

# Mathematisch-algorithmische Grundlagen für Data Science

Numerische Algorithmen für Datenanalyse und Optimierung

Prof. Dr. Peter Becker

Fachbereich Informatik  
Hochschule Bonn-Rhein-Sieg

Sommersemester 2022



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

# Allgemeines zur Vorlesung

Homepage:

<http://www2.inf.h-brs.de/~pbecke2m/dsm/>

▶ [Link zur Homepage](#)

Die Vorlesung wird **überwiegend folienbasiert** gehalten.

Die Folien enthalten **nur die wichtigsten Aspekte** (Definitionen, Sätze, knappe Beispiele, wichtige Bemerkung).

Alles was sonst eine Vorlesung ausmacht (Erläuterungen, ausführliche Beispiele, Beweise, Anwendungen, Querverweise auf andere Gebiete der Informatik, etc.) gibt es nur in der Vorlesung selbst.

Die Folien zur Vorlesung (Skript) stehen auf der Homepage **vor der Vorlesung** zur Verfügung.

# Übungen und Vorleistung

- Wöchentlich erscheint ein Aufgabenblatt, das in der folgenden Woche in den Übungen besprochen wird.
- **Heute** Ausgabe des ersten Aufgabenblatts!
- Jedes Aufgabenblatt hat eine Sollpunktzahl.
- Zu erfüllende **Vorleistung**: 50% der gesamten Sollpunktzahl (aller Aufgabenblätter) müssen erreicht werden.
- Die Aufgabenblätter können in **Zweiergruppen** bearbeitet werden.
- teilweise ACAT-Aufgaben

# Lernziele

- Grundlegende numerische Algorithmen zur Lösung von **linearen Gleichungssystemen** kennen und anwenden können,
- Verfahren der **linearen Optimierung** und deren Anwendung auf **Ausgleichsprobleme** beherrschen,
- das Konzept von **Eigenwerten** verstehen und deren Bedeutung für die **Analyse von Netzwerken** kennen,
- weitere Verfahren für die Datenanalyse kennen und anwenden können (z. B. Verfahren der **Dimensionsreduktion** oder der **Interpolation**) und
- solche Algorithmen **implementieren** können.

# Inhalt (geplant)

- 1 Wiederholung: Grundlagen aus linearen Algebra
- 2 Direkte Methoden für lineare Gleichungssysteme
- 3 Ausgleichsrechnung
- 4 Optimierungsverfahren für Data Science
- 5 Eigenwerte und Netzwerkanalyse
- 6 Dimensionsreduktion
- 7 Maximum-Likelihood und der EM-Algorithmus (optional)

# Organisatorisches und Formales

Inhaltliche Voraussetzungen:

- Mathematik, insbesondere **Lineare Algebra**, **Wahrscheinlichkeitstheorie**, **Analysis**, **Graphentheorie**
- **Programmieren können**

Umfang: 2V + 2Ü

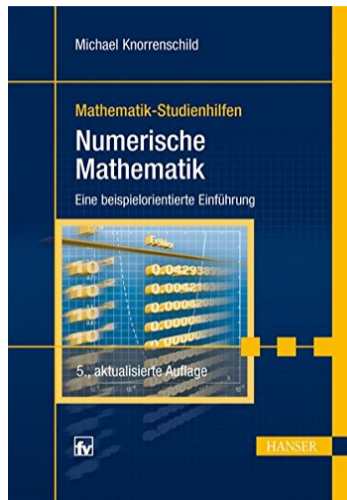
**Übungen/Vorleistung:** Abgabe von Lösungen zu Übungsaufgaben,  $\geq 50\%$  der gesamten Sollpunktzahl

**Prüfungsform:** gerne mündlich, wahrscheinlich schriftlich

# Literatur

Es gibt **nicht das Lehrbuch** zu dieser Veranstaltung!

- Bücher zu Numerik und Wissenschaftlichem Rechnen haben einen stärkeren Fokus auf die mathematischen Grundlagen der Verfahren und sind ohne Anwendungsbezug geschrieben.
- Bücher zu Data Mining Algorithmen vernachlässigen die Themen Mathematik und Optimierung.



Michael Knorrenschild  
*Numerische Mathematik: Eine  
beispielorientierte Einführung*  
Carl Hanser Verlag  
2013

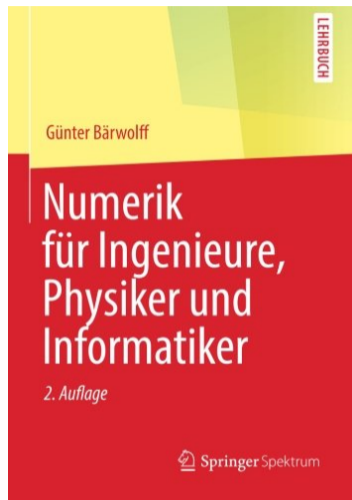
- knapp und anschaulich
- für die Grundlagen der Kapitel 2 und 3





Martin Hanke-Bourgeois  
*Grundlagen der Numerischen Mathematik und  
des Wissenschaftlichen Rechnens*  
Vieweg Teubner Verlag  
2008

- theoretischer, umfassender, tiefgehender als Knorrenschild
- für die Grundlagen der Kapitel 2, 3, 5 und 6



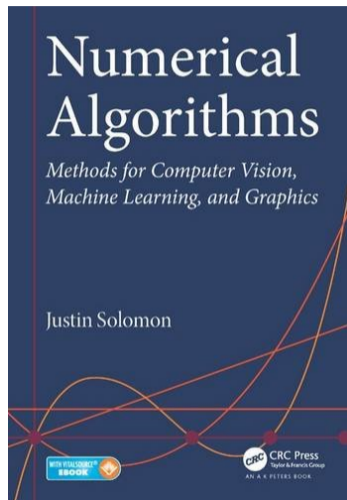
Günter Bärwolff

*Numerik*

Springer Spektrum

2015

- etwas ausführlicher als Knorrenschild
- für die Grundlagen der Kapitel 2, 3, 5 und 6



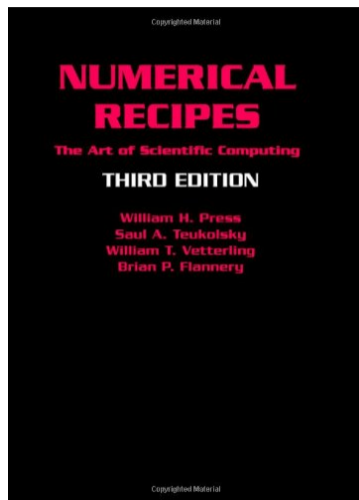
Justin Solomon

*Numerical Algorithms: Methods for Computer Vision, Machine Learning, and Graphics*

CRC Press

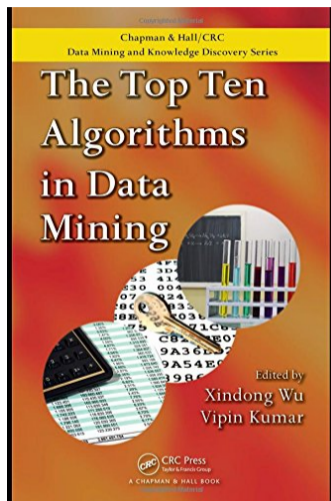
2015

- mit stärkerem Anwendungsbezug
- insbesondere Bildverarbeitung



William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery  
*Numerical Recipes: The Art of Scientific Computing*  
Cambridge University Press  
2007

- Klassiker für Anwender
- mit starken Quelltextanteilen
- vollständige Quelltexte auch auf CD erhältlich
- geht weit über klassische Numerik hinaus

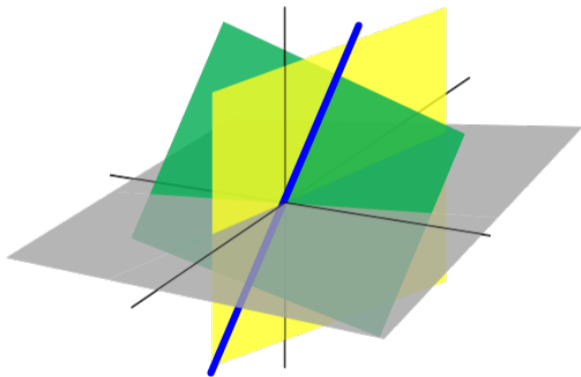


Xindong Wu, Vipin Kumar (Hrsg.)  
*The Top Ten Algorithms in Data Mining*  
CRC Press  
2009

- Sammelband mit wichtigen Data Mining Algorithmen
- knapp und aber trotzdem gut verständlich

# Kapitel 1

Wiederholung: Grundlagen aus der linearen Algebra



# Inhalt

## 1 Wiederholung: Grundlagen aus der linearen Algebra

- Notationen
- Dimension, Basis, Rang
- Lineare Gleichungssysteme
- Determinante

## Notationen für $K$ -Vektorraum

Zur Unterscheidung zwischen den Vektoren  $\in V$  und den Skalaren  $\in K$  schreiben wir die **Vektoren mit fettgedruckten lateinischen Kleinbuchstaben**, z. B.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = (x_j) \in \mathbb{R}^n.$$

Für die Skalare nutzen wir üblicherweise **griechische Kleinbuchstaben in Normalschrift**, z. B.

$$\lambda \in \mathbb{R}.$$

Zur Abkürzung schreiben wir die Vektoren teilweise auch zeilenorientiert, also

$$\mathbf{x} = (x_1, x_2, \dots, x_n).$$



Den **Nullvektor** bezeichnen wir mit  $\mathbf{0}$ . Demgegenüber bezeichnet  $0$  das neutrale Element der Addition des Körpers.

In den meisten nachfolgenden Fällen **verzichten wir auf die Verwendung des Multiplikationssymbols  $\cdot$** , sowohl bei der Multiplikation im Körper als auch bei der Multiplikation mit Skalaren. D. h.

$$\lambda\mu \quad := \quad \lambda \cdot \mu$$

$$\lambda\mathbf{v} \quad := \quad \lambda \cdot \mathbf{v}$$

für  $\lambda, \mu \in K, \mathbf{v} \in V$ .

☞ Wir bewegen uns im Folgenden fast ausschließlich im  **$\mathbb{R}$ -Vektorraum  $\mathbb{R}^n$** .

## Notationen für Matrizen

Die Menge der reellen Matrizen mit  $m$  Zeilen und  $n$  Spalten bezeichnen wir mit  $\mathbb{R}^{m \times n}$ .

Zur Darstellung solcher Matrizen nutzen wir i. d. R. **fette lateinische Großbuchstaben**, z. B.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = (a_{ij}) \in \mathbb{R}^{m \times n}.$$

Die **Nullmatrix** stellen wir ebenfalls durch **0** dar. Aus dem Kontext ergibt sich, ob damit der Nullvektor oder die Nullmatrix gemeint ist.

Für eine Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  bezeichnet  $\mathbf{A}^T \in \mathbb{R}^{n \times m}$  die **transponierte Matrix** von  $\mathbf{A}$ .

Einen Vektor  $\mathbf{x} \in \mathbb{R}^n$  können wir auch als **einspaltige Matrix**  $\mathbf{x} \in \mathbb{R}^{n \times 1}$  auffassen.

Das **Skalarprodukt**  $\langle \mathbf{x}, \mathbf{y} \rangle$  zweier Vektoren  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  können wir dann auch als **Matrixprodukt**  $\mathbf{x}^T \mathbf{y}$  schreiben.

Für eine Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  bezeichnet

- $\mathbf{a}_i \in \mathbb{R}^n$  (**fetter Kleinbuchstabe mit tiefgestelltem Index**) den  $i$ -ten Zeilenvektor und
- $\mathbf{a}^j \in \mathbb{R}^m$  (**fetter Kleinbuchstabe mit hochgestelltem Index**) den  $j$ -ten Spaltenvektor von  $\mathbf{A}$ .

Also:

$$\mathbf{A} = (\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n) = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_m \end{pmatrix}$$

Die Einheitsvektoren  $\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^n$  mit  $\mathbf{e}^i = (x_j) \in \mathbb{R}^n$  und

$$x_j = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{sonst} \end{cases}$$

bilden die **kanonische Basis** des  $\mathbb{R}^n$ .

Die Matrix

$$\mathbf{E} := (\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^n) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{n \times n}$$

bezeichnet die **Einheitsmatrix**.

# Lineare Unabhängigkeit

## Definition 1.1

Ein Vektor  $\mathbf{w} \in \mathbb{R}^n$  mit

$$\mathbf{w} = \sum_{i=1}^k \lambda_i \mathbf{v}_i$$

und  $\lambda_i \in \mathbb{R}$  für  $1 \leq i \leq k$  heißt **Linearkombination** der Vektoren  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k \in \mathbb{R}^n$ .

## Definition 1.2

Vektoren  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k \in \mathbb{R}^n$  heißen **linear unabhängig**, wenn aus

$$\lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \dots + \lambda_k \mathbf{v}_k = \sum_{i=1}^k \lambda_i \mathbf{v}_i = \mathbf{0}$$

stets  $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$  folgt. Andernfalls heißen sie **linear abhängig**.

Vektoren sind genau dann linear unabhängig, wenn sich  $\mathbf{0}$  nur auf triviale Weise als Linearkombination darstellen lässt.

### Beispiel 1.3

- Folgende Vektoren sind linear unabhängig:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}.$$

- Dagegen sind diese Vektoren linear abhängig:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} 0 \\ -1 \\ -1 \end{pmatrix}$$

Tafel .

## Lemma 1.4

Die Vektoren  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k \in \mathbb{R}^n$  seien linear abhängig. Dann existiert ein  $j, 1 \leq j \leq k$  mit

$$\mathbf{v}_j = \sum_{\substack{i=1 \\ i \neq j}}^k \mu_i \mathbf{v}_i.$$

- ☞ Wenn Vektoren linear abhängig sind, kann also **mindestens einer dieser Vektoren als Linearkombination der anderen Vektoren** dargestellt werden.

Beweis.

Tafel 

# Basis und Dimension eines Vektorraums

## Definition 1.5

- Der **Spann** (oder die **lineare Hülle**) der Vektoren  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k \in \mathbb{R}^n$  ist die Menge alle Linearkombinationen dieser Vektoren:

$$\text{Span}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) := \left\{ \sum_{i=1}^k \lambda_i \mathbf{v}_i \mid \lambda_1, \lambda_2, \dots, \lambda_k \in \mathbb{R} \right\}.$$

- Für einen Vektorraum  $V \subseteq \mathbb{R}^n$  heißt eine Menge  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k \in V$  von Vektoren **Erzeugendensystem von  $V$** , wenn

$$\text{Span}(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k) = V$$

gilt.



## Fortsetzung Definition.

- Ein linear unabhängiges Erzeugendensystem von  $V$  heißt **Basis von  $V$** .
- Die Anzahl der Vektoren in einer Basis von  $V$  heißt **Dimension von  $V$**  und wird mit  $\dim(V)$  bezeichnet.
  
- Alle Basen eines Vektorraums  $V$  haben die gleiche Anzahl an Vektoren.  $\dim(V)$  ist somit sinnvoll definiert.
- Vektorräume können auch eine unendliche Dimension haben. Beispiel:  $C^0([a, b])$ .
- Der  $\mathbb{R}^n$  hat natürlich die Dimension  $n$ , aber (echte) Unterräume des  $\mathbb{R}^n$  haben eine kleinere Dimension.
- Der Unterraum  $(\{\mathbf{0}\}, +, \cdot)$  ist der sogenannte **Nullvektorraum** oder auch **Nullraum**. Er hat die Dimension 0.

## Rang einer Matrix

### Definition 1.6

Der **Rang**  $r(\mathbf{A})$  einer Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  bezeichnet die maximale Anzahl linear unabhängiger Spaltenvektoren von  $\mathbf{A}$ .

### Bemerkung:

$$r(\mathbf{A}) = \dim(\text{Span}(\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n))$$

### Satz 1.7

Es sei  $\mathbf{A} \in \mathbb{R}^{m \times n}$  eine Matrix. Dann gilt:

- (a)  $r(\mathbf{A}) = 0 \Leftrightarrow \mathbf{A} = \mathbf{0}$
- (b)  $r(\mathbf{A}) = r(\mathbf{A}^T)$
- (c)  $r(\mathbf{A}) \leq \min\{m, n\}$

## Beispiel zum Matrixrang

### Beispiel 1.8

Für

$$\mathbf{A} = \begin{pmatrix} 3 & 2 & 1 \\ 4 & -5 & 1 \end{pmatrix}$$

ergibt sich  $r(\mathbf{A}) = 2$ .

**Begründung:** Die Spaltenvektoren  $\mathbf{a}^1$  und  $\mathbf{a}^2$  sind linear unabhängig, somit folgt  $r(\mathbf{A}) \geq 2$ .

Aus Satz 1.7 (c) folgt  $r(\mathbf{A}) \leq \min\{2, 3\} = 2$ .

Also muss  $r(\mathbf{A}) = 2$  gelten.

## Lineare Gleichungssysteme

Ein lineares Gleichungssystem (LGS) mit  $m$  Gleichungen und  $n$  Unbekannten hat die allgemeine Gestalt

$$\begin{array}{cccccc} a_{11} x_1 & + & a_{12} x_2 & + & \cdots & + & a_{1n} x_n & = & b_1 \\ a_{21} x_1 & + & a_{22} x_2 & + & \cdots & + & a_{2n} x_n & = & b_2 \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ a_{m1} x_1 & + & a_{m2} x_2 & + & \cdots & + & a_{mn} x_n & = & b_m \end{array}$$

Hierbei sind  $a_{ij}, b_i \in \mathbb{R}$  die **Koeffizienten** und  $x_j$  die **Unbekannten** des LGS.

Wir können das LGS auch schreiben als

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad \text{für } i = 1, \dots, m$$

oder in Matrixschreibweise mit

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

als

$$\mathbf{Ax} = \mathbf{b}.$$

# Beispiel LGS

## Beispiel 1.9

Das LGS

$$\begin{aligned} 3x_1 + 2x_2 + x_3 &= 5 \\ 4x_1 - 5x_2 + x_3 &= 2 \end{aligned}$$

und seine Darstellung in Matrixschreibweise

$$\begin{pmatrix} 3 & 2 & 1 \\ 4 & -5 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}.$$

# Lösbarkeit linearer Gleichungssysteme

## Definition 1.10

Für ein LGS  $\mathbf{Ax} = \mathbf{b}$  bezeichne

$$\mathbf{A|b} := (\mathbf{a}^1, \mathbf{a}^2, \dots, \mathbf{a}^n, \mathbf{b}) \in \mathbb{R}^{m \times n+1}$$

die um den Vektor  $\mathbf{b}$  der rechten Seite **erweiterte Matrix**  $\mathbf{A}$ .

## Satz 1.11

- *Es sei  $\mathbf{A} \in \mathbb{R}^{m \times n}$  und  $\mathbf{b} \in \mathbb{R}^m$ . Dann ist das LGS*

$$\mathbf{Ax} = \mathbf{b}$$

*genau dann lösbar, wenn  $r(\mathbf{A}) = r(\mathbf{A|b})$  gilt.*

- *Gilt zusätzlich  $r(\mathbf{A}) = n$ , dann gibt es genau eine Lösung  $\mathbf{x} \in \mathbb{R}^n$ .*
- *Falls  $r(\mathbf{A}) < n$  gilt, gibt es unendlich viele Lösungen mit  $n - r(\mathbf{A})$  freien Parametern.*

## Bemerkungen

Für  $m < n$  kann die Bedingung  $r(\mathbf{A}) = n$  nicht erfüllt werden und es gibt daher keine eindeutige Lösung (falls es überhaupt eine Lösung gibt).

Wenn keine Lösung existiert, dann heißt das LGS **überbestimmt**, existieren unendlich viele Lösungen, dann ist es **unterbestimmt**.

Für  $\mathbf{b} = \mathbf{0}$  spricht man von einem **homogenen LGS**.

Hat ein homogenes LGS nur die Lösung  $\mathbf{x} = \mathbf{0}$ , dann sind die Spaltenvektoren von  $\mathbf{A}$  linear unabhängig.



## Beispiele für Lösbarkeit von LGS

### Beispiel 1.12

Für das LGS aus Beispiel 1.9 gilt  $r(\mathbf{A}) = r(\mathbf{A}|\mathbf{b}) = 2$  (siehe auch Beispiel 1.8). Also ist das LGS lösbar mit unendlich vielen Lösungen und einem freien Parameter.

### Beispiel 1.13

(a)

$$\mathbf{A} = \begin{pmatrix} 3 & 0 & 1 \\ 6 & 2 & 1 \\ -3 & -2 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 3 \\ -1 \end{pmatrix}, \quad r(\mathbf{A}) = 3, \quad r(\mathbf{A}|\mathbf{b}) = 3$$

lösbar, da  $r(\mathbf{A}) = r(\mathbf{A}|\mathbf{b})$  und mit eindeutiger Lösung, da  $r(\mathbf{A}) = 3 = n$ .

## Fortsetzung Beispiel.

(b)

$$\mathbf{A} = \begin{pmatrix} -2 & 1 & 1 \\ -2 & 2 & 3 \\ -6 & 2 & 1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 4 \\ -4 \end{pmatrix}, \quad r(\mathbf{A}) = 2, \quad r(\mathbf{A}|\mathbf{b}) = 2$$

lösbar, da  $r(\mathbf{A}) = r(\mathbf{A}|\mathbf{b})$  und mit unendlich vielen Lösung, da  $r(\mathbf{A}) = 2 < n$ .

(c)

$$\mathbf{A} = \begin{pmatrix} -1 & 4 & 2 \\ -2 & 8 & 4 \\ 3 & -12 & -6 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 3 \\ 6 \\ -9 \end{pmatrix}, \quad r(\mathbf{A}) = 1, \quad r(\mathbf{A}|\mathbf{b}) = 1$$

lösbar, da  $r(\mathbf{A}) = r(\mathbf{A}|\mathbf{b})$  und mit unendlich vielen Lösung, da  $r(\mathbf{A}) = 1 < n$ .

## Fortsetzung Beispiel.

(d)

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 2 & -2 & 1 \\ -2 & -10 & 3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}, \quad r(\mathbf{A}) = 2, \quad r(\mathbf{A}|\mathbf{b}) = 3$$

unlösbar, da  $r(\mathbf{A}) \neq r(\mathbf{A}|\mathbf{b})$ .

**Bemerkungen:**

- Bei **direkten Methoden für Gleichungssysteme** betrachtet man eindeutig lösbare Gleichungssysteme. ➡ **Kapitel 2**
- In der **Ausgleichsrechnung** hat man überbestimmte Gleichungssysteme und versucht diese, mit einem möglichst geringen Gesamtfehler zu "lösen". ➡ **Kapitel 3**
- In der **Optimierung mit Nebenbedingungen** treten überwiegend unterbestimmte Systeme auf. ➡ **Kapitel 4**

# Permutationen

## Definition 1.14

Es sei  $X = \{x_1, x_2, \dots, x_n\}$  eine  $n$ -elementige Menge. Dann heißt eine bijektive Abbildung

$$\sigma : X \rightarrow X$$

**Permutation.**

## Satz 1.15

*Für eine  $n$ -elementige Menge  $X$  gibt es  $n!$  verschiedene Permutationen.*

- Für die mathematische Betrachtung von Permutationen beschränkt man sich üblicherweise auf  $X = \{1, 2, \dots, n\}$ .
- Für uns ist eine Permutation also stets eine bijektive Abbildung

$$\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}.$$

## Schreibweise von Permutationen

Eine Permutation  $\sigma : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$  stellt man üblicherweise in Form einer zweizeiligen Matrix

$$\sigma = \begin{pmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{pmatrix}$$

oder verkürzt in Tupelform

$$\sigma = (\sigma(1) \ \sigma(2) \ \cdots \ \sigma(n))$$

dar.

### Beispiel 1.16

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix} = (2 \ 4 \ 1 \ 3)$$

$$\sigma^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix} = (3 \ 1 \ 4 \ 2)$$

# Symmetrische Gruppe

## Definition 1.17

$S_n$  bezeichne die Menge aller Permutationen auf der Menge  $\{1, 2, \dots, n\}$ .

### Bemerkung:

- $(S_n, \circ)$  bildet mit der Komposition  $\circ$  von Abbildungen als Verknüpfung eine Gruppe.
- $S_n$  wird auch als **symmetrische Gruppe** bezeichnet.
- Eine **Permutationsgruppe** ist eine Untergruppe von  $S_n$ .
- Nach dem sogenannten Satz von Cayley ist **jede endliche Gruppe isomorph zu einer Permutationsgruppe**.

# Fehlstand und Signum einer Permutation

## Definition 1.18

- Für eine Permutation  $\sigma \in S_n$  heißt ein Paar  $(i, j)$  mit  $i < j$  und  $\sigma(i) > \sigma(j)$  **Fehlstand**.
- Die Zahl

$$F(\sigma) = |\{(i, j) \mid 1 \leq i < j \leq n \wedge \sigma(i) > \sigma(j)\}|$$

bezeichnet die **Anzahl der Fehlstände** von  $\sigma$ .

- Das **Signum**  $\text{sign}(\sigma)$  einer Permutation  $\sigma \in S_n$  ist definiert als

$$\text{sign}(\sigma) = (-1)^{F(\sigma)}.$$

- Eine Permutation mit  $\text{sign}(\sigma) = 1$  heißt **gerade**, eine Permutation mit  $\text{sign}(\sigma) = -1$  **ungerade**.

## Beispiel 1.19

Die Permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix}$$

aus Beispiel 1.16 hat die Fehlstände

$$\{(1, 3), (2, 3), (2, 4)\}.$$

Also gilt

$$F(\sigma) = 3$$

und damit

$$\text{sign}(\sigma) = -1.$$

Die Permutation ist somit ungerade.



# Eigenschaften des Signum

## Satz 1.20

Es sei  $\sigma \in S_n$  eine Permutation. Dann gilt

$$\text{sign}(\sigma) = \prod_{1 \leq i < j \leq n} \frac{\sigma(j) - \sigma(i)}{j - i}.$$

Für zwei Permutationen  $\sigma, \tau \in S_n$  gilt

$$\text{sign}(\tau \circ \sigma) = \text{sign}(\tau) \cdot \text{sign}(\sigma).$$

# Determinante

## Definition 1.21

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine quadratische Matrix. Dann heißt

$$\det(\mathbf{A}) := \sum_{\sigma \in S_n} \left( \text{sign}(\sigma) \prod_{i=1}^n a_{i\sigma(i)} \right)$$

**Determinante** der Matrix  $\mathbf{A}$ .

## Bemerkungen:

- Die Formel in Definition 1.21 heißt **Leibniz-Formel** und geht zurück auf **Gottfried Wilhelm Leibniz**.
- Summiert wird über die Elemente der symmetrischen Gruppe  $S_n$ . Die Summe besteht demnach aus  $n!$  **Summanden**.

## Beispiel zur Berechnung der Determinante

### Beispiel 1.22

Für  $n = 3$  enthält die symmetrische Gruppe  $S_n$  die Permutationen

$$(1 \ 2 \ 3), \quad (2 \ 3 \ 1), \quad (3 \ 1 \ 2)$$

mit  $\text{sign}(\sigma) = +1$  und die Permutationen

$$(3 \ 2 \ 1), \quad (1 \ 3 \ 2), \quad (2 \ 1 \ 3)$$

mit  $\text{sign}(\sigma) = -1$ . Damit folgt

$$\begin{aligned} \det \begin{pmatrix} 3 & 0 & 1 \\ 6 & 2 & 1 \\ -3 & -2 & 1 \end{pmatrix} &= \begin{array}{l} 3 \cdot 2 \cdot 1 + 0 \cdot 1 \cdot (-3) + 1 \cdot 6 \cdot (-2) \\ - 1 \cdot 2 \cdot (-3) - 3 \cdot 1 \cdot (-2) - 0 \cdot 6 \cdot 1 \end{array} \\ &= 6 + 0 - 12 + 6 + 6 - 0 \\ &= 6. \end{aligned}$$

## Eigenschaften der Determinante (1)

### Satz 1.23 (Laplacescher Entwicklungssatz)

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine Matrix.  $\mathbf{A}_{ij} \in \mathbb{R}^{(n-1) \times (n-1)}$  bezeichne die Matrix, die aus  $\mathbf{A}$  durch Streichen der  $i$ -ten Zeile und  $j$ -ten Spalte entsteht.

Dann gilt:

$$\begin{aligned}\det(\mathbf{A}) &= \sum_{i=1}^n (-1)^{i+j} a_{ij} \det(\mathbf{A}_{ij}) \\ &= \sum_{j=1}^n (-1)^{i+j} a_{ij} \det(\mathbf{A}_{ij})\end{aligned}$$

Die erste Gleichung beschreibt die *Entwicklung nach der  $j$ -ten Spalte*, die zweite Gleichung die *Entwicklung nach der  $i$ -ten Zeile*.

## Eigenschaften der Determinante (2)

### Satz 1.24

Es seien  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  quadratische Matrizen. Dann gilt:

- (a)  $\det(\mathbf{A} \cdot \mathbf{B}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$
- (b) Für  $r \in \mathbb{R}$  gilt  $\det(r\mathbf{A}) = r^n \det(\mathbf{A})$
- (c)  $\det(\mathbf{A}^T) = \det(\mathbf{A})$

# Determinante und Lösbarkeit eines LGS

## Satz 1.25

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine Matrix. Dann sind die folgenden Aussagen äquivalent:

- (a)  $\det(\mathbf{A}) \neq 0$
- (b) Die Spaltenvektoren (bzw. Zeilenvektoren) von  $\mathbf{A}$  sind linear unabhängig.
- (c)  $r(\mathbf{A}) = n$
- (d) Für alle  $\mathbf{b} \in \mathbb{R}^n$  ist das LGS  $\mathbf{A}\mathbf{x} = \mathbf{b}$  eindeutig lösbar.
- (e) Es existiert eine Matrix  $\mathbf{A}^{-1} \in \mathbb{R}^n$  mit  $\mathbf{A}^{-1}\mathbf{A} = \mathbf{E}$ .

# Cramersche Regel

## Satz 1.26

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine quadratische Matrix mit  $\det(\mathbf{A}) \neq 0$ . Für das LGS  $\mathbf{Ax} = \mathbf{b}$  sei

$$\mathbf{A}_j := (\mathbf{a}^1, \dots, \mathbf{a}^{j-1}, \mathbf{b}, \mathbf{a}^{j+1}, \dots, \mathbf{a}^n),$$

also die Matrix, die entsteht, wenn in  $\mathbf{A}$  die  $j$ -te Spalte durch den Vektor  $\mathbf{b}$  ersetzt wird.

Dann gilt für die eindeutige Lösung  $\mathbf{x} = (x_j) \in \mathbb{R}^n$  des Gleichungssystems  $\mathbf{Ax} = \mathbf{b}$ :

$$x_j = \frac{\det(\mathbf{A}_j)}{\det(\mathbf{A})}.$$

## Beispiel 1.27

Wir betrachten das LGS aus Beispiel 1.13 (a). Die Determinante  $\det(\mathbf{A}) = 6$  haben wir bereits in Beispiel 1.22 berechnet.

$$\det(\mathbf{A}_1) = \det \begin{pmatrix} 1 & 0 & 1 \\ 3 & 2 & 1 \\ -1 & -2 & 1 \end{pmatrix} = 2 + 0 - 6 + 2 + 2 - 0 = 0$$

$$\det(\mathbf{A}_2) = \det \begin{pmatrix} 3 & 1 & 1 \\ 6 & 3 & 1 \\ -3 & -1 & 1 \end{pmatrix} = 9 - 3 - 6 + 9 + 3 - 6 = 6$$

$$\det(\mathbf{A}_3) = \det \begin{pmatrix} 3 & 0 & 1 \\ 6 & 2 & 3 \\ -3 & -2 & -1 \end{pmatrix} = -6 + 0 - 12 + 6 + 18 - 0 = 6$$

Daraus folgt:

$$x_1 = \frac{0}{6} = 0, \quad x_2 = \frac{6}{6} = 1, \quad x_3 = \frac{6}{6} = 1.$$



# Zusammenfassung

- Lineare Unabhängigkeit
- Matrixrang
- Zusammenhang: Matrixrang und Lösbarkeit von linearen Gleichungssystemen
- Determinante in Form der Leibniz-Formel
- Cramersche-Regel

# Kapitel 2

Direkte Verfahren für lineare Gleichungssysteme

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

# Inhalt

- 2 Direkte Verfahren für lineare Gleichungssysteme
  - Lineare Gleichungssysteme einfacher Struktur
  - LR-Zerlegung ohne Pivotisierung
  - LR-Zerlegung mit Pivotisierung
  - LR-Zerlegung für Tridiagonalmatrizen
  - Cholesky-Zerlegung für positiv definite Matrizen

# Lösungsverfahren für lineare Gleichungssysteme

- **Direkte Verfahren**  
liefern in endlich vielen Schritten eine (exakte) Lösung.
- **Iterative Verfahren**  
erzeugen eine Folge von Vektoren, die gegen die Lösung des linearen Gleichungssystems konvergiert (approximative Lösung).

# Diagonalmatrix

## Definition 2.1

Eine Matrix  $\mathbf{D} \in \mathbb{R}^{n \times n}$  mit  $d_{ij} = 0$  für  $i \neq j$  heißt **Diagonalmatrix**.

Eine Diagonalmatrix hat also die Gestalt

$$\mathbf{D} = \begin{pmatrix} * & & & 0 \\ & * & & \\ & & \ddots & \\ 0 & & & * \end{pmatrix}.$$

Für eine Diagonalmatrix mit den Diagonalelementen  $d_{11}, d_{22}, \dots, d_{nn}$  schreiben wir auch  $\text{diag}(d_{11}, d_{22}, \dots, d_{nn})$ .

## Lineares Gleichungssystem mit Diagonalmatrix

Sei  $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n) \in \mathbb{R}^{n \times n}$ .

Dann ist das lineare Gleichungssystem

$$\mathbf{D}\mathbf{x} = \mathbf{b}$$

für beliebiges  $\mathbf{b} \in \mathbb{R}^n$  genau dann lösbar, wenn  $d_i \neq 0$  für alle  $i = 1, \dots, n$  gilt.

Die Lösung lautet dann

$$x_i = \frac{b_i}{d_i}.$$

# Algorithmus für Diagonalmatrix

## Algorithmus 2.2

### Eingabe:

- Diagonalmatrix  $\mathbf{D} \in \mathbb{R}^{n \times n}$  mit  $d_{ii} \neq 0$  für  $i = 1, \dots, n$
- rechte Seite  $\mathbf{b} \in \mathbb{R}^n$

**Ausgabe:** Lösungsvektor  $\mathbf{x} \in \mathbb{R}^n$

**for**  $i := 1$  **to**  $n$  **do**

$x_i := b_i / d_{ii}$

**end**

**return**  $\mathbf{x}$

# Untere Dreiecksmatrix

## Definition 2.3

Eine Matrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  mit  $l_{ij} = 0$  für  $i < j$  heißt **untere Dreiecksmatrix**.

$$\mathbf{L} = \begin{pmatrix} * & & & 0 \\ * & * & & \\ \vdots & \vdots & \ddots & \\ * & * & \cdots & * \end{pmatrix}$$

Gilt zusätzlich  $l_{ii} = 1$  für  $i = 1, \dots, n$ , dann ist  $\mathbf{L}$  eine **normierte untere Dreiecksmatrix**.



