Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(0) Einführung

Prof. Dr. Dirk Ostwald (dirk.ostwald@ovgu.de)

Seit 2021	W2 Professur Methodenlehre I
2014 - 2020	W1 Professur Freie Universität Berlin
2010 - 2014	Postdoc BCCN & MPIB Berlin
2007 - 2010	PhD Psychologie Birmingham
2004 - 2006	MSc Neurowissenschaften Tübingen
2005 - 2012	BSc Mathematik Hagen
2000 - 2003	BSc Medizin Hamburg

Forschung	Komputationale Kognitive Neurowissenschaften
Lehre	Datenwissenschaft

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INSTITUT FÜR PSYCHOLOGIE

Sitemap Impressum Kontakt

Suchbegriff

INSTITUT | STUDIUM | FORSCHUNG | PERSONEN DIREKTLINKS

Home › Institut › Abteilungen des I... › Methodenlehre I : Experimentelle und N... › Forschung | Lehre | CBBS Imaging Platform | Team

Methodenlehre I : Experimentelle und Neurowissenschaftliche Psychologie

Forschung

Lehre

CBBS Imaging Platform

Team

Kontakt

Abteilungsleitung

›Prof. Dr. Dirk Ostwald

dirk.ostwald@ovgu.de

Tel.: + 49 391 67 57370

Abteilungsassistenz

›Birgit Müller

birgit.mueller@ovgu.de

Tel.: +49 391 67 58464

Anschrift

Otto-von-Guericke-Universität
Magdeburg
Institut für Psychologie
Universitätsplatz 2
Gebäude 24
39106 Magdeburg

›Anfahrt

Motivation

Datenwissenschaft

Formalia

Motivation

Datenwissenschaft

Formalia

Multivariate Verfahren

- Einblick in die moderne multivariate Datenanalyse
- Methoden der Statistik und des Maschinellen Lernens

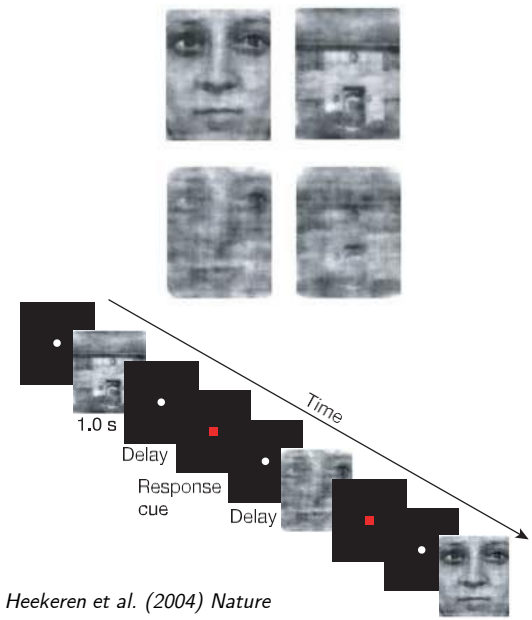
	Unabhängige Variable	Abhängige Variable
Univariate Verfahren	Eindimensional, mehrdimensional	Eindimensional
Multivariate Verfahren	Eindimensional, mehrdimensional	Mehrdimensional

- Grundlagen (Vektoren, Matrizen, Eigenanalyse, Multivariate Normalverteilung)
- Verfahren der Datenreduktion (Hauptkomponentenanalyse, Faktoranalyse)
- Verfahren der frequentistischen Inferenz (ANOVA, Regression, Korrelation)
- Verfahren der Klassifikation (Logistische Regression, SVMs, Neuronale Netze)

Neurobiologische Verarbeitung von Sinnesreizen

Wie werden visuelle Stimuli im Gerhin verarbeitet?

Wie entscheiden Menschen, ob sie ein Haus oder ein Gesicht wahrnehmen?



Heekeren et al. (2004) Nature

→ Allgemeine Psychologie, Biologische Psychologie, Kognitive Neurowissenschaften

Neurobiologische Verarbeitung von Sinnesreizen - Verhaltensdaten

Trial	Stimulus	Kohärenz	Antwort	Reaktionszeit (ms)
1	Gesicht	20	Gesicht	854
2	Gesicht	20	Haus	843
3	Haus	60	Gesicht	369
4	Gesicht	60	Haus	564
5	Gesicht	20	Haus	449
6	Haus	40	Gesicht	565
7	Gesicht	40	Gesicht	715
8	Gesicht	60	Haus	416
9	Gesicht	60	Haus	828
10	Haus	40	Haus	479
11	Haus	60	Gesicht	483
12	Gesicht	20	Gesicht	321

$$\text{Unabhängige Variable} = (\text{Stimulus, Kohärenz})$$

$$\text{Abhängige Variable} = (\text{Antwort, Reaktionszeit})$$

Neurobiologische Verarbeitung von Sinnesreizen - Neurophysiologiedaten

ms	Stimulus	E1 (O1)	E2 (O2)	E3 (Cz)	E4 (Pz)	E5 (AF1)	E6 (AF2)
0	Gesicht	-0.827	-0.063	-2.26	-1.50	-0.537	1.34
2	Gesicht	15.317	16.571	19.52	17.09	17.946	16.54
4	Gesicht	19.121	18.636	17.82	19.54	15.569	17.79
6	Gesicht	2.999	5.590	3.02	3.09	2.656	2.02
8	Gesicht	-14.892	-15.081	-14.07	-13.80	-14.709	-16.20
10	Gesicht	-17.555	-18.604	-19.84	-18.66	-19.557	-19.97
12	Gesicht	-5.477	-5.482	-4.46	-5.01	-5.153	-7.33
14	Gesicht	13.005	11.218	12.90	13.00	14.211	12.46
16	Gesicht	17.877	20.651	18.61	19.96	20.834	19.23
18	Gesicht	7.963	8.011	7.26	8.18	8.194	7.69
20	Gesicht	-11.194	-11.076	-9.82	-8.93	-10.404	-10.67
22	Gesicht	-18.932	-19.981	-19.87	-18.70	-18.327	-19.02
24	Gesicht	-10.662	-10.713	-10.23	-10.01	-11.076	-10.92

$$\text{Unabhängige Variable} = (\text{Stimulus})$$

$$\text{Abhängige Variable} = (\text{E1,E2,E3,E4,E5,E6})$$

Evidenzbasierte Evaluation von Psychotherapieformen bei Depression

Welche Therapieform ist bei Depression wirksamer?

Online Psychotherapie

Klassische Psychotherapie

→ Klinische Psychologie, Klinische Diagnostik

Evidenzbasierte Evaluation von Psychotherapieformen bei Depression

Patient ID	Bedingung	Prae BDI	Post BDI	Prae Glukokortikoide	Post Glukokortikoide
1	Online	25	10	192	197
2	Online	21	30	372	200
3	Online	42	11	209	233
4	Online	26	18	212	150
5	Online	17	20	468	212
6	Online	30	10	339	267
7	Online	22	25	183	299
8	Online	26	24	399	241
9	Online	16	20	191	166
10	Klassisch	21	29	455	117
11	Online	32	10	168	242
12	Online	40	19	233	191

$$\text{Unabhängige Variable} = (\text{Bedingung})$$

$$\text{Abhängige Variable} = (\Delta\text{BDI}, \Delta\text{Glukokortikoide})$$

Approbationsordnung für Psychotherapeutinnen und Psychotherapeuten (2020)

Inhalte, die im Masterstudiengang im Rahmen der hochschulischen Lehre zu vermitteln und bei dem Antrag auf Zulassung zur psychotherapeutischen Prüfung nachzuweisen sind

2. vertiefte Forschungsmethodik

Die studierenden Personen

a) wenden komplexe und multivariate Erhebungs- und Auswertungsmethoden zur Evaluierung und Qualitätssicherung von Interventionen an,

b) nutzen und beurteilen einschlägige Forschungsstudien und deren Ergebnisse für die Psychotherapie

c) planen selbstständig Studien zur Neu- oder Weiterentwicklung der Psychotherapieforschung oder der Forschung in angrenzenden Bereichen, führen solche Studien durch, werten sie aus und fassen sie zusammen, (. . .)

⇒ Masterarbeit

Zur Vermittlung der Inhalte der vertieften Forschnungsmethodik sind bei der Planung der hochschulischen Lehre (. . .) die folgenden Wissensbereiche abzudecken (. . .)

a) multivariate Verfahren und Messtheorie

Motivation

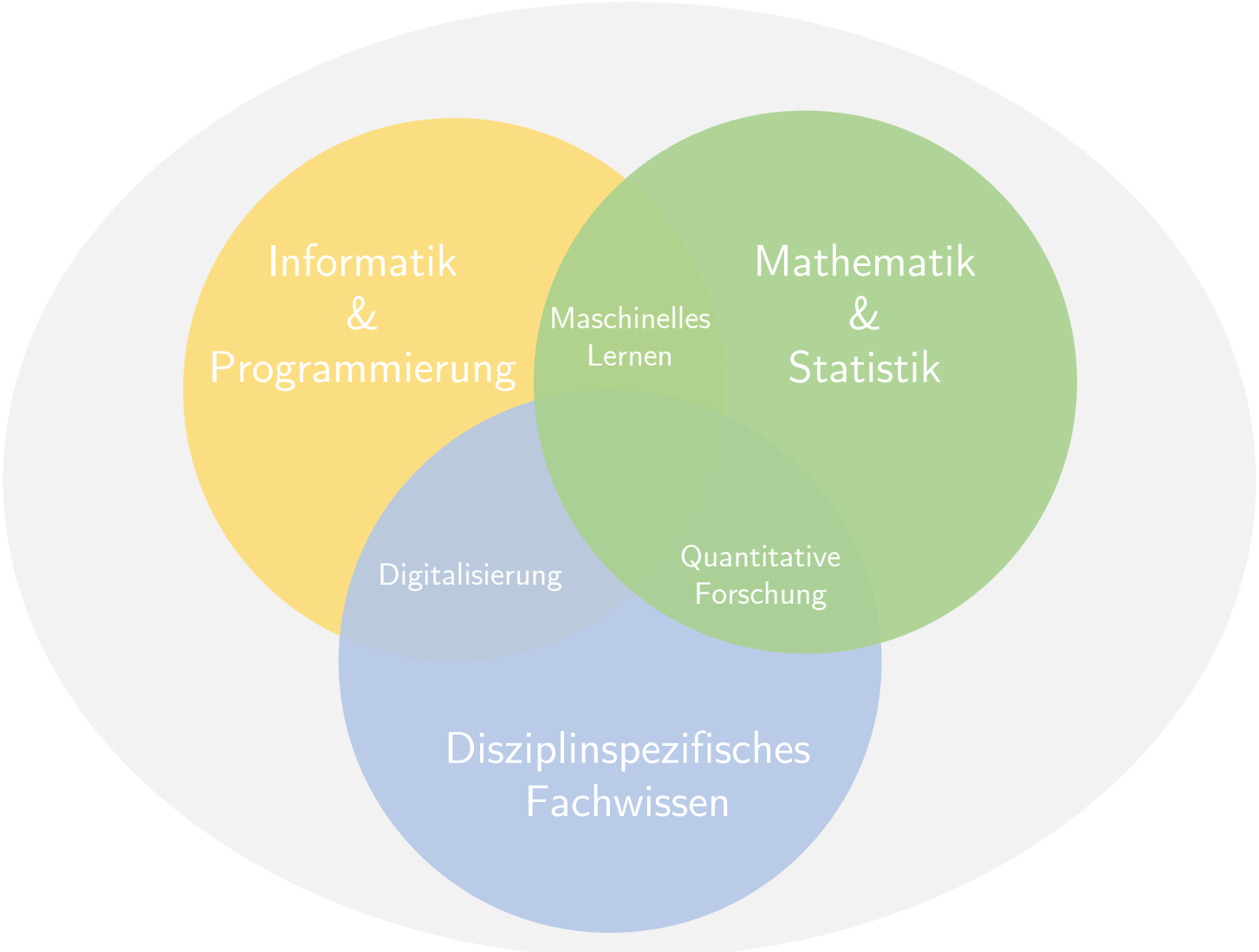
Datenwissenschaft

Formalia

Datenwissenschaft

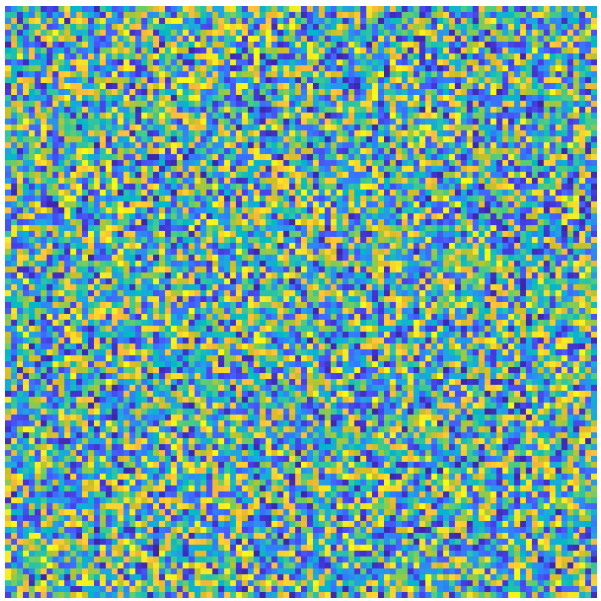
Die Kunst, aus Daten Sinn zu generieren

Datenwissenschaft

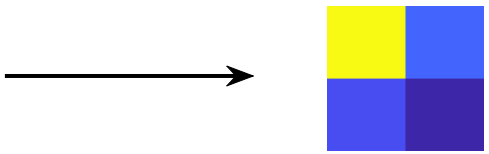


Datenwissenschaft ist Datenreduktion

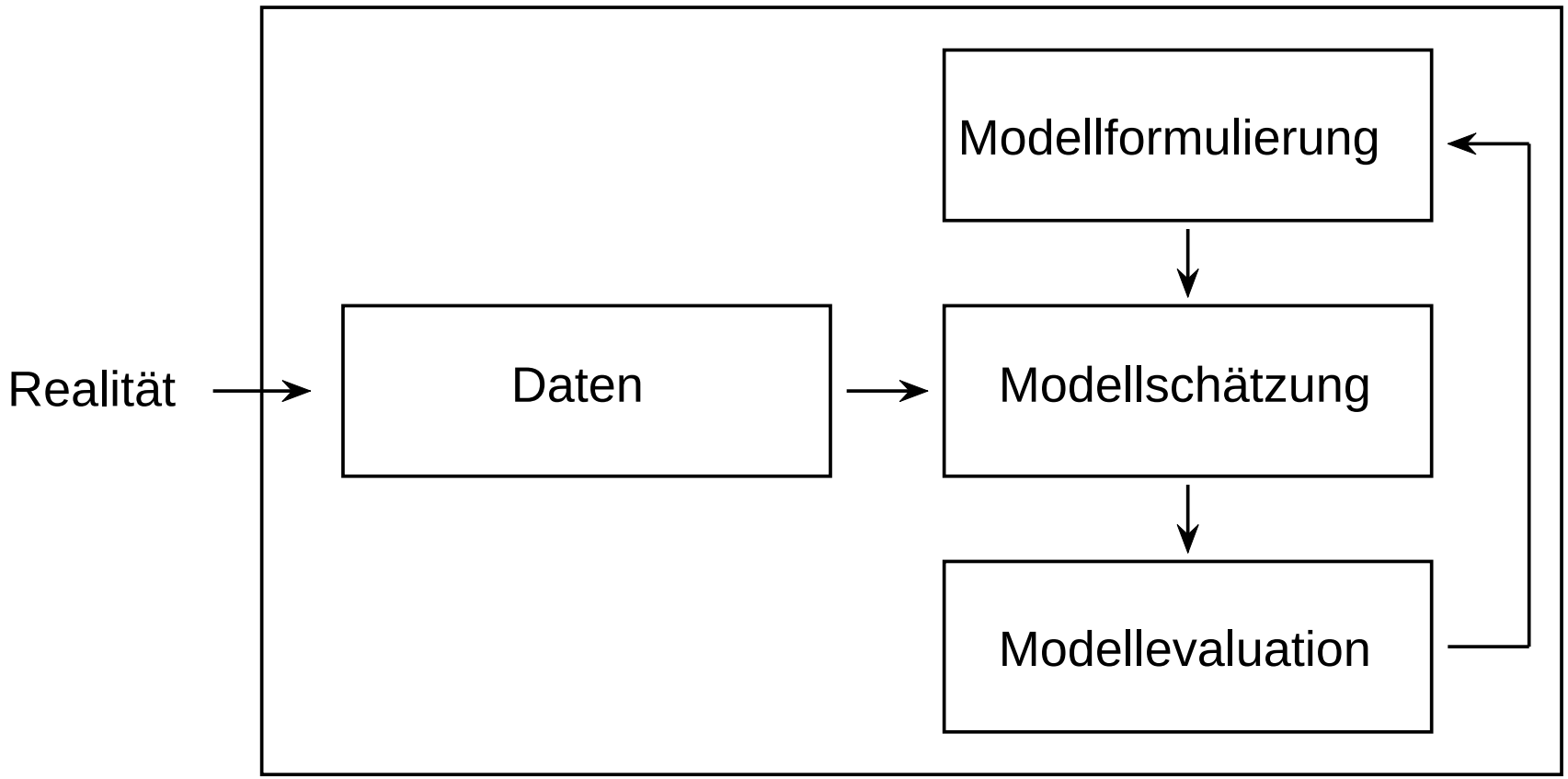
Rohdaten



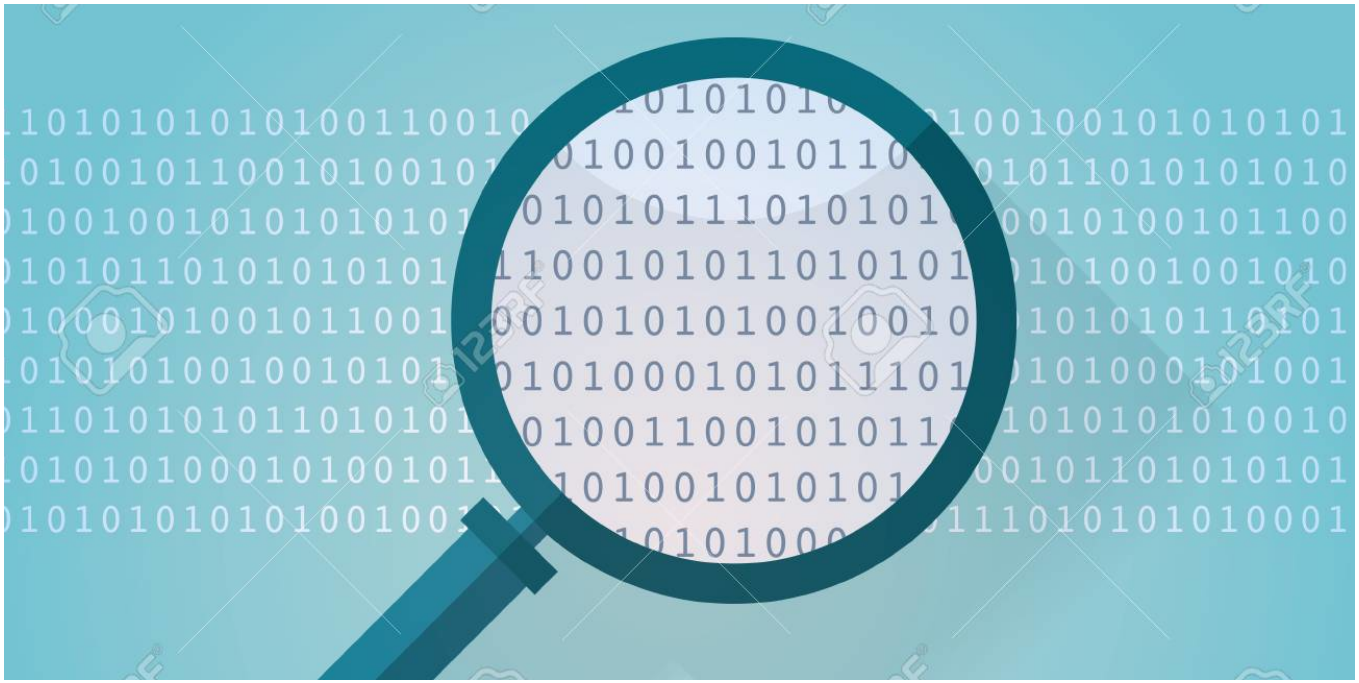
Reduzierte Daten



Datenwissenschaft ist Naturwissenschaft



Datenwissenchaft ist Dateninterpretation



Terminologie der Datenwissenschaft

Statistik = Maschinelles Lernen = Künstliche Intelligenz

Statistik	Maschinelles Lernen	Künstliche Intelligenz
Probabilistische Modelle	Deterministische Modelle	Agenten-basierte Modelle
Theoretische Analyse	Klassifikation	Reinforcement learning
Optimalitätstheorie	Bayesianische Modelle	Symbolik
Asymptotische Theorie	Anwendung	Anwendung
Wissenschaftsphilosophie	Benchmarking	Hype

Datenwissenschaft in der Psychologie

Die Kunst, aus Verhaltens- und Neurophysiologiedaten psychologischen Sinn zu generieren

Motivation

Datenwissenschaft

Formalia

Formalia

Modul A1/A3 Forschungsmethoden: Multivariate Verfahren

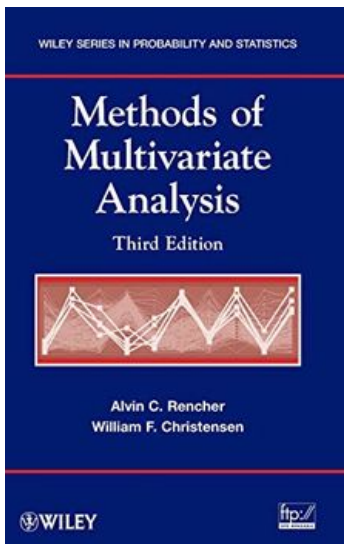
- Freitags 9 - 11 Uhr, G22A-020, 11 - 13 Uhr G40B-238
- Integrierter Kurs zu Theorie und Anwendung in R
- Kursmaterialien (Folien, Videos) auf der Kurswebseite
- Ankündigen über die Moodleseite
- Mathematik Grundlagenkurs im aktuellen BSc Curriculum hier
- R Grundlagenkurs im aktuellen BSc Curriculum und hier
- Benotete Multiple Choice Ende Sommersemester 2022 (?)
- Klausurwiederholungstermin am Ende des Wintersemesters 2022/2023 (?)
- Klausurtermin und Klausurort gemäß Prüfungsplan des FNW Prüfungsamtes

Modul A1/A3 Forschungsmethoden: Multivariate Verfahren

Datum	Einheit	Thema
15.10.2021	Einführung	(0) Einführung
15.10.2021	Grundlagen	(1) Vektoren
22.10.2021	Grundlagen	(2) Matrizen
29.10.2021	Grundlagen	(3) Eigenanalyse
05.11.2021	Grundlagen	(4) Multivariate Normalverteilung
12.11.2021	Achsentransformationen	(5) Hauptkomponentenanalyse
19.11.2021	Achsentransformationen	(6) Faktoranalyse
26.11.2021	Frequentistische Inferenz	(7) T-Tests
03.12.2021	Frequentistische Inferenz	(8) Einfaktorielle Varianzanalyse
10.12.2021	Frequentistische Inferenz	(9) Multivariate Regression
17.12.2021	Frequentistische Inferenz	(10) Kanonische Korrelation
	Weihnachtspause	
07.01.2022	Maschinelles Lernen	(11) Statistische Lerntheorie
14.01.2022	Maschinelles Lernen	(12) LDA und logistische Regression
21.01.2022	Maschinelles Lernen	(13) Support Vektor Maschinen
28.01.2022	Maschinelles Lernen	(14) Neuronale Netze
Feb 2022	Klausurtermin	
Jul 2022	Klausurwiederholungstermin	

Modul A1/A3 Forschungsmethoden: Multivariate Verfahren

- Vorlesungsfolien inklusive Selbstkontrollfragen sind klausurrelevant
- Beispielklausurfragen werden im Januar 2022 bereit gestellt
- Als weiterführende Literatur bietet sich an



Rencher & Christensen (2012) Methods of Multivariate Analysis, 3rd Edition

Q&A

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(1) Vektoren

Motivation

- In der multivariaten Datenanalyse bestehen Datenpunkte aus mehreren Zahlen.
  - Abhängige Variable = (E1,E2,E3,E4,E5,E6) bei EEG-Studie
  - Abhängige Variable = (ΔBDI, ΔGlucorticoids) bei Psychotherapie-Studie.
- Einzelne Datenpunkte aus mehreren Zahlen nennt man Vektoren.
- In der multivariaten Datenanalyse wollen wir mit Vektoren rechnen.
- Vektorraumstrukturen definieren den mathematischen Umgang mit Vektoren.

Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Definition (Vektorraum)

Es seien V eine nichtleere Menge und S eine Menge von Skalaren. Weiterhin sei eine Abbildung

$$+ : V \times V \to V, (v_1, v_2) \mapsto +(v_1, v_2) =: v_1 + v_2, \tag{1}$$

genannt *Vektoraddition*, definiert. Schließlich sei eine Abbildung

$$\cdot : S \times V \to V, (s, v) \mapsto \cdot(s, v) =: sv, \tag{2}$$

genannt *Skalarmultiplikation* definiert. Dann wird das Tupel $(V, S, +, \cdot)$ genau dann *Vektorraum* genannt, wenn für beliebige Elemente $v, w, u \in V$ und $a, b \in S$ folgende Bedingungen gelten:

(1) Kommutativität der Vektoraddition	$v + w = w + v$
(2) Assoziativität der Vektoraddition	$(v + w) + u = v + (w + u)$
(3) Existenz eines neutralen Elements der Vektoraddition	$\exists\, 0 \in V$ mit $v + 0 = 0 + v = v.$
(4) Existenz inverser Elemente der Vektoraddition	$\forall v \in V \exists\ -v \in V$ mit $v + (-v) = 0.$
(5) Existenz eines neutralen Elements der Skalarmultiplikation	$\exists\, 1 \in S$ mit $1 \cdot v = v.$
(6) Assoziativität der Skalarmultiplikation	$a \cdot (b \cdot c) = (a \cdot b) \cdot c.$
(7) Distributivität hinsichtlich der Vektoraddition	$a \cdot (v + w) = a \cdot v + a \cdot w.$
(8) Distributivität hinsichtlich der Skalaraddition	$(a + b) \cdot v = a \cdot v + b \cdot v.$

Reeller Vektorraum

Bemerkungen

- Es gibt viele sehr verschiedene Vektorräume
- Beispiele für Mengen, auf denen eine Vektorraumstruktur definiert werden kann, sind
  - Die Menge der Matrizen
  - Die Menge der Polynome
  - Die Menge der Lösungen eines linearen Gleichungssystems
  - Die Menge der reellen Folgen
  - Die Menge der stetigen Funktionen
- Wir sind hier nur an der Vektorraumstruktur auf den reellen n-Tupeln interssiert
- Zur Erinnerung: die reellen m-Tupel bezeichnen wir mit

$$\mathbb{R}^m := \left\{ \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \mid x_i \in \mathbb{R} \text{ für alle } 1 \leq i \leq m \right\} \tag{3}$$

- Wir sprechen $\mathbb{R}^m$ als “R hoch m” aus.
- Die Elemente $x \in \mathbb{R}^m$ nennen wir *reelle Vektoren* oder einfach *Vektoren*

Theorem (Reeller Vektorraum)

Für alle $x, y \in \mathbb{R}^m$ und $a \in \mathbb{R}$ definieren wir die *Vektoraddition* durch

$$+ : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m, (x, y) \mapsto x + y = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} + \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} := \begin{pmatrix} x_1 + y_1 \\ \vdots \\ x_m + y_m \end{pmatrix} \tag{4}$$

und die *Skalarmultiplikation* durch

$$\cdot : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m, (a, x) \mapsto ax = a \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} := \begin{pmatrix} ax_1 \\ \vdots \\ ax_m \end{pmatrix} \tag{5}$$

Dann bildet $(\mathbb{R}^m, +, \cdot)$ mit den Rechenregeln der Addition und Multiplikation in $\mathbb{R}$ einen Vektorraum, den wir den *reellen Vektorraum* nennen.

Bemerkungen

- Wir verzichten auf einen Beweis.
- Man sagt, dass Vektoraddition und Skalarmultiplikation *komponentenweise* durchgeführt werden.

Beispiele

- Für $x := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}$ und $y := \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ gilt $x + y = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1+2 \\ 2+1 \\ 3+0 \\ 4+1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 5 \end{pmatrix}$.

- Für $x := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ und $y := \begin{pmatrix} 1 \\ 3 \end{pmatrix}$ gilt $x - y = \begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2-1 \\ 3-3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

- Für $x := \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$ und $a := 3$ gilt $ax = 3 \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \cdot 2 \\ 3 \cdot 1 \\ 3 \cdot 3 \end{pmatrix} = \begin{pmatrix} 6 \\ 3 \\ 9 \end{pmatrix}$.

Vektorrechnung in R

```
x = matrix(c(1,2,3,4), nrow = 4)      # Vektordefinition
y = matrix(c(2,1,0,1), nrow = 4)      # Vektordefinition
x + y                                 # Vektoraddition
```

```
>      [,1]
> [1,]    3
> [2,]    3
> [3,]    3
> [4,]    5
```

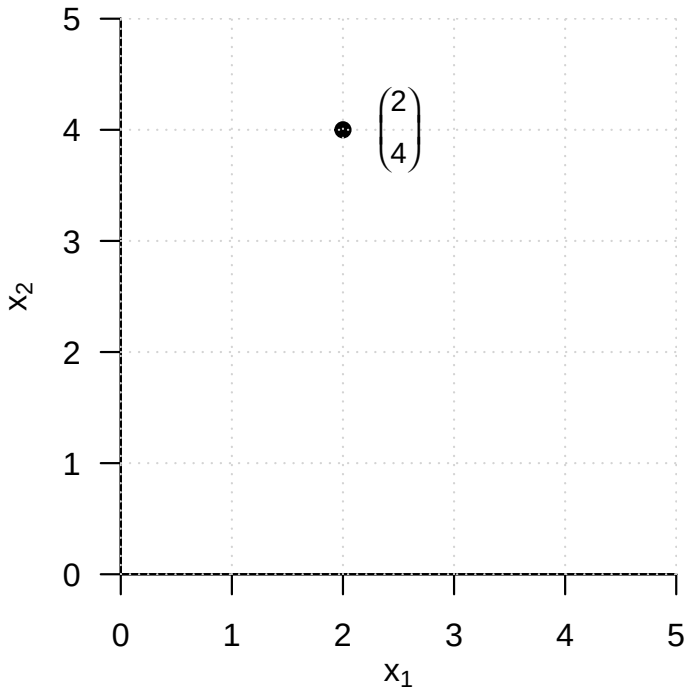
```
x = matrix(c(2,3), nrow = 2)          # Vektordefinition
y = matrix(c(1,3), nrow = 2)          # Vektordefinition
x - y                                 # Vektorsubtraktion
```

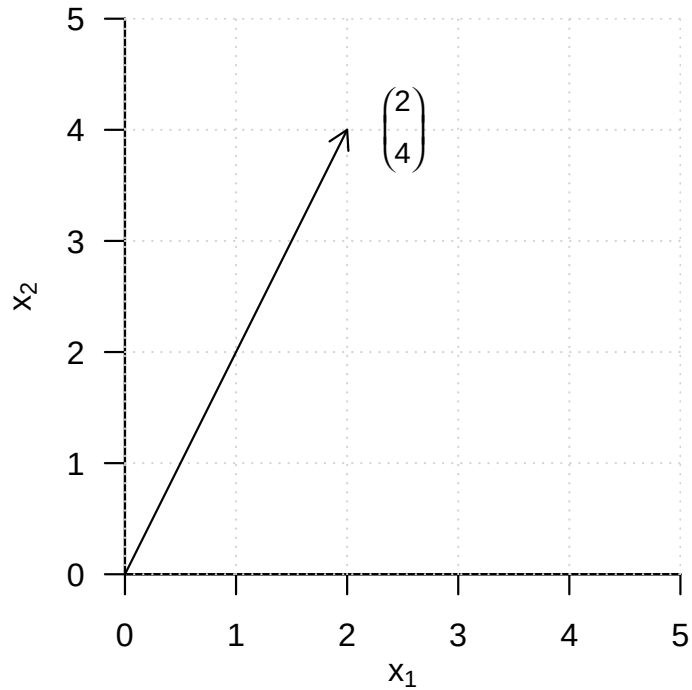
```
>      [,1]
> [1,]    1
> [2,]    0
```

```
x = matrix(c(2,1,3), nrow = 3)        # Vektordefinition
a = 3                                 # Skalardefinition
a*x                                   # Skalarmultiplikation
```

```
>      [,1]
> [1,]    6
> [2,]    3
> [3,]    9
```

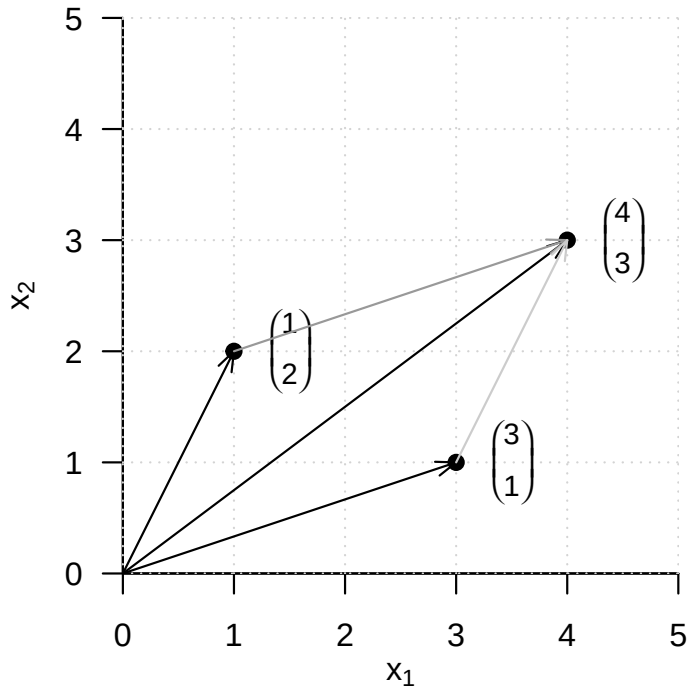
Visualisierung von Vektoren in $\mathbb{R}^2$





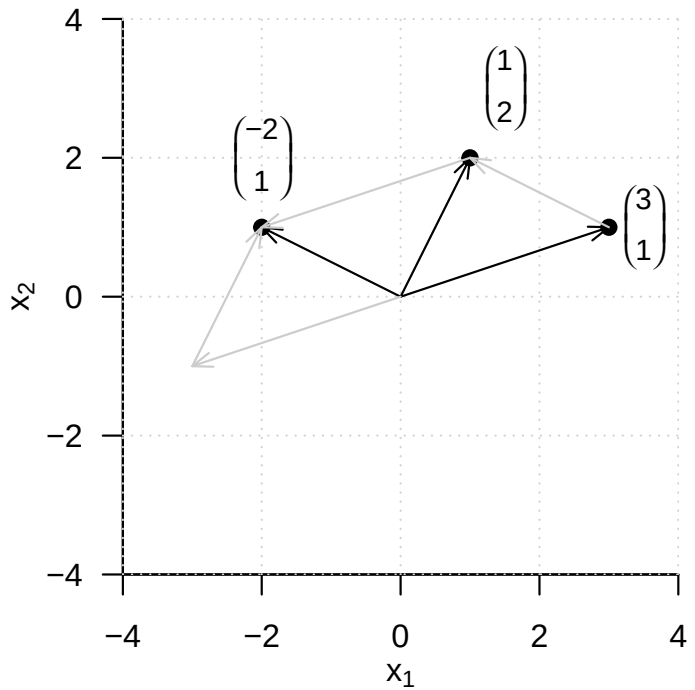
Vektoraddition in $\mathbb{R}^2$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix} \tag{6}$$



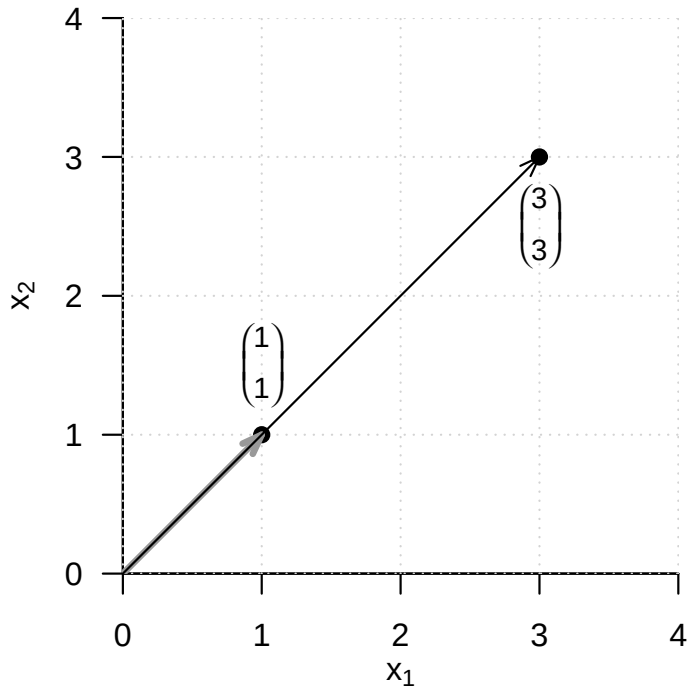
Vektorsubtraktion in $\mathbb{R}^2$

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} -3 \\ -1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \tag{7}$$



Skalarmultiplikation in $\mathbb{R}^2$

$$3 \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \tag{8}$$



Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Definition (Skalarprodukt auf $\mathbb{R}^m$)

Das *Skalarprodukt auf* $\mathbb{R}^m$ ist definiert als die Abbildung

$$\langle\rangle : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}, (x, y) \mapsto \langle (x, y) \rangle := \langle x, y \rangle := \sum_{i=1}^{m} x_i y_i. \tag{9}$$

Bemerkungen

- Das Skalarprodukt heißt Skalarprodukt, weil es einen Skalar ergibt, nicht weil Skalare multipliziert werden.
- Wir sehen später, dass mit der Identifikation $\mathbb{R}^m = \mathbb{R}^{m \times 1}$ und der Matrixtransposition gilt, dass

$$\langle x, y \rangle = x^T y. \tag{10}$$

Beispiel

Es seien

$$x := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \text{ und } y := \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} \tag{11}$$

Dann ergibt sich

$$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + x_3 y_3 = 1 \cdot 2 + 2 \cdot 0 + 3 \cdot 1 = 2 + 0 + 3 = 5. \tag{12}$$

```
# Vektordefinitionen
x = matrix(c(1,2,3), nrow = 3)
y = matrix(c(2,0,1), nrow = 3)
```

```
# Skalarprodukt mithilfe von R's komponentenweiser Multiplikation und sum() Funktion
sum(x*y)
```

```
> [1] 5
```

```
# Skalarprodukt mithilfe von R's Matrixtransposition und -multiplikation
t(x) %*% y
```

```
>      [,1]
> [1,]    5
```

Definition (Euklidischer Vektorraum)

Das Tupel $((\mathbb{R}^m, +, \cdot), \langle\rangle)$ aus dem reellen Vektorraum $(\mathbb{R}^m, +, \cdot)$ und dem Skalarprodukt $\langle\rangle$ auf $\mathbb{R}^m$ heißt *reeller kanonischer Euklidischer Vektorraum*.

Bemerkungen

- Generell heißt jedes Tupel aus einem Vektorraum und einem Skalarprodukt "Euklidischer Vektorraum".
- Informell sprechen wir aber oft auch einfach von $\mathbb{R}^m$ als "Euklidischer Vektorraum" und insbesondere bei $((\mathbb{R}^m, +, \cdot), \langle\rangle)$ von "Euklidischen Vektorraum".
- Ein Euklidischer Vektorraum ist ein Vektorraum mit geometrischer Struktur, die durch das Skalarprodukt induziert wird.
- Insbesondere bekommen im Euklidischen Vektorraum Begriffe wie die *Länge* eines Vektors, der *Abstand* zweier Vektoren und der *Winkel* zwischen zwei Vektoren mithilfe des Skalarproduktes eine Bedeutung.

Definition (Länge, Abstand, Winkel)

$((\mathbb{R}^m, +, \cdot), \langle\rangle)$ sei der Euklidische Vektorraum.

- Die *Länge* eines Vektors $x \in \mathbb{R}^m$ ist definiert als

$$\|x\| := \sqrt{\langle x, x\rangle}. \tag{13}$$

- Der *Abstand* zweier Vektoren $x, y \in \mathbb{R}^m$ ist definiert als

$$d(x, y) := \|x - y\|. \tag{14}$$

- Der *Winkel* α zwischen zwei Vektoren $x, y \in \mathbb{R}^m$ mit $x, y \neq 0$ ist definiert durch

$$0 \leq \alpha \leq \pi \text{ und } \cos\alpha := \frac{\langle x, y\rangle}{\|x\|\|y\|} \tag{15}$$

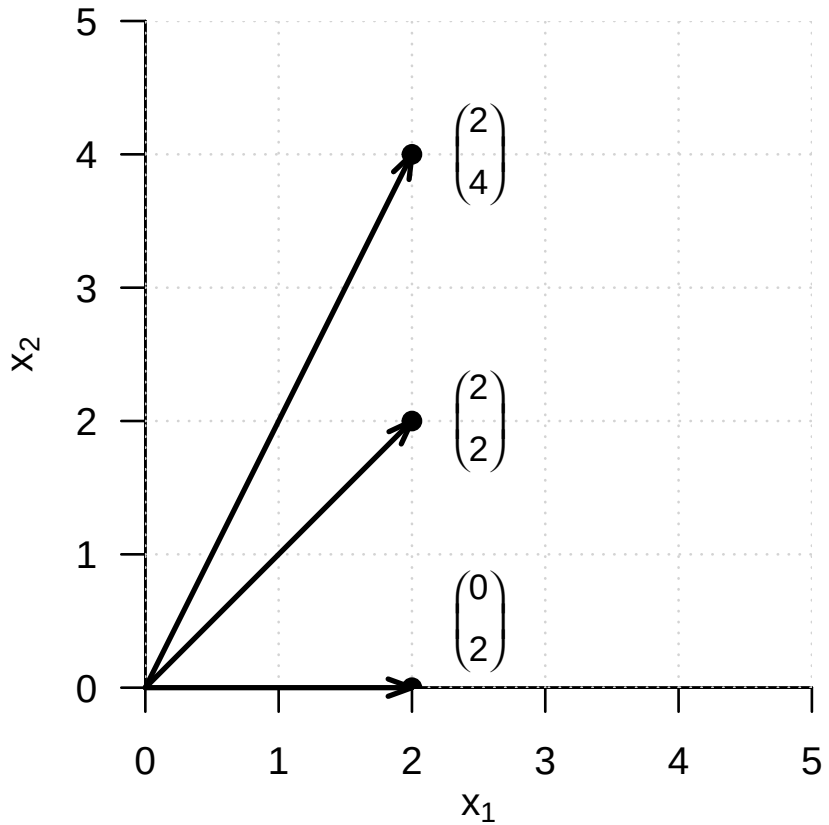
Bemerkungen

- $\|x\|$ heißt auch *Norm von* x oder ℓ_2*-Norm von* x.
- Ohne Beweis halten wir fest, dass für den Abstand gilt, dass

$$d(x, y) \geq 0, d(x, x) = 0, d(x, y) = d(y, x) \text{ und } d(x, y) \leq d(x, z) + d(z, y). \tag{16}$$

- $\cos$ ist auf $[0, \pi]$ bijektiv, also invertierbar.

Vektorlängen in $\mathbb{R}^2$



Vektorlängen in $\mathbb{R}^2$

$$\left\| \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\| = \sqrt{\left\langle \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\rangle} = \sqrt{2^2 + 0^2} = \sqrt{4} = 2.00 \tag{17}$$

```
norm(matrix(c(2,0),nrow = 2), type = "2")              # Vektorlänge = l_2 Norm
```

```
> [1] 2
```

$$\left\| \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\| = \sqrt{\left\langle \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\rangle} = \sqrt{2^2 + 2^2} = \sqrt{8} \approx 2.83 \tag{18}$$

```
norm(matrix(c(2,2),nrow = 2), type = "2")              # Vektorlänge = l_2 Norm
```

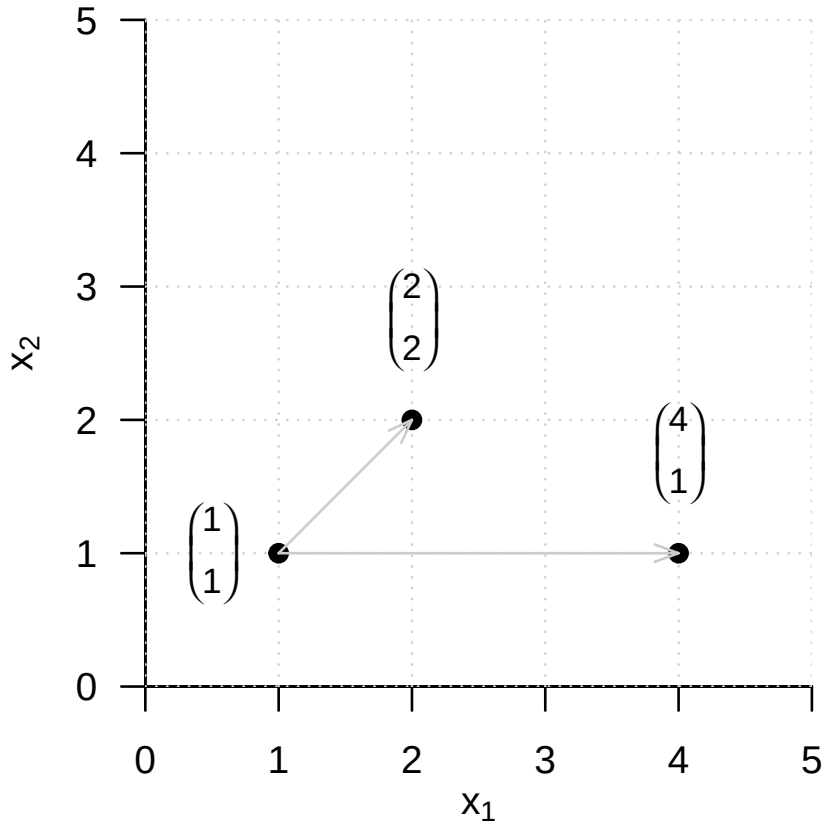
```
> [1] 2.83
```

$$\left\| \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right\| = \sqrt{\left\langle \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right\rangle} = \sqrt{2^2 + 4^2} = \sqrt{20} \approx 4.47 \tag{19}$$

```
norm(matrix(c(2,4),nrow = 2), type = "2")              # Vektorlänge = l_2 Norm
```

```
> [1] 4.47
```

Abstände in $\mathbb{R}^2$



Abstände in $\mathbb{R}^2$

$$d\left(\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}2\\2\end{pmatrix}\right) = \left\|\begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}2\\2\end{pmatrix}\right\| = \left\|\begin{pmatrix}-1\\-1\end{pmatrix}\right\| = \sqrt{(-1)^2 + (-1)^2} = \sqrt{2} \approx 1.41 \tag{20}$$

```
norm(matrix(c(1,1),nrow = 2) - matrix(c(2,2),nrow = 2), type = "2")
```

```
> [1] 1.41
```

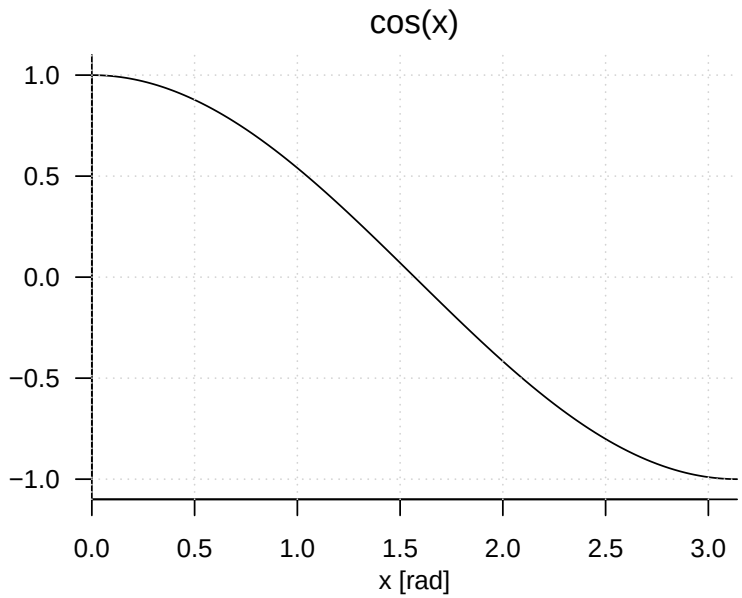
$$d\left(\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}4\\1\end{pmatrix}\right) = \left\|\begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}4\\1\end{pmatrix}\right\| = \left\|\begin{pmatrix}-3\\0\end{pmatrix}\right\| = \sqrt{(-3)^2 + 0^2} = \sqrt{9} = 3 \tag{21}$$

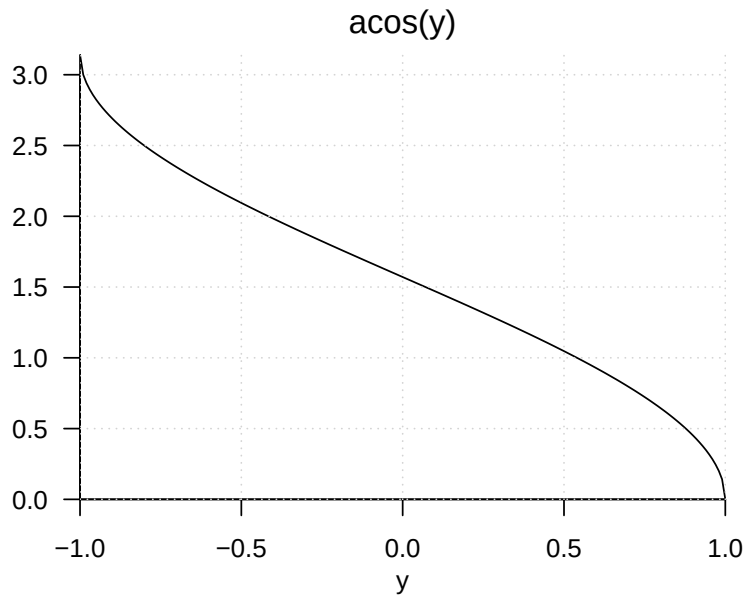
```
norm(matrix(c(1,1),nrow = 2) - matrix(c(1,4),nrow = 2), type = "2")
```

```
> [1] 3
```

Winkel in $\mathbb{R}^2$

Kosinus und Arkuskosinus auf $[0, \pi]$

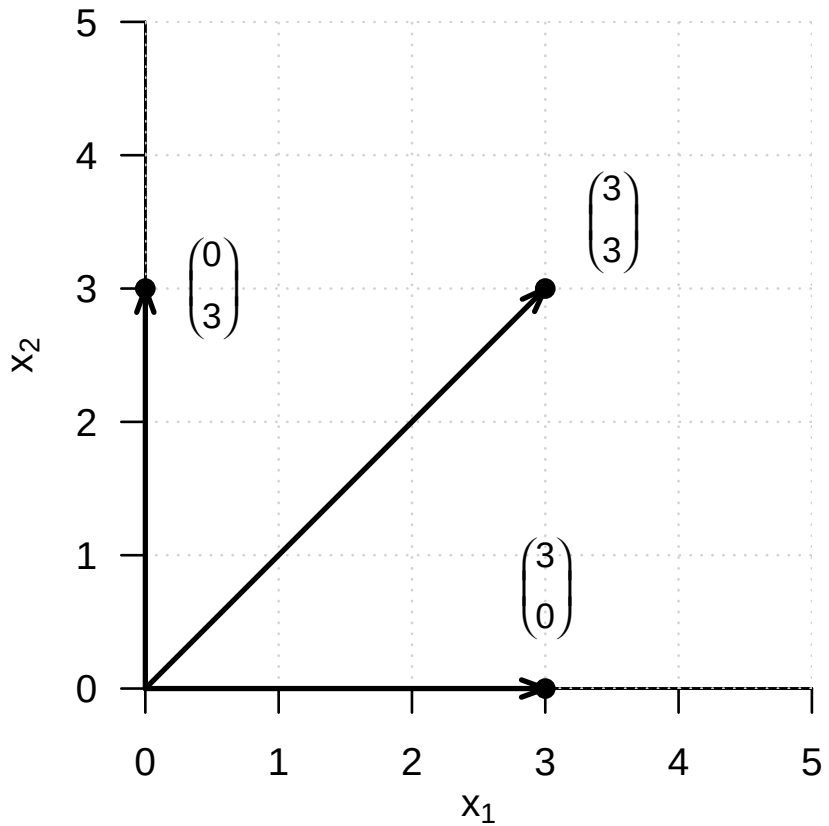




$$\text{deg} = \text{rad} \cdot \frac{180}{\pi}, \ \text{rad} = \text{deg} \cdot \frac{\pi}{180} \tag{22}$$

$$0\pi \text{ rad } = 0.00 \text{ rad } = 0 \text{ deg }, \ \frac{\pi}{2} \text{ rad } \approx 1.57 \text{ rad } = 90 \text{ deg }, \ \pi \text{ rad } \approx 3.14 \text{ rad } = 180 \text{ deg} \tag{23}$$

Winkel in $\mathbb{R}^2$



Winkel in $\mathbb{R}^2$

Winkel in Radians

$$\text{acos}\left(\frac{\left\langle \begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}3\\3\end{pmatrix} \right\rangle}{\left\| \begin{pmatrix}3\\0\end{pmatrix} \right\| \left\| \begin{pmatrix}3\\3\end{pmatrix} \right\|}\right) = \text{acos}\left(\frac{3\cdot 3 + 3\cdot 0}{\sqrt{3^2+0^2}\cdot\sqrt{3^2+3^2}}\right) = \text{acos}\left(\frac{9}{3\cdot\sqrt{18}}\right) = \frac{\pi}{4} \approx 0.785 \tag{24}$$

Winkel in Grad

$$0.785 \cdot 180/\pi = 45 \tag{25}$$

Berechnung in R

```
x = matrix(c(3,0), nrow = 2)                                  # Vektor 1
y = matrix(c(3,3), nrow = 2)                                  # Vektor 2
w = acos(sum(x*y)/(sqrt(sum(x*x))*sqrt(sum(y*y)))) * 180/pi   # Winkel in Grad
print(w)
```

```
> [1] 45
```

Winkel in $\mathbb{R}^2$

Winkel in Radians

$$\alpha = \text{acos}\left(\frac{\left\langle \begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}0\\3\end{pmatrix} \right\rangle}{\left\| \begin{pmatrix}3\\0\end{pmatrix} \right\| \left\| \begin{pmatrix}0\\3\end{pmatrix} \right\|}\right) = \text{acos}\left(\frac{3\cdot 0 + 0\cdot 3}{\sqrt{3^2+0^2}\cdot\sqrt{0^2+3^2}}\right) = \text{acos}\left(\frac{0}{3\cdot 3}\right) = \frac{\pi}{2} \approx 1.57 \tag{26}$$

Winkel in Grad

$$\frac{\pi}{2}\cdot\frac{180}{\pi} = 90 \tag{27}$$

Berechnung in R

```
x = matrix(c(3,0), nrow = 2)                                  # Vektor 1
y = matrix(c(0,3), nrow = 2)                                  # Vektor 2
w = acos(sum(x*y)/(sqrt(sum(x*x))*sqrt(sum(y*y)))) * 180/pi   # Winkel in Grad
print(w)
```

```
> [1] 90
```

Definition (Orthogonalität und Orthonormalität von Vektoren)

$((\mathbb{R}^m, +, \cdot), \langle\rangle)$ sei der Euklidische Vektorraum.

- Zwei Vektoren $x, y \in \mathbb{R}^m$ heißen *orthogonal*, wenn gilt, dass

$$\langle x, y \rangle = 0 \tag{28}$$

- Zwei Vektoren $x, y \in \mathbb{R}^m$ heißen *orthonormal*, wenn gilt, dass

$$\langle x, y \rangle = 0 \text{ und } \|x\| = \|y\| = 1. \tag{29}$$

Bemerkung

- Für orthogonale und orthonormale Vektoren gilt insbesondere auch

$$\cos \alpha = \frac{\langle x, y \rangle}{\|x\| \|y\|} = \frac{0}{\|x\| \|y\|} = 0 \tag{30}$$

  also

$$\alpha = \frac{\pi}{2} = 90^\circ \tag{31}$$

Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Definition (Linearkombination)

$\{v_1, v_2, ..., v_k\}$ sei eine Menge von k Vektoren eines Vektorraums V. Dann ist die *Linearkombination* der Vektoren in $v_1, v_2, ..., v_k$ mit den skalaren Koeffizienten $a_1, a_2, ..., a_k$ definiert als der Vektor

$$w := \sum_{i=1}^{k} a_i v_i \in V. \tag{32}$$

Beispiel

Es seien

$$v_1 := \begin{pmatrix} 2 \\ 1 \end{pmatrix}, v_2 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}, v_3 := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ und } a_1 := 2, a_2 := 3, a_3 := 0. \tag{33}$$

Dann ergibt sich

$$\begin{aligned} w = a_1 v_1 + a_2 v_2 + a_3 v_3 &= 2 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} + 3 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 4 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 7 \\ 5 \end{pmatrix} \end{aligned} \tag{34}$$

Definition (Lineare Unabhängigkeit)

V sei ein Vektorraum. Eine Menge $W := \{w_1, w_2, ..., w_k\}$ von Vektoren in V heißt *linear unabhängig*, wenn die einzige Repräsentation des Nullelements $0 \in V$ durch eine Linearkombination der $w \in W$ die triviale Repräsentation

$$0 = a_1 w_1 + a_2 w_2 + \cdots + a_n w_n \text{ mit } a_1 = a_2 = \cdots = a_n = 0 \tag{35}$$

ist. Wenn die Menge W nicht linear unabhängig ist, dann heißt sie *linear abhängig*.

Bemerkungen

- Prinzipiell müsste man für jede Linearkombination der $w \in W$ prüfen, ob sie Null ist.
- Die beiden folgenden Theoreme zeigen, dass es auch einfacher geht.

Theorem (Lineare Abhängigkeit von zwei Vektoren)

V sei ein Vektorraum. Zwei Vektoren $v_1, v_2 \in V$ sind linear abhängig, wenn einer der Vektoren ein skalares Vielfaches des anderen Vektors ist.

Beweis

v_1 sei ein skalares Vielfaches von v_2, also

$$v_1 = \lambda v_2 \text{ mit } \lambda \neq 0. \tag{36}$$

Dann gilt

$$v_1 - \lambda v_2 = 0. \tag{37}$$

Dies aber entspricht der Linearkombination

$$a_1 v_1 + a_2 v_2 = 0 \tag{38}$$

mit $a_1 = 1 \neq 0$ und $a_2 = -\lambda \neq 0$. Es gibt also eine Linearkombination des Nullelementes, die nicht die triviale Repräsentation ist, und damit sind v_1 und v_2 nicht linear unabhängig.

Theorem (Lineare Abhängigkeit einer Menge von Vektoren)

V sei ein Vektorraum und $w_1, ..., w_k \in V$ sei eine Menge von Vektoren in V. Wenn einer der Vektoren $w_i, i = 1, ..., k$ eine Linearkombination der anderen Vektoren ist, dann ist die Menge der Vektoren linear abhängig.

Beweis

Die Vektoren $w_1, ..., w_k$ sind genau dann linear abhängig, wenn gilt, dass $\sum_{i=1}^{n} a_i w_i = 0$ mit mindestens einem $a_i \neq 0$. Es sei also zum Beispiel $a_j \neq 0$. Dann gilt

$$0 = \sum_{i=1}^{n} a_i w_i = \sum_{i=1, i\neq j}^{n} a_i w_i + a_j w_j \tag{39}$$

Also folgt

$$a_j w_j = - \sum_{i=1, i\neq j}^{n} a_i w_i \tag{40}$$

und damit

$$w_j = -a_j^{-1} \sum_{i=1, i\neq j}^{n} a_i w_i = - \sum_{i=1, i\neq j}^{n} (a_j^{-1} a_i) w_i \tag{41}$$

Also ist w_j eine Linearkombination der $w_i, i = 1, ..., k$ mit $i \neq j$. □

Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Definition (Lineare Hülle und Aufspannen)

V sei ein Vektorraum und es sei $W := \{w_1, ..., w_k\} \subset V$. Dann ist die *lineare Hülle* von W definiert als die Menge aller Linearkombinationen der Elemente von W,

$$\text{Span}(W) := \left\{ \sum_{i=1}^{k} a_i w_i | a_1, ..., a_k \text{ sind skalare Koeffizienten} \right\} \tag{42}$$

Man sagt, dass eine Menge von Vektoren $W \subseteq V$ *einen Vektorraum* V *aufspannt*, wenn jedes $v \in V$ als eine Linearkombination von Vektoren in W geschrieben werden kann.

Definition (Basis)

V sei ein Vektorraum und es sei $B \subseteq V$. Dann heißt B eine *Basis von* V, wenn

- die Vektoren in B linear unabhängig sind und
- die Vektoren in B den Vektorraum V aufspannen.

Theorem (Eigenschaften von Basen)

- Alle Basen eines Vektorraums beinhalten die gleiche Anzahl von Vektoren.
- Die Anzahl der Vektoren einer Basis heißt die *Dimension* des Vektorraums.
- Jede Menge von m linear unabhängigen Vektoren ist Basis eines m-dimensionalen Vektorraums.

Bemerkung

- Wir verzichten auf einen Beweis des sehr tiefen Theorems.
- Vektorräume haben in der Regel unendlich viele Basen.

Definition (Basisdarstellung und Koordinaten)

$B := \{b_1, ..., b_m\}$ sei eine Basis eines m-dimensionalen Vektorraumes V und es sei $v \in V$. Dann heißt die Linearkombination

$$v = \sum_{i=1}^{m} c_i b_i \tag{43}$$

die *Darstellung von v bezüglich der Basis B* und die Koeffizienten $c_1, ..., c_m$ heißen die *Koordinaten von v bezüglich der Basis B*.

Theorem (Eindeutigkeit der Basisdarstellung)

Die Basisdarstellung eines $v \in V$ bezüglich einer Basis B ist eindeutig.

Beweis

Ohne Beschränkung der Allgemeinheit nehmen wir an, dass der Vektorraum von Dimension m ist. Nehmen wir an, dass zwei Darstellungen von v bezüglich der Basis B existieren, also dass

$$\begin{aligned} v &= a_1 b_1 + \cdots + a_m b_m \\ v &= c_1 b_1 + \cdots + c_m b_m \end{aligned} \tag{44}$$

Subtraktion der unteren von dern oberen Gleichung ergibt

$$0 = (a_1 - c_1)b_1 + \cdots + (a_m - c_m)b_m \tag{45}$$

Weil die $b_1, \ldots, b_m$ linear unabhängig sind, gilt aber, dass $(a_i - c_i) = 0$ für alle $i = 1, \ldots, m$ und somit sind die beiden Darstellungen von v bezüglich der Basis B identisch.

□

Definition (Orthonormalbasis von $\mathbb{R}^m$)

Eine Menge von m Vektoren $v_1, ..., v_m \in \mathbb{R}^m$ heißt *Orthonormalbasis* von $\mathbb{R}^m$, wenn $v_1, ..., v_m$ jeweils die Länge 1 haben und wechselseitig orthogonal sind, also wenn

$$\langle v_i, v_j \rangle = \begin{cases} 1 & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}. \tag{46}$$

Beispiel 1

Es ist

$$B_1 := \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\} \tag{47}$$

eine Orthonormalbasis von $\mathbb{R}^2$, denn B_1 besteht aus zwei Vektoren und es gelten

$$\left\langle \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\rangle = 1 \cdot 1 + 0 \cdot 0 = 1 + 0 = 1, \tag{48}$$

sowie

$$\left\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle = 0 \cdot 0 + 1 \cdot 1 = 0 + 1 = 1 \tag{49}$$

und

$$\left\langle \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle = 1 \cdot 0 + 0 \cdot 1 = 0 + 0 = 0 \tag{50}$$

Definition (Kanonische Basis und kanonische Einheitsvektoren)

Die Orthonormalbasis

$$B := \left\{ e_1, ..., e_m | e_{i_j} = 1 \text{ für } i = j \text{ und } e_{i_j} = 0 \text{ für } i \neq j \right\} \subset \mathbb{R}^m \tag{51}$$

heißt die *kanonische Basis* von $\mathbb{R}^m$ und die e_{i_j} heißen *kanonische Einheitsvektoren*.

Beispiele

- B_1 aus Beispiel 1 ist die kanonische Basis von $\mathbb{R}^2$.
- Die kanonische Basis von $\mathbb{R}^3$ ist $B := \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$.

Beispiel 2

Es ist auch

$$B_2 := \left\{ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\} \tag{52}$$

eine Orthonormalbasis von $\mathbb{R}^2$, denn B_2 besteht aus zwei Vektoren und es gelten

$$\left\langle \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = \frac{1}{2} + \frac{1}{2} = 1, \tag{53}$$

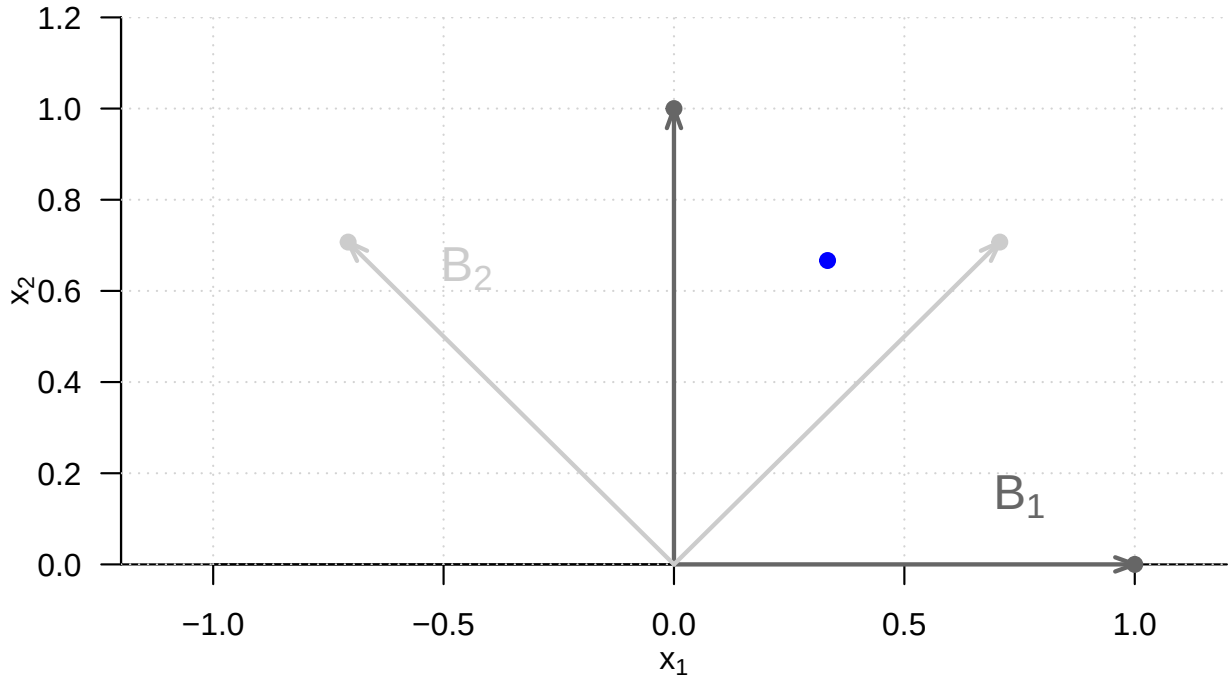
sowie

$$\left\langle \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle = \left(-\frac{1}{\sqrt{2}} \right) \cdot \left(-\frac{1}{\sqrt{2}} \right) + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = \frac{1}{2} + \frac{1}{2} = 1 \tag{54}$$

und

$$\left\langle \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\rangle = -\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} = -\frac{1}{2} + \frac{1}{2} = 0 \tag{55}$$

Beispiele 1 & 2



- Im Rahmen von Hauptkomponentenranalyse werden wir daran interessiert sein, basierend auf den Koordinaten eines Vektors bezüglich einer Basis die Koordinaten desselben Vektors bezüglich einer anderen Basis zu berechnen.

Reeller Vektorraum

Euklidischer Vektorraum

Lineare Unabhängigkeit

Vektorraumbasen

Selbstkontrollfragen

Selbstkontrollfragen

1. Geben Sie die Definition eines Vektorraums wieder.
2. Geben Sie die Definition des reellen Vektorraums wieder.
3. Es seien

$$x := \begin{pmatrix} 2 \\ 1 \end{pmatrix}, y := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ und } a := 2. \tag{56}$$

   Berechnen Sie

$$v = a(x + y) \text{ und } w = \frac{1}{a}(y - x) \tag{57}$$

   und überprüfen Sie ihre Rechnung mit R.
4. Geben Sie die Definition des Skalarproduktes auf $\mathbb{R}^m$ wieder.
5. Für

$$x := \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}, y := \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, z := \begin{pmatrix} 3 \\ 1 \\ 0 \end{pmatrix} \tag{58}$$

   berechnen Sie

$$\langle x, y \rangle, \langle x, z \rangle, \langle y, z \rangle \tag{59}$$

   und überprüfen Sie ihre Rechnung mithilfe von R.
6. Geben Sie die Definition des Euklidischen Vektorraums wieder.
7. Definieren Sie die Länge eines Vektors im Euklidischen Vektorraum.
8. Berechnen Sie die Längen der Vektoren x, y, z aus Aufgabe 5 und überprüfen Sie ihre Rechnung mit R.

Selbstkontrollfragen

9. Geben Sie Definition des Abstands zweier Vektoren im Euklidischen Vektorraum wieder.
10. Berechnen Sie $d(x, y)$, $d(x, z)$ und $d(y, z)$ für x, y, z aus Aufgabe 5.
11. Geben Sie die Definition des Winkels zwischen zwei Vektoren im Euklidischen Vektorraum wieder.
12. Berechnen Sie die Winkel zwischen den Vektoren x und y, x und z, sowie y und z aus Aufgabe 5 mit R.
13. Definieren Sie die Begriffe der Orthogonalität und Orthonormalität von Vektoren.
14. Definieren Sie den Begriff der Linearkombination von Vektoren.
15. Definieren Sie den Begriff der linearen Unabhängigkeit von Vektoren.
16. Woran kann man erkennen, dass zwei Vektoren linear abhängig sind?
17. Definieren Sie den Begriff der linearen Hülle einer Menge von Vektoren.
18. Definieren Sie den Begriff der Basis eines Vektorraums.
19. Geben Sie das Theorem zu den Eigenschaften von Vektorraumbasen wieder.
20. Definieren Sie den Begriff der Basisdarstellung eines Vektors.
21. Definieren Sie den Begriff einer Orthonormalbasis von $\mathbb{R}^m$.
22. Definieren Sie die kanonische Basis von $\mathbb{R}^m$
23. Dokumentieren Sie alle in dieser Einheit eingeführten R Befehle in einem Skript.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(2) Matrizen

Motivation

Matrizen sind die Worte der Sprache der multivariaten Datenanalyse.

Vektoren sind nur spezielle Matrizen.

Matrizen können als Tabellen der Datenrepräsentation dienen.

Matrizen können lineare Abbildungen repräsentieren.

Matrizen können Vektorräume repräsentieren.

Ein sicherer Umgang mit Matrizen ist für

das Verständnis multivariater Verfahren unverzichtbar.

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Definition (Matrix)

Eine Matrix ist eine rechteckige Anordnung von Zahlen, die wie folgt bezeichnet wird

$$A := \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} := (a_{ij})_{1 \leq i \leq n,\, 1 \leq j \leq m}. \tag{1}$$

Bemerkungen

- Matrizen bestehen aus *Zeilen (rows)* und *Spalten (columns)*.
- Die Matrixeinträge a_{ij} werden mit einem *Zeilenindex* i und einem *Spaltenindex* j indiziert.
- Zum Beispiel gilt für $A := \begin{pmatrix} 2 & 7 & 5 & 2 \\ 8 & 2 & 5 & 6 \\ 6 & 4 & 0 & 9 \\ 9 & 2 & 1 & 2 \end{pmatrix}$, dass $a_{32} = 4$.

Definition

Bemerkungen (fortgeführt)

- Die *Größe* oder *Dimension* einer Matrix ergibt sich aus der Anzahl ihrer Zeilen $m \in \mathbb{N}$ und Spalten $n \in \mathbb{N}$.
- Matrizen mit $m = n$ heißen *quadratische Matrizen*.
- In der Folge benötigen wir nur Matrizen mit reellen Einträgen, also $a_{ij} \in \mathbb{R}\ \forall i = 1, \ldots, m, j = 1, \ldots, n$.
- Wir nennen die Matrizen mit reellen Einträge *reelle Matrizen*.
- Die Menge der reellen Matrizen mit m Zeilen und n Spalten bezeichnen wir mit $\mathbb{R}^{m \times n}$
- Aus dem Ausdruck $A \in \mathbb{R}^{2 \times 3}$ lesen wir ab, dass A eine reelle Matrix mit zwei Zeilen und drei Spalten ist.
- Wir identifizieren die Menge $\mathbb{R}^{1 \times 1}$ mit der Menge $\mathbb{R}$.
- Wir identifizieren die Menge $\mathbb{R}^{m \times 1}$ mit der Menge $\mathbb{R}^m$.
- Reelle Matrizen mit einer Spalte und m Zeilen sind also dasselbe wie m-dimensionale reelle Vektoren.

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Matrixoperationen

Man kann mit Matrizen rechnen.

In der Folge betrachten wir folgende grundlegende Matrixoperationen

- Addition und Subtraktion von Matrizen gleicher Größe (Matrixaddition und Matrixsubtraktion)
- Multiplikation einer Matrix mit einem Skalar (Skalarmultiplikation)
- Vertauschen der Zeilen- und Spaltenanordnung (Matrixtransposition)
- Multiplikation einer Matrix mit einer passenden zweiten Matrix (Matrixmultiplikation)
- "Teilen" durch eine Matrix (Matrixinversion)

Definition (Matrixaddition)

Es seien $A, B \in \mathbb{R}^{m \times n}$. Dann ist die *Addition* von A und B definiert als die Abbildung

$$+ : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n},\ (A, B) \mapsto +(A, B) := A + B \tag{2}$$

mit

$$\begin{aligned} A + B &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} \\ &:= \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{pmatrix}. \end{aligned} \tag{3}$$

Bemerkungen

- Nur Matrizen identischer Größe können miteinander addiert werden.
- Die Addition zweier gleich großer Matrizen ist elementweise definiert.

Definition (Matrixsubtraktion)

Es seien $A, B \in \mathbb{R}^{m \times n}$. Dann ist die *Subtraktion* von A und B definiert als die Abbildung

$$- : \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n},\ (A, B) \mapsto -(A, B) := A - B \tag{4}$$

mit

$$\begin{aligned} A - B &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} - \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{pmatrix} \\ &:= \begin{pmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \cdots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} & \cdots & a_{2n} - b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn} \end{pmatrix}. \end{aligned} \tag{5}$$

Bemerkungen

- Nur Matrizen identischer Größe können voneinander subtrahiert werden.
- Die Subktration zweier gleich großer Matrizen ist elementweise definiert.

Operationen

Beispiel

Es seien $A, B \in \mathbb{R}^{2\times 3}$ definiert als

$$A := \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} \text{ und } B := \begin{pmatrix} 4 & 1 & 0 \\ -4 & 2 & 0 \end{pmatrix}. \tag{6}$$

Da A und B gleich groß sind, können wir sie addieren

$$\begin{aligned} C = A + B &= \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} + \begin{pmatrix} 4 & 1 & 0 \\ -4 & 2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 2+4 & -3+1 & 0+0 \\ 1-4 & 6+2 & 5+0 \end{pmatrix} \\ &= \begin{pmatrix} 6 & -2 & 0 \\ -3 & 8 & 5 \end{pmatrix} \end{aligned} \tag{7}$$

und voneinander subtrahieren

$$\begin{aligned} D = A - B &= \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} - \begin{pmatrix} 4 & 1 & 0 \\ -4 & 2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 2-4 & -3-1 & 0-0 \\ 1+4 & 6-2 & 5-0 \end{pmatrix} \\ &= \begin{pmatrix} -2 & -4 & 0 \\ 5 & 4 & 5 \end{pmatrix}. \end{aligned} \tag{8}$$

Operationen

Beispiel

```
# Spaltenweise Definition von A (R default)
A = matrix(c(2,1,-3,6,0,5), nrow = 2)
print(A)
```

```
>      [,1] [,2] [,3]
> [1,]    2   -3    0
> [2,]    1    6    5
```

```
# Zeilenweise Definition von B
B = matrix(c(4,1,0,-4,2,0), nrow = 2, byrow = TRUE)
print(B)
```

```
>      [,1] [,2] [,3]
> [1,]    4    1    0
> [2,]   -4    2    0
```

Operationen

Beispiel

```
# Addition
C = A + B
print(C)
```

```
>      [,1] [,2] [,3]
> [1,]    6   -2    0
> [2,]   -3    8    5
```

```
# Subtraktion
D = A - B
print(D)
```

```
>      [,1] [,2] [,3]
> [1,]   -2   -4    0
> [2,]    5    4    5
```

Definition (Skalarmultiplikation)

Es sei $c \in \mathbb{R}$ ein Skalar und $A \in \mathbb{R}^{m \times n}$. Dann ist die *Skalarmultiplikation* von c und A definiert als die Abbildung

$$\cdot : \mathbb{R} \times \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n},\ (c, A) \mapsto \cdot(c, A) := cA \tag{9}$$

mit

$$cA = c \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} := \begin{pmatrix} ca_{11} & ca_{12} & \cdots & ca_{1n} \\ ca_{21} & ca_{22} & \cdots & ca_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ca_{m1} & ca_{m2} & \cdots & ca_{mn} \end{pmatrix}. \tag{10}$$

Bemerkungen

- Die Skalarmultiplikation ist elementweise definiert.

Beispiel

Es seien $c := -3$ und $A \in \mathbb{R}^{4\times 3}$ definiert als

$$A := \begin{pmatrix} 3 & 1 & 1 \\ 5 & 2 & 5 \\ 2 & 7 & 1 \\ 3 & 4 & 2 \end{pmatrix}. \tag{11}$$

Dann ergibt sich

$$B := cA = -3 \begin{pmatrix} 3 & 1 & 1 \\ 5 & 2 & 5 \\ 2 & 7 & 1 \\ 3 & 4 & 2 \end{pmatrix} = \begin{pmatrix} -3\cdot 3 & -3\cdot 1 & -3\cdot 1 \\ -3\cdot 5 & -3\cdot 2 & -3\cdot 5 \\ -3\cdot 2 & -3\cdot 7 & -3\cdot 1 \\ -3\cdot 3 & -3\cdot 4 & -3\cdot 2 \end{pmatrix} = \begin{pmatrix} -9 & -3 & -3 \\ -15 & -6 & -15 \\ -6 & -21 & -3 \\ -9 & -12 & -6 \end{pmatrix}. \tag{12}$$

Operationen

Beispiel

```
# Definitionen
A = matrix(c(3,1,1,
             5,2,5,
             2,7,1,
             3,4,2),
           nrow = 4,
           byrow = TRUE)
c = -3

# Skalarmultiplikation
B = c*A
print(B)
```

```
>      [,1] [,2] [,3]
> [1,]   -9   -3   -3
> [2,]  -15   -6  -15
> [3,]   -6  -21   -3
> [4,]   -9  -12   -6
```

Theorem (Vektorraum $\mathbb{R}^{m\times n}$)

Das Tripel $(\mathbb{R}^{m\times n}, +, \cdot)$ mit der oben definierten Matrixaddition und Skalarmultiplikation ist ein Vektorraum. Insbesondere gelten also für $A, B, C \in \mathbb{R}^{m\times n}$ und $r, s, t \in \mathbb{R}$ folgende Rechenregeln:

(1) Kommutativität der Addition	$A + B = B + A$
(2) Assoziativität der Addition	$(A + B) + C = A + (B + C)$
(3) Existenz eines neutralen Elements der Addition	$\exists\, 0 \in \mathbb{R}^{m\times n}$ mit $A + 0 = 0 + A = A$.
(4) Existenz inverser Elemente der Addition	$\forall A\, \exists\, -A$ mit $A + (-A) = 0$.
(5) Existenz eines neutralen Elements der Skalarmultiplikation	$\exists\, 1 \in \mathbb{R}$ mit $1 \cdot A = A$.
(6) Assoziativität der Skalarmultiplikation	$r \cdot (s \cdot t) = (r \cdot s) \cdot t$.
(7) Distributivität hinsichtlich der Matrixaddition	$r \cdot (A + B) = r \cdot A + r \cdot B$.
(8) Distributivität hinsichtlich der Skalaraddition	$(r + s) \cdot A = r \cdot A + s \cdot A$.

Bemerkungen

- Wir verzichten auf einen Beweis.
- Der Beweis ergibt sich mit dem elementweisen Charakter von $+, -, \cdot$ und den Rechenregeln in $(\mathbb{R}, +, \cdot)$.
- Das neutrale Element der Addition heißt *Nullmatrix*; wir schreiben $0_{nm} := (0)_{1\leq i\leq m, 1\leq j\leq n}$ mit $0 \in \mathbb{R}$.
- Die inversen Elemente der Addition sind durch $-A := (-a_{ij})_{1\leq i\leq m, 1\leq j\leq n}$ gegeben.
- Das neutrale Element der Skalarmultiplikation ist $1 \in \mathbb{R}$.

Definition (Matrixtransposition)

Es sei $A \in \mathbb{R}^{m \times n}$. Dann ist die *Transposition* von A definiert als die Abbildung

$$\cdot^T : \mathbb{R}^{m \times n} \to \mathbb{R}^{n \times m},\ A \mapsto \cdot^T(A) := A^T \tag{13}$$

mit

$$A^T = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} := \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{pmatrix} \tag{14}$$

Bemerkungen

- Die Matrixtransposition "vertauscht" Zeilen und Spalten.
- Für $A \in \mathbb{R}^{m \times n}$ gilt immer $A^T \in \mathbb{R}^{n \times m}$.
- Für $A \in \mathbb{R}^{1 \times 1}$ gilt immer $A^T = A$.
- Es gilt $\left(A^T\right)^T = A$.
- Es gilt $(a_{ii})_{1 \leq i \leq \min(m,n)} = (a_{ii})^T_{1 \leq i \leq \min(m,n)}$
- Matrixelemente auf der Hauptdiagonalen einer Matrix bleiben bei Transposition also unberührt.

Beispiel

Es sei $A \in \mathbb{R}^{2\times 3}$ definiert durch

$$A := \begin{pmatrix} 2 & 3 & 0 \\ 1 & 6 & 5 \end{pmatrix}, \tag{15}$$

Dann gilt $A^T \in \mathbb{R}^{3\times 2}$ und speziell

$$A^T := \begin{pmatrix} 2 & 1 \\ 3 & 6 \\ 0 & 5 \end{pmatrix}. \tag{16}$$

Weiterhin gilt offenbar $\min(m, n) = 2$ und folglich

$$(a_{11}) = (a_{11})^T \text{ und } (a_{22}) = (a_{22})^T. \tag{17}$$

Operationen

Beispiel

```
# Definition
A = matrix(c(2,3,0,
             1,6,5),
           nrow = 2,
           byrow = TRUE)
print(A)
```

```
>      [,1] [,2] [,3]
> [1,]    2    3    0
> [2,]    1    6    5
```

```
# Transposition
AT = t(A)
print(AT)
```

```
>      [,1] [,2]
> [1,]    2    1
> [2,]    3    6
> [3,]    0    5
```

Definition (Matrixmultiplikation)

Es seien $A \in \mathbb{R}^{m \times n}$ und $B \in \mathbb{R}^{n \times p}$. Dann ist die *Matrixmultiplikation* von A und B definiert als die Abbildung

$$\cdot : \mathbb{R}^{m \times n} \times \mathbb{R}^{n \times p} \to \mathbb{R}^{m \times p},\ (A, B) \mapsto \cdot(A, B) := AB \tag{18}$$

mit

$$\begin{aligned} AB &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{pmatrix} \\ &:= \begin{pmatrix} \sum_{i=1}^{n} a_{1i} b_{i1} & \sum_{i=1}^{n} a_{1i} b_{i2} & \cdots & \sum_{i=1}^{n} a_{1i} b_{ip} \\ \sum_{i=1}^{n} a_{2i} b_{i1} & \sum_{i=1}^{n} a_{2i} b_{i2} & \cdots & \sum_{i=1}^{n} a_{2i} b_{ip} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} a_{mi} b_{i1} & \sum_{i=1}^{n} a_{mi} b_{i2} & \cdots & \sum_{i=1}^{n} a_{mi} b_{ip} \end{pmatrix} \end{aligned} \tag{19}$$

Operationen

Bemerkungen

- Das Matrixprodukt AB ist nur dann definiert, wenn A genau so viele Spalten hat wie B Zeilen.
- Informell gilt für die beteiligten Matrixgrößen immer $(m \times n)(n \times p) = (m \times p)$.
- In AB ist $(AB)_{ij}$ die Summe der multiplizierten $i$ten Zeilen von A und $j$ten Spalten von B.
- Zum Berechnen von $(AB)_{ij}$ für $1 \leq i \leq m, 1 \leq j \leq p$ geht man also wie folgt vor:
  1. Man legt in Gedanken die Tranposition der $i$ten Zeile von A über die $j$te Spalte von B.
  2. Weil A genau n Spalten hat und B genau n Zeilen hat, gibt es zu jedem Element der Zeile aus A ein korrespondierendes Element in der Spalte von B.
  3. Man multipliziert die korrespondierenden Elemente miteinander.
  4. Die Summe dieser Produkte ist dann der Eintrag mit Index ij in AB.
- Die Multiplikation von Matrizen ist im Allgemeinen nicht kommutativ (also meist $AB \neq BA$).

Beispiel

$A \in \mathbb{R}^{2\times 3}$ und $B \in \mathbb{R}^{3\times 2}$ seien definiert als

$$A := \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} \text{ und } B := \begin{pmatrix} 4 & 2 \\ -1 & 0 \\ 1 & 3 \end{pmatrix}. \tag{20}$$

Wir wollen $C := AB$ und $D := BA$ berechnen.

Mit $m = 2, n = 3$ und $p = 2$ wissen wir schon, dass $C \in \mathbb{R}^{2\times 2}$ und $D \in \mathbb{R}^{3\times 3}$, weil

$$(2 \times 3)(3 \times 2) = (2 \times 2) \tag{21}$$

und

$$(3 \times 2)(2 \times 3) = (3 \times 3) \tag{22}$$

Es gilt hier also sicher $AB \neq BA$.

Beispiel (fortgeführt)

Es ergibt sich zum einen

$$
\begin{aligned}
C &= AB \\
&= \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} \begin{pmatrix} 4 & 2 \\ -1 & 0 \\ 1 & 3 \end{pmatrix} \\
&= \begin{pmatrix} 2 \cdot 4 + (-3) \cdot (-1) + 0 \cdot 1 & 2 \cdot 2 + (-3) \cdot 0 + 0 \cdot 3 \\ 1 \cdot 4 + 6 \cdot (-1) + 5 \cdot 1 & 1 \cdot 2 + 6 \cdot 0 + 5 \cdot 3 \end{pmatrix} \\
&= \begin{pmatrix} 8 + 3 + 0 & 4 + 0 + 0 \\ 4 - 6 + 5 & 2 + 0 + 15 \end{pmatrix} \\
&= \begin{pmatrix} 11 & 4 \\ 3 & 17 \end{pmatrix}.
\end{aligned}
\tag{23}
$$

Beispiel (fortgeführt)

```
# Definitionen
A = matrix(c(2,-3,0,
             1, 6,5),
           nrow  = 2,
           byrow = TRUE)
B = matrix(c( 4,2,
             -1,0,
              1,3),
           nrow  = 3,
           byrow = TRUE)

# Matrixmultiplikation
C = A %*% B
print(C)
```

```
>      [,1] [,2]
> [1,]   11    4
> [2,]    3   17
```

Beispiel (fortgeführt)

Es ergibt sich zum anderen

$$
\begin{aligned}
D &= BA \\
&= \begin{pmatrix} 4 & 2 \\ -1 & 0 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} 2 & -3 & 0 \\ 1 & 6 & 5 \end{pmatrix} \\
&= \begin{pmatrix} 4 \cdot 2 + 2 \cdot 1 & 4 \cdot (-3) + 2 \cdot 6 & 4 \cdot 0 + 2 \cdot 5 \\ (-1) \cdot 2 + 0 \cdot 1 & (-1) \cdot (-3) + 0 \cdot 6 & (-1) \cdot 0 + 0 \cdot 5 \\ 1 \cdot 2 + 3 \cdot 1 & 1 \cdot (-3) + 3 \cdot 6 & 1 \cdot 0 + 3 \cdot 5 \end{pmatrix} \\
&= \begin{pmatrix} 8 + 2 & -12 + 12 & 0 + 5 \\ -2 + 0 & 3 + 0 & 0 + 0 \\ 2 + 3 & -3 + 18 & 0 + 15 \end{pmatrix} \\
&= \begin{pmatrix} 10 & 0 & 10 \\ -2 & 3 & 0 \\ 5 & 15 & 15 \end{pmatrix}
\end{aligned}
\tag{24}
$$

Beispiel (fortgeführt)

```
# Definitionen
A = matrix(c(2,-3,0,
             1, 6,5),
           nrow  = 2,
           byrow = TRUE)
B = matrix(c( 4,2,
             -1,0,
              1,3),
           nrow  = 3,
           byrow = TRUE)

# Matrixmultiplikation
D = B %*% A
print(D)
```

```
>      [,1] [,2] [,3]
> [1,]   10    0   10
> [2,]   -2    3    0
> [3,]    5   15   15
```

```
# Beispiel für eine undefinierte Matrixmultipliation
E = t(A) %*% B      # (3 x 2)(3 x 2)
```

```
> Error in t(A) %*% B: nicht passende Argumente
```

Theorem (Matrixmultiplikation und Skalarprodukt)

Es seien $x, y \in \mathbb{R}^n$. Dann gilt

$$\langle x, y \rangle = x^T y. \tag{25}$$

Weiterhin seien für $A \in \mathbb{R}^{m \times n}$ für $i = 1, \ldots, m$

$$\bar{a}_i := (a_{ji})_{1 \leq j \leq n} \in \mathbb{R}^n \tag{26}$$

die Spalten von A^T und für $B \in \mathbb{R}^{n \times p}$ für $i = 1, \ldots, p$

$$\bar{b}_j := (b_{ij})_{1 \leq j \leq n} \in \mathbb{R}^n \tag{27}$$

die Spalten von B, also

$$A^T = \begin{pmatrix} \bar{a}_1 & \bar{a}_2 & \cdots & \bar{a}_m \end{pmatrix} \in \mathbb{R}^{n \times m} \text{ und } B = \begin{pmatrix} \bar{b}_1 & \bar{b}_2 & \cdots & \bar{b}_p \end{pmatrix} \in \mathbb{R}^{n \times p}. \tag{28}$$

Dann gilt

$$AB = \left(\langle \bar{a}_i, \bar{b}_j \rangle \right)_{1 \leq i \leq m, 1 \leq j \leq p} \tag{29}$$

Bemerkungen

- Wir verzichten auf einen ausführlichen Beweis.
- Die erste Aussage folgt mit der Identifikation von $\mathbb{R}^n = \mathbb{R}^{n \times 1}$
- Der Eintrag $(AB)_{ij}$ enstpricht dem Skalarprodukt von $i$ter Spalte von A^T und $j$ter Spalte von B.

Motivation für Begriff der Inversen einer quadratischen Matrix

- Es seien $A \in \mathbb{R}^{m \times m}$, $x \in \mathbb{R}^m$ und $b \in \mathbb{R}^m$, A und b seien als bekannt vorausgesetzt, x sei unbekannt.
- Zum Beispiel sei $A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ und $b := \begin{pmatrix} 5 \\ 11 \end{pmatrix}$
- In diesem Fall gilt $Ax = b \Leftrightarrow \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \end{pmatrix} \Leftrightarrow \begin{array}{ll} 1x_1 + 2x_2 & = 5 \\ 3x_1 + 4x_2 & = 11 \end{array}$
- Wir haben also ein *lineares Gleichungssystem (LGS)* mit zwei Gleichungen und zwei Unbekannten.
- Wir stellen uns vor, dass wissen möchten, für welche(s) x das LGS erfüllt ist.
- Wären $A = a \in \mathbb{R}$, $x \in \mathbb{R}$ und $b \in \mathbb{R}$, also $ax = b$ gegeben so würden mit dem *multiplikativem Inversen* von a multiplizieren, also dem Wert, der mit a multipliziert 1 ergibt und durch $a^{-1} = \frac{1}{a}$ gegeben ist.
- Dann würde nämlich gelten $ax = b \Leftrightarrow a^{-1}ax = a^{-1}b \Leftrightarrow 1 \cdot x = a^{-1}b \Leftrightarrow x = \frac{b}{a}$
- Konkret etwa $2x = 6 \Leftrightarrow 2^{-1}2x = 2^{-1}6 \Leftrightarrow \frac{1}{2}2x = \frac{1}{2}6 \Leftrightarrow x = 3$.
- Analog möchte mit dem *multiplikativen Inversen* A^{-1} von A multiplizieren können, sodass "$A^{-1}A = 1$".
- Dann hätte man nämlich $Ax = b \Leftrightarrow A^{-1}Ax = A^{-1}b \Leftrightarrow x = A^{-1}b$
- Die Idee des multiplikativen Inversen wird im folgenden als *Inverse eine quadratischen Matrix* formalisiert.

Definition (Einheitsmatrix)

Die Matrix

$$I_m := (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq m} \in \mathbb{R}^{m \times m} := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \tag{30}$$

mit $a_{ij} = 1$ für $i = j$ und $a_{ij} = 0$ für $i \neq j$ heißt *m-dimensionale Einheitsmatrix*.

- I_m wird in R mit dem Befehl `diag(m)` erzeugt.

Theorem (Neutrales Element der Matrixmultiplikation)

I_m ist das neutrale Element der Matrixmultiplikation, d.h. es gilt für $A \in \mathbb{R}^{m \times n}$, dass

$$I_m A = A \text{ und } A I_n = A. \tag{31}$$

Beweis

Es sei $B = (b_{ij}) = I_m A \in \mathbb{R}^{m \times n}$. Dann gilt für alle $1 \leq i \leq m$ und alle $1 \leq j \leq n$

$$d_{ij} = 0 \cdot a_{1j} + 0 \cdot a_{2j} + \cdots + 0 \cdot a_{i-1,j} + 1 \cdot a_{ij} + + \cdots + 0 \cdot a_{i+1,j} + 0 \cdot a_{mj} = a_{ij} \tag{32}$$

und analog für $A I_n$. □

Definition (Invertierbare Matrix und inverse Matrix)

$A \in \mathbb{R}^{m \times m}$ heißt *invertierbar*, wenn es eine Matrix $A^{-1} \in \mathbb{R}^{m \times m}$ gibt, so dass

$$A^{-1}A = AA^{-1} = I_m \tag{33}$$

ist. Die Matrix A^{-1} heißt die *inverse Matrix von* A.

Bemerkungen

- Invertierbarkeit und inverse Matrizen beziehen sich nur auf quadratische Matrizen.
- Inverse Matrizen heißen auch einfach *Inverse*.
- Quadratische Matrizen können, müssen aber nicht invertierbar sein.
- Nicht invertierbare Matrizen nennt man *singuläre* Matrizen
- Für $A = a \in \mathbb{R}^{1 \times 1}$ gilt $A^{-1} = \frac{1}{a}$.
- Die Definition sagt nur aus, was eine inverse Matrix ist, nicht wie man sie berechnet.

Beispiel für eine invertierbare Matrix

Die Matrix $A = \begin{pmatrix} 2.0 & 1.0 \\ 3.0 & 4.0 \end{pmatrix}$ ist invertierbar mit inverser Matrix $A^{-1} = \begin{pmatrix} 0.8 & -0.2 \\ -0.6 & 0.4 \end{pmatrix}$, denn

$$\begin{pmatrix} 2.0 & 1.0 \\ 3.0 & 4.0 \end{pmatrix} \begin{pmatrix} 0.8 & -0.2 \\ -0.6 & 0.4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.8 & -0.2 \\ -0.6 & 0.4 \end{pmatrix} \begin{pmatrix} 2.0 & 1.0 \\ 3.0 & 4.0 \end{pmatrix}, \tag{34}$$

wovon man sich durch Nachrechnen überzeugt.

Beispiel für eine nicht-invertierbare Matrix

Die Matrix $B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ ist nicht invertierbar, denn wäre B invertierbar, dann gäbe es $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ mit

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{35}$$

Das würde aber bedeuten, dass $0 = 1$ in $\mathbb{R}$ und das ist ein Widerspruch. Also kann B nicht invertierbar sein.

Berechnen inverser Matrizen

- 2×2 bis etwa 5×5 Matrizen kann man prinzipiell per Hand invertieren.
- Dazu lernt man im BSc Mathematik verschiedene Verfahren.
- Wir verzichten auf eine Einführung in die Matrizeninvertierung per Hand.
- Ein kurzes (30 min) Erklärvideo findet sich hier.
- In der Anwendung werden Matrizen standardmäßig numerisch invertiert.
- Matrixinversion ist ein weites Feld in der numerischen Mathematik.
- Es gibt sehr viele Algorithmen zur Invertierung invertierbarer Matrizen.
- Elegant berechnet man inverse Matrizen in R zum Beispiel mit dem Paket `matlib`.

Berechnen inverser Matrizen

```
# Einmalige Installation des R Pakets matlib
install.packages("matlib")
```

```
# Laden der matlib Funktionen
library(matlib)

# Definition
A = matrix(c(2,1,
             3,4),
           nrow  = 2,
           byrow = TRUE)

# Berechnen von A^{-1}
inv(A)
```

```
>      [,1] [,2]
> [1,]  0.8 -0.2
> [2,] -0.6  0.4
```

Berechnen inverser Matrizen

```
print(inv(A) %*% A)
```

```
>           [,1] [,2]
> [1,] 1.00e+00    0
> [2,] 2.22e-16    1
```

```
print(A %*% inv(A))
```

```
>           [,1] [,2]
> [1,] 1.00e+00    0
> [2,] 4.44e-16    1
```

```
# Nicht-invertierbare Matrizen sind auch numerisch nicht-invertierbar (singulär)
B = matrix(c(1,0,
             0,0),
           nrow  = 2,
           byrow = 2)
inv(B)
```

```
> Error in Inverse(X, tol = sqrt(.Machine$double.eps), ...): X is numerically singular
```

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Definition (Determinante)

Für $A = (a_{ij})_{1 \leq i,j \leq m} \in \mathbb{R}^{m \times m}$ mit $m > 1$ sei $A_{ij} \in \mathbb{R}^{m-1 \times m-1}$ die Matrix, die aus A durch Entfernen der $i$ten Zeile und der $j$ten Spalte entsteht. Dann heißt die Zahl

$$\det(A) := a_{11} \qquad \text{für } m = 1 \tag{36}$$

$$\det(A) := \sum_{j=1}^{m} a_{1j}(-1)^{1+j} \det\left(A_{1j}\right) \quad \text{für } m > 1 \tag{37}$$

die *Determinante von* A.

Bemerkungen

- Für

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \tag{38}$$

  ergeben sich zum Beispiel

$$A_{11} = \begin{pmatrix} 5 & 6 \\ 8 & 9 \end{pmatrix}, A_{12} = \begin{pmatrix} 4 & 6 \\ 7 & 9 \end{pmatrix}, A_{21} = \begin{pmatrix} 2 & 3 \\ 8 & 9 \end{pmatrix}, A_{22} = \begin{pmatrix} 1 & 3 \\ 7 & 9 \end{pmatrix} \tag{39}$$

- Determinanten sind nichtlineare Abbildungen der Form $\det : \mathbb{R}^{m \times m} \to \mathbb{R}$, $A \mapsto \det(A)$

Theorem (Determinanten von 2 ×2 und 3 × 3 Matrizen)

(1) Es sei $A = (a_{ij})_{1 \leq i,j \leq 2} \in \mathbb{R}^{2 \times 2}$. Dann gilt

$$\det(A) = a_{11}a_{22} - a_{12}a_{21}. \tag{40}$$

(2) Es sei $A = (a_{ij})_{1 \leq i,j \leq 3} \in \mathbb{R}^{3 \times 3}$. Dann gilt

$$\det(A) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32} - a_{13}a_{22}a_{31}. \tag{41}$$

Bemerkungen

- Für 2×2 und 3×3 Matrizen (und nur für diese) gilt die *Sarrusche Merkregel*

  "Summe der Produkte auf den Diagonalen minus Summe der Produkte auf den Gegendiagonalen"

- Bei 3×3 Matrizen bezieht sich die Merkregel auf das Schema

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & | & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & | & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & | & a_{31} & a_{32} \end{pmatrix} \tag{42}$$

Determinanten

Beweis

Für $A \in \mathbb{R}^{2\times 2}$ gilt nach Definition

$$\begin{aligned}\det(A) &= \sum_{j=1}^{m} a_{1j}(-1)^{1+j} \det\left(A_{1j}\right) \\ &= a_{11}(-1)^{1+1}\det(A_{11}) + a_{12}(-1)^{1+2}\det(A_{12}) \\ &= a_{11}\det((a_{22})) - a_{12}\det((a_{21})) \\ &= a_{11}a_{22} - a_{12}a_{21}\end{aligned} \tag{43}$$

Für $A \in \mathbb{R}^{3\times 3}$ gilt nach Definition und mit der Formel für Determinanten von 2×2 Matrizen

$$\begin{aligned}\det(A) &= \sum_{j=1}^{m} a_{1j}(-1)^{1+j} \det\left(A_{1j}\right) \\ &= a_{11}(-1)^{1+1}\det\left(A_{1j}\right) + a_{12}(-1)^{1+2}\det\left(A_{12}\right) + a_{13}(-1)^{1+3}\det\left(A_{13}\right)) \\ &= a_{11}\det(A_{11}) - a_{12}\det(A_{12}) + a_{13}\det(A_{13}) \\ &= a_{11}\det\left(\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33}\end{pmatrix}\right) - a_{12}\det\left(\begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33}\end{pmatrix}\right) + a_{13}\det\left(\begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32}\end{pmatrix}\right) \\ &= a_{11}\left(a_{22}a_{33} - a_{23}a_{32}\right) - a_{12}\left(a_{21}a_{33} - a_{23}a_{31}\right) + a_{13}\left(a_{21}a_{32} - a_{22}a_{31}\right) \\ &= a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} \\ &= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32} - a_{13}a_{22}a_{31}.\end{aligned} \tag{44}$$

Beispiel 1

Es seien

$$A := \begin{pmatrix} 2 & 1 \\ 3 & 4 \end{pmatrix} \text{ und } B := \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tag{45}$$

Dann ergeben sich

$$\det(A) = 2 \cdot 4 - 1 \cdot 3 = 8 - 3 = 5. \tag{46}$$

und

$$\det(B) = 1 \cdot 0 - 0 \cdot 0 = 0 - 0 = 0. \tag{47}$$

Beispiel 2 Es sei

$$C := \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix} \tag{48}$$

Dann ergibt sich

$$\det(C) = 2 \cdot 1 \cdot 3 + 0 \cdot 0 \cdot 0 + 0 \cdot 0 \cdot 0 - 0 \cdot 0 \cdot 3 - 0 \cdot 0 \cdot 0 - 0 \cdot 1 \cdot 0 = 2 \cdot 1 \cdot 3 = 6 \tag{49}$$

Determinanten

```
# Beispiel 1
A = matrix(c(2,1,                    # Matrixdefinition
             3,4),
           nrow = 2,
           byrow = TRUE)
det(A)                               # Determinantenberechnung
```

```
> [1] 5
```

```
B = matrix(c(1,0,                    # Matrixdefinition
             0,0),
           nrow = 2,
           byrow = TRUE)
det(B)                               # Determinantenberechnung
```

```
> [1] 0
```

```
# Beispiel 2
C = matrix(c(2,0,0,                  # Matrixdefinition
             0,1,0,
             0,0,3),
           nrow = 3,
           byrow = TRUE)
det(C)                               # Determinantenberechnung
```

```
> [1] 6
```

Theorem (Rechenregeln für Determinanten)

Determinantenmultiplikationssatz

- Für $A, B \in \mathbb{R}^{m \times m}$ gilt

$$\det(AB) = \det(A)\det(B). \tag{50}$$

Transposition

- Für $A \in \mathbb{R}^{m \times m}$ gilt

$$\det(A) = \det\left(A^T\right). \tag{51}$$

Dreiecksmatrizen

- Für Matrizen $A = (a_{ij})_{1 \leq i,j \leq m} \in \mathbb{R}^{m \times m}$ mit $a_{ij} = 0$ für $i > j$ oder $a_{ij} = 0$ für $j > i$ gilt

$$\det(A) = \prod_{i=1}^{m} a_{ii} \tag{52}$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Bei Dreiecksmatrizen sind alle Elemente unterhalb ($i > j$) oder oberhalb ($j > i$) der Diagonalen 0
- Bei I_m sind alle nicht-diagonalen Elemente 0 und alle diagonalen Elemente 1, also folgt $\det(I_m) = 1$.

Theorem (Invertierbarkeit und Determinante)

$A \in \mathbb{R}^{m \times m}$ ist dann und nur dann invertierbar, wenn gilt, dass $\det(A) \neq 0$. Es gilt also

$$A \text{ ist invertierbar} \Leftrightarrow \det(A) \neq 0 \text{ und} A \text{ ist nicht invertierbar} \Leftrightarrow \det(A) = 0. \tag{53}$$

Beweisandeutung

Wir zeigen lediglich, dass aus der Invertierbarkeit von A folgt, dass $\det(A)$ nicht null sein kann. Nehmen wir also an, dass A ist invertierbar. Dann gibt es eine Matrix B mit $AB = I_m$ und mit dem Determinantenmultiplikationssatz folgt

$$\det(AB) = \det(A)\det(B) = \det(I_m) = 1. \tag{54}$$

Also kann $\det(A) = 0$ nicht gelten, denn sonst wäre $0 = 1$.

□

Bemerkung

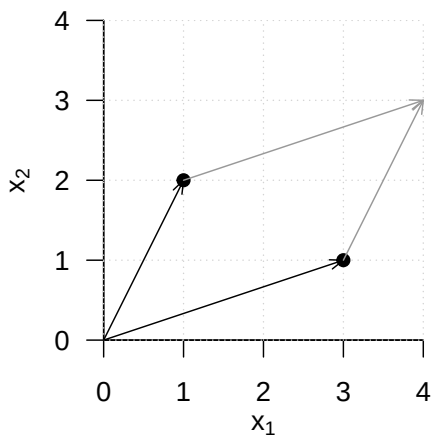
- A ist nicht invertierbar $\Leftrightarrow \det(A) = 0$ ist für die Eigenanalyse essentiell.

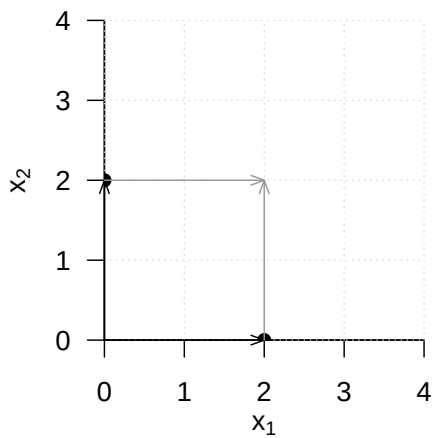
Visuelle Intuition

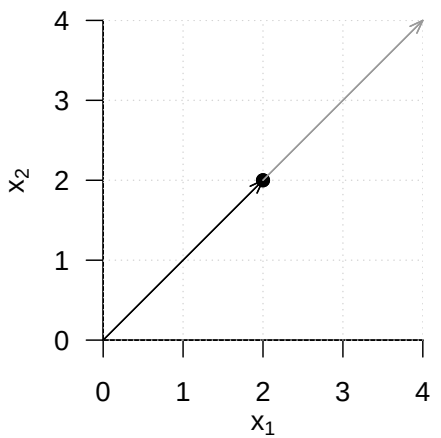
$a_1, ..., a_m \in \mathbb{R}^m$ seien die Spalten von $A \in \mathbb{R}^{m \times m}$.

$\Rightarrow \det(A)$ enstpricht dem signierten Volumen des von $a_1, ..., a_m \in \mathbb{R}^m$ aufgespannten Parallelotops.

$$A_1 = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix} \qquad A_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \qquad A_3 = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$







$\det(A_1) = 3 \cdot 2 - 1 \cdot 1 = 5$ $\qquad$ $\det(A_2) = 2 \cdot 2 - 0 \cdot 0 = 4$ $\qquad$ $\det(A_3) = 2 \cdot 2 - 2 \cdot 2 = 0$

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Definition (Nullmatrizen, Einheitsmatrizen, Einheitsvektoren, Einsvektoren)

- Wir bezeichnen *Nullmatrizen* mit

$$0_{mn} := (0)_{1 \leq i \leq m, 1 \leq j \leq n} \in \mathbb{R}^{m \times n} \text{ und } 0_m := (0)_{1 \leq i \leq m} \in \mathbb{R}^m \tag{55}$$

- Wir bezeichnen die *Einheitsmatrix* mit

$$I_m := (i_{jk})_{1 \leq i \leq m, 1 \leq j \leq n} \in \mathbb{R}^{m \times n} \text{ mit } i_{jk} = 1 \text{ für } j = k \text{ und } i_{jk} = 0 \text{ für } j \neq k \tag{56}$$

- Wir bezeichnen die *Einheitsvektoren* $e_1, i = 1, \ldots, m$ mit

$$e_i := (e_{i_j})_{1 \leq j \leq m} \in \mathbb{R}^m \text{ mit } e_{i_j} = 1 \text{ für } i = j \text{ und } e_{i_j} = 0 \text{ für } i \neq j \tag{57}$$

- Wir bezeichnen den *Einsvektor* mit

$$1_m := (1)_{1 \leq i \leq m} \in \mathbb{R}^m \tag{58}$$

Bemerkungen

- 0_{mn} und 0_m bestehen nur aus Nullen.
- I_m besteht nur aus Nullen und Diagonalelementen gleich Eins.
- $e_i, i1, \ldots., m$ besteht nur aus Nullen und einer Eins in der $i$ten Komponente.
- 1_m besteht nur aus Einsen.

Definition (Symmetrische, diagonale, und orthogonale Matrizen)

- Eine Matrix $S \in \mathbb{R}^{m \times m}$ heißt *symmetrisch*, wenn gilt dass $S^T = S$.
- Eine Matrix $D \in \mathbb{R}^{m \times m}$ heißt *Diagonalmatrix*, wenn $d_{ij} = 0$ für $1 \leq i, j \leq m, i \neq j$.
- Eine Matrix $Q \in \mathbb{R}^{m \times m}$ heißt *orthogonal*, wenn ihre Spaltenvektoren wechselseitig *orthonormal* sind.

Bemerkungen

- Eine Diagonalmatrix D mit Diagonalelementen $d_1, \ldots, d_n$ schreibt man auch als $D = \text{diag}(d_1, \ldots, d_n)$
- Symmetrische, diagonale, und orthogonale Matrizen haben viele "gute" Eigenschaften.
- Im folgenden wichtige Eigenschaften sind
  - $D := \text{diag}(d_1, \ldots, d_n)$ ist eine Diagonalmatrix $\Rightarrow \det(D) = \prod_{i=1}^{n} d_i$.
  - Q ist orthogonal $\Rightarrow Q^{-1} = Q^T$ (weil $Q^T Q = Q Q^T = I_m$).

Definition (Positiv-definite und positiv-semidefinite Matrizen)

- Eine Matrix $C \in \mathbb{R}^{m \times m}$ heißt *positiv-definit*, wenn für alle $x \in \mathbb{R}^m$ mit $x \neq 0_m$ gilt, dass

$$x^T A x > 0. \tag{59}$$

- Eine Matrix $C \in \mathbb{R}^{m \times m}$ heißt *positiv-semidefinit*, wenn für alle $x \in \mathbb{R}^m$ mit $x \neq 0_m$ gilt, dass

$$x^T A x \geq 0. \tag{60}$$

Bemerkungen

- Positiv-definite und positiv-semidefinite Matrizen haben viele "gute" Eigenschaften.
- Im folgenden wichtige Eigenschaften sind
  - C ist positiv-definit $\Rightarrow \det(C) > 0$ und C ist invertierbar.
  - C ist positiv-definit $\Rightarrow$ Es gibt eine Matrix $K \in \mathbb{R}^{m \times m}$ mit $C = KK^T$.
  - C ist positiv-definit $\Rightarrow$ Alle Eigenwerte von C sind positiv.

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Definition (Spaltenraum und Rang einer Matrix)

Es sei $A \in \mathbb{R}^{m \times n}$ und

$$a_1 := \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix}, a_2 := \begin{pmatrix} a_{12} \\ \vdots \\ a_{m2} \end{pmatrix}, \ldots, a_n := \begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix} \in \mathbb{R}^m \tag{61}$$

seien die *Spalten(vektoren)* von A. Dann heißt

$$\text{col}(A) := \left\{ \sum_{i=1}^{n} c_i a_i | c_i \text{ mit } i = 1, \ldots, n \in \mathbb{R} \right\} \tag{62}$$

der *Spaltenraum* von A. Die Dimension von col(A) heißt *der Rang von* A. Ist die Dimension von col(A) gleich n, so sagt man, dass A *vollen Spaltenrang hat*.

Bemerkungen

- col(A) ist die Menge aller Linearkombinationen der Spaltenvektoren von A
- Die Dimension von col(A) entspricht der Anzahl der linear unabhängigen Spaltenvektoren von A.
- Mit $c := (c_i)_{1 \leq i \leq n} \in \mathbb{R}^n$ gilt col$(A) = \{Ac | c \in \mathbb{R}^n\}$.

Selbstkontrollfragen

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Definition (Eigenvektor, Eigenwert)

$A \in \mathbb{R}^{m \times m}$ sei eine quadratische Matrix. Dann heißt jeder Vektor $v \in \mathbb{R}^m, v \neq 0$, für den gilt, dass

$$Av = \lambda v \tag{63}$$

mit $\lambda \in \mathbb{R}$ ein *Eigenvektor* von A. λ heißt zugehöriger *Eigenwert* von A.

Bemerkungen

- Ein Eigenvektor v von A wird durch A mit einem Faktor λ verlängert.
- Jeder Eigenvektor hat einen zugehörigen Eigenwert.
- Die Eigenwerte verschiedener Eigenvektor können identisch sein.

Theorem

$A \in \mathbb{R}^{m \times m}$ sei eine quadratische Matrix. Wenn $v \in \mathbb{R}^m$ Eigenvektor von A mit Eigenwert $\lambda \in \mathbb{R}$ ist, dann ist auch $av \in \mathbb{R}^m$ mit $a \in \mathbb{R}$ Eigenvektor von A und zwar mit Eigenwert $a\lambda \in \mathbb{R}$.

Beweis

Es gilt

$$Av = \lambda v \Leftrightarrow a(Av) = a(\lambda)v \Leftrightarrow A(av) = (a\lambda)v \tag{64}$$

Also ist av ein Eigenvektor von A mit Eigenwert $a\lambda$.

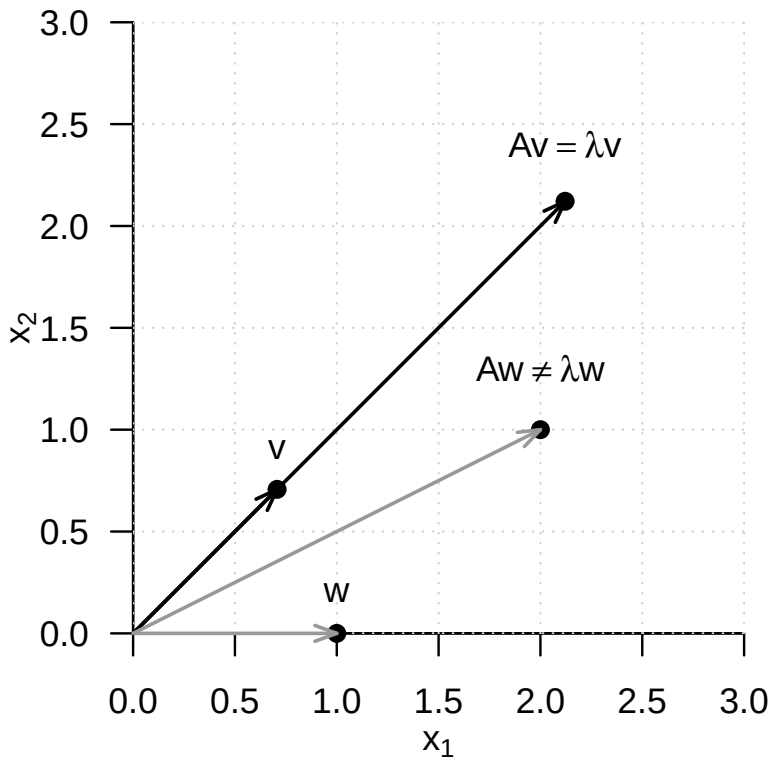
□

Konvention

Wir betrachten im Folgenden nur Eigenvektoren mit $\|v\| = 1$.

Visualisierung eines Eigenvektors

Für $A := \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ ist $v := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ Eigenvektor zum Eigenwert $\lambda = 3$, $w := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ist kein Eigenvektor.



Theorem (Bestimmung von Eigenwerten und Eigenvektoren)

$A \in \mathbb{R}^{m \times m}$ sei eine quadratische Matrix. Dann ergeben sich die Eigenwerte von A als die Nullstellen des *charakteristischen Polynoms*

$$\chi_A(\lambda) := \det(A - \lambda I_m). \tag{65}$$

von A. Weiterhin seien $\lambda_i^*, i = 1, 2, \ldots$ die auf diese Weise bestimmten Eigenwerte von A. Die entsprechenden Eigenvektoren $v_i, i = 1, 2, \ldots$ von A können dann durch Lösen der linearen Gleichungssysteme

$$(A - \lambda_i^* I_m) v_i = 0_m \text{ für } i = 1, 2, \ldots \tag{66}$$

bestimmt werden.

Bemerkungen

- Für kleine Matrizen mit $m \leq 3$ können Eigenwerte und Eigenvektoren manuell bestimmt werden.
- Bei großen Matrizen werden Eigenwerte und Eigenvektor im Allgemeinen numerisch bestimmt.
- R's `eigen()`, Scipy's `linalg.eig()`, Matlab's `eig()`.

Beweis

(1) Bestimmen von Eigenwerten

Wir halten zunächst fest, dass mit der Definition von Eigenvektoren und Eigenwerten gilt, dass

$$Av = \lambda v \Leftrightarrow Av - \lambda v = 0_m \Leftrightarrow (A - \lambda I_m)v = 0_m\,. \tag{67}$$

Für den Eigenwert λ wird der Eigenvector v also durch $(A - \lambda I_m)$ auf den Nullvektor 0_m abgebildet. Weil aber per Definition $v \neq 0_m$ gilt, ist die Matrix $(A - \lambda I_m)$ somit nicht invertierbar: sowohl der Nullvektor als auch v werden durch A auf 0_m abgebildet, die Abbildung

$$f : \mathbb{R}^m \to \mathbb{R}^m, x \mapsto (A - \lambda I_m)x \tag{68}$$

ist also nicht bijektiv, und $(A - \lambda I_m)^{-1}$ kann nicht existieren. Die Tatsache, dass $(A - \lambda I_m)$ nicht invertierbar ist, ist aber äquivalent dazu, dass die Determinante von $(A - \lambda I_m)$ Null ist. Also ist

$$\chi_A(\lambda) = \det(A - \lambda I_m) = 0 \tag{69}$$

notwendige und hinreichende Bedingung dafür, dass λ ein Eigenwert von A ist.

(2) Bestimmen von Eigenvektoren

Es sei λ_i^* ein Eigenwert von A. Dann gilt mit den obigen Überlegungen, dass Auflösen von

$$(A - \lambda_i^* I_m)v_i^* = 0_m \tag{70}$$

nach v_i^* einen Eigenvektor zum Eigenwert λ^* ergibt. □

Beispiel 1

Es sei

$$A := \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{71}$$

Wir wollen die Eigenwerte und Eigenvektoren von A bestimmen.

(1) Berechnen von Eigenwerten

Die Eigenwerte von A sind die Nullstellen des charakteristischen Polynoms von A.

Das charakteristische Polynom von A ergibt als

$$\chi_A(\lambda) = \det\left(\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right) = \det\begin{pmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{pmatrix} = (2-\lambda)^2 - 1. \tag{72}$$

Nullsetzen und Auflösen nach λ ergibt mit der pq-Formel

$$(2-\lambda)^2 - 1 = 0 \Rightarrow \lambda_1 = 3, \lambda_2 = 1. \tag{73}$$

Die Eigenwerte von A sind also $\lambda_1 = 3$ und $\lambda_2 = 1$.

Beispiel 1 (fortgeführt)

(2) Berechnen von Eigenvektoren

Die Eigenvektoren zu den Eigenwerten $\lambda_1 = 3$ und $\lambda_2 = 1$ ergeben sich durch Lösen der linearen Gleichungssysteme

$$(A - \lambda_i I_2)v_i = 0_2 \tag{74}$$

Für $\lambda_1 = 3$ ergibt sich

$$(A - 3I_2)v_1 = 0_2 \Leftrightarrow \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow v_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \text{ ist eine Lösung.} \tag{75}$$

Fpr $\lambda_2 = 1$ ergibt sich

$$(A - 1I_2)v_2 = 0_2 \Leftrightarrow \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow v_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \text{ ist eine Lösung.} \tag{76}$$

Weiterhin gilt $v_1^T v_2 = 0$ und $||v_1||_2 = ||v_2||_2 = 1$.

Eigenanalyse

```
# Matrixdefinition
A = matrix(c(2,1,
             1,2),
           nrow  = 2,
           byrow = TRUE)

# Eigenanalyse
eigen(A)
```

```
> eigen() decomposition
> $values
> [1] 3 1
>
> $vectors
>      [,1]   [,2]
> [1,] 0.707 -0.707
> [2,] 0.707  0.707
```

Theorem (Eigenwerte und Eigenvektoren symmetrischer Matrizen)

Eine symmetrische Matrix $S \in \mathbb{R}^{m \times m}$ hat m verschiedene Eigenwerte $\lambda_1, ..., \lambda_m$ mit zugehörigen orthogonalen Eigenvektoren $q_1, ..., q_m \in \mathbb{R}^m$.

Bemerkungen

- Das Theorem ist eine Konsequenz aus dem Spektralsatz der Linearen Algebra.
- Ein vollständiger Beweis findet sich in Strang (2009), Section 6.4.

Teilbeweis

Wir setzen die Tatsache, dass S m verschiedene Eigenwerte hat, als gegeben voraus und zeigen lediglich, dass die Eigenvektoren von S orthogonal sind. Ohne Beschränkung der Allgemeinheit seien also λ_i und λ_j mit $1 \leq i, j \leq m$ znd $\lambda_i \neq \lambda_j$ zwei der m verschiedenen Eigenwerte von S mit zugehörigen Eigenvektoren q_i und q_j, respektive. Dann ergibt sich

$$Sq_i = \lambda_i q_i \Leftrightarrow (Sq_i)^T = (\lambda_i q_i)^T \Leftrightarrow q_i^T S = q_i^T \lambda_i \Leftrightarrow q_i^T S q_j = \lambda_i q_i^T q_j \,. \tag{77}$$

Ähnlicherweise gilt

$$Sq_j = \lambda_j q_j \Leftrightarrow q_i^T S q_j = \lambda_j q_i^T q_j \,. \tag{78}$$

Also folgt

$$\lambda_i q_i^T q_j = \lambda_j q_i^T q_j \text{ mit } q_i \neq 0, q_j \neq 0, \text{ und } \lambda_i \neq \lambda_j \tag{79}$$

und damit die Orthogonalität $q_i^T q_j = 0$. □

Theorem (Orthonormale Zerlegung einer symmetrischen Matrix)

$S \in \mathbb{R}^{m \times m}$ sei eine symmetrische Matrix. Dann kannn S geschrieben werden als

$$S = Q \Lambda Q^T, \tag{80}$$

wobei $Q \in \mathbb{R}^{m \times m}$ eine orthogonale Matrix ist und $\Lambda \in \mathbb{R}^{m \times m}$ eine Diagonalmatrix ist.

Beweis

Weil S symmetrisch ist, hat sie m verschiedene Eigenwerte $\lambda_i, i = 1, \ldots, m$ und m zugehörige orthogonale Eigenvektoren $q_i, i = 1, \ldots, m$, so dass

$$S q_i = \lambda_i q_i \text{ for } i = 1, \ldots, m. \tag{81}$$

Mit den Definitionen

$$Q := \begin{pmatrix} q_1 & q_2 & \cdots & q_m \end{pmatrix} \text{ und } \Lambda := \text{diag} \begin{pmatrix} \lambda_1, \lambda_2, \ldots, \lambda_m \end{pmatrix}, \tag{82}$$

folgt dann

$$SQ = \Lambda Q \Leftrightarrow SQ = Q\Lambda. \tag{83}$$

Rechtseitige Multiplikation mit Q^T ergibt dann

$$SQQ^T = Q\Lambda Q^T \Leftrightarrow SI_m = Q\Lambda Q^T \Leftrightarrow S = Q\Lambda Q^T \tag{84}$$

und damit ist alles gezeigt. □

Beispiel 1 (fortgeführt)

Für

$$Q := \begin{pmatrix} v_1 & v_2 \end{pmatrix} \text{ and } \Lambda = \text{diag}(\lambda_1, \lambda_2) \tag{85}$$

ergibt sich

$$\begin{aligned}
Q\Lambda Q^T &= \begin{pmatrix} v_1 & v_2 \end{pmatrix} \text{diag}(\lambda_1, \lambda_2) \begin{pmatrix} v_1 & v_2 \end{pmatrix}^T \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 3 & 1 \\ 3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix} \\
&= \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \\
&= A
\end{aligned}$$

Definition

Operationen

Determinanten

Spezielle Matrizen

Spaltenraum und Rang

Eigenanalyse

Selbstkontrollfragen

Selbstkontrollfragen

1. Geben Sie Definition einer Matrix wieder.
2. Nennen Sie sechs Matrixoperationen.
3. Geben Sie Definitionen der Matrixaddition und -subtraktion wieder.
4. Geben Sie die Definition der Skalarmultiplikation für Matrizen wieder.
5. Geben Sie die Definition der Matrixtransposition wieder.
6. Es seien
$$A := \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}, B := \begin{pmatrix} 3 & 0 \\ 1 & 2 \end{pmatrix}, \text{ und } c := 2 \tag{86}$$
Berechnen Sie
$$D := c\left(A - B^T\right) \text{ und } E := (cA)^T + B. \tag{87}$$
per Hand und überprüfen Sie Ihre Rechnung mit R.
7. Geben Sie die Definition der Matrixmultiplikation wieder.
8. Es seien $A \in \mathbb{R}^{3\times 2}$, $B \in \mathbb{R}^{2\times 4}$ und $C \in \mathbb{R}^{3\times 4}$. Prüfen Sie, ob folgende Matrixprodukte definiert sind, und wenn ja, geben Sie die Größe der resultierenden Matrix and
$$ABC, \quad ABC^T, \quad , A^TCB^T \quad , BAC \tag{88}$$

Selbstkontrollfragen

9. Es seien

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 0 \end{pmatrix} \; B := \begin{pmatrix} 1 & 2 & 2 \\ 1 & 3 & 1 \\ 2 & 0 & 0 \end{pmatrix} \text{ und } C := \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \tag{89}$$

   Berechnen Sie die Matrixprodukte

$$AB, \qquad B^T A^T, \qquad \left(B^T A^T\right)^T, \qquad AC \tag{90}$$

   per Hand und überprüfen Sie Ihre Rechnung mit R.

10. Invertieren Sie die Matrizen A und B aus der vorherigen Aufgabe mithilfe von matlib::inv und überprüfen Sie die Inverseeigenschaft der inversen Matrizen mithilfe von R.

11. Geben Sie die Formel für die Determinante von $A := (A_{ij})_{1 \leq i,j \leq 2} \in \mathbb{R}^2$ wieder.

12. Geben Sie die Formel für die Determinante von $A := (A_{ij})_{1 \leq i,j \leq 3} \in \mathbb{R}^3$ wieder.

13. Berechnen Sie die Determinanten von

$$A := \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \; B := \begin{pmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{pmatrix} \text{ und } C := \text{diag}(1, 2, 3) \tag{91}$$

   per Hand und überprüfen Sie Ihre Rechnung mit R.

