Einführung in die Mustererkennung

ist nicht ganz Vollständig, kann auf Anfrage auch gerne die Tex-Files rausrücken

`deepunicorn@protonmail.com`

Technische Universität Wien — December 2, 2019

1 Lineare Algebra

Inverse eine 2D Matrix

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \Longrightarrow \qquad \mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \tag{1}$$

wobei $|\mathbf{A}|$ die Determinante von $\mathbf{A}$ ist.

2 Grundlagen

2.1 Regression

Regressionsanalyse ist die Schätzung der Zusammenhänge zwischen den Variablen, also die Suche nach einem Modell.

Polynomial Curve Fitting An Datenpunkten können beliebige Polynome M-ter Ordnung gefittet, Koeffizientenwahl durch Minimierung der Fehlerfunktion, werden.

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j x^j \tag{2}$$

Fehlerfunktion gibt die Abweichung zwischen der Beobachtung t_n und der Schätzung $y(x_n, \mathbf{w})$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 \tag{3}$$

Overfitting Das Modell wird zu komplex (zu viele Freiheitsgrade), d.h. Störungen bzw. Rauschen beeinflusst das Modell zu stark.

2.2 Klassifikator: Nearest Neighbor (kNN)

Voronoi-Diagramm

Distanzfunktionen

- Euklidische Distanz, auch L_2 Norm bezeichnet

$$d(x,y) = ||x-y||_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

- Manhatten Distanz, auch L_1 Norm bezeichnet

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{5}$$

Eigenschaften von kNN

- k ist vom Benutzer festgelegt
- kNN ist nicht parametrisch, d.h. es wird kein Training benötigt und es werden keine Koeffizienten angepasset wie bei Regression
- Speicheraufwand $\mathcal{O}(N)$, wächst linear
- Laufzeitaufwand ist für Brute force $\mathcal{O}(N)$, für vorbereitet Daten $\mathcal{O}(\log N)$ jedoch erst nach $\mathcal{O}(N \log N)$ Vorbereitungszeit

2.3 Statistische Grundlagen für diskrete Zufallsvariablen

Zufallsvaraible Eine Zufallsvaraible x ist diskrete, wenn sie nur endlich viele Werte einer Menge $X \in \{x_1, \ldots, x_m\}$ annehmen kann.

Wahrscheinlichkeitsfunktion Wie Wahrscheinlichkeit das ein gewisses Ereignis x_i eintritt, ist gegeben durch die Wahrscheinlichkeitsfunktion $p_i = P(x = x_i)$.

Eigenschaften einer Wahrscheinlichkeitsfunktion Wahrscheinlichkeitsfunktionen müssen immer größer Null $p_i \geq 0$ und normiert $\sum_{x \in X} P(x) = 1$ sein.

Erwartung $\mathcal{E}[\cdot]$ einer Funktion $h(x)$ einer Zufallsvaraible x ist defineirt als

$$\mathcal{E}[h(x)] = \sum_{x \in X} h(x) P(x) \tag{6}$$

Moment Der Erwartungswert von $h(x) = x^i$ wird i-tes Moment bezeichnet.

Mittelwert Das erste Moment $\mathcal{E}[h(x) = x] = \mu$ wird ebenefalls Mittelwert bezeichent und gibt den Schwerpunkt der Verteilungsfunktion $P(x)$ an.

Varianz

$$\sigma^2 = \mathcal{E}\left[(x-\mu)^2\right] = \sum_{x \in X} (x-\mu)^2 P(x) = \mathcal{E}[x^2] - (\mathcal{E}[x])^2 \tag{7}$$

Standardabweichung Die Standardabweichung beschreibt, wie stark Werte einer Verteilung von dem Mittelwert abweichen.

$$\sigma = \sqrt{\mathcal{E}\left[(x-\mu)^2\right]} \tag{8}$$

Verbundwahrscheinlichkeit Die Verbundwahrscheinlichkeit (joint probability) gibt die Wahrscheinlichkeit $p_{ij} = P(x = x_i, y = y_i)$, dass ein Paar ein Ereignissen (x_i, y_j) eintritt an. Wobei auch hier $P(x, y) \geq 0$ und $\sum_{x \in X} \sum_{y \in Y} P(x, y) = 1$ zutreffen muss.

Randverteilung Die Randverteilung (marginal distribution) gibt die Wahrscheinlichkeit des Eintreten eines Ereignisses ohne Berücksichtigung des anderen an.

$$p_i = p_{i\cdot} = \sum_j p_{ij} = \sum_{y \in Y} P(x, y) \tag{9}$$

Sind die Zufallsvariablen x und y unabhängig voneinander, dann kann die Verbundwahrscheinlichkeit als Produkt der korrespondierenden Randverteilungen berechnet werden, d.h. $p_{ij} = p_{i\cdot} p_{\cdot j}$.

Bedingte Wahrscheinlichkeit Die bedingte Wahrscheinlichkeit (conditional probability) $P(A|B)$ von A unter B, ist die Wahrscheinlichkeit, dass A eintritt, nachdem B bereits eingetreten ist.

$$P(A|B) = \frac{P(A, B)}{P(B)} = \frac{P(x = v_i, y = u_j)}{P(y = u_j)} = \frac{p_{ij}}{p_{\cdot j}} \tag{10}$$

Die Verbundwahrscheinlichkeit kann aus den bedingten Wahrscheinlichkeiten und den Randverteilungen berechnet werden:

$$P(A, B) = P(A|B)P(B) \qquad P(A, B) = P(B|A)P(A) \tag{11}$$

Für unabhängige Wahrescheinlichkeiten gilt, daher $P(A|B) = P(A)$.