# Lineares Gleichungssystem mit unterer Dreiecksmatrix

Das lineare Gleichungssystem

$$\mathbf{Lx} = \mathbf{b}$$

entspricht

$$\begin{array}{rccccccc}
 l_{11}x_1 & & & & & & = & b_1 \\
 l_{21}x_1 & + & l_{22}x_2 & & & & = & b_2 \\
 \vdots & & & \ddots & & & \vdots & \\
 \vdots & & & & \ddots & & \vdots & \\
 l_{n1}x_1 & + & l_{n2}x_2 & + & \cdots & + & l_{nn}x_n & = & b_n
 \end{array}$$

Aus der ersten Zeile erhalten wir  $x_1$ , eingesetzt in die zweite Zeile erhalten wir  $x_2$ , usw.

Diese Vorgehensweise heißt **Vorwärtssubstitution**.

# Vorwärtssubstitution

## Algorithmus 2.4

### Eingabe:

- Untere Dreiecksmatrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  mit  $l_{ii} \neq 0$  für  $i = 1, \dots, n$
- rechte Seite  $\mathbf{b} \in \mathbb{R}^n$

### Ausgabe: Lösungsvektor $\mathbf{x} \in \mathbb{R}^n$

$$x_1 := b_1 / l_{11}$$

**for**  $i := 2$  **to**  $n$  **do**

$$x_i = \left( b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right) / l_{ii}$$

**end**

**return**  $\mathbf{x}$

## Beispiel zur Vorwärtssubstitution

### Beispiel 2.5

Gegeben sei das lineare Gleichungssystem

$$\begin{pmatrix} 5 & 0 & 0 \\ -3 & 2 & 0 \\ 1 & 6 & -7 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 25 \\ -9 \\ 9 \end{pmatrix}.$$

Mit der Vorwärtssubstitution ergibt sich:

$$\begin{aligned} x_1 &= \frac{25}{5} = 5 \\ x_2 &= \frac{-9 - (-3) \cdot 5}{2} = 3 \\ x_3 &= \frac{9 - 1 \cdot 5 - 6 \cdot 3}{-7} = 2 \end{aligned}$$

## Aufwand der Vorwärtssubstitution

- für  $i = 1$ :  
eine Division
- für  $i > 1$ :
  - ▶ eine Division
  - ▶  $i - 1$  Additionen/Subtraktionen
  - ▶  $i - 1$  Multiplikationenalso  $2i - 1$  Rechenoperationen

Gesamtaufwand:

$$1 + \sum_{i=2}^n (2i - 1) = 1 + \sum_{i=1}^{n-1} (2i + 1) = n + 2 \sum_{i=1}^{n-1} i = n + 2 \cdot \frac{n(n-1)}{2} = n^2$$

# Obere Dreiecksmatrix

## Definition 2.6

Eine Matrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$  mit  $r_{ij} = 0$  für  $i > j$  heißt **obere Dreiecksmatrix**.

$$\mathbf{R} = \begin{pmatrix} * & * & \cdots & * \\ & * & \cdots & * \\ & & \ddots & \vdots \\ 0 & & & * \end{pmatrix}$$

Gilt zusätzlich  $r_{ii} = 1$  für  $i = 1, \dots, n$ , dann ist  $\mathbf{R}$  eine **normierte obere Dreiecksmatrix**.

# Lineares Gleichungssystem mit oberer Dreiecksmatrix

Das lineare Gleichungssystem

$$\mathbf{R}\mathbf{x} = \mathbf{b}$$

entspricht

$$\begin{array}{cccccc} r_{11}x_1 & + & r_{12}x_2 & + & \cdots & + & r_{1n}x_n & = & b_1 \\ & & r_{22}x_2 & + & \cdots & + & r_{2n}x_n & = & b_2 \\ & & & & \ddots & & \vdots & & \vdots \\ & & & & & & r_{nn}x_n & = & b_n \end{array}$$

Aus der letzten Zeile erhalten wir  $x_n$ , eingesetzt in die vorletzte Zeile erhalten wir  $x_{n-1}$ , usw.

Diese Vorgehensweise heißt **Rückwärtssubstitution**.

# Rückwärtssubstitution

## Algorithmus 2.7

### Eingabe:

- Obere Dreiecksmatrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$  mit  $r_{ii} \neq 0$  für  $i = 1, \dots, n$
- rechte Seite  $\mathbf{b} \in \mathbb{R}^n$

### Ausgabe: Lösungsvektor $\mathbf{x} \in \mathbb{R}^n$

$$x_n := b_n / r_{nn}$$

**for**  $i := n - 1$  **downto** 1 **do**

$$x_i = \left( b_i - \sum_{j=i+1}^n r_{ij} x_j \right) / r_{ii}$$

**end**

**return**  $\mathbf{x}$

# Permutationsmatrix

## Definition 2.8

Eine Matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  heißt **Permutationsmatrix**, wenn sich die Zeilen von  $\mathbf{P}$  durch eine Permutation aus den Zeilen der Einheitsmatrix  $\mathbf{E} \in \mathbb{R}^{n \times n}$  ergeben.

## Beispiel 2.9

Die Matrix

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

ist eine Permutationsmatrix.



Für eine Permutation  $\pi \in S_n$  bezeichne  $\mathbf{P}(\pi)$  die Permutationsmatrix

$$\begin{pmatrix} \mathbf{e}_{\pi(1)} \\ \vdots \\ \mathbf{e}_{\pi(n)} \end{pmatrix}.$$

### Beispiel 2.10

Die Permutationsmatrix aus Beispiel 2.9 ist  $\mathbf{P}(\pi)$  mit

$$\pi = ( 3 \ 1 \ 2 ).$$

## Eigenschaften einer Permutationsmatrix

- Eine Matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  ist genau dann eine Permutationsmatrix, wenn in jeder Zeile und in jeder Spalte genau eine Eins steht (und sonst nur Nullen).
- Dies ist auch äquivalent zu einer Vertauschung der Spalten einer Einheitsmatrix.
- Definition erfolgte über Zeilenvertauschung, da wir später in erster Linie Zeilenvertauschungen vornehmen.

### Lemma 2.11

Es sei  $\mathbf{P}$  eine Permutationsmatrix. Dann gilt:

$$\mathbf{P}^{-1} = \mathbf{P}^T.$$

## Lösung eines LGS mit Permutationsmatrix

### Algorithmus 2.12

Lösung von  $\mathbf{P}\mathbf{x} = \mathbf{b}$  für eine Permutationsmatrix  $\mathbf{P} = \mathbf{P}(\pi)$ .

**for**  $i := 1$  **to**  $n$  **do**

$x_{\pi(i)} := b_i$

**end**

**return**  $\mathbf{x}$

### Algorithmus 2.13

Lösung von  $\mathbf{P}^T \mathbf{x} = \mathbf{b}$  für eine Permutationsmatrix  $\mathbf{P} = \mathbf{P}(\pi)$ .

**for**  $i := 1$  **to**  $n$  **do**

$x_i := b_{\pi(i)}$

**end**

**return**  $\mathbf{x}$

# LR-Zerlegung

- bekannt: **Eliminationsverfahren von Gauß**
- Verfahren führt zu einer Zerlegung der Koeffizientenmatrix:  $\mathbf{A} = \mathbf{LR}$

## Definition 2.14

Unter einer **LR-Zerlegung** einer Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  verstehen wir eine Zerlegung von  $\mathbf{A}$  in der Form

$$\mathbf{A} = \mathbf{LR}$$

mit einer **normierten unteren Dreiecksmatrix**  $\mathbf{L} \in \mathbb{R}^{n \times n}$  sowie einer **oberen Dreiecksmatrix**  $\mathbf{R} \in \mathbb{R}^{n \times n}$ .

# Beispiel einer LR-Zerlegung

## Beispiel 2.15

$$\underbrace{\begin{pmatrix} 2 & 1 & 7 \\ 8 & 8 & 33 \\ -4 & 10 & 4 \end{pmatrix}}_A = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ -2 & 3 & 1 \end{pmatrix}}_L \underbrace{\begin{pmatrix} 2 & 1 & 7 \\ 0 & 4 & 5 \\ 0 & 0 & 3 \end{pmatrix}}_R$$

# Lösung eines LGS mittels LR-Zerlegung

## Algorithmus 2.16

### Eingabe:

- reguläre Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , für die eine LR-Zerlegung existiert
- rechte Seite  $\mathbf{b} \in \mathbb{R}^n$

**Ausgabe:** Lösungsvektor  $\mathbf{x} \in \mathbb{R}^n$

- 1 Bestimme eine LR-Zerlegung von  $\mathbf{A}$ .
- 2 Löse  $\mathbf{L}\mathbf{y} = \mathbf{b}$  durch Vorwärtssubstitution.
- 3 Löse  $\mathbf{R}\mathbf{x} = \mathbf{y}$  durch Rückwärtssubstitution.

**Bemerkung:** Nicht für jede reguläre Matrix  $\mathbf{A}$  existiert eine LR-Zerlegung.

## Beispiel 2.17

Wir betrachten das LGS:

$$\underbrace{\begin{pmatrix} 2 & 1 & 7 \\ 8 & 8 & 33 \\ -4 & 10 & 4 \end{pmatrix}}_{\mathbf{A}} \mathbf{x} = \begin{pmatrix} 15 \\ 73 \\ 12 \end{pmatrix}$$

- 1 Die LR-Zerlegung von  $\mathbf{A}$  kennen wir aus Beispiel 2.15.
- 2 Vorwärtssubstitution zur Lösung von

$$\begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ -2 & 3 & 1 \end{pmatrix} \mathbf{y} = \begin{pmatrix} 15 \\ 73 \\ 12 \end{pmatrix}$$

liefert:  $y_1 = 15$ ,  $y_2 = 73 - 4 \cdot 15 = 13$ ,  $y_3 = 12 - 3 \cdot 13 + 2 \cdot 15 = 3$ .

## Fortsetzung Beispiel.

③ Rückwärtssubstitution zur Lösung von

$$\begin{pmatrix} 2 & 1 & 7 \\ 0 & 4 & 5 \\ 0 & 0 & 3 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 15 \\ 13 \\ 3 \end{pmatrix}$$

liefert:

$$\begin{aligned} x_3 &= \frac{3}{3} = 1 \\ x_2 &= \frac{13 - 5}{4} = 2 \\ x_1 &= \frac{15 - 2 - 7}{2} = 3 \end{aligned}$$



## Fragen zur LR-Zerlegung

Schritt 1 aus Algorithmus 2.16 wirft folgende Fragen auf:

- 1 Wie bestimmt man eine LR-Zerlegung einer Matrix  $\mathbf{A}$ ?
- 2 Wann existiert solch eine LR-Zerlegung?
- 3 Ist solch eine LR-Zerlegung eindeutig bestimmt?

## Beispiel zur Konstruktion einer LR-Zerlegung

### Beispiel 2.18

Wir betrachten das lineare Gleichungssystem  $\mathbf{Ax} = \mathbf{b}$  mit

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} =: \mathbf{A}^{(1)}$$

und beliebiger rechter Seite  $\mathbf{b} \in \mathbb{R}^4$ .

Wir subtrahieren:

- von der zweiten Zeile das zweifache der ersten Zeile,
- von der dritten Zeile das vierfache der ersten Zeile,
- von der vierten Zeile das dreifache der ersten Zeile.

## Fortsetzung Beispiel.

In Matrixschreibweise:

$$\mathbf{L}_1 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{pmatrix}$$

Dann gilt:

$$\mathbf{L}_1 \mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{pmatrix} =: \mathbf{A}^{(2)}$$

## Fortsetzung Beispiel.

Sei

$$\mathbf{L}_2 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & -4 & 0 & 1 \end{pmatrix}$$

Dann gilt:

$$\mathbf{L}_2 \mathbf{L}_1 \mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{pmatrix} =: \mathbf{A}^{(3)}$$

## Fortsetzung Beispiel.

Sei

$$\mathbf{L}_3 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Dann gilt:

$$\mathbf{L}_3 \mathbf{L}_2 \mathbf{L}_1 \mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix} =: \mathbf{A}^{(4)} =: \mathbf{R}$$

Damit haben wir die gewünschte obere Dreiecksmatrix  $\mathbf{R}$ .

## Fortsetzung Beispiel.

Fazit:

- Eventuelle Zerlegung mittels  $\mathbf{L} := \mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\mathbf{L}_3^{-1}$ .
- Damit gilt dann  $\mathbf{A} = \mathbf{LR}$ .

## Entscheidende Fragen:

- Wie sehen die  $\mathbf{L}_j^{-1}$  aus?
- Ist  $\mathbf{L}$  eine normierte untere Dreiecksmatrix?

# Frobenius-Matrix

## Definition 2.19

Eine Matrix  $\mathbf{L}_j \in \mathbb{R}^{n \times n}$  mit der Gestalt

$$\mathbf{L}_j = \begin{pmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & -l_{j+1,j} & 1 & & & \\ & & \vdots & & \ddots & & \\ 0 & & -l_{n,j} & & & \ddots & 1 \end{pmatrix}$$

heißt **Frobenius-Matrix**.

## Diskussion zur Frobenius-Matrix

- Einheitsmatrix mit Ausnahme der  $j$ -ten Spalte
- in Spalte  $j$  unterhalb der Diagonalen die Einträge  $-l_{j+1,j}, \dots, -l_{n,j}$
- Wirkung von  $\mathbf{L}_j \mathbf{A}$ : Das  $l_{i,j}$ -fache der  $j$ -ten Zeile von  $\mathbf{A}$  wird von Zeile  $i$  in  $\mathbf{A}$  abgezogen.



## Inverse einer Frobenius-Matrix

### Lemma 2.20

Es sei  $\mathbf{L}_j \in \mathbb{R}^{n \times n}$  eine *Frobenius-Matrix* mit der Darstellung wie in Definition 2.19.

Dann gilt:

$$\mathbf{L}_j^{-1} = \begin{pmatrix} 1 & & & & & & & & 0 \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & +l_{j+1,j} & 1 & & & & & \\ & & \vdots & & \ddots & & & & \\ 0 & & +l_{n,j} & & & \ddots & & & 1 \end{pmatrix}.$$

- Das *Inverse* einer Frobenius-Matrix ist *wieder eine Frobenius-Matrix*.
- Zur Bestimmung der inversen Matrix müssen wir *nur die Vorzeichen* in Spalte  $j$  unterhalb der Diagonalen *herumdrehen*.

## Lemma 2.21

Es seien  $\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_{n-1}$  *Frobenius-Matrizen* mit der Darstellung wie in Definition 2.19, also für  $\mathbf{L}_j$  mit Einträgen in Spalte  $j$ .

Dann gilt:

$$\mathbf{L} := \mathbf{L}_1^{-1} \mathbf{L}_2^{-1} \cdots \mathbf{L}_{n-1}^{-1} = \begin{pmatrix} 1 & & & 0 \\ l_{2,1} & 1 & & \\ \vdots & \ddots & \ddots & \\ l_{n,1} & \cdots & l_{n,n-1} & 1 \end{pmatrix}.$$

- Damit ist  $\mathbf{L}$  tatsächlich eine *normierte untere Dreiecksmatrix*.
- Wenn die Matrizen  $\mathbf{L}_1, \dots, \mathbf{L}_{n-1}$  vorliegen, können wir  $\mathbf{L}$  direkt (also ohne weitere Rechnung) angeben.

## Beispiel 2.22

Mit Lemma 2.21 folgt als LR-Zerlegung für Beispiel 2.18:

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

# Grundversion einer LR-Zerlegung

## Algorithmus 2.23

```
A(1) := A  
for  $j := 1$  to  $n - 1$  do  
    for  $i := j + 1$  to  $n$  do  
         $l_{i,j} := a_{i,j}^{(j)} / a_{j,j}^{(j)}$   
    end  
    Definiere  $L_j$  gemäß 2.19  
    A( $j+1$ ) := L $j$ A( $j$ )  
end  
R := A( $n$ )
```

# Hauptabschnittsmatrix

## Definition 2.24

Sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Dann heißt

$$\mathbf{A}[k] := \begin{pmatrix} a_{1,1} & \cdots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{k,1} & \cdots & a_{k,k} \end{pmatrix} \in \mathbb{R}^{k \times k} \quad \text{für } k = 1, \dots, n$$

die **führende  $k \times k$ -Hauptabschnittsmatrix** von  $\mathbf{A}$  und  $\det(\mathbf{A}[k])$  die **führende  $k$ -te Hauptabschnittsdeterminante**.

## Beispiel 2.25

Für die  $3 \times 3$ -Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

sind die drei führenden Hauptabschnittsmatrizen

$$\mathbf{A}[1] = (1), \quad \mathbf{A}[2] = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}, \quad \mathbf{A}[3] = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

## Existenz einer LR-Zerlegung

### Satz 2.26

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine reguläre Matrix.

Dann besitzt  $\mathbf{A}$  genau dann eine LR-Zerlegung, wenn  $\det(\mathbf{A}[k]) \neq 0$  für alle  $k = 1, \dots, n$  gilt.

### Beispiel 2.27

- Die Matrix  $\mathbf{A}$  aus Beispiel 2.25 besitzt eine LR-Zerlegung.
- Für die reguläre Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 3 & 7 \end{pmatrix}$$

existiert keine LR-Zerlegung.

# Eigenschaften von Dreiecksmatrizen

## Lemma 2.28

- (i) Sind  $\mathbf{L} \in \mathbb{R}^{n \times n}$  und  $\mathbf{L}' \in \mathbb{R}^{n \times n}$  untere (normierte) Dreiecksmatrizen, so ist auch  $\mathbf{L}\mathbf{L}'$  eine untere (normierte) Dreiecksmatrix.
- (ii) Ist  $\mathbf{L} \in \mathbb{R}^{n \times n}$  eine reguläre untere (normierte) Dreiecksmatrix, so ist auch  $\mathbf{L}^{-1}$  eine untere (normierte) Dreiecksmatrix.
- (iii) Die Aussagen (i) und (ii) gelten analog für obere (normierte) Dreiecksmatrizen.



# Eindeutigkeit der LR-Zerlegung

## Satz 2.29

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine reguläre Matrix mit  $\det(\mathbf{A})[k] \neq 0$  für  $k = 1, \dots, n$ .

Dann ist die existierende LR-Zerlegung von  $\mathbf{A}$  eindeutig bestimmt.

## Beweis.

- Es seien  $\mathbf{L}_1 \mathbf{R}_1 = \mathbf{A} = \mathbf{L}_2 \mathbf{R}_2$  zwei LR-Zerlegungen für  $\mathbf{A}$ .
- Daraus folgt  $\mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{L}_1^{-1} \mathbf{L}_2$ .
- Nach Lemma 2.28 ist  $\mathbf{R}_1 \mathbf{R}_2^{-1}$  eine obere Dreiecksmatrix und  $\mathbf{L}_1^{-1} \mathbf{L}_2$  eine untere normierte Dreiecksmatrix.
- Aus der Gleichheit und den speziellen Eigenschaften folgt  $\mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{E} = \mathbf{L}_1^{-1} \mathbf{L}_2$ .
- Damit folgt  $\mathbf{R}_1 = \mathbf{R}_2$  und  $\mathbf{L}_1 = \mathbf{L}_2$ .

## Aufwand der LR-Zerlegung

### Algorithmus 2.30

*Konkreter Kern der LR-Zerlegung:*

```
for  $j := 1$  to  $n - 1$  do  
    for  $i := j + 1$  to  $n$  do  
         $l_{i,j} := a_{i,j} / a_{j,j}$   
        for  $k := j + 1$  to  $n$  do  
             $a_{i,k} := a_{i,k} - l_{i,j} * a_{j,k}$   
        end  
    end  
end
```

Hauptaufwand: innerste Schleife, dort zwei Operationen

Wenn wir über alle Schleifendurchläufe der innersten Schleife summieren, erhalten wir:

$$\begin{aligned}\sum_{j=1}^{n-1} \sum_{i=j+1}^n \sum_{k=j+1}^n 2 &= 2 \sum_{j=1}^{n-1} \sum_{i=j+1}^n (n-j) \\ &= 2 \sum_{j=1}^{n-1} (n-j)^2 \\ &= 2 \sum_{j=1}^{n-1} j^2 \\ &= 2 \frac{(n-1)n(2n-1)}{6} \approx \frac{2}{3} n^3\end{aligned}$$

**Fazit:** Der Zeitaufwand der LR-Zerlegung beträgt  $O(n^3)$ .

# LR-Zerlegung mit Pivotisierung

- Die Matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

hat **keine LR-Zerlegung**.

- Die Matrix  $\mathbf{A}$  ist aber **regulär** und somit wäre  $\mathbf{Ax} = \mathbf{b}$  **stets eindeutig lösbar**.
- Wie können wir dieses Problem lösen? Zeilenvertauschung!
- Mit geeigneten **Zeilenvertauschungen** existiert stets eine LR-Zerlegung.
- Formal repräsentieren wir Zeilenvertauschungen durch eine **Permutationsmatrix**.

# Existenz einer Zerlegung mit Zeilenvertauschung

## Satz 2.31

Jede reguläre Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  besitzt eine Zerlegung der Gestalt

$$\mathbf{PA} = \mathbf{LR}$$

mit

- einer *Permutationsmatrix*  $\mathbf{P} \in \mathbb{R}^{n \times n}$ ,
- einer *normierten unteren Dreiecksmatrix*  $\mathbf{L} \in \mathbb{R}^{n \times n}$  und
- einer *oberen Dreiecksmatrix*  $\mathbf{R} \in \mathbb{R}^{n \times n}$ .

## Beispiel 2.32

Wir betrachten die reguläre Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 3 & 7 \end{pmatrix}$$

aus Beispiel 2.27, für die **keine LR-Zerlegung existiert**.

Dagegen ist, wenn wir die zweite und dritte Zeile von  $\mathbf{A}$  vertauschen, eine LR-Zerlegung möglich:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{\mathbf{P}} \mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 3 & 7 \\ 1 & 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 0 & 3 & 7 \\ 0 & 0 & -1 \end{pmatrix}}_{\mathbf{R}}.$$

# Lösung eines LGS mittels LR-Zerlegung und Pivotisierung

$$\mathbf{Ax} = \mathbf{b} \iff \mathbf{PAx} = \mathbf{Pb} \iff \mathbf{LRx} = \mathbf{Pb}$$

führt zu folgendem Algorithmus:

## Algorithmus 2.33

### Eingabe:

- reguläre Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$
- rechte Seite  $\mathbf{b} \in \mathbb{R}^n$

### Ausgabe:

- 1 Bestimme eine Zerlegung  $\mathbf{PA} = \mathbf{LR}$ .
- 2 Setze  $\mathbf{c} := \mathbf{Pb}$ .
- 3 Löse  $\mathbf{Ly} = \mathbf{c}$  durch Vorwärtssubstitution.
- 4 Löse  $\mathbf{Rx} = \mathbf{y}$  durch Rückwärtssubstitution.





## Beispiel zu LR-Zerlegung mit Pivotisierung

### Beispiel 2.34

Wir betrachten die Matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & -1 & -2 \\ 2 & -2 & 4 & -1 \\ 1 & 1 & 1 & 1 \\ -2 & 1 & -2 & 1 \end{pmatrix}.$$

Wir vertauschen die beiden ersten Zeilen und erhalten damit

$$\mathbf{P}_{1,2}\mathbf{A} = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 1 & 1 & 1 & 1 \\ -2 & 1 & -2 & 1 \end{pmatrix}.$$

## Fortsetzung Beispiel.

Mit der Frobenius-Matrix

$$\mathbf{L}_1 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -0.5 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

erhalten wir

$$\mathbf{L}_1 \mathbf{P}_{1,2} \mathbf{A} = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 2 & -1 & 1.5 \\ 0 & -1 & 2 & 0 \end{pmatrix}.$$

## Fortsetzung Beispiel.

Jetzt ist keine Zeilenvertauschung notwendig, dies machen wir durch

$$\mathbf{P}_{2,2} := \mathbf{E}$$

deutlich. Mit der Frobenius-Matrix

$$\mathbf{L}_2 := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0.5 & 0 & 1 \end{pmatrix}$$

erhalten wir

$$\mathbf{L}_2 \mathbf{P}_{2,2} \mathbf{L}_1 \mathbf{P}_{1,2} \mathbf{A} = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 0 & 0 & 3.5 \\ 0 & 0 & 1.5 & -1 \end{pmatrix}.$$

### Fortsetzung Beispiel.

In der letzten Iteration vertauschen wir mittels  $\mathbf{P}_{3,4}$  die dritte und vierte Zeile.

Anschließend ist keine weitere Transformation mehr notwendig, d. h. wir wählen  $\mathbf{L}_3 := \mathbf{E}$  und erhalten damit

$$\mathbf{L}_3 \mathbf{P}_{3,4} \mathbf{L}_2 \mathbf{P}_{2,2} \mathbf{L}_1 \mathbf{P}_{1,2} \mathbf{A} = \begin{pmatrix} 2 & -2 & 4 & -1 \\ 0 & 2 & -1 & -2 \\ 0 & 0 & 1.5 & -1 \\ 0 & 0 & 0 & 3.5 \end{pmatrix} =: \mathbf{R}.$$

## Fortsetzung Beispiel.

Wenn wir jetzt durch

$$\mathbf{c} := \mathbf{L}_3 \mathbf{P}_{3,4} \mathbf{L}_2 \mathbf{P}_{2,2} \mathbf{L}_1 \mathbf{P}_{1,2} \mathbf{b}$$

die gleichen Transformationen auch auf die rechte Seite  $\mathbf{b}$  anwenden, könnten wir das ursprüngliche LGS  $\mathbf{Ax} = \mathbf{b}$  mittels Rückwärtssubstitution für

$$\mathbf{Rx} = \mathbf{c}$$

lösen.

# Grundversion einer LR-Zerlegung mit Pivotisierung

## Algorithmus 2.35

$\mathbf{A}^{(1)} := \mathbf{A}$

**for**  $j := 1$  **to**  $n - 1$  **do**

*Wähle aus der  $j$ -ten Spalte von  $\mathbf{A}^{(j)}$  ein Element  $a_{ij}^{(j)} \neq 0$  mit  $i \geq j$*

$\tilde{\mathbf{A}}^{(j)} := \mathbf{P}_{j,i} \mathbf{A}^{(j)}$

**for**  $i := j + 1$  **to**  $n$  **do**

$l_{i,j} := \tilde{a}_{i,j}^{(j)} / \tilde{a}_{j,j}^{(j)}$

**end**

*Definiere  $\mathbf{L}_j$  gemäß 2.19*

$\mathbf{A}^{(j+1)} := \mathbf{L}_j \tilde{\mathbf{A}}^{(j)}$

**end**

$\mathbf{R} := \mathbf{A}^{(n)}$

## Bemerkungen zur Pivotisierung

- Im Allgemeinen wählt man das Pivotelement  $a_{ij}^{(j)}$  so, dass

$$a_{ij}^{(j)} = \max_{j \leq k \leq n} |a_{kj}^{(j)}|$$

gilt.

- Insbesondere führt man auch dann eine Zeilenvertauschung durch, wenn  $a_{jj}^{(j)} \neq 0$  gilt.
- Grund: **bessere numerische Stabilität**
- Man nennt das hier gezeigte Vorgehen **Spaltenpivotisierung**.
- Algorithmus 2.35 liefert tatsächlich eine Zerlegung der Form  $\mathbf{PA} = \mathbf{LR}$ . Auf einen Beweis hierfür verzichten wir.

## Bemerkungen zu einer Implementierung

- Die durchgeführten **Zeilenvertauschungen** merken wir uns einfach in einer Permutation  $\pi$  repräsentiert als **Array**.
- Initialisierung:  $\pi = ( 1 \ 2 \ \dots \ n )$
- Aus  $\pi$  können am Ende **P** bestimmen.
- Die  $l_{i,j}$  tragen wir in der Matrix **L** ein.
- **Wenn wir in Iteration  $j$  eine Zeilenvertauschung der Zeilen  $i$  und  $j$  vornehmen, tauschen wir auch in **L** die Einträge in diesen Zeilen, aber nur bis zur Spalte  $j - 1$ .**



## Beispiele zur LR-Zerlegung mit Pivotisierung

### Beispiel 2.36

Für Beispiel 2.34 erhalten wir

$$\mathbf{P} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -0.5 & 1 & 0 \\ 0.5 & 1 & 0 & 1 \end{pmatrix}.$$

Probe und Herleitung, Tafel 

## Beispiel 2.37


Wir betrachten die Matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 6 & 1 \\ 2 & 3 & 2 \\ 4 & 2 & 1 \end{pmatrix}$$

und führen eine LR-Zerlegung mit Pivotisierung durch, wobei wir jeweils durch Zeilenvertauschung das betragsmäßig größte Element auf die Diagonale ziehen.

Als Ergebnis erhalten wir

$$\mathbf{P} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & \frac{4}{11} & 1 \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} 4 & 2 & 1 \\ 0 & \frac{11}{2} & \frac{3}{4} \\ 0 & 0 & \frac{27}{22} \end{pmatrix}.$$

Durchführung Tafel .

## Fortsetzung Beispiel.

Jetzt betrachten wir noch das LGS

$$\mathbf{Ax} = \begin{pmatrix} 16 \\ 14 \\ 11 \end{pmatrix} =: \mathbf{b}.$$

Wir berechnen  $\mathbf{Pb}$ :

$$\mathbf{Pb} = \begin{pmatrix} 11 \\ 16 \\ 14 \end{pmatrix} =: \mathbf{c}.$$

Wir lösen  $\mathbf{Ly} = \mathbf{c}$ :

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & \frac{4}{11} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 11 \\ 16 \\ 14 \end{pmatrix}$$

ergibt  $y_1 = 11, y_2 = \frac{53}{4}, y_3 = \frac{81}{22}$ .

## Fortsetzung Beispiel.

Wir lösen  $\mathbf{R}\mathbf{x} = \mathbf{y}$ :

$$\begin{pmatrix} 4 & 2 & 1 \\ 0 & \frac{11}{2} & \frac{3}{4} \\ 0 & 0 & \frac{27}{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 11 \\ \frac{53}{4} \\ \frac{81}{22} \end{pmatrix}$$

ergibt  $x_3 = 3, x_2 = 2, x_1 = 1$ .

# Tridiagonalmatrix

## Definition 2.38

Eine Matrix  $\mathbf{T} \in \mathbb{R}^{n \times n}$  mit  $d_{ij} = 0$  für  $|i - j| \geq 2$  heißt **Tridiagonalmatrix**.

Eine Tridiagonalmatrix hat also die Gestalt

$$\mathbf{T} = \begin{pmatrix} * & * & & & 0 \\ * & * & * & & \\ & * & * & \ddots & \\ & & \ddots & \ddots & * \\ 0 & & & * & * \end{pmatrix}.$$

## Bemerkungen

- Eine Tridiagonalmatrix hat also nur auf der **Hauptdiagonalen** und den **beiden Nebendiagonalen** Einträge  $\neq 0$ .
- Gleichungssysteme mit Tridiagonalmatrix treten z. B. bei einer **Interpolation mit kubischen Splines** auf.
- Ziel: Definition einer **Spezialversion der LR-Zerlegung für Tridiagonalmatrizen**
- Motivation: geringerer Berechnungsaufwand als bei der normalen LR-Zerlegung

## Lösungsansatz

Wir vermuten, dass bei einer LR-Zerlegung einer Tridiagonalmatrix die Matrizen **L** und **R** **Bidiagonalmatrizen** sind.

$$\begin{aligned}
 \mathbf{T} &= \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \gamma_2 & \alpha_2 & \beta_3 & & \\ & \gamma_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_n \\ 0 & & & \gamma_n & \alpha_n \end{pmatrix} \\
 &= \underbrace{\begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} d_1 & r_2 & & & \\ & d_2 & r_3 & & \\ & & d_3 & \ddots & \\ & & & \ddots & r_n \\ & & & & d_n \end{pmatrix}}_{\mathbf{R}}
 \end{aligned}$$

Mit diesem Ansatz erhalten wir:

$$\begin{array}{rcl}
 & \alpha_1 = d_1 & \beta_2 = r_2 \\
 \gamma_2 = l_2 d_1 & \alpha_2 = l_2 r_2 + d_2 & \beta_3 = r_3 \\
 \gamma_3 = l_3 d_2 & \alpha_3 = l_3 r_3 + d_3 & \beta_4 = r_4 \\
 \vdots & \vdots & \vdots \\
 \vdots & \vdots & \beta_n = r_n \\
 \gamma_n = l_n d_{n-1} & \alpha_n = l_n r_n + d_n & 
 \end{array}$$

Wenn wir nach den gesuchten Komponenten  $d_i$ ,  $r_i$  und  $l_i$  auflösen, erhalten wir die gesuchten Elemente für die LR-Zerlegung (ohne Pivotisierung).



# LR-Zerlegung einer Tridiagonalmatrix

## Algorithmus 2.39

$$d_1 := \alpha_1$$

$$r_2 := \beta_2$$

**for**  $j := 2$  **to**  $n - 1$  **do**

$$l_j := \gamma_j / d_{j-1}$$

$$d_j := \alpha_j - l_j \cdot d_j$$

$$r_{j+1} := \beta_{j+1}$$

**end**

$$l_n := \gamma_n / d_{n-1}$$

$$d_n := \alpha_n - l_n \cdot r_n$$

## Anpassung von Vorwärts- und Rückwärtssubstitution

### Algorithmus 2.40

*Vorwärtssubstitution für  $\mathbf{L}\mathbf{y} = \mathbf{b}$ .*

$$y_1 := b_1$$

**for**  $j := 2$  **to**  $n$  **do**

$$y_j := b_j - l_j \cdot y_{j-1}$$

**end**

### Algorithmus 2.41

*Rückwärtssubstitution für  $\mathbf{R}\mathbf{x} = \mathbf{y}$ .*

$$x_n := y_n / d_n$$

**for**  $j := n - 1$  **downto**  $1$  **do**

$$x_j := (y_j - r_{j+1} \cdot x_{j+1}) / d_j$$

**end**

## Aufwand und Anwendbarkeit

Zeitaufwand:  $O(n)$

### Lemma 2.42

Sei  $\mathbf{T}$  eine Tridiagonalmatrix. Wenn die Komponenten der Matrix die Ungleichungen

$$\begin{aligned} |\alpha_1| &> |\beta_2| \\ |\alpha_j| &\geq |\gamma_j| + |\beta_{j+1}| \quad \text{für } j = 2, \dots, n-1 \\ |\alpha_n| &\geq |\gamma_n| \\ \gamma_j &\neq 0 \quad \text{für } j = 2, \dots, n \end{aligned}$$

erfüllen, dann ist der Algorithmus zur LR-Zerlegung einer Tridiagonalmatrix (2.39) wohldefiniert.

## Positiv definite Matrix

### Definition 2.43

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine **symmetrische** Matrix.

$\mathbf{A}$  heißt **positiv definit**, wenn für alle  $\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}$  gilt:

$$\mathbf{x}^T \mathbf{A} \mathbf{x} > 0.$$

### Beispiel 2.44

Die Einheitsmatrix  $\mathbf{E}$  ist positiv definit, denn:

$$\mathbf{x}^T \mathbf{E} \mathbf{x} = \mathbf{x}^T \mathbf{x} = \langle \mathbf{x}, \mathbf{x} \rangle = \|\mathbf{x}\|_2^2 > 0 \quad \text{für } \mathbf{x} \neq \mathbf{0}.$$

**Bemerkung:**  $\|\mathbf{x}\|_2$  bezeichnet die **euklidische Norm** von  $\mathbf{x}$ .

# Definitheit allgemein

Natürlich existieren weitere Definitheitsbegriffe:

- positiv semidefinit:  $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$
- negativ definit:  $\mathbf{x}^T \mathbf{A} \mathbf{x} < 0$
- negativ semidefinit:  $\mathbf{x}^T \mathbf{A} \mathbf{x} \leq 0$
- indefinit:  $\exists \mathbf{x} : \mathbf{x}^T \mathbf{A} \mathbf{x} > 0$  und  $\exists \mathbf{y} : \mathbf{y}^T \mathbf{A} \mathbf{y} < 0$

Wir untersuchen aber nur positiv definite Matrizen.