Selbstkontrollfragen

14. Geben Sie den Determinantenmultiplikationssatz wieder.
15. Geben Sie das Theorem zur Invertierbarkeit und Determinante von Matrizen wieder.
16. Geben Sie die Definition einer symmetrischen Matrix wieder.
17. Geben Sie die Definition einer Diagonalmatrix wieder.
18. Geben Sie die Definition einer orthogonalen Matrix wieder.
19. Geben Sie die Definition einer positiv-definiten und einer positiv-semidefiniten Matrix wieder.
20. Geben Sie die Definition des Spaltenraums einer Matrix wieder.
21. Wann sagt man, dass eine Matrix vollen Spaltenrang hat?
22. Geben Sie die Definition eines Eigenvektors und eines Eigenwertes einer quadratischen Matrix wieder.
23. Geben Sie das Theorem zur Bestimmung von Eigenwerten und Eigenvektoren wieder.
24. Geben Sie das Theorem zu den Eigenwerten und Eigenvektoren symmetrischer Matrizen wieder.
25. Geben Sie das Theorem zur orthonormalen Zerlegung einer symmetrischen Matrix wieder.
26. Dokumentieren Sie die in dieser Einheit eingeführten R Befehle in einem R Skript.

References

Strang, Gilbert. 2009. *Introduction to Linear Algebra*.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(3) Wahrscheinlichkeitstheorie

Realisierungen von Zufallsvariablen

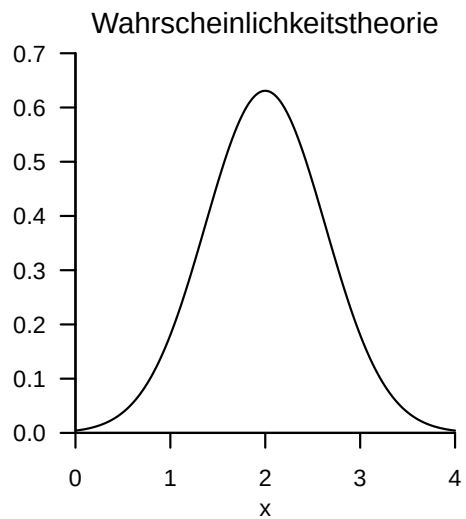
```
# Univariate Normalverteilungsparameter
mu      = 2.0                          # Erwartungswertparameter
sigsqr  = 0.5                          # Varianzparameter
n       = 10                           # Anzahl von u.i.v. Realisierungen

# 10 Realisierungen
X       = rnorm(n,mu,sqrt(sigsqr))     # X_i \sim N(\mu,\sigma^2), i = 1,...,n
print(X)
```

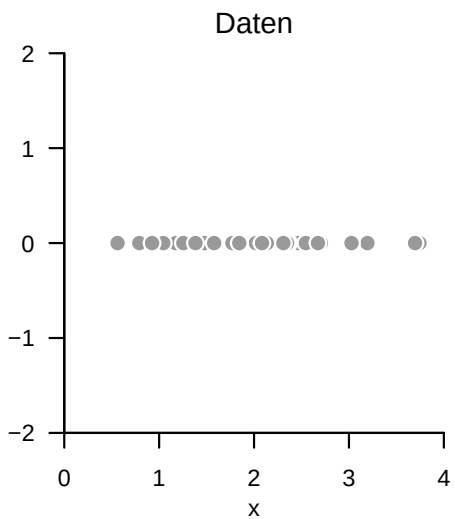
```
>  [1] 1.010 2.181 0.277 1.996 2.440 2.812 0.712 1.825 1.827 1.800
```

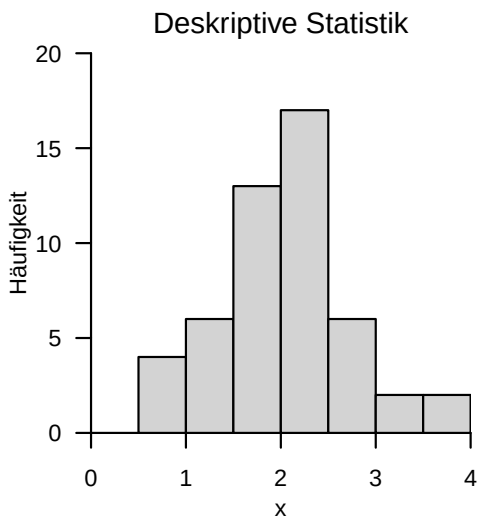
```
# 10 Realisierungen
X       = rnorm(n,mu,sqrt(sigsqr))     # X_i \sim N(\mu,\sigma^2), i = 1,...,n
print(X)
```

```
>  [1] 1.608 2.445 3.460 0.847 2.362 0.683 1.631 1.963 2.384 1.354
```

Wahrscheinlichkeitstheorie, Daten, Deskriptive Statistik







Realisierungen von Zufallsvektoren

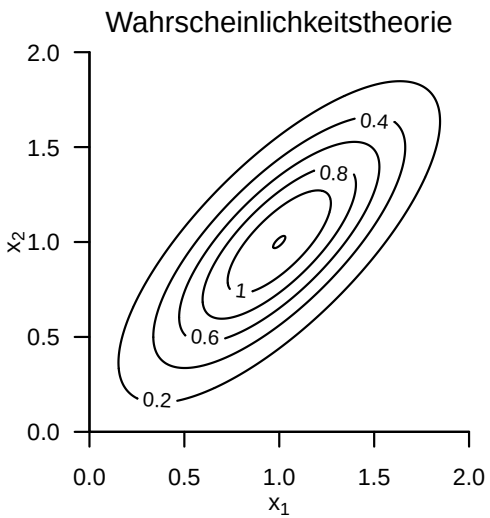
```
# R Paket für multivariate Normalverteilungsrealisierung
library(MASS)

# Multivariate Normalverteilungsparameter
mu      = c(2.0,5.0)                  # Erwartungswertparameter
Sigma   = matrix(c(0.5,0.1,           # Kovarianzmatrixparamter
                   0.1,0.5),
                 nrow = 2,
                 byrow = TRUE)
n       = 10                          # Anzahl von u.i.v. Realisierungen

# 10 Realisierungen
X       = t(mvrnorm(n,mu,Sigma))      # X_i \sim N(\mu,\Sigma), i = 1,...,n
print(X)
```

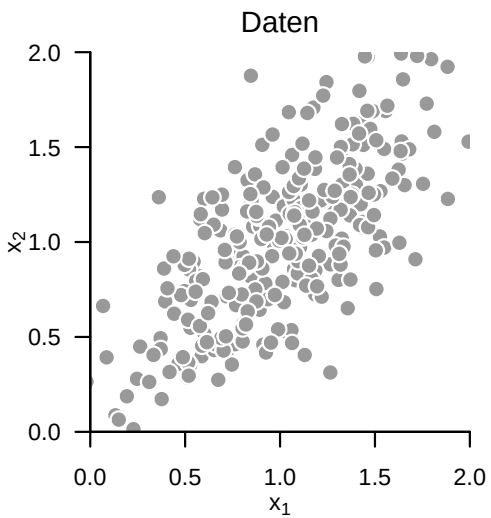
```
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
> [1,] 1.84 2.12 1.18 1.68 2.98 2.01 2.34 2.12 1.69 0.694
> [2,] 5.67 5.28 4.39 6.13 6.08 4.89 3.63 4.87 4.41 4.649
```

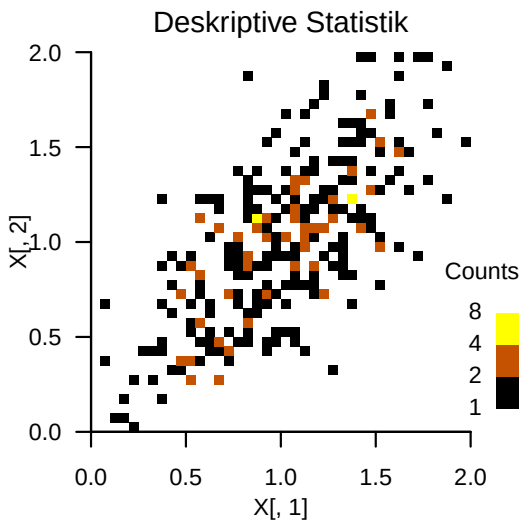
```
# 10 Realisierungen
X       = t(mvrnorm(n,mu,Sigma))      # X_i \sim N(\mu,\sigma^2), i = 1,...,n
print(X)
```

```
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
> [1,] 1.65 1.76 2.06 1.85 2.23 3.26 3.04 1.12 2.23 0.857
> [2,] 5.42 4.54 4.88 4.87 5.26 6.76 4.01 6.52 4.90 4.048
```

Multivariate Wahrscheinlichkeitstheorie, Multivariate Daten, Multivariate Deskriptive Statistik







Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Definition (Zufallsvektor)

$(\Omega, \mathcal{A}, \mathbb{P})$ sei ein Wahrscheinlichkeitsraum und $(\mathcal{X}, \mathcal{S})$ sei ein m-dimensionaler Messraum. Ein m-dimensionaler *Zufallsvektor* ist definiert als eine Abbildung

$$X : \Omega \to \mathcal{X}, \omega \mapsto X(\omega) := \begin{pmatrix} X_1(\omega) \\ \vdots \\ X_m(\omega) \end{pmatrix} \tag{1}$$

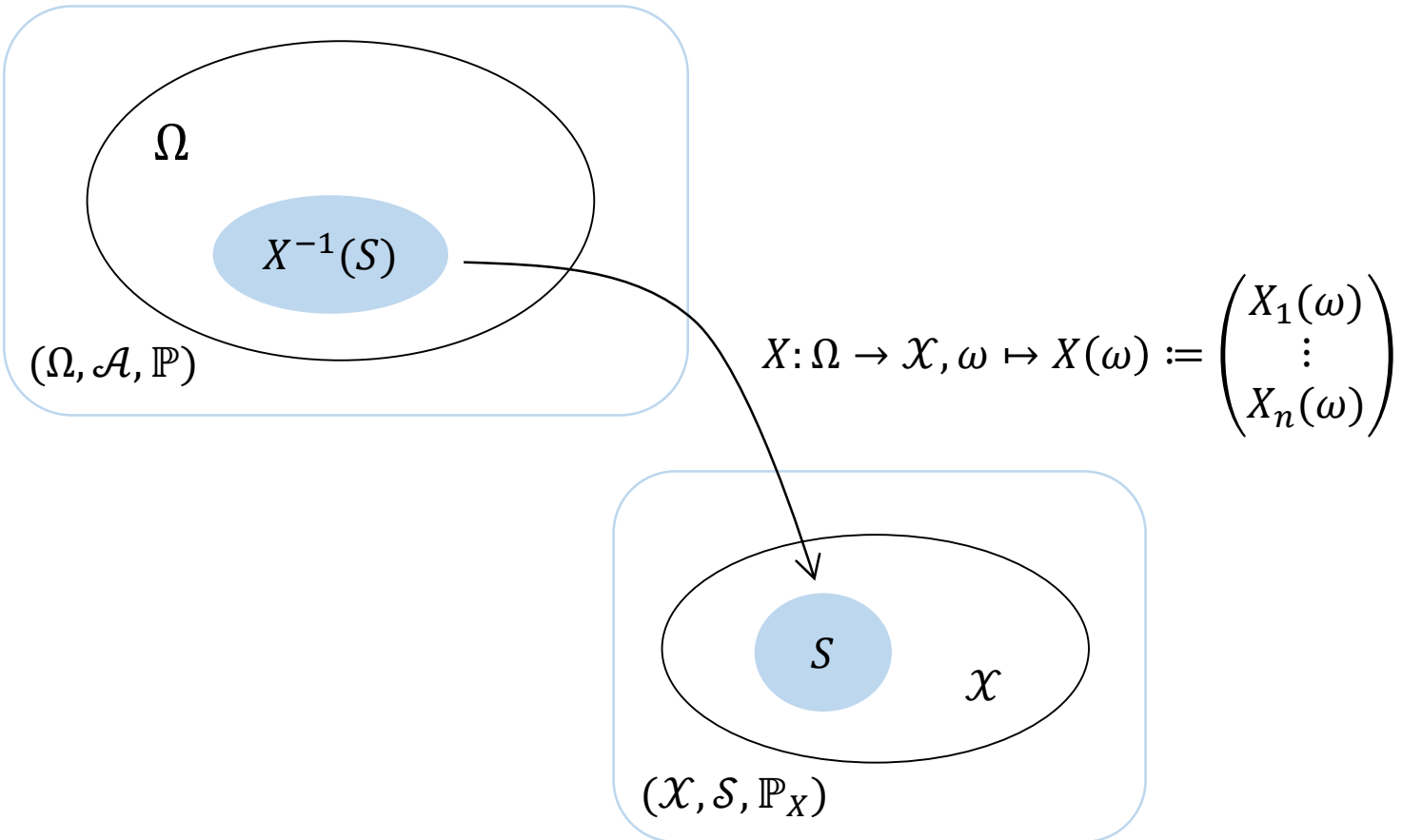
mit der *Messbarkeitseigenschaft*

$$\{\omega \in \Omega | X(\omega) \in S\} \in \mathcal{A} \text{ für alle } S \in \mathcal{S}. \tag{2}$$

Bemerkungen

- X ist messbar, wenn die Komponentenfunktionen $X_1, ..., X_m$ messbar sind.
- Die Komponentenfunktionen eines Zufallsvektors sind Zufallsvariablen.
- Ein m-dimensionaler Zufallsvektor ist die Konkatenation von m Zufallsvariablen.
- Für einen Zufallsvektor schreiben wir auch häufig $X := (X_1, ..., X_m)$.
- Für $m := 1$ ist ein Zufallsvektor eine Zufallsvariable.

Zufallsvektoren und multivariate Verteilungen



$$\mathbb{P}\left(X^{-1}(S)\right) = \mathbb{P}(\{\omega \in \Omega | X(\omega) \in S\}) =: \mathbb{P}_X(S)$$

Definition (Multivariate Verteilung)

$(\Omega, \mathcal{A}, \mathbb{P})$ sei ein Wahrscheinlichkeitsraum, $(\mathcal{X}, \mathcal{S})$ sei ein m-dimensionaler Messraum und

$$X : \Omega \to \mathcal{X}, \omega \mapsto X(\omega) \tag{3}$$

sei ein Zufallsvektor. Dann heißt das Wahrscheinlichkeitsmaß $\mathbb{P}_X$, definiert durch

$$\mathbb{P}_X : \mathcal{S} \to [0,1], S \mapsto \mathbb{P}_X(S) := \mathbb{P}(X^{-1}(S)) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) \in S\}\right) \tag{4}$$

die *multivariate Verteilung des Zufallsvektor* X.

Bemerkungen

- Der Einfachheit halber spricht man oft auch nur von "der Verteilung des Zufallsvektors X".
- Die Notationskonventionen für Zufallsvariablen gelten für Zufallsvektoren analog, z.B.

$$\begin{aligned}
\mathbb{P}_X(X \in S) &:= \mathbb{P}\left(\{X \in S\}\right) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) \in S\}\right) \\
\mathbb{P}_X(X = x) &:= \mathbb{P}\left(\{X = x\}\right) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) = x\}\right) \\
\mathbb{P}_X(X \leq x) &:= \mathbb{P}\left(\{X \leq x\}\right) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) \leq x\}\right) \\
\mathbb{P}_X(x_1 \leq X \leq x_2) &:= \mathbb{P}\left(\{x_1 \leq X \leq x_2\}\right) = \mathbb{P}\left(\{\omega \in \Omega | x_1 \leq X(\omega) \leq x_2\}\right)
\end{aligned} \tag{5}$$

- Relationsoperatoren wie $\leq$ werden hier *komponentenweise* verstanden.
- Zum Beispiel heißt $x \leq y$ für $x, y \in \mathbb{R}^m$, dass $x_i \leq y_i$ für alle $i = 1, \ldots, m$.

Definition (Multivariate kumulative Verteilungsfunktionen)

X sei ein Zufallsvektor mit Ergebnisraum $\mathcal{X}$. Dann heißt eine Funktion der Form

$$P_X : \mathcal{X} \to [0,1],\ x \mapsto P_X(x) := \mathbb{P}_X(X \leq x) \tag{6}$$

multivariate kumulative Verteilungsfunktion von X.

Bemerkung

- Multivariate kumulative Verteilungsfunktionen können zur Definition von multivariaten Verteilungen genutzt werden, häufiger ist allerdings die Definition multivariater Verteilungen durch multivariate Wahrscheinlichkeitsmasse- oder Wahrscheinlichkeitsdichtefunktionen.

Definition (Diskreter Zufallsvektor, multivariate WMF)

$(\Omega, \mathcal{A}, \mathbb{P})$ sei ein Wahrscheinlichkeitsraum und $X : \Omega \to \mathcal{X}$ ein Zufallsvektor. X heißt *diskreter Zufallsvektor*, wenn der Ergebnisraum $\mathcal{X}$ endlich viele oder höchsten abzählbar viele Elemente $x_i, i = 1, 2, \ldots$ enthält. Die *multivariate Wahrscheinlichkeitsmassefunktion (WMF)* eines diskreten Zufallsvektors X wird mit p_X bezeichnet und ist definiert durch

$$p_X : \mathcal{X} \to [0, 1], x \mapsto p_X(x) := \mathbb{P}_X(X = x). \tag{7}$$

Bemerkungen

- Der Begriff der multivariaten WMF ist analog zum Begriff der WMF.
- Man spricht oft einfach von der WMF eines Zufallsvektors.
- Wie univariate WMFen sind multivariate WMFen nicht-negativ und normiert.

Beispiel (Multivariate Wahrscheinlichkeitsmassefunktion)

Wir betrachten einen zweidimensionalen Zufallsvektor $X := (X_1, X_2)$ der Werte in $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2$ annimmt, wobei $\mathcal{X}_1 := \{1, 2, 3\}$ und $\mathcal{X}_2 = \{1, 2, 3, 4\}$ seien.

Eine exemplarische bivariate WMF der Form

$$p_X : \{1, 2, 3\} \times \{1, 2, 3, 4\} \to [0, 1], (x_1, x_2) \mapsto p_X(x_1, x_2) \tag{8}$$

ist dann durch nachfolgende Tabelle definiert.

$p_X(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$
$x_1 = 1$	0.1	0.0	0.2	0.1
$x_1 = 2$	0.1	0.2	0.0	0.0
$x_1 = 3$	0.0	0.1	0.1	0.1

Man beachte, dass $\sum_{x_1=1}^{3} \sum_{x_2=1}^{4} p_X(x_1, x_2) = 1$.

Definition (Kontinuierlicher Zufallsvektor, multivariate WDF)

Es sei $(\Omega, \mathcal{A}, \mathbb{P})$ ein Wahrscheinlichkeitsraum. Ein Zufallsvektor der Form $X : \Omega \to \mathbb{R}^m$ heißt *kontinuierlicher* Zufallsvektor. Die *multivariate Wahrscheinlichkeitsdichtefunktion (WDF)* eines kontinuierlichen Zufallsvektors X ist eine Funktion

$$p_X : \mathbb{R}^m \to \mathbb{R}_{\geq 0}, x \mapsto p_X(x), \tag{9}$$

mit den Eigenschaften

(1) $\int_{\mathbb{R}^m} p_X(x)\, dx = 1$

(2) $\mathbb{P}_X(x_1 \leq X \leq x_2) = \int_{x_{1_1}}^{x_{2_1}} \cdots \int_{x_{1_m}}^{x_{2_m}} p_X(s_1, ..., s_m)\, ds_1 \cdots ds_m$

Bemerkungen

- Der Begriff der multivariaten WDF ist analog zum Begriff der WDF.
- Man spricht häufig auch einfach von der WDF eines Zufallsvektors
- Wie univariate WDFen sind multivariate WDFen nicht-negativ und normiert.
- Wie für kontinuierliche Zufallsvariablen gilt für kontinuierliche Zufallsvektoren

$$\mathbb{P}_X(X = x) = \mathbb{P}_X(x \leq X \leq x) = \int_{x_1}^{x_1} \cdots \int_{x_m}^{x_m} p_X(s_1, ..., s_m)\, ds_1 \cdots ds_m = 0 \tag{10}$$

Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Definition (Univariate Marginalverteilung)

$(\Omega, \mathcal{A}, \mathbb{P})$ sei ein Wahrscheinlichkeitsraum, $(\mathcal{X}, \mathcal{S})$ sei ein m-dimensionaler Messraum, $X : \Omega \to \mathcal{X}$ sei ein Zufallsvektor, $\mathbb{P}_X$ sei die Verteilung von X, $\mathcal{X}_i \subset \mathcal{X}$ sei der Ergebnisraum der $i$ten Komponente X_i von X, und $\mathcal{S}_i$ sei eine σ-Algebra auf X_i. Dann heißt die durch

$$\mathbb{P}_{X_i} : \mathcal{S}_i \to [0,1], S \mapsto \mathbb{P}_X\left(\mathcal{X}_1 \times \cdots \times \mathcal{X}_{i-1} \times S \times \mathcal{X}_{i+1} \times \cdots \times \mathcal{X}_m\right) \text{ für } S \in \mathcal{S}_i \quad (11)$$

definierte Verteilung die *$i$te univariate Marginalverteilung von X*.

Bemerkungen

- Univariate Marginalverteilungen sind die Verteilungen der Komponenten eines Zufallsvektors.
- Univariate Marginalverteilungen sind Verteilungen von Zufallsvariablen.
- Die Festlegung der multivariaten Verteilung von X legt auch die Verteilungen der X_i fest.

Theorem (Marginale Wahrscheinlichkeitsmasse- und dichtefunktionen)

(1) $X = (X_1, ..., X_m)$ sei ein m-dimensionaler diskreter Zufallsvektor mit Wahrscheinlichkeitsmassefunktion p_X und Komponentenergebnisräumen $\mathcal{X}_1, ..., \mathcal{X}_m$. Dann ergibt sich die Wahrscheinlichkeitsmassefunktion der $i$ten Komponente X_i von X als

$$p_{X_i} : \mathcal{X}_i \to [0,1], x_i \mapsto p_{X_i}(x_i) := \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_m} p_X(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_m) \quad (12)$$

(2) $X = (X_1, ..., X_m)$ sei ein m-dimensionaler kontinuierlicher Zufallsvektor mit Wahrscheinlichkeitsdichtefunktion p_X und Komponentenergebnisraum $\mathbb{R}$. Dann ergibt sich die Wahrscheinlichkeitsdichtefunktion der $i$ten Komponente X_i von X als

$$p_{X_i} : \mathbb{R} \to \mathbb{R}_{\geq 0}, x_i \mapsto p_{X_i}(x_i) := \int_{x_1} \cdots \int_{x_{i-1}} \int_{x_{i+1}} \cdots \int_{x_m} p_X(x_1, ..., x_{i-1}, x_i, x_{i+1}, ..., x_m)\, dx_1 ... dx_{i-1}\, dx_{i+1} ... dx_m. \quad (13)$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Die WMFen der univariaten Marginalverteilungen diskreter Zufallsvektoren ergeben sich durch Summation.
- Die WDFen der univariaten Marginalverteilungen kontinuierlicher Zufallsvektoren ergeben sich durch Integration.

Beispiel (Marginale Wahrscheinlichkeitsmassefunktionen)

Wir betrachten erneut den zweidimensionalen Zufallsvektor $X := (X_1, X_2)$ der Werte in $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2$ annimmt, wobei $\mathcal{X}_1 := \{1, 2, 3\}$ und $\mathcal{X}_2 = \{1, 2, 3, 4\}$ seien.

Basierend auf der oben definierten WMF ergeben sich folgende marginalen WMFen p_{X_1} und p_{X_2}

$p_X(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$p_{X_1}(x_1)$
$x_1 = 1$	0.1	0.0	0.2	0.1	0.4
$x_1 = 2$	0.1	0.2	0.0	0.0	0.3
$x_1 = 3$	0.0	0.1	0.1	0.1	0.3
$p_{X_2}(x_2)$	0.2	0.3	0.3	0.2	

Man beachte, dass $\sum_{x_1=1}^{3} p_{X_1}(x_1) = 1$ und $\sum_{x_2=1}^{3} p_{X_2}(x_2) = 1$ gilt.

Vorbemerkungen

Wir erinnern uns, dass für einen Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, \mathbb{P})$ und zwei Ereignisse $A, B \in \mathcal{A}$ mit $\mathbb{P}(B) > 0$ die bedingte Wahrscheinlichkeit von A gegeben B definiert ist als

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}. \tag{14}$$

Analog wird für zwei Zufallsvariablen X_1, X_2 mit Ereignisräumen $\mathcal{X}_1, \mathcal{X}_2$ und (messbaren) Mengen $S_1 \in \mathcal{X}_1, S_2 \in \mathcal{X}_2$ die bedingte Verteilung von X_1 gegeben X_2 mithilfe der Ereignisse $A := \{X_1 \in S_1\}$ und $B := \{X_2 \in S_2\}$ definiert.

So ergibt sich zum Beispiel die bedingte Wahrscheinlichkeit, dass $X_1 \in S_1$ gegeben dass $X_2 \in S_2$ unter der Annahme, dass $\mathbb{P}(\{X_2 \in S_2\}) > 0$, zu

$$\mathbb{P}(\{X_1 \in S_1\}|\{X_2 \in S_2\}) = \frac{\mathbb{P}(\{X_1 \in S_1\} \cap \{X_2 \in S_2\})}{\mathbb{P}(\{X_2 \in S_2\})}. \tag{15}$$

In der Folge betrachten wir zunächst durch die WMFen/WDFen zweidimensionaler Zufallsvektoren definierte bedingte Verteilungen.

Definition (Bedingte WMF, diskrete bedingte Verteilung)

$X := (X_1, X_2)$ sei ein diskreter Zufallsvektor mit Ergebnisraum $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2$, WMF $p_X = p_{X_1,X_2}$ und marginalen WMFen p_{X_1} und p_{X_2}. Die bedingte WMF von X_1 gegeben $X_2 = x_2$ ist dann für $p_{X_2}(x_2) > 0$ definiert als

$$p_{X_1|X_2=x_2} : \mathcal{X}_1 \to [0,1], x_1 \mapsto p_{X_1|X_2=x_2}(x_1|x_2) := \frac{p_{X_1,X_2}(x_1,x_2)}{p_{X_2}(x_2)} \tag{16}$$

Analog ist für $p_{X_1}(x_1) > 0$ die bedingte WMF von X_2 gegeben $X_1 = x_1$ definiert als

$$p_{X_2|X_1=x_1} : \mathcal{X}_2 \to [0,1], x_2 \mapsto p_{X_2|X_1=x_2}(x_1|x_2) := \frac{p_{X_1,X_2}(x_1,x_2)}{p_{X_1}(x_1)} \tag{17}$$

Die bedingten Verteilungen mit WMFen $p_{X_1|X_2=x_2}$ und $p_{X_2|X_1=x_1}$ heißen dann die *diskreten bedingten Verteilungen von X_1 gegeben $X_2 = x_2$ und X_2 gegeben $X_1 = x_1$*, respektive.

Bemerkungen

- In Analogie zur Definition der bedingten Wahrscheinlichkeit von Ereignissen gilt also

$$p_{X_1|X_2}(x_1|x_2) = \frac{p_{X_1,X_2}(x_1,x_2)}{p_{X_2}(x_2)} = \frac{\mathbb{P}(\{X_1 = x_1\} \cap \{X_2 = x_2\})}{\mathbb{P}(\{X_2 = x_2\})}. \tag{18}$$

- Bedingte Verteilungen sind (lediglich) normalisierte gemeinsame Verteilungen.

Beispiel (Bedingte Wahrscheinlichkeitsmassefunktionen)

Wir betrachten erneut den zweidimensionalen Zufallsvektor $X := (X_1, X_2)$ der Werte in $\mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2$ annimmt, wobei $\mathcal{X}_1 := \{1, 2, 3\}$ und $\mathcal{X}_2 = \{1, 2, 3, 4\}$ seien.

Basierend auf der oben definierten WMF und den entsprechenden oben evaluierten marginalen WMFen ergeben sich folgende bedingte WMFen für $p_{X_2|X_1=x_1}$

$p_{X_1\|X_2}(x_1, x_2)$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$
$p_{X_2\|X_1=1}(x_2\|x_1 = 1)$	$\frac{0.1}{0.4} = 0.25$	$\frac{0.0}{0.4} = 0.00$	$\frac{0.2}{0.4} = 0.50$	$\frac{0.1}{0.4} = 0.25$
$p_{X_2\|X_1=2}(x_2\|x_1 = 2)$	$\frac{0.1}{0.3} = 0.3\bar{3}$	$\frac{0.2}{0.3} = 0.6\bar{6}$	$\frac{0.0}{0.3} = 0.00$	$\frac{0.0}{0.3} = 0.00$
$p_{X_2\|X_1=3}(x_2\|x_1 = 3)$	$\frac{0.0}{0.3} = 0.00$	$\frac{0.1}{0.3} = 0.3\bar{3}$	$\frac{0.1}{0.3} = 0.3\bar{3}$	$\frac{0.1}{0.3} = 0.3\bar{3}$

Bemerkungen

- Man beachte, dass $\sum_{x_2=1}^{4} p_{X_2|X_1=x_1}(x_2|x_1) = 1$ für alle $x_1 \in \mathcal{X}_1$.
- Man beachte die qualitative Ähnlichkeit der WMFen $p_{X_1,X_2}(x_1, x_2)$ und $p_{X_2|X_1}(x_2|x_1)$.
- Bedingte Verteilungen sind (lediglich) normalisierte gemeinsame Verteilungen.

Definition (Bedingte WDF, kontinuierliche bedingte Verteilungen)

$X := (X_1, X_2)$ sei ein kontinuierlicher Zufallsvektor mit Ergebnisraum $\mathbb{R}^2$, WDF $p_X = p_{X_1,X_2}$ und marginalen WDFen p_{X_1} und p_{X_2}. Die bedingte WDF von X_1 gegeben $X_2 = x_2$ ist dann für $p_{X_2}(x_2) > 0$ definiert als

$$p_{X_1|X_2=x_2} : \mathbb{R} \to \mathbb{R}_{\geq 0}, x_1 \mapsto p_{X_1|X_2=x_2}(x_1|x_2) := \frac{p_{X_1,X_2}(x_1, x_2)}{p_{X_2}(x_2)} \tag{19}$$

Analog ist für $p_{X_1}(x_1) > 0$ die bedingte WMF von X_2 gegeben $X_1 = x_1$ definiert als

$$p_{X_2|X_1=x_1} : \mathbb{R} \to \mathbb{R}_{\geq 0}, x_2 \mapsto p_{X_2|X_1=x_2}(x_2|x_1) := \frac{p_{X_1,X_2}(x_1, x_2)}{p_{X_1}(x_1)} \tag{20}$$

Die Verteilungen mit WDFen $p_{X_1|X_2=x_2}$ und $p_{X_2|X_1=x_1}$ heißen dann die *kontinuierlichen bedingten Verteilungen von X_1 gegeben $X_2 = x_2$ und X_2 gegeben $X_1 = x_1$*, respektive.

Bemerkung

- Im kontinuierlichen Fall gilt zwar $\mathbb{P}(X = x) = 0$, aber nicht notwendig auch $p_X(x) = 0$.

Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Definition (Erwartungswert)

X sei ein m-dimensionaler Zufallvektor. Dann ist der *Erwartungwert* von X definiert als der m-dimensionale Vektor

$$\mathbb{E}(X) := \begin{pmatrix} \mathbb{E}(X_1) \\ \vdots \\ \mathbb{E}(X_m) \end{pmatrix}. \tag{21}$$

Bemerkungen

- Der Erwartungswert von X ist der Vektor der Erwartungswerte $\mathbb{E}(X_1), ..., \mathbb{E}(X_m)$.
- Im allgemeinen linearen Modell $X = D\beta + \varepsilon$ gilt zum Beispiel $\mathbb{E}(\varepsilon) = 0_m$ und $\mathbb{E}(X) = D\beta$.

Definition (Kovarianzmatrix)

X sei ein m-dimensionaler Zufallsvektor. Die *Kovarianzmatrix* von X ist definiert als die $m \times m$ matrix

$$\mathbb{C}(X) := \mathbb{E}\left((X - \mathbb{E}(X))(X - \mathbb{E}(X))^T\right) = \begin{pmatrix} \mathbb{C}(X_1, X_1) & \cdots & \mathbb{C}(X_1, X_m) \\ \vdots & \ddots & \vdots \\ \mathbb{C}(X_m, X_1) & \cdots & \mathbb{C}(X_m, X_m) \end{pmatrix}. \quad (22)$$

Bemerkungen

- Die Kovarianzmatrix $\mathbb{C}(X)$ ist die Matrix der Kovarianzen $\mathbb{C}(X_i, X_j), i, j = 1, \ldots, m$.
- Die Korrelation von X_i and X_j ist definiert als

$$\rho_{ij} = \frac{\mathbb{C}(X_i, X_j)}{\sqrt{\mathbb{C}(X_i, X_i)}\sqrt{\mathbb{C}(X_j, X_j)}} \in [-1, 1]. \quad (23)$$

- Die Kovarianzmatrix repräsentiert also die Korrelationsstruktur der Zufallsvariablen $X_1, \ldots, X_m$
- Im ALM mit sphärischer Kovarianmatrix gilt per Definition $\mathbb{C}(\varepsilon) = \mathbb{C}(X) = \sigma^2 I_m$.

Erwartungswerte und Kovarianzmatrizen

Die Äquivalenz der Kovarianzschreibweisen folgt mit

$$
\begin{aligned}
&\mathbb{E}\left((X-\mathbb{E}(X))(X-\mathbb{E}(X))^T\right)\\
&=\mathbb{E}\left(\left(\begin{pmatrix}X_1\\ \vdots\\ X_m\end{pmatrix}-\begin{pmatrix}\mathbb{E}(X_1)\\ \vdots\\ \mathbb{E}(X_m)\end{pmatrix}\right)\left(\begin{pmatrix}X_1\\ \vdots\\ X_m\end{pmatrix}-\begin{pmatrix}\mathbb{E}(X_1)\\ \vdots\\ \mathbb{E}(X_m)\end{pmatrix}\right)^T\right)\\
&=\mathbb{E}\left(\begin{pmatrix}X_1-\mathbb{E}(X_1)\\ \vdots\\ X_m-\mathbb{E}(X_m)\end{pmatrix}\begin{pmatrix}X_1-\mathbb{E}(X_1)\\ \vdots\\ X_m-\mathbb{E}(X_m)\end{pmatrix}^T\right)\\
&=\mathbb{E}\left(\begin{pmatrix}X_1-\mathbb{E}(X_1)\\ \vdots\\ X_m-\mathbb{E}(X_m)\end{pmatrix}\begin{pmatrix}X_1-\mathbb{E}(X_1) & \dots & X_m-\mathbb{E}(X_m)\end{pmatrix}\right)\\
&=\mathbb{E}\begin{pmatrix}(X_1-\mathbb{E}(X_1))(X_1-\mathbb{E}(X_1)) & \dots & (X_1-\mathbb{E}(X_1))(X_m-\mathbb{E}(X_m)\\ \vdots & \ddots & \vdots\\ (X_m-\mathbb{E}(X_m))(X_1-\mathbb{E}(X_1)) & \dots & (X_m-\mathbb{E}(X_m))(X_m-\mathbb{E}(X_m))\end{pmatrix}\\
&=\left(\mathbb{E}\left((X_i-\mathbb{E}(X_i))(X_j-\mathbb{E}(X_j))\right)\right)_{1\le i,j\le m}\\
&=:\left(\mathbb{C}(X_i,X_j)\right)_{1\le i,j\le m}\\
&=:\mathbb{C}(X).
\end{aligned}
$$

Definition (Stichprobenmittel, Stichprobenkovarianmatrix)

$X^{(1)}, ..., X^{(n)}$ seien n m-dimensionale Zufallsvektoren. Dann ist das Stichprobenmittel der $X^{(1)}, ..., X^{(n)}$ definiert als

$$\bar{X} := \frac{1}{n} \sum_{i=1}^{n} X^{(i)} \tag{24}$$

und die Stichprobenkovarianzmatrix der $X^{(1)}, ..., X^{(m)}$ ist definiert als

$$C := \frac{1}{n-1} \sum_{i=1}^{n} (X^{(i)} - \bar{X})(X^{(i)} - \bar{X})^T. \tag{25}$$

Bemerkungen

- $\bar{X}$ ist ein unverzerrter Schätzer von $\mathbb{E}(X)$.
- C ist ein unverzerrter Schätzer von $\mathbb{C}(X)$.

Theorem (Datenmatrix und Stichprobenkovarianzmatrix)

$$Y := \begin{pmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(n)} \end{pmatrix} \in \mathbb{R}^{m \times n} \tag{26}$$

sei eine $m \times n$ *Datenmatrix*, die durch die spaltenweise Konkatenation von n m-dimensionalen Zufallvektoren $X^{(1)}, \ldots, X^{(n)}$ gegeben sei. Dann kann das Stichprobenmittel berechnet werden als

$$\bar{Y} := \frac{1}{n} Y 1_n = \bar{X} \tag{27}$$

Weiterhin sei

$$Y_c := Y - \bar{Y} 1_n^T \in \mathbb{R}^{m \times n} \tag{28}$$

die $m \times n$ *zentrierte Datenmatrix*. Dann kann die Stichprobenkovarianzmatrix der $X^{(1)}, \ldots, X^{(n)}$ durch

$$C = \frac{1}{n-1} Y_c Y_c^T \text{ und } C = \frac{1}{n-1} \left(Y \left(I_n - \frac{1}{n} 1_{nn} \right) Y^T \right) \tag{29}$$

berechnet werden.

Bemerkungen

- $C = \frac{1}{n-1} Y_c Y_c^T$ ist in der Theorie der Hauptkomponentenanalyse von Bedeutung.
- $C = \frac{1}{n-1} \left(Y \left(I_n - \frac{1}{n} 1_{nn} \right) Y^T \right)$ erleichtert die numerische Implementation.

Beweis

Die Darstellung des Stichprobenmittels ergibt sich direkt aus

$$
\begin{aligned}
\bar{Y} &:= \frac{1}{n} Y 1_n \\
&= \frac{1}{n} \left(\begin{pmatrix} X_1^{(1)} & \cdots & X_1^{(n)} \\ \vdots & \ddots & \vdots \\ X_m^{(1)} & \cdots & X_m^{(n)} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right) \\
&= \frac{1}{n} \begin{pmatrix} \sum_{i=1}^{n} X_1^{(i)} \\ \vdots \\ \sum_{i=1}^{n} X_m^{(i)} \end{pmatrix} \\
&= \frac{1}{n} \sum_{i=1}^{n} X^{(i)} =: \bar{X}
\end{aligned}
\tag{30}
$$

Beweis (fortgeführt)

Wir halten weiterhin zunächt fest, dass gilt

$$
\begin{aligned}
Y_c &= Y - \bar{X} 1_n^T \\
&= \begin{pmatrix} X_1^{(1)} & \dots & X_1^{(n)} \\ \vdots & \ddots & \vdots \\ X_m^{(1)} & \dots & X_m^{(n)} \end{pmatrix} - \begin{pmatrix} \bar{X}_1 \\ \vdots \\ \bar{X}_m \end{pmatrix} \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \\
&= \begin{pmatrix} X_1^{(1)} & \dots & X_1^{(n)} \\ \vdots & \ddots & \vdots \\ X_m^{(1)} & \dots & X_m^{(n)} \end{pmatrix} - \begin{pmatrix} \bar{X}_1 & \dots & \bar{X}_1 \\ \vdots & \ddots & \vdots \\ \bar{X}_m & \dots & \bar{X}_m \end{pmatrix} \\
&= \begin{pmatrix} X_1^{(1)} - \bar{X}_1 & \dots & X_1^{(n)} - \bar{X}_1 \\ \vdots & \ddots & \vdots \\ X_m^{(1)} - \bar{X}_m & \dots & X_m^{(n)} - \bar{X}_m \end{pmatrix} \\
&= \begin{pmatrix} X^{(1)} - \bar{X} & \dots & X^{(n)} - \bar{X} \end{pmatrix}
\end{aligned}
$$

Beweis (fortgeführt)

Dann aber gilt

$$\begin{aligned}\frac{1}{n-1} Y_c Y_c^T &= \frac{1}{n-1} \begin{pmatrix} X^{(1)} - \bar{X} & \dots & X^{(n)} - \bar{X} \end{pmatrix} \begin{pmatrix} \left(X^{(1)} - \bar{X}\right)^T \\ \vdots \\ \left(X^{(n)} - \bar{X}\right)^T \end{pmatrix} \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \left(X^{(i)} - \bar{X}\right) \left(X^{(i)} - \bar{X}\right)^T \\ &= C.\end{aligned}$$

□

Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Definition (Normalverteilte Zufallsvariable)

X sei eine Zufallsvariable mit Ergebnisraum $\mathbb{R}$ und WDF

$$p : \mathbb{R} \to \mathbb{R}_{>0}, x \mapsto p(x) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{31}$$

Dann sagen wir, dass X einer *Normalverteilung (oder Gauß-Verteilung) mit Parametern* $\mu \in \mathbb{R}$ *und* $\sigma^2 > 0$ unterliegt und nennen X eine *normalverteilte Zufallsvariable*. Wir kürzen dies mit $X \sim N\left(\mu, \sigma^2\right)$ ab. Die WDF einer normalverteilten Zufallsvariable bezeichnen wir mit

$$N\left(x; \mu, \sigma^2\right) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right). \tag{32}$$

Bemerkungen

- Der Parameter μ entspricht dem Wert höchster Wahrscheinlichkeitsdichte.
- Der Parameter σ^2 spezifiziert die Breite der WDF.

Definition (Multivariate Normalverteilung)

X sei ein n-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^n$ und WDF

$$p : \mathbb{R}^n \to \mathbb{R}_{>0},\ x \mapsto p(x) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{33}$$

Dann sagen wird, dass X einer *multivariaten (oder n-dimensionalen) Normalverteilung* mit *Erwartungswertparameter* $\mu \in \mathbb{R}^n$ und positive-definitem *Kovarianzmatrixparameter* $\Sigma \in \mathbb{R}^{n \times n}$ unterliegt und nennen X einen *(multivariat) normalverteilten Zufallsvektor*. Wir kürzen dies mit $X \sim N(\mu, \Sigma)$ ab. Die WDF eines multivariat normalverteilten Zufallsvektors bezeichnen wir mit

$$N(x; \mu, \Sigma) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{34}$$

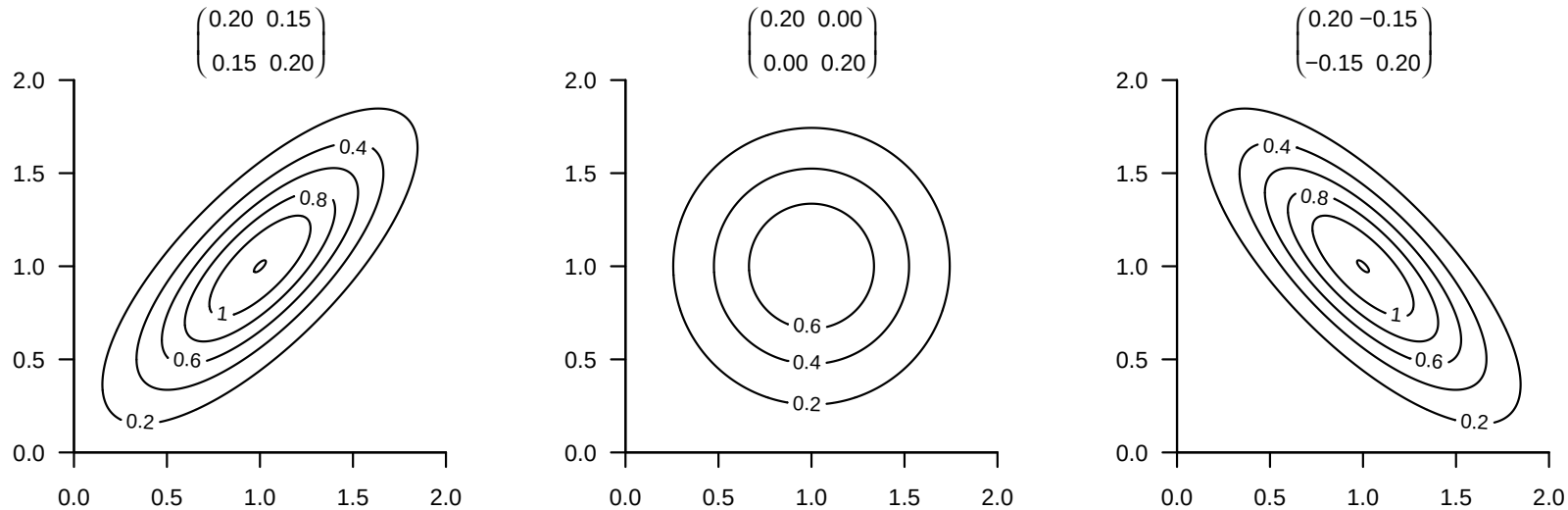
Bemerkungen

- Der Parameter $\mu \in \mathbb{R}^n$ entspricht dem Wert höchster Wahrscheinlichkeitsdichte
- Die Diagonalelemente von Σ spezifizieren die Breite der WDF bezüglich $X_1, ..., X_n$.
- Das i, jte Element von Σ spezifiziert die Kovarianz on X_i und X_j.
- Der Term $(2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}}$ ist die Normalsieriungskonstante für den Exponentialfunktionsterm.

Visualisierung bivariater Normalverteilungsdichtefunktionen

```
# multivariate Normalverteilungstools
# install.packages("mvtnorm")
library(mvtnorm)


# Ergebnisraumdefintion
x_min  = 0                                          # x_i Minimum
x_max  = 2                                          # x_i Maxim
x_res  = 1e3                                        # x_i Auflösung
x_1    = seq(x_min, x_max, length.out = x_res)      # x_1 Raum
x_2    = seq(x_min, x_max, length.out = x_res)      # x_2 Raum
X      = expand.grid(x_1,x_2)                       # X = (x_1,x_2)^T Raum

# Parameterdefinition
mu     = c(1,1)                                     # \mu \in \mathbb{R}^2
S      = list(matrix(c(0.2,  0.15,  0.15, 0.2), 2), # \Sigma in \mathbb{R}^{2 \times 2}
              matrix(c(0.2,  0.00,  0.00, 0.2), 2), # \Sigma in \mathbb{R}^{2 \times 2}
              matrix(c(0.2, -0.15, -0.15, 0.2), 2)) # \Sigma in \mathbb{R}^{2 \times 2}

# Kovarianzparametervariantenschleife
for (Sigma in S){

  # Wahrscheinlichkeitsdichtefunktionauswertung
  p      = matrix(                                  # Matrixkonversion des von
                  dmvnorm(as.matrix(X), mu, Sigma), # dmvnorm() ausgegebenen Vektors
                  nrow = x_res)

  # Visualisierung
  contour(
  x_1,
  x_2,
  p,
  xlim      =  c(x_min,x_max),
  ylim      =  c(x_min,x_max),
  nlevels   = 5)
}
```

Visualisierung bivariater Normalverteilungsdichtefunktionen

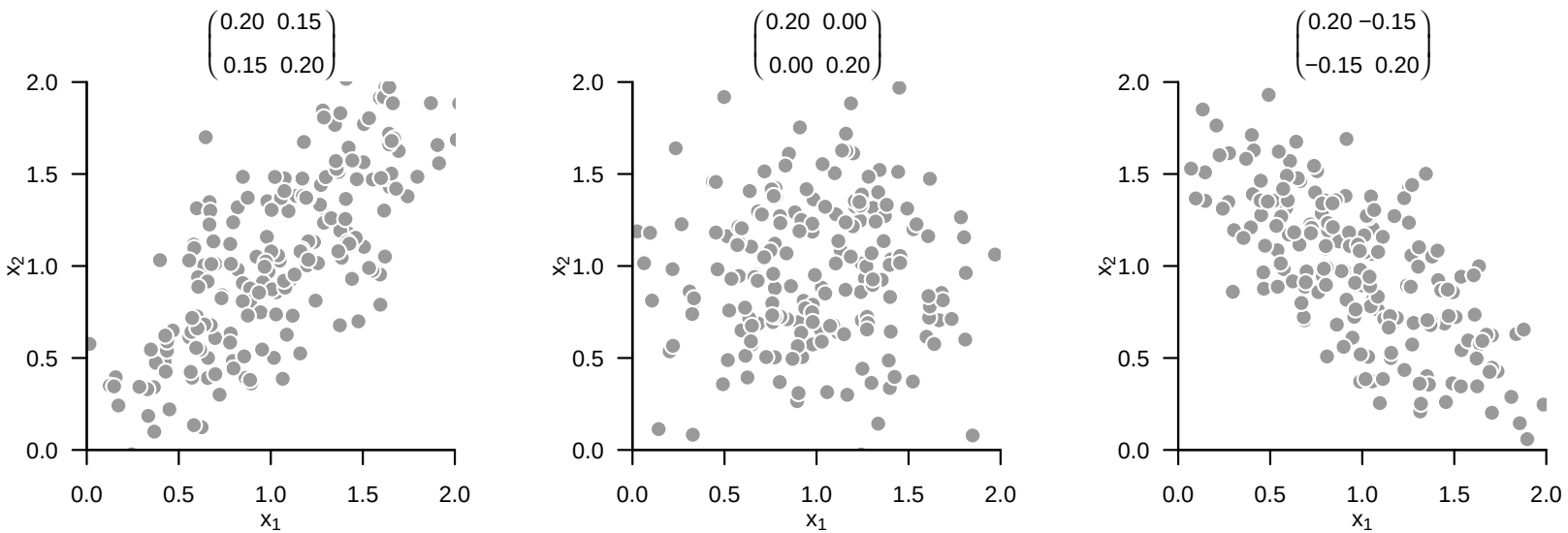


Realisierung bivariater normalverteilter Zufallsvektoren

```r
# R Paket für multivariate Normalverteilungen
library(mvtnorm)

# Parameterdefinition
mu     = c(1,1)                                  # \mu \in \mathbb{R}^2
Sigma  = matrix(c(0.2,  0.15,  0.15, 0.2), 2)   # \Sigma in \mathbb{R}^{2 \times 2}

# Zufallsvektorrealisierungen
rmvnorm(n = 10, mu, Sigma)
```

```
>         [,1]  [,2]
>  [1,] 1.316 0.927
>  [2,] 1.409 1.105
>  [3,] 0.340 0.386
>  [4,] 1.458 1.250
>  [5,] 0.886 0.397
>  [6,] 0.453 0.845
>  [7,] 1.589 1.343
>  [8,] 1.634 1.456
>  [9,] 1.227 1.067
> [10,] 1.054 0.995
```

Realisierung bivariater normalverteilter Zufallsvektoren



Theorem (Marginale Normalverteilungen)

Es sei $m := k + l$ und $X = (X_1, ..., X_m)$ sei ein m-dimensionaler normalverteilter Zufallsvektor mit Erwartungswertparameter

$$\mu = \begin{pmatrix} \mu_y \\ \mu_z \end{pmatrix} \in \mathbb{R}^m, \tag{35}$$

mit $\mu_y \in \mathbb{R}^k$ and $\mu_z \in \mathbb{R}^l$ und Kovarianzmatrixparameter

$$\Sigma = \begin{pmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_{zz} \end{pmatrix} \in \mathbb{R}^{m \times m}, \tag{36}$$

mit $\Sigma_{yy} \in \mathbb{R}^{k \times k}$, $\Sigma_{yz} \in \mathbb{R}^{k \times l}$, $\Sigma_{zy} \in \mathbb{R}^{l \times k}$, und $\Sigma_{zz} \in \mathbb{R}^{l \times l}$. Dann sind $Y := (X_1, ..., X_k)$ und $Z := (X_{k+1}, ..., X_m)$ k- und l-dimensionale normalverteilte Zufallsvektoren, respektive, und es gilt

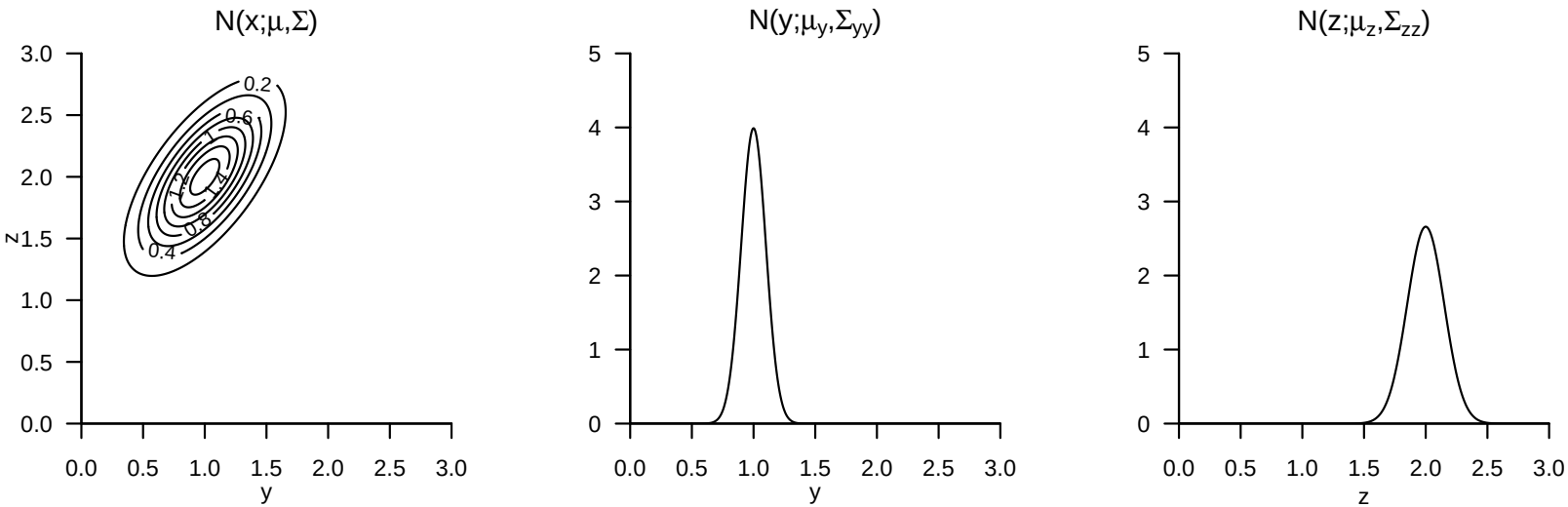
$$Y \sim N(\mu_y, \Sigma_{yy}) \text{ and } Z \sim N(\mu_z, \Sigma_{xx}), \tag{37}$$

Bemerkungen

- Wir verzichten auf einen Beweis
- Die Marginalverteilungen einer multivariaten Normalverteilung sind auch Normalverteilungen.
- Die Parameter der Marginalverteilungen ergeben sich aus den Parametern der gemeinsamen Verteilung.

Marginale Normalverteilungen

$$m := 2, k = 1, l = 1, \mu := \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \Sigma := \begin{pmatrix} 0.10 & 0.08 \\ 0.08 & 0.15 \end{pmatrix}$$



Theorem (Gemeinsame Normalverteilunge)

X sei ein m-dimensionaler normalverteilter Zufallsvektor mit WDF

$$p_X : \mathbb{R}^m \to \mathbb{R}_{>0},\ x \mapsto p_X(x) := N(x; \mu_x, \Sigma_{xx}) \text{ mit } \mu_x \in \mathbb{R}^m, \Sigma_{xx} \in \mathbb{R}^{m \times m}, \tag{38}$$

$A \in \mathbb{R}^{n \times m}$ sei eine Matrix, $b \in \mathbb{R}^n$ sei ein Vektor und Y sei ein n-dimensionaler bedingt normalverteilter Zufallsvektor mit bedingter WDF

$$p_{y|X}(\cdot|x) : \mathbb{R}^n \to \mathbb{R}_{>0},\ y \mapsto p_{Y|X}(y|x) := N(y; AX + b, \Sigma_{yy}) \text{ mit } \Sigma_{yy} \in \mathbb{R}^{n \times n}. \tag{39}$$

Dann ist der $m + n$-dimensionale Zufallsvektor (X, Y) normalverteilt mit (gemeinsamer) WDF

$$p_{X,Y} : \mathbb{R}^{m+n} \to \mathbb{R}_{>0},\ (x, y) \mapsto p_{X,Y}(x, y) = N\left((x, y); \mu_{x,y}, \Sigma_{x,y}\right), \tag{40}$$

mit $\mu_{x,y} \in \mathbb{R}^{n+m}$ and $\Sigma_{x,y} \in \mathbb{R}^{m+n \times m+n}$ und insbesondere

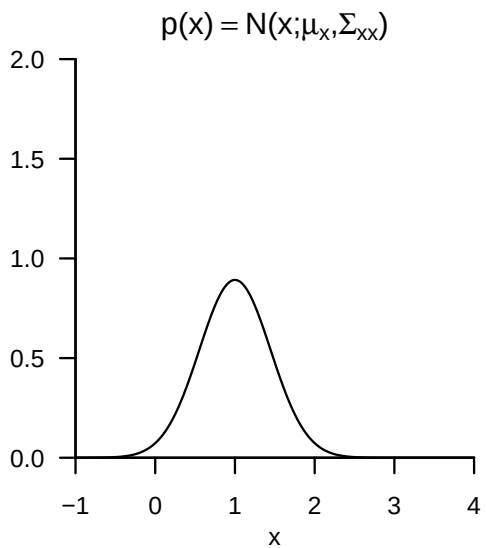
$$\mu_{x,y} = \begin{pmatrix} \mu_x \\ A\mu_x + b \end{pmatrix} \text{ und } \Sigma_{x,y} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xx}A^T \\ A\Sigma_{xx} & \Sigma_{yy} + A\Sigma_{xx}A^T \end{pmatrix}. \tag{41}$$

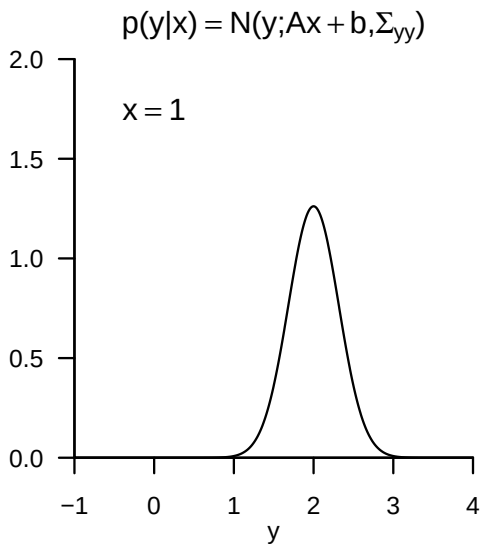
Bemerkungen

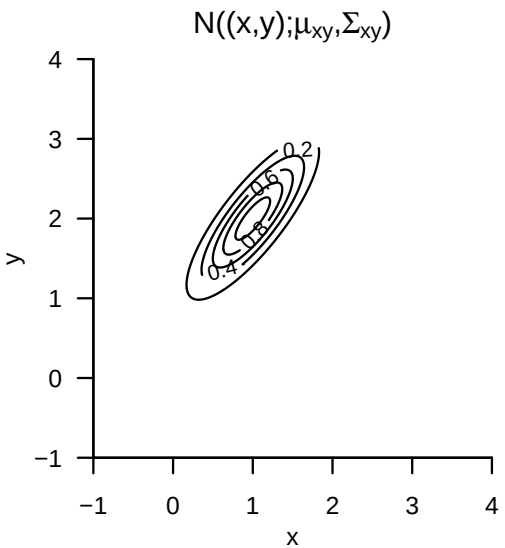
- Wir verzichten auf einen Beweis.
- Eine marginale und eine bedingte multivariate Normalverteilung induzieren eine gemeinsame Normalverteilung.
- Die Parameter der gemeinsamen Verteilungen ergeben sich als linear-affine Transformation der Parameter der induzierenden Verteilungen.

Gemeinsame Normalverteilungen

$$m := 1, n := 1, \mu_x := 1, \Sigma_{xx} := 0.2, A := 1, b := 1, \Sigma_{yy} := 0.1$$







Theorem (Bedingte Normalverteilungen)

(X, Y) sei ein $m + n$-dimensionaler normalverteilter Zufallsvektor mit WDF

$$p_{X,Y} : \mathbb{R}^{m+n} \to \mathbb{R}_{>0}, (x, y) \mapsto p_{X,Y}(x, y) := N\left((x, y); \mu_{x,y}, \Sigma_{x,y}\right), \tag{42}$$

mit

$$\mu_{x,y} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \Sigma_{x,y} = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}, \tag{43}$$

mit $x, \mu_x \in \mathbb{R}^m, y, \mu_y \in \mathbb{R}^n$ and $\Sigma_{xx} \in \mathbb{R}^{m \times m}, \Sigma_{xy} \in \mathbb{R}^{m \times n}, \Sigma_{yy} \in \mathbb{R}^{n \times n}$. Dann ist die bedingte Verteilung von X gegeben Y eine m-dimensionale Normalverteilung mit bedingter WDF

$$p_{X|Y}(\cdot|y) : \mathbb{R}^m \to \mathbb{R}_{>0}, x \mapsto p_{X|Y}(x|y) := N(x; \mu_{x|y}, \Sigma_{x|y}) \tag{44}$$

mit

$$\mu_{x|y} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(Y - \mu_y) \in \mathbb{R}^m \tag{45}$$

und

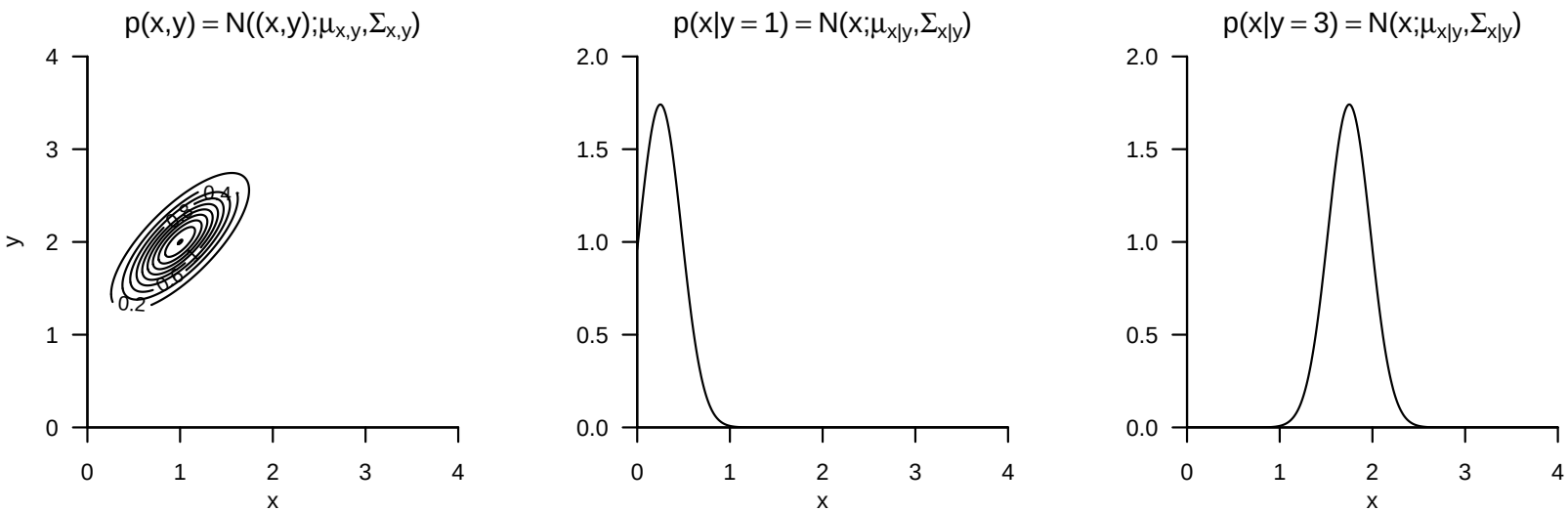
$$\Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \in \mathbb{R}^{m \times m}. \tag{46}$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Die Parameter einer bedingten (multivariaten) Normalverteilung ergeben sich aus den Parametern einer gemeinsamen multivariaten Normalverteilung. Im Zusammenspiel mit den vorherigen Theoremen können die Parameter bedingter und marginale Normalverteilungen aus den Parametern der komplementären bedingten und marginalen Normalverteilungen bestimmt werden.

Bedingte Normalverteilungen

$$m := 2, n := 1, \mu := \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \Sigma := \begin{pmatrix} 0.12 & 0.09 \\ 0.09 & 0.12 \end{pmatrix}$$



Zufallsvektoren und multivariate Verteilungen

Marginale und bedingte Verteilungen

Erwartungswerte und Kovarianzmatrizen

Multivariate Normalverteilungen

Selbstkontrollfragen

Selbstkontrollfragen

1. Definieren Sie den Begriff des Zufallsvektors.
2. Definieren Sie den Begriff der multivariaten Verteilung eines Zufallsvektors.
3. Definieren Sie den Begriff der multivariaten WMF.
4. Definieren Sie den Begriff der multivariaten WDF.
5. Definieren Sie den Begriff der univariaten Marginalverteilung eines Zufallsvektors.
6. Wie berechnet man die WMF der $i$ten Komponente eines diskreten Zufallsvektors?
7. Wie berechnet man die WDF der $i$ten Komponente eines kontinuierlichen Zufallsvektors?
8. Definieren Sie die Begriffe der bedingten WMF und der diskreten bedingten Verteilung.
9. Definieren Sie die Begriffe der bedingten WDF und der kontinuierlichen bedingten Verteilung.
10. Geben Sie die Definition des Erwartungswerts eines Zufallsvektors wieder.
11. Geben Sie die Definition der Kovarianzmatrix eines Zufallsvektors wieder.
12. Geben Sie die Definition des Stichprobenmittels und der Stichprobenkovarianzmatrix wieder.
13. Erläutern Sie, warum für eine Stichprobenkovarianzmatrix $C = \frac{1}{n-1} Y_c Y_c^T$ gilt.
14. Definieren Sie die WDF einer univariaten normalverteilten Zufallsvariable und erläutern Sie diese.
15. Definieren Sie die WDF eines multivariaten normalverteilten Zufallsvektors wieder und erläutern Sie diese.
16. Geben Sie das Theorem zu Marginalen Normalverteilungen wieder.
17. Geben Sie das Theorem zu Gemeinsamen Normalverteilungen wieder.
18. Geben Sie das Theorem zu Bedingten Normalverteilungen wieder.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(4) Hauptkomponentenanalyse

Modul A1/A3 Forschungsmethoden: Multivariate Verfahren

Datum	Einheit	Thema
15.10.2021	Einführung	(0) Einführung
15.10.2021	Grundlagen	(1) Vektoren
22.10.2021	Grundlagen	(2) Matrizen I
29.10.2021	Grundlagen	(3) Matrizen II
05.11.2021	Grundlagen	(4) Multivariate Normalverteilung
12.11.2021	**Achsentransformationen**	**(5) Hauptkomponentenanalyse**
19.11.2021	Achsentransformationen	(6) Faktoranalyse
26.11.2021	Maschinelles Lernen	(7) LDA und Optimierung
03.12.2021	Maschinelles Lernen	(8) Logistische Regression
10.12.2021	Maschinelles Lernen	(9) Support Vektor Maschinen
17.12.2021	Maschinelles Lernen	(10) Neuronale Netze
	Weihnachtspause	
07.01.2022	Frequentistische Inferenz	(11) T-Tests
14.01.2022	Frequentistische Inferenz	(12) Einfaktorielle Varianzanalyse
21.01.2022	Frequentistische Inferenz	(13) Multivariate Regression
28.01.2022	Frequentistische Inferenz	(14) Kanonische Korrelation
22.02.2022	Klausur	12 - 13 Uhr, G26-H1
Jul 2022	Klausurwiederholungstermin	

Überblick

- Hauptkomponentenanalyse heißt auf Englisch Principal Component Analysis (PCA).
- PCA ist eine Featureselektionsmethode.
  - "Features" sind die Komponenten multidimensionaler Zufallsvektoren.
  - Korrelierte Features repräsentieren redundate Information.
- PCA generiert ein korrelationsfreies Featureset durch lineare Featurekombination.
- PCA basiert auf
  - einer Eigenanalyse/Orthonormalzerlegung der Stichprobenkovarianzmatrix und
  - einer anschließenden Vektorkoordinatentransformation.
- Implementiert wird eine PCA oft mithilfe einer Singulärwertzerlegung.
- In der Psychologie dient PCA zum Beispiel
  - der Datenkompression beim Umgang mit neurophysiologischen Zeitseriendaten,
  - der Inspiration im Rahmen der "Exploratorischen Faktoranalyse".

Vektorkoordinatentransformation

Definition und Theorem

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Vektorkoordinatentransformation

Definition

Datenkompression

Singulärwertzerlegung

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Vektorkoordinatentransformation

Im Folgenden wichtige Begriffe aus (1) Vektoren

Euklidischer Vektorraum. Das Tupel $((\mathbb{R}^m, +, \cdot), \langle\rangle)$ aus dem reellen Vektorraum $(\mathbb{R}^m, +, \cdot)$ und dem Skalarprodukt $\langle\rangle$ auf $\mathbb{R}^m$ heißt *reeller kanonischer Euklidischer Vektorraum*.

Basis. V sei ein Vektorraum und es sei $B \subseteq V$. Dann heißt B eine *Basis von* V, wenn die Vektoren in B linear unabhängig sind und die Vektoren in B den Vektorraum V aufspannen.

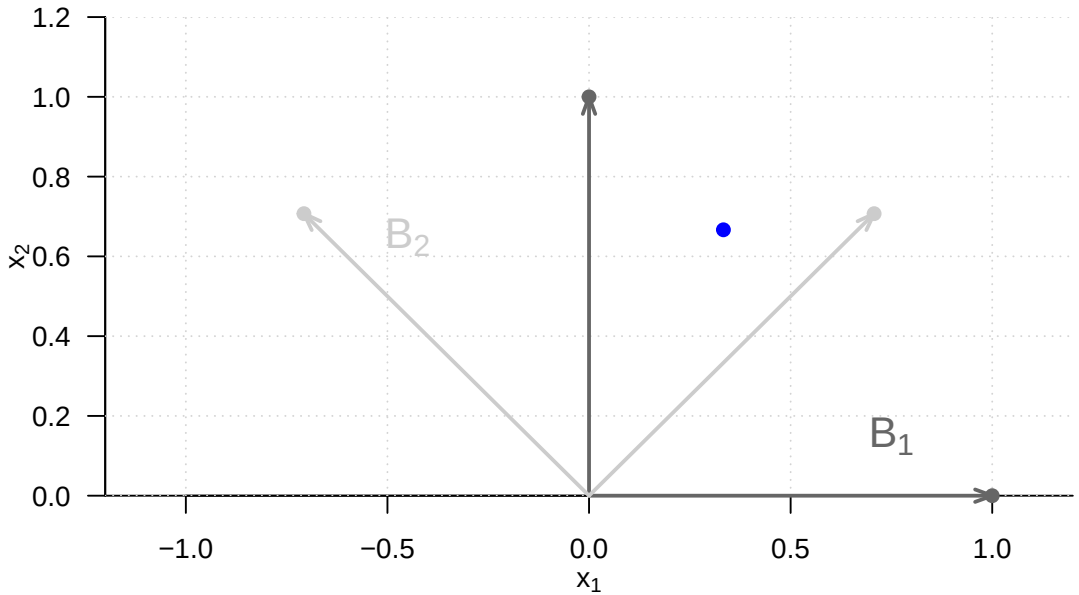
Basisdarstellung und Koordinaten. $B := \{b_1, ..., b_m\}$ sei eine Basis eines m-dimensionalen Vektorraumes V und es sei $x \in V$. Dann heißt die Linearkombination $x = \sum_{i=1}^{m} a_i b_i$ die *Darstellung von* x *bezüglich der Basis* B und die Koeffizienten $a_1, ..., a_m$ heißen die *Koordinaten von* x *bezüglich der Basis* B.

Orthonormalbasis von $\mathbb{R}^m$. Eine Menge von m Vektoren $q_1, ..., q_m \in \mathbb{R}^m$ heißt *Orthonormalbasis* von $\mathbb{R}^m$, wenn $q_1, ..., q_m$ jeweils die Länge 1 haben und wechselseitig orthogonal sind.

Im Folgenden wichtiger Begriff aus (2) Matrizen

Orthonormale Zerlegung einer symmetrischen Matrix. $S \in \mathbb{R}^{m \times m}$ sei eine symmetrische Matrix. Dann kannn S geschrieben werden als $S = Q\Lambda Q^T$, wobei $Q \in \mathbb{R}^{m \times m}$ eine orthogonale Matrix ist und $\Lambda \in \mathbb{R}^{m \times m}$ eine Diagonalmatrix ist. Dabei sind die Spalten von Q die Eigenvektoren von S und die Diagonalelemente von Λ sind die entsprechenden Eigenwerte.

Im Folgenden wichtige Intuition aus (1) Vektoren



- Im Rahmen von Hauptkomponentenranalyse werden wir daran interessiert sein, basierend auf den Koordinaten eines Vektors bezüglich einer Basis die Koordinaten desselben Vektors bezüglich einer anderen Basis zu berechnen.

Definition (Orthogonalprojektion)

x und q seien Vektoren im Euklidischen Vektorraum $\mathbb{R}^m$. Dann ist die *Orthogonalprojektion von x auf q* definiert als der Vektor

$$\tilde{x} = aq \text{ mit } a := \frac{q^T x}{q^T q}, \tag{1}$$

wobei der Skalar a *Projektionsfaktor* genannt wird.

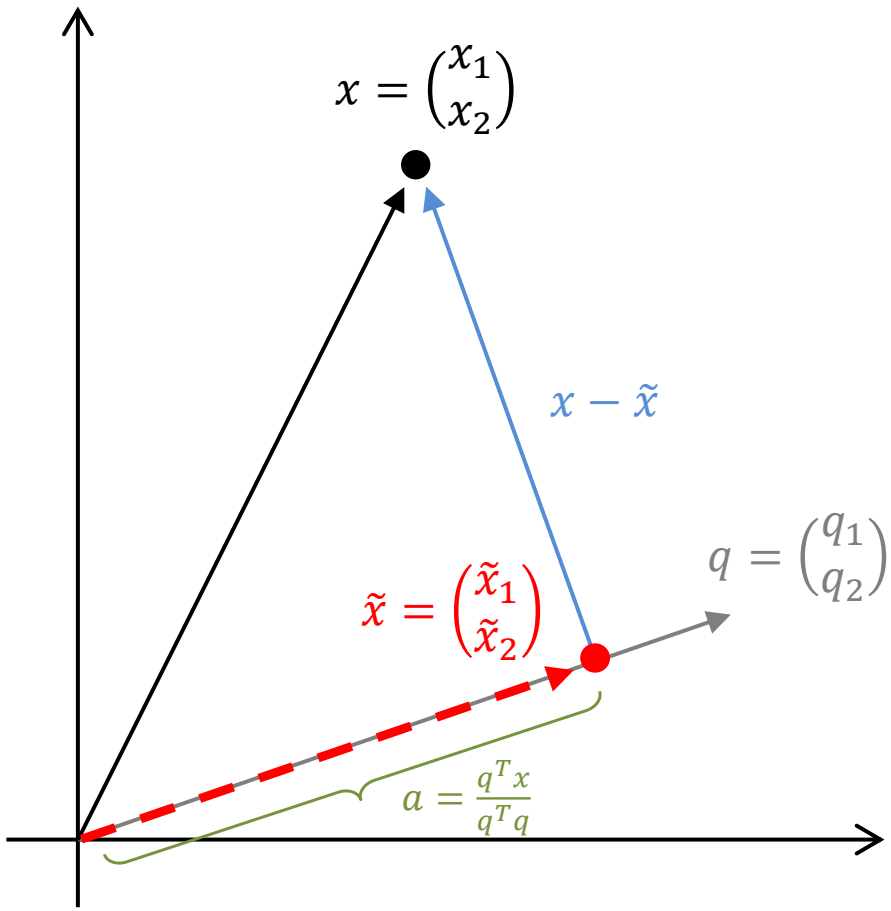
Bemerkungen

- Per definition ist $\tilde{x} = aq$ mit $a \in \mathbb{R}$ der Punkt in Richtung von q der x am nähesten ist.
- Diese minimierte Distanzeigenschaft impliziert die Orthogonalität von q und $x - \tilde{x}$.
- Die Formel von a folgt direkt aus der Orthogonalität von $x - \tilde{x}$ und q, da gilt

$$q^T(x - \tilde{x}) = 0 \Leftrightarrow q^T(x - aq) = 0 \Leftrightarrow q^T x - aq^T q = 0 \Leftrightarrow a = \frac{q^T x}{q^T q}.$$

- Wenn q die Länge $||q|| = \sqrt{q^T q} = 1$ hat, dann gilt $a = \frac{q^T x}{||q||^2} = q^T x$.

Orthogonalprojektion



Theorem (Vektorkoordinaten bezüglich einer Orthogonalbasis)

Es sei $x \in \mathbb{R}^m$ und es sei $B := \{q_1, \ldots, q_m\}$ eine Orthonormalbasis von von $\mathbb{R}^m$. Dann ergeben sich für $i = 1, \ldots, m$ die Koordinaten a_i in der Basisdarstellung von x bezüglich B als die Projektionsfaktoren

$$a_i = x^T q_i \tag{2}$$

in der Orthogonalprojektion von x auf q_i. Äquivalent ist die Basisdarstellung von x bezüglich B gegeben durch

$$x = \sum_{i=1}^{m} (x^T q_i) q_i . \tag{3}$$

Beweis

Für $i = 1, \ldots, m$ gilt

$$x = \sum_{j=1}^{m} a_j q_j \Leftrightarrow q_i^T x = q_i^T \sum_{j=1}^{m} a_j q_j \Leftrightarrow q_i^T x = \sum_{j=1}^{m} a_j q_i^T q_j \Leftrightarrow q_i^T x = a_i \Leftrightarrow a_i = x^T q_i . \tag{4}$$

□

Theorem (Vektorkoordinatentransformation)

$B_v := \{v_1, \ldots, v_m\}$ und $B_w := \{w_1, \ldots, w_m\}$ seien zwei Orthonormalbasen eines Vektorraums. $A \in \mathbb{R}^{m \times m}$ sei die Matrix, die durch die spaltenweise Konkatenation der Koordinaten der Vektoren in B_w in der Basisdarstellung bezüglich der Basis B_v ergibt. Dann können die Koordinaten $x_i, i = 1, \ldots, m$ eines Vektors x bezüglich der Basis B_v in die Koordinaten $\tilde{x}_1, \ldots, \tilde{x}_m$ des Vektors bezüglich der Basis B_w durch

$$\tilde{x} = A^T x \tag{5}$$

transformiert werden. Analog können die Koordinaten $\tilde{y}_1, \ldots, \tilde{y}_m$ des Vektors hinsichtlich der Basis B_w in die Koordinaten $y_1, \ldots, y_m$ des Vektors hinsichtlich B_v durch

$$x = A\tilde{x}. \tag{6}$$

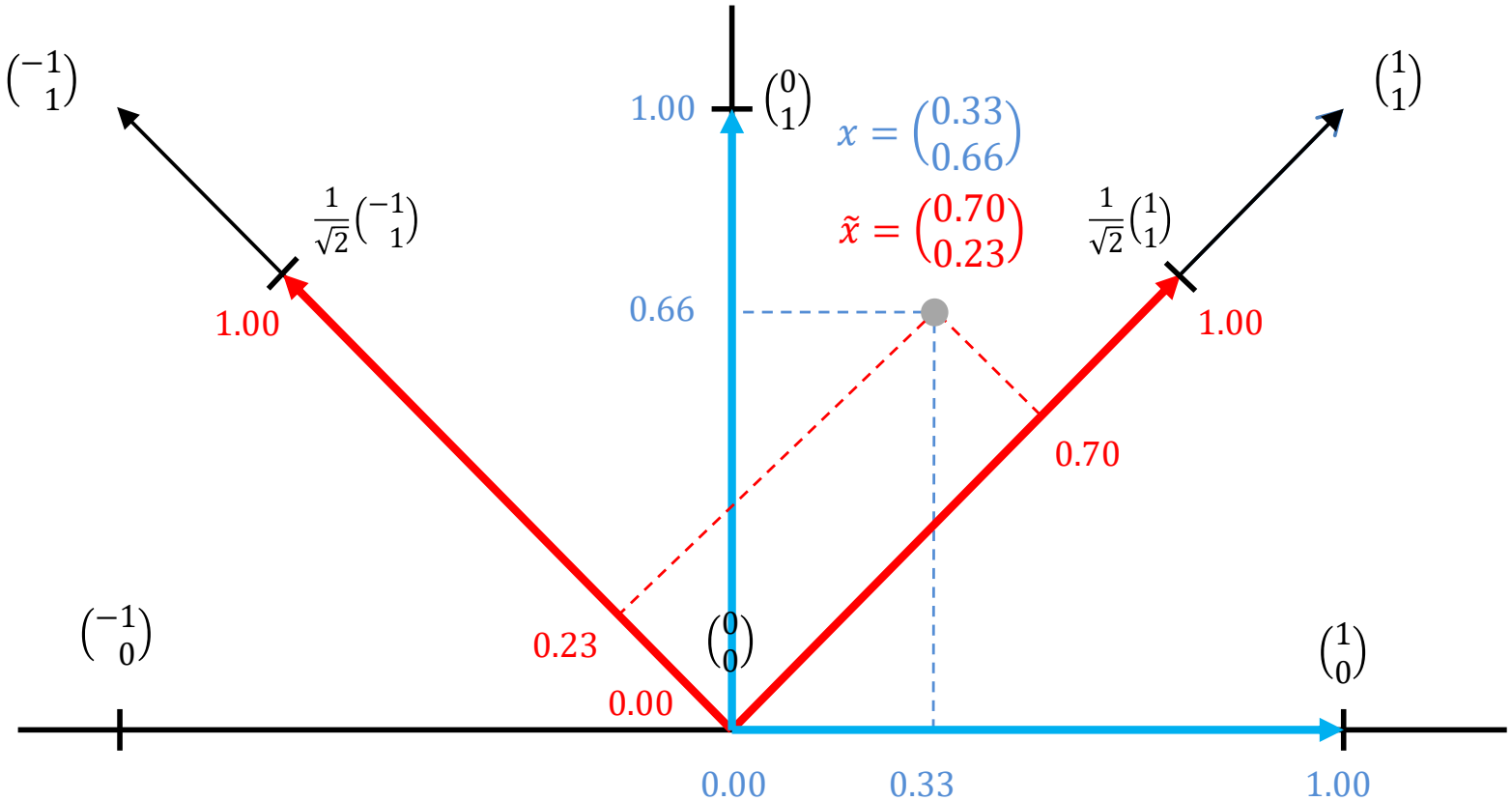
transformiert werden.

Bemerkungen

- Das Theorem erlaubt die Berechnung von Vektorkoordinaten bezüglich einer anderen Orthonormalbasis.
- Für die Berechnung muss zunächst die Matrix A gebildet und dann (nur) entsprechend multipliziert werden.
- Wir verzichten auf einen Beweis und demonstrieren das Theorem an einem Beispiel.

Ein Vektor wird hier als fester Punkt in $\mathbb{R}^m$ betrachtet; die Komponenten (Zahlen) des Vektors werden dagegen nur als Koordinaten bezüglich einer spezifischen Basis interpretiert!

Beispiel



Man beachte, dass x and $\tilde{x}$ am selben Ort in $\mathbb{R}^2$ liegen!

Beispiel

Wir nehmen an, dass wir die Koordinaten von $x = (1/3, 2/3)^T \in \mathbb{R}^2$ hinsichtlich der kanonischen Orthonormalbasis $B_v := \{e_1, e_2\}$ in die Koordinaten bezüglich der Basis

$$B_w := \left\{ \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \right\} \tag{7}$$

transformieren wollen. Die Basisdarstellungen der in Vektoren B_w bezüglich der Basisvektoren in B_v sind

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = a_{11} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_{21} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ and } \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = a_{12} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + a_{22} \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{8}$$

Die Projektionsfaktoren der Orthogonalprojektionen der Vektoren in B_w auf die Vektoren in B_v sind

$$a_{11} = \frac{1}{\sqrt{2}}, a_{21} = \frac{1}{\sqrt{2}}, a_{12} = -\frac{1}{\sqrt{2}}, a_{22} = \frac{1}{\sqrt{2}}. \tag{9}$$

Die Transformationsmatrix $A \in \mathbb{R}^{m \times m}$ in obigem Theorem ergibt sich also zu

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{10}$$

Die Vektorkoordinatentransformation von $x \in \mathbb{R}^2$ ergibt sich also zu

$$\tilde{x} = A^T x = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \end{pmatrix} \approx \begin{pmatrix} 0.70 \\ 0.23 \end{pmatrix}. \tag{11}$$

Vektorkoordinatentransformation

Definition und Theorem

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Definition (Hauptkomponentenanalyse)

$\mathbb{C}(X)$ sei die Kovarianzmatrix eines m-dimensionalen Zufallsvektors X. Dann heißt die orthonormale Zerlegung

$$\mathbb{C}(X) = Q\Lambda Q^T, \tag{12}$$

wobei

- $Q \in \mathbb{R}^{m \times m}$ die Matrix der spaltenweisen Konkatenation der Eigenvektoren von $\mathbb{C}(X)$ ist und
- $\Lambda \in \mathbb{R}^{m \times m}$ die Diagonalmatrix der zugehörigen Eigenwerte bezeichnen,

die *Hauptkomponentenanalyse* von $\mathbb{C}(X)$ und die Spalten von Q heißen die *Hauptkomponenten* von $\mathbb{C}(X)$. Der m-dimensionale Zufallsvektor

$$\tilde{X} = Q^T X \tag{13}$$

heißt *PCA-transformierter Zufallsvektor*.

Bemerkungen

- Man spricht auch von der Hauptkomponentenanalyse/den Hauptkomponenten von X.

Theorem (Hauptkomponentenanalyse)

$\mathbb{C}(X) \in \mathbb{R}^{m \times m}$ sei die Kovarianzmatrix eines m-dimensionalen Zufallsvektors X, es sei $\mathbb{E}(X) = 0_m$ und es sei

$$\mathbb{C}(X) = Q \Lambda Q^T, \tag{14}$$

die Hauptkomponentenanalyse von $\mathbb{C}(X)$. Dann gelten

(1) Die Spalten von Q bilden eine Orthonormalbasis von $\mathbb{R}^m$.

(2) Multiplikation mit Q^T transformiert die kanonischen Koordinaten von X in Koordinaten bezüglich der Hauptkomponenten von $\mathbb{C}(X)$.

(3) Die Kovarianzmatrix des PCA-transformierten Zufallsvektors ist die Diagonalmatrix Λ.

(4) In dem Koordinatensystem, das von den Hauptkomponenten von $\mathbb{C}(X)$ aufgespannt wird, gilt

$$\mathbb{V}(\tilde{X}_i) = \lambda_i \text{ für } i = 1, \ldots, m \text{ und } \mathbb{C}(\tilde{X}_i, \tilde{X}_j) = 0 \text{ für } i \neq j, 1 \leq i, j \leq m. \tag{15}$$

Beweis

(1) Mit dem Theorem zu den Eigenschaften von Basen aus Einheit (1) Vektoren gilt, dass jede Menge von m linear unabhängigen Vektoren Basis eines m-dimensionalen Vektorraums ist. Die Spalten $q_1, \ldots q_m \in \mathbb{R}^m$ von Q sind m orthonormale Vektoren und damit insbesondere auch linear unabhängig, denn für $i = 1, \ldots, m$ gilt

$$\begin{aligned} a_1 q_1 + a_2 q_2 + \cdots + a_m q_m &= 0_m \\ \Leftrightarrow (a_1 q_1 + a_2 q_2 + \cdots + a_m q_m)^T &= 0_m^T \\ \Leftrightarrow (a_1 q_1 + a_2 q_2 + \cdots + a_m q_m)^T q_i &= 0_m^T q_i \\ \Leftrightarrow \sum_{j=1}^{m} a_j q_j^T q_i &= 0 \\ \Leftrightarrow a_i &= 0. \end{aligned} \tag{16}$$

Es ist also $a_i = 0$ für $i = 1, \ldots, m$ und die einzige Repräsentation des Nullelements 0_m durch eine Linearkombination der Spalten von Q ist die triviale Repräsentation. Die Spalten von Q sind also m unabhängige Vektoren und damit eine Basis von $\mathbb{R}^m$. Da die Spalten von Q auch orthonormal sind, bilden sie eine Orthonormalbasis von $\mathbb{R}^m$.

Beweis (fortgeführt)

(2) Wir betrachten das Theorem zur Vektorkoordinatentransformation aus dieser Einheit und setzen $B_v := \{e_1, ..., e_m\}$ und $B_w := \{q_1, ...q_m\}$ mit den Spalten $q_1, ..., q_m \in \mathbb{R}^m$ von Q. Dann gilt, dass $Q \in \mathbb{R}^{m \times m}$ die Matrix ist, die sich durch die spaltenweise Konkatenation der Koordinaten der Vektoren in B_w in der Basisdarstellung bezüglich der Basis B_v ergibt, denn für $i = 1, ..., m$ gilt, dass die Basisdarstellung von q_i bezüglich der kanonischen Basis B_v gegeben ist durch

$$q_i = \sum_{j=1}^{m} (q_i^T e_j) e_j = \sum_{j=1}^{m} q_{i_j} e_j = q_i. \tag{17}$$

Äquivalent ist natürlich jeder Vektor $q \in \mathbb{R}^m$ schon immer identisch mit der Basisdarstellung von q bezüglich der kanonischen Basis. Damit folgt aber mit Theorem zur Vektorkoordinatentransformation direkt, dass der PCA-transfomierte Zufallsvektor

$$\tilde{X} = Q^T X \tag{18}$$

aus den Koordinaten des Vektors bezüglich der Hauptkomponenten von $\mathbb{C}(X)$ besteht.

(3) Wir erinnern zunächst daran, dass die inverse Matrix einer orthogonalen Matrix Q durch Q^T gegeben ist (vgl. Einheit (2) Definition symmetrischer, diagonaler, und orthogonaler Matrize). Mit $QQ^T = Q^T Q = I_m$ gilt dann, dass

$$\mathbb{C}(X) = Q\Lambda Q^T \Leftrightarrow Q^T \mathbb{C}(X) Q = Q^T Q \Lambda Q^T Q \Leftrightarrow Q^T \mathbb{C}(X) Q = \Lambda. \tag{19}$$

Beweis (fortgeführt)

Weiterhin gilt, dass mit $\mathbb{E}(X) = 0_m$ die Kovarianzmatrix von X gegeben ist durch (vgl. Einheit (3) Definition der Kovarianzmatrix)

$$\mathbb{C}(X) = \mathbb{E}\left((X - \mathbb{E}(X))\,(X - \mathbb{E}(X))^T\right) = \mathbb{E}\left(XX^T\right). \tag{20}$$

Damit ergibt sich für die Kovarianzmatrix des PCA-transformierte Vektors $\tilde{X} = Q^T X$ aber, dass

$$\begin{aligned}
\mathbb{C}(\tilde{X}) &= \mathbb{E}\left(\left(\tilde{X} - \mathbb{E}(\tilde{X})\right)\left(\tilde{X} - \mathbb{E}(\tilde{X})\right)^T\right) \\
&= \mathbb{E}\left(\left(Q^T X - \mathbb{E}(Q^T X)\right)\left(Q^T X - \mathbb{E}(Q^T X)\right)^T\right) \\
&= \mathbb{E}\left(\left(Q^T X - Q^T \mathbb{E}(X)\right)\left(Q^T X - Q^T \mathbb{E}(X)\right)^T\right) \\
&= \mathbb{E}\left((Q^T X)(Q^T X)^T\right) \\
&= Q^T \mathbb{E}\left(XX^T\right) Q \\
&= Q^T \mathbb{C}(X) Q \\
&= \Lambda.
\end{aligned} \tag{21}$$

(4) Die Koordinaten von $\tilde{X}$ entsprechen den Koordinaten von X in dem Koordinatensystem, dass von den Hauptkomponenten $q_1, \ldots, q_m$ von $\mathbb{C}(X)$ aufgespannt wird. Mit $\mathbb{C}(\tilde{X}) = \Lambda$ folgt Aussage (4) dann direkt mit der Definition der Kovarianzmatrix in Einheit (3). □

Definition und Theorem

Bemerkungen

- Die Eigenwerte $\lambda_1, ..., \lambda_m$ von $\mathbb{C}(X)$ sind die Varianzen von $\tilde{X}_1, ..., \tilde{X}_m$.

- Bei Annahme von $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ mit zugehörigen Eigenvektoren $q_1, ..., q_m$ gilt

$$\mathbb{V}(\tilde{X}_1) > \mathbb{V}(\tilde{X}_2) > \cdots > \mathbb{V}(\tilde{X}_m) \Leftrightarrow \mathbb{V}(q_1^T X) > \mathbb{V}(q_2^T X) > \cdots > \mathbb{V}(q_m^T X) \tag{22}$$

- Die paarweise nicht-identischen Kovarianzen der Komponenten von $\tilde{X}$ sind Null.

  $\Rightarrow$ Die Komponenten von $\tilde{X}$ sind unkorreliert.

  $\Rightarrow$ Die Komponenten von $\tilde{X}$ repräsentieren keine redundante Information.

- $q_1^T X$ maximiert die Varianz der unkorrelierten Linearkombinationen der Komponenten von X.

Definition (Hauptkomponentenanalyse eines Datensatzes)

$Y \in \mathbb{R}^{m \times n}$ sei ein Datensatz aus n unabhängigen Realisierungen eines m-dimensionalen Zufallsvektors und es sei $C \in \mathbb{R}^{m \times m}$ die Stichprobenkovarianzmatrix des Datensatzes. Dann heißt die Orthonormalzerlegung

$$C = Q \Lambda Q^T \tag{23}$$

wobei

- $Q \in \mathbb{R}^{m \times m}$ die spaltenweise Konkatenation der Eigenvektoren von C ist und
- $\Lambda \in \mathbb{R}^{m \times m}$ die Diagonalmatrix der zugehörigen Eigenwerten bezeichnen,

die *Hauptkomponentenanalyse von* C und die Spalten von Q heißen die *Hauptkomponenten von* C. Der $m \times n$-dimensionale Datensatz

$$\tilde{Y} = Q^T Y \tag{24}$$

heißt *PCA-transformierter Datensatz*.

Bemerkungen

- Man spricht auch von der Hauptkomponentenanalyse/den Hauptkomponenten des Datensatzes Y.

Theorem (Hauptkomponentenanalyse eines Datensatzes)

$C \in \mathbb{R}^{m \times m}$ sei die Stichprobenkovarianzmatrix eines Datensatzes $Y \in \mathbb{R}^{m \times n}$ und es sei

$$C = Q \Lambda Q^T, \tag{25}$$

die Hauptkomponentenanalyse von C. Dann gelten

(1) Die Spalten von Q bilden eine Orthonormalbasis von $\mathbb{R}^m$.

(2) Multiplikation mit Q^T transformiert die kanonischen Koordinaten der Spalten von Y in Koordinaten bezüglich der Hauptkomponenten von C.

(3) Die Stichprobenkovarianzmatrix des PCA-transformierten Datensatzes ist die Diagonalmatrix Λ.

(4) In dem Koordinatensystem, das von den Hauptkomponenten von C aufgespannt wird, gilt

$$S^2(\tilde{Y}_i) = \lambda_i \text{ für } i = 1, \ldots, m \text{ und } C(\tilde{Y}_i, \tilde{Y}_j) = 0 \text{ für } i \neq j, 1 \leq i, j \leq m. \tag{26}$$

wobei $S^2(\tilde{Y}_i)$ die Stichprobenvarianz der $i$ten Komponente des Datensatzes und $C(\tilde{Y}_i, \tilde{Y}_j)$ die Stichprobenkovarianz der $i$ten und $j$ten Komponente des Datensatzes bezeichnen.

Bemerkungen

- Der Beweis ergibt sich in Analogie zum Beweis Theorems zur Hauptkomponentenanalyse
- Wir verzichten aus eine Ausformulierung des Beweises.

Hauptkomponentenanalyse eines simulierten Datensatzes

Datensatzgeneration

```
# R Pakete
library(matrixcalc)                          # Matrix Paket (is.positive.definite())
library(MASS)                                # Multivariate Normalverteilung (mvrnorm())

# Simulationsparameter
set.seed(1)                                  # Reproduzierbare Randomisierung
m     = 5                                    # Datenpunktdimension
n     = 20                                   # Anzahl Realisierung
mu    = rep(0,m)                             # Erwartungswertparameter
Sigma = matrix(runif(m^2), nrow = m)         # zufällige Matrix
Sigma = 0.5*(Sigma+t(Sigma))                 # symmetrische Matrix
Sigma = Sigma + m*diag(m)                    # positiv definite Matrix
print(is.positive.definite(Sigma))           # Positiv-Definitheits Check
```

```
> [1] TRUE
```

```
# Datensatzgeneration
Y     = t(mvrnorm(n,mu,Sigma))
```

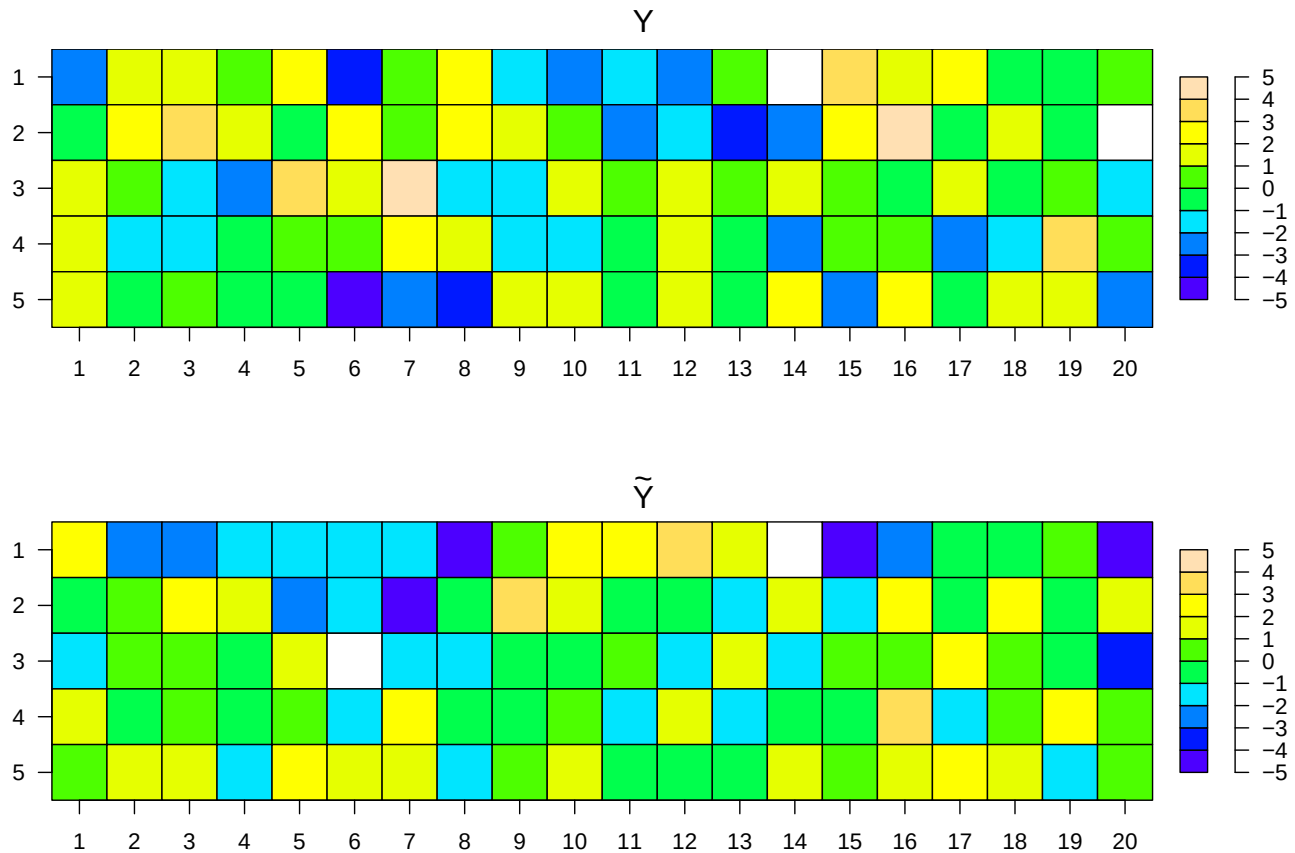
Hauptkomponentenanalyse eines simulierten Datensatzes

Hauptkomponentenanalyse durch Eigenanalyse

```
# Hauptkomponentenanalyse durch Eigenanalyse
I_n     = diag(n)                                    # Einheitsmatrix I_n
J_n     = matrix(rep(1,n^2), nrow = n)               # 1_{nn}
C       = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y)) # Stichprobenkovarianzmatrix
D       = diag(1/sqrt(diag(C)))                      # Kov-Korr-Transformationsmatrix
R       = D %*% C %*% D                              # Stichprobenkorrelationsmatrix
EA      = eigen(C)                                   # Eigenanalyse von C
lambda  = EA$values                                  # Eigenwerte von C
Q       = EA$vectors                                 # Eigenvektoren von C
Y_tilde = t(Q) %*% Y                                 # Transformierter Datensatz

# Stichproben- und Korrelationsmatrix des transformierten Datensatzes
C_tilde = (1/(n-1))*(Y_tilde %*% (I_n-(1/n)*J_n) %*% t(Y_tilde))
D_tilde = diag(1/sqrt(diag(C_tilde)))
R_tilde = D_tilde %*% C_tilde %*% D_tilde
```

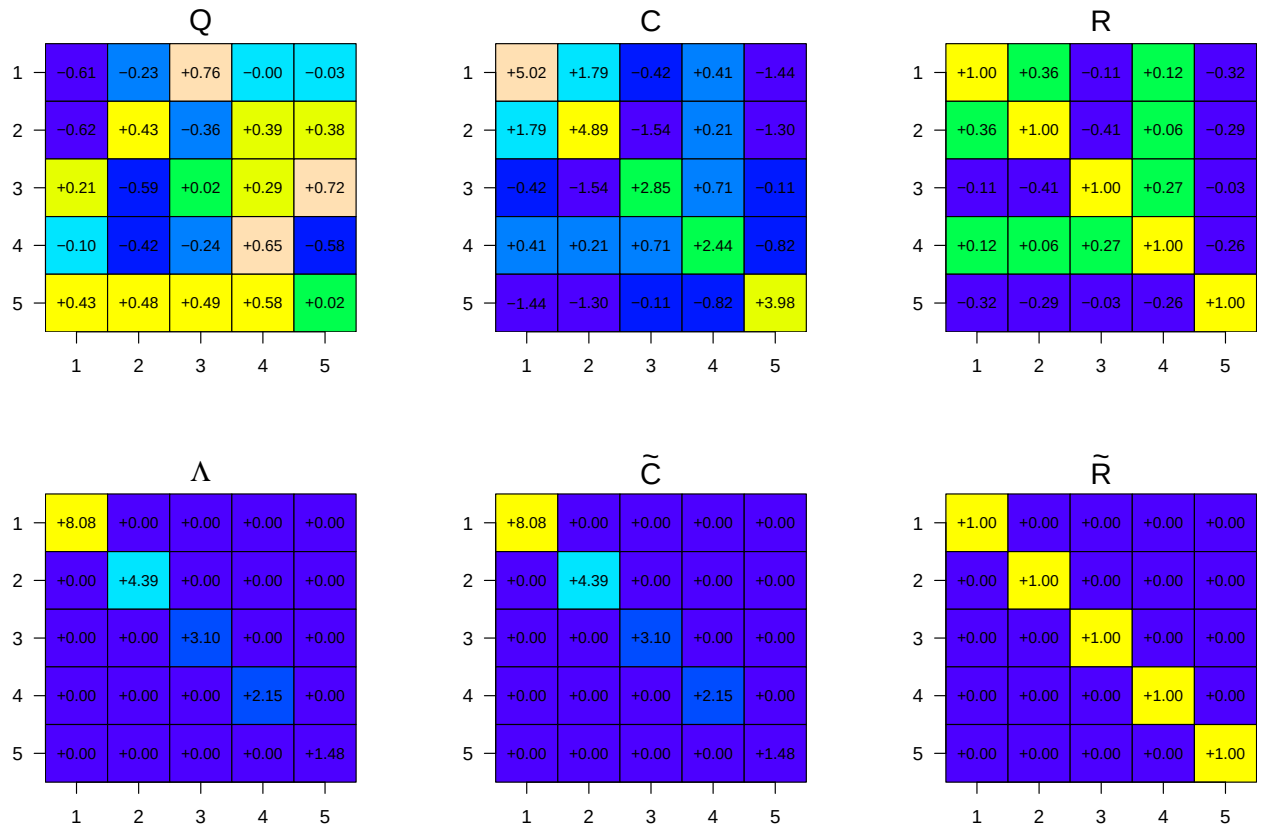
Hauptkomponentenanalyse eines simulierten Datensatzes

Hauptkomponentenanalyse durch Eigenanalyse



Hauptkomponentenanalyse eines simulierten Datensatzes

Hauptkomponentenanalyse durch Eigenanalyse



Vektorkoordinatentransformation

Definition

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Definition (Singulärwertzerlegung)

$X \in \mathbb{R}^{m \times n}$ sei eine Matrix. Dann heißt die Zerlegung

$$X = USV^T, \tag{27}$$

wobei $U \in \mathbb{R}^{m \times m}$ eine orthogonale matrix ist, $S \in \mathbb{R}^{m \times n}$ eine Diagonalmatrix ist und $V \in \mathbb{R}^{n \times n}$ eine orthogonale Matrix ist, *Singulärwertzerlegung (Singular Value Decomposition (SVD))* von X. Die Diagonalelemente von S heißen die *Singulärwerte* von X.

Bemerkungen

- Die Existenz der Singulärwertzerlegung folgt aus dem Spektralsatz der Linearen Algebra.
- Singulärwertzerlegungen können in R mit svd() berechnet werden.

Theorem (Singulärwertzerlegung und Eigenanalyse)

$X \in \mathbb{R}^{m \times n}$ sei eine Matrix und

$$X = USV^T \tag{28}$$

sei ihre Singulärwertzerlegung. Dann gilt:

- Die Spalten von U sind die Eigenvektoren von XX^T,
- die Spalten von V sind die Eigenvektoren von X^TX und
- die entsprechenden Singulärwerte sind die Quadratwurzeln der zugehörigen Eigenwerte.

Bemerkung

- Singulärwertzerlegung und Eigenanalyse sind eng verwandt.

Beweis

Wir halten zunächst fest, dass mit

$$\left(XX^T\right)^T = XX^T \text{ and } \left(X^TX\right)^T = X^TX, \tag{29}$$

XX^T und X^TX symmetrische Matrizen sind und somit Orthornomalzerlegungen haben. Wir halten weiterhin fest, dass mit der Definition der Singulärwertzerlegung gelten, dass sowohl

$$XX^T = USV^T \left(U\Sigma V^T\right)^T = USV^TVS^TU^T = USSU^T = U\Lambda U^T \tag{30}$$

als auch

$$X^TX = \left(USV^T\right)^T USV^T = VS^TUUS^TV^T = V\Lambda V^T \tag{31}$$

ist, wobei wir $\Lambda := SS$ definiert haben. Weil das Produkt von Diagonalmatrizen wieder eine Diagonalmatrix ist, ist Λ eine Diagonalmatrix und per Definition sind U und V orthogonale Matrizen. Wir haben also XX^T und X^TX in Form der Orthonormalzerlegungen

$$XX^T = U\Lambda U^T \text{ and } X^TX = V\Lambda V^T \tag{32}$$

geschrieben und damit ist alles gezeigt.

□

Theorem (Datenhauptkomponentenanalyse durch Singulärwertzerlegung)

$Y \in \mathbb{R}^{m \times n}$ sei ein Datensatz von n unabhängigen Realisierungen eines m-dimensionalen Zufallsvektors. Weiterhin sei

$$\frac{1}{\sqrt{n-1}} Y_c = USV^T \tag{33}$$

die Singulärwertzerlegung des skalierten und zentrierten Datenmatrix. Dann sind die Spalten von U die Eigenvektoren der Stichprobenkovarianzmatrix des Datensatzes Y und die quadrierten Singulärwerte sind die zugehörigen Eigenwerte.

Beweis

Nach Definition der Singulärwertzerlegung sind die Spalten von U die Eigenvektoren von

$$\frac{1}{\sqrt{n-1}} Y_c \frac{1}{\sqrt{n-1}} Y_c^T = \frac{1}{n-1} Y_c Y_c^T =: C, \tag{34}$$

und damit identisch mit den Eigenvektoren der Stichprobenkovarianzmatrix. Weil die Singulärwerte die Quadratwurzeln der zugehörigen Eigenwerte sind, sind ihre quadrierten Werte identisch zu den zugehörigen Eigenwerten.

□

Eine PCA kann durch Eigenanalyse der Stichprobenkovarianzmatrix berechnet werden

Eine PCA kann auch durch Singulärwertzerlegung der skaliert-zentrierten Datenmatrix berechnet werden

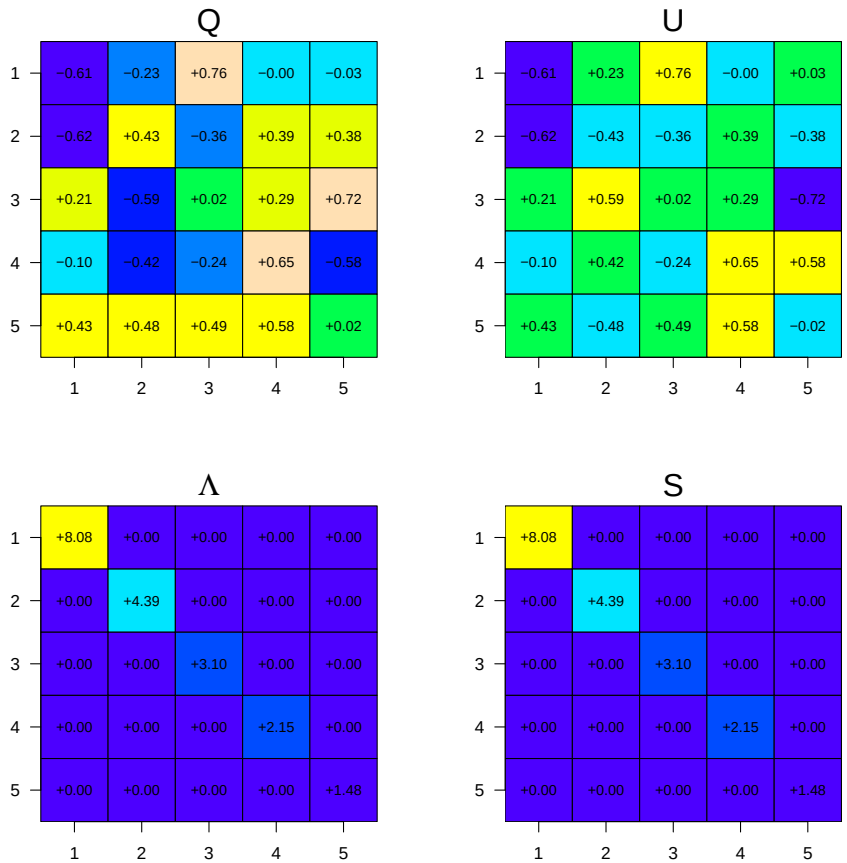
Hauptkomponentenanalyse eines simulierten Datensatzes

Hauptkomponentenanalyse durch Singulärwertzerlegung

```
# Hauptkomponentenanalyse durch Singulärwertzerlegung
Y_sc    = (1/sqrt(n-1))*(Y-as.matrix(rowMeans(Y)) %*% rep(1,n)) # skaliert-zentrierter Datensatz
SWD     = svd(Y_sc)                                              # Singulärwertzerlegung
U       = SWD$u                                                  # Eigenvektoren von C
s       = SWD$d^2                                                # Eigenwerte von C
Y_tilde = t(U) %*% Y                                             # Transformierter Datensatz

# Stichproben- und Korrelationsmatrix des transformierten Datensatzes
C_tilde = (1/(n-1))*(Y_tilde %*% (I_n-(1/n)*J_n) %*% t(Y_tilde))
D_tilde = diag(1/sqrt(diag(C_tilde)))
R_tilde = D_tilde %*% C_tilde %*% D_tilde
```

Hauptkomponentenanalyse eines simulierten Datensatzes

Hauptkomponentenanalyse durch Singulärwertzerlegung



Vektorkoordinatentransformation

Definition

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Überblick

- Datenkompression entspricht einer Reduktion der Dimension m von Daten.
- Im Rahmen der prädiktiven Modellierung wird PCA zur *Dimensionsreduktion* eingesetzt.
- Ziel ist es hier, dem Undersampling hochdimensionaler Datenräume entgegen zu wirken.
- Dimensionsreduktion entspricht dem Verwerfen von $k < m$ Komponenten von $\tilde{Y}$.
- Wahl von Komponenten mit hohen Eigenwerten hält den *Datenrekonstruktionsfehler* klein.

Definition (Dimensionsreduzierter Datensatz)

$Y \in \mathbb{R}^{m \times n}$ sei ein Datensatz,

$$C = \frac{1}{n-1}\left(Y\left(I_n - \frac{1}{n}1_{nn}\right)Y^T\right) \in \mathbb{R}^{m \times m} \tag{35}$$

sei die zugehörige Stichprobenkovarianzmatrix,

$$C = Q\Lambda Q^T \tag{36}$$

sei die Hauptkomponentenanalyse von C und es gelte $\lambda_1 > \lambda_2 > \cdots > \lambda_m$ für die Diagonalelemente von Λ. Schließlich sei für $k \leq m$ Q_k die Matrix, die aus Q durch Streichen der Spalten $k+1, \ldots, m$ entsteht. Dann heißt

$$\tilde{Y}_k = Q_k^T Y \in \mathbb{R}^{k \times n} \tag{37}$$

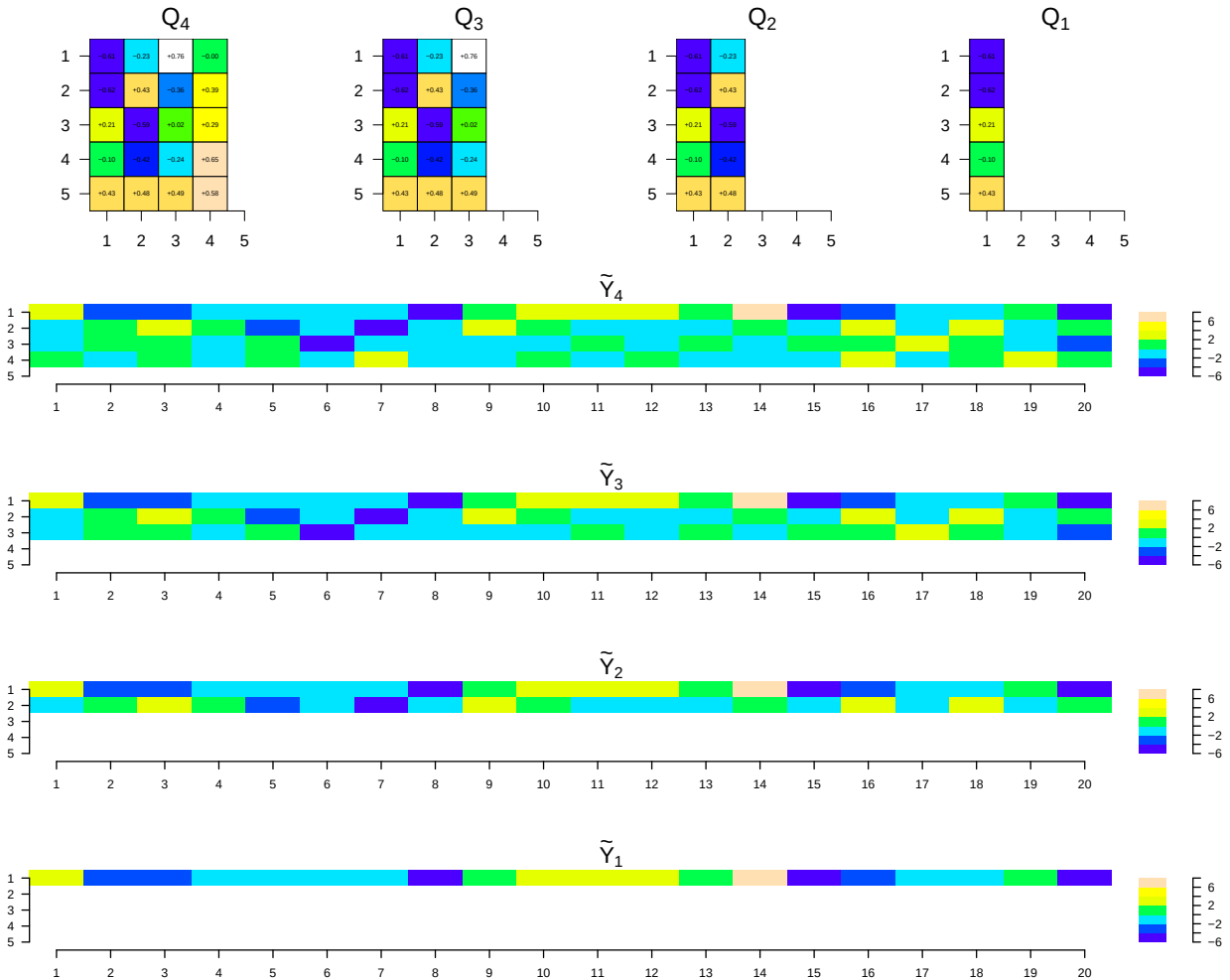
PCA-dimensionsreduzierter Datensatz.

Bemerkung

- $\tilde{Y}_k = Q_k^T Y$ entspricht einer $(k \times n) = (k \times m) \cdot (m \times n)$ Matrixmultiplikation
- $\tilde{Y}_k$ ist der Datensatz, der aus $\tilde{Y}$ durch Streichen der $(k+1)$-ten bis m-ten Zeile ensteht.

Dimensionalitätsreduktion eines simulierten Datensatzes

Dimensionsreduzierte Datensätze



Definition (Rekonstruierter Datensatz, Datenrekonstruktionsfehler)

$Y \in \mathbb{R}^{m \times n}$ sei ein Datensatz und für $k \leq m$ sei

$$\tilde{Y}_k = Q_k^T Y \in \mathbb{R}^{k \times n} \tag{38}$$

ein PCA-dimensionsreduzierter Datensatz. Dann heißt

$$Y_k = Q_k \tilde{Y}_k \in \mathbb{R}^{m \times n} \tag{39}$$

rekonstruierter Datensatz und

$$e = \|\text{vec}(Y - Y_k)\| \geq 0 \tag{40}$$

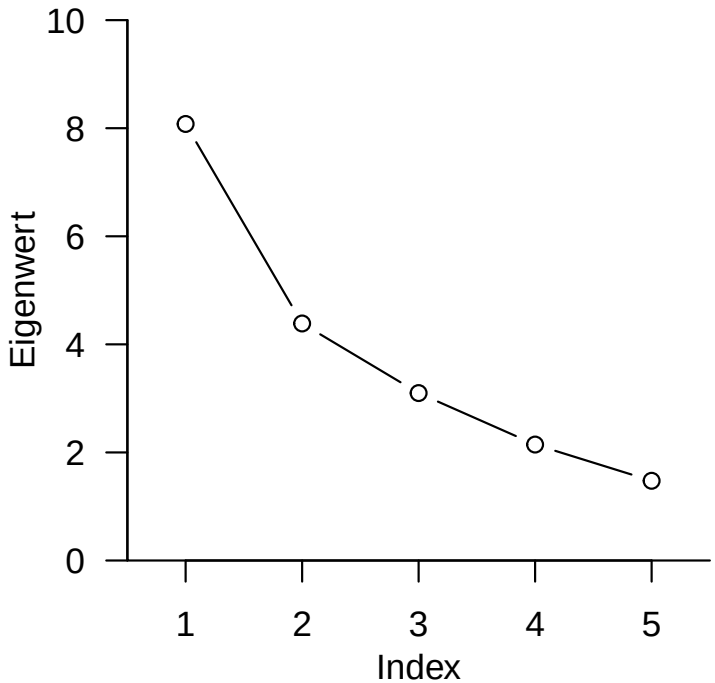
heißt *Datenrekonstruktionsfehler*.

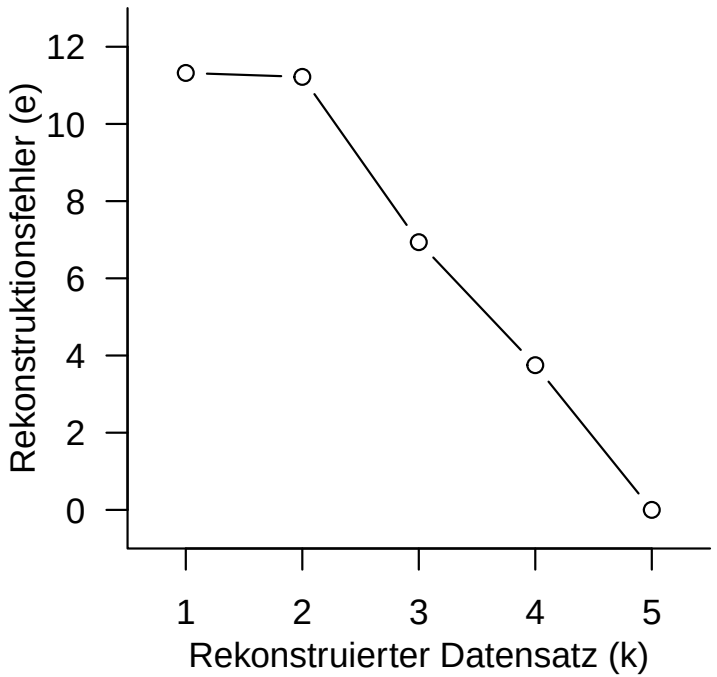
Bemerkungen

- $Y_k = Q_k \tilde{Y}_k$ entspricht einer $(m \times n) = (m \times k) \cdot (k \times n)$ Matrixmultiplikation
- Für $M \in \mathbb{R}^{m \times n}$ ist $\text{vec}(M) \in \mathbb{R}^{mn}$ der Vektor, der durch Stapeln der Spalten von M entsteht.
- Für $k = m$ gilt $Q\tilde{Y}_k = QQ^T Y = Y$ und damit $e = 0$.

Datensatzrekonstruktion und -rekonstruktionsfehler eines simulierten Datensatzes

Eigenwerte (“Scree-Plot”) und Rekonstruktionsfehler



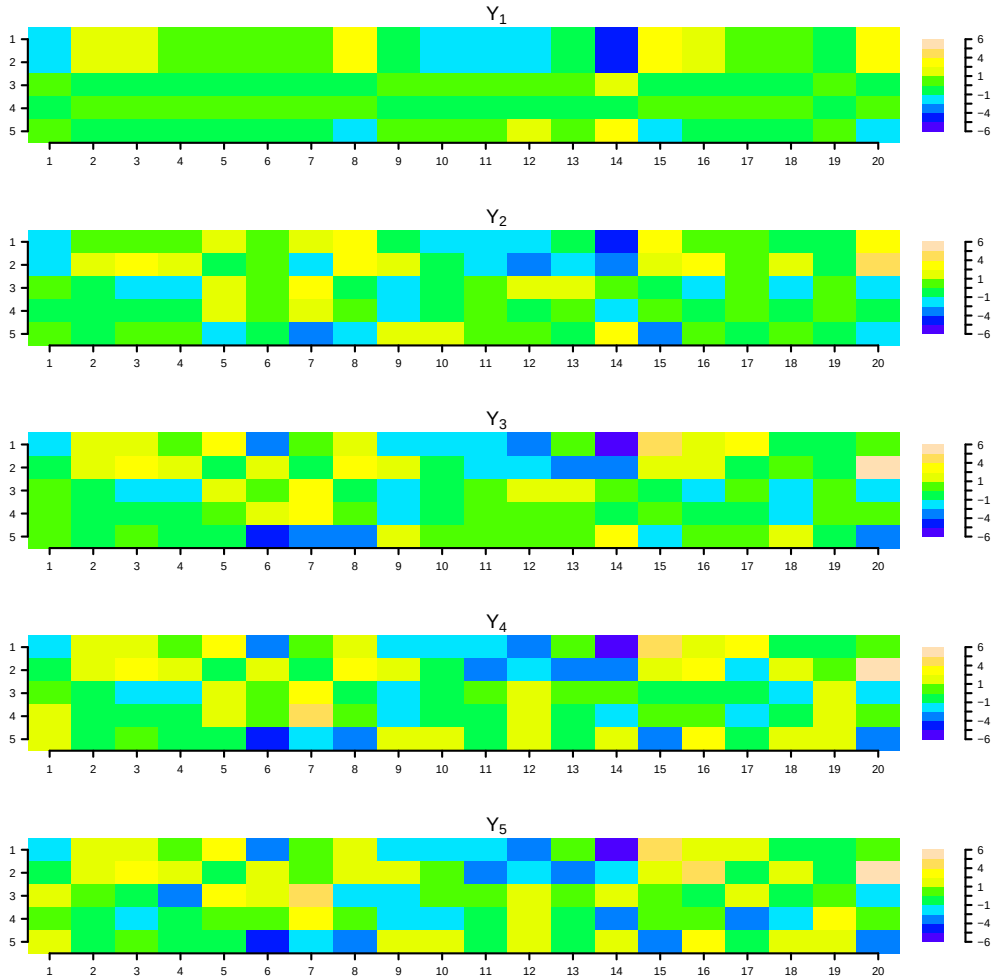


Datensatzrekonstruktion und -rekonstruktionsfehler eines simulierten Datensatzes

Scree (engl.) Schutthalde

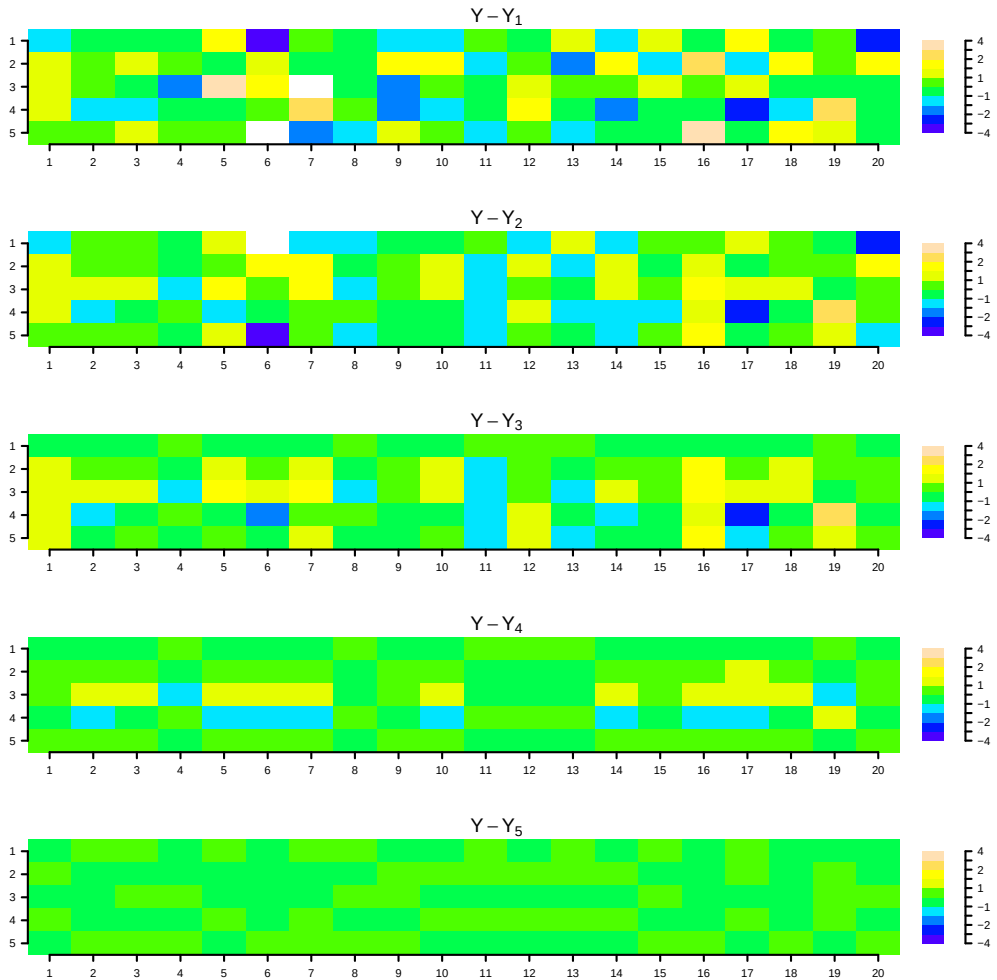
Datensatzrekonstruktion und -rekonstruktionsfehler eines simulierten Datensatzes

Rekonstruierte Datensätze



Datensatzrekonstruktion und -rekonstruktionsfehler eines simulierten Datensatzes

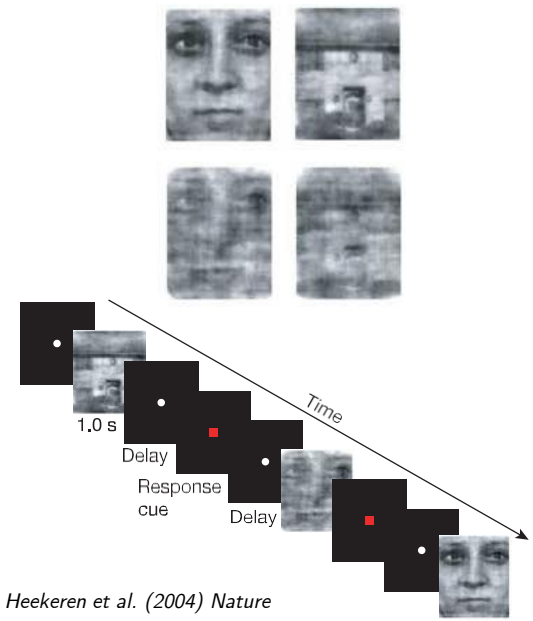
Originaldatensatz minus rekonstruierter Datensatz



Featureselektion bei EEG Daten

Wie werden visuelle Stimuli im Gehirn verarbeitet?