Example 1: Berechnung der Bedingten Wahrscheinlichkeiten

$y \backslash x$	1	2	3	4	$p_{\cdot j} = P(y)$
1	0.08	0.12	0.15	0.15	0.5
2	0.02	0.18	0.25	0.05	0.5
$p_{i\cdot} = P(x)$	0.1	0.3	0.4	0.2	1

Mit dieser Tabelle können sehr einfach die Bedingten Wahrscheinlichkeiten errechnet werden.

$$P(x = x_i|y = y_j) = \frac{P(x = x_i, y = y_j)}{P(y = y_j)} \qquad \text{bzw.} \qquad P(y = y_j|x = x_i) = \frac{P(x = x_i, y = y_j)}{P(x = x_i)}$$

Dies ergibt z.B. für $i = 3$ und $j = 1$

$$P(x = x_3|y = y_1) = \frac{P(x = x_3, y = y_1)}{P(y = y_1)} = \frac{0.15}{0.5} = 0.3$$

$$P(y = y_1|x = x_3) = \frac{P(x = x_3, y = y_1)}{P(x = x_3)} = \frac{0.15}{0.4} = 0.375$$

Bayes-Theorem Das Bayes-Theorem erlaubt es, die bedingte Wahrscheinlichkeit $P(B|A)$ der bedingten als Funktion Wahrscheinlichkeit der Randverteilungen $P(A)$, $P(B)$ und der bedingten Wahrscheinlichkeit $P(A|B)$ auszudrücken.

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \tag{12}$$

$P(B)$: *a priori* Wahrscheinlichkeit (prior) von B
$P(B|A)$: *a posteriori* Wahrscheinlichkeit (posterior) von B und A

Bayes-Theorem für Mustererkennung Das Bayes-Theorem gibt die Wahrscheinlichkeit an, dass das Muster zur Klasse j gehört, basierend auf der Merkmalsausprägung $x = i$.

$$P(u = j|x = i) = \frac{P(x = i|u = j)P(u = j)}{P(x = i)} \tag{13}$$

x: Merkmal
u: Klassenzugehörigkeit von Mustern

Fehlerwahrscheinlichkeit $P(error|x = i)$ ist die Summe der Wahrscheinlichkeiten, dass $x = i$ falsch klassifiziert wird. (siehe Beispiel darunter)

Fehlerrate $P(error) = \sum_i P(error|x = i)$ ist die Wahrscheinlichkeit, dass irgendein Merkmalsvektor falsch klassifiziert wird.

Example 2: Berechnung des Posterior

$$P(\omega_j|x = i) = \frac{P(x = i|u = j)P(u = j)}{P(x = i)} \tag{14}$$

Gegeben sind die a priori Wahrscheinlichkeiten $P(\omega_1) = 0.6$ und $P(\omega_2) = 0.4$ und die gemessene Verteilung der Merkmale

$\omega_j \backslash i$	1	2	3	4
$P(x = i\|\omega_1)$	0.5	0.3	0.15	0.05
$P(x = i\|\omega_1)$	0.3	0.25	0.25	0.2
$P(x = i)$	0.42	0.28	0.19	0.11

$$P(x = i) = \sum_{j=1}^{c} P(x = i|\omega_j)P(\omega_j) \quad \text{z.B. für } i = 1 \quad P(x = 1) = 0.5 \cdot 0.6 + 0.3 \cdot 0.4 = 0.42 \tag{15}$$

Damit lässt sich nun auch die a posteriori Wahrscheinlichkeiten für jede Klasse-/Merkmalskombination berechnen, welche zur Bestimmung der Klassenzugehörigkeit verwendet werden kann.

$P(\omega_j\|x = i)$ $i \backslash j$	1	2
1	0.71	0.29
2	0.64	0.36
3	0.47	0.53
4	0.27	0.73

Nun kann noch die (bedingte) Fehlerwahrscheinlichkeiten $P(error|x = i)$ und die Fehlerrate (error rate) $P(error)$ berechnet werden.

$P(\omega_j\|x = i)$ $i \backslash j$	1	2	$P(error\|x = i)$
1	0.71	0.29	0.29
2	0.64	0.36	0.36
3	0.47	0.53	0.47
4	0.27	0.73	0.27

$$\begin{aligned} P(error) &= \sum_{i=1}^{4} P(error|x = i)P(x = i) \\ &= 0.29 \cdot 0.42 + 0.36 \cdot 0.28 + 0.47 \cdot 0.19 + 0.27 \cdot 0.11 = 0.34 \end{aligned} \tag{16}$$

Bayes-Entscheidungsregel Gegeben die Beobachtung (Merkmalsausprägung) $x = i$, entscheide Wahrscheinlichkeit für die Klasse k, welche die größte a posteriori aufweist.

$$k = \arg\max_j P\left(\omega_j | x = i\right) \tag{17}$$

2.4 Statistische Grundlagen für stetige Zufallsvariablen

Zufallsvariable $x \in \mathbb{R}$ heißt stetig, weil sie unendlich viele verschiedene Werte annehmen kann. Die Wahrscheinlichkeit das genau ein Ereigniss $x = a$ eintritt muss gleich 0 sein.

