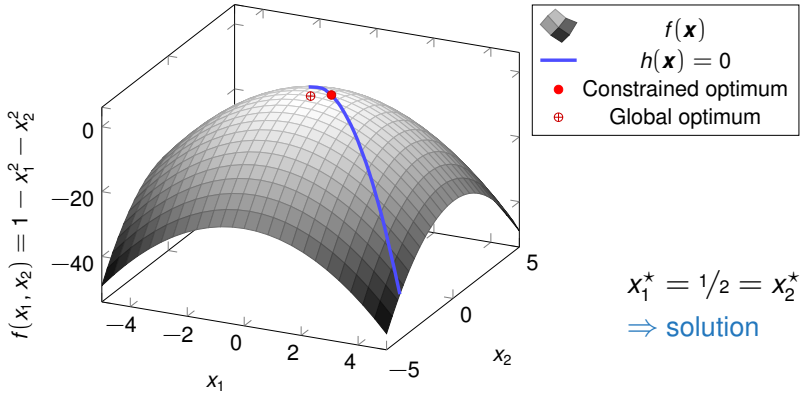
Exercise

Solve the following optimization problem:

$$\begin{aligned} &\textit{maximize} && f(x_1, x_2) := 1 - x_1^2 - x_2^2 \\ &\textit{subject to} && h(x_1, x_2) := x_1 + x_2 - 1 = 0 \end{aligned}$$

(This is a concave function with linear constraints, therefore there is only one global maximum which you don't have to show!)

Exercise Solution



$x_1^\star = 1/2 = x_2^\star$

$\Rightarrow$ solution

Dealing with additional Inequality Constraints

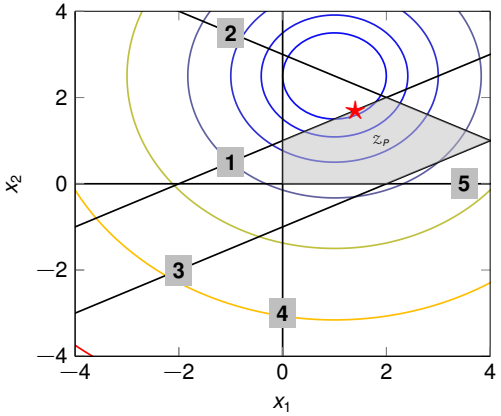
- For simple problems we can try to find a solution by directly solving the KKT system given by (12) - (16)
- We usually start with the complementary slackness conditions in (15) and consider all combinations for which they are satisfied
- We then plug these values into the other conditions and check if all of them hold true, if yes we have found a KKT point
- This approach can become tedious for large problems
- Optimization libraries implement efficient algorithms to solve such problems numerically

Dealing with additional Inequality Constraints (Ctd.)

Let the following optimization problem (P) be given:

$$
\begin{aligned}
&\textit{minimize} && q(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2.5)^2 && \\
&\textit{subject to} && -x_1 + 2x_2 - 2 \leqslant 0 && \textbf{1} \\
& && x_1 + 2x_2 - 6 \leqslant 0 && \textbf{2} \\
& && x_1 - 2x_2 - 2 \leqslant 0 && \textbf{3} \\
& && x_1 \geqslant 0 && \textbf{4} \\
& && x_2 \geqslant 0 && \textbf{5}
\end{aligned}
$$

Dealing with additional Inequality Constraints (Ctd.)



LAGRANGE Duality

- **Duality** in optimization is the idea of converting an optimization problem **(primal problem)** from one set of variables $\boldsymbol{x}$ **(primal variables)** into a different set of variables $\boldsymbol{\mu}$ **(dual variables)** which gives rise to the **dual problem**
- The dual problem might be easier to solve than the primal problem
- The primal and dual optimization problems have the same solution, but the value of the objective function may differ **(duality gap)**
- For convex optimization problems the duality gap is zero if **Slater's condition** holds
- There are different approaches to duality
- We shall now introduce the concept of **LAGRANGE duality**

Primal and dual Optimization Problems

- Consider the **primal optimization problem**

$$\begin{aligned} &\textit{minimize} && f(\boldsymbol{x}) \\ &\textit{subject to} && g_\ell(\boldsymbol{x}) \leqslant 0 \qquad (\ell = 1, \ldots, L). \end{aligned}$$