Wie entscheiden Menschen, ob sie ein Haus oder ein Gesicht wahrnehmen?



Heekeren et al. (2004) Nature

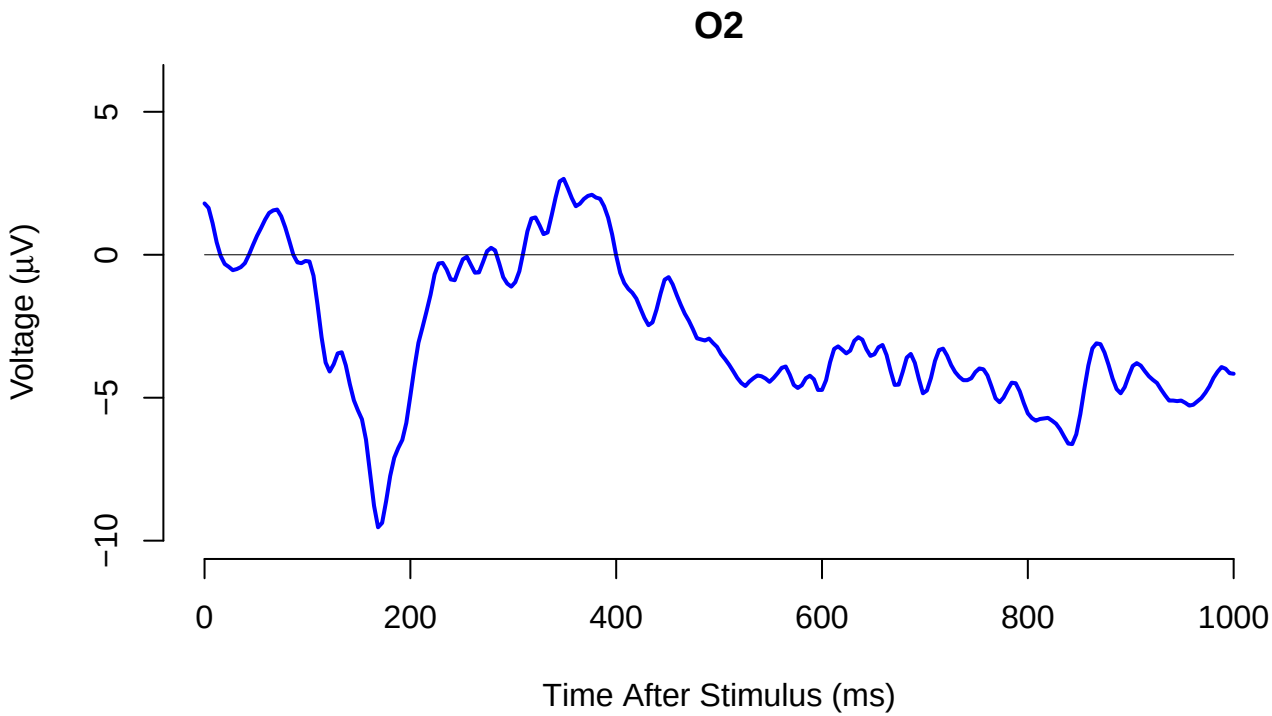
→ Allgemeine Psychologie, Biologische Psychologie, Kognitive Neurowissenschaften

Featureselektion bei EEG Daten

- In der prädiktiven EEG Analyse werden mentale Zustände aus Raumzeitserien dekodiert.
- Dabei wird ein Klassifikationsalgorithmus anhand von Single-Trial-Daten trainiert.
- Oft ist die Datendimension (Elektroden $\times$ Perieventsamples) sehr viel höher als die Trialanzahl.
- Wegen des Curse of Dimensionality performen Classifier auf Rohdatensätzen nicht optimal.
- Im Rahmen der Featureselektion können redundante Features durch PCA entfernt werden.
- Featureselektion entspricht einer Dimensionsreduktion und kann Klassifikationsraten erhöhen.
- Wir visualisieren das Vorgehen exemplarisch für eine Single-Trial-Elektrodenraum-Zeitserie.

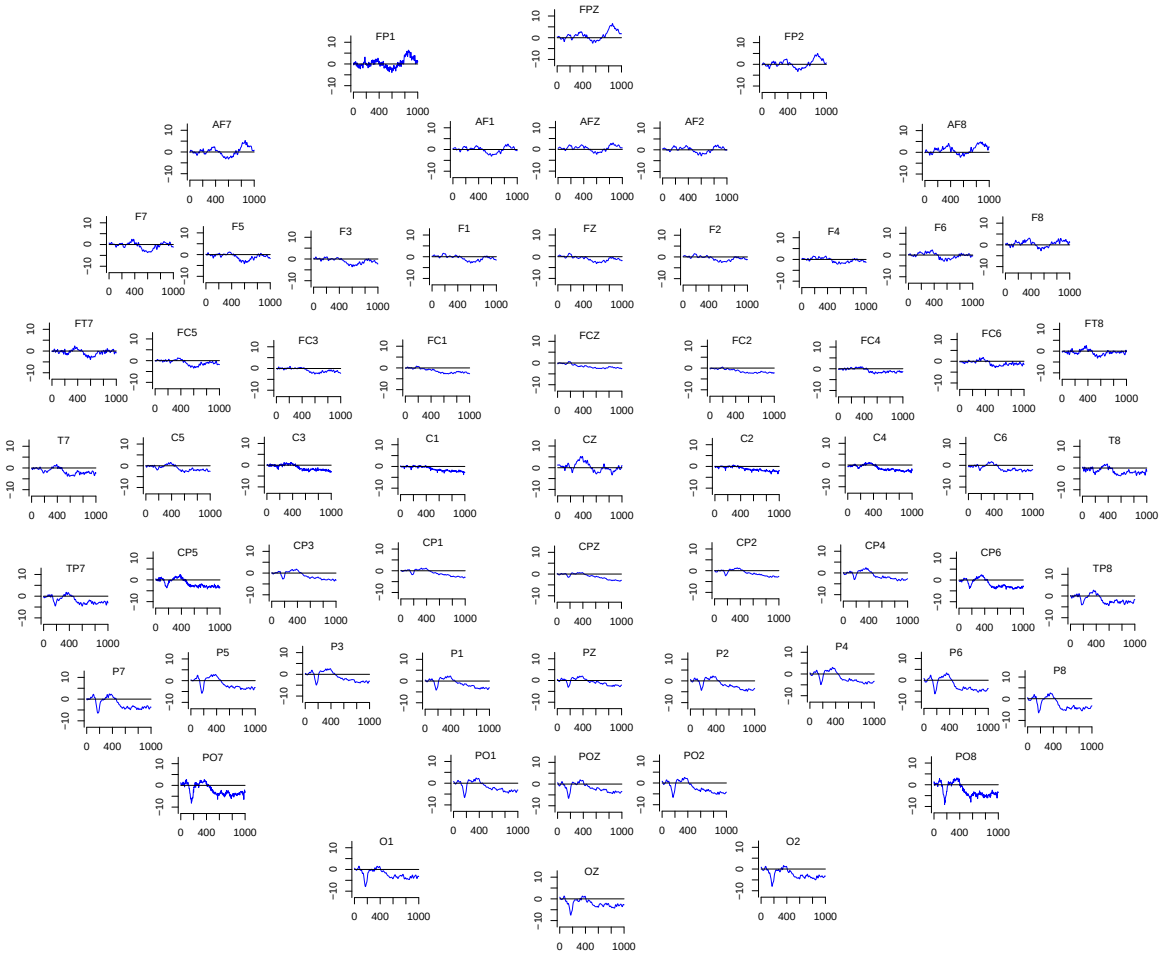
Featureselektion bei EEG Daten

Single-Trial Evoziertes Potential einer Elektrode



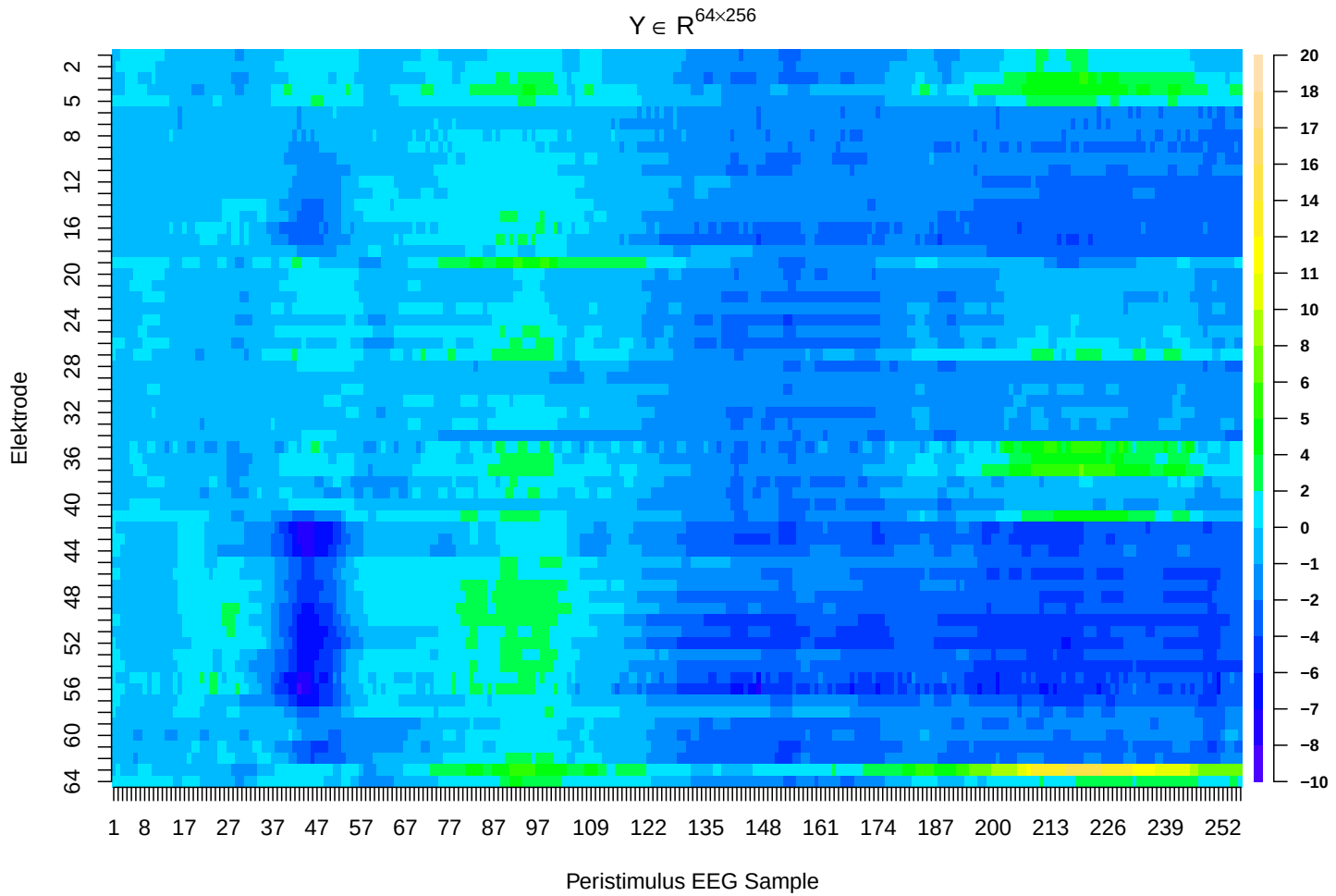
Featureselektion bei EEG Daten

Single-Trial Evoziertes Potential aller Elektroden



Featureselektion bei EEG Daten

Single-Trial Evoziertes Potential aller Elektroden in Matrixform



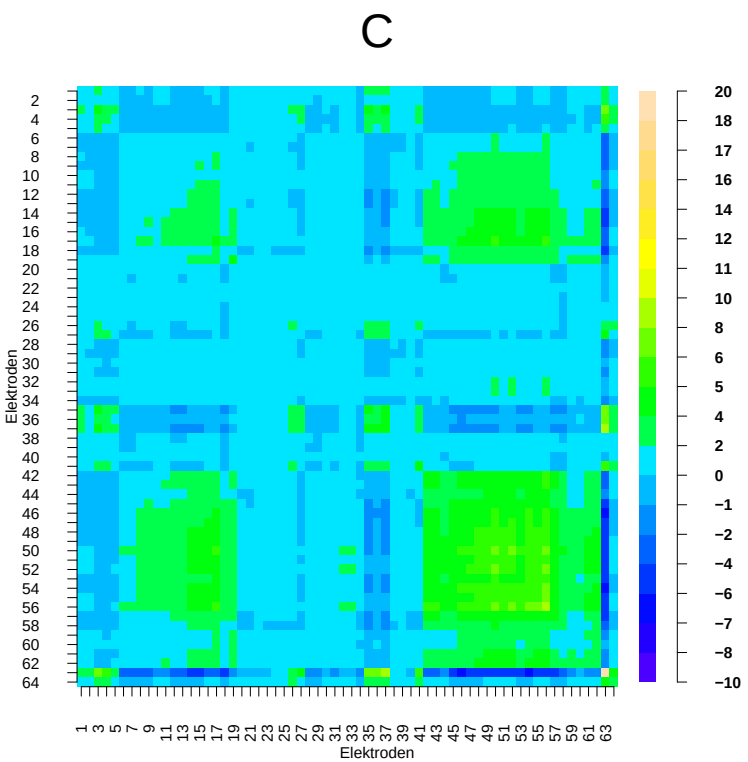
Featureselektion bei EEG Daten

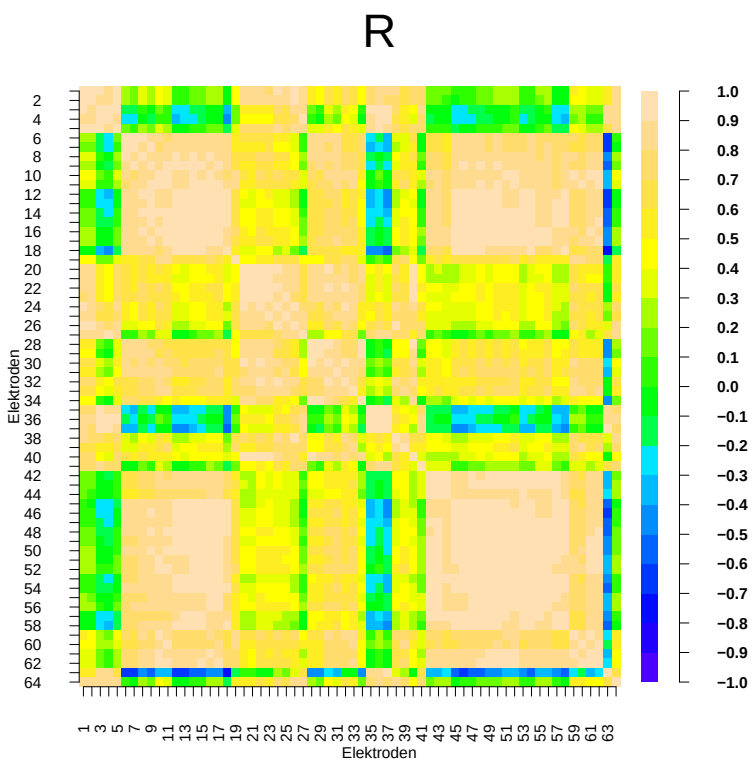
Hauptkomponentenanalyse

```
# Laden der Daten
Y       = as.matrix(readRDS(file.path(getwd(), "4_Daten", "eeg.csv")))

# Hauptkomponentenanalyse durch Eigenanalyse
m       = nrow(Y)                                    # Datendimension (Anzahl Elektroden)
n       = ncol(Y)                                    # Datenpunktanzahl (Anzahl Time-Bins)
I_n     = diag(n)                                    # Einheitsmatrix I_n
J_n     = matrix(rep(1,n^2), nrow = n)               # 1_{nn}
C       = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y)) # Stichprobenkovarianzmatrix
D       = diag(1/sqrt(diag(C)))                      # Kov-Korr-Transformationsmatrix
R       = D %*% C %*% D                              # Stichprobenkorrelationsmatrix
EA      = eigen(C)                                   # Eigenanalyse von C
lambda  = EA$values                                  # Eigenwerte von C
Q       = EA$vectors                                 # Eigenvektoren von C
Y_tilde = t(Q) %*% Y                                 # Transformierter Datensatz
```

Featureselektion bei EEG Daten

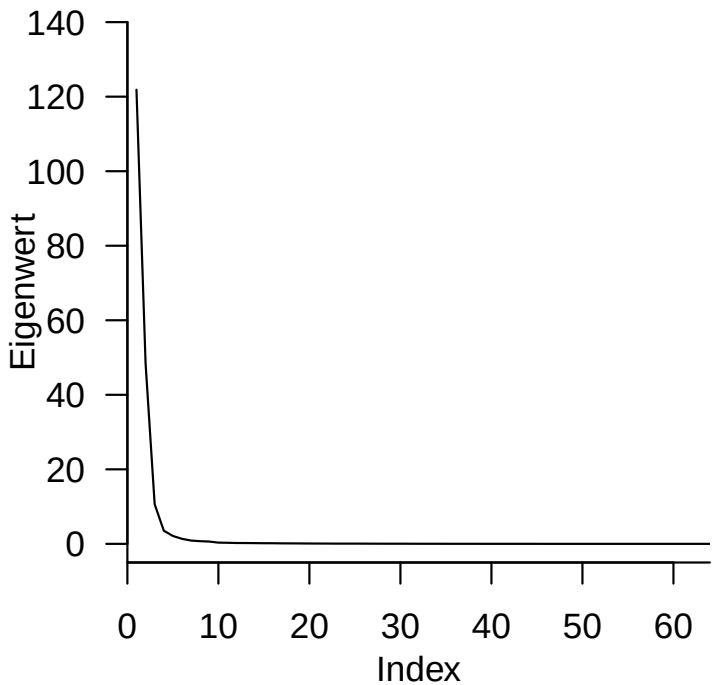
Stichprobenkovarianzmatrix und Stichprobenkorrelationsmatrix

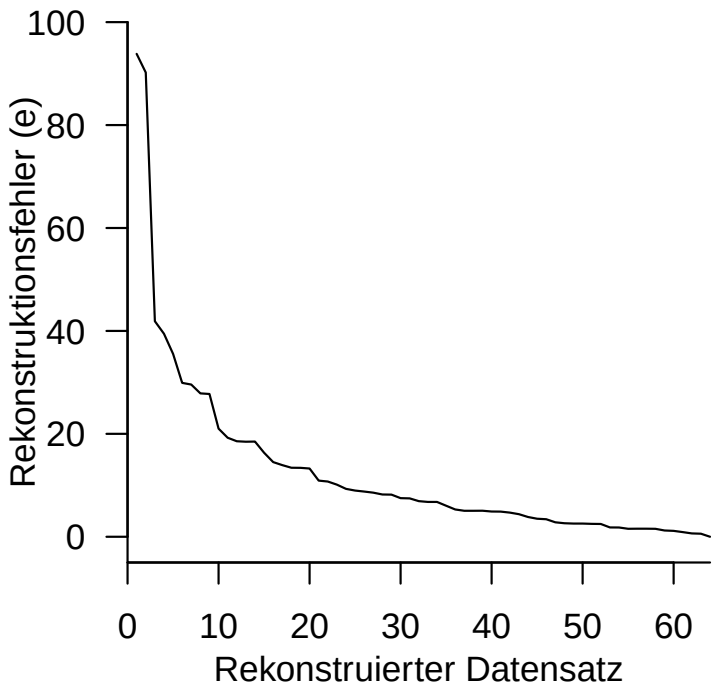




Featureselektion bei EEG Daten

Eigenwerte ("Scree (engl. Schutthalde)-Plot") und Rekonstruktionsfehler





Vektorkoordinatentransformation

Definition

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Überblick

Faktorenanalyse (oder Faktoranalyse) ist ein Sammelbegriff für viele datenanalytische Verfahren.

Prinzipiell entsprechen "Faktoren" latenten Zufallsvariablen in probabilistischen Modellen.

Latente Zufallsvariablen sind nur indirekt beobachtbar.

In der Psychologie dienen latente Zufallsvariablen oft der Modellierung von "Konstrukten".

Wir betrachten probabilistische Modelle mit latenten Zufallsvariablen in (5) Faktorenanalyse.

"Exploratorische Faktorenanalyse (EFA)" ist eine intuitive Vorform des Latent-Variable-Modellings.

Im Wesentlichen entspricht EFA einer speziellen Interpretation einer Hauptkomponentenanalyse.

Wir erläutern EFA hier anhand der Sedimentationshypothese der Persönlichkeitspsychologie.

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

- Alle wichtigen Persönlichkeitseigenschaften sind durch Adjektive repräsentiert.
- Persönlichkeitsadjektive haben sich analog zu Persönlichkeitseigenschaften entwickelt
- Persönlichkeitsadjektive decken die relevanten individuellen Differenzen ab.
- $\Rightarrow$ Big-Five der Persönlichkeitspsychologie (OCEAN-Modell)

Datengeneration

- 30 Proband:innen schätzen eine Person hinsichtlich des Zutreffens von Adjektiven ein.
- Neunstufige Skala (1: trifft überhaupt nicht zu, 9: trifft voll zu)
- Hochkorrelierte Ajdektive beschreiben eine "übergeordnete Persönlichkeitseigenschaften".

Rudolf & Buse (2020) Multvariate Verfahren Kapitel 9

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

Datensatz (Proband:innen 1 bis 15)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
angriffslustig	4	9	5	8	7	2	4	3	1	4	2	6	4	8	7
penibel	6	3	1	1	3	4	3	5	4	7	6	4	8	7	8
streitbar	3	8	4	6	6	3	3	3	1	3	2	4	3	7	7
kämpferisch	4	6	2	8	7	3	4	4	5	6	5	7	5	8	8
grimmig	5	4	3	4	4	5	5	4	3	4	2	3	5	7	6
gründlich	5	2	2	2	3	4	5	5	4	6	6	5	6	6	7
akkurat	5	2	1	1	3	4	4	5	4	6	6	5	7	5	5
gewissenhaft	1	2	3	2	4	3	6	4	4	5	6	2	5	4	4
kleinlich	5	1	1	1	3	1	2	1	2	3	6	1	1	7	8
übergenau	6	1	1	1	3	4	3	5	4	7	6	1	1	7	8
herausfordernd	4	1	2	8	7	3	4	4	1	1	2	3	5	7	6
hitzig	5	4	3	4	4	5	6	4	3	4	2	3	5	7	6

Rudolf & Buse (2020) Multvariate Verfahren Kapitel 9

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

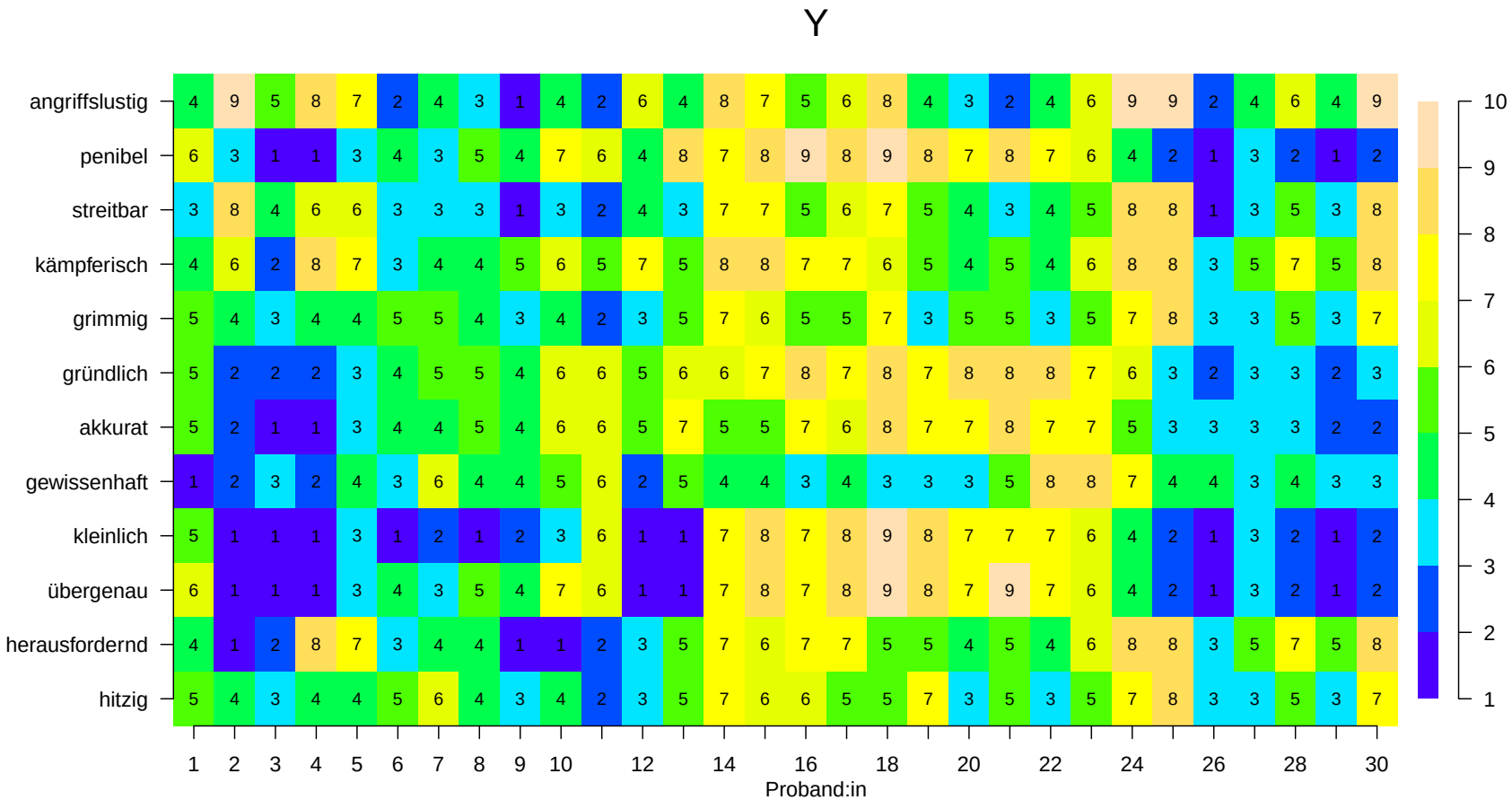
Datensatz (Proband:innen 16 bis 30)

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
angriffslustig	5	6	8	4	3	2	4	6	9	9	2	4	6	4	9
penibel	9	8	9	8	7	8	7	6	4	2	1	3	2	1	2
streitbar	5	6	7	5	4	3	4	5	8	8	1	3	5	3	8
kämpferisch	7	7	6	5	4	5	4	6	8	8	3	5	7	5	8
grimmig	5	5	7	3	5	5	3	5	7	8	3	3	5	3	7
gründlich	8	7	8	7	8	8	8	7	6	3	2	3	3	2	3
akkurat	7	6	8	7	7	8	7	7	5	3	3	3	3	2	2
gewissenhaft	3	4	3	3	3	5	8	8	7	4	4	3	4	3	3
kleinlich	7	8	9	8	7	7	7	6	4	2	1	3	2	1	2
übergenau	7	8	9	8	7	9	7	6	4	2	1	3	2	1	2
herausfordernd	7	7	5	5	4	5	4	6	8	8	3	5	7	5	8
hitzig	6	5	5	7	3	5	3	5	7	8	3	3	5	3	7

Rudolf & Buse (2020) Multvariate Verfahren Kapitel 9

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

Datensatz



Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

```
# Hauptkomponentenanalyse durch Eigenanalyse
m       = nrow(Y)                                  # Variablenanzahl (Anzahl Adjektive)
n       = ncol(Y)                                  # Datenpunktanzahl (Anzahl Proband:innen)
I_n     = diag(n)                                  # Einheitsmatrix I_n
J_n     = matrix(rep(1,n^2), nrow = n)             # 1_{nn}
C       = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))  # Stichprobenkovarianzmatrix
D       = diag(1/sqrt(diag(C)))                    # Kov-Korr-Transformationsmatrix
R       = D %*% C %*% D                            # Stichprobenkorrelationsmatrix
EA      = eigen(C)                                 # Eigenanalyse von C
lambda  = EA$values                                # Eigenwerte von C
Q       = EA$vectors                               # Eigenvektoren von C
Y_tilde = t(Q) %*% Y                               # Transformierter Datensatz

# Stichproben- und Korrelationsmatrix des transformierten Datensatzes
C_tilde = (1/(n-1))*(Y_tilde %*% (I_n-(1/n)*J_n) %*% t(Y_tilde))
D_tilde = diag(1/sqrt(diag(C_tilde)))
R_tilde = D_tilde %*% C_tilde %*% D_tilde
```

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

Stichprobenkovarianzmatrix und Stichprobenkorrelationsmatrix

C

	angriffslustig	penibel	streitbar	kämpferisch	grimmig	gründlich	akkurat	gewissenhaft	kleinlich	übergenau	herausfordernd	hitzig
angriffslustig	+5.9	−0.9	+4.8	+3.2	+2.3	−0.8	−1.4	−0.4	+0.3	−1.1	+3.2	+2.1
penibel	−0.9	+7.1	+0.2	+0.3	+0.8	+5.3	+5.1	+0.9	+6.1	+6.4	+0.0	+0.8
streitbar	+4.8	+0.2	+4.4	+2.7	+2.2	+0.1	−0.5	−0.3	+1.4	+0.3	+2.9	+2.1
kämpferisch	+3.2	+0.3	+2.7	+3.0	+1.4	+0.2	−0.2	−0.0	+0.9	+0.2	+2.6	+1.4
grimmig	+2.3	+0.8	+2.2	+1.4	+2.3	+0.7	+0.4	+0.1	+1.0	+0.8	+2.1	+1.9
gründlich	−0.8	+5.3	+0.1	+0.2	+0.7	+4.6	+4.2	+1.4	+5.0	+5.2	+0.4	+0.7
akkurat	−1.4	+5.1	−0.5	−0.2	+0.4	+4.2	+4.3	+1.4	+4.3	+4.6	−0.2	+0.4
gewissenhaft	−0.4	+0.9	−0.3	−0.0	+0.1	+1.4	+1.4	+2.8	+1.1	+1.1	+0.2	+0.1
kleinlich	+0.3	+6.1	+1.4	+0.9	+1.0	+5.0	+4.3	+1.1	+8.0	+7.2	+1.4	+1.1
übergenau	−1.1	+6.4	+0.3	+0.2	+0.8	+5.2	+4.6	+1.1	+7.2	+7.8	+0.3	+0.9
herausfordernd	+3.2	+0.0	+2.9	+2.6	+2.1	+0.4	−0.2	+0.2	+1.4	+0.3	+4.8	+2.2
hitzig	+2.1	+0.8	+2.1	+1.4	+1.9	+0.7	+0.4	+0.1	+1.1	+0.9	+2.2	+2.4

R

	angriffslustig	penibel	streitbar	kämpferisch	grimmig	gründlich	akkurat	gewissenhaft	kleinlich	übergenau	herausfordernd	hitzig
angriffslustig	+1.0	−0.1	+0.9	+0.8	+0.6	−0.2	−0.3	−0.1	+0.0	−0.2	+0.6	+0.5
penibel	−0.1	+1.0	+0.0	+0.1	+0.2	+0.9	+0.9	+0.2	+0.8	+0.9	+0.0	+0.2
streitbar	+0.9	+0.0	+1.0	+0.7	+0.7	+0.0	−0.1	−0.1	+0.2	+0.1	+0.6	+0.6
kämpferisch	+0.8	+0.1	+0.7	+1.0	+0.6	+0.1	−0.0	−0.0	+0.2	+0.0	+0.7	+0.5
grimmig	+0.6	+0.2	+0.7	+0.6	+1.0	+0.2	+0.1	+0.0	+0.2	+0.2	+0.6	+0.8
gründlich	−0.2	+0.9	+0.0	+0.1	+0.2	+1.0	+1.0	+0.4	+0.8	+0.9	+0.1	+0.2
akkurat	−0.3	+0.9	−0.1	−0.0	+0.1	+1.0	+1.0	+0.4	+0.7	+0.8	−0.0	+0.1
gewissenhaft	−0.1	+0.2	−0.1	−0.0	+0.0	+0.4	+0.4	+1.0	+0.2	+0.2	+0.1	+0.1
kleinlich	+0.0	+0.8	+0.2	+0.2	+0.2	+0.8	+0.7	+0.2	+1.0	+0.9	+0.2	+0.2
übergenau	−0.2	+0.9	+0.1	+0.0	+0.2	+0.9	+0.8	+0.2	+0.9	+1.0	+0.0	+0.2
herausfordernd	+0.6	+0.0	+0.6	+0.7	+0.6	+0.1	−0.0	+0.1	+0.2	+0.0	+1.0	+0.6
hitzig	+0.5	+0.2	+0.6	+0.5	+0.8	+0.2	+0.1	+0.1	+0.2	+0.2	+0.6	+1.0

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

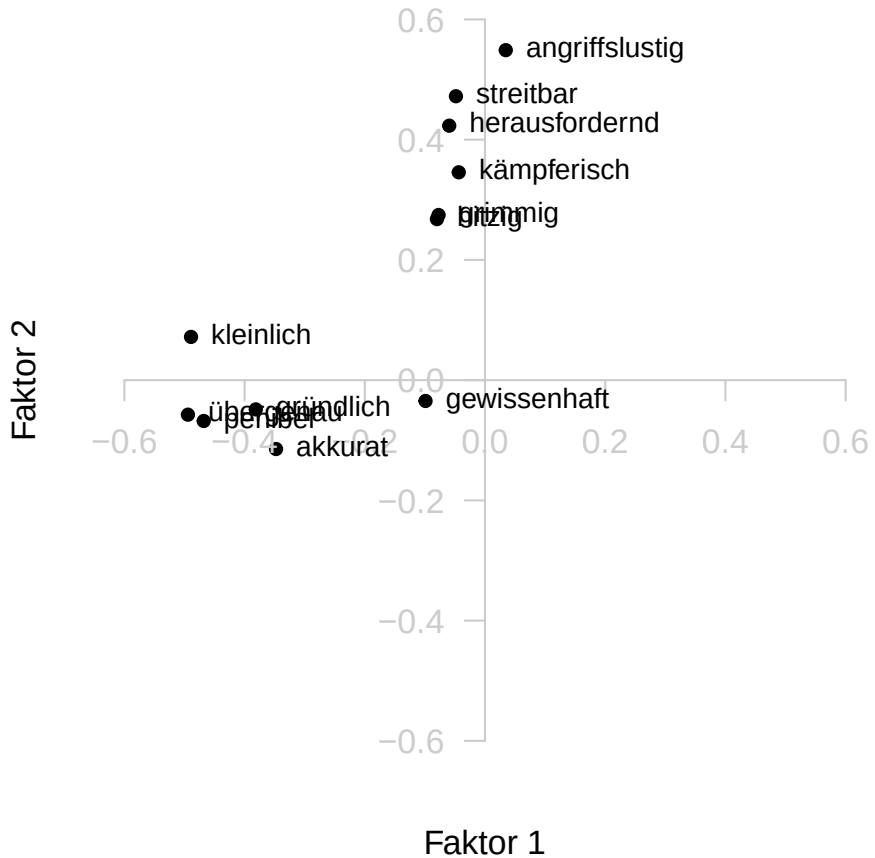
Ladungen (= Komponenten) der Faktoren (= Eigenvektoren) mit den höchsten beiden Eigenwerten

```
# "Ladungen" der "Faktoren" mit den höchsten beiden Eigenwerten
L = Q[, 1:2]
```

	Faktor 1	Faktor 2
angriffslustig	0.035	0.549
penibel	-0.468	-0.068
streitbar	-0.048	0.472
kämpferisch	-0.044	0.346
grimmig	-0.077	0.275
gründlich	-0.381	-0.049
akkurat	-0.348	-0.115
gewissenhaft	-0.099	-0.035
kleinlich	-0.489	0.072
übergenau	-0.494	-0.057
herausfordernd	-0.060	0.423
hitzig	-0.080	0.268

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

Ladungen (Komponenten) der Faktoren (Eigenvektoren) mit den höchsten beiden Eigenwerten

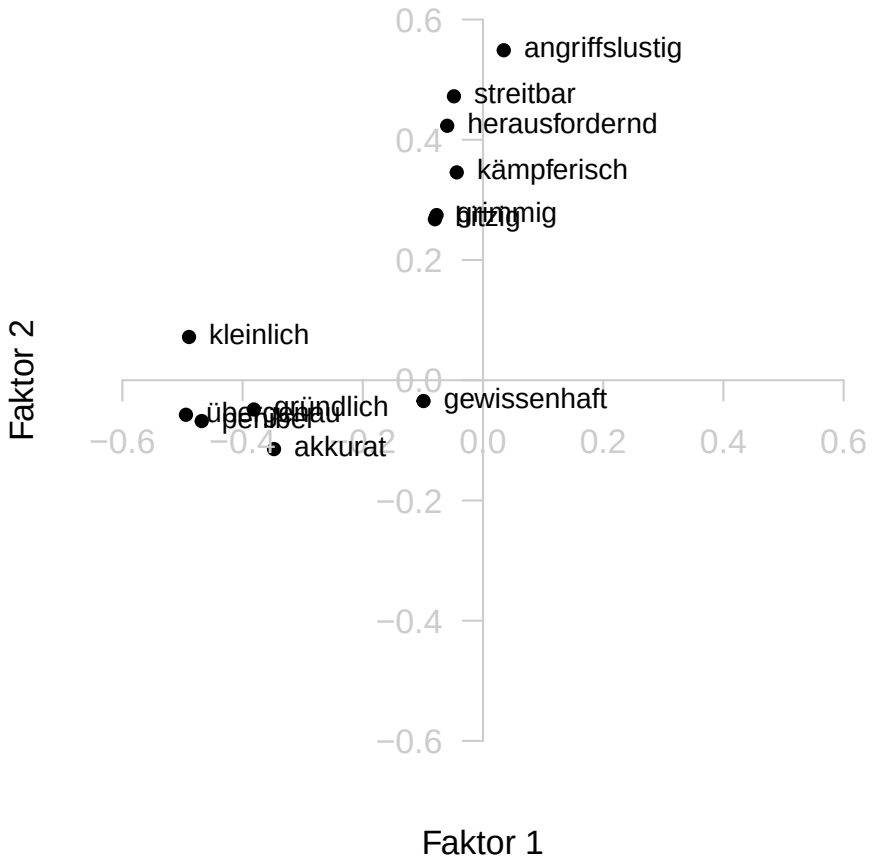


Angriffslustig, streitbar, herausfordernd, kämpferisch, hitzig ≈ Hohe Werte Faktor 1, Niedrige Werte Faktor 2

Kleinlich, akkurat, übergenau, gründlich, gewissenhaft, penibel ≈ Niedrige Werte Faktor 1, Hohe Werte Faktor 2

Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

Ladungen (Komponenten) der Faktoren (Eigenvektoren) mit den höchsten beiden Eigenwerten

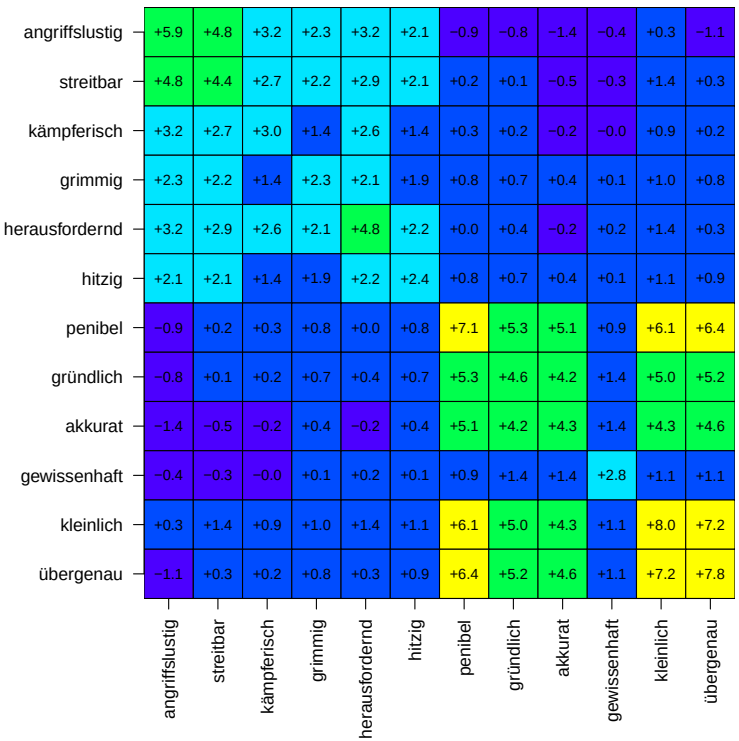


Faktor 1 = Aggressivität und Faktor 2 = Perfektionismus als Persönlichkeitseigenschaften

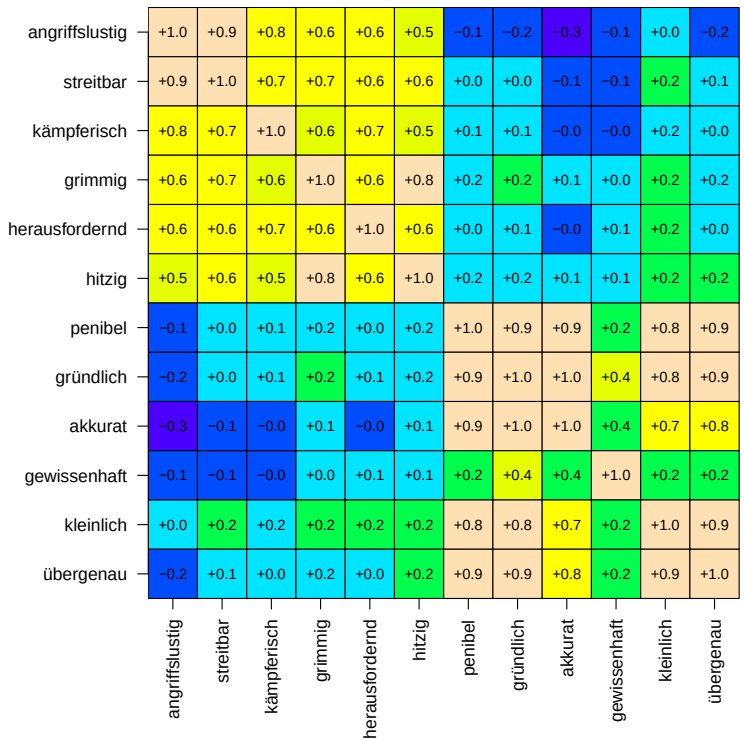
Sedimentationshypothese (Lexikalischer Ansatz) der Persönlichkeitspsychologie

. . . es ginge auch direkter . . .

C



R



Vektorkoordinatentransformation

Definition

Singulärwertzerlegung

Datenkompression

Exploratorische Faktorenanalyse

Selbstkontrollfragen

Selbstkontrollfragen

1. Erläutern Sie den Begriff “Featureselektion”.
2. Definieren Sie den Begriff Orthogonalprojektion.
3. Geben Sie das Theorem zu Vektorkoordinaten bezüglich einer Orthogonalbasis wieder.
4. Geben Sie das Vektorkoordinatentransformationstheorem wieder.
5. Erläutern Sie das Vektorkoordinatentransformationstheorem.
6. Geben Sie die Definition einer Hauptkomponentenanalyse wieder.
7. Geben Sie das Theorem zur Hauptkomponentenanalyse wieder.
8. Geben Sie die Definition der Hauptkomponentenanalyse eines Datensatzes wieder.
9. Geben Sie das Theorem zur Hauptkomponentenanalyse eines Datensatzes wieder.
10. Schreiben Sie R Code zur Implementation einer Hauptkomponentenanalyse durch Eigenanalyse.
11. Geben Sie die Definition einer Singulärwertzerlegung wieder.
12. Geben Sie das Theorem zum Zusammenhang von Singulärwertzerlegung und Eigenanalyse wieder.

Selbstkontrollfragen

13. Geben Sie das Theorem zur Datenhauptkomponentenanalyse durch Singulärwertzerlegung wieder.
14. Schreiben Sie R Code zur Implementation einer Hauptkomponentenanalyse durch Singulärwertzerlegung.
15. Erläutern Sie das Prinzip der Datenkompression durch Hauptkomponentenanalyse
16. Definieren Sie den Begriff des PCA-dimensionreduzierten Datensatzes.
17. Definieren Sie den Begriff des (PCA)-rekonstruierten Datensatzes.
18. Definieren Sie den Begriff des (PCA)-Rekonstruktionsfehlers.
19. Erläutern Sie die Idee eines Scree-Plots.
20. Erläutern Sie ein Beispiel zur Datendimensionreduktion in der Analyse von EEG Daten.
21. Erläutern Sie den Begriff "Exploratorische Faktorenanalyse".
22. Erläutern Sie die Idee der Sedimenationshypothese/des lexikalischen Ansatzes der Persönlichkeitspsychologie.
23. Erläutern Sie, wie man mithilfe einer Hauptkomponentenanalyse und eines Datensatzes von Einschätzungen einer ihnen bekannten Person durch Proband:innen hinsichtlich einer Menge von Adjektiven zur Identifikation von latenten Persönlichkeitseigenschaften gelangen kann.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(5) Faktorenanalyse

Überblick

Generative Perspektive zu konfirmatorischer und exploratorischer Faktoranalyse

- "Generativ" bedeutet hier probabilistisch und modell-basiert.

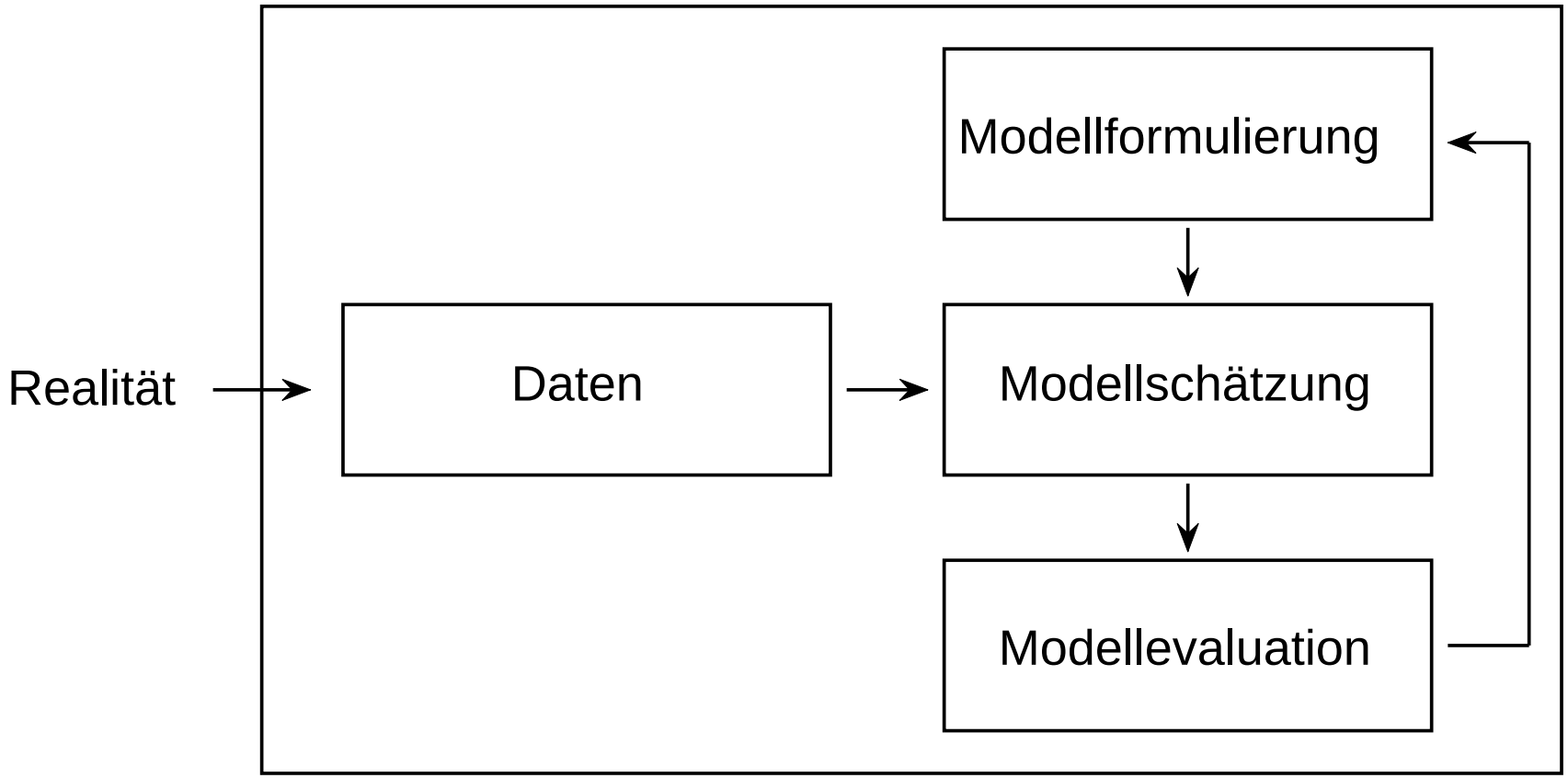
Einführung zum Expectation-Maximization (EM) Algorithmus.

- Generischer Algorithmus zur Schätzung von Parametern in Modellen mit latenten Variablen.
- Moderne Sichtweise von EM als Evidende Lower Bound Maximierung.
- Ein Schritt in Richtung eines Verständnisses von Variational Inference.

$\Rightarrow$ Integrative Perspektive von Inferenz und Lernen in Modellen mit latenten Variablen mit einer natürlichen Generalisierung zu kontemporären Modellen des maschinellen Lernens und der künstlicher Intelligenz (Variational Bayesian Filtering oder "Variational Autoencoders" für die Generation Deep Learning)

$\Rightarrow$ Ein Schritt zu einem Verständnis zeitgenössischer Theorien zur Funktionsweise des Gehirns (Free Energy Principle, Active Inference, Agent-based behavioral models)

Überblick


Realität
Daten
Modellformulierung
Modellschätzung
Modellevaluation

Literaturhinweise

Wir folgen der Darstellung Linearer Normalverteilungsmodelle in Roweis and Ghahramani (1999). Dempster, Laird, and Rubin (1977) bietet eine inhaltsreiche Einführung zum EM Algorithmus. Die Anwendung des EM Algorithmus im Rahmen der Faktorenanalyse geht auf Rubin and Thayer (1982) zurück. Probabilistische Hauptkomponentenanalye wird in Tipping and Bishop (1999) und Roweis (1998) diskutiert und geht auf Arbeiten von Lawley (1953) zurück. Ursprünglich wurde die Hauptkomponentenanalyse von Pearson (1901) vorgeschlagen und insbesondere von Hotelling (1933) verfeinert. Das Anwendungsbeispiel aus dem Gebiet der kognitiven Fähigkeitsforschung geht auf das Beispiel zur konfirmatorischen Faktorenanalyse in Rosseel (2012) basierend auf Joreskog (1969) und Holzinger and Swineford (1939).

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Anwendungsbeispiel

Selbstkontrollfragen

Appendix

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Anwendungsbeispiel

Selbstkontrollfragen

Appendix

Definition (Multivariate Normalverteilung)

X sei ein n-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^n$ und WDF

$$p : \mathbb{R}^n \to \mathbb{R}_{>0},\ x \mapsto p(x) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \quad (1)$$

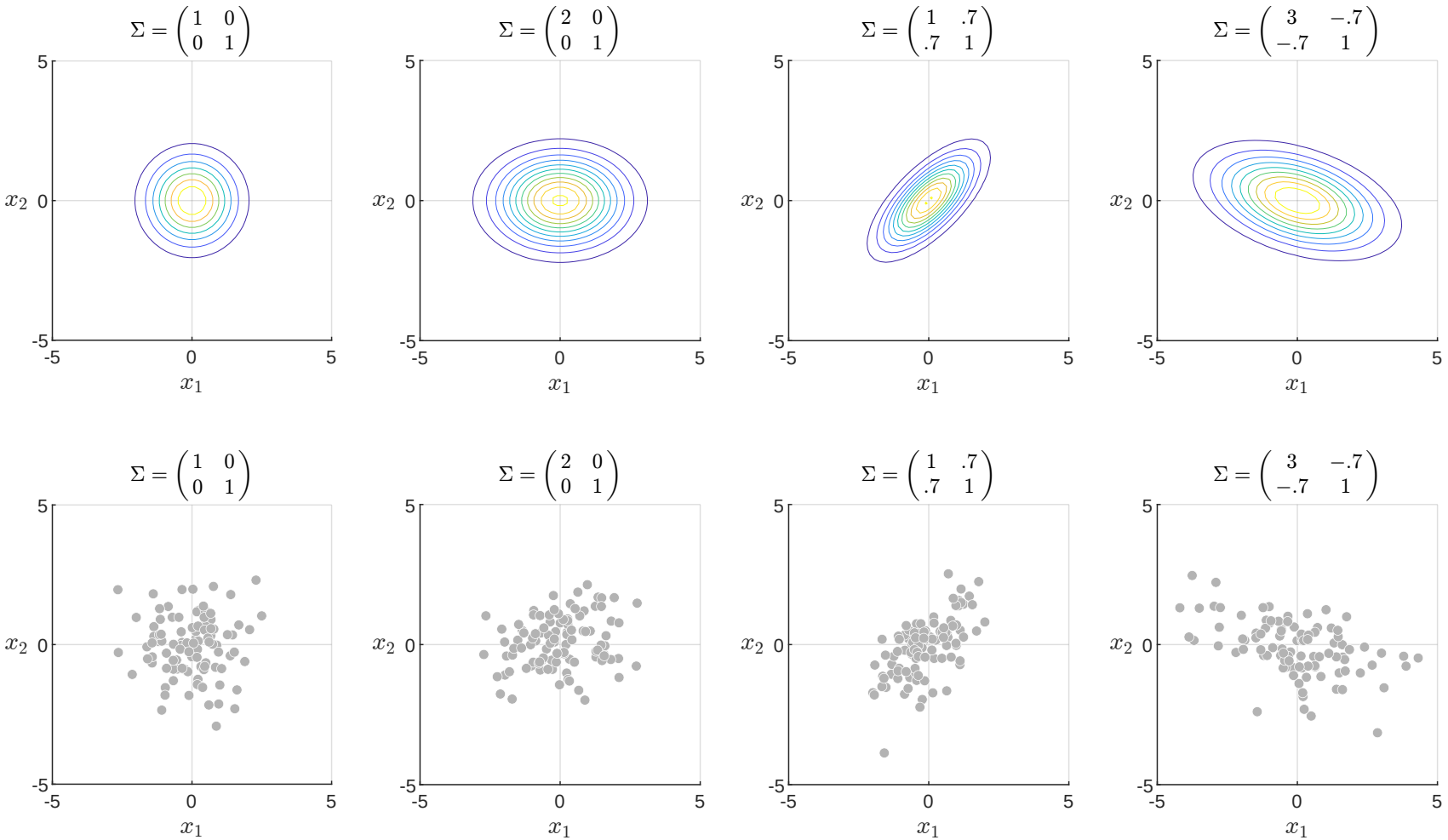
Dann sagen wird, dass X einer *multivariaten (oder n-dimensionalen) Normalverteilung* mit *Erwartungswertparameter* $\mu \in \mathbb{R}^n$ und positive-definitem *Kovarianzmatrixparameter* $\Sigma \in \mathbb{R}^{n \times n}$ unterliegt und nennen X einen *(multivariat) normalverteilten Zufallsvektor*. Wir kürzen dies mit $X \sim N(\mu, \Sigma)$ ab. Die WDF eines multivariat normalverteilten Zufallsvektors bezeichnen wir mit

$$N(x; \mu, \Sigma) := (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \quad (2)$$

Bemerkungen

- Der Parameter $\mu \in \mathbb{R}^n$ entspricht dem Wert höchster Wahrscheinlichkeitsdichte
- Die Diagonalelemente von Σ spezifizieren die Breite der WDF bezüglich $X_1, \ldots, X_n$.
- Das i, jte Element von Σ spezifiziert die Kovarianz on X_i und X_j.
- Der Term $(2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}}$ ist die Normalsieriungskonstante für den Exponentialfunktionsterm.

Zweidimensionale Normalverteilungen



Definition (Lineares Normalverteilungsmodell)

X sei ein kontinuierlicher nicht-beobachtbarer k-dimensionaler Zufallsvektor und Y sei ein kontinuierlicher beobachbarer m-dimensionaler Zufallsvektor. $B \in \mathbb{R}^{m \times k}$ sei eine Matrix und $R \in \mathbb{R}^{m \times m}$ sei eine positiv-definite Matrix. Dann heißt ein probabilistisches Modell mit WDF

$$p(x, y) = p(y|x)p(x), \tag{3}$$

wobei

$$p(y|x) := N(y; Bx, R) \text{ und } p(x) := N(x; 0_k, I_k) \tag{4}$$

gilt, ein *lineares Normalverteilungsmodell (LNM)*. Die Parametermenge eines LNMs ist $\theta := \{B, R\}$ und wir schreiben die WDFen von LNMen allgemein als

$$p_\theta(x, y) := N(y; Bx, R)N(x; 0_k, I_k). \tag{5}$$

Bemerkungen

- Der Zufallsvektor X heißt auch *Zustandsvektor* oder *latenter Vektor*
- Der Zufallsvektor Y heißt auch *Datenvektor*.
- In hierarchischer Form kann ein LNM geschrieben werden als

$$\begin{aligned} X &= \xi & \xi &\sim N(0_k, I_k) \\ Y &= BX + \eta & \eta &\sim N(0_m, R). \end{aligned} \tag{6}$$

- ξ heißt dabei *Zustandsrauschen*, η heißt dabei *Beobachtungsrauschen/fehler*.
- Samplen eines LNMs resultiert in Realisierungen $(x^{(i)}, y^{(i)})$ mit $i = 1, ..., n$.
- Die $x^{(i)} \in \mathbb{R}^k$ modellieren nicht beobachtbare/latente/virtuelle Daten.
- Die $y^{(i)} \in \mathbb{R}^m$ modellieren beobachtbare Daten.
- LNMe sind spezielle lineare normalverteilte Zustandsraummodelle.

Theorem (LNM Datenverteilung)

Die Datenverteilung des LNMs

$$p_\theta(x, y) = N(y; Bx, R)N(x; 0_k, I_k) \tag{7}$$

ist gegeben durch

$$p_\theta(y) = N(y; 0_m, BB^T + R) \tag{8}$$

und wird auch als *marginale Datenverteilung* oder *marginale Likelihood* bezeichnet.

Beweis

Wir halten zunächst fest, dass mit dem Theorem zu Gemeinsamen Normalverteilungen aus (3) Wahrscheinlichkeitstheorie direkt folgt, dass die WDF der gemeinsamen Verteilung von X und Y gegeben ist durch

$$p_\theta(x, y) = N\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} 0_k \\ 0_m \end{pmatrix}, \begin{pmatrix} I_k & B^T \\ B & BB^T + R \end{pmatrix}\right). \tag{9}$$

Mit dem Theorem zu Marginalen Normalverteilungen aus (3) Wahrscheinlichkeitstheorie folgt dann aber soforft, dass die WDF der marginalen Verteilung von Y durch

$$p_\theta(y) = N\left(y; 0_m, BB^T + R\right). \tag{10}$$

gegeben ist. □

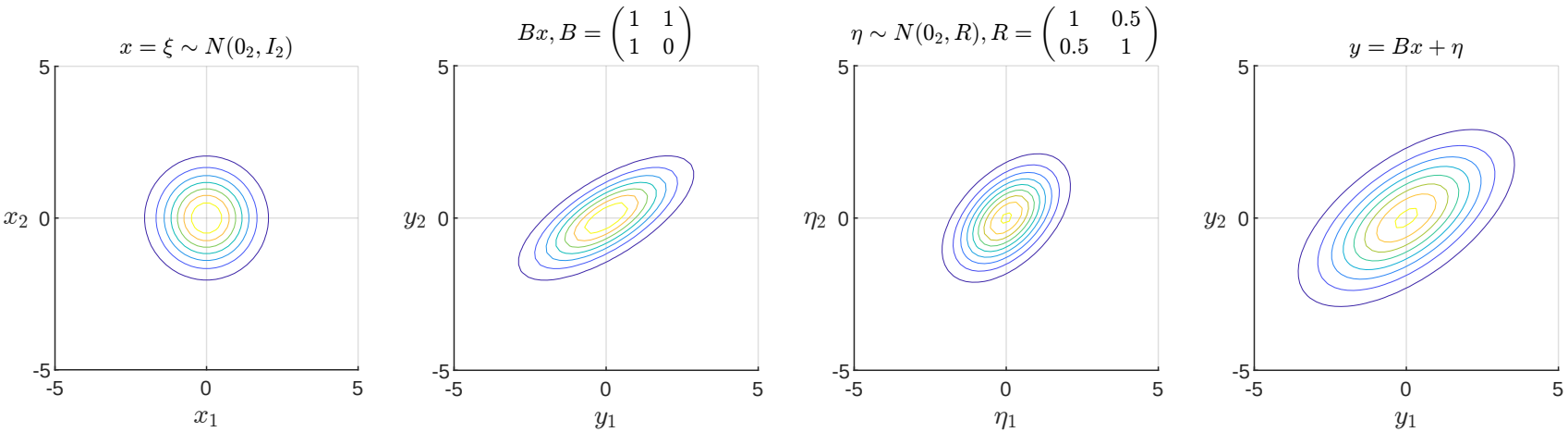
Bemerkungen

- LNMs modellieren zentrierte multivariat normalverteilte Datensätze mit Kovarianzmatrix.

$$\mathbb{C}(Y) = BB^T + R. \tag{11}$$

- "Modellieren" bedeutet hier insbesondere, die Datenkovarianzmatrix $\mathbb{C}(Y)$ zu erklären.
- Die Form von $\mathbb{C}(Y)$ resultiert dabei aus der Transformation einer latenten Normalverteilung.
- Die Matrizen B und R generieren/erklären $\mathbb{C}(Y)$ mechanistisch.
- B und R ermöglichen so oft eine kondensiertere Erklärung als $\mathbb{C}(Y)$ *per se*.
- Verschiedene LNMe haben unterschiedliche Potentiale, Datenkovarianzmatrizen zu erklären.

Grudlagen



- Der latente Zufallsvektor X ist in $\mathbb{R}^k$ (hier $k = 2$) spärisch verteilt.
- Durch B wird diese Sphäre gedehnt, rotiert, und nach $\mathbb{R}^m$ (hier $m = 2$) transformiert.
- Bei $m < p$ sieht die Sphäre von X zum Beispiel wie ein Pfannkuchen aus.
- Dieser Pfannkuchen wird dann noch mit der Kovarianz von des Beobachtungsrauschens η konvolviert.
- Es mag helfen, sich diese Konvolution (Faltung) als “Addition von Rauschen” vorzustellen.

Spezielle lineare Normalverteilungsmodelle

- Das Ziel der Datenmodellierung mit LNMe ist die Erklärung der Datenkovarianmatrixstruktur.
- Die Datenkovarianmatrixstruktur kann durch Wahl von B und R erklärt werden
- Spezielle LNMe entsprechen spezifischen Randbedingungen für R.

  $\Rightarrow$ In der konfirmatorischen Faktorenanalyse wird R als Diagonalmatrix vorausgesetzt.

  $\Rightarrow$ In der probabilistischen PCA wird R als sphärisch vorausgesetzt.

  $\Rightarrow$ In der exploratorischen Faktorenanalyse (PCA) wird $R = 0_{mm}$ vorausgesetzt.

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Anwendungsbeispiel

Selbstkontrollfragen

Appendix

Definition (Verteilungen von LNM Datensätzen)

$$Y := \begin{pmatrix} y^{(1)} & \dots & y^{(n)} \end{pmatrix} \in \mathbb{R}^{m \times n} \text{ und } X := \begin{pmatrix} x^{(1)} & \dots & x^{(n)} \end{pmatrix} \in \mathbb{R}^{k \times n} \tag{12}$$

seien ein beobachteter Datensatz (Realisierungen des beobachtbaren Zufallsvektors) und der assoziierte unbeobachtete Datensatz (Realisierungen des latenten Zufallsvektors). Unter der Annahme unabhängier und identischer Verteilung der gemeinsamen Realisationen $(x^{(i)}, y^{(i)})$ für $i = 1, \dots, n$ ist die WDF der gemeinsamen Verteilung eines LNM Datensatz durch

$$p_\theta(X, Y) = \prod_{i=1}^{n} p_\theta \left(x^{(i)}, y^{(i)} \right) = \prod_{i=1}^{n} N \left(y^{(i)}; Bx^{(i)}, R \right) N \left(x^{(i)}; 0_k, I_k \right) \tag{13}$$

und die marginale WDF des beobachtbaren Datensatzes gegeben durch

$$p_\theta(Y) = \prod_{i=1}^{n} N \left(y^{(i)}; 0_m, BB^T + R \right). \tag{14}$$

Bemerkungen

- X **und** Y **bezeichnen ab jetzt keine Zufallsvektoren mehr.**
- $X \in \mathbb{R}^{k \times n}$ **und** $Y \in \mathbb{R}^{m \times n}$ **bezeichnen ab jetzt Matrizen.**

Inferenz

Was ist die Verteilung der latenten Zufallsvektoren und was sind ihre wahrscheinlichsten Werte für feste Werte der LNM Parameter $\theta := \{B, R\}$?

$\Rightarrow$ Die Antwort gibt das Theorem zu bedingten multivariaten Normalverteilungen.

Lernen

Welche Parameterwerte maximieren die Marginal-Likelihood Funktion

$$L : \Theta \to \mathbb{R}_{\geq 0}, \theta \mapsto L(\theta) := p_\theta(Y) = \int p_\theta(X, Y)\, dX, \tag{15}$$

oder, äquivalent, die Log Marginal-Likelihood Funktion

$$\ell : \Theta \to \mathbb{R}, \theta \mapsto \ell(\theta) := \ln p_\theta(Y) = \ln \int p_\theta(X, Y)\, dX \ ? \tag{16}$$

für einen festen beobachteten Datensatz $Y \in \mathbb{R}^{m \times n}$?

$\Rightarrow$ Die Antwort gibt der Expectation-Maximization Algorithmus.

Theorem (LNM Inferenz)

Es sei

$$p_\theta(x, y) = N(y; Bx, R)N(x; 0, I_k),$$

ein LNM. Dann ist die WDF der bedingten Verteilung des latenten Zufallsvektors gegeben durch

$$p_\theta(x|y) = N\left(x; B^T(BB^T + R)^{-1}y, I_k - B^T(BB^T + R)^{-1}B\right). \tag{17}$$

Bemerkungen

- Die bedingte Verteilung des latenten Zufallsvektors ist eine Normalverteilung.
- Der wahrscheinlichste Wert des latenten ZVs gegeben eine Beobachtung des beobachtbaren ZVs ist also

$$\hat{x} := \mu_{x|y} = B^T(BB^T + R)^{-1}y. \tag{18}$$

- Die mit diesem Wert assoziierte Unsicherheit ist $\Sigma_{x|y} := I_k - B^T(BB^T + R)^{-1}B$.

Beweis

Wir hatten oben bereits gesehen, dass die WDF der gemeinsamen Verteilung von latentem und beobachtbarem Zufallsvektor gegeben ist durch

$$p_\theta(x,y) = N\left(\begin{pmatrix} x \\ y \end{pmatrix}; \begin{pmatrix} 0_k \\ 0_m \end{pmatrix}, \begin{pmatrix} I_k & B^T \\ B & BB^T + R \end{pmatrix}\right) \tag{19}$$

Mit dem Theorem zu Bedingten Normalverteilungen aus Einheit (3) Wahrscheinlichkeitstheorie gilt dann mit Identifikation von

$$\mu_x := 0_k, \mu_y := 0_m, \Sigma_{xx} := I_k, \Sigma_{xy} := B^T, \Sigma_{yx} := B, \text{ und } \Sigma_{yy} := BB^T + R, \tag{20}$$

dass

$$p_\theta(x|y) = N\left(x; \mu_{x|y}, \Sigma_{x|y}\right), \tag{21}$$

wobei

$$\mu_{x|y} = \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y) = B^T(BB^T + R)^{-1}y, \tag{22}$$

und wobei

$$\Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} = I_k - B^T(BB^T + R)^{-1}B. \tag{23}$$

ist. □

Theorem (LNM Datensatzinferenz)

$p_\theta(X, Y)$ sei die gemeinsame Verteilung eines LNM Datenatzes. Dann ist die WDF der bedingten Vertilung von X gegeben Y gegeben durch

$$p_\theta(X|Y) = \prod_{i=1}^{n} N\left(x^{(i)}; B^T(BB^T + R)^{-1}y^{(i)}, I_k - B^T(BB^T + R)^{-1}B\right). \quad (24)$$

Beweis

Mit den Ausdrücken für die WDFen der gemeinsamen und marginalen LNM Datensatzverteilungen gilt

$$p_\theta(X|Y) = \frac{p_\theta(X, Y)}{p_\theta(Y)} = \frac{\prod_{i=1}^{n} p_\theta(x^{(i)}, y^{(i)})}{\prod_{i=1}^{n} p_\theta\left(y^{(i)}\right)} = \prod_{i=1}^{n} \frac{p_\theta(x^{(i)}, y^{(i)})}{p_\theta(y^{(i)})} = \prod_{i=1}^{n} p_\theta\left(x^{(i)}|y^{(i)}\right).$$

Das Theorem folgt dann direkt mit dem Theorem zur LNM Inferenz.

□

Theorem (Evidence Lower Bound)

Für einen Datensatz $Y \in \mathbb{R}^{m \times n}$ sei $\ln p_\theta(Y)$ die WDF der Log Marginal-Likelihood Verteilung eines LNMs. Dann gilt für jede WDF $q(X)$, dass

$$\ln p_\theta(Y) \geq \int q(X) \ln p_\theta(X, Y)\, dX - \int q(X) \ln q(X)\, dX =: \text{ELBO}(q(X), \theta).$$

$\text{ELBO}(q(X), \theta)$ heißt *Evidence Lower Bound*.

Beweis

Mit der Jensenschen Ungleichung (Appendix) gilt

$$\ln p_\theta(Y) := \ln \int p_\theta(X, Y)\, dX = \ln \int q(X) \left(\frac{p_\theta(X, Y)}{q(X)} \right) dX \geq \int q(X) \ln \left(\frac{p_\theta(X, Y)}{q(X)} \right) dX.$$

Damit aber folgt

$$\begin{aligned} \ln p_\theta(Y) \geq \int q(X) \ln \left(\frac{p_\theta(X, Y)}{q(X)} \right) dX &= \int q(X) \left(\ln p_\theta(X, Y) - \ln q(X) \right) dX \\ &= \int q(X) \ln p_\theta(X, Y)\, dX - \int q(X) \ln q(X)\, dX. \end{aligned}$$

Bemerkungen

- Für einen festen Datensatz Y ist ELBO$(q(X), \theta)$ eine Funktion von $q(X)$ und θ.
- Die Bezeichnung Evidence Lower Bound geht auf den Term "Evidence" für $p_\theta(Y)$ zurück.
- In den kognitiven Neurowissenschaften ist die ELBO als "Freie Energie" bekannt.
- Die Signifikanz der ELBO geht weit über LNMe hinaus:
  - Die ELBO ist für die Variational Inference zentral.
  - Variational Inference ist für moderne Theorien zur Funktion des Gehirns zentral.
- Für Einführungen zur Variational Inference siehe zum Beispiel
  - Ostwald et al. (2014), Starke and Ostwald (2017), Blei and Smyth (2017).

Definition (Expectation-Maximization Algorithmus)

Die iterative koordinatenweise Maximierung der ELBO hinsichtlich $q(X)$ und θ heißt *Expectation-Maximization (EM) Algorithmus*. Der Algorithmus hat die allgemeine Form

EM Algorithmus

0. Initialisierung von $q^{(0)}(X)$ und $\theta^{(0)}$

Für $j = 1, 2, \ldots$

1. E Schritt: Setze $q^{(j)}(X) := \arg\max_{q(X)} \text{ELBO}\left(q(X), \theta^{(j-1)}\right)$
2. M Schritt: Setze $\theta^{(j)} := \arg\max_{\theta} \text{ELBO}\left(q^{(j)}(X), \theta\right)$

Nach Konvergenz, nutze $\hat{\theta} := \theta^{(j)}$ als Parameterschätzer.

Bemerkungen

- "Expectation Schritt" ist eine Fehlbezeichnung, es handelt sich auch um einen Maximization Schritt...
- ... allerdings ergibt die Bezeichnung im sogenannten "exakten" EM Algorithmus Sinn.

Theorem (Exakter Expectation-Maximization Algorithmus)

Das Setzen von

$$q^{(j)}(X) := p_{\theta^{(j-1)}}(X|Y) \text{ für alle } j = 1, 2, \ldots \tag{25}$$

im E Schritt des EM Algorithmus maximiert die ELBO hinsichtlich $q(X)$ und heißt *exakter E Schritt*. Der Algorithmus hat dann die folgende Form

Exakter EM Algorithmus

0. Initialisierung von $\theta^{(0)}$

Für $j = 1, 2, \ldots$

1. E Step $q^{(j)}(X) := p_{\theta^{(j-1)}}(X|Y)$
2. M Step $\theta^{(j)} := \arg\max_\theta \int p_{\theta^{(j-1)}}(X|Y) \ln p_\theta(X, Y)\, dX$

Nach Konvergenz, nutze $\hat{\theta} := \theta^{(j)}$ als Parameterschätzer.

Bemerkungen

- Für LNMe kann $p_{\theta^{(j-1)}}(X|Y)$ analytisch evaluiert werden $\Rightarrow$ Inferenz.
- Der M Schritt des exakten EM Algorithmus für LNM Parameterschätzung $\Rightarrow$ Lernen.

Beweis

Wir zeigen zunächst, dass die ELBO $q^{(j)}(X) := p_{\theta^{(j-1)}}(X|Y)$ den maximalen Wert $\ln p_{\theta^{(j-1)}}(Y)$ annimmt:

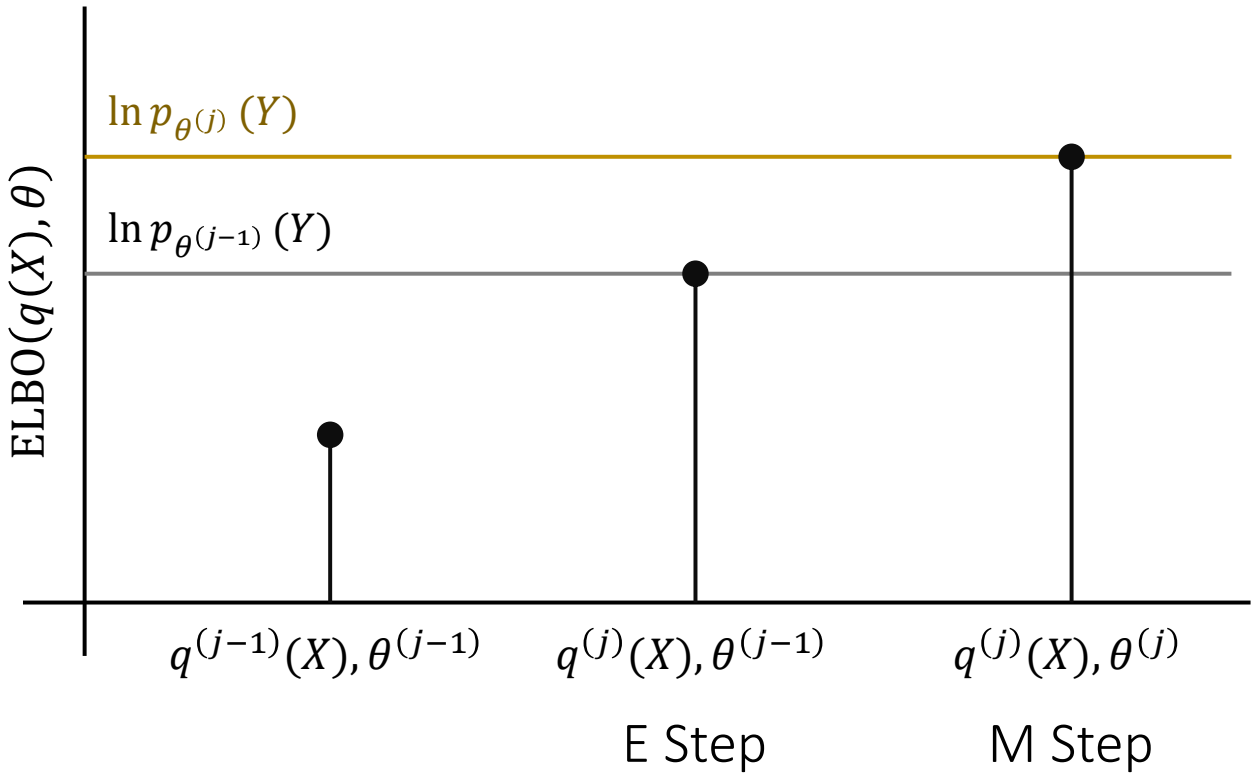
$$\begin{aligned}
\text{ELBO}(p_{\theta^{(j-1)}}(X|Y), \theta) &= \int p_{\theta^{(j-1)}}(X|Y) \ln \left(\frac{p_{\theta^{(j-1)}}(X, Y)}{p_{\theta^{(j-1)}}(X|Y)} \right) dX \\
&= \int p_{\theta^{(j-1)}}(X|Y) \ln \left(\frac{p_{\theta^{(j-1)}}(Y) p_{\theta^{(j-1)}}(X|Y)}{p_{\theta^{(j-1)}}(X|Y)} \right) dX \\
&= \int p_{\theta^{(j-1)}}(X|Y) \ln p_{\theta^{(j-1)}}(Y)\, dX \\
&= \ln p_{\theta^{(j-1)}}(Y) \int p_{\theta^{(j-1)}}(X|Y)\, dX \\
&= \ln p_{\theta^{(j-1)}}(Y).
\end{aligned}$$

Der M Schritt hat dementsprechend die Form

$$\begin{aligned}
\theta^{(j)} &= \arg\max_{\theta} \text{ELBO} \left(p_{\theta^{(j-1)}}(X|Y), \theta \right) \\
&= \arg\max_{\theta} \int p_{\theta^{(j-1)}}(X|Y) \ln p_{\theta}(X, Y)\, dX - \int p_{\theta^{(j-1)}}(X|Y) \ln p_{\theta^{(j-1)}}(X|Y)\, dX.
\end{aligned}$$

Der M Schritt des exakten EM Algorithmus folgt dann damit, dass der zweite Integralterm hier nicht von θ abhängt.

Visuelle Intuition



Weitere Bemerkungen

- Der M Step der $i$ten Iteration des exakten EM Algorithmus entspricht der Maximierung des Erwartungswertes der logarithmierten WDF der gemeinsamen Datenverteilung $p_\theta(X, Y)$ hinsichtlich θ, wobei der Erwartungswert hinsichtlihch der WDF der bedingten Datenverteilung von X gegeben Y basierend auf der Parameterschätzung $\theta^{(j-1)}$, die in der $(j-1)$ten Iteration des exakten EM Algorithmus gewonnen wurde.
- Überraschenderweise garantiert aufgrund der inherenten Logik des EM Algrithmus die Maximierung des Erwartungswertes

$$\mathbb{E}_{p_{\theta^{(j-1)}}(X|Y)}\left(\ln p_\theta(X, Y)\right) = \int p_{\theta^{(j-1)}}(X|Y) \ln p_\theta(X, Y)\, dX \tag{26}$$

  auch die Maximierung der tatsächlichen Funktion von Interesse, $\ln p_\theta(Y)$.
- Für konkrete Algorithmen und für spezifische LNMe muss obiger Erwartungswert analytisch als Funktion von $\theta^{(j-1)}$ ausgewertet werden und dann hinsichtlich θ entweder analytisch oder numerisch maximiert werden um die Parameterschätzung $\theta^{(j)}$. zu erhalten.

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Anwendungsbeispiel

Selbstkontrollfragen

Appendix

Definition (Faktorenanalysemodell)

Es sei

$$p_\theta(x, y) = N(y; Bx, R)N(x; 0, I_k) \tag{27}$$

ein LGM mit diagonaler Beobachtungsrauschen Kovarianzmatrix

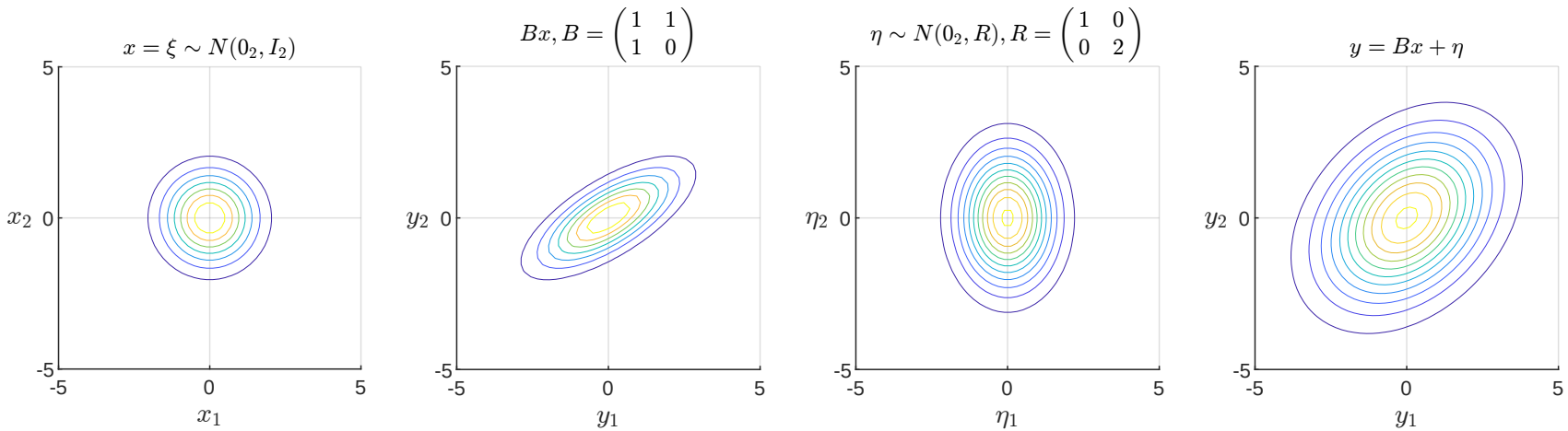
$$R = \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_m^2) \in \mathbb{R}^{m \times m}, \sigma_i^2 > 0,\ i = 1, ..., m. \tag{28}$$

Dann heißt $p_\theta(x, y)$ *Faktorenanalysemodell*.

Bemerkungen

- Das Modell heißt auch **Modell der konfirmatorischen Faktorenanalyse**.
- Die Komponenten (Zufallsvariablen) des latenten Zufallsvektors werden **Faktoren** genannt.
- Die Matrix B des Faktorenanalysemodells wird **Faktorenladungsmatrix** genannt.
- Die Diagonalelemente von R werden **Uniquenesses** genannt.

Visuelle Intuition



Bemerkungen

- Faktorenenalysemodelle erklären die Struktur Datenkovarianzmatrizen dadurch, dass
  - alle Korrelationen zwischen Datendimensionen durch B erklärt werden,
  - alle Varianzen dimensionsspezifisch durch R erklärt werden und
  - die Komponenten des beobachtbaren ZVs als bedingt unabhängig angenommen werden.
- Faktorenanalysemodelle behandeln Datenkovarianzen und Varianzen nicht identisch.
- Die Datenkovarianzstruktur wird als bedeutsam angesehen

  $\Leftrightarrow$ Das Beobachtungsrauschen wird als unkorreliert angenommen.
- Exploratorische und konfirmatorische FA entsprechen unterschiedlichen Bedingungen an R.
- Die Parameter des Modells können mit dem exakten EM Algorithmus geschätzt werden.

Theorem (Exakter EM Algorithmus für Faktorenanalysemodelle)

0. Initialisiere $B^{(0)}$ und $R^{(0)}$.

Für $i = 1, 2, \ldots$

1. E Schritt. Setze $\tilde{B} := B^{(j-1)}$ und $\tilde{R} := R^{(j-1)}$ und

$$q^{(j)}(X) := \prod_{j=1}^{n} N\left(x^{(i)}; \hat{x}^{(i)}, \hat{\Sigma}^{(i)}\right), \tag{29}$$

wobei

$$\hat{x}^{(i)} := \tilde{B}^T(\tilde{B}\tilde{B}^T + \tilde{R})^{-1} y^{(i)} \text{ und } \hat{\Sigma}^{(i)} := I_k - \tilde{B}^T(\tilde{B}\tilde{B}^T + \tilde{R})^{-1}\tilde{B}. \tag{30}$$

2. M Schritt. Setze

$$B^{(j)} := \sum_{i=1}^{n} y^{(i)} \hat{x}^{(i)^T} \left(\sum_{i=1}^{n} \hat{x}^{(i)} \hat{x}^{(i)^T} + \hat{\Sigma}^{(i)} \right)^{-1} \tag{31}$$

und

$$R^{(j)} := \frac{1}{n} \text{diag} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \sum_{i=1}^{n} y^{(i)} \hat{x}^{(i)^T} B^{(i)^T} \right). \tag{32}$$

Siehe Appendix für einen Beweis.

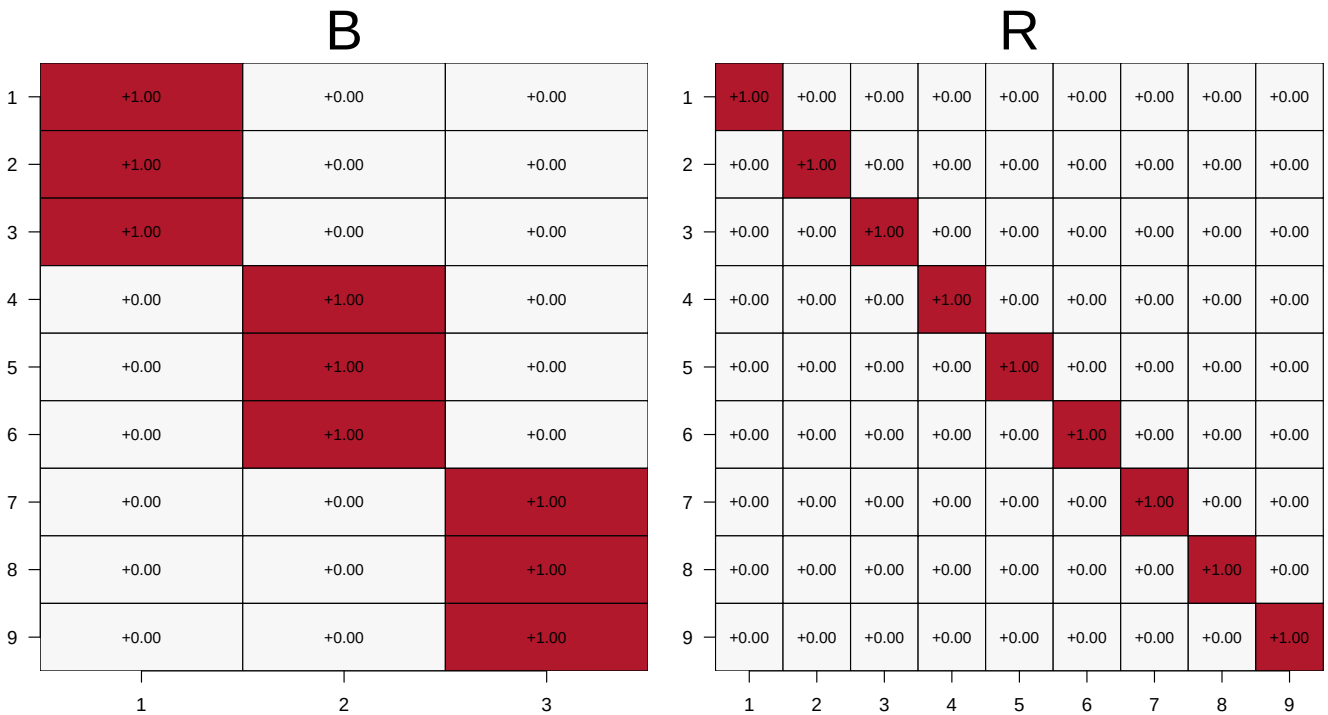
Simulationsbeispiel

Datengeneration

```
# mvrnorm() Paket
library(MASS)

# Paramterspezifikation
k      = 3                                  # Dimension des latenten Zufallsvektors
m      = 9                                  # Dimension des beobachtbaren Zufallsvektors
B      = matrix(c( 1,0,0,                   # Faktorenladungsmatrix
                   1,0,0,
                   1,0,0,
                   0,1,0,
                   0,1,0,
                   0,1,0,
                   0,0,1,
                   0,0,1,
                   0,0,1),
                 nrow  = m,
                 byrow = TRUE)
u      = rep(1,m)                           # Uniquenesses
R      = diag(u)                            # Beobachtungsrauschenkovarianmatrix
mu     = rep(0,k+m)                         # Erwartungswertparameter  p_\theta(x,y)
Sigma  = rbind(cbind(diag(k), t(B)),        # Kovarianzmatrixparameter p_\theta(x,y)
          cbind(B, B %*% t(B) + R))

# Datengeneration
n      = 1e3                                # Beobachtungsanzahl
XY     = t(mvrnorm(n,mu,Sigma))             # p_theta(x,y) sampling und z-score normalization
X      = XY[1:k,]                           # virtuelle Daten    X \in R^{k x n}
Y      = XY[(k+1):nrow(XY),]                # Beobachtete Daten  Y \in R^{m x n}
```

Simulationsbeispiel

Wahre, aber unbekannte, Parameter



Simulationsbeispiel

Marginale Kovarianzmatrix und marginale Korrelationsmatrix

Σ_y

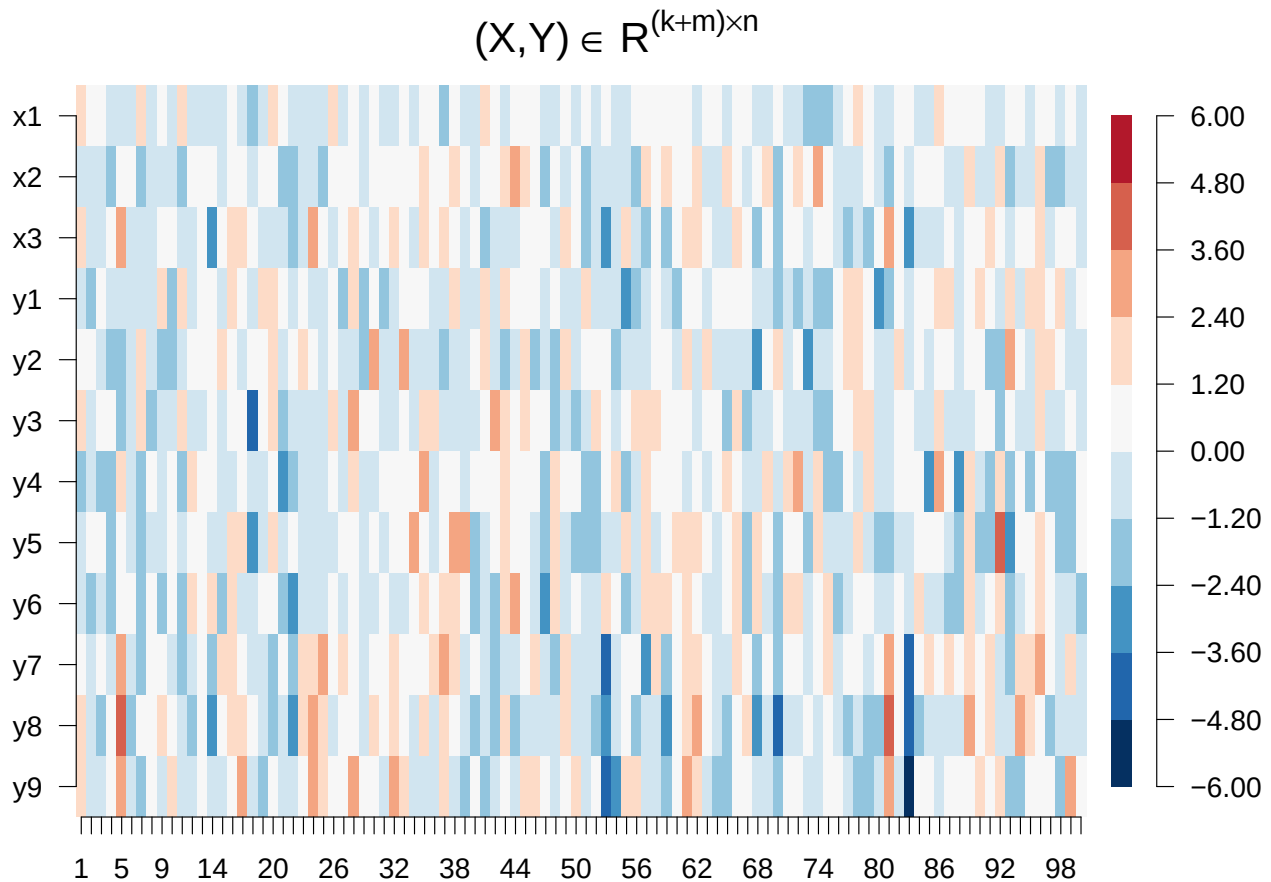
	1	2	3	4	5	6	7	8	9
1	+2.0	+1.0	+1.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
2	+1.0	+2.0	+1.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
3	+1.0	+1.0	+2.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
4	+0.0	+0.0	+0.0	+2.0	+1.0	+1.0	+0.0	+0.0	+0.0
5	+0.0	+0.0	+0.0	+1.0	+2.0	+1.0	+0.0	+0.0	+0.0
6	+0.0	+0.0	+0.0	+1.0	+1.0	+2.0	+0.0	+0.0	+0.0
7	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+2.0	+1.0	+1.0
8	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+1.0	+2.0	+1.0
9	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+1.0	+1.0	+2.0

ρ_y

	1	2	3	4	5	6	7	8	9
1	+1.0	+0.5	+0.5	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
2	+0.5	+1.0	+0.5	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
3	+0.5	+0.5	+1.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0
4	+0.0	+0.0	+0.0	+1.0	+0.5	+0.5	+0.0	+0.0	+0.0
5	+0.0	+0.0	+0.0	+0.5	+1.0	+0.5	+0.0	+0.0	+0.0
6	+0.0	+0.0	+0.0	+0.5	+0.5	+1.0	+0.0	+0.0	+0.0
7	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+1.0	+0.5	+0.5
8	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.5	+1.0	+0.5
9	+0.0	+0.0	+0.0	+0.0	+0.0	+0.0	+0.5	+0.5	+1.0

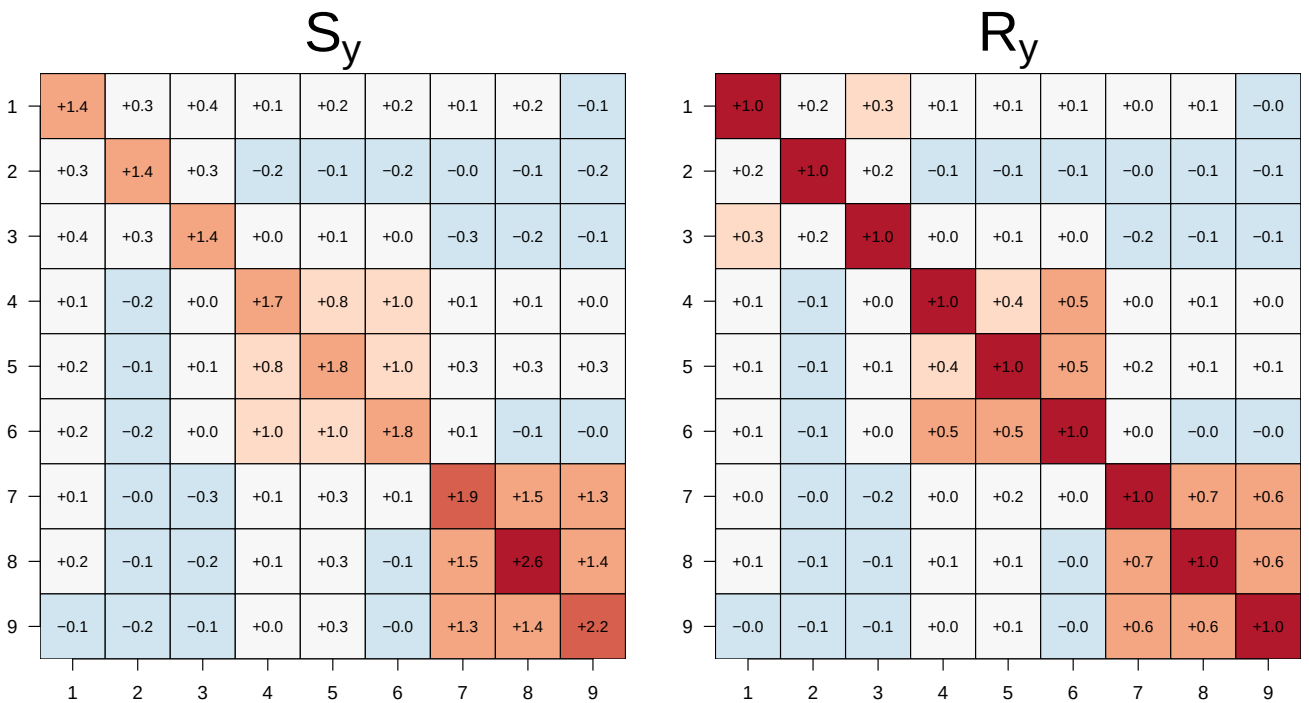
Simulationsbeispiel

Vollständiger Datensatz (n = 100)



Simulationsbeispiel

Stichprobenkovarianzmatrix und Stichprobenkorrelationsmatrix



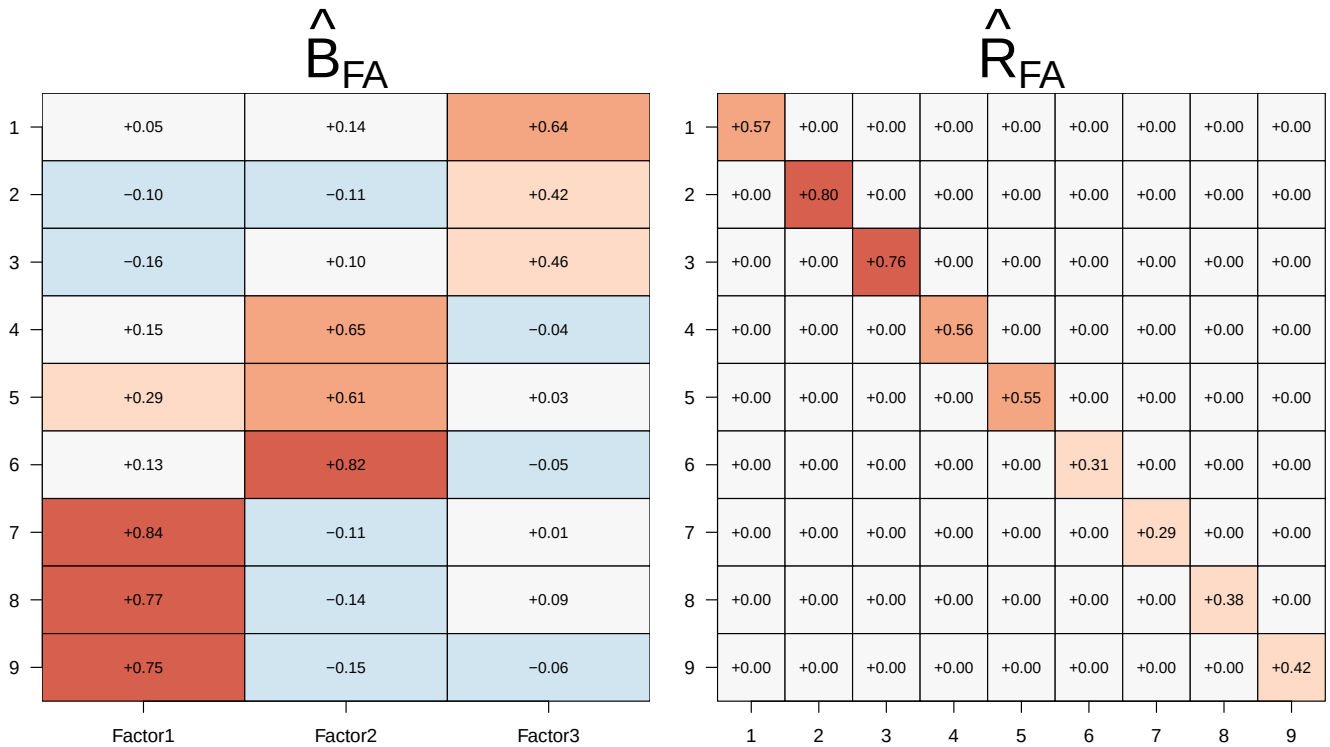
Simulationsbeispiel

Parameterschätzung mit R's `factanal()` Funktion

```
# Lawley & Maxwell (1971) Schätzverfahren für Faktorenanalyse
fa              = factanal(t(Y), factors = 3, rotation = "none")  # Parameterschätzung
B_hat_fa        = fa$loadings[1:9,1:3]                            # \hat{B}
R_hat_fa        = diag(fa$uniquenesses)                           # \hat{R}

# Parametschätzer-basierte Kovarianz und Korrelationsmatrix
Sigma_hat_y_fa  = B_hat_fa %*% t(B_hat_fa) + R_hat_fa
D_hat_y_fa      = diag(1/sqrt(diag(Sigma_hat_y_fa)))
rho_hat_y_fa    = D_hat_y_fa %*% Sigma_hat_y_fa %*% D_hat_y_fa
```

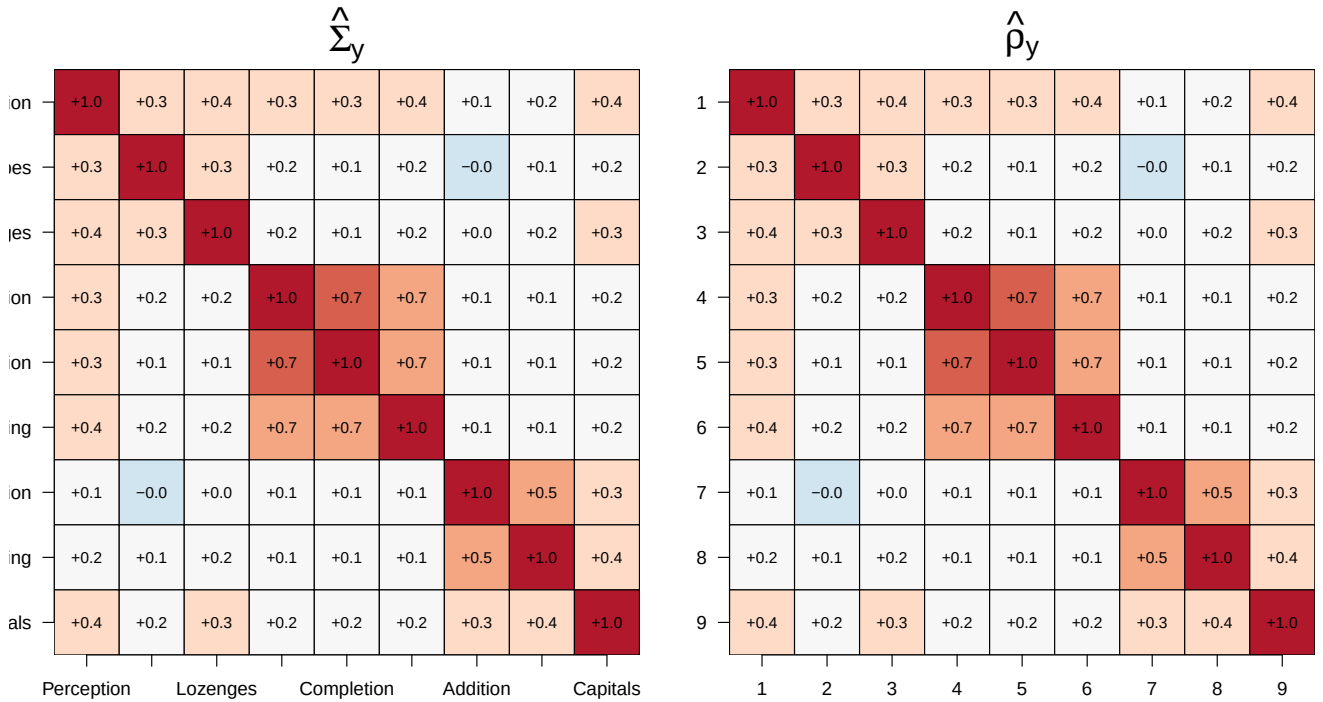
Simulationsbeispiel

`factanal()` basierte Parameterschätzer



Simulationsbeispiel

`factanal()` basierte marginale Kovarianmatrix und Korrelationsmatrix



Simulationsbeispiel

EM Algorithmus

```
library(matlib)                                    # Matrizenrechnungspaket
max_j = 2^2                                        # Maximale Anzahl Iterationen

# EM Initialisierung
B_j = matrix(runif(m*k), nrow = m)
R_j = diag(runif(0,m))

# Iterations
for(j in 1:max_j){

  # E Schritt
  X_hat_j      = t(B_j) %*% inv(B_j %*% t(B_j) + R) %*% Y
  Sigma_hat_j  = diag(k) - (t(B_j) %*% inv(B_j %*% t(B_j) + R) %*% B_j)

  # M Schritt
  yxT = Y %*% t(X_hat_j)
  xxT = X_hat_j %*% t(X_hat_j)
  yyT = Y %*% t(Y)
  B_j = yxT %*% inv(xxT + Sigma_hat_j)
  R_j = (1/n) * diag(diag((yyT - (yxT %*% t(B_j)))))
}
print(yyT)
print(Y %*% t(Y))
# Parameterschätzer
B_hat_em = B_j
R_hat_em = R_j

# Parametschätzer-basierte Kovarianz und Korrelationsmatrix
Sigma_hat_y_em  = B_hat_em %*% t(B_hat_em) + R_hat_em
D_hat_y_em      = diag(1/sqrt(diag(Sigma_hat_y_em)))
rho_hat_y_em    = D_hat_y_em %*% Sigma_hat_y_em %*% D_hat_y_em
```

Simulationsbeispiel

EM Algorithmus-basierte Parameterschätzer

$\hat{B}_{EM}$

−0.47	+0.99	+1.71	1
−0.55	+1.16	+1.77	2
−0.41	+0.95	+1.94	3
+1.14	+2.02	−0.30	4
+1.22	+1.89	−0.50	5
+1.36	+1.84	−0.47	6
+1.76	−1.21	+0.95	7
+1.62	−1.30	+0.98	8
+1.70	−1.44	+0.99	9
1	2	3	

$\hat{R}_{EM}$

+0.69	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
+0.00	+0.62	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
+0.00	+0.00	+0.63	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
+0.00	+0.00	+0.00	+0.61	+0.00	+0.00	+0.00	+0.00	+0.00
+0.00	+0.00	+0.00	+0.00	+0.64	+0.00	+0.00	+0.00	+0.00
+0.00	+0.00	+0.00	+0.00	+0.00	+0.61	+0.00	+0.00	+0.00
+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.67	+0.00	+0.00
+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.65	+0.00
+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.68
1	2	3	4	5	6	7	8	9

Simulationsbeispiel

EM Algorithmus basierte marginale Kovarianmatrix und Korrelationsmatrix

$\hat{\Sigma}_y^{EM}$

	1	2	3	4	5	6	7	8	9
1	+4.8	+4.4	+4.4	+0.9	+0.4	+0.4	−0.4	−0.4	−0.5
2	+4.4	+5.4	+4.8	+1.2	+0.6	+0.5	−0.7	−0.7	−0.9
3	+4.4	+4.8	+5.4	+0.9	+0.3	+0.3	−0.0	+0.0	−0.2
4	+0.9	+1.2	+0.9	+6.1	+5.4	+5.4	−0.7	−1.1	−1.3
5	+0.4	+0.6	+0.3	+5.4	+6.0	+5.4	−0.6	−1.0	−1.2
6	+0.4	+0.5	+0.3	+5.4	+5.4	+6.1	−0.3	−0.7	−0.8
7	−0.4	−0.7	−0.0	−0.7	−0.6	−0.3	+6.1	+5.4	+5.7
8	−0.4	−0.7	+0.0	−1.1	−1.0	−0.7	+5.4	+5.9	+5.6
9	−0.5	−0.9	−0.2	−1.3	−1.2	−0.8	+5.7	+5.6	+6.6

$\hat{\rho}_y^{EM}$

	1	2	3	4	5	6	7	8	9
1	+1.0	+0.9	+0.9	+0.2	+0.1	+0.1	−0.1	−0.1	−0.1
2	+0.9	+1.0	+0.9	+0.2	+0.1	+0.1	−0.1	−0.1	−0.1
3	+0.9	+0.9	+1.0	+0.2	+0.1	+0.0	−0.0	+0.0	−0.0
4	+0.2	+0.2	+0.2	+1.0	+0.9	+0.9	−0.1	−0.2	−0.2
5	+0.1	+0.1	+0.1	+0.9	+1.0	+0.9	−0.1	−0.2	−0.2
6	+0.1	+0.1	+0.0	+0.9	+0.9	+1.0	−0.0	−0.1	−0.1
7	−0.1	−0.1	−0.0	−0.1	−0.1	−0.0	+1.0	+0.9	+0.9
8	−0.1	−0.1	+0.0	−0.2	−0.2	−0.1	+0.9	+1.0	+0.9
9	−0.1	−0.1	−0.0	−0.2	−0.2	−0.1	+0.9	+0.9	+1.0

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Selbstkontrollfragen

Appendix

Definition (Modell der probabilistischen Hauptkomponentenanalyse)

Es sei

$$p_\theta(x, y) = N(y; Bx, R)N(x; 0_k, I_k) \tag{33}$$

ein LNM mit der sphärischen Beobachtungsrauschen Kovarianzmatrix

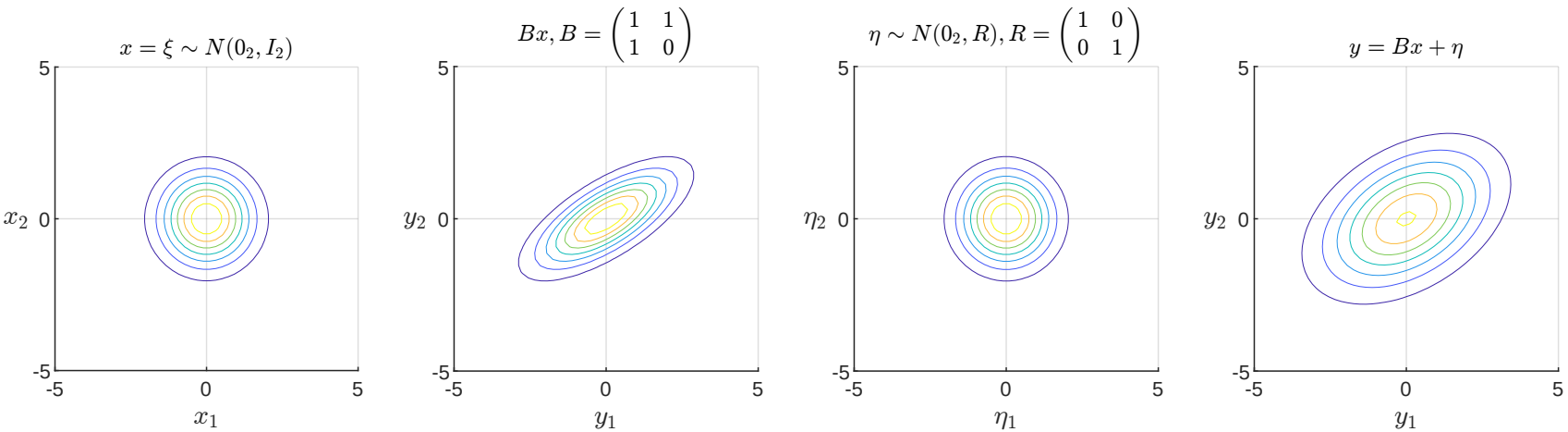
$$R := \sigma^2 I_m . \tag{34}$$

Dann heißt $p_\theta(x, y)$ *probabilistisches Hauptkomponentanalysemodell*.

Bemerkungen

- Die Matrix B bedingt die Beziehung zur klassischen Hauptkomponentenanalyse.
- σ^2 heißt *globales Rauschlevel*.
- B and σ^2 können durch den EM Algorithmus geschätzt werden.
- B and σ^2 können auch durch direkte Maximierung der marginalen Likelihood Funktion geschätzt werden.

Visuelle Intuition zur probabilistischen Hauptkomponentenanalyse



Theorem (Parameter der probabilistischen Hauptkomponentnanalyse)

Es sei

$$p_\theta(x, y) = N(y; Bx, \sigma^2 I_m) N(x; 0, I_k) \tag{35}$$

ein probabilitisches Hauptkomponentenanalysemodell und es sei

$$\mathbb{C}(Y) = Q \Lambda Q^T \tag{36}$$

die Orthogonalzerlegung der Kovarianzmatrix des beobachtbaren Zufallsvektors. Dann kann der Parameter B des probabilistischen Hauptkomponentnanalysemodells geschrieben werden als

$$B = Q \left(\Lambda - \sigma^2 I_m \right)^{1/2}. \tag{37}$$

Bemerkungen

- Wir haben $\mathbb{C}(Y) = Q \Lambda Q^T$ Hauptkomponentenanalyse von $\mathbb{C}(Y)$ genannt.
- Q besteht aus den Eigenvektoren von $\mathbb{C}(y)$, Λ aus den assoziierten Eigenwerten.
- Wir haben die Eigenvektoren von $\mathbb{C}(Y)$ die Hauptkomponenten von $\mathbb{C}(Y)$ genannt.
- Die Spalten von B sind die Hauptkomponenten gewichtet mit $(\Lambda - \sigma^2 I_m)^{1/2}$.

Beweis

Wir halten zunächst fest, dass die marginale Datenverteilung eines probabilistischen Hauptkomponentenanalysemodells gegeben ist durch

$$p_\theta(y) = N(y; 0_m, BB^T + \sigma^2 I_m) \text{ and thus } \mathbb{C}(Y) = BB^T + \sigma^2 I_m. \tag{38}$$

Einsetzen von $B = Q(\Lambda - \sigma^2 I_m)^{1/2}$ ergibt dann

$$\begin{aligned}
\mathbb{C}(Y) &= BB^T + \sigma^2 I_m \\
&= Q(\Lambda - \sigma^2 I_m)^{1/2}(Q(\Lambda - \sigma^2 I_m)^{1/2})^T + \sigma^2 I_m \\
&= Q(\Lambda - \sigma^2 I_m)^{1/2}((\Lambda - \sigma^2 I_m)^{1/2})^T Q^T + \sigma^2 I_m \\
&= Q(\Lambda - \sigma^2 I_m)^{1/2}(\Lambda - \sigma^2 I_m)^{1/2} Q^T + \sigma^2 I_m \\
&= Q(\Lambda - \sigma^2 I_m) Q^T + \sigma^2 I_m \\
&= (Q\Lambda - \sigma^2 Q I_m) Q^T + \sigma^2 I_m \\
&= Q\Lambda Q^T - \sigma^2 Q Q^T + \sigma^2 I_m \\
&= Q\Lambda Q^T - \sigma^2 I_m + \sigma^2 I_m \\
&= Q\Lambda Q^T.
\end{aligned} \tag{39}$$

Also ergibt sich die Äquivalenz

$$\mathbb{C}(y) = Q\Lambda Q^T \Leftrightarrow B = Q(\Lambda - \sigma^2 I_m)^{\frac{1}{2}}. \tag{40}$$

Theorem (Direkte Marginal-Maximum Likelihood Schätzung)

Es sei

$$p_\theta(x, y) = N(y; Bx, \sigma^2 I_m) N(x; 0, I_k) \tag{41}$$

ein probabilistisches Hauptkomponentenanalyse model und $Y \in \mathbb{R}^{m \times n}$ sei ein von diesem Modell generierter Datensatz. Weiterhin sei

$$C = \frac{1}{n} Y Y^T \text{ und } C = Q \Lambda Q^T \tag{42}$$

die unkorrigierte Stichprobenkovarianzmatrix und ihre Orhtogonazerlegung, respektive. Schließlich seien $Q_q \in \mathbb{R}^{m \times q}$ und $\Lambda_q \in \mathbb{R}^{q \times q}$ die Matrizen die durch die spaltenweise Konkatenation der höchsten Eigenwerte und ihrer assoziierten Eigenvektoren entstehen. Dann sind Maximum Marginal Likelihood Schätzer von B und σ^2 durch

$$\hat{B} = Q_q (\Lambda_q - \sigma^2 I_m)^{1/2} \text{ and } \hat{\sigma}^2 = \frac{1}{m - l} \sum_{j=l+1}^{m} \lambda_j, \tag{43}$$

gegeben.

Siehe Appendix für einen Beweis.

Definition (Modell der Hauptkomponentenanalyse)

Es sei

$$p_\theta(x, y) = N(y; Bx, R)N(x; 0, I_k) \tag{44}$$

ein LNM mit der Beobachtungsrauschenkovarianzmatrix

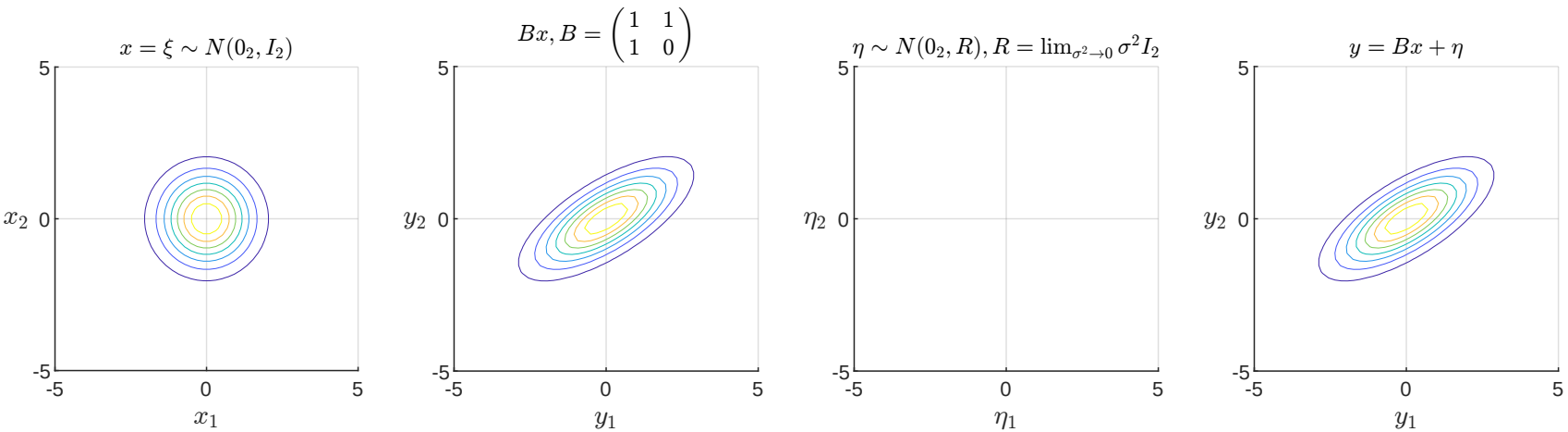
$$R := \lim_{\sigma^2 \to 0} \sigma^2 I_m \in \mathbb{R}^{m \times m}. \tag{45}$$

Dann heißt $p_\theta(x, y)$ *Modell der Hauptkomponentenanalyse*.

Bemerkungen

- B enkodiert die Beziehung zur klassischen Hauptkomponentenanalyse.
- Das Beobachtungsrauschen wird als nicht-existent angenommen.

Visuelle Intuition zur Hauptkomponentenanalyse



Theorem (Parameter der Hauptkomponentenanalyse)

Es sei

$$p_\theta(x, y) = N\left(y; Bx, \lim_{\sigma^2 \to 0} \sigma^2 I_m\right) N(x; 0_k, I_k) \tag{46}$$

ein Hauptkomponentenanalysemodell und es sei

$$\mathbb{C}(Y) = Q^T \Lambda Q \tag{47}$$

die Hauptkomponentenanalyse der zugehörigen marginalen Datenverteilung. Dann gilt

$$B = Q\Lambda^{\frac{1}{2}}. \tag{48}$$

Beweis

Wir halten zunächst fest, dass die marginale Datenverteilung des Hauptkomponentenanalysemodells im Limit $\sigma^2 \to 0$ gegeben ist durch

$$p_\theta(y) = N(y; 0_m, BB^T) \text{ and thus } \mathbb{C}(y) = BB^T. \tag{49}$$

Es ergibt sich also

$$\mathbb{C}(y) = BB^T \Leftrightarrow Q\Lambda Q^T = BB^T \Leftrightarrow B = Q\Lambda^{\frac{1}{2}}. \tag{50}$$

□

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Anwendungsbeispiel

Selbstkontrollfragen

Appendix

Nine mental ability tests | Intelligenzforschungsdatensatz

Holzinger and Swineford (1939)

Visualization

1. Visual Perception
2. Cubes
3. Lozenges

Verbal intelligence

4. Paragraph Comprehension
5. Sentence Completion
6. Word Meaning

Speed

7. Addition
8. Counting dots
9. Straight-Curved Capitals

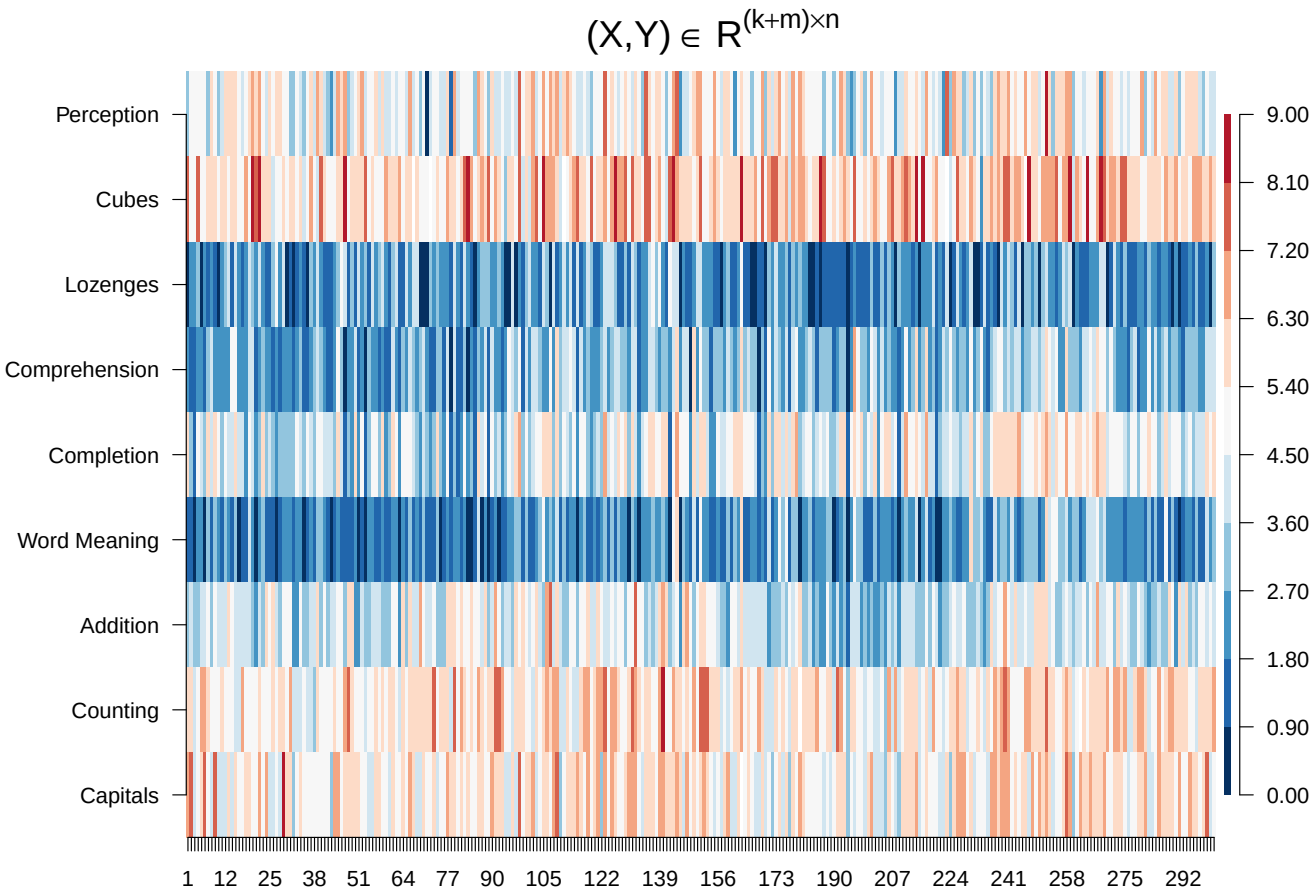
Beobachteter Datensatz (n = 301)

301 Proband:innen | 11 - 16 Jahre

Probandin:innen 1 - 10

	1	2	3	4	5	6	7	8	9	10
Perception	3.33	5.33	4.50	5.33	4.83	5.33	2.83	5.67	4.50	3.50
Cubes	7.75	5.25	5.25	7.75	4.75	5.00	6.00	6.25	5.75	5.25
Lozenges	0.38	2.12	1.88	3.00	0.88	2.25	1.00	1.88	1.50	0.75
Comprehension	2.33	1.67	1.00	2.67	2.67	1.00	3.33	3.67	2.67	2.67
Completion	5.75	3.00	1.75	4.50	4.00	3.00	6.00	4.25	5.75	5.00
Word Meaning	1.29	1.29	0.43	2.43	2.57	0.86	2.86	1.29	2.71	2.57
Addition	3.39	3.78	3.26	3.00	3.70	4.35	4.70	3.39	4.52	4.13
Counting	5.75	6.25	3.90	5.30	6.30	6.65	6.20	5.15	4.65	4.55
Capitals	6.36	7.92	4.42	4.86	5.92	7.50	4.86	3.67	7.36	4.36

Beobachteter Datensatz (n = 301)



Stichprobenkovarianzmatrix und Stichprobenkorrelationsmatrix

S_y

	Perception	Cubes	Lozenges	Comprehension	Completion	Word Meaning	Addition	Counting	Capitals
Perception	+1.4	+0.4	+0.6	+0.5	+0.4	+0.5	+0.1	+0.3	+0.5
Cubes	+0.4	+1.4	+0.5	+0.2	+0.2	+0.2	−0.1	+0.1	+0.2
Lozenges	+0.6	+0.5	+1.3	+0.2	+0.1	+0.2	+0.1	+0.2	+0.4
Comprehension	+0.5	+0.2	+0.2	+1.4	+1.1	+0.9	+0.2	+0.1	+0.2
Completion	+0.4	+0.2	+0.1	+1.1	+1.7	+1.0	+0.1	+0.2	+0.3
Word Meaning	+0.5	+0.2	+0.2	+0.9	+1.0	+1.2	+0.1	+0.2	+0.2
Addition	+0.1	−0.1	+0.1	+0.2	+0.1	+0.1	+1.2	+0.5	+0.4
Counting	+0.3	+0.1	+0.2	+0.1	+0.2	+0.2	+0.5	+1.0	+0.5
Capitals	+0.5	+0.2	+0.4	+0.2	+0.3	+0.2	+0.4	+0.5	+1.0

R_y

	Perception	Cubes	Lozenges	Comprehension	Completion	Word Meaning	Addition	Counting	Capitals
Perception	+1.0	+0.3	+0.4	+0.4	+0.3	+0.4	+0.1	+0.2	+0.4
Cubes	+0.3	+1.0	+0.3	+0.2	+0.1	+0.2	−0.1	+0.1	+0.2
Lozenges	+0.4	+0.3	+1.0	+0.2	+0.1	+0.2	+0.1	+0.2	+0.3
Comprehension	+0.4	+0.2	+0.2	+1.0	+0.7	+0.7	+0.2	+0.1	+0.2
Completion	+0.3	+0.1	+0.1	+0.7	+1.0	+0.7	+0.1	+0.1	+0.2
Word Meaning	+0.4	+0.2	+0.2	+0.7	+0.7	+1.0	+0.1	+0.1	+0.2
Addition	+0.1	−0.1	+0.1	+0.2	+0.1	+0.1	+1.0	+0.5	+0.3
Counting	+0.2	+0.1	+0.2	+0.1	+0.1	+0.1	+0.5	+1.0	+0.4
Capitals	+0.4	+0.2	+0.3	+0.2	+0.2	+0.2	+0.3	+0.4	+1.0

Modellschätzung

Die Art der Tests motiviert ein 3-Faktor-Modell mit Faktoren

- Visualisierungsvermögen
- Verbale Intelligenz
- Schnelligkeit

```
# Lawley & Maxwell (1971) Schätzverfahren für Faktorenanalyse
fa              = factanal(t(Y), factors = 3, rotation = "none")  # Parameterschätzung
B_hat_fa        = fa$loadings[1:9,1:3]                            # \hat{B}
R_hat_fa        = diag(fa$uniquenesses)                           # \hat{R}

# Parametschätzer-basierte Kovarianz und Korrelationsmatrix
Sigma_hat_y_fa  = B_hat_fa %*% t(B_hat_fa) + R_hat_fa
D_hat_y_fa      = diag(1/sqrt(diag(Sigma_hat_y_fa)))
rho_hat_y_fa    = D_hat_y_fa %*% Sigma_hat_y_fa %*% D_hat_y_fa
```

Modellschätzung

$\hat{B}_{FA}$

	Factor1	Factor2	Factor3
1	+0.15	+0.54	−0.39
2	+0.05	+0.70	−0.33
3	+0.07	+0.76	−0.21
4	−0.40	+0.31	+0.52
5	−0.53	+0.29	+0.40
6	−0.57	+0.34	+0.44
7	+0.59	−0.01	+0.23
8	+0.74	+0.15	+0.43
9	+0.63	+0.26	+0.20

$\hat{R}_{FA}$

	1	2	3	4	5	6	7	8	9
1	+0.54	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
2	+0.00	+0.40	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
3	+0.00	+0.00	+0.38	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00
4	+0.00	+0.00	+0.00	+0.47	+0.00	+0.00	+0.00	+0.00	+0.00
5	+0.00	+0.00	+0.00	+0.00	+0.47	+0.00	+0.00	+0.00	+0.00
6	+0.00	+0.00	+0.00	+0.00	+0.00	+0.36	+0.00	+0.00	+0.00
7	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.59	+0.00	+0.00
8	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.24	+0.00
9	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.00	+0.49

Marginale Kovarianmatrix und Korrelationsmatrix

$\hat{\Sigma}_y$

	Perception	Cubes	Lozenges	Comprehension	Completion	Word Meaning	Addition	Counting	Capitals
Perception	+1.0	+0.3	+0.4	+0.3	+0.3	+0.4	+0.1	+0.2	+0.4
Cubes	+0.3	+1.0	+0.3	+0.2	+0.1	+0.2	−0.0	+0.1	+0.2
Lozenges	+0.4	+0.3	+1.0	+0.2	+0.1	+0.2	+0.0	+0.2	+0.3
Comprehension	+0.3	+0.2	+0.2	+1.0	+0.7	+0.7	+0.1	+0.1	+0.2
Completion	+0.3	+0.1	+0.1	+0.7	+1.0	+0.7	+0.1	+0.1	+0.2
Word Meaning	+0.4	+0.2	+0.2	+0.7	+0.7	+1.0	+0.1	+0.1	+0.2
Addition	+0.1	−0.0	+0.0	+0.1	+0.1	+0.1	+1.0	+0.5	+0.3
Counting	+0.2	+0.1	+0.2	+0.1	+0.1	+0.1	+0.5	+1.0	+0.4
Capitals	+0.4	+0.2	+0.3	+0.2	+0.2	+0.2	+0.3	+0.4	+1.0

$\hat{\rho}_y$

	Perception	Cubes	Lozenges	Comprehension	Completion	Word Meaning	Addition	Counting	Capitals
Perception	+1.0	+0.3	+0.4	+0.3	+0.3	+0.4	+0.1	+0.2	+0.4
Cubes	+0.3	+1.0	+0.3	+0.2	+0.1	+0.2	−0.0	+0.1	+0.2
Lozenges	+0.4	+0.3	+1.0	+0.2	+0.1	+0.2	+0.0	+0.2	+0.3
Comprehension	+0.3	+0.2	+0.2	+1.0	+0.7	+0.7	+0.1	+0.1	+0.2
Completion	+0.3	+0.1	+0.1	+0.7	+1.0	+0.7	+0.1	+0.1	+0.2
Word Meaning	+0.4	+0.2	+0.2	+0.7	+0.7	+1.0	+0.1	+0.1	+0.2
Addition	+0.1	−0.0	+0.0	+0.1	+0.1	+0.1	+1.0	+0.5	+0.3
Counting	+0.2	+0.1	+0.2	+0.1	+0.1	+0.1	+0.5	+1.0	+0.4
Capitals	+0.4	+0.2	+0.3	+0.2	+0.2	+0.2	+0.3	+0.4	+1.0

$\Rightarrow$ Modellierung des Datensatzes mit einem 3-Faktor Modell ist möglich.