## Anwendungen

- Wir betrachten positiv definite Matrizen im Zusammenhang mit **Ausgleichsproblemen** im nächsten Kapitel.
- Dort treten positiv definite **quadratische Formen** auf.
- Eine **quadratische Form** ist eine Funktion  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  der Art:

$$f(\mathbf{x}) = \sum_{i=1}^n \sum_{j=i}^n \alpha_{ij} x_i x_j$$

- Beispiel:

$$f(x_1, x_2) = 2x_1^2 + 4x_1x_2 + 3x_2^2$$

- In Matrixschreibweise:

$$f(x_1, x_2) = \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 2 & 2 \\ 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

# Cholesky-Zerlegung

## Satz 2.45

Eine symmetrische Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  ist genau dann positiv definit, wenn eine reguläre untere Dreiecksmatrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  existiert, mit

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T.$$

- Wir können leicht zeigen, dass aus der Existenz von  $\mathbf{L}$  die positive Definitheit folgt:

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \mathbf{L} \mathbf{L}^T \mathbf{x} = \left( \mathbf{L}^T \mathbf{x} \right)^T \left( \mathbf{L}^T \mathbf{x} \right) = \|\mathbf{L}^T \mathbf{x}\|_2^2 > 0 \quad \text{für } \mathbf{x} \neq \mathbf{0}.$$

- Die Umkehrung ergibt sich aus der Konstruktion des folgenden Algorithmus.

- Wenn wir zusätzlich  $l_{ii} > 0$  verlangen, ist die Matrix  $\mathbf{L}$  sogar eindeutig bestimmt.
- Diese eindeutige Zerlegung nennen wir **Cholesky-Zerlegung**

### Definition 2.46

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine positiv definite Matrix.

Dann heißt eine Zerlegung von  $\mathbf{A}$  in der Form

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T$$

mit einer unteren Dreiecksmatrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  und  $l_{ii} > 0$  für  $i = 1, \dots, n$  **Cholesky-Zerlegung** von  $\mathbf{A}$ .



### Beispiel 2.47

Die Matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 2 \\ 2 & 3 \end{pmatrix}$$

hat die Cholesky-Zerlegung

$$\begin{pmatrix} \sqrt{2} & 0 \\ \sqrt{2} & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 1 \end{pmatrix}.$$

# Lösung eines LGS mittels Cholesky-Zerlegung

## Algorithmus 2.48

Lösung eines linearen Gleichungssystems  $\mathbf{Ax} = \mathbf{b}$  für eine positiv definite Matrix  $\mathbf{A}$ .

- 1 Bestimme eine Cholesky-Zerlegung  $\mathbf{A} = \mathbf{LL}^T$ .
- 2 Löse  $\mathbf{Ly} = \mathbf{b}$  durch Vorwärtssubstitution.
- 3 Löse  $\mathbf{L}^T \mathbf{x} = \mathbf{y}$  durch Rückwärtssubstitution.

# Herleitung der Cholesky-Zerlegung

Wir vergleichen:

$$\underbrace{\begin{pmatrix} a_{1,1} & a_{2,1} & \cdots & a_{n,1} \\ a_{2,1} & a_{2,2} & \cdots & a_{n,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}}_{\mathbf{A}} = \underbrace{\begin{pmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & \vdots & \ddots & \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} l_{1,1} & l_{2,1} & \cdots & l_{n,1} \\ & l_{2,2} & \cdots & l_{n,2} \\ & & \ddots & \vdots \\ & & & l_{n,n} \end{pmatrix}}_{\mathbf{L}^T}$$

Daraus ergibt sich:

$$\begin{array}{l} a_{1,1} = l_{1,1}^2 \quad \implies \quad l_{1,1} = \sqrt{a_{1,1}} \\ \hline a_{2,1} = l_{2,1} l_{1,1} \quad \implies \quad l_{2,1} = a_{2,1} / l_{1,1} \\ \hline a_{2,2} = l_{2,1}^2 + l_{2,2}^2 \quad \implies \quad l_{2,2} = \sqrt{a_{2,2} - l_{2,1}^2} \\ \hline \end{array}$$

$$\begin{array}{lcl}
 a_{3,1} = l_{3,1}l_{1,1} & \implies & l_{3,1} = a_{3,1}/l_{1,1} \\
 a_{3,2} = l_{3,1}l_{2,1} + l_{3,2}l_{2,2} & \implies & l_{3,2} = (a_{3,2} - l_{3,1}l_{2,1})/l_{2,2} \\
 a_{3,3} = l_{3,1}^2 + l_{3,2}^2 + l_{3,3}^2 & \implies & l_{3,3} = \sqrt{a_{3,3} - l_{3,1}^2 - l_{3,2}^2} \\
 \vdots & \vdots & \vdots \\
 \hline
 a_{n,1} = l_{n,1}l_{1,1} & \implies & l_{n,1} = a_{n,1}/l_{1,1} \\
 a_{n,2} = l_{n,1}l_{2,1} + l_{n,2}l_{2,2} & \implies & l_{n,2} = (a_{n,2} - l_{n,1}l_{2,1})/l_{2,2} \\
 \vdots & \vdots & \vdots \\
 a_{n,n} = l_{n,1}^2 + l_{n,2}^2 + \cdots + l_{n,n}^2 & \implies & l_{n,n} = \sqrt{a_{n,n} - l_{n,1}^2 - \cdots - l_{n,n-1}^2}
 \end{array}$$

## Beispiel 2.49

Für die Matrix

$$\mathbf{A} = \begin{pmatrix} 4 & -2 & 6 \\ -2 & 5 & -1 \\ 6 & -1 & 26 \end{pmatrix}$$

erhalten wir

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & 0 \\ -1 & 2 & 0 \\ 3 & 1 & 4 \end{pmatrix}.$$

# Algorithmus zur Cholesky-Zerlegung

## Algorithmus 2.50

**Eingabe:** positiv definite Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$

**Ausgabe:** Matrix  $\mathbf{L} \in \mathbb{R}^{n \times n}$  mit  $l_{i,i} > 0$  und  $\mathbf{A} = \mathbf{L}\mathbf{L}^T$

**for**  $i := 1$  **to**  $n$  **do**

**for**  $j := 1$  **to**  $i - 1$  **do**

$$l_{i,j} := \left( a_{i,j} - \sum_{k=1}^{j-1} l_{j,k} l_{i,k} \right) / l_{j,j}$$

**end**

$$l_{i,i} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2}$$

**end**

Aufwand:  $\approx \frac{1}{3}n^3$  Operationen, also die  **Hälfte der LR-Zerlegung**.

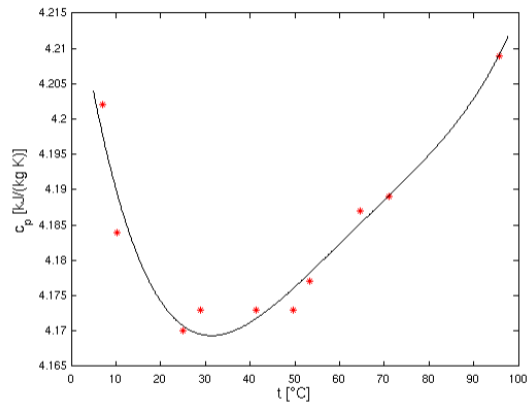
# Zusammenfassung

Lösungsalgorithmen für Gleichungssysteme:

- Dreiecksmatrizen: Vorwärts- und Rückwärtssubstitution
- regulär: LR-Zerlegung
- Problem bei LR-Zerlegung: existiert evtl. nicht, dann Pivotisierung
- tridiagonal: mit effizientem Spezialalgorithmus in linearer Zeit
- positiv definit: Cholesky-Zerlegung

## Kapitel 3

## Ausgleichsrechnung





# Inhalt

## 3 Ausgleichsrechnung

- Ausgleichsproblem
- Lineare Ausgleichsprobleme
- Orthogonalisierungsverfahren
- Modellauswahl

# Ausgleichsproblem

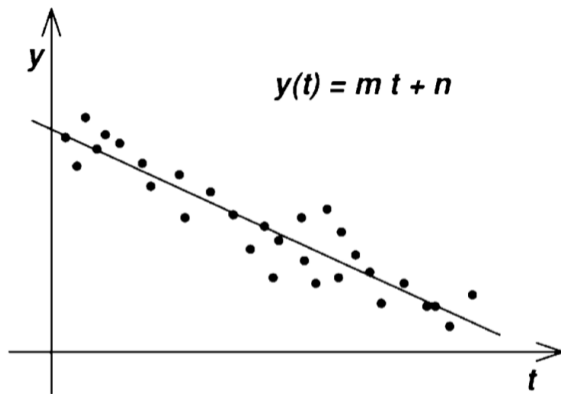
## Gegeben:

- $n$  Wertepaare  $(x_i, y_i)$ ,  $i = 1, \dots, n$  mit  $x_i \neq x_j$  für  $i \neq j$ .

## Gesucht:

- Eine stetige Funktion  $f$ , die in einem gewissen Sinne die Wertepaare bestmöglich annähert,
- d. h. es soll möglichst genau  $f(x_i) \approx y_i$  für alle  $i = 1, \dots, n$  gelten.

## Beispiel: Messwerte durch eine Gerade annähern



Wie sollen wir  $m$  und  $n$  wählen?

## Bemerkungen zum Ausgleichsproblem

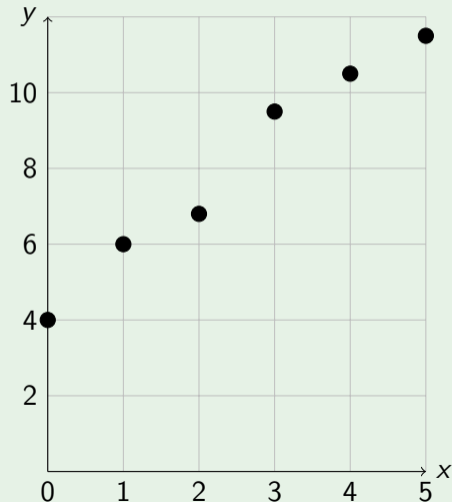
- Wenn wir die Menge der zulässigen stetigen Funktionen nicht weiter einschränken, können wir ein **Interpolationspolynom mit Grad  $n - 1$**  finden, das  $f(x_i) = y_i$  für  $i = 1, \dots, n$  erfüllt..
- Solche Interpolationspolynome sind aber **für die Praxis meist untauglich**.
- Die **Wertepaare sind oft empirischer Natur**, sie entstammen z. B. Messungen, die **fehlerbehaftet** sind.
- Daher ist die **Forderung einer exakten Anpassung nicht sinnvoll**.
- Weiteres Problem: Die **Anzahl der Parameter würde linear mit der Anzahl der Messungen wachsen**.
- Der tatsächliche funktionale Zusammenhang ist aber natürlich unabhängig von der Anzahl der Messungen.

# Beispiel

## Beispiel 3.1

Wir haben als Wertepaare ( $n = 6$ ):

$x_i$	0	1	2	3	4	5
$y_i$	4	6	6.8	9.5	10.5	11.5



# Ausgleichsfunktion, Fehlerfunktional

## Definition 3.2

Gegeben sei:

- eine Menge  $\mathcal{F}$  von stetigen Funktionen auf  $[a, b]$ , also  $\mathcal{F} \subseteq C([a, b])$ ,
- sowie  $n$  Wertepaare  $(x_i, y_i)$ ,  $i = 1, \dots, n$ .

Eine Funktion  $f \in \mathcal{F}$  heißt **Ausgleichsfunktion** von  $\mathcal{F}$  zu den gegebenen Wertepaaren, falls  $f$  das **Fehlerfunktional**

$$E(f) := \sum_{i=1}^n (y_i - f(x_i))^2$$

minimiert, d. h.  $E(f) = \min\{E(g) | g \in \mathcal{F}\}$ .

Die Menge  $\mathcal{F}$  heißt auch **Menge der Ansatzfunktionen**.

# Bemerkungen

Typische Situation:

- Wir haben eine Anzahl fehlerbehafteter Messdaten vorliegen und versuchen nun,
- die Parameter eines Modells so anzupassen, dass das Modell die beobachteten Messdaten möglichst gut wiedergibt.
- Das Modell besteht meist aus einer parameterbehafteten Kombination von Funktionen.
- Üblicherweise haben wir viel mehr Messdaten als Modellparameter.

## Fehlerfunktionale

- In Definition 3.2 haben wir das Fehlerfunktional  $E(f)$  auf Basis der **euklidische Norm**  $\| \cdot \|_2$  definiert. Prinzipiell könnten wir auch andere Normen verwenden.
- Für die **Summennorm**  $\| \cdot \|_1$  würde sich

$$E(f) := \sum_{i=1}^n |y_i - f(x_i)|$$

als Fehlerfunktional ergeben.

- Auch möglich: **Gewichtung der Fehler**. Damit entsteht

$$E(f) := \sum_{i=1}^n w_i (y_i - f(x_i))^2$$

mit  $w_i > 0$ .

- Anwendung: Berücksichtigung der Qualität oder **Zuverlässigkeit einer Messung**.



- Das Fehlerfunktional auf Basis der Maximumsnorm  $\| \cdot \|_{\infty}$  ist:

$$E(f) = \max_{i=1}^n |y_i - f(x_i)|$$

- Die Fehlerfunktionale unterscheiden sich insbesondere in ihrer **Empfindlichkeit gegenüber Ausreißern**. Dies können wir auch als **Robustheit** betrachten.
- Wir nutzen in der Regel das Fehlerfunktional basierend auf der Norm  $\| \cdot \|_2$ . Dies führt zu speziellen **linearen Gleichungssystemen**.
- Ausgleichsprobleme mit Fehlerfunktional basierend auf  $\| \cdot \|_1$  oder  $\| \cdot \|_{\infty}$  führen dagegen zu **linearen Programmen**.  
☞ nächstes Kapitel

# Ansatzfunktionen

- Im einfachsten Fall wählt man als  $\mathcal{F}$  die Menge aller Geraden, also

$$\mathcal{F} = \{f(x) = c x + d \mid c, d \in \mathbb{R}\}$$

Dies nennt man **lineare Regression**.

- In diesem Fall heißt die Ausgleichsfunktion **Ausgleichsgerade** oder auch **Regressionsgerade**.

## Lagrangesche Interpolationsformel

Bereits erwähnt: Ein **Interpolationspolynom** könnte das Ausgleichsproblem exakt (mit Fehler 0) lösen.

### Definition 3.3

Für  $x_1 < x_2 < \dots < x_n$  heißt das Polynom

$$L_i(x) = \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j}$$

**$i$ -tes Lagrange-Polynom** (zu den Stützstellen  $x_1, \dots, x_n$ ).

Offensichtlich gilt:

$$L_i(x_j) = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$$

Aus den Lagrange-Polynomen bilden wir nun ein geeignetes **Interpolationspolynom**  $P(x)$  wie folgt:

$$P(x) = \sum_{j=1}^n y_j L_j(x)$$

Damit gilt dann:

$$\begin{aligned} P(x_i) &= \sum_{j=1}^n y_j L_j(x_i) \\ &= y_i \underbrace{L_i(x_i)}_{=1} \\ &= y_i \end{aligned}$$

Sowohl manuell als auch numerisch ist dieses Vorgehen aber **umständlich**.

# Vandermonde-Matrix

- Wir modellieren

$$P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_0$$

und setzen die Stützstellen  $x_1, \dots, x_n$  ein.

- Dann entsteht das LGS:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

## Definition 3.4

Für  $x_1, x_2, \dots, x_n \in \mathbb{R}$  heißt die Matrix

$$V(x_1, \dots, x_n) = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{pmatrix}$$

Vandermonde-Matrix.

## Lemma 3.5

Es gilt:

$$\det(V(x_1, \dots, x_n)) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$$

## Folgerung 3.6

Für  $x_1 < x_2 < \dots < x_n$  ist die Vandermonde-Matrix *regulär*.

Damit können wir unsere Methoden zur Lösung eines LGS einsetzen.

## Beispiel 3.7

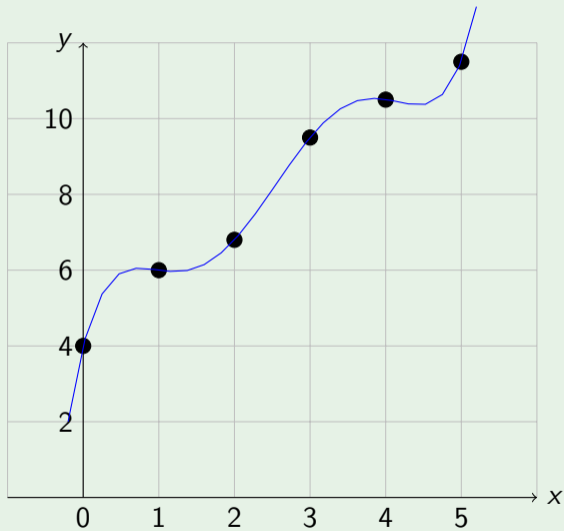
Für Beispiel 3.1 erhalten wir:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 & 16 & 32 \\ 1 & 3 & 9 & 27 & 81 & 243 \\ 1 & 4 & 16 & 64 & 256 & 1024 \\ 1 & 5 & 25 & 125 & 625 & 3125 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 6.8 \\ 9.5 \\ 10.5 \\ 11.5 \end{pmatrix}$$

Es ergibt sich (Koeffizienten gerundet):

$$P(x) = 0.1x^5 - 1.28x^4 + 5.69x^3 - 10.22x^2 + 7.71x + 4$$

## Fortsetzung Beispiel.





# Lineare Regression

- Die Grafik von Beispiel 3.7 zeigt das **Problem der Polynominterpolation**: Die gefundene Funktion ist **zu wellig**,
- zumal die  $y_i$  üblicherweise fehlerbehaftet sind.
- Polynominterpolation führt zu einer **Überanpassung (Overfitting)**, siehe Übungen).
- Besserer Ansatz: Lineare Funktionen:

$$\mathcal{F} = \{af_1 + bf_2 \mid a, b \in \mathbb{R}\}$$

mit den **Ansatzfunktionen**  $f_1(x) = x$  und  $f_2(x) = 1$ .

- Das **Fehlerfunktional** hat dann die Form:

$$E(f) = E(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

# Extremwertaufgabe

Die Minimierung der Funktion

$$E(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

ist eine **Extremwertaufgabe** im  $\mathbb{R}^2$ .

Hierfür benötigen wir Differentialrechnung im  $\mathbb{R}^n$ .

☞ **Einführung in die Analysis, Kapitel 7**

## Lineare Regression revisited

Zur Minimierung von

$$E(a, b) = \sum_{i=1}^n (y_i - (ax_i + b))^2$$

müssen die partiellen Ableitungen gleich 0 sein.

$$\frac{\partial E(a, b)}{\partial a} = -2 \sum_{i=1}^n (y_i - (ax_i + b))x_i \stackrel{!}{=} 0$$

$$\frac{\partial E(a, b)}{\partial b} = -2 \sum_{i=1}^n (y_i - (ax_i + b)) \stackrel{!}{=} 0$$

Umformung ergibt das LGS:

$$a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i$$
$$a \sum_{i=1}^n x_i + b \sum_{i=1}^n 1 = \sum_{i=1}^n y_i$$

bzw.

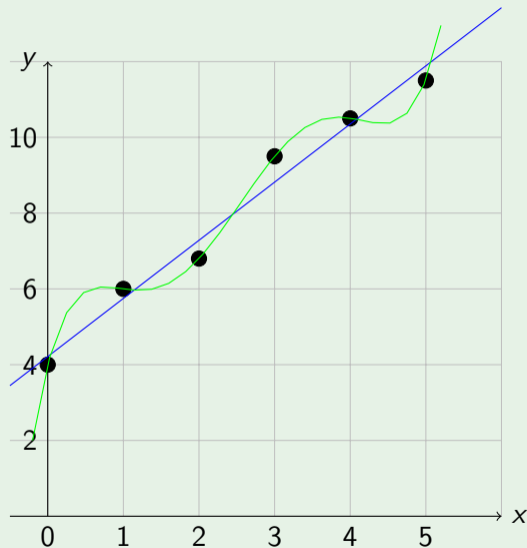
$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n y_i \end{pmatrix}$$

## Beispiel 3.8

Für Beispiel 3.1 erhalten wir:

$$\begin{pmatrix} 55 & 15 \\ 15 & 6 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 147.6 \\ 48.3 \end{pmatrix}$$

mit der Lösung  $a = 1.5343$  und  $b = 4.2143$ .



# Lineares Ausgleichsproblem

## Definition 3.9

Gegeben seien:

- Ansatzfunktionen  $f_1, \dots, f_m$ ,
- $\mathcal{F} = \{\lambda_1 f_1 + \lambda_2 f_2 + \dots + \lambda_m f_m \mid \lambda_i \in \mathbb{R} \text{ für } i = 1, \dots, m\}$ ,
- $n$  Wertepaare  $(x_i, y_i)$ ,  $i = 1, \dots, n$ .

Weiterhin sei  $f := \sum_{j=1}^m \lambda_j f_j \in \mathcal{F}$

Dann besteht das **lineare Ausgleichsproblem** darin, die Parameter  $\lambda_1, \dots, \lambda_m$  so zu bestimmen, dass das folgende **Fehlerfunktional minimal wird**:

$$E(\lambda_1, \lambda_2, \dots, \lambda_m) = \sum_{i=1}^n (y_i - f(x_i))^2 = \sum_{i=1}^n \left( y_i - \sum_{j=1}^m \lambda_j f_j(x_i) \right)^2.$$

## Fehlergleichungssystem

Im Folgenden gelte:

$$\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_m(x_2) \\ \vdots & \vdots & & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_m(x_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix}$$

### Definition 3.10

Das Gleichungssystem

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{y}$$

heißt **Fehlergleichungssystem**.

### Folgerung 3.11

$$E(\boldsymbol{\lambda}) = E(\lambda_1, \dots, \lambda_m) = \|\mathbf{y} - \mathbf{A}\boldsymbol{\lambda}\|_2^2$$

## Normalgleichungen

### Definition 3.12

Die Gleichungen

$$\frac{\partial E(\lambda_1, \dots, \lambda_m)}{\partial \lambda_i} = 0, \quad \text{für } i = 1, \dots, m$$

heißen **Normalgleichungen** des linearen Ausgleichsproblem und das zugehörige Gleichungssystem heißt **Normalgleichungssystem**.

### Satz 3.13

*Das Normalgleichungssystem lässt sich in der Form:*

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y}$$

*schreiben.*



## Beweis.

Es gilt

$$\frac{\partial E(\lambda_1, \dots, \lambda_m)}{\partial \lambda_k} = \sum_{i=1}^n 2 \left( y_i - \sum_{j=1}^m \lambda_j f_j(x_i) \right) f_k(x_i) \cdot (-1)$$

Damit folgt:

$$\begin{aligned} \frac{\partial E(\lambda_1, \dots, \lambda_m)}{\partial \lambda_k} &= 0 \\ \Leftrightarrow \sum_{i=1}^n \left( y_i - \sum_{j=1}^m \lambda_j f_j(x_i) \right) f_k(x_i) &= 0 \\ \Leftrightarrow \sum_{i=1}^n y_i f_k(x_i) &= \sum_{i=1}^n f_k(x_i) \sum_{j=1}^m \lambda_j f_j(x_i) \end{aligned}$$

## Fortsetzung Beweis.

$$\Leftrightarrow \sum_{i=1}^n y_i f_k(x_i) = \sum_{j=1}^m \lambda_j \sum_{i=1}^n f_k(x_i) f_j(x_i)$$

Mit der Notation

$$f_l(\mathbf{x}) := \begin{pmatrix} f_l(x_1) \\ f_l(x_2) \\ \vdots \\ f_l(x_n) \end{pmatrix} \in \mathbb{R}^n$$

können wir diese Gleichung schreiben als:

$$\langle f_k(\mathbf{x}), \mathbf{y} \rangle = \sum_{j=1}^m \lambda_j \langle f_k(\mathbf{x}), f_j(\mathbf{x}) \rangle$$

## Fortsetzung Beweis.

Weiterhin gilt:

$$\mathbf{A}^T = \begin{pmatrix} f_1(x_1) & f_1(x_2) & \dots & f_1(x_n) \\ f_2(x_1) & f_2(x_2) & \dots & f_2(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_m(x_1) & f_m(x_2) & \dots & f_m(x_n) \end{pmatrix}$$

Sei  $\mathbf{B} := \mathbf{A}^T \mathbf{A}$ . Dann folgt:

$$\begin{aligned} b_{ij} &= (f_i(x_1) \cdots f_i(x_n)) \begin{pmatrix} f_j(x_1) \\ \vdots \\ f_j(x_n) \end{pmatrix} \\ &= \langle f_i(\mathbf{x}), f_j(\mathbf{x}) \rangle \end{aligned}$$

## Fortsetzung Beweis.

Wir schauen uns die  $k$ -te Zeile von  $\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda}$  an. Es gilt:

$$\text{Zeile } k \text{ von } \mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \sum_{j=1}^m b_{kj} \lambda_j = \sum_{j=1}^m \lambda_j \langle f_k(\mathbf{x}), f_j(\mathbf{x}) \rangle$$

Dies ist die **rechte Seite** der Gleichung auf Folie 154 unten.

Jetzt schauen wir uns die  $k$ -te Zeile von  $\mathbf{A}^T \mathbf{y}$  an.

$$\text{Zeile } k \text{ von } \mathbf{A}^T \mathbf{y} = \sum_{i=1}^n f_k(x_i) y_i = \langle f_k(\mathbf{x}), \mathbf{y} \rangle$$

Dies ist die **linke Seite** der Gleichung auf Folie 154 unten. Also:

$$\frac{\partial E(\lambda_1, \dots, \lambda_m)}{\partial \lambda_k} = 0 \text{ für } k = 1, \dots, m \iff \mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y}$$

## Bemerkungen

- Das Normalgleichungssystem hat die  $m \times m$  Koeffizientenmatrix  $\mathbf{A}^T \mathbf{A}$ . Die **Größe** ist also durch die **Anzahl der Parameter** bestimmt.
- Üblicherweise gilt, dass die **Anzahl der Messpunkte deutlich größer** ist als die **Anzahl der Parameter**.
- Die Matrix  $\mathbf{A}^T \mathbf{A}$  ist **positiv definit**.
- Daher können wir eine Lösung mit einer **Cholesky-Zerlegung** berechnen oder eine LR-Zerlegung verwenden.
- Später besprechen wir mit der **QR-Zerlegung** noch eine alternative Zerlegung.
- Der Vektor  $\mathbf{r} := \mathbf{A}\boldsymbol{\lambda} - \mathbf{y}$  heißt **Residuumsvektor**.

## Beispiel 3.14

Wir betrachten das Normalgleichungssystem für die Regressionsgerade von Beispiel 3.1. Unsere Ansatzfunktionen sind bekanntlich  $f_1(x) = x$  und  $f_2(x) = 1$ . Damit erhalten wir

$$\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) \\ f_1(x_2) & f_2(x_2) \\ f_1(x_3) & f_2(x_3) \\ f_1(x_4) & f_2(x_4) \\ f_1(x_5) & f_2(x_5) \\ f_1(x_6) & f_2(x_6) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{pmatrix}$$

und somit

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 55 & 15 \\ 15 & 6 \end{pmatrix}$$

sowie

$$\mathbf{A}^T \mathbf{y} = \begin{pmatrix} 147.6 \\ 48.3 \end{pmatrix}.$$

### Fortsetzung Beispiel.

Damit stellt das LGS

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y}$$

genau das LGS dar, welches wir schon aus Beispiel 3.7 kennen.

Dementsprechend ist die Lösung  $\lambda_1 = 1.5343$  und  $\lambda_2 = 4.2143$ .

### Bemerkung:

- Bei einem linearen Ausgleichsproblem suchen wir nach einer **optimalen Linearkombination von Ansatzfunktionen**,
- d. h. die Kombination der Funktionen kann nur linear sein.
- **Die Ansatzfunktionen selbst können aber auch nicht linear sein.**

# Nichtlineare Ansatzfunktionen

## Beispiel 3.15

Gegeben seien die Daten

$x_i$	0	1	2	3	4
$y_i$	6	12	30	80	140

Wir wollen eine Funktion der Form

$$f(x) = \lambda_1 e^x + \lambda_2$$

finden, die die Daten bestmöglich approximiert.

Unsere Ansatzfunktionen sind also  $f_1(x) = e^x$  und  $f_2(x) = 1$ .



## Fortsetzung Beispiel.

Das Fehlergleichungssystem lautet damit

$$\begin{pmatrix} e^0 & 1 \\ e^1 & 1 \\ e^2 & 1 \\ e^3 & 1 \\ e^4 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 12 \\ 30 \\ 80 \\ 140 \end{pmatrix}.$$

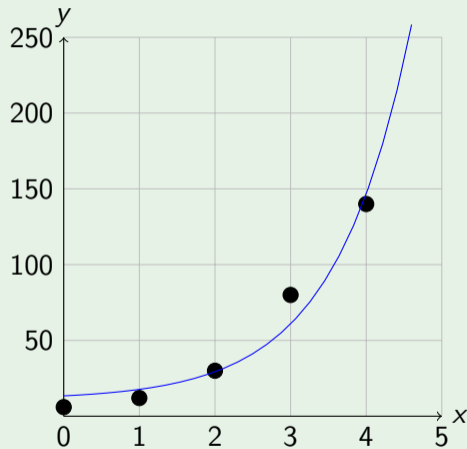
Für das Normalgleichungssystem entsteht

$$\begin{pmatrix} 3447.373987 & 85.79102488 \\ 85.79102488 & 5.0 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 9510.875023 \\ 268.0 \end{pmatrix}$$

mit Lösung

$$\lambda = \begin{pmatrix} 2.4869 \\ 10.9295 \end{pmatrix}.$$

## Fortsetzung Beispiel.



# Logarithmisch-lineare Regression

Wählt man den Ansatz

$$f(x) = \lambda_1 e^{\lambda_2 x}$$

dann spricht man von **logarithmisch linearer Regression**. Hierbei handelt es sich zunächst um ein nichtlineares Regressionsproblem.

Wir können diesen Ansatz aber linearisieren, indem wir **zum Logarithmus übergehen**:

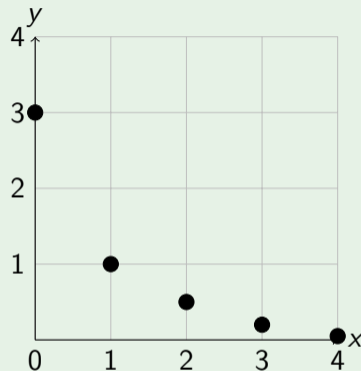
$$\log(f(x)) = \log(\lambda_1) + \lambda_2 x$$

Voraussetzung:  $\mathbf{y} > \mathbf{0}$

## Beispiel 3.16

Gegeben seien die Daten

$x_i$	0	1	2	3	4
$y_i$	3	1	0.5	0.2	0.05



Wir suchen eine Funktion der Form

$$f(x) = \lambda_1 e^{\lambda_2 x},$$

die die Daten bestmöglich approximiert.

## Fortsetzung Beispiel.

Wir gehen zum Logarithmus über:

$x_i$	0	1	2	3	4
$y_i$	3	1	0.5	0.2	0.05
$\log(y_i)$	1.099	0	-0.693	-1.609	-2.996

und berechnen die Regressionsgerade für die  $x_i$  und die  $\log(y_i)$ . Wir erhalten:

$$\lambda_1 = 1.11968, \quad \lambda_2 = -0.9798$$

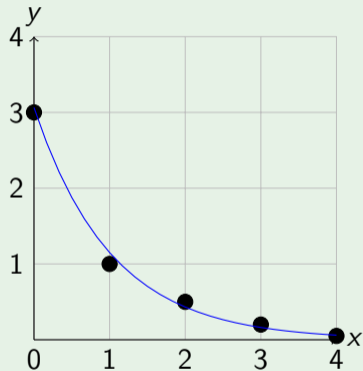
also

$$\log(f(x)) = -0.9798x + 1.11968.$$

Daraus ergibt sich

$$f(x) = e^{1.11968} e^{-0.9798x} = 3.0639 e^{-0.9798x}.$$

## Fortsetzung Beispiel.



# Multiple Regression

- Anschaulich betrachtet sprechen wir von **multipler Regression**, wenn die  $x_j$ -Werte der Wertepaare selber wieder Vektoren  $\in \mathbb{R}^d$  sind.
- Mit anderen Worten: Die approximierende Funktion  $f$  ist von der Form  $f : D \rightarrow \mathbb{R}$  mit  $D \subseteq \mathbb{R}^d$ .
- Wir haben also **nicht nur eine Einflussgröße sondern  $d$** .
- Zur Übersicht:
  - $n$  Anzahl an Wertepaaren/Stützstellen
  - $m$  Anzahl Parameter für die Ansatzfunktionen
  - $d$  Anzahl der Werte/Dimensionen für eine einzelne Stützstelle

### Beispiel 3.17

Gegeben seien die Daten

$x_{i,1}$	1	1	1	2	2	2	3	3	3
$x_{i,2}$	1	2	3	1	2	3	1	2	3
$y_i$	1.2	1.4	2.0	2.5	3.2	3.4	3.7	3.9	4.3

Wir haben also als Stützstellen ein **äquidistantes Gitter der Größe  $3 \times 3$** .

Wir vermuten einen funktionalen Zusammenhang der Form


$$f(x_1, x_2) = \lambda_0 + \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_1^2 + \lambda_4 x_2^2 + \lambda_5 x_1 x_2.$$



## Fortsetzung Beispiel.

Allgemeine Form des Fehlergleichungssystems:

$$\begin{pmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,1}^2 & x_{1,2}^2 & x_{1,1}x_{1,2} \\ 1 & x_{2,1} & x_{2,2} & x_{2,1}^2 & x_{2,2}^2 & x_{2,1}x_{2,2} \\ \vdots & & & & & \vdots \\ 1 & x_{9,1} & x_{9,2} & x_{9,1}^2 & x_{9,2}^2 & x_{9,1}x_{9,2} \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \vdots \\ \lambda_5 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_9 \end{pmatrix}$$

Konkretes Fehlergleichungssystems: Tafel 

## Fortsetzung Beispiel.

Lösung:

$$\lambda_0 = -1.44$$

$$\lambda_1 = 2.45$$

$$\lambda_2 = 0.42$$

$$\lambda_3 = -0.28$$

$$\lambda_4 = 0.016$$

$$\lambda_5 = -0.05$$

# Orthogonale Matrix

- Ausgleichsprobleme sind häufig **schlecht konditioniert**.
- Matrix des Normalgleichungssystems kann **“nahezu singulär”** sein.
- Spezielle Matrixzerlegung für höhere numerische Stabilität: **QR-Zerlegung**

## Definition 3.18

Es sei  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  eine quadratische Matrix. Gilt

$$\mathbf{Q}^{-1} = \mathbf{Q}^T,$$

dann heißt  $\mathbf{Q}$  **orthogonal**.