$$P(x = a) = \int_a^a p(x)dx = 0 \qquad (\forall a \in \mathbb{R})$$

Dichtefunktion Für stetige Zufallsvariablen wird eine Dichtefunktion $p(x)$ (PDF, probability density function) statt einer Wahrscheinlichkeitsfunktion $p_i = P(x = x_i)$ verwendet. Diese muss die selbe Eigenschaften wie die Wahrscheinlichkeitsfunktion erfüllen, d.h. $p(x) \geq 0$ und $\int_{-\infty}^{\infty} p(x)dx = 1$. Das ein eintretendes Ereignis x sich im Intervall $[a, b]$ befindet, ist gegeben durch

$$P[x \in (a, b)] = \int_a^b p(x)dx \tag{18}$$

Erwartungswert Der Erwartungswert $\mathcal{E}[\cdot]$ einer Funktion $h(x)$ einer stetigen Zufallsvaraible x ist defineirt als

$$\mathcal{E}[h(x)] = \int_{\infty}^{\infty} h(x)p(x)dx \tag{19}$$

Ist $h(x) = x^i$ so wird dies ebenfalls das i-te Moment genannt. Das erste Moment ($i = 1$) wird auch hier als **Mittelwert** $\mathcal{E}[x] = \mu$ bezeichent.

Varianz

$$\sigma^2 = \mathcal{E}\left[(x - \mu)^2\right] = \int_{-\infty}^{\infty} (x - \mu)^2 p(x)dx \tag{20}$$

2.5 Bayes-Theorem für stetige Merkmale

Für stetige Zufallsvariablen wird die Wahrscheinlichkeitsfunktion mit der Dichtefunktion ersetzt.

$$P\left(\omega_j | x = i\right) = \frac{P\left(x = i | \omega_j\right) P\left(\omega_j\right)}{P(x = i)} \qquad \Longrightarrow \qquad P\left(\omega_j | x\right) = \frac{p\left(x | \omega_j\right) P\left(\omega_j\right)}{p(x)} \tag{21}$$

Dabei wir $p\left(x|\omega_j\right)$ als "class conditional" PDF bezeichent, welches die stetige Verteilung eines Merkmals x für eine gegeben Klasse x_i beschreibt und besitzt alle Eigenschaften einer üblichen PDF.

Likelihood Falls man $p\left(x|\omega_j\right)$ als Funktion der Klasse j für ein festes x betrachtet, spricht man von der "Likelihood" von j bezüglich x.

$$P\left(\omega_j | x\right) = \frac{p\left(x | \omega_j\right) P\left(\omega_j\right)}{p(x)} \qquad \Longrightarrow \qquad posterior = \frac{likelihood \cdot prior}{evidence} \tag{22}$$

Evidence Die Evidence enspricht dem Normalisierungsfaktor. Im Falle, dass $p(x)$ für alle Klassen identisch ist, hat diese keinen Einfluss auf die posterior.

Likelihood-Verhältnis Im Fall von $c = 2$ (zwei Klassen), kann man die Bayes-Entscheidungsregel auch formulieren als,

$$P(\omega_1|x) > P(\omega_2|x) \tag{23}$$

$$p(x|\omega_1)\,P(\omega_1) > p(x|\omega_2)\,P(\omega_2) \tag{24}$$

$$\underbrace{\frac{p(x|\omega_1)}{p(x|\omega_2)}}_{\text{Likelihood-Verhältnis}} > \underbrace{\frac{P(\omega_2)}{P(\omega_1)}}_{\text{Schwellwert}} \tag{25}$$

Fehlerwahrscheinlichkeit Die Fehlerwahrscheinlichkeit die (bedingte) **Wahrscheinlichkeit der Fehlklassifikation** (conditional error) $P(error|x)$.

- $P(\omega_1|x)$, falls man für ω_2 entscheidet bzw.
- $P(\omega_2|x)$, falls man für ω_1 entscheidet

Fehlerrate Die Fehlerrate (oder der mittlere Fehler) $P(error)$ berechnet sich als

$$P(error) = \int_{-\infty}^{\infty} P(error|x)p(x)dx \tag{26}$$

3 Bayes-Theorem

Entscheidung mti "Loss" Dabei wird nach dem geringsten Loss klassifiziert.

$$\text{argmin}_j \sum_k L_{kj} P(\omega_k|x) \tag{27}$$

"Reject" Option Sollten die Posterior sehr nahe bei einander liege, z.B. bei zwei Klassen beide 0.5, dann kann bewusst keine Entscheidung getroffen werden.

3.1 Parameterschätzung für Bayes-Theorem

Parameterschätzung Bei der Parameterschätzung spricht man auch von *Training* oder *Supervised Learning*. Dabei werden mit Hilfe von Trainingsdaten die priors $P(\omega_j)$, evidence $p(x)$ und die class conditional PDFs $p(x|\omega_j)$ abgeschätzt. Das Ziel ist damit Testdaten richtig zu klassifizieren.

3.1.1 Schätzung der Priors

Die Schätzung basiert meist auf Klassenverteilung der Trainingsdaten oder/und zusätzliches Wissen über die Verteilung. Oft werden dabei einfach gleiche Priors fur alle Klassen angenommen.