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

Bayesian Information Criterion (cf. Horvath et al. (2021), Schwarz (1978))

$$\text{BIC} := \ln p_{\hat{\theta}}(Y) - \frac{l}{2} \ln n \tag{51}$$

$\ln p_{\hat{\theta}}(Y)$

- Logarithmierte marginale Datenwahrscheinlichkeit(sdichte) bei optimierten Parametern
- Maß für die Passungsgüte des Modells

$\frac{l}{2} \ln n$

- Stichprobengröße gewichtete Anzahl an Parametern l
- Maß für die Komplexität des Modells

$$\text{BIC} = \text{Passungsgüte} - \text{Komplexität} \tag{52}$$

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

Theorem (BIC für Faktorenanalysemodelle)

$p_\theta(X, Y)$ sei die WDF der gemeinsamen Verteilung eines Faktorenanalysemodelldatensatzes mit $X \in \mathbb{R}^{k \times n}$ und $Y \in \mathbb{R}^{m \times n}$. Weiterhin seien $\hat{\theta} := \{\hat{B}, \hat{R}\}$ (marginale) Maximum Likelihood Schätzer von $\theta := \{B, R\}$ und es sei

$$\hat{\Sigma} := \hat{B}\hat{B}^T + \hat{R}. \tag{53}$$

Dann ergibt sich das Bayesian Information Criterion zu

$$\text{BIC} = -\frac{nm}{2} \ln 2\pi - \frac{n}{2} \ln |\hat{\Sigma}| - \frac{1}{2} \text{tr}\left(\hat{\Sigma}^{-1} Y Y^T\right) - \frac{mk + m}{2} \ln n \tag{54}$$

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

Beweis

Nach Definition der Verteilung von LNM Datensätzen gilt

$$p_{\hat{\theta}}(Y) = \prod_{i=1}^{n} N\left(y^{(i)}; 0_m, \hat{B}\hat{B}^T + \hat{R}\right). \tag{55}$$

Die Anzahl der Parameter eines Faktorenanalysemodells ergibt sich als $l = mk + m$, wobei mk die Anzahl der Einträge in B und m die Anzahl der Einträge in R sind. Mit der Eigenschaft

$$x^T Ax = \mathsf{tr}(Axx^T) \tag{56}$$

der Matrix Spur

$$\mathsf{tr}(A) := \sum_{i=1}^{n} a_i i \text{ für } A := (a_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n} \tag{57}$$

und der Definition von

$$\hat{\Sigma} := \hat{B}\hat{B}^T + \hat{R} \tag{58}$$

ergibt sich dann

$$\mathsf{BIC} = \ln p_{\hat{\theta}}(Y) - \frac{l}{2} \ln n = \ln \left(\prod_{i=1}^{n} N\left(y^{(i)}; 0_m, \hat{\Sigma}\right) \right) - \frac{mk + m}{2} \ln n \tag{59}$$

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

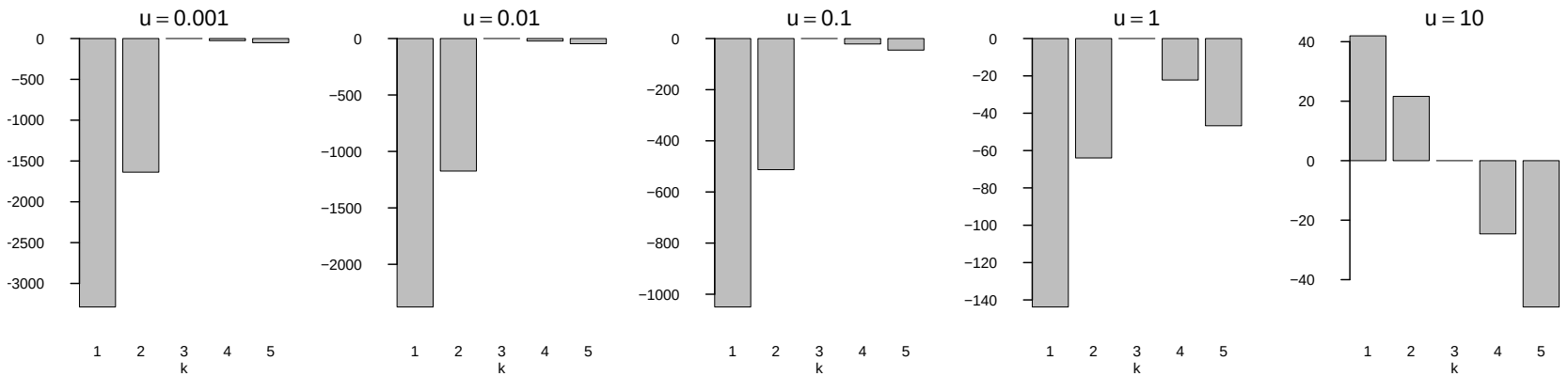
Beweis (fortgeführt)

Es ergibt sich also

$$
\begin{aligned}
\text{BIC} &= \sum_{i=1}^{n} \ln N\left(y^{(i)}; 0_m, \hat{\Sigma}\right) - \frac{mk+m}{2} \ln n \\
&= \sum_{i=1}^{n} \ln\left((2\pi)^{-\frac{m}{2}} |\hat{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} {y^{(i)}}^T \Sigma^{-1} y^{(i)}\right)\right) - \frac{mk+m}{2} \ln n \\
&= \sum_{i=1}^{n} \left(-\frac{m}{2} \ln 2\pi - \frac{1}{2} \ln |\hat{\Sigma}| - \frac{1}{2} {y^{(i)}}^T \Sigma^{-1} y^{(i)}\right) - \frac{mk+m}{2} \ln n \\
&= -\frac{nm}{2} \ln 2\pi - \frac{n}{2} \ln |\hat{\Sigma}| - \frac{1}{2} \sum_{i=1}^{n} {y^{(i)}}^T \Sigma^{-1} y^{(i)} - \frac{mk+m}{2} \ln n \\
&= -\frac{nm}{2} \ln 2\pi - \frac{n}{2} \ln |\hat{\Sigma}| - \frac{1}{2} \text{tr}\left(\Sigma^{-1} \sum_{i=1}^{n} y^{(i)} {y^{(i)}}^T\right) - \frac{mk+m}{2} \ln n \\
&= -\frac{nm}{2} \ln 2\pi - \frac{n}{2} \ln |\hat{\Sigma}| - \frac{1}{2} \text{tr}\left(\hat{\Sigma}^{-1} Y Y^T\right) - \frac{mk+m}{2} \ln n
\end{aligned}
\tag{60}
$$

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

Simulationsbasierte Validierung des BIC bei $k = 3, n = 301$ und $R := uI_m$ mit $u > 0$.



$\Rightarrow$ BIC erlaubt Model Recovery in Szenarien mit niedrigem bis mittlerem Beobachtungsrauschen.

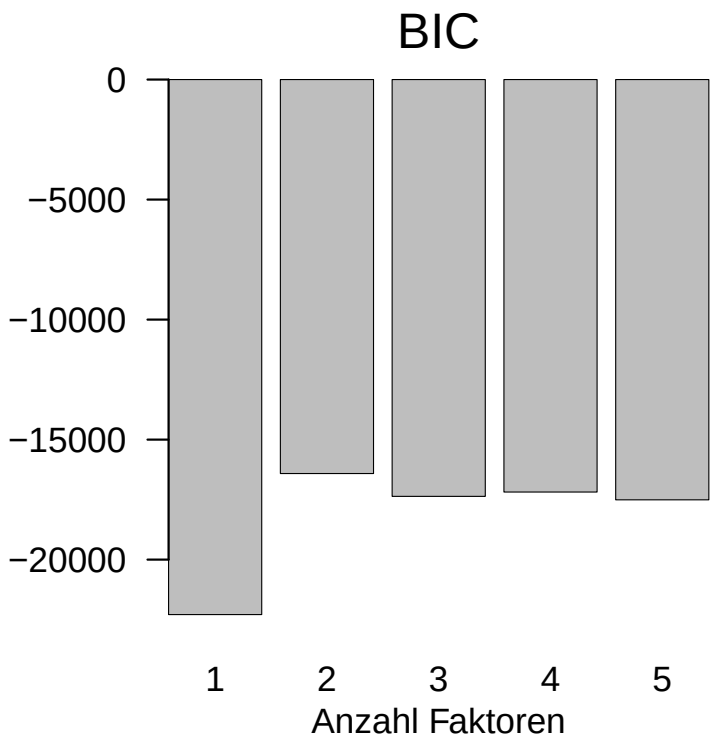
Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

BIC Modellevaluation

```
# Datensatz
library(lavaan)                                          # Lavaan SEM Paket
data(HolzingerSwineford1939)                             # Datensatz
Y              = t(HolzingerSwineford1939[,7:15])        # Datenmatrix
m              = nrow(Y)                                 # Anzahl Tests/Variablen
n              = ncol(Y)                                 # Anzahl Proband:innen
max_k          = 5                                       # Maximale Faktorenanzahl
BIC            = rep(NaN,max_k)                          # BIC

# Modell Iterationen
for(k in 1:max_k){

  # Modellformulierung und Modellschätzung
  fa         = factanal(t(Y), factors = k, rotation = "none")
  B_hat      = fa$loadings[,1:k]
  R_hat      = diag(fa$uniquenesses)

  # Modellschätzer
  Sigma_hat = B_hat %*% t(B_hat) + R_hat

  # Modellevaluation
  BIC[k]      = (-((n*m)/2)*log(2*pi)
                 -(n/2*log(det(Sigma_hat)))
                 -(1/2)*sum(diag(inv(Sigma_hat)%*%Y%*%t(Y)))
                 -(((m*k)+m)/2)*log(n))
}
```

Modellvergleich | Welche Anzahl von Faktoren zwischen 1 und 5 ist am besten?

BIC Modellevaluation



$\Rightarrow$ Das BIC Modellvergleichskriterium legt ein Modell mit 2 Faktoren nahe.

Selbstkontrollfragen

1. Definieren Sie den Begriff des Linearen Normalverteilungsmodells (LNMs).
2. Erläutern Sie die hierarchische Form eines LNMs.
3. Geben Sie das Theorem zur LNM Datenverteilung wieder.
4. Was modellieren/erklären LNMs?
5. Definieren Sie die WDFen der gemeinsamen und marginalen Datenverteilungen eines LNM Datensatzes.
6. Erläutern Sie die Fragestellungen der Inferenz und des Lernens bei der Schätzung von LNMen.
7. Geben Sie das Theorem zum Exakten Expectation-Maximization Algorithmus wieder.
8. Erläutern Sie das Theorem zum Exakten Expectation-Maximization Algorithmus.
9. Geben Sie die Definition eines Faktorenanalysemodells wieder.
10. Erläutern Sie das Verhältnis von Datenkomponentenkovarianzen und -varianzen im Faktorenanalysemodell.
11. Geben Sie die Definition eines Hauptkomponentenanalysemodells wieder.
12. Geben Sie das Theorem zum Parameter des Hauptkomponentenanalysemodells wieder.
13. Erläutern Sie den Zusammenhang von modell-freier PCA (Einheit (4)) und modell-basierter PCA (Einheit (5)).
14. Definieren Sie das Bayesian Information Criterion (BIC).
15. Erläutern Sie die Intuition zum BIC im Kontext von Modellvergleichen.

Grundlagen

Inferenz und Lernen

Faktorenanalyse

Hauptkomponentenanalyse Revisited

Selbstkontrollfragen

Appendix

Theorem (Jensen's inequality)

Let x be a random variable and g be a convex function, i.e.

$$g(\lambda x_1 + (1-\lambda)x_2) \leq \lambda g(x_1) + (1-\lambda)g(x_2). \tag{61}$$

Then

$$\mathbb{E}(g(x)) \geq g(\mathbb{E}(x)). \tag{62}$$

Conversely, let g be a concave function, i.e.

$$g(\lambda x_1 + (1-\lambda)x_2) \geq \lambda g(x_1) + (1-\lambda)g(x_2). \tag{63}$$

Then

$$\mathbb{E}(g(x)) \leq g(\mathbb{E}(x)). \tag{64}$$

Proof

By adapting the proof of Theorem 4.7.8 in Casella and Berger (2012) , we show the inequality for the concave case. Let f be a tangent line at the point $g(\mathbb{E}(x))$, i.e. is a linear-affine function of the form $f(x) := ax + b$ for some $a, b \in \mathbb{R}$ with $f(\mathbb{E}(x)) = g(\mathbb{E}(x))$. Because g is concave, we have $g(x) \leq ax + b$ for all $x \in \mathbb{R}$ and thus als $g(x) \leq ax + b$. Hence,

$$\mathbb{E}(g(x)) \leq \mathbb{E}(ax + b) = a\mathbb{E}(x) + b = f(\mathbb{E}(x)) = g(\mathbb{E}(x)). \tag{65}$$

Remarks

- For convex g the function's graph lies below the straight line $g(x_1)$ to $g(x_2)$.
- For concave g the function's graph lies above the straight line $g(x_1)$ to $g(x_2)$.
- The logarithm is a concave function, hence $\mathbb{E}(\ln x) \leq \ln \mathbb{E}(x)$.

Beweis des Theorems zum exakten EM Algorithmus eines Faktorenanalysemodells

The E Step of the algorithm follows directly with the LGM data set inference theorem. We thus focus on the derivation of the M Step. To this end, recall that the M Step of the exact EM algorithm takes the form

$$\theta^{(j)} := \arg\max_{\theta} \int p_{\theta^{(j-1)}}(X|Y) \ln p_{\theta}(X, Y)\, dX \tag{66}$$

For LGMs, the maximization of the expected joint likelihood with respect to θ can be carried out analytically and in the sense of the necessary condition for a maximum.

To ease notation, we will write $\tilde{\theta} := \theta^{(j-1)}$ in the following and denote the expectation of a function f of the unobserved data X under the conditional distribution $p_{\tilde{\theta}}(X|Y)$ as a conditional expectation:

$$\mathbb{E}_{\tilde{\theta}}(f(X)|Y) := \mathbb{E}_{p_{\tilde{\theta}}(X|Y)}(f(X)) = \int f(X)\, p_{\tilde{\theta}}(X|Y)\, dX. \tag{67}$$

With these simplifications, we thus aim to evaluate

$$\frac{\partial}{\partial B}\mathbb{E}_{p_{\tilde{\theta}}(X|Y)}\left(\ln p_{\theta}(X, Y)\right) \text{ and } \frac{\partial}{\partial R}\mathbb{E}_{p_{\tilde{\theta}}(X|Y)}\left(\ln p_{\theta}(X, Y)\right), \tag{68}$$

set the results to zero, and solve for update equations for $B^{(j)}$ and $R^{(j)}$.

We proceed in four steps. (1) We first use the IID data assumption and the linearity of conditional expectations and derivatives to simplify the problem of evaluating the expected data set joint likelihood and its partial derivatives for a single data point $(x^{(i)}, y^{(i)})$. We then (2) evaluate the conditional expectation and (3) evaluate the respective partial derivatives. By capitalizing on the results form the first step, we then (4) evaluate and simplify the ensuing parameter update equations.

Appendix

(1) Expected joint likelihood partial derivatives under IID data assumptions

$$
\begin{aligned}
\frac{\partial}{\partial B} \mathbb{E}_{p_{\tilde{\theta}}(X|Y)} \left(\ln p_\theta(X, Y) \right) &= \frac{\partial}{\partial B} \left(\mathbb{E}_{p_{\tilde{\theta}}(X|Y)} \left(\ln \prod_{i=1}^{n} p_\theta \left(x^{(i)}, y^{(i)} \right) \right) \right) \\
&= \frac{\partial}{\partial B} \left(\mathbb{E}_{p_{\tilde{\theta}}(X|Y)} \left(\sum_{i=1}^{n} \ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right) \right) \\
&= \frac{\partial}{\partial B} \sum_{i=1}^{n} \mathbb{E}_{\prod_{i=1}^{n} p_{\tilde{\theta}} \left(x^{(i)}, y^{(i)} \right)} \left(\ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right) \qquad (69) \\
&= \frac{\partial}{\partial B} \sum_{i=1}^{n} \mathbb{E}_{p_{\tilde{\theta}} \left(x^{(i)}, y^{(i)} \right)} \left(\ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial B} \mathbb{E}_{p_{\tilde{\theta}} \left(x^{(i)}, y^{(i)} \right)} \left(\ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right),
\end{aligned}
$$

$$
\frac{\partial}{\partial R} \mathbb{E}_{p_{\tilde{\theta}}(X|Y)} \left(\ln p_\theta(X, Y) \right) = \sum_{i=1}^{n} \frac{\partial}{\partial R} \mathbb{E}_{p_{\tilde{\theta}} \left(x^{(i)}, y^{(i)} \right)} \left(\ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right). \qquad (70)
$$

(2) Expected joint likelihood for a single data point

For ease of notation, we omit the (i) superscript indexing the data realizations in this step.

$$
\begin{aligned}
&\mathbb{E}_{p_{\tilde{\theta}}(x|y)}\left(\ln p_\theta(x,y)\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(\ln p_\theta(x,y)|y\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(\ln\left(N(y;Bx,R)N(x;0,I_k)\right)|y\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(\ln N(y;Bx,R) + \ln N(x;0,I_k)|y\right) \\
&= \mathbb{E}\left(\ln\left((2\pi)^{-\frac{m}{2}}|R|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(y-Bx)^T R^{-1}(y-Bx)\right)\right) + \ln\left((2\pi)^{-\frac{k}{2}}|I_k|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}x^T x\right)\right)|y\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(-\frac{m+k}{2}\ln 2\pi - \frac{1}{2}\ln|R| - \frac{1}{2}\left(y^T R^{-1}y - 2y^T R^{-1}Bx + x^T B^T R^{-1}Bx\right) - \frac{1}{2}\ln 1 - \frac{1}{2}x^T x|y\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(-\frac{m+k}{2}\ln 2\pi - \frac{1}{2}\ln|R| - \frac{1}{2}y^T R^{-1}y + y^T R^{-1}Bx - \frac{1}{2}x^T B^T R^{-1}Bx - \frac{1}{2}x^T x|y\right) \\
&= \mathbb{E}_{\tilde{\theta}}\left(-\frac{m+k}{2}\ln 2\pi - \frac{1}{2}\ln|R| - \frac{1}{2}y^T R^{-1}y + y^T R^{-1}Bx - \frac{1}{2}\mathrm{tr}\left(B^T R^{-1}Bxx^T\right) - \frac{1}{2}x^T x|y\right) \\
&= -\frac{m+k}{2}\ln 2\pi - \frac{1}{2}\ln|R| - \frac{1}{2}y^T R^{-1}y + y^T R^{-1}B\mathbb{E}_{\tilde{\theta}}(x|y) - \frac{1}{2}\mathrm{tr}\left(B^T R^{-1}B\mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right)\right) - \frac{1}{2}\mathbb{E}_{\tilde{\theta}}\left(x^T x|y\right)
\end{aligned}
$$

where in the 7th equality, we made use of the fact that $x^T Ax = \mathrm{tr}(Axx^T)$ for $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$.

(3) Partial derivatives

To evaluate the partial derivatives of the conditional expected joint likelihood with respect to the matrices B and R, we require the following identities from matrix calculus (z.B. Petersen and Pedersen (2012)):

$$\frac{\partial}{\partial X} A^T X B = A B^T, \frac{\partial}{\partial X} \mathsf{tr}(XA) = A^T, \frac{\partial}{\partial X} \mathsf{tr}(X^T A X B) = A X B + A^T X B^T, \frac{\partial}{\partial X} \ln |X| = \left(X^{-1}\right)^T. \tag{71}$$

We then have

$$\begin{aligned}
\frac{\partial}{\partial B} \mathbb{E}_{p_{\tilde{\theta}}(x|y)} \left(\ln p_\theta(x, y)\right) &= \frac{\partial}{\partial B} \left(y^T R^{-1} B \mathbb{E}_{\tilde{\theta}}(x|y) - \frac{1}{2} \mathsf{tr}\left(B^T R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right)\right)\right) \\
&= \frac{\partial}{\partial B} \left(y^T R^{-1} B \mathbb{E}_{\tilde{\theta}}(x|y)\right) - \frac{1}{2} \frac{\partial}{\partial B} \mathsf{tr}\left(B^T R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right)\right) \\
&= \left(y^T R^{-1}\right)^T \mathbb{E}_{\tilde{\theta}}(x|y)^T - \frac{1}{2} R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right) - \frac{1}{2} \left(R^{-1}\right)^T B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right)^T \\
&= R^{-1} y \mathbb{E}_{\tilde{\theta}}(x|y)^T - \frac{1}{2} R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right) - \frac{1}{2} R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right) \\
&= R^{-1} y \mathbb{E}_{\tilde{\theta}}(x|y)^T - R^{-1} B \mathbb{E}_{\tilde{\theta}}\left(xx^T|y\right).
\end{aligned} \tag{72}$$

Similarly, we have

$$
\begin{aligned}
&\frac{\partial}{\partial R^{-1}} \mathbb{E}_{p_{\tilde{\theta}}(x|y)} \left(\ln p_\theta(x, y) \right) \\
&= \frac{\partial}{\partial R^{-1}} \left(-\frac{1}{2} \ln |R| - \frac{1}{2} y^T R^{-1} y + y^T R^{-1} B \mathbb{E}_{\tilde{\theta}}(x|y) - \frac{1}{2} \mathsf{tr} \left(B^T R^{-1} B \mathbb{E}_{\tilde{\theta}} \left(x x^T | y \right) \right) \right) \\
&= -\frac{1}{2} \frac{\partial}{\partial R^{-1}} \ln |R| - \frac{1}{2} \frac{\partial}{\partial R^{-1}} y^T R^{-1} y + \frac{\partial}{\partial R^{-1}} y^T R^{-1} C \mathbb{E}_{\tilde{\theta}}(x|y) - \frac{1}{2} \frac{\partial}{\partial R^{-1}} \mathsf{tr} \left(R^{-1} C \mathbb{E}_{\tilde{\theta}} \left(x x^T | y \right) B^T \right) \\
&= \frac{1}{2} R - \frac{1}{2} y y^T + y \left(B \mathbb{E}_{\tilde{\theta}}(x|y) \right)^T - \frac{1}{2} \left(B \mathbb{E}_{\tilde{\theta}} \left(x x^T | y \right) B^T \right)^T \\
&= \frac{1}{2} R - \frac{1}{2} y y^T + y \mathbb{E}_{\tilde{\theta}}(x|y)^T B^T - \frac{1}{2} B \mathbb{E}_{\tilde{\theta}} \left(x x^T | y \right) B^T .
\end{aligned}
$$

(4) Parameter update equations

Re-substitution then yields for the partial derivative with respect to B

$$\begin{aligned}
\frac{\partial}{\partial B}\mathbb{E}_{p_{\tilde{\theta}}(X|Y)}\left(\ln p_\theta(X,Y)\right) &= \sum_{i=1}^{n}\frac{\partial}{\partial B}\mathbb{E}_{p_{\tilde{\theta}}\left(x^{(i)},y^{(i)}\right)}\left(\ln p_\theta\left(x^{(i)},y^{(i)}\right)\right) \\
&= \sum_{i=1}^{n} R^{-1}{y^{(i)}}\mathbb{E}_{\tilde{\theta}}\left(x^{(i)}|y^{(i)}\right)^T - R^{-1}B\mathbb{E}_{\tilde{\theta}}\left(x^{(i)}{x^{(i)}}^T|y^{(i)}\right) \\
&= R^{-1}\sum_{i=1}^{n} y^{(i)}\mathbb{E}_{\tilde{\theta}}\left(x^{(i)}|y^{(i)}\right)^T - R^{-1}B\sum_{i=1}^{n}\mathbb{E}_{\tilde{\theta}}\left(x^{(i)}{x^{(i)}}^T|y^{(i)}\right).
\end{aligned} \tag{73}$$

Setting to zero and solving for $B^{(j)}$ then yields

$$R^{-1} B^{(j)} \sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) = R^{-1} \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T .$$

$$\Leftrightarrow B^{(j)} = \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T \left(\sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) \right)^{-1} . \tag{74}$$

Similarly, re-substitution yields for the partial derivative with respect to R

$$
\begin{aligned}
&\frac{\partial}{\partial R^{-1}} \mathbb{E}_{p_{\tilde{\theta}}(X|Y)} \left(\ln p_\theta(X, Y) \right) \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial R^{-1}} \mathbb{E}_{p_{\tilde{\theta}}\left(x^{(i)}, y^{(i)}\right)} \left(\ln p_\theta \left(x^{(i)}, y^{(i)} \right) \right) \\
&= \sum_{i=1}^{n} \frac{1}{2} R - \frac{1}{2} y^{(i)} y^{(i)T} + y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^T - \frac{1}{2} B \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)T} | y^{(i)} \right) B^T \\
&= \frac{n}{2} R - \frac{1}{2} \sum_{i=1}^{n} y^{(i)} y^{(i)T} + \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^T - \frac{1}{2} B \sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)T} | y^{(i)} \right) B^T .
\end{aligned}
\tag{75}
$$

Beweis (fortgeführt)

Setting to zero and solving for $R^{(j)}$ then yields

$$\frac{n}{2} R^{(j)} = \frac{1}{2} \sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^T + \frac{1}{2} B \sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) B^T$$

$$R^{(j)} = \frac{1}{n} \sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \frac{2}{n} \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^T + \frac{1}{n} B \sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) B^T$$

Substitution of the update equation for B further yields

$$
\begin{aligned}
R^{(j)} =& \frac{1}{n} \sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \frac{2}{n} \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^{(j)^T} \\
&+ \frac{1}{n} \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T \left(\sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) \right)^{-1} \sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) B^{(j)^T} \\
=& \frac{1}{n} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)^T} - 2 \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^{(j)^T} + \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^{(j)^T} \right) \\
=& \frac{1}{n} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^{(j)^T} \right)
\end{aligned}
$$

We thus obtained the parameter update equations

$$B^{(j)} = \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T \left(\sum_{i=1}^{n} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) \right)^{-1}$$

$$R^{(j)} = \frac{1}{n} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)^T} - \sum_{i=1}^{n} y^{(i)} \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right)^T B^{(j)^T} \right)$$

The computational forms of these update equations can be further simplified by noting that with

$$\mathbb{E} \left(x x^T | y \right) = \mu_{x|y} \mu_{x|y}^T + \Sigma_{x|y} \tag{76}$$

and the LGM inference theorem it holds that

$$\mathbb{E}_{\tilde{\theta}} \left(x^{(i)} | y^{(i)} \right) = \hat{x}^{(i)} \text{ and } \mathbb{E}_{\tilde{\theta}} \left(x^{(i)} x^{(i)^T} | y^{(i)} \right) = \hat{x}^{(i)} \hat{x}^{(i)^T} + \hat{\Sigma}^{(i)}. \tag{77}$$

We thus obtain

$$B^{(j)} = \sum_{i=1}^{n} y^{(i)} \hat{x}^{(i)T} \left(\sum_{i=1}^{n} \hat{x}^{(i)} \hat{x}^{(i)T} + \hat{\Sigma}^{(i)} \right)^{-1}$$

$$R^{(j)} = \frac{1}{n} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)T} - \sum_{i=1}^{n} y^{(i)} \hat{x}^{(i)T} B^{(j)T} \right).$$

Finally, enforcing the diagonality constraint on R can be achieved by setting

$$R^{(j)} := -\frac{1}{n} \mathsf{diag} \left(\sum_{i=1}^{n} y^{(i)} y^{(i)T} + \sum_{i=1}^{n} y^{(i)} \hat{x}^{(i)T} B^{(j)T} \right) \tag{78}$$

Beweis des Theorems zur direkten Maximum Marginal Likelihood Schätzung des probabilistischen PCA Modells

We only show that $\hat{B}$ as defined in the theorem corresponds to maximum of the log marginal likelihood function. To this end, we closely follow the respective proof in Tipping and Bishop (1999), and proceed in three steps: (1) We first rewrite the marginal data set log likelihood function of the PPCA model in a suitable manner. (2) We then evaluate its gradient with respect to B and (3) finally evaluate the resulting maximum marginal likelihood estimator.

(1) Log likelihood function

We first rewrite the marginal data set log likelihood function of the PPCA model

$$\ell : \Theta \to \mathbb{R}, \theta \mapsto \ell(\theta) := \ln p_\theta(Y) = \sum_{i=1}^{n} \ln N\left(y^{(i)}; 0_m, BB^T + \sigma^2 I_m\right) \tag{79}$$

with $\theta := \{B, \sigma^2\}$ in a way more amenable to direct maximization.

With the definitions of

$$\Sigma := BB^T + \sigma^2 I_m \text{ and } C := \frac{1}{n} \sum_{i=1}^{n} y^{(i)} y^{(i)T} \tag{80}$$

and the trace operator properties

$$x^T A x = \mathrm{tr}(A x x^T) \text{ and } \mathrm{tr}(A) + \mathrm{tr}(B) = \mathrm{tr}(A + B), \tag{81}$$

we have

$$
\begin{aligned}
\ln p_\theta(Y) &= \sum_{i=1}^{n} \ln N\left(y^{(i)}; 0_m, \Sigma\right) \\
&= \sum_{i=1}^{n} \ln\left((2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} {y^{(i)}}^T \Sigma^{-1} y^{(i)}\right)\right) \\
&= -\frac{n}{2}\left(m \ln 2\pi + \ln|\Sigma| + \sum_{i=1}^{n} \frac{1}{n} {y^{(i)}}^T \Sigma^{-1} y^{(i)}\right) \\
&= -\frac{n}{2}\left(m \ln 2\pi + \ln|\Sigma| + \sum_{i=1}^{n} \mathrm{tr}\left(\Sigma^{-1} \frac{1}{n} y^{(i)} {y^{(i)}}^T\right)\right) \\
&= -\frac{n}{2}\left(m \ln 2\pi + \ln|\Sigma| + \mathrm{tr}\left(\sum_{i=1}^{n} \Sigma^{-1} \frac{1}{n} y^{(i)} {y^{(i)}}^T\right)\right) \\
&= -\frac{n}{2}\left(m \ln 2\pi + \ln|\Sigma| + \mathrm{tr}\left(\Sigma^{-1} \frac{1}{n} \sum_{i=1}^{n} y^{(i)} {y^{(i)}}^T\right)\right) \\
&= -\frac{n}{2}\left(m \ln 2\pi + \ln|\Sigma| + \mathrm{tr}\left(\Sigma^{-1} C\right)\right).
\end{aligned}
\tag{82}
$$

(2) Gradient of the log marginal likelihood function

With

$$\frac{\partial}{\partial X} \ln |X| = \left(X^{-1}\right)^T, \frac{\partial}{\partial X} X X^T = 2X, \text{ and } \frac{\partial}{\partial X} \text{tr}(XA) = A^T, \tag{83}$$

the gradient of the log marginal likelihood function with respect to B evaluates to

$$\begin{aligned}
\frac{\partial}{\partial B} \ell(\theta) &= -\frac{n}{2} \frac{\partial}{\partial B} \left(m \ln 2\pi + \ln |BB^T + \sigma^2 I_m| + \text{tr}\left(\left(BB^T + \sigma^2 I_m\right)^{-1} C \right) \right) \\
&= -\frac{n}{2} \frac{\partial}{\partial B} \ln |BB^T + \sigma^2 I_m| - \frac{n}{2} \frac{\partial}{\partial B} \text{tr}\left(\left(BB^T + \sigma^2 I_m\right)^{-1} C \right) \\
&= -\frac{n}{2} 2 \left(\left(BB^T + \sigma^2 I_m\right)^{-1} B \right)^T + \frac{n}{2} 2 \left(BB^T + \sigma^2 I_m\right)^{-1} C \left(BB^T + \sigma^2 I_m\right)^{-1} B \\
&= n \left(-\Sigma^{-1} B + \Sigma^{-1} C \Sigma^{-1} B \right) \\
&= n \left(\Sigma^{-1} C \Sigma^{-1} B - \Sigma^{-1} B \right).
\end{aligned} \tag{84}$$

(3) Maximum marginal likelihood estimator evaluation

Setting the gradient of ℓ with respect to B to zero then yields

$$\Sigma^{-1} C \Sigma^{-1} \hat{B} - \Sigma^{-1} \hat{B} = 0 \Leftrightarrow \Sigma^{-1} \hat{B} = \Sigma^{-1} C \Sigma^{-1} \hat{B} \Leftrightarrow \hat{B} = C \Sigma^{-1} \hat{B}. \tag{85}$$

We consider solutions of this necessary condition for a stationary point of the log marginal likelihood function with $B \neq 0$ and $\Sigma \neq C$. To find these, we first express $\hat{B}$ in terms of its singular value decomposition

$$\hat{B} = U L V^T \tag{86}$$

where $U = (u_1, u_2, ..., u_l)$ is an $m \times q$ matrix of orthonormal column vectors, $L = \text{diag}(l_1, l_2, ..., l_q)$ is a $q \times q$ diagonal matrix of singular values, and V is a $q \times q$ orthogonal matrix. Substitution in the necessary condition for a stationary point then yields

$$CUL = U(\sigma^2 I_m + L^2)L. \tag{87}$$

For $l_j \neq 0$, eq. (87) implies that

$$Cu_j = (\sigma^2 + l_j^2)u_j \tag{88}$$

Hence, each column of U must be an eigenvector of C with corresponding eigenvalue $\lambda_j = \sigma^2 + l_j$, and thus

$$\lambda_j = \sigma^2 + l_j^2 \Leftrightarrow l_j^2 = \lambda_j - \sigma^2 \Leftrightarrow l_j = (\lambda_j - \sigma^2)^{\frac{1}{2}}. \tag{89}$$

For $l_j = 0$, u_j is arbitrary. Under the assumption that $l_j \neq 0$ for $j = 1, ..., m$, all potential solutions for $\hat{B}$ can thus be written in the form

$$\hat{B} = U_q \left(\Lambda_q - \sigma^2 I_m \right)^{\frac{1}{2}} R, \tag{90}$$

where U_q is a $m \times q$ matrix whose q columns are the eigenvectors of C, Λ_q is the diagonal matrix of the corresponding eigenvalues, and R is an arbitrary $q \times q$ orthogonal matrix, for example, $R = I_q$.

□

References I

Blei, David M., and Padhraic Smyth. 2017. "Science and Data Science." *Proceedings of the National Academy of Sciences* 114 (33): 8689–92. https://doi.org/10.1073/pnas.1702076114.

Casella, G, and R Berger. 2012. *Statistical Inference*. Duxbury.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x.

Holzinger, K. J., and F. Swineford. 1939. *A Study in Factor Analysis: The Stability of a Bifactor Solution*. Vol. 48. Supplementary Educational Monographs. University of Chicago.

Horvath, Lilla, Stanley Colcombe, Michael Milham, Shruti Ray, Philipp Schwartenbeck, and Dirk Ostwald. 2021. "Human Belief State-Based Exploration and Exploitation in an Information-Selective Symmetric Reversal Bandit Task." *Computational Brain & Behavior*, August. https://doi.org/10.1007/s42113-021-00112-3.

Hotelling, Harold. 1933. "Analysis of Complext Variables into Principal Components." *Journal of Educational Psychology* 24: 417-441 and 498-520.

Joreskog, K. G. 1969. "A General Approach to Confirmatory Maximum Likelihood Factor Analysis." *Psychometrika* 34: 183–202.

Lawley, N. D. 1953. "A Modified Method of Estimation in Factor Analysis and Some Large Sample Results." *Nord. Psyko. Monogr. Ser* 3: 35–42.

Ostwald, Dirk, Evgeniya Kirilina, Ludger Starke, and Felix Blankenburg. 2014. "A Tutorial on Variational Bayes for Latent Linear Stochastic Time-Series Models." *Journal of Mathematical Psychology* 60 (June): 1–19. https://doi.org/10.1016/j.jmp.2014.04.003.

Pearson, Karl. 1901. "On Lines and Planes of Closest Fit to Systems of Points in Space." *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (11): 559–72. https://doi.org/10.1080/14786440109462720.

Petersen, Kaare Brandt, and Michael Syskind Pedersen. 2012. "The Matrixcookbook," 72.

References II

Rosseel, Yves. 2012. "Lavaan: An R Package for Structural Equation Modeling." *Journal of Statistical Software* 48 (2). https://doi.org/10.18637/jss.v048.i02.

Roweis, Sam. 1998. "EM Algorithms for PCA and SPCA," 7.

Roweis, Sam, and Zoubin Ghahramani. 1999. "A Unifying Review of Linear Gaussian Models." *Neural Computation* 11 (2): 305–45. https://doi.org/10.1162/089976699300016674.

Rubin, Donald B., and Dorothy T. Thayer. 1982. "EM Algorithms for ML Factor Analysis." *Psychometrika* 47 (1): 69–76. https://doi.org/10.1007/BF02293851.

Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics* 6 (2): 461–64.

Starke, Ludger, and Dirk Ostwald. 2017. "Variational Bayesian Parameter Estimation Techniques for the General Linear Model." *Frontiers in Neuroscience* 11 (September). https://doi.org/10.3389/fnins.2017.00504.

Tipping, Michael E, and Christopher M Bishop. 1999. "Probabilistic Principal Component Analysis," 12.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(6) Optimierung

Multivariate Datenanalyse

Datum	Einheit	Thema
15.10.2021	Einführung	(0) Einführung
15.10.2021	Grundlagen	(1) Vektoren
22.10.2021	Grundlagen	(2) Matrizen I
29.10.2021	Grundlagen	(3) Matrizen II
05.11.2021	Grundlagen	(4) Multivariate Normalverteilung
12.11.2021	Latente Variablenmodelle	(5) Hauptkomponentenanalyse
19.11.2021	Latente Variablenmodell	(6) Faktorenanalyse
26.11.2021	Prädiktive Modellierung	(7) Optimierung
03.12.2021	Prädiktive Modellierung	(8) Lineare Diskriminanzanalyse und Logistische Regression
10.12.2021	Prädiktive Modellierung	(9) Support Vektor Maschinen
17.12.2021	Prädiktive Modellierung	(10) Neuronale Netze
	Weihnachtspause	
07.01.2022	Frequentistische Inferenz	(11) T-Tests
14.01.2022	Frequentistische Inferenz	(12) Einfaktorielle Varianzanalyse
21.01.2022	Frequentistische Inferenz	(13) Kanonische Korrelationsanalyse I
28.01.2022	Frequentistische Inferenz	(14) Kanonische Korrelationsanalyse II
22.02.2022	Klausur	12 - 13 Uhr, G26-H1
Jul 2022	Klausurwiederholungstermin	

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Prädiktive Modellierung

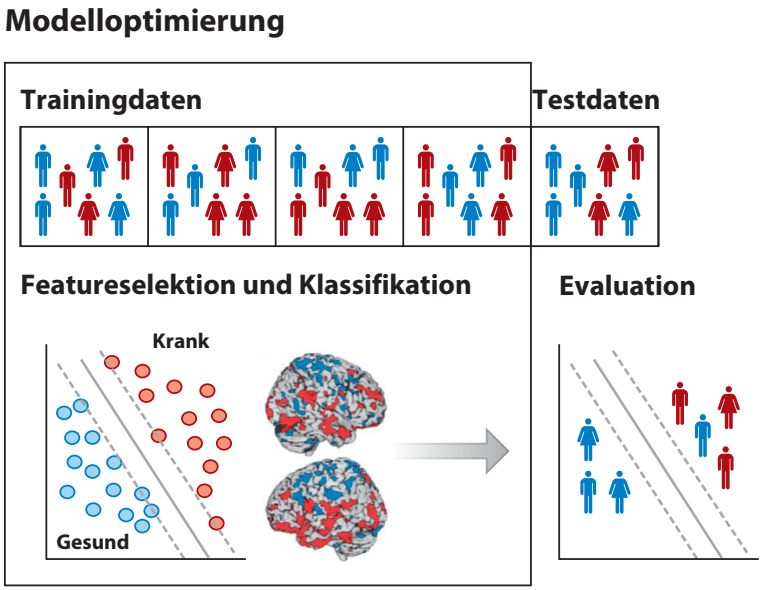
In der Psychologie möchte man gerne Dinge prädizieren, zum Beispiel

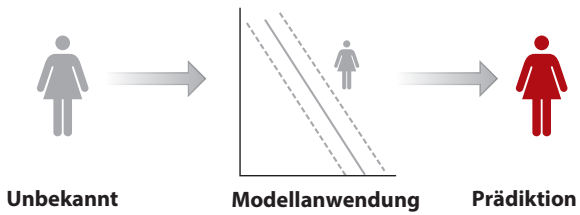
- Risiko psychiatrischer Erkrankung basierend auf Fragebogendaten
- Prognose psychiatrischer Erkrankung basierend auf Hirnbildgebungsdaten
- Psychotherapieerfolg basierend auf klinisch-psychologischen Tests
- Subjektive Wahrnehmung (Bewusstsein) basierend auf funktionellen Hirnbildgebungsdaten
- ...

Prädiktive Modellierung ist ein datenanalytisches Paradigma, das die prädiktive Rhethorik bedient.

- Die Datengrundlage ist hier oft multivariat, die Vorhersage ist oft univariat.
- Prädiktive Modellierung wird oft mit "Maschinellem Lernen" gleichgesetzt oder verwechselt.
- Prädiktive Modellierung wird oft mit "Künstlicher Intelligenz" gleichgesetzt oder verwechselt.
- Prädiktive Modellierung wird oft mit "Deep Learning" gleichgesetzt oder verwechselt.

Struktur der Prädiktiven Modellierung





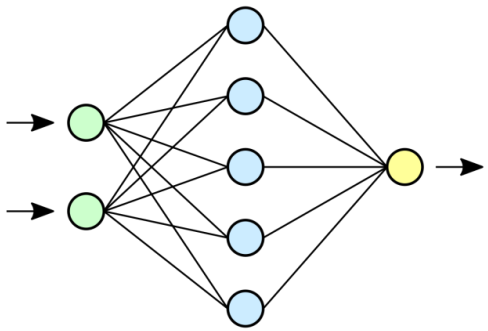
nach Dwyer, Falkai, and Koutsouleris (2018)

Rhethorik der Prädiktiven Modellierung

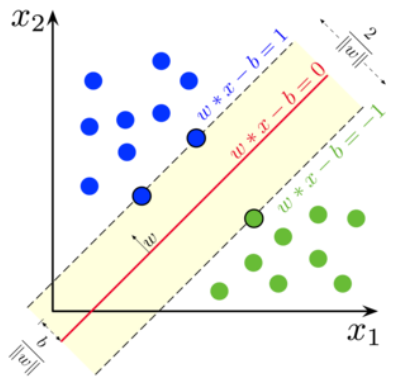
Daten	Trainingsdaten und Testdaten
Statistisches Modell	Modell, Machine Learning Algorithmus
Schätzen von Parametern	Trainieren des Modells, Lernen von Parametern, Supervised Learning

Typische Modelle der Prädiktiven Modellierung

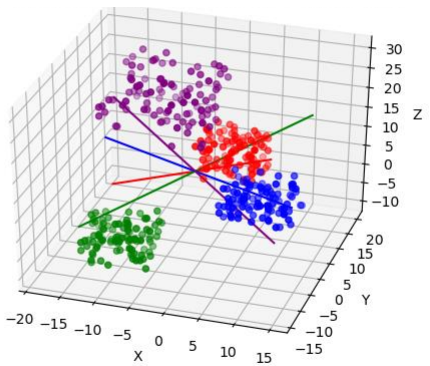
Neuronale Netze | Deep Learning



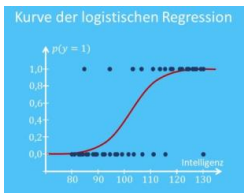
Support Vektor Maschinen



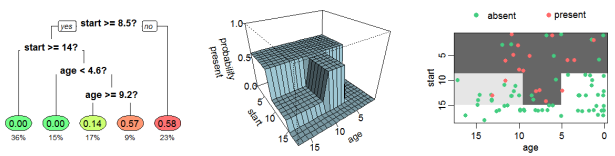
Lineare Diskriminanzanalyse



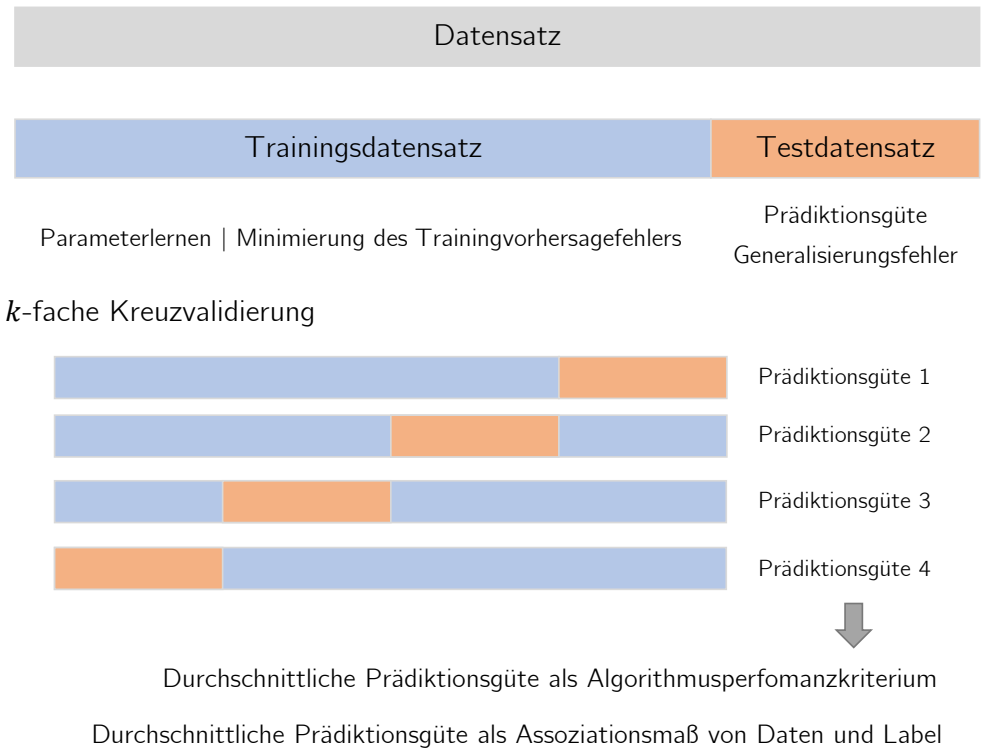
Logistische Regression



Entscheidungsbäume



Modellschätzung- und Modellevaluationsansatz der Prädiktiven Modellierung

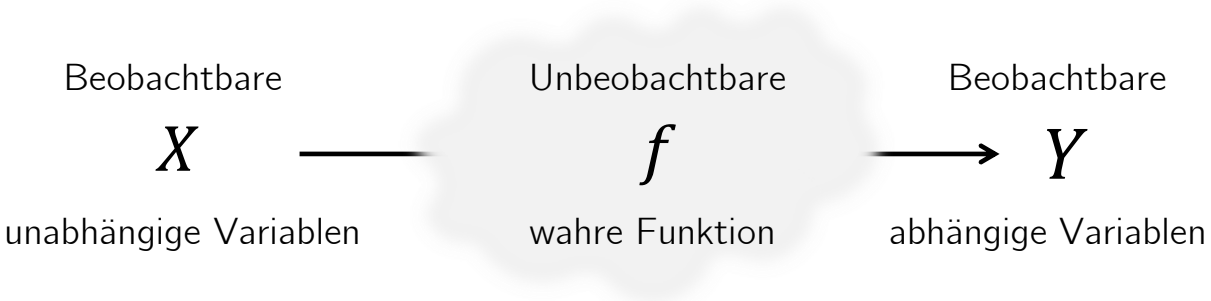


Die theoretische Analyse dieses Ansatzes heißt “Statistische Lerntheorie” (Vapnik 2010)

Explanatorische Modellierung vs. Prädiktive Modellierung

Explanatorische Modellierung ⇔ Wissenschaft

Bestimmung von $\hat{f} \coloneqq \text{argmin} \| f - \hat{f} \|$



Bestimmung von $\tilde{f} \coloneqq \text{argmin} \| Y - \tilde{f}(X) \|$

Prädiktive Modellierung ⇔ Anwendung

- Es gibt keinen Grund anzunehmen, dass immer $\tilde{f} = \hat{f}$ gilt.
- Für ein Beispiel mit $\tilde{f} \neq \hat{f}$. , siehe z.B. Shmueli (2010).

Technik der Prädiktiven Modellierung

Das Lernen von Modellparametern ist das zentrale Problem der Prädiktiven Modellierung

- Parameterlernen impliziert die Minimierung einer Zielfunktion (objective function).
- An der Stelle eines Minimums ist die erste Ableitung einer Funktion gleich Null.
- An der Stelle eines Minimums ist die zweite Ableitung einer Funktion positiv.

Optimierungsverfahren identifizieren Parameterwerte, für die diese Minimumsbedingungen gelten.

Modell	Optimierungsverfahren
Lineare Diskriminanzanalyse	Analytische Optimierung
Logistische Regression	Gradientenverfahren
Support Vektor Maschinen	Optimierung mit Nebenbedingungen
Neuronale Netze	Gradientenverfahren

Anmerkungen

- In der statistischen Modellierung hat die Zielfunktion meist probabilistische Konnotation.
- Typische Zielfunktionen der statistischen Modellierung sind Likelihood Funktionen.
- Prädiktive Modellierung verzichtet zum Teil auf die Repräsentation von Unsicherheit.

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Definition (Funktionenarten)

In der statistischen Anwendung unterscheiden wir

- *univariate reellwertige Funktionen* der Form

$$f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x), \tag{1}$$

- *multivariate reellwertige Funktionen* der Form

$$f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x) = f(x_1, ..., x_n), \tag{2}$$

- *multivariate vektorwertige Funktionen* der Form

$$f : \mathbb{R}^n \to \mathbb{R}^m, x \mapsto f(x) = \begin{pmatrix} f_1(x_1, ..., x_n) \\ \vdots \\ f_m(x_1, ..., x_n) \end{pmatrix}. \tag{3}$$

Bemerkung

- In der Physik werden multivariate reellwertige Funktionen auch *Skalarfelder* genannt.
- In der Physik werden multivariate vektorwertige Funktionen auch *Vektorfelder* genannt.

In diesem Abschnitt betrachten wir univariate reellwertige Funktionen.

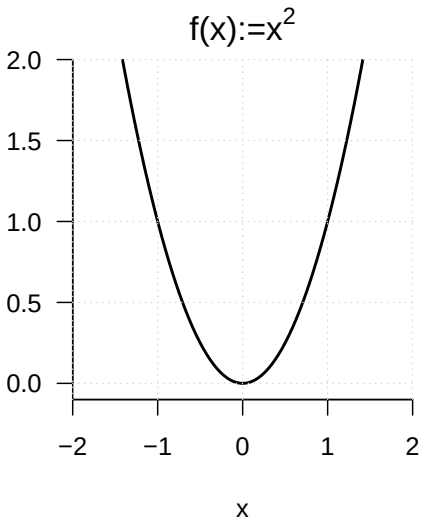
Beispiele

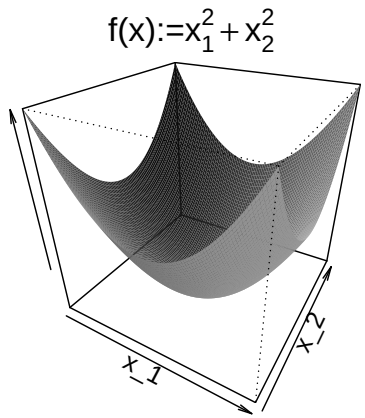
Univariate, reellwertige Funktion

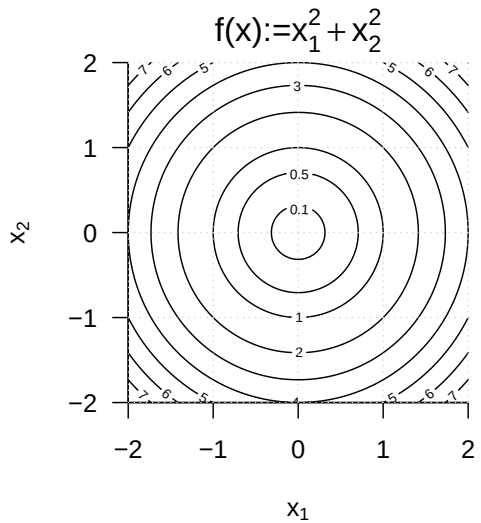
$$f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x) := x^2 \tag{4}$$

Multivariate (bivariate), reellwertige Funktion

$$f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2 \tag{5}$$







Definition (Ableitung)

Es sei $I \subseteq \mathbb{R}$ ein Intervall und

$$f : I \to \mathbb{R}, x \mapsto f(x) \tag{6}$$

eine univariate reellwertige Funktion. f heißt in $a \in I$ *differenzierbar*, wenn der Grenzwert

$$f'(a) := \lim_{h \to 0} \frac{f(a+h) - f(a)}{h} \tag{7}$$

existiert. $f'(a)$ heißt dann die *Ableitung von f an der Stelle a*. Ist f differenzierbar für alle $x \in I$, so heißt f *differenzierbar* und die Funktion

$$f' : I \to \mathbb{R}, x \mapsto f'(x) \tag{8}$$

heißt *Ableitung von f*

Bemerkungen

- Für $h > 0$ heißt $\frac{f(a+h)-f(a)}{h}$ *Differenzquotient*.
- Der Differenzquotient misst die Änderung $f(a+h) - f(a)$ von f pro Strecke h.
- Für $h \to 0$ misst der Differenzquotient die Änderungsrate von f in a.
- $f'(a)$ ist eine Zahl, f' ist eine Funktion.
- Wir werden keine Grenzwertbildung zur Berechnung von Ableitungen benötigen.

Definition (Notation für Ableitungen univariater reellwertiger Funktionen)

Es sei f eine univariate reellwertige Funktion. Äquivalente Schreibweisen für die Ableitung von f und die Ableitung von f an einer Stelle x sind

(1) die *Lagrange-Notation* f' und $f'(x)$,

(2) die *Newton-Notation* $\dot{f}$ und $\dot{f}(x)$,

(3) die *Leibniz-Notation* $\frac{df}{dx}$ und $\frac{df(x)}{dx}$ und

(4) die *Euler-Notation* Df und $Df(x)$.

Bemerkungen

- Für univariate reellwertige Funktionen benutzen wir f' und $f'(x)$ als Bezeichner.
- In Berechnungen benutzen wir auch die "Operator-Schreibweise" $\frac{d}{dx} f(x)$.
- Wir verstehen $\frac{d}{dx} f(x)$ als den Auftrag, die Ableitung von f zu berechnen.

Definition (Höhere Ableitungen)

Es sei f eine univariate reellwertige Funktion und

$$f^{(1)} := f' \tag{9}$$

sei die Ableitung von f. Die k-te Ableitung von f ist rekursiv definiert durch

$$f^{(k)} := \left(f^{(k-1)}\right)' \text{ für } k \geq 0, \tag{10}$$

unter der Annahme, dass $f^{(k-1)}$ differenzierbar ist. Insbesondere ist die *zweite Ableitung von* f definiert durch die Ableitung von f', also

$$f'' := (f')'. \tag{11}$$

Bemerkungen

- Wir schreiben auch $\frac{d^2}{dx^2} f(x)$ für den Auftrag, die zweite Ableitung von f zu bestimmen.
- Die nullte Ableitung $f^{(0)}$ von f ist f selbst.
- Üblicherweise schreibt man für $k < 4$ f', f'', f''' statt $f^{(1)}$, $f^{(2)}$, $f^{(3)}$.
- Im Allgemeinen benötigen wir nur f' und f''.

Theorem (Rechenregeln für Ableitungen)

Für $i = 1, \ldots, n$ seien g_i reellwertige univariate differenzierbare Funktionen. Dann gelten folgende Rechenregeln:

(1) Summenregel

$$\text{Für } f(x) := \sum_{i=1}^{n} g_i(x) \text{ gilt } f'(x) = \sum_{i=1}^{n} g_i'(x). \tag{12}$$

(2) Produktregel

$$\text{Für } f(x) := g_1(x)g_2(x) \text{ gilt } f'(x) = g_1'(x)g_2(x) + g_1(x)g_2'(x). \tag{13}$$

(3) Quotientenregel

$$\text{Für } f(x) := \frac{g_1(x)}{g_2(x)} \text{ gilt } f'(x) = \frac{g_1'(x)g_2(x) - g_1(x)g_2'(x)}{g_2^2(x)}. \tag{14}$$

(4) Kettenregel

$$\text{Für } f(x) := g_1(g_2(x)) \text{ gilt } f'(x) = g_1'(g_2(x))g_2'(x). \tag{15}$$

Bemerkung

- Für Beweise der Rechenregeln wird auf die einschlägige Literatur verwiesen.

Theorem (Ableitungen elementarer Funktionen)

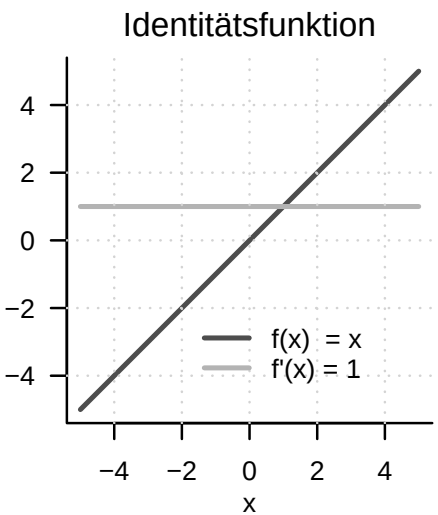
Für einige elementare Funktionen der Datenanalyse ergeben sich folgende Ableitungen:

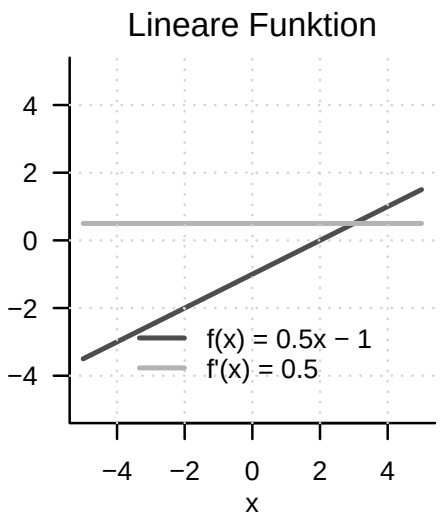
Name	Definition	Ableitung
Polynomfunktionen	$f(x) := \sum_{i=0}^{n} a_i x^i$	$f(x) = \sum_{i=1}^{n} i a_i x^{i-1}$
Konstante Funktion	$f(x) := a$	$f'(x) = 0$
Identitätsfunktion	$f(x) := x$	$f'(x) = 1$
Lineare Funktion	$f(x) := ax + b$	$f'(x) = a$
Quadratfunktion	$f(x) := x^2$	$f'(x) = 2x$
Exponentialfunktion	$f(x) := \exp(x)$	$f'(x) = \exp(x)$
Logarithmusfunktion	$f(x) := \ln(x)$	$f'(x) = \frac{1}{x}$

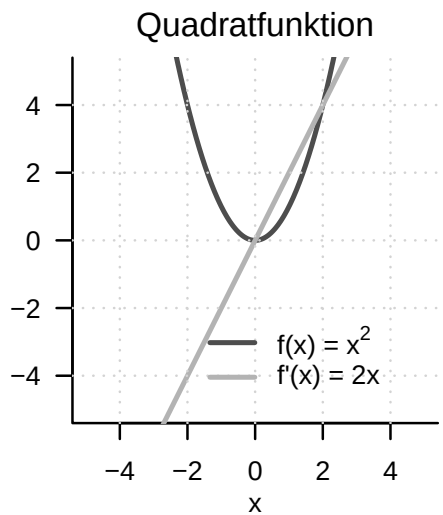
Bemerkung

- Für Beweise wird auf die einschlägige Literatur verwiesen.

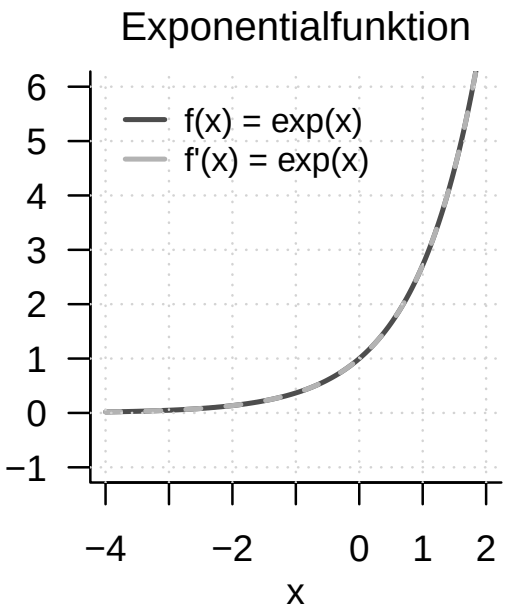
Ableitungen elementarer Funktionen

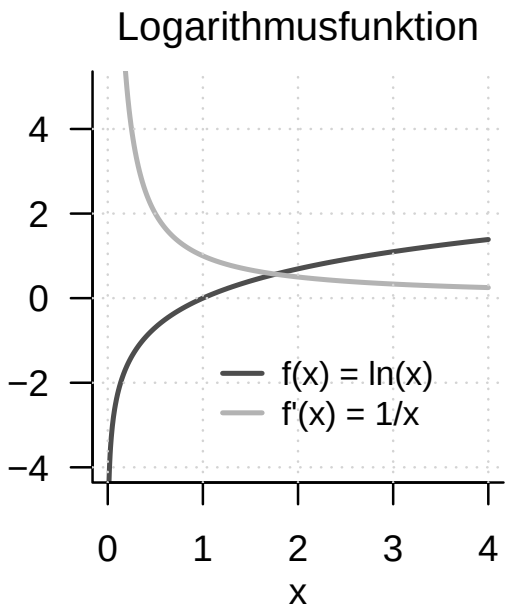






Ableitungen elementarer Funktionen





Definition (Extremstellen und Extremwerte)

Es sei $U \subseteq \mathbb{R}$ und $f : U \to \mathbb{R}$ eine univariate reellwertige Funktion. Dann hat f an der Stelle $x_0 \in U$

- ein *lokales Minimum*, wenn es ein Intervall $I :=]a, b[$ gibt mit $x_0 \in]a, b[$ und
$$f(x_0) \leq f(x) \text{ für alle } x \in I \cap U, \tag{16}$$
- ein *globales Minimum*, wenn gilt, dass
$$f(x_0) \leq f(x) \text{ für alle } x \in U, \tag{17}$$
- ein *lokales Maximum*, wenn es ein Intervall $I :=]a, b[$ gibt mit $x_0 \in]a, b[$ und
$$f(x_0) \geq f(x) \text{ für alle } x \in I \cap U, \tag{18}$$
- *lokales Maximum*, wenn gilt, dass
$$f(x_0) \geq f(x) \text{ für alle } x \in U. \tag{19}$$

Der Wert $x_0 \in U$ der Definitionsmenge von f heißt entsprechend *lokale* oder *globale Minimalstelle* oder *Maximalstelle*, der Funktionswert $f(x_0) \in \mathbb{R}$ heißt entsprechend *lokales* oder *globales Minimum* oder *Maximum*. Generell heißt der Wert $x_0 \in U$ *Extremstelle* und der Funktionswert $f(x_0) \in \mathbb{R}$ *Extremwert*.

Bemerkungen

- Extremstellen werden auch mit $\arg\min_{x \in I \cap U} f(x)$ oder $\arg\max_{x \in I \cap U} f(x)$ bezeichnet.
- Extremwerte werden auch mit $\min_{x \in I \cap U} f(x)$ oder $\max_{x \in I \cap U} f(x)$ bezeichnet.

Definition (Notwendige Bedingung für Extrema)

f sei eine univariate reellwertige Funktion. Dann gilt

$$x_0 \text{ ist Extremstelle von } f \Rightarrow f'(x_0) = 0. \tag{20}$$

Bemerkungen

- Wenn x_0 eine Extremstelle von f ist, dann ist die erste Ableitung von f in x_0 null.
- Sei zum Beispiel x_0 eine lokale Maximalstelle von f. Dann gilt
  - Links von x_0 steigt f an, rechts von x_0 fällt f ab.
  - In x_0 steigt f weder an, noch fällt f ab, also ist $f'(x_0) = 0$.

Definition (Hinreichende Bedingungen für lokale Extrema)

f sei eine zweimal differenzierbare univariate reellwertige Funktion.

- Wenn für $x_0 \in U \subseteq \mathbb{R}$

$$f'(x_0) = 0 \text{ und } f''(x_0) > 0 \tag{21}$$

  gilt, dann hat f an der Stelle x_0 ein Minimum.

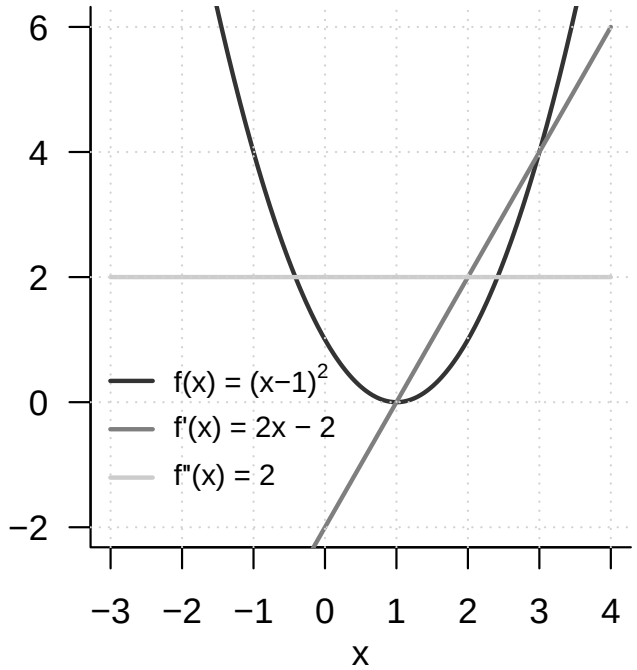
- Wenn für $x_0 \in U \subseteq \mathbb{R}$

$$f'(x_0) = 0 \text{ und } f''(x_0) < 0 \tag{22}$$

  gilt, dann hat f an der Stelle x_0 ein Maximum.

Bemerkung

- Eine Intuition vermittelt nachfolgende Abbildung.



Hier ist offenbar $x_0 = 1$ eine lokale Minimalstelle von $f(x) = (x-1)^2$. Man erkennt:

- Links von x_0 fällt f ab, rechts von x_0 steigt f an.
- In x_0 steigt f weder an, noch fällt f ab, also ist $f'(x_0) = 0$.
- Links und rechts von x_0 und in x_0 ist die Änderung f'' von f' positiv.
- Links von x_0 schwächt sich die Negativität von f' zu 0 ab.
- Rechts von x_0 verstärkt sich die Positivität von f'.

Definition (Standardverfahren der analytischen Optimierung)

f sei eine univariate reellwertige Funktion. Lokale Extremstellen von f können mit folgendem *Standardverfahren der analytischen Optimierung* identifiziert werden:

(1) Berechnen der ersten und zweiten Ableitung von f.

(2) Bestimmen von Nullstellen x^* von f' durch Auflösen von $f'(x^*) = 0$ nach x^*.

$\Rightarrow$ Nullstellen von f' sind Kandidaten für Extremstellen von f.

(3) Evaluation von $f''(x^*)$.

$\Rightarrow$ Wenn $f''(x^*) > 0$, dann ist x^* lokale Minimumstelle von f.

$\Rightarrow$ Wenn $f''(x^*) < 0$, dann ist x^* lokale Maximumstelle von f.

$\Rightarrow$ Wenn $f''(x^*) = 0$, dann ist x^* keine Extremstelle von f.

Beispiel

Wir betrachten die Funktion

$$f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x) := (x-1)^2. \tag{23}$$

Die erste Ableitung von f ergibt sich mit der Kettenregel zu

$$f'(x) = \frac{d}{dx}\left((x-1)^2\right) = 2(x-1) \cdot \frac{d}{dx}(x-1) = 2x - 2. \tag{24}$$

Die zweite Ableitung von f ergibt sich zu

$$f''(x) = \frac{d}{dx} f'(x) = \frac{d}{dx}(2x-2) = 2 > 0 \text{ für alle } x \in \mathbb{R}. \tag{25}$$

Auflösen von $f'(x^*) = 0$ nach x^* ergibt

$$f'(x^*) = 0 \Leftrightarrow 2x^* - 2 = 0 \Leftrightarrow 2x^* = 2 \Leftrightarrow x^* = 1. \tag{26}$$

$x^* = 1$ ist folglich eine Minimalstelle von f mit zugehörigen Minimalwert $f(1) = 0$.

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Definition (Funktionenarten)

In der statistischen Anwendung unterscheiden wir

- *univariate reellwertige Funktionen* der Form

$$f : \mathbb{R} \to \mathbb{R}, x \mapsto f(x), \tag{27}$$

- *multivariate reellwertige Funktionen* der Form

$$f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x) = f(x_1, ..., x_n), \tag{28}$$

- *multivariate vektorwertige Funktionen* der Form

$$f : \mathbb{R}^n \to \mathbb{R}^m, x \mapsto f(x) = \begin{pmatrix} f_1(x_1, ..., x_n) \\ \vdots \\ f_m(x_1, ..., x_n) \end{pmatrix}. \tag{29}$$

In diesem Abschnitt betrachten wir multivariate reellwertige Funktionen.

Beim Parameterlernen der Prädiktiven Modellierung ist

$x \in \mathbb{R}^n$ **ein Vektor von Parameterwerten und** $f : \mathbb{R}^n \to \mathbb{R}$ **eine Zielfunktion**.

Definition (Partielle Ableitung)

Es sei $D \subseteq \mathbb{R}^n$ eine Menge und

$$f : D \to \mathbb{R}, x \mapsto f(x) \tag{30}$$

eine multivariate reellwertige Funktion. f heißt in $x \in D$ *nach* x_i *partiell differenzierbar*, wenn der Grenzwert

$$\frac{\partial}{\partial x_i} f(x) := \lim_{h \to 0} \frac{f(x + he_i) - f(x)}{h} \tag{31}$$

existiert. $\frac{\partial}{\partial x_i} f(x)$ heißt dann die *partielle Ableitung von* f *nach* x_i *an der Stelle* x. Wenn f für alle $x \in D$, nach x_i partiell differenzierbar ist, dann heißt f *nach* x_i *partiell differenzierbar* und die Funktion

$$\frac{\partial}{\partial x_i} f : D \to \mathbb{R}, x \mapsto \frac{\partial}{\partial x_i} f(x) \tag{32}$$

heißt *partielle Ableitung von* f *nach* x_i.

f heißt *partiell differenzierbar in* $x \in D$, wenn f für alle $i = 1, \ldots, n$ in $x \in D$ nach x_i partiell differenzierbar ist, und f heißt partiell differenzierbar, wenn f für alle $i = 1, \ldots, n$ in allen $x \in D$ nach x_i partiell differenzierbar ist.

Bemerkungen

- $e_i \in \mathbb{R}^n$ bezeichnet den $i$ten Einheitsvektor.
- $\frac{f(x+he_i)-f(x)}{h}$ misst die Änderung $f(x + he_i) - f(x)$ von f pro Strecke h in Richtung e_i.
- Für $h \to 0$ misst der Differenzquotient die Änderungsrate von f in x in Richtung e_i.
- $\frac{\partial}{\partial x_i} f(x)$ ist eine Zahl, $\frac{\partial}{\partial x_i} f$ ist eine Funktion.
- Praktisch berechnet man $\frac{\partial}{\partial x_i} f$ als die (einfache) Ableitung

$$\frac{d}{dx_i} \tilde{f}_{x_1,\dots x_{i-1},x_{i+1},\dots,x_n}(x_i) \tag{33}$$

der univariaten reellwertigen Funktion

$$\tilde{f} : \mathbb{R} \to \mathbb{R}, x_i \mapsto \tilde{f}_{x_1,\dots x_{i-1},x_{i+1},\dots,x_n}(x_i) := f(x_1, ..., x_i, ..., x_n). \tag{34}$$

- Man betrachtet alle x_j mit $j \neq i$ also als Konstanten.

Beispiel (1)

Wir betrachten die Funktion

$$f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2. \tag{35}$$

Weil die Definitionsmenge dieser Funktion zweidimensional ist, kann man zwei partielle Ableitungen berechnen

$$\frac{\partial}{\partial x_1} f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto \frac{\partial}{\partial x_1} f(x) \text{ und } \frac{\partial}{\partial x_2} f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto \frac{\partial}{\partial x_2} f(x). \tag{36}$$

Um die erste dieser partiellen Ableitungen zu berechnen, betrachtet man die Funktion

$$f_{x_2} : \mathbb{R} \to \mathbb{R}, x_1 \mapsto f_{x_2}(x_1) := x_1^2 + x_2^2, \tag{37}$$

wobei x_2 hier die Rolle einer Konstanten einnimmt. Um explizit zu machen, dass x_2 kein Argument der Funktion ist, die Funktion aber weiterhin von x_2 abhängt haben wir die Subskriptnotation $f_{x_2}(x_1)$ verwendet. Um nun die partielle Ableitung zu berechnen, berechnen wir die (einfache) Ableitung von f_{x_2},

$$f'_{x_2}(x) = 2x_1. \tag{38}$$

Es ergibt sich also

$$\frac{\partial}{\partial x_1} f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto \frac{\partial}{\partial x_1} f(x) = \frac{\partial}{\partial x_1}(x_1^2 + x_2^2) = f'_{x_2}(x) = 2x_1. \tag{39}$$

Analog gilt mit der entsprechenden Formulierung von f_{x_1}, dass

$$\frac{\partial}{\partial x_2} f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto \frac{\partial}{\partial x_2} f(x) = \frac{\partial}{\partial x_2}(x_1^2 + x_2^2) = f'_{x_1}(x) = 2x_2. \tag{40}$$

Definition (Zweite partielle Ableitungen)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine multivariate reellwertige Funktion und $\frac{\partial}{\partial x_i} f$ sei die partielle Ableitung von f nach x_i. Dann ist die zweite partielle Ableitung von f nach x_i und x_j definiert als

$$\frac{\partial^2}{\partial x_j x_i} f(x) := \frac{\partial}{\partial x_j} \left(\frac{\partial}{\partial x_i} f \right) \tag{41}$$

Bemerkungen

- Wie die zweite Ableitung ist auch die zweite partielle Ableitung rekursiv definiert.
- Zu jeder partiellen Ableitung $\frac{\partial}{\partial x_i} f$ gibt es n zweite partiellen Ableitungen $\frac{\partial^2}{\partial x_j \partial x_i} f, j = 1, \ldots, n$.

Theorem (Satz von Schwarz)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine partiell differenzierbare multivariate reellwertige Funktion. Dann gilt

$$\frac{\partial^2}{\partial x_j \partial x_i} f(x) = \frac{\partial^2}{\partial x_i \partial x_j} f(x) \text{ für alle } 1 \leq i, j \leq n. \tag{42}$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Das Theorem von Schwarz besagt, dass die Reihenfolge des partiellen Ableitens irrelevant ist
- Das Theorem erleichtert die Berechnung von zweiten partiellen Ableitungen.
- Das Theorem hilft, Fehler bei der Berechnung zweiter partieller Ableitungen aufzudecken.

Beispiel (1) (fortgeführt)

Wir wollen die partiellen Ableitungen zweiter Ordnung der Funktion

$$f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2. \tag{43}$$

berechnen. Mit den Ergebnissen für die partiellen Ableitungen erster Ordnung dieser Funktion ergibt sich

$$\begin{aligned}
\frac{\partial^2}{\partial x_1 x_1} f(x) &= \frac{\partial}{\partial x_1} \left(\frac{\partial}{\partial x_1} f(x) \right) = \frac{\partial}{\partial x_1} (2x_1) = 2 \\
\frac{\partial^2}{\partial x_1 x_2} f(x) &= \frac{\partial}{\partial x_1} \left(\frac{\partial}{\partial x_2} f(x) \right) = \frac{\partial}{\partial x_1} (2x_2) = 0 \\
\frac{\partial^2}{\partial x_2 x_1} f(x) &= \frac{\partial}{\partial x_2} \left(\frac{\partial}{\partial x_1} f(x) \right) = \frac{\partial}{\partial x_2} (2x_1) = 0 \\
\frac{\partial^2}{\partial x_2 x_2} f(x) &= \frac{\partial}{\partial x_2} \left(\frac{\partial}{\partial x_2} f(x) \right) = \frac{\partial}{\partial x_2} (2x_2) = 2
\end{aligned} \tag{44}$$

Offenbar gilt

$$\frac{\partial^2}{\partial x_1 x_2} f(x) = \frac{\partial^2}{\partial x_2 x_1} f(x). \tag{45}$$

Beispiel (2)

Wir wollen die partiellen Ableitungen erster und zweiter Ordnung der Funktion

$$f : \mathbb{R}^3 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_1 x_2 + x_2 \sqrt{x_3}. \tag{46}$$

berechnen.

Mit den Rechenregeln für Ableitungen ergibt sich für die partiellen Ableitungen erster Ordnung

$$\begin{aligned}
\frac{\partial}{\partial x_1} f(x) &= \frac{\partial}{\partial x_1} \left(x_1^2 + x_1 x_2 + x_2 \sqrt{x_3} \right) = 2x_1 + x_2, \\
\frac{\partial}{\partial x_2} f(x) &= \frac{\partial}{\partial x_2} \left(x_1^2 + x_1 x_2 + x_2 \sqrt{x_3} \right) = x_1 + \sqrt{x_3}, \\
\frac{\partial}{\partial x_3} f(x) &= \frac{\partial}{\partial x_3} \left(x_1^2 + x_1 x_2 + x_2 \sqrt{x_3} \right) = \frac{x_2}{2\sqrt{x_3}}.
\end{aligned} \tag{47}$$

Beispiel (2) (fortgeführt)

Für die zweiten partiellen Ableitungen hinsichtlich x_1 ergibt sich

$$\begin{aligned}
\frac{\partial^2}{\partial x_1 \partial x_1} f(x) &= \frac{\partial}{\partial x_1}\left(\frac{\partial}{\partial x_1} f(x)\right) = \frac{\partial}{\partial x_1}(2x_1 + x_2) = 2, \\
\frac{\partial^2}{\partial x_2 \partial x_1} f(x) &= \frac{\partial}{\partial x_2}\left(\frac{\partial}{\partial x_1} f(x)\right) = \frac{\partial}{\partial x_2}(2x_1 + x_2) = 1, \\
\frac{\partial^2}{\partial x_3 \partial x_1} f(x) &= \frac{\partial}{\partial x_3}\left(\frac{\partial}{\partial x_1} f(x)\right) = \frac{\partial}{\partial x_3}(2x_1 + x_2) = 0.
\end{aligned} \tag{48}$$

Für die zweiten partiellen Ableitungen hinsichtlich x_2 ergibt sich

$$\begin{aligned}
\frac{\partial^2}{\partial x_1 \partial x_2} f(x) &= \frac{\partial}{\partial x_1}\left(\frac{\partial}{\partial x_2} f(x)\right) = \frac{\partial}{\partial x_1}(x_1 + \sqrt{x_3}) = 1, \\
\frac{\partial^2}{\partial x_2 \partial x_2} f(x) &= \frac{\partial}{\partial x_2}\left(\frac{\partial}{\partial x_2} f(x)\right) = \frac{\partial}{\partial x_2}(x_1 + \sqrt{x_3}) = 0, \\
\frac{\partial^2}{\partial x_3 \partial x_2} f(x) &= \frac{\partial}{\partial x_3}\left(\frac{\partial}{\partial x_2} f(x)\right) = \frac{\partial}{\partial x_3}(x_1 + \sqrt{x_3}) = \frac{1}{2\sqrt{x_3}}.
\end{aligned} \tag{49}$$

Beispiel (2) (fortgeführt)

Für die zweiten partiellen Ableitungen hinsichtlich x_3 ergibt sich

$$\begin{aligned}
\frac{\partial^2}{\partial x_1 \partial x_3} f(x) &= \frac{\partial}{\partial x_1}\left(\frac{\partial}{\partial x_3} f(x)\right) = \frac{\partial}{\partial x_1}\left(\frac{x_2}{2}\sqrt{x_3}\right) = 0, \\
\frac{\partial^2}{\partial x_2 \partial x_3} f(x) &= \frac{\partial}{\partial x_2}\left(\frac{\partial}{\partial x_3} f(x)\right) = \frac{\partial}{\partial x_2}\left(\frac{x_2}{2\sqrt{x_3}}\right) = \frac{1}{2\sqrt{x_3}}, \\
\frac{\partial^2}{\partial x_3 \partial x_3} f(x) &= \frac{\partial}{\partial x_3}\left(\frac{\partial}{\partial x_3} f(x)\right) = \frac{\partial}{\partial x_3}\left(x_2 \frac{1}{2} x_3^{-\frac{1}{2}}\right) = -\frac{1}{4} x_2 x_3^{-\frac{3}{2}}.
\end{aligned} \tag{50}$$

Weiterhin erkennt man, dass die Reihenfolge der partiellen Ableitungen irrelevant ist, denn es gilt

$$\begin{aligned}
\frac{\partial^2}{\partial x_1 \partial x_2} f(x) &= \frac{\partial^2}{\partial x_2 \partial x_1} f(x) = 1, \\
\frac{\partial^2}{\partial x_1 \partial x_3} f(x) &= \frac{\partial^2}{\partial x_3 \partial x_1} f(x) = 0, \\
\frac{\partial^2}{\partial x_2 \partial x_3} f(x) &= \frac{\partial^2}{\partial x_3 \partial x_2} f(x) = \frac{1}{2\sqrt{x_3}}.
\end{aligned} \tag{51}$$

Definition (Gradient)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine multivariate reellwertige Funktion. Dann ist der *Gradient* $\nabla f(x)$ von f an der Stelle $x \in \mathbb{R}^n$ definiert als

$$\nabla f(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{pmatrix} \in \mathbb{R}^n. \tag{52}$$

Bemerkung

- $\nabla f(x)$ fasst die partiellen Ableitungen von f an der Stelle $x \in \mathbb{R}^n$ in einem Vektor zusammen.
- Gradienten sind multivariate vektorwertige Abbildungen der Form $\nabla f : \mathbb{R}^n \to \mathbb{R}^n, x \mapsto \nabla f(x)$.
- Wir zeigen später, dass $-\nabla f(x)$ die Richtung des steilsten Abstiegs von f in $\mathbb{R}^n$ anzeigt.
- Für $n = 1$ gilt $\nabla f(x) = f'(x)$.

Beispiele

Für die in Beispiel (1) betrachtete Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ gilt

$$\nabla f(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \in \mathbb{R}^2. \tag{53}$$

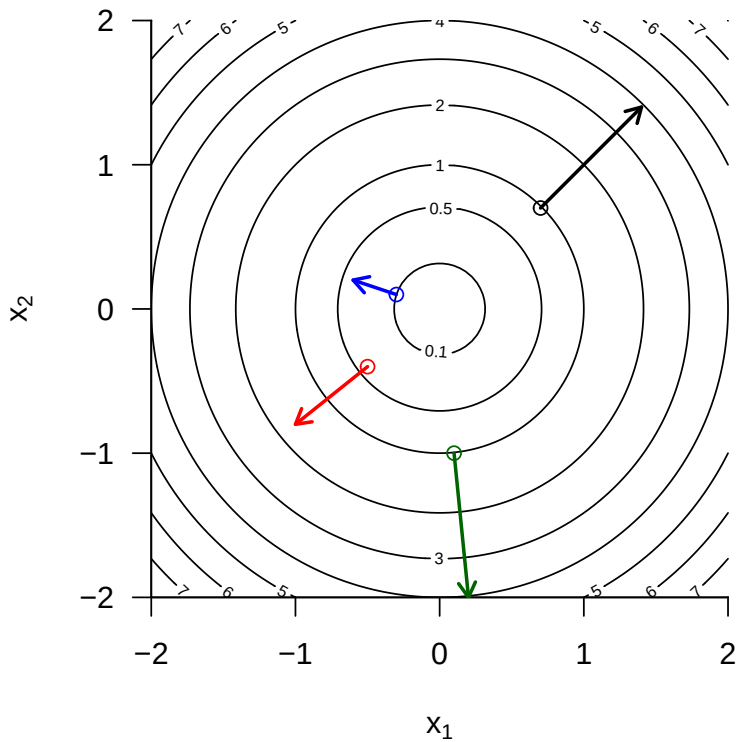
Für die in Beispiel (2) betrachtete Funktion $f : \mathbb{R}^3 \to \mathbb{R}$ gilt

$$\nabla f(x) := \begin{pmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \frac{\partial}{\partial x_3} f(x) \end{pmatrix} = \begin{pmatrix} 2x_1 + x_2 \\ x_1 + \sqrt{x_3} \\ \frac{x_2}{2\sqrt{x_3}} \end{pmatrix} \in \mathbb{R}^3. \tag{54}$$

Beispiel (1) (fortgeführt)

Gradienten von $f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2$ bei

$$x = \begin{pmatrix} 0.7 \\ 0.7 \end{pmatrix} \qquad x = \begin{pmatrix} -0.3 \\ 0.1 \end{pmatrix} \qquad x = \begin{pmatrix} -0.5 \\ -0.4 \end{pmatrix} \qquad x = \begin{pmatrix} 0.1 \\ -1.0 \end{pmatrix}$$



Definition (Hesse-Matrix)

$f : \mathbb{R}^n \to \mathbb{R}$ sei ein multivariate reellwertige Funktion. Dann ist die *Hesse-Matrix* $\nabla^2 f(x)$ von f an der Stelle $x \in \mathbb{R}^n$ definiert als

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2}{\partial x_1 x_1} f(x) & \frac{\partial^2}{\partial x_1 x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_1 x_n} f(x) \\ \frac{\partial^2}{\partial x_2 x_1} f(x) & \frac{\partial^2}{\partial x_2 x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_2 x_n} f(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_n x_1} f(x) & \frac{\partial^2}{\partial x_n x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_n x_n} f(x) \end{pmatrix} \in \mathbb{R}^{n \times n}. \tag{55}$$

Bemerkung

- $\nabla^2 f(x)$ fasst die partiellen Ableitungen zweiter Ordnung von f in einer Matrix zusammen.
- Hesse-Matrizen sind multivariate matrixwertige Abbildungen der Form $\nabla^2 f : \mathbb{R}^n \to \mathbb{R}^{n \times n}, x \mapsto \nabla^2 f(x)$.
- Für $n = 1$ gilt $\nabla^2 f(x) = f''(x)$.
- Mit $\frac{\partial^2}{\partial x_i \partial x_j} f(x) = \frac{\partial^2}{\partial x_j \partial x_i} f(x)$ für $1 \leq i, j \leq n$ folgt, dass $\left(\nabla^2 f(x)\right)^T = \nabla^2 f(x)$.

Beispiel

Für die in Beispiel (1) betrachtete Funktion $f : \mathbb{R}^2 \to \mathbb{R}$ gilt

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2}{\partial x_1 x_1} f(x) & \frac{\partial^2}{\partial x_1 x_2} f(x) \\ \frac{\partial^2}{\partial x_2 x_1} f(x) & \frac{\partial^2}{\partial x_2 x_2} f(x) \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

Für die in Beispiel (2) betrachtete Funktion $f : \mathbb{R}^3 \to \mathbb{R}$ gilt

$$\begin{aligned} \nabla^2 f(x) &:= \begin{pmatrix} \frac{\partial^2}{\partial x_1 x_1} f(x) & \frac{\partial^2}{\partial x_1 x_2} f(x) & \frac{\partial^2}{\partial x_1 x_3} f(x) \\ \frac{\partial^2}{\partial x_2 x_1} f(x) & \frac{\partial^2}{\partial x_2 x_2} f(x) & \frac{\partial^2}{\partial x_2 x_3} f(x) \\ \frac{\partial^2}{\partial x_3 x_1} f(x) & \frac{\partial^2}{\partial x_3 x_2} f(x) & \frac{\partial^2}{\partial x_3 x_3} f(x) \end{pmatrix} \\ &:= \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & \frac{1}{2\sqrt{3}} \\ 0 & \frac{1}{2\sqrt{3}} & -\frac{1}{4} x_2 x_3^{-3/2} \end{pmatrix} \end{aligned}$$

Definition (Glatte multivariate reellwertige Funktion)

Eine multivariate reellwertige Funktion

$$f : \mathbb{R}^n \to \mathbb{R}, x \mapsto f(x) \tag{56}$$

heißt *glatt*, wenn ihr Gradient und ihre Hesse-Matrix existieren und für alle $x \in \mathbb{R}^n$ stetig sind.

Bemerkungen

- Der Gradient und die Hesse-Matrix einer glatten Funktion könnten überall in $\mathbb{R}^n$ berechnet werden.

Theorem (Multivariater Mittelwertsatz erster Ordnung)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion und es sei $p \in \mathbb{R}^n$. Dann gibt es ein $t \in]0, 1[$, so dass gilt

$$f(x + p) = f(x) + \nabla f(x + tp)^T p. \tag{57}$$

Theorem (Multivariater Mittelwertsatz zweiter Ordnung)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion und es sei $p \in \mathbb{R}^n$. Dann gibt es ein $t \in]0, 1[$, so dass gilt

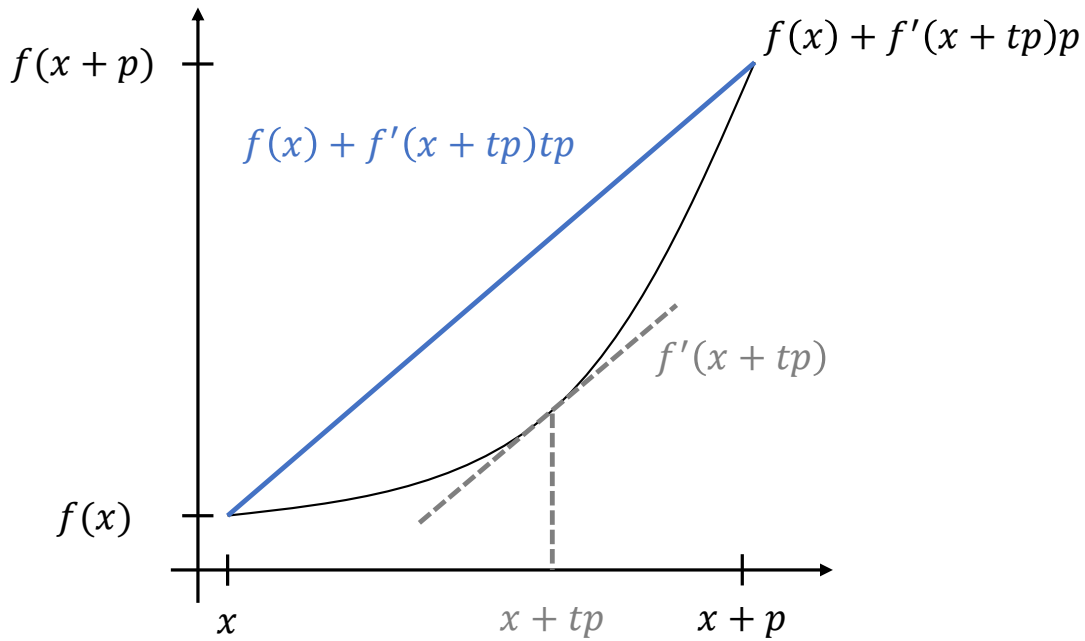
$$f(x + p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x + tp) p. \tag{58}$$

Bemerkung

- Wir verzichten auf Beweise.
- Nocedal and Wright (2006) bezeichnen die Theoreme als “Taylor’s Theorem”, das ist ein wenig misleading.
- ∇f und $\nabla^2 f$ werden an einer Stelle zwischen x und $x + p$ evaluiert.

Univariate visuelle Intuition zum Mittelwertsatz 1. Ordnung

$$\text{Es gibt ein } t \in]0,1[\text{ mit } f(x+p) = f(x) + f'(x+tp)p \tag{59}$$



Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Definition (Optimierungsproblem)

Ein *Optimierungsproblem* hat die allgemeine Form

$$\min_x f(x), \tag{60}$$

wobei $x \in \mathbb{R}^n$ und $f : \mathbb{R}^n \to \mathbb{R}$ eine glatte multivariate reellwertige Funktion ist. Weil gilt, dass

$$\max_x f(x) = \min_x -f(x) \tag{61}$$

genügt es, sich mit Minimierungsproblemen zu befassen.

Definition (Globale und lokale Minimierer, globale und lokale Minima)

$f : \mathbb{R}^n \to \mathbb{R}$ sei ein multivariate reellwertige Funktion.

- $x^* \in \mathbb{R}^n$ heißt globaler Minimalstelle von f, wenn $f(x^*) \leq f(x)$ für alle $x \in \mathbb{R}^n$ gilt. $f(x^*) \in \mathbb{R}$ heißt das globale Minimum von f.

- $x^* \in \mathbb{R}^n$ heißt lokale Minimalstelle von f, wenn es eine Umgebung N von x^* gibt, so dass $f(x^*) \leq f(x)$ für alle $x \in N \subset \mathbb{R}^n$. In diesem Fall heißt $f(x^*) \in \mathbb{R}$ lokales Minimum von f.

- $x^* \in \mathbb{R}^n$ heißt strikte lokale Minimalstelle von f, wenn es eine Umgebung N von x^* gibt, so dass $f(x^*) < f(x)$ für alle $x \in N \subset \mathbb{R}^n$. In diesem Fall heißt $f(x^*) \in \mathbb{R}$ striktes lokales Minimum von f.

Bemerkung

- Eine Umgebung von $x \in \mathbb{R}^n$ ist eine offene Menge, die x enthält.

Theorem (Notwendige Bedingung erster Ordnung)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion. Wenn x^* eine lokale Minimalstelle von f ist, dann gilt

$$\nabla f(x^*) = 0_n\,. \tag{62}$$

Beweis

Wir beweisen das Theorem mithilfe eines indirekten Beweises (Beweis durch Widerspruch). Dazu nehmen wir an, dass x^* zwar eine lokale Minimalstelle von f ist, aber $\nabla f(x^*) \neq 0_n$ ist. Dazu definieren wir zunächst $p := -\nabla f(x^*)$. Dann gilt, dass

$$p^T \nabla f(x^*) = -\nabla f(x^*)^T \nabla f(x^*) = -||\nabla f(x^*)|| < 0. \tag{63}$$

Weil ∇f in einer Umgebung von x^* stetig ist, existiert ein Skalar $T > 0$, so dass auch

$$p^T \nabla f(x^* + tp) < 0 \text{ für alle } t \in [0, T]. \tag{64}$$

gilt. Nun gilt für $\tilde{t} \in]0, T]$ aber mit dem Mittelwertssatz erster Ordnung, dass

$$f(x^* + \tilde{t}p) = f(x^*) + \nabla f(x^* + tp)^T \tilde{t}p = f(x^*) + \tilde{t}p^T \nabla f(x^* + tp) \text{ für ein } t \in]0, \tilde{t}[. \tag{65}$$

Also folgt $f(x^* + \tilde{t}p) < f(x^*)$ für alle $\tilde{t} \in]0, T]$. Wir haben also eine Richtung von x^* weg gefunden, in der f abnimmt. Also kann x^* keine Minimalstelle sein, wenn $\nabla f(x^*) \neq 0_n$ gilt. Dies ist aber ein Widerspruch, zur Annahme, dass es möglich ist, dass x^* eine lokale Minimalstelle von f ist und $\nabla f(x^*) \neq 0_n$ gilt. Also muss $\nabla f(x^*) = 0_n$ gelten, wenn x^* eine lokale Minimalstelle ist.

Theorem (Notwendige Bedingung zweiter Ordnung)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion. Wenn x^* eine lokale Minimalstelle von f ist, dann ist $\nabla f(x^*) = 0_n$ und $\nabla^2 f(x^*)$ ist positiv semidefinit.

Beweis

Wir beweisen das Theorem mithilfe eines indirekten Beweises (Beweis durch Widerspruch). Wir haben schon gesehen, dass $\nabla f(x^*) = 0_n$ ist, wenn x^* eine lokale Minimalstelle von f. ist. Für einen Widerspruchsbeweis nehmen wir nun an, dass x^* zwar eine lokale Minimalstelle von f ist, aber dass $\nabla^2 f(x^*)$ nicht positiv semidefinit ist. Dann ist es möglichen einen Vektor p zu finden, so dass gilt

$$p^T \nabla^2 f(x^*) p < 0. \tag{66}$$

Weil $\nabla^2 f(x^*)$ in einer Umgebung von x^* stetig ist, existiert ein Skalar $T > 0$, so dass

$$p^T \nabla^2 f(x^* + tp) p < 0 \text{ für alle } t \in [0, T]. \tag{67}$$

gilt. Mithilfe des Mittelwertsatzes zweiter Ordnung gilt dann für alle $\bar{t} \in]0, T]$ und ein $t \in]0, \bar{t}[$, dass

$$f(x^* + \bar{t}p) = f(x^*) + \bar{t} p^T \nabla f(x^*) + \frac{1}{2} \bar{t}^2 p^T \nabla^2 f(x^* + tp) p < f(x^*). \tag{68}$$

Wir haben also wieder eine Richtung von x^* weg gefunden, in der f abnimmt. Also kann x^* keine Minimalstelle sein, wenn $\nabla^2 f(x^*)$ nicht positiv semidefinit ist. Dies ist aber ein Widerspruch, zur Annahme, dass es möglich ist, dass x^* eine lokale Minimalstelle von f ist und $\nabla^2 f(x^*)$ nicht positiv semidefinit ist. Also muss $\nabla^2 f(x^*)$ positiv semidefinit sein, wenn x^* eine lokale Minimalstelle ist.

Theorem (Hinreichende Bedingungen zweiter Ordnung)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion und es seien $\nabla f(x^*) = 0_n$ und $\nabla^2 f(x^*)$ positiv definit. Dann ist x^* eine strikte Minimalstelle von f.

Beweis

Wir halten zunächst fest, dass weil die Hesse-Matrix stetig und positiv definit in x^* ist, wir ein $r > 0$ wählen können, so dass $\nabla^2 f(x)$ positiv definit für alle x in

$$D = \{x | ||x - x^*|| < r\} \tag{69}$$

ist. Für einen Vektor p mit $||p|| > 0$ und $||p|| < r$ gilt $x^* + p \in D$. Für ein $t \in]0, 1[$ gilt dann mit dem Mittelwertssatz zweiter Ordnung, dass

$$\begin{aligned} f(x^* + p) &= f(x^*) + \nabla f(x^*) p^T + \frac{1}{2} p^T \nabla^2 f(x^* + tp) p \\ &= f(x^*) + \frac{1}{2} p^T \nabla^2 f(x^* + tp) p. \end{aligned} \tag{70}$$

Weil aber $x^* + tp \in D$ ist, gilt, dass $p^T \nabla^2 f(x^* + tp) p > 0$ ist und somit $f(x^* + p) > f(x^*)$. In jeder Richtung p von x^* weg erhöht sich also der Wert von f und damit ist x^* eine strikte Minimalstelle.

Theorem (Minmalstellen konvexer Funktionen)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine *konvexe Funktion*, das heißt, für alle $x, y \in \mathbb{R}^n$ gelte

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \text{ für alle } \lambda \in [0,1]. \tag{71}$$

Dann ist eine lokale Minimalstelle x^* von f auch die globale Minimalstelle von f.

Beweis

Wir beweisen das Theorem mithilfe eines indirekten Beweises (Beweis durch Widerspruch). Nehmen wir dazu an, x^* sei eine lokale, aber keine globale Minimalstelle. Dann können wir ein $z \in \mathbb{R}^n$ mit $f(z) < f(x^*)$ finden. Wir betrachten nun die Strecke, die x^* und z in $\mathbb{R}^n$ verbindet, also

$$x = \lambda z + (1-\lambda) x^* \text{ mit } \lambda \in]0,1] \tag{72}$$

Mit der Konvexität von f folgt dann aber, dass

$$f(x) \leq \lambda f(z) + (1-\lambda) f(x^*) < f(x^*) \tag{73}$$

Jede Umgebung N von x^* enthält ein Stück dieser Strecke, also gibt es immer Punkte $x \in N$ mit $f(x) < f(x^*)$. Also ist $x*$ keine lokale Minimalstelle und wir haben einen Widerspruch.

Zusammenfassung

Optimierungsproblem

$$\min_x f(x) = \max_x -f(x) \text{ für } f : \mathbb{R}^n \to \mathbb{R}$$

Lokale Minimalstelle

$$x^* = \arg\min_x f(x), x^* \in \mathbb{R}^n \Leftrightarrow f(x^*) \leq f(x) \text{ für alle } x \in N \subset \mathbb{R}^n$$

Notwendige Bedingung erster Ordnung

$$x^* = \arg\min_x f(x) \Rightarrow \nabla f(x^*) = 0_n$$

Notwendige Bedingung zweiter Ordnung

$$x^* = \arg\min_x f(x) \Rightarrow \nabla f(x^*) = 0_n \text{ und } \nabla^2 f(x) \text{ positiv semidefinit}$$

Hinreichende Bedingung zweiter Ordnung

$$\nabla f(x^*) = 0 \text{ und } \nabla^2 f(x) \text{ positiv definit } \Rightarrow x^* = \arg\min_x f(x)$$

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Allgemeine Form von Optimierungsalgorithmen

Initialisierung

0. Wahl eines Startpunktes $x_0 \in \mathbb{R}^n$.

Iterationen

Für $k = 0, 1, 2, \ldots$

1. Berechnung von x_{k+1} basierend auf Information über f an der Stelle x_k.
2. STOP, wenn Minimalstelle gefunden ist oder kein Fortschritt mehr erzielt wird.

Gradientenverfahren

Initialisierung

0. Wahl von Startpunkt $x_0 \in \mathbb{R}^n$, Lernrate $\alpha > 0$, Konvergenzkriteriums $\delta > 0$.

Iterationen

Für $k = 0, 1, 2, \ldots$

1. Setze $x_{k+1} := x_k - \alpha \nabla f(x_k)$.
2. STOP, wenn $||\nabla f(x_{k+1})|| < \delta$, ansonsten gehe zu 1.

Theorem (Gradientenverfahren)

$f : \mathbb{R}^n \to \mathbb{R}$ sei eine glatte Funktion und es sei $x_k \in \mathbb{R}^n$. Dann ist die Gradientenrichtung

$$p_k^G := -\nabla f(x_k) \tag{74}$$

die Richtung des steilsten Abstiegs von f in x_k.

Bemerkungen

- Es gibt unendliche viele mögliche Richtungen p in x_k.
- $\nabla f(x) \in \mathbb{R}^n$ ist eine Richtung in der Definitionsmenge von f (Parameterraum).
- Die Gradientenrichtung ist davon die Richtung, in der die Zielfunktion f am schnellsten abnimmt.
- Zum Vergleich von Richtungen genügt es, Richtungen der Länge $||p|| = 1$ zu vergleichen.

Beweis

Mit dem Mittelwertsatz zweiter Ordnung gilt für jede Richtung p und Schrittlängenparameter α, dass

$$f(x_k + \alpha p) = f(x_k) + \alpha p^T \nabla f(x_k) + \frac{1}{2}\alpha^2 p^T \nabla^2 f(x_k + tp)p \text{ für ein } t \in]0, \alpha[. \tag{75}$$

Die Änderungsrate von f in Richtung p in x_k ist also der Koeffizient von α, also $p^T \nabla f(x_k)$ (man denke an $x = tv$ für einen Ort x, eine Geschwindigkeit v und eine Zeit t). Also gilt, dass die Richtung des steilsten Abstiegs p in x_k mit Länge 1 die Lösung des Optimierungsproblems

$$\min_p p^T \nabla f(x_k) \text{ mit der Nebenbedingung } ||p|| = 1. \tag{76}$$

ist. Wir erinnern nun zunächst daran, dass für $x, y \in \mathbb{R}^n$ gilt der Kosinus des Winkel zwischen x und y durch

$$\cos\alpha = \frac{\langle x, y\rangle}{||x||||y||} = \frac{x^T y}{||x||||y||} \tag{77}$$

gegeben ist. Damit aber gilt, dass

$$p^T \nabla f(x_k) = ||p|| \cdot ||\nabla f(x_k)|| \cos\theta = 1 \cdot ||\nabla f(x_k)|| \cos\theta = ||\nabla f(x_k)|| \cos\theta \tag{78}$$

und somit liegt hier bei $\cos\theta = -1$ eine Minimalstelle vor. Dies bedeutet aber, dass die minimierende Länge p exakt antiparallel zu $\nabla f(x_k)$ und von Länge 1 sein muss. Also ist die Minimalstelle des Optimierungsproblems

$$p = \frac{-\nabla f(x_k)}{||\nabla f(x_k)||}. \tag{79}$$

Damit ist $p_k^G := -\nabla f(x_k)$ aber der Richtungsvektor beliebiger Länge in der die Abnahme von f maximal ist.

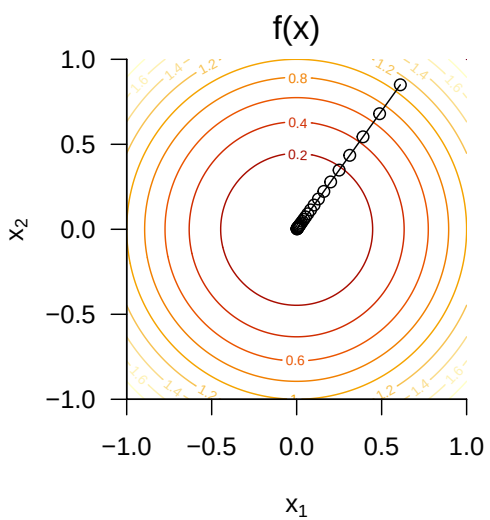
Beispiel

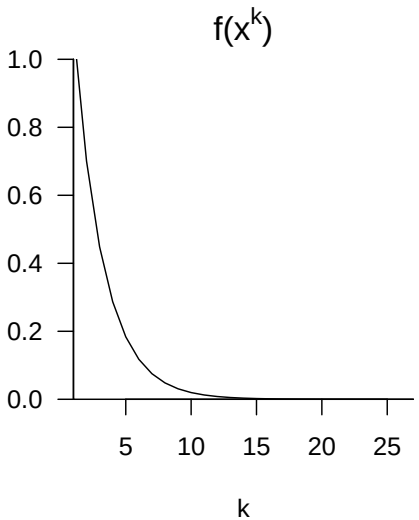
Minimierung von $f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2$

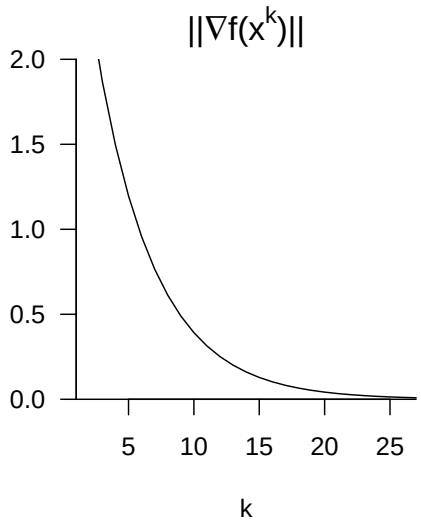
```
# Funktionsdefinitionen
# ------------------------------------------------------------------------------
# Zielfunktion
f = function(x) {
  return(x[1]^2 + x[2]^2)                  # f(x) := x_1^2 + x_2^2
}
# Gradient der Zielfunktion
nabla_f = function(x) {
  return(matrix(c(2*x[1],2*x[2]),          # \nabla f(x) := (2x_1, 2x_2)^T
                nrow = 2))
}
# Gradientenverfahren
# ------------------------------------------------------------------------------
# Parameter
n      = 2                               # Dimension
alpha  = 1e-1                            # Lernrate
delta  = 1e-2                            # Konvergenzkriterium

# Initialisierung
x_k    = matrix(c(.61,.85), nrow = 2)    # Zufälliger Startpunkt in [0,1]^2
x      = x_k                             # Initialisierung Iteranden
fx     = f(x_k)                          # Initialisierung Funktionswerte
crt    = norm(nabla_f(x_k))              # Initialisierung Kriterium

# Iterationen
while(norm(nabla_f(x_k)) > delta){

  # Argumentupdate
  x_k    = x_k - alpha*nabla_f(x_k)

  # Dokumentation
  x      = cbind(x,x_k)
  fx     = c(fx,f(x_k))
  crt    = c(crt, norm(nabla_f(x_k)))
}
```

Beispiel

Minimierung von $f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := x_1^2 + x_2^2$







Liniensuchverfahren als generalisierte Gradientenverfahren

Initialisierung

0. Wahl eines Startpunktes $x_0 \in \mathbb{R}^n$.

Iterationen

Für $k = 0, 1, 2, \ldots$

1. Wahl einer Abstiegsrichtung p_k
2. Wahl eines Lernparameters $\alpha_k \approx \min_\alpha f(x_k + \alpha p_k)$.
3. Setze $x_{k+1} := x_k + \alpha_k p_k$.
4. Konvergenztest.

$\Rightarrow$ Die Wahl sinnvoller Lernraten α_k ist für eine gute Performanz entscheidend!

(vgl. Ostwald and Starke (2016))

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Definition (Optimierungsproblem mit Nebenbedingungen)

Ein *Optimierungsproblem mit Nebenbedingungen* hat die allgemeine Form

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I, \tag{80}$$

wobei $f : \mathbb{R}^n \to \mathbb{R}$ und $c_i : \mathbb{R}^n \to \mathbb{R}, i \in E \cup I$ glatte multivariate reellwertige Funktions und E, I endliche Indexmengen sind. f heißt *Zielfunktion*, die $c_i, i \in E$ heißen *Gleichungsnebenbedingungen* und die $c_i, i \in I$ heißen *Ungleichungsnebenbedingungen*. Die Menge

$$\mathcal{X} := \{x \in \mathbb{R}^n | c_i(x) = 0, i \in E \text{ und } c_i(x) \geq 0, i \in I\} \tag{81}$$

heißt *feasible set*.

Bemerkung

- Die notwendigen Bedingungen für Minimalstellen bei Optimierungsproblem ohne Nebenbedingungen sind für $n = 1$: $f'(x^*) = 0$ und für $n > 1$: $\nabla f(x^*) = 0_n$. Im Folgenden führen wir analoge notwendige Bedingungen erster Ordnung für Minimalstellen bei Optimierungsproblemen mit Nebenbedingungen ein.

Beispiel

Definition (Quadratisches Programm)

Ein *Quadratisches Programm* ist das konvexe Optimierungsproblem mit den Nebenbedingungen

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T P x + q^T x \text{ u.d.N. } Ax = b \text{ und } -Gx + h \geq 0, \tag{82}$$

wobei

- $P \in \mathbb{R}^{n \times n}$ eine positiv definite Matrix ist,
- $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ sind und
- $G \in \mathbb{R}^{m \times n}$, und $h \in \mathbb{R}^m$ sind.

Bemerkungen

- Quadratische Programme sind Optimierungsprobleme mit Nebenbedingungen.
- Parameterlernen bei Support Vektor Maschinen führt auf ein Quadratisches Programm.
- Optimierungstoolboxen enhalten Funktionen zur Lösung Quadratischer Programme.
- In R bietet sich das Paket `quadprog` an.

Definition (Lagrange Funktion, Lagrange Multiplikatoren)

Es sei

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I, \tag{83}$$

ein Optimierungsproblem mit Nebenbedingungen. Dann ist die *Lagrange Funktion* dieses Problems definiert als

$$L : \mathbb{R}^n \times \mathbb{R}^{|E \cup I|} \to \mathbb{R}, (x, \lambda) \mapsto L(x, \lambda) := f(x) - \sum_{i \in E \cup I} \lambda_i c_i(x). \tag{84}$$

Hierbei wird $\lambda \in \mathbb{R}^{|E \cup I|}$ *Lagrange-Multiplikatoren Vektor* genannt und die einzelnen $\lambda_i \in \mathbb{R}$ mit $i \in E \cup I$ werden *Lagrange Multiplikatoren*genannt.

Bemerkung

- Die Lagrange Funktion und die Lagrange Multiplikatoren nehmen in den notwendigen Bedingungen der Optimierung mit Nebenbedingungen eine zentrale Rolle ein.

Definition (Notwendige Bedingungen erster Ordnung)

x^* sei eine lokale Lösung des Optimierungsproblems

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I. \tag{85}$$

Dann gibt es einen Lagrange-Multiplikatoren Vektor $\lambda^* \in \mathbb{R}^{|E \cup I|}$ mit den Komponenten $\lambda_i^*, i \in E \cup I$, so dass die folgenden Bedingungen an der Stelle $(x^*, \lambda^*) \in \mathbb{R}^{n+|E \cup I|}$ gelten

$$\begin{aligned}
\nabla_x L(x^*, \lambda^*) &= 0 \\
c_i(x^*) &= 0 \text{ für alle } i \in E \\
c_i(x^*) &\geq 0 \text{ für alle } i \in I \\
\lambda_i^* &\geq 0 \text{ für alle } i \in I \\
\lambda_i^* c_i(x^*) &= 0 \text{ für alle } i \in E \cup I
\end{aligned}$$

Bemerkungen

- Die Bedingungen werden auch *Karush-Kuhn-Tucker (KKT)* Bedingungen genannt.
- Für einen Beweis und Regularitätsbedingungen, siehe Nocedal and Wright (2006) Section 12.4.
- Die letzte Bedingung impliziert $\lambda_i^* > 0 \Rightarrow c_i(x^*) = 0$.

Definition (Duales Problem)

Es sei

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0, \tag{86}$$

ein Optimierungsproblem ohne Gleichungsnebenbedingungen, $c(x) := (c_1(x), c_2(x), ..., c_m(x))^T$ sei die multivariate vektorwertige Funktion der Ungleichungsnebenbedingungen und die zugehörige Lagrange Funktion und der Lagrange Multiplikatoren Vektoren $\lambda \in \mathbb{R}^m$ seien durch

$$L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, (x, \lambda) \mapsto L(x, \lambda) := f(x) - \lambda^T c(x). \tag{87}$$

gegeben. Dann ist die *duale Zielfunktion* (auch *duale Lagrange Funktion genannt*) definiert als

$$q : \mathbb{R}^m \to \mathbb{R}, \lambda \mapsto q(\lambda) := \min_x L(x, \lambda), \tag{88}$$

und das *duale Problem* ist definiert als

$$\max_{\lambda \in \mathbb{R}^m} q(\lambda) \text{ u.d.N. } \lambda \geq 0. \tag{89}$$

Bemerkung

- Duale Probleme sind manchmal einfacher zu lösen als die (primären) Ausgangsprobleme.
- Duale Probleme sind für das Parameterlernen von Support Vektor Maschinen zentral.

Theorem (Schwache Dualität)

Für jede Lösung $\bar{x}$ von

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0, \tag{90}$$

und jedes $\bar{\lambda} \geq 0$ gilt, dass

$$q(\bar{\lambda}) \leq f(\bar{x}). \tag{91}$$

Beweis

Mit den Definitionen von q, $\bar{\lambda} \geq 0$, und $c(\bar{x}) \geq 0$, gilt, dass

$$q(\bar{\lambda}) = \min_x f(x) - \bar{\lambda}^T c(x) \leq f(\bar{x}) - \bar{\lambda}^T c(\bar{x}) \leq f(\bar{x}). \tag{92}$$

□

Bemerkung

- Das Theorem besagt, dass der optimierte Wert des dualen Problems eine untere Grenze für den optimalen Wert der Zielfunktion des Ausgangsproblems ist.

Theorem (Starke Dualität)

Gegeben seien das Optimierungsproblem

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0 \tag{93}$$

und seine zugehörigen notwendigen Bedingungen erster Ordnung

$$\begin{aligned} \nabla f(\bar{x}) - \nabla c(\bar{x})\bar{\lambda} &= 0, \\ c(\bar{x}) &\geq 0, \\ \bar{\lambda} &\geq 0, \\ \bar{\lambda}_i c_i(\bar{x}) &= 0, i = 1, 2, \ldots, m, \end{aligned} \tag{94}$$

mit $\nabla c(x) = (\nabla c_1(x), \nabla c_2(x), \ldots, \nabla c_m(x)) \in \mathbb{R}^{n \times m}$. $\bar{x}$ sei eine Lösung des Ausgangsproblems und f sowie $-c_i, i = 1, 2, \ldots, m$ konvexe Funktionen auf $\mathbb{R}^n$, die in $\bar{x}$ differenzierbar sind. Dann ist jedes $\bar{\lambda}$, für das $(\bar{x}, \bar{\lambda})$ die notwendigen Bedingungen des Ausgangsproblem erfüllt, eine Lösung des dualen Problems

Bemerkungen

- Die optimalen Lagrange Multiplikatoren des Ausgangsproblems sind Lösungen des dualen Problems.
- SVM Training als Quadratisches Programm benötigt das Konzept der starken Dualität.

Beweis

Wir nehmen an, dass $(\bar{x}, \bar{\lambda})$ die notwendigen Bedingungen erster Ordnung für ein Minimum des Ausgangsproblem erfüllen und dass $L(\cdot, \bar{\lambda})$ konvex und differenzierbar ist. Dann gilt für jedes $x \in \mathbb{R}^n$, dass

$$L(x, \bar{\lambda}) \geq L(\bar{x}, \bar{\lambda}) + \nabla_x L(\bar{x}, \bar{\lambda})(x - \bar{x}) = L(\bar{x}, \bar{\lambda}), \tag{95}$$

weil $\nabla_x L(\bar{x}, \bar{\lambda}) = 0$. Also gilt für die duale Zielfunktion

$$q(\bar{\lambda}) = \inf_x L(x, \bar{\lambda}) = L(\bar{x}, \bar{\lambda}). \tag{96}$$

Mit der letzten der notwendigen Bedingungen erster Ordnung folgt weiterhin

$$q(\bar{\lambda}) = L(\bar{x}, \bar{\lambda}) = f(\bar{x}) - \bar{\lambda}^T c(\bar{x}) = f(\bar{x}) \tag{97}$$

Schließlich gilt mit dem Theorem zur Schwachen Dualität, dass $q(\lambda) \leq f(\bar{x})$ für alle $\lambda \geq 0$. Also folgt mit $q(\bar{\lambda}) = f(\bar{x})$, dass $\bar{\lambda}$ eine Lösung des dualen Problems ist. □

Prädiktive Modellierung

Differentialrechnung und Analytische Optimierung

Multivariate Differentialrechnung

Grundlagen der Optimierung

Gradientenverfahren

Grundlagen der Optimierung mit Nebenbedingungen

Selbstkontrollfragen

Selbstkontrollfragen

1. Erläutern Sie das allgemeine datenanalytische Vorgehen im Rahmen der Prädiktiven Modellierung.
2. Nennen Sie drei in der Prädiktiven Modellierung typischerweise vewendete Verfahren.
3. Erläutern Sie den Begriff der k-fachen Kreuzvalidierung.
4. Erläutern Sie Gemeinsamkeiten und Unterschiede explanatorischer und prädiktiver Modellierung.
5. Warum ist die Kenntnis von Optimierungsprinzipien im Rahmen der Prädiktiven Modellierung wichtig?
6. Definieren Sie die Begriffe der univariat-und multivariat-reellwertigen Funktion.
7. Definieren Sie Begriff der multivariaten vektorwertigen Funktion.
8. Definieren Sie den Begriff der Ableitung $f'(a)$ einer Funktion f an einer Stelle a.
9. Definieren den Begriff der Ableitung f' einer Funktion f.
10. Erläutern Sie die Symbole $f'(x)$, $\dot{f}(x)$, $\frac{df(x)}{dx}$, und $\frac{d}{dx}f(x)$.
11. Definieren Sie den Begriff der zweiten Ableitung f'' einer Funktion f.
12. Geben Sie die Summenregel für Ableitungen wieder.
13. Geben Sie die Produktregel für Ableitungen wieder.
14. Geben Sie die Quotientenregel für Ableitungen wieder.
15. Geben Sie die Kettenregel für Ableitungen wieder.

Selbstkontrollfragen

16. Bestimmen Sie die Ableitung der Funktion $f(x) := 3x^2 + \exp\left(-x^2\right) - x\ln(x)$
17. Bestimmen Sie die Ableitung der Funktion $f(x) := \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2$ für $\mu \in \mathbb{R}$.
18. Definieren Sie die Begriffe des globalen und lokalen Maximums/Minimums einer Funktion.
19. Geben Sie die notwendige Bedingung für ein Extremum einer Funktion wieder.
20. Geben Sie die hinreichende Bedingung für ein lokales Extremum einer Funktion wieder.
21. Geben Sie das Standardverfahren der analytischen Optimierung wieder.
22. Bestimmen Sie einen Extremwert von $f(x) := \exp\left(-\frac{1}{2}(x-\mu)^2\right)$ für $\mu \in \mathbb{R}$.
23. Berechnen Sie die (ersten) partiellen Ableitungen der Funktion

$$f : \mathbb{R}^2 \to \mathbb{R}, x \mapsto f(x) := \exp\left(-\frac{1}{2}\left(x_1^2 + x_2^2\right)\right) \tag{98}$$

24. Berechnen Sie die zweiten partiellen Ableitungen obiger Funktion f.
25. Geben Sie den Satz von Schwarz wieder.
26. Definieren Sie den Gradienten einer multivariaten reellwertigen Funktion.
27. Geben Sie den Gradienten obiger Funktion f an und werten Sie ihn in $x = (1,2)^T$ aus.

Selbstkontrollfragen

30. Definieren Sie die Hesse-Matrix einer multivariaten reellwertigen Funktion.
31. Geben Sie die Hesse-Matrix obiger Funktion f an und werten Sie sie in $x = (1, 2)^T$ aus.
32. Definieren Sie die allgemeine Form eines Optimierungsproblems.
33. Was sind x und f eines Optimierungsproblems in der Prädiktiven Modellierung?
34. Warum betrachtet man in der Theorie der Optimierung nur die Minimierung?
35. Definieren Sie die Begriffe der globalen und lokalen Minimalstellen einer multivariaten reellwertigen Funktion.
36. Geben Sie die notwendige Bedingung erster Ordnung für ein Minimum von $f : \mathbb{R}^n \to \mathbb{R}$ an.
37. Geben Sie die notwendige Bedingung zweiter Ordnung für ein Minimum von $f : \mathbb{R}^n \to \mathbb{R}$ an.
38. Geben Sie die hinreichende Bedingung zweiter Ordnung für ein Minimum von $f : \mathbb{R}^n \to \mathbb{R}$ an.
39. Geben Sie das Theorem zu Minimalstellen konvexer Funktionen an.
40. Formulieren Sie den Algorithmus des Gradientenverfahrens.
41. Geben Sie das Theorem zum Gradientenverfahren wieder.
42. Erläutern Sie die Bedeutung der Lernrate $\alpha > 0$ und des Konvergenzkriteriums $\delta > 0$ im Gradientenverfahren.

References

Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. 2018. "Machine Learning Approaches for Clinical Psychology and Psychiatry." *Annual Review of Clinical Psychology* 14 (1): 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research. New York: Springer.

Ostwald, Dirk, and Ludger Starke. 2016. "Probabilistic Delay Differential Equation Modeling of Event-Related Potentials." *NeuroImage* 136 (August): 227–57. https://doi.org/10.1016/j.neuroimage.2016.04.025.

Shmueli, Galit. 2010. "To Explain or to Predict?" *Statistical Science* 25 (3). https://doi.org/10.1214/10-STS330.