## Beispiele orthogonaler Matrizen

### Beispiel 3.19

(i) Die Matrix

$$\mathbf{Q} = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

ist orthogonal. Sie beschreibt als lineare Abbildung eine **Drehung um  $45^\circ$** .

(ii) Die Verallgemeinerung der Matrix aus (i) ist

$$\mathbf{R} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}.$$

Sie beschreibt eine **Drehung um den Winkel  $\alpha$** .

(iii) **Permutationsmatrizen** sind orthogonal.

# Eigenschaften orthogonaler Matrizen

## Satz 3.20

Es sei  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  eine orthogonale Matrix. Dann gilt:

- (i) Die Spaltenvektoren  $\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \dots, \mathbf{q}^{(n)}$  bilden eine *orthonormale Basis des  $\mathbb{R}^n$* , d. h.
  - ▶ die Vektoren  $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(n)}$  sind linear unabhängig,
  - ▶  $\text{Span}(\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(n)}) = \mathbb{R}^n$ ,
  - ▶  $\|\mathbf{q}^{(i)}\| = 1$  für  $i = 1, \dots, n$  und
  - ▶  $\langle \mathbf{q}^{(i)}, \mathbf{q}^{(j)} \rangle = 0$  für  $i \neq j$ .
- (ii)  $\mathbf{Q}^T$  ist auch orthogonal.
- (iii)  $\mathbf{Q}$  entspricht einer *längentreuen linearen Abbildung*, d. h.  $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$  für alle  $\mathbf{x} \in \mathbb{R}^n$ .
- (iv)  $\det(\mathbf{Q}) = \pm 1$
- (v) Das *Produkt*  $\mathbf{Q} \cdot \mathbf{R}$  zweier orthogonaler Matrizen  $\mathbf{Q}$  und  $\mathbf{R}$  ist wieder *orthogonal*.

# QR-Zerlegung

## Definition 3.21

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times m}$ . Gilt

$$\mathbf{A} = \mathbf{QR}$$

- (i) für eine Matrix  $\mathbf{Q} \in \mathbb{R}^{n \times m}$  mit orthonormierten Spalten und eine obere Dreiecksmatrix  $\mathbf{R} \in \mathbb{R}^{m \times m}$ , dann heißt diese Zerlegung **reduzierte QR-Zerlegung**.
- (ii) für eine Orthogonalmatrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  und eine Matrix  $\mathbf{R} \in \mathbb{R}^{n \times m}$  mit oberer Dreiecksgestalt, dann heißt diese Zerlegung **volle QR-Zerlegung**.

## Bemerkungen zu Definition 3.21:

(i)  $\mathbf{Q}$  hat orthonormierte Spalten heißt

$$\langle \mathbf{q}^{(i)}, \mathbf{q}^{(j)} \rangle = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$$

und  $1 \leq i, j \leq m$  und  $\mathbf{q}^{(i)}, \mathbf{q}^{(j)} \in \mathbb{R}^n$ .

(ii) Obere Dreiecksgestalt bedeutet  $r_{i,j} = 0$  für  $i > j$ .

$$\mathbf{R} = \begin{pmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ 0 & 0 & \ddots & \vdots \\ \vdots & & \ddots & * \\ & & & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & & 0 \end{pmatrix} \in \mathbb{R}^{n \times m}$$

## Eindeutigkeit der reduzierten QR-Zerlegung

### Satz 3.22

Sei  $\mathbf{A} \in \mathbb{R}^{n \times m}$  mit  $r(\mathbf{A}) = m \leq n$  und seien

$$\mathbf{Q}_1 \mathbf{R}_1 = \mathbf{A} = \mathbf{Q}_2 \mathbf{R}_2$$

zwei reduzierte QR-Zerlegungen von  $\mathbf{A}$ .

Dann existiert eine Diagonalmatrix  $\mathbf{D} \in \mathbb{R}^{m \times m}$  mit  $d_{ii} \pm 1$  für alle  $i = 1, \dots, m$ , so dass gilt:

$$\mathbf{Q}_1 = \mathbf{Q}_2 \mathbf{D} \quad \text{und} \quad \mathbf{R}_2 = \mathbf{D} \mathbf{R}_1$$

### Folgerung 3.23

Es existiert genau eine QR-Zerlegung mit  $r_{ii} > 0$  für alle  $i = 1, \dots, m$ .



## QR-Zerlegung und Normalgleichungssystem

Die Lösung eines linearen Ausgleichsproblems ergibt sich durch Lösung des Normalgleichungssystems

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y}.$$

Sei nun  $\mathbf{A} = \mathbf{QR}$  eine reduzierte QR-Zerlegung. Wegen  $\mathbf{Q}^T \mathbf{Q} = \mathbf{E}$  folgt

$$\mathbf{R}^T \mathbf{R} \boldsymbol{\lambda} = \mathbf{R}^T \mathbf{Q}^T \mathbf{Q} \mathbf{R} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y} = \mathbf{R}^T \mathbf{Q}^T \mathbf{y}.$$

Da  $\mathbf{R}$  und damit auch  $\mathbf{R}^T$  regulär ist, ergibt sich

$$\mathbf{R} \boldsymbol{\lambda} = \mathbf{Q}^T \mathbf{y}.$$

Dieses Gleichungssystem können wir durch Rückwärtssubstitution lösen.

# Lösung von $\mathbf{A}\boldsymbol{\lambda} \approx \mathbf{y}$ mittels QR-Zerlegung

## Algorithmus 3.24

- 1 Bestimme eine reduzierte QR-Zerlegung  $\mathbf{A} = \mathbf{QR}$ .
- 2 Setze  $\mathbf{c} := \mathbf{Q}^T \mathbf{y}$ .
- 3 Löse  $\mathbf{R}\boldsymbol{\lambda} = \mathbf{c}$  durch Rückwärtssubstitution.

### Bemerkung:

- Für  $n = m$  besteht kein Unterschied zwischen einer reduzierten und einer vollen QR-Zerlegung.
- In diesem Fall können wir eine QR-Zerlegung auch zur Lösung eines LGS verwenden.
- Dies wird aus Gründen der numerischen Stabilität auch verwendet.

# Gram-Schmidt-Orthogonalisierung

Wir schauen uns  $\mathbf{A} = \mathbf{QR}$  näher an.

$$\left( \begin{array}{c|c|c|c} \mathbf{a}^{(1)} & \mathbf{a}^{(2)} & \dots & \mathbf{a}^{(m)} \\ \hline & & & \end{array} \right) = \left( \begin{array}{c|c|c|c} \mathbf{q}^{(1)} & \mathbf{q}^{(2)} & \dots & \mathbf{q}^{(m)} \\ \hline & & & \end{array} \right) \begin{pmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,m} \\ & r_{2,2} & \dots & r_{2,m} \\ & & \ddots & \vdots \\ & & & r_{m,m} \end{pmatrix}$$

Dies ist äquivalent zu:

$$\begin{aligned} \mathbf{a}^{(1)} &= r_{1,1} \mathbf{q}^{(1)} \\ \mathbf{a}^{(2)} &= r_{1,2} \mathbf{q}^{(1)} + r_{2,2} \mathbf{q}^{(2)} \\ &\vdots \\ \mathbf{a}^{(m)} &= r_{1,m} \mathbf{q}^{(1)} + r_{2,m} \mathbf{q}^{(2)} + \dots + r_{m,m} \mathbf{q}^{(m)} \end{aligned}$$

In Kurzform:

$$\mathbf{a}^{(j)} = \sum_{i=1}^j r_{i,j} \mathbf{q}^{(i)}$$

Hieraus können wir nun die  $\mathbf{q}^{(j)}$  und die  $r_{i,j}$  induktiv herleiten.

Für  $j = 1$  ergibt sich:

$$r_{1,1} = \pm \|\mathbf{a}^{(1)}\| \quad \text{und} \quad \mathbf{q}^{(1)} = \frac{1}{r_{1,1}} \mathbf{a}^{(1)}$$

Sei nun  $j \geq 2$ , sowie  $\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(j-1)}$  ein orthonormiertes System von Vektoren, so dass die ersten  $j - 1$  Gleichungen mit bekannten Zahlen  $r_{i,k}$ ,  $1 \leq i \leq k \leq j - 1$  erfüllt sind.

Wir machen den Ansatz:

$$\mathbf{q}^{(j)} = \frac{1}{r_{j,j}} \left( \mathbf{a}^{(j)} - \sum_{i=1}^{j-1} r_{i,j} \mathbf{q}^{(i)} \right)$$

Da  $\mathbf{q}^{(j)}$  normiert sein soll, folgt

$$r_{j,j} = \left\| \mathbf{a}^{(j)} - \sum_{i=1}^{j-1} r_{i,j} \mathbf{q}^{(i)} \right\|$$

Die noch unbekanntenen  $r_{i,j}$  für  $i < j$  ergeben sich aus der geforderdeten Orthogonalität:

$$0 = \langle \mathbf{q}^{(i)}, \mathbf{q}^{(j)} \rangle = \frac{1}{r_{j,j}} \left( \langle \mathbf{q}^{(i)}, \mathbf{a}^{(j)} \rangle - r_{i,j} \right)$$

für alle  $i = 1, \dots, j - 1$ . Damit folgt

$$r_{i,j} = \langle \mathbf{q}^{(i)}, \mathbf{a}^{(j)} \rangle$$

# Gram-Schmidt-Verfahren

## Algorithmus 3.25

$$r_{1,1} := \|\mathbf{a}^{(1)}\|$$

$$\mathbf{q}^{(1)} := \frac{1}{r_{1,1}}\mathbf{a}^{(1)}$$

**for**  $j := 2$  **to**  $m$  **do**

$$\mathbf{q}^{(j)} := \mathbf{a}^{(j)}$$

**for**  $i := 1$  **to**  $j - 1$  **do**

$$r_{i,j} := \langle \mathbf{q}^{(i)}, \mathbf{a}^{(j)} \rangle$$

$$\mathbf{q}^{(j)} := \mathbf{q}^{(j)} - r_{i,j}\mathbf{q}^{(i)}$$

**end**

$$r_{j,j} := \|\mathbf{q}^{(j)}\|$$

$$\mathbf{q}^{(j)} := \frac{1}{r_{j,j}}\mathbf{q}^{(j)}$$

**end**

## Existenz der QR-Zerlegung

- In Folgerung 3.23 können wir aus Satz 3.22 nur die **Eindeutigkeit der QR-Zerlegung** herleiten, aber nicht deren Existenz.
- Das Gram-Schmidt-Verfahren liefert aber eine **Konstruktion für eine QR-Zerlegung**.
- Wenn wir zeigen können, dass dieser Algorithmus immer durchführbar ist, haben wir auch den **Existenzbeweis**.
- Das Gram-Schmidt-Verfahren kann nur durch  $r_{j,j} = 0$  scheitern.
- Wäre aber  $r_{j,j} = 0$ , dann würde

$$\mathbf{a}^{(j)} \in \text{Span}\{\mathbf{q}^{(1)}, \dots, \mathbf{q}^{(j-1)}\} = \text{Span}\{\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(j-1)}\}$$

gelten.

- Dies ist ein Widerspruch zur linearen Unabhängigkeit der Spaltenvektoren von  $\mathbf{A} \in \mathbb{R}^{n \times m}$ , denn n. V. gilt  $r(\mathbf{A}) = m$ .

# Modifiziertes Gram-Schmidt-Verfahren

- Wenn die Spalten von  $\mathbf{A}$  nahezu linear abhängig sind, ist das Gram-Schmidt-Verfahren immer noch **numerisch instabil**. Die berechneten Vektoren  $\mathbf{q}^{(j)}$  sind also nicht orthogonal.
- Besser wird die Stabilität, wenn wir die Berechnung der  $r_{i,j}$  ersetzen durch

$$r_{i,j} = \langle \mathbf{q}^{(i)}, \mathbf{q}^{(j)} \rangle.$$

- Diese zum klassischen Gram-Schmidt-Verfahren **äquivalente Methode** heißt **modifiziertes Gram-Schmidt-Verfahren**.



## Algorithmus 3.26

$$r_{1,1} := \|\mathbf{a}^{(1)}\|$$

$$\mathbf{q}^{(1)} := \frac{1}{r_{1,1}}\mathbf{a}^{(1)}$$

**for**  $j := 2$  **to**  $m$  **do**

$$\mathbf{q}^{(j)} := \mathbf{a}^{(j)}$$

**for**  $i := 1$  **to**  $j - 1$  **do**

$$r_{i,j} := \langle \mathbf{q}^{(i)}, \mathbf{q}^{(j)} \rangle$$

$$\mathbf{q}^{(j)} := \mathbf{q}^{(j)} - r_{i,j}\mathbf{q}^{(i)}$$

**end**

$$r_{j,j} := \|\mathbf{q}^{(j)}\|$$

$$\mathbf{q}^{(j)} := \frac{1}{r_{j,j}}\mathbf{q}^{(j)}$$

**end**

## Beispiel 3.27

Wir lösen Beispiel 3.1 mittels Algorithmus 3.24.

**Schritt 1:** Wir berechnen eine QR-Zerlegung mit dem Gram-Schmidt-Verfahren. Es gilt

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{pmatrix}$$

und damit  $\|\mathbf{a}^{(1)}\| = \sqrt{55} = r_{1,1}$ , sowie

$$\mathbf{q}^{(1)} = \frac{1}{\sqrt{55}} \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}.$$

## Fortsetzung Beispiel.

Daraus ergibt sich

$$r_{1,2} = \langle \mathbf{q}^{(1)}, \mathbf{a}^{(2)} \rangle = \frac{15}{\sqrt{55}}.$$

und für den Vektor  $\mathbf{q}^{(2)}$  vor der Normierung

$$\mathbf{q}^{(2)} = \mathbf{a}^{(2)} - r_{1,2}\mathbf{q}^{(1)} = \frac{1}{11} \begin{pmatrix} 11 \\ 8 \\ 5 \\ 2 \\ -1 \\ -4 \end{pmatrix}.$$

## Fortsetzung Beispiel.

Die Normierung von  $\mathbf{q}^{(2)}$  liefert  $r_{2,2} = \frac{\sqrt{231}}{11}$  und

$$\mathbf{q}^{(2)} = \frac{1}{\sqrt{231}} \begin{pmatrix} 11 \\ 8 \\ 5 \\ 2 \\ -1 \\ -4 \end{pmatrix}.$$

**Schritt 2:** Wir berechnen  $\mathbf{c} = \mathbf{Q}^T \mathbf{y}$ .

$$\mathbf{c} = \mathbf{Q}^T \mathbf{y} = \begin{pmatrix} \langle \mathbf{q}^{(1)}, \mathbf{y} \rangle \\ \langle \mathbf{q}^{(2)}, \mathbf{y} \rangle \end{pmatrix} = \begin{pmatrix} \frac{147.6}{\sqrt{55}} \\ \frac{88.5}{\sqrt{231}} \end{pmatrix}$$

## Fortsetzung Beispiel.

**Schritt 3:** Wir lösen  $\mathbf{R}\boldsymbol{\lambda} = \mathbf{c}$ .

$$\begin{pmatrix} \sqrt{55} & \frac{15}{\sqrt{55}} \\ 0 & \frac{\sqrt{231}}{11} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \frac{147.6}{\sqrt{55}} \\ \frac{88.5}{\sqrt{231}} \end{pmatrix}$$

Daraus ergibt sich

$$\begin{aligned} \lambda_2 &= \frac{11 \cdot 88.5}{231} = 4.2143 \\ \lambda_1 &= \frac{147.6 - 15\lambda_2}{55} = 1.5343 \end{aligned}$$

# Householder-Matrix


## Definition 3.28

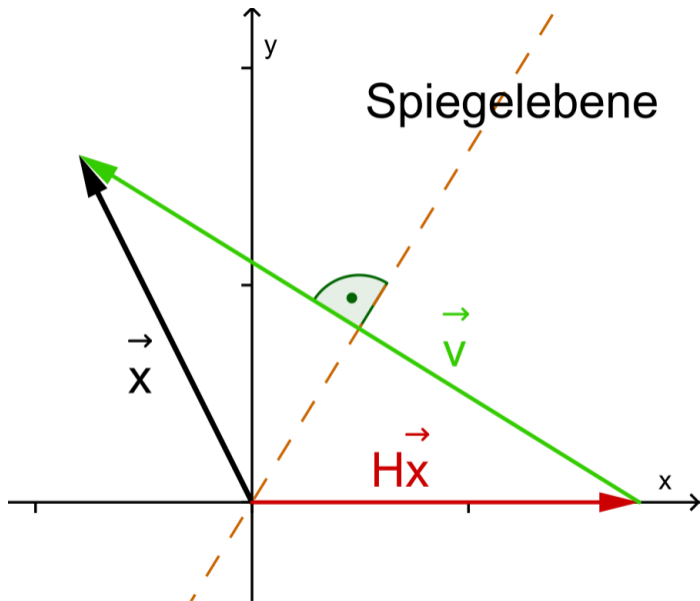
Es sei  $\mathbf{v} \in \mathbb{R}^n$  mit  $\mathbf{v} \neq \mathbf{0}$ .

Dann heißt die Matrix

$$\mathbf{H} = \mathbf{E} - \frac{2}{\mathbf{v}^T \mathbf{v}} \mathbf{v} \mathbf{v}^T$$

**Householder-Matrix** zum Vektor  $\mathbf{v}$ .

- Ist  $\mathbf{v}$  auf die Länge 1 normiert, ergibt sich  $\mathbf{H} = \mathbf{E} - 2\mathbf{v}\mathbf{v}^T$ .
- Householder-Matrizen beschreiben **Spiegelungen** an einer durch den Ursprung gehenden Spiegelebene mit Normalenvektor  $\mathbf{v}$ . 



# Eigenschaften von Householder-Matrizen

- 1 Jede Householder-Matrix  $\mathbf{H}$  ist **orthogonal**.
- 2 Householder-Matrizen sind **symmetrisch**, es gilt also

$$\mathbf{H} = \mathbf{H}^T.$$

- 3 Aus (1) und (2) folgt

$$\mathbf{H} = \mathbf{H}^{-1}.$$

- 4 Die Multiplikation von beliebig vielen Householder-Matrizen liefert wieder eine orthogonale Matrix.



### Beispiel 3.29

Wir berechnen die Householder-Matrix  $\mathbf{H}$  zum Vektor

$$\mathbf{v} = \begin{pmatrix} 2 \\ -1 \\ 3 \end{pmatrix}$$

und überprüfen die genannten Eigenschaften.



# Schritt zur Erzeugung einer oberen Dreiecksmatrix

## Lemma 3.30

Es sei

- $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine quadratische Matrix,
- $k \in \{1, \dots, n\}$ ,
- $\mathbf{e}^k \in \mathbb{R}^n$  der  $k$ -te Einheitsvektor,
- $\mathbf{a}^k = (0, \dots, 0, a_{k,k}, \dots, a_{n,k})^T \in \mathbb{R}^n$  der  $k$ -te Spaltenvektor von  $\mathbf{A}$ , wobei die Elemente oberhalb der Diagonalen durch Null ersetzt wurden,
- $\mathbf{v}_k = \mathbf{a}^k + \|\mathbf{a}^k\|_2 \mathbf{e}^k \in \mathbb{R}^n$  und
- $\mathbf{H}_k = \mathbf{E} - \frac{2}{\mathbf{v}_k^T \mathbf{v}_k} \mathbf{v}_k \mathbf{v}_k^T \in \mathbb{R}^{n \times n}$ .

Dann besitzt die Matrix

$$\mathbf{B} = \mathbf{H}_k \mathbf{A}$$

unterhalb der Diagonalen in Spalte  $k$  nur Nulleinträge.

## Herleitung einer QR-Zerlegung

- Wir multiplizieren eine Matrix  $\mathbf{A}$  mit Householder-Matrizen  $\mathbf{H}_k$ , so dass wir jeweils in Spalte  $k$  unterhalb der Diagonalen Nullen erhalten.
- Wiederholtes Anwenden mit  $k = 1, \dots, n - 1$  ergibt eine obere Dreiecksmatrix  $\mathbf{R}$  mit

$$\mathbf{R} = \mathbf{H}_{n-1} \cdot \dots \cdot \mathbf{H}_1 \cdot \mathbf{A}.$$

- Wegen  $\mathbf{H}_k^{-1} = \mathbf{H}_k$  folgt

$$\mathbf{A} = \mathbf{H}_1 \cdot \dots \cdot \mathbf{H}_{n-1} \cdot \mathbf{R}.$$

- Die Matrix

$$\mathbf{Q} = \mathbf{H}_1 \cdot \dots \cdot \mathbf{H}_{n-1}$$

ist eine orthogonale Matrix.

# Algorithmus zur QR-Zerlegung

## Algorithmus 3.31

Gegeben sei eine Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ .

Sei  $\mathbf{B}_1 = \mathbf{A}$  mit den Matrixeinträgen  $b_{i,j}^1$ .

Für  $k = 1, \dots, n-1$  führe die folgenden Schritte aus:

- 1 Setze  $\mathbf{a}_k = (0, \dots, 0, b_{k,k}^k, \dots, b_{n,k}^k)^T$  und damit

$$\mathbf{v}_k = \mathbf{a}_k + \|\mathbf{a}_k\|_2 \cdot \mathbf{e}_k.$$

- 2 Berechne die Householder-Matrix

$$\mathbf{H}_k = \mathbf{E} - \frac{2}{\mathbf{v}_k^T \mathbf{v}_k} \mathbf{v}_k \mathbf{v}_k^T.$$

### Fortsetzung Algorithmus.

- 3 Berechne  $\mathbf{B}_{k+1} = \mathbf{H}_k \cdot \mathbf{B}_k$  mit den Matrixeinträgen  $b_{i,j}^{k+1}$ .

Die QR-Zerlegung von  $\mathbf{A}$  ist dann gegeben durch

$$\mathbf{Q} = \mathbf{H}_1 \cdot \dots \cdot \mathbf{H}_{n-1}$$

und  $\mathbf{R} = \mathbf{B}_n$ .



## Beispiel zur QR-Zerlegung

### Beispiel 3.32

Es sei

$$\mathbf{A} = \begin{pmatrix} 2 & 4 & -9 \\ -2 & -10 & 3 \\ 1 & -1 & 6 \end{pmatrix}.$$

Es gilt  $\mathbf{v}_1 = (5, -2, 1)^T$  (warum?) und damit erhalten wir

$$\mathbf{H}_1 \cdot \mathbf{B}_1 = \begin{pmatrix} -\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & \frac{11}{15} & \frac{2}{15} \\ -\frac{1}{3} & \frac{2}{15} & \frac{14}{15} \end{pmatrix} \cdot \begin{pmatrix} 2 & 4 & -9 \\ -2 & -10 & 3 \\ 1 & -1 & 6 \end{pmatrix} = \begin{pmatrix} -3 & -9 & 6 \\ 0 & -\frac{24}{5} & -3 \\ 0 & -\frac{18}{5} & 9 \end{pmatrix} = \mathbf{B}_2.$$

## Fortsetzung Beispiel.

Wir nutzen  $\mathbf{v}_2 = (0, \frac{6}{5}, -\frac{18}{5})^T$  und erhalten


$$\mathbf{H}_2 \cdot \mathbf{B}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & \frac{3}{5} & -\frac{4}{5} \end{pmatrix} \begin{pmatrix} -3 & -9 & 6 \\ 0 & -\frac{24}{5} & -3 \\ 0 & -\frac{18}{5} & 9 \end{pmatrix} = \begin{pmatrix} -3 & -9 & 6 \\ 0 & -6 & 3 \\ 0 & 0 & -9 \end{pmatrix} = \mathbf{B}_3 = \mathbf{R}.$$

Die orthogonale Matrix  $\mathbf{Q}$  ergibt sich durch

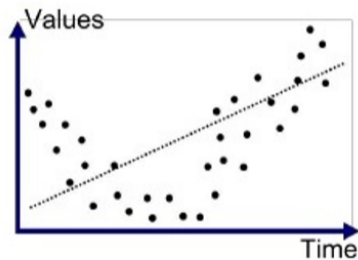
$$\mathbf{Q} = \mathbf{H}_1 \cdot \mathbf{H}_2 = \begin{pmatrix} -\frac{2}{3} & \frac{2}{3} & -\frac{1}{3} \\ \frac{2}{3} & \frac{11}{15} & \frac{2}{15} \\ -\frac{1}{3} & \frac{2}{15} & \frac{14}{15} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{4}{5} & \frac{3}{5} \\ 0 & \frac{3}{5} & -\frac{4}{5} \end{pmatrix} = \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{2}{3} \end{pmatrix}.$$



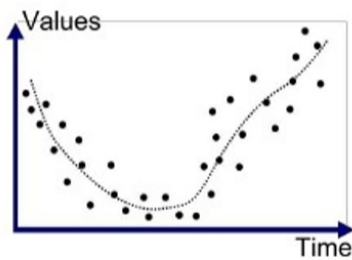
# Diskussion QR-Zerlegung

- Komplexität  $O(n^3)$
- Diese Komplexität ist **nur mit einer effizienten Matrixmultiplikation erreichbar**, die aber existiert (Übungsaufgabe .
- Eine QR-Zerlung kann **für jede beliebige Matrix durchgeführt werden**, es ist keine Pivotisierung notwendig.
- Die Verwendung einer QR-Zerlegung zur Lösung eines LGS ist i. A. numerisch stabiler als eine LR-Zerlegung.

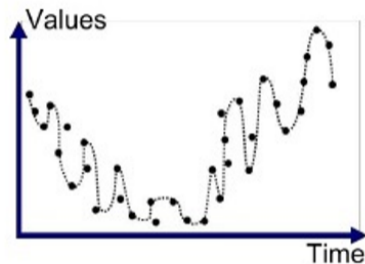
# Underfitting und Overfitting



Underfitted



Good Fit/Robust



Overfitted

# Prognosemodelle bewerten

- Geht es um das Finden von Ansatzfunktionen, so sollte das **Prognosemodell so genau aber auch so einfach wie möglich** sein.
- Hat man ein sparsames Modell, also mit wenigen Einflussvariablen, ermittelt, so benötigt man auch für die Prognose nur diese wenigen Informationen.
- Als Maßzahl für die Güte der Anpassung einer Funktion an die Daten haben wir das Fehlerfunktional kennengelernt. Dieses Maß ist jedoch für die Suche nach einem sparsamen Modell ungeeignet, da es nicht kleiner wird, wenn ein Merkmal hinzugefügt wird.
- Andererseits führen komplexe Modelle dazu, dass  $E(f)$  immer kleiner wird, aber **wo beginnt die Überanpassung?**
- Besser geeignet ist das **Akaike Informationskriterium (AIC)**.

# Akaike Informationsindex

## Definition 3.33

Gegeben sei:

- $p$  die Anzahl der Koeffizienten und damit der Variablen im Modell und
- $n$  Wertepaare  $(x_i, y_i), i = 1, \dots, n$ .

$$AIC = \log \left( \frac{1}{n-p} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \right) + 2 * \frac{p}{n}$$

Dabei ist  $\hat{y}_i$  der für  $x_i$  prognostizierte Wert.

**Bemerkung:** In der Literatur findet man verschiedene Formulierungen des AIC's, alle folgen jedoch der allgemeinen Formel:

$$AIC = 2p - 2 \log(E(f)).$$

Auf die Theorie, die der Formel zugrunde liegt, wird hier nicht eingegangen. Man kann aber erkennen, dass

- eine **gute Anpassung der Funktion an die Daten** zu einer kleinen Summe  $E(f)$  führt, und damit das Gesamtergebnis reduziert;
- eine **große Anzahl von Beobachtungen**  $n$  das Ergebnis ebenfalls reduziert;
- jedoch eine große Anzahl von Parametern wie z.B. Koeffizienten von Variablen oder Polynomtermen das Ergebnis vergrößert, die Anzahl  $p$  wird also **strafend** berücksichtigt.

Das heißt:

- wenn der AIC-Wert kleiner wird, wenn eine Variable, ein Polynomterm o. ä. hinzugefügt wird, kann man von einer Verbesserung des Modells ausgehen.
- tritt keine Reduktion ein, so wird die Vergrößerung von  $p$  nicht durch die Reduktion der Summe der quadrierten Abweichungen kompensiert, der Term verbessert das Modell nicht.

# Parameterauswahl

Zur Auswahl des nach diesem Kriterium besten Modells geht man wie folgt vor:

- 1 Man erstellt ein Modell, das die Funktionen enthält, die man für sinnvoll erachtet, am besten nach Visualisierung der Zusammenhänge anhand einer Grafik.
- 2 Man entfernt jeweils einen Term (ein Merkmal oder einen Polynomterm o. ä.) und vergleicht die AIC-Werte.
- 3 Hat sich der AIC-Wert verringert, so wird der Term aus dem Modell endgültig entfernt.

Beispiel: in der Übung

# Zusammenfassung

- **Lineares Ausgleichsproblem:** Finde eine Linearkombination  $f = \sum_{j=1}^m \lambda_j f_j$  von Funktionen, so dass das **Fehlerfunktional**

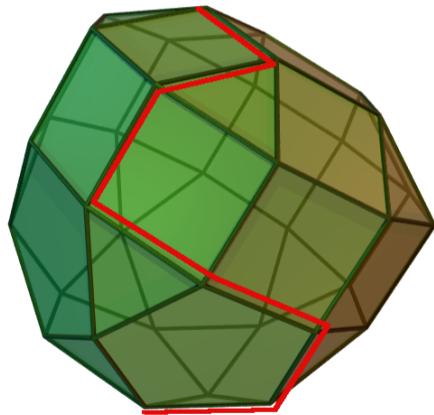
$$E(f) = \sum_{i=1}^n (y_i - f(x_i))^2$$

minimiert wird.

- Interpolation liefert i. d. R. keine befriedigende Lösung.
- Lösung des Ausgleichsproblems auf Basis des **Normalgleichungssystems** mittels Cholesky-Zerlegung.
- Numerisch stabiler: **QR-Zerlegung**
- Berechnung einer QR-Zerlegung mit dem **(modifizierten) Gram-Schmidt-Verfahren**
- Lösung für andere Fehlerfunktionale mittels linearer Programmierung

# Kapitel 4

Lineare Optimierung mit Anwendungen





# Inhalt

## 4 Lineare Optimierung mit Anwendungen

- Lineares Programm
- Grafische Lösung linearer Programme
- Normalform
- Geometrie linearer Programme
- Basislösungen
- Simplexalgorithmus
- Anwendungen

# Lineares Programm

## Definition 4.1

Es seien  $b_i, c_j, a_{ij} \in \mathbb{R}$  für  $1 \leq i \leq m$  und  $1 \leq j \leq n$ .

Ein **lineares Programm (LP)** ist die Aufgabe, eine **lineare Zielfunktion**

$$z = F(x_1, \dots, x_n) = c_1 x_1 + \dots + c_n x_n$$

für  $x_j \in \mathbb{R}$  zu **maximieren** oder zu **minimieren** unter Beachtung von **linearen Nebenbedingungen** der Form

$$a_{i,1}x_1 + \dots + a_{i,n}x_n \leq b_i \quad (i = 1, \dots, m_1)$$

$$a_{i,1}x_1 + \dots + a_{i,n}x_n = b_i \quad (i = m_1 + 1, \dots, m_2)$$

$$a_{i,1}x_1 + \dots + a_{i,n}x_n \geq b_i \quad (i = m_2 + 1, \dots, m)$$

und meist auch von **Vorzeichenbedingungen**  $x_j \geq 0$  für einige oder alle  $j = 1, \dots, n$ .

## Beispiel 4.2

Ein Eisverkäufer stellt stündlich bis zu 10 kg Eis der Sorten  $A$  bzw.  $B$  her.

	$A$	$B$
Verkaufspreis	80 EUR/kg	65 EUR/kg
Kosten	50 EUR/kg	40 EUR/kg
Energieaufwand	5 kWh/kg	2 kWh/kg
absetzbar	6 kg	9 kg

Es stehen höchstens 30 kWh stündlich zur Verfügung.

**Entscheidungsvariablen** seien die stündlich herzustellenden Mengen  $x_1$  kg bzw.  $x_2$  kg.

Zu maximieren sei die Differenz aus Preis und Kosten.

## Modellierung für Beispiel 4.2

Maximiere

$$z = F(x_1, x_2) = 80x_1 + 65x_2 - 50x_1 - 40x_2 = 30x_1 + 25x_2$$

unter den Nebenbedingungen

$$\begin{array}{rclcl} x_1 & + & x_2 & \leq & 10 \\ 5x_1 & + & 2x_2 & \leq & 30 \\ x_1 & & & \leq & 6 \\ & & & & x_2 \leq 9 \end{array}$$

und Vorzeichenbedingungen  $x_1, x_2 \geq 0$ .

## Zulässige und optimale Lösung

### Definition 4.3

Ein Punkt oder Vektor  $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ , der alle Neben- und Vorzeichenbedingungen erfüllt, heißt **zulässige Lösung** eines LP.

Eine zulässige Lösung  $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$  heißt **optimale Lösung** eines LP, wenn es keine zulässige Lösung  $\mathbf{x}$  mit besserem Zielfunktionswert als  $F(\mathbf{x}^*)$  gibt.

Mit  $\mathcal{X}_{LP}$  bezeichnen wir die **Menge der zulässigen Lösungen** des linearen Programms  $LP$  und mit  $\mathcal{X}_{LP}^*$  die **Menge der optimalen Lösungen** von  $LP$ .

**Bemerkung:** Wenn aus dem Kontext heraus das lineare Programm eindeutig ist, schreiben wir auch  $\mathcal{X}$  und  $\mathcal{X}^*$  statt  $\mathcal{X}_{LP}$  bzw.  $\mathcal{X}_{LP}^*$ .

# Grafische Lösung linearer Programme

Wir betrachten zwei Entscheidungsvariablen  $x_1$  und  $x_2$ :

$$a_1x_1 + a_2x_2 = b$$

ist die Gleichung einer Geraden im  $\mathbb{R}^2$ .

$$a_1x_1 + a_2x_2 \leq b \text{ und } a_1x_1 + a_2x_2 \geq b$$

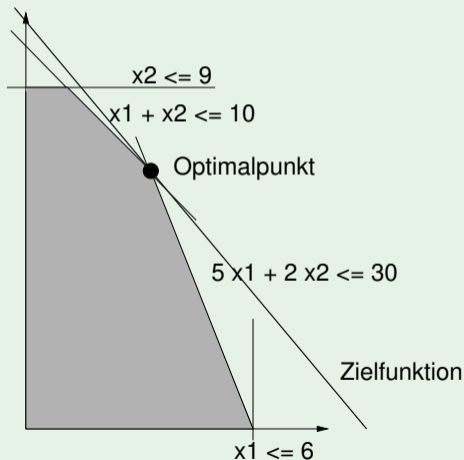
beschreiben jeweils eine Halbebene mit der Geraden  $a_1x_1 + a_2x_2 = b$  als Rand.

Auch  $x_1 \geq 0$  und  $x_2 \geq 0$  stellen Halbebenen dar.

Der zulässige Bereich ist der **Durchschnitt endlich vieler Halbebenen** und somit ein sogenanntes **konvexes Polyeder** mit endlich vielen Eckpunkten.

# Grafische Lösung zu Beispiel 4.2

## Beispiel 4.4



Die Zielfunktion  $z = 30x_1 + 25x_2$  wird ebenfalls durch eine Gerade dargestellt.