- The **dual optimization problem** is then given by

$$\begin{aligned} &\underline{\textit{maximize}}\ (!) && \mathfrak{D}(\boldsymbol{\mu}) := \inf_{\boldsymbol{x} \in \mathbb{R}^D} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu}) \\ &\textit{subject to} && \boldsymbol{\mu} \geqslant \mathbf{0}. \end{aligned}$$

The dual optimization Problem

- $\mathfrak{D}$ is a **concave function** *(which is why we have to maximize)*
- We can turn the dual optimization problem into a minimization problem by flipping the sign of $\mathfrak{D}$
- $-\mathfrak{D}$ is then a **convex function**
- We assume f and g_ℓ to be differentiable functions
- We can then compute $\inf_{\boldsymbol{x} \in \mathbb{R}^D} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu})$ for a given value of $\boldsymbol{\mu}$ by differentiating $\mathfrak{L}$ with respect to $\boldsymbol{x}$, setting the differential to zero, and solving for the optimal value
- The next slides will demonstrate this for a quadratic objective function

Example: Duality for quadratic Programs

- Consider the case of a quadratic objective function with affine constraints, e. g.

$$\begin{aligned} \textit{minimize} \quad & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \textit{subject to} \quad & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \leqslant \boldsymbol{0}, \end{aligned}$$

  where $\boldsymbol{A} \in \mathbb{R}^{L \times D}$, $\boldsymbol{b} \in \mathbb{R}^{L}$, and $\boldsymbol{c} \in \mathbb{R}^{D}$

- We assume that $\boldsymbol{Q} \in \mathbb{R}^{D \times D}$ is symmetric and positive definite, and therefore, the objective function is a convex function
- This problem is known as a **quadratic program**

Example: Duality for quadratic Programs (Ctd.)

- The Lagrangian $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu})$ for this problem is given by

$$\begin{aligned}\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu}) &= \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{\mu}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + (\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\mu})^\top \boldsymbol{x} - \boldsymbol{\mu}^\top \boldsymbol{b}.\end{aligned}$$

- We take the derivative of $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu})$ with respect to $\boldsymbol{x}$ and set it to zero:

$$\nabla_{\boldsymbol{x}} \mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{Q}\boldsymbol{x} + \left(\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\mu}\right) = \boldsymbol{0} \tag{24}$$

Example: Duality for quadratic Programs (Ctd.)

- $\boldsymbol{Q}$ is invertible (because we assumed it to be positive definite)
- We can therefore rearrange equation (24) for $\boldsymbol{x}$ and obtain

$$\boldsymbol{x} = -\boldsymbol{Q}^{-1}\left(\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\mu}\right).$$

- We plug this result into $\mathfrak{L}(\boldsymbol{x}, \boldsymbol{\mu})$ which yields the dual objective function

$$\mathfrak{D}(\boldsymbol{\mu}) = -\frac{1}{2}\left(\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\mu}\right)^\top \boldsymbol{Q}^{-1}\left(\boldsymbol{c} + \boldsymbol{A}^\top \boldsymbol{\mu}\right) - \boldsymbol{\mu}^\top \boldsymbol{b}.$$

Example: Duality for quadratic Programs (Ctd.)

- The dual optimization problem is therefore given by

$$\begin{aligned} &\textit{maximize} && -\frac{1}{2}\left(\boldsymbol{c}+\boldsymbol{A}^\top\boldsymbol{\mu}\right)^\top\boldsymbol{Q}^{-1}\left(\boldsymbol{c}+\boldsymbol{A}^\top\boldsymbol{\mu}\right)-\boldsymbol{\mu}^\top\boldsymbol{b} \\ &\textit{subject to} && \boldsymbol{\mu}\geqslant\mathbf{0}. \end{aligned}$$

- A quadratic optimization problem subject to constraints has to be solved when training **support vector machines**

Section:

Numerical Optimization

Introduction to numerical Optimization Techniques
First-Order Method: Gradient Descent
First-Order Method: Gradient Descent with Momentum
Second-Order Method: NEWTON's Method

Numerical Optimization

Problem:

Some optimization problems are too complex to solve them analytically!