Vapnik, Vladimir. 2010. *The Nature of Statistical Learning Theory*. 2., nd ed. Softcover version of original hardcover edition 2000. Information Science and Statistics. New York, NY: Springer New York.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(7) Lineare Diskriminanzanalyse und Logistische Regression

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Vorbemerkungen

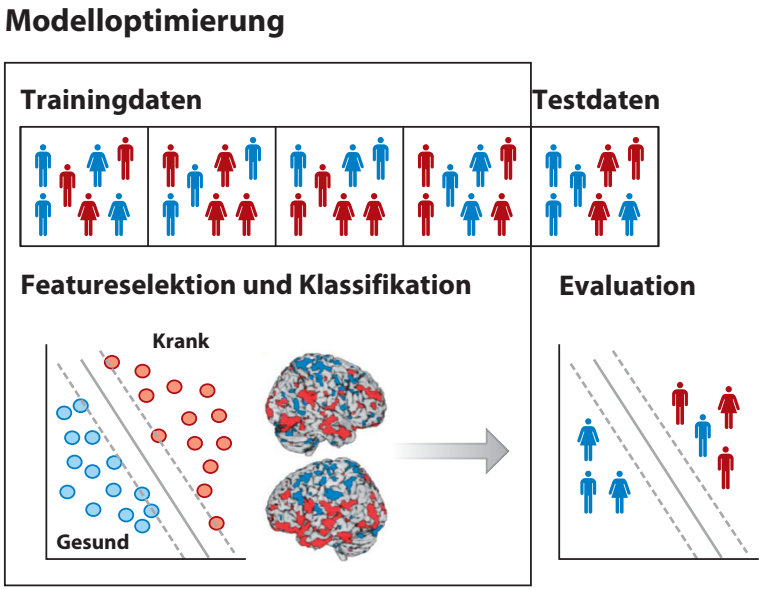
Lineare Diskriminanzanalyse

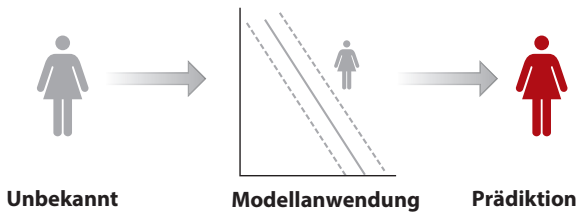
Logistische Regression

Selbstkontrollfragen

Appendix

Struktur der Prädiktiven Modellierung





nach Dwyer, Falkai, and Koutsouleris (2018)

Rhethorik der Prädiktiven Modellierung

Daten	Trainingsdaten und Testdaten
Statistisches Modell	Modell, Machine Learning Algorithmus
Schätzen von Parametern	Trainieren des Modells, Lernen von Parametern, Supervised Learning

Definition (Binärer Klassifikationstrainingdatensatz)

Ein *binärer Klassifikationsdatensatz*

$$\mathcal{D} := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(n)}, y^{(n)})\} \tag{1}$$

ist eine Menge von n *Trainingsdatenpunkten*

$$(x^{(i)}, y^{(i)}) \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{0, 1\} \text{ for } i = 1, ..., n, \tag{2}$$

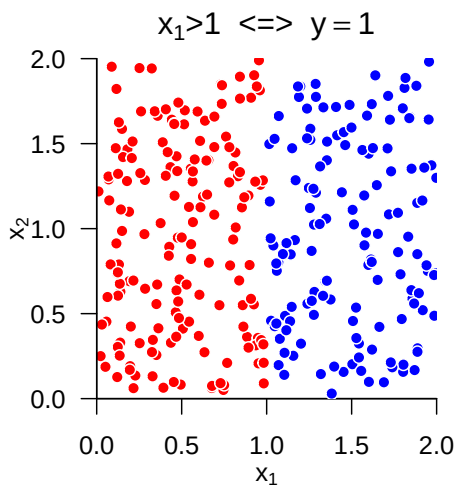
wobei $x^{(i)}$ m-*dimensionaler Featurevektor* und $y^{(i)}$ *Label* genannt wird

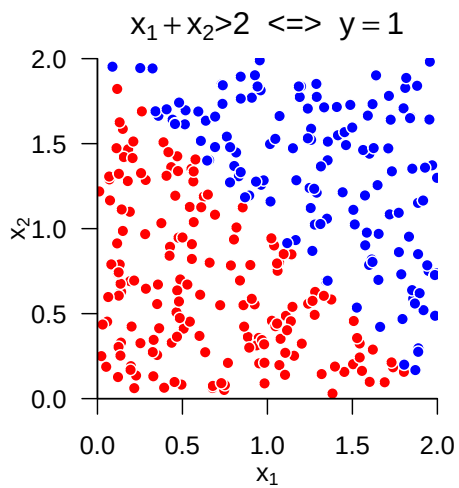
Bemerkungen

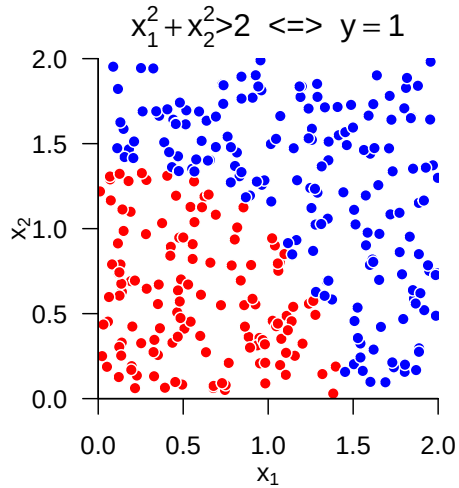
- $y^{(i)} \in \{0, 1\}$ bezeichnet die Klassenzugehörigkeit des Featurevektors $x^{(i)} \in \mathbb{R}^m$.
- Man beachte, dass hier $y^{(i)} \in \{0, 1\}$ gilt, wohingegen bei SVMs $y^{(i)} \in \{-1, 1\}$ ist.

Bivariate Featureplots mit $x^{(i)} \in \mathbb{R}^2, y^{(i)} \in \{0, 1\}, i = 1, ..., n$

$\bullet\ y = 0 \qquad \bullet\ y = 1$







Überblick

	Lineare Diskriminanzanalyse	Logistische Regression
Zufallsvektoren	$x \in \mathbb{R}^m, y \in \{0, 1\}$	$y \in \{0, 1\}$
Modell	$p(x, y) := B(y; \mu)N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y$	$p(y) := B\left(y; \frac{1}{1+\exp(-x^T\beta)}\right)$
Inferenz	$p(y\|x) = \frac{1}{1+\exp(-\tilde{x}^T\beta)}$	Keine
Klassifikation	$\delta(x) = 1$ für $p(y = 1\|x) > p(y = 0\|x)$	$\delta(x) = 1$ für $p(y = 1) > p(y = 0)$
Diskriminanz	$f(x) = w^T x + x_0, (w_0, w) = \beta$	$f(x) = w^T x + x_0, (w_0, w) = \beta$
Parameterlernen	Analytische Likelihood Maximierung	Numerische Likelihood Maximierung

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Definition (Multivariate Normalverteilung)

X sei ein m-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^m$ und WDF

$$p : \mathbb{R}^n \to \mathbb{R}_{>0},\ x \mapsto p(x) := (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{3}$$

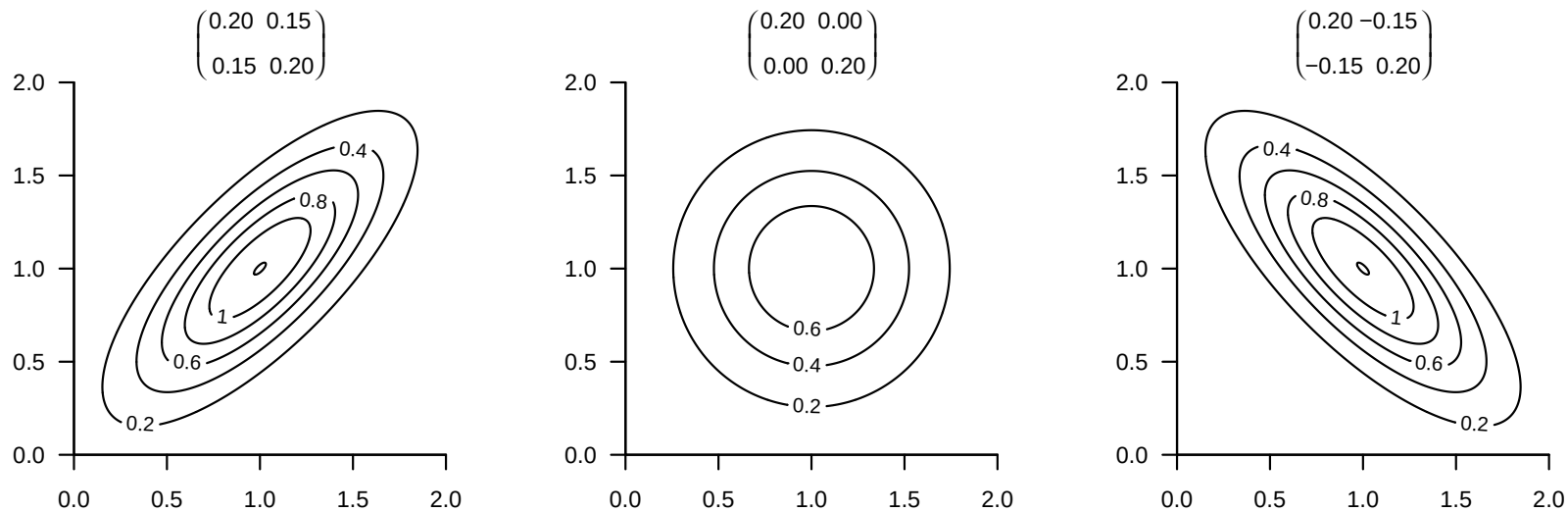
Dann sagen wird, dass X einer *multivariaten (oder m-dimensionalen) Normalverteilung* mit *Erwartungswertparameter* $\mu \in \mathbb{R}^m$ und positive-definitem *Kovarianzmatrixparameter* $\Sigma \in \mathbb{R}^{m \times m}$ unterliegt und nennen X einen *(multivariat) normalverteilten Zufallsvektor*. Wir kürzen dies mit $X \sim N(\mu, \Sigma)$ ab. Die WDF eines multivariat normalverteilten Zufallsvektors bezeichnen wir mit

$$N(x; \mu, \Sigma) := (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right). \tag{4}$$

Bemerkungen

- Der Parameter $\mu \in \mathbb{R}^m$ entspricht dem Wert höchster Wahrscheinlichkeitsdichte
- Die Diagonalelemente von Σ spezifizieren die Breite der WDF bezüglich $X_1, \ldots, X_m$.
- Das i,jte Element von Σ spezifiziert die Kovarianz on X_i und X_j.
- Der Term $(2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}}$ ist die Normalsieriungskonstante für den Exponentialfunktionsterm.

Zweidimensionale Normalverteilungen



Definition (Bernoulli Verteilung)

Es sei X eine Zufallsvariable mit Ergebnisraum $\mathcal{X} = \{0, 1\}$ und WMF

$$p : \mathcal{X} \to [0, 1], x \mapsto p(x) := \mu^x (1 - \mu)^{1-x} \text{ mit } \mu \in [0, 1]. \tag{5}$$

Dann sagen wir, dass X einer *Bernoulli-Verteilung mit Parameter* $\mu \in [0, 1]$ unterliegt und nennen X eine *Bernoulli-Zufallsvariable*. Wir kürzen dies mit $X \sim B(\mu)$ ab. Die WMF einer Bernoulli-Zufallsvariable bezeichnen wir mit

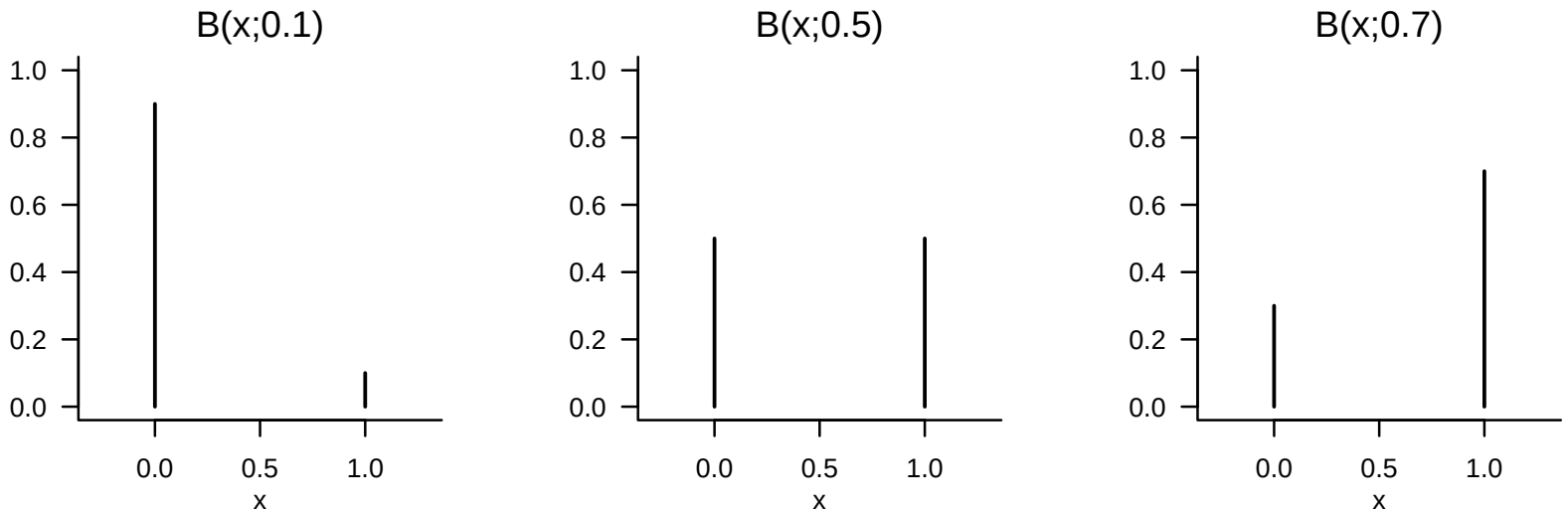
$$B(x; \mu) := \mu^x (1 - \mu)^{1-x}. \tag{6}$$

Bemerkungen

- Eine Bernoulli-Zufallsvariable kann als Modell eines Münzwurfs dienen.
- $\mu \in [0, 1]$ ist die Wahrscheinlichkeit dafür, dass X den Wert 1 annimmt,

$$\mathbb{P}(X = 1) = \mu^1 (1 - \mu)^{1-1} = \mu. \tag{7}$$

Bernoulli Verteilungen



Definition (Modell der Linearen Diskriminanzanalyse)

X sei ein m-dimensionaler Zufallsvektor mit Ergebnisraum $\mathbb{R}^m$ und Y sei eine Zufallsvariable mit Ergebnisraum $\{0, 1\}$. Dann ist das *Modell der Linearen Diskriminanzanalyse* die gemeinsame Verteilung

$$\mathbb{P}(X, Y) = \mathbb{P}(Y)\mathbb{P}(X|Y) \tag{8}$$

wobei die diskrete marginale Verteilung $\mathbb{P}(Y)$ durch die WMF

$$p(y) = B(y; \mu) \tag{9}$$

mit $\mu \in]0, 1[$ definiert und die kontinuierliche bedingte Verteilung $\mathbb{P}(X|Y)$ durch die WDF

$$p(x|y) = N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{10}$$

mit $\mu_0, \mu_1 \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. definiert ist. Wir bezeichnen die gemischte WDF/WMF des LDA Modells mit

$$p(x, y) := p(y)p(x|y) = B(y; \mu)N(x; \mu_0, \Sigma)^{1-y} N(x; \mu_1, \Sigma)^y \tag{11}$$

Bemerkung

Aus generativer Sicht wird ein Trainingsdatensatz

$$\left\{ \left(x^{(i)}, y^{(i)} \right) \right\}_{i=1}^{n} \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{0, 1\} \tag{12}$$

eines LDA Modells wie folgt erzeugt:

(1) $y^{(i)}$ wird zunächst durch Ziehen aus einer Bernoulliverteilung mit Parameter μ erzeugt.

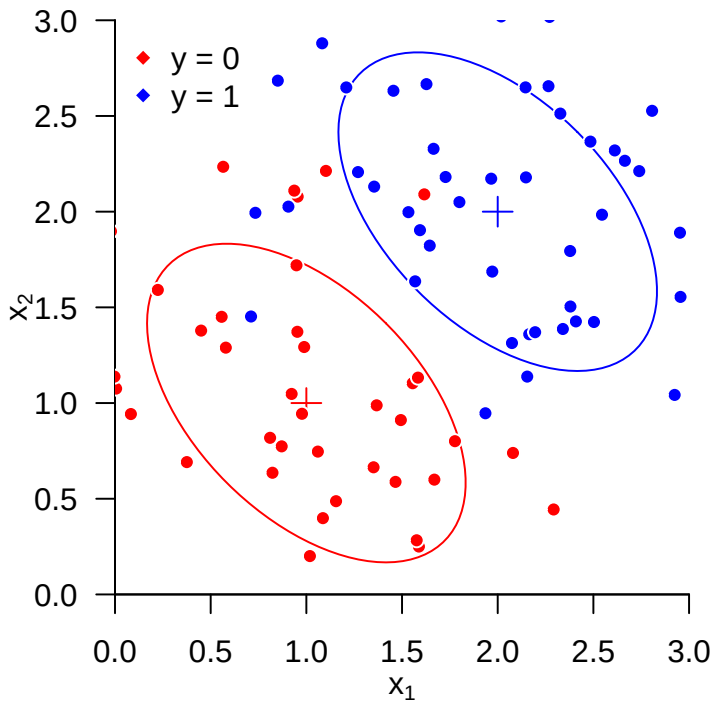
(2) In Abhängigkeit vom Wert von $y^{(i)}$ wird $x^{(i)}$ dann durch Ziehen aus einer multivariaten Normalverteilung mit Kovarianzmatrixparameter Σ und Erwartungswertparameter μ_0 für $y^{(i)} = 0$ oder μ_1 für $y^{(i)} = 1$ erzeugt.

Datengeneration

```
# R Paket für multivariate Normalverteilung
library(mvtnorm)
set.seed(0)

# Modellparameter
m      = 2                                          # Featurevektordimension
n      = 2e2                                        # Anzahl Trainingsdatenpunkte
mu     = 0.5                                        # wahrer, aber unbekannter, Bernoulliparameter \mu
mu_0   = c(1,1)                                     # wahrer, aber unbekannter, Normalverteilungsparameter \mu_0
mu_1   = c(2,2)                                     # wahrer, aber unbekannter, Normalverteilungsparameter \mu_1
Sigma  = matrix(c( 0.50, -0.25,                     # Kovarianzmatrixparameter
                  -0.25,  0.50),
                byrow = TRUE,
                nrow = m)

# Modellsampling
y      = matrix(rep(NaN,n)  , nrow = 1)             # Labeldatenarray
x      = matrix(rep(NaN,n*m), nrow = m)             # Featurevektorarray
for(i in 1:n){
  y[i]   = rbinom(1,1,mu)                           # y^{(i)} \sim B(\mu)
  x[,i] = ((y[i] == 0)*rmvnorm(1, mu_0, Sigma)      # x^{i} \sim N(\mu_0,\Sigma)^{1-y}N(\mu_1,\Sigma)^y
          +(y[i] == 1)*rmvnorm(1, mu_1, Sigma))
}

# Datensatzkonkatenation
D = rbind(x,y)
```

Datengeneration

$$\mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.50 & -0.10 \\ -0.10 & 0.50 \end{pmatrix}$$



Theorem (LDA Inferenz)

$p(x, y)$ sei die WMF/WDF eines LDA Model. Dann gilt

$$p(y = 1|x) = \frac{1}{1 + \exp(-\tilde{x}^T \beta)} \text{ und } p(y = 0|x) = 1 - p(y = 1|x), \tag{13}$$

wobei

$$\tilde{x} := \begin{pmatrix} 1 \\ x \end{pmatrix} \in \mathbb{R}^{m+1} \tag{14}$$

der *erweiterten Featurevektor* und

$$\beta := \begin{pmatrix} \frac{1}{2}\left(\mu_0^T \Sigma^{-1} \mu_0 - \mu_1^T \Sigma^{-1} \mu_1\right) + \ln\left(\frac{\mu}{1-\mu}\right) \\ -\Sigma^{-1}(\mu_0 - \mu_1) \end{pmatrix} \in \mathbb{R}^{m+1}. \tag{15}$$

der *Inferenzparametervektor* sind.

Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.
- $p(y|x)$ kann zur Prädiktion der Klasse eines $x \in \mathbb{R}^m$ genutzt werden.
- Diese Prädiktion hängt von den LDA Modellparametern $\mu, \mu_0, \mu_1, \Sigma$ ab.

Definition (Klassifikationsregel der linearen Diskriminanzanalyse)

$p(x, y)$ sei die WMF/WDF eines LDA Modells. Dann ist die *Klassifikationregel* definiert als

$$\delta : \mathbb{R}^m \to \{0, 1\}, x \mapsto \delta(x) := \begin{cases} 0 & \text{für } p(y = 0|x) \geq p(y = 1|x) \\ 1 & \text{für } p(y = 0|x) < p(y = 1|x) \end{cases} \tag{16}$$

Bemerkung

- Es gilt

$$\delta(x) = 1 \Leftrightarrow p(y = 1|x) > p(y = 0|x) \Leftrightarrow p(y = 1|x) > 0.5. \tag{17}$$

Inferenz und Klassifikation bei bekannten Modellparametern

$$\mu = 0.5, \mu_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.10 & -0.05 \\ -0.05 & 0.10 \end{pmatrix}$$

```
# Inferenz und Klassifikation für die ersten k Datenpunkte
library(matlib)                                          # Matrixtools
k           =  10                                        # Anzahl Datenpunkte
x_tilde     = rbind(rep(1,k), x[,1:k])                   # erweiterte Featurevektoren
beta        = matrix(                                    # Inferenzparametervektor
              c((1/2)*(  t(mu_0) %*% inv(Sigma) %*% mu_0
                       - t(mu_1) %*% inv(Sigma) %*% mu_1)
                       + log(mu/(1-mu)),
                -inv(Sigma) %*% (mu_0-mu_1)), nrow = 3)
p_y_giv_x   = 1/(1+exp(-t(x_tilde) %*% beta))            # p(y = 1|x)
delta       = as.numeric(p_y_giv_x >= 0.5)               # Klassifikationsregel
```

	1	2	3	4	5	6	7	8	9	10
x_1	1.137	1.800	2.380	0.871	2.485	2.145	0.083	0.823	2.61	1.466
x_2	3.243	2.050	1.504	0.774	2.366	2.649	0.943	0.636	2.32	0.588
p(y = 1|x)	0.996	0.968	0.972	0.004	0.999	0.999	0.000	0.002	1.00	0.022
delta(x)	1.000	1.000	1.000	0.000	1.000	1.000	0.000	0.000	1.00	0.000
y	1.000	1.000	1.000	0.000	1.000	1.000	0.000	0.000	1.00	0.000

Theorem (LDA Diskriminanzfunktion)

$p(x, y)$ sei die WMF/WDF eines LDA Mmodells und $\beta \in \mathbb{R}^{m+1}$ sei der Inferenzparametervektor. Dann kann die LDA Klassifikationsregel δ als eine lineare Diskriminanzfunktion der Form

$$h : \mathbb{R}^m \to \{0, 1\}, x \mapsto h(x) := g(f(x)), \tag{18}$$

mit

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{19}$$

und

$$g : \mathbb{R} \to \{0, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} 0, & f(x) \geq 0 \\ 1, & f(x) < 0 \end{cases} \tag{20}$$

geschrieben werden, d.h. es gilt $\delta(x) = h(x)$ für alle $x \in \mathbb{R}^m$. Inbesondere gilt dabei

$$w_0 = \beta_1 \text{ und } w = (\beta_2, \ldots, \beta_{m+1})^T \tag{21}$$

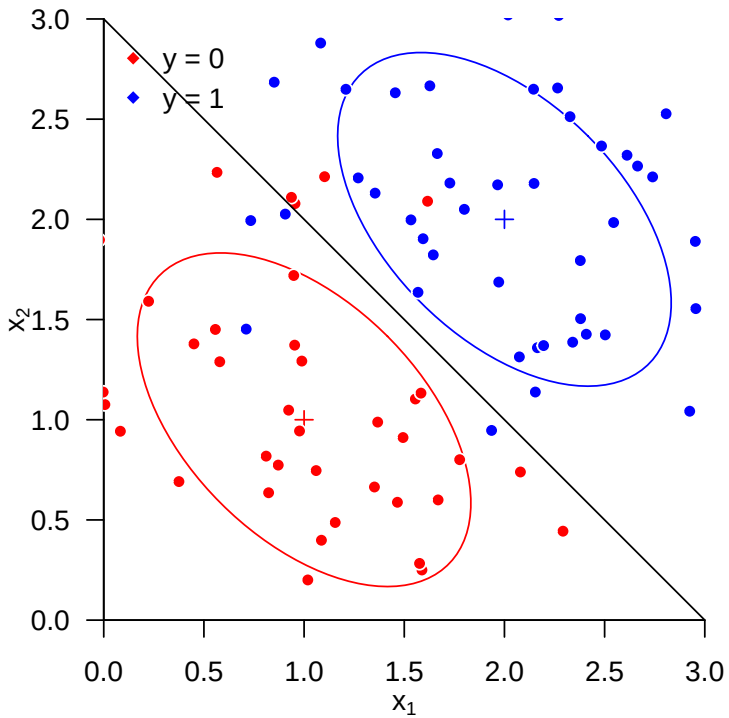
Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.
- Zur Visualisierung in zweidimensionalen Featureräumen dient die Graphgleichungen für Hyperebenen

$$f(x) = 0 \Leftrightarrow w^T x + w_0 = 0 \Leftrightarrow w_1 x_1 + w_2 x_2 + w_0 = 0 \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \tag{22}$$

Implementation der Diskriminanfunktion bei bekannten Modellparametern

```
# Diskriminanzfunktion
x_1  = seq(0,3,len = 1e2)                    # x_1
w_0  = beta[1]                               # w_0
w    = beta[2:(m+1)]                         # w
x_2  = -(w[1]/w[2])*x_1 - (w_0/w[2])         # x_2
```



Theorem (LDA Maximum Likelihood Schätzer)

$p(x, y)$ sei die WMF/WDF eines LDA Modells mit Parametern $\{\mu, \mu_0, \mu_1, \Sigma\}$, $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ sei ein LDA Trainingsdatensatz, und $1_{\{S\}}$ sei die Indikatorfunktion der Aussage A, d.h. $1_{\{A\}} = 1$, wenn A WAHR ist uns $1_{\{A\}} = 0$, wenn A FALSCH ist. Dann sind die Maximum Likelihood Schätzer für μ, μ_0, μ_1 und Σ gegeben durch

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{n} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}, \\
\hat{\mu}_0 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} x^{(i)}, \\
\hat{\mu}_1 &= \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} x^{(i)}, \\
\hat{\Sigma} &= \frac{1}{n} \sum_{i=1}^{n} \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right) \left(x^{(i)} - \hat{\mu}_{y^{(i)}} \right)^T .
\end{aligned}
\tag{23}
$$

Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix

Bemerkungen (fortgeführt)

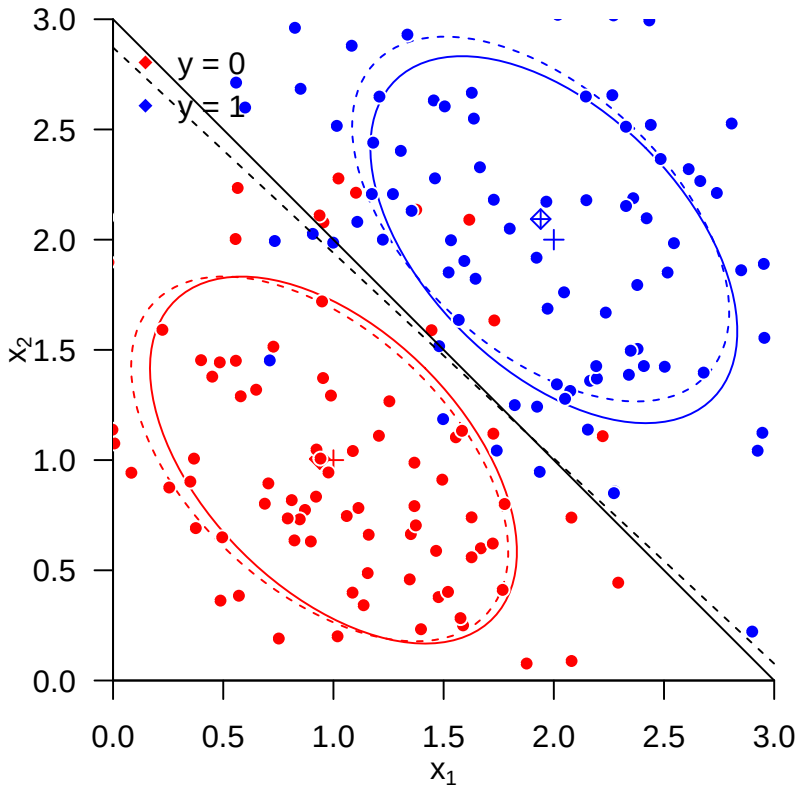
- μ wird als die relative Häufigkeit der 1en im Trainingsdatensatz geschätzt.
- μ_0 und μ_1 werden als Stichprobenmittel aller $x^{(i)}$ mit $y^{(i)} = 0$ bzw. $y^{(i)} = 1$ geschätzt.
- Σ wird durch die empirische Kovarianzmatrix aller $x^{(i)}, i = 1, \ldots, n$ geschätzt.
- Substitution ergibt die Schätzer $\hat{\beta}$, $\hat{w}$, $\hat{w}_0$ und $\hat{h}$

Implementation

```
# Parameterlernen bei gegebenen Featurevektorset x \in \mathbb{R}^{m \times n} und Labelset y \in {0,1}^n
n             = ncol(x)                                                    # n
m             = nrow(x)                                                    # m
mu_hat        = mean(y)                                                    # \hat{\mu}
mu_0_hat      = rowMeans(x[, y == 0])                                      # \hat{\mu}_0
mu_1_hat      = rowMeans(x[, y == 1])                                      # \hat{\mu}_1
Sigma_hat     = matrix(rep(0,m^2), nrow = m)                               # \hat{\Sigma}
for(i in 1:n){
  Sigma_hat   = (Sigma_hat + (1/n)*
                ((y[i] == 0)*(x[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
                +(y[i] == 1)*(x[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))
}
beta_hat      = matrix(c((1/2)*( t(mu_0_hat) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                               - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                               + log(mu_hat/(1-mu_hat)),
                         -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)),
                       nrow = m+1)
w_0_hat       = beta_hat[1]                                                # \hat{w}
w_hat         = beta_hat[2:(m+1)]                                          # \hat{w_0}
x_2_hat       = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])              # \hat{h}
```

Implementation

$$\hat{\mu} = 0.52, \hat{\mu}_0 = \begin{pmatrix} 0.94 \\ 1.01 \end{pmatrix}, \hat{\mu}_1 = \begin{pmatrix} 1.94 \\ 2.09 \end{pmatrix}, \hat{\Sigma} = \begin{pmatrix} 0.53 & -0.26 \\ -0.26 & 0.49 \end{pmatrix}$$



Prädiktion des Studienerfolgs

Nach Rudolf and Buse (2020) Kapitel 4

Datensatz zum Verhältnis psychologischer Diagnostik und Studienerfolg

$n = 30$ Studierende naturwissenschaftlicher Studiengänge

Featurevektor $x \in \mathbb{R}^4$

- x_1 Intelligenztestscore
- x_2 Mathematiktestscore
- x_3 Gewissenhaftigkeitscore
- x_4 Verträglichkeitscore

Label $y \in \{0, 1\}$

- 0: ungenügend (Studienabbruch aufgrund nicht bestandener Prüfungen)
- 1: gut (Abschlussnote besser als 2.5)

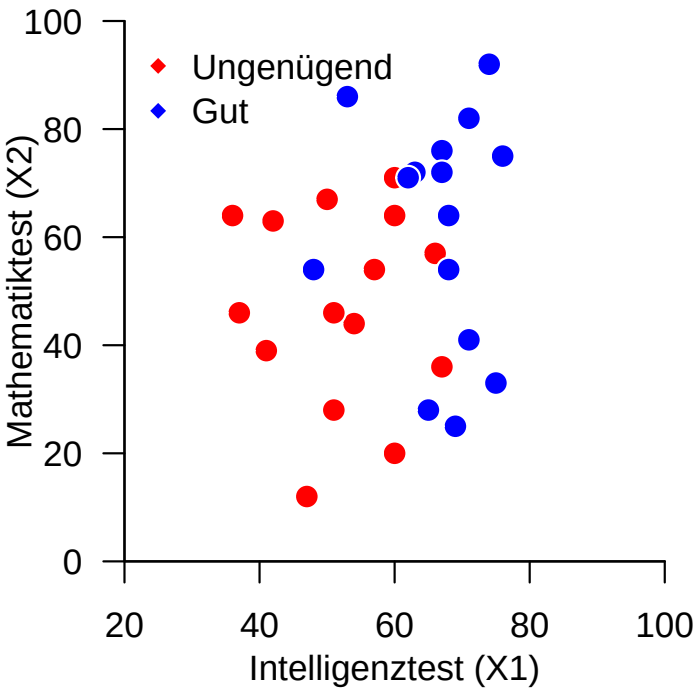
Datensatz $i = 1, ..., 15$

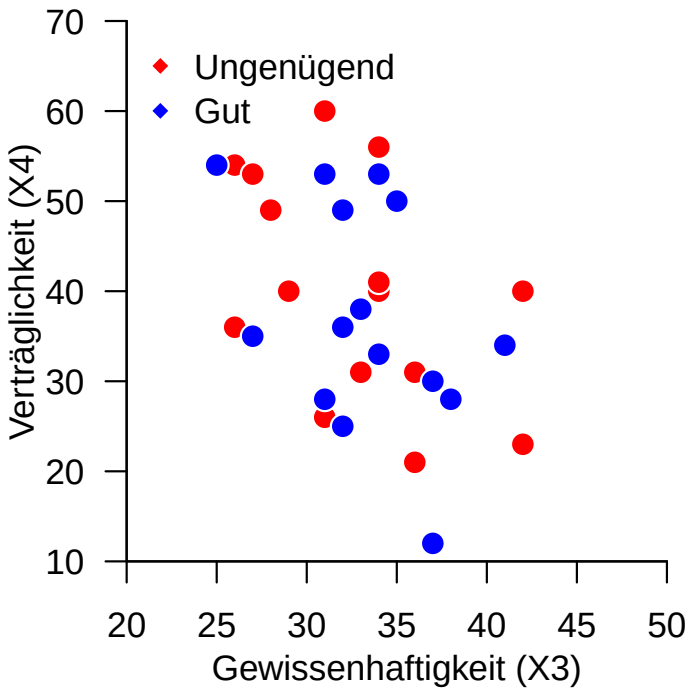
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
X1	54	60	67	41	66	51	51	37	57	47	50	42	60	36	60
X2	44	20	36	39	57	28	46	46	54	12	67	63	64	64	71
X3	31	33	26	31	34	42	34	36	28	34	27	26	29	36	42
X4	60	31	54	26	56	23	40	31	49	41	53	36	40	21	40
y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Datensatz $i = 16, ..., 30$

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
X1	71	65	67	68	75	71	68	63	48	53	62	69	67	74	76
X2	41	28	76	54	33	82	64	72	54	86	71	25	72	92	75
X3	37	33	38	34	25	32	34	32	41	27	31	32	31	35	37
X4	30	38	28	53	54	49	33	36	34	35	53	25	28	50	12
y	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Datensatz


Ungenügend
Gut
Mathematiktest (X2)
Intelligenztest (X1)
0
20
40
60
80
100
20
40
60
80
100

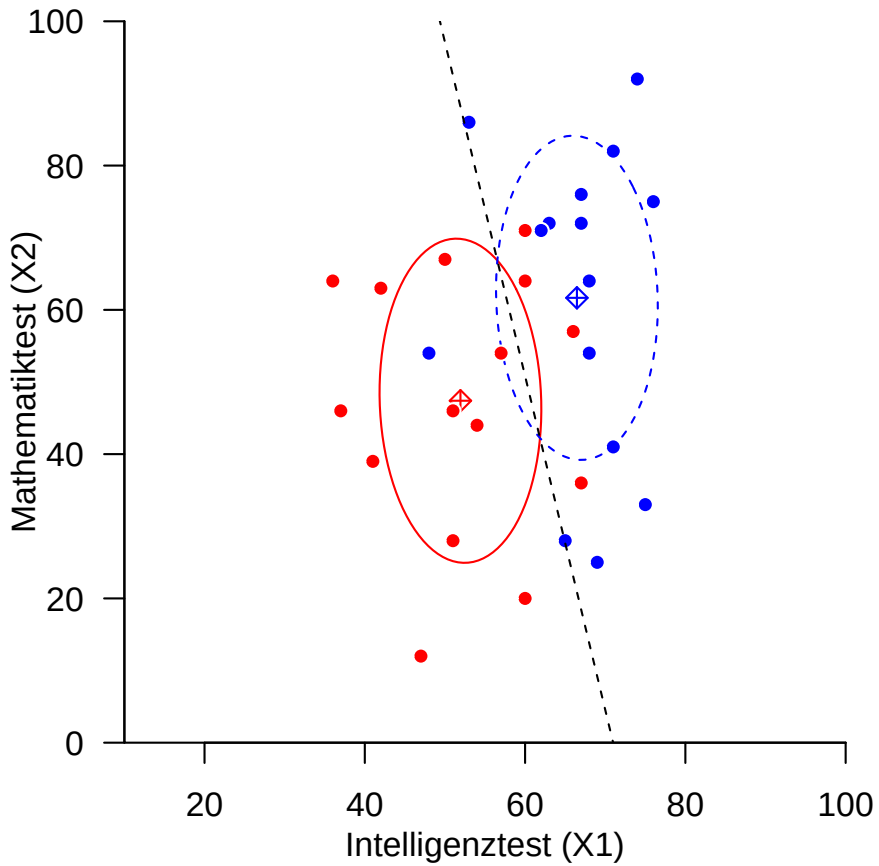

Ungenügend
Gut
Verträglichkeit (X4)
Gewissenhaftigkeit (X3)
10
20
30
40
50
60
70
20
25
30
35
40
45
50

Parameterlernen und lineare Diskriminanzfunktion

```
library(matlib)                                                          # Matrixtools
D             = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv")) # Datensatz
x             = as.matrix(D[1:2,])                                       # Featureselektion
y             = as.matrix(D[5,])                                         # Label
D             = rbind(x,y)                                               # Featureselected Datensatz
n             = ncol(x)                                                  # Datensatzgröße
m             = nrow(x)                                                  # Featurevektordimensionalität
mu_hat        = mean(y)                                                  # \hat{\mu}
mu_0_hat      = rowMeans(x[, y == 0])                                    # \hat{\mu}_0
mu_1_hat      = rowMeans(x[, y == 1])                                    # \hat{\mu}_1
Sigma_hat     = matrix(rep(0,4), nrow = 2)                               # \hat{\Sigma}
for(i in 1:n){
  Sigma_hat   = (Sigma_hat + (1/n)*
                ( (y[i] == 0)*(x[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
                 +(y[i] == 1)*(x[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))
}
beta_hat      = matrix(c((1/2)*( t(mu_0_hat) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                               - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                               + log(mu_hat/(1-mu_hat)),
                       -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)), nrow = 3)
w_0_hat       = beta_hat[1]                                              # \hat{w}
w_hat         = beta_hat[2:(m+1)]                                        # \hat{w_0}
x_1           = seq(min(x[1,]), max(x[1,]), len = 1e2)                   # x_1
x_2_hat       = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])            # \hat{h}
x_tilde       = rbind(1, x)
p_y_giv_x     = 1/(1+exp(-t(x_tilde) %*% beta_hat))
delta         = as.numeric(p_y_giv_x >= 0.5)
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.833
```

Parameterlernen und lineare Diskriminanzfunktion



Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 2$

```
# Datensatz
D                = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv"))
x                = as.matrix(D[1:2,])
y                = as.matrix(D[5,])
D                = rbind(x,y)
K                = ncol(D)
p_y_giv_x        = matrix(rep(NaN, ncol(y)), nrow = 1)
delta            = matrix(rep(NaN, ncol(y)), nrow = 1)

# K-fache Leave-One-Out Cross-Validation
for(k in 1:K){

    # Datensatzpartition
    x_train      = as.matrix(x[,-k])
    y_train      = as.matrix(y[,-k])
    x_test       = as.matrix(x[, k])
    y_test       = as.matrix(y[, k])

    # Trainingsdatensatz-basiertes Parameterlernen
    n            = ncol(x_train)
    m            = nrow(x_train)
    mu_hat       = mean(y_train)
    mu_0_hat     = rowMeans(x_train[, y_train == 0])
    mu_1_hat     = rowMeans(x_train[, y_train == 1])
    Sigma_hat    = matrix(rep(0,m^2), nrow = m)
    for(i in 1:n){
      Sigma_hat  = (Sigma_hat + (1/n)*
                   ((y_train[i] == 0)*(x_train[,i]-mu_0_hat) %*% t((x[,i]-mu_0_hat))
                   +(y_train[i] == 1)*(x_train[,i]-mu_1_hat) %*% t((x[,i]-mu_1_hat))))
    }
    beta_hat     = matrix(c((1/2)*( t(mu_0_hat) %*% inv(Sigma_hat) %*% mu_0_hat  # \hat{beta}
                                  - t(mu_1_hat) %*% inv(Sigma_hat) %*% mu_1_hat)
                                  + log(mu_hat/(1-mu_hat)),
                         -inv(Sigma_hat) %*% (mu_0_hat-mu_1_hat)), nrow = m+1)

    # Prädiktion
    x_test_tilde = rbind(1, x_test)
    p_y_giv_x[k] = 1/(1+exp(-t(x_test_tilde) %*% beta_hat))
    delta[k]     = as.numeric(p_y_giv_x[k] >= 0.5)
}
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.667
```

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 2$

k	x_1	x_2	p(y = 1|x)	delta(x)	y
1	54	44	1.0	1	0
2	60	20	1.0	1	0
3	67	36	1.0	1	0
4	41	39	1.0	1	0
5	66	57	0.0	0	0
6	51	28	0.0	0	0
7	51	46	0.0	0	0
8	37	46	0.0	0	0
9	57	54	0.1	0	0
10	47	12	0.0	0	0
11	50	67	0.2	0	0
12	42	63	0.0	0	0
13	60	64	0.9	1	0
14	36	64	0.0	0	0
15	60	71	1.0	1	0
16	71	41	0.7	1	1
17	65	28	0.1	0	1
18	67	76	1.0	1	1
19	68	54	0.9	1	1
20	75	33	0.8	1	1
21	71	82	1.0	1	1
22	68	64	0.9	1	1
23	63	72	0.9	1	1
24	48	54	0.0	0	1
25	53	86	0.5	0	1
26	62	71	0.8	1	1
27	69	25	0.5	0	1
28	67	72	0.9	1	1
29	74	92	1.0	1	1
30	76	75	1.0	1	1

Prediction Accuracy = 0.67.

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 3$

k	x_1	x_2	x_3	p(y = 1|x)	delta(x)	y
1	54	44	31	1.0	1	0
2	60	20	33	1.0	1	0
3	67	36	26	1.0	1	0
4	41	39	31	0.0	0	0
5	66	57	34	0.0	0	0
6	51	28	42	0.0	0	0
7	51	46	34	0.0	0	0
8	37	46	36	0.0	0	0
9	57	54	28	0.0	0	0
10	47	12	34	0.0	0	0
11	50	67	27	0.0	0	0
12	42	63	26	0.0	0	0
13	60	64	29	0.0	0	0
14	36	64	36	0.0	0	0
15	60	71	42	0.4	0	0
16	71	41	37	0.0	0	1
17	65	28	33	0.0	0	1
18	67	76	38	0.4	0	1
19	68	54	34	0.2	0	1
20	75	33	25	0.2	0	1
21	71	82	32	0.9	1	1
22	68	64	34	0.6	1	1
23	63	72	32	0.5	0	1
24	48	54	41	0.0	0	1
25	53	86	27	0.2	0	1
26	62	71	31	0.7	1	1
27	69	25	32	0.3	0	1
28	67	72	31	0.9	1	1
29	74	92	35	1.0	1	1
30	76	75	37	1.0	1	1

Prediction Accuracy = 0.60.

Lineare Diskriminanzanalyse | Leave-One-Out Cross-Validation, $m = 4$

k	x_1	x_2	x_3	x_4	p(y = 1|x)	delta(x)	y
1	54	44	31	60	0.0	0	0
2	60	20	33	31	0.0	0	0
3	67	36	26	54	0.0	0	0
4	41	39	31	26	0.0	0	0
5	66	57	34	56	0.0	0	0
6	51	28	42	23	0.0	0	0
7	51	46	34	40	0.0	0	0
8	37	46	36	31	0.0	0	0
9	57	54	28	49	0.0	0	0
10	47	12	34	41	0.0	0	0
11	50	67	27	53	0.0	0	0
12	42	63	26	36	0.0	0	0
13	60	64	29	40	0.0	0	0
14	36	64	36	21	0.0	0	0
15	60	71	42	40	0.0	0	0
16	71	41	37	30	0.0	0	1
17	65	28	33	38	0.0	0	1
18	67	76	38	28	0.1	0	1
19	68	54	34	53	0.0	0	1
20	75	33	25	54	1.0	1	1
21	71	82	32	49	1.0	1	1
22	68	64	34	33	1.0	1	1
23	63	72	32	36	1.0	1	1
24	48	54	41	34	0.0	0	1
25	53	86	27	35	0.9	1	1
26	62	71	31	53	0.3	0	1
27	69	25	32	25	1.0	1	1
28	67	72	31	28	1.0	1	1
29	74	92	35	50	1.0	1	1
30	76	75	37	12	1.0	1	1

Prediction Accuracy = 0.80.

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Definition (Generalisiertes Lineares Modell)

$x \in \mathbb{R}^m$ sei ein erweiterter Featurevektor und y das assoziierte Label. Weiterhin sei für einen *Parametervektor* $\beta \in \mathbb{R}^m$

$$\eta := x^T \beta \tag{24}$$

ein *linearer Prädiktor*. Dann ist ein generalisierte lineares Modell definiert mithilfe einer zweimal differenzierbaren und invertierbaren *link function*

$$g : \mathbb{R} \to \mathbb{R}, \mathbb{E}(y) \mapsto g(\mathbb{E}(y)) =: \eta. \tag{25}$$

definiert. Die Inverse der link function,

$$g^{-1} : \mathbb{R} \to \mathbb{R}, \eta \mapsto g^{-1}(\eta) = \mathbb{E}(y) \tag{26}$$

heißt *mean function* und wird mit f bezeichnet, so dass

$$f : \mathbb{R} \to \mathbb{R}, \eta \mapsto f(\eta) = \mathbb{E}(y). \tag{27}$$

Definition (Allgemeines Lineares Modell als Generalisiertes Lineares Modell)

Das Allgemeine Lineare Modell mit u.i.v. Störvariablen ist das Generalisierte Lineare Model, bei dem

1. die Labelvariable eine univariat normalverteilte Zufallsvariable

$$y \sim N(\mu, \sigma^2), \tag{28}$$

   ist und

2. die link function durch die Identität

$$g : \mathbb{R} \to \mathbb{R}, \mu \mapsto g(\mu) := \mu =: \eta. \tag{29}$$

   gegeben ist.

Weil die Inverse der Identität wiederrum die Identität ist, folgt, dass die mean function des ALM durch

$$f : \mathbb{R} \to \mathbb{R}, \eta \mapsto f(\eta) = \eta = \mu. \tag{30}$$

gegeben ist. Die Parameter des Allgemeinen Linearen Modells sind die Komponenten des Vektors $\beta \in \mathbb{R}^m$ des linearen Prädiktors $\eta = x^T \beta$ und der Parameter $\sigma^2 > 0$.

Definition (Logistische Regression als Generalisiertes Lineares Modell)

Das Modell der Logistischen Regression (LR) ist das Generalisierte Lineare Modell, bei dem

1. die Labelvariable eine Bernoulli-Zufallsvariable

$$y \sim B(\mu) \tag{31}$$

   ist und

2. die link function durch die *standard logit function*

$$g : [0,1] \to \mathbb{R}, \mu \mapsto g(\mu) := \ln\left(\frac{\mu}{1-\mu}\right) =: \eta \tag{32}$$

   gegeben ist.

Die Parameter des Logistischen Regressionsmodels sind die Komponenten des Vektors $\beta \in \mathbb{R}^m$ des linearen Prädiktors $\eta = x^T\beta$.

Theorem (Mean function der Logistischen Regression)

Die Inverse der link function des Modells der Logistischen Regression und somit seine mean function ist die *standard logistic function*

$$f : \mathbb{R} \to [0, 1], \eta \mapsto f(\eta) = \frac{1}{1 + \exp(-\eta)}. \tag{33}$$

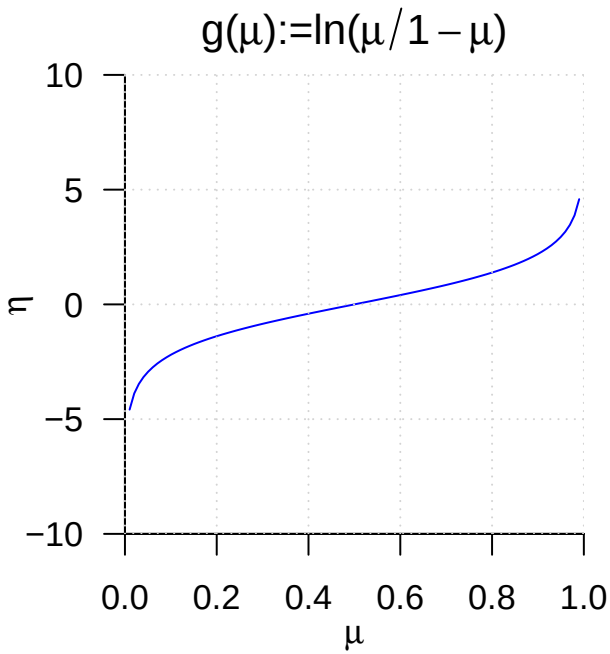
Beweis

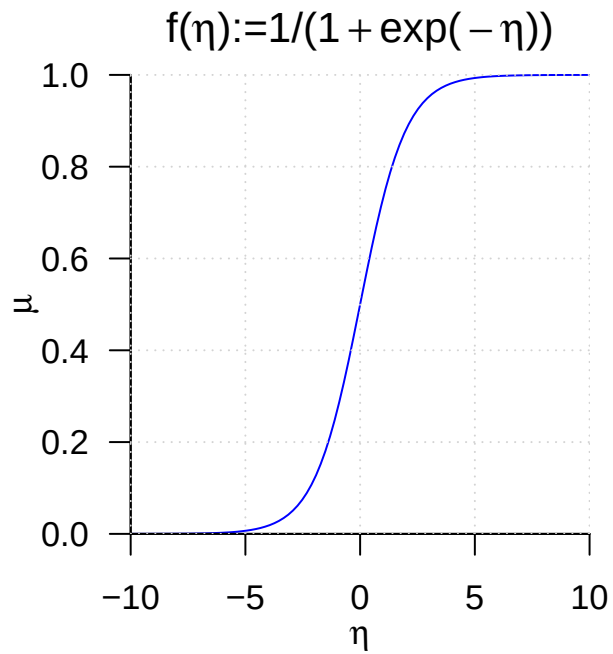
Umformen der logit function ergibt

$$\begin{aligned}
\eta &= \ln(\mu/(1-\mu)) \\
\Leftrightarrow -\eta &= -\ln(\mu/(1-\mu)) \\
\Leftrightarrow -\eta &= \ln((1-\mu)/\mu) \\
\Leftrightarrow \exp(-\eta) &= (1-\mu)/\mu \\
\Leftrightarrow \mu \exp(-\eta) &= 1-\mu \\
\Leftrightarrow \exp(-\eta) &= \mu^{-1} - 1 \\
\mu &= 1/(\exp(-\eta) + 1)
\end{aligned}$$

□

Link und Mean Funktionen





Definition (Modell der Logistischen Regression)

y sei eine Zufallsvariable mit Ergebnisraum $\{0, 1\}$. Dann ist das *Modell der Logistischen Regressionsmodell* definiert als die WMF

$$p(y) = B\left(y; \frac{1}{1 + \exp(-x^T \beta)}\right), \tag{34}$$

wobei $x \in \mathbb{R}^{m+1}$ einen erweiterten Featurevektor und $\beta \in \mathbb{R}^{m+1}$ den *Parametervektor* bezeichnen

Bemerkung

- Aus generativer Sicht wird ein Trainingsdatensatz

$$\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n} \text{ mit } x^{(i)} \in \mathbb{R}^{m+1} \text{ und } y^{(i)} \in \{0, 1\} \tag{35}$$

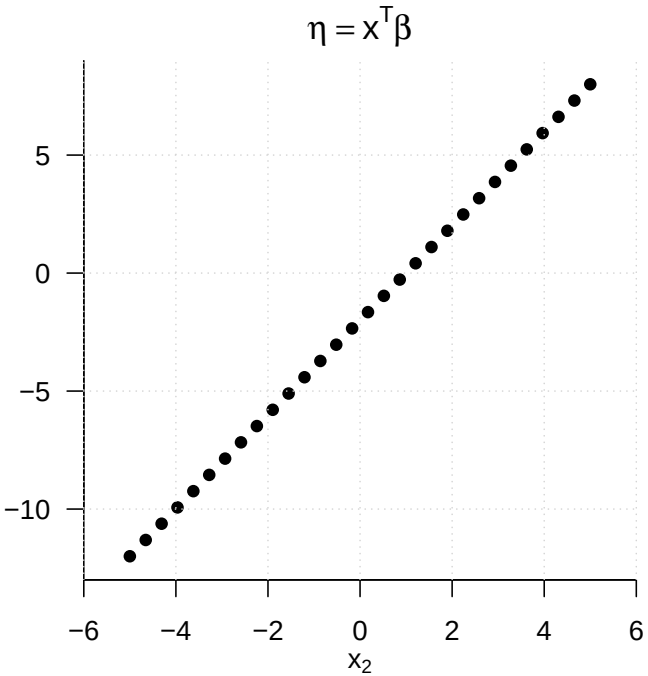
  eines LR Modells wie folgt erzeugt:

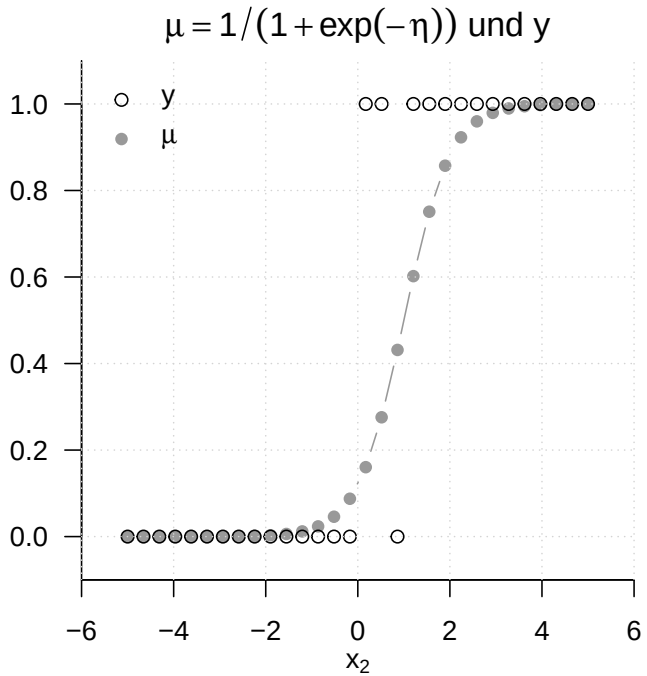
(1) Definition von $x^{(i)}$,
(2) Ziehen von $y^{(i)}$ aus $p(y) = B(y; \mu)$ mit Erwartungswertparameter $\mu = \frac{1}{1+\exp(-x^{(i)T}\beta)}$.

Datengeneration bei einfacher Logistischer Regression ($m = 1$)

```
# Modellparameter
m    = 1                                # Featurevektoredimensionalität
n    = 30                               # Anzahl Datenpunkte
x    = matrix(c(rep(1,n),               # Definition des erweiterten Featurevektors
                seq(-5,5, len = n)),
              nrow  = 2,
              byrow = TRUE)
beta = matrix(c(-2,2), nrow = 2)        # wahrer, aber unbekannter, Parametervektor
eta  = t(x) %*% beta                    # wahrer, aber unbekannter linearer Prädiktor
mu   = 1/(1+exp(-eta))                  # wahrer, aber unbekannter, Bernoulliparametervektor

# Dategeneration
set.seed(2)                             # Zufallsgeneratorzustand
y    = rep(NaN,2)                       # Datenarray
for(i in 1:n){
  y[i] = rbinom(1,1,mu[i])              # Bernoullivariablenrealisierung
}
```

Datengeneration bei einfacher Logistischer Regression ($m = 1, \beta = (-2, 2)^T, n = 30$)





Definition (Klassifikationsregel der Logistischen Regression)

$p(y)$ sei die WMF eines Logistischen Regressionsmodells. Dann ist die *Klassifikationregel* definiert als

$$\delta : \mathbb{R}^m \to \{0, 1\}, x \mapsto \delta(x) := \begin{cases} 0 & \text{für } p(y = 0) \geq p(y = 1) \\ 1 & \text{für } p(y = 0) < p(y = 1) \end{cases} \tag{36}$$

Bemerkung

- Es gilt

$$\delta(x) = 1 \Leftrightarrow p(y = 1) > p(y = 0) \Leftrightarrow p(y = 1) > 0.5. \tag{37}$$

Theorem (Lineare Diskriminanzfunktion der Logistischen Regression)

$p(y)$ sei die WMF eines Logistischen Regressionsmodells und $\beta \in \mathbb{R}^{m+1}$ sei der Parametervektor. Dann kann die Klassifikationsregel δ der Logistischen Regression als eine lineare Diskriminanzfunktion der Form

$$h : \mathbb{R}^m \to \{0, 1\}, x \mapsto h(x) := g(f(x)), \tag{38}$$

mit

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{39}$$

und

$$g : \mathbb{R} \to \{0, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} 0, & f(x) \geq 0 \\ 1, & f(x) < 0 \end{cases} \tag{40}$$

geschrieben werden, d.h. es gilt $\delta(x) = h(x)$ für alle $x \in \mathbb{R}^m$. Inbesondere gilt dabei

$$w_0 = \beta_1 \text{ und } w = (\beta_2, \ldots, \beta_{m+1})^T \tag{41}$$

Bemerkung

- CAVE: f bezeichnet hier eine linear-affine Funktion, nicht die standard logistic function.
- Ein Beweis ergibt sich in Analogie zum Fall der Linearen Diskriminanzanalyse

Theorem (Log Likelihood Funktion der Logistischen Regression)

$\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ sei ein Trainingsdatensatz aus erweiterten Featurevektoren und assoziierten Labelvariablenrealisierungen und f sei die standard logistic function. Dann hat die Log Likelihood Funktion der Logistischen Regression die Form

$$\ell : \mathbb{R}^m \to \mathbb{R}, \beta \mapsto \ell(\beta) := \sum_{i=1}^{n} y^{(i)} \ln(f({x^{(i)}}^T \beta)) + (1 - y^{(i)}) \ln(1 - f({x^{(i)}}^T \beta)).$$

Beweis

Wir halten zunächst fest, dass für u.i.v. Labelvariablen gilt, dass

$$\ell(\beta) := \ln p(y^{(1)}, \ldots, y^{(n)}) = \ln \prod_{i=1}^{n} p(y^{(i)}) = \sum_{i=1}^{n} \ln p(y^{(i)})$$

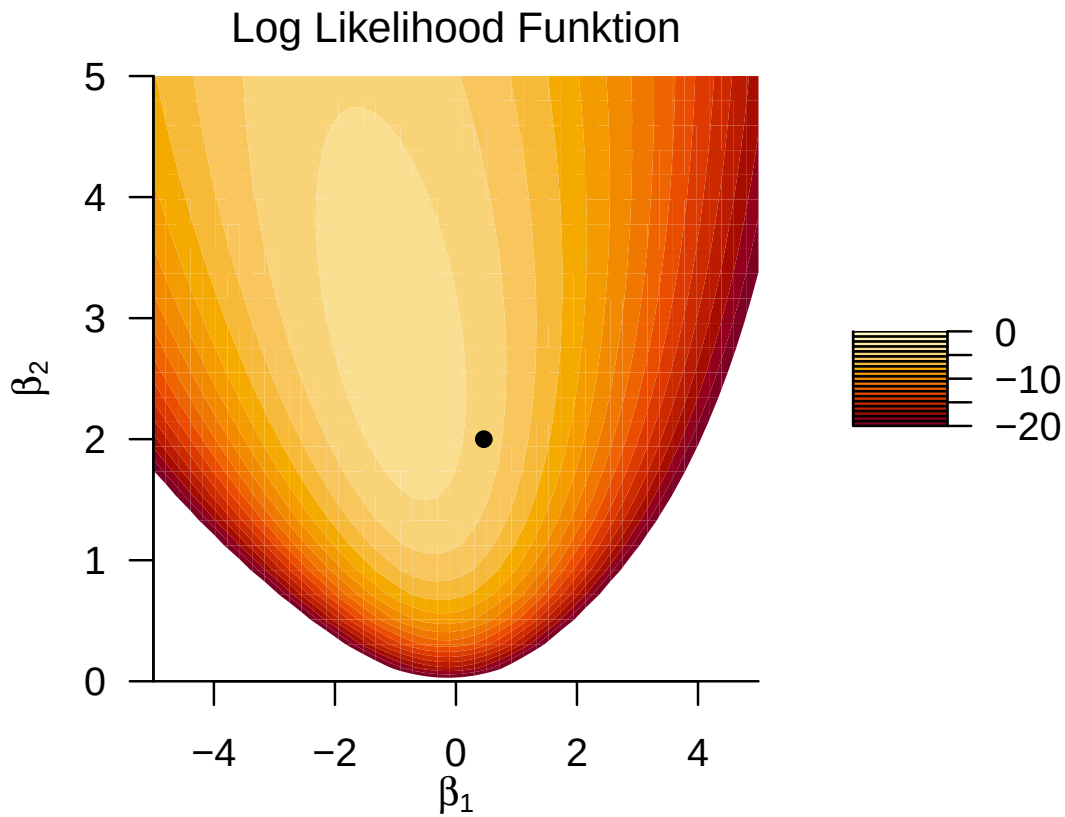
Mit der WMF der Bernoulliverteilung folgt dann

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^{n} \ln(f({x^{(i)}}^T \beta)^{y^{(i)}} (1 - f({x^{(i)}}^T \beta))^{1-y^{(i)}}) \\ &= \sum_{i=1}^{n} y^{(i)} \ln(f({x^{(i)}}^T \beta)) + (1 - y^{(i)}) \ln(1 - f({x^{(i)}}^T \beta)) \end{aligned}$$

Implementation der Log Likelihood Funktion

```
# Funktionsdefinitionen
# ------------------------------------------------------------------------------------------
# Standard Logistic Function
f = function(eta){
    return(1/(1 + exp(-eta)))
}

# Log Likelihood Function
llh = function(x,y,beta){
    n   = ncol(x)
    ell = 0
    for(i in 1:n){
        ell = ell + y[i]*log(f(t(x[,i]) %*% beta)) + (1-y[i])*log(1-f(t(x[,i]) %*% beta))
    }
    return(ell)
}

# Log Likelihood Funktion Auswertung
# ------------------------------------------------------------------------------------------
beta_min  = -5                                                    # beta Minimum
beta_max  =  5                                                    # beta Maximum
beta_res  = 5e1                                                   # beta Auflösung
beta_1    = seq(beta_min, beta_max, length.out = beta_res)        # beta_1 Raum
beta_2    = seq(beta_min, beta_max, length.out = beta_res)        # beta_2 Raum
ell       = matrix(rep(NaN, beta_res*beta_res), nrow = beta_res) # Log Likelihood Funktion Array
for(i in 1:beta_res){
    for(j in 1:beta_res){
        beta12   = matrix(c(beta_1[i], beta_2[j]), nrow = 2)
        ell[i,j] = llh(x,y,beta12)
    }
}
```

Visualisierung der Log Likelihood Funktion



Theorem (Gradientverfahren der Logistischen Regression)

$p(y)$ sei das Modell einer Logistischen Regression und $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ sei ein entsprechender Trainingsdatensatz. Dann kann eine Maximum Likelihood Schätzer $\hat{\beta}$ für den Parametervektor β des LRM durch folgendes Gradientenverfahren gewonnen werden:

(0) Wähle $\beta^0 \in \mathbb{R}, \alpha > 0, \delta > 0$

(1) Für $k = 0, 1, 2, \ldots$ bis zur Konvergenz setze

$$\beta^{(k+1)} := \beta^{(k)} + \alpha \nabla \ell(\beta^{(k)}). \tag{42}$$

wobei $\nabla \ell \left(\beta^k\right)$ den Gradienten der Log Likelihood Funktion der Logistischen Regression bezeichnet und die Form

$$\nabla \ell(\beta) = \begin{pmatrix} \frac{\partial}{\partial \beta_1} \ell(\beta) \\ \frac{\partial}{\partial \beta_2} \ell(\beta) \\ \vdots \\ \frac{\partial}{\partial \beta_m} \ell(\beta) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_1^{(i)} \\ \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_2^{(i)} \\ \vdots \\ \sum_{i=1}^{n} (y^{(i)} - f({x^{(i)}}^T \beta)) x_m^{(i)} \end{pmatrix} \tag{43}$$

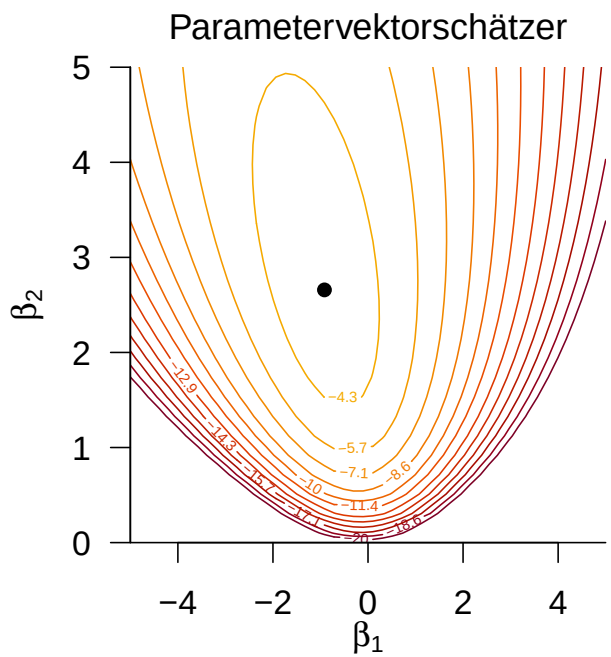
Bemerkungen

- Für einen Beweis verweisen wir auf den Appendix.

Bemerkungen (fortgeführt)

- Das reine Gradientenverfahren zum Lernen der Parameter eines LR Modells ist recht instabil.
- Iteratively Weighted Least Squares Verfahren werden zur ML Schätzung in GLMs bevorzugt (Green 1984).
- IWLS Verfahren nutzen Gradienten und Hesse-Matrix ähnlich wie Gauss-Newton Verfahren.
- R implementiert in der `glm()` ein IWLS Verfahren.

```
lr       = glm(y ~ x[2,], family = 'binomial')        # generalized linear model fit
beta_hat = lr$coefficients                            # Parametervektorschätzer
```



Prädiktion des Studienerfolgs

Nach Rudolf and Buse (2020) Kapitel 4

Datensatz zum Verhältnis psychologischer Diagnostik und Studienerfolg

$n = 30$ Studierende naturwissenschaftlicher Studiengänge

Featurevektor $x \in \mathbb{R}^4$

- x_1 Intelligenztestscore
- x_2 Mathematiktestscore
- x_3 Gewissenhaftigkeitscore
- x_4 Verträglichkeitscore

Label $y \in \{0, 1\}$

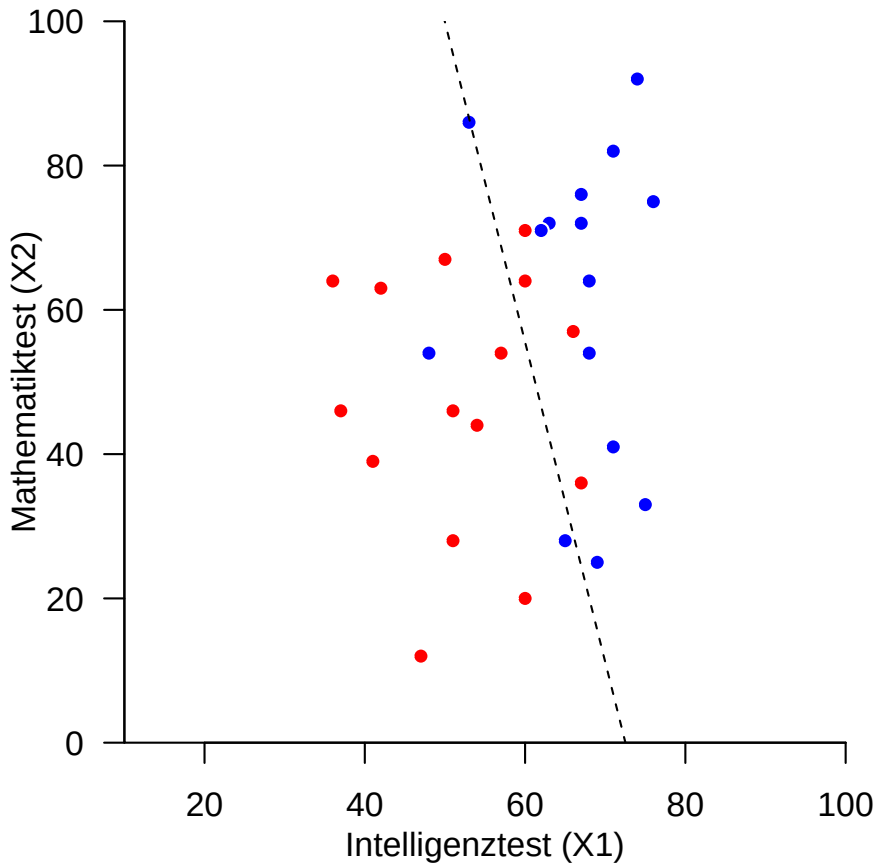
- 0: ungenügend (Studienabbruch aufgrund nicht bestandener Prüfungen)
- 1: gut (Abschlussnote besser als 2.5)

Parameterlernen, Prädiktion und lineare Diskriminanzfunktion

```
library(matlib)                                                          # Matrixtools
D            = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv")) # Datensatz
x            = as.matrix(D[1:2,])                                        # Featureselektion
y            = as.matrix(D[5,])                                          # Label
D            = rbind(x,y)                                                # Featureselected Datensatz
n            = ncol(x)                                                   # Datensatzgröße
m            = nrow(x)                                                   # Featurevektordimensionalität
lr           = glm(t(y) ~ t(x), family = 'binomial')                       # generalized linear model fit
beta_hat     = as.matrix(lr$coefficients, nrow = m + 1)                  # Parameterschätzung
w_0_hat      = beta_hat[1]                                               # \hat{w}
w_hat        = beta_hat[2:(m+1)]                                         # \hat{w_0}
x_1          = seq(min(x[1,]), max(x[1,]), len = 1e2)                    # x_1
x_2_hat      = -(w_hat[1]/w_hat[2])*x_1 - (w_0_hat/w_hat[2])             # \hat{h}
x_tilde      = rbind(1, x)
p_y          = 1/(1+exp(-t(x_tilde) %*% beta_hat))
delta        = as.numeric(p_y >= 0.5)
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.767
```

Parameterlernen und lineare Diskriminanzfunktion



Logistische Regression | Leave-One-Out Cross-Validation, $m = 2$

```
# Datensatz
D                = read.table(file.path(getwd(), "7_Daten", "studienerfolg.csv"))
x                = as.matrix(D[1:2,])
y                = as.matrix(D[5,])
D                = rbind(x,y)
K                = ncol(D)
p_y              = matrix(rep(NaN, ncol(y)), nrow = 1)
delta            = matrix(rep(NaN, ncol(y)), nrow = 1)

# K-fache Leave-One-Out Cross-Validation
for(k in 1:K){

    # Datensatzpartition
    x_train      = as.matrix(x[,-k])
    y_train      = as.matrix(y[,-k])
    x_test       = as.matrix(x[, k])
    y_test       = as.matrix(y[, k])

    # Trainingsdatensatz-basiertes Parameterlernen
    n            = ncol(x_train)
    m            = nrow(x_train)
    lr           = glm(y_train ~ t(x_train), family = 'binomial')
    beta_hat     = as.matrix(lr$coefficients, nrow = m + 1)

    # Prädiktion
    x_test_tilde = rbind(1, x_test)
    p_y[k]       = 1/(1+exp(-t(x_test_tilde) %*% beta_hat))
    delta[k]     = as.numeric(p_y[k] >= 0.5)
}
cat("Accuracy: ", mean(delta == y))
```

```
> Accuracy:  0.733
```

Logistische Regression | Leave-One-Out Cross-Validation, $m = 2$

k	x_1	x_2	p(y = 1)	delta(x)	y
1	54	44	0.2	0	0
2	60	20	0.2	0	0
3	67	36	0.7	1	0
4	41	39	0.0	0	0
5	66	57	0.8	1	0
6	51	28	0.1	0	0
7	51	46	0.1	0	0
8	37	46	0.0	0	0
9	57	54	0.4	0	0
10	47	12	0.0	0	0
11	50	67	0.2	0	0
12	42	63	0.0	0	0
13	60	64	0.6	1	0
14	36	64	0.0	0	0
15	60	71	0.7	1	0
16	71	41	0.8	1	1
17	65	28	0.3	0	1
18	67	76	0.9	1	1
19	68	54	0.8	1	1
20	75	33	0.8	1	1
21	71	82	1.0	1	1
22	68	64	0.9	1	1
23	63	72	0.8	1	1
24	48	54	0.0	0	1
25	53	86	0.3	0	1
26	62	71	0.7	1	1
27	69	25	0.5	0	1
28	67	72	0.9	1	1
29	74	92	1.0	1	1
30	76	75	1.0	1	1

Prediction Accuracy = 0.73.

Logistische Regression | Leave-One-Out Cross-Validation, $m = 3$

k	x_1	x_2	x_3	p(y = 1)	delta(x)	y
1	54	44	31	0.2	0	0
2	60	20	33	0.2	0	0
3	67	36	26	0.7	1	0
4	41	39	31	0.0	0	0
5	66	57	34	0.9	1	0
6	51	28	42	0.1	0	0
7	51	46	34	0.1	0	0
8	37	46	36	0.0	0	0
9	57	54	28	0.3	0	0
10	47	12	34	0.0	0	0
11	50	67	27	0.2	0	0
12	42	63	26	0.0	0	0
13	60	64	29	0.6	1	0
14	36	64	36	0.0	0	0
15	60	71	42	1.0	1	0
16	71	41	37	0.9	1	1
17	65	28	33	0.3	0	1
18	67	76	38	0.9	1	1
19	68	54	34	0.8	1	1
20	75	33	25	0.7	1	1
21	71	82	32	1.0	1	1
22	68	64	34	0.9	1	1
23	63	72	32	0.8	1	1
24	48	54	41	0.0	0	1
25	53	86	27	0.1	0	1
26	62	71	31	0.7	1	1
27	69	25	32	0.5	0	1
28	67	72	31	0.9	1	1
29	74	92	35	1.0	1	1
30	76	75	37	1.0	1	1

Prediction Accuracy = 0.73.

Logistische Regression | Leave-One-Out Cross-Validation, $m = 4$

k	x_1	x_2	x_3	x_4	p(y = 1)	delta(x)	y
1	54	44	31	60	0.0	0	0
2	60	20	33	31	0.5	0	0
3	67	36	26	54	0.4	0	0
4	41	39	31	26	0.0	0	0
5	66	57	34	56	0.6	1	0
6	51	28	42	23	0.3	0	0
7	51	46	34	40	0.1	0	0
8	37	46	36	31	0.0	0	0
9	57	54	28	49	0.2	0	0
10	47	12	34	41	0.0	0	0
11	50	67	27	53	0.0	0	0
12	42	63	26	36	0.0	0	0
13	60	64	29	40	0.9	1	0
14	36	64	36	21	0.1	0	0
15	60	71	42	40	1.0	1	0
16	71	41	37	30	1.0	1	1
17	65	28	33	38	0.4	0	1
18	67	76	38	28	1.0	1	1
19	68	54	34	53	0.5	1	1
20	75	33	25	54	0.6	1	1
21	71	82	32	49	1.0	1	1
22	68	64	34	33	1.0	1	1
23	63	72	32	36	0.9	1	1
24	48	54	41	34	0.0	0	1
25	53	86	27	35	0.4	0	1
26	62	71	31	53	0.4	0	1
27	69	25	32	25	0.9	1	1
28	67	72	31	28	1.0	1	1
29	74	92	35	50	1.0	1	1
30	76	75	37	12	1.0	1	1

Prediction Accuracy = 0.77.

Vorbemerkungen

Lineare Diskriminanzanalyse

Logistische Regression

Selbstkontrollfragen

Appendix

Selbstkontrollfragen

1. Definieren Sie den Begriff des Binären Klassifikationsdatensatzes.
2. Definieren Sie die Bernoulli Verteilung.
3. Definieren Sie das Modell der Linearen Diskriminanzanalyse.
4. Erläutern Sie die Erzeugung von Daten unter dem Modell der Linearen Diskriminanzanalyse.
5. Erläutern Sie den Begriff der Inferenz im Modell der Linearen Diskriminanzanalyse.
6. Definieren Sie die Klassifikationsregel der Linearen Diskriminanzanalyse.
7. Wie werden die Parameter eines Linearen Diskriminanzanalysemodells gelernt?
8. Erläutern Sie den Ablauf einer Leave-One-Out Cross-Validation mithilfe der Linearen Diskriminanzanalyse.
9. Definieren Sie die standard logistic function.
10. Definieren Sie das Modell der Logistischen Regression.
11. Erläutern Sie die Erzeugung von Daten unter dem Modell der Logistischen Regression.
12. Warum gibt es im Modell der Logistischen Regression keine Inferenz?
13. Definieren Sie die Klassifikationsregel der Logistischen Regression.
14. Wie werden die Parameter eines Logistischen Regressionsmodells gelernt?
15. Erläutern Sie den Ablauf einer Leave-One-Out Cross-Validation mithilfe der Logistischen Regression.

Beweis des LDA Inferenz Theorems

Wir halten zunächst fest, dass

$$\begin{aligned} p(y=1|x) &= \frac{p(x,y=1)}{p(x,y=0)+p(x,y=1)} \\ &= \frac{\frac{p(x,y=1)}{p(x,y=1)}}{\frac{p(x,y=0)}{p(x,y=1)}+\frac{p(x,y=1)}{p(x,y=1)}} \\ &= \frac{1}{1+\frac{p(x,y=0)}{p(x,y=1)}} \\ &= \frac{1}{1+\exp\left(\ln\left(\frac{p(x,y=0)}{p(x,y=1)}\right)\right)} \\ &= \frac{1}{1+\exp\left(-\ln\left(\frac{p(x,y=1)}{p(x,y=0)}\right)\right)} \end{aligned} \tag{44}$$

Mit der Definition des LDA Modells gilt dann

$$p(x,y=1) = p(x|y=1)p(y=1) = N(x;\mu_1,\Sigma)\mu \tag{45}$$

und

$$p(x,y=0) = p(x|y=0)p(y=0) = N(x;\mu_0,\Sigma)(1-\mu) \tag{46}$$

Beweis des LDA Inferenz Theorems (fortgeführt)

Wir erhalten also

$$
\begin{aligned}
&= \ln\left(\frac{p(x, y=1)}{p(x, y=0)}\right) \\
&= \ln\left(\frac{N(x; \mu_1, \Sigma)\mu}{N(x; \mu_0, \Sigma)(1-\mu)}\right) \\
&= \ln\left(N(x; \mu_1, \Sigma)\mu\right) - \ln\left(N(x; \mu_0, \Sigma)(1-\mu)\right) \\
&= \ln(\mu) + \ln N(x; \mu_1, \Sigma) - \ln(1-\mu) - \ln N(x; \mu_0, \Sigma) \\
&= \ln\mu - \ln(1-\mu) - \frac{m}{2}\ln 2\pi - \ln|\Sigma| - \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\
&\qquad + \frac{m}{2}\ln 2\pi + \ln|\Sigma| + \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) \\
&= -\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) + \frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \ln\mu - \ln(1-\mu) \\
&= -\frac{1}{2}x^T\Sigma^{-1}x + x^T\Sigma^{-1}\mu_1 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_0 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \ln\left(\frac{\mu}{1-\mu}\right) \\
&= \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 - x^T\Sigma^{-1}(\mu_0-\mu_1) + \ln\left(\frac{\mu}{1-\mu}\right) \\
&= \begin{pmatrix}1 & x^T\end{pmatrix}\begin{pmatrix}\frac{1}{2}\left(\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1\right) + \ln\left(\frac{\mu}{1-\mu}\right) \\ -\Sigma^{-1}(\mu_0-\mu_1)\end{pmatrix} =: \tilde{x}^T\beta
\end{aligned}
$$

Beweis des LDA Diskriminanzfunktion Theorems

Wir zeigen die Äquivalenz von $\delta(x) = 0$ und $f(x) \geq 0$ womit der Rest des Theorems sofort folgt. Es ergibt sich

$$
\begin{aligned}
\delta(x) &= 0 \\
\Leftrightarrow p(y=0|x) &\geq p(y=1|x) \\
\Leftrightarrow \frac{p(y=0|x)}{p(y=1|x)} &\geq 1 \\
\Leftrightarrow \ln\left(\frac{p(y=0|x)}{p(y=1|x)}\right) &\geq \ln 1 \\
\Leftrightarrow \ln\left(\frac{1 - \frac{1}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\frac{\frac{1+\exp(-\tilde{x}^T\beta)}{1+\exp(-\tilde{x}^T\beta)} - \frac{1}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\frac{\frac{\exp(-\tilde{x}^T\beta)}{1+\exp(-\tilde{x}^T\beta)}}{\frac{1}{1+\exp(-\tilde{x}^T\beta)}}\right) &\geq 0 \\
\Leftrightarrow \ln\left(\exp(-\tilde{x}^T\beta)\right) &\geq 0
\end{aligned}
$$

Beweis des LDA Diskriminanzfunktion Theorems (fortgeführt)

Es ergibt sich also

$$
\begin{aligned}
\delta(x) = 0 \\
\Leftrightarrow -\tilde{x}^T \beta \geq 0 \\
\Leftrightarrow -\begin{pmatrix} 1 & x^T \end{pmatrix} \begin{pmatrix} \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln(1-\mu) - \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln \mu \\ -\Sigma^{-1}(\mu_0 - \mu_1) \end{pmatrix} \geq 0 \\
\Leftrightarrow x^T \Sigma^{-1}(\mu_0 - \mu_1) + \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln(1-\mu) - \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln \mu \geq 0 \\
\Leftrightarrow \left(x^T \Sigma^{-1}(\mu_0 - \mu_1)\right)^T + \frac{1}{2}\mu_1 \Sigma \mu_1 + \ln(1-\mu) - \frac{1}{2}\mu_0 \Sigma^{-1} \mu_0 - \ln \mu \geq 0 \\
\Leftrightarrow (\mu_0 - \mu_1)^T \Sigma^{-1} x + \frac{1}{2}(\mu_1 - \mu_0)^T \Sigma^{-1}(\mu_1 - \mu_0) + \ln\left(\frac{1-\mu}{\mu}\right) \geq 0 \\
\Leftrightarrow: w^T x + w_0 \geq 0
\end{aligned}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems

(1) Formulierung der Log Likelihood Funktion

$$
\begin{aligned}
&\ell(\mu, \mu_0, \mu_1, \Sigma) := \\
&\ln \prod_{i=1}^{n} p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)}, y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)} | y^{(i)}\right) p\left(y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln p\left(x^{(i)} | y^{(i)}\right) + \ln p\left(y^{(i)}\right) \\
&= \sum_{i=1}^{n} \ln \left(N(x^{(i)}; \mu_0, \Sigma)\right)^{1-y^{(i)}} \left(N(x^{(i)}; \mu_1, \Sigma)\right)^{y^{(i)}} + \ln \left(\mu^{y^{(i)}} (1-\mu)^{1-\mu^{y^{(i)}}}\right)
\end{aligned}
$$

Appendix

$$
\begin{aligned}
&\ell(\mu, \mu_0, \mu_1, \Sigma) = \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \ln N\left(x^{(i)}; \mu_0, \Sigma\right) + y^{(i)} \ln N\left(x^{(i)}; \mu_1, \Sigma\right) + y^{(i)} \ln \mu + \left(1 - y^{(i)}\right) \ln(1 - \mu) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \ln \mu + \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \ln(1 - \mu).
\end{aligned}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

(2) Gradient der Log Likelihood Funktion

Der Gradient der Log Likelihood Funktion des LDA Modells besteht aus den partiellen Ableitungen von ℓ hinsichtlich von μ, μ_0, μ_1 und Σ. Wie unten gezeigt ergibt er sich als

$$\nabla \ell(\mu, \mu_0, \mu_1, \Sigma) = \begin{pmatrix} \frac{\partial}{\partial \mu} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \mu_0} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \mu_1} \ell(\mu, \mu_0, \mu_1, \Sigma) \\ \frac{\partial}{\partial \Sigma} \ell(\mu, \mu_0, \mu_1, \Sigma) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{\mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \\ -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(\left(x^{(i)} - \mu_0 \right)^T \Sigma^{-1} \right) \\ -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} \left(\left(x^{(i)} - \mu_1 \right)^T \Sigma^{-1} \right) \\ \frac{n}{2} \Sigma - \frac{1}{2} \sum_{i=1}^{n} \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}} \right) \left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}} \right)^T \end{pmatrix}.$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Für die partielle Ableitung hinsichtlich μ_0 und ähnlich für μ_1 ergibt sich

$$
\begin{aligned}
\frac{\partial}{\partial \mu_0} \ell(\mu, \mu_0, \mu_1, \Sigma) &= \frac{\partial}{\partial \mu_0} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \frac{\partial}{\partial \mu_0} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \frac{\partial}{\partial \mu_0} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right) \\
&= -\frac{1}{2} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(\left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right). \\
&= -\frac{1}{2} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(\left(x^{(i)} - \mu_0\right)^T \Sigma^{-1}\right).
\end{aligned}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Für die partielle Ableitung hinsichtlich Σ ergibt sich

$$
\begin{aligned}
&\frac{\partial}{\partial \Sigma} \ell(\mu, \mu_1, \mu_0, \Sigma) \\
&= \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \frac{\partial}{\partial \Sigma} \sum_{i=1}^{n} y^{(i)} \left(-\frac{n}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma| - \frac{1}{2} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right) \\
&= \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_0\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_0\right)\right) \\
&\quad + \sum_{i=1}^{n} y^{(i)} \left(-\frac{1}{2} \frac{\partial}{\partial \Sigma} \ln |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \left(x^{(i)} - \mu_1\right)^T \Sigma^{-1} \left(x^{(i)} - \mu_1\right)\right)
\end{aligned}
\tag{47}
$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

... und damit

$$\begin{aligned}\frac{\partial}{\partial \Sigma}\ell(\mu, \mu_1, \mu_0, \Sigma) = & \sum_{i=1}^{n}\left(1 - y^{(i)}\right)\left(-\frac{1}{2}\Sigma - \frac{1}{2}\left(x^{(i)} - \mu_0\right)\left(x^{(i)} - \mu_0\right)^T\right) \\ & + \sum_{i=1}^{n} y^{(i)}\left(-\frac{1}{2}\Sigma - \frac{1}{2}\left(x^{(i)} - \mu_1\right)\left(x^{(i)} - \mu_1\right)^T\right) \\ = & \frac{n}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1\{y^{(i)}=1\}}\right)\left(x^{(i)} - \mu_{1\{y^{(i)}=1\}}\right)^T.\end{aligned} \tag{48}$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Für die partielle Ableitung hinsichtlich μ ergibt sich

$$\begin{aligned}
\frac{\partial}{\partial \mu} \ell(\mu, \mu_1, \mu_0, \Sigma) &= \frac{\partial}{\partial \mu} \left(\sum_{i=1}^{n} y^{(i)} \ln \mu + \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \ln(1 - \mu) \right) \\
&= \sum_{i=1}^{n} y^{(i)} \frac{\partial}{\partial \mu} \ln \mu + \sum_{i=1}^{n} \left(1 - y^{(i)}\right) \frac{\partial}{\partial \mu} \ln(1 - \mu) \\
&= \frac{1}{\mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1 - \mu} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}.
\end{aligned}$$

(4) Auflösen der Maximum Likelihood Gleichungen

Nullsetzen der partiellen Ableitungen des Gradienten der Log Likelihood Funktion und Auflösen der resultierenden Log Likelihood Gleichungen ergibt dann die Maximum Likelihood Schätzer des LDA Modells.

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der ersten Gradientenkomponente ergibt

$$\frac{1}{\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=1\}} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} = 0$$

$$\Leftrightarrow \frac{1}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \frac{1}{1-\hat{\mu}} \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0$$

$$\Leftrightarrow \frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - \sum_{i=1}^{n} \left(1 - y^{(i)}\right) = 0$$

$$\Leftrightarrow \frac{1-\hat{\mu}}{\hat{\mu}} \sum_{i=1}^{n} y^{(i)} - n + \sum_{i=1}^{n} y^{(i)} = 0$$

$$\Leftrightarrow (1-\hat{\mu}) \sum_{i=1}^{n} y^{(i)} - \hat{\mu} n + \hat{\mu} \sum_{i=1}^{n} y^{(i)} = 0$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

... und weiter

$$\Leftrightarrow (1-\hat{\mu})\sum_{i=1}^{n} y^{(i)} - \hat{\mu}n + \hat{\mu}\sum_{i=1}^{n} y^{(i)} = 0$$

$$\Leftrightarrow (1-\hat{\mu}+\hat{\mu})\sum_{i=1}^{n} y^{(i)} = \hat{\mu}n$$

$$\Leftrightarrow \hat{\mu}n = \sum_{i=1}^{n} y^{(i)}$$

$$\Leftrightarrow \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} 1_{\{y^{(i)}=1\}}.$$

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der zweiten Gradientenkomponente ergibt

$$\sum_{i=1}^{n} \left(1 - y^{(i)}\right) \left(\left(x^{(i)} - \hat{\mu}_0\right)^T \Sigma^{-1}\right) = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \left(x^{(i)} - \hat{\mu}_0\right)^T = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)} - \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = 0$$

$$\Leftrightarrow \sum_{i=1}^{n} 1_{\{y^{(i)}=0\}} \hat{\mu}_0 = \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)}$$

$$\Leftrightarrow \hat{\mu}_0 = \frac{1}{\sum_{i=1}^{n} 1_{\{y^{(i)}=0\}}} \sum_{i=1}^{n} 1_{\{y^{(i)}\}} x^{(i)}.$$

Nullsetzen der dritten Gradientenkomponente ergibt dann in ähnlicher Weise den Maximum Likelihood Schätzer $\hat{\mu}_1$.

Beweis des LDA Maximum Likelihood Schätzer Theorems (fortgeführt)

Nullsetzen der vierten Gradientenkomponente ergibt dann schließlich

$$
\begin{aligned}
0 &= \frac{n}{2}\hat{\Sigma} - \frac{1}{2}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T \\
\Leftrightarrow\ n\hat{\Sigma} &= \sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T \\
\hat{\Sigma} &= \frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)\left(x^{(i)} - \mu_{1_{\{y^{(i)}=1\}}}\right)^T
\end{aligned}
$$

Beweis des LR Gradientenverfahrens

Um die $j$te partielle Ableitung der Log Likelihood Funktion zu bestimmen, halten wir zunächst fest, dass sich die Ableitung der logistic function f hinsichtlich η zu

$$\frac{d}{d\eta} f : \mathbb{R} \to \mathbb{R}, \eta \mapsto \frac{d}{d\eta} f(\eta) = f(\eta)(1 - f(\eta)) \tag{49}$$

ergibt. Dies kann wie folgt eingesehen werden:

$$\begin{aligned}
\frac{d}{d\eta} f(\eta) &= \frac{d}{d\eta} \left(1 + \exp(-\eta)\right)^{-1} \\
&= -\left(1 + \exp(-\eta)\right)^{-2} \cdot \exp(-\eta) \cdot (-1) \\
&= \frac{\exp(-\eta)}{(1 + \exp(-\eta))^2} \\
&= \frac{1 + \exp(-\eta) - 1}{(1 + \exp(-\eta))^2} \\
&= \frac{1 + \exp(-\eta)}{(1 + \exp(-\eta))^2} - \frac{1}{(1 + \exp(-\eta))^2} \\
&= \frac{1}{1 + \exp(-\eta)} - \frac{1}{(1 + \exp(-\eta))^2} \\
&= \frac{1}{1 + \exp(-\eta)} \left(1 - \frac{1}{1 + \exp(-\eta)}\right) \\
&= f(\eta)(1 - f(\eta))
\end{aligned}$$

Beweis des LR Gradientenverfahrens (fortgeführt)

Damit ergibt sich dann für $\frac{\partial}{\partial \beta_j} \ell, j = 1, \ldots, m$:

$$
\begin{aligned}
&\frac{\partial}{\partial \beta_j} \ell(\beta) \\
&= \frac{\partial}{\partial \beta_j} \left(\sum_{i=1}^{n} y^{(i)} \ln \left(f \left({x^{(i)}}^T \beta \right) \right) + \left(1 - y^{(i)} \right) \ln \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \right) \\
&= \sum_{i=1}^{n} y^{(i)} \frac{\partial}{\partial \beta_j} \left(\ln \left(f \left({x^{(i)}}^T \beta \right) \right) \right) + \left(1 - y^{(i)} \right) \frac{\partial}{\partial \beta_j} \left(\ln \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \right) \\
&= \sum_{i=1}^{n} y^{(i)} \frac{1}{f \left({x^{(i)}}^T \beta \right)} \left(\frac{\partial}{\partial \beta_j} \left(f \left({x^{(i)}}^T \beta \right) \right) \right) + \left(1 - y^{(i)} \right) \frac{1}{1 - f \left({x^{(i)}}^T \beta \right)} \frac{\partial}{\partial \beta_j} \left(1 - f \left({x^{(i)}}^T \beta \right) \right) \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f \left({x^{(i)}}^T \beta \right)} - \left(1 - y^{(i)} \right) \frac{1}{1 - f \left({x^{(i)}}^T \beta \right)} \right) \frac{\partial}{\partial \beta_j} \left(f \left({x^{(i)}}^T \beta \right) \right)
\end{aligned}
$$

Appendix

Beweis des LR Gradientenverfahrens (fortgeführt)

$$
\begin{aligned}
\frac{\partial}{\partial \beta_j} \ell(\beta) &= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{1}{1 - f\left(x^{(i)^T} \beta\right)} \right) \\
&\quad \times f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right) \frac{\partial}{\partial \beta_j} \left(x^{(i)^T} \beta\right) \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{1}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{1}{1 - f\left(x^{(i)^T} \beta\right)} \right) f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} \frac{f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right)}{f\left(x^{(i)^T} \beta\right)} - \left(1 - y^{(i)}\right) \frac{f\left(x^{(i)^T} \beta\right) \left(1 - f\left(x^{(i)^T} \beta\right)\right)}{1 - f\left(x^{(i)^T} \beta\right)} \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} \left(1 - f\left(x^{(i)^T} \beta\right)\right) - \left(1 - y^{(i)}\right) f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} - y^{(i)} f\left(x^{(i)^T} \beta\right) - f\left(x^{(i)^T} \beta\right) + y^{(i)} f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)} \\
&= \sum_{i=1}^{n} \left(y^{(i)} - f\left(x^{(i)^T} \beta\right) \right) x_j^{(i)}.
\end{aligned}
$$

References

Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. 2018. "Machine Learning Approaches for Clinical Psychology and Psychiatry." *Annual Review of Clinical Psychology* 14 (1): 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037.

Green, P. J. 1984. "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and Some Robust and Resistant Alternatives." *Journal of the Royal Statistical Society: Series B (Methodological)* 46 (2): 149–70. https://doi.org/10.1111/j.2517-6161.1984.tb01288.x.

Rudolf, Matthias, and Johannes Buse. 2020. *Multivariate Verfahren*. Göttingen: Hogrefe.

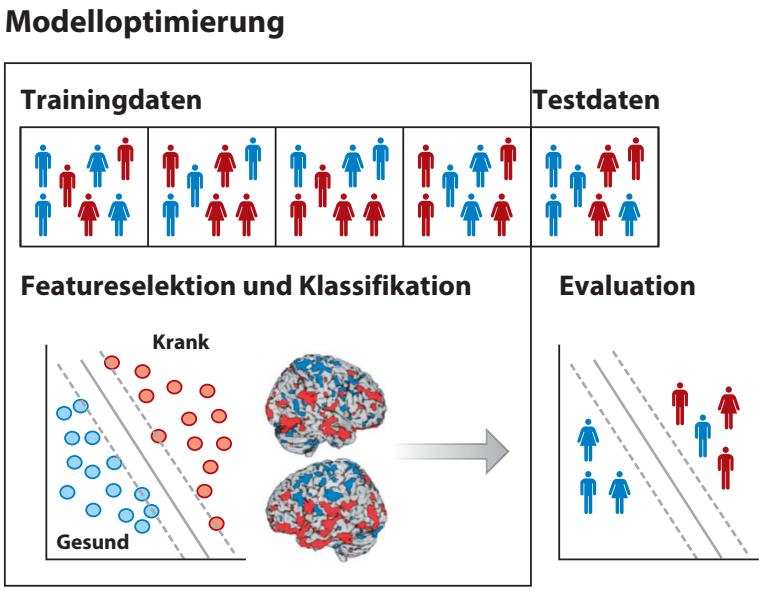
Multivariate Datenanalyse

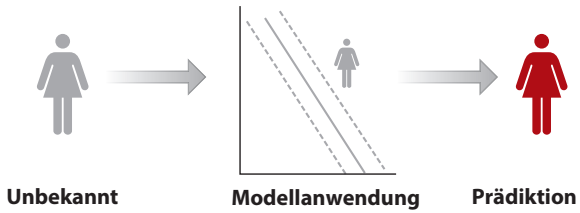
MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(8) Support Vektor Maschinen

Struktur der Prädiktiven Modellierung





nach Dwyer, Falkai, and Koutsouleris (2018)

Rhethorik der Prädiktiven Modellierung

Daten	Trainingsdaten und Testdaten
Statistisches Modell	Modell, Machine Learning Algorithmus
Schätzen von Parametern	Trainieren des Modells, Lernen von Parametern, Supervised Learning

Definition (Binärer Klassifikationstrainingdatensatz)

Ein *binärer Klassifikationsdatensatz*

$$\mathcal{D} := \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(n)}, y^{(n)})\} \tag{1}$$

ist eine Menge von n *Trainingsdatenpunkten*

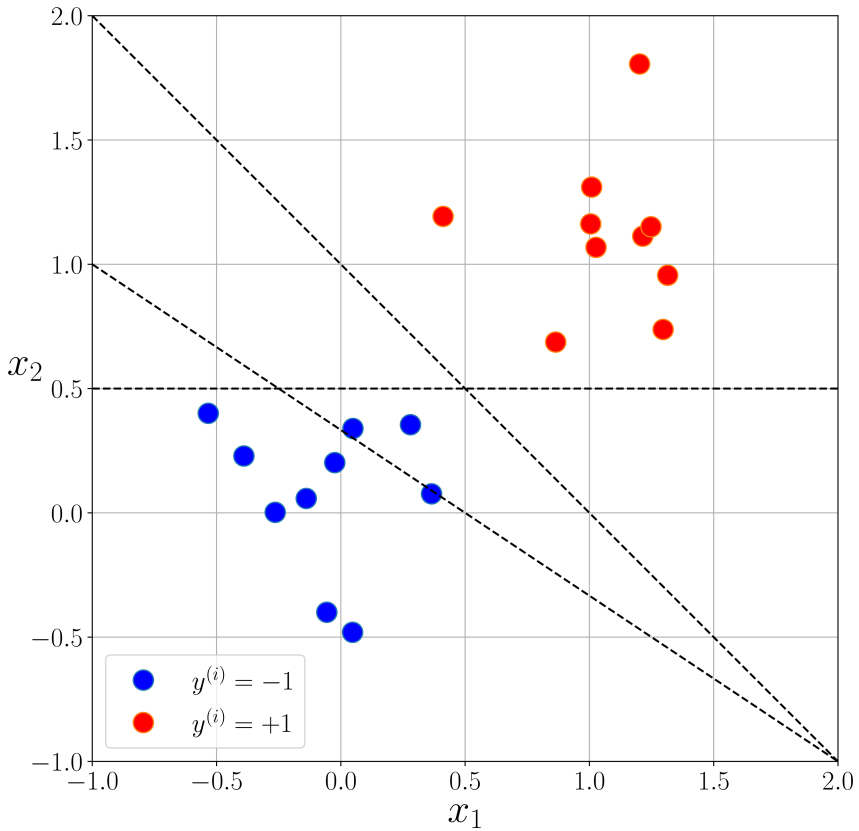
$$(x^{(i)}, y^{(i)}) \text{ mit } x^{(i)} \in \mathbb{R}^m \text{ und } y^{(i)} \in \{-1, 1\} \text{ für } i = 1, ..., n, \tag{2}$$

wobei $x^{(i)}$ m*-dimensional Featurevektor* und $y^{(i)}$ *Label* genannt wird

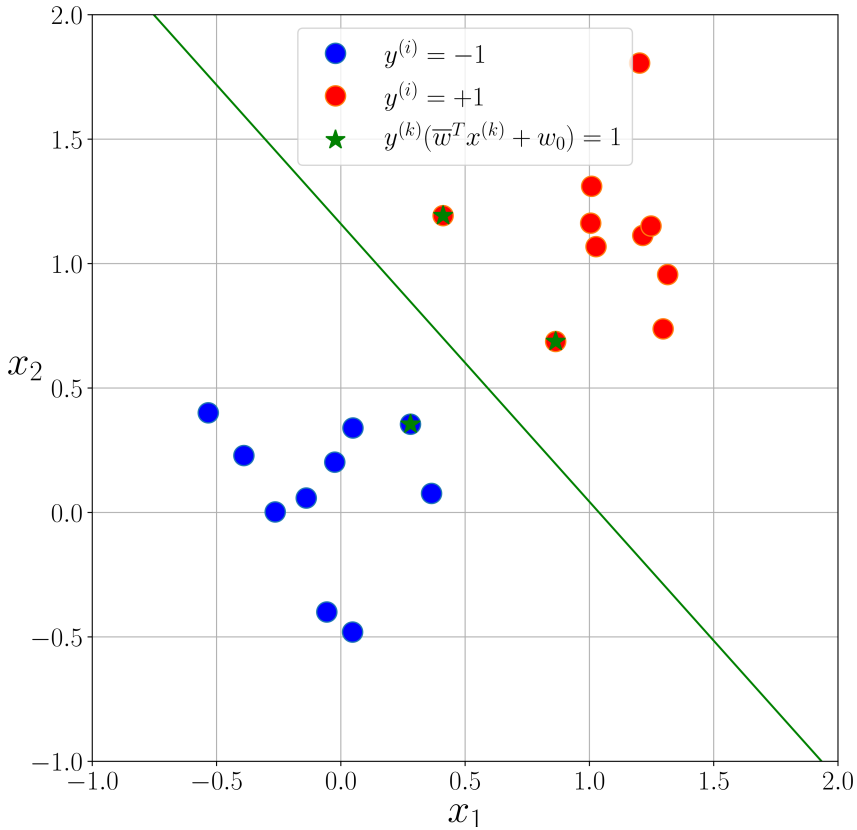
Bemerkung

- $y^{(i)} \in \{-1, 1\}$ bezeichnet die Klassenzugehörigkeit des Featurevektors $x^{(i)} \in \mathbb{R}^m$.
- Man beachte, dass hier $y^{(i)} \in \{-1, 1\}$, wohingegen bei LDA/LR $y^{(i)} \in \{0, 1\}$.

Welche lineare Diskriminanzfunktion (Hyperebene) soll hier man wählen ?



Nach der Theorie der Maximum Margin Support Vektor Maschinen diese:



Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Definition (Lineare Diskriminanzfunktion)

Eine *Lineare Diskriminanzfunktion* ist eine multivariate reellwertige Funktion der Form

$$h : \mathbb{R}^m \to \{-1, +1\}, x \mapsto h(x) := g(f(x)), \tag{3}$$

wobei

- f eine multivariate reellwertige, parameterabhängige linear-affine Function der Form

  $$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0, \tag{4}$$

  mit *Parametervektor* $w \in \mathbb{R}^m$ und *Biasparameter* $w_0 \in \mathbb{R}$ ist und
- g eine univariate reellwertige, parameterunabhängige *Klassifikationsfunktion* der Form

  $$g : \mathbb{R} \to \{-1, 1\}, f(x) \mapsto g(f(x)) := \begin{cases} -1, & f(x) < 0 \\ +1, & f(x) \geq 0 \end{cases} \tag{5}$$

ist. Eine LDF induziert im Featurevektorenraum $\mathbb{R}^m$

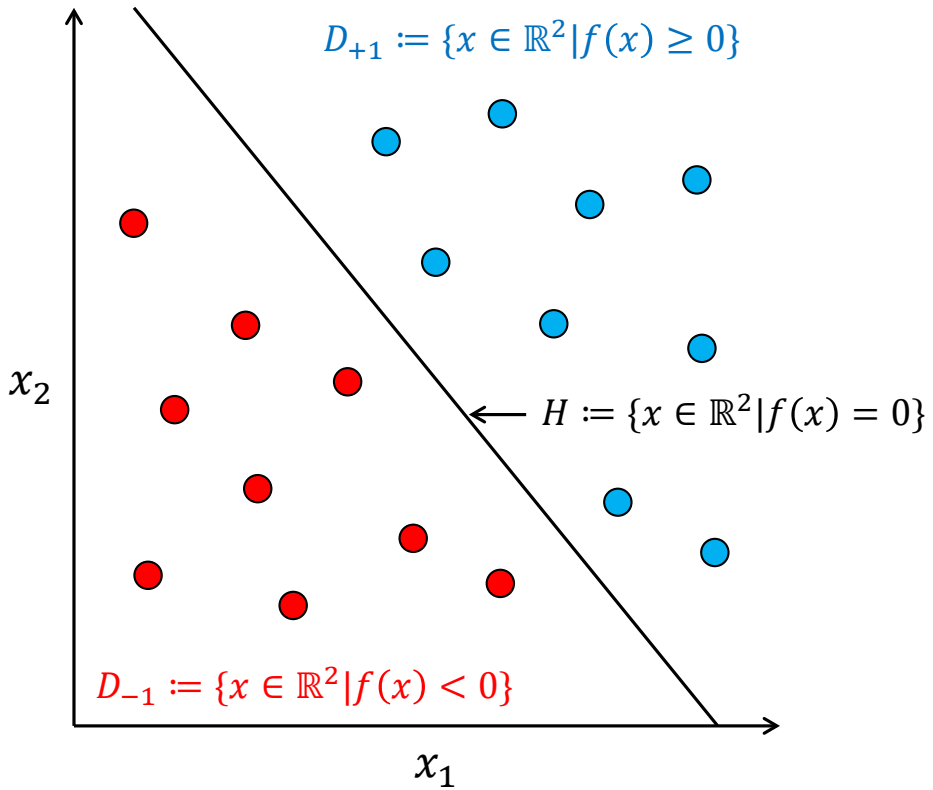
- eine *Entscheidungsgrenze* $H := \{x \in \mathbb{R}^m | f(x) = 0\}$, genannt *Hyperebene*,
- eine *Entscheidungsregion* $D_{-1} := \{x \in \mathbb{R}^m | f(x) < 0\}$, und
- eine *Entscheidungsregion* $D_{+1} := \{x \in \mathbb{R}^m | f(x) \geq 0\}$.

Geometrie linearer Diskriminanzfunktionen

$$m := 2$$

$$\Rightarrow x^{(i)} = \left(x_1^{(i)}, x_2^{(i)}\right)^T \in \mathbb{R}^2$$

- $x^{(i)}$ mit $y^{(i)} = -1$
- $x^{(i)}$ mit $y^{(i)} = +1$



Graphgleichungen für Hyperebenen in $\mathbb{R}^2$

$$f(x) = 0 \Leftrightarrow w^T x + w_0 = 0 \Leftrightarrow w_1 x_1 + w_2 x_2 + w_0 = 0 \Leftrightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2} \qquad (6)$$

Theorem (Geometrie linearer Diskriminanzfunktionen)

$$f : \mathbb{R}^m \to \mathbb{R}, x \mapsto f(x) := w^T x + w_0 \tag{7}$$

sei eine multivariate reellwertige, parameterabhängige linear-affine Funktion und

$$H := \{x \in \mathbb{R}^m | f(x) = 0\} \subset \mathbb{R}^m \tag{8}$$

sei die zugehörige Hyperebene. Weiterhin sei

$$||v||_2 := \sqrt{v^T v} \tag{9}$$

die Euklidische Länge von $v \in \mathbb{R}^m$. Dann gelten die folgenden geometrischen Beziehungen:

(1) w ist zu jedem Vektor, der in der Richtung von H orientiert ist, orthogonal.

(2) Der minimale Euklidische Abstand d zwischen $x \in \mathbb{R}^m$ und einem Punkt auf H ist

$$d = \frac{1}{||w||_2} f(x). \tag{10}$$

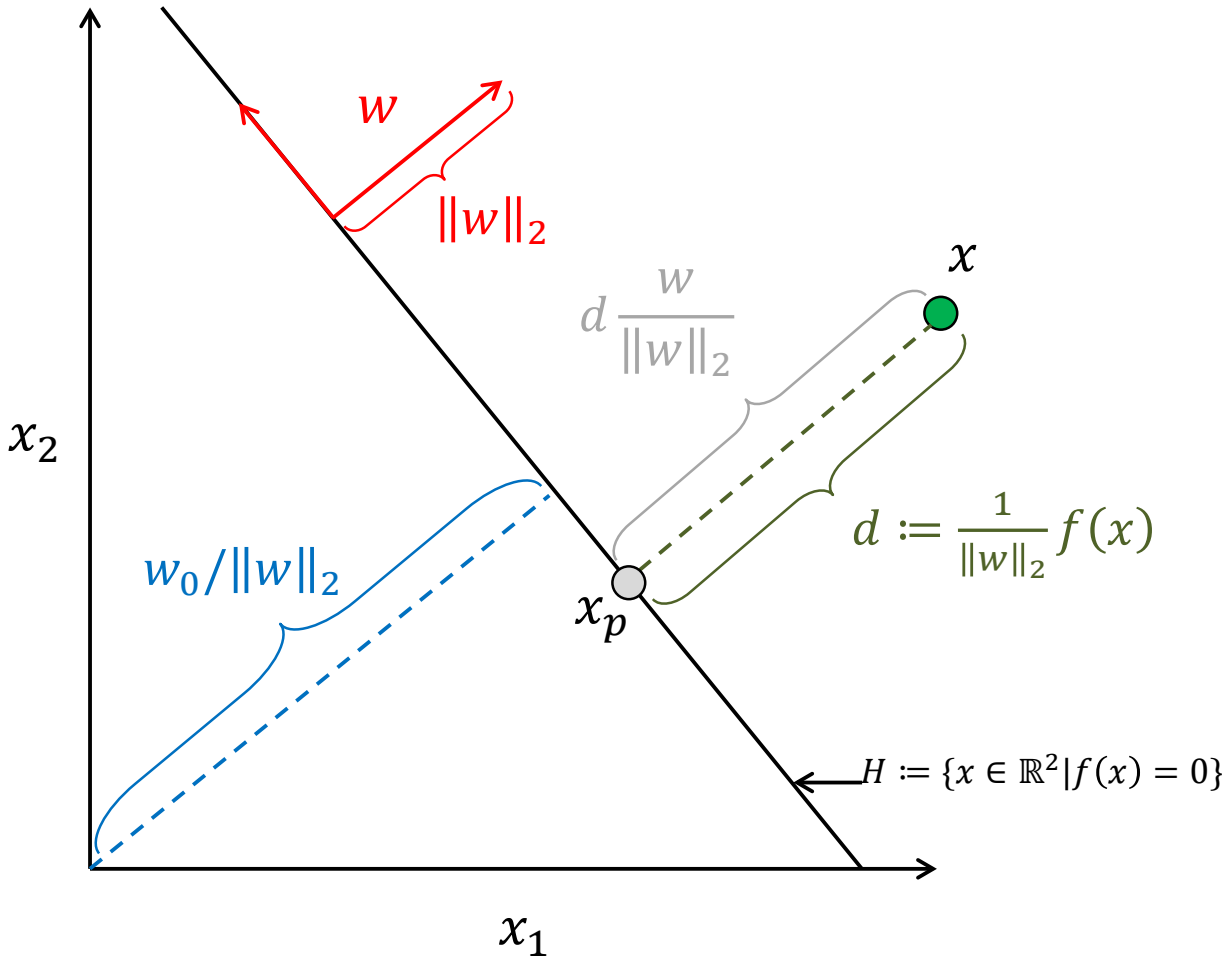
(3) Der minimale Euklidische Abstand d_0 zwischen dem Nullpunkt und einem Punkt auf H ist

$$d_0 = \frac{w_0}{||w||_2}. \tag{11}$$

Bemerkung

- w bestimmt die Orientierung der Hyperebene im Featurevektorenraum.
- w_0 bestimmt die Lage der Hyperebene im Featurevektorenraum.

Geometrie linearer Diskriminanzfunktionen



Geometrie linearer Diskriminanzfunktionen

Beweis von (1)

$x_a \in H_w$ und $x_b \in H_w$ seien zwei beliebige Punkte auf der Hyperebene. Dann gilt folgendes lineares Gleichungssystem:

$$w^T x_a + w_0 = 0 \tag{12}$$

$$w^T x_b + w_0 = 0. \tag{13}$$

Subtraktion von (13) von (12) ergibt

$$w^T x_a - w^T x_b = 0 \Leftrightarrow w^T (x_a - x_b) = 0. \tag{14}$$

Also ist der Parametervektor orthogonal zu dem Vektor $y := (x_a - x_b)$, welcher in Richtung der Hyperebene orientiert ist.

Beweis von (2)

Wir betrachten die Zerlegung eines Punktes $x \in \mathbb{R}^m$ in seine orthogonale Projektion auf eine Hyperebene $x_p \in \mathbb{R}^m$ und seinen Abstand von der Hyperebene $d \frac{w}{\|w\|_2}$

$$x = x_p + d \frac{w}{\|w\|_2}. \tag{15}$$

Diese Zerlegung ist möglich, weil w orthogonal zu jedem in Richtung der Hyperebene orientiertem Vektor ist und $\| \frac{w}{\|w\|_2} \|_2 = 1$ gilt.

Als nächstes betrachten wir die Transformation dieses so zerlegten x durch die lineare Diskriminanzfunktion:

$$f(x) = w^T x + w_0 = w^T \left(x_p + d\frac{w}{\|w\|_2} \right) + w_0 = w^T x_p + w_0 + d\frac{w^T w}{\|w\|_2}. \tag{16}$$

Dann gilt, weil $x_p \in H_w$ und somit $w^T x_p + w_0 = 0$, dass

$$f(x) = d\frac{w^T w}{\|w\|_2} = d\frac{\|w\|_2^2}{\|w\|_2} = d\|w\|_2. \tag{17}$$

Also folgt

$$d = \frac{1}{\|w\|_2} f(x). \tag{18}$$

Beweis von (3)

Für den minimalen Abstand des Nullpunktes $x_0 = (0, \ldots, 0)^T \in \mathbb{R}^m$ zu Punkten auf der Hyperebene gilt

$$d_0 = \frac{1}{\|w\|_2} f(x_0) = \frac{1}{\|w\|_2} (w^T x_0 + w_0) = \frac{1}{\|w\|_2} w^T \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} + \frac{w_0}{\|w\|_2} = \frac{w_0}{\|w\|_2}. \tag{19}$$

□

Definition (Hyperebenenmargin und Support Vektoren)

$\mathcal{D}$ sei ein Trainingsdatensatz, f sei eine multivariate reellwertige linear-affine Funktion und H sei die durch f induzierte Hyperebene. Weiterhin sei

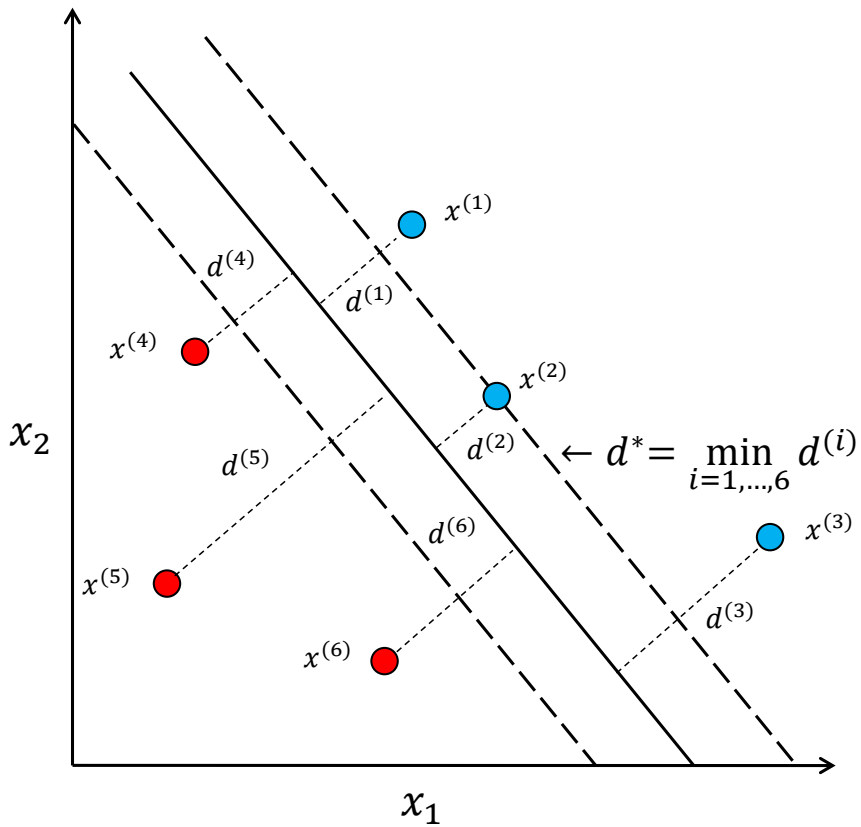
$$\left|d^{(i)}\right| := \left|\frac{1}{\|w\|_2} f\left(x^{(i)}\right)\right| = \frac{y^{(i)}}{\|w\|_2} f\left(x^{(i)}\right) = \frac{y^{(i)}(w^T x^{(i)} + w_0)}{\|w\|_2} \geq 0 \tag{20}$$

der absolute Wert des minimalen Euklidischen Abstands eines Featurevektors $x^{(i)}, i = 1, \ldots, n$ von H. Dann ist der *Margin* d^* von H hinsichtlich $\mathcal{D}$ definiert als das Minimum der absoluten minimalen Euklidischen Abstände von Featurevektoren zur Hyperebene,

$$d^* := \min_{i=1,\ldots,n} \left\{\left|d^{(i)}\right|\right\} = \min_{i=1,\ldots,n} \left\{\frac{y^{(i)}(w^T x^{(i)} + w_0)}{\|w\|_2}\right\}. \tag{21}$$

Ein Featurevektor $x^{(i)}$ wird *Support Vektor* genannt, wenn $|d^{(i)}| = d^*$, d.h., wenn $x^{(i)}$ auf dem Margin der Hyperebene liegt.

Hyperebenenmargin und Support Vektoren



Definition (Äquivalente Hyperebenen und kanonische Hyperebene)

f sei eine multivariate reellwertige linear-affine Funktion und

$$H := \{x \in \mathbb{R}^m | f(x) = 0\} \tag{22}$$

sei die durch f induzierte Hyperebene. Dann induzieren alle skalaren Vielfachen von f (und damit von w und w_0) die identische Hyperebene, denn aus $f(x) = 0$ folgt, dass $af(x) = 0$ für jedes $a \in \mathbb{R} \setminus \{0\}$. Die Hyperebenen

$$H_a := \{x \in \mathbb{R}^m | af(x) = 0, a \in \mathbb{R} \setminus \{0\}\} \tag{23}$$

heißen die zu H äquivalenten Hyperebenen. Zu einem Support Vektor x^* und einer Menge äquivalenter Hyperebenen (und somit einer Menge Parametervektoren und Biasparametern, welche die äquivalenten Hyperebenen induzieren) ist die *kanonische Hyperebene* definiert als die Hyperebene (und somit der spezifische Parametervektor w und Biasparameter w_0), für die gilt

$$|f(x^*)| = y^*(w^T x^* + w_0) = 1. \tag{24}$$

Aus der Definition der kanonischen Hyperebene folgt dann sofort, dass der Margin der kanonischen Hyperebene durch

$$d^* = \frac{1}{\|w\|_2}. \tag{25}$$

gegeben ist.

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Definition (Linear separierbarer Trainingsdatensatz)

Ein Trainingsdatensatz heißt *linear separierbarer Trainingsdatensatz*, wenn eine lineare Diskriminanzfunktion existiert, so dass alle Trainingsdatenpunkte korrekt klassifiziert werden können. Ein Trainingsdatensatz heißt nicht-linear separierbarer Trainingsdatensatz, wenn keine solche lineare Diskriminanzfunktion existiert.

Definition (Maximum Margin-Klassifikation)

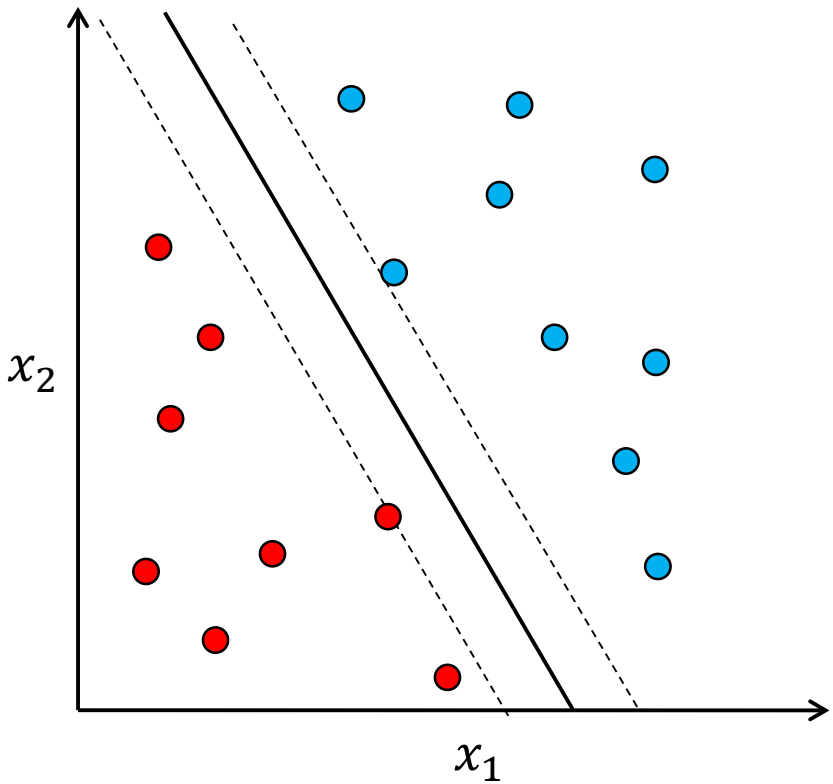
$\mathcal{D}$ sei ein linear separierbarer Trainingsdatensatz. Dann ist das Training einer Support Vektor Maschine für *Maximum Margin-Klassifikation* gegeben durch das Optimierungsproblem

$$\min_w \frac{1}{2}\|w\|_2^2 \text{ mit den Nebenbedingungen } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, \ldots, n. \tag{26}$$

Speziell entsprechen hierbei

- das Ziel
$$\min_w \frac{1}{2}\|w\|_2^2 \Leftrightarrow \max_w \frac{1}{\|w\|_2} \tag{27}$$
der Maximierung des Margins der von der SVM induzierten Hyperebene,
- die Nebenbedingungen
$$y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, \ldots, n \tag{28}$$
dem Ziel, dass alle Featurevektoren auf der korrekten Seite der Hyperebene liegen oder Support Vektoren sind.

Maximum Margin-Klassifikation



$$\min \ \frac{1}{2}\|w\|_2^2 \text{ u. d. Nbg. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1$$

Definition (Soft Margin-Klassifikation)

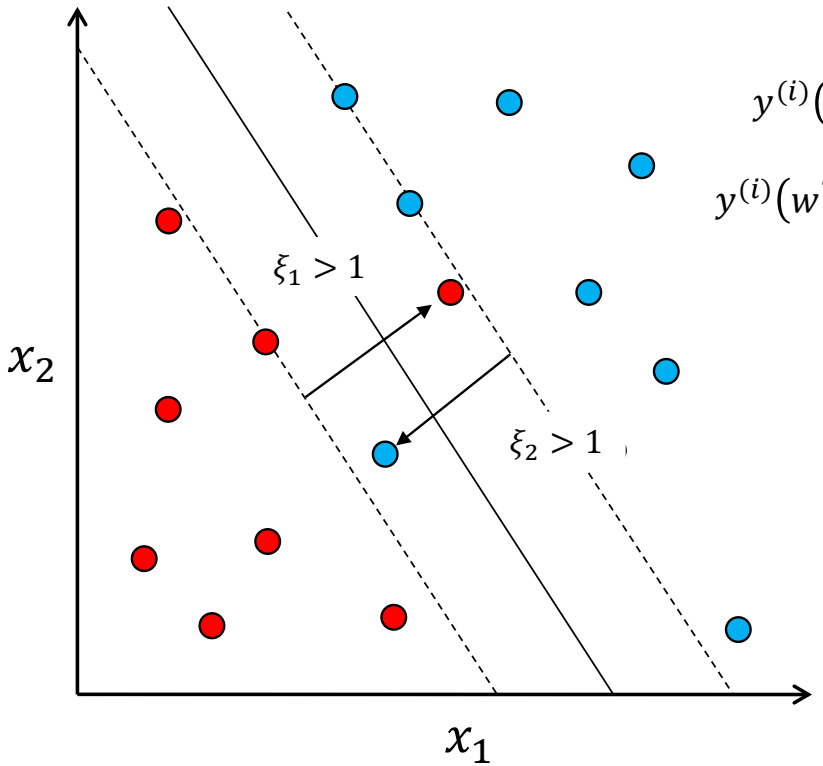
$\mathcal{D}$ sei ein nicht notwendigerweise linear separierbarer Trainingsdatensatz. Dann ist das Training einer Support Vektor Maschine für *Soft Margin-Klassifikation* gegeben durch das Optimierungsproblem

$$\min_{w,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n} \xi_i^k \text{ unter den Nebenbedingungen } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 - \xi_i, \xi \geq 0 \tag{29}$$

wobei $\xi := (\xi_1, \ldots, \xi_n)$ ein Vektor sogenannter *Schlupfvariablen (slack variables)* $\xi_i, i = 1, \ldots, n$ ist, der term $\sum_{i=1}^{n} \xi_i^k$ *Loss* genannt wird, $k \in \mathbb{N}$ eine Konstante ist, welche die genaue Form des Losses bestimmt (z.B. *hinge loss* für $k = 1$, *quadratic loss* für $k = 2$), und $C \in \mathbb{R}$ eine empirisch gewählte Konstante ist. Speziell entsprechen hierbei

- das Optimierungsziel dem Ziel, den Margin der durch die SVM induzierten Hyperebene zu maximieren und gleichzeitig den Loss zu minimieren, wobei die relative Gewichtung dieser beiden Ziele durch C gegeben ist,
- die Nebenbedingungen den Zielen
  (1) der korrekten Trainingsdatenpunktklassifikation und der Maximierung des Margins für $\xi_i = 0$„
  (2) der korrekten Trainingsdatenpunktklassifikation für $0 < \xi \leq 1$, und
  (3) inkorrekter Trainingsdatenpunktklassifikation für $\xi > 1$.

Soft Margin-Klassifikation



$$y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1$$
$$\downarrow$$
$$y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi_i$$

$$\min_{w,w_0,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i^k \text{ u. d. Nbg. } y^{(i)}\left(w^T x^{(i)} + w_0\right) \geq 1 - \xi_i, \xi_i \geq 0$$

Soft Margin SVM Training mit R Paket e1071

```
# Einlesen der Studienerfolgdaten und Fokus auf binäre Klassifikation
library(foreign)
D         = read.spss(file.path(getwd(), "8_Daten", "studienerfolg.sav"),
                      to.data.frame = T)
D         = D[D[,1] != "befriedigend",]

# SVM Training und Trainingsdatenprädiktion
library(e1071)                                                  # SVM Paket
acc = rep(NaN,4)                                                # Accuracyarray
for(i in 1:4){                                                  # Featuresetiteration
    x         = D[,2:(2+i-1)]                                   # Featurevektoren
    y         = D[,1]                                           # Labelvariable
    svm.train = svm(x,y, kernel = "linear")                     # SVM Training
    svm.pred  = predict(svm.train, x, kernel = "linear")        # Vorhersage der Trainingsdatenlabel
    acc[i]    = mean(svm.pred == y)                             # Trainingsdatenvorhersageaccuracy
}
print(acc)
```

```
> [1] 0.767 0.800 0.767 0.900
```

Soft Margin SVM Leave-One-Out Cross-Validation, $m = 2$

```
# Einlesen der Studienerfolgdaten und Fokus auf binäre Klassifikation
library(foreign)
D         = read.spss(file.path(getwd(), "8_Daten", "studienerfolg.sav"),
                      to.data.frame = T)
D         = D[D[,1] != "befriedigend",]
K         = nrow(D)
x         = D[,2:3]
y         = D[,1]

# K-fache Leave-One-Out Cross-Validation
library(e1071)
correct   = rep(NaN,K)
h         = rep(NaN,K)
for(k in 1:K){

    # Datensatzpartition
    x_train       = x[-k,]
    y_train       = y[-k]
    x_test        = x[k,]
    y_test        = y[k]

    # Trainingsdatensatz-basiertes Parameterlernen
    svm.train     = svm(x_train,y_train, kernel = "linear")

    # Testdatensatz-basierte Prädiktion
    svm.pred      = predict(svm.train, x_test, kernel = "linear")
    h[k]          = as.numeric(svm.pred)
    correct[k]    = svm.pred == y_test
}
cat("Accuracy: ", mean(correct))
```

Soft Margin SVM Leave-One-Out Cross-Validation, $m = 2$

	x_1	x_2	y	h(x)
1	54	44	-1	-1
2	60	20	-1	-1
3	67	36	-1	1
4	41	39	-1	-1
5	66	57	-1	1
6	51	28	-1	-1
7	51	46	-1	-1
8	37	46	-1	-1
9	57	54	-1	-1
10	47	12	-1	-1
11	50	67	-1	-1
12	42	63	-1	-1
13	60	64	-1	1
14	36	64	-1	-1
15	60	71	-1	1
31	71	41	1	1
32	65	28	1	-1
33	67	76	1	1
34	68	54	1	1
35	75	33	1	1
36	71	82	1	1
37	68	64	1	1
38	63	72	1	1
39	48	54	1	-1
40	53	86	1	-1
41	62	71	1	1
42	69	25	1	1
43	67	72	1	1
44	74	92	1	1
45	76	75	1	1

Prediction Accuracy = 0.77.