3.1.2 Schätzung der Evidence

Die Evidence erhält man durch

$$p(x) = \sum_{j=1}^{c} p(x|\omega_j)\,P(\omega_j)\,, \tag{28}$$

dabei werden jedoch die class conditional PDFs benötigt.

3.1.3 Schätzung der Dichtefunktion

Das Schätzen der class conditional PDFs $p(x|\omega_j)$ ist am schwierigsten. Dafür gibt es verschiedene Verfahren welche man in **nicht-parametrische Verfahren** (keine fixe Anzahl an Parametern, keine Ann. über die Verteilung) und **parametrische Verfahren** (fixe Anzahl an Parametern, trifft zu Beginn Ann. über Verteilung) einteilen kann.

nicht-parametrische Verfahren Ein einfaches nicht-parametrische Verfahren zur Dichteschätzung ist das **kNN Estimation**.
Dabei werden Zellen über x angeordnet. Diese wachsen so lange bis k Stichproben enthalten sind. V_x gibt dabei die Größe der Zelle an (bei 1D ist dies die Länge der Linie). Somit gibt $\frac{1}{V_x}$ den Schätzer der Dichte. Die Form der Dichtefunktion ist bei diesem Verfahren sehr stark abhängig von dem Wert k. Auch ist die Dichtefunktion üblichweise nciht glatt. Dieses kNN-Schätzverfahren kann auch in zwei oder mehreren Dimensionen durchgeführt werden.
Eine Abwandlung des kNN-Schätzers ist der kNN-Klassifikator, welcher anstatt die Likelihood, direkt die posteriors, schätzt. Dabei werden zu einem Testdatenpunkt x genau k Trainingsdatenpunkte gefunden mit der kleinsten Distanz. Anhand diesen k Trainingsdatenpunkte kann je nach Klassenzugehörigkeit die posterior Wahrscheinlichkeit $P(\omega_j|x) = \frac{k_j}{k}$ berechnet werden.

parametrische Verfahren Bei parametrische Verfahren wird eine Annahme über die Dichtefunktion getroffen, dabei wird häufig eine Normalverteilung angenommen. Bei dieser Art von Verfahren werden weniger Parameter abgeschätzt und auch weniger Trainingsdaten benötigt, was jedoch sich negativ auf die Flexibilität auswirkt.
Sehr oft wird eine Normalverteilung angenommen.

$$p(x) \sim N\left(\mu, \sigma^2\right) \qquad p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{29}$$

$$\begin{aligned} P[|x-\mu| \leq \sigma] &\approx 0,68 \rightarrow 68\% \text{ der Daten} \\ P[|x-\mu| \leq 2\sigma] &\approx 0,95 \rightarrow 95\% \text{ der Daten} \\ P[|x-\mu| \leq 3\sigma] &\approx 0,997 \rightarrow \text{ mehr als } 99\% \text{ der Daten} \end{aligned}$$

Schätzung des Mittelwertes

$$\hat{\mu}_j = \frac{1}{N_j} \sum_{x_i \in \omega_j} x_i$$

Schätzung der Varianz

$$\hat{\sigma}_j^2 = \frac{1}{N_j} \sum_{x_i \in \omega_j} (x_i - \mu_j)^2 \qquad \text{falls } \mu_j \text{ bekannt}$$

$$\hat{\sigma}_j^2 = \frac{1}{N_j - 1} \sum_{x_i \in \omega_j} (x_i - \mu_j)^2 \qquad \text{falls } \mu_j \text{ nicht bekannt}$$

3.2 Abstecher: Erwartungstreue Schätzer

Eigenschaften von Schätzern Wünschenswerte Eigenschaften eines Schätzers $\hat{\theta}$ des Parameters θ, wären:

- Erwartungstreue: $\mathcal{E}[\hat{\theta}] = \theta$
- Effizienz: je geringer die Varianz des Schätzers $\sigma_{\hat{\theta}}^2$, desto effizienter ist er.
- Konsistenz: $\hat{\theta} \rightarrow \theta$ für $N \rightarrow \infty$ (“Gesetz der Großen Zahlen”)

Erwartungstreue ist gegeben, wegen

$$\mathcal{E}[\hat{\mu}] = \mathcal{E}\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N} \mathcal{E}\left[x_i\right] = \frac{1}{N} \cdot N \cdot \mu = \mu$$

und

$$\mathcal{E}\left[\hat{\sigma}^2\right] = \mathcal{E}\left[\frac{1}{N}\sum_{l=1}^{N} (x_i - \mu)^2\right] = \frac{1}{N}\sum_{l=1}^{N} \mathcal{E}\left[(x_i - \mu)^2\right] = \frac{1}{N} \cdot N \cdot \sigma^2 = \sigma^2.$$

Unkorrigierte Stichprobenvarianz - verzerrte Varianz:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 = \frac{(x_1 - \hat{\mu})^2 + \cdots + (x_N - \hat{\mu})^2}{N} \tag{30}$$

Die unkorrigierte Varianz ist nicht erwartungstreu, daher stammt der Besselsche Korrekturfaktor $\frac{N}{N-1}$.

Korrigierte Stichprobenvarianz - unverzerrte Varianz:

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu})^2 = \frac{N}{N-1} \cdot \frac{(x_1 - \hat{\mu})^2 + \cdots + (x_N - \hat{\mu})^2}{N} \tag{31}$$

Diese ist nun auch erwartungstreu.