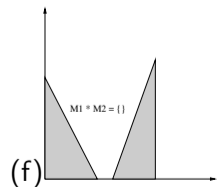
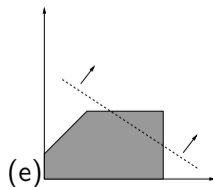
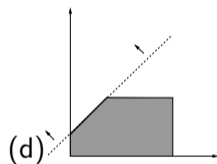
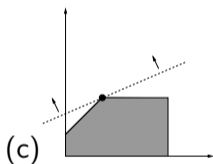
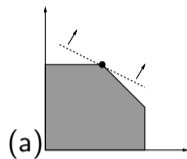
Wachsendes  $z$  bedeutet eine Verschiebung nach rechts oben.

Verschiebe nach oben, solange die Gerade durch  $\mathcal{X}$  verläuft.

Optimale Lösung im Schnittpunkt der Geraden  $x_1 + x_2 = 10$  und  $5x_1 + 2x_2 = 30$ , also  $\mathbf{x}^* = \left(\frac{10}{3}, \frac{20}{3}\right)$  mit  $z^* = \frac{800}{3}$ .

# Mögliche Situationen bei grafischer Lösung

- (a) beschränktes  $\mathcal{X}$ , eindeutige optimale Lösung
- (b) beschränktes  $\mathcal{X}$ , nicht-eindeutige optimale Lösung
- (c) unbeschränktes  $\mathcal{X}$ , eindeutige optimale Lösung
- (d) unbeschränktes  $\mathcal{X}$ , nicht-eindeutige optimale Lösung
- (e) unbeschränktes  $\mathcal{X}$ , keine optimale Lösung
- (f)  $\mathcal{X} = \emptyset$





# Maximumproblem

## Definition 4.5

Ein LP der Form

$$\text{Maximiere } z = F(x_1, \dots, x_n) = c_1x_1 + \dots + c_nx_n = \sum_{j=1}^n c_jx_j$$

unter den Nebenbedingungen

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \quad (i = 1, \dots, m)$$

und den Vorzeichenbedingungen  $x_j \geq 0 \quad (j = 1, \dots, n)$

heißt **Maximumproblem**.

## Kompakte Schreibweise

Mit  $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{b} \in \mathbb{R}^m$  und  $\mathbf{A} \in \mathbb{R}^{m \times n}$  können wir ein Maximumproblem auch schreiben als:

Maximiere

$$\mathbf{c}^T \mathbf{x}$$

unter den Nebenbedingungen

$$\mathbf{Ax} \leq \mathbf{b}$$

und den Vorzeichenbedingungen

$$\mathbf{x} \geq \mathbf{0}.$$

## Beispiel 4.2 in kompakter Schreibweise

### Beispiel 4.6

Maximiere

$$(30, 25) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

unter den Nebenbedingungen

$$\begin{pmatrix} 1 & 1 \\ 5 & 2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 10 \\ 30 \\ 6 \\ 9 \end{pmatrix}$$

und den Vorzeichenbedingungen

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

# Umformung in ein Maximumproblem

## Satz 4.7

*Zu jedem LP lässt sich ein äquivalentes LP in Form eines Maximumproblems formulieren.*

### Beweis.

- Ersetze zu minimierende Zielfunktion  $z = F(\mathbf{x})$  durch zu maximierende Zielfunktion  $-z = -F(\mathbf{x})$
- Transformiere  $\geq$ -Nebenbedingung durch Multiplikation beider Seiten mit  $-1$  in eine  $\leq$ -Nebenbedingung.
- Eine Gleichung  $\sum_{j=1}^n a_{ij}x_j = b_i$  kann durch zwei Ungleichungen  $\sum_{j=1}^n a_{ij}x_j \leq b_i$  und  $\sum_{j=1}^n -a_{ij}x_j \leq -b_i$  ersetzt werden.
- Falls für  $x_j$  beliebige Werte aus  $\mathbb{R}$  erlaubt sind, so ersetze  $x_j$  durch die zwei Variablen  $x_j^+ \geq 0$  und  $x_j^- \geq 0$  mit  $x_j = x_j^+ - x_j^-$ .



## Beispiel 4.8

Wir überführen das folgende LP in ein Maximumproblem:

Minimiere

$$z = 3x_1 - 4x_2$$

unter den Nebenbedingungen:

$$2x_1 + 3x_2 \leq 7$$

$$x_1 - 2x_2 \geq 4$$

$$3x_1 + 2x_2 = 6$$

$$x_1 \geq 0, x_2 \in \mathbb{R}$$

## Fortsetzung Beispiel 4.8

Zunächst sorgen wir für eine Maximierung und stellen alle Nebenbedingungen als  $\leq$  Nebenbedingungen dar:

Maximiere

$$-z = -3x_1 + 4x_2$$

unter den Nebenbedingungen:

$$2x_1 + 3x_2 \leq 7$$

$$-x_1 + 2x_2 \leq -4$$

$$3x_1 + 2x_2 \leq 6$$

$$-3x_1 - 2x_2 \leq -6$$

$$x_1 \geq 0, x_2 \in \mathbb{R}$$

## Fortsetzung Beispiel 4.8

Nun wird durch  $x_2 = x_2^+ - x_2^-$  mit  $x_2^+, x_2^- \geq 0$  die fehlende Vorzeichenbeschränkung eliminiert.  
Wir erhalten:

Maximiere

$$-z = -3x_1 + 4x_2^+ - 4x_2^-$$

unter den Nebenbedingungen:

$$2x_1 + 3x_2^+ - 3x_2^- \leq 7$$

$$-x_1 + 2x_2^+ - 2x_2^- \leq -4$$

$$3x_1 + 2x_2^+ - 2x_2^- \leq 6$$

$$-3x_1 - 2x_2^+ + 2x_2^- \leq -6$$

$$x_1, x_2^+, x_2^- \geq 0$$

# Normalform

## Definition 4.9

Ein LP liegt in **Normalform** vor, wenn es die Form hat:

Maximiere 
$$z = F(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j$$

unter den Nebenbedingungen 
$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = 1, \dots, m)$$

und Vorzeichenbedingungen 
$$x_j \geq 0 \quad (j = 1, \dots, n)$$

In kompakter Darstellung:

Maximiere 
$$\mathbf{c}^T \mathbf{x}$$

unter den Nebenbedingungen 
$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

und den Vorzeichenbedingungen 
$$\mathbf{x} \geq \mathbf{0}.$$



# Umformung in Normalform

## Satz 4.10

*Zu jedem LP lässt sich ein äquivalentes LP in Normalform formulieren.*

## Beweis.

Nach Satz 4.7 lässt sich zu jedem LP ein äquivalentes Maximumproblem formulieren. Es reicht daher zu zeigen, dass jedes Maximumproblem in Normalform überführt werden kann.

Hierzu führen wir für die  $m$  Ungleichungen die **Schlupfvariablen**  $x_{n+1}, \dots, x_{n+m}$  ein, die in der Zielfunktion mit 0 bewertet werden.

Die Variablen  $x_1, \dots, x_n$  heißen **Strukturvariablen**.

## Fortsetzung Beweis.

Die Normalform ergibt sich dann durch:

Maximiere  $z = \sum_{j=1}^n c_j x_j + \sum_{j=n+1}^{n+m} 0 \cdot x_j$  unter den Bedingungen

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i \quad (i = 1, \dots, m)$$

und Vorzeichenbedingungen

$$x_j \geq 0 \quad (j = 1, \dots, n + m)$$

In Matrixschreibweise lautet die Normalform

$$z = F(\mathbf{x}) = \mathbf{c}^T \mathbf{x}$$

unter den Bedingungen

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}$$

# Kanonische Normalform

## Definition 4.11

Gelten in der Matrixschreibweise für die Normalform die Eigenschaften

$$\mathbf{b} \geq \mathbf{0}, \mathbf{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{und} \quad \mathbf{A} = \left( \begin{array}{ccc|cc} a_{1,1} & \cdots & a_{1,n} & 1 & 0 \\ \vdots & & \vdots & & \ddots \\ a_{m,1} & \cdots & a_{m,n} & 0 & 1 \end{array} \right)$$

so ist das LP in **kanonischer Form**.

## Beispiel 4.12

Maximiere

$$z = 30x_1 + 25x_2$$

unter den Bedingungen

$$\begin{array}{rclcl} x_1 & + & x_2 & \leq & 10 \\ 5x_1 & + & 2x_2 & \leq & 30 \\ x_1 & & & \leq & 6 \\ & & x_2 & \leq & 9 \\ x_1 & & & \geq & 0 \\ & & x_2 & \geq & 0 \end{array}$$

Für die Nebenbedingungen führen wir die Schlupfvariablen  $x_3, x_4, x_5, x_6$  ein und erhalten ...

## Fortsetzung Beispiel 4.12.

Maximiere

$$z = 30x_1 + 25x_2 + 0 \cdot x_3 + \cdots + 0 \cdot x_6$$

unter den Bedingungen

$$\begin{array}{rcccccccl} x_1 & + & x_2 & + & x_3 & & & = & 10 \\ 5x_1 & + & 2x_2 & & & + & x_4 & = & 30 \\ x_1 & & & & & & + & x_5 & = & 6 \\ & & x_2 & & & & & + & x_6 & = & 9 \end{array}$$

und  $x_j \geq 0$  ( $j = 1, \dots, 6$ ).

## Fortsetzung Beispiel 4.12.

In Matrixschreibweise:

Maximiere

$$z = \begin{pmatrix} 30 & 25 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_6 \end{pmatrix}$$

unter den Bedingungen

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 5 & 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 6 \\ 9 \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} x_1 \\ \vdots \\ x_6 \end{pmatrix} \geq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

# Existenz von Extremwerten

## Satz 4.13 (Weierstraß)

Es sei  $M \subseteq \mathbb{R}^n$  eine Menge und es sei  $f : M \rightarrow \mathbb{R}$  eine stetige Funktion.

- Ist  $M$  kompakt, dann ist die Funktion  $f$  auf  $M$  beschränkt und es existieren Minimum und Maximum für  $f$ .
- Ist  $M$  abgeschlossen und  $f$  auf  $M$  nach unten beschränkt, dann existiert das Minimum für  $f$ .
- Ist  $M$  abgeschlossen und  $f$  auf  $M$  nach oben beschränkt, dann existiert das Maximum für  $f$ .

# Konsequenzen für lineare Programme

- Ist  $\mathcal{X}_{LP}$  nicht leer und beschränkt, dann existiert eine Lösung.  
vgl. Folie 216, Fälle (a) und (b)
- Ist  $\mathcal{X}_{LP}$  nicht leer und die Zielfunktion auf dem zulässigen Bereich nach oben beschränkt, dann existiert für Maximierungsprobleme eine Lösung.  
vgl. Folie 216, Fälle (c) und (d)
- Ist  $\mathcal{X}_{LP}$  nicht leer und die Zielfunktion auf dem zulässigen Bereich nach unten beschränkt, dann existiert für Minimierungsprobleme eine Lösung.



# Konvexität

## Definition 4.14

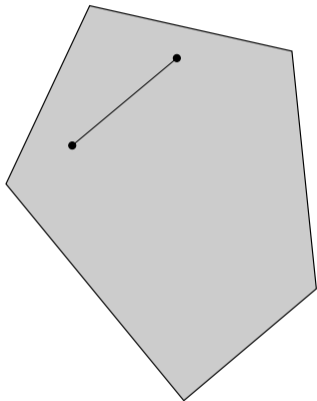
Eine Menge  $M \subseteq \mathbb{R}^n$  heißt **konvex** gdw. für je zwei Punkte  $\mathbf{x} \in M$  und  $\mathbf{y} \in M$  und alle  $\lambda \in \mathbb{R}$  mit  $0 \leq \lambda \leq 1$  gilt:

$$\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in M.$$

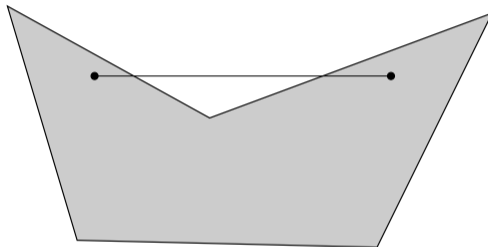
Die **konvexe Hülle**  $\text{conv}(M)$  einer Menge  $M \subseteq \mathbb{R}^n$  ist die kleinste konvexe Menge, die  $M$  enthält, d.h.

$$\text{conv}(M) = \bigcap_{\substack{M \subseteq K \\ K \text{ konvex}}} K$$

# Konvexe Menge



konvexe Menge



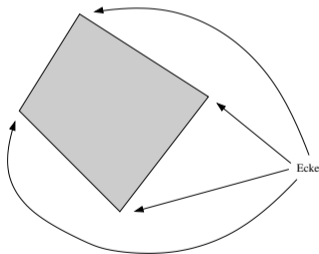
nicht konvexe Menge

# Ecke

## Definition 4.15

Es  $M \subseteq \mathbb{R}^n$  eine konvexe Menge.

$z$  heißt **Ecke** von  $M$ , wenn sich  $z$  nicht als echte Konvexkombination zweier verschiedener Punkte  $x \in M$  und  $y \in M$  darstellen lässt.



**Frage:** Wie viele Ecken hat ein Kreis?

# Konvexität und LP-Lösungen

## Folgerung 4.16

*Es gilt:*

- *Die Menge der hinsichtlich jeder einzelnen Nebenbedingung zulässigen Lösungen ist konvex.*
- *Die Menge  $\mathcal{X}_{LP}$  der zulässigen Lösungen eines LP ist konvex.*
- *Die Menge  $\mathcal{X}_{LP}^*$  der optimalen Lösungen eines LP ist konvex.*

# Ecken und LP-Lösungen

## Satz 4.17

Gegeben sei ein LP (als Maximumproblem).

- Die Menge  $\mathcal{X}$  der zulässigen Lösungen des LP hat endlich viele Ecken.
- Ist  $\mathcal{X}$  beschränkt, so nimmt die Zielfunktion ihr Maximum in mindestens einer Ecke von  $\mathcal{X}$  an.
- Ist  $\mathcal{X}$  unbeschränkt, aber die Zielfunktion  $F(\mathbf{x})$  auf  $\mathcal{X}$  nach oben beschränkt, so nimmt  $F(\mathbf{x})$  das Maximum in mindestens einer Ecke von  $\mathcal{X}$  an.
- Ist  $\mathcal{X}$  unbeschränkt und  $F(\mathbf{x})$  auf  $\mathcal{X}$  nach oben unbeschränkt, so hat das LP keine Lösung.

vgl. Folie 216

# Basis

## Definition 4.18

Gegeben sei ein LP in der Normalform mit  $m$  als Rang der Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ .

- $\mathbf{x} \in \mathbb{R}^n$  mit  $\mathbf{Ax} = \mathbf{b}$  heißt **Basislösung** gdw.  $n - m$  Komponenten  $x_j$  gleich Null und die zu den restlichen Variablen gehörenden Spaltenvektoren  $\mathbf{a}^j$  linear unabhängig sind.
- Eine Basislösung, die zulässig ist ( $\mathbf{x} \geq \mathbf{0}$ ), heißt **zulässige Basislösung**.
- Die  $m$  linear unabhängigen Spaltenvektoren  $\mathbf{a}^j$  einer (zulässigen) Basislösung heißen **Basisvektoren**, die zugehörigen Variablen  $x_j$  **Basisvariablen (BV)**.
- Alle übrigen Spaltenvektoren heißen **Nichtbasisvektoren**, die zugehörigen Variablen **Nichtbasisvariablen (NBV)**.
- Die Menge aller Basisvariablen  $x_j$  einer Basislösung bezeichnet man als **Basis**.

# Charakterisierung zulässiger Basislösungen

## Satz 4.19

*$x$  ist genau dann eine zulässige Basislösung eines LP, wenn  $x$  ist Ecke von  $\mathcal{X}_{LP}$  ist.*

**Bemerkung:** Dieser Satz ist der **Schlüssel zur algebraischen Lösung** von linearen Programmen.

# Ecken-Algorithmus

## Algorithmus 4.20

Gegeben sei ein LP in Normalform mit  $\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{x} \in \mathbb{R}^n$  und  $\mathbf{b} \in \mathbb{R}^m$ .

Für  $N := \binom{n}{m}$  seien  $B_1, B_2, \dots, B_N$  die  $m$ -elementigen Teilmengen der Menge  $\{1, \dots, n\}$ .

Für eine Menge  $B_k = \{j_1, \dots, j_m\}$  bezeichne  $\mathbf{A}_{B_k} = (\mathbf{a}^{j_1}, \dots, \mathbf{a}^{j_m}) \in \mathbb{R}^{m \times m}$  die Matrix, die aus den Spaltenvektoren  $j_1$  bis  $j_m$  von  $\mathbf{A}$  besteht.

Der Vektor  $\mathbf{x}_{B_k}$  ist der entsprechende Variablenvektor dessen Komponenten einen Index aus  $B_k$  haben.



## Fortsetzung Algorithmus 4.20.

- 1  $k := 1, z^* := -\infty$
- 2 Erzeuge  $B_k, \mathbf{A}_{B_k}$  und  $\mathbf{x}_{B_k}$ .
- 3 Falls  $r(\mathbf{A}_{B_k}) < m$  dann weiter mit 6.
- 4 Löse das LGS  $\mathbf{A}_{B_k} \mathbf{x}_{B_k} = \mathbf{b}$ . Es sei  $\mathbf{x}$  die Basislösung zur Lösung dieses LGS. Falls  $\mathbf{x}$  nicht zulässig ist, weiter mit 6.
- 5 Falls  $\mathbf{c}^T \mathbf{x} > z^*$ , setze  $z^* := \mathbf{c}^T \mathbf{x}$  und  $\mathbf{x}^* := \mathbf{x}$ .
- 6  $k := k + 1$ . Falls  $k \leq N$  gehe zu 2, sonst STOP!

# Bemerkungen zum Ecken-Algorithmus

- Wenn ein LP eine Lösung hat, dann liefert der Ecken-Algorithmus eine optimale Lösung  $\mathbf{x}^*$  mit Zielfunktionswert  $z^*$ .
- Die Bestimmung des Rang von  $\mathbf{A}_{B_k}$  in Schritt 3 und die Lösung des LGS in Schritt 4 kann mit dem [Gaußschen Algorithmus](#) oder der [Cramer-Regel](#) erfolgen.
- Der Algorithmus hat keine praktische Bedeutung und ist nur für kleine  $n$  und  $m$  durchführbar.

## Beispiel zum Eckenalgorithmus

### Beispiel 4.21

Wir greifen das Beispiel mit dem Eisverkäufer (Beispiel 4.2 bzw. 4.4) wieder auf.

Maximiere  $z = F(\mathbf{x}) = 30x_1 + 25x_2$  unter den Nebenbedingungen

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 5 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \quad x_1, \dots, x_5 \geq 0$$

Man beachte: Die redundante Nebenbedingung  $x_1 \leq 6$  wurde weggelassen.

## Fortsetzung Beispiel 4.21.

Man erhält  $\binom{5}{3} = 10$  verschiedene Spaltenmengen für die Matrix **A**:

$$B_1 = \{1, 2, 3\} : \begin{pmatrix} 1 & 1 & 1 \\ 5 & 2 & 0 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \text{ ergibt } x_3 < 0$$

$$B_2 = \{1, 2, 4\} : \begin{pmatrix} 1 & 1 & 0 \\ 5 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \text{ Ecke } \mathbf{x} = \begin{pmatrix} 1 \\ 9 \\ 0 \\ 7 \\ 0 \end{pmatrix}$$

mit  $F(\mathbf{x}) = 255$

$$B_3 = \{1, 2, 5\} : \begin{pmatrix} 1 & 1 & 0 \\ 5 & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \text{ Ecke } \mathbf{x} = \begin{pmatrix} 10/3 \\ 20/3 \\ 0 \\ 0 \\ 7/3 \end{pmatrix}$$

mit  $F(\mathbf{x}) = 266\frac{2}{3}$

## Fortsetzung Beispiel 4.21.

$B_4 = \{1, 3, 4\}$  :  $\begin{pmatrix} 1 \\ 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$  sind linear abhängig.

$B_5 = \{1, 3, 5\}$  :  $\begin{pmatrix} 1 & 1 & 0 \\ 5 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_3 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix}$  Ecke  $\mathbf{x} = \begin{pmatrix} 6 \\ 0 \\ 4 \\ 0 \\ 9 \end{pmatrix}$

mit  $F(\mathbf{x}) = 180$

$B_6 = \{1, 4, 5\}$  :  $\begin{pmatrix} 1 & 0 & 0 \\ 5 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix}$  liefert  $x_4 < 0$

## Fortsetzung Beispiel 4.21.

$$B_7 = \{2, 3, 4\} : \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \quad \text{Ecke } \mathbf{x} = \begin{pmatrix} 0 \\ 9 \\ 1 \\ 12 \\ 0 \end{pmatrix}$$

mit  $F(\mathbf{x}) = 225$

$$B_8 = \{2, 3, 5\} : \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ x_3 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \quad \text{liefert } x_3 < 0$$

$$B_9 = \{2, 4, 5\} : \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_2 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \quad \text{liefert } x_5 < 0$$

## Fortsetzung Beispiel 4.21.

$$B_{10} = \{3, 4, 5\} : \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 10 \\ 30 \\ 9 \end{pmatrix} \quad \text{Ecke } \mathbf{x} = \begin{pmatrix} 0 \\ 0 \\ 10 \\ 30 \\ 9 \end{pmatrix}$$

mit  $F(\mathbf{x}) = 0$

Für die Ecke  $\begin{pmatrix} 10/3 \\ 20/3 \\ 0 \\ 0 \\ 7/3 \end{pmatrix}$  wird der maximale Zielfunktionswert

$z = F(\mathbf{x}) = 266\frac{2}{3}$  angenommen.

# Grundideen

- Der **Simplexalgorithmus** ist das Standardverfahren zur Lösung von linearen Programmen.
- von **George Bernard Dantzig** in den 40ern des 20. Jahrhunderts entwickelt
- Vorgehen: versuche ausgehend von einer **Startecke** mit einer **Ausgangsbasis** durch **Basisaustausch** zu einer Ecke mit besserem Zielfunktionswert fortzuschreiten.
- Da es nur endlich viele Ecken gibt, erhalten wir nach endlich vielen Schritten die optimale Lösung.
- Der **Basistausch geschieht dabei so sparsam wie möglich**: Es wird stets genau eine Basisvariable gegen eine Nichtbasisvariable ausgetauscht.



## Fazit zu Grundideen

Konstruktion einer Folge  $(\mathbf{x}^{(r)})$  von Basislösungen mit

$$\mathbf{c}^T \mathbf{x}^{(r+1)} \geq \mathbf{c}^T \mathbf{x}^{(r)}$$

und Abbruch, wenn keine Verbesserung mehr möglich ist.

## Beispiel für ein Simplextableau

### Beispiel 4.22

Wir bleiben beim Problem des Eisverkäufers (Beispiel 4.21) und ordnen die Daten in einem **Simplextableau** an:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$z$	<b>b</b>
$x_3$	1	1	1	0	0	0	10
$x_4$	5	2	0	1	0	0	30
$x_5$	0	1	0	0	1	0	9
$z$	-30	-25	0	0	0	1	0

## Fortsetzung Beispiel 4.22.

- Die **Strukturvariablen**  $x_1, x_2$  sind **NBV**, die **Schlupfvariablen**  $x_3, x_4, x_5$  sind **BV**.
- Die Werte der BV ergeben sich aus den Nebenbedingungsgleichungen, die durch die Zeilen des Tableaus repräsentiert werden. Hierdurch ist eine Ecke gegeben.
- Für den Zielfunktionswert  $z = \mathbf{c}^T \mathbf{x}$  wird eine neue Variable  $z$  eingeführt und die Gleichung  $-\mathbf{c}^T \mathbf{x} + z = 0$  wird wie eine zusätzliche Nebenbedingung aufgefasst. Wir betrachten also eigentlich das Problem:

$$\begin{array}{rcl} & \max z & \\ \text{u.d.N.} & \mathbf{Ax} = \mathbf{b} & \\ & -\mathbf{c}^T \mathbf{x} + z = 0 & \\ & \mathbf{x} \geq \mathbf{0} & \end{array}$$

- Letzte Zeile ist **Zielfunktionszeile**, aufgefasst als Nebenbedingung. Der Zielfunktionswert  $z$  steht ganz rechts.

## Beispiel für einen Basiswechsel

### Beispiel 4.23

Im Tableau von Beispiel 4.22 verspricht  $x_1$  den größeren Zuwachs,  $x_1$ -Spalte ist daher die **Pivotspalte**.

$x_1$  kann höchstens den Wert  $30/5 = 6$  annehmen,  $x_4$  würde dann 0.  $x_4$ -Zeile ist daher die **Pivotzeile**.

Wir nehmen  $x_1$  in die Basis auf, dafür wird  $x_4$  aus der Basis herausgenommen. Dies ist der **Basisaustausch**.

Pivotspalte und Pivotzeile schneiden sich im **Pivotelement**, hier  $a_{21} = 5$ . Wir teilen die Pivotzeile durch das Pivotelement. Damit entsteht

$$x_1 + \frac{2}{5}x_2 + \frac{1}{5}x_4 = 6$$

Aus dieser Gleichung folgt  $x_1 = 6 - \frac{2}{5}x_2 - \frac{1}{5}x_4$ . Dies setzen wir in alle übrigen Gleichungen ein.

### Fortsetzung Beispiel 4.23.

Für die erste Zeile erhalten wir  $(6 - \frac{2}{5}x_2 - \frac{1}{5}x_4) + x_2 + x_3 = 10$ . Dies ergibt  $\frac{3}{5}x_2 + x_3 - \frac{1}{5}x_4 = 4$ .

Die dritte Zeile bleibt unverändert, da  $x_1$  dort nicht auftritt.

Die Zielfunktionszeile wird zu  $-30(6 - \frac{2}{5}x_2 - \frac{1}{5}x_4) - 25x_2 + z = 0$ , also  $-13x_2 + 6x_4 + z = 180$ .

Jetzt können wir das neue Tableau aufstellen:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$z$	<b>b</b>
$x_3$	0	$\frac{3}{5}$	1	$-\frac{1}{5}$	0	0	4
$x_1$	1	$\frac{2}{5}$	0	$\frac{1}{5}$	0	0	6
$x_5$	0	1	0	0	1	0	9
$z$	0	-13	0	6	0	1	180

## Fortsetzung Beispiel 4.23.

Durch Vertauschen der Spalten für  $x_1$  und  $x_4$  können wir das Tableau wieder in die übliche Form bringen:

	$x_4$	$x_2$	$x_3$	$x_1$	$x_5$	$z$	<b>b</b>
$x_3$	$-\frac{1}{5}$	$\frac{3}{5}$	1	0	0	0	4
$x_1$	$\frac{1}{5}$	$\frac{2}{5}$	0	1	0	0	6
$x_5$	0	1	0	0	1	0	9
$z$	6	-13	0	0	0	1	180

Die zugehörige Ecke ist  $\begin{pmatrix} 6 \\ 0 \\ 4 \\ 0 \\ 9 \end{pmatrix}$ .

## Fortsetzung Beispiel 4.23.

Der nächste Austauschschritt liefert das Tableau:

	$x_4$	$x_3$	$x_2$	$x_1$	$x_5$	$b_i$
$x_2$	$-\frac{1}{3}$	$\frac{5}{3}$	1	0	0	$\frac{20}{3}$
$x_1$	$\frac{1}{3}$	$-\frac{2}{3}$	0	1	0	$\frac{10}{3}$
$x_5$	$\frac{1}{3}$	$-\frac{5}{3}$	0	0	1	$\frac{7}{3}$
$z$	$\frac{5}{3}$	$\frac{65}{3}$	0	0	0	$\frac{800}{3}$

Das heißt in der Ecke  $\begin{pmatrix} \frac{10}{3} \\ \frac{20}{3} \\ 0 \\ 0 \\ \frac{7}{3} \end{pmatrix}$  wird das Optimum mit  $z = \frac{800}{3}$  angenommen.

## Fazit für Beispiel 4.23

- Wir dividieren also die Pivotzeile durch den Pivotwert.
- Zu den übrigen Zeilen addieren wir ein Vielfaches der Pivotzeile, so dass in der Pivotspalte Nullen entstehen (analog zum Gaußalgorithmus).
- optionale Spaltenvertauschung, um wieder die reine kanonische Form zu erlangen
- Aus der Spalte **b** und den Basisvariablen ergibt sich die Ecke.
- Solange in der Zielfunktionszeile Koeffizienten  $< 0$  auftreten, ist eine Verbesserung möglich.
- Der Algorithmus terminiert, wenn in der Zielfunktionszeile alle Koeffizienten  $\geq 0$  sind.



# Kanonische Form

- Dieses Vorgehen funktioniert in der gezeigten Weise nur, wenn zu Beginn ein **kanonisches Maximumproblem** vorliegt, d.h. ein Problem der Form

$$\max \mathbf{c}^T \mathbf{x} \text{ u.d.N. } \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq 0$$

und **zusätzlich**  $\mathbf{b} \geq \mathbf{0}$  gilt.

- Für dieses kanonische Maximumproblem erhalten wir mit Hilfe von  $m$  Schlupfvariablen ein **Problem in kanonischer Normalform**.
- Die **erste Basislösung** ist dann durch die Schlupfvariablen bestimmt (siehe Beispiel 4.22).
- Wegen  $\mathbf{b} \geq 0$  ist diese Basislösung zulässig und stellt damit eine Ecke (die **Startecke**) dar.

## Starttableau für kanonisches Maximumproblem

BV	$x_1$	$\cdots$	$x_n$	$x_{n+1}$	$\cdots$	$x_{n+m}$	$z$	$\mathbf{b}$
$x_{n+1}$	$a_{1,1}$	$\cdots$	$a_{1,n}$	1	$\cdots$	0	0	$b_1$
$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$x_{n+m}$	$a_{m,1}$	$\cdots$	$a_{m,n}$	0	$\cdots$	1	0	$b_m$
$z$	$-c_1$	$\cdots$	$-c_n$	0	$\cdots$	0	1	0

### Definition 4.24

Für  $\mathbf{b} \geq \mathbf{0}$  heißt solch ein Tableau **primal zulässig**.

**Bemerkung:** Da sich die Spalte  $z$  nie ändert, können wir auf diese Spalte im Tableau auch verzichten.

# Primaler Simplexalgorithmus

## Algorithmus 4.25

Es liege ein kanonisches Maximumproblem ( $\mathbf{b} \geq \mathbf{0}$ ) vor mit  $n$  Variablen und  $m$  Nebenbedingungen, also  $n$  Struktur- und  $m$  Schlupfvariablen in Normalform.

Start: Ecke des Starttableaus ist:

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix}$$

mit  $z = 0$ . Schlupfvariablen sind BV, Strukturvariablen sind NBV.

## Fortsetzung Algorithmus 4.25.

Starttableau:

	$x_1$	$\cdots$	$x_t$	$\cdots$	$x_n$	$x_{n+1}$	$\cdots$	$x_{n+s}$	$\cdots$	$x_{n+m}$	$b_i$
$x_{n+1}$	$a_{n+1,1}$	$\cdots$	$a_{n+1,t}$	$\cdots$	$a_{n+1,n}$	1	$\cdots$	0	$\cdots$	0	$b_1$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_s$	$a_{s,1}$	$\cdots$	$a_{s,t}$	$\cdots$	$a_{s,n}$	0	$\cdots$	1	$\cdots$	0	
$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$x_{n+m}$	$a_{n+m,1}$	$\cdots$	$a_{n+m,t}$	$\cdots$	$a_{n+m,n}$	0	$\cdots$	0	$\cdots$	1	$b_m$
$Z$	$-c_1$	$\cdots$	$-c_t$	$\cdots$	$-c_n$	0	$\cdots$	0	$\cdots$	0	0

Von hier ab stehen Indices nicht für eine Spalten- oder Zeilennummer, sondern für den Variablenindex in der entsprechenden Spalte- bzw. Zeile des Simplextableaus.

## Fortsetzung Algorithmus 4.25.

**Wahl der Pivotspalte:** Ist die Zielfunktionszeile von der Gestalt

$$\overline{z \mid d_1 \quad \cdots \quad d_t \quad \cdots \quad d_n \mid 0 \quad \cdots \quad 0 \mid d}$$

mit  $d_j \geq 0$ , ( $j = 1, \dots, n$ ), so liegt eine Optimallösung vor.

Andernfalls machen wir eine Spalte  $t$  mit negativem  $d_t$  zur Pivotspalte und die NBV  $x_t$  zur BV.

**Wahl der Pivotzeile:** Sind in der Pivotspalte alle  $a_{i,t} \leq 0$ , so wächst  $z$  unbeschränkt, da  $x_t$  unbeschränkt wachsen kann. Es gibt dann keine Optimallösung.

Andernfalls bestimmen wir eine Zeile  $s$  durch

$$\frac{b_s}{a_{s,t}} = \min_{i=1}^m \frac{b_i}{a_{i,t}} \text{ für } a_{i,t} > 0$$

Die NBV  $x_t$  wird BV und bekommt den Wert  $\frac{b_s}{a_{s,t}}$ .

Die bisherige BV  $x_s$  wird NBV und nimmt den Wert 0 an.

## Fortsetzung Algorithmus 4.25.

Austauschschritt: Das neue Tableau lautet: Linke Hälfte:

	$x_1$	$\dots$	$x_t$	$\dots$	$x_n$	
$x_1$	$a_{1,1} - \frac{a_{1,t}}{a_{s,t}} a_{s,1}$	$\dots$	0	$\dots$	$a_{1,n} - \frac{a_{1,t}}{a_{s,t}} a_{s,n}$	
$\vdots$	$\vdots$		$\vdots$		$\vdots$	
$x_t$	$\frac{a_{s,1}}{a_{s,t}}$	$\dots$	1	$\dots$	$\frac{a_{s,n}}{a_{s,t}}$	$\dots$
$\vdots$	$\vdots$		$\vdots$		$\vdots$	
$x_m$	$a_{m,1} - \frac{a_{m,t}}{a_{s,t}} a_{s,1}$	$\dots$	0	$\dots$	$a_{m,n} - \frac{a_{m,t}}{a_{s,t}} a_{s,n}$	
$z$	$d_1 - \frac{d_t}{a_{s,t}} a_{s,1}$	$\dots$	0	$\dots$	$d_n - \frac{d_t}{a_{s,t}} a_{s,n}$	

## Fortsetzung Algorithmus 4.25.

Rechte Hälfte:

	$x_{n+1}$	$\dots$	$x_s$	$\dots$	$x_{n+m}$	$b_i$
	1	$\dots$	$-\frac{a_{1,t}}{a_{s,t}}$	$\dots$	0	$b_1 - \frac{b_s}{a_{s,t}} a_{1,t}$
	$\vdots$		$\vdots$		$\vdots$	$\vdots$
$\dots$	0	$\dots$	$\frac{1}{a_{s,t}}$	$\dots$	0	$\frac{b_s}{a_{s,t}}$
	$\vdots$		$\vdots$		$\vdots$	$\vdots$
	0	$\dots$	$-\frac{a_{m,t}}{a_{s,t}}$	$\dots$	1	$b_m - \frac{b_s}{a_{s,t}} a_{m,t}$
	0	$\dots$	$-\frac{d_t}{a_{s,t}}$	$\dots$	0	$d - \frac{b_s}{a_{s,t}} d_t$

**Terminierung:** Wenn alle Koeffizienten der Zielfunktionszeile nichtnegative Werte haben, beschreibt das Tableau eine optimale Ecke. Rechts unten steht dann  $z^*$ .