Leave-One-Out Cross-Validation, $m = 3$

	x_1	x_2	x_3	y	h(x)
1	54	44	31	-1	-1
2	60	20	33	-1	-1
3	67	36	26	-1	1
4	41	39	31	-1	-1
5	66	57	34	-1	1
6	51	28	42	-1	-1
7	51	46	34	-1	-1
8	37	46	36	-1	-1
9	57	54	28	-1	-1
10	47	12	34	-1	-1
11	50	67	27	-1	-1
12	42	63	26	-1	-1
13	60	64	29	-1	1
14	36	64	36	-1	-1
15	60	71	42	-1	1
31	71	41	37	1	1
32	65	28	33	1	-1
33	67	76	38	1	1
34	68	54	34	1	1
35	75	33	25	1	1
36	71	82	32	1	1
37	68	64	34	1	1
38	63	72	32	1	1
39	48	54	41	1	-1
40	53	86	27	1	-1
41	62	71	31	1	1
42	69	25	32	1	1
43	67	72	31	1	1
44	74	92	35	1	1
45	76	75	37	1	1

Prediction Accuracy = 0.77.

Soft Margin SVM Leave-One-Out Cross-Validation, $m = 4$

	x_1	x_2	x_3	x_4	y	h(x)
1	54	44	31	60	-1	-1
2	60	20	33	31	-1	-1
3	67	36	26	54	-1	1
4	41	39	31	26	-1	-1
5	66	57	34	56	-1	1
6	51	28	42	23	-1	-1
7	51	46	34	40	-1	-1
8	37	46	36	31	-1	-1
9	57	54	28	49	-1	-1
10	47	12	34	41	-1	-1
11	50	67	27	53	-1	-1
12	42	63	26	36	-1	-1
13	60	64	29	40	-1	1
14	36	64	36	21	-1	-1
15	60	71	42	40	-1	1
31	71	41	37	30	1	1
32	65	28	33	38	1	-1
33	67	76	38	28	1	1
34	68	54	34	53	1	-1
35	75	33	25	54	1	-1
36	71	82	32	49	1	1
37	68	64	34	33	1	1
38	63	72	32	36	1	1
39	48	54	41	34	1	-1
40	53	86	27	35	1	-1
41	62	71	31	53	1	-1
42	69	25	32	25	1	1
43	67	72	31	28	1	1
44	74	92	35	50	1	1
45	76	75	37	12	1	1

Prediction Accuracy = 0.60.

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Kernelisierung der Maximum Margin SVM

Kernelmethoden basieren auf dem dualen Problem des Maximum Margin SVM Trainingproblems.

Die zentrale Einsicht ist dabei, dass die Zielfunktion des dualen SVM Trainingproblems

$$q(\lambda) := \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{30}$$

lediglich von den Skalarprodukten der Featurevektoren

$${x^{(i)}}^T x^{(j)} \text{ für } i, j = 1, ..., n. \tag{31}$$

abhängt.

Die Projektion der Featurevektoren in einen "hochdimensionalen Featureraum", in welchem auf lineare Separabilität gehofft wird, benötigt also nur die Auswertung von Skalarprodukten.

Skalarprodukte in den Projektionsräumen werden *Kernel* genannt.

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Beginn Exkurs

Grundlagen der

Optimierung mit Nebenbedingungen

Definition (Optimierungsproblem mit Nebenbedingungen)

Ein *Optimierungsproblem mit Nebenbedingungen* hat die allgemeine Form

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I, \tag{32}$$

wobei $f : \mathbb{R}^n \to \mathbb{R}$ und $c_i : \mathbb{R}^n \to \mathbb{R}, i \in E \cup I$ glatte multivariate reellwertige Funktions und E, I endliche Indexmengen sind. f heißt *Zielfunktion*, die $c_i, i \in E$ heißen *Gleichungsnebenbedingungen* und die $c_i, i \in I$ heißen *Ungleichungsnebenbedingungen*. Die Menge

$$\mathcal{X} := \{x \in \mathbb{R}^n | c_i(x) = 0, i \in E \text{ und } c_i(x) \geq 0, i \in I\} \tag{33}$$

heißt *feasible set*.

Bemerkung

- Die notwendigen Bedingungen für Minimalstellen bei Optimierungsproblem ohne Nebenbedingungen sind für $n = 1$: $f'(x^*) = 0$ und für $n > 1$: $\nabla f(x^*) = 0_n$. Im Folgenden führen wir analoge notwendige Bedingungen erster Ordnung für Minimalstellen bei Optimierungsproblemen mit Nebenbedingungen ein.

Beispiel

Definition (Quadratisches Programm)

Ein *Quadratisches Programm* ist das konvexe Optimierungsproblem mit den Nebenbedingungen

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T P x + q^T x \text{ u.d.N. } Ax = b \text{ und } -Gx + h \geq 0, \tag{34}$$

wobei

- $P \in \mathbb{R}^{n \times n}$ eine positiv definite Matrix ist,
- $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$ sind und
- $G \in \mathbb{R}^{m \times n}$, und $h \in \mathbb{R}^m$ sind.

Bemerkungen

- Quadratische Programme sind Optimierungsprobleme mit Nebenbedingungen.
- Parameterlernen bei Support Vektor Maschinen führt auf ein Quadratisches Programm.
- Optimierungstoolboxen enhalten Funktionen zur Lösung Quadratischer Programme.
- In R bietet sich das Paket `quadprog` an.

Definition (Lagrange Funktion, Lagrange Multiplikatoren)

Es sei

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I, \tag{35}$$

ein Optimierungsproblem mit Nebenbedingungen. Dann ist die *Lagrange Funktion* dieses Problems definiert als

$$L : \mathbb{R}^n \times \mathbb{R}^{|E \cup I|} \to \mathbb{R}, (x, \lambda) \mapsto L(x, \lambda) := f(x) - \sum_{i \in E \cup I} \lambda_i c_i(x). \tag{36}$$

Hierbei wird $\lambda \in \mathbb{R}^{|E \cup I|}$ *Lagrange-Multiplikatoren Vektor* genannt und die einzelnen $\lambda_i \in \mathbb{R}$ mit $i \in E \cup I$ werden *Lagrange Multiplikatoren*genannt.

Bemerkung

- Die Lagrange Funktion und die Lagrange Multiplikatoren nehmen in den notwendigen Bedingungen der Optimierung mit Nebenbedingungen eine zentrale Rolle ein.

Definition (Notwendige Bedingungen erster Ordnung)

x^* sei eine lokale Lösung des Optimierungsproblems

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c_i(x) = 0, i \in E, c_i(x) \geq 0, i \in I. \tag{37}$$

Dann gibt es einen Lagrange-Multiplikatoren Vektor $\lambda^* \in \mathbb{R}^{|E \cup I|}$ mit den Komponenten $\lambda_i^*, i \in E \cup I$, so dass die folgenden Bedingungen an der Stelle $(x^*, \lambda^*) \in \mathbb{R}^{n+|E \cup I|}$ gelten

$$\begin{aligned}
\nabla_x L(x^*, \lambda^*) &= 0 \\
c_i(x^*) &= 0 \text{ für alle } i \in E \\
c_i(x^*) &\geq 0 \text{ für alle } i \in I \\
\lambda_i^* &\geq 0 \text{ für alle } i \in I \\
\lambda_i^* c_i(x^*) &= 0 \text{ für alle } i \in E \cup I
\end{aligned}$$

Bemerkungen

- Die Bedingungen werden auch *Karush-Kuhn-Tucker (KKT)* Bedingungen genannt.
- Für einen Beweis und Regularitätsbedingungen, siehe Nocedal and Wright (2006) Section 12.4.
- Die letzte Bedingung impliziert $\lambda_i^* > 0 \Rightarrow c_i(x^*) = 0$.

Definition (Duales Problem)

Es sei

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0, \tag{38}$$

ein Optimierungsproblem ohne Gleichungsnebenbedingungen, $c(x) := (c_1(x), c_2(x), \ldots, c_m(x))^T$ sei die multivariate vektorwertige Funktion der Ungleichungsnebenbedingungen und die zugehörige Lagrange Funktion und der Lagrange Multiplikatoren Vektoren $\lambda \in \mathbb{R}^m$ seien durch

$$L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, (x, \lambda) \mapsto L(x, \lambda) := f(x) - \lambda^T c(x). \tag{39}$$

gegeben. Dann ist die *duale Zielfunktion* (auch *duale Lagrange Funktion genannt*) definiert als

$$q : \mathbb{R}^m \to \mathbb{R}, \lambda \mapsto q(\lambda) := \min_x L(x, \lambda), \tag{40}$$

und das *duale Problem* ist definiert als

$$\max_{\lambda \in \mathbb{R}^m} q(\lambda) \text{ u.d.N. } \lambda \geq 0. \tag{41}$$

Bemerkung

- Duale Probleme sind manchmal einfacher zu lösen als die (primären) Ausgangsprobleme.
- Duale Probleme sind für das Parameterlernen von Support Vektor Maschinen zentral.

Theorem (Schwache Dualität)

Für jede Lösung $\bar{x}$ von

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0, \tag{42}$$

und jedes $\bar{\lambda} \geq 0$ gilt, dass

$$q(\bar{\lambda}) \leq f(\bar{x}). \tag{43}$$

Beweis

Mit den Definitionen von q, $\bar{\lambda} \geq 0$, und $c(\bar{x}) \geq 0$, gilt, dass

$$q(\bar{\lambda}) = \min_x f(x) - \bar{\lambda}^T c(x) \leq f(\bar{x}) - \bar{\lambda}^T c(\bar{x}) \leq f(\bar{x}). \tag{44}$$

□

Bemerkung

- Das Theorem besagt, dass der optimierte Wert des dualen Problems eine untere Grenze für den optimalen Wert der Zielfunktion des Ausgangsproblems ist.

Theorem (Starke Dualität)

Gegeben seien das Optimierungsproblem

$$\min_{x \in \mathbb{R}^n} f(x) \text{ u.d.N. } c(x) \geq 0 \tag{45}$$

und seine zugehörigen notwendigen Bedingungen erster Ordnung

$$\begin{aligned} \nabla f(\bar{x}) - \nabla c(\bar{x})\bar{\lambda} &= 0, \\ c(\bar{x}) &\geq 0, \\ \bar{\lambda} &\geq 0, \\ \bar{\lambda}_i c_i(\bar{x}) &= 0, i = 1, 2, \ldots, m, \end{aligned} \tag{46}$$

mit $\nabla c(x) = (\nabla c_1(x), \nabla c_2(x), \ldots, \nabla c_m(x)) \in \mathbb{R}^{n \times m}$. $\bar{x}$ sei eine Lösung des Ausgangsproblems und f sowie $-c_i, i = 1, 2, \ldots, m$ konvexe Funktionen auf $\mathbb{R}^n$, die in $\bar{x}$ differenzierbar sind. Dann ist jedes $\bar{\lambda}$, für das $(\bar{x}, \bar{\lambda})$ die notwendigen Bedingungen des Ausgangsproblem erfüllt, eine Lösung des dualen Problems

Bemerkungen

- Die optimalen Lagrange Multiplikatoren des Ausgangsproblems sind Lösungen des dualen Problems.
- SVM Training als Quadratisches Programm benötigt das Konzept der starken Dualität.

Beweis

Wir nehmen an, dass $(\bar{x}, \bar{\lambda})$ die notwendigen Bedingungen erster Ordnung für ein Minimum des Ausgangsproblem erfüllen und dass $L(\cdot, \bar{\lambda})$ konvex und differenzierbar ist. Dann gilt für jedes $x \in \mathbb{R}^n$, dass

$$L(x, \bar{\lambda}) \geq L(\bar{x}, \bar{\lambda}) + \nabla_x L(\bar{x}, \bar{\lambda})(x - \bar{x}) = L(\bar{x}, \bar{\lambda}), \tag{47}$$

weil $\nabla_x L(\bar{x}, \bar{\lambda}) = 0$. Also gilt für die duale Zielfunktion

$$q(\bar{\lambda}) = \inf_x L(x, \bar{\lambda}) = L(\bar{x}, \bar{\lambda}). \tag{48}$$

Mit der letzten der notwendigen Bedingungen erster Ordnung folgt weiterhin

$$q(\bar{\lambda}) = L(\bar{x}, \bar{\lambda}) = f(\bar{x}) - \bar{\lambda}^T c(\bar{x}) = f(\bar{x}) \tag{49}$$

Schließlich gilt mit dem Theorem zur Schwachen Dualität, dass $q(\lambda) \leq f(\bar{x})$ für alle $\lambda \geq 0$. Also folgt mit $q(\bar{\lambda}) = f(\bar{x})$, dass $\bar{\lambda}$ eine Lösung des dualen Problems ist. □

Ende Exkurs

Grundlagen der

Optimierung mit Nebenbedingungen

Theorem (SVM Training als quadratisches Programmierungsproblem I)

Das duale Problem des Maximum Margin SVM Trainingproblems

$$\min_{w} \frac{1}{2}\|w\|_2^2 \text{ mit den Nebenbedingungen } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, \ldots, n \tag{50}$$

ist gegeben als

$$\max_{\lambda \in \mathbb{R}^n} q(\lambda) := \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{51}$$

unter den Nebenbedingungen

$$\lambda \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i y^{(i)} = 0. \tag{52}$$

Basierend auf einer Lösung $\bar{\lambda}$ des dualen Problems sind alle $x^{(i)}$ mit $\bar{\lambda}_i > 0, i = 1, \ldots, n$ Support Vektoren und die Lösungen für den Parametervektor und den Biasparameter des primären Problems sind

$$\bar{w} = \sum_{i=1}^{n} \bar{\lambda_i} y^{(i)} x^{(i)} \text{ and } \bar{w}_0 = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \bar{w}^T x^{(i)} \right), \tag{53}$$

respective.

Theorem (SVM Training als quadratisches Programmierungsproblem II)

Weiterhin gilt, dass bei Definition von

$$y := \left(y^{(i)}\right)_{i=1,\ldots,n} \in \mathbb{R}^n \text{ and } K := \left(x^{(i)^T} x^{(j)}\right)_{i,j=1,\ldots,n} \in \mathbb{R}^{n\times n}, \tag{54}$$

as well as

$$P := yy^T K \in \mathbb{R}^{n\times n}, q := -1_n, G := -I_n, h := 0_n, A := y^T, \text{ and } b := 0 \tag{55}$$

das duale Problem des Maximum Marging SVM Trainingproblems als quadratisches Programmierproblem der Form

$$\min_{\lambda\in\mathbb{R}^n} \frac{1}{2}\lambda^T P\lambda + q^T\lambda \text{ mit den Nebenbedingungen } -G\lambda + h \geq 0_n \text{ and } A\lambda = b \tag{56}$$

geschrieben werden kann, und somit mit allen Standardalgorithmen der quadratischen Programmierung gelöst werden kann.

Bemerkungen

- Einerseits führt die QP Formulierung von Maximum Margin SVM Problem auf ein Standardproblem.
- Andererseits motiviert die QP Formulierung des Maximum Margin SVM Problems auch Kernelmethoden

Beweis

(1) Lagrangefunktion des primären Problems

Per definition ist die Lagrangefunktion des primären Problems

$$\min_{w,w_0} \frac{1}{2}\|w\|_2^2 \text{ mit den Nebenbedingungen } y^{(i)}(w^T x^{(i)} + w_0) \geq 1 \text{ für } i = 1, \ldots, n \tag{57}$$

gegeben durch

$$L(w, w_0, \lambda) := \frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)}(w^T x^{(i)} + w_0) - 1). \tag{58}$$

(2) Bestimmung der Zielfunktion des dualen Problems

Per definition ist die Zielfunktion des dualen Problems gegeben durch

$$q : \mathbb{R}^n \to \mathbb{R}, \lambda \mapsto q(\lambda) := \min_{w,w_0} L(w, w_0, \lambda). \tag{59}$$

Die analytische Bestimmung des Minimums der Lagrangefunktion L hinsichtlich w und w_0 entspricht der Bestimmung der partiellen Ableitungen von L hinsichtlich w und w_0, Nullsetzen, und lösen. Wir definieren

$$\bar{w} := \arg\min_{w \in \mathbb{R}^m} L(w, w_0, \lambda) \text{ and } \bar{w_0} := \arg\min_{w_0 \in \mathbb{R}} L(w, w_0, \lambda). \tag{60}$$

Beweis (fortgeführt)

Für die Minimierung von L hinsichtlich w ergibt sich

$$\begin{aligned}\nabla_w L(w, w_0, \lambda) &= \nabla_w \left(\frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)} (w^T x^{(i)} + w_0) - 1) \right) \\ &= w - \nabla_w \left(\sum_{i=1}^{n} \lambda_i y^{(i)} w^T x^{(i)} + \lambda_i y^{(i)} w_0 - \lambda_i \right) \qquad (61) \\ &= w - \sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)}\end{aligned}$$

und somit

$$\bar{w} = \sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)}. \qquad (62)$$

Beweis (fortgeführt)

In ähnlicher Weise ergibt sich für die Minimierung von L bezüglich w_0

$$\begin{aligned}\nabla_{w_0} L(w, w_0, \lambda) &= \nabla_{w_0} \left(\frac{1}{2} w^T w - \sum_{i=1}^{n} \lambda_i (y^{(i)} (w^T x^{(i)} + w_0) - 1) \right) \\ &= \nabla_{w_0} \left(\sum_{i=1}^{n} \lambda_i y^{(i)} w^T x^{(i)} + \lambda_i y^{(i)} w_0 - \lambda_i \right) \\ &= -\sum_{i=1}^{n} \lambda_i y^{(i)}.\end{aligned} \tag{63}$$

An der Minimalstelle von L hinsichtlich w_0 ergibt sich also

$$-\sum_{i=1}^{n} \lambda_i y^{(i)} = 0. \tag{64}$$

Man beachte, dass wir hier lediglich die Bedingung $-\sum_{i=1}^{n} \lambda_i y^{(i)} = 0$ an der Minimalstelle von L hinsichtlich w_0 erhalten, nicht aber die Minimalstelle $\bar{w}_0$ selbst.

SVM Training als quadratisches Programmierungsproblem

Beweis (fortgeführt)

Für die Zielfunktion des dualen Problems ergibt sich also

$$
\begin{aligned}
& q(\lambda) \\
& = \min_{w, w_0} L(w, w_0, \lambda) \\
& = L(\bar{w}, \bar{w}_0, \lambda) \\
& = \frac{1}{2} \bar{w}^T \bar{w} - \sum_{i=1}^{n} \lambda_i \left(y^{(i)} (\bar{w}^T x^{(i)} + \bar{w}_0) - 1 \right) \\
& = \frac{1}{2} \left(\sum_{i=1}^{n} \lambda_i y^{(i)} x^{(i)} \right)^T \left(\sum_{j=1}^{n} \lambda_i y^{(j)} x^{(j)} \right) - \sum_{i=1}^{n} \lambda_i \left(y^{(i)} \left(\left(\sum_{j=1}^{n} \lambda_j y^{(j)} x^{(j)} \right)^T x^{(i)} + \bar{w}_0 \right) - 1 \right. \\
& = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} x^{(i)^T} x^{(j)} - \sum_{i=1}^{n} \lambda_i y^{(i)} \left(\left(\sum_{j=1}^{n} \lambda_j y^{(j)} x^{(j)} \right)^T x^{(i)} + \bar{w}_0 \right) + \sum_{i=1}^{n} \lambda_i
\end{aligned}
$$

Beweis (fortgeführt)

und weiterhin

$$
\begin{aligned}
q(\lambda) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)} - \sum_{i=1}^{n}\lambda_i\lambda_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)} - \bar{w}_0\sum_{i=1}^{n}\lambda_i y^{(i)} + \sum_{i=1}^{n}\lambda_i \\
&= \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)} - \bar{w}_0\sum_{i=1}^{n}\lambda_i y^{(i)} \\
&= \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y^{(i)}y^{(j)}x^{(i)^T}x^{(j)}.
\end{aligned}
$$

Hierbei folgt die letzte Gleichung mit der Tatsache, dass an der Stelle $\bar{w}_0$ gilt, dass $\sum_{i=1}^{n}\lambda_i y^{(i)} = 0$.

Beweis (fortgeführt)

Wir haben also gezeigt, dass die Zielfunktion des dualen Problems des Maximum Margin SVM Trainingsproblems von der Form

$$q : \mathbb{R}^n \to \mathbb{R}, \lambda \to q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)}. \tag{65}$$

ist.

(3) Formulierung des dualen Problems

Das duale Problem zum Maximum Margin SVM Trainingsproblem ergibt sich also zu

$$\max_{\lambda \in \mathbb{R}} q(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{66}$$

unter den Nebenbedingunge

$$\lambda_i \geq 0, i = 1, \ldots, n \text{ and } \sum_{i=1}^{n} \lambda_i y^{(i)} = 0, \tag{67}$$

wobei die letzte Nebenbedingung das Minimum der Lagrangefunktion hinsichtlich w_0 sicherstellt.

Beweis (fortgeführt)

Lösen des dualen Problems mithilfe eines Standardalgorithmus ergibt einen Vektor optimaler Lagrangemultiplikatoren

$$\bar{\lambda} = \underset{\lambda \in \mathbb{R}^n}{\arg\max}\, q(\lambda) = \underset{\lambda \in \mathbb{R}^n}{\arg\max}\, L(\bar{w}, \bar{w}_0, \lambda). \tag{68}$$

Basierend auf der Minimierung von L hinsichtlich von w ergibt sich also

$$\bar{w} = \sum_{i=1}^{n} \bar{\lambda}_i y^{(i)} x^{(i)}. \tag{69}$$

Schließlich ergibt sich für den optimalen Biasparameter $\bar{w}_0$ zunächst mit den KKT Bedingungen, dass für alle $\hat{\lambda}_i > 0, i = 1, \ldots, n$ gilt, dass

$$\begin{aligned} y^{(i)}(\bar{w}^T x^{(i)} + w_0) - 1 &= 0 \\ \Leftrightarrow y^{(i)}(\bar{w}^T x^{(i)} + w_0) &= 1 \\ \Leftrightarrow y^{(i)} y^{(i)}(\bar{w}^T x^{(i)} + w_0) &= y^{(i)} \\ \Leftrightarrow \bar{w}^T x^{(i)} + w_0 &= y^{(i)}. \end{aligned} \tag{70}$$

SVM Training als quadratisches Programmierungsproblem

Beweis (fortgeführt)

Dies impliziert, erstens, dass alle $x^{(i)}$ mit $\bar{\lambda}_i > 0$ Support Vektoren sind, weil ihre Distanz zur optimalen Hyperebene gleich 1 ist, und zweitens, dass

$$\sum_{i=1}^{n} \bar{w}^T x^{(i)} + n w_0 = \sum_{i=1}^{n} y^{(i)} \Leftrightarrow w_0 = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - \bar{w}^T x^{(i)} \right). \tag{71}$$

(4) Standardform eines QP Problems

Die Äquivalenzen

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \Leftrightarrow \lambda^T y y^T K \lambda \Leftrightarrow \lambda^T P \lambda$$

$$\sum_{i=1}^{n} \lambda_i \Leftrightarrow 1_n^T \lambda \Leftrightarrow q^T \lambda \tag{72}$$

$$\lambda \geq 0 \Leftrightarrow I_n \lambda + 0_n \geq 0_n \Leftrightarrow -G\lambda + h \leq 0_n$$

$$\sum_{i=1}^{n} \lambda_i y^{(i)} = 0 \Leftrightarrow y^T \lambda = 0 \Leftrightarrow A\lambda = b$$

ergeben sich direkt mit den Regeln der Matrixmultiplikation.

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Kernelisierung der Maximum Margin SVM

Kernelmethoden basieren auf dem dualen Problem des Maximum Margin SVM Trainingproblems.

Die zentrale Einsicht ist dabei, dass die Zielfunktion des dualen SVM Trainingproblems

$$q(\lambda) := \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)} \tag{73}$$

lediglich von den Skalarprodukten der Featurevektoren

$${x^{(i)}}^T x^{(j)} \text{ für } i, j = 1, ..., n. \tag{74}$$

abhängt.

Die Projektion der Featurevektoren in einen "hochdimensionalen Featureraum", in welchem auf lineare Separabilität gehofft wird, benötigt also nur die Auswertung von Skalarprodukten.

Skalarprodukte in den Projektionsräumen werden *Kernel* genannt.

Geometrie linearer Diskriminanzfunktionen

Support Vektor Maschinen Training

SVM Training als quadratisches Optimierungsproblem

Kernelisierung der Maximum Margin SVM

Selbstkontrollfragen

Selbstkontrollfragen

1. Wie unterscheiden sich binäre Klassifikationsdatensätze für Support Vektor Maschinen von binären Klassifikationsdatensätzen für Lineare Diskriminanzanalyse und Logistische Regression?
2. Definieren Sie den Begriff der linearen Diskriminanzfunktion.
3. Definieren Sie in Bezug zum Begriff der linearen Diskriminanzfunktion die Begriffe der Entscheidungsgrenze, der Hyperebene und der Entscheidungsregion.
4. Geben Sie das Theorem zur Geometrie linearer Diskriminanzfunktionen wieder.
5. Erläutern Sie die Bedeutung des Theorems zur Geometrie linearer Diskriminanzfunktionen bei der Bestimmung von Hyperebenen.
6. Definieren Sie den Begriff des Hyperebenenmargins.
7. Definieren Sie den Begriff des Support Vektors.
8. Definieren Sie den Begriff der kanonischen Hyperebene.
9. Definieren Sie den Begriff des linear separierbaren Trainingsdatensatzes.
10. Definieren und erläutern Sie das Optimierungsproblem zur Maximum Margin Klassifikation bei SVMs.
11. Definieren und erläutern Sie das Optimierungsproblem zur Soft Margin Klassifikation bei SVMs.
12. Lesen Sie den Datensatz studienerfolg.sav mit R ein und bestimmen Sie den Trainingsdatenprädiktionsfehler nach Trainieren einer Support Vektor Maschine mithilfe des R Pakets e1071 zur prädiktiven Modellierung des Studienerfolgs (gut, ungenügend) basierend auf (1) den Intelligenztestdaten, (2) den Intelligenz- und Mathematiktestdaten und (3) den Intelligenztest-, Mathematiktest-, und Gewissenhaftigkeitsdaten.

References

Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. 2018. "Machine Learning Approaches for Clinical Psychology and Psychiatry." *Annual Review of Clinical Psychology* 14 (1): 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037.

Nocedal, Jorge, and Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research. New York: Springer.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(9) Neuronale Netze

Vorbemerkungen

Funktionale Architektur

Lernen

Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Vorbemerkungen

Funktionale Architektur

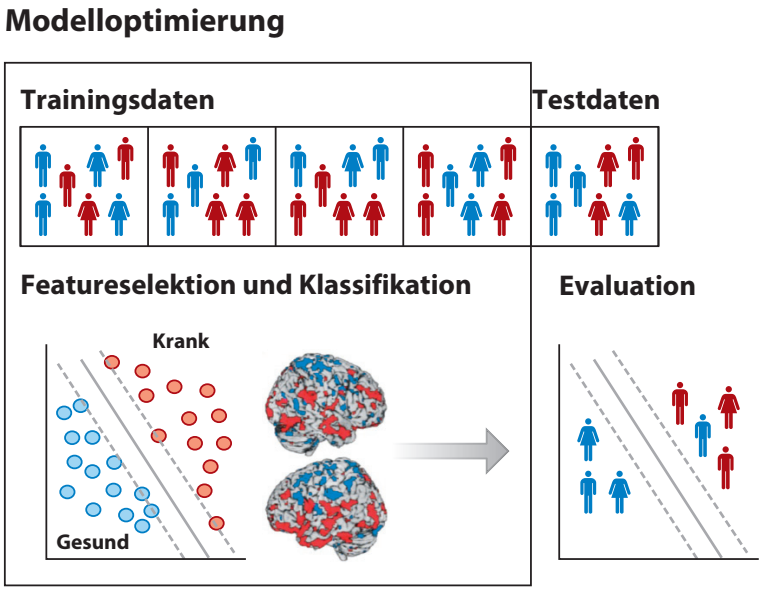
Lernen

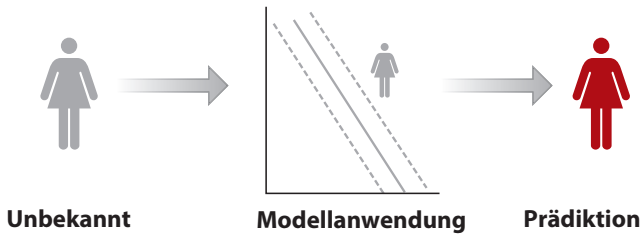
Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Struktur der Prädiktiven Modellierung





nach Dwyer, Falkai, and Koutsouleris (2018)

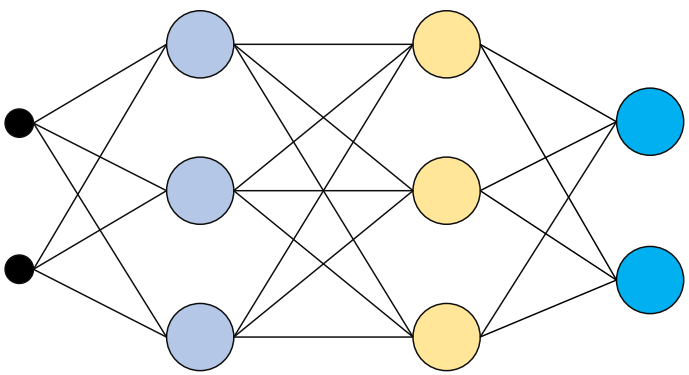
Rhetorik der Prädiktiven Modellierung

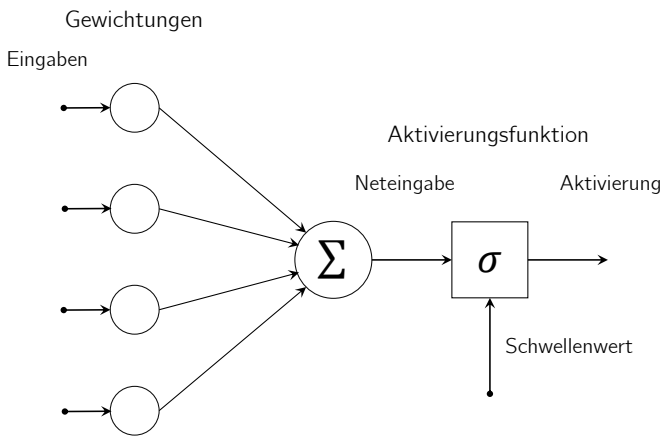
Daten	Trainingsdaten und Testdaten
Statistisches Modell	Modell, Machine Learning Algorithmus
Schätzen von Parametern	Trainieren des Modells, Lernen von Parametern, Supervised Learning

Neuronale Netze (Neural Networks)

- AKA *Künstliche Neuronale Netze (Artificial Neural Networks)*.
- Keine Modelle für biologische neuronale Netze.
- Mathematische Modelle zur Approximation multivariater vektorwertiger Funktionen.

Typische Visualisierungen





Zur Geschichte neuronaler Netze

Anfänge

- McCulloch and Pitts (1943) | Analyse der mit biologischen Neuronen möglichen logischen Operationen.
- Rosenblatt (1958) | Implementation eines Mustererkennungsalgorithmus in einem frühen Computer.
- Minsky and Papert (1969) | Mathematische Analyse der logischen Stärken und Schwächen eines Perzeptrons.

$\Rightarrow$ Erster Winter Neuronaler Netze

Erste Renaissance

- Hopfield (1982) | Mehrschichtige neuronale Netze beleben das Interesse an neuronalen Netzen erneut.
- Rumelhart, Hinton, and Williams (1986) | Popularisierung des Backpropagation Algorithmus.
- Hauptinteresse in den 1990er und 2000er Jahren im Machine Learning gilt aber SVMs und Bayesian Inference.

$\Rightarrow$ Zweiter Winter Neuronaler Netze

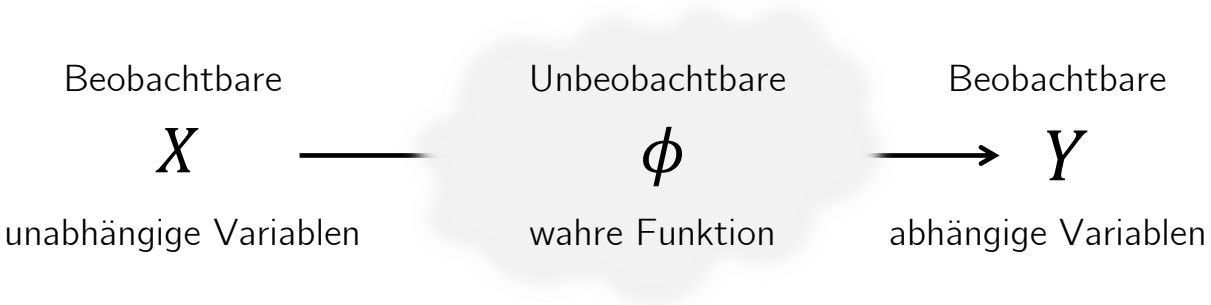
Zweite Renaissance

- 2009 - 2012 | Schmidhuber (2015) gewinnen Klassifikationswettbewerbe mit neuronalen Netzen.
- LeCun, Bengio, and Hinton (2015) | Neuronale Netze unter dem Label “Deep Learning” wieder sehr in Mode.
- 2015 - 2022 | Viele Menschen verwechseln die Begriffe “Künstliche Intelligenz” und “Neuronales Netz”.
- Ostwald and Usée (2021) | Beweis der Validität des Backpropagation Algorithmus in Matrixform.

Neuronale Netze und Prädiktive Modellierung

Explanatorische Modellierung ⇔ Wissenschaft

Bestimmung von $\hat{\phi} \coloneqq \operatorname{argmin} \|\hat{\phi} - \phi\|$



Bestimmung von $f \coloneqq \operatorname{argmin}_{\tilde{f} \in F} \|Y - \tilde{f}(X)\|$

Prädiktive Modellierung ⇔ Anwendung

⇒ Neuronale Netze zur Approximation multivariater vektorwertiger Funktionen im prädiktiven Sinn.

Universelle Approximationstheoreme

Topologische Aussagen über die Dichten von Funktionenräumen (cf. Friedman (1970)).

Neuronale Netze können eine Vielzahl von Funktionen sehr genau approximieren, wenn

- die Anzahl der Neurone gegen Unendlich geht (*arbitrary width case*) bzw.
- die Anzahl der Neuronenschichten gegen Unendlich geht (*arbitrary depth case*).

Arbitrary width case $\Rightarrow$ Cybenko (1989), Hornik (1991), Leshno et al. (1993), Pinkus (1999)

Arbitrary depth case $\Rightarrow$ Lu et al. (2017), Hanin and Sellke (2018), Kidger and Lyons (2020)

Universelle Approximationstheoreme sind Existenzaussagen, keine Konstruktionsaussagen.

$\Rightarrow$ Parameter neuronaler Netze müssen durch Gradientenverfahren gelernt werden.

Universelle Approximationstheoreme

Beispiel

Theorem (Universelles Approximationstheorem nach Kidger (2020))

$\mathcal{X}$ sei eine kompakte Teilmenge von $\mathbb{R}^m$, $\sigma : \mathbb{R} \to \mathbb{R}$ sei eine nicht-affine stetige und zumindest in einem Punkt stetig differenzierbare Funktion mit einer von Null verschiedenen Ableitung in diesem Punkt. F sei die Menge der neuronalen Netze f mit Inputdimension m, Outputdimension n_k und einer beliebigen Anzahl verdeckter Schichten mit jeweils $m + n_k + 2$ Neuronen und Aktivierungsfunktion σ, sowie der Identitätsabbildung als Aktivierungsfunktion der Outputschicht. Dann existiert zu jeder stetigen multivariaten vektorwertigen Funktion

$$g : \mathcal{X} \to \mathbb{R}^{n_k}, x \mapsto g(x) \tag{1}$$

ein neuronales Netz $f \in F$, so dass für ein beliebig kleines $\epsilon > 0$ gilt, dass

$$\sup_{x \in \mathcal{X}} \|f(x) - g(x)\| < \epsilon. \tag{2}$$

Bemerkungen

- Das Supremum $\sup$ kann intuitiv als Maximum verstanden werden.
- $\|\cdot\|$ bezeichnet eine *Metrik* (Abstandsfunktion) auf $\mathbb{R}^{n_k}$.
- Für jedes $x \in \mathcal{X}$ wird der Abstand zwischen dem Wert von g und dem Wert von f also beliebig klein.
- Man sagt dazu auch, dass F im Raum der stetigen multivariaten vektorwertigen Funktionen *dicht* ist.
- Man kann das Theorem sicherlich noch präziser formulieren und sollte es beweisen.
- Wir verzichten hier darauf und führen das Theorem nur als "intuitives Beispiel" auf.

Vorbemerkungen

Funktionale Architektur

Lernen

Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Definition (Potentialfunktionen)

$W \in \mathbb{R}^{m \times (n+1)}$ sei eine Matrix, die wir *Wichtungsmatrix* nennen und $a \in \mathbb{R}^n$ sei ein Vektor, den wir *Aktivierungsvektor* nennen. Dann nennen wir eine Funktion der Form

$$\Phi : \mathbb{R}^{m \times (n+1)} \times \mathbb{R}^n \to \mathbb{R}^m, (W, a) \mapsto \Phi(W, a) := W \cdot \begin{pmatrix} a \\ 1 \end{pmatrix} \tag{3}$$

eine *bivariate Potentialfunktion*. Für ein festes $a \in \mathbb{R}^n$ nennen wir eine Funktion der Form

$$\Phi_a : \mathbb{R}^{m \times (n+1)} \to \mathbb{R}^m, \ W \mapsto \Phi_a(W) := \Phi(W, a) \tag{4}$$

eine *Wichtungsmatrix-variate Potentialfunktion*. Weiterhin nennen wir für eine feste Matrix $W \in \mathbb{R}^{m \times (n+1)}$ eine Funktion der Form

$$\Phi_W : \mathbb{R}^n \to \mathbb{R}^m, a \mapsto \Phi_W(a) := \Phi(W, a) \tag{5}$$

eine *Potentialfunktion*. Schließlich nennen wir $z := \Phi_W(a)$ einen *Potentialvektor*.

Definition (Aktivierungsfunktion)

Wir nennen eine multivariate vektorwertige Funktion der Form

$$\Sigma : \mathbb{R}^n \to \mathbb{R}^n, z \mapsto \Sigma(z) := (\sigma(z_1), \ldots, \sigma(z_n))^T, \tag{6}$$

mit

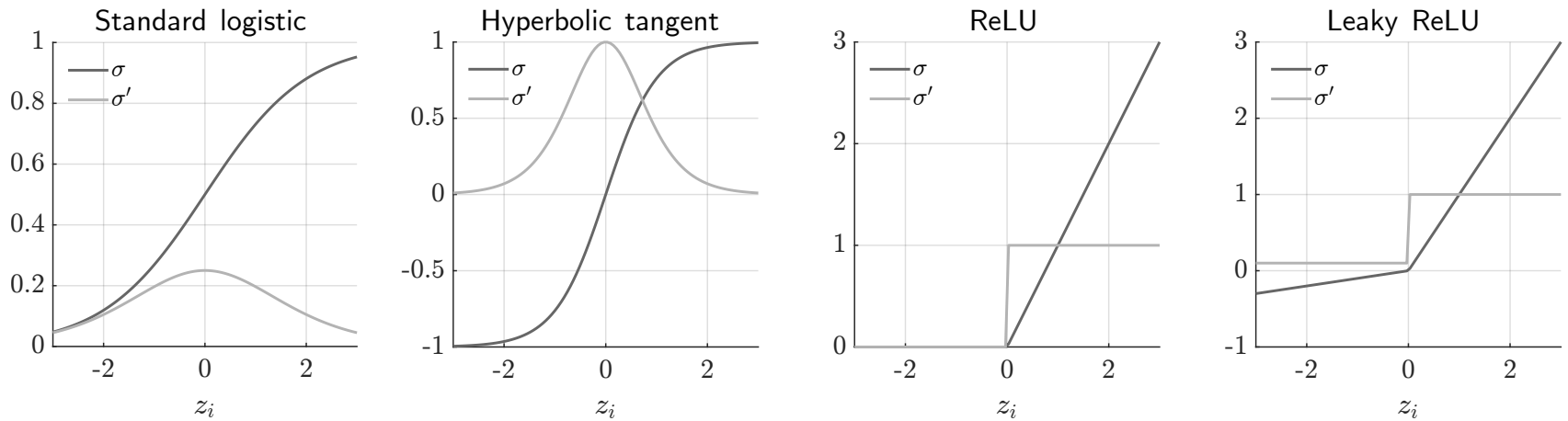
$$\sigma : \mathbb{R} \to \mathbb{R}, z_i \mapsto \sigma(z_i) =: a_i \text{ für alle } i = 1, \ldots, n, \tag{7}$$

eine *komponentenweise Aktivierungsfunktion* und die univariate reellwertige Funktion σ eine *Aktivierungsfunktion*.

Typische Aktivierungsfunktionen und ihre Ableitungen

Name	Definition	Ableitung
Standard logistic	$\sigma(z_i) := \frac{1}{1+\exp(-z_i)}$	$\sigma'(z_i) = \frac{\exp(z_i)}{(1+\exp(z_i))^2}$
Hyperbolic tangent	$\sigma(z_i) := \tanh(z_i)$	$\sigma'(z_i) = 1 - \tanh^2(z_i)$
ReLU	$\sigma(z_i) := \max(0, z_i)$	$\sigma'(z_i) = \begin{cases} 0, & z_i < 0 \\ \emptyset, & z_i = 0 \\ 1, & z_i > 0 \end{cases}$
Leaky ReLU	$\sigma(z_i) := \begin{cases} 0.1z_i, & z_i \leq 0 \\ z_i, & z_i > 0 \end{cases}$	$\sigma'(z_i) = \begin{cases} 0.01, & z_i \leq 0 \\ 1, & z_i > 0 \end{cases}$

Typische Aktivierungsfunktionen und ihre Ableitungen



Definition (k-schichtiges neuronales Netz)

Eine multivariate vektorwertige Funktion

$$f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_k}, x \mapsto f(x) =: y \tag{8}$$

heißt *k-schichtiges neuronales Netz*, wenn f von der Form

$$f : \mathbb{R}^{n_0} \xrightarrow{\Phi^1_{W^1}} \mathbb{R}^{n_1} \xrightarrow{\Sigma^1} \mathbb{R}^{n_1} \xrightarrow{\Phi^2_{W^2}} \mathbb{R}^{n_2} \xrightarrow{\Sigma^2} \mathbb{R}^{n_2} \xrightarrow{\Phi^3_{W^3}} \dots \xrightarrow{\Phi^{k-1}_{W^{k-1}}} \mathbb{R}^{n_{k-1}} \xrightarrow{\Sigma^{k-1}} \mathbb{R}^{n_{k-1}} \xrightarrow{\Phi^k_{W^k}} \mathbb{R}^{n_k} \xrightarrow{\Sigma^k} \mathbb{R}^{n_k}, \tag{9}$$

ist, wobei für $l = 1, \dots, k$

$$\Phi^l_{W^l} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}, a^{l-1} \mapsto \Phi^l_{W^l}(a^{l-1}) := W^l \cdot \begin{pmatrix} a^{l-1} \\ 1 \end{pmatrix} =: z^l \tag{10}$$

Potentialfunktionen und

$$\Sigma^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_l}, z^l \to \Sigma^l(z^l) =: a^l \tag{11}$$

komponentenweise Aktvierungsfunktionen sind. Für ein $x \in \mathbb{R}^{n_0}$ nimmt ein k-schichtiges neuronales Netz den Wert

$$f(x) := \Sigma^k(\Phi^k_{W^k}(\Sigma^{k-1}(\Phi^{k-1}_{W^{k-1}}(\Sigma^{k-2}(\dots(\Sigma^1(\Phi^1_{W^1}(x))\dots)))))) \in \mathbb{R}^{n_k}. \tag{12}$$

an.

Funktionale Architektur

Bemerkungen

- Die Vektoren $a^l = (a^l_1, \ldots, a^l_{n_l})^T \in \mathbb{R}^{n_l}, l = 0, 1, \ldots, k$ heißen *Aktivierungsvektoren der $l$ten Schicht*.
- Die Komponenten $a^l_i \in \mathbb{R}, i = 1, \ldots, n_l, l = 0, 1, \ldots, k$ heißen *Aktivierungen der $l$ten Schicht*.
- Die Schicht mit Index $l = 0$ und Dimension n_0 heißt *Inputschicht*.
- Der Aktivierungsvektor mit Index $l = 0$ heißt *Input* und wird mit $x := a^0$ bezeichnet.
- Die Schicht mit Index $l = k$ und Dimension n_k heißt *Outputschicht*
- Der Aktivierungsvektor mit Index $l = k$ heißt Output und wird mit $y := a^k$ bezeichnet.
- Die Schichten mit den Indizes $l = 1, \ldots, k-1$ heißen *verdeckte Schichten (hidden layers)*.
- $w^l_{ij} \in \mathbb{R}$ sei der ijte Eintrag der $l$ten Wichtungsmatrix, d.h.

$$W^l = (w^l_{ij})_{1 \leq i \leq n_l, 1 \leq j \leq n_{l-1}+1} \in \mathbb{R}^{n_l \times (n_{l-1}+1)} \text{ für } l = 1, \ldots, k. \tag{13}$$

- w^l_{ij} heißt *(synaptisches) Gewicht* der Verbindung von Neuron j in Schicht $l-1$ and Neuron i in Schicht l.
- Für $i = 1, \ldots, n_l$ heißt $w_{i,n_{l-1}+1}$ *Bias* von Neuron i in Schicht l.
- Die letzte Spalte von W^l enkodiert also die Biases für die Neuronen in Schicht l.

Bemerkungen

Auf der Ebene einzelner Neurone ergibt sich damit folgende Nomenklatur:

- Das *Potential* von Neuron i in Schicht l für $i = 1, \ldots, n_l$ und $l = 1, \ldots, k$ ist gegeben durch

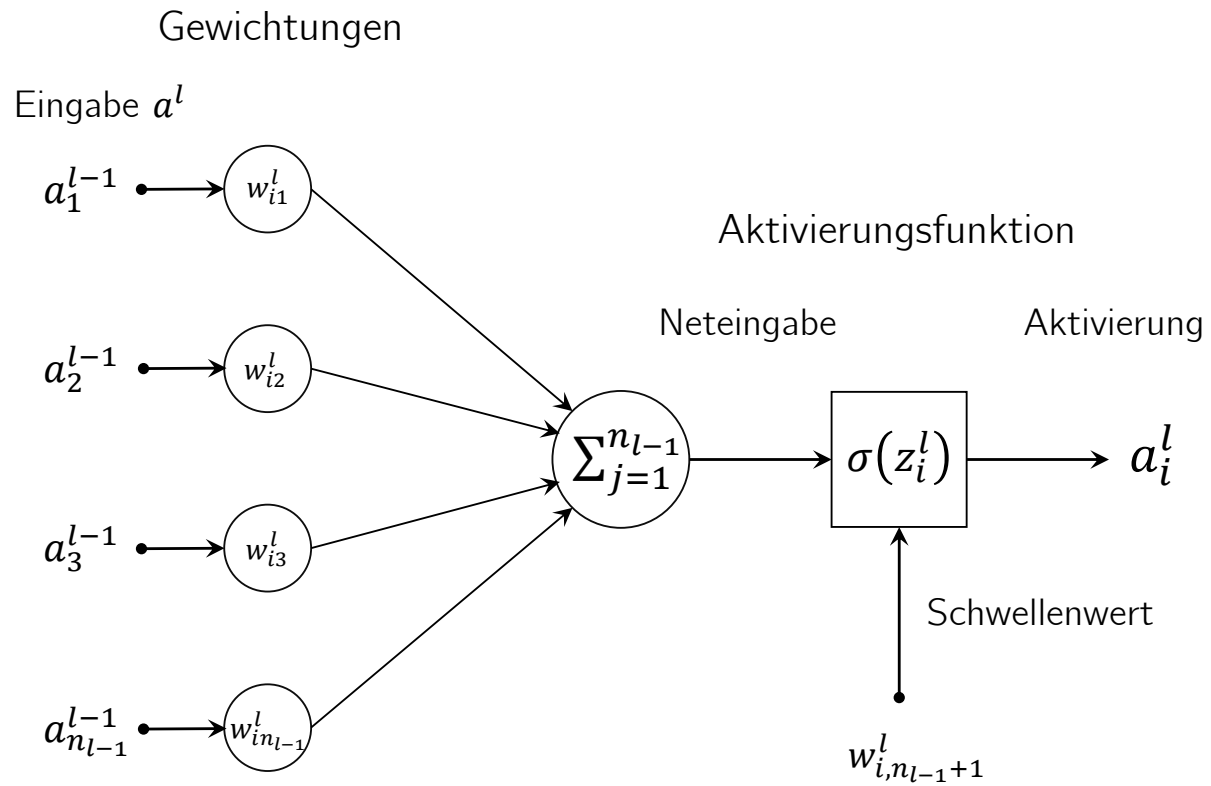
$$z_i^l = \sum_{j=1}^{n_{l-1}} w_{ij}^l a_j^{l-1} + w_{i,n_{l-1}+1} \in \mathbb{R}. \tag{14}$$

- Die *Aktivierung* von Neuron i in Schicht l für $i = 1, \ldots, n_l$ und $l = 1, \ldots, k$ ist gegeben durch

$$a_i^l = \sigma \left(\sum_{j=1}^{n_{l-1}} w_{ij}^l a_j^{l-1} + w_{i,n_{l-1}+1} \right) \in \mathbb{R}, \tag{15}$$

- Die Aktivierung a_i^l kann als die mittlere Feuerungsrate des $i$ten Neuron in der $l$ten Schicht verstanden werden.

Funktionale Architektur



Beispiel

$k = 3, n_0 = 2, n_1 = 3,\ n_2 = 3, n_3 = 2$

$$\begin{pmatrix} a^0 \\ 1 \end{pmatrix} = \begin{pmatrix} a_1^0 \\ a_2^0 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} a^1 \\ 1 \end{pmatrix} = \begin{pmatrix} a_1^1 \\ a_2^1 \\ a_3^1 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} a^2 \\ 1 \end{pmatrix} = \begin{pmatrix} a_1^2 \\ a_2^2 \\ a_3^2 \\ 1 \end{pmatrix} \qquad a^3 = \begin{pmatrix} a_1^3 \\ a_2^3 \end{pmatrix}$$

$$W^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 \\ w_{31}^1 & w_{32}^1 & w_{33}^1 \end{pmatrix} \quad W^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & w_{13}^2 & w_{14}^2 \\ w_{21}^2 & w_{22}^2 & w_{23}^2 & w_{24}^2 \\ w_{31}^2 & w_{32}^2 & w_{33}^2 & w_{34}^2 \end{pmatrix} \quad W^3 = \begin{pmatrix} w_{11}^3 & w_{12}^3 & w_{13}^3 & w_{14}^3 \\ w_{21}^3 & w_{22}^3 & w_{23}^3 & w_{24}^3 \end{pmatrix}$$

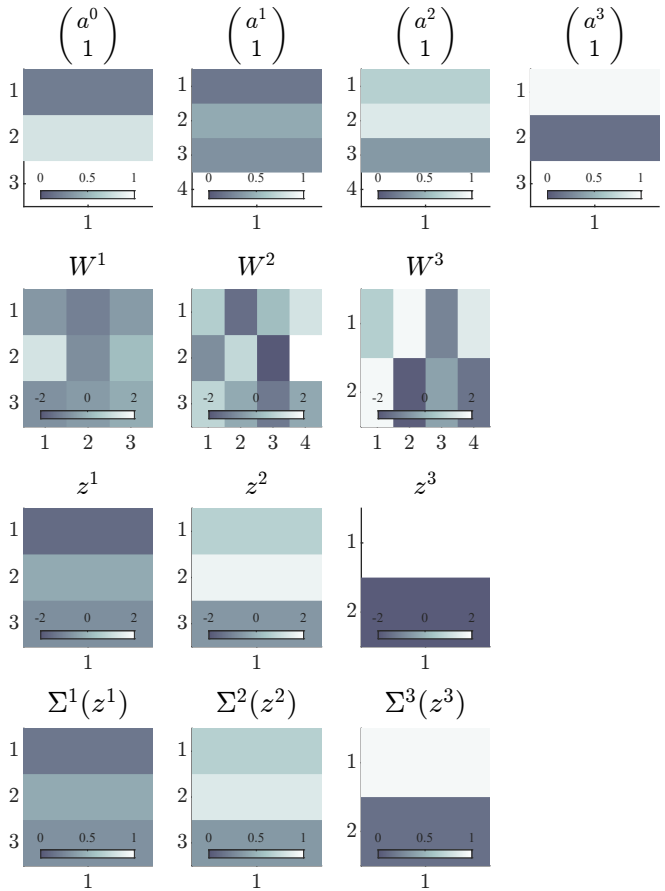
$$z^1 = \begin{pmatrix} z_1^1 \\ z_2^1 \\ z_3^1 \end{pmatrix} \qquad z^2 = \begin{pmatrix} z_1^2 \\ z_2^2 \\ z_3^2 \end{pmatrix} \qquad z^3 = \begin{pmatrix} z_1^3 \\ z_2^3 \end{pmatrix}$$

$$\Sigma^1(z^1) = \begin{pmatrix} \sigma(z_1^1) \\ \sigma(z_2^1) \\ \sigma(z_3^1) \end{pmatrix} \qquad \Sigma^2(z^2) = \begin{pmatrix} \sigma(z_1^2) \\ \sigma(z_2^2) \\ \sigma(z_3^2) \end{pmatrix} \qquad \Sigma^3(z^3) = \begin{pmatrix} \sigma(z_1^3) \\ \sigma(z_2^3) \end{pmatrix}$$

Es gilt $x = a^0$ und $a^3 = y$.

Beispiel

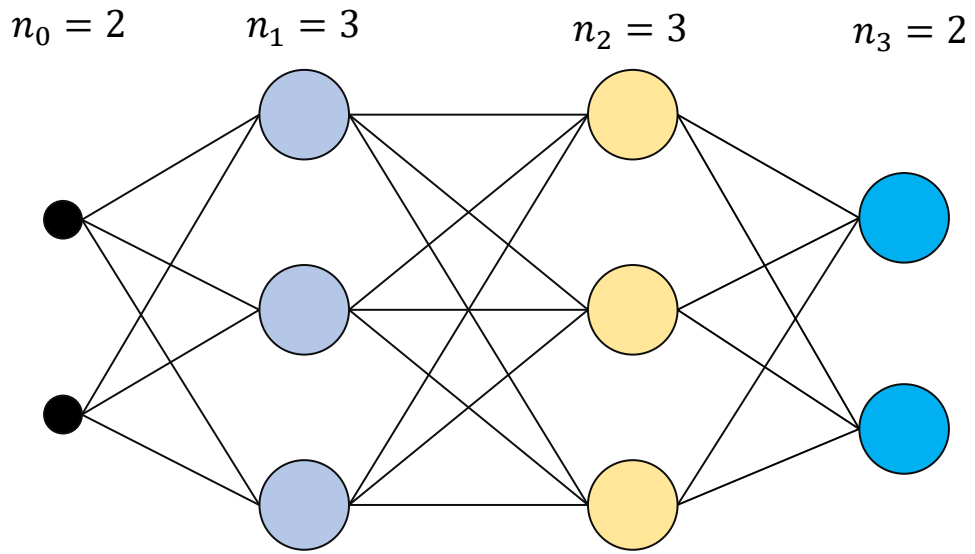
$k = 3, n_0 = 2, n_1 = 3,\ n_2 = 3, n_3 = 2$



Es gilt $x = a^0$ und $a^3 = y$.

Beispiel

$k = 3, n_0 = 2, n_1 = 3, n_2 = 3, n_3 = 2$



- Die Biases sind hier nicht visualiert.

Vorbemerkungen

Funktionale Architektur

Lernen

Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Überblick

Anhand eines Trainingsdatensatzes werden die Parameter eines neuronalen Netzes wie folgt gelernt:

- Zunächst wird eine Funktion definiert, die misst, inwiefern sich bei einem gegebenen Inputvektor und Zielvektor der Output des neuronalen Netzes basierend auf einem Wert der Parameter unterscheidet. Diese Funktion nennt man eine *Kostenfunktion* oder *Zielfunktion*.
- Der summierte Wert der Kostenfunktion über alle Trainingsdatenpunkte wird dann durch Veränderung der Parameter minimiert, so dass Parameterwerte gefunden werden, für die die Abweichung zwischen Zielvektor und Output des neuronalen Netzes bei gegebenemen Inputvektor möglichst gering ist.
- Zur Minimierung der Kostenfunktion wird üblicherweise ein Gradientenverfahren benutzt.
- Zur Berechnung der in diesem Verfahren auftretenden Zielfunktionsgradienten wird ein komputational effizienter Algorithmus eingesetzt, der die spezielle Struktur neuronaler Netze ausnutzt und unter dem Namen *Backpropagation Algorithmus* bekannt ist.

In der Folge wollen wir die Aspekte dieses Lernprozesses genauer betrachten.

Definition (Trainingsdatensatz)

Ein *Trainingsdatensatz* für ein neuronales Netz ist eine Menge

$$\mathcal{D} := \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}, \tag{16}$$

wobei $x^{(i)} \in \mathbb{R}^{n_0}$ *Featurvektor* und $y^{(i)} \in \mathbb{R}^{n_k}$ *Zielvektor* genannt werden.

Bemerkungen

- Im Kontext der zuvor betrachteten multivariaten Verfahren gilt hier $n_0 = m$.
- Typische Zielvektorformate beim Training neuronaler Netze sind
  - $y^{(i)} \in \{0, 1\}$ für binäre Klassifikationsprobleme,
  - $y^{(i)} \in \{0, 1\}^{n_k}$ mit $\sum_{i=1}^{n_k} y_i = 1, n_k > 1$ für n_k-fache Klassifikationsprobleme,
  - $y^{(i)} \in \mathbb{R}^{n_k}, n_k > 1$ für Regressionsprobleme.

Definition (Trainieren eine neuronalen Netzes)

f sei ein k-schichtiges neuronales Netz und $\mathcal{D}$ sei ein Trainingsdatensatz. Dann bezeichnet der Begriff des *Trainierens* den Prozess der Adaptation der Wichtungsmatrizen $W^1, ..., W^k$ des neuronalen Netzes mit dem Ziel, ein Abweichungskriterium zwischen der Ouptutaktivierung $f(x^{(i)})$ und dem assoziierten Wert des Zielvektors $y^{(i)}$ über alle Trainingsdatenpunkte $(x^{(i)}, y^{(i)}), i = 1, ..., n$ des Trainingsdatensatzes $\mathcal{D}$ hinweg zu minimieren.

Bemerkungen

- Wir erinnern an das Ziel $f := \text{argmin}_{\tilde{f} \in F} \|Y - \tilde{f}(X)\|$ der prädiktiven Modellierung.
- Das erwähnte Abweichungskriterium wird in Form von *Kostenfunktionen* definiert.
- Wir benötigen noch den Begriff der *Wichtungsmatrix-variaten neuronalen Netzfunktion*.

Definition (Wichtungsmatrix-variate neuronale Netzfunktion)

f sei ein k-schichtiges neuronales Netz und x sei ein Input von f. Dann ist *Wichtungsmatrix-variate neuronale Netzfunktion f_x von f* definiert als die Funktion

$$\begin{aligned} f_x : \mathbb{R}^{n_1 \times (n_0+1)} \times \cdots \times \mathbb{R}^{n_k \times (n_{k-1}+1)} &\to \mathbb{R}^{n_k}, (W^1, \ldots, W^k) \mapsto f_x(W^1, \ldots, W^k) \\ &:= \Sigma^k(\Phi^k(W^k, \Sigma^{k-1}(\Phi^{k-1}(W^{k-1}, \cdots (W^2, \Sigma^1(\Phi^1(W^1, x))) \cdots)))), \end{aligned} \tag{17}$$

wobei für $l = 1, \ldots, k$, Φ^l die bivariate Potentialfunktion bezeichnet, die der Potentialfunktion $\Phi^l_{W^l}$ in der Definition des neuronalen Netzes entspricht. Weiterhin definieren wir für $l = 1, \ldots, k$, die *Wichtungsmatrix-variate neuronale Netzfunktion der $l$ten Schicht* f_x^l für festes fixed $W^\ell \in \mathbb{R}^{n_\ell \times (n_{\ell-1}+1)}$m mit $\ell = 1, \ldots, k$ und $\ell \neq l$ als

$$f_x^l : \mathbb{R}^{n_1 \times (n_{l-1}+1)} \to \mathbb{R}^{n_k}, W^l \mapsto f_x^l(W^l) := f_x^l(W^1, \ldots, W^k). \tag{18}$$

Bemerkungen

- Die Definition von f in der Definition eines k-schichtiges neuronales Netzes ist eine Funktion des Inputs x bei festen Wichtungsmatrizen $W^1, \ldots, W^l$. Zum Trainieren eines neuronalen Netzes ist es aber entscheidend, bei festem Input den Output des neuronalen Netzes bei Variation der Parameter $W^1, \ldots, W^l$ zu monitoren. Dies motiviert den Begriff der Wichtungsmatrix-variaten neuronalen Netzfunktion: Die Definition von f_x in (17) ist eine Funktion der Wichtungsmatrizen $W^1, \ldots, W^l$ bei festem Input x.

Definition (Output-spezifische Kostenfunktionen)

f sei ein k-schichtiges neuronales Netz und y sei ein Zielvektor von f. Dann wird eine multivariate reelwertige Funktion der Form

$$c_y : \mathbb{R}^{n_k} \to \mathbb{R}, a^k \mapsto c_y(a^k) \tag{19}$$

Output-spezifische Kostenfunktion genannt.

Bemerkung

- Eine Output-spezifische Kostenfunktion c_y misst die Abweichung des Outputs a^k eines neuronalen Netzes von einem Zielvektor y. Untenstehende Tabelle führt zwei typische Beispiele für Output-spezifische Kostenfunktionen und ihre Gradienten, die in der Folge wichtig werden, auf.

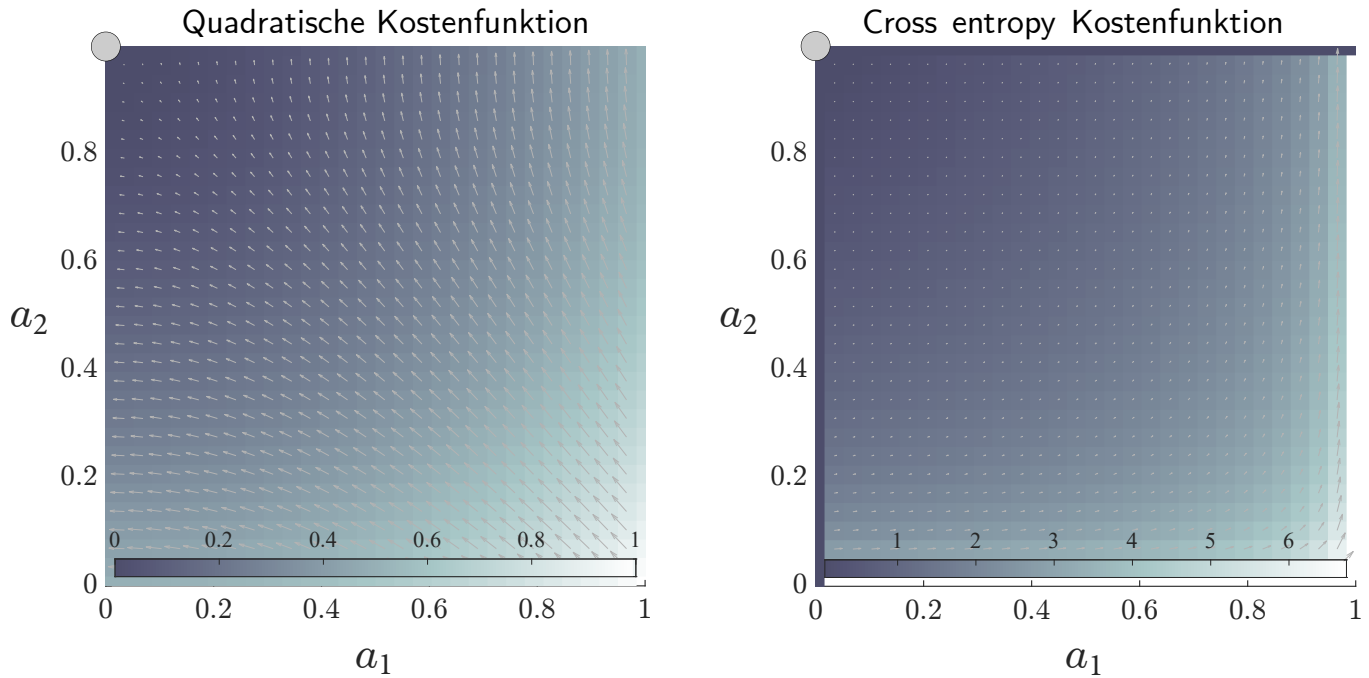
Quadratische Kostenfunktion

Definition	Gradient
$c_y(a^k) := \frac{1}{2}\sum_{j=1}^{n_k}(a_j^k - y_j)^2$	$\nabla c_y(a^k) := \left(a_j^k - y_j\right)_{j=1,\ldots,n_k}$

Cross-entropy Kostenfunktion

Definition	Gradient
$c_y(a^k) := -\sum_{i=1}^{n_k} y_j \ln a_j^k + (1-y_j)\ln(1-a_j^k)$	$\nabla c_y(a^k) := \left(-\frac{y_j}{a_j^k} + \frac{1-y_j}{1-a_j^k}\right)_{j=1,\ldots,n_k}$

Output-spezifische Kostenfunktionswerte für $y = (0, 1)^T$ bei logistischer Aktivierungsfunktion



- Beide Funktionen haben ihr Minimum bei $a = y$.
- Die Pfeile bilden die skalierten Gradientenwerte der jeweiligen Funktion ab.

Definition (Trainingsdatenpunkt-spezifische Kostenfunktionen)

f sei ein k-schichtiges neuronales Netz, f_x sei die zugehörige wichtungsmatrix-variate neuronale Netzfunktion, x und y seien Inputs und Outputs des neuronalen Netzes, $\mathcal{D}$ sei ein Trainingsdatensatz und c_y sei eine Output-spezifische Kostenfunktion. Dann heißt eine multimatrixvariate reellwertige Funktion der Form

$$c_{xy} : \mathbb{R}^{n_1 \times (n_0+1)} \times \cdots \times \mathbb{R}^{n_k \times (n_{k-1}+1)} \to \mathbb{R},$$
$$(W^1, \ldots, W^k) \mapsto c_{xy}(W^1, \ldots, W^k) := c_y(f_x(W^1, \ldots, W^k)) \quad (20)$$

Trainingsdatenpunkt-spezifische Kostenfunktion.

Bemerkung

- Eine Trainingsdatenpunkt-spezifische Kostenfunktion c_{xy} misst die Abweichung des Outputs a^k eines neuronalen Netzes von einem Zielvektor y mithilfe einer Output-spezifischen Kostenfunktion c_y für einen festen Input x als Funktion der (also bei variablen) Wichtungsmatrizen.

Definition (Gewichtsvektor)

f sei ein k-schichtige neuronales Netz mit $n_l \times n_{l-1} + 1$-dimensionalen Gewichtsmatrizen $W^l, l = 1, .., k$ und es sei

$$p := \sum_{l=1}^{n_k} n_l(n_{l-1} + 1). \tag{21}$$

die Anzahl der Gewichtsparameter des neuronalen Netzes. Dann heißt

$$\mathcal{W} := \left(\text{vec}\left(W^l\right)\right)_{1 \leq l \leq k} \in \mathbb{R}^p \tag{22}$$

der *Gewichtsvektor* des neuronalen Netzes.

Bemerkung

- Die Vektorisierung und Konkatenation der Wichtungsmatrizen im Sinne des Gewichtsvektors erlaubt es, dass Trainieren eines neuronalen Netzes als ein Standardoptimierungsproblem einer multivariaten (nicht multimatrixvariaten) reellwertigen zu formulieren.

Definition (Additive Kostenfunktion)

$\mathcal{D}$ sei ein Trainingsdatensatz und c_{xy} sei eine Trainingsdatenpunkt-spezifische Kostenfunktion. Dann nennt man eine multivariate reellwertige Funktion der Form

$$c_{\mathcal{D}} : \mathbb{R}^p \to \mathbb{R}, \mathcal{W} \mapsto c_{\mathcal{D}}(\mathcal{W}) := \frac{1}{n} \sum_{i=1}^{n} c_{x^{(i)}\, y^{(i)}}(W^1, \ldots, W^k) \tag{23}$$

eine *additive Kostenfunktion*.

Bemerkung

- Die additive Kostenfunktion ist die zentrale Zielfunktion beim Trainierung eines neuronalen Netzes.
- $c_{\mathcal{D}}$ ist eine multivariate reellwertige Funktion, es liegt also ein Standardoptimierungsproblem vor.
- Wir nehmen dabei stillschweigend an, dass die sinnvolle Aufteilung des Gewichtsvektors $\mathcal{W}$ auf die Wichtungsmatrizen $W^1, \ldots, W^k$ in der Auswertung der Funktion $c_{\mathcal{D}}$ geschieht.

Definition (Batch Gradientenverfahren für neuronale Netze)

f sei ein k-schichtiges neuronales Netz mit Gewichtsvektor $\mathcal{W}$, $\mathcal{D}$ sei ein Trainingsdatensatz bestehend aus n Trainingsdatenpunkten, und $c_{\mathcal{D}}$ sei eine additive Kostenfunktion mit assoziierter Trainingsexemplar-spezifischer Kostenfunktion c_{xy}. Dann ist ein Gradientenverfahren zur Minimierung der additiven Kostenfunktion $c_{\mathcal{D}}$ (und damit zum Lernen der Parameter von f) definiert durch

Initialisierung

Wahl eines Startpunktes $\mathcal{W}^{(0)}$ und einer Lernrate $\alpha > 0$.

Iterationen

Für $j = 1, 2, \ldots$ setze

$$\mathcal{W}^{(j)} := \mathcal{W}^{(j-1)} - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla c_{x^{(i)}\, y^{(i)}}(W^1, \ldots, W^k), \tag{24}$$

wobei

$$\nabla c_{x^{(i)}\, y^{(i)}}(W^1, \ldots, W^k) = \left(\nabla_{W^l} c_{x^{(i)}\, y^{(i)}}(W^1, \ldots, W^k) \right)_{1 \leq l \leq k} \tag{25}$$

für $i = 1, \ldots, n$ den Gradienten der $i$ten Trainingsexemplar-spezifischen Kostenfunktion bezeichnet

Bemerkungen

- $\mathcal{W}^{(j)}$ wird in (22) in die negative Richtung des Gradientenmittelwerts über Trainingsdatenpunkte adaptiert. Wird der Gradientenmittelwert dagegen nur über eine zufällig gewählte Teilmenge der Trainingsdatenpunkte berechnet, so spricht man von einem *stochastischen Gradientenverfahren*.

Vorbemerkungen

Funktionale Architektur

Training

Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Wesen und Motivation des Backpropagation Algorithmus

Der Backpropagation (BP) Algorithmus dient der numerischen Bestimmung der Komponenten

$$\frac{\partial}{\partial w_{ij}^l} c_{xy}(W^1, ..., W^k) \text{ für alle } i = 1, ..., n_l, j = 1, ..., n_{l-1} + 1, \text{ und } l = 1, ..., k. \quad (26)$$

des Gradienten $\nabla c_{x^{(i)}y^{(i)}}(W^1, ..., W^k)$ der $i$ten Trainingsexemplar-spezifischen Kostenfunktion.

Prinzipiell können diese partiellen Ableitungen numerisch durch

$$\frac{\partial}{\partial w_{ij}^l} c_{xy}(W^1, ..., W^k) \approx \frac{c_{xy}(W^1, ..., \tilde{W}^l, ..., W^k) - c_{xy}(W^1, ..., W^l, ..., W^k)}{\epsilon}, \quad (27)$$

mit

- $\tilde{W}^l := W^l + 1_{ij}^l \epsilon$,
- einer Matrix $1_{ij}^l \in \mathbb{R}^{n_l \times (n_{l-1}+1)}$ aus 0en mit einer 1 an der w_{ij}^l Stelle in W^l und
- einem Schrittweitenparameter $\epsilon > 0$

approximiert werden (vgl. Definition der partiellen Ableitung).

Wesen und Motivation des Backpropagation Algorithmus

Dieses Vorgehen würde für jede Iteration des Gradientenverfahren und für jeden Trainingsdatenpunkt

$$K := 1 + \sum_{l=1}^{k} n_l (n_{l-1} + 1) \tag{28}$$

Auswertungen der Trainingsdatenpunkt-spezifischen Kostenfunktion c_{xy} und somit von f erfordern. Man nennt die Auswertung von f für einen Trainingsdatenpunkt x einen *Forward Pass*.

Die zentrale Eigenschaft des Backpropagation Algorithmus ist es, für die Auswertung von ∇c_{xy} die Anzahl der notwendigen *Forward Passes* pro Gradientenverfahrensiteration von K auf 1 zu reduzieren.

Um dies zu erreichen, nutzt der Backpropagation Algorithmus einen sogenannten *Backward Pass*, der die gleiche komputationale Komplexität wie der *Forward Pass* hat und auf einer multivariate Version der Kettenregel der Differentialrechnung sowie der repetitiven funktionalen Architektur neuronaler Netze beruht.

Der Backpropagation Algorithmus reduziert die Anzahl nötiger *Passes* zur Auswertung von ∇c_{xy} also von K *Forward Passes* auf 1 *Forward Pass* und 1 *Backward Pass*.

Theorem (Backpropagation Algorithmus)

f sei ein k-schichtiges neuronales Netz, $W^l_\bullet \in \mathbb{R}^{n_l \times n_{l-1}}$ seien für $l = 1, \ldots, k$ Matrizen, die durch das Entfernen der letzten Spalte der Wichtungsmatrizen $W^l \in \mathbb{R}^{n_l \times n_{l-1}+1}$ entstehen, c_{xy} sei eine Trainingsdatenpunkt-spezifische Kostenfunktion, $\nabla c_y(a^k)$ sei der Gradient der Output-spezifischen Kostenfunktion, $\tilde{\Sigma}^l(z^l) := (\sigma'(z^l_1), \ldots, \sigma'(z^l_{n_l}))^T$ sei der Vektor der Aktivierungsfunktionenableitungen ausgewertet an der Stelle z^l und $\Sigma^l(z^l)$ die komponentenweise Aktivierungsfunktion evaluiert an der Stelle z^l. Dann können die partiellen Gradienten von c_{xy} hinsichtlich der Wichtungsmatrizen W^l für $l = k, k-1, \ldots, 1$ mit folgendem Algorithmus berechnet werden:

Initialisierung

Setze $W^{k+1} := (1 \quad 0)$ und $\delta^{k+1} := \nabla c_y(a^k)$.

Iterationen

Für $l = k, k-1, k-2, \ldots, 1$, setze

$$\delta^l := \left(\left(W^{l+1}_\bullet \right)^T \cdot \delta^{l+1} \right) \circ \tilde{\Sigma}^l(z^l) \tag{29}$$

und

$$\nabla_{W^l} c_{xy}(W^1, \ldots, W^k) := \text{vec}\left(\delta^l \cdot \left(\Sigma^{l-1}(z^{l-1})^T \quad 1 \right) \right), \tag{30}$$

mit Rekursionstermination durch $\Sigma^0(z^0) := x^T$ und dem Hadamard-Produkt $\circ$.

Für weitere Details und einen Beweis verweisen wir auf Ostwald and Usée (2021).

Vorbemerkungen

Funktionale Architektur

Training

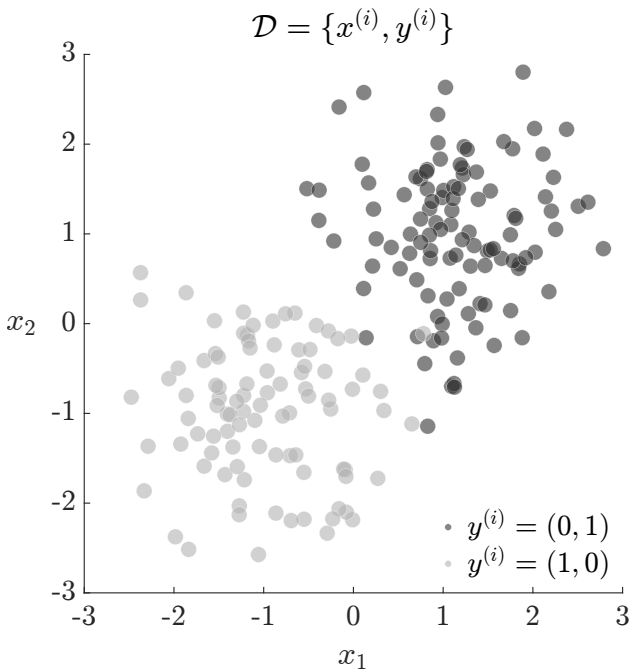
Backpropagation

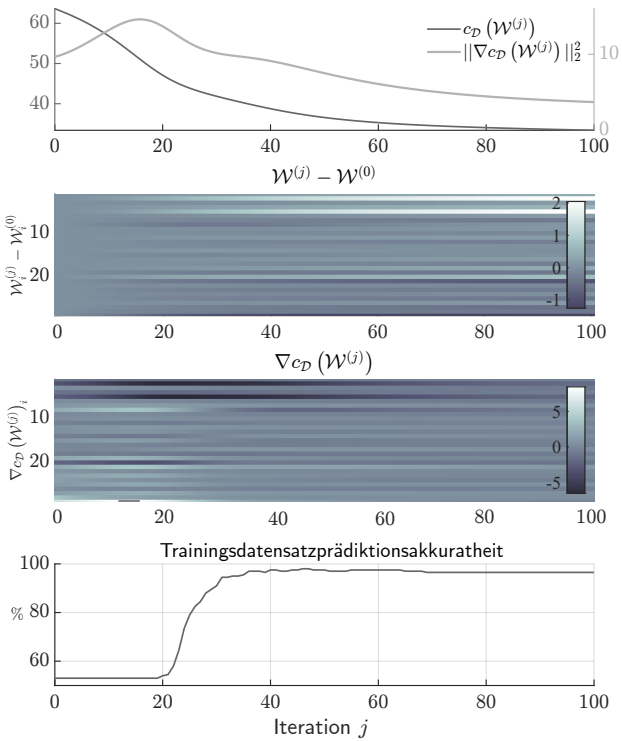
Anwendungsbeispiele

Selbstkontrollfragen

Simulation und Analyse mit Matlab Implementation (Ostwald and Usée (2021))

- Neuronales Netz mit $k = 3, n_0 = 2, n_1 = 3, n_2 = 3, n_3 = 2$.
- Trainingsdatensatz anhand eines LDA Modells simuliert.





Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

$\Rightarrow$ Eine verdeckte Schicht mit zwei Neuronen

```r
# R Pakete
library(foreign)
library(neuralnet)

# Einlesen der Studienerfolgdaten und Fokus auf binäre Klassifikation
D              = read.spss(file.path(getwd(), "9_Daten", "studienerfolg.sav"),
                           to.data.frame = T)
D              = D[D[,1] != "befriedigend",]                  # Binärisierung des Datensatzes
D$Intelligenz  = D$X1                                         # Variablenbennenung
D$Mathematik   = D$X2                                         # Variablenbenennung
D$Erfolg       = c(rep(0,15), rep(1,15))                      # Recoding der Binärisierung


# Trainieren eines neuronalen Netzes mit 2 Hidden Neurons
set.seed(7)                                                   # Der Algorithmus ist nicht deterministisch!
nn             = neuralnet(Erfolg ~ Intelligenz + Mathematik, # y^{(i)}, x^{(i)} Definitionen
                           data          = D,                 # Datensatz
                           hidden        = 2,                 # 2 Neurone in 1 verdeckte Schicht
                           err.fct       = "ce",              # Cross Entropie Kostenfunktion
                           linear.output = FALSE)             # Sigmoide Aktivierungsfunktion


# Resultatsaufbereitung
R              = data.frame(nn$covariate, nn$response, nn$net.result[[1]], as.numeric(nn$net.result[[1]] > 0.5))
colnames(R)    = c("Intelligenztest", "Mathematiktest", "Erfolg", "Output", "Prädiktion")
print(sprintf("Prädiktionsakkuratheit = %0.2f", mean(R$Prädiktion == R$Erfolg)))
```

```
> [1] "Prädiktionsakkuratheit = 0.80"
```

Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

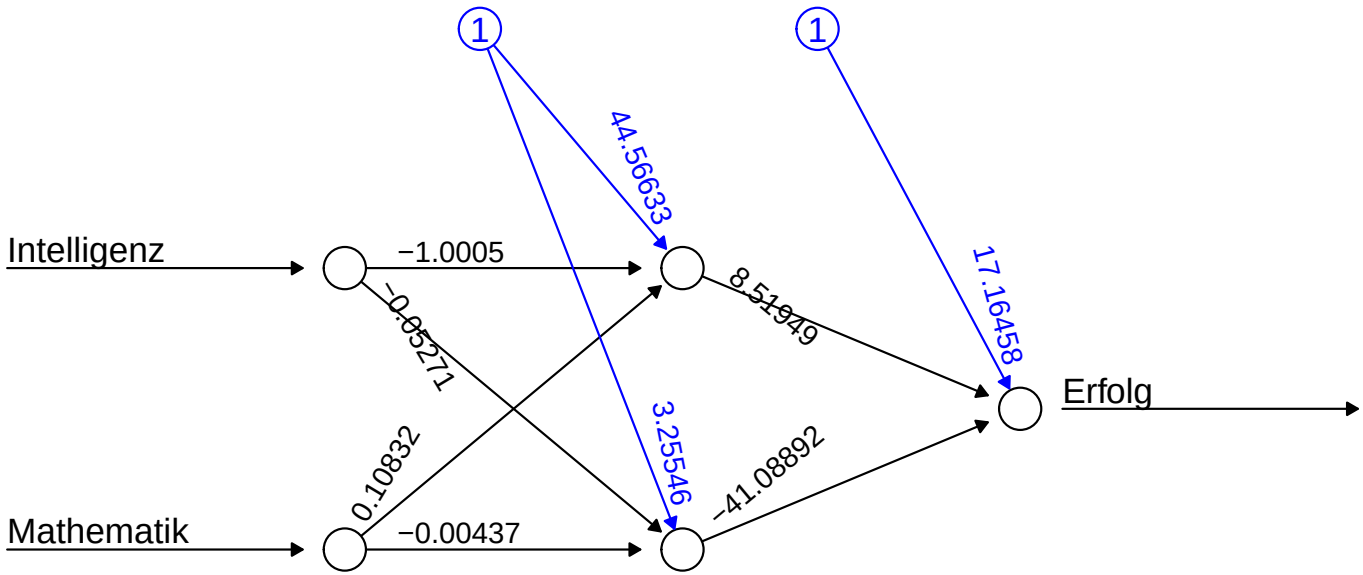
$\Rightarrow$ Eine verdeckte Schicht mit zwei Neuronen

	Intelligenztest	Mathematiktest	Erfolg	Output	Prädiktion
1	54	44	0	0.0	0
2	60	20	0	0.0	0
3	67	36	0	0.7	1
4	41	39	0	0.0	0
5	66	57	0	0.8	1
6	51	28	0	0.0	0
7	51	46	0	0.0	0
8	37	46	0	0.0	0
9	57	54	0	0.0	0
10	47	12	0	0.0	0
11	50	67	0	0.6	1
12	42	63	0	0.1	0
13	60	64	0	0.2	0
14	36	64	0	0.0	0
15	60	71	0	0.2	0
31	71	41	1	1.0	1
32	65	28	1	0.4	0
33	67	76	1	0.9	1
34	68	54	1	0.9	1
35	75	33	1	1.0	1
36	71	82	1	1.0	1
37	68	64	1	0.9	1
38	63	72	1	0.6	1
39	48	54	1	0.4	0
40	53	86	1	0.8	1
41	62	71	1	0.5	0
42	69	25	1	0.8	1
43	67	72	1	0.9	1
44	74	92	1	1.0	1
45	76	75	1	1.0	1

Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

⇒ Eine verdeckte Schicht mit zwei Neuronen

```
plot(nn)
```



Error: 8.46792 Steps: 64768

Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

⇒ Zwei verdeckte Schichten mit drei Neuronen

```
# R Pakete
library(foreign)
library(neuralnet)

# Einlesen der Studienerfolgdaten und Fokus auf binäre Klassifikation
D              = read.spss(file.path(getwd(), "9_Daten", "studienerfolg.sav"),
                           to.data.frame = T)
D              = D[D[,1] != "befriedigend",]                  # Binärisierung des Datensatzes
D$Intelligenz  = D$X1                                         # Variablenbennenung
D$Mathematik   = D$X2                                         # Variablenbenennung
D$Erfolg       = c(rep(0,15), rep(1,15))                      # Recoding der Binärisierung


# Trainieren eines neuronalen Netzes mit 2 Hidden Neurons
set.seed(7)                                                   # Der Algorithmus ist nicht deterministisch!
nn             = neuralnet(Erfolg ~ Intelligenz + Mathematik, # y^{(i)}, x^{(i)} Definitionen
                           data          = D,                 # Datensatz
                           hidden        = c(3,3),            # 3 Neurone in 2 verdeckten Schichten
                           err.fct       = "ce",              # Cross Entropie Kostenfunktion
                           linear.output = FALSE)             # Sigmoide Aktivierungsfunktion


# Resultstatsaufbereitung
R              = data.frame(nn$covariate, nn$response, nn$net.result[[1]], as.numeric(nn$net.result[[1]] > 0.5))
colnames(R)    = c("Intelligenztest", "Mathematiktest", "Erfolg", "Output", "Prädiktion")
print(sprintf("Prädiktionsakkuratheit = %0.2f", mean(R$Prädiktion == R$Erfolg)))
```

```
> [1] "Prädiktionsakkuratheit = 0.97"
```

Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

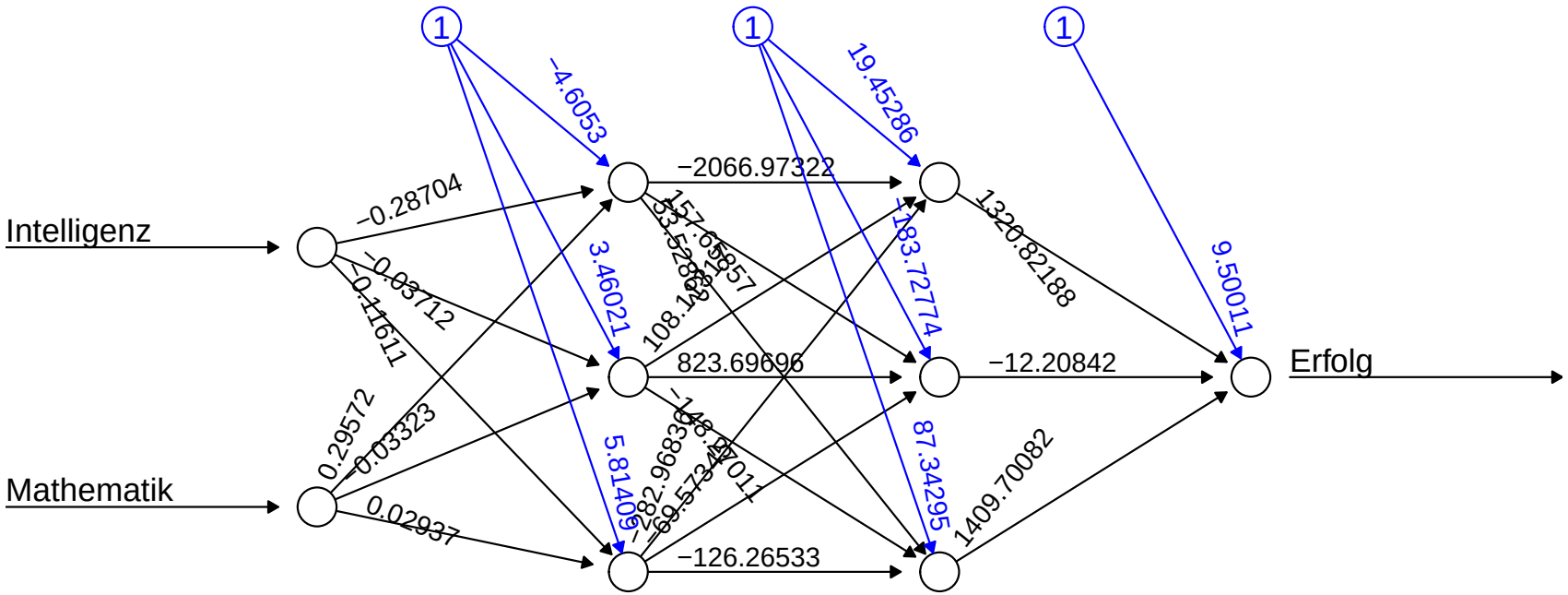
$\Rightarrow$ Zwei verdeckte Schichten mit drei Neuronen

	Intelligenztest	Mathematiktest	Erfolg	Output	Prädiktion
1	54	44	0	0.1	0
2	60	20	0	0.1	0
3	67	36	0	0.1	0
4	41	39	0	0.1	0
5	66	57	0	0.1	0
6	51	28	0	0.1	0
7	51	46	0	0.1	0
8	37	46	0	0.1	0
9	57	54	0	0.1	0
10	47	12	0	0.1	0
11	50	67	0	0.1	0
12	42	63	0	0.1	0
13	60	64	0	0.1	0
14	36	64	0	0.1	0
15	60	71	0	0.1	0
31	71	41	1	1.0	1
32	65	28	1	1.0	1
33	67	76	1	1.0	1
34	68	54	1	1.0	1
35	75	33	1	1.0	1
36	71	82	1	1.0	1
37	68	64	1	1.0	1
38	63	72	1	1.0	1
39	48	54	1	0.1	0
40	53	86	1	1.0	1
41	62	71	1	1.0	1
42	69	25	1	1.0	1
43	67	72	1	1.0	1
44	74	92	1	1.0	1
45	76	75	1	1.0	1

Prädiktive Modellierung von *studienerfolg.sav* mit neuralnet() (Günther and Fritsch (2010))

⇒ Zwei verdeckte Schichten mit drei Neuronen

```
plot(nn)
```



Vorbemerkungen

Funktionale Architektur

Lernen

Backpropagation

Anwendungsbeispiel

Selbstkontrollfragen

Selbstkontrollfragen

1. Erläutern Sie die zentralen Ideen Universeller Approximationstheoreme im Kontext neuronaler Netze.
2. Nennen Sie die Formel für das Potential z_i^l eines Neurons i in einer Schicht l eines neuronalen Netzes und erläutern Sie die verschiedenen Komponenten dieser Formel und ihre intuitive Bedeutung.
3. Skizzieren Sie die Standard Logistic und ReLU Aktivierungsfunktionen und ihre Ableitungen.
4. Nennen Sie die Formel für die Aktivierung a_i^l eines Neurons i in einer Schicht l eines neuronalen Netzes und erläutern Sie ihre Bestandteile und deren intuitive Bedeutung.
5. Erläutern Sie das prinzipielle Vorgehen zum Trainieren eines neuronalen Netzes.
6. Definieren Sie die (Output-spezifische) Quadratische Kostenfunktion und erläutern Sie ihre Bestandteile.
7. Geben Sie das Batch Gradientenverfahren zum Trainieren neuronaler Netze wieder.
8. Differenzieren Sie die Begriffe Batch und Stochastischen Gradientenverfahren zum Trainieren neuronaler Netze.
9. Erläutern Sie Wesen und Motivation des Backpropagation Algorithmus.
10. Lesen Sie den Datensatz studienerfolg.sav mit R ein und bestimmen Sie den Trainingsdatenprädiktionsfehler nach Trainieren eines neuronalen Netzes mithilfe des R Pakets neuralnet zur prädiktiven Modellierung des Studienerfolgs (gut, ungenügend) basierend auf (1) den Intelligenztestdaten, (2) den Intelligenz- und Mathematiktestdaten und (3) den Intelligenztest-, Mathematiktest-, und Gewissenhaftigkeitsdaten. Wiederholen Sie Ihre Analyse zur Bestimmung des Generalisierungsfehlers im Rahmen einer Leave-One-Out Kreuzvalidierung.

References I

Cybenko, G. 1989. "Approximation by Superpositions of a Sigmoidal Function." *Mathematics of Control, Signals, and Systems*, 12.

Dwyer, Dominic B., Peter Falkai, and Nikolaos Koutsouleris. 2018. "Machine Learning Approaches for Clinical Psychology and Psychiatry." *Annual Review of Clinical Psychology* 14 (1): 91–118. https://doi.org/10.1146/annurev-clinpsy-032816-045037.

Friedman, Avner. 1970. *Foundations of Modern Analysis*. Dover Publications.

Günther, Frauke, and Stefan Fritsch. 2010. "Neuralnet: Training of Neural Networks." *The R Journal* 2 (1): 30. https://doi.org/10.32614/RJ-2010-006.

Hanin, Boris, and Mark Sellke. 2018. "Approximating Continuous Functions by ReLU Nets of Minimal Width." *arXiv:1710.11278 [Cs, Math, Stat]*, March. https://arxiv.org/abs/1710.11278.

Hopfield, J. J. 1982. "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences* 79 (8): 2554–58. https://doi.org/10.1073/pnas.79.8.2554.

Hornik, Kurt. 1991. "Approximation Capabilities of Multilayer Feedforward Networks." *Neural Networks* 4 (2): 251–57. https://doi.org/10.1016/0893-6080(91)90009-T.

Kidger, Patrick, and Terry Lyons. 2020. "Universal Approximation with Deep Narrow Networks." *arXiv:1905.08539 [Cs, Math, Stat]*, June. https://arxiv.org/abs/1905.08539.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–44. https://doi.org/10.1038/nature14539.

Leshno, Moshe, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. 1993. "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function." *Neural Networks* 6 (6): 861–67. https://doi.org/10.1016/S0893-6080(05)80131-5.

Lu, Zhou, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. 2017. "The Expressive Power of Neural Networks: A View from the Width." *arXiv:1709.02540 [Cs]*, November. https://arxiv.org/abs/1709.02540.

McCulloch, Warren S, and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5: 115–33.

Minsky, Marvin, and Seymour A. Papert. 1969. *Perceptrons: An Introduction to Computational Geometry*. 2. print. with corr. Cambridge/Mass.: The MIT Press.

Ostwald, Dirk, and Franziska Usée. 2021. "An Induction Proof of the Backpropagation Algorithm in Matrix Notation." *arXiv:2107.09384 [Cs, Math, q-Bio, Stat]*, July. https://arxiv.org/abs/2107.09384.

Pinkus, Allan. 1999. "Approximation Theory of the MLP Model in Neural Networks." *Acta Numerica* 8 (January): 143–95. https://doi.org/10.1017/S0962492900002919.

References II

Rosenblatt, F. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review* 65 (6): 386–408. https://doi.org/10.1037/h0042519.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1986. "Learning Representations by Back-Propagating Errors." *Nature*, no. 323: 533–36.

Schmidhuber, Jürgen. 2015. "Deep Learning in Neural Networks: An Overview." *Neural Networks* 61 (January): 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(10) T^2-Tests

Vorbemerkungen

Einstichproben-T^2-Tests

Zweistichproben-T^2-Tests

Univariates vs. multivariates Testen

Selbstkontrollfragen

Ausgewählte multivariate Methoden der Frequentistischen Inferenz

Multivariate Generalisierungen bekannter Frequentistischer Verfahren

T^2-Tests als Generalisierung von T-Tests

- Inferentieller Vergleich von ein bis zwei Gruppen multivariater Daten

Multivariate **Einfaktorielle Varianzanalyse** als Generalisierung der einfaktoriellen Varianzanalyse

- Inferentieller Vergleich von drei oder mehr Gruppen multivariater Daten

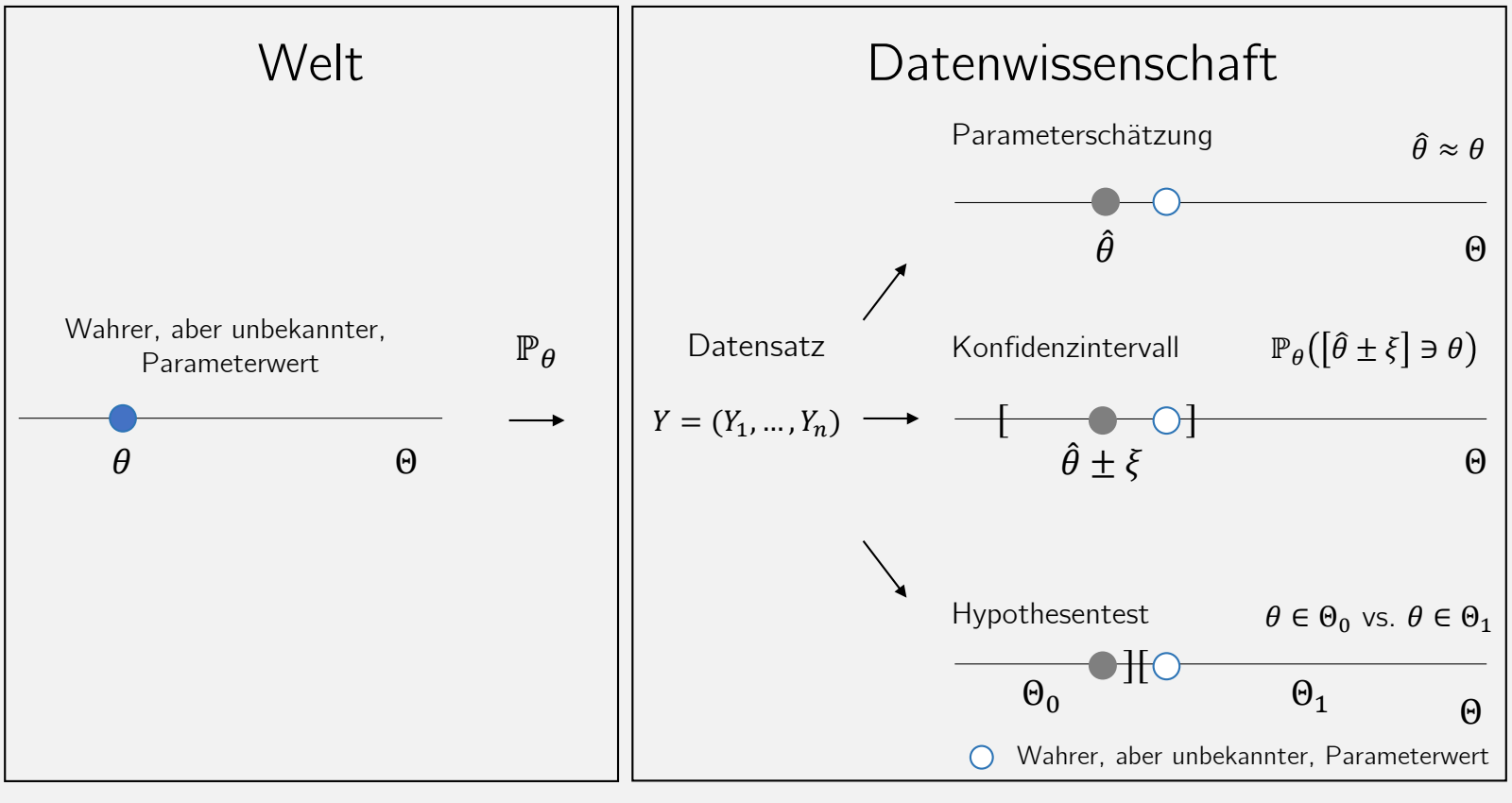
Kanonische Korrelationsanalyse als Generalisierung (multipler) Korrelation

- Korrelative Zusammenhänge von Zufallsvektoren

Zur Revision univariater Frequentistischer Verfahren

- Inferenzstatistik SoSe 21
- Wahrscheinlichkeitstheorie und Frequentistische Inferenz WiSe 21/22

Modell und Standardprobleme der Frequentistischen Inferenz



Standardprobleme Frequentistischer Inferenz

(1) **Parameterschätzung**

Ziel der Parameterschätzung ist es, einen möglichst guten Tipp für den wahren, aber unbekannten, Parameterwert (oder eine Funktion dessen) abzugeben, typischerweise basierend auf der Beobachtung einer Realisierung von $Y_1, ..., Y_n \sim \mathbb{P}_\theta$.

(2) **Konfidenzintervalle**

Das Ziel der Bestimmung von Konfidenzintervallen ist es, basierend auf der Verteilung möglicher Parameterschätzwerte eine quantitative Aussage über die mit dem Schätzwert assoziierte Unsicherheit zu treffen.

(3) **Hypothesentests**

Das Ziel der Auswertung von Hypothesentests ist es, basierend auf der angenommenen Verteilung der Beobachtungen $Y_1, ..., Y_n$ in einer möglichst sinnvollen Form zu entscheiden, ob der wahre, aber unbekannte Parameterwert, in einer von zwei sich gegenseitig ausschließenden Untermengen des Parameterraumes, welche man als Hypothesen bezeichnet, liegt.

Standardannahmen Frequentistischer Inferenz

$\mathcal{M}$ sei ein statistisches Modell mit unabhängig und identisch verteilten Zufallsvektoren $Y_1, \ldots, Y_n \sim \mathbb{P}_\theta$. Es wird angenommen, dass ein vorliegender Datensatz eine der möglichen Realisierungen von $Y_1, \ldots, Y_n \sim \mathbb{P}_\theta$ ist. Aus frequentistischer Sicht kann man die Erhebung von Datensätzen unendlich oft wiederholen und zu jedem Datensatz Statistiken auswerten.

Datensatz (1) : $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, \ldots, y_n^{(1)}\right)$, Statistik (1): $S : \mathbb{R}^{m \times n} \to \Sigma, y^{(1)} \mapsto S\left(y^{(1)}\right)$

Datensatz (2) : $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, \ldots, y_n^{(2)}\right)$, Statistik (2): $S : \mathbb{R}^{m \times n} \to \Sigma, y^{(2)} \mapsto S\left(y^{(2)}\right)$

Datensatz (3) : $y^{(3)} = \left(y_1^{(3)}, y_2^{(3)}, \ldots, y_n^{(3)}\right)$, Statistik (3): $S : \mathbb{R}^{m \times n} \to \Sigma, y^{(3)} \mapsto S\left(y^{(3)}\right)$

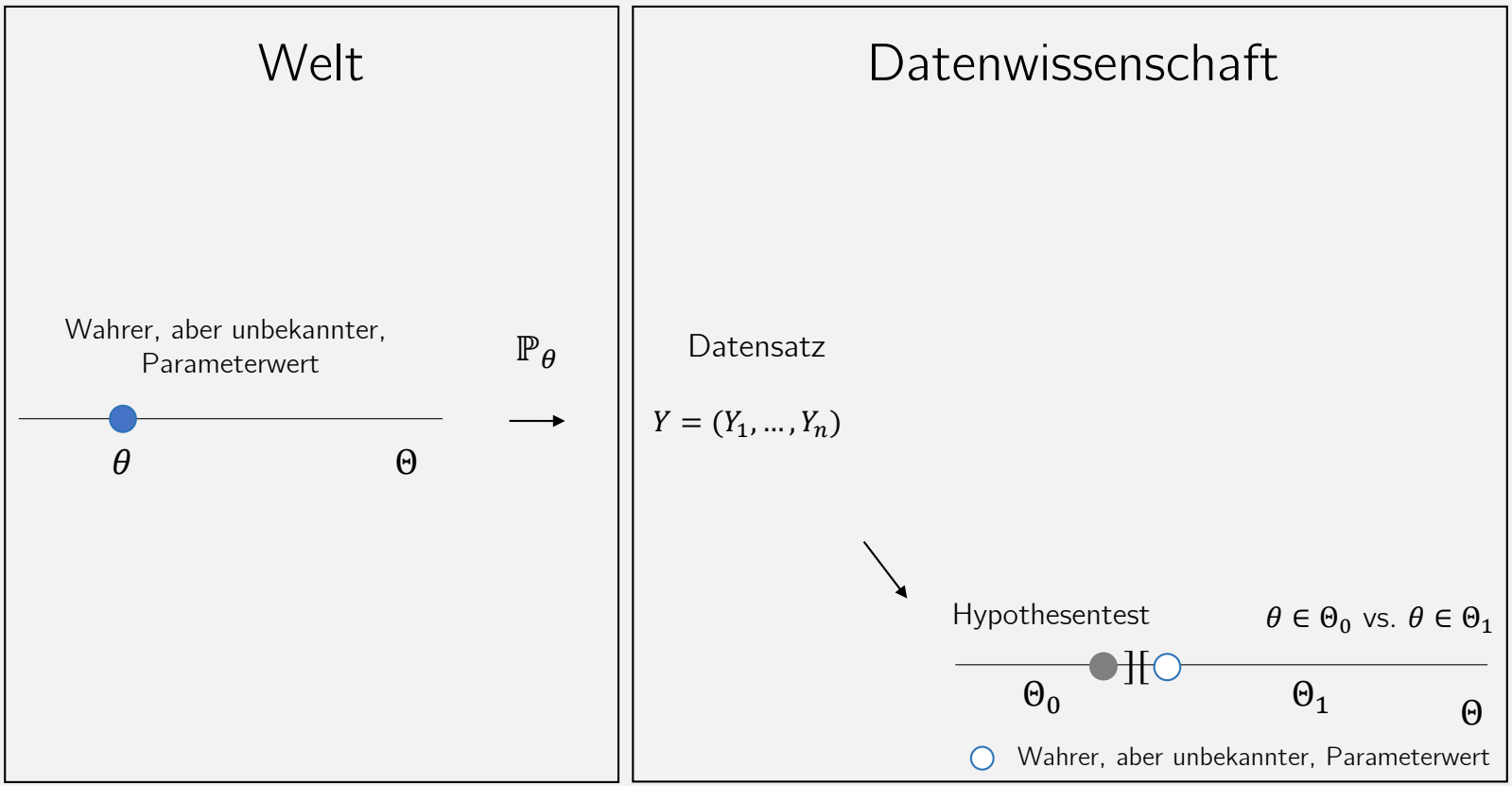
Datensatz (4) : $y^{(4)} = \left(y_1^{(4)}, y_2^{(4)}, \ldots, y_n^{(4)}\right)$, Statistik (4): $S : \mathbb{R}^{m \times n} \to \Sigma, y^{(4)} \mapsto S\left(y^{(4)}\right)$

...

Um die Qualität statistischer Methoden zu beurteilen betrachtet die frequentistische Statistik deshalb die Wahrscheinlichkeitsverteilungen von Statistiken und Schätzern unter der Annahme von $Y_1, \ldots, Y_n \sim \mathbb{P}_\theta$.

Wenn eine statistische Methode im Sinne der frequentitischen Standardannahmen “gut” ist, dann heißt das also, dass sie bei häufiger Anwendung “im Mittel gut” ist. Im Einzelfall, also im realen Normalfall nur eines vorliegenden Datensatzes, kann sie auch “schlecht” sein.

Modell und Standardprobleme der Frequentistischen Inferenz



Definition (Mahalanobis Distanz)

ξ_1 sei ein Zufallsvektor, eine Realisiation eines Zufallsvektors, ein multivariater Erwartungswert oder ein multivariates Stichprobenmittel, ξ_2 sei ein Zufallsvektor, eine Realisiation eines Zufallsvektors, ein multivariater Erwartungswert oder ein multivariates Stichprobenmittel und Ξ sei eine Kovarianzmatrix oder eine Stichprobenkovarianzmatrix. Dann heißt

$$D = (\xi_1 - \xi_2)^T \, \Xi^{-1} \, (\xi_1 - \xi_2) \tag{1}$$

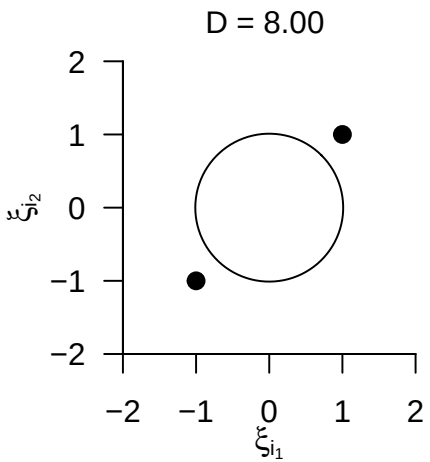
Mahalanobis Distanz von ξ_1 und ξ_2 hinsichtlich Ξ.

Bemerkungen

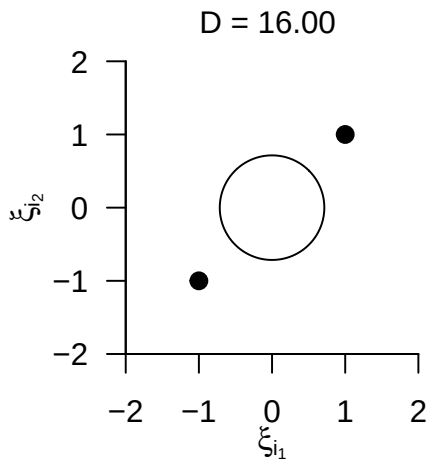
- Eine Mahalanobis Distanz ist eine Kovarianzmatrix-normalisierte quadrierte Euklidische Distanz.
- Ähnliche Maße in der univariaten Statistik sind die z-Transformation $z = \frac{y-\mu}{\sigma}$ und Cohen's $d = \frac{\bar{y}_1 - \bar{y}_2}{s_{12}}$.
- Ähnlich wie bei z-Werten wird bei der Mahalanobis Distanz in "Einheiten von Kovarianzen" gemessen.
- Stark variante Komponenten von ξ_1 und ξ_2 tragen weniger zur Distanz bei.
- Stark kovariante Komponenten von ξ_1 und ξ_2 tragen weniger zur Distanz bei.

Mahalanobis Distanzen als Funktion von Komponentenvarianzen

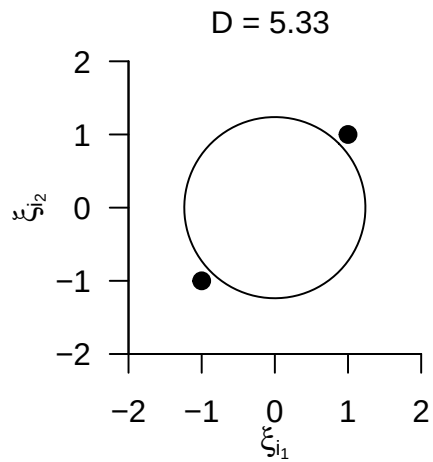
$$\Sigma := \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$



$$\Sigma := \begin{pmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$$

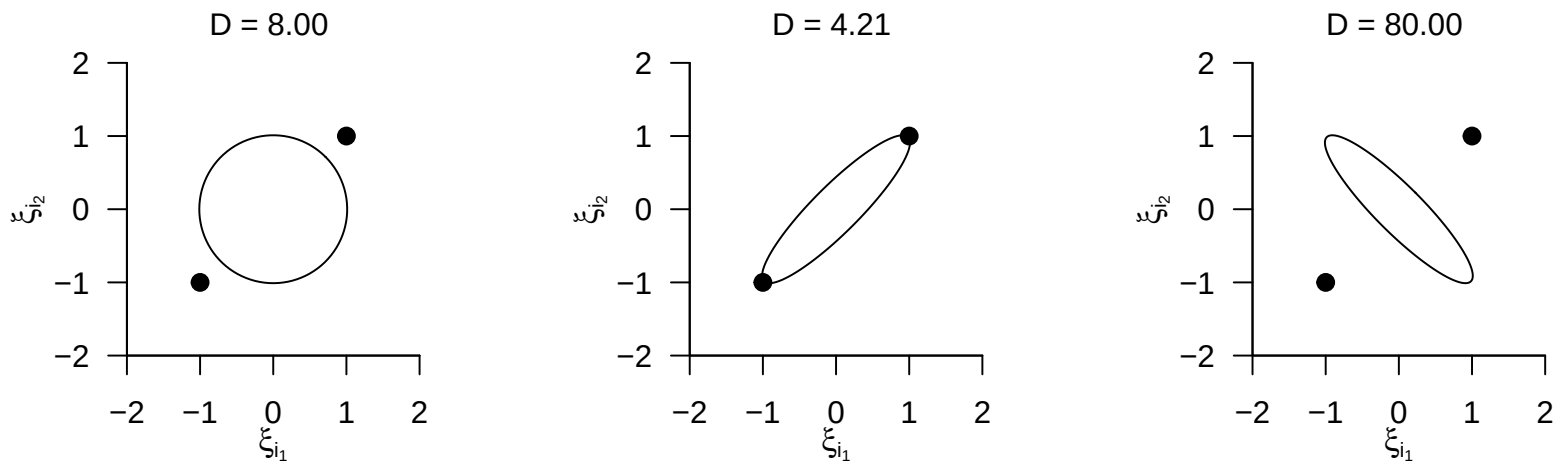


$$\Sigma := \begin{pmatrix} 1.5 & 0.0 \\ 0.0 & 1.5 \end{pmatrix}$$



Mahalanobis Distanzen als Funktion von Komponentenkovarianzen

$$\Sigma := \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} \qquad \Sigma := \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix} \qquad \Sigma := \begin{pmatrix} 1.0 & -0.9 \\ -0.9 & 1.0 \end{pmatrix}$$



Definition (f-Zufallsvariable)

X sei eine Zufallsvariable mit Ergebnisraum $\mathbb{R}_{>0}$ und Wahrscheinlichkeitsdichtefunktion

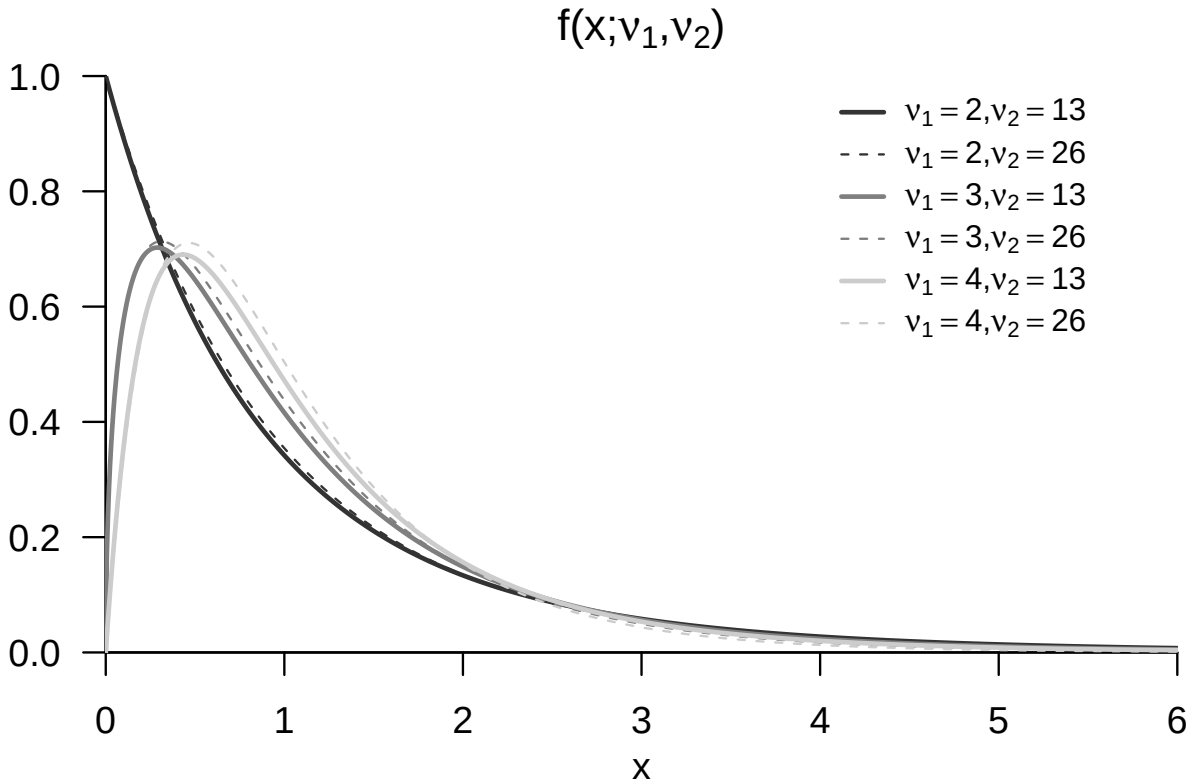
$$p_X : \mathbb{R} \to \mathbb{R}_{>0}, x \mapsto p_X(x) := {\nu_1}^{\frac{\nu_1}{2}} {\nu_2}^{\frac{\nu_2}{2}} \frac{\Gamma\left(\frac{\nu_1+\nu_2}{2}\right)}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)} \frac{x^{\frac{\nu_1}{2}-1}}{(\nu_1 x + \nu_2)^{\frac{\nu_1+\nu_2}{2}}}, \tag{2}$$

wobei Γ die Gammafunktion bezeichne. Dann sagen wir, dass X einer f-Verteilung mit Freiheitsgradparametern ν_1 und ν_2 unterliegt und nennen X eine f-Zufallsvariable mit Freiheitsgradparametern ν_1 und ν_2. Wir kürzen dies mit $X \sim f(\nu_1, \nu_2)$ ab. Die Wahrscheinlichkeitsdichtefunktion (WDF) einer f-Zufallsvariable bezeichnen wir mit $f(x; \nu_1, \nu_2)$, die kumulative Verteilungsfunktion (KVF) einer f-Zufallsvariable bezeichnen wir mit $F(x; \nu_1, \nu_2)$, und die inverse kumulative Verteilungsfunktion einer f-Zufallsvariable bezeichnen wir mit $F^{-1}(x; \nu_1, \nu_2)$.

Bemerkungen

- Im univariaten Fall ist die F-Statistik der Varianzanalyse bei Zutreffen der Nullhypothese f-verteilt
- Im multivariaten Fall ist z.B. die T^2-Statistik bei Zutreffen der Nullhypothese f-verteilt.

Wahrscheinlichkeitsdichtefunktionen von f-Verteilungen



Definition (Nichtzentrale f-Zufallsvariable)

X sei eine Zufallsvariable mit Ergebnisraum $\mathbb{R}_{>0}$ und Wahrscheinlichkeitsdichtefunktion

$$p_X : \mathbb{R} \to \mathbb{R}_{>0}, x \mapsto$$

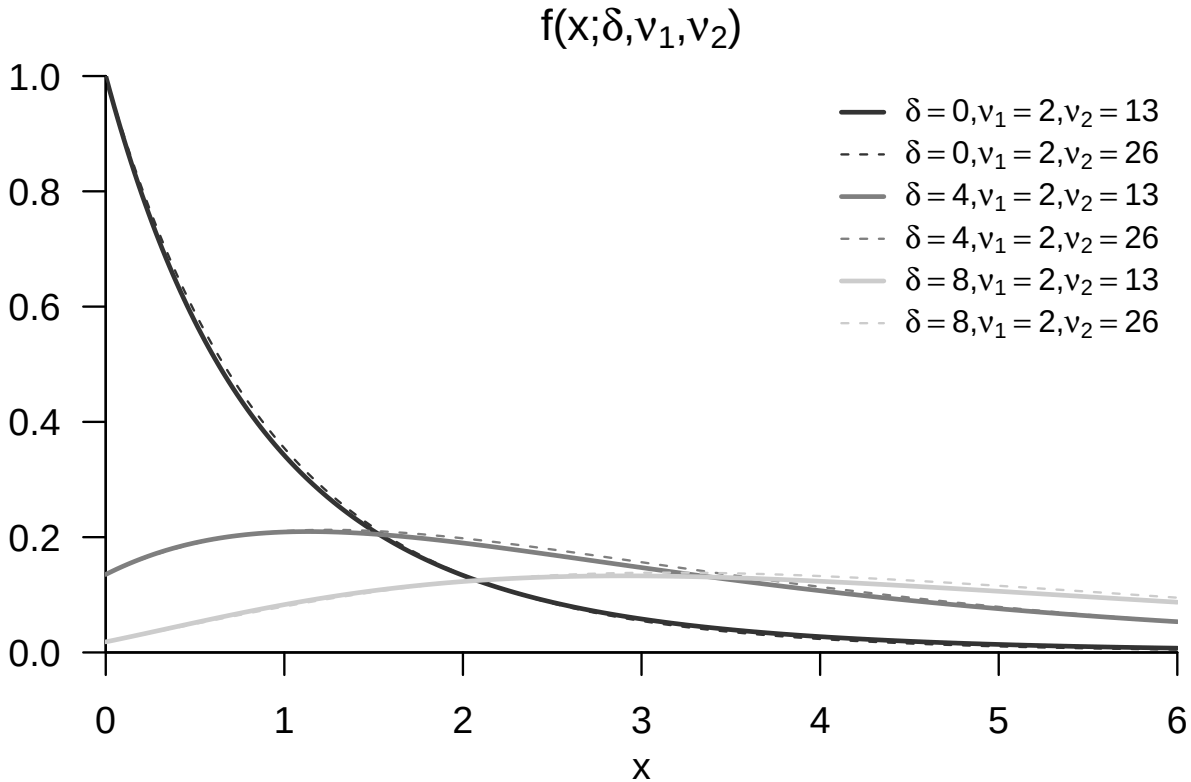
$$p_X(x) := \sum_{k=0}^{\infty} \frac{e^{-\delta/2}(\delta/2)^k}{\frac{\Gamma(\nu_2/2)\Gamma(\nu_1/2+k)}{\Gamma(\nu_2/2+\nu_1/2+k)}k!} \left(\frac{\nu_1}{\nu_2}\right)^{\nu_1/2+k} \left(\frac{\nu_2}{\nu_2+\nu_1 x}\right)^{(\nu_1+\nu_2)/2+k} x^{\nu_1/2-1+k} \quad (3)$$

wobei Γ die Gammafunktion bezeichne. Dann sagen wir, dass X einer nichtzentralen f-Verteilung mit Nichtzentralitätsparameter δ und Freiheitsgradparametern ν_1 und ν_2 unterliegt und nennen X eine nichtzentrale f-Zufallsvariable mit Nichtzentralitätsparameter δ und Freiheitsgradparametern ν_1 und ν_2. Wir kürzen dies mit $X \sim f(\delta, \nu_1, \nu_2)$ ab. Die Wahrscheinlichkeitsdichtefunktion (WDF) einer f-Zufallsvariable bezeichnen wir mit $f(x; \delta, \nu_1, \nu_2)$, die kumulative Verteilungsfunktion (KVF) einer nichtzentralen f-Zufallsvariable bezeichnen wir mit $F(x; \delta, \nu_1, \nu_2)$, und die inverse kumulative Verteilungsfunktion einer nichtzentralen f-Zufallsvariable bezeichnen wir mit $F^{-1}(x; \delta, \nu_1, \nu_2)$.

Bemerkungen

- Es gilt $f(0, \nu_1, \nu_2) = f(\nu_1, \nu_2)$.
- Im univariaten Fall ist die F-Statistik bei Nichtzutreffen der Nullhypothese nichtzentral f-verteilt
- Im multivariaten Fall ist z.B. die T^2-Statistik bei Nichtzutreffen der Nullhypothese nichtzentral f-verteilt.

WDFen von nichtzentralen f-Verteilungen



Vorbemerkungen

Einstichproben-T^2-Tests

Zweistichproben-T^2-Tests

Univariates vs. multivariates Testen

Selbstkontrollfragen

Einstichproben-T^2-Tests

Anwendungszenario

- **Eine Stichprobe experimenteller Einheiten**.
- Annahme unabhängiger und identisch nach $N(\mu, \Sigma)$ multivariat normalverteilter Daten.
- μ und Σ unbekannt.
- Quantifizieren der Unsicherheit beim inferentiellen Vergleich von μ mit μ_0 beabsichtigt.

Anwendungsbeispiele

- Gruppenanalyse von BDI und Glukokortikoid Daten
  - $\mu \neq \mu_0$ als Evidenz für eine multivariate Abweichung von einem Normwert μ_0.
- Gruppenanalyse von Kognitionstestdaten
  - $\mu \neq \mu_0$ als Evidenz für eine multivariate Abweichung von einem Normwert μ_0.

Anwendungsbeispiel

VP	IQ Test Score	Math Test Score
1	54	44
2	60	20
3	67	36
4	41	39
5	66	57
6	51	28
7	51	46
8	37	46
9	57	54
10	47	12
11	50	67
12	42	63
13	60	64
14	36	64
15	60	71

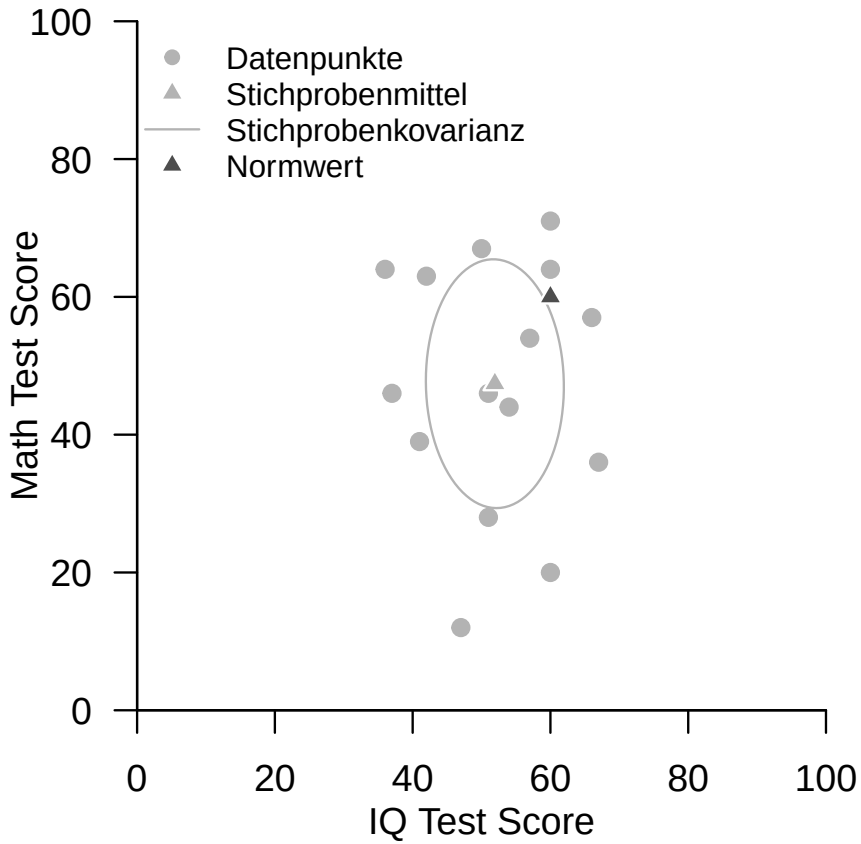
Abweichung des (IQ Test Score, Math Test Score) Erwartungswertparameters vom Normwert $\mu_0 = \begin{pmatrix} 60 \\ 60 \end{pmatrix}$?