3.3 Statistische Grundlagen für zwei oder mehrere

3.3.1 Bivariat

Erwartungswert Die Erwartung $\mathcal{E}[\cdot]$ einer bivariaten Funktion $h(x, y)$ von zwei diskreten Zufallsvariablen ist:

$$\mathcal{E}[h(x,y)] = \sum_{x\in X}\sum_{y\in Y} h(x,y)P(x,y) \tag{32}$$

Mittelwerte

$$\mu_x = \mathcal{E}[x] = \sum_{x\in X}\sum_{y\in Y} xP(x,y) \tag{33}$$

$$\mu_y = \mathcal{E}[y] = \sum_{x\in X}\sum_{y\in Y} yP(x,y) \tag{34}$$

Varianzen

$$\sigma_x^2 = \mathcal{E}\left[(x-\mu_x)^2\right] = \sum_{x\in X}\sum_{y\in Y}(x-\mu_x)^2 P(x,y) \tag{35}$$

$$\sigma_y^2 = \mathcal{E}\left[(y-\mu_y)^2\right] = \sum_{x\in X}\sum_{y\in Y}(y-\mu_y)^2 P(x,y) \tag{36}$$

Kovarianz Die Kovarianz ist ein Maß für die statistische Abhängigkeit zwischen Zufallsvariablen und gegeben durch

$$\sigma_{xy} = \mathcal{E}\left[(x-\mu_x)(y-\mu_y)\right] = \sum_{x\in X}\sum_{y\in Y}(x-\mu_x)(y-\mu_y)P(x,y), \tag{37}$$

mit dessen Schätzung

$$\hat{\sigma}_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \hat{\mu}_x)(y_i - \hat{\mu}_y) \tag{38}$$

Wenn die Kovarianz $\sigma_{xy} = 0$ ist kann dies zwei Bedeutungen haben. (1) x und y sind unabhängig oder (2) x und y sind unkorreliert. Ist $\sigma_{xy} > 0$, so sind x und y positiv korreliert und wachensen sie bzw. verkleinern sich gemeinsam. Bei $\sigma_{xy} < 0$ sind sie negativ korreliert, was bedeutet wenn x sich vergrößert so wird y kleiner und umgekehrt.

3.3.2 Multivariat

Merkmalsvektor

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \in \mathbb{R}^p \tag{39}$$

Mittelwertsvektor

$$\boldsymbol{\mu} = \mathcal{E}[\boldsymbol{x}] = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix} \in \mathbb{R}^p = \begin{bmatrix} \mathcal{E}[x]_1 \\ \mathcal{E}[x]_2 \\ \vdots \\ \mathcal{E}[x]_p \end{bmatrix} \qquad \hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{40}$$

Kovarianzmatrix Eine Kovarianzmatrix $\boldsymbol{\Sigma}$ setzt sich aus zentralen Momenten 2. Ordnung zusammen. Die Elemente σ_{ij} bezeichnet man als Varianz wenn $(i = j)$ und als Kovarianz wenn $(i \neq j)$.

$$\begin{aligned} &\sigma_{ij} = \mathcal{E}\left[(x_i - \mu_i)\,(x_j - \mu_j)\right] \\ &\Rightarrow \qquad \mathrm{Cov}(\boldsymbol{x}) = \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_{pp} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1p} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \cdots & \sigma_p^2 \end{bmatrix} = \mathcal{E}\left[(\boldsymbol{x} - \boldsymbol{\mu})((\boldsymbol{x} - \boldsymbol{\mu}))^T\right] \end{aligned} \tag{41}$$

Schätzer der Kovarianzmatrix:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T \tag{42}$$

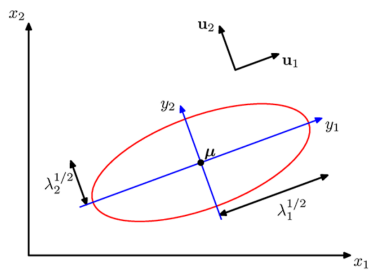
3.4 Multivariate Normalverteilungen

$$p(\boldsymbol{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \qquad p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]} \tag{43}$$

Wenn $\boldsymbol{\Sigma}$ singulär ist, d.h. $|\boldsymbol{\Sigma}| = 0$ ist, dann existiert keine Inverse.
$\boldsymbol{x}$: p-dimensionaler Merkmalsvektor (Spaltenvektor)
$\boldsymbol{\mu}$: p-dimensionaler Mittelwertvektor (Spaltenvektor)
$\boldsymbol{\Sigma}$: $p \times p$-dimensionale Kovarianzmatrix
$|\boldsymbol{\Sigma}|$: Determinante von $\boldsymbol{\Sigma}$
$\boldsymbol{\Sigma}^{-1}$: Inverse von $\boldsymbol{\Sigma}$
$(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$: Mahalanobisdistanz

Mahalanobisdistanz

$$d^2(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}) \tag{44}$$



- alle Punkte $\{\boldsymbol{x} : d^2(\boldsymbol{x}) = c\}$ liegen auf einem Hyperellipsoid im $\mathcal{R}^p$ mit Mittelpunkt $\boldsymbol{\mu}$
- für alle diese $\boldsymbol{x}$ liefert die Dichtefunktion $p(\boldsymbol{x})$ denselben Wert
- Hauptachsen der Ellipse sind definiert durch Eigenvektoren $\boldsymbol{u}_1$ und $\boldsymbol{u}_2$ von $\boldsymbol{\Sigma}$
- Eigenwerte λ_1 und λ_2 bestimmen die Länge der Achsen

3.5 Bayes-Theorem für multivariate Normalverteilungen

Hypothese Die Dichtefunktion $p(x|\omega_j)$ jeder Klasse j kann durch eine multivariate Normalverteilung $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ beschrieben werde.