Andernfalls vertauschen wir die Spalten, so dass das Tableau wieder kanonische Form annimmt. Wir beginnen nun wieder von vorne.

# Eigenschaften des primalen Simplexalgorithmus

## Satz 4.26

*Das  $r$ -te Tableau beim Simplexalgorithmus sei primal zulässig. Wählen wir die Pivotspalte und die Pivotzeile gemäß Algorithmus 4.25, so ist das  $(r + 1)$ -te Tableau wieder primal zulässig, und es gilt  $z^{(r+1)} \geq z^{(r)}$ .*



## Beispiel zum Simplexalgorithmus

### Beispiel 4.27

In einem Betrieb sind drei Maschinen vorhanden, die für die Herstellung zweier Produkte benötigt werden. Bei der Produktion müssen die Produkte auf mehreren Maschinen bearbeitet werden, wobei die folgenden Bearbeitungszeiten anfallen:

	Maschine 1	Maschine 2	Maschine 3
Produkt A	40	24	0
Produkt B	24	48	60

Die tägliche Maschinenlaufzeit beträgt 480 Minuten. Der Ertrag pro Einheit beträgt 10 € für Produkt A und 40 € für Produkt B.

Welche Anzahl der Produkte ist täglich zu fertigen, so dass der Ertrag maximal wird?

## Fortsetzung Beispiel 4.27.

**Mathematische Modellierung:** $x_1$  produzierte Menge von Produkt A $x_2$  produzierte Menge von Produkt B

LP:

$$\text{Maximiere } z = F(x_1, x_2) = 10x_1 + 40x_2$$

unter den Nebenbedingungen

$$40x_1 + 24x_2 \leq 480$$

$$24x_1 + 48x_2 \leq 480$$

$$60x_2 \leq 480$$

$$x_1, x_2 \geq 0$$

## Fortsetzung Beispiel 4.27.

Starttableau:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$z$	<b>b</b>
$x_3$	40	24	1	0	0	0	480
$x_4$	24	48	0	1	0	0	480
$x_5$	0	60	0	0	1	0	480
$z$	-10	-40	0	0	0	1	0

2. Tableau:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$z$	<b>b</b>
$x_3$	40	0	1	0	$-2/5$	0	288
$x_4$	24	0	0	1	$-4/5$	0	96
$x_2$	0	1	0	0	$1/60$	0	8
$z$	-10	0	0	0	$2/3$	1	320

## Fortsetzung Beispiel 4.27.

3. Tableau:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$z$	<b>b</b>
$x_3$	0	0	1	$-5/3$	$14/15$	0	128
$x_1$	1	0	0	$1/24$	$-1/30$	0	4
$x_2$	0	1	0	0	$1/60$	0	8
$z$	0	0	0	$5/12$	$1/3$	1	360

Also ist

$$x^* = \begin{pmatrix} 4 \\ 8 \\ 128 \\ 0 \\ 0 \end{pmatrix}$$

eine optimale Lösung.

## Opportunitätskosten und Schattenpreise

Das Endtableau von Beispiel 4.27 entspricht dem LGS:

$$\begin{array}{rclcl}
 & x_3 & -\frac{5}{3}x_4 & +\frac{14}{15}x_5 & = & 128 \\
 x_1 & & +\frac{1}{24}x_4 & -\frac{1}{30}x_5 & = & 4 \\
 & x_2 & & +\frac{1}{60}x_5 & = & 8 \\
 & & \frac{5}{12}x_4 & +\frac{1}{3}x_5 & +z & = & 360
 \end{array}$$

Aufgelöst nach den Basisvariablen entsteht

$$\begin{array}{rcl}
 x_3 & = & 128 + \frac{5}{3}x_4 - \frac{14}{15}x_5 \\
 x_1 & = & 4 - \frac{1}{24}x_4 + \frac{1}{30}x_5 \\
 x_2 & = & 8 - \frac{1}{60}x_5 \\
 z & = & 360 - \frac{5}{12}x_4 - \frac{1}{3}x_5
 \end{array}$$

- Maschine 2 ( $x_4 = 0$ ) und Maschine 3 ( $x_5 = 0$ ) sind voll ausgelastet,
- dagegen steht Maschine 1 pro Tag  $x_3 = 128$  Minuten still.
- Der Gewinn würde sich um  $\frac{5}{12}$  € bzw.  $\frac{1}{3}$  € pro Maschinenminute bei Maschine 2 bzw. 3 verringern,
- bzw. um  $\frac{5}{12}$  € bzw.  $\frac{1}{3}$  € erhöhen, wenn eine Maschinenminute mehr zur Verfügung stehen würde.
- Diese Werte nennen wir **Opportunitätskosten** bzw. **Schattenpreise**.
- Sie entsprechen dem entgangenen Gewinn durch die nicht mehr verfügbare Kapazität
- bzw. den Preisen, die der Hersteller bereit wäre, für eine Maschinenminute von Maschine 2 bzw. 3 zu zahlen.

## Lineare Ausgleichsprobleme auf Basis der Summennorm

Wir betrachten nun das Fehlerfunktional

$$E(\lambda_1, \dots, \lambda_m) = \sum_{i=1}^n |y_i - f(x_i)| = \sum_{i=1}^n \left| y_i - \sum_{j=1}^m \lambda_j f_j(x_i) \right|$$

für lineare Ausgleichsprobleme.

Durch den Betrag haben wir **keine Differenzierbarkeit** mehr. Stattdessen modellieren wir das **Ausgleichsproblem als LP**.

- Wir führen einen **Residuumsvektor**  $\mathbf{r}$  ein, um das Fehlergleichungssystem lösen zu können.

$$\mathbf{A}\boldsymbol{\lambda} + \mathbf{r} = \mathbf{y}$$

- Das Fehlerfunktional entspricht damit:  $E(f) = \sum_{i=1}^n |r_i| = \|\mathbf{r}\|_1$ .
- Problem:  $\mathbf{r}$  ist nicht vorzeichenbeschränkt.
- Daher setzen wir  $\mathbf{r} = \mathbf{u} - \mathbf{v}$  mit  $\mathbf{u} \geq \mathbf{0}$  und  $\mathbf{v} \geq \mathbf{0}$ .
- Damit erhalten wir:

$$\mathbf{A}\boldsymbol{\lambda} + \mathbf{u} - \mathbf{v} = \mathbf{y}$$

- Da von den beiden Variablen  $u_j$  und  $v_j$  stets eine als 0 gewählt werden kann, gilt:

$$\min \|\mathbf{r}\|_1 = \min (\|\mathbf{u}\|_1 + \|\mathbf{v}\|_1) = \min \left( \sum_{i=1}^n u_i + \sum_{i=1}^n v_i \right)$$

- Damit haben wir ein LP.



## $\| \cdot \|_1$ -Ausgleichsproblem als LP

Wir lösen das folgende LP:

$$\min \sum_{i=1}^n u_i + \sum_{i=1}^n v_i$$

unter den Nebenbedingungen:

$$\mathbf{A}\boldsymbol{\lambda} + \mathbf{u} - \mathbf{v} = \mathbf{y}$$

und den Vorzeichenbedingungen:

$$\mathbf{u} \geq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0}$$

Beachten Sie: Die Variablen  $\lambda_i$  des Variablenvektors  $\boldsymbol{\lambda}$  sind **nicht vorzeichenbeschränkt**.

## Beispiel 4.28

Für die Daten von Beispiel 3.1 erhalten wir das LP

$$\min u_0 + u_1 + u_2 + u_3 + u_4 + u_5 + v_0 + v_1 + v_2 + v_3 + v_4 + v_5$$

unter den Nebenbedingungen

$$\lambda_1 + u_0 - v_0 = 4$$

$$\lambda_1 + \lambda_2 + u_1 - v_1 = 6$$

$$\lambda_1 + 2\lambda_2 + u_2 - v_2 = 6.8$$

$$\lambda_1 + 3\lambda_2 + u_3 - v_3 = 9.5$$

$$\lambda_1 + 4\lambda_2 + u_4 - v_4 = 10.5$$

$$\lambda_1 + 5\lambda_2 + u_5 - v_5 = 11.5$$

und Vorzeichenbedingungen

$$u_i, v_i \geq 0 \text{ für } i = 0, \dots, 5, \quad \lambda_1, \lambda_2 \in \mathbb{R}$$

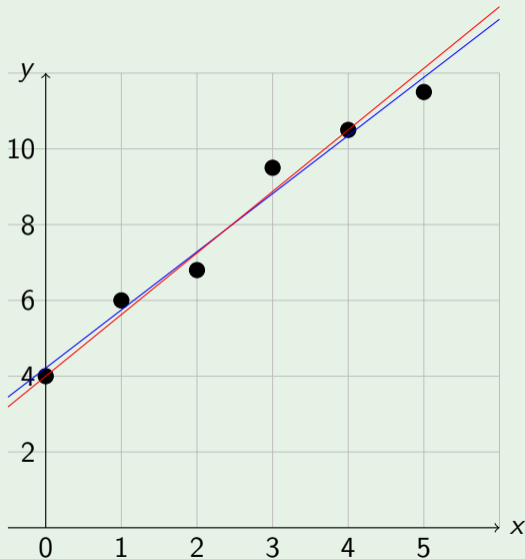
## Fortsetzung Beispiel.

Optimale Lösung (Strukturvariablen):

$$\lambda_1 = 4, \quad \lambda_2 = 1.625$$

Also lautet die Ausgleichsgerade:

$$f(x) = 1.625x + 4$$



## || $\|\infty$ -Ausgleichsproblem als LP

- Wir führen die Variable

$$z := \max\{u_1, \dots, u_n, v_1, \dots, v_n\}$$

ein.

- Damit lautet die Zielfunktion:  $\min z$
- Als zusätzliche Nebenbedingungen erhalten wir:

$$z - u_j \geq 0$$

$$z - v_j \geq 0$$

für  $i = 1, \dots, n$ .

## Beispiel 4.29

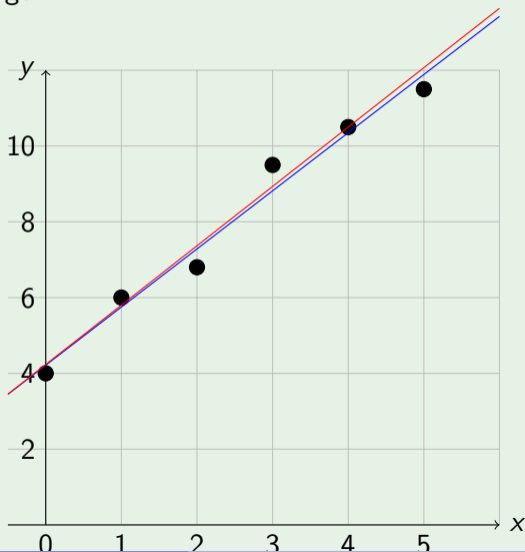
LP für die Daten von Beispiel 3.1: siehe Homepage

Optimale Lösung (Strukturvariablen):

$$\lambda_1 = 4.2333, \lambda_2 = 1.5667$$

Also lautet die Ausgleichsgerade:

$$f(x) = 1.5667x + 4.2333$$



# Robustheit

Gegeben seien  $n$  (fehlerbehaftete) Messwerte  $x_1 \leq x_2 \leq \dots \leq x_n$ .

Wir wollen

- eine Schätzung für den realen Wert  $x$  mithilfe verschiedener Fehlerfunktionale durchführen
- und die Robustheit dieser Schätzung untersuchen.

Ein Schätz- oder Testverfahren heißt **robust**, wenn es nicht sensibel auf Ausreißer reagiert.

Ein **Ausreißer** ist ein Messwert, der nicht den Erwartungen entspricht. Diese wird meistens als Streubereich um den Erwartungswert herum definiert.

# Fehlerfunktionale

Fehlerfunktionale hier für die Normen  $\| \cdot \|_2$ ,  $\| \cdot \|_1$  und  $\| \cdot \|_\infty$ :

$$E_2(x) = \sum_{i=1}^n (x - x_i)^2$$

$$E_1(x) = \sum_{i=1}^n |x - x_i|$$

$$E_\infty(x) = \max_{i=1}^n |x - x_i|$$

# Schätzungen

- für  $E_2$ :

$$x = \frac{1}{n} \sum_{i=1}^n x_i,$$

also das **arithmetische Mittel aller Messwerte**.

- für  $E_1$ :

$$x = \begin{cases} x_{\frac{n+1}{2}} & \text{wenn } n \text{ ungerade} \\ \frac{1}{2} \left( x_{\frac{n}{2}} + x_{\frac{n+1}{2}} \right) & \text{wenn } n \text{ gerade} \end{cases}$$

also der **Median** der Messwerte.

- für  $E_\infty$ :

$$x = \frac{1}{2} (\max\{x_1, \dots, x_n\} + \min\{x_1, \dots, x_n\})$$

also das **arithmetische Mittel von Maximum und Minimum** der Messwerte.



## Robustheit der Schätzungen

- Wie ändert sich der Schätzwert für  $x$  bezüglich  $E_2(x)$ ,  $E_1(x)$  und  $E_\infty(x)$ , wenn
  - ▶  $x_n \rightarrow \infty$  strebt und
  - ▶  $x_1, \dots, x_{n-1}$  unverändert bleiben?
- Wie stark ist dabei der Einfluss von  $x_n$ ?

für  $E_1$ : **kein Einfluss**,  $x$  bleibt unverändert.

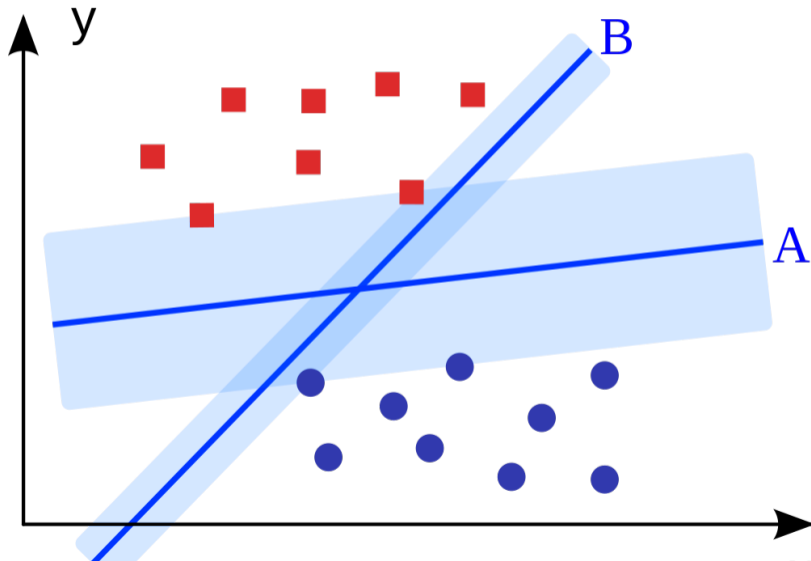
für  $E_\infty$ : **sehr starker Einfluss**,  $x$  wächst linear mit  $x_n$ , Einfluss ist unabhängig von der Anzahl der Messwerte.

für  $E_2$ : **gedämpfter Einfluss**,  $x$  wächst prinzipiell linear in  $x_n$ , aber mit gedämpftem Faktor  $\frac{1}{n}$ , also abhängig von der Anzahl der Messwerte.

# Support Vector Machine

- Wir haben **zwei Klassen** und
- eine **Menge von Objekten**, für die bekannt ist, zu welcher Klasse sie gehören.
- Jedes **Objekt** wird durch einen **Vektor in einem Vektorraum** repräsentiert.
- Aufgabe: Man finde eine **Hyperebene**, die die **Objekte in zwei Klassen** teilt.
- Dabei soll der **Abstand der Objekte, die der Hyperebene am nächsten liegen, maximiert** werden.
- Dies soll dafür sorgen, dass auch Objekte, die nicht genau den Trainingsobjekten entsprechen, möglichst zuverlässig klassifiziert werden.

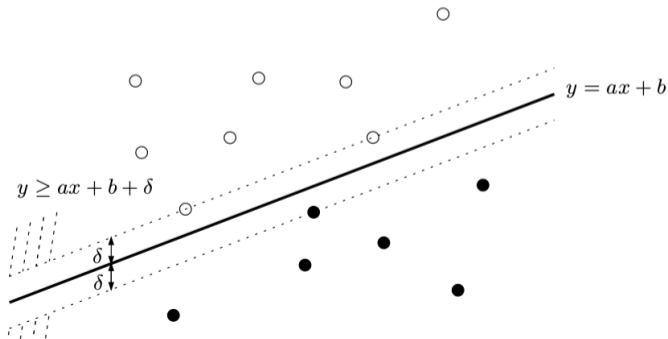
# Gute und schlechte Hyperebene



## Optimale Hyperebene formaler

Gesucht ist eine Gerade  $y = ax + b$ , so dass

- alle Punkte der Klasse  $A$  oberhalb,
- alle Punkte der Klasse  $B$  unterhalb liegen und
- und der vertikale Abstand  $\delta$  der Punkte, welche der Geraden am nächsten sind, maximiert wird.



# Zugehöriges LP

$$\max \delta$$

unter den Nebenbedingungen

$$x_i a + b + \delta \leq y_i \quad \text{für die Punkte aus } A$$

$$x_i a + b - \delta \geq y_i \quad \text{für die Punkte aus } B$$

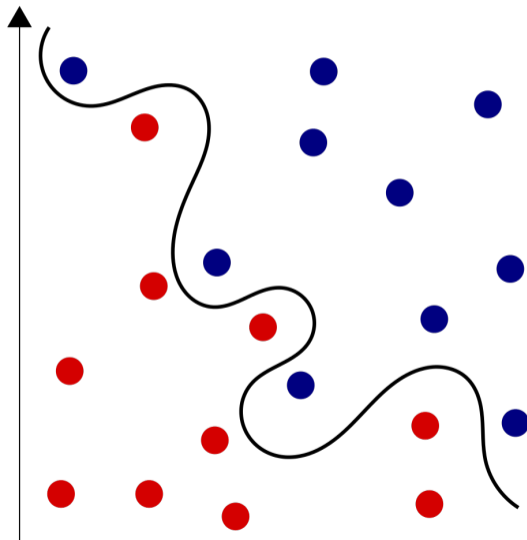
und Vorzeichenbedingungen

$$\delta \geq 0, a, b \in \mathbb{R}.$$

## Nicht linear trennbar

Wie als LP formulieren?

- vgl. Ausgleichproblem bei Summennorm
- Variablen  $u_i$  und  $v_i$  für positiven bzw. negativen Abstand von Gerade
- Für Punkte der Klasse über der Geraden: nur  $u_i$  geht in Zielfunktion ein.
- Für Punkte der Klasse unterhalb der Geraden: nur  $v_i$ .
- Übungsaufgabe!

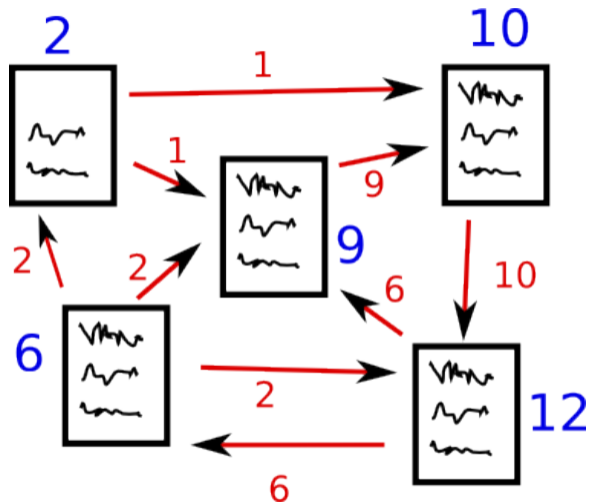


# Zusammenfassung

- LP und Normalform
- Konvexität, Ecken, optimale Lösungen treten in Ecken auf
- Ecke  $\Leftrightarrow$  zulässige Basislösung, Eckenalgorithmus
- Lösung von LPs mit dem Simplexalgorithmus
- Anwendungen: Lineare Ausgleichsrechnung für Summen- und Maximumsnorm, Optimale trennende Hyperebenen für die Klassifikation

## Kapitel 5

## Eigenwerte und Netzwerkanalyse





# Inhalt

## 5 Eigenwerte und Netzwerkanalyse

- Eigenwerte
- Numerische Berechnung von Eigenwerten
- Page Rank

# Eigenwerte und Eigenvektoren

## Definition 5.1

Es sei  $\mathbf{A} \in \mathbb{C}^{n \times n}$  eine quadratische Matrix.

Ein Vektor  $\mathbf{x} \in \mathbb{C}^n$  mit  $\mathbf{x} \neq \mathbf{0}$  heißt **Eigenvektor** zum **Eigenwert**  $\lambda \in \mathbb{C}$ , wenn gilt:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

Die **Eigenwerte der Matrix**  $\mathbf{A}$  sind alle Werte  $\lambda$ , für die ein Eigenvektor existiert.

Die Menge

$$\sigma(\mathbf{A}) = \{\lambda \in \mathbb{C} \mid \lambda \text{ ist Eigenwert von } \mathbf{A}\}$$

aller Eigenwerte von  $\mathbf{A}$  wird **Spektrum** genannt.

Der **Spektralradius**  $\rho(\mathbf{A})$  ist der größte Betrag aller Eigenwerte.

## Beispiel 5.2

Sei

$$\mathbf{A} = \begin{pmatrix} 0 & 4 \\ 1 & 0 \end{pmatrix}$$

$\mathbf{A}$  hat die **Eigenvektoren**  $\mathbf{u} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$  und  $\mathbf{v} = \begin{pmatrix} -2 \\ 1 \end{pmatrix}$  mit den **Eigenwerten**  $\lambda = 2$  und  $\mu = -2$ .

Also gilt für das **Spektrum**

$$\sigma(\mathbf{A}) = \{2, -2\}$$

und der **Spektralradius** ist gleich 2.

## Bemerkungen zu Eigenwerten und Eigenvektoren

- Da der Nullvektor natürlich immer auf sich selbst abgebildet wird, **verlangen wir von einem Eigenvektor stets  $\mathbf{x} \neq \mathbf{0}$** .
- In der Ebene beschreiben Eigenvektoren **Fixgeraden** linearer Abbildungen.
- Die durch die Eigenvektoren definierten Geraden werden auf sich selbst abgebildet (als Menge, nicht punktweise).
- Eine punktweise Abbildung wäre eine **Fixpunktgerade**. Dies sind Geraden definiert durch Eigenvektoren zum Eigenwert 1.

## Reelle Eigenwerte müssen nicht existieren

- Eigenwertprobleme betrachtet man **üblicherweise nicht in  $\mathbb{R}$ , sondern in  $\mathbb{C}$** , weil für eine allgemeine reelle Matrix die Existenz von reellen Eigenwerten nicht garantiert ist.
- So hat i. A. **eine Drehung keine Fixgeraden**. Dementsprechend hat die Drehmatrix

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

**nur komplexe Eigenwerte** (Drehung um  $90^\circ$  bzw.  $\frac{\pi}{2}$ ).

- Allerdings werden wir uns überwiegend mit **reellen symmetrischen Matrizen** befassen, die stets **nur reelle Eigenwerte** haben.

# Eigenwerte symmetrischer Matrizen

## Satz 5.3

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine symmetrische Matrix. Dann gilt:

- (i) Alle *Eigenwerte* von  $\mathbf{A}$  sind *reell*.
- (ii) Es gibt  $n$  *linear unabhängige Eigenvektoren* von  $\mathbf{A}$ .
- (iii) Sind  $\mathbf{u} \in \mathbb{R}^n$  und  $\mathbf{v} \in \mathbb{R}^n$  zwei *linear unabhängige Eigenvektoren* von  $\mathbf{A}$ , so sind  $\mathbf{u}$  und  $\mathbf{v}$  *zueinander orthogonal*.
- (iv) Es existiert eine *orthogonale Matrix*  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  und eine *Diagonalmatrix*  $\mathbf{D} \in \mathbb{R}^{n \times n}$ , so dass gilt:

$$\mathbf{A} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$$

Wie die Matrizen  $\mathbf{Q}$  und  $\mathbf{D}$  aussehen, können wir uns leicht denken. Es sei

$$\mathbf{Q} = \left( \mathbf{q}^{(1)} \quad \mathbf{q}^{(2)} \quad \dots \quad \mathbf{q}^{(n)} \right)$$

wobei die Vektoren  $\mathbf{q}^{(i)}$  linear unabhängige Eigenvektoren von  $\mathbf{A}$  mit  $\|\mathbf{q}^{(i)}\| = 1$  und zugehörigen Eigenwerten  $\lambda_i$  sind.

Dann gilt:

$$\begin{aligned} \mathbf{Q}^T \mathbf{A} \mathbf{Q} &= \mathbf{Q}^T \left( \mathbf{A} \mathbf{q}^{(1)} \quad \mathbf{A} \mathbf{q}^{(2)} \quad \dots \quad \mathbf{A} \mathbf{q}^{(n)} \right) \\ &= \mathbf{Q}^T \left( \lambda_1 \mathbf{q}^{(1)} \quad \lambda_2 \mathbf{q}^{(2)} \quad \dots \quad \lambda_n \mathbf{q}^{(n)} \right) \\ &= \begin{pmatrix} \mathbf{q}^{(1)T} \\ \vdots \\ \mathbf{q}^{(n)T} \end{pmatrix} \left( \lambda_1 \mathbf{q}^{(1)} \quad \lambda_2 \mathbf{q}^{(2)} \quad \dots \quad \lambda_n \mathbf{q}^{(n)} \right) \end{aligned}$$

$$= \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & \lambda_n \end{pmatrix} = \mathbf{D}$$

### Fazit:

- $\mathbf{Q}$  besteht spaltenweise aus den **normierten Eigenvektoren**.
- $\mathbf{D}$  ist eine Diagonalmatrix mit den **Eigenwerten auf der Diagonalen**.



## Beispiel 5.4

Es sei

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$


$\mathbf{A}$  hat die Eigenvektoren

$$\mathbf{q}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{und} \quad \mathbf{q}^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

mit den Eigenwerten  $\lambda_1 = 4$  und  $\lambda_2 = -2$ .

Also:

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 4 & 0 \\ 0 & -2 \end{pmatrix}$$

Probe: Tafel .

## Analytische Berechnung der Eigenwerte und -vektoren

Es gilt:

$$\mathbf{Ax} = \lambda \mathbf{x} \quad \Leftrightarrow \quad \mathbf{Ax} - \lambda \mathbf{x} = \mathbf{0} \quad \Leftrightarrow \quad (\mathbf{A} - \lambda \mathbf{E}) \mathbf{x} = \mathbf{0}$$

Wegen  $\mathbf{x} \neq \mathbf{0}$  hat die rechte Gleichung genau dann eine Lösung  $\neq \mathbf{0}$ , wenn die Matrix  $\mathbf{A} - \lambda \mathbf{E}$  **singulär** ist.

### Satz 5.5

$\lambda \in \mathbb{C}$  ist genau dann Eigenwert der Matrix  $\mathbf{A} \in \mathbb{C}^{n \times n}$ , wenn gilt:

$$\det(\mathbf{A} - \lambda \mathbf{E}) = 0$$

### Definition 5.6

Für  $\mathbf{A} \in \mathbb{C}^{n \times n}$  heißt

$$P(\lambda) := \det(\mathbf{A} - \lambda \mathbf{E})$$

**charakteristisches Polynom** von  $\mathbf{A}$ .

Die Eigenwerte sind also die **Nullstellen des charakteristischen Polynoms** einer Matrix.

### Beispiel 5.7

Sei

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

Damit erhalten wir:

$$\begin{aligned} P(\lambda) = \det(\mathbf{A} - \lambda \mathbf{E}) &= \det \begin{pmatrix} 1 - \lambda & 3 \\ 3 & 1 - \lambda \end{pmatrix} \\ &= (1 - \lambda)^2 - 9 \\ &= \lambda^2 - 2\lambda - 8 \end{aligned}$$

Damit folgt:

$$\lambda_{1,2} = 1 \pm \sqrt{1 + 8} = 4 \text{ bzw. } -2$$

## Fortsetzung Beispiel.

- Aus den Eigenwerten können wir nun Eigenvektoren berechnen.
- Dazu setzen wir einen Eigenwert  $\lambda$  einfach in das lineare Gleichungssystem  $\mathbf{Ax} = \lambda\mathbf{x}$  ein.
- Jede Lösung ( $\neq \mathbf{0}$ ) dieses Gleichungssystems ist ein Eigenvektor (zum Eigenwert  $\lambda$ ).
- Da  $\mathbf{A} - \lambda\mathbf{E}$  singularär ist, haben wir **freie Variablen**.

Für  $\lambda = 4$  lautet die erste Zeile von  $\mathbf{Ax} = \lambda\mathbf{x}$ :

$$x_1 + 3x_2 = 4x_1$$

woraus  $x_2 = x_1$  folgt. Analog erhalten wir für  $\lambda = -2$ :

$$x_1 + 3x_2 = -2x_1$$

woraus  $x_2 = -x_1$  folgt.

### Fortsetzung Beispiel.

$x_1$  ist unsere freie Variable. Wir setzen  $x_1 = 1$  und erhalten damit die bekannten Eigenvektoren

$$\mathbf{q}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{und} \quad \mathbf{q}^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

## Beispiel 5.8


Wir betrachten die Matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 6 \\ 1 & 0 \end{pmatrix}$$

Als Eigenwerte erhalten wir  $\lambda = 3$  und  $\mu = -2$ .

Zugehörige Eigenvektoren sind z. B.

$$\mathbf{u} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \text{ für Eigenwert } \lambda \text{ und } \mathbf{v} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \text{ für Eigenwert } \mu.$$

Herleitung: Tafel 

Bemerkung:  $\mathbf{u}$  und  $\mathbf{v}$  sind nicht orthogonal, da  $\mathbf{A}$  nicht symmetrisch ist.

## Was bringen uns Eigenwerte?

Wir können bspw. die  $n$ -fache Anwendung einer linearen Abbildung **deutlich beschleunigen**.

### Beispiel 5.9

Wo landet ein Punkt  $\mathbf{p} \in \mathbb{R}^2$ , wenn wir ihn  $n$ -mal mit der Matrix  $\mathbf{A}$  aus Beispiel 5.8 abbilden?

Bei  $\mathbf{p}' := \mathbf{A}^n \mathbf{p}$ .

**Naive Berechnung:**  $n$ -fache Matrix-Vektor-Multiplikation.

$$\begin{aligned}\mathbf{p}^{(0)} &= \mathbf{p} \\ \mathbf{p}^{(i+1)} &= \mathbf{A}\mathbf{p}^{(i)}\end{aligned}$$

Damit gilt:  $\mathbf{p}' = \mathbf{A}^n \mathbf{p} = \mathbf{p}^{(n)}$ .

**Aufwand:**  $O(n)$

## Fortsetzung Beispiel.

Wir nehmen nun an, dass wir  $\mathbf{p}$  als Linearkombination der Eigenvektoren  $\mathbf{u}$  und  $\mathbf{v}$  darstellen können, also

$$\mathbf{p} = \alpha \mathbf{u} + \beta \mathbf{v}$$

Damit ergibt sich

$$\begin{aligned} \mathbf{p}' = \mathbf{A}^n \mathbf{p} &= \mathbf{A}^n (\alpha \mathbf{u} + \beta \mathbf{v}) \\ &= \alpha \mathbf{A}^n \mathbf{u} + \beta \mathbf{A}^n \mathbf{v} \\ &= \alpha \lambda^n \mathbf{u} + \beta \mu^n \mathbf{v} \\ &= \alpha 3^n \mathbf{u} + \beta (-2)^n \mathbf{v} \end{aligned}$$

**Aufwand:**  $O(1)$  (wenn die Exponentenoperation Aufwand  $O(1)$  haben)



## Fortsetzung Beispiel.

Aber wie bestimmen wir  $\alpha$  und  $\beta$ ?

Als eindeutige Lösung des LGS

$$\alpha \mathbf{u} + \beta \mathbf{v} = \mathbf{p}$$

also

$$\begin{aligned} 3\alpha - 2\beta &= p_1 \\ \alpha + \beta &= p_2 \end{aligned}$$

Allgemeine Lösung:

$$\alpha = \frac{p_1 + 2p_2}{5}, \quad \beta = \frac{3p_2 - p_1}{5}$$

Aufwand ist ebenfalls  $O(1)$ .

**Fazit:** Für jeden Vektor  $\mathbf{p} \in \mathbb{R}^2$  können wir  $\mathbf{A}^n \mathbf{p}$  in Zeit  $O(1)$  berechnen.

# Anwendungsbeispiele

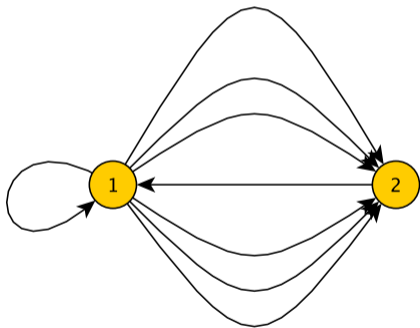
Natürlich geht es nicht nur um schnellere und numerisch stabilere Berechnungen. Eigenwerte spielen auch bei Analysen eine wichtige Rolle.

So können wir mithilfe von Eigenwerten:

- aus einer Adjazenzmatrix Formeln für die Anzahl an Kantenzügen zwischen zwei Knoten in einem Graphen herleiten,
- Wachstumsprozesse analysieren und die damit verbundenen Differenzgleichungen lösen und
- stochastische Systeme analysieren und z. B. Zustandsverteilungen oder stabile Endzustände ermitteln.

# Anzahl Kantenzüge in gerichteten Graphen

## Beispiel 5.10



$$\mathbf{A} = \begin{pmatrix} 1 & 6 \\ 1 & 0 \end{pmatrix}$$

$$\mathbf{a}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{3}{5} \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \frac{2}{5} \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

$$\mathbf{a}^{(2)} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} = \frac{6}{5} \begin{pmatrix} 3 \\ 1 \end{pmatrix} - \frac{6}{5} \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

## Fortsetzung Beispiel.

- Aus der Graphentheorie: Die **Anzahl der Kantenzüge der Länge  $k$**  können wir in der Matrix  $\mathbf{A}^k = \left( a_{i,j}^{(k)} \right)$  ablesen.
- $a_{i,j}^{(k)}$  ist die Anzahl der Kantenzüge der Länge  $k$  von Knoten  $i$  zu Knoten  $j$ .

Berechnung:

$$\begin{aligned}
 \mathbf{A}^k &= \left( \mathbf{A}^{k-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \mathbf{A}^{k-1} \begin{pmatrix} 6 \\ 0 \end{pmatrix} \right) \\
 &= \left( \frac{3}{5} 3^{k-1} \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \frac{2}{5} (-2)^{k-1} \begin{pmatrix} -2 \\ 1 \end{pmatrix} \frac{6}{5} 3^{k-1} \begin{pmatrix} 3 \\ 1 \end{pmatrix} - \frac{6}{5} (-2)^{k-1} \begin{pmatrix} -2 \\ 1 \end{pmatrix} \right) \\
 &= \frac{1}{5} \begin{pmatrix} 3^{k+1} + (-1)^k \cdot 2^{k+1} & 2 \cdot 3^{k+1} - 3 \cdot (-1)^k \cdot 2^{k+1} \\ 3^k + (-1)^{k-1} \cdot 2^k & 2 \cdot 3^k - 3 \cdot (-1)^{k-1} \cdot 2^k \end{pmatrix}
 \end{aligned}$$

# Wachstumsprozesse und Differenzengleichungen

Den bekanntesten Wachstumsprozess beschrieb **Leonardo Fibonacci** im Jahr 1202:

- In der 1. Zeitperiode haben wir ein (neugeborenes) Kaninchenpaar.
- Jedes Kaninchenpaar benötigt zwei Zeitperioden bis zur Geschlechtsreife.
- Jedes geschlechtsreife Kaninchenpaar hat in einer Periode genau ein junges Kaninchenpaar als Nachkommen.
- Kaninchen sterben nicht.