Anwendungsbeispiel

```
# Deskriptivstatistik
library(foreign)                                                  # R Paket
library(ellipse)                                                  # R Paket
library(matlib)                                                   # R Paket
mu_0   = matrix(c(60,60), nrow = 2)                               # Normwert
fname  = file.path(getwd(), "10_Daten", "studienerfolg.sav")      # Dateinamen
D      = read.spss(fname, to.data.frame = T)                      # Dateneinlesen
Y      = rbind(D$X1[D$Gruppe == "ungenügend"],                    # Y_{i_1} IQ Test Score
               D$X2[D$Gruppe == "ungenügend"])                    # Y_{i_2} Math Test Score
n      = ncol(Y)                                                  # Anzahl Datenpunkte
j_n    = matrix(rep(1,n), nrow = n)                               # 1_n
I_n    = diag(n)                                                  # I_n
J_n    = matrix(rep(1,n^2), nrow = n)                             # 1_{nn}
Y_bar  = (1/n)*(Y %*% j_n)                                        # Stichprobenmittel (cf.(3) Wahrscheinlichkeitstheorie)
C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))               # Stichprobenkovarianzmatrix (cf. ibid.)
D      = t(Y_bar - mu_0) %*% inv(C) %*% (Y_bar - mu_0)            # Mahalanobis Distanz

# Ausgabe
cat("Y_bar =", Y_bar,
    "\nD     =", D)
```

```
Y_bar = 51.9 47.4
D     = 1.19
```

Anwendungsbeispiel



Einstichproben-T^2-Tests

Hypothesenszenarien

Einfache Nullhypothese, einfache Alternativhypothese $H_0 : \mu = \mu_0, H_1 : \mu = \mu_1$

- Theoretisch wichtiges Szenario (Neymann-Pearson Lemma)
- Praktische Relevanz eher gering

Einfache Nullhypothese, zusammengesetzte Alternativhypothese $H_0 : \mu = \mu_0, H_1 : \mu \neq \mu_0$

- Zweiseitiger Einstichproben-T-Test mit ungerichteter Hypothese
- Ungerichtete Fragestellung nach einem Unterschied

Zusammengesetzte Nullhypothese/Alternativhypothese $H_0 : \mu \leq \mu_0, H_1 : \mu > \mu_0$

- Einseitiger Einstichproben-T-Test mit gerichteter Hypothese
- Gerichtete Fragestellung nach einem positiven Unterschied

Zusammengesetzte Nullhypothese/Alternativhypothese $H_0 : \mu \geq \mu_0, H_1 : \mu < \mu_0$

- Gerichtete Fragestellung nach einem negativen Unterschied
- Qualitativ äquivalente Theorie zum umgekehrten Fall

Einstichproben-T^2-Tests

Im Folgenden näher betrachtetes Hypothesenszenario

Einfache Nullhypothese, einfache Alternativhypothese $H_0 : \mu = \mu_0, H_1 : \mu = \mu_1$

- Theoretisch wichtiges Szenario (Neymann-Pearson Lemma)
- Praktische Relevanz eher gering

Einfache Nullhypothese, zusammengesetzte Alternativhypothese $H_0 : \mu = \mu_0, H_1 : \mu \neq \mu_0$

- Zweiseitiger Einstichproben-T-Test mit ungerichteter Hypothese
- Ungerichtete Fragestellung nach einem Unterschied

Zusammengesetzte Nullhypothese/Alternativhypothese $H_0 : \mu \leq \mu_0, H_1 : \mu > \mu_0$

- Einseitiger Einstichproben-T-Test mit gerichteter Hypothese
- Gerichtete Fragestellung nach einem positiven Unterschied

Zusammengesetzte Nullhypothese/Alternativhypothese $H_0 : \mu \geq \mu_0, H_1 : \mu < \mu_0$

- Gerichtete Fragestellung nach einem negativen Unterschied
- Qualitativ äquivalente Theorie zum umgekehrten Fall

Einstichproben-T^2-Tests

Gliederung (vgl. (12) Hypothesentests)

(1) Statistisches Modell in klassischer Form

(2) Statistisches Modell in generativer Form

(3) Testhypothesen, Teststatistik, Test

(4) Analyse der Teststatistik

(5) Analyse der Testgütefunktion

(6) Testumfangkontrolle

(7) p-Werte

(8) Analyse der Powerfunktion

Zur Wiederholung des univariaten Falls, siehe (13) Einstichproben-T-Tests.

(1) Statistisches Modell in klassischer Form

$Y_1, ..., Y_n \sim N(\mu, \Sigma)$ sei die Stichproben eines m-dimensionalen Normalverteilungsmodells mit unbekanntem Erwartungswertparameter $\mu \in \mathbb{R}^m$ und unbekanntem Kovarianzmatrixparameter $\Sigma \in \mathbb{R}^{m \times m}$ p.d. Als Parameter von Interesse betrachten wir $\theta = \mu$, so dass sich der Parameterraum von Interesse zu $\Theta = \mathbb{R}^m$ ergibt.

(2) Statistisches Modell in generativer Form

Es sei

$$Y_i = \mu + \varepsilon_i \text{ mit } \varepsilon_i \sim N(0_m, \Sigma) \text{ für } i = 1, ..., n \tag{4}$$

wobei

- $Y_i, i = 1, ..., m$ beobachtbare Zufallsvektoren,
- $\mu \in \mathbb{R}^m$ den festen und identischen Erwartungswertparameter über Zufallsvektoren und
- $\varepsilon_i, i = 1, ..., n$ unabhängige normalverteilte nicht beobachtbare Zufallsvektoren

bezeichnen.

(2) Statistisches Modell in generativer Form (fortgeführt)

Die generative Form betont, dass im vorliegenden Modell beobachtete Daten durch einen systematischen deterministischen Prozess (hier $\mu \in \mathbb{R}^m$) unter dem additiven Einfluss einer Vielzahl unabhängiger und deshalb in ihrer Summe normalverteilter Störprozesse (hier in der Summe $\varepsilon_i, i = 1, ..., n$) erzeugt konzipiert werden.

Beweis

Wir halten zunächst fest, dass wenn $\xi \sim N(\alpha, S)$ ein m_1-dimensionaler normalverteilter Zufallsvektor mit Erwartungswertparameter $\alpha \in \mathbb{R}^{m1}$ und Kovarianzmatrixparameter $S \in \mathbb{R}^{m1 \times m1}$ p.d. ist und für $A \in \mathbb{R}^{m2 \times m1}$ und $b \in \mathbb{R}^{m2}$ ein m_2-dimensionaler Zufallsvektor definiert ist als

$$\zeta := A\xi + b, \tag{5}$$

dann gilt, dass

$$\zeta \sim N\left(A\alpha + b, ASA^T\right) \tag{6}$$

(vgl. Anderson (2003), Section 2.4). Aus $\varepsilon \sim N(0_m, \Sigma)$ folgt hier mit

$$Y_i = I_m \varepsilon + \mu \tag{7}$$

dann aber sofort, dass

$$Y \sim N\left(I_m 0_m + \mu, I_m \Sigma I_m^T\right) = N(\mu, \Sigma). \tag{8}$$

□

(3) Testhypothesen, Teststatistik, Test

Für ein $\mu_0 \in \mathbb{R}^m$ betrachten wir die einfache Nullhypothese und die zusammengesetzte Alternativhypothese

$$H_0 := \mu = \mu_0 \Leftrightarrow \Theta_0 := \{\mu_0\} \text{ und } H_0 := \mu \neq \mu_0 \Leftrightarrow \Theta_1 := \mathbb{R}^m \setminus \{\mu_0\} \tag{9}$$

Weiterhin betrachten wir die Einstichproben-T^2-Teststatistik

$$T^2 := n(\bar{Y} - \mu_0)^T C^{-1} (\bar{Y} - \mu_0) \tag{10}$$

wobei $\bar{Y}$ und C das Stichprobenmittel und die Stichprobenkovarianzmatrix der $Y_1, \ldots, Y_n$, respektive, bezeichnen (vgl. (3) Wahrscheinlichkeitstheorie).

- T^2 ist die mit dem Stichprobenumfang skalierte Mahalanobis Distanz von $\bar{Y}$ und μ_0 hinsichtlich C.
- $T^2 \uparrow$ für $||\bar{Y} - \mu_0|| \uparrow$, $C \downarrow$ und $n \uparrow$.

Schließlich definieren wir den kritischen Wert-basierten Test

$$\phi(Y) := 1_{\{T^2 > k\}} := \begin{cases} 1 & T^2 > k \\ 0 & T^2 \leq k \end{cases}. \tag{11}$$

wobei wie üblich 1 den Vorgang des Ablehnens von H_0 und 0 den Vorgang des Nichtablehnens von H_0 repräsentieren.

(4) Analyse der Teststatistik

Theorem (Verteilung der Einstichproben-T^2-Teststatistik)

Es seien $Y_1, \ldots, Y_n \sim N(\mu, \Sigma)$ mit $\mu \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d.,

$$\nu := \frac{n-m}{(n-1)m} \tag{12}$$

und für $\mu \in \mathbb{R}^m$ sei die Einstichproben-T^2-Teststatistik definiert als

$$T^2 := n(\bar{Y} - \mu_0)^T C^{-1} (\bar{Y} - \mu_0). \tag{13}$$

Dann gilt

$$\nu T^2 \sim f(\delta, m, n-m) \tag{14}$$

wobei $f(\delta, m, n-m)$ die nichtzentrale f-Verteilung mit Nichtzentralitätsparameter

$$\delta := n(\mu - \mu_0)^T \Sigma^{-1} (\mu - \mu_0) \tag{15}$$

sowie mit Freiheitsgradparametern m und $n-m$ bezeichnet.

Bemerkungen

- Für einen Beweis von (14) verweisen wir auf Anderson (2003) und Hotelling (1931).
- Für $\mu = \mu_0$ und damit $\delta = 0$ entspricht $f(m, n-m, \delta)$ der f-Verteilung $f(m, n-m)$

Theorem (WDF und KVF der Einstichproben-T^2-Teststatistik)

Im Einstichproben-T^2-Testszenario sei

$$\nu := \frac{n-m}{(n-1)m} \tag{16}$$

Dann ist eine WDF der Einstichproben-T^2-Teststatistik gegeben durch

$$p_{T^2} : \mathbb{R}_{\geq 0} \to \mathbb{R}, t^2 \mapsto p_{T^2}(t^2) := \nu f(\nu t^2; \delta, m, n-m) \tag{17}$$

und eine WDF der Einstichproben-T^2-Teststatistik ist gegeben durch

$$P_{T^2} : \mathbb{R}_{\geq 0} \to [0,1], t^2 \mapsto P_{T^2}(t^2) := F(\nu t^2; \delta, m, n-m) \tag{18}$$

Bemerkungen

- νT^2 hat die WDF $f(\delta, m, n-m)$, T^2 dagegen hat die WDF $\nu f(\nu t^2; \delta, m, n-m)$.
- Für $m := 1$ ist $\nu = (n-1)/(n-1) \cdot 1 = 1$ und mit der Stichprobenvarianz S^2 gilt

$$T^2 = n\frac{(\bar{Y}-\mu_0)^2}{S^2} = \left(\sqrt{n}\frac{\bar{Y}-\mu_0}{S}\right)^2 \tag{19}$$

- Das Quadrat der univariate Einstichproben-T-Teststatistik $T := \sqrt{n}\frac{\bar{Y}-\mu_0}{S}$ ist also $f(\delta, 1, n-1)$ verteilt.

Beweis

Wir halten zunächst fest, dass das Theorem zur univariate WDF Transformation bei linear-affinen Abbildungen (vgl. (7) Transformationen der Normalverteilung) besagt, dass für eine Zufallsvariable X mit WDF p_X und der Definition $Y = f(X)$ mit $f(X) := aX + b$ für $a \neq 0$ eine WDF von Y definiert ist $p_Y(y) := (1/|a|)p_X((y-b)/a)$. Im vorliegenden Fall ist $X = \nu T^2$ mit WDF $f(\delta, m, n-m)$ und $Y := T^2 = \frac{1}{\nu}\nu T^2$, also $a = 1/\nu$ und $b = 0$. Mit $\nu > 0$ ergibt sich (17) also aus

$$p_{T^2}(t^2) = \frac{1}{a} p_X\left(\frac{t^2}{a}\right) = \nu f(\nu t^2; m, n-m) \tag{20}$$

(18) folgt dann mit der Tatsache, dass WDFen bei kontinuierlichen Zufallsvariablen die Ableitungen der entsprechenden KVF sind, sowie der Kettenregel der Differentiation

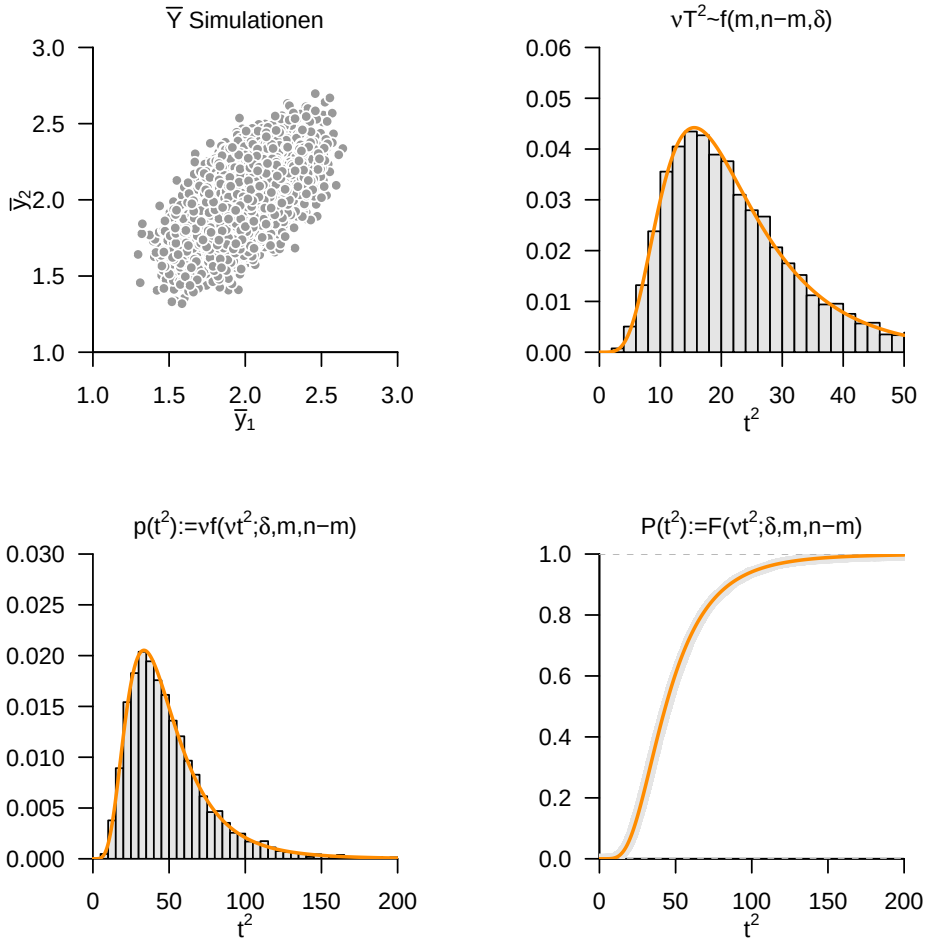
$$\begin{aligned} \frac{d}{dt^2} P_{T^2}\left(t^2\right) &= \frac{d}{dt^2}\left(F(\nu t^2; m, n-m, \delta)\right) \\ &= \frac{d}{dt^2} F(\nu t^2; m, n-m, \delta) \frac{d}{dt^2}\left(\nu t^2\right) \\ &= \nu f(\nu t^2; m, n-m, \delta) \\ &= p_{T^2}(t^2) \end{aligned} \tag{21}$$

□

(4) Analyse der Teststatistik

```
# Modellparameter
m     = 2                                                  # Dimensionalität der Zufallsvektoren/Daten
n     = 15                                                 # Anzahl der Datenpunkte
mu_0  = matrix(c(1,1) , nrow = 2)                          # H0 Hypothesenparameter
mu    = matrix(c(2,2) , nrow = 2)                          # wahrer, aber unbekannter, Erwartungswertparameter
Sigma = matrix(c(0.5,0.3, 0.3,0.5), nrow = 2, byrow = TRUE)  # wahrer, aber unbekannter, Kovarianzmatrixparameter

# Simulation
library(MASS)                                              # R Paket für multivariate Normalverteilungen
library(matlib)                                            # R Paket für Matrizenrechnung
nsim  = 1e4                                                # Anzahl Simulationen/Datensatzrealisierungen
Yb    = matrix(rep(NaN,m*nsim), nrow = 2)                  # Stichprobenmittelarray
T2    = rep(NaN,nsim)                                      # T2 Statistik Array
j_n   = matrix(rep(1,n), nrow = n)                         # 1_n
I_n   = diag(n)                                            # I_n
J_n   = matrix(rep(1,n^2), nrow = n)                       # 1_{nn}
for(s in 1:nsim){                                          # Simulationsiterationen
    Y      = t(mvrnorm(n,mu,Sigma))                        # Y_i \sim N(\mu,\Sigma), i = 1,...,n
    Y_bar  = (1/n)*(Y %*% j_n)                             # Stichprobenmittel
    C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))    # Stichprobenkovarianzmatrix
    T2[s]  = n*t(Y_bar - mu_0) %*% inv(C) %*% (Y_bar - mu_0)  # T2 Statistik
    Yb[,s] = Y_bar                                         # Stichprobenmittel für Visualisierung
}
```

(4) Analyse der Teststatistik



(5) Analyse der Testgütefunktion

Theorem (Testgütefunktion)

ϕ sei der im obigen Testszenario definiert Test. Dann ist die Testgütefunktion von ϕ gegeben durch

$$q_\phi : \mathbb{R}^m \to [0,1], \mu \mapsto q_\phi(\mu) := 1 - F(\nu k; \delta_\mu, m, n-m) \tag{22}$$

wobei $F(\cdot; \delta_\mu, m, n-m)$ die KVF der nichtzentralen f-Verteilung mit Freiheitsgradparametern$m$ und $n-m$ sowie mit Nichtzentralitätsparameter

$$\delta_\mu := n(\mu - \mu_0)^T \Sigma^{-1} (\mu - \mu_0) \tag{23}$$

bezeichnet.

Bemerkungen

- q_ϕ kann zur Bestimmung kritischer Werte für einen erwünschten Testumfang genutzt werden.
- q_ϕ kann zur Bestimmung der Testpower genutzt werden.

(5) Analyse der Testgütefunktion

Beweis

Die Testgütefunktion des betrachteten Tests im vorliegenden Testszenario ist definiert als

$$q_\phi : \mathbb{R}^m \to [0,1], \mu \mapsto q_\phi(\mu) := \mathbb{P}_\mu(\phi = 1) \tag{24}$$

Da die Wahrscheinlichkeiten für $\phi = 1$ und dafür, dass die zugehörige Teststatistik im Ablehnungsbereich des Tests liegt, gleich sind, benötigen wir also zunächst die Verteilung der Teststatistik. Wir haben oben aber bereits gesehen, dass

$$\frac{n-m}{m(n-1)} T^2 \sim f(m, n-m, \delta_\mu) \text{ mit } \delta_\mu := n(\mu - \mu_0)^T \Sigma^{-1} (\mu - \mu_0) \tag{25}$$

gilt. Der Ablehnungsbereich des betrachteten Tests ist $A :=]k, \infty[$. Also ergibt sich

$$\begin{aligned} q_\phi(\mu) &= \mathbb{P}_\mu(\phi = 1) \\ &= \mathbb{P}_\mu \left(T^2 \in]k, \infty[\right) \\ &= \mathbb{P}_\mu \left(T^2 > k \right) \\ &= 1 - \mathbb{P}_\mu \left(T^2 \leq k \right) \\ &= 1 - F(\nu k; \delta_\mu, m, m-n) \end{aligned} \tag{26}$$

(5) Analyse der Testgütefunktion

Beispiele

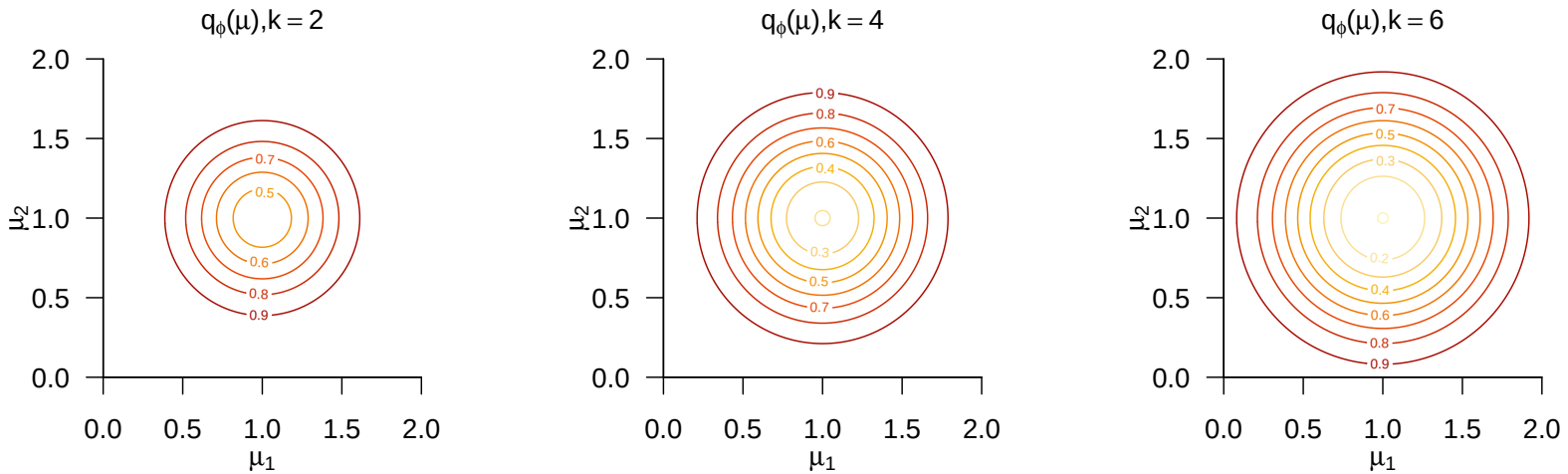
$$m := 2, n := 15, \Sigma := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mu_0 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

```
# Modellparameter
library(matlib)                                                    # R Paket
m           = 2                                                    # m
n           = 15                                                   # n
nu          = (n-m)/((n-1)*m)                                      # \nu
Sigma       = diag(m)                                              # \Sigma = I_2
iSigma      = inv(Sigma)                                           # \Sigma^{-1}

# Testparameter
mu_0        = matrix(c(1,1), nrow = 2)                             # \mu_0
k_all       = c(2,4,6)                                             # k <-> \phi
n_k         = length(k_all)                                        # Anzahl k Werte/Tests

# q_\phi(\mu) Evaluation
mu_min      = 0                                                    # \mu_i Minimum
mu_max      = 2                                                    # \mu_i Maximum
mu_res      = 1e3                                                  # \mu_i Auflösung
mu_i        = seq(mu_min,mu_max,len = mu_res)                      # mu_i
q_phi       = array(dim = c(mu_res, mu_res, length(k_all)))        # q_\phi Array
for(k in 1:n_k){
    for(i in 1:mu_res){
        for(j in 1:mu_res){
            mu           = matrix(c(mu_i[i],mu_i[j]), nrow = 2)    # \mu
            delta_mu     = n*t(mu - mu_0) %*% iSigma %*% (mu -mu_0) # \delta_\mu
            q_phi[i,j,k] = 1 - pf(nu*k_all[k], m, n-m, delta_mu)}}} # q_\phi(\mu)
```

(5) Analyse der Testgütefunktion

Beispiele

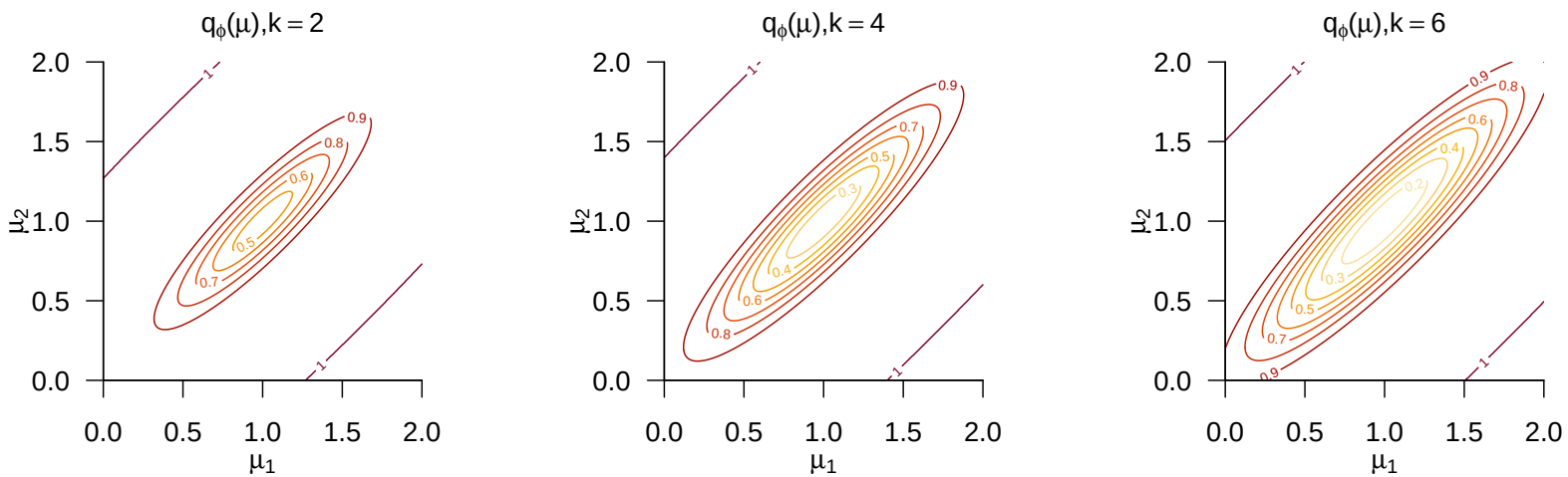
$$m := 2, n := 15, \Sigma := \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}, \mu_0 := \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}$$



(5) Analyse der Testgütefunktion

Beispiele

$$m := 2, n := 15, \Sigma := \begin{pmatrix} 1.0 & 0.9 \\ 0.9 & 1.0 \end{pmatrix}, \mu_0 := \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}$$



(6) Testumfangkontrolle

Theorem (Testumfangkontrolle)

ϕ sei der im obigen Testszenario definierte Test. Dann ist ϕ ein Level-α_0-Test mit Testumfang α_0, wenn der kritische Wert definiert ist durch

$$k_{\alpha_0} := \nu^{-1} F^{-1}\left(1-\alpha_0; m, n-m\right) \tag{27}$$

wobei $\nu := (n-m)/((n-1)m)$ und $F^{-1}(\cdot; m, n-m)$ die inverse KVF der f-Verteilung mit Freiheitsgrad-parametern m und $n-m$ ist.

Beweis

Damit der betrachtete Test ein Level-α_0-Test ist, muss bekanntlich $q_\phi(\mu) \leq \alpha_0$ für alle $\mu \in \{\mu_0\}$, also hier $q_\phi(\mu_0) \leq \alpha_0$ gelten. Weiterhin ist der Testumfang des betrachteten Tests durch $\alpha = \max_{\mu \in \{\mu_0\}} q_\phi(\mu)$, also hier durch $\alpha = q_\phi(\mu_0)$ gegeben. Wir müssen also zeigen, dass die Wahl von k_{α_0} garantiert, dass ϕ ein Level-α_0-Test mit Testumfang α_0 ist. Dazu merken wird zunächst an, dass für $\mu = \mu_0$ gilt, dass

$$q_\phi(\mu_0) = 1 - F(\nu k; m, n-m, \delta_{\mu_0}) = 1 - F(\nu k; m, n-m, 0) = 1 - F(\nu k; m, n-m) \tag{28}$$

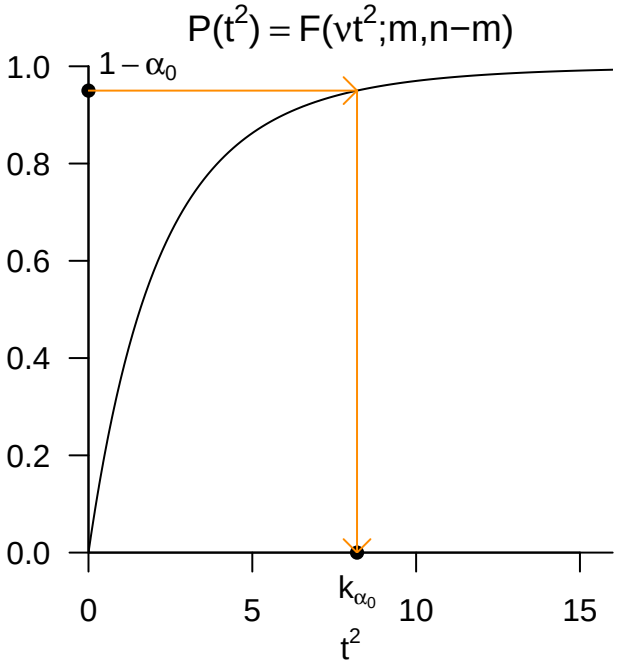
wobei$F(\nu k; \delta, m, n-m)$ und $F(\nu k; m, n-m)$ die KVF der nichtzentralen f-Verteilung mit Nichtzentralitätsparameter δ und Freiheitsgradparametern m und $n-m$ sowie der f-Verteilung mit Freiheitsgradparametern m und $n-m$, respektive, bezeichnen. Sei nun also $k := k_{\alpha_0}$. Dann gilt

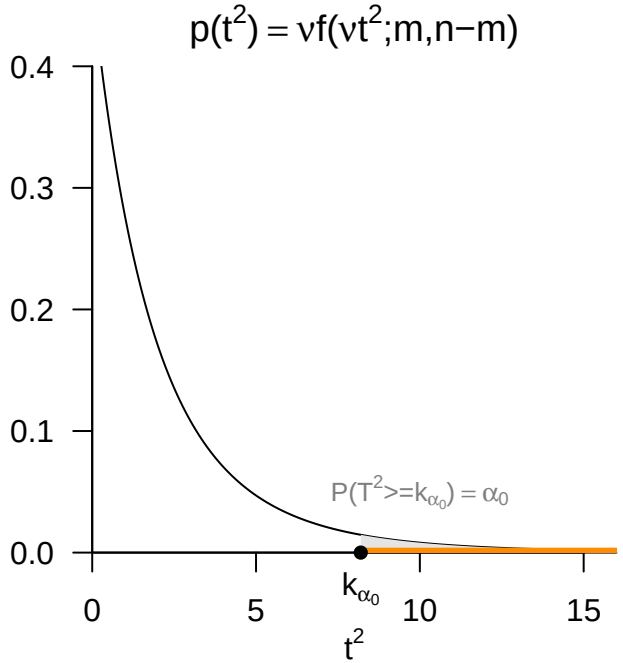
$$\begin{aligned} q_\phi(\mu_0) &= 1 - F(\nu k_{\alpha_0}; m, n-m) \\ &= 1 - F\left(\nu \nu^{-1} F^{-1}(1-\alpha_0; m, n-m); m, n-m\right) \\ &= 1 - F\left(F^{-1}(1-\alpha_0; m, n-m); m, n-m\right) \\ &= 1 - (1-\alpha_0) = \alpha_0 \end{aligned} \tag{29}$$

Es folgt also direkt, dass bei der Wahl von $k = k_{\alpha_0}$, $q_\phi(\mu_0) \leq \alpha_0$ ist der betrachtete Test somit ein Level-α_0-Test ist. Weiterhin folgt direkt, dass der Testumfang des betrachteten Tests bei der Wahl von $k = k_{\alpha_0}$ gleich α_0 ist.

(6) Testumfangkontrolle

Wahl von $k_{\alpha_0} := \nu^{-1} F^{-1}(1 - \alpha_0; m, n - m)$ mit $m = 2, n = 15$ und $\alpha_0 := 0.05$





(6) Testumfangkontrolle

```
# Modellparameter
m          = 2                                                    # Dimensionalität der Zufallsvektoren/Daten
n          = 15                                                   # Anzahl der Datenpunkte
nu         = (n-m)/(m*(n-1))                                      # Parameter
mu_0       = matrix(c(1,1) , nrow = 2)                            # H0 Hypothesenparameter
mu         = mu_0                                                 # w.a.u. Erwartungswertparameter bei Zutreffen von H0
Sigma      = matrix(c(0.5,0.3, 0.3,0.5), nrow = 2, byrow = TRUE)  # wahrer, aber unbekannter, Kovarianzmatrixparameter

# Testparameter
alpha_0    = 0.05                                                 # Signifikanzlevel
k_alpha_0 = (1/nu)*qf(1-alpha_0, m,n-m)                           # kritischer Wert

# Simulation der Testumfangkontrolle
library(MASS)                                                     # R Paket für multivariate Normalverteilungen
library(matlib)                                                   # R Paket für Matrizenrechnung
nsim   = 1e5                                                      # Testentscheidungsarray
phi    = rep(NaN,nsim)                                            # Testentscheidungsarray
j_n    = matrix(rep(1,n), nrow = n)                               # 1_n
I_n    = diag(n)                                                  # I_n
J_n    = matrix(rep(1,n^2), nrow = n)                             # 1_{nn}
for(s in 1:nsim){                                                 # Simulationsiterationen
    Y      = t(mvrnorm(n,mu,Sigma))                               # Y_i \sim N(\mu,\Sigma), i = 1,...,n
    Y_bar  = (1/n)*(Y %*% j_n)                                    # Stichprobenmittel
    C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))           # Stichprobenkovarianzmatrix
    T2     = n*t(Y_bar - mu_0) %*% inv(C) %*% (Y_bar - mu_0)      # T^2 Statistik
    if(T2 > k_alpha_0){                                           # Test 1_{T^2 >= k_alpha_0}
        phi[s] = 1                                                # Ablehnen von H_0
    } else {
        phi[s] = 0                                                # Nicht Ablehnen von H_0
    }
}
cat("\nKritischer Wert              = ", k_alpha_0,
    "\nGeschätzter Testumfang alpha = ", mean(phi))              # Ausgabe
```

```
Kritischer Wert              =  8.2
Geschätzter Testumfang alpha =  0.0494
```

(6) Testumfangkontrolle

Praktisches Vorgehen

- Man nimmt an, dass ein vorliegender Datensatz $y_1, ..., y_n$ eine Realisation von $Y_1, ..., Y_n \sim N(\mu, \Sigma)$ mit unbekannten Parametern $\mu \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. ist.
- Man möchte entscheiden ob für ein $\mu_0 \in \mathbb{R}^m$ eher $H_0 : \mu = \mu_0$ oder $H_1 : \mu \neq \mu_0$ zutrifft.
- Man wählt ein Signifikanzlevel α_0 und bestimmt den zugehörigen Freiheitsgradparameter-abhängigen kritischen Wert k_{α_0}. Zum Beispiel gilt bei Wahl von $\alpha_0 := 0.05, m = 2$ und $n = 15$, also Freiheitsgradparametern 2 und 13, dass $k_{0.05} = \nu^{-1} F^{-1}(1-0.05; 2, 13) \approx 8.2$.
- Anhand von $m, n, \mu_0, \bar{Y}$ und C berechnet man die Realisierung der Einstichproben-T^2-Teststatistik
$$T^2 := n(\bar{Y} - \mu_0)^T C^{-1} (\bar{Y} - \mu_0) \tag{30}$$
- Wenn T^2 größer als k_{α_0} ist, lehnt man die Nullhypothese ab, andernfalls nicht.
- Die oben entwickelte Theorie garantiert dann, dass man in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.

(7) p-Werte

Bestimmung des p-Wertes

- Per Definition ist der p-Wert das kleinste Signifikanzlevel α_0, bei welchem man die Nullhypothese basierend auf einem vorliegenden Wert der Teststatistik ablehnen würde.
- Bei $T^2 = t^2$ würde H_0 für jedes α_0 mit $t^2 \geq \nu^{-1} F^{-1}(1-\alpha_0; m, n-m)$ abgelehnt werden. Für diese α_0 gilt, wie unten gezeigt

$$\alpha_0 \geq \mathbb{P}\left(T^2 \geq t^2\right) \tag{31}$$

- Das kleinste $\alpha_0 \in [0,1]$ mit $\alpha_0 \geq \mathbb{P}\left(T^2 \geq t^2\right)$ ist dann $\alpha_0 = \mathbb{P}(T^2 \geq t^2)$, also folgt

$$\text{p-Wert } = \mathbb{P}\left(T^2 \geq t^2\right) = 1 - F(\nu t^2; m, n-m) \tag{32}$$

- Zum Beispiel ergibt sich bei

- $m = 2$ und $n = 15$ der p-Wert für $t^2 = 7.00$ zu 0.071
- $m = 2$ und $n = 15$ der p-Wert für $t^2 = 9.00$ zu 0.040
- $m = 2$ und $n = 99$ der p-Wert für $t^2 = 7.00$ zu 0.035
- $m = 4$ und $n = 15$ der p-Wert für $t^2 = 7.00$ zu 0.304

(7) p-Werte

Bestimmung des p-Wertes

- Es bleibt zu zeigen, dass gilt

$$\begin{aligned} t^2 &\geq \nu^{-1} F^{-1}(1-\alpha_0; m, n-m) \\ \Leftrightarrow \nu t^2 &\geq F^{-1}(1-\alpha_0; m, n-m) \\ \Leftrightarrow \alpha_0 &\geq \mathbb{P}\left(T^2 \geq t^2\right) \end{aligned} \tag{33}$$

- Dies aber folgt aus

$$\begin{aligned} t^2 &\geq \nu^{-1} F^{-1}(1-\alpha_0; m, n-m) \\ \nu t^2 &\geq F^{-1}(1-\alpha_0; m, n-m) \\ F(\nu t^2; m, n-m) &\geq F\left(F^{-1}(1-\alpha_0; m, n-m); m, n-m\right) \\ F(\nu t^2; m, n-m) &\geq 1-\alpha_0 \\ \mathbb{P}\left(T^2 \leq t^2\right) &\geq 1-\alpha_0 \\ \alpha_0 &\geq 1-\mathbb{P}\left(T^2 \leq t^2\right) \end{aligned} \tag{34}$$

Einstichproben-T^2-Tests

Anwendungsbeispiel

```
# R Pakete
library(foreign)                                                    # Dateneinlesen
library(matlib)                                                     # Matrixalgebra

# Datenpräprozessierung
fname     = file.path(getwd(), "10_Daten", "studienerfolg.sav")     # Dateinamen
D         = read.spss(fname, to.data.frame = T)                     # Dateneinlesen
Y         = rbind(D$X1[D$Gruppe == "ungenügend"],                   # Y_{i_1} IQ Test Score
                  D$X2[D$Gruppe == "ungenügend"])                   # Y_{i_2} Math Test Score

# Testparameter
m         = nrow(Y)                                                 # Dimensionalität der Zufallsvektoren/Daten
n         = ncol(Y)                                                 # Anzahl der Datenpunkte
nu        = (n-m)/(m*(n-1))                                         # Parameter
mu_0      = matrix(c(60,60) , nrow = 2)                             # H0 Hypothesenparameter ("Normwert")
alpha_0   = 0.05                                                    # Signifikanzlevel
k_alpha_0 = (1/nu)*qf(1-alpha_0,m,n-m)                              # kritischer Wert

# Testevaluation
j_n       = matrix(rep(1,n), nrow = n)                              # 1_n
I_n       = diag(n)                                                 # I_n
J_n       = matrix(rep(1,n^2), nrow = n)                            # 1_{nn}
Y_bar     = (1/n)*(Y %*% j_n)                                       # Stichprobenmittel
C         = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))              # Stichprobenkovarianzmatrix
T2        = n*t(Y_bar - mu_0) %*% inv(C) %*% (Y_bar - mu_0)         # T^2 Statistik
if(T2 > k_alpha_0){                                                 # Test 1_{T^2 >= k_alpha_0}
    phi = 1                                                         # Ablehnen von H_0
} else {
    phi = 0                                                         # Nicht Ablehnen von H_0
}
p         = 1 - pf(nu*T2,m,n-m)                                     # p-Wert
```

Einstichproben-T^2-Tests

Anwendungsbeispiel

```
# Ausgabe
cat("Y_bar   = ", Y_bar,
    "\nC       = ", C,
    "\nT^2     = ", T2,
    "\nalpha_0 = ", alpha_0,
    "\nk       = ", k_alpha_0,
    "\nphi     = ", phi,
    "\np       = ", p)
```

```
Y_bar   =  51.9 47.4
C       =  98.2 -4.11 -4.11 319
T^2     =  17.8
alpha_0 =  0.05
k       =  8.2
phi     =  1
p       =  0.00482
```

Anwendungsbeispiel mit `MVTests::OneSampleHT2()`

```
library(MVTests)                                # R Pakete
Y     = D[1:15,2:3]                             # Dataframe von Interesse
phi   = OneSampleHT2(Y, mu_0, alpha_0)          # Einstichproben-T^2-Test

# Ausgabe
cat("Y_bar   = "  , phi$Descriptive[2,],
    "\nT^2     = ", phi$HT2,
    "\nalpha_0 = ", phi$alpha,
    "\nk       = " ,phi$F,
    "\np       = ", phi$p.value)
```

```
Y_bar   =  51.9 47.4
T^2     =  17.8
alpha_0 =  0.05
k       =  8.27
p       =  0.00482
```

(8) Analyse der Powerfunktion

Wir betrachten die Testgütefunktion

$$q_\phi : \mathbb{R}^m \to [0,1], \mu \mapsto q_\phi(\mu) := 1 - F(\nu k; \delta_\mu, m, n-m) \tag{35}$$

bei kontrolliertem Testumfang, also für

$$k_{\alpha_0} := \nu^{-1} F^{-1}\left(1 - \alpha_0; m, n-m\right) \tag{36}$$

mit festem α_0 als Funktion des Nichtzentralitätsparameters und des Stichprobenumfangs. Namentlich hängt hier k_{α_0} auch von n ab.

Es ergibt sich die bivariate reellwertige Funktion

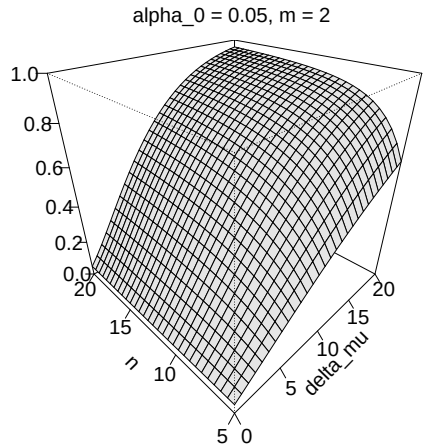
$$\pi : \mathbb{R} \times \mathbb{N} \to [0,1], (\delta_\mu, n) \mapsto \pi(\delta_\mu, n) := 1 - F(\nu k_{\alpha_0}; \delta_\mu, m, n-m) \tag{37}$$

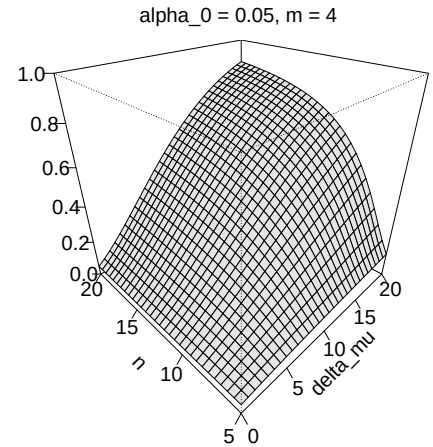
Bei festgelegtem α_0 hängt die Powerfunktion des Einstichproben-T^2-Tests also vom unbekannten Wert δ_μ, von der Datendimensionalität m und von der Stichprobengröße n ab. Wir evaluieren und visualisieren diese Abhängigkeiten untenstehend.

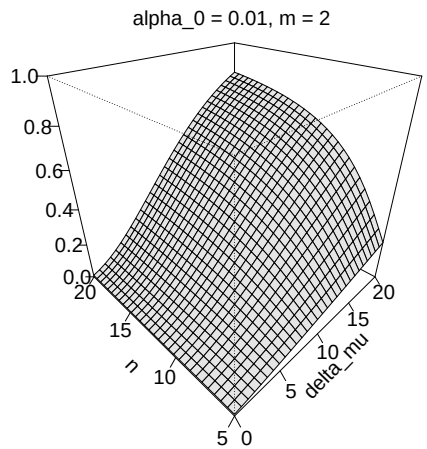
(8) Analyse der Powerfunktion

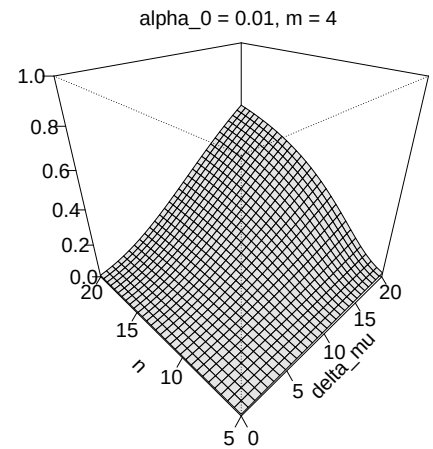
```
# Szenariospezifikationen
a_0_all   = c(0.05,0.01)                                          # \alpha_0 raum
d_mu_min  = 0                                                     # \delta_\mu Minimum
d_mu_max  = 20                                                    # \delta_\mu Maximum
d_mu_res  = 30                                                    # \delta_\mu Auflösung
d_mu_all  = seq(d_mu_min, d_mu_max, len = d_mu_res)               # \delta_\mu d Raum
n_min     = 5                                                     # n Minimum
n_max     = 20                                                    # n Maximum
n_res     = 30                                                    # n Auflösung
n_all     = seq(n_min,n_max, len = n_res)                         # n Raum
m_all     = c(2,4)                                                # m Raum

# Evaluation der Powerfunktion
pi        = array(dim = c(d_mu_res, n_res, 2,2))                  # Powerfunktionsarray
for (a in 1:length(a_0_all)){
    for (l in 1:length(m_all)){                                   # m Iterationen
        for(i in 1:length(d_mu_all)){                             # \delta_\mu Iterationen
            for(j in 1:length(n_all)){                            # n Iterationen
                m           = m_all[l]                            # Datendimensionalität
                n           = n_all[j]                            # Stichprobenumfang
                d_mu        = d_mu_all[i]                         # wahrer, aber unbekannter, Parameter
                nu          = (n-m)/(m*(n-1))                     # Parameter
                alpha_0     = a_0_all[a]                          # Signifikanzlevel
                k_alpha_0   = (1/nu)*qf(1-alpha_0,m,n-m)          # kritischer Wert
                pi[i,j,l,a] = 1 - pf(nu*k_alpha_0, m, n-m, d_mu)}}}}  # Powerfunktionswert
```

(8) Analyse der Powerfunktion









(8) Analyse der Powerfunktion

Praktisches Vorgehen

Mit größerem n steigt die Powerfunktion des Tests an

- Ein großer Stichprobenumfang ist besser als ein kleiner Stichprobenumfang.
- Kosten für die Erhöhung des Stichprobenumfangs werden aber nicht berücksichtigt.

$\Rightarrow$ Die Theorie statistischer Hypothesentests ist nicht besonders lebensnah.

Die Powerfunktion hängt vom wahren, aber unbekannten, Parameterwert $\delta_\mu = n(\mu - \mu_0)^T \Sigma^{-1} (\mu - \mu_0)$ ab.

$\Rightarrow$ Wenn man δ_μ schon kennen würde, würde man den Test nicht durchführen.

Generell wird folgendes Vorgehen favorisiert

- Man legt das Signifikanzlevel α_0 fest und evaluiert die Powerfunktion.
- Man wählt einen Mindestparameterwert δ_μ^*, den man mit $\pi(\delta_\mu, n) = \beta$ detektieren möchte.
- Ein konventioneller Wert ist $\beta = 0.8$.
- Man liest die für $\pi(\delta_\mu = \delta_\mu^*, n) = \beta$ nötige Stichprobengröße n ab.

(7) Analyse der Powerfunktion

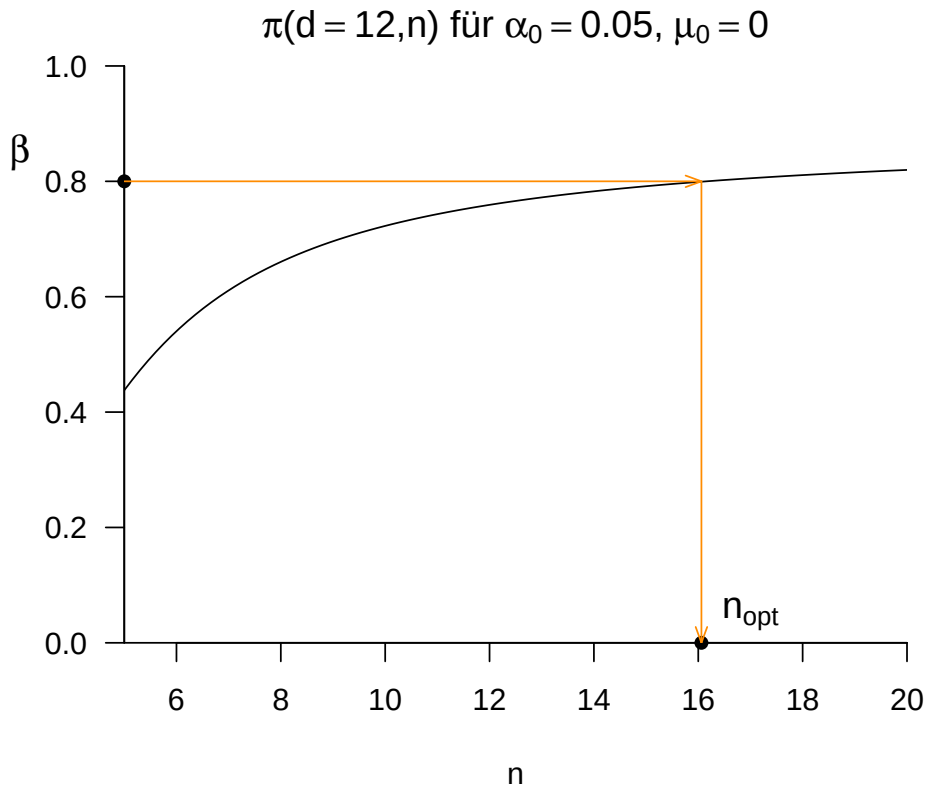
Praktisches Vorgehen

```
# Szenariospezifikation
n_min      = 5                                          # n Minimum
n_max      = 20                                         # n Maximum
n_res      = 1e2                                        # n Auflösung
n          = seq(n_min,n_max, len = n_res)              # n Raum
alpha_0    = 0.05                                       # Signifikanzlevel

# Poweranalyse
m          = 2                                          # Datendimensionalität
d_mu_fix   = 12                                         # fester Nichtzentralitätsparameter
nu         = (n-m)/(m*(n-1))                            # Parameter
k_alpha_0  = (1/nu)*qf(1-alpha_0,m,n-m)                 # kritischer Wert
pi_n       = 1 - pf(nu*k_alpha_0, m, n-m, d_mu_fix)     # Powerfunktionswert
beta       = 0.8                                        # gewünschter Powerfunktionswert
i          = 1                                          # Indexinitialisierung
n_min      = NaN                                        # minimales n Initialisierung
while(pi_n[i] < beta){                                  # Solange \pi(\delta_\mu*,n) < \beta
    n_min = n[i]                                        # Aufnahme des minimal nötigen ns
    i     = i + 1                                       # und Erhöhung des Indexes
}
cat("Minimal nötiges n =", ceiling(n_min))              # Ausgabe
```

```
Minimal nötiges n = 17
```

(7) Analyse der Powerfunktion

Praktisches Vorgehen



Vorbemerkungen

Einstichproben-T^2-Tests

Zweistichproben-T^2-Tests

Univariates vs. multivariates Testen

Selbstkontrollfragen

Zweistichproben-T^2-Tests **bei unabhängigen Stichproben**

Anwendungsszenario

- **Zwei Stichproben** experimenteller Einheiten.
- Annahme unabhängiger und identisch nach $N(\mu_1, \Sigma_1)$ und $N(\mu_2, \Sigma_2)$ verteilter Daten.
- $\mu_1, \Sigma_1, \mu_2, \Sigma_2$ unbekannt.
- Quantifizieren der Unsicherheit beim inferentiellen Vergleich von μ_1 und μ_2 beabsichtigt.

Anwendungsbeispiele

- Gruppenvergleich von BDI und Glukokortikoid Daten
  - $\mu_1 \neq \mu_2$ als Evidenz für multivariate Gruppenunterschiede
- Gruppenvergleich von Testdaten bei erfolgreichen vs. nicht-erfolgreichem Studienabschluss
  - $\mu_1 \neq \mu_2$ als Evidenz für multivariate Gruppenunterschiede

Mögliche Modellszenarien

- Annahme identischer Kovarianzmatrixparameter
- Annahme eines bekannten Varianzverhältnisses
- Keine Annahmen zu Varianzen

Mögliche Hypothesenszenarien

- $H_0 : \mu_1 = \mu_2$ und $H_1 : \mu_1 \neq \mu_2$
- $H_0 : \mu_1 \leq \mu_2$ und $H_1 : \mu_1 > \mu_2$
- $H_0 : \mu_1 \geq \mu_2$ und $H_1 : \mu_1 < \mu_2$

Zweistichproben-T^2-Tests

Hier betrachtetes Modellszenario

- Annahme identischer Kovarianzmatrixparameter
- Annahme eines bekannten Varianzverhältnisses
- Keine Annahmen zu Varianzen

Hier betrachtetes Hypothesenszenario

- $H_0 : \mu_1 = \mu_2$ und $H_1 : \mu_1 \neq \mu_2$
- $H_0 : \mu_1 \leq \mu_2$ und $H_1 : \mu_1 > \mu_2$
- $H_0 : \mu_1 \geq \mu_2$ und $H_1 : \mu_1 < \mu_2$

Anwendungsbeispiel

Gruppe	VP	IQ Test Score	Math Test Score
1	1	54	44
1	2	60	20
1	3	67	36
1	4	41	39
1	5	66	57
1	6	51	28
1	7	51	46
1	8	37	46
1	9	57	54
1	10	47	12
1	11	50	67
1	12	42	63
1	13	60	64
1	14	36	64
1	15	60	71
2	1	71	41
2	2	65	28
2	3	67	76
2	4	68	54
2	5	75	33
2	6	71	82
2	7	68	64
2	8	63	72
2	9	48	54
2	10	53	86
2	11	62	71
2	12	69	25
2	13	67	72
2	14	74	92
2	15	76	75

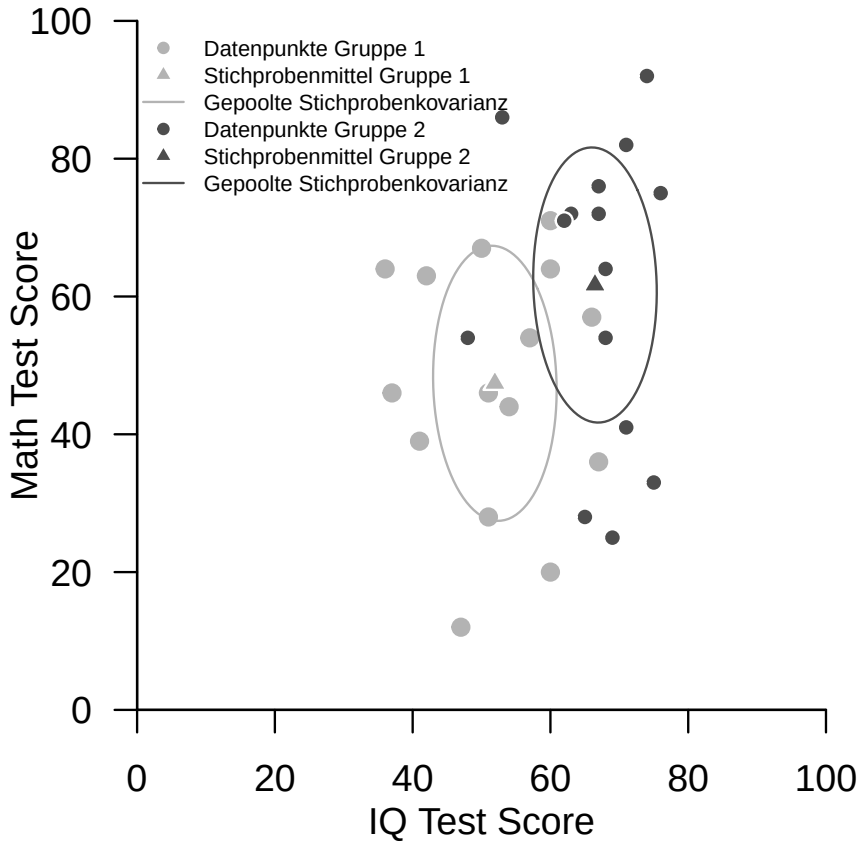
Unterschiede zwischen den (IQ Test Score, Math Test Score) Erwartungswertparametern zwischen Gruppen?

Anwendungsbeispiel

```
# Deskriptivstatistik
library(foreign)                                                    # R Paket
library(ellipse)                                                    # R Paket
library(matlib)                                                     # R Paket
fname    = file.path(getwd(), "10_Daten", "studienerfolg.sav")      # Dateiname
D        = read.spss(fname, to.data.frame = T)                      # Dateneinlesen
m        = 2                                                        # Datendimensionalität
n        = 15                                                       # Datenpunkte pro Gruppe
n_1      = n                                                        # Datenpunkte Gruppe 1
n_2      = n                                                        # Datenpunkte Gruppe 2
j_n      = matrix(rep(1,n), nrow = n)                               # 1_n
I_n      = diag(n)                                                  # I_n
J_n      = matrix(rep(1,n^2), nrow = n)                             # 1_{nn}
Y_1      = t(as.matrix(D[D$Gruppe == "ungenügend", 2:3]))           # Daten Gruppe 1
Y_2      = t(as.matrix(D[D$Gruppe == "gut", 2:3]))                  # Daten Gruppe 2
Y_1_bar  = (1/n)*(Y_1 %*% j_n)                                      # Stichprobenmittel
Y_2_bar  = (1/n)*(Y_2 %*% j_n)                                      # Stichprobenmittel
C_1      = (1/(n_1-1))*(Y_1 %*% (I_n-(1/n)*J_n) %*% t(Y_1))         # Stichprobenkovarianzmatrix
C_2      = (1/(n_2-1))*(Y_2 %*% (I_n-(1/n)*J_n) %*% t(Y_2))         # Stichprobenkovarianzmatrix
C        = ((n_1-1)*C_1+(n_2-1)*C_2)/(n_1+n_2-2)                    # Gepoolte Stichprobenkovarianzmatrix
D        = t(Y_1_bar - Y_2_bar) %*% inv(C) %*% (Y_1_bar - Y_2_bar)  # Mahalanobis Distanz

# Ausgabe
cat("Y_1_bar  =", Y_1_bar,
    "\nY_2_bar  =", Y_2_bar,
    "\nD        =", D)
```

```
Y_1_bar  = 51.9 47.4
Y_2_bar  = 66.5 61.7
D        = 3.33
```

Anwendungsbeispiel



Zweistichproben-T^2-Tests

Gliederung (vgl. (12) Hypothesentests)

(1) Statistisches Modell in klassischer Form

(2) Statistisches Modell in generativer Form

(3) Testhypothesen, Teststatistik, Test

(4) Analyse der Teststatistik

(5) Analyse der Testgütefunktion

(6) Testumfangkontrolle

(7) p-Werte

Zur Wiederholung des univariaten Falls, siehe (14) Zweistichproben-T-Tests

Zweistichproben-T^2-Tests

(1) Statistisches Modell in klassischer Form

$Y_{11}, \ldots, Y_{1n_1} \sim N(\mu_1, \Sigma)$ sei eine Stichprobe eines multivariaten Normalverteilungsmodells mit unbekanntem Erwartungswertparameter $\mu_1 \in \mathbb{R}^m$ und unbekanntem Kovarianzmatrixparameter $\Sigma \in \mathbb{R}^{m \times m}$ p.d. $Y_{21}, \ldots, Y_{2n_2} \sim N(\mu_2, \Sigma)$ sei eine weitere Stichprobe eines multivariaten Normalverteilungsmodells mit unbekanntem Erwartungswertparameter $\mu_2 \in \mathbb{R}^m$ und unbekanntem Kovarianzmatrixparameter $\Sigma \in \mathbb{R}^{m \times m}$ p.d. Die Kovarianzmatrixparameter beider Stichproben werden also als identisch vorausgesetzt. Der Parameter von Interesse ist (μ_1, μ_2), der Parameterraum des Modells ist $\Theta := \mathbb{R}^m \times \mathbb{R}^m$.

(2) Statistisches Modell in generativer Form

Es sei

$$Y_{ij} = \mu_i + \varepsilon_{ij} \text{ mit } \varepsilon_{ij} \sim N(0, \Sigma) \text{ für } i = 1, 2 \text{ und } j = 1, \ldots, n_i \tag{38}$$

wobei

- i die Stichproben indiziert,
- j die experimentellen Einheiten indiziert,
- n_i die Stichprobengrößen sind,
- Y_{ij} beobachtbare Zufallsvariablen sind,
- $\mu_i \in \mathbb{R}^m$ feste Erwartungswertparameter der Stichprobenvariablen sind
- $\Sigma \in \mathbb{R}^{m \times m}$ p.d. der identische Kovarianzmatrixparameter über Stichproben, und
- ε_{ij} unabhängige multivariat-normalverteilte nicht-beobachtbare Zufallsvariablen sind.

Der Zusammenhang zwischen klassischer und generativer Modellform ergibt sich analog zum Einstichprobenfall.

(3) Testhypothesen, Teststatistik, Test

Wir betrachten die einfache Nullhypothese

$$H_0 : \mu_1 = \mu_2 \Leftrightarrow \Theta_0 := \{(\mu_1, \mu_2) \in \mathbb{R}^m \times \mathbb{R}^m | \mu_1 = \mu_2\} \tag{39}$$

und die zusammengesetzte Alternativhypothese

$$H_1 : \mu_1 \neq \mu_2 \Leftrightarrow \Theta_1 := \{(\mu_1, \mu_2) \in \mathbb{R}^m \times \mathbb{R}^m | \mu_1 \neq \mu_2\} \tag{40}$$

Weiterhin betrachten die *Zweistichproben-T^2-Teststatistik*

$$T^2 := \frac{n_1 n_2}{n_1 + n_2} \left(\bar{Y}_1 - \bar{Y}_2\right)^T C^{-1} \left(\bar{Y}_1 - \bar{Y}_2\right) \tag{41}$$

wobei für $i = 1, 2$ und respektiven Stichprobenkovarianzmatrizen C_1 und C_2

$$\bar{Y}_i := \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} \text{ und } C := \frac{(n_1 - 1)C_1 + (n_2 - 1)C_2}{n_1 + n_2 - 2} \tag{42}$$

die Stichprobenmittel und die *gepoolte Stichprobenkovarianzmatrix*, respektive, bezeichnen.

- T^2 ist die mit den Stichprobenumfängen skalierte Mahalanobis Distanz von $\bar{Y}_1$ und $\bar{Y}_2$ hinsichtlich C.
- Größere T^2 Werte ergeben sich für größere Abstände, geringere Kovarianzen und größere Stichprobenumfänge.

Schließlich definieren wir den kritischen Wert-basierten Test

$$\phi(X) := 1_{\{T^2 \geq k\}}. \tag{43}$$

(4) Analyse der Teststatistik

Theorem (Verteilung der Zweistichproben-T^2 Statistik)

Für $i = 1, 2$ seien $Y_{i1}, \ldots, Y_{in_i} \sim N(\mu_i, \Sigma)$ mit $\mu_i \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d.,,

$$\nu := \frac{n_1 + n_2 - m - 1}{m(n_1 + n_2 - 2)} \tag{44}$$

und die Zweistichproben-T^2-Teststatistik sei definiert als

$$T^2 := \frac{n_1 n_2}{n_1 + n_2} (\bar{Y}_1 - \bar{Y}_2)^T C^{-1} (\bar{Y}_1 - \bar{Y}_2) \tag{45}$$

mit den Stichprobenmitteln $\bar{Y}_i$ und der gepoolten Stichprobenkovarianzmatrix C. Dann gilt

$$\nu T^2 \sim f(\delta, m, n_1 + n_2 - m - 1), \tag{46}$$

wobei $f(\delta, m, n_1 + n_2 - m - 1)$ die nichtzentrale f-Verteilung mit Nichtzentralitätsparameter

$$\delta := \frac{n_1 n_2}{n_1 + n_2} (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \tag{47}$$

sowie mit Freiheitsgradparametern m und $n_1 + n_2 - m - 1$ bezeichnet.

Bemerkungen

- Für einen Beweis verweisen wir auf Anderson (2003).
- Für $\mu_1 = \mu_2$ und damit $\delta = 0$ entspricht $f(\delta, m, n_1 + n_2 - m - 1)$ der f-Verteilung $f(m, n_1 + n_2 - m - 1)$.

(4) Analyse der Teststatistik

Theorem (WDF und KVF der Zweistichproben-T^2 Statistik)

Im Zweistichproben-T^2-Testszenario sei

$$\nu := \frac{n_1 + n_2 - m - 1}{m(n_1 + n_2 - 2)} \tag{48}$$

Dann ist eine WDF der Zweistichproben-T^2 Statistik gegeben durch

$$p_{T^2} : \mathbb{R}_{\geq 0} \to \mathbb{R}, t^2 \mapsto p_{T^2}(t^2) := \nu f(\nu t^2; \delta, m, n_1 + n_2 - m - 1) \tag{49}$$

und eine KVF der Zweistichproben-T^2 Statistik ist gegeben durch

$$P_{T^2} : \mathbb{R}_{\geq 0} \to \mathbb{R}, t^2 \mapsto P_{T^2}(t^2) := Ft(\nu t^2; \delta, m, n_1 + n_2 - m - 1) \tag{50}$$

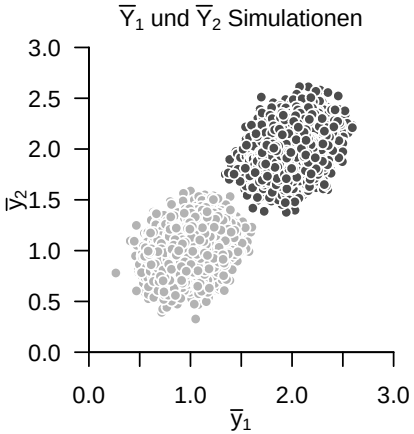
Bemerkung

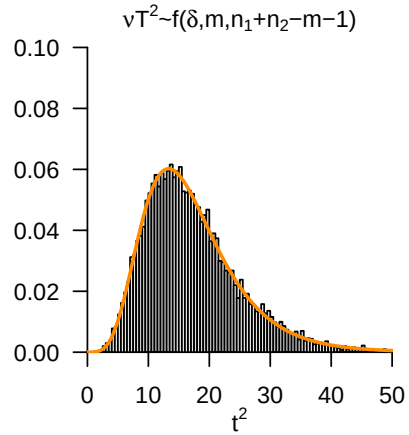
- Der Beweis erfolgt in Analogie zum Einstichproben-T^2-Testszenario.

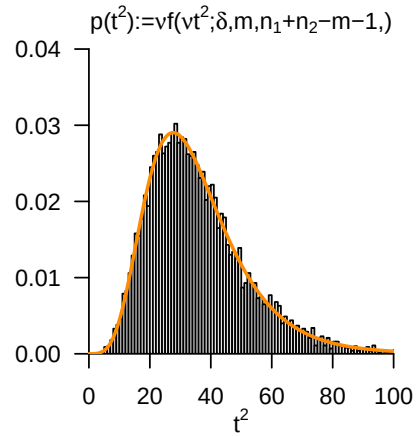
(4) Analyse der Teststatistik

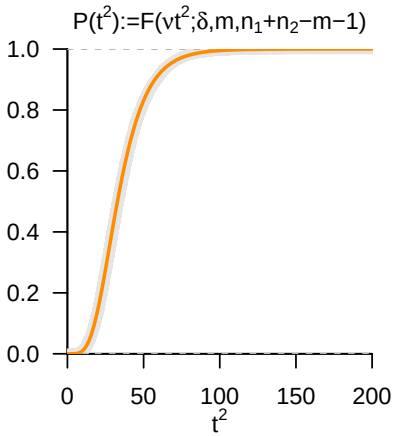
```
# Modellparameter
m     = 2                                                    # Dimensionalität der Zufallsvektoren/Daten
n     = 15                                                   # Anzahl Datenpunkte pro Stichprobe
n_1   = n                                                    # Anzahl Datenpunkte Stichprobe 1
n_2   = n                                                    # Anzahl Datenpunkte Stichprobe 1
mu_1  = matrix(c(1,1) , nrow = 2)                            # wahrer, aber unbekannter, Erwartungswertparameter
mu_2  = matrix(c(2,2) , nrow = 2)                            # wahrer, aber unbekannter, Erwartungswertparameter
Sigma = matrix(c(0.4,0.1, 0.1,0.4), nrow = 2, byrow = TRUE)  # wahrer, aber unbekannter, Kovarianzmatrixparameter

# Simulation
library(MASS)                                                # R Paket für multivariate Normalverteilungen
library(matlib)                                              # R Paket für Matrizenrechnung
nsim  = 1e4                                                  # Anzahl Simulationen/Datensatzrealisierungen
Y1b   = matrix(rep(NaN,m*nsim), nrow = 2)                    # Stichprobenmittelarray
Y2b   = matrix(rep(NaN,m*nsim), nrow = 2)                    # Stichprobenmittelarray
T2    = rep(NaN,nsim)                                        # T2 Statistik Array
j_n   = matrix(rep(1,n), nrow = n)                           # 1_n
I_n   = diag(n)                                              # I_n
J_n   = matrix(rep(1,n^2), nrow = n)                         # 1_{nn}
for(s in 1:nsim){                                            # Simulationsiterationen
    Y_1      = t(mvrnorm(n,mu_1,Sigma))                      # Y_1j \sim N(\mu_1,\Sigma), j = 1,...,n_1
    Y_2      = t(mvrnorm(n,mu_2,Sigma))                      # Y_2j \sim N(\mu_2,\Sigma), j = 1,...,n_2
    Y_1_bar  = (1/n)*(Y_1 %*% j_n)                           # Stichprobenmittel
    Y_2_bar  = (1/n)*(Y_2 %*% j_n)                           # Stichprobenmittel
    C_1      = (1/(n_1-1))*(Y_1 %*% (I_n-(1/n)*J_n) %*% t(Y_1))  # Stichprobenkovarianzmatrix
    C_2      = (1/(n_2-1))*(Y_2 %*% (I_n-(1/n)*J_n) %*% t(Y_2))  # Stichprobenkovarianzmatrix

    # Gepoolte Stichprobenkovarianzmatrix und T2 Statistik
    C        = ((n_1-1)*C_1+(n_2-1)*C_2)/(n_1+n_2-2)
    T2[s]    = (((n_1*n_2)/(n_1+n_2))*t(Y_1_bar-Y_2_bar) %*% inv(C) %*% (Y_1_bar-Y_2_bar))

    # Stichprobenmittel für Visualisierung
    Y1b[,s]  = Y_1_bar
    Y2b[,s]  = Y_2_bar
}
```

(4) Analyse der Teststatistik









(5) Analyse der Testgütefunktion

Theorem (Testgütefunktion)

ϕ sei der im obigen Testszenario definiert Test. Dann ist die Testgütefunktion von ϕ gegeben durch

$$q_\phi : \mathbb{R}^m \times \mathbb{R}^m \to [0,1], \mu \mapsto q_\phi(\mu_1, \mu_2) := 1 - F(\nu k; \delta_{\mu_{1,2}}, m, n_1 + n_2 - m - 1), \tag{51}$$

wobei

$$\nu := \frac{n_1 + n_2 - m - 1}{m(n_1 + n_2 - 2)}, \quad \delta_{\mu_{1,2}} := \frac{n_1 n_2}{n_1 + n_2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) \tag{52}$$

und $F(\cdot; \delta_{\mu_{1,2}}, m, n_1 + n_2 - m - 1)$ die KVF der nichtzentralen f-Verteilung mit Freiheitsgradparametern m und $n_1 + n_2 - m - 1$ sowie mit Nichtzentralitätsparameter $\delta_{\mu_{1,2}}$ bezeichnet.

Bemerkungen

- q_ϕ kann zur Bestimmung kritischer Werte für einen erwünschten Testumfang genutzt werden.
- q_ϕ kann zur Bestimmung der Testpower genutzt werden.

(5) Analyse der Testgütefunktion

Beweis

Die Testgütefunktion des betrachteten Tests im vorliegenden Testszenario ist definiert als

$$q_\phi : \mathbb{R}^m \times \mathbb{R}^m \to [0,1], (\mu_1, \mu_2) \mapsto q_\phi(\mu_1, \mu_2) := \mathbb{P}_{\mu_1,\mu_2}(\phi = 1) \tag{53}$$

Da die Wahrscheinlichkeiten für $\phi = 1$ und dafür, dass die zugehörige Teststatistik im Ablehnungsbereich des Tests liegt, gleich sind, benötigen wir also zunächst die Verteilung der Teststatistik. Wir haben oben aber bereits gesehen, dass für

$$\nu := \frac{n_1 + n_2 - m - 1}{(m(n_1 + n_2 - 2)} \text{ und } \delta_{\mu_{1,2}} := \frac{n_1 n_2}{n_1 + n_2}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) \tag{54}$$

gilt, dass

$$\nu T^2 \sim f(\delta_{\mu_{1,2}}, m, n_1 + n_2 - m - 1). \tag{55}$$

Der Ablehnungsbereich des betrachteten Tests ergibt sich als $A :=]k, \infty[$ Also ergibt sich

$$\begin{aligned} q_\phi(\mu) &= \mathbb{P}_\mu(\phi = 1) \\ &= \mathbb{P}_\mu\left(T^2 \in]k, \infty[\right) \\ &= \mathbb{P}_\mu\left(T^2 > k\right) \\ &= 1 - \mathbb{P}_\mu\left(T^2 \leq k\right) \\ &= 1 - F\left(\nu k; \delta_{\mu_{1,2}}, m, n_1 + n_2 - m - 1\right) \end{aligned} \tag{56}$$

(5) Analyse der Testgütefunktion

Beispiele

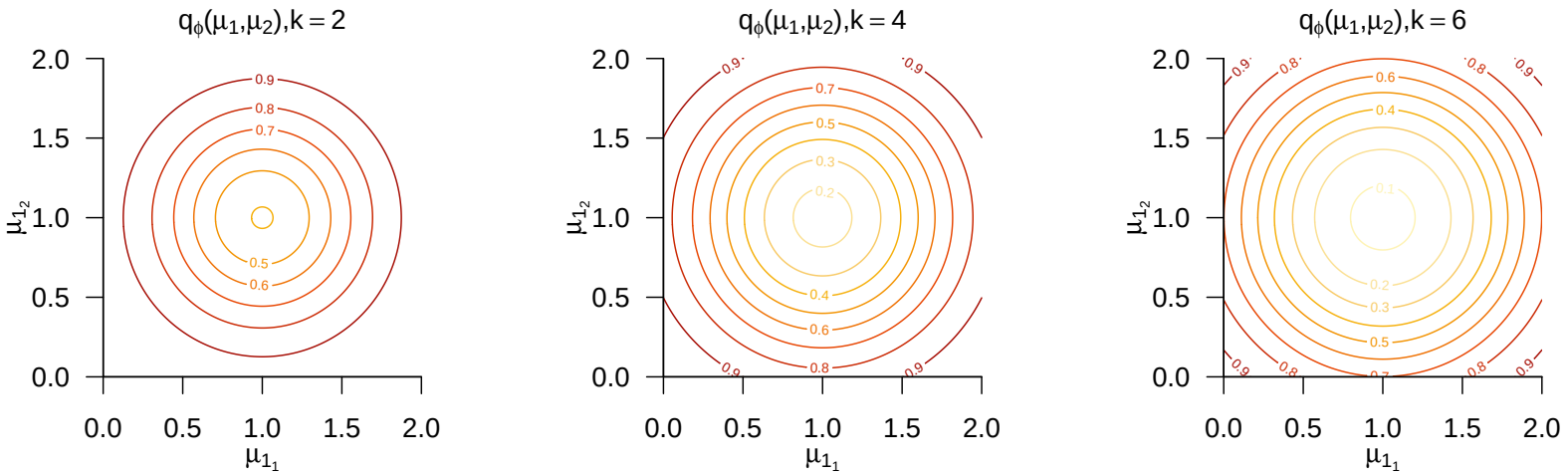
$$m := 2, n_1 := 15, n_2 := 15, \mu_2 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

```
# Modellparameter
library(matlib)                                                    # R Paket
m           = 2                                                    # m
n_1         = 15                                                   # n_1
n_2         = 15                                                   # n_2
nu          = (n_1+n_2-m-1)/(m*(n_1+n_2-2))                        # \nu
Sigma       = diag(m)                                              # \Sigma = I_2
iSigma      = inv(Sigma)                                           # \Sigma^{-1}
mu_2        = matrix(c(1,1), nrow = 2)                             # \mu_2

# Testparameter
k_all       = c(2,4,6)                                             # k <-> \phi
n_k         = length(k_all)                                        # Anzahl k Werte/Tests

# q_\phi(\mu) Evaluation
mu_min      = 0                                                    # \mu_1 Minimum
mu_max      = 2                                                    # \mu_1 Maximum
mu_res      = 1e3                                                  # \mu_1 Auflösung
mu_i        = seq(mu_min,mu_max,len = mu_res)                      # mu_1i
q_phi       = array(dim = c(mu_res, mu_res, length(k_all)))        # q_\phi Array
for(k in 1:n_k){
    for(i in 1:mu_res){
        for(j in 1:mu_res){
            mu_1         = matrix(c(mu_i[i],mu_i[j]), nrow = 2)    # \mu_1
            delta        = (((n_1*n_2)/(n_1+n_2))*                 # \delta_{\mu_{1,2}}
                            t(mu_1 - mu_2) %*% iSigma %*% (mu_1-mu_2))
            q_phi[i,j,k] = 1-pf(nu*k_all[k],m,n_1+n_2-m-1,delta)}}}    # q_\phi(\mu)
```

(5) Analyse der Testgütefunktion

Beispiele

$$m := 2, n_1 := 15, n_2 := 15, \mu_2 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Sigma := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



(6) Testumfangkontrolle

Theorem (Testumfangkontrolle)

ϕ sei der im obigen Testszenario definierte Test. Dann ist ϕ ein Level-α_0-Test mit Testumfang α_0, wenn der kritische Wert definiert ist durch

$$k_{\alpha_0} := \frac{1}{\nu} F^{-1}\left(1-\alpha_0; m, n_1+n_2-m-1\right) \tag{57}$$

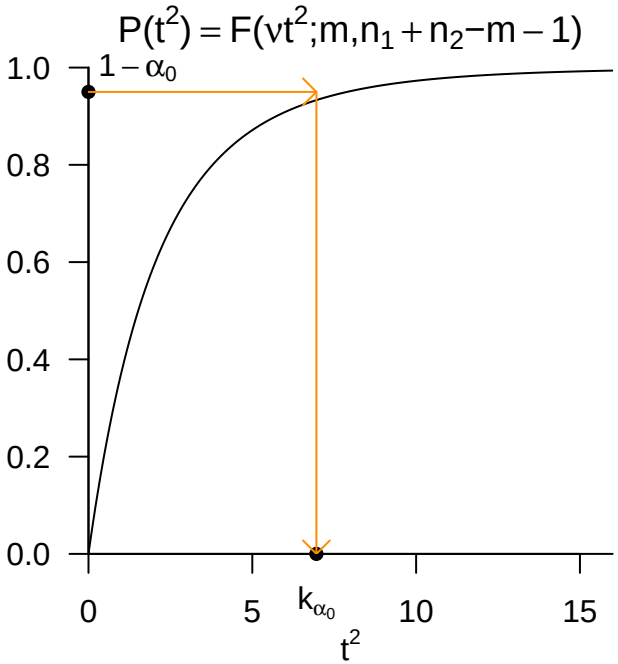
wobei $F^{-1}(\cdot; m, n_1+n_2-m-1)$ die inverse KVF der f-Verteilung mit Freiheitsgradparametern m und n_1+n_2-m-1 ist.

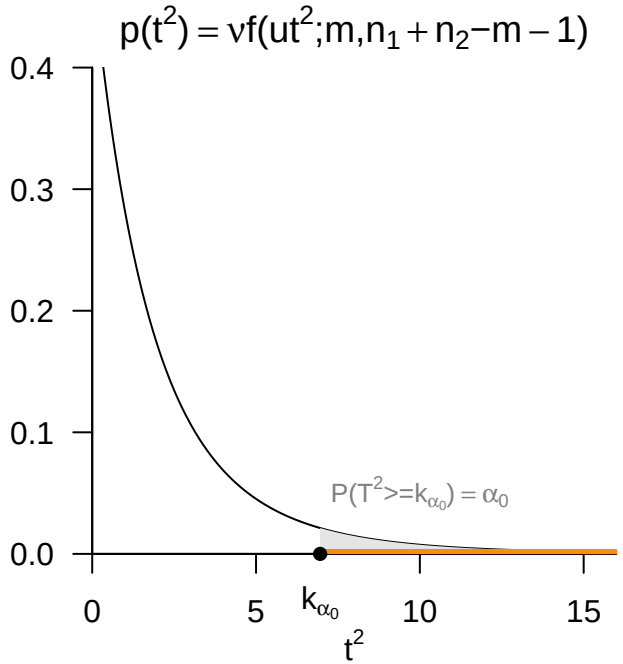
Bemerkung

- Der Beweis erfolgt analog zum Einstichprobenszenario.

(6) Testumfangkontrolle

Wahl von $k_{\alpha_0} := \frac{1}{\nu} F^{-1}(1-\alpha_0; m, n_1+n_2-m-1)$ mit $m=2, n_1=15, n_2=15, \alpha_0 := 0.05$





(6) Testumfangkontrolle

```
# Modellparameter bei Zutreffen von H0
m     = 2                                                    # Dimensionalität der Zufallsvektoren/Dat
n     = 15                                                   # Anzahl Datenpunkte pro Stichprobe
n_1   = n                                                    # Anzahl Datenpunkte Stichprobe 1
n_2   = n                                                    # Anzahl Datenpunkte Stichprobe 1
mu_1  = matrix(c(1,1) , nrow = 2)                            # wahrer, aber unbekannter, Erwartungswertparameter
mu_2  = mu_1                                                 # wahrer, aber unbekannter, Erwartungswertparameter
Sigma = matrix(c(0.4,0.1, 0.1,0.4), nrow = 2, byrow = TRUE)  # wahrer, aber unbekannter, Kovarianzmatrixparameter

# Testparameter
nu          = (n_1+n_2-m-1)/(m*(n_1+n_2-2))                  # \nu
alpha_0     = 0.05                                           # Signifikanzlevel
k_alpha_0   = (1/nu)*qf(1-alpha_0,m,n_1+n_2-m-1)             # kritischer Wert

# Simulation
library(MASS)                                                # R Paket für multivariate Normalverteilungen
library(matlib)                                              # R Paket für Matrizenrechnung
nsim  = 1e4                                                  # Anzahl Simulationen/Datensatzrealisierungen
Y1b   = matrix(rep(NaN,m*nsim), nrow = 2)                    # Stichprobenmittelarray
Y2b   = matrix(rep(NaN,m*nsim), nrow = 2)                    # Stichprobenmittelarray
phi   = rep(NaN,nsim)                                        # phi Array
j_n   = matrix(rep(1,n), nrow = n)                           # 1_n
I_n   = diag(n)                                              # I_n
J_n   = matrix(rep(1,n^2), nrow = n)                         # 1_{nn}
for(s in 1:nsim){                                            # Simulationsiterationen
    Y_1     = t(mvrnorm(n,mu_1,Sigma))                       # Y_1j \sim N(\mu_1,\Sigma), j = 1,...,n_1
    Y_2     = t(mvrnorm(n,mu_2,Sigma))                       # Y_2j \sim N(\mu_2,\Sigma), j = 1,...,n_2
    Y_1_bar = (1/n)*(Y_1 %*% j_n)                            # Stichprobenmittel
    Y_2_bar = (1/n)*(Y_2 %*% j_n)                            # Stichprobenmittel
    C_1     = (1/(n_1-1))*(Y_1 %*% (I_n-(1/n)*J_n) %*% t(Y_1))   # Stichprobenkovarianzmatrix
    C_2     = (1/(n_2-1))*(Y_2 %*% (I_n-(1/n)*J_n) %*% t(Y_2))   # Stichprobenkovarianzmatrix
    C       = ((n_1-1)*C_1+(n_2-1)*C_2)/(n_1+n_2-2)          # Gepoolte Stichprobenkovarianzmatrix
    T2      = (((n_1*n_2)/(n_1+n_2))*                        # T^2 Statistik
              t(Y_1_bar-Y_2_bar) %*% inv(C) %*% (Y_1_bar-Y_2_bar))
    if(T2 > k_alpha_0){phi[s] = 1} else {phi[s] = 0}         # Evaluation von phi
}
cat("\nKritischer Wert              =", k_alpha_0,
    "\nGeschätzter Testumfang alpha =", mean(phi))
```

```
Kritischer Wert              = 6.96
Geschätzter Testumfang alpha = 0.0476
```

(6) Testumfangkontrolle

Praktisches Vorgehen

- Man nimmt an, dass zwei vorliegende Datensätze $y_{11}, ..., y_{1n_1}$ und $y_{21}, ..., y_{2n_2}$ Realisationen von m-dimensionalen Zufallsvektoren

$$Y_{11}, ..., Y_{1n_1} \sim N(\mu_1, \Sigma) \text{ und } Y_{21}, ..., Y_{2n_2} \sim N(\mu_2, \Sigma) \tag{58}$$

  mit unbekannten Parametern μ_1, μ_2, Σ sind.

- Man möchte entscheiden, ob eher $H_0 : \mu_1 = \mu_2$ oder $H_1 : \mu_1 \neq \mu_2$ zutrifft.
- Man wählt ein Signifikanzlevel α_0 und bestimmt den zugehörigen kritischen Wert k_{α_0}.
- Anhand von $m, n_1, n_2, \bar{Y}_1, \bar{Y}_2$ und der gepoolten Stichprobenstandardabweichung C berechnet man die Realisierung der Zweistichproben-T^2-Teststatistik t^2.
- Wenn t^2 größer als k_{α_0} ist lehnt man die Nullhypothese ab, andernfalls lehnt man sie nicht ab.
- Die oben entwickelte Theorie des Zweistichproben-T-^{2}Tests garantiert dann, dass man in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.

(7) p-Werte

Bestimmung des p-Wertes

- Per Definition ist der p-Wert das kleinste Signifikanzlevel α_0, bei welchem man die Nullhypothese basierend auf einem vorliegenden Wert der Teststatistik ablehnen würde.
- Bei $T^2 = t^2$ würde H_0 für jedes α_0 mit $t^2 \geq \frac{1}{\nu} F^{-1}(1-\alpha_0; m, n_1+n_2-m-1)$ abgelehnt werden. Für diese α_0 gilt, wie analog im Eintsichprobenstzenario gezeigt

$$\alpha_0 \geq \mathbb{P}\left(T^2 \geq t^2\right) \tag{59}$$

- Das kleinste $\alpha_0 \in [0,1]$ mit $\alpha_0 \geq \mathbb{P}\left(T^2 \geq t^2\right)$ ist dann $\alpha_0 = \mathbb{P}(T^2 \geq t^2)$, also folgt

$$\text{p-Wert } = \mathbb{P}\left(T^2 \geq t^2\right) = 1 - F(\nu t^2; m, n_1+n_2-m-1) \tag{60}$$

- Zum Beispiel ergibt sich bei
  - $m = 2, n_1 = 15$ und $n_2 = 15$ der p-Wert für $t^2 = 5$ zu 0.11
  - $m = 2, n_1 = 15$ und $n_2 = 15$ der p-Wert für $t^2 = 7$ zu 0.05
  - $m = 2, n_1 = 99$ und $n_2 = 99$ der p-Wert für $t^2 = 5$ zu 0.09
  - $m = 4, n_1 = 15$ und $n_2 = 15$ der p-Wert für $t^2 = 7$ zu 0.21

Zweistichproben-T^2-Tests

Anwendungsbeispiel

```
# R Pakete
library(foreign)                                                    # Dateneinlesen
library(matlib)                                                     # Matrixalgebra

# Datenpräprozessierung
fname     = file.path(getwd(), "10_Daten", "studienerfolg.sav")     # Dateiname
D         = read.spss(fname, to.data.frame = T)                     # Dateneinlesen
Y_1       = t(as.matrix(D[D$Gruppe == "ungenügend", 2:3]))          # Daten Gruppe 1
Y_2       = t(as.matrix(D[D$Gruppe == "gut", 2:3]))                 # Daten Gruppe 2

# Testparameter
m         = 2                                                       # Datendimensionalität
n         = 15                                                      # Datenpunkte pro Gruppe
n_1       = n                                                       # Datenpunkte Gruppe 1
n_2       = n                                                       # Datenpunkte Gruppe 2
nu        = (n_1+n_2-m-1)/(m*(n_1+n_2-2))                           # \nu
alpha_0   = 0.05                                                    # Signifikanzlevel
k_alpha_0 = (1/nu)*qf(1-alpha_0,m,n_1+n_2-m-1)                      # kritischer Wert

# Testevaluation
j_n      = matrix(rep(1,n), nrow = n)                               # 1_n
I_n      = diag(n)                                                  # I_n
J_n      = matrix(rep(1,n^2), nrow = n)                             # 1_{nn}
Y_1_bar  = (1/n)*(Y_1 %*% j_n)                                      # Stichprobenmittel
Y_2_bar  = (1/n)*(Y_2 %*% j_n)                                      # Stichprobenmittel
C_1      = (1/(n_1-1))*(Y_1 %*% (I_n-(1/n)*J_n) %*% t(Y_1))         # Stichprobenkovarianzmatrix
C_2      = (1/(n_2-1))*(Y_2 %*% (I_n-(1/n)*J_n) %*% t(Y_2))         # Stichprobenkovarianzmatrix
C        = ((n_1-1)*C_1+(n_2-1)*C_2)/(n_1+n_2-2)                    # Gepoolte Stichprobenkovarianzmatrix
T2       = (((n_1*n_2)/(n_1+n_2))*                                  # T^2 Statistik
            t(Y_1_bar-Y_2_bar) %*% inv(C) %*% (Y_1_bar-Y_2_bar))
if(T2 > k_alpha_0){phi = 1} else {phi = 0}                          # Test 1_{T^2 > k_alpha_0}
p         = 1 - pf(nu*T2,m,n_1+n_2-m-1)                             # p-Wert
```

Zweistichproben-T^2-Tests

Anwendungsbeispiel

```
# Ausgabe
cat("Y_1_bar = ", Y_1_bar,
    "\nY_2_bar = ", Y_2_bar,
    "\nC       = ", C,
    "\nT^2     = ", T2,
    "\nalpha_0 = ", alpha_0,
    "\nk       = ", k_alpha_0,
    "\nphi     = ", phi,
    "\np       = ", p)
```

```
Y_1_bar =  51.9 47.4
Y_2_bar =  66.5 61.7
C       =  78.7 -8.94 -8.94 390
T^2     =  25
alpha_0 =  0.05
k       =  6.96
phi     =  1
p       =  0.000181
```

Anwendungsbeispiel mit `MVTests::TwoSamplesHT2()`

```
library(MVTests)                                # R Pakete
Y     = D[c(1:15, 31:45), 2:3]                  # Dataframe von Interesse
G     = c(rep(1,15), rep(2,15))                 # Gruppenindikatoren
phi   = TwoSamplesHT2(Y,G, alpha_0)             # Zweistichproben-T^2-Test

# Ausgabe
cat("Y_1_bar = " , phi$Descriptive1[1,],
    "\nY_2_bar = ",  phi$Descriptive2[1,],
    "\nT^2     = ", phi$HT2,
    "\nalpha_0 = ", phi$alpha,
    "\nk       = " ,phi$F,
    "\np       = ", phi$p.value)
```

```
Y_1_bar =  51.9 47.4
Y_2_bar =  66.5 61.7
T^2     =  25
alpha_0 =  0.05
k       =  12.1
p       =  0.000181
```

Vorbemerkungen

Einstichproben-T^2-Tests

Zweistichproben-T^2-Tests

Univariates vs. multivariates Testen

Selbstkontrollfragen

Univariates vs. multivariates Testen

Gegeben sei ein Anwendungsszenario mit n Beobachtungen von m Variablen

Bei Durchführung von m univariaten Tests entsteht ein multiples Testproblem

- Induktion multipler Typ I und Typ II Fehlerraten (cf. Ostwald et al. (2019))
- Familywise Error Rate (FWER) $= \mathbb{P}(\geq 1$ Typ I Fehler$)$
- Bei unabhängigen Variablen bietet zur FWER Kontrolle die *Bonferroni Korrektur* an.
- Bei Durchführung eines multivariaten Tests entsteht kein multiples Testproblem.

Multivariate Tests beziehen Variablenkovarianzen explizit mit ein.

- Bei Durchführung von m univariaten Tests werden Variablenkorrelationen aktiv ignoriert.

Sollten m univariate Tests oder ein multivariater Test durchgeführt werden?

- Je nachdem, ob die Daten als multi- oder univariate Realisierung konzipiert werden.
- Je nachdem, welche geometrische Form des Annahmebereiches gewünscht ist.
- Prinzipiell sollte im wissenschaftlichen Diskurs überhaupt nicht getestet werden.

Wir betrachten in der Folge Simulationsszenarien mit $m := 2$.

Univariate vs. multivariate Einstichproben-T$^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$

```
# R Pakete
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrizenrechnung

# Modellparameter
m     = 2                                              # Dimensionalität der Zufallsvektoren/Daten
n     = 15                                             # Anzahl der Datenpunkte
mu    = matrix(c(0,0) , nrow = 2)                      # Erwartungswertparameter
Sigma = matrix(c(1,0,0,1), nrow = 2, byrow = TRUE)     # Kovarianzmatrixparameter

# Testparameter
alpha = 0.05                                           # Signifikanzlevel
nu    = (n-m)/(m*(n-1))                                # T^2-Test Parameter
k_T2  = (1/nu)*qf(1-alpha, m, n-m)                     # T^2-Test kritischer Wert
k_Tu  = qt(1-(1/2)*alpha  ,n-1)                        # T-Test kritischer Wert unkorrigiert
k_Tc  = qt(1-(1/2)*alpha/m,n-1)                        # T-Test kritischer Wert Bonferonni korrigiert

# Simulation der Testumfangkontrolle
nsim  = 1e4                                            # Anzahl Simulation
phi   = matrix(rep(NaN,nsim*5), nrow = 5)              # Testentscheidungsarray
j_n   = matrix(rep(1,n), nrow = n)                     # 1_n
I_n   = diag(n)                                        # I_n
J_n   = matrix(rep(1,n^2), nrow = n)                   # 1_{nn}
for(s in 1:nsim){                                      # Simulationsiterationen
    Y        = t(mvrnorm(n,mu,Sigma))                  # Y_i \sim N(\mu,\Sigma), i = 1,...,n
    Y_bar    = (1/n)*(Y %*% j_n)                       # Stichprobenmittel
    C        = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))  # Stichprobenkovarianzmatrix
    phi[1,s] =  n*t(Y_bar) %*% inv(C) %*% (Y_bar)  > k_T2  # T^2-Test mit \mu_0
    for(i in 1:m){                                     # T-Test Iterationen
      y_bar      = Y_bar[i]                            # Stichprobenmittel
      sigma_hat  = sqrt(C[i,i])                        # Stichprobenstandardabweichung
      phi[i+1,s] = abs(sqrt(n)*y_bar/sigma_hat)  > k_Tu    # Unkorrigierter T-Test mit \mu_0 = 0
      phi[i+3,s] = abs(sqrt(n)*y_bar/sigma_hat)  > k_Tc}}  # Korrigierter   T-Test mit \mu_0 = 0
```

Univariate vs. multivariate Einstichproben-$T^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$

```
Kritischer Wert T^2-Test                                    = 8.2
Kritischer Wert T-Test                                      = 2.14
Kritischer Wert T-Test Bonferroni                           = 2.51

Geschätzte Typ I Fehlerwahrscheinlichkeit T^2-Test          = 0.049

Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 1 unkorrigiert = 0.0474
Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 2 unkorrigiert = 0.0523
Geschätzte FWER T-Tests unkorrigiert                        = 0.0978

Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 1 Bonferroni   = 0.0245
Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 2 Bonferroni   = 0.0264
Geschätzte FWER T-Tests Bonferroni                          = 0.0506
```

Univariate vs. multivariate Einstichproben-T$^{(2)}$-Tests mit $\Sigma \neq \sigma^2 I_2$

```
Kritischer Wert T^2-Test                                    = 8.2
Kritischer Wert T-Test                                      = 2.14
Kritischer Wert T-Test Bonferroni                           = 2.51

Geschätzte Typ I Fehlerwahrscheinlichkeit T^2-Test          = 0.0445

Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 1 unkorrigiert = 0.0471
Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 2 unkorrigiert = 0.0465
Geschätzte FWER T-Tests unkorrigiert                        = 0.0675

Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 1 Bonferroni   = 0.0214
Geschätzte Typ I Fehlerwahrscheinlichkeit T-Test 2 Bonferroni   = 0.0226
Geschätzte FWER T-Tests Bonferroni                          = 0.0226
```

- Kovariabilität von Variablen reduziert die FWER.
- Die Bonferroni FWER Korrektur wird *konservativ*, also $\mathbb{P}(\geq 1 \text{ Typ I Fehler}) < \alpha_0$.

Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests

Zur Visualisierung von Stichprobenmittel und Testentscheidung bieten sich (nur) Z^2-Test an.

Z^2-Test $\approx$ T^2-Test mit als bekannt vorausgesetzter Kovarianzmatrix bei $Y_i \sim N(\mu, \Sigma)$.

- Einstichproben-Z^2-Teststatistik:

$$Z^2 := n(\bar{Y} - \mu_0)^T \Sigma^{-1} (\bar{Y} - \mu_0) \tag{61}$$

- Verteilung der Einstichproben-Z^2-Teststatistik bei $H_0 : \mu = \mu_0$:

$$Z^2 \sim \chi^2(m) \tag{62}$$

- Kritischer Wert für Testumfangkontrolle:

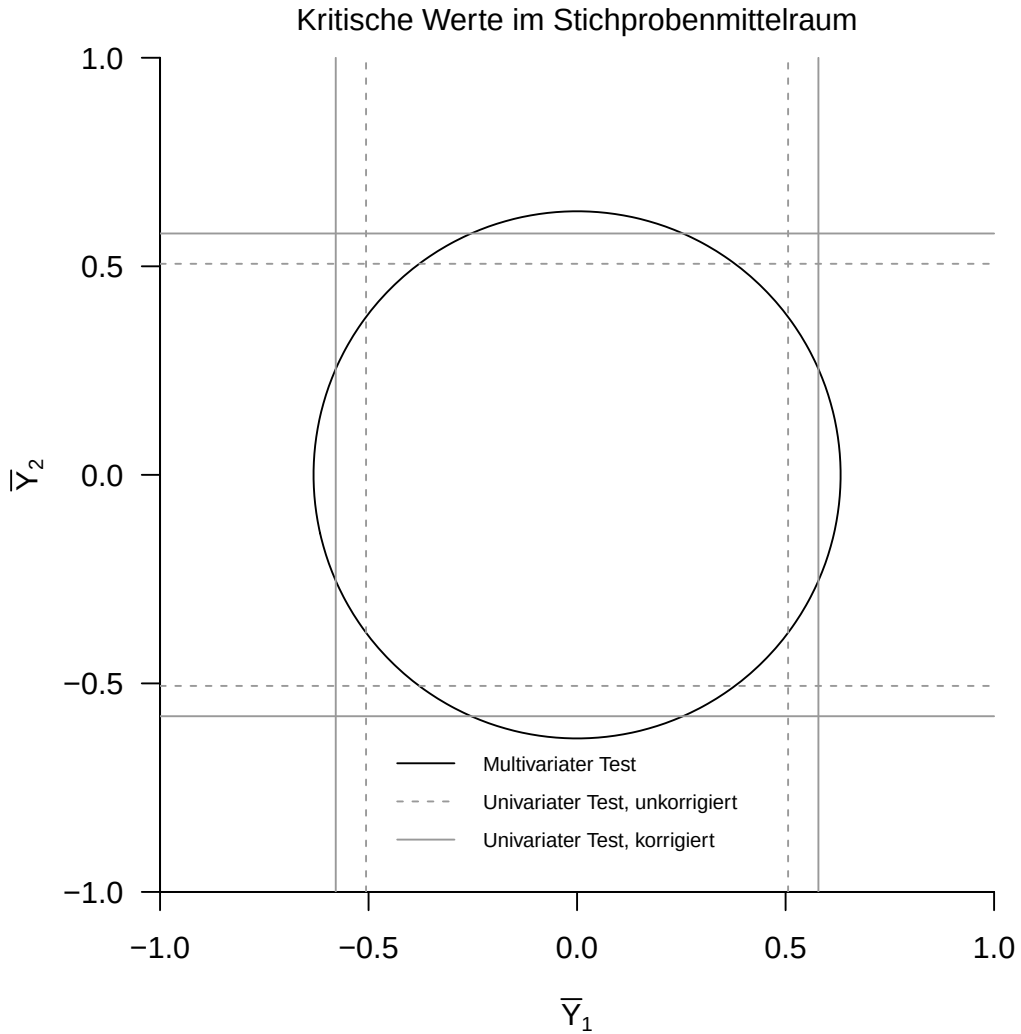
$$k_{\alpha_0} := \Xi^{2^{-1}}(1 - \alpha_0; m) \tag{63}$$

wobei $\Xi^{2^{-1}}$ die inversen KVF der χ^2 Verteilung bezeichnet.

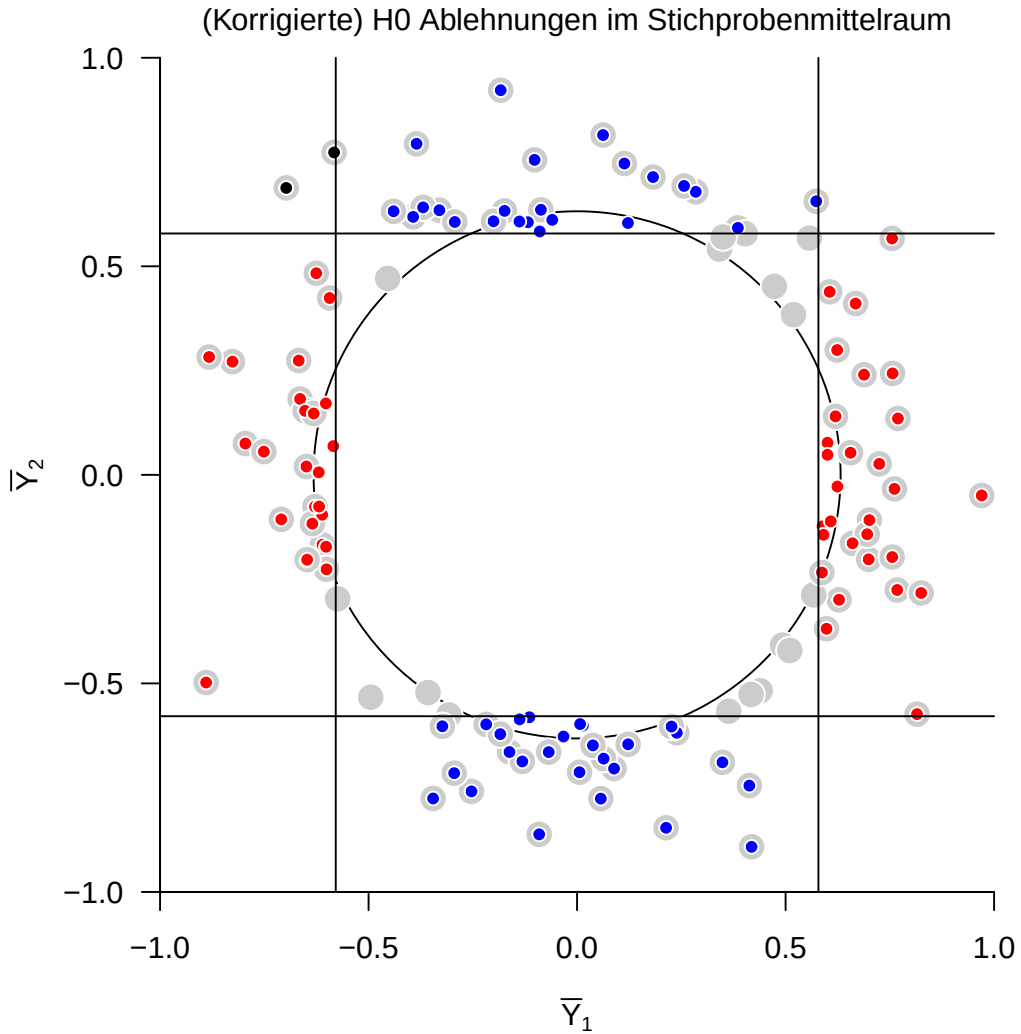
Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$

```
# R Pakete
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrizenrechnung

# Modellparameter
m     = 2                                              # Dimensionalität der Zufallsvektoren/Daten
n     = 15                                             # Anzahl der Datenpunkte
mu    = matrix(c(0,0) , nrow = 2)                      # Erwartungswertparameter
Sigma = matrix(c(1,0,0,1), nrow = 2, byrow = TRUE)     # Kovarianzmatrixparameter

# Testparameter
alpha = 0.05                                           # Signifikanzlevel
k_Z2  = qchisq(1-alpha, m)                             # Z^2-Test kritischer Wert
k_Zu  = qnorm(1-(1/2)*alpha)                           # Z-Test kritischer Wert unkorrigiert
k_Zc  = qnorm(1-(1/2)*alpha/m)                         # Z-Test kritischer Wert Bonferonni korrigiert

# Simulation der Testumfangkontrolle
nsim  = 2e3                                            # Anzahl Simulation
YB    = matrix(rep(NaN,nsim*2), nrow  = 2)             # Stichprobenmittelarray
phi   = matrix(rep(NaN,nsim*5), nrow  = 5)             # Testentscheidungsarray
j_n   = matrix(rep(1,n), nrow = n)                     # 1_n
I_n   = diag(n)                                        # I_n
J_n   = matrix(rep(1,n^2), nrow = n)                   # 1_{nn}
for(s in 1:nsim){                                      # Simulationsiterationen
    Y        = t(mvrnorm(n,mu,Sigma))                  # Y_i \sim N(\mu,\Sigma), i = 1,...,n
    YB[,s]   = (1/n)*(Y %*% j_n)                       # Stichprobenmittel
    phi[1,s] = (n*t(YB[,s])%*%inv(Sigma)%*%YB[,s]) > k_Z2   # T^2-Test mit \mu_0
    for(i in 1:m){                                     # T-Test Iterationen
      y_bar      = YB[i,s]                             # Stichprobenmittel
      sigma      = sqrt(Sigma[i,i])                    # Stichprobenstandardabweichung
      phi[i+1,s] = abs(sqrt(n)*y_bar/sigma)  > k_Zu    # Unkorrigierter T-Test mit \mu_0 = 0
      phi[i+3,s] = abs(sqrt(n)*y_bar/sigma)  > k_Zc}}  # Korrigierter   T-Test mit \mu_0 = 0
```

Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$

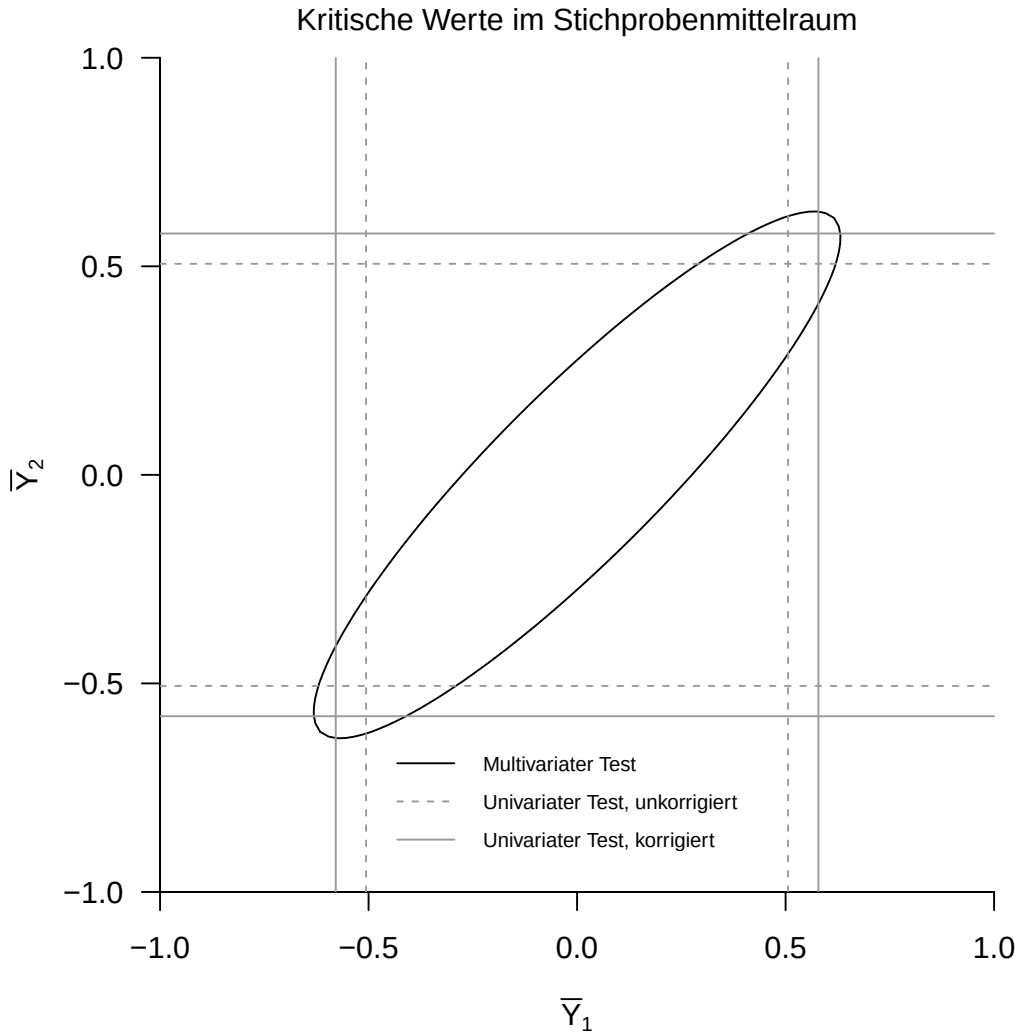


Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$

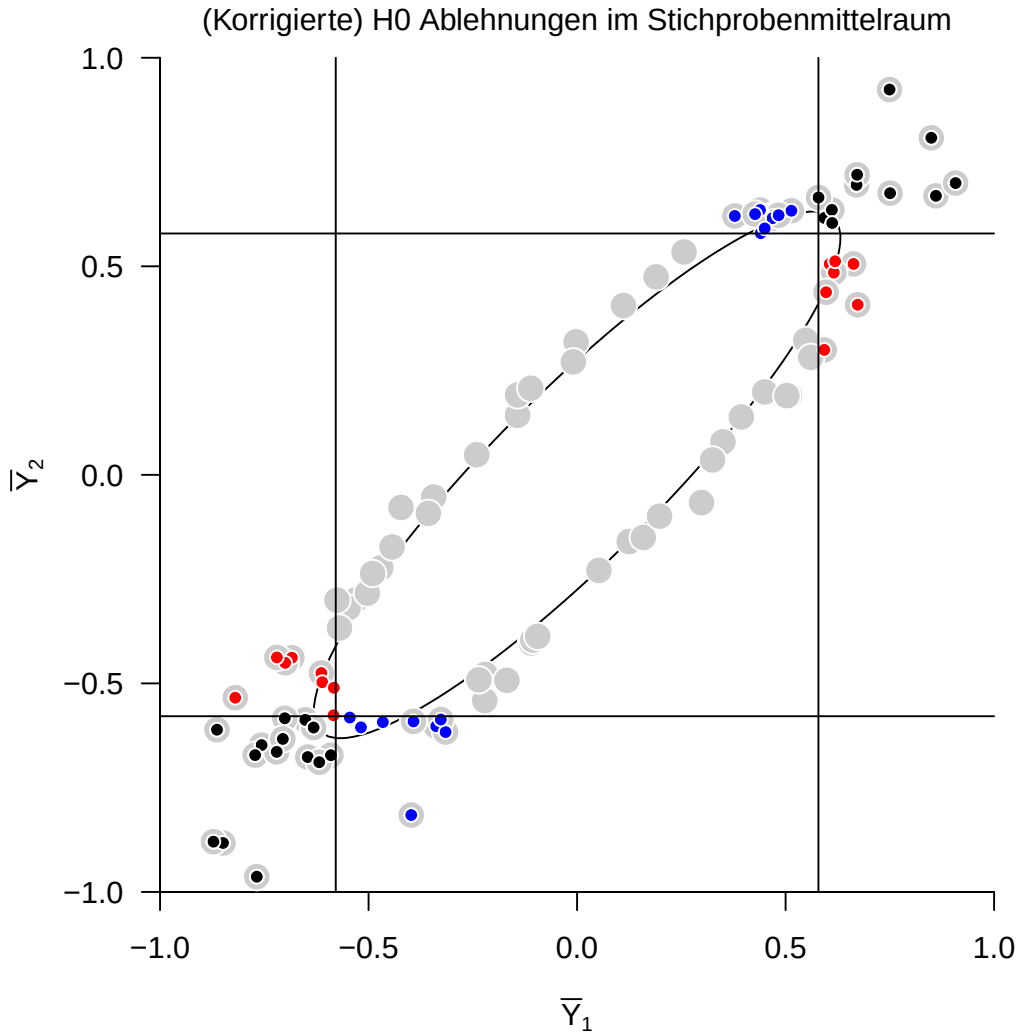


● $\phi(Y) = 1$, ● $\phi(Y_1) = 1$, ● $\phi(Y_2) = 1$, ● $\phi(Y_1) = 1$ und $\phi(Y_2) = 1$

Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests mit $\Sigma \neq \sigma^2 I_2$



Univariate vs. multivariate Einstichproben-$Z^{(2)}$-Tests mit $\Sigma = \sigma^2 I_2$



● $\phi(Y) = 1$, ● $\phi(Y_1) = 1$, ● $\phi(Y_2) = 1$, ● $\phi(Y_1) = 1$ und $\phi(Y_2) = 1$

Vorbemerkungen

Einstichproben-T^2-Tests

Zweistichproben-T^2-Tests

Univariates vs. multivariates Testen

Selbstkontrollfragen

Selbstkontrollfragen

1. Erläutern Sie das Modell und die Standardprobleme der Frequentistischen Inferenz.
2. Erläutern Sie die Standardannahmen Frequentistischer Inferenz.
3. Definieren Sie den Begriff der Mahalanobis Distanz.
4. Erläutern Sie den Unterschied zwischen einer f-Verteilung und einer nichtzentralen f-Verteilung.
5. Beschreiben Sie das Anwendungsszenario für einen Einstichproben-T^2-Test.
6. Geben Sie das statistische Modell eines Einstichproben-T^2-Test in klassischer Form an.
7. Geben Sie das statistische Modell eines Einstichproben-T^2-Test in generativer Form an.
8. Erläutern Sie das statistische Modell eines Einstichproben-T^2-Test in generativer Form.
9. Definieren Sie die Testhypothesen eines Einstichproben-T^2-Test mit einfacher Nullhypothese und zusammengesetzter Alternativhypothese.
10. Definieren Sie die Einstichproben-T^2-Teststatistik.
11. Erläutern Sie, wann die Einstichproben-T^2-Teststatistik hohe Werte annimmt.
12. Geben Sie die WDF und KVF der Einstichproben-T^2-Teststatistik an.
13. Geben Sie den kritischen Wert für einen Level-α_0-Einstichproben-T^2-Testmit einfacher Nullhypothese und zusammengesetzter Alternativhypothese und Testumfang α_0 an.
14. Erläutern Sie das praktische Vorgehen bei der Durchführung eines Einstichproben-T^2-Tests.
15. Definieren Sie den Begriff des p-Wertes.
16. Geben Sie den p-Wert für einen Einstichproben-T^2-Test mit einfacher Nullhypothese und zusammengesetzter Alternativhypothese an und erläutern Sie die Komponenten des entsprechenden Ausdrucks.
17. Definieren Sie die Powerfunktion eines Einstichproben-T^2-Test mit einfacher Nullhypothese und zusammengesetzter Alternativhypothese und erläutern Sie ihre Komponenten.

References

Anderson, T. W. 2003. *An Introduction to Multivariate Statistical Analysis*. 3rd ed. Wiley Series in Probability and Statistics. Hoboken, N.J: Wiley-Interscience.

Hotelling, Harold. 1931. "The Generalization of Student's Ratio." *The Annals of Mathematical Statistics* 2 (3): 360–78. https://doi.org/10.1214/aoms/1177732979.

Ostwald, Dirk, Sebastian Schneider, Rasmus Bruckner, and Lilla Horvath. 2019. "Power, Positive Predictive Value, and Sample Size Calculations for Random Field Theory-Based fMRI Inference." *BioRxiv: Doi.org/10.1101/613331*, April. https://doi.org/10.1101/613331.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(11) Einfaktorielle Varianzanalyse

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation mit

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Anwendungsszenario und Datendeskription

Anwendungsszenario

- Zwei oder mehr Stichproben (oft Gruppen genannt) experimenteller Einheiten.
- Annahme der unabhängigen und identischen Normalverteilung $N(\mu_i, \Sigma)$ der Daten.
- $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. unbekannt.
- Absicht des inferentiellen Testens der Nullhypothese identischer Gruppenerwartungswerte.

Generalisierung des Zweistichproben-T^2-Tests bei unabhängigen Stichproben mit einfacher Nullhypothese für mehr als zwei Stichproben

Anwendungsbeispiele

- BDI/Glukokortikoid Analyse von drei Gruppen psychiatrischer Patient:innen nach PA, CBT,ST
  - Inferentielle Evidenz für Gruppenerwartungswertunterschiede?
  - Evidenz für unterschiedliche Therapiewirksamkeit?
- Gruppenvergleich von Testdaten bei gutem, befriedigendem, ungenügenden Studienabschluss
  - Inferentielle Evidenz für Gruppenerwartungswertunterschiede?
  - Evidenz für Studienerfolgsprädiktivität der Testdaten?

Anwendungsbeispiel

Nach Rudolf and Buse (2020) Kapitel 4, vgl. Einheiten zur Prädiktiven Modellierung

Datensatz zum Verhältnis psychologischer Testdiagnostik und Studienerfolg

Faktor Studienerfolg mit $k = 3$ Leveln (*Ungenügend, Befriedigend, Gut*)

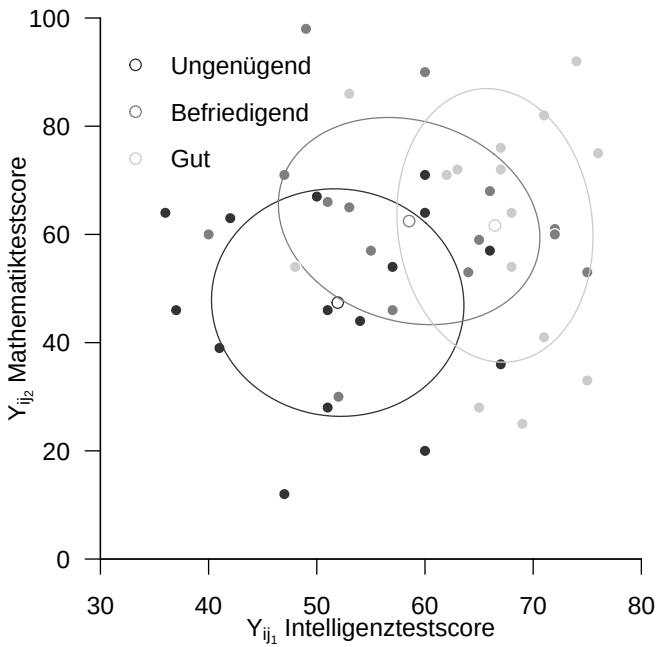
Datendimension $m = 2$ mit $y_{ij} \in \mathbb{R}^2$ (y_{ij_1} *Intelligenztestscore*, y_{ij_2} *Mathematiktestscore*)

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library("foreign")
S       = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datenpräprozessierung
m       = 2                                     # Datendimension von Interesse
k       = 3                                     # Anzahl Gruppen
l       = 15                                    # Anzahl Datenpunkte pro Gruppe
Y       = array(dim = c(m,l,k))                 # Datenarrayinitialisierung
Y[,,1]  = rbind(S$X1[S$Gruppe == "ungenügend"],   # Y_{1j_1} Intelligenztestscore
                S$X2[S$Gruppe == "ungenügend"])   # Y_{1j_2} Mathematiktestscore
Y[,,2]  = rbind(S$X1[S$Gruppe == "befriedigend"], # Y_{2j_1} Intelligenztestscore
                S$X2[S$Gruppe == "befriedigend"]) # Y_{2j_2} Mathematiktestscore
Y[,,3]  = rbind(S$X1[S$Gruppe == "gut"],          # Y_{3j_1} Intelligenztestscore
                S$X2[S$Gruppe == "gut"])          # Y_{3j_2} Mathematiktestscore
```

Anwendungsbeispiel

Datendeskription

```
Y_bar_i = array(dim = c(m,k))                                                    # Stichprobenmittelarray
C_i     = array(dim = c(m,m,k))                                                  # Stichprobenkovarianzmatrizenarray
j_l     = matrix(rep(1,l), nrow = l)                                             # 1_{l}
I_l     = diag(l)                                                                # Einheitsmatrix I_l
J_l     = matrix(rep(1,l^2), nrow = l)                                           # 1_{ll}
for (i in 1:k){                                                                  # Gruppeniterationen
    Y_bar_i[,i] = (1/l)*(Y[,,i] %*% j_l)                                         # Stichprobenmittel \bar{Y}_i
    C_i[,,i]    = (1/(l-1))*(Y[,,i] %*% (I_l-(1/l)*J_l) %*% t(Y[,,i]))}          # Stichprobenkovarianzmatrix C_i
```



Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Definition (Modell der einfaktoriellen Varianzanalyse)

Für $i = 1, \ldots, k$ sei

$$Y_{i1}, \ldots, Y_{il} \sim N(\mu_i, \Sigma) \tag{1}$$

eine Stichprobe eines multivariaten Normalverteilungsmodells von Größe l mit unbekanntem Erwartungswertparameter $\mu_i \in \mathbb{R}^m$ und unbekannten Kovarianzmatrixparameter $\Sigma \in \mathbb{R}^{m \times m}$ p.d. Dann heißt (1) *Klassisches Modell der einfaktoriellen Varianzanalyse*. Die Gesamtstichprobengröße bezeichnen wir mit $n := kl$. Äquivalent sei

$$Y_{ij} = \mu_i + \varepsilon_{ij} \text{ mit } \varepsilon_{ij} \sim N(0_m, \Sigma) \text{ für } i = 1, \ldots, k \text{ und } j = 1, \ldots, l, \tag{2}$$

wobei

- i die Stichproben und j die experimentellen Einheiten indizieren,
- l die Stichprobengrößen und $n := lk$ die Gesamtstichprobengröße sind,
- Y_{ij} beobachtbare Zufallsvektoren sind,
- $\mu_i \in \mathbb{R}^m$ feste Erwartungswertparameter der Stichprobenvariablen sind, und
- ε_{ij} unabhängige normalverteilte nicht-beobachtbare Zufallsvariablen mit $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.

Dann heißt (2) *Generatives Modell der einfaktoriellen Varianzanalyse.*

Bemerkungen

- Der Einfachheit halber setzen wir hier identische Stichprobengrößen voraus.
- Die Kovarianzmatrixparameter aller Stichproben werden als identisch vorausgesetzt.
- Wir verzichten auf einen Beweis der Äquivalenz von (1) und (2).
- Die Gesamtheit aller Stichprobenzufallsvektoren bezeichen wir mit $Y := \{Y_{ij}\}_{i=1,\ldots,k, j=1,\ldots,l}$.

Theorem (Parameterschätzer der einfaktoriellen Varianzanalyse)

Für $i = 1, \ldots, k$ sei mit

$$Y_{i1}, \ldots, Y_{il} \sim N(\mu_i, \Sigma) \tag{3}$$

für $\mu_i \in \mathbb{R}^m$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. das Modell der einfaktoriellen Varianzanalyse gegeben. Dann ist für $i = 1, \ldots, k$

$$\hat{\mu}_i := \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \tag{4}$$

ein unverzerrte Schätzer des Erwartungswertparameters μ_i und

$$\hat{\Sigma} := \frac{1}{lk - k} \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \hat{\mu}_i \right) \left(Y_{ij} - \hat{\mu}_i \right)^T \tag{5}$$

ein unverzerrter Schätzer des Kovarianzmatrixparameters Σ.

Bemerkungen

- $\hat{\mu}_i$ ist das Stichprobenmittel der $i$ten Gruppe.
- $\hat{\Sigma}$ ist die mit $(lk - k)$ skalierte Within Group Sum of Squares Matrix (siehe unten).
- Anstelle eines Beweis validieren wir die Aussage des Theorems mithilfe einer Simulation.

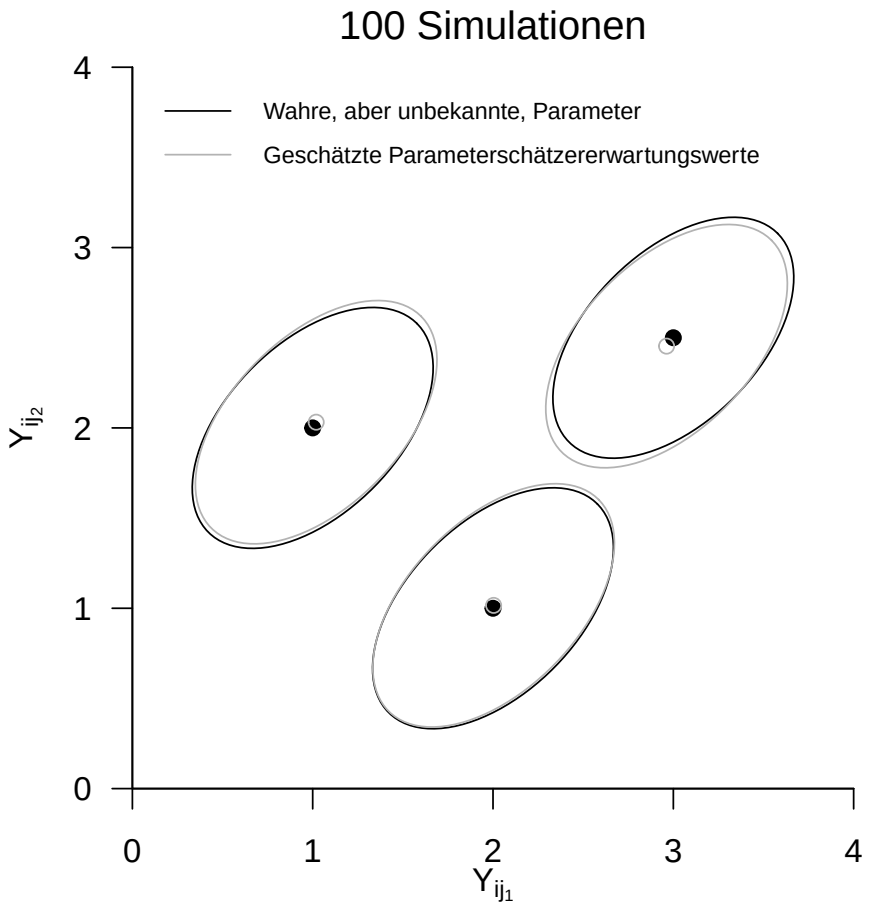
Parameterschätzer der einfaktoriellen Varianzanalyse

```
estimate = function(Y){

  # Diese Funktion evaluiert die Parameterschätzer einer einfaktoriellen
  # Varianzanalyse basierend auf einen m x l x k Datensatz Y.
  #
  # Input
  #     Y          : m x l x k Datenarray
  #
  # Output
  #     $mu_hat    : m x k \mu_i Parameterschätzer
  #     $Sigma_hat : m x m \Sigma Parametschätzer
  # ------------------------------------------------------------------------
  # Dimensionsparameter
  d         = dim(Y)                                    # Datensatzdimensionen
  m         = d[1]                                      # Datendimension
  l         = d[2]                                      # Anzahl Datenpunkte pro Gruppe
  k         = d[3]                                      # Anzahl Gruppen

  # Erwartungswertparameterschätzer
  mu_hat_i  = matrix(apply(Y,3,rowMeans), nrow = m)

  # Kovarianzmatrixparameterschätzer
  Sigma_hat = matrix(rep(0,m*m), nrow = m)
  for(i in 1:k){
      for(j in 1:l){
          Sigma_hat = Sigma_hat + (1/(l*k-k))*(Y[,j,i] - mu_hat_i[,i]) %*% t(Y[,j,i] - mu_hat_i[,i])
    }
  }


  # Outputspezifikation
  return(list(mu_hat_i = mu_hat_i, Sigma_hat = Sigma_hat))}
```

Parameterschätzer der einfaktoriellen Varianzanalyse

```r
# R Pakete
library(MASS)                                       # multivariate Normalverteilungen
library(matlib)                                     # Matrixalgebra


# Modellparameter
k           = 3                                     # Anzahl Gruppen
l           = 15                                    # Anzahl Datenpunkte pro Gruppe
m           = 2                                     # Datendimensionalität
mu_i        = matrix(c(1,2,2,1,3,2.5), ncol = k)    # Erwartungswertparameter
Sigma       = matrix(c(1,.5,.5,1)   , ncol = m)     # Kovarianzmatrixparameter

# Simulationsparameter und Arrays
nsm         = 1e2                                   # Anzahl Simulation
mu_hat_is   = array(dim = c(m,k,nsm))               # \hat{\mu}_i Array
Sigma_hats  = array(dim = c(m,m,nsm))               # \hat{\Sigma} Array

# Simulationen
for(s in 1:nsm){

    # Datengeneration
    Y               = array(dim = c(m,l,k))       # Datenarray
    for(i in 1:k){
        Y[,,i] = t(mvrnorm(l,mu_i[,i],Sigma))     # Datengeneration
    }
    S               = estimate(Y)                 # Parameterschätzung
    mu_hat_is[,,s]  = S$mu_hat_i                  # \hat{\mu}_i
    Sigma_hats[,,s] = S$Sigma_hat                 # \hat{\Sigma}
}

# Erwartungswertschätzung
E_hat_mu_i_hat  = apply(mu_hat_is , c(1,2), mean)
E_hat_Sigma_hat = apply(Sigma_hats, c(1,2), mean)
```

Parameterschätzer der einfaktoriellen Varianzanalyse



Anwendungsbeispiel

Parameterschätzung

```
# Parameterschätzung
S = estimate(Y)

# Ausgabe
print(S$mu_hat_i)
```

```
>      [,1] [,2] [,3]
> [1,] 51.9 58.5 66.5
> [2,] 47.4 62.5 61.7
```

```
print(S$Sigma_hat)
```

```
>       [,1]  [,2]
> [1,]  87.6 -14.9
> [2,] -14.9 348.3
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Überblick

Primäres Ziel der einfaktoriellen Varianzanalyse ist zumeist das Testen der Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k. \tag{6}$$

Die Nullhypothese gesagt also, dass zwischen den Gruppen keine Erwartungswertparameterunterschiede bestehen. Die Alternativhypothese lautet somit

$$H_1 : \mu_{i_c} \neq \mu_{j_c} \text{ für mindestens ein Paar } (i, j) \text{ mit } i \neq j \text{ und mindestens ein } c \text{ mit } 1 \leq c \leq m. \tag{7}$$

Die Alternativhypothese besagt also, dass sich mindestens zwei Erwartungswertparameter in mindestens einer ihrer Komponenten unterscheiden.