Solution: Use numerical optimization techniques!

There is a plethora of techniques which differ in the **number of iterations** required, the **computational cost**, the **convergence guarantees**, the **robustness** with noisy cost functions, and their **memory usage**

Numerical Optimization (Ctd.)

- In the following we shall assume a cost function – consider e. g. (1)

$$\mathfrak{J} : \mathbb{R}^M \to \mathbb{R}, \boldsymbol{\theta} \mapsto \mathfrak{J}(\boldsymbol{\theta}),$$

  where $\boldsymbol{\theta} \in \mathbb{R}^M$ is the vector of model parameters

- **General idea:** Incrementally update an initial estimate of the parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^t + \boldsymbol{\delta}_{\boldsymbol{\theta}}^t \tag{25}$$

- Choose the **step** $\boldsymbol{\delta}_{\boldsymbol{\theta}}^t$ such that $\mathfrak{J}(\boldsymbol{\theta}^{t+1}) < \mathfrak{J}(\boldsymbol{\theta}^t)$
- Optimization algorithms differ in how they determine $\boldsymbol{\delta}_{\boldsymbol{\theta}}^t$

Numerical Optimization Algorithms (some Examples)

Gradient based methods:

- **Gradient descent**
- Conjugate gradient descent
- **NEWTON's method**
- (L-)BFGS
  (BROYDEN-FLETCHER-GOLDFARB-SHANNO)

Non-gradient based methods:

- Genetic algorithms
- Particle swarm
- Non-linear simplex
- NELDER-MEAD

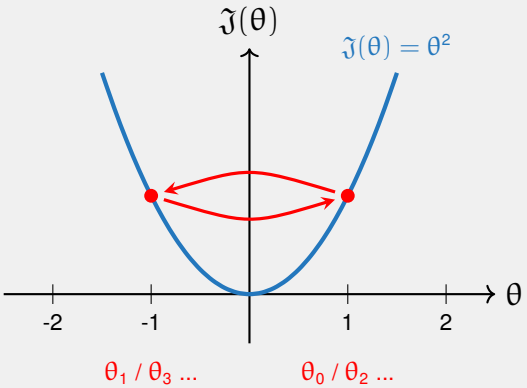
Numerical optimization techniques may not find the global optimum!

Gradient Descent

- **General idea:**
  1. Start with a random (or informed) guess of the parameters $\boldsymbol{\theta}^0$
  2. Then go into the direction of the **steepest descent**
  3. Iterate until the solution is *'good enough'*
- We have already seen that the negative gradient points into the direction of steepest descent, therefore:
- **First idea:**

$$\boldsymbol{\delta}_{\boldsymbol{\theta}}^t \longleftarrow -\nabla_{\boldsymbol{\theta}} \tilde{\jmath}(\boldsymbol{\theta}^t) \qquad \boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^t + \boldsymbol{\delta}_{\boldsymbol{\theta}}^t \tag{26}$$

Example: Gradient Descent



- Consider $\mathfrak{J}(\theta) = \theta^2$
- $\mathfrak{J}'(\theta) = 2\theta$
- Start with $\theta_0 = 1$

$$\theta_1 \longleftarrow 1 - 2 \qquad = -1$$
$$\theta_2 \longleftarrow -1 - (-2) = 1$$
$$\vdots$$

Gradient Descent: Update Rule

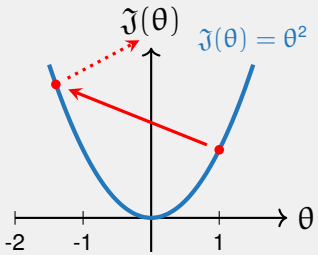
- It turns out that (26) is numerically instable
- Therefore, we introduce the **learning rate** $\alpha \in (0, 1)$ which stabilizes the learning process
- **Better idea:**

$$\boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \longleftarrow -\alpha \nabla_{\boldsymbol{\theta}} \mathfrak{J}(\boldsymbol{\theta}^{t}) \qquad \boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^{t} + \boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \tag{27}$$

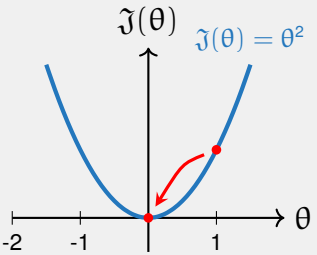
Gradient descent is a basic, but most commonly used optimization algorithm im machine learning!