Parameterschätzung Dabei werden aus Trainingsdaten der Mittelwertvektor $\hat{\boldsymbol{\mu}}_j$ und die Kovarianzmatrix $\hat{\boldsymbol{\Sigma}}_j$ geschätzt. Um die Anzahl an Parametern zu reduzieren können Annahmen über die Kovarianzmatrix getroffen werden.

Naive Bayes-Klassifikator Der Naive Bayes-Klassifikator nimmt an, dass es keine Korrelation zwischen den Merkmalen gibt, d.h. die off-diagonale der Kovarianzmatrix sind 0.

4 Diskriminantenfunktionen. Merkmale und ihre Dimensionalität

4.1 Diskriminantenfunktionen

Ein Klassifikator kann als eine Gruppe von Diskriminantenfunktionen $g_i(\boldsymbol{x}), i = 1, \ldots, c$ dargestellt werden. Der Klassifikator weist den Merkmalsvektor $\boldsymbol{x}$ der Klasse ω_i zu, wenn gilt: $g_i(\boldsymbol{x}) > g_j(\boldsymbol{x})$ für alle $j \neq i$.
Man kann sich einen Klassifikator als ein Netzwerk vorstellen, dass c Diskriminantenfunktionen berechnet und die Klasse mit dem größten Diskriminanten auswählt.
Die Diskriminantenfunktionen unterteilen den Merkmalsraum in c **Entscheidungsregionen** $\mathcal{R}_1, \ldots, \mathcal{R}_c$. Wenn $g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}$ für alle $j \neq i$, dann liegt $\boldsymbol{x}$ in $\mathcal{R}_i$, d.h. $\boldsymbol{x}$ wird der Klasse j zugelwiesen. Entscheidungsregionen sind durch Entscheidungsgrenzen ("decision boundaries") getrennt. Diese Entscheidungsgrenze zwischen $\mathcal{R}_i$ und $\mathcal{R}_j$ ist durch die Gleichung $g_i(\boldsymbol{x}) = g_j(\boldsymbol{x})$ gegeben.

Zwei Klassen Im Fall von 2 Klassen ist es üblich nur eine Diskriminantenfunktion zu definieren $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$. Diese Funktion entscheidet für w_1 wenn $g(\boldsymbol{x}) > 0$, ansonst für Klasse w_2.
Die Wahl der Diskriminantenfunktion ist nicht eindeutig und kann durch eine kluge Wahl optimiert werden.

$$g_i(\boldsymbol{x}) = P(\omega_i|\boldsymbol{x}) \tag{45}$$

$$g_i(\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)P(\omega_i) \tag{46}$$

$$g_i(\boldsymbol{x}) = \ln\left(\frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}\right) \tag{47}$$

Diese Diskriminantenfunktionen führen alle zum gleichen Ergebnis, sind jedoch nicht gleich einfach zu verstehen bzw. zu berechnen.
Für multivariate Normalverteilungen unter Verwendung des Bayes-Klassifikators kann durch logarithmieren der Prosteriors die Diskriminantenfunktion stark vereinfacht werden.

$$\begin{aligned} g_j(\boldsymbol{x}) &= \ln\left(\frac{p(\boldsymbol{x}|\omega_j)P(\omega_j)}{p(\boldsymbol{x})}\right) = \ln\left(\frac{\mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_j)^T\hat{\boldsymbol{\Sigma}}_j^{-1}(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_j)}}{(2\pi)^{\frac{p}{2}}\ |\hat{\boldsymbol{\Sigma}}_j|^{\frac{1}{2}}}\frac{P(\omega_i)}{p(\boldsymbol{x})}\right) \\ &= -\frac{1}{2}\left(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_j\right)^T\hat{\boldsymbol{\Sigma}}_j^{-1}(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_j) - \frac{1}{2}\ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right) + \ln\left(P(\omega_i)\right) - \frac{p}{2}\ln\left(2\pi\right) - \ln\left(p(\boldsymbol{x})\right) \end{aligned} \tag{48}$$

Die letzten Terme können weggelassen werden, da sie unabhängig von ω_j sind. Der erste Term kann durch die Mahalanobisdistanz ausgedruckt werden. Dies führt zu

$$g_j(\boldsymbol{x}) = -\frac{1}{2}d_j^2(\boldsymbol{x}) - \frac{1}{2}\ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right) + \ln\left(P(\omega_i)\right) \tag{49}$$

Hierbei kann man jetzt in drei verschiedene Fälle unterscheiden.