**Mathematisches Modell** für die Anzahl  $f_n$  der **Kaninchenpaare in Zeitperiode  $n$** :

$$f_1 = f_2 = 1, f_n = f_{n-1} + f_{n-2} \text{ für } n \geq 3$$

Allgemeines Ziel für solche Wachstumsmodelle:

- aus der rekursiven Formulierung Aussagen über die Größenordnung des Wachstums ermitteln
- Idealfall: explizite Formel für die  $f_n$

Im Fall der Fibonacci-Zahlen kennen wir die Antwort.

### Satz 5.11 (Formel von Moivre-Binet)

Für die durch  $f_0 = 0$ ,  $f_1 = 1$  und  $f_n = f_{n-1} + f_{n-2}$  (für  $n \geq 2$ ) rekursiv definierte Fibonacci-Folge gilt:

$$f_n = \frac{1}{\sqrt{5}} \left( \left( \frac{1 + \sqrt{5}}{2} \right)^n - \left( \frac{1 - \sqrt{5}}{2} \right)^n \right)$$

## Bezug zu Eigenwerten

$$\begin{aligned}\begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{n-1} \\ f_{n-2} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_{n-2} \\ f_{n-3} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} f_1 \\ f_0 \end{pmatrix}\end{aligned}$$

Herleitung der Formel von Moivre-Binet:

- Eigenwerte und -vektoren der Matrix bestimmen.
- Startvektor als Linearkombination von Eigenvektoren darstellen.
- Daraus ergibt sich eine explizite Formel für die Fibonacci-Zahlen.

# KFZ-Schadenfreiheitsklassen

- Versicherungsprämien der KFZ-Versicherung richten sich nach dem Risiko
- Bester Indikator für das Risiko: Anzahl der Schadensfälle in der Vergangenheit
- Systems der **Schadenfreiheitsklassen**
- Wir gehen von einer Menge  $\{K_1, \dots, K_n\}$  solcher Klassen und Übergangsregeln zwischen diesen Klassen aus.
- $K_1$  sei dabei die Klasse für Neueinsteiger.



## Beispiel 5.12

Wir gehen von vier Klassen aus:

	$K_1$	$K_2$	$K_3$	$K_4$
Rabatt	0	10	20	40

Übergangsregeln:

- Kein Schadensfall: Im Folgejahr eine Klasse höher (oder in  $K_4$  bleiben).
- Ein Schadensfall: Im Folgejahr eine Klasse zurück (oder in  $K_1$  bleiben).
- Mehr als ein Schadensfall: Im Folgejahr zurück in Klasse  $K_1$ .

## Fortsetzung Beispiel.

Es sei  $p_{i,j}$  die Wahrscheinlichkeit dafür, dass ein Kunde, der dieses Jahr in Klasse  $j$  ist, im nächsten Jahr in Klasse  $i$  ist und es sei

$$\mathbf{P} = (p_{i,j}) \in \mathbb{R}^{4 \times 4}$$

Weiterhin beschreibe  $\mathbf{x} \in \mathbb{R}^4$  die Anzahl der Kunden in den vier Schadenfreiheitsklassen. Dann beschreibt

$$\mathbf{P}^k \mathbf{x}$$

die Anzahl der Kunden in den Schadenfreiheitsklassen in  $k$  Jahren.

Interessant ist insbesondere wie sich dies langfristig verhält:

$$\lim_{k \rightarrow \infty} \mathbf{P}^k \mathbf{x}$$

## Fortsetzung Beispiel.

Es sei

$$\mathbf{P} = \begin{pmatrix} 0.15 & 0.15 & 0.05 & 0.05 \\ 0.85 & 0.00 & 0.10 & 0.00 \\ 0.00 & 0.85 & 0.00 & 0.10 \\ 0.00 & 0.00 & 0.85 & 0.85 \end{pmatrix} \quad \text{und} \quad \mathbf{x} = \begin{pmatrix} 0.3 \\ 0.2 \\ 0.1 \\ 0.4 \end{pmatrix}$$

Wir untersuchen  $\mathbf{x}^{(k)} := \mathbf{P}^k \mathbf{x}$ :

	$x_1$	$x_2$	$x_3$	$x_4$
$\mathbf{x}^{(0)}$	0.3	0.2	0.1	0.4
$\mathbf{x}^{(1)}$	0.1	0.265	0.21	0.425
$\mathbf{x}^{(2)}$	0.087	0.106	0.268	0.540
$\mathbf{x}^{(3)}$	0.069	0.100	0.144	0.686
$\mathbf{x}^{(5)}$	0.064	0.072	0.133	0.731
$\mathbf{x}^{(10)}$	0.063	0.067	0.131	0.740
$\mathbf{x}^{(20)}$	0.063	0.066	0.131	0.740

# Gerschgorin-Kreise

## Definition 5.13

Sei  $\mathbf{A} \in \mathbb{C}^{n \times n}$ . Die Mengen

$$K_j := \left\{ z \in \mathbb{C} \mid |z - a_{j,j}| \leq \sum_{k=1, k \neq j}^n |a_{j,k}| \right\}$$

heißen **Gerschgorin-Kreise**.

Gerschgorin-Kreise sind Kreise in  $\mathbb{C}$  mit

- dem Diagonalelement  $a_{j,j}$  als **Mittelpunkt** und
- einem **Radius** so groß wie die  $j$ -te Zeilenbetragssumme minus Diagonalelement.

# Lokalisierung von Eigenwerten

## Satz 5.14

Sei  $\mathbf{A} \in \mathbb{C}^{n \times n}$  mit den Gerschgorin-Kreisen  $K_j, j = 1, \dots, n$ . Dann gilt:

(i)

$$\sigma(\mathbf{A}) \subseteq \bigcup_{j=1}^n K_j$$

Die Eigenwerte von  $\mathbf{A}$  liegen also in der *Vereinigung der Gerschgorin Kreise*.

(ii) Ist die Vereinigung von  $1 \leq p < n$  Gerschgorin-Kreisen  $K'$  disjunkt zur Vereinigung  $K''$  der anderen Gerschgorin-Kreise, dann liegen in  $K'$  genau  $p$  und in  $K''$  genau  $n - p$  Eigenwerte von  $\mathbf{A}$ .

## Folgerung 5.15

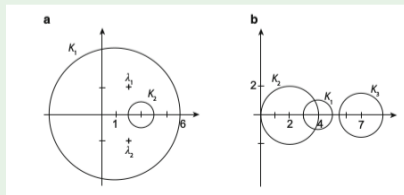
Wenn die Gerschgorin-Kreise  $K_j$  alle disjunkt sind, dann liegt in jedem Kreis  $K_j$  genau ein Eigenwert von  $\mathbf{A}$ .

## Beispiel 5.16

Seien

$$\mathbf{A} = \begin{pmatrix} 1 & 5 \\ -1 & 3 \end{pmatrix} \quad \text{und} \quad \mathbf{B} = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 0.5 & 7 \end{pmatrix}$$

Gerschgorin-Kreise:



## Von-Mises-Vektoriteration

- Bei vielen Anwendungen ist der **betragsgrößte Eigenwert** am wichtigsten.
- Wir gehen davon aus, dass der betragsgrößte Eigenwert **eindeutig** ist, also

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

- Wir starten mit einem Vektor  $\mathbf{x}^{(0)}$  und bilden dann

$$\mathbf{x}^{(1)} = \mathbf{A}\mathbf{x}^{(0)}, \mathbf{x}^{(2)} = \mathbf{A}\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$$

- Wenn  $\mathbf{x}$  eine Linearkombination der Eigenvektoren  $\mathbf{q}^{(i)}$  ist, also

$$\mathbf{x} = \sum_{i=1}^n \alpha_i \mathbf{q}^{(i)}$$

dann erhalten wir

$$\mathbf{x}^{(k+1)} = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{q}^{(i)}$$

- Gilt  $\alpha_1 \neq 0$ , dann wird

$$\mathbf{x}^{(k+1)} = \sum_{i=1}^n \alpha_i \lambda_i^k \mathbf{q}^{(i)}$$

für genügend großes  $k$  durch  $\lambda_1$  dominiert.

- Für einen Vektor  $\mathbf{y}$  mit  $\langle \mathbf{y}, \mathbf{q}^{(1)} \rangle \neq 0$  gilt außerdem

$$\langle \mathbf{x}^{(k+1)}, \mathbf{y} \rangle \approx \lambda_1 \langle \mathbf{x}^{(k)}, \mathbf{y} \rangle$$

für genügend großes  $k$ . Damit folgt

$$\lambda_1 = \lim_{k \rightarrow \infty} \frac{\langle \mathbf{x}^{(k+1)}, \mathbf{y} \rangle}{\langle \mathbf{x}^{(k)}, \mathbf{y} \rangle}$$



- Außerdem konvergiert dann die Folge

$$\mathbf{z}^{(k)} := \psi_k \frac{\mathbf{x}^{(k)}}{\|\mathbf{x}^{(k)}\|}$$

gegen den normierten Eigenvektor  $\mathbf{q}^{(1)}$ .

- Dabei müssen wir  $\psi_k$  so wählen, dass die erste von null verschiedene Komponente von  $\mathbf{z}^{(k)}$  positiv ist.
- Das hier beschriebene Verfahren heißt **Von-Mises-Vektoriteration**.

## Beispiel 5.17

Wir berechnen den größten Eigenwert der Matrix  $\mathbf{B}$  von Beispiel 5.16.

Quelltext siehe Homepage.

Als Ergebnis erhalten wir  $\lambda_1 = 7.1681$  und

$$\mathbf{q}^{(1)} = \begin{pmatrix} 0.0636 \\ 0.2014 \\ 0.9774 \end{pmatrix}$$

## Deflation

Wie kann man **weitere Eigenwerte** einer Matrix  $\mathbf{A}$  berechnen, wenn man einen kennt?

Wir betrachten zunächst eine symmetrische Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  mit bekanntem Eigenwert  $\lambda$  und zugehörigem Eigenvektor  $\mathbf{q}$ .

Sei

$$\mathbf{B} = \mathbf{A} - \frac{\lambda}{\langle \mathbf{q}, \mathbf{q} \rangle} \mathbf{q} \mathbf{q}^T$$

Dann gilt:

$$\mathbf{B} \mathbf{q} = \left( \mathbf{A} - \frac{\lambda}{\langle \mathbf{q}, \mathbf{q} \rangle} \mathbf{q} \mathbf{q}^T \right) \mathbf{q} = \mathbf{A} \mathbf{q} - \frac{\lambda}{\langle \mathbf{q}, \mathbf{q} \rangle} \mathbf{q} \mathbf{q}^T \mathbf{q} = \lambda \mathbf{q} - \lambda \mathbf{q} \frac{\mathbf{q}^T \mathbf{q}}{\langle \mathbf{q}, \mathbf{q} \rangle} = \mathbf{0}$$

Für die Matrix  $\mathbf{B}$  ist also  $\mathbf{q}$  ein **Eigenvektor zum Eigenwert 0**.

Was passiert mit einem Eigenvektor  $\mathbf{r}$  zu einem anderen Eigenwert  $\mu$ ?

$$\mathbf{B}\mathbf{r} = \mathbf{A}\mathbf{r} - \frac{\lambda}{\langle \mathbf{q}, \mathbf{q} \rangle} \mathbf{q}\mathbf{q}^T \mathbf{r} = \mu \mathbf{r}$$

wegen  $\mathbf{q}^T \mathbf{r} = 0$ . Die **anderen Eigenwerte bleiben also erhalten**.

Durch Untersuchung der Matrix  $\mathbf{B}$  können wir somit weitere Eigenwerte berechnen.

Dieses Verfahren nennt man **Deflation**.

## Beispiel 5.18

Die Matrix

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

hat die Eigenwerte  $\lambda_1 = 2$ ,  $\lambda_2 = 2 - \sqrt{2}$ ,  $\lambda_3 = 2 + \sqrt{2}$  mit den Eigenvektoren

$$\mathbf{q}^{(1)} = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \quad \mathbf{q}^{(2)} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ \frac{1}{2} \end{pmatrix}, \quad \mathbf{q}^{(3)} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{2} \end{pmatrix}$$

Die Matrix

$$\mathbf{B} = \mathbf{A} - \frac{\lambda_1}{\langle \mathbf{q}^{(1)}, \mathbf{q}^{(1)} \rangle} \mathbf{q}^{(1)} \mathbf{q}^{(1)T} = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

hat die Eigenwerte  $\lambda_1 = 0$ ,  $\lambda_2 = 2 - \sqrt{2}$ ,  $\lambda_3 = 2 + \sqrt{2}$  mit den gleichen Eigenvektoren wie  $\mathbf{A}$ .

# Ähnliche Matrizen

## Definition 5.19

Zwei Matrizen  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  heißen ähnlich, wenn eine reguläre Matrix  $\mathbf{S} \in \mathbb{R}^{n \times n}$  existiert mit

$$\mathbf{B} = \mathbf{S}^{-1} \mathbf{A} \mathbf{S}.$$

## Beispiel 5.20

Die Matrizen

$$\mathbf{A} = \begin{pmatrix} -3 & 2 \\ -1 & 0 \end{pmatrix} \quad \text{und} \quad \mathbf{B} = \begin{pmatrix} 2 & 3 \\ -4 & -5 \end{pmatrix}$$

sind ähnlich, denn mit

$$\mathbf{S} = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \quad \text{ergibt sich} \quad \mathbf{S}^{-1} = \begin{pmatrix} 2 & -1 \\ -3 & 2 \end{pmatrix}$$

und damit

$$\mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \begin{pmatrix} 2 & -1 \\ -3 & 2 \end{pmatrix} \begin{pmatrix} -3 & 2 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ -4 & -5 \end{pmatrix} = \mathbf{B}.$$

# Ähnliche Matrizen und Eigenwerte

## Satz 5.21

Zwei Matrizen  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , die zueinander ähnlich sind, haben die gleichen Eigenwerte.

## Beweis.

$$\begin{aligned}\det(\mathbf{B} - \lambda \mathbf{E}) &= \det(\mathbf{S}^{-1} \mathbf{A} \mathbf{S} - \lambda \mathbf{E}) \\ &= \det(\mathbf{S}^{-1} \mathbf{A} \mathbf{S} - \mathbf{S}^{-1} \lambda \mathbf{E} \mathbf{S}) \\ &= \det(\mathbf{S}^{-1} (\mathbf{A} - \lambda \mathbf{E}) \mathbf{S}) \\ &= \det(\mathbf{S}^{-1}) \det(\mathbf{A} - \lambda \mathbf{E}) \det(\mathbf{S}) \\ &= \det(\mathbf{S})^{-1} \det(\mathbf{A} - \lambda \mathbf{E}) \det(\mathbf{S}) \\ &= \det(\mathbf{A} - \lambda \mathbf{E})\end{aligned}$$



## Eigenvektoren ähnlicher Matrizen

### Satz 5.22

Es seien  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  ähnliche Matrizen.

Ist  $\mathbf{u} \in \mathbb{R}^n$  Eigenvektor von  $\mathbf{B}$  zum Eigenwert  $\lambda$ , dann ist  $\mathbf{Sx}$  Eigenvektor von  $\mathbf{A}$  zum Eigenwert  $\lambda$ .

### Beweis.

$$\begin{aligned} \mathbf{A}(\mathbf{Sx}) &= \mathbf{SBS}^{-1}(\mathbf{Sx}) \\ &= \mathbf{SBx} \\ &= \lambda \mathbf{Sx} \end{aligned}$$

## QR-Verfahren zur Berechnung der Eigenwerte (1)

Wir wollen mit **einem** Algorithmus alle Eigenwerte einer Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  bestimmen.

### Grundidee:

- Transformiere  $\mathbf{A}$  mit geschickt gewählter orthogonalen Matrix  $\mathbf{Q}$ :

$$\mathbf{A}' = \mathbf{Q}^T \mathbf{A} \mathbf{Q}$$

- Sorge dabei dafür, dass  $\mathbf{A}'$  größere Diagonal- und kleinere Nichtdiagonalelemente hat.
- Wiederhole dies, bis  $\mathbf{A}'$  nahezu diagonal ist.

# Hauptachsentransformation

## Satz 5.23

Sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  und sei  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  eine orthogonale Matrix.

Dann besitzen die Matrizen

$$\mathbf{A} \quad \text{und} \quad \mathbf{Q}^T \mathbf{A} \mathbf{Q}$$

dieselben Eigenwerte.

## QR-Verfahren zur Berechnung der Eigenwerte (2)

- $\mathbf{A}^{(0)} = \mathbf{A}$
- Wiederhole:
  - ▶ Berechne eine QR-Zerlegung von  $\mathbf{A}^{(k)}$ :  $\mathbf{A}^{(k)} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$ .
  - ▶ Setze  $\mathbf{A}^{(k+1)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$ .
  - ▶  $k := k + 1$

bis  $\mathbf{A}^{(k)}$  nahezu eine Diagonalmatrix ist.

**Bemerkung:** Der Schritt  $\mathbf{A}^{(k+1)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)}$  ist eine Ähnlichkeitstransformation.

$$\begin{aligned}\mathbf{A}^{(k)} &= \mathbf{Q}^{(k)}\mathbf{R}^{(k)} \\ \Rightarrow \mathbf{R}^{(k)} &= \left(\mathbf{Q}^{(k)}\right)^T \mathbf{A}^{(k)} \\ \Rightarrow \mathbf{A}^{(k+1)} &= \mathbf{R}^{(k)}\mathbf{Q}^{(k)} = \left(\mathbf{Q}^{(k)}\right)^T \mathbf{A}^{(k)}\mathbf{Q}^{(k)}\end{aligned}$$

## Beispiel 5.24

$$\mathbf{A} = \mathbf{A}^{(0)} = \begin{pmatrix} 2 & -1 \\ -1 & 4 \end{pmatrix} = \mathbf{Q}^{(0)}\mathbf{R}^{(0)} \begin{pmatrix} -0.8944 & 0.4472 \\ 0.4472 & 0.8944 \end{pmatrix} \begin{pmatrix} -2.2361 & 2.6833 \\ 0 & 3.1305 \end{pmatrix}$$

$$\mathbf{A}^{(1)} = \mathbf{R}^{(0)}\mathbf{Q}^{(0)} = \mathbf{Q}^{(1)}\mathbf{R}^{(1)} = \begin{pmatrix} -0.9162 & -0.4008 \\ -0.4008 & 0.9162 \end{pmatrix} \begin{pmatrix} -3.4928 & -2.4049 \\ 0 & 2.0041 \end{pmatrix}$$

$$\mathbf{A}^{(2)} = \mathbf{R}^{(1)}\mathbf{Q}^{(1)} = \begin{pmatrix} 4.1639 & -0.8033 \\ -0.8033 & 1.8361 \end{pmatrix}$$

Nach zehn Iterationen entsteht

$$\mathbf{A}^{(10)} = \begin{pmatrix} 4.4142 & -0.0002 \\ -0.0002 & 1.5858 \end{pmatrix}$$

## Fortsetzung Beispiel.

Die Eigenvektoren sind die Spalten der Matrix

$$\mathbf{Q} = \mathbf{Q}^{(0)} \cdot \mathbf{Q}^{(1)} \cdot \mathbf{Q}^{(2)} \dots$$

Nach zehn Iterationen ergibt sich

$$\mathbf{Q}^{(10)} = \begin{pmatrix} 0.3828 & 0.9238 \\ -0.9238 & 0.3828 \end{pmatrix}.$$

## Konvergenz des QR-Verfahrens

### Satz 5.25

Sei  $\mathbf{A}$  eine symmetrische Matrix und für alle Eigenwerte  $\lambda_i$  ( $i = 1, \dots, n$ ) von  $\mathbf{A}$  gelte

$$|\lambda_i| \neq |\lambda_j| \quad \text{für } 1 \leq i < j \leq n.$$

Dann konvergiert das QR-Verfahren gegen eine Diagonalmatrix.

- insbesondere **keine Konvergenz bei zwei Eigenwerten, die sich nur im Vorzeichen unterscheiden**, also  $\lambda_i = -\lambda_j$
- in praktischen Fällen **auch dann Konvergenz, wenn Eigenwerte mehrfach auftreten**, also  $\lambda_i = \lambda_j$
- Symmetrische und positiv definite Matrizen haben nur positive Eigenwerte.
- Daher können wir **bei diesen Matrizen von einer Konvergenz ausgehen**.

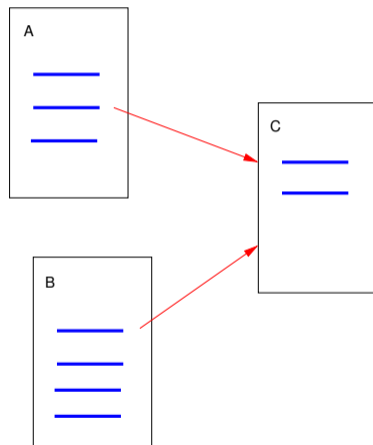
# Effizienz und Konvergenzgeschwindigkeit

- In jedem Iterationsschritt muss eine QR-Zerlegung berechnet werden. Aufwand  $O(n^3)$ .
- Weiterhin müssen zwei Matrixmultiplikation durchgeführt werden. Aufwand  $O(n^3)$ .
- Das Verfahren konvergiert u. U. sehr langsam, insbesondere wenn zwei Eigenwerte existieren für die  $|\lambda_i - \lambda_j|$  klein ist.
- Die Anzahl der notwendigen Iterationen ist typischerweise deutlich größer als  $n$ .
- Es gibt **Techniken, die das QR-Verfahren deutlich beschleunigen**:
  - ▶ Tridiagonalmatrizen: QR-Zerlegungen und Matrixmultiplikationen können damit in Zeit  $O(n^2)$  ausgeführt werden.
  - ▶ Shifts: Die Anzahl der notwendigen Iterationen kann deutlich verringert werden.

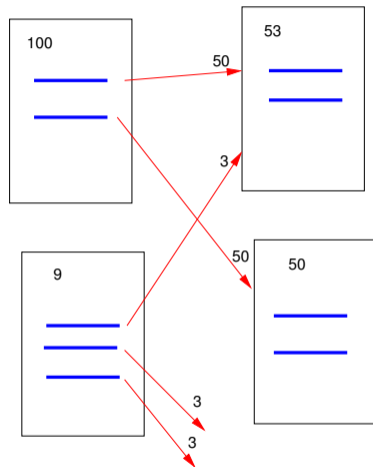


# Page Rank

- Google versucht die Bedeutung von Webseiten mithilfe des sogenannten **Page Rank** zu ermitteln.
- Der Page Rank einer Seite basiert **ausschließlich auf der Verweisstruktur** des Webs.
- Der Inhalt einer Seite hat dagegen keinen direkten Einfluss auf den Page Rank.



- Für eine Seite  $v$  heißt ein Verweis von einer anderen Seite hin zu  $v$  **Backlink**.
- Ein Verweis von  $v$  zu einer anderen Seite heißt **Forward Link**.
- Dokumente, die viele Backlinks haben, werden als wichtig erachtet und sollen eine hohen Page Rank erhalten.
- Über die Forward Links gibt eine Seite wiederum ihren Page Rank an andere Seiten weiter.
- Der **Page Rank verteilt sich dabei gleichmäßig auf die Forward Links**.



# Simplified Page Rank

Die Page Rank Werte sind so zu wählen,

- dass sich der **Page Rank** jeder Seite als **Summe der über die Backlinks erhaltenen Page Rank Werte ergibt** und
- für jede Seite ihr **Page Rank gleichmäßig auf die Forward Links verteilt** wird.

Dies entspricht einem (unterbestimmten) linearen Gleichungssystem.

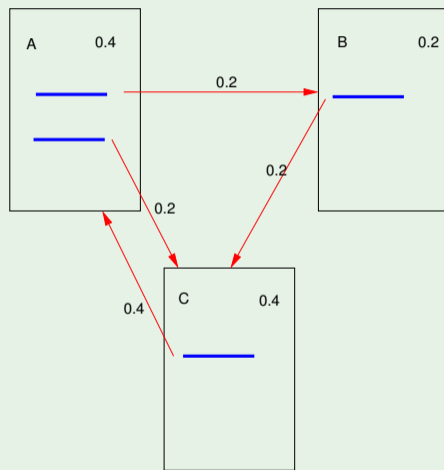
## Beispiel 5.26

Es seien  $a, b, c$  der Page Rank für die Seiten  $A, B, C$ .

$$\begin{aligned} c &= a \\ \frac{1}{2}a &= b \\ \frac{1}{2}a + b &= c \end{aligned}$$

Eine Lösung ist:

$$a = 0.4, b = 0.2, c = 0.4$$



## Definition: Simplified Page Rank

### Definition 5.27

Es sei  $G = (V, E)$  ein gerichteter Graph mit  $n = |V|$ . Dann ist der **Simplified Page Rank** bis auf einen konstanten Faktor  $c$  für einen Knoten  $v \in V$  definiert durch:

$$P(v) = \sum_{(w,v) \in E} \frac{P(w)}{\text{outdeg}(w)}$$

## Page Rank und Eigenvektor

Es sei  $\mathbf{A} \in \mathbb{R}^{n \times n}$  eine quadratische Matrix ( $n = |V|$ ) mit

$$a_{i,j} = \begin{cases} 1/\text{outdeg}(v_j) & \text{falls } (v_i, v_j) \in E \\ 0 & \text{sonst} \end{cases}$$

Weiterhin sei  $\mathbf{p}$  der Vektor der Page Rank Werte für die Knoten  $v_1, v_2, \dots, v_n$ .

Dann entspricht die Gleichung aus Definition 5.27:

$$\mathbf{A}\mathbf{p} = \mathbf{p}$$

Der **Page Rank Vektor** ist also ein **Eigenvektor** zum **Eigenwert 1** der Matrix  $\mathbf{A}$ .

## Lemma 5.28

Für den Spektralradius  $\sigma(\mathbf{A})$  der Matrix  $\mathbf{A}$  gilt:

$$\sigma(\mathbf{A}) \leq 1$$

### Beweis.

- Mithilfe der Gerschgorin-Kreise und Satz 5.14
- Es gilt  $a_{j,j} = 0$  und für die transponierte Matrix

$$\sum_{k=1}^n |a_{j,k}| = 1$$

- Damit folgt  $|z| \leq 1$  für alle  $z$ .

# Berechnung des Simplified Page Rank

- Zur Berechnung nutzen wir **Vektoriteration**.
- Wir starten mit einem beliebigen Vektor  $\mathbf{p} \in \mathbb{R}^n$ , der eine **Wahrscheinlichkeitsverteilung** darstellt, d. h.
  - ▶  $p_i \geq 0$  für  $i = 1, \dots, n$
  - ▶  $\sum_{i=1}^n p_i = 1$
- Unter gewissen Voraussetzungen (diskutieren wir nachfolgend) ist **Konvergenz der Vektoriteration** garantiert.
- Wir berechnen also einfach  $\mathbf{A}^n \mathbf{p}$  für genügend großes  $n$ .



# Stochastische Matrix

## Definition 5.29

Eine Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  heißt **spaltenstochastisch**, wenn jeder Spaltenvektor  $\mathbf{a}^{(j)}$  eine Wahrscheinlichkeitsverteilung darstellt, also:

- $a_{i,j} \geq 0$  für alle  $1 \leq i, j \leq n$  und
- $\sum_{i=1}^n a_{i,j} = 1$  für alle  $1 \leq j \leq n$ .

Wir nennen solch eine Matrix auch **Übergangsmatrix**.

## Folgerung 5.30

*Es sei  $G = (V, E)$  ein gerichteter Graph.*

*Gilt  $\text{outdeg}(v) \geq 1$  für alle  $v \in V$ , dann ist die Matrix  $\mathbf{A}$  für die Berechnung des Simplified Page Rank **spaltenstochastisch**.*

# Berechnungsbeispiel

## Beispiel 5.31

Für den Graphen aus Beispiel 5.26 lautet die Matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & 0 \end{pmatrix}.$$

Wir starten mit  $\mathbf{p} = (1, 0, 0)^T$ .

	$p_1$	$p_2$	$p_3$
1	0	0.5	0.5
2	0.5	0	0.5
5	0.5	0.125	0.375
10	0.4063	0.1875	0.4063
20	0.3994	0.2002	0.4004
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\infty$	0.4	0.2	0.4

# Random Surfer Modell

- Wir können den Page Rank auch mittels eines **Random Walk** auf einem Graphen interpretieren.
- Man startet an einem beliebigen Knoten.
- Wenn man sich an einem Knoten befindet, wählt man **zufällig (und gleichverteilt) eine der ausgehenden Kanten**.
- Der Page Rank entspricht dann der **Grenzverteilung der Besuchswahrscheinlichkeiten** (Wahrscheinlichkeit, dass man sich zum Zeitpunkt  $t$  an Knoten  $v$  befindet) für  $t \rightarrow \infty$ .
- Diese Sichtweise liefert auch einen naiven aber ineffizienten Berechnungsansatz: **Simulation**.

# Konvergenzvoraussetzungen

Aus dem Random Surfer Modell wird deutlich, dass wir nicht für jeden Graphen

- Konvergenz und
- Unabhängigkeit des Grenzwertes von der Startverteilung

erwarten können.

Mögliche Schwierigkeiten:

- Es kann Knoten  $v$  ohne ausgehende Kanten geben, also  $\text{outdeg}(v) = 0$ . Dann ist die Matrix  $\mathbf{A}$  nicht spaltenstochastisch.
- Der gerichtete Graph  $G$  ist nicht stark zusammenhängend.
- Der gerichtete Graph ist nicht aperiodisch.

# Aperiodizität

## Definition 5.32

Es sei  $G = (V, E)$  ein gerichteter Graph.

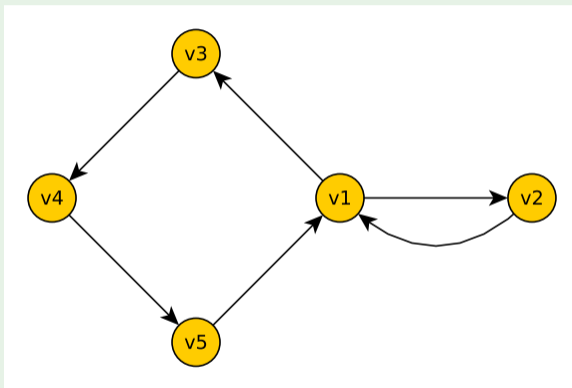
Ein Knoten  $v \in V$  heißt **periodisch** mit Periode  $k > 1$ , wenn  $k$  die größte Zahl ist, so dass alle geschlossenen Kantenzüge mit  $v$  als Start- und Endknoten eine Länge haben, die ein Mehrfaches von  $k$  ist.

Ein Knoten, der nicht periodisch ist (d. h.  $k = 1$ ), heißt **aperiodisch**.

$G$  heißt **aperiodisch**, wenn alle Knoten  $v \in V$  aperiodisch sind.

## Beispiel 5.33

Der folgende Graph  $G = (V, E)$  ist **nicht aperiodisch**:



Es gilt  $k = 2$  für alle Knoten  $v \in V$ .

# Vom Simplified Page Rank zum Page Rank

- Idee: Erweitere den Simplified Page Rank, so dass **stets Konvergenz gegen einen eindeutigen Grenzwert gewährleistet ist**.
- Hierzu erweitert man den gegebenen Graphen zu einem **vollständigen Graphen** (inklusive Schlingen).
- Dabei haben die **originären Kanten eine deutlich höhere Übergangswahrscheinlichkeit** als die neu hinzugefügten.
- Durch den Übergang zu einem vollständigen Graphen
  - ▶ **hat jeder Knoten ausgehende Kanten**,
  - ▶ ist der Graph **stark zusammenhängend** und
  - ▶ **aperiodisch**.

# Mathematische Konstruktion des Page Rank

## Definition 5.34

Es sei  $G = (V, E)$  ein Graph und es sei weiterhin:

- $\tilde{G} = (V, \tilde{E})$  der Graph der entsteht, wenn für in  $G$  für jeden Knoten  $v \in V$  mit  $\text{outdeg}(v) = 0$  zusätzliche Kanten  $(v, w)$  für alle  $w \in V$  einfügen,
- $\mathbf{A}$  die Matrix wie beim Simplified Page Rank für den Graphen  $\tilde{G}$ ,
- $n$  die Anzahl der Knoten von  $G$ ,
- $\mathbf{1}$  eine  $n \times n$  Matrix, deren Komponenten alle 1 sind,
- $d \in (0, 1)$ , der sogenannte **Abschwächungsfaktor (damping factor)**.

Dann ist der **Page Rank** bis auf einen konstanten Faktor  $c$  für einen Knoten  $v \in V$  definiert durch:

$$P(v) = \frac{1-d}{n} + d \sum_{(w,v) \in E} \frac{P(w)}{\text{outdeg}(w)}$$



### Folgerung 5.35

*Der Page Rank entspricht einem Eigenvektor der Matrix:*

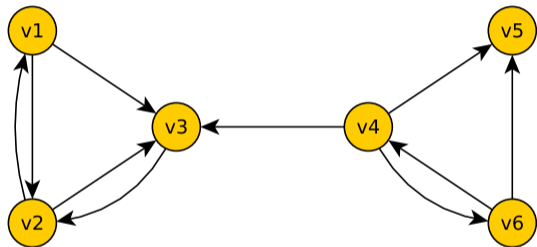
$$(1 - d)\frac{1}{n}\mathbf{1} + d\mathbf{A}$$

*wobei  $\mathbf{A}$  die Simplified Page Rank Matrix des erweiterten Graphen  $\tilde{G}$  ist.*


# Berechnungsbeispiel

## Beispiel 5.36

Wir betrachten den Graphen:



$$\mathbf{A} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{6} & 0 \\ \frac{1}{2} & 0 & 1 & 0 & \frac{1}{6} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{6} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{6} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{6} & 0 \end{pmatrix}$$

Berechnung des Page Rank für  $d = 0.85$  mit Sage .

$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
0.19	0.35	0.28	0.06	0.07	0.05

## Diskussion Page Rank

- In den Originalveröffentlichungen zum Page Rank ist  $d = 0.85$ .
- Die Matrix  $\mathbf{A}$  ist bei großen Graphen in der Regel **dünn besetzt**.
- Daher sollte man bei der Berechnung auch **nicht zu einer gemeinsamen Matrix** für  $(1 - d)\frac{1}{n}\mathbf{1} + d\mathbf{A}$  übergehen.
- Stattdessen:

$$\begin{aligned}\mathbf{q} &= \mathbf{A}\mathbf{p} \\ \mathbf{p} &= \frac{1-d}{n}\mathbf{1}\mathbf{p} + d \cdot \mathbf{q}\end{aligned}$$

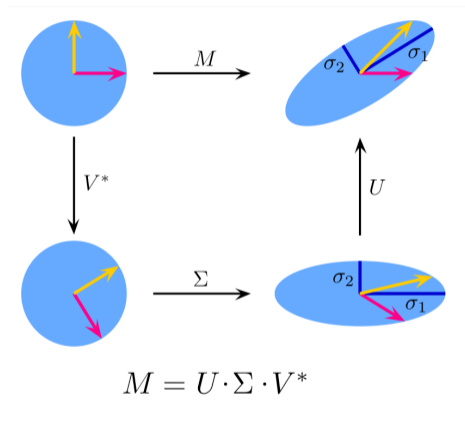
Der Vektor  $\frac{1-d}{n}\mathbf{1}\mathbf{p}$  besteht dabei aus lauter identischen Komponenten und ändert sich nie (warum?), muss also nur einmal ausgewertet werden!