Example revisited: Gradient Descent

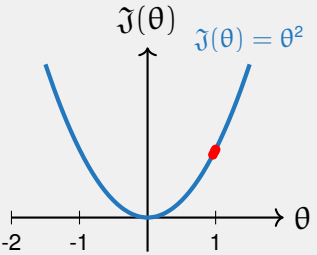
$\alpha = 1.2$ (too high)



$\alpha = 0.5$ (about right)



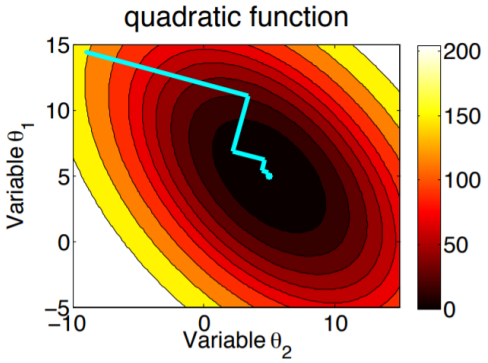
$\alpha = 0.001$ (too low)



Caution: The value of the learning rate depends on the optimization problem!

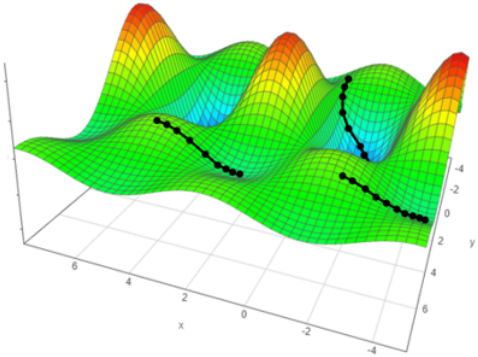
Problems of Gradient Descent

The parameter updates tend to 'zig-zag' down the valley (as the **curvature** of the function is not considered):



Problems of Gradient Descent (Ctd.)

Initialization also matters...



Gradient Descent with Momentum

- If the function is not convex, we run into the risk of not finding the optimal solution
- We adjust the algorithm to make it less likely to **get stuck in local solutions** and simultaneously **accelerate its convergence**
- Gradient descent with **momentum**:

$$\boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \longleftarrow \mu \boldsymbol{\delta}_{\boldsymbol{\theta}}^{t-1} + (1-\mu)\nabla_{\boldsymbol{\theta}}\mathfrak{J}(\boldsymbol{\theta}^{t}) \tag{28}$$

$$\boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^{t} - \alpha \boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \tag{29}$$

- $\mu \in [0, 1]$ is the momentum parameter ($\mu = 0$ recovers gradient descent)

NEWTON's Method

- Unlike gradient descent, NEWTON's method is a **second-order method**
- As a second-order method it also takes the **curvature** of the function into account
- Another name is NEWTON-RAPHSON algorithm

Lemma: The NEWTON step size $\boldsymbol{\delta}_{\boldsymbol{\theta}}$ to optimize a function $\mathfrak{J} : \mathbb{R}^M \rightarrow \mathbb{R}$ is given by the expression

$$\boldsymbol{\delta}_{\boldsymbol{\theta}} := -\boldsymbol{H}^{-1}(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta}), \tag{30}$$

where $\boldsymbol{g}(\boldsymbol{\theta})$ is the gradient, and $\boldsymbol{H}(\boldsymbol{\theta})$ the HESSE matrix of $\mathfrak{J}$ evaluated at $\boldsymbol{\theta}$