1. **Allgemeiner Fall** - unterschiedliche Kovarianzen, d.h. $\Sigma_i \neq \Sigma_j$
2. **Spezialfall 2** - identische Kovarianzen, d.h. $\Sigma_i = \Sigma_j$
3. **Spezialfall 1** - identische und isotrope Kovarianzen, d.h. $\Sigma_i = \Sigma_j = \sigma^2\boldsymbol{I}$

Allgemeiner Fall

- beliebige Kovarianzmatrix $\boldsymbol{\Sigma}$
- Entscheidungsgrenzen sind durch so genannte Hyperquadrics gegeben
- Hyperquadrics sind unter anderem:
  Hyperebenen, Hyperkugeln, Hyperellipsoide, Hyperparaboloide, etc.
- korrespondierenden Entscheidungsregionen müssen nicht einfach zusammenhängend sein

Spezialfall 2 (hier nicht im Detail behandelt)

- alle Klassen haben dieselbe Kovarianzmatrix
- Form der Verteilungen ist durch Hyperellipsoide in $\mathcal{R}^p$ gegeben
- Entscheidungsgrenzen sind linear
- jedoch nicht normal zur Verbindungsstrecke zwischen den beiden Klassenmitteln

Spezialfall 1 Die Diskriminantenfunktion kann in diesem Fall stark vereinfacht werden. Da sich die Kovarianzmatrix auf eine Dimmenson, also eine Konstante, reduziert kann darauf geschlossen werden, dass es sich dabei um eine lineare Entscheidungsgrenze handelt. Dies kann auch folgendermaßen gezeigt werden.

$$\begin{aligned} g_j(\boldsymbol{x}) &= -\frac{1}{2\sigma^2}\left(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j\right)^T\left(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j\right) - \frac{1}{2}\ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right) + \ln\left(P(\omega_i)\right) \\ &= -\frac{1}{2\sigma^2}\left(\boldsymbol{x}^T\boldsymbol{x} - 2\hat{\boldsymbol{\mu}}_j^T\boldsymbol{x} + \hat{\boldsymbol{\mu}}_j^T\hat{\boldsymbol{\mu}}_j\right) - \frac{1}{2}\ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right) + \ln\left(P(\omega_i)\right) \end{aligned} \tag{50}$$

Zwei dieser Terme können weggelassen werden, da sie in beiden Diskriminantenfunktionen enthalten sind, das ist der $\boldsymbol{x}^T\boldsymbol{x}$ Term und der $\ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right)$ Term, da die Kovarianzmatrix/Kovarianzen für beide Klassen gleich sind.

$$g_j(\boldsymbol{x}) = -\frac{1}{2\sigma^2}\left(-2\hat{\boldsymbol{\mu}}_j^T\boldsymbol{x} + \hat{\boldsymbol{\mu}}_j^T\hat{\boldsymbol{\mu}}_j\right) + \ln\left(P(\omega_i)\right) \tag{51}$$

Durch umdefinieren mancher Konstanten kann die Formel auf die Darstellung einer Geradengleichen gebracht werden.

$$g_j(\boldsymbol{x}) = \left(\frac{1}{\sigma^2}\hat{\boldsymbol{\mu}}_j^T\right)\boldsymbol{x} + \left(-\frac{1}{2\sigma^2}\hat{\boldsymbol{\mu}}_j^T\hat{\boldsymbol{\mu}}_j + \ln\left(P(\omega_i)\right)\right) = \boldsymbol{w}_j^T\boldsymbol{x} + w_{j0} \tag{52}$$

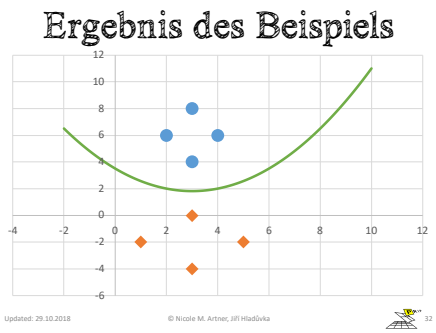
Example 3: Bestimmung der Diskriminantenfunktionen und der Entscheidungsgrenze

Merkmalsvektoren der Klasse 1 (K1):

$$\begin{pmatrix}2\\4\end{pmatrix}, \begin{pmatrix}3\\8\end{pmatrix}, \begin{pmatrix}2\\6\end{pmatrix}, \begin{pmatrix}4\\6\end{pmatrix}$$

Klasse 2 (K2):

$$\begin{pmatrix}3\\0\end{pmatrix}, \begin{pmatrix}3\\-4\end{pmatrix}, \begin{pmatrix}1\\-2\end{pmatrix}, \begin{pmatrix}5\\-2\end{pmatrix}$$



$$\hat{\boldsymbol{\mu}}_1 = \frac{1}{N_{\mathrm{K1}}} \sum_{i=1}^{N_{\mathrm{K1}}} \boldsymbol{x}_i^{(\mathrm{K1})} = \begin{pmatrix}3\\6\end{pmatrix} \qquad \text{bzw.} \qquad \hat{\boldsymbol{\mu}}_2 = \frac{1}{N_{\mathrm{K2}}} \sum_{i=1}^{N_{\mathrm{K2}}} \boldsymbol{x}_i^{(\mathrm{K2})} = \begin{pmatrix}3\\-2\end{pmatrix}$$