- So kann man den
  - ▶ Speicherplatz für die Matrix kompakt halten und
  - ▶ effiziente SpMV-Verfahren (Sparse matrix-vector multiplication) nutzen.

# Zusammenfassung

- Eigenwerte und Eigenvektoren zur **Analyse von Prozessen oder Graphen**
- Herleitung von Formeln für  $\mathbf{A}^n \mathbf{x}$
- **Gerschgorin-Kreise** zur Lokalisierung von Eigenwerten
- Berechnung des betragsgrößten Eigenwerts mithilfe der **von-Mises-Vektoriteration**
- Berechnung weiterer Eigenwerte mittels **Deflation**
- **Page Rank** zur Bewertung von Knoten in gerichteten Graphen
- Page Rank entspricht **Eigenvektor zum Eigenwert 1** einer **Übergangsmatrix** für einen stark **zusammenhängenden** und **aperiodischen** Graphen.

## Kapitel 6

Hauptkomponentenanalyse und  
Singulärwertzerlegung

# Inhalt

- 6 Hauptkomponentenanalyse und Singulärwertzerlegung
  - Hauptkomponentenanalyse

# Hauptkomponentenanalyse: Idee (1)

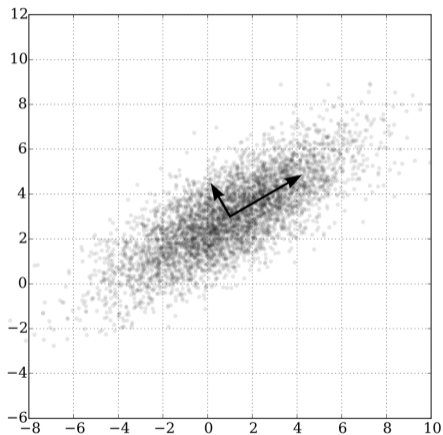
- auch **Principal Component Analysis** oder kurz **PCA**
- Wir haben einen Datensatz mit  $p$  **numerischen Merkmalen** und  $n$  **Beobachtungen**.
- Wir repräsentieren den Datensatz in einer Matrix  $\mathbf{A} \in \mathbb{R}^{n \times p}$ .
- Jede Zeile ist eine Beobachtung und stellt einen Punkt im  $\mathbb{R}^p$  dar.
- Wir wollen nun die Beobachtungen **in einen Raum niedriger Dimension überführen**, also als Matrix  $\mathbf{B} \in \mathbb{R}^{n \times q}$  mit  $q < p$  darstellen.
- Dazu lassen wir aber nicht einfach Merkmale weg, sondern die neuen  $q$  Merkmale sollen jeweils eine geeignete Linearkombination der alten  $p$  Merkmale sein.
- Der Informationsverlust sollte gering sein.
- Wie sollen wir diese Linearkombinationen wählen?

## Hauptkomponentenanalyse: Idee (2)

- Nutze Korrelationen zwischen den Merkmalen!
- Finde die Richtung mit maximaler Varianz im höherdimensionalen Raum und nutze diese Richtung für die Projektion in den Raum niedriger Dimension.
- Die Richtung der maximalen Varianz bildet die 1. Hauptkomponente.
- Die  $k$ -te Hauptkomponente ist orthogonal zu den Hauptkomponenten  $1, \dots, k - 1$ .
- Rechnerisch erhalten wir die Hauptkomponenten als Eigenvektoren der Kovarianzmatrix.
- Die zugehörigen Eigenwerte sagen etwas über den Anteil der Varianz in Richtung der Hauptkomponente in Bezug zur Gesamtvarianz aus.
- Wenn durch die ersten  $q$  Hauptkomponenten ein Großteil der Varianz abgedeckt ist, können wir die Daten mit geringem Informationsverlust in den  $\mathbb{R}^q$  transformieren.



# Beispiel: Hauptkomponenten einer zweidimensionalen Normalverteilung



- Mittelwert  $(1, 3)$ ,
- Standardabweichung circa 3 in Richtung  $(0.866, 0.5)^T$  und 1 in die dazu orthogonale Richtung.
- Die Vektoren sind die Eigenvektoren der Kovarianzmatrix und haben als Länge die Wurzel des zugehörigen Eigenwertes.

# Kovarianz und Korrelationskoeffizient

## Definition 6.1

Es seien  $X$  und  $Y$  zwei Zufallsvariablen, die auf dem gleichen Wahrscheinlichkeitsraum definiert sind. Weiterhin gelte  $E(X) = \mu_X$  und  $E(Y) = \mu_Y$ .

Dann bezeichnet  $\text{Cov}(X, Y)$  die **Kovarianz** von  $X$  und  $Y$ , die definiert ist durch

$$\text{Cov}(X, Y) = E((X - \mu_X)(Y - \mu_Y)).$$

Existieren die Standardabweichungen  $\sigma_X$  und  $\sigma_Y$  für  $X$  bzw.  $Y$  und gilt  $\sigma_X, \sigma_Y > 0$ , dann ist

$$\rho_{X,Y} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

der **Korrelationskoeffizient** von  $X$  und  $Y$ .

## Diskussion: Kovarianz und Korrelationskoeffizient

- Sowohl die Kovarianz als auch der Korrelationskoeffizient messen die **lineare Beziehung zwischen  $X$  und  $Y$** .
- Die Kovarianz  $\text{Cov}(X, Y)$  ist **positiv**, wenn  $X - \mu_X$  und  $Y - \mu_Y$  dazu tendieren, **mit hoher Wahrscheinlichkeit das gleiche Vorzeichen** zu haben.
- Die Kovarianz  $\text{Cov}(X, Y)$  ist negativ, wenn  $X - \mu_X$  und  $Y - \mu_Y$  dazu tendieren, mit hoher Wahrscheinlichkeit verschiedene Vorzeichen zu haben.
- Die Größe von  $\text{Cov}(X, Y)$  ist relativ bedeutungslos, da sie von der Varianz von  $X$  und  $Y$  abhängt.
- Der **Korrelationskoeffizient eliminiert diese individuelle Variabilität**.
- Es gilt

$$-1 \leq \rho_{X,Y} \leq 1.$$

# Stichprobenkovarianz (1)

- Kovarianz und Korrelationskoeffizient sind **theoretische Größen**, deren Definition auf den Verteilungen von  $X$  und  $Y$  basiert.
- In der Datenanalyse kennen wir diese Verteilungen üblicherweise nicht. Wir sehen nur eine **Stichprobe** dieser Verteilungen.
- Wir behelfen uns, indem wir die Kovarianz oder den Korrelationskoeffizienten **schätzen**, z. B. durch die **Stichprobenkovarianz**.

## Stichprobenkovarianz (2)

### Definition 6.2

Ist  $(x_1, y_1), \dots, (x_n, y_n)$  eine Stichprobe zweier Zufallsvariablen  $X$  und  $Y$ , dann ist die **Stichprobenkovarianz**  $s_{x,y}$  definiert durch

$$s_{x,y} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}).$$

mit den arithmetischen Mittelwerten  $\bar{x} = \sum_{i=1}^n x_i$  und  $\bar{y} = \sum_{i=1}^n y_i$ .

## Diskussion: Stichprobenkovarianz

- Es gibt noch die **korrigierte Stichprobenkovarianz**  $\hat{\sigma}_{x,y}$  mit

$$\hat{\sigma}_{x,y} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}).$$

- Die Stichprobenkovarianz entspricht dem **Maximum-Likelihood-Schätzer**

$$S_{X,Y} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

für  $\text{Cov}(X, Y)$ , aber dieser Schätzer ist nicht **erwartungstreu**.

- Konkret gilt

$$E(S_{X,Y}) = \frac{n-1}{n} \text{Cov}(X, Y).$$

- Die **korrigierte Stichprobenkovarianz** ist jedoch **erwartungstreu**.
- In der induktiven Statistik wird daher die korrigierte Stichprobenkovarianz genutzt.
- Die Formel für die **Stichprobenkovarianz** kann verwendet werden, wenn ein **ganzer Datensatz** betrachtet wird und die **Erwartungswerte** bekannt oder **ausreichend gut geschätzt** werden können.
- Insbesondere **bei wenigen Messwerten** sollte man dagegen die Formel für die **korrigierte Stichprobenkovarianz** nutzen.

# Kovarianzmatrix

- Für  $n$  Zufallsvariablen  $X_1, \dots, X_n$  heißt die Matrix  $\mathbf{\Sigma} = (\sigma_{i,j}) \in \mathbb{R}^{n \times n}$  mit

$$\sigma_{i,j} = \text{Cov}(X_i, X_j)$$

## Kovarianzmatrix.

- Für eine Datenanalyse eines Datensatzes nutzen wir die **Stichprobenkovarianzmatrix**  $\mathbf{S} = (s_{i,j}) \in \mathbb{R}^{n \times n}$  mit

$$s_{i,j} = s_{X_i, X_j}.$$

- Die Stichprobenkovarianzmatrix besteht also aus den **Stichprobenkovarianzen der einzelnen Merkmale**.



## Beispiel 6.3

Datensatz:

$x_1$	$x_2$
1.0	1.41
2.0	1.56
2.0	2.19
4.0	2.79
5.0	3.04
6.0	2.23
9.0	3.74
9.0	3.84
9.0	2.80
13.0	4.18

Geschätzte Erwartungswerte:

$$\bar{x}_1 = 6 \quad \text{und} \quad \bar{x}_2 = 2.78$$

Stichprobenkovarianzmatrix:

$$\mathbf{S} = \begin{pmatrix} 13.80 & 2.97 \\ 2.97 & 0.81 \end{pmatrix}$$

## Vorgehen zur Hauptkomponentenanalyse

### 1 Zentriere die Daten

Man subtrahiere von jedem der  $p$  Merkmal den Erwartungswert (geschätzt durch das arithmetische Mittel).

### 2 Optional: Normiere die Daten bzgl. der Varianz

Berechne für jedes Merkmal die Standardabweichung und dividiere die Merkmalswerte durch diese.

### 3 Berechne die Stichprobenkovarianzmatrix $\mathbf{S}$

### 4 Berechne Eigenwerte und zugehörige Eigenvektoren von $\mathbf{S}$

Es seien  $\lambda_1, \dots, \lambda_p$  die Eigenwerte. Dann gibt

$$\frac{\lambda_i}{\sum_{i=1}^p \lambda_i}$$

den Anteil der zugehörigen Hauptkomponente an der Gesamtvarianz an.

5 Sortiere die Eigenwerte absteigend nach ihren Betragswerten

Zusammen mit den zugehörigen Eigenvektoren ergeben sich dadurch die Hauptkomponenten.

Es lohnt sich, hier den Anteil der Hauptkomponente an der Gesamtvarianz auszugeben.

6 Optional: Verwerfe die betragsmäßig kleinsten Eigenwerte und -vektoren und transformiere die Daten

- ▶ Baue aus den verbleibenden Eigenvektoren eine Transformationsmatrix  $\mathbf{W}$ .
- ▶ Generiere durch

$$\mathbf{x}' = \mathbf{W}\mathbf{x}$$

einen neuen Datensatz mit niedriger Dimension. Dabei ist  $\mathbf{x}$  ein Datenpunkt im **alten zentrierten oder normierten Datensatz**.

# Hauptkomponentenanalyse für den IRIS Datensatz

## Klassifikation von Schwertlilien

Attribute:

- Kelchblattlänge
- Kelchblattbreite
- Kronblattlänge
- Kronblattbreite
- Art

Alle Längen und Breiten in cm.

$n = 150$

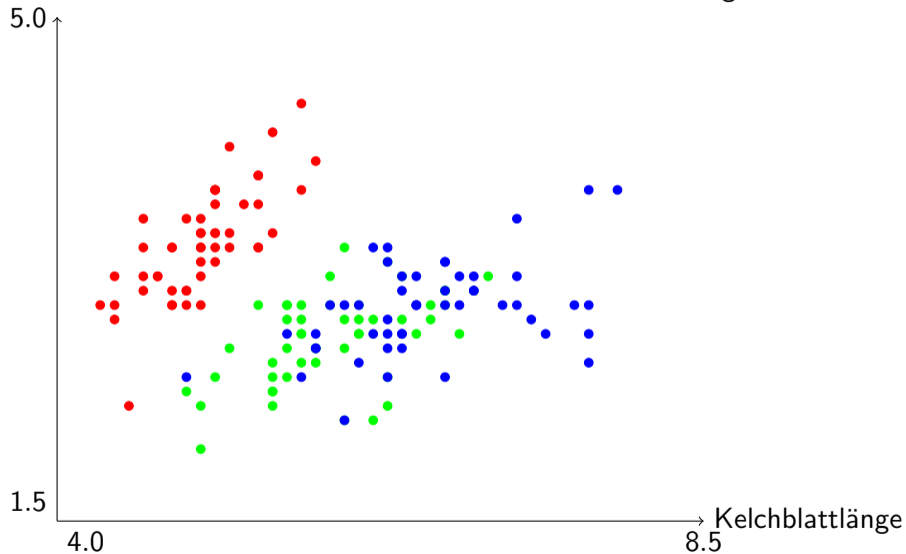
$p = 4$

Auszug aus den Daten:

5.1	3.5	1.4	.2	Setosa
4.9	3	1.4	.2	Setosa
⋮	⋮	⋮	⋮	⋮
7	3.2	4.7	1.4	Versicolor
6.4	3.2	4.5	1.5	Versicolor
⋮	⋮	⋮	⋮	⋮
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica

Kelchblattbreite

• Setosa • Versicolor • • Virginica



Der Datensatz sei als Matrix  $\mathbf{A} \in \mathbb{R}^{n \times p}$  gegeben.

- ① **Mittelwerte** der einzelnen Merkmale:

Kelchblattlänge $\bar{x}_1$	Kelchblattbreite $\bar{x}_2$	Kronblattlänge $\bar{x}_3$	Kronblattbreite $\bar{x}_4$
5.843333	3.057333	3.758000	1.199333

Damit **zentrieren** wir den Datensatz, d. h. wir bilden die Matrix  $\mathbf{B} = (b_{i,j}) \in \mathbb{R}^{n \times p}$  mit

$$b_{i,j} = a_{i,j} - \bar{x}_j.$$

- ② **Keine Normierung** der Daten bzgl. der Varianz.

- 3 Da die Daten nun zentriert sind, nutzen wir konkret die Formel

$$s_{i,j} = \frac{1}{n} \sum_{k=1}^n b_{k,i} b_{k,j}$$

zur Berechnung der Stichprobenkovarianz von Merkmal  $i$  und Merkmal  $j$ .

Damit ergibt sich die Stichprobenkovarianzmatrix

$$\mathbf{S} = \begin{pmatrix} 0.6811 & -0.0422 & 1.2658 & 0.5128 \\ -0.0422 & 0.1887 & -0.3275 & -0.1208 \\ 1.2658 & -0.3275 & 3.0955 & 1.2870 \\ 0.5128 & -0.1208 & 1.2870 & 0.5771 \end{pmatrix}.$$

- ④ Wir berechnen mit dem **QR-Verfahren** die Eigenwerte und Eigenvektoren von **S**.  
Für eine einzelne **QR-Zerlegung** nutzen wir das **Gram-Schmidt-Verfahren**.  
Die **Eigenwerte** sind

$$\lambda_1 = 4.2001, \quad \lambda_2 = 0.2411, \quad \lambda_3 = 0.0777, \quad \lambda_4 = 0.0237.$$

Der Anteil der Hauptkomponenten an der Varianz beträgt

$$92.46\%, \quad 5.31\%, \quad 1.71\%, \quad 0.52\%.$$

Die **Eigenvektoren** und damit die **Hauptkomponenten** sind

1. HK	2. HK	3. HK	4. HK
0.3614	0.6566	-0.5820	0.3155
-0.0845	0.7302	0.5979	-0.3197
0.8567	-0.1734	0.0762	-0.4798
0.3583	-0.0755	0.5458	0.7537



- 5 Die Eigenwerte und die zugehörigen Eigenvektoren sind zufälliger Weise schon richtig sortiert.
- 6 Wir nehmen die ersten beiden Hauptkomponenten ( $q = 2$ ), da sie zusammen fast 98% der Varianz erklären.

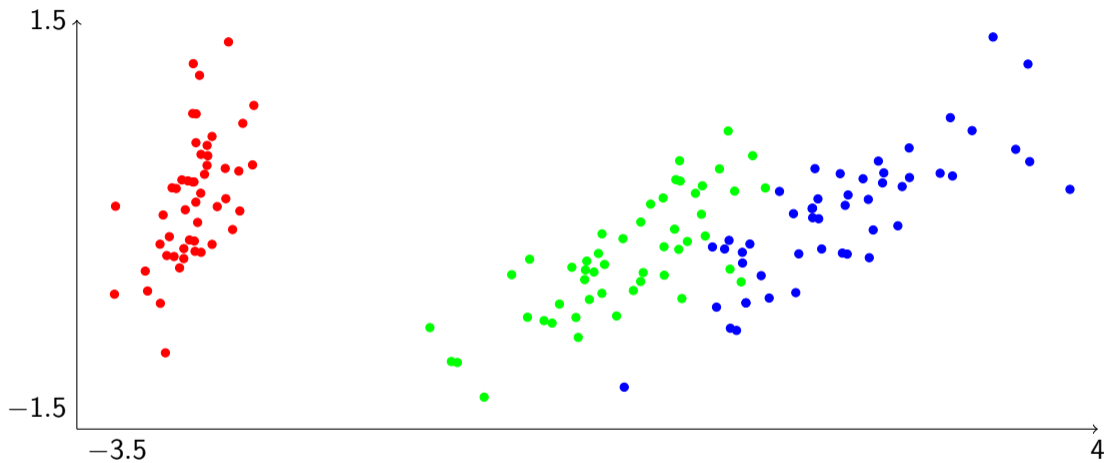
Mit diesen beiden Hauptkomponenten definieren wir die Transformatrix

$$\mathbf{W} = \begin{pmatrix} 0.3614 & 0.6566 \\ -0.0845 & 0.7302 \\ 0.8567 & -0.1734 \\ 0.3583 & -0.0755 \end{pmatrix} \in \mathbb{R}^{p \times q}$$

und bauen mit

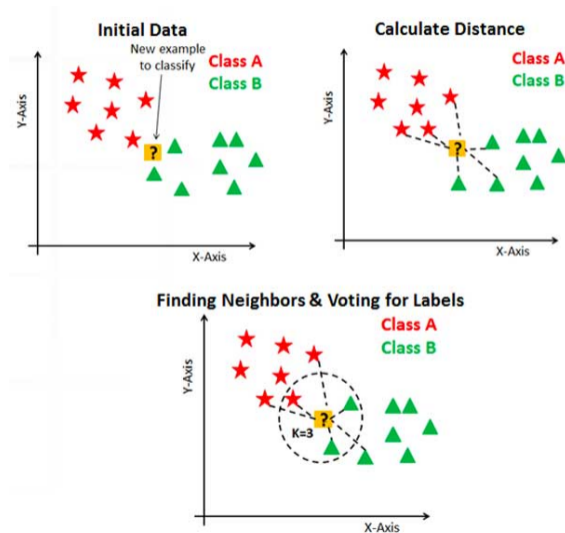
$$\mathbf{C} = \mathbf{B}\mathbf{W}$$

den transformierten Datensatz  $\mathbf{C} \in \mathbb{R}^{n \times q}$ .



## Kapitel 7

## Instanzbasiertes Lernen



# Inhalt

## 7 Instanzbasiertes Lernen

- Lokale gewichtete Regression
- kNN-Verfahren

# Einleitung

- Prognosen basieren auf der Annahme, dass ein funktionaler Zusammenhang zwischen der zu prognostizierenden Größe  $y$  und anderen  $d$ -dimensionalen Variablen  $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d$  besteht.
- Kann der funktionale Zusammenhang geschätzt werden, so kann im Prinzip für jedes  $\mathbf{x}$  der zugehörige Wert  $y$  prognostiziert werden.
- Bei den meisten Anwendungen wird angenommen, dass der **funktionale Zusammenhang einem niedrigdimensionalen parametrischen Modell entspricht** oder durch solch einen Ansatz gut wiedergegeben wird.

## ☞ Kapitel 3

- **Problem:** Der wirkliche funktionale Zusammenhang entspricht u. U. nicht dem gewählten Modell.
- **Folge:** schlechte Prognosen

## Einleitung (2)

- Nichtparametrische Verfahren gehen ebenfalls von einem funktionalen Zusammenhang aus,
- geben aber **kein festes parametrisches Modell** vor.
- Grundidee: Finde für ein gegebenes  $x$  **lokal eine beste Anpassung** an die vorliegenden Daten und prognostiziere damit  $y$ .
- **Vorteil:** Wir benötigen keine Trainingsphase mehr, da keine Parameter eines Modells **vor** der Prognose an die Trainingsbeispiele angepasst werden müssen.  
Die Prognose findet sofort mithilfe der abgespeicherten Trainingsbeispiele statt.  
☞ **Instanzbasiertes Lernen, Lazy Learning**
- **Nachteil:** Wir erhalten keinen einfachen globalen funktionalen Zusammenhang.

## Gewichtete Regression

- Situation wie in Kapitel 3 bei der Ausgleichsrechnung: Wir wollen eine Funktion  $f(x; \lambda_1, \dots, \lambda_m)$  möglichst gut an einen Datensatz

$$(x_1, y_1), \dots, (x_n, y_n)$$

anpassen.

- Die Funktion  $f$  ist eine Linearkombination von Ansatzfunktionen, also

$$f(x; \lambda_1, \dots, \lambda_m) = \sum_{j=1}^m \lambda_j f_j(x).$$

- **Aber:** Für die Anpassung sollen die Abweichungen nicht (wie in Kapitel 3) mit gleichen Anteilen eingehen.

- Vielmehr wird jede Abweichung mit einem Gewicht  $w_i \geq 0$ ,  $i = 1, \dots, n$  gewichtet.
- Damit betrachten wir das Fehlerfunktional

$$E(\lambda_1, \dots, \lambda_m) = \sum_{i=1}^n w_i (y_i - f(x_i; \lambda_1, \dots, \lambda_m))^2.$$

- Um die  $\lambda_i$  zu bestimmen, bilden wir wie üblich die partiellen Ableitungen und setzen diese auf Null:

$$\frac{\partial E(\lambda_1, \dots, \lambda_m)}{\partial \lambda_i} \stackrel{!}{=} 0.$$

- Wie sieht das zugehörige **Normalgleichungssystem** aus?



# Lineare gewichtete Regression

Zur Minimierung von

$$E(a, b) = \sum_{i=1}^n w_i (y_i - (ax_i + b))^2$$

müssen die partiellen Ableitungen gleich 0 sein.

$$\frac{\partial E(a, b)}{\partial a} = -2 \sum_{i=1}^n w_i ((y_i - (ax_i + b))x_i) \stackrel{!}{=} 0$$

$$\frac{\partial E(a, b)}{\partial b} = -2 \sum_{i=1}^n w_i ((y_i - (ax_i + b))) \stackrel{!}{=} 0$$

Umformung ergibt das LGS

$$\begin{aligned} a \sum_{i=1}^n w_i x_i^2 + b \sum_{i=1}^n w_i x_i &= \sum_{i=1}^n w_i x_i y_i \\ a \sum_{i=1}^n w_i x_i + b \sum_{i=1}^n w_i &= \sum_{i=1}^n w_i y_i \end{aligned}$$

bzw.

$$\begin{pmatrix} \sum_{i=1}^n w_i x_i^2 & \sum_{i=1}^n w_i x_i \\ \sum_{i=1}^n w_i x_i & \sum_{i=1}^n w_i \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n w_i x_i y_i \\ \sum_{i=1}^n w_i y_i \end{pmatrix}.$$

# Normalgleichungssystem für die gewichtete Regression

## Satz 7.1

*Das Normgleichungssystem für das gewichtete Regressionsproblem lautet:*

$$\mathbf{A}^T \mathbf{W} \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{W} \mathbf{y}.$$

*Dabei ist  $\mathbf{W}$  eine  $n \times n$ -Diagonalmatrix mit den Gewichten  $w_i$  auf der Diagonalen.*

# Lokale gewichtete Regression

**Gegeben:**  $n$  Beobachtungen  $(x_1, y_1), \dots, (x_n, y_n)$  für die wir einen funktionalen Zusammenhang  $f(x) = y$  vermuten.

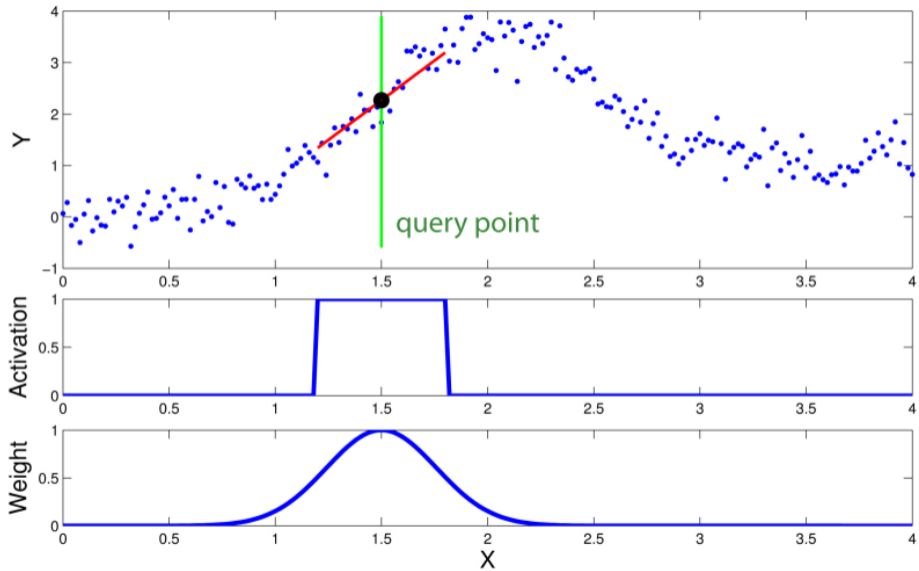
Die Funktion  $f(x)$  ist uns nicht bekannt.

**Gewünscht:** Für einen Anfragewert  $\hat{x}$  hätten wir gerne eine Prognose  $\hat{y} = f(\hat{x})$ .

**Prinzip:** Wir führen eine (lineare) Regression durch, bei der wir die  $x_i$  in Abhängigkeit von  $|\hat{x} - x_i|$  gewichten.

Es sei  $f_{\hat{x}}(x)$  die zugehörige Regressionfunktion.

Dann prognostizieren wir  $\hat{y} = f_{\hat{x}}(\hat{x})$ .



# Gewichtung und Auswahl

Fragen:

- 1 Wie sollen wir die  $x_j$  in Abhängigkeit von  $\hat{x}$  gewichten?  
☞ **Weight**
- 2 Welche  $x_j$  sollen überhaupt für die Prognose genutzt werden?  
☞ **Activation**

Zu 1.:

- Zunächst berechnet man den Abstand  $d(x_j, \hat{x})$ .
- Diesen Abstand setzt man dann in eine Gewichtsfunktion  $K(\cdot)$ , auch **Kernfunktion** genannt, ein.

Zu 2.:

- Üblicherweise berücksichtigt man nur Beobachtungen  $(x_i, y_i)$  für die  $x_i$  nicht zu weit von  $\hat{x}$  entfernt ist.
- Dies drückt man durch eine Indikatorfunktion  $I(\cdot)$  aus, z. B.

$$I_{\hat{x}}(x) = \begin{cases} 1 & \text{für } |x - \hat{x}| \leq 1 \\ 0 & \text{sonst} \end{cases}$$

Die Gewichtung erfolgt dann als Kombination von Kern- und Indikatorfunktion:

$$w_i = K(d(x_i, \hat{x})) \cdot I_{\hat{x}}(x_i).$$

## Typische Kernfunktionen

- **Gaußkern:**  $K(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right)$
- **Cauchy-Kern:**  $K(t) = \frac{1}{\pi(1+t^2)}$
- **Picard-Kern:**  $K(t) = \frac{1}{2} \exp(-|t|)$

Für eine Aussage  $B(t)$  sei die **Indikatorfunktion**  $I(B(t))$  definiert durch

$$I(B(t)) = \begin{cases} 1 & \text{falls } B(t) \text{ wahr ist} \\ 0 & \text{sonst.} \end{cases}$$

- **Epanechnikov-Kern:**  $K(t) = \frac{3}{4}(1 - t^2)I(|t| \leq 1)$ .
- **Tricube-Kern:**  $K(t) = \frac{70}{81}(1 - |t|^3)^3 I(|t| \leq 1)$ .



## Diskussion: Kernfunktionen

- Für  $t$  ist der Abstand  $d(x_i, \hat{x})$  einzusetzen.
- Die Kernfunktionen sind hier als **Wahrscheinlichkeitsdichten** angegeben (Probability Density Function, PDF).
- Wenn wir die Gewichte aber ausschließlich für eine gewichtete Regression nutzen, **spielen die konstanten Faktoren keine Rolle**.
- Beim Gauß-, Cauchy- und Picard-Kern können **prinzipiell alle Beobachtungen in das Ergebnis einfließen**.
- Wir können diese Kernfunktionen aber auch mit einer zusätzlichen Indikatorfunktion kombinieren.
- Beim Epanechnikov- und Tricube-Kern fließen nur die Beobachtungen ein, die von  $\hat{x}$  einen Abstand  $\leq 1$  haben.
- Dieser erlaubte Abstand kann natürlich angepasst werden.

## Diskussion: Lokale gewichtete Regression

- Bei reiner linearer Regression (lokal) werden **Spitzen in den Daten evtl. nicht gut erkannt**.
- Lokale gewichtete Regression ist aber **auch mit Polynom-Modellen möglich**. Damit ist eine bessere Anpassung an Spitzen möglich.
- Wichtig: Es sollten **genügend Daten** vorhanden sein.
- Kern- und Indikatorfunktion müssen gut eingestellt sein, um Über- und Unteranpassung zu vermeiden.
- Mit einer zusätzlichen gewichteten zeitlichen Komponente kann man auch Änderungen in den Daten lernen.

# kNN-Verfahren

- **kNN-Verfahren** können sowohl zur Regression als auch zur Klassifikation eingesetzt werden.
- Sie basieren auf den Ideen der lokal gewichteten Regression.
- Zu einem Anfragepunkt werden die  $k$  nächsten Nachbarn genutzt, um einen Funktionswert zu prognostizieren (Regression) oder das Objekt einer Klasse zuzuordnen (Klassifikation).
- $k = 1$  führt meist zu einer starken Überanpassung, es sollte ein größerer Wert für  $k$  gewählt werden.
- Meistens gehen die  $k$  nächsten Nachbarn gewichtet in das Ergebnis ein. Dann können für die Gewichtung die bekannten Kernfunktionen genutzt werden.

# Regression mit kNN

## Gegeben:

- Beobachtungen  $(\mathbf{x}^{(1)}, y_1), \dots, (\mathbf{x}^{(n)}, y_n)$ . mit  $\mathbf{x}^{(i)} \in \mathbb{R}^d$  und  $y_i \in \mathbb{R}$ .
- Ein Anfragepunkt  $\hat{\mathbf{x}} \in \mathbb{R}^d$ .

**Gesucht:** Ein prognostizierter Funktionswert  $\hat{y} \in \mathbb{R}$  für  $\hat{\mathbf{x}}$ .

- 1 Bestimme die  $k$  nächsten Nachbarn  $\mathbf{x}^{(j_1)}, \dots, \mathbf{x}^{(j_k)}$  des Anfragepunkts  $\hat{\mathbf{x}}$ .
- 2 Ordne jedem der  $k$  nächsten Nachbarn  $\mathbf{x}^{(j_i)}$  ein Gewicht  $w_i$  zu, abhängig von der Entfernung zwischen  $\mathbf{x}^{(j_i)}$  und  $\hat{\mathbf{x}}$ .
- 3 Prognostiziere

$$\hat{y} = \frac{\sum_{i=1}^k w_i y_{j_i}}{\sum_{i=1}^k w_i}$$

als Funktionswert für  $\hat{\mathbf{x}}$ .

# Klassifikation mit kNN

## Gegeben:

- Eine Menge  $\mathcal{K} = \{K_1, \dots, K_r\}$  von Klassen.
- Beobachtungen  $(\mathbf{x}^{(1)}, k_1), \dots, (\mathbf{x}^{(n)}, k_n)$  mit  $\mathbf{x}^{(i)} \in \mathbb{R}^d$  und  $k_i \in \mathcal{K}$ .
- Ein Anfragepunkt  $\hat{\mathbf{x}} \in \mathbb{R}^d$ .

**Gesucht:** Eine Klassenzuordnung  $\hat{k} \in \mathcal{K}$  für  $\hat{\mathbf{x}}$ .

- 1 Bestimme die  $k$  nächsten Nachbarn  $\mathbf{x}^{(j_1)}, \dots, \mathbf{x}^{(j_k)}$  des Anfragepunkts  $\hat{\mathbf{x}}$ .
- 2 Ordne jedem der  $k$  nächsten Nachbarn  $\mathbf{x}^{(j_i)}$  ein Gewicht  $w_i$  zu, abhängig von der Entfernung zwischen  $\mathbf{x}^{(j_i)}$  und  $\hat{\mathbf{x}}$ .

3

$$\hat{k} = \operatorname{argmax}_{r \in \mathcal{K}} \frac{\sum_{i=1}^k w_i \cdot I(\mathbf{x}^{(j_i)} \text{ hat Klasse } r)}{\sum_{i=1}^k w_i}$$

## Suche der $k$ nächsten Nachbarn

- **Problem:** Suche in einem evtl. hochdimensionalem Raum
- Unterscheidung zwischen **Vektorraum** und **metrischem Raum**
- **Vektorraum:** Die Beobachtungen sind Punkte im  $\mathbb{R}^d$ .  
Uns stehen alle Operationen eines Vektorraums zur Verfügung, wir können z. B. **Mittelpunkte** bilden oder die Punktmenge mit einer Hyperebene aufteilen.
- Spezielle Datenstrukturen für die Suche:  $k$ - $d$ -Baum, R-Baum
- **Metrischer Raum:** Wir haben nur die Objekte und eine Abstandsfunktion, aber keine Vektorraumoperationen.
- Es gibt spezielle Datenstrukturen für die Suche in metrischen Räumen, z. B. **M-Tree**.
- **Hauptproblem:** Der Datenraum kann i. d. R. nicht disjunkt aufgeteilt werden.
- Die Effizienz der Suche ist dadurch (teilweise deutlich) geringer als im eindimensionalen Fall (B-Baum, AVL-Baum).

# Zusammenfassung

- Bei instanzbasiertem Lernen wird kein globales Modell gelernt,
- sondern zu einem Anfragepunkt erfolgt eine lokale Anpassung an die vorliegenden Beobachtungen.
- Anpassung mit linearer Funktion oder einem Polynom mit kleinem Grad
- Gewichtung der Beobachtungen mit Kernfunktionen
- einsetzbar für Regressions- und Klassifikationsprobleme