Proof of NEWTON's Method

Proof: We consider the second-order TAYLOR expansion of $\mathfrak{J}$ at the point $\boldsymbol{\theta}$ which is given by

$$\mathfrak{J}(\boldsymbol{\theta} + \boldsymbol{\delta}_{\boldsymbol{\theta}}) \approx \mathfrak{J}(\boldsymbol{\theta}) + \boldsymbol{g}(\boldsymbol{\theta})^{\top}\boldsymbol{\delta}_{\boldsymbol{\theta}} + \frac{1}{2}\boldsymbol{\delta}_{\boldsymbol{\theta}}^{\top}\boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{\delta}_{\boldsymbol{\theta}}, \tag{31}$$

where $\boldsymbol{g}(\boldsymbol{\theta})$ is the gradient, and $\boldsymbol{H}(\boldsymbol{\theta})$ the HESSE matrix of $\mathfrak{J}$ evaluated at $\boldsymbol{\theta}$. Subsequently, we differentiate (31) with respect to $\boldsymbol{\delta}_{\boldsymbol{\theta}}$ and set the derivative to zero:

$$\nabla_{\boldsymbol{\delta}_{\boldsymbol{\theta}}}\mathfrak{J}(\boldsymbol{\theta} + \boldsymbol{\delta}_{\boldsymbol{\theta}}) = \boldsymbol{g}(\boldsymbol{\theta}) + \boldsymbol{H}(\boldsymbol{\theta})\boldsymbol{\delta}_{\boldsymbol{\theta}} \overset{!}{=} \mathbf{0} \tag{32}$$

Rearranging for $\boldsymbol{\delta}_{\boldsymbol{\theta}}$ yields

$$\boldsymbol{\delta}_{\boldsymbol{\theta}} = -\boldsymbol{H}^{-1}(\boldsymbol{\theta})\boldsymbol{g}(\boldsymbol{\theta}) \tag{33}$$

which concludes the proof. ■

NEWTON's Method: Update Rule

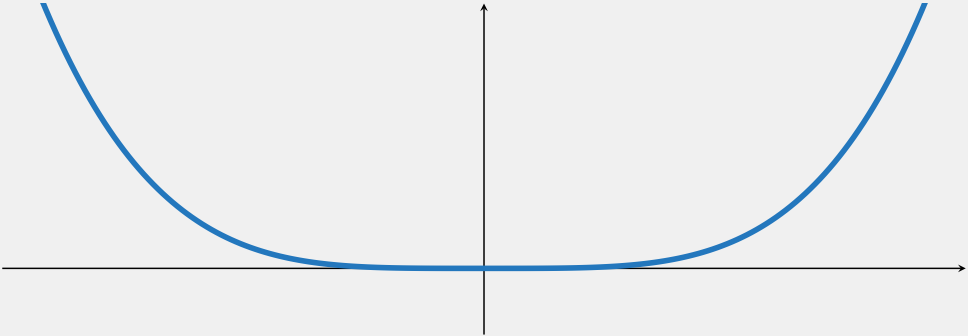
- The NEWTON update rule therefore takes the form

$$\boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \longleftarrow -\boldsymbol{H}^{-1}(\boldsymbol{\theta}^{t})\boldsymbol{g}(\boldsymbol{\theta}^{t}) \tag{34}$$

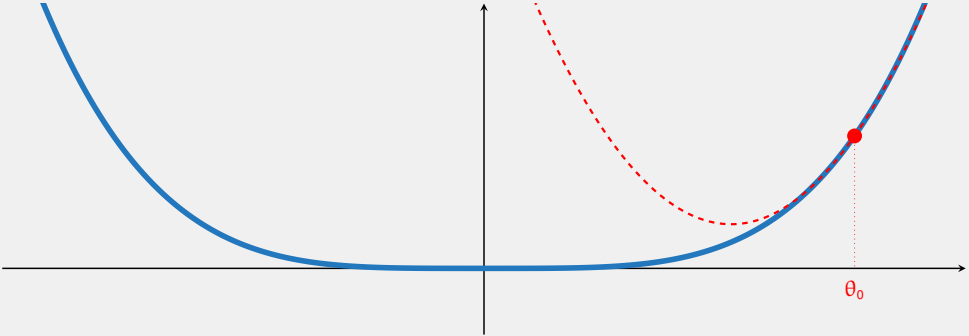
$$\boldsymbol{\theta}^{t+1} \longleftarrow \boldsymbol{\theta}^{t} + \boldsymbol{\delta}_{\boldsymbol{\theta}}^{t} \tag{35}$$