Kritische Wert-basierte Tests der Nullhypothesen können mit verschiedenen Teststatistiken (*Wilks'* Λ*, Pillai Statisik*) konstruiert werden. Diesen Teststatistiken ist gemein, dass sie auf eine Generalisierung der vom univariaten Fall bekannten Quadratsummenzerlegung zurück gehen. Wir führen also zunächst diese sogenannte *Kreuzproduktsummenmatrizenzerlegung* ein und betrachten dann die durch die Wilks' Λ und die Bartlett-Pillai-Spur induzierten Tests anhand der Gliederung (1) Teststatistik und Test, (2) Analyse der Teststatistik und (3) Testumfangkontrolle.

Einen Überblick über die Modellevaluation bei der univariaten einfaktoriellen Varianzanalyse gibt (12) Varianzanalysen.

Theorem (Kreuzproduktsummenmatrizenzerlegung)

Für $i = 1, \ldots, k$ und $j = 1, \ldots, l$ bezeichne Y_{ij} den $j$ten Stichprobenvektor der $i$ten Stichprobengruppe eines einfaktoriellen Varianzanalysemodells. Weiterhin seien

$$\bar{Y} := \frac{1}{n} \sum_{i=1}^{k} \sum_{j=1}^{l} Y_{ij} \text{ und } \bar{Y}_i := \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \tag{8}$$

das *Gesamtstichprobenmittel* und das i*te Gruppenstichprobenmittel*, respektive. Schließlich seien

$T := \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \bar{Y}\right) \left(Y_{ij} - \bar{Y}\right)^T$ die *Totale Sum of Squares Matrix*

$B := \sum_{i=1}^{k} l \left(\bar{Y}_i - \bar{Y}\right) \left(\bar{Y}_i - \bar{Y}\right)^T$ die *Between Group Sum of Squares Matrix*

$W := \sum_{i=1}^{k} \sum_{j=1}^{l} \left(Y_{ij} - \bar{Y}_i\right) \left(Y_{ij} - \bar{Y}_i\right)^T$ die *Within Group Sum of Squares Matrix*.

Dann gilt

$$T = B + W. \tag{9}$$

Bemerkungen

- $T \in \mathbb{R}^{m \times m}$ repräsentiert die totale Variabilität der Datenvektoren um das Gesamtstichprobenmittel.
- $B \in \mathbb{R}^{m \times m}$ repräsentiert die Variabilität der Gruppenstichprobenmittel um das Gesamtstichprobenmittel.
- $W \in \mathbb{R}^{m \times m}$ repräsentiert die Variabilität der Datenvektoren um ihre jeweiligen Gruppenstichprobenmittel.
- Die totale Variabilität wird hier also in zwei unabhängige Beiträge von Variabilität zerlegt.
- W heißt auch *Residualvariabiliät*, weil sie die verbleibende Variabilität nach Schätzung der Gruppenerwartungswertparameter quantifiziert und gilt das

$$W = (lk - k)\hat{\Sigma}. \tag{10}$$

- Die Kreuzproduktsummenmatrixzerlegung ergibt sich anhand von algebraischen Identitäten wie unten gezeigt.

Beweis

$$
\begin{aligned}
T &= \sum_{i=1}^{k}\sum_{j=1}^{l}(Y_{ij}-\bar{Y})(Y_{ij}-\bar{Y})^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}(Y_{ij}-\bar{Y}_i+\bar{Y}_i-\bar{Y})(Y_{ij}-\bar{Y}_i+\bar{Y}_i-\bar{Y})^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\Big((Y_{ij}-\bar{Y}_i)+(\bar{Y}_i-\bar{Y})\Big)\Big((Y_{ij}-\bar{Y}_i)+(\bar{Y}_i-\bar{Y})\Big)^T \\
&= \sum_{i=1}^{k}\sum_{j=1}^{l}\Big((Y_{ij}-\bar{Y}_i)(Y_{ij}-\bar{Y}_i)^T+2(Y_{ij}-\bar{Y}_i)(\bar{Y}_i-\bar{Y})^T+(\bar{Y}_i-\bar{Y})(\bar{Y}_i-\bar{Y})^T\Big) \\
&= \sum_{i=1}^{k}\left(\sum_{j=1}^{l}(Y_{ij}-\bar{Y}_i)(Y_{ij}-\bar{Y}_i)^T+\sum_{j=1}^{l}2(Y_{ij}-\bar{Y}_i)(\bar{Y}_i-\bar{Y})^T+\sum_{j=1}^{l}(\bar{Y}_i-\bar{Y})(\bar{Y}_i-\bar{Y})^T\right)
\end{aligned}
$$

Beweis (fortgeführt)

$$
\begin{aligned}
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i) \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} \left(Y_{ij} - \frac{1}{l} \sum_{j=1}^{l} Y_{ij} \right) \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + 2 \left(\sum_{j=1}^{l} Y_{ij} - \sum_{j=1}^{l} Y_{ij} \right) (\bar{Y}_i - \bar{Y})^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} \left(\sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T + l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T \right) \\
&= \sum_{i=1}^{k} l(\bar{Y}_i - \bar{Y})(\bar{Y}_i - \bar{Y})^T + \sum_{i=1}^{k} \sum_{j=1}^{l} (Y_{ij} - \bar{Y}_i)(Y_{ij} - \bar{Y}_i)^T \\
&= B + W.
\end{aligned}
$$

Modellevaluation

```r
sos = function(Y){

  # Diese Funktion evaluiert die Kreuzproduktsummenmatrizen T,B,W einer
  # einfaktoriellen Varianzanalyse basierend auf einen m x l x k Datensatz Y.
  #
  # Input
  #     Y        : m x l x k Datenarray
  #
  # Output
  #     $Y_bar   : m x 1 Gesamtmittelwert
  #     $Y_bar_i : m x k Gruppenmittelwerte
  #     $T       : m x m Total Sum of Squares Matrix
  #     $B       : m x m Between Group Sum of Squares Matrix
  #     $W       : m x m Within  Group  Sum of Squares Matrix
  # ------------------------------------------------------------------------
  d        = dim(Y)                                    # Datensatzdimensionen
  m        = d[1]                                      # Datendimension
  l        = d[2]                                      # Anzahl Datenpunkte pro Gruppe
  k        = d[3]                                      # Anzahl Gruppen

  # Mittelwerte
  Y_bar_i  = matrix(apply(Y,3,rowMeans), nrow = m)     # Gruppenstichprobenmittel
  Y_bar    = matrix(rowMeans(Y_bar_i)  , nrow = m)     # Gesamtstichprobenmittel

  # Totale sum of Squares Matrix
  T = matrix(rep(0,m*m), nrow = m)
  for(i in 1:k){
      for(j in 1:l){
          T = T + (Y[,j,i] - Y_bar) %*% t(Y[,j,i] - Y_bar)}}

  # Between Sum of Squares Matrix
  B = matrix(rep(0,m*m), nrow = m)
  for(i in 1:k){
    B = B + l*(Y_bar_i[,i] - Y_bar) %*% t(Y_bar_i[,i] - Y_bar)}

  # Within Sum of Squares Matrix
  W = matrix(rep(0,m*m), nrow = m)
  for(i in 1:k){
      for(j in 1:l){
          W = W + (Y[,j,i] - Y_bar_i[,i]) %*% t(Y[,j,i] - Y_bar_i[,i])}}

  # Outputspezifikation
  return(list(Y_bar_i = Y_bar_i, Y_bar = Y_bar, T = T, B = B, W = W))}
```

Überblick zu Teststatistiken und ihren Verteilungen

Basierend auf der Kreuzproduktsummenmatrixzerlegung $T = B + W$ wurden eine Reihe von Teststatisken für Hypothesentests der Nullhypothese $H_0 : \mu_1 = \cdots \mu_k$ vorgeschlagen. Wir betrachten hier (nur)

$$\text{Wilks' } \Lambda = \frac{|W|}{|B+W|} \text{ und Pillai's } V := \text{tr}\left((B+W)^{-1}B\right) \tag{11}$$

Im Gegensatz zur T^2-Teststatistik und zur F-Teststatistik der univariaten Varianzanalyse sind die Verteilungen von Λ und V nur für bestimmte Anwendungsszenarien, d.h. kleine Werte der Datendimensionalität m und die Anzahl der Gruppen k, analytisch exakt beschreibbar und führen, wie im Fall der T^2-Teststatistik auf f-Verteilungen.

Für Anwendungsszenarien mit größeren Werten von m und oder k existieren lediglich Approximationen der Verteilungen von Λ und V, die für unendlich große Stichprobenumfänge $n \to \infty$ exakt sind und wiederum durch f-Verteilungen gegeben sind.

Anderson (2003), Kapitel 8 gibt eine Einführung in die Approximationstheorie für multivariate Modelle, das Wissen um die exakten Verteilungen der Teststatistiken um 1970 wird von Rao (1972) zusammengefasst. Im Sinne der Anwendung unterscheiden sich Testentscheidungen basierend auf exakten oder approximativen Verteilungen von Wilk's Λ und Pillai's V nicht. Zur Absicherung dieser Aussage mögen im konkreten Fall von m und k Simulationen wie im Folgenden diskutiert helfen, Unterschiede zwischen Λ und V und der Approximation ihrer Verteilungen abzuschätzen.

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- **Wilks' Λ**
- Pillai's V

Selbstkontrollfragen

Definition (Wilks' Λ)

Es seien das Modell der einfaktoriellen Varianzanalyse sowie die Between Sum of Squares Matrix B und die Within Sum of Squares Matrix W definiert wie oben. Dann ist die Wilks' Λ Teststatistik definiert als

$$\Lambda := \frac{|W|}{|B+W|}, \tag{12}$$

wobei $|\cdot|$ die Determinante bezeichnet.

Bemerkungen

- Intuitiv misst Λ das Verhältnis von Residualvariabilität und Gesamtvariabilität.
- Ohne Beweis halten wir fest, dass $\Lambda \in [0,1]$
- Für $\bar{Y}_1 = \cdots = \bar{Y}_p = \bar{Y}$ gilt $B = 0_{mm}$ und damit $\Lambda = 1$.
- Für steigende Unterschiede zwischen den $\bar{Y}_i$ nimmt $|B+W|$ gegenüber $|W|$ zu, Λ also ab.
- Kleine Werte von Λ sprechen also für eine Abweichung von der Nullhypothese.

Theorem (Eigenwertform von Wilks' Λ)

Es seien das Modell der einfaktoriellen Varianzanalyse, die Between Sum of Squares Matrix B, die Within Sum of Squares Matrix W und Wilks' Λ definiert wie oben. Weiterhin seien $\lambda_1, \ldots, \lambda_s$ die Eigenwerte von $W^{-1}B$. Dann gilt

$$\Lambda = \prod_{i=1}^{s} \frac{1}{1+\lambda_i} \tag{13}$$

Bemerkungen

- Wir verzichten auf einen Beweis.
- Die Matrix $W^{-1}B$ ist das multivariate Analogon zu $\frac{\text{SQB}}{\text{SQB}}$.

Theorem (Spezielle H_0 Verteilungen von Wilks' Λ Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Wilks' Λ definiert wie oben und es gelte außerdem

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m \tag{14}$$

Dann sind für die in den ersten beiden Tabellenspalten aufgeführten Spezialfällen die in der dritten Tabellenspalte aufgeführten Statistiken f-Zufallsvariablen und zwar mit den in der vierten Tabellenspalte aufgeführten Parametern.

Datendimension m	Gruppenanzahl k	Statistik	f-Verteilungsparameter
Beliebig	2	$\frac{1-\Lambda}{\Lambda}\,\frac{n-k-m+1}{m}$	$m, n-k-m+1$
Beliebig	3	$\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\,\frac{n-k-m+1}{m}$	$2m, 2(n-k-m+1)$
1	Beliebig	$\frac{1-\Lambda}{\Lambda}\,\frac{n-k}{k-1}$	$k-1, n-k$
2	Beliebig	$\frac{1-\sqrt{\Lambda}}{\sqrt{\Lambda}}\,\frac{n-k-1}{k-1}$	$2(k-1), 2(n-k-1)$

Bemerkungen

- Die Verteilungen gehen zurück auf Wilks (1932).

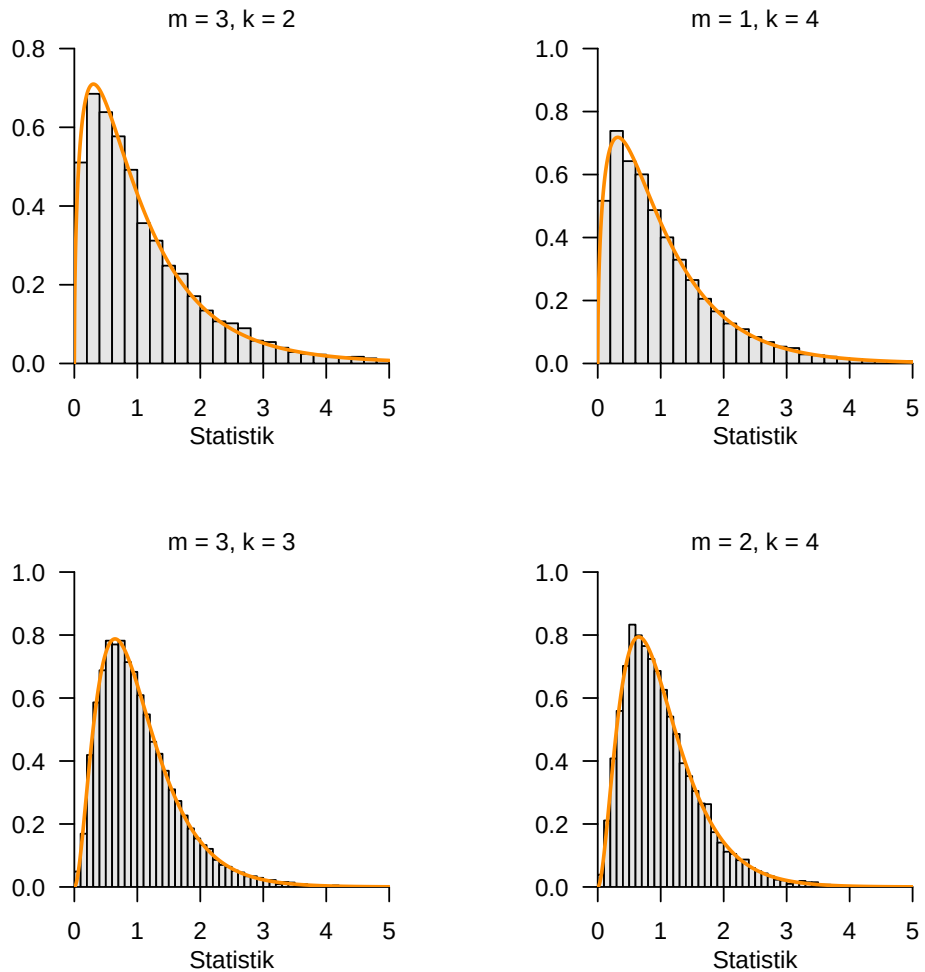
Modellevaluation mit der Wilks' Λ Statistik

Simulation spezieller H_0 Verteilungen von Wilks' Λ Transformationen

```
# Szenarioparameter
nsm = 1e4                                          # Datensimulationsanzahl
M   = c(3,3,1,2)                                   # Datendimension
K   = c(2,3,4,4)                                   # Gruppenanzahl
l   = 15                                           # Datenpunkte pro Gruppe
N   = l*K                                          # Gesamtanzahl Datenpunkte

# Szenariensimulationen
library(MASS)                                      # R Paket für multivariate Normalverteilungen
nsc = length(M)                                    # Szenarienanzahl
P   = matrix(rep(NaN,nsm*nsc), ncol = nsc)         # Statistik ("Prüfgröße") Array
for(sc in 1:nsc){                                  # Szenarioiterationen

    # Modellparameter
    m     = M[sc]                                  # Datendimension
    k     = K[sc]                                  # Gruppenanzahl
    n     = N[sc]                                  # Gesamtanzahl Datenpunkte
    mu_i  = matrix(rep(1,m), nrow = m)             # Identische Gruppenerwartungswertparameter bei H_0
    Sigma = diag(m)                                # Identische Gruppenkovarianzmatrixparameter

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))              # Datenanarrayinitialisierung
        for(i in 1:k){                             # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}    # Datensimulation

        # Analyse
        S        = sos(Y)                          # Stichprobenmittel und Sum of Squares Matrizen
        L        = det(S$W)/det(S$W + S$B)         # Wilks' Lambda

        # Szenarioabhängige Statistik ("Prüfgröße")
        if     (sc == 1){p = ((1-L)/L)*((n-k-m+1)/m)}
        else if(sc == 2){p = ((1-sqrt(L))/sqrt(L))*((n-k-m+1)/m)}
        else if(sc == 3){p = ((1-L)/L)*((n-k)/(k-1))}
        else if(sc == 4){p = ((1-sqrt(L))/sqrt(L))*((n-k-1)/(k-1))}

        # Statistikrealisation
        P[sm,sc] = p }}
```

Modellevaluation mit der Wilks' Λ Statistik

Simulation spezieller H_0 Verteilungen von Wilks' Λ Transformationen



Theorem (Approximative H_0 Verteilungen von Wilks' Λ Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Wilks' Λ definiert wie oben und es gelte außerdem die Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m \tag{15}$$

Dann ist die Statistik

$$\tau := \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \frac{\nu_2}{\nu_1} \tag{16}$$

mit

$$\nu_1 := m(k-1) \text{ und } \nu_2 := wt - \frac{1}{2}(m(k-1) - 2) \tag{17}$$

sowie

$$w := n - 1 - \frac{1}{2}(m + k) \text{ und } t := \sqrt{\frac{m^2(k-1)^2 - 4}{m^2 + (k-1)^2 - 5}} \tag{18}$$

approximativ f-verteilt mit Freiheitsgradparametern ν_1 und ν_2.

Bemerkungen

- Die Approximation geht zurück auf Rao (1951).

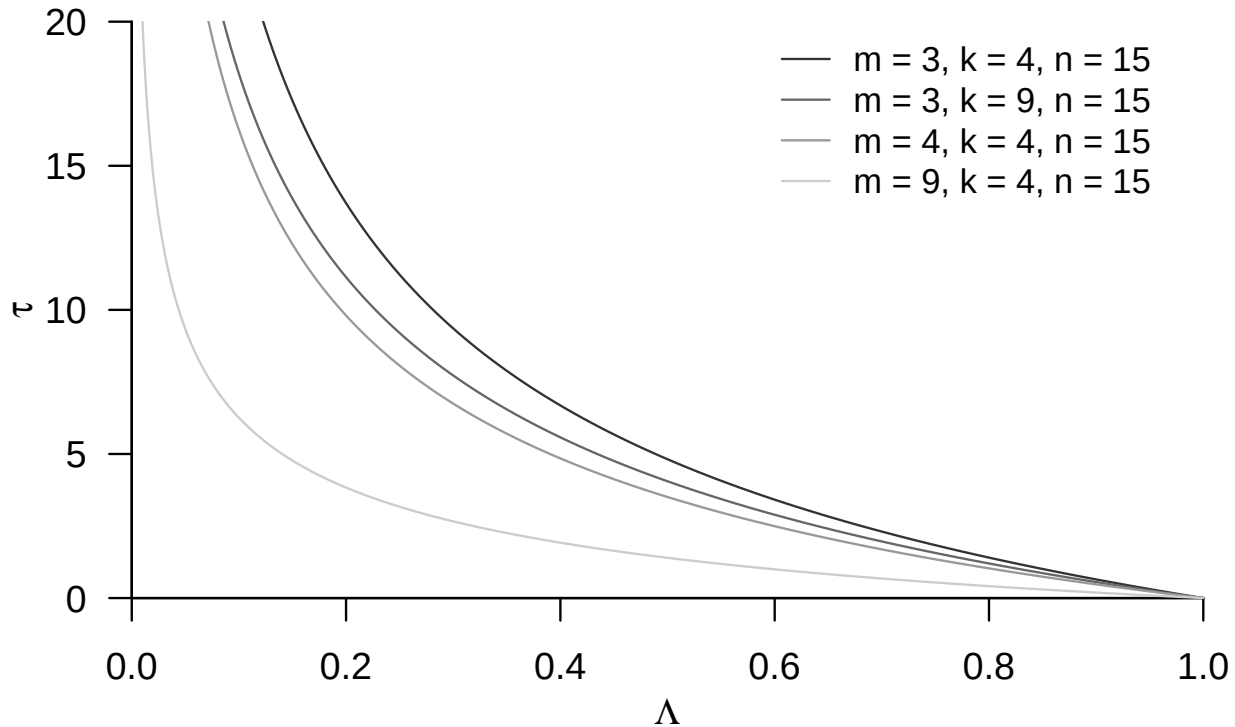
Modellevaluation mit der Wilks' Λ Statistik

Simulation approximativer H_0 Verteilungen von Wilks' Λ Transformationen

```
# Szenarioparameter
library(MASS)                                          # # R Paket für multivariate Normalverteilungen
nsm = 1e4                                              # Datensimulationsanzahl
M   = c(3,3,4,9)                                       # Datendimension
K   = c(4,9,4,4)                                       # Gruppenanzahl
l   = 15                                               # Datenpunkte pro Gruppe
N   = l*K                                              # Gesamtanzahl Datenpunkte
nsc = length(M)                                        # Szenarienanzahl
TAU = matrix(rep(NaN,nsm*nsc), ncol = nsc)             # Statistik Array
NU  = matrix(rep(NaN,2*nsc)  , ncol = nsc)             # Parameter Array
WL  = seq(0,1,len = 1e3)                               # Wilk's Lambda Values
TL  = matrix(rep(NaN,length(WL)*nsc), nrow = nsc)      # \tau(\Lambda) Array
for(sc in 1:nsc){                                      # Szenarioiterationen

    # Modellparameter
    m       = M[sc]                                    # Datendimension
    k       = K[sc]                                    # Gruppenanzahl
    n       = N[sc]                                    # Gesamtanzahl Datenpunkte
    mu_i    = matrix(rep(1,m), nrow = m)               # Identische Gruppenerwartungswertparameter bei H_0
    Sigma   = diag(m)                                  # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyse Parameter
    w       = n-1-(1/2)*(m+k)                          # w
    t       = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5))    # t
    nu_1    = m*(k-1)                                  # \nu_1
    nu_2    = w*t-(1/2)*(m*(k-1)-2)                    # \nu_2
    TL[sc,] = ((1-WL^(1/t))/WL^(1/t))*(nu_2/nu_1)      # \tau(\Lambda)

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation

        # Varianzanalyse
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        L          = det(S$W)/det(S$W + S$B)           # Wilks' Lambda
        tau        = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1) # Statistik
        TAU[sm,sc] = tau                               # Statistik
        NU[1,sc]   = nu_1                              # \nu_1
        NU[2,sc]   = nu_2}}                            # \nu_2
```

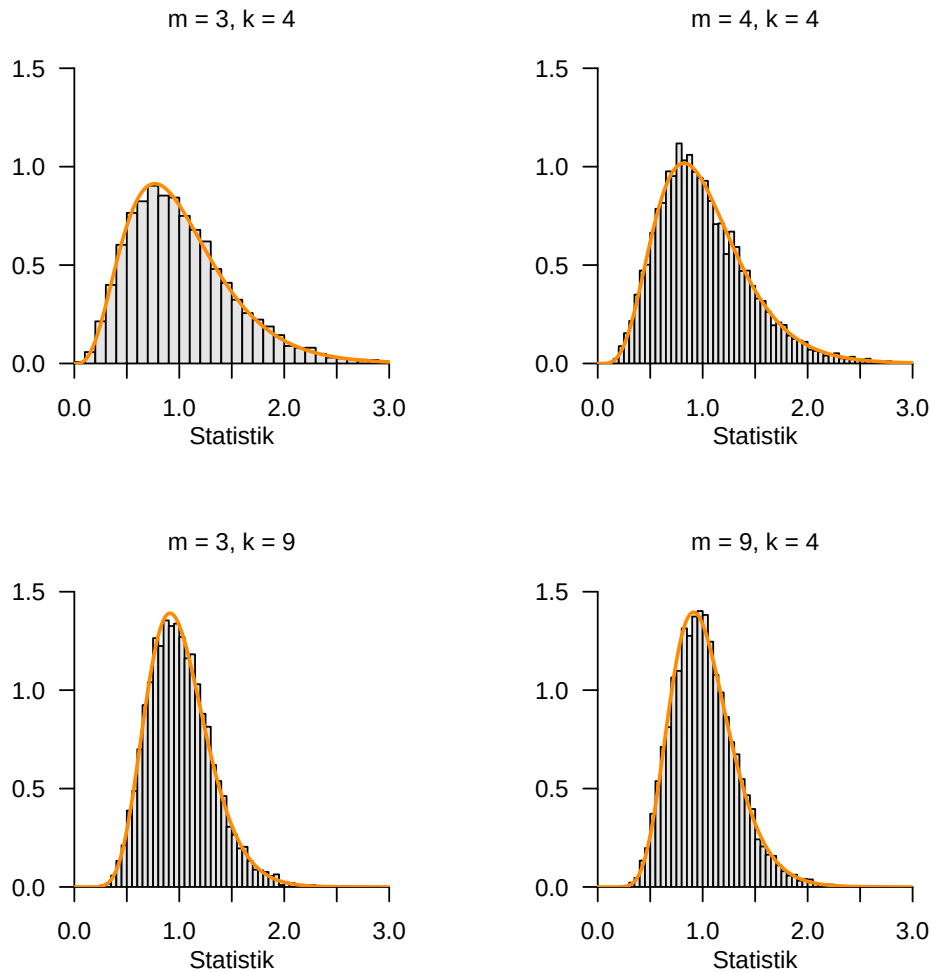
Modellevaluation mit der Wilks' Λ Statistik

τ als Funktion von Λ : $\Lambda \downarrow \Rightarrow \tau \uparrow$



Modellevaluation mit der Wilks' Λ Statistik

Simulation approximativer H_0 Verteilungen von Wilks' Λ Transformationen



Definition (Wilks' Λ-basierter Test, Testumfangkontrolle, p-Wert)

Es seien das Modell der einfaktoriellen Varianzanalyse und die Wilks' Λ basierte Teststatistik τ mit Verteilungsparametern ν_1, ν_2 wie oben definiert. Weiterhin sei der kritische Wert-basierte Test

$$\phi(Y) := 1_{\{\tau > k\}} := \begin{cases} 1 & \tau > k \\ 0 & \tau \leq k \end{cases} \tag{19}$$

definiert. ϕ ist genau dann ein Level-α_0-Test mit Testumfang α, wenn

$$k := k_{\alpha_0} := F^{-1}(1 - \alpha_0; \nu_1, \nu_2) \tag{20}$$

ist und der p-Wert einer realisiertern τ-Teststatistik $\tilde{\tau}$ ergibt sich zu

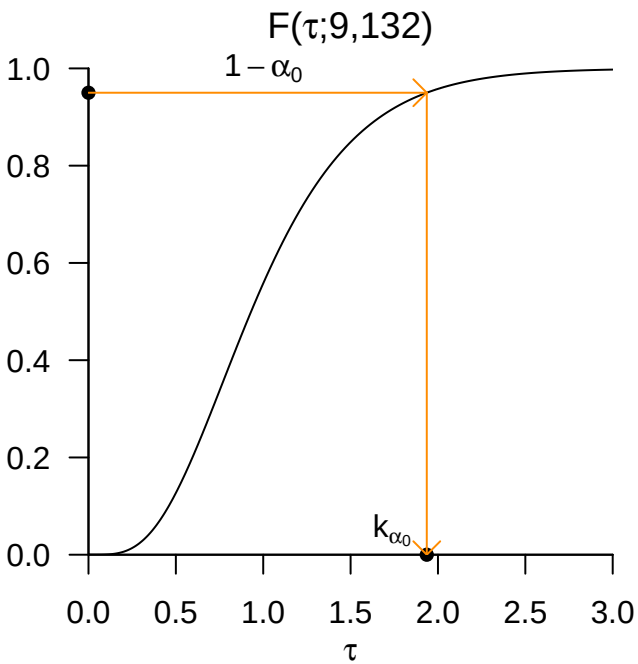
$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{21}$$

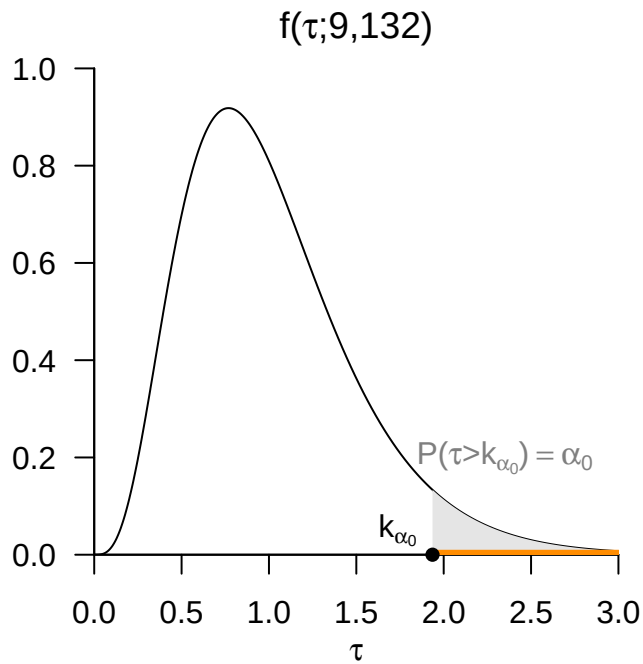
Bemerkungen

- Ein Beweis kann in Analogie zum Einstichproben-T^2-Test Fall geführt werden.
- Wir validieren die Testumfangkontrolle mithilfe von k_{α_0} in untenstehender Simulation.

Testumfangkontrolle

Wahl von k_{α_0} bei $m = 3, k = 4, n = 15 \Rightarrow \nu_1 = 9, \nu_2 = 132$ und $\alpha_0 = 0.05$.





Modellevaluation mit der Wilks' Λ Statistik

Testumfangkontrolle

```
# Szenarioparameter
nsm       = 1e4                                       # Datensimulationsanzahl
M         = c(3,3,4,9)                                # Datendimension
K         = c(4,9,4,4)                                # Gruppenanzahl
l         = 15                                        # Datenpunkte pro Gruppe
N         = l*K                                       # Gesamtanzahl Datenpunkte
alpha_0   = 0.05                                      # \alpha_0
nsc       = length(M)                                  # Szenarienanzahl
TAU       = matrix(rep(NaN,nsm*nsc), ncol = nsc)      # Statistik Array
NU        = matrix(rep(NaN,2*nsc)  , ncol = nsc)      # Parameter Array
KA        = rep(NaN, nsc)                             # Kritische Werte
PHI       = matrix(rep(0,nsm*nsc)  , ncol = nsc)      # Testarray

# Simulationen
for(sc in 1:nsc){                                     # Szenarioiterationen

    # Modellparameter
    m         = M[sc]                                 # Datendimension
    k         = K[sc]                                 # Gruppenanzahl
    n         = N[sc]                                 # Gesamtanzahl Datenpunkte
    mu_i      = matrix(rep(1,m), nrow = m)            # Identische Gruppenerwartungswertparameter bei H_0
    Sigma     = diag(m)                               # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyse Parameter
    w         = n-1-(1/2)*(m+k)                       # w
    t         = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5)) # t
    nu_1      = m*(k-1)                               # \nu_1
    nu_2      = w*t-(1/2)*(m*(k-1)-2)                 # \nu_2
    KA[sc]    = qf(1-alpha_0,nu_1,nu_2)               # kritischer Wert

    # Datensimulationen
    for(sm in 1:nsm){
        Y     = array(dim = c(m,l,k))                 # Datenanarrayinitialisierung
        for(i in 1:k){                                # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}       # Datensimulation
        S         = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        L         = det(S$W)/det(S$W + S$B)           # Wilks' Lambda
        tau       = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1) # Statistik
        PHI[sm,sc] = tau > KA[sc]}}                   # Test
```

```
> Kritische Werte       :  1.95 1.55 1.82 1.57
> Geschätzte Testumfänge:  0.0526 0.0477 0.0513 0.0502
```

Praktisches Vorgehen

- Man nimmt an, dass ein vorliegender Datensatz von k Gruppendatensätzen $y_{11}, ..., y_{1l}$, $y_{21}, ..., y_{2l}$, $\cdots$, $y_{k1}, ..., y_{kl}$ Realisationen von $Y_{ij} \sim N(\mu_i, \Sigma)$ mit unbekannten Parametern $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.
- Man möchte entscheiden ob $H_0 : \mu_1 = \cdots = \mu_k$ eher zutrifft oder eher nicht.
- Man wählt ein Signifikanzniveau α_0 und bestimmt den zugehörigen Freiheitsgradparameter-abhängigen kritischen Wert k_{α_0}. Zum Beispiel gilt bei Wahl von $\alpha_0 := 0.05, m = 3, k = 4, l = 15$ und somit $n = 60$ sowie $\nu_1 = 9, \nu_2 = 132$, dass $k_{\alpha_0} = F^{-1}(1 - 0.05; 9, 132) \approx 1.95$ ist.
- Anhand der Wilk's Λ Statistik sowie m, k und n berechnet man man den realisierten Wert der τ-Teststatistik, den wir hier mit $\tilde{\tau}$ bezeichnen.
- Wenn $\tilde{\tau}$ größer als k_{α_0} ist, lehnt man die Nullhypothese ab, andernfalls nicht.
- Die oben entwickelte Theorie garantiert dann, dass man im Mittel in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.
- Schließlich ergibt sich der assoziierte p-Wert der realisiertern τ-Teststatistik $\tilde{\tau}$ zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{22}$$

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit R's `lm()` und `Manova()`

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(car)
D        = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)
model    = lm(cbind(D$X1,D$X2) ~ D$Gruppe, D)      # Modellspezifikation
Manova(model, test.statistic = "Wilks")            # Einfaktorielle Varianzanalyse
```

```
>
> Type II MANOVA Tests: Wilks test statistic
>          Df test stat approx F num Df den Df  Pr(>F)
> D$Gruppe  2      0.61     5.76      4     82 0.00039 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit sos()

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library("foreign")
D      = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datepräprozessierung
m      = 2                                          # Datendimension von Interesse
k      = 3                                          # Anzahl Gruppen
l      = 15                                         # Anzahl Datenpunkte pro Gruppe
n      = k*l                                        # Gesamtdatenpunktanzahl
Y      = array(dim = c(m,l,k))                      # Datenarrayinitialisierung
Y[,,1] = rbind(D$X1[D$Gruppe == "ungenügend"],      # Y_{1j_1} Intelligenztestscore
               D$X2[D$Gruppe == "ungenügend"])      # Y_{1j_2} Mathematiktestscore
Y[,,2] = rbind(D$X1[D$Gruppe == "befriedigend"],    # Y_{2j_1} Intelligenztestscore
               D$X2[D$Gruppe == "befriedigend"])    # Y_{2j_2} Mathematiktestscore
Y[,,3] = rbind(D$X1[D$Gruppe == "gut"],             # Y_{3j_1} Intelligenztestscore
               D$X2[D$Gruppe == "gut"])             # Y_{3j_2} Mathematiktestscore

# Einfaktorielle Varianzanalyse
S      = sos(Y)                                     # Sum of Squares Matrizen
L      = det(S$W)/det(S$W + S$B)                    # Wilks' Lambda
w      = n-1-(1/2)*(m+k)                            # w
t      = sqrt((m^2*(k-1)^2-4)/(m^2+(k-1)^2-5))      # t
nu_1   = m*(k-1)                                    # \nu_1
nu_2   = w*t-(1/2)*(m*(k-1)-2)                      # \nu_2
tau    = ((1-L^(1/t))/L^(1/t))*(nu_2/nu_1)          # Teststatistik
P      = 1-pf(tau,nu_1,nu_2)                        # Überschreitungswahrscheinlichkeit

# Ausgabe
cat("Wilks' Lambda     "   , L,
    "\ntau               ", tau,
    "\nnu_1              ", nu_1,
    "\nnu_2              ", nu_2,
    "\nP(tau > tau_tilde)", P)
```

```
> Wilks' Lambda      0.61
> tau                5.76
> nu_1               4
> nu_2               82
> P(tau > tau_tilde) 0.000392
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- **Pillai's** V

Selbstkontrollfragen

Definition (Pillai's V Statistik)

Es seien das Modell der einfaktoriellen Varianzanalyse sowie die Between Sum of Squares Matrix B und die Within Sum of Squares Matrix W definiert wie oben. Dann ist die *Pillai's V Statistik* definiert als

$$V := \text{tr}\left((B+W)^{-1}B\right) \tag{23}$$

wobei tr$(\cdot)$ die Spur, also die Summe der Diagonalelemente, einer Matrix bezeichnet.

Bemerkungen

- Die Pillai's V Statistik betrachtet die Diagonalelemente von $T^{-1}B$
- Ein hoher Wert von V spricht also für einen großen Anteil der Between-Varianz an der Gesamtvarianz.
- Für einen hohen Wert von V würde man also die Nullhypothese $B = 0_{mm}$ verwerfen.

Theorem (Eigenwertform der Pillai's V Statistik)

Es seien das Modell der einfaktoriellen Varianzanalyse, die Between Sum of Squares Matrix B, die Within Sum of Squares Matrix W und die Pillai's V Statistik definiert wie oben. Weiterhin seien $\lambda_1, ..., \lambda_s$ die Eigenwerte von $W^{-1}B$. Dann gilt

$$V = \sum_{i=1}^{s} \frac{\lambda_i}{1 + \lambda_i} \tag{24}$$

footnotesize Bemerkungen

- Wir verzichten auf einen Beweis.
- Die Matrix $W^{-1}B$ ist das multivariate Analogon zu $\frac{\text{SQB}}{\text{SQW}}$

Theorem (Approximative H_0 Verteilungen von Pillai's V Transformationen)

Es seien das Modell der einfaktoriellen Varianzanalyse und Pillai's V definiert wie oben und es gelte außerdem die Nullhypothese

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_p \text{ bzw. } H_0 : \alpha_1 = \alpha_2 = \cdots = \alpha_p = 0_m. \tag{25}$$

Weiterhin seien die Parameter

$$s := \min(k-1, m), t := \frac{1}{2}(|k-1-m| - 1) \text{ und } w := \frac{1}{2}(n-k-m-1) \tag{26}$$

definiert. Dann ist

$$\tau = \frac{(2w+s+1)V}{(2t+s+1)(s-V)} \tag{27}$$

approximativ f-verteilt mit Freiheitsgradparametern

$$\nu_1 := s(2t+s+1) \text{ und } \nu_2 := s(2w+s+1) \tag{28}$$

Bemerkungen

- Die Approximation geht zurück auf Pillai (1955).

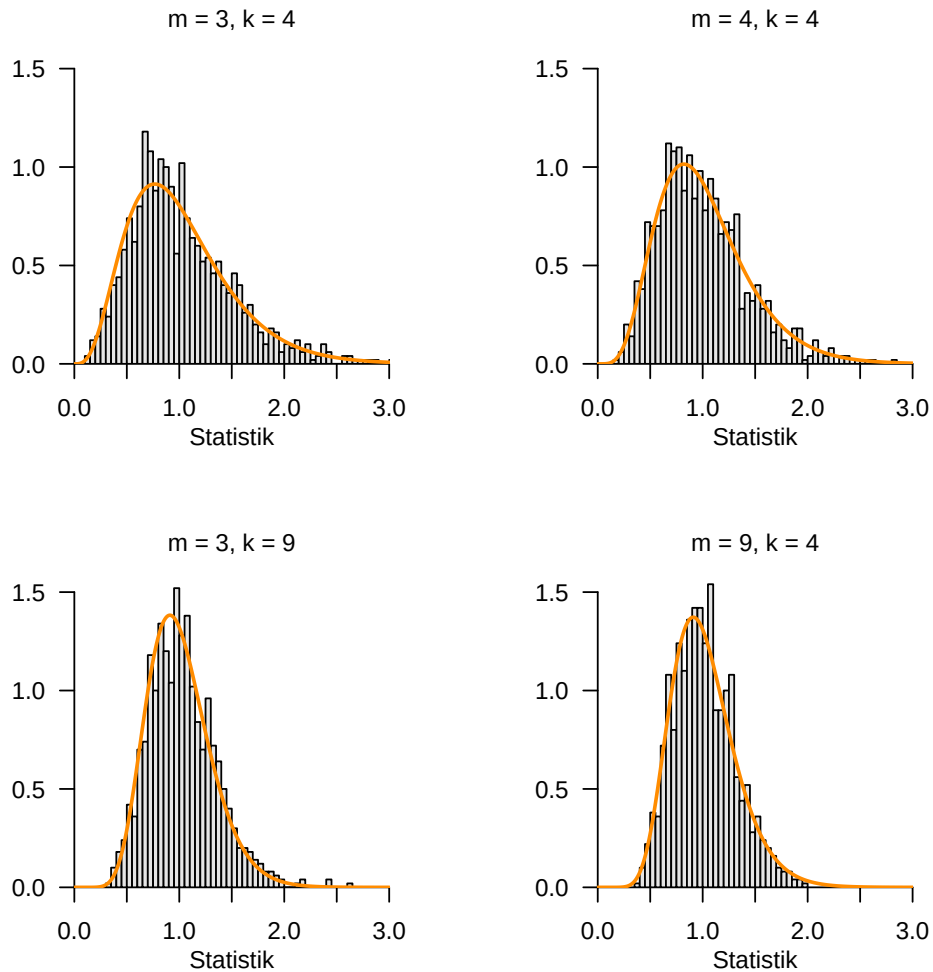
Modellevaluation mit der Pillai's V Statistik

Simulation approximativer H_0 Verteilungen von Pillai's V Transformatione

```
# Szenarioparameter
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrixalgebra
nsm = 1e1                                              # Datensimulationsanzahl
M   = c(3,3,4,9)                                       # Datendimension
K   = c(4,9,4,4)                                       # Gruppenanzahl
l   = 15                                               # Datenpunkte pro Gruppe
N   = l*K                                              # Gesamtanzahl Datenpunkte
nsc = length(M)                                        # Szenarienanzahl
TAU = matrix(rep(NaN,nsm*nsc), ncol = nsc)             # Statistik Array
NU  = matrix(rep(NaN,2*nsc)  , ncol = nsc)             # Parameter Array
for(sc in 1:nsc){                                      # Szenarioiterationen


    # Modellparameter
    m     = M[sc]                                      # Datendimension
    k     = K[sc]                                      # Gruppenanzahl
    n     = N[sc]                                      # Gesamtanzahl Datenpunkte
    mu_i  = matrix(rep(1,m), nrow = m)                 # Identische Gruppenerwartungswertparameter bei H_0
    Sigma = diag(m)                                    # Identische Gruppenkovarianzmatrixparameter

    # Varianzanalyseparameter
    s     = min(k-1,m)                                 # s
    t     = (1/2)*(abs(k-1-m)-1)                       # t
    w     = (1/2)*(n-k-m-1)                            # w
    nu_1  = s*(2*t+s+1)                                # \nu_1
    nu_2  = s*(2*w+s+1)                                # \nu_2

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation

        # Varianzanalyse
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        V          = sum(diag(inv(S$B+S$W) %*% S$B))   # Pillai's V
        tau        = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))   # Statistik
        TAU[sm,sc] = tau                               # Statistik
        NU[1,sc]   = nu_1                              # \nu_1
        NU[2,sc]   = nu_2}}                            # \nu_2
```

Modellevaluation mit der Pillai's V Statistik

Simulation approximativer H_0 Verteilungen von Pillai's V Transformationen



Definition (Pillai's V -basierter Test, Testumfangkontrolle, p-Wert)

Es seien das Modell der einfaktoriellen Varianzanalyse und die Pillai's V basierte Teststatistik τ mit Verteilungsparametern ν_1, ν_2 wie oben definiert. Weiterhin sei der kritische Wert-basierte Test

$$\phi(Y) := 1_{\{\tau > k\}} := \begin{cases} 1 & \tau > k \\ 0 & \tau \leq k \end{cases} \tag{29}$$

definiert. ϕ ist genau dann ein Level-α_0-Test mit Testumfang α, wenn

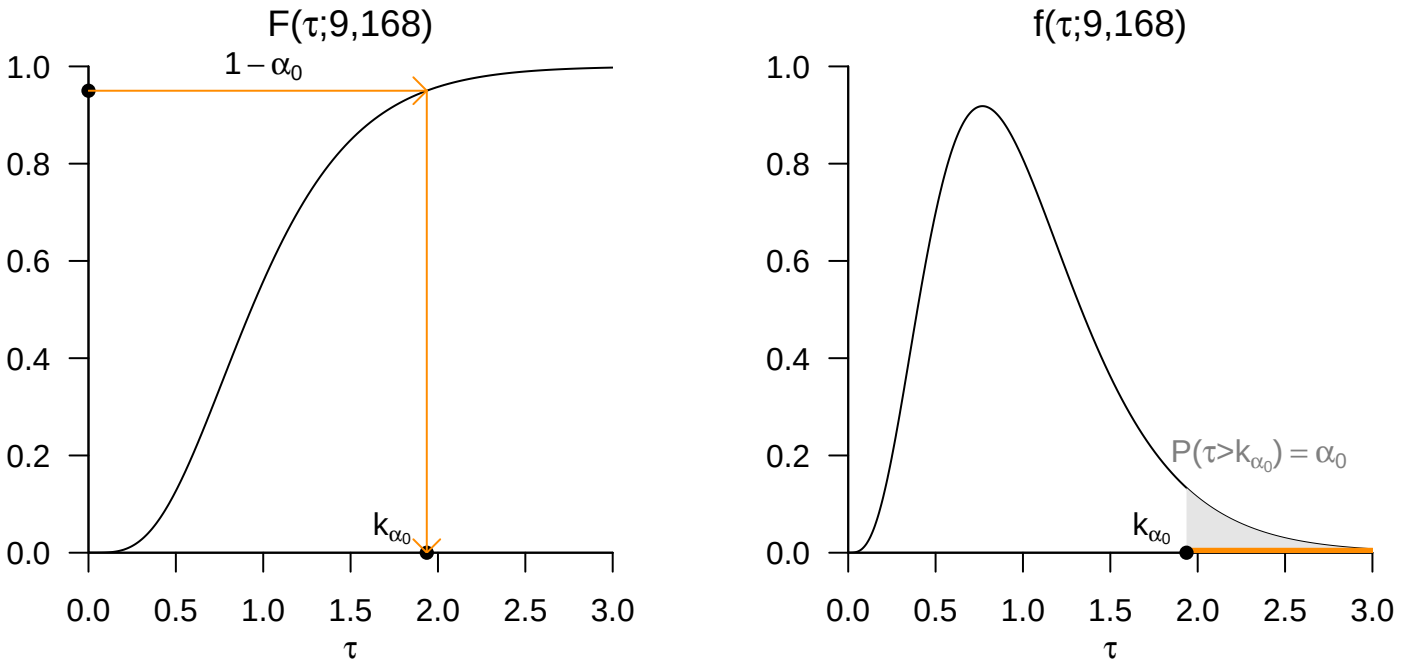
$$k := k_{\alpha_0} := F^{-1}(1 - \alpha_0; \nu_1, \nu_2) \tag{30}$$

ist und der p-Wert einer realisiertern τ-Teststatistik $\tilde{\tau}$ ergibt sich zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{31}$$

Testumfangkontrolle

Wahl von k_{α_0} bei $m = 3, k = 4, n = 15 \Rightarrow \nu_1 = 9, \nu_2 = 168$ und $\alpha_0 = 0.05$.



Modellevaluation mit der Pillai's V Statistik

Testumfangkontrolle

```
# Szenarioparameter
library(MASS)                                          # R Paket für multivariate Normalverteilungen
library(matlib)                                        # R Paket für Matrixalgebra
nsm     = 1e3                                          # Datensimulationsanzahl
M       = c(3,3,4,9)                                   # Datendimension
K       = c(4,9,4,4)                                   # Gruppenanzahl
l       = 15                                           # Datenpunkte pro Gruppe
N       = l*K                                          # Gesamtanzahl Datenpunkte
alpha_0 = 0.05                                         # Signifikanzlevel
nsc     = length(M)                                    # Szenarienanzahl
KA      = rep(NaN, nsc)                                # kritische Werte
PHI     = matrix(rep(0,nsm*nsc), ncol = nsc)           # Testarray

# Szenariosimulationen
for(sc in 1:nsc){                                      # Szenarioiterationen

    # Modell- und Varianzanalyseparameter
    m      = M[sc]                                     # Datendimension
    k      = K[sc]                                     # Gruppenanzahl
    n      = N[sc]                                     # Gesamtanzahl Datenpunkte
    mu_i   = matrix(rep(1,m), nrow = m)                # Identische Gruppenerwartungswertparameter bei H_0
    Sigma  = diag(m)                                   # Identische Gruppenkovarianzmatrixparameter
    s      = min(k-1,m)                                # s
    t      = (1/2)*(abs(k-1-m)-1)                      # t
    w      = (1/2)*(n-k-m-1)                           # w
    nu_1   = s*(2*t+s+1)                               # \nu_1
    nu_2   = s*(2*w+s+1)                               # \nu_2
    KA[sc] = qf(1-alpha_0,nu_1,nu_2)                   # kritischer Wert

    # Datensimulationen
    for(sm in 1:nsm){

        # Datengeneration und Varianzanalyse
        Y     = array(dim = c(m,l,k))                  # Datenanarrayinitialisierung
        for(i in 1:k){                                 # Gruppeniterationen
            Y[,,i] =  t(mvrnorm(l,mu_i,Sigma))}        # Datensimulation
        S          = sos(Y)                            # Stichprobenmittel und Sum of Squares Matrizen
        V          = sum(diag(inv(S$B+S$W) %*% S$B))   # Pillai's V
        tau        = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))   # Statistik
        PHI[sm,sc] = tau > KA[sc]}}                    # Test
```

```
> Kritische Werte       :  1.95 1.55 1.82 1.57
> Geschätzte Testumfänge:  0.0526 0.0477 0.0513 0.0502
```

Praktisches Vorgehen

- Man nimmt an, dass ein vorliegender Datensatz von k Gruppendatensätzen $y_{11}, ..., y_{1l}$, $y_{21}, ..., y_{2l}$, $\cdots$, $y_{k1}, ..., y_{kl}$ Realisationen von $Y_{ij} \sim N(\mu_i, \Sigma)$ mit unbekannten Parametern $\mu_i \in \mathbb{R}^m, i = 1, ..., k$ und $\Sigma \in \mathbb{R}^{m \times m}$ p.d. sind.
- Man möchte entscheiden ob $H_0 : \mu_1 = \cdots = \mu_k$ eher zutrifft oder eher nicht.
- Man wählt ein Signifikanzniveau α_0 und bestimmt den zugehörigen Freiheitsgradparameter-abhängigen kritischen Wert k_{α_0}. Zum Beispiel gilt bei Wahl von $\alpha_0 := 0.05, m = 3, k = 4, l = 15$ und somit $n = 60$ sowie $\nu_1 = 9, \nu_2 = 168$, dass $k_{\alpha_0} = F^{-1}(1 - 0.05; 9, 168) \approx 1.94$ ist.
- Anhand der Pillai's V Statistik sowie m, k und n berechnet man man den realisierten Wert der τ-Teststatistik, den wir hier mit $\tilde{\tau}$ bezeichnen.
- Wenn $\tilde{\tau}$ größer als k_{α_0} ist, lehnt man die Nullhypothese ab, andernfalls nicht.
- Die oben entwickelte Theorie garantiert dann, dass man im Mittel in höchstens $\alpha_0 \cdot 100$ von 100 Fällen die Nullhypothese fälschlicherweise ablehnt.
- Schließlich ergibt sich der assoziierte p-Wert der realisiertern τ-Teststatistik $\tilde{\tau}$ zu

$$\text{p-Wert} = \mathbb{P}(\tau \geq \tilde{\tau}) = 1 - F(\tilde{\tau}; \nu_1, \nu_2) \tag{32}$$

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit R's `lm()` und `Manova()`

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(car)
D        = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)
model    = lm(cbind(D$X1,D$X2) ~ D$Gruppe, D)      # Modellspezifikation
Manova(model, test.statistic = "Pillai")           # Einfaktorielle Varianzanalyse
```

```
>
> Type II MANOVA Tests: Pillai test statistic
>          Df test stat approx F num Df den Df  Pr(>F)
> D$Gruppe  2     0.404     5.31      4     84 0.00073 ***
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modellevaluation mit der Pillai's V Statistik

Anwendungsbeispiel

Einfaktorielle Varianzanalyse mit sos()

```
# Einlesen des Datensatzes aus Rudolf & Buse (2019) Multivariate Verfahren
library(foreign)
library(matlib)
D      = read.spss(file.path(getwd(), "11_Daten", "studienerfolg.sav"), to.data.frame = T)

# Datepräprozessierung
m      = 2                                          # Datendimension von Interesse
k      = 3                                          # Anzahl Gruppen
l      = 15                                         # Anzahl Datenpunkte pro Gruppe
n      = k*l                                        # Gesamtdatenpunktanzahl
Y      = array(dim = c(m,l,k))                      # Datenarrayinitialisierung
Y[,,1] = rbind(D$X1[D$Gruppe == "ungenügend"],      # Y_{1j_1} Intelligenztestscore
               D$X2[D$Gruppe == "ungenügend"])      # Y_{1j_2} Mathematiktestscore
Y[,,2] = rbind(D$X1[D$Gruppe == "befriedigend"],    # Y_{2j_1} Intelligenztestscore
               D$X2[D$Gruppe == "befriedigend"])    # Y_{2j_2} Mathematiktestscore
Y[,,3] = rbind(D$X1[D$Gruppe == "gut"],             # Y_{3j_1} Intelligenztestscore
               D$X2[D$Gruppe == "gut"])             # Y_{3j_2} Mathematiktestscore

# Einfaktorielle Varianzanalyse
S      = sos(Y)                                     # Sum of Squares Matrizen
V      = sum(diag(inv(S$B+S$W) %*% S$B))            # Pillai's V
s      = min(k-1,m)                                 # s
t      = (1/2)*(abs(k-1-m)-1)                       # t
w      = (1/2)*(n-k-m-1)                            # w
nu_1   = s*(2*t+s+1)                                # \nu_1
nu_2   = s*(2*w+s+1)                                # \nu_2
tau    = ((2*w+s+1)*V)/((2*t+s+1)*(s-V))            # Statistik
P      = 1-pf(tau,nu_1,nu_2)                        # Überschreitungswahrscheinlichkeit

# Ausgabe
cat("Pillai's  V        "  , V,
    "\ntau                ", tau,
    "\nnu_1               ", nu_1,
    "\nnu_2               ", nu_2,
    "\nP(tau > tau_tilde) ", P)
```

```
> Pillai's  V         0.404
> tau                 5.31
> nu_1                4
> nu_2                84
> P(tau > tau_tilde)  0.000732
```

Anwendungsszenario und Datendeskription

Modellformulierung und Modellschätzung

Modellevaluation

- Wilks' Λ
- Pillai's V

Selbstkontrollfragen

Selbskontrolfragen

1. Erläutern Sie das Anwendungsszenario einer multivariaten einfaktoriellen Varianzanalyse.
2. Definieren Sie das Klassische Modell der einfaktoriellen Varianzanalyse.
3. Geben Sie das Theorem zur Kreuzproduktsummenmatrixzerlegung wieder.
4. Erläutern Sie die intuitive Bedeutung der T, B und W Matrizen der Kreuzproduktsummenmatrixzerlegung.
5. Geben Sie einen Überblick zu den Teststatistiken der einfaktoriellen Varianzanalyse und ihren Verteilungen.
6. Definieren Sie die Wilk's Λ Teststatistik.
7. Erläutern Sie die intuitive Bedeutung der Wilk's Λ Teststatistik.
8. Definieren Sie die Pillai's V Teststatistik.
9. Erläutern Sie die intuitive Bedeutung der Pillai's V Teststatistik

References

Anderson, T. W. 2003. *An Introduction to Multivariate Statistical Analysis*. 3rd ed. Wiley Series in Probability and Statistics. Hoboken, N.J: Wiley-Interscience.

Pillai, K. C. S. 1955. "Some New Test Criteria in Multivariate Analysis." *The Annals of Mathematical Statistics* 26 (1): 117–21. https://doi.org/10.1214/aoms/1177728599.

Rao, C Radhakrishna. 1951. "An Asymptotic Expansion of the Distribution of Wilk's Criterion." *Bulletin of the International Statistical Institute* 33 (2): 177–80.

———. 1972. "Recent Trends of Research Work in Multivariate Analysis." *Biometrics* 28 (1): 21.

Rudolf, Matthias, and Johannes Buse. 2020. *Multivariate Verfahren*. Göttingen: Hogrefe.

Wilks, S. S. 1932. "Certain Generalizations in the Analysis of Variance." *Biometrika* 24 (3-4): 471–94. https://doi.org/10.1093/biomet/24.3-4.471.

Multivariate Datenanalyse

MSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(12) Kanonische Korrelationsanalyse

Vorbemerkungen

Modellformulierung

Modellschätzung

Modellevaluation

Datenanalyseszenarien

UV	AV	Datenanalysemethoden
Univariat	Univariat	Korrelation, Einfache Regression,T-Tests
Univariat	Multivariat	T^2-Tests, MANOVA
Multivariat	Univariat	Multiple Korrelation, Multiple Regression, ALM, SVMs, NNs
Multivariat	Multivariat	Kanonische Korrelation, MALM, NNs

Datenanalyseszenarien

UV	AV
x_1	y_1
x_{11}	y_{11}
x_{12}	y_{12}
x_{13}	y_{13}
$\vdots$	$\vdots$
$\vdots$	$\vdots$
$\vdots$	$\vdots$
x_{1n}	y_{1n}

Korrelation

Einfache Regression

T-Tests

UV			AV
x_1	$\cdots$	x_m	y_1
x_{11}	$\cdots$	x_{m1}	y_{11}
x_{12}	$\cdots$	x_{m2}	y_{12}
x_{13}	$\cdots$	x_{m3}	y_{13}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{1n}	$\cdots$	x_{mn}	y_{1n}

Multiple Korrelation

Multiple Regression

ALM, SVMs, NNs

Datenanalyseszenarien

UV	AV		
x_1	y_1	$\cdots$	y_m
x_{11}	y_{12}	$\cdots$	y_{m1}
x_{12}	y_{13}	$\cdots$	y_{m2}
x_{13}	y_{14}	$\cdots$	y_{m3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{1n}	y_{1n}	$\cdots$	y_{mn}

T^2-Tests
MANOVA

UV			AV		
x_1	$\cdots$	x_{m_x}	y_1	$\cdots$	y_{m_y}
x_{11}	$\cdots$	x_{m_x1}	y_{11}	$\cdots$	y_{m_y1}
x_{12}	$\cdots$	x_{m_x2}	y_{12}	$\cdots$	y_{m_y2}
x_{13}	$\cdots$	x_{m_x3}	y_{13}	$\cdots$	y_{m_y3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{1n}	$\cdots$	x_{m_xn}	y_{1n}	$\cdots$	y_{m_yn}

Kanonische Korrelation
MALM, NNs

Überblick und Notation

Wir folgen in der Darstellung Mardia, Kent, and Bibby (1979), Chapter 10.

Die Datenvektoren $x_{1i}, \ldots, x_{m_x i}, i = 1, \ldots, n$ werden als u.i.v. Realisierungen eines Zufallsvektors X interpretiert.
Die Datenvektoren $y_{1i}, \ldots, y_{m_y i}, i = 1, \ldots, n$ werden als u.i.v. Realisierungen eines Zufallsvektors Y interpretiert.

Die "erste kanonische Korrelation" ist die maximale Korrelation von Linearkombinationen von X und Y; wir bezeichnen die Linearkombinationen von X und Y mit Vektoren $a \in \mathbb{R}^{m_x}$ und $b \in \mathbb{R}^{m_y}$ mit

$$\xi = a^T X = a_1 X_1 + \cdots + a_{m_x} X_{m_x} \text{ und } \upsilon = b^T Y = b_1 Y_1 + \cdots + b_{m_y} Y_{m_y} \tag{1}$$

ξ und υ sind dann als Linearkombinationen von Zufallsvariablen selbst Zufallsvariablen; Die Korrelation von ξ und υ bezeichnen wir mit $\rho(\xi, \upsilon)$

Wenn die Zufallsvektoren X als unabhängige Variable und Y als abhängige Variable interpretiert werden, dann kann $\xi = a^T X$ als "bester Prädiktor" und $\upsilon = b^T Y$ als "am besten prädizierbares Kriterium" interpretiert werden. Kanonische Korrelationsanalyse fragt damit nach Parametern $a \in \mathbb{R}^{m_x}$ und $b \in \mathbb{R}^{m_y}$ für die $\rho(\xi, \upsilon)$ maximal ist.

Für Skalare $\alpha, \beta \in \mathbb{R}$ sind die Korrelationen $\rho\left(a^T X, b^T Y\right)$ und $\rho\left(\left(\alpha a^T\right) X, \left(\beta b^T\right) Y\right)$ allerdings identisch (siehe unten). Man sucht deshalb Parameter $a \in \mathbb{R}^{m_x}$ und $b \in \mathbb{R}^{m_y}$ für die $\rho(\xi, \upsilon)$ maximal ist und für die $a^T X$ und $b^T Y$ jeweils eine Varianz von 1 haben, also $\mathbb{V}(\xi) = \mathbb{V}(\upsilon) = 1$ gilt.

Wahrscheinlichkeitstheorie

Definition (Korrelation)

Die Korrelation zweier Zufallsvariablen ξ und υ ist definiert als

$$\rho(\xi, \upsilon) := \frac{\mathbb{C}(\xi, \upsilon)}{\mathbb{S}(\xi)\mathbb{S}(\upsilon)}, \tag{2}$$

wobei

$$\mathbb{C}(\xi, \upsilon) := \mathbb{E}((\xi - \mathbb{E}(\xi))(\upsilon - \mathbb{E}(\upsilon))) \tag{3}$$

die Kovarianz von ξ und υ und

$$\mathbb{S}(\xi) := \sqrt{\mathbb{V}(\xi)} = \sqrt{\mathbb{C}(\xi, \xi)} \text{ und } \mathbb{S}(\upsilon) := \sqrt{\mathbb{V}(\upsilon)} = \sqrt{\mathbb{C}(\upsilon, \upsilon)} \tag{4}$$

die Standardabweichungen von ξ und υ, respektive, bezeichnen.

Bemerkungen

- $\rho \in [-1, 1]$.
- Wenn ξ und υ unabhängig sind, dann gilt $\rho(\xi, \upsilon) = 0$.
- $\rho(\xi, \upsilon)$ misst dem Grad des linear-affinen Zusammenhangs $\upsilon = a\xi + b$.

Wahrscheinlichkeitstheorie

Theorem (Kovarianz und Korrelation bei linear-affinen Transformationen)

$\rho(\xi, \upsilon)$ sei die Korrelation von ξ und υ und es seien $\alpha, \beta, \gamma, \delta \in \mathbb{R}$. Dann gelten

$$\mathbb{C}(\alpha\xi + \beta, \gamma\upsilon + \delta) = \alpha\beta\mathbb{C}(\xi, \upsilon) \tag{5}$$

und

$$\rho(\alpha\xi + \beta, \gamma\upsilon + \delta) = \rho(\xi, \upsilon). \tag{6}$$

Bemerkung

- Die Varianzen von $a^T X$ und $b^T y$ und die Varianzen von Linearkombinationen von X und Y mit beliebigen skalaren Vielfachen von a und b sind im Sinne der ersten Aussage zur Kovarianz verschieden.
- Insbesondere gilt einen m_x-dimensionalen Zufallsvektoren X und einen m_y-dimensionalen Zufallsvektor Y, $a \in \mathbb{R}^{m_x}$, $b \in \mathbb{R}^{m_y}$, $\alpha, \beta \in \mathbb{R}$ und die Linearkombinationen $\xi := a^T X$ und $\upsilon := b^T Y$ also auch

$$\begin{aligned} & \rho(\xi, \upsilon) = \rho(\alpha\xi, \beta\upsilon) \\ \Leftrightarrow\ & \rho(a^T X, b^T Y) = \rho(\alpha(a^T X), \beta(b^T Y)) \\ \Leftrightarrow\ & \rho(a^T X, b^T Y) = \rho((\alpha a^T)X), (\beta b^T)Y). \end{aligned} \tag{7}$$

  Die Korrelation von $a^T X$ und $b^T y$ und die Korrelationen von Linearkombinationen von X und Y mit beliebigen skalaren Vielfachen von a und b sind also gleich.

Wahrscheinlichkeitstheorie

Beweis

Es gilt zunächst

$$\begin{aligned}
\mathbb{C}(\alpha\xi+\beta, c\upsilon+d) &= \mathbb{E}((\alpha\xi+\beta-\mathbb{E}(\alpha\xi+\beta))(\gamma\upsilon+\delta-\mathbb{E}(\gamma\upsilon+\delta))) \\
&= \mathbb{E}((\alpha\xi+\beta-\alpha\mathbb{E}(\xi)-\beta)(\gamma\upsilon+\delta-\gamma\mathbb{E}(\gamma\upsilon)-\delta)) \\
&= \mathbb{E}(\alpha(\xi-\mathbb{E}(\xi))(\gamma(\upsilon-\gamma\mathbb{E}(\upsilon))) \\
&= \mathbb{E}(\alpha\gamma((\xi-\mathbb{E}(\xi))(\upsilon-\gamma\mathbb{E}(\upsilon)))) \\
&= \alpha\gamma\mathbb{C}(\xi,\upsilon)
\end{aligned} \tag{8}$$

Also folgt

$$\begin{aligned}
\rho(\alpha\xi+\beta,\gamma\upsilon+\delta) &= \frac{\mathbb{C}(\alpha\xi+\beta,\gamma\upsilon+\delta)}{\sqrt{\mathbb{V}(\alpha\xi+\beta)}\sqrt{\mathbb{V}(\gamma\upsilon+\delta)}} \\
&= \frac{\alpha\gamma\mathbb{C}(\xi,\upsilon)}{\sqrt{\alpha^2\mathbb{V}(\xi)}\sqrt{\gamma^2\mathbb{V}(\upsilon)}} \\
&= \frac{\alpha\gamma\mathbb{C}(\xi,\upsilon)}{\alpha\mathbb{S}(\xi)c\mathbb{S}(\upsilon)} \\
&= \frac{\mathbb{C}(\xi,\upsilon)}{\mathbb{S}(\xi)\mathbb{S}(\upsilon)} \\
&= \rho(\xi,\upsilon).
\end{aligned} \tag{9}$$

Lineare Algebra

Definition (Symmetrische Quadratwurzel einer Matrix)

$A \in \mathbb{R}^{m \times m}$ sei eine invertierbare symmetrische Matrix mit positiven Eigenwerten. Dann sind für $r \in \mathbb{N}^0$ und $s \in \mathbb{N}$ die rationalen Potenzen von A einer orthonormalen Matrix $Q \in \mathbb{R}^{m \times m}$ der Eigenvektoren von A und einer Diagonalmatrix $\Lambda = \text{diag}(\lambda_i) \in \mathbb{R}^{m \times m}$ der zugehörigen Eigenwerte $\lambda_1, \ldots, \lambda_m$ von A definiert als

$$A^{r/s} = Q\Lambda^{r/s}Q^T \text{ mit } \Lambda^{r/s} = \text{diag}\left(\lambda_i^{r/s}\right). \tag{10}$$

Der Spezialfall $r := 1, s := 2$ wird als symmetrische Quadratwurzel von A bezeichnet und hat die Form

$$A^{1/2} = Q\Lambda^{1/2}Q^T \text{ mit } \Lambda^{1/2} = \text{diag}\left(\lambda_i^{1/2}\right). \tag{11}$$

Bemerkungen

- Offenbar gilt

$$\left(A^{1/2}\right)^2 = Q\Lambda^{1/2}Q^TQ\Lambda^{1/2}Q^T = Q\Lambda^{1/2}\Lambda^{1/2}Q^T = Q\Lambda Q^T = A. \tag{12}$$

- Weiterhin gilt

$$\left(A^{-1/2}\right)^2 = Q\Lambda^{-1/2}Q^TQ\Lambda^{-1/2}Q^T = Q\Lambda^{-1}Q^T = A^{-1}. \tag{13}$$

  Die vorletzte Gleichung mag überraschen, aber es gilt ja zum Beispiel

$$4^{-1/2} \cdot 4^{-1/2} = \frac{1}{\sqrt{4}} \cdot \frac{1}{\sqrt{4}} = \frac{1}{4} = 4^{-1}. \tag{14}$$

Lineare Algebra

Theorem (Eigenwerte und Eigenvektoren von Matrixprodukten)

Für $A \in \mathbb{R}^{n \times m}$ und $B \in \mathbb{R}^{m \times n}$ sind die Eigenwerte von $AB \in \mathbb{R}^{n \times n}$ und $BA \in \mathbb{R}^{m \times m}$ gleich. Weiterhin gilt, dass für einen Eigenvektor v zu einem von Null verschiedenen Eigenwert λ von AB $w := Bv$ ein Eigenvektor von BA ist.

Bemerkungen

- Für einen Beweis verweisen wir auf Mardia, Kent, and Bibby (1979), S. 468.

```
A   = matrix(1:6, nrow = 2,  byrow = T)          # Matrix A \in \mathbb{R}^{2 x 3}
B   = matrix(1:6, ncol = 2,  byrow = T)          # Matrix B \in \mathbb{R}^{3 x 2}
EAB = eigen(A %*% B)                             # Eigenanalyse von AB \in \mathbb{R}^{2 \times 2}
EBA = eigen(B %*% A)                             # Eigenanalyse von BA \in \mathbb{R}^{3 \times 3}
w   = B %*% EAB$vectors[,1]                      # Eigenvektor von BA
cat("Eigenwerte von AB :"  , EAB$values[1:2],
    "\nEigenwerte von BA :", EBA$values[1:2],
    "\nBAw mit w = Bv    :", B %*% A %*% w,
    "\nlw mit w = Bv     :", EBA$values[1] * w)
```

```
> Eigenwerte von AB : 85.6 0.421
> Eigenwerte von BA : 85.6 0.421
> BAw mit w = Bv    : -191 -417 -642
> lw mit w = Bv     : -191 -417 -642
```

Lineare Algebra

Theorem (Eigenwert und Eigenvektor eines Matrixvektorprodukts)

Für $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{p \times n}$, $a \in \mathbb{R}^m$ und $b \in \mathbb{R}^p$ gilt, dass der einzige von Null verschiedene Eigenwert von $Aab^T B \in \mathbb{R}^{n \times n}$ gleich $b^T BAa$ mit zugehörigem Eigenvektor Aa ist.

Bemerkungen

- Für einen Beweis verweisen wir auf Mardia, Kent, and Bibby (1979), S. 468.

```
A      = matrix(1:6, nrow = 2,  byrow = T)     # Matrix A \in \mathbb{R}^{2 x 3}
B      = matrix(1:8, ncol = 2,  byrow = T)     # Matrix B \in \mathbb{R}^{4 x 2}
a      = matrix(1:3, nrow = 3,  byrow = T)     # Vektor a \in \mathbb{R}^{3 x 1}
b      = matrix(1:4, nrow = 4,  byrow = T)     # Vektor b \in \mathbb{R}^{4 x 1}
EAabTB = eigen(A %*% a %*% t(b) %*% B)         # Eigenanalyse von Aab^TB \in \mathbb{R}^{4 x 4}
cat("Eigenwerte von AabTB :", EAabTB$values,
    "\nbTBAa                :", t(b) %*% B %*% A %*% a,
    "\nAa                   :", A %*% a,
    "\n(AabTB)Aa            :",(A %*% a %*% t(b) %*% B) %*% A %*% a,            # Mv
    "\n(bTBAa)Aa            :",as.vector((t(b) %*% B %*% A %*% a)) * (A %*% a)) # = \lambda v
```

```
> Eigenwerte von AabTB : 2620 0
> bTBAa                : 2620
> Aa                   : 14 32
> (AabTB)Aa            : 36680 83840
> (bTBAa)Aa            : 36680 83840
```

Lineare Algebra

Theorem (Maximierung quadratischer Formen mit Nebenbedingungen)

$A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{m \times m}$ p.d. seien symmetrische Matrizen und λ_1 sei der größte Eigenwert von $B^{-1}A$ mit assoziertem Eigenvektor $v_1 \in \mathbb{R}^m$. Dann ist λ_1 eine Lösung des Optimierungsproblems

$$\max_x x^T Ax \text{ unter der Nebenbedingung } x^T Bx = 1. \tag{15}$$

Bemerkungen

- Das Theorem ist direkt durch die kanonische Korrelationsanalyse motiviert.
- Nach Wortlaut des Theorems gilt also

$$v_1 = \arg\max_x x^T Ax \text{ unter der Nebenbedingung } x^T Bx = 1 \tag{16}$$

- Nach Wortlaut des Theorems gilt weiterhin

$$\lambda_1 = \max_x x^T Ax \text{ unter der Nebenbedingung } x^T Bx = 1. \tag{17}$$

Lineare Algebra

Beweis

$B^{1/2}$ sei die symmetrische Quadratwurzel von B und es sei

$$y := B^{1/2}x \Leftrightarrow x = B^{-1/2}y \tag{18}$$

Dann kann mit der symmetrischen Matrix

$$K := B^{-1/2}AB^{-1/2} \in \mathbb{R}^{m\times m} \tag{19}$$

das Optimierungsproblem (15) geschrieben werden als

$$\max_y y^T Ky \text{ unter der Nebenbedingung } y^T y = 1. \tag{20}$$

Dies gilt, weil

$$\max_x x^T Ax \Leftrightarrow \max_y \left(B^{-1/2}y\right)^T A \left(B^{-1/2}y\right) \Leftrightarrow \max_y y^T B^{-1/2}AB^{-1/2}y \Leftrightarrow \max_y y^T Ky \tag{21}$$

und

$$x^T Bx = 1 \Leftrightarrow y^T B^{-1/2}BB^{-1/2}y = 1 \Leftrightarrow y^T y = 1. \tag{22}$$

Weil K eine symmetrische Matrix ist, existiert die Orthonormalzerlegung (vgl. (2) Matrizen)

$$K = Q\Lambda Q^T, \tag{23}$$

wobei die Spalten der orthogonalen Matrix Q die Eigenvektoren von K und die Diagonalemente von Λ die zugehörigen Eigenwerte von K sind.

Lineare Algebra

Beweis (fortgeführt)

Mit der orthogonalen Matrix Q aus obiger Orthornomalzerlegung sei nun

$$z := Q^T y \Leftrightarrow y := Qz. \tag{24}$$

Dann kann das Optimierungsproblem (20) geschrieben werden als

$$\max_z \sum_{i=1}^{m} \lambda_i z_i^2 \text{ unter der Nebenbedingung } z^T z = 1, \tag{25}$$

weil

$$\max_y y^T K y \Leftrightarrow \max_z (Qz)^T K (Qz) \Leftrightarrow \max_z z^T Q^T Q \Lambda Q^T Q z \Leftrightarrow \max_z z^T \Lambda z \Leftrightarrow \max_z \sum_{i=1}^{m} \lambda_i z_i^2 \tag{26}$$

und

$$y^T y = 1 \Leftrightarrow (Qz)^T Qz = 1 \Leftrightarrow z^T Q^T Q z = 1 \Leftrightarrow z^T z = 1. \tag{27}$$

Lineare Algebra

Beweis (fortgeführt)

Die Eigenwerte von K seien nun absteigend sortiert, also $\lambda_1 \geq \cdots \geq \lambda_m$. Dann gilt für das Optimierungsproblem (25), dass

$$\max_z \sum_{i=1}^{m} \lambda_i z_i^2 \leq \lambda_1 , \tag{28}$$

weil

$$\max_z \sum_{i=1}^{m} \lambda_i z_i^2 \leq \max_z \sum_{i=1}^{m} \lambda_1 z_i^2 = \lambda_1 \max_z \sum_{i=1}^{m} z_i^2 = \lambda_1 \tag{29}$$

wobei sich die letzte Gleichung aus der Nebenbedingung $z^T z = 1$ ergibt. Schließlich gilt

$$\max_z \sum_{i=1}^{m} \lambda_i z_i^2 = \lambda_1 , \tag{30}$$

für $z := e_1 = (1, 0, \ldots, 0)^T$. Zusammenfassend heißt das, dass $z = e_1$ eine Lösung des Optimierungsproblem (25) ist und das λ_1 das entsprechende Maximum ist.

Lineare Algebra

Beweis (fortgeführt)

Damit ergibt sich aber sofort, dass dann

$$y = Qz = Qe_1 = q_1 \text{ und } x = B^{-1/2}q_1 \tag{31}$$

Lösungen der äquivalenten Optimierungsprobleme (20) und (15), respektive, sind. Nach Konstruktion ist q_1 ein Eigenvektor von $B^{-1/2}AB^{-1/2}$ und nach obigem Theorem zu Eigenwerten und Eigenvektoren von Matrixprodukten damit auch ein Eigenvektor von

$$B^{-1/2}B^{-1/2}A = B^{-1}A \tag{32}$$

und die zugehörigen Eigenwerte sind gleich. Damit aber folgt, dass der größte Eigenwert von $B^{-1}A$ und sein assoziierter Eigenvektor eine Lösung von

$$\max_x x^T Ax \text{ unter der Nebenbedingung } x^T Bx = 1. \tag{33}$$

ist. □

Vorbemerkungen

Modellformulierung

Modellschätzung

Modellevaluation

Überblick

Zur Entwicklung der kanonischen Korrelationsanalyse werden X und Y als

$$Z := \begin{pmatrix} X \\ Y \end{pmatrix} \tag{34}$$

zusammengefasst.

Wir nehmen durchgängig an, dass $\mathbb{E}(Z) = 0_m$ mit $m = m_x + m_y$.

Der mathematische Fokus ist auf der Kovarianzmatrix $\mathbb{C}(Z)$.

- Kovarianzen von Linearkombinationen von X und Y ergeben sich aus Matrixprodukten von $\mathbb{C}(Z)$.
- Die Matrixtheoreme aus den Vorbemerkungen können auf diese Matrixprodukte angewendet werden.

Generell wird im folgenden ein restringierter Optimierungsansatz mithilfe der Lagrangefunktion zugunsten der Eigenanalyse von Matrixprodukten supprimiert. Für den Lagrangeansatz, siehe zum Beispiel Anderson (2003), Kapitel 12.

Theorem (Kovarianzmatrizen von Zufallsvektoren)

Z sei ein m-dimensionaler Zufallsvektor mit Erwartungswert $\mathbb{E}(Z) = 0_m$ und es sei

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix} \text{ mit } \mathbb{E}(Z) := 0_m \tag{35}$$

ein $m_x + m_y$-dimensionaler Zufallsvektor und sein Erwartungswertvektor, respektive. Dann kann die $m \times m$ Kovarianzmatrix Z geschrieben werden als

$$\mathbb{C}(Z) = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{36}$$

wobei

$$\begin{aligned} \Sigma_{xx} &:= \mathbb{E}\left(XX^T\right) \in \mathbb{R}^{m_x \times m_x} \\ \Sigma_{xy} &:= \mathbb{E}\left(XY^T\right) \in \mathbb{R}^{m_x \times m_y} \\ \Sigma_{yx} &:= \mathbb{E}\left(YX^T\right) \in \mathbb{R}^{m_y \times m_x} \\ \Sigma_{yy} &:= \mathbb{E}\left(YY^T\right) \in \mathbb{R}^{m_x \times m_y} \end{aligned} \tag{37}$$

Beweis

Nach Definition der Kovarianzmatrix eines Zufallsvektors (vgl. (3) Wahrscheinlichkeitshtheorie) gilt

$$
\begin{aligned}
\mathbb{C}(Z) &= \mathbb{E}\left((Z - \mathbb{E}(Z))(Z - \mathbb{E}(Z))^T\right) \\
&= \mathbb{E}\left((Z - 0_m)(Z - 0_m)^T\right) \\
&= \mathbb{E}\left(ZZ^T\right) \\
&= \mathbb{E}\left(\begin{pmatrix} X \\ Y \end{pmatrix}\begin{pmatrix} X^T & Y^T \end{pmatrix}\right) \\
&= \mathbb{E}\left(\begin{pmatrix} XX^T & XY^T \\ YX^T & YY^T \end{pmatrix}\right) \\
&= \begin{pmatrix} \mathbb{E}\left(XX^T\right) & \mathbb{E}\left(XY^T\right) \\ \mathbb{E}\left(YX^T\right) & \mathbb{E}\left(YY^T\right) \end{pmatrix} \\
&= \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}
\end{aligned}
\tag{38}
$$

□

Theorem (Linearkombinationen von Zufallsvektorpartitionen)

Es sei

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix} \text{ mit } \mathbb{E}(X) = 0_m \text{ und } \mathbb{C}(Z) = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \tag{39}$$

ein m-dimensionaler partitionierter Zufallsvektor sowie sein Erwartungswertvektor und seine Kovarianzmatrix, respektive. Weiterhin seien für $a \in \mathbb{R}^{m_x}$ und $b \in \mathbb{R}^{m_y}$ die Zufallsvariablen

$$\xi := a^T X \text{ und } \upsilon := b^T Y \tag{40}$$

als Linearkombinationen der Komponenten von X und Y definiert. Dann gelten

(1) $\mathbb{V}(\xi) = a^T \Sigma_{xx} a$

(2) $\mathbb{V}(\upsilon) = b^T \Sigma_{yy} b$

(2) $\rho(\xi, \upsilon) = a^T \Sigma_{xy} b$, wenn $\mathbb{V}(\xi) = 1$ und $\mathbb{V}(\upsilon) = 1$.

Bemerkungen

- Die Varianz der Zufallsvariable $a^T X$ ergibt sich als "doppelte Linearkombination" von Σ_{xx}.
- Die Varianz der Zufallsvariable $b^T Y$ ergibt sich als "doppelte Linearkombination" von Σ_{yy}.
- Die Korrelation der Zufallsvariablen $a^T X$ und $b^T Y$ ergibt sich "doppellte Linearkombination" von Σ_{xy}.

Beweis von (1) und (2)

Wir betrachten zunächst die Varianz von ξ. Mit dem Varianzverschiebungssatz gilt

$$\begin{aligned}
\mathbb{V}(\xi) &= \mathbb{E}\left(\xi\xi\right) - \mathbb{E}(\xi)\mathbb{E}(\xi) \\
&= \mathbb{E}\left((a^T X)(a^T X)\right) - \mathbb{E}\left(a^T X\right)\mathbb{E}\left(a^T X\right) \\
&= \mathbb{E}\left((a^T X)(a^T X)^T\right) - \mathbb{E}\left(a^T X\right)\mathbb{E}\left(a^T X\right) \\
&= \mathbb{E}\left(a^T X X^T a\right) - \mathbb{E}\left(a^T X\right)\mathbb{E}\left(a^T X\right) \qquad (41) \\
&= a^T \mathbb{E}\left(X X^T\right) a - a^T \mathbb{E}(X) a^T \mathbb{E}(X) \\
&= a^T \mathbb{E}\left(X X^T\right) a - a^T 0_{m_x} a^T 0_{m_x} \\
&= a^T \Sigma_{xx} a.
\end{aligned}$$

Der Beweis zur Varianz von υ folgt dann analog.

Beweis von (3)

Mit der Definition der Korrelation von Zufallsvariablen und mit $\mathbb{V}(\xi) = \mathbb{V}(\upsilon) = 1$ und dem Kovarianzverschiebungssatz gilt

$$\begin{aligned}
\rho(\xi, \upsilon) &= \frac{\mathbb{C}(\xi, \upsilon)}{\sqrt{\mathbb{V}(\xi)}\sqrt{\mathbb{V}(\upsilon)}} \\
&= \frac{\mathbb{C}(\xi, \upsilon)}{\sqrt{1}\sqrt{1}} \\
&= \mathbb{C}(\xi, \upsilon) \\
&= \mathbb{E}(\xi\upsilon) - \mathbb{E}(\xi)\mathbb{E}(\upsilon) \\
&= \mathbb{E}\left((a^T X)(b^T Y)\right) - \mathbb{E}(a^T X)\mathbb{E}(b^T Y) \\
&= \mathbb{E}\left((a^T X)(b^T Y)^T\right) - \mathbb{E}(a^T X)\mathbb{E}(b^T Y) \\
&= \mathbb{E}\left(a^T X Y^T b\right) - \mathbb{E}(a^T X)\mathbb{E}(b^T Y) \\
&= a^T \mathbb{E}\left(X Y^T\right) b - a^T \mathbb{E}(X) b^T \mathbb{E}(Y) \\
&= a^T \mathbb{E}\left(X Y^T\right) b - a^T 0_{m_x} b^T 0_{m_y} \\
&= a^T \Sigma_{xy} b.
\end{aligned} \tag{42}$$

□

Definition (Kanonische Koeffizientenvektoren, Variate, Korrelationen)

Es seien

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix} \text{ mit } \mathbb{E}(Z) := 0_m \text{ und } \mathbb{C}(Z) := \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{43}$$

ein m-dimensionaler partitionierter Zufallsvektor sowie sein Erwartungswert und seine Kovarianzmatrix, respektive. Weiterhin sei

$$K := \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \in \mathbb{R}^{m_x \times m_y} \tag{44}$$

mit der Singulärwertzerlegung

$$K = A \Lambda B^T, \tag{45}$$

wobei

$$A := \begin{pmatrix} \alpha_1 & \cdots & \alpha_k \end{pmatrix} \in \mathbb{R}^{m_x \times m_y} \text{ und } B := \begin{pmatrix} \beta_1 & \cdots & \beta_k \end{pmatrix} \in \mathbb{R}^{m_y \times m_y} \tag{46}$$

die orthogonalen Matrix der Eigenvektoren von KK^T und die orthogonale Matrix der Eigenvektoren von $K^T K$, respektive, bezeichnen und

$$\Lambda := \text{diag} \left(\lambda_1^{1/2}, \ldots, \lambda_k^{1/2} \right) \in \mathbb{R}^{m_y \times m_y}, \tag{47}$$

die Diagonalmatrix der Quadratwurzeln der zugehörigen Eigenvektoren bezeichnet. Schließlich seien für $i = 1, \ldots, k$

$$a_i := \Sigma_{xx}^{-1/2} \alpha_i \in \mathbb{R}^{m_x} \text{ und } b_i := \Sigma_{yy}^{-1/2} \beta_i \in \mathbb{R}^{m_y}. \tag{48}$$

Dann heißen für $i = 1, \ldots, k$

(1) $a_i \in \mathbb{R}^{m_x}$ und $b_i \in \mathbb{R}^{m_y}$ die *$i$ten kanonischen Koeffizientenvektoren*,

(2) die Zufallsvektoren $\xi_i := a_i^T X$ und $\upsilon_i := b_i^T Y$ die $i$ten *$i$ten kanonischen Variaten* und

(3) $\rho_i := \lambda_i^{1/2}$ die *$i$te kanonische Korrelation*.

Theorem (Eigenschaften kanonischer Korrelationen und Variaten)

Es seien

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix} \text{ mit } \mathbb{E}(Z) := 0_m \text{ und } \mathbb{C}(Z) := \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{49}$$

ein m-dimensionaler partitionierter Zufallsvektor sowie sein Erwartungswert und seine Kovarianzmatrix, respektive. Weiterhin seien für $i = 1, \ldots, k$ die kanonischen Koeffizientenvektoren a_i, b_i, die kanonischen Variaten ξ, υ_i und die kanonischen Korrelationen ρ_i definiert wie oben. Dann gilt, dass für $1 \leq r \leq k$ das Maximum des $r$ten restringierten Optimierungsproblems

$$\phi_r = \max_{a,b} a^T \Sigma_{xy} b \tag{50}$$

unter den Nebenbedingungen

$$a^T \Sigma_{xx} a = 1, \quad b^T \Sigma_{yy} b = 1, \quad a_i^T \Sigma_{xx} a = 0 \text{ für } i = 1, \ldots, r-1 \tag{51}$$

(1) den Wert $\phi_r = \rho_r$ hat und (2) bei $a = a_r$ und $b = b_r$ angenommen wird.

Bemerkungen

- ϕ_1 ist die größtmögliche Korrelation von $\xi = a^T X$ und $\upsilon = b^T Y$ unter den Nebenbedingungen
  - $\mathbb{V}(\xi) = a^T \Sigma_{xx} a = 1$ und $\mathbb{V}(\upsilon) = b^T \Sigma_{yy} b = 1$
- ϕ_r ist die größtmögliche Korrelation von $\xi = a^T X$ und $\upsilon = b^T Y$ unter den Nebenbedingungen
  - $\mathbb{V}(\xi) = a^T \Sigma_{xx} a = 1$ und $\mathbb{V}(\upsilon) = b^T \Sigma_{yy} b = 1$
  - $\mathbb{C}(\xi_i, \xi) = a_i^T \Sigma_{xx} a = 0$ für die ersten $i = 1, \ldots, r-1$ kanonischen Variaten ξ_i

Beweis

Wir betrachten das restringierte Optimierungsproblem

$$\phi_r^2 = \max_{a,b} \left(a^T \Sigma_{xy} b \right)^2 \text{ u.d.N. } a^T \Sigma_{xx} a = 1,\ b^T \Sigma_{yy} b = 1,\ a_i^T \Sigma_{xx} a = 0, i = 1, \ldots, r-1 \tag{52}$$

Wir folgen Mardia, Kent, and Bibby (1979), S. 284 und gehen schrittweise vor, d.h. wir lösen das restringierte Optimierungsproblem

$$\phi_r^2 = \max_{a} \left(\max_{b} \left(a^T \Sigma_{xy} b \right)^2 \text{ u.d.N.} b^T \Sigma_{yy} b = 1 \right) \text{ u.d.N. } a^T \Sigma_{xx} a = 1,\ a_i^T \Sigma_{xy} a = 0, i = 1, \ldots, r-1 \tag{53}$$

von innen nach außen.

Schritt (1)

Wir wählen wir zunächst ein festes $a \in \mathbb{R}^m$ und betrachten das restringierte Optimierungsproblem

$$\max_{b} \left(a^T \Sigma_{xy} b \right)^2 \text{ u.d.N. } b^T \Sigma_{yy} b = 1 \tag{54}$$

Dieses Optimierungsproblem kann geschrieben werden als

$$\max_{b} b^T \Sigma_{yx} a a^T \Sigma_{xy} b \text{ u.d.N. } b^T \Sigma_{yy} b = 1, \tag{55}$$

weil gilt, dass

$$\left(a^T \Sigma_{xy} b \right)^2 = \left(a^T \Sigma_{xy} b \right) \left(a^T \Sigma_{xy} b \right) = \left(a^T \Sigma_{xy} b \right)^T a^T \Sigma_{xy} b = b^T \Sigma_{yx} a a^T \Sigma_{xy} b. \tag{56}$$

Beweis (fortgeführt)

Das Optimierungsproblem (55) kann nun mithilfe des Theorems zur Maximierung quadratischer Formen mit Nebenbedingen gelöst werden. Im Sinne dieses Theorems setzen wir dazu

$$A := \Sigma_{yx} a a^T \Sigma_{xy} \text{ und } B := \Sigma_{yy}. \tag{57}$$

Dann hat (55) die Form

$$\max_b b^T A b \text{ unter der Nebenbedingung } b^T B b = 1, \tag{58}$$

Das Maximum von (58) entspricht nach dem Theorem zur Maximierung quadratischer Formen mit Nebenbedingungen dem größten Eigenwert von

$$B^{-1} A = \Sigma_{yy}^{-1} \Sigma_{yx} a a^T \Sigma_{xy} \tag{59}$$

Der größte Eigenwert von $\Sigma_{yy}^{-1} \Sigma_{yx} a a^T \Sigma_{xy}$ wiederum kann mithilfe des Theorems zum Eigenwert und Eigenvektor eines Matrixvektorprodukts bestimmt werden. Im Sinne dieses Theorems setzen wir dazu

$$A := \Sigma_{yy}^{-1} \Sigma_{yx}, \quad b := a, \quad B := \Sigma_{xy} \tag{60}$$

und erhalten für den betreffenden Eigenwert

$$\lambda_a = b^T B A a = a^T \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} a. \tag{61}$$

als Lösung (Maximum) des restringierten Optimierungsproblems

$$\max_b \left(a^T \Sigma_{xy} b \right)^2 \text{ u.d.N. } b^T \Sigma_{yy} b = 1 \tag{62}$$

Beweis (fortgeführt)

Schritt (2)

Basierend auf Schritt (1) verbleibt die Lösung des restringierten Optimierungsproblem

$$\phi_r^2 = \max_a a^T \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} a \text{ u.d.N. } a^T \Sigma_{xx} a = 1,\ a_i^T \Sigma_{xx} a = 0, i = 1, \ldots, r-1 \tag{63}$$

Dazu halten wir zunächst fest, dass (63) mit den Definitionen von α_i und K in der Definition der Kanonischen Koeffizientenvektoren, Variaten, und Korrelationen geschrieben werden kann als

$$\phi_r^2 = \max_\alpha \alpha^T K K^T \alpha \text{ u.d.N. } \alpha^T \alpha = 1,\ \alpha_i^T \alpha = 0, i = 1, \ldots, r-1, \tag{64}$$

denn

$$\begin{aligned}
\phi_r^2 &= \max_a a^T \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} a \text{ u.d.N. } a^T \Sigma_{xx} a = 1,\ a_i^T \Sigma_{xx} a = 0 \Leftrightarrow \\
\phi_r^2 &= \max_a a^T \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} a \text{ u.d.N. } \alpha^T \Sigma_{xx}^{-1/2} \Sigma_{xx} \Sigma_{xx}^{-1/2} \alpha = 1,\ \alpha_i^T \Sigma_{xx}^{-1/2} \Sigma_{xx} \Sigma_{xx}^{-1/2} \alpha = 0 \\
\phi_r^2 &= \max_a \alpha^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} \Sigma_{xx}^{-1/2} \alpha \text{ u.d.N. } \alpha^T \alpha = 1,\ \alpha_i^T \alpha = 0 \\
\phi_r^2 &= \max_a \alpha^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \Sigma_{yy}^{-1/2} \Sigma_{yx} \Sigma_{xx}^{-1/2} \alpha \text{ u.d.N. } \alpha^T \alpha = 1,\ \alpha_i^T \alpha = 0 \\
\phi_r^2 &= \max_a \alpha^T K K^T \alpha \text{ u.d.N. } \alpha^T \alpha = 1,\ \alpha_i^T \alpha = 0
\end{aligned} \tag{65}$$

Beweis (fortgeführt)

Dabei sind nach der betreffenden Definition die α_i die Eigenvektoren von KK^T mit den $i = 1, \ldots, r-1$ größten Eigenwerten. Nach dem Theorem zur Maximierung quadratischer Formen mit Nebenbedingungen ist die Lösung von (64) der größte Eigenwert von KK^T mit seinem assoziierten Eigenvektor. Die Nebenbedingung $\alpha_i^T \alpha = 0$ schränkt diese Wahl auf den rt-größten Eigenwert und seinen assoziierten Eigenvektor α_r ein. Mit der Definition von Eigenwerten und Eigenvektoren gilt also

$$\phi_r^2 = \alpha_r^T KK^T \alpha_r = \alpha_r^T \lambda_r \alpha_r = \lambda_r \alpha_r^T \alpha_r = \lambda_r. \tag{66}$$

Wir haben also gezeigt, dass das restringierte Optimierungsproblem des Theorems den Maximumwert $\phi_r = \lambda_r^{1/2}$ hat. Es bleibt zu zeigen, dass dieser Maximumwert für a_r und b_r angenommen wird.

Schritt (3)

Einsetzen von a_r und b_r in $a^T \Sigma_{xy} b$ ergibt mit

$$K = A\Lambda B^T \Leftrightarrow KB = A\Lambda B^T B \Leftrightarrow KB = A\Lambda \Leftrightarrow K\beta_r = \alpha_r \lambda_r^{1/2} \tag{67}$$

dass

$$a_r^T \Sigma_{xy} b_r = \alpha_r^T \Sigma_{xx}^{-1/2} \Sigma_{xy} \Sigma_{yy}^{-1/2} \beta_r = \alpha_r^T K \beta_r = \alpha_r^T \alpha_r \lambda_r^{1/2} = \rho_r \tag{68}$$

Also nimmt $a^T \Sigma_{xy} b$ bei a_r und b_r seinen restringierten Maximalwert λ_r an.

□

Simulationsbeispiel

Wir betrachten das Beispiel (vgl. Uurtio et al. (2018))

$$p(X) = N(x; 0_4, I_4) \text{ und } p(Y|X) = N(y; LX, G) \tag{69}$$

mit

$$L := \begin{pmatrix} 0.0 & 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -1.0 \end{pmatrix} \text{ und } G := \begin{pmatrix} 0.2 & 0.0 & 0.0 \\ 0.0 & 0.4 & 0.0 \\ 0.0 & 0.0 & 0.3 \end{pmatrix} \tag{70}$$

Hier gilt offenbar $m_x = 4, m_y = 3, m = 7$ und

$$\begin{aligned} Y_1 &= X_3 + \varepsilon_1 \\ Y_2 &= X_1 + \varepsilon_2 \\ Y_3 &= -X_4 + \varepsilon_3 \end{aligned} \tag{71}$$

mit

$$X_1 \sim N(0,1), X_3 \sim N(0,1), X_4 \sim N(0,1) \tag{72}$$

und

$$\varepsilon_1 \sim N(0, 0.2), \varepsilon_2 \sim N(0, 0.4), \varepsilon_3 \sim N(0, 0.3) \tag{73}$$

Simulationsbeispiel

Mit dem Theorem zu gemeinsamen Normalverteilungen (vgl. Einheit (3) Matrizen) ergibt sich, dass

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N(0_7, \Sigma) \tag{74}$$

mit

$$\Sigma = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix}, \tag{75}$$

wobei

$$\Sigma_{xx} = I_4, \quad \Sigma_{xy} = L^T, \quad \Sigma_{yx} = L \text{ und } \Sigma_{yy} = G + LL^T. \tag{76}$$

Explizit ergibt sich also

$$\Sigma = \begin{pmatrix} I_4 & L^T \\ L & G + LL^T \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & -1.0 \\ 0.0 & 0.0 & 1.0 & 0.0 & 1.2 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.4 & 0.0 \\ 0.0 & 0.0 & 0.0 & -1.0 & 0.0 & 0.0 & 1.3 \end{pmatrix} \tag{77}$$

Simulationsbeispiel

```
# R Pakete für Matrizenrechnung
library(matlib)
library(expm)

# Modellparameter
L = matrix(c(0,0,1, 0,
             1,0,0, 0,
             0,0,0,-1),
           nrow  = 3,
           byrow = T)
G = diag(c(0.2,0.4,0.3))

# Kovarianzmatrixpartition
Sigma_xx = diag(4)
Sigma_xy = t(L)
Sigma_yx = L
Sigma_yy = G + L %*% t(L)
Sigma    = rbind(cbind(Sigma_xx, Sigma_xy), cbind(Sigma_yx, Sigma_yy))
print(Sigma)
```

```
>      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
> [1,]    1    0    0    0  0.0  1.0  0.0
> [2,]    0    1    0    0  0.0  0.0  0.0
> [3,]    0    0    1    0  1.0  0.0  0.0
> [4,]    0    0    0    1  0.0  0.0 -1.0
> [5,]    0    0    1    0  1.2  0.0  0.0
> [6,]    1    0    0    0  0.0  1.4  0.0
> [7,]    0    0    0   -1  0.0  0.0  1.3
```

Simulationsbeispiel

```
# Evaluation der iten kanonischen Koeffizientenvektoren und Korrelationen
K      = sqrtm(inv(Sigma_xx)) %*% Sigma_xy %*% sqrtm(inv(Sigma_yy)) # K
ALB    = svd(K)                                                     # K = A\LambdaV
A      = ALB$u                                                      # A
Lambda = ALB$d                                                      # Lambda
B      = ALB$v                                                      # B
rho    = Lambda                                                     # \rho_i = \lambda_i^{1/2}
a      = sqrtm(inv(Sigma_xx)) %*% A                                 # a_i = \Sigma_{xx}^{-1/2}\alpha_i
b      = sqrtm(inv(Sigma_yy)) %*% B                                 # b_i = \Sigma_{yy}^{-1/2}\beta_i
```

Die kanonische Korrelationen und kanonischen Koeffizientenvektoren ergeben sich zu

```
> rho_1 =  0.913 , a_1^T = ( 0 0 -1 0 ), b_1^T = ( -0.913 0 0 )
> rho_2 =  0.877 , a_2^T = ( 0 0 0 1 ) , b_2^T = ( 0 0 -0.877 )
> rho_3 =  0.845 , a_3^T = ( -1 0 0 0 ), b_3^T = ( 0 -0.845 0 )
```

Vorbemerkungen

Modellformulierung

Modellschätzung

Modellevaluation

Definition (Schätzer kanonischer Korrelationen und Koeffizientenvektoren)

Für $i = 1, \ldots, n$ seien

$$Z_i = \begin{pmatrix} X_i \\ Y_i \end{pmatrix} \text{ mit } \mathbb{E}(Z_i) := 0_m \text{ und } \mathbb{C}(Z_i) := \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{78}$$

unabhängig und identisch verteilte m-dimensionale partitionierte Zufallsvektoren sowie ihr Erwartungswert und ihre Kovarianzmatrix, respektive, und

$$C := \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix} \in \mathbb{R}^{m \times m} \tag{79}$$

sei ihre Stichprobenkovarianzmatrix. Dann sind für $i = 1, \ldots, k := \min\{m_x, m_y\}$

$$\hat{a}_i := C_{xx}^{-1/2} \hat{\alpha}_i \in \mathbb{R}^{m_x}, \quad \hat{b}_i := C_{yy}^{-1/2} \hat{\beta}_i \in \mathbb{R}^{m_y} \text{ und } \hat{\rho}_i := \hat{\lambda}_i^{1/2} \tag{80}$$

Schätzer der $i$ten kanonischen Koeffizientenvektoren und kanonischen Korrelationen, respektive. Dabei sind mit

$$\hat{K} := C_{xx}^{-1/2} C_{xy} C_{yy}^{-1/2} \in \mathbb{R}^{m_x \times m_y} \tag{81}$$

$\hat{\alpha}_i$ und $\hat{\lambda}_i$ der $i$te Eigenvektor und sein zugehöriger Eigenwert von $\hat{K}\hat{K}^T$ und $\hat{\beta}_i$ der entsprechende Eigenkvektor von $\hat{K}^T\hat{K}$.

Bemerkungen

- Zur Modellschätzung wird $\mathbb{C}(Z)$ also durch C ersetzt.

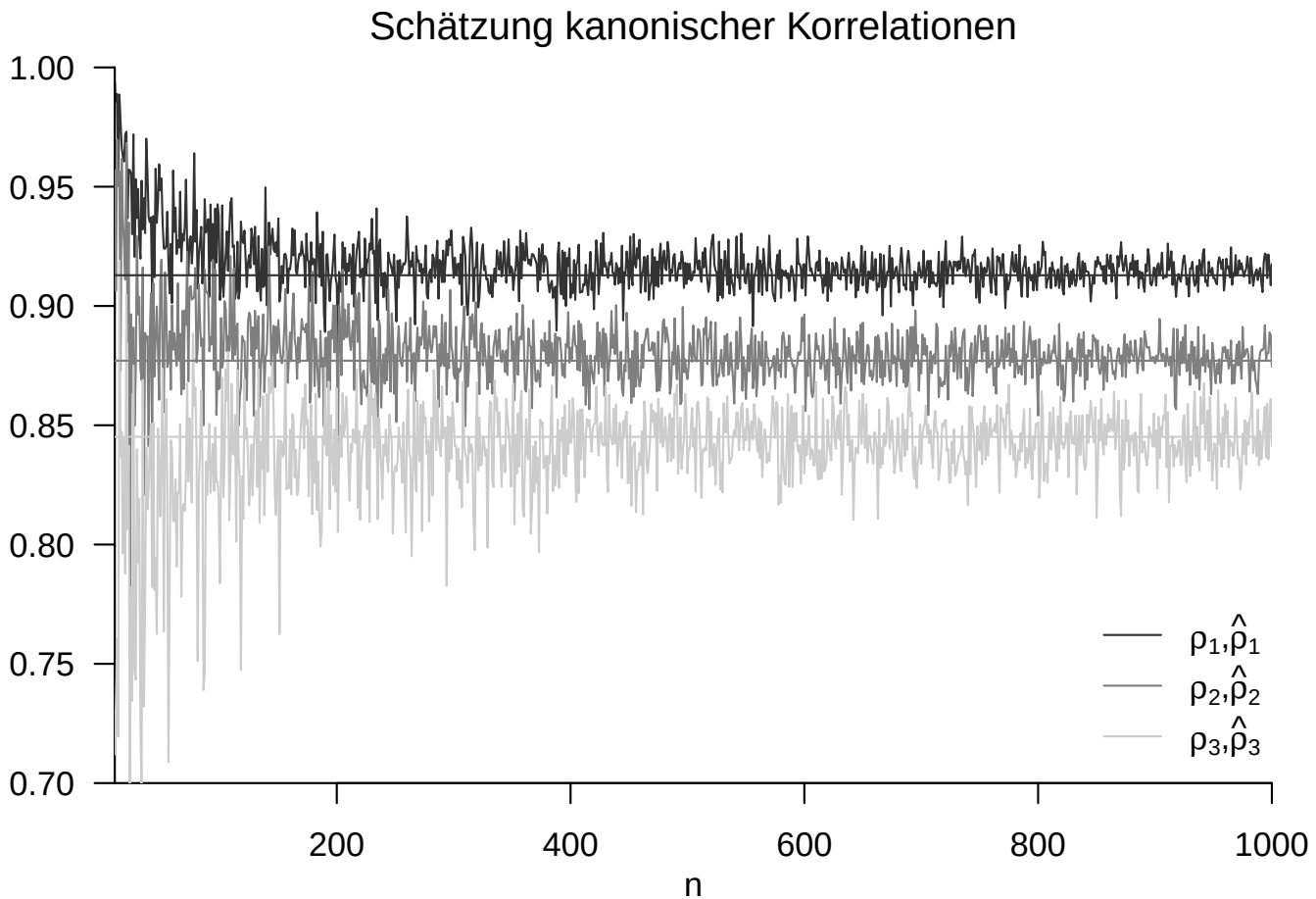
Simulationsbeispiel

```
# R Pakete
library(MASS)
library(matlib)
library(expm)

# Modellparameter
m_x      = 4
m_y      = 3
k        = min(m_x,m_y)
L        = matrix(c(0,0,1,0,1,0,0,0,0,0,0,-1), nrow = 3,byrow = 3)
G        = diag(c(0.2,0.4,0.3))
Sigma_xx = diag(4)
Sigma_xy = t(L)
Sigma_yx = L
Sigma_yy = G + L %*% t(L)
Sigma    = rbind(cbind(Sigma_xx, Sigma_xy), cbind(Sigma_yx, Sigma_yy))
K        = sqrtm(inv(Sigma_xx)) %*% Sigma_xy %*% sqrtm(inv(Sigma_yy))
ALB      = svd(K)
A        = ALB$u
Lambda   = ALB$d
B        = ALB$v
rho      = Lambda
a        = sqrtm(inv(Sigma_xx)) %*% A
b        = sqrtm(inv(Sigma_yy)) %*% B
```

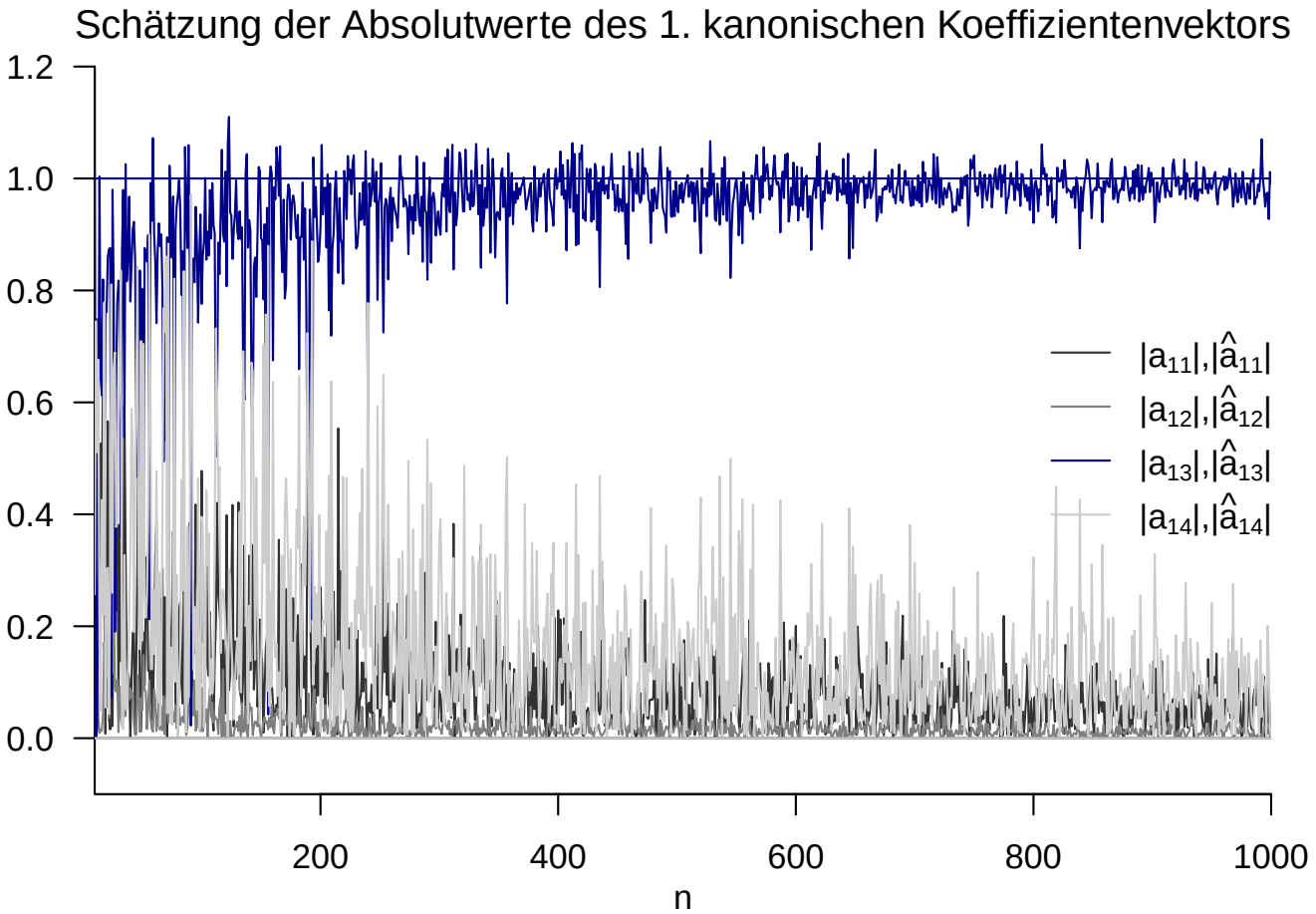
Simulationsbeispiel

```
# Simulationen
n       = 1e1:1e3
rho_hat = matrix(rep(NaN, length(n)*k)  , nrow = k)
a_1_hat = matrix(rep(NaN, length(n)*m_x), nrow = m_x)
for(i in 1:length(n)){

    # Datengeneration
    Y           = t(mvrnorm(n[i],rep(0, m_x+m_y),Sigma))
    I_n         = diag(n[i])
    J_n         = matrix(rep(1,n[i]^2), nrow = n[i])

    # Stichprobenkovarianzmatrixpartition
    C           = (1/(n[i]-1))*(Y %*% (I_n-(1/n[i])*J_n) %*% t(Y))
    C_xx        = C[1:m_x,1:m_x]
    C_xy        = C[1:m_x,(m_x+1):(m_x+m_y)]
    C_yx        = C[(m_x+1):(m_x+m_y),1:m_x]
    C_yy        = C[(m_x+1):(m_x+m_y),(m_x+1):(m_x+m_y)]

    # Kanonische Korrelationsanalyse
    K_hat       = sqrtm(inv(C_xx)) %*% C_xy %*% sqrtm(inv(C_yy))
    ALB_hat     = svd(K_hat)
    A_hat       = ALB_hat$u
    Lambda_hat  = ALB_hat$d
    B_hat       = ALB_hat$v
    a_hat       = sqrtm(inv(C_xx)) %*% A_hat
    b_hat       = sqrtm(inv(C_yy)) %*% B_hat
    rho_hat[,i] = as.matrix(Lambda_hat)
    a_1_hat[,i] = a_hat[,1]
}
```

Simulationsbeispiel



Simulationsbeispiel



Anwendungsbeispiel

Wir betrachten erneut den Datensatz nach Rudolf and Buse (2020) Kapitel 4

Wir betrachten die psychodiagnostischen Daten der $n = 45$ Studierenden

- X_1 Intelligenztestscore
- X_2 Mathematiktestscore
- Y_1 Gewissenhaftigkeitscore
- Y_2 Verträglichkeitscore

als 45 unabhängige Realisierungen eines 4-dimensionaler Zufallsvektors Z. Wir sind also an den kanonischen Korrelationen der Kognitionstestwerte (Intelligenz, Mathematik) mit den Charaktertestwerten (Gewissenhaftigkeit, Veträglichkeit) interessiert.

Anwendungsbeispiel

Daten der ersten 12 Proband:innen

x1	x2	y1	y2
54	44	31	60
60	20	33	31
67	36	26	54
41	39	31	26
66	57	34	56
51	28	42	23
51	46	34	40
37	46	36	31
57	54	28	49
47	12	34	41
50	67	27	53
42	63	26	36

Anwendungsbeispiel

Kanonische Korrelationsanalyse

```
# Datenpräprozessierung
D          = read.spss(file.path(getwd(), "12_Daten", "studienerfolg.sav"), to.data.frame = T)
x          = as.matrix(cbind(D$X1, D$X2))
y          = as.matrix(cbind(D$X3, D$X4))
n          = nrow(x)
m_x        = ncol(x)
m_y        = ncol(y)
Y          = rbind(t(x),t(y))

# Stichprobenkovarianzmatrixpartition
I_n        = diag(n)
J_n        = matrix(rep(1,n^2), nrow = n)
C          = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))
C_xx       = C[1:m_x,1:m_x]
C_xy       = C[1:m_x,(m_x+1):(m_x+m_y)]
C_yx       = C[(m_x+1):(m_x+m_y),1:m_x]
C_yy       = C[(m_x+1):(m_x+m_y),(m_x+1):(m_x+m_y)]

# Kanonische Korrelationsanalyse
K_hat      = sqrtm(inv(C_xx)) %*% C_xy %*% sqrtm(inv(C_yy))
ALB_hat    = svd(K_hat)
A_hat      = ALB_hat$u
Lambda_hat = ALB_hat$d
B_hat      = ALB_hat$v
a_hat      = sqrtm(inv(C_xx)) %*% A_hat
b_hat      = sqrtm(inv(C_yy)) %*% B_hat
rho_hat    = as.matrix(Lambda_hat)
```

```
> rho_hat_1 :  0.245
> a_hat_1   :  0.0152 0.0497
> b_hat_1   :  0.143 0.0412
> rho_hat_2 :  0.133
> a_hat_2   :  0.0905 -0.0132
> b_hat_2   :  -0.0484 0.0764
```

Anwendungsbeispiel

Kanonische Korrelationsanalyse mir R's `cancor()` Funktion

```
# Datenpräprozessierung
D           = read.spss(file.path(getwd(), "12_Daten", "studienerfolg.sav"), to.data.frame = T)
x           = as.matrix(cbind(D$X1, D$X2))
y           = as.matrix(cbind(D$X3, D$X4))
cca         = cancor(x,y)
```

```
> $cor
> [1] 0.245 0.133
>
> $xcoef
>         [,1]     [,2]
> [1,] 0.00229  0.01364
> [2,] 0.00749 -0.00199
>
> $ycoef
>         [,1]    [,2]
> [1,] 0.02161 -0.0073
> [2,] 0.00621  0.0115
>
> $xcenter
> [1] 59.0 57.2
>
> $ycenter
> [1] 36.4 39.1
```

Anwendungsbeispiel

Die geschätzte maximale Korrelation von Linearkombinationen von (x_1, x_2) und (y_1, y_2) ist 0.25.

- (x_1, x_2) und (y_1, y_2) sind multivariat also "gering bis mäßig" korreliert.

Basierend auf der simulationsvalidierten Schätzung ergibt sich

- $\xi = 0.02X_1 + 0.05X_2$ als "bester Prädiktor"
- $\upsilon = 0.09Y_1 - 0.01Y_2$ als "am besten prädizierbares Kriterium"

"Mathematikfähigkeiten" scheinen zur Prädiktion der betrachteten "Charaktereigenschaften" etwas wichtiger als "Intelligenz"; bei den betrachteten "Charaktereigenschaften" trägt "Gewissenhaftigkeit" mehr zum Kriterium bei als "Verträglichkeit".

Vorbemerkungen

Modellformulierung

Modellschätzung

Modellevaluation

Überblick

Ein Ziel der Modellevaluation bei der Kanonischen Korrelationsanalyse kann das Testen von

$$H_0 : \Sigma_{xy} = 0_{m_x m_y} \tag{82}$$

sein. Diese Nullhypothese besagt, dass zwischen keiner der Variablen $X_1, \ldots, X_{m_x}$ und $Y_1, \ldots, Y_{m_y}$ eine lineare Abhängigkeit besteht. Dies impliziert, dass alle kanonischen Korrelationen gleich 0 sind, denn es gilt

$$\Sigma_{xy} = 0_{m_x m_y} \Rightarrow K = 0_{m_x m_y} \Rightarrow \Lambda = 0_{m_y m_y}\,. \tag{83}$$

Die Alternativhypothese zu dieser Nullhypothese lautet

$$H_1 : \Sigma_{xy_{i,j}} \neq 0 \text{ für mindestens ein Paar } (i,j) \text{ mit } 1 \leq i \leq m_x \text{ und } 1 \leq j \leq m_y. \tag{84}$$

Die Alternativhypothese besagt also, dass mindestens eine der Variablen $X_1, \ldots, X_{m_x}$ und eine der Variablen $Y_1, \ldots, Y_{m_y}$ linear abhängig sind und damit nicht alle kanonischen Korrelationen gleich null sind.

Wie bei der einfaktoriellen Varianzanalyse (und generell im Kontext multivariater allgemeiner lineare Modelle) können kritische Wert-basierte Tests der obigen Nullhypothese mit verschiedenen Teststatistiken (*Wilks' Λ, Pillai Statisik*) konstruiert werden, der Verteilungen wiederum nur in Speziallfällen analytisch beschrieben sind und ansonsten mit f-Verteilungen approximiert werden.

Wir betrachten hier lediglich das Testen der obigen Nullhypothese mit der Wilk's Λ Statistik.

Theorem (Wilks' Λ für die kanonische Korrelationsanalyse)

Es seien das Modell der kanonischen Korrelationsanalyse, die Partition der Stichprobenkovarianzmatrix, und die Schätzer der kanonischen Korrelationen definiert wie oben. Dann hat die die Wilks' Λ Teststatistik die Form

$$\Lambda = \frac{|C|}{|C_{xx}||C_{yy}|} = \prod_{i=1}^{k} \left(1 - \hat{\rho}_i^2\right), \tag{85}$$

wobei $|\cdot|$ die Determinante bezeichnet. Weiterhin ist für

$$H_0 : \Sigma_{xy} = 0_{m_x m_y} \tag{86}$$

die Statistik

$$\tau := \frac{1 - \Lambda^{1/t}}{\Lambda^{1/t}} \frac{\nu_2}{\nu_1} \tag{87}$$

mit

$$\nu_1 := m_x m_y \text{ und } \nu_2 := wt - \frac{1}{2} m_x m_y + 1 \tag{88}$$

sowie

$$w := n - \frac{1}{2}(m_x + m_y + 3) \text{ und } t := \sqrt{\frac{m_x^2 m_y^2 - 4}{m_x^2 + m_y^2 - 5}} \tag{89}$$

approximativ f-verteilt mit den Freiheitsgradparametern ν_1 und ν_2.

Bemerkungen

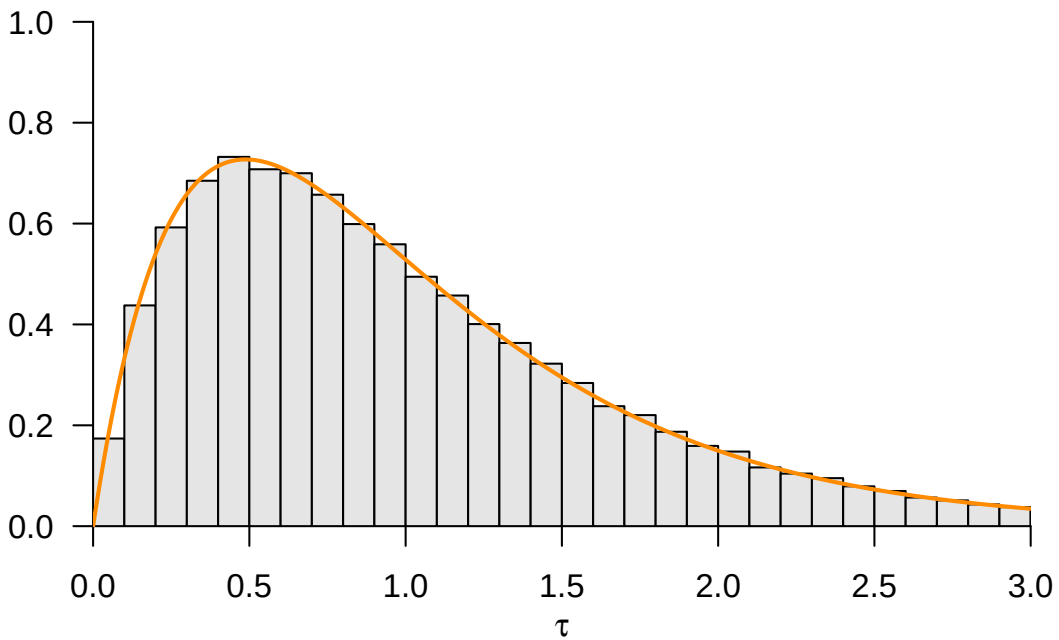
- Wir verzichten auf Beweise aller Aussagen dieses Theorems.
- Λ wird klein, wenn die absoluten Werte der Einträge in C_{xy} groß werden.
- Man denke in diesem Zusammenhang an das Berechnen der Determinante einer 2×2 Matrix.
- Λ wird klein, wenn zumindest ein $\hat{\rho}_i$ groß ist
- Für $\hat{\rho}_i = 0$ für alle $i = 1, ..., k$ gilt $\Lambda = 1$; für $\hat{\rho}_i = 1$ für alle $i = 1, ..., k$ gilt $\Lambda = 0$
- Kleine Werte von Λ und damit große Werte von τ sprechen also gegen H_0.

Simulationsbeispiel

```
# Modellparameter
library(MASS)
m_x      = 2
m_y      = 2
Sigma_xx = diag(m_x)
Sigma_xy = matrix(rep(0,m_x*m_y), nrow = 2)              # H_0 : \Sigma_{xy} = 0_{m_xm_y}
Sigma_yx = Sigma_xy
Sigma_yy = diag(m_y)
Sigma    = rbind(cbind(Sigma_xx, Sigma_xy),
                 cbind(Sigma_yx, Sigma_yy))
# Testszenarioparameter
n        = 45                                            # Anzahl Datenpunkte
w        = n-((1/2)*(m_x+m_y+3))
t        = sqrt((m_x^2*m_y^2 - 4)/((m_x^2+m_y^2)-5))
nu_1     = m_x*m_y
nu_2     = w*t-((1/2)*m_x*m_y) + 1
alpha_0  = 0.05                                          # Signifikanzlevel
kW       = qf(1-alpha_0,nu_1,nu_2)                       # kritischer Wert

# Simulationen
sim      = 1e5                                           # Anzahl an Simulationen
tau      = rep(NaN, sim)                                 # Teststatistikarray
phi      = rep(0,   sim)                                 # Testarray
for(s in 1:sim){
    Y      = t(mvrnorm(n,rep(0, m_x+m_y),Sigma))         # Datengeneration
    I_n    = diag(n)
    J_n    = matrix(rep(1,n^2), nrow = n)
    C      = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))  # Stichprobenkovarianzmatrix
    C_xx   = C[1:m_x,1:m_x]
    C_xy   = C[1:m_x,(m_x+1):(m_x+m_y)]
    C_yy   = C[(m_x+1):(m_x+m_y),(m_x+1):(m_x+m_y)]
    Lambda = det(C)/(det(C_xx)*det(C_yy))                # Wilk's Lambda
    tau[s] = ((1-Lambda^(1/t))/Lambda^(1/t))*(nu_2/nu_1) # \tau
    phi[s] = tau[s] > kW                                 # Test
}
```

Simulationsbeispiel

$$n = 45, m_x = 2, m_y = 2, \nu_1 = 4, \nu_2 = 82, k = 2.48, \hat{\alpha} = 0.0498$$



Simulationsbeispiel

```
# Datenpräprozessierung
library(foreign)
D           = read.spss(file.path(getwd(), "12_Daten", "studienerfolg.sav"), to.data.frame = T)
x           = as.matrix(cbind(D$X1, D$X2))
y           = as.matrix(cbind(D$X3, D$X4))
n           = nrow(x)
m_x         = ncol(x)
m_y         = ncol(y)
Y           = rbind(t(x),t(y))

# Testszenarioparameter
w         = n-((1/2)*(m_x+m_y+3))
t         = sqrt((m_x^2*m_y^2 - 4)/((m_x^2+m_y^2)-5))
nu_1      = m_x*m_y
nu_2      = w*t-((1/2)*m_x*m_y) + 1
alpha_0   = 0.05
k         = qf(1-alpha_0,nu_1,nu_2)

# Stichprobenkovarianzmatrixpartition
I_n         = diag(n)
J_n         = matrix(rep(1,n^2), nrow = n)
C           = (1/(n-1))*(Y %*% (I_n-(1/n)*J_n) %*% t(Y))
C_xx        = C[1:m_x,1:m_x]
C_xy        = C[1:m_x,(m_x+1):(m_x+m_y)]
C_yx        = C[(m_x+1):(m_x+m_y),1:m_x]
C_yy        = C[(m_x+1):(m_x+m_y),(m_x+1):(m_x+m_y)]

# Nullhypothesentest
Lambda      = det(C)/(det(C_xx)*det(C_yy))
tau         = ((1-Lambda^(1/t))/Lambda^(1/t))*(nu_2/nu_1)
phi         = tau > k
pval        = 1 - pf(tau,nu_1,nu_2)
```

```
> Lambda : 0.924
> tau    : 0.831
> k      : 2.48
> phi    : 0
> p      : 0.509
```

$\Rightarrow H_0 : \Sigma_{xy} = 0_{m_x m_y}$ wird nicht verworfen.

References

Anderson, T. W. 2003. *An Introduction to Multivariate Statistical Analysis*. 3rd ed. Wiley Series in Probability and Statistics. Hoboken, N.J: Wiley-Interscience.

Mardia, K. V., J. T. Kent, and J. M. Bibby. 1979. *Multivariate Analysis*. Probability and Mathematical Statistics. London ; New York: Academic Press.

Rudolf, Matthias, and Johannes Buse. 2020. *Multivariate Verfahren*. Göttingen: Hogrefe.

Uurtio, Viivi, João M. Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. 2018. "A Tutorial on Canonical Correlation Methods." *ACM Computing Surveys* 50 (6): 1–33. https://doi.org/10.1145/3136624.