- The learning rate α has been replaced by the inverse of the Hessian matrix
- Newton's method **converges faster** (quadratic convergence) than gradient descent, but the inverse of the HESSE matrix is **expensive to compute** $\approx \mathcal{O}(n^3)$

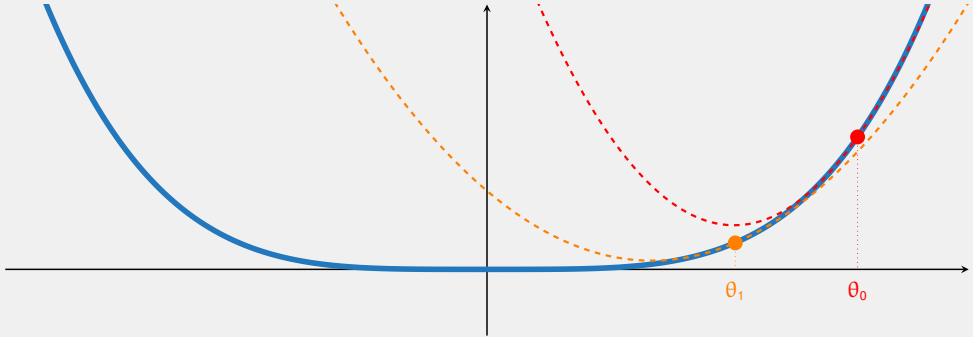
Visualization: NEWTON's Method



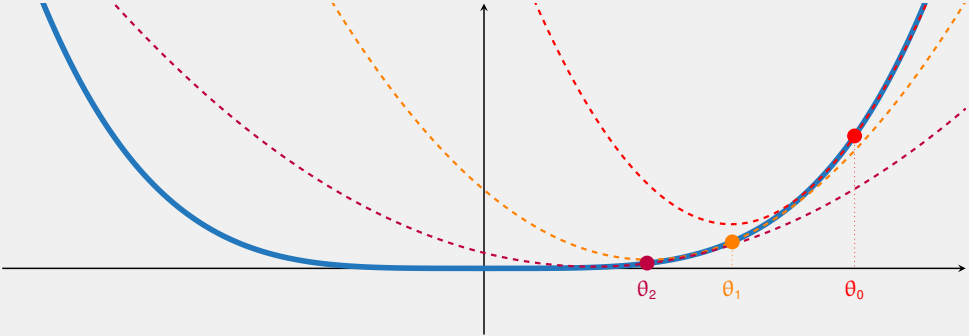
Visualization: NEWTON's Method



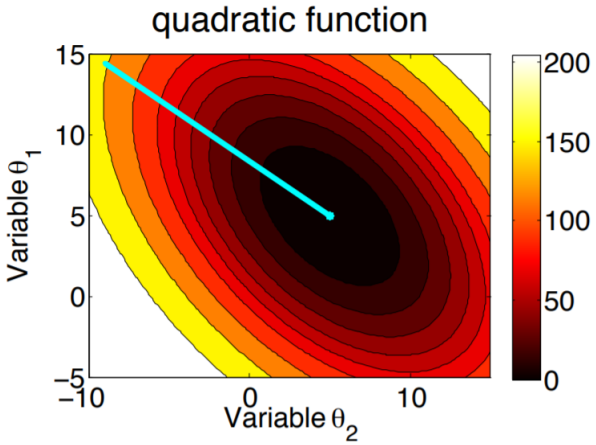
Visualization: NEWTON's Method



Visualization: NEWTON's Method

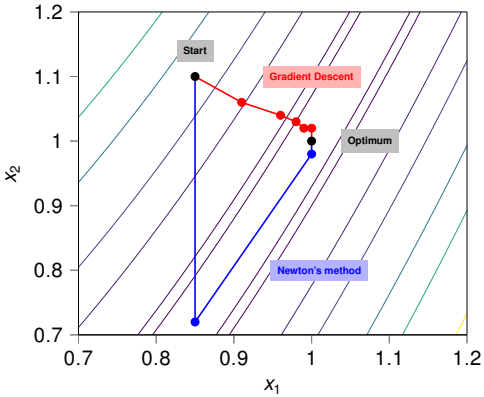


Visualization: NEWTON's Method (Ctd.)



NEWTON's Method in Comparison to Gradient Descent

- NEWTON's methods (blue) needs fewer interations than gradient descent (red)
- **However, matrix inversion is expensive!**



Section:

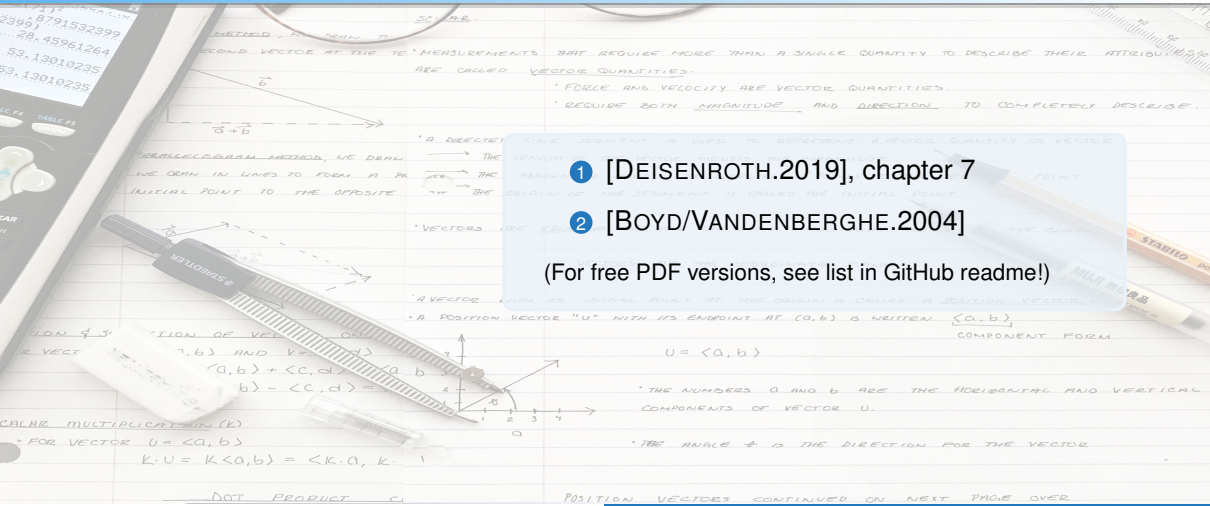
Wrap-Up

Summary
Recommended Literature
Self-Test Questions
Lecture Outlook

Summary

- **Every machine learning problem is an optimization problem!**
- We have learned about gradients, the HESSE matrix, and the TAYLOR expansion
- Good cost functions are **convex**
- You should be familiar with unconstrained and constrained optimization, especially with the concept of LAGRANGE optimization
- Closed-form solutions are often not feasible. In such cases we resort to **numerical optimization techniques**
- The most prominent numerical technique is called **gradient descent**
- NEWTON's method requires less iterations, but is computationally more demanding

Recommended Literature



1. [DEISENROTH.2019], chapter 7
2. [BOYD/VANDENBERGHE.2004]

(For free PDF versions, see list in GitHub readme!)

Self-Test Questions

1. What is a gradient and how is it used in optimization?
2. Which property of gradients is exploited in gradient descent?
3. What is convexity? Why should cost functions be convex?
4. How does LAGRANGE optimization work? What is the geometric interpretation?
5. Describe the gradient descent algorithm! Which issues does it have?
6. Explain the role of the learning rate in the gradient descent algorithm!
7. What is momentum?
8. How does NEWTON's method differ from gradient descent?

What's next...?

	I	Machine Learning Introduction
	II	Optimization Techniques
•	**III**	Bayesian Decision Theory
	IV	Non-parametric Density Estimation
	V	Probabilistic Graphical Models
	VI	Linear Regression
	VII	Logistic Regression
	VIII	Deep Learning
	IX	Evaluation
	X	Decision Trees
	XI	Support Vector Machines
	XII	Clustering
	XIII	Principal Component Analysis
	XIV	Reinforcement Learning
	XV	Advanced Regression

Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

Topic: Optimization Techniques

Term: Summer term 2025

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?