Schätzung der Kovarianzmatrizen mit unbekannten $\boldsymbol{\mu}_i$, d.h. nur geschätzter Mittelwert wird verwendet

$$\hat{\boldsymbol{\Sigma}}_i = \frac{1}{N_{\mathrm{K}i} - 1} \sum_{j=1}^{N_{\mathrm{K}i}} \left(\boldsymbol{x}_j^{\mathrm{K}i} - \hat{\boldsymbol{\mu}}_i\right) \left(\boldsymbol{x}_j^{\mathrm{K}i} - \hat{\boldsymbol{\mu}}_i\right)^T \quad \Rightarrow \quad \hat{\boldsymbol{\Sigma}}_1 = \begin{pmatrix}\frac{2}{3} & 0\\ 0 & \frac{8}{3}\end{pmatrix} \qquad \text{bzw.} \qquad \hat{\boldsymbol{\Sigma}}_2 = \begin{pmatrix}\frac{8}{3} & 0\\ 0 & \frac{8}{3}\end{pmatrix}$$

Weiters nehmen wir an das die vorhin berechneten Mittelwerte die tatsächlichen Mittelwerte sind, d.h. $\boldsymbol{\mu}_i = \hat{\boldsymbol{\mu}}_i$. Dies ändert die Kovarianzmatrizen folgendermaßen

$$\boldsymbol{\Sigma}_i = \frac{1}{N_{\mathrm{K}i}} \sum_{j=1}^{N_{\mathrm{K}i}} \left(\boldsymbol{x}_j^{\mathrm{K}i} - \boldsymbol{\mu}_i\right) \left(\boldsymbol{x}_j^{\mathrm{K}i} - \boldsymbol{\mu}_i\right)^T$$

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix}\frac{1}{2} & 0\\ 0 & 2\end{pmatrix} \qquad \text{bzw.} \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix}2 & 0\\ 0 & 2\end{pmatrix}$$

Die Determinanten und Inversen lassen sind leicht zu berechnen und sehen wie folgt aus.

$$|\boldsymbol{\Sigma}_1| = 1, \quad \boldsymbol{\Sigma}_1^{-1} = \begin{pmatrix}2 & 0\\ 0 & \frac{1}{2}\end{pmatrix} \qquad \text{bzw.} \qquad |\boldsymbol{\Sigma}_2| = 4, \quad \boldsymbol{\Sigma}_2^{-1} = \begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}$$

Mit diesen Ergebnissen und der Annahme von $P(\omega_1) = P(\omega_2) = 0.5$ kann nun eine Diskriminantenfunktion bestimmt werden.

$$g_j(\boldsymbol{x}) = -\frac{1}{2} d_j^2(\boldsymbol{x}) - \frac{1}{2} \ln\left(|\hat{\boldsymbol{\Sigma}}_j|\right) + \ln\left(P(\omega_i)\right)$$

mit $d_j^2(\boldsymbol{x}) = \left(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j\right)^T \left(\boldsymbol{x} - \hat{\boldsymbol{\mu}}_j\right)$.

$$g_1(\boldsymbol{x}) = -x_1^2 + 6x_1 - \frac{1}{4}x_2^2 + 3x_2 - 18 + \ln(0.5) g_2(\boldsymbol{x}) \quad = -\frac{1}{4}x_1^2 + \frac{3}{2}x_1 - \frac{1}{4}x_2^2 - x_2 - \frac{13}{4} - \ln(2) + \ln(0.5)$$

Durch das Gleichsetzen von $g_1(x)$ und $g_2(x)$ kann nun eine Entscheidungsgrenze bestimmt werden.

$$\begin{aligned}
g_1(x) &= g_2(x) \\
-x_1^2 + 6x_1 - \frac{1}{4}x_2^2 + 3x_2 - 18 + \ln(0.5) &= -\frac{1}{4}x_1^2 + \frac{3}{2}x_1 - \frac{1}{4}x_2^2 - x_2 - \frac{13}{4} - \ln(2) + \ln(0.5) \\
-\frac{3}{4}x_1^2 + \frac{9}{2}x_1 &= -4x_2 + \frac{59}{4} - \ln(2) \\
x_2 &= \frac{1}{4}\left[\frac{3}{4}x_1^2 - \frac{9}{2}x_1 + \frac{59}{4} - \ln(2)\right]
\end{aligned}$$

4.1.1 Allgemeiner Fall und Spezialfall 1

Beispiel: Angabe

Aufgabe: Ermittle die Diskriminantenfunktionen von Klasse A und B und ihre Entscheidungsgrenze…

1. für den allgemeinen Fall
2. für Spezialfall 1

Annahmen:

$p(\boldsymbol{x}|\omega_j)$ → multivariate Normalverteilungen

$P(w_A) = P(w_B) = 0{,}5$ → gleiche Priors

für Spezialfall 1: $\Sigma_A = \Sigma_B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Beispiel: Angabe

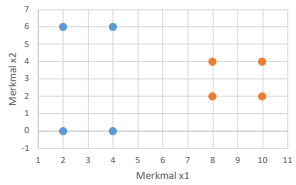
Trainingsdaten:

Klasse A

$\begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 6 \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}$

Klasse B

$\begin{bmatrix} 8 \\ 2 \end{bmatrix}, \begin{bmatrix} 8 \\ 4 \end{bmatrix}, \begin{bmatrix} 10 \\ 2 \end{bmatrix}, \begin{bmatrix} 10 \\ 4 \end{bmatrix}$



Beispiel: Lösung

1. Schätzung der Mittelwertvektoren

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

$$\hat{\boldsymbol{\mu}}_A = \left(\begin{bmatrix} 2 \\ 6 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 4 \\ 6 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \end{bmatrix}\right) \cdot \frac{1}{4} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$\hat{\boldsymbol{\mu}}_B = \left(\begin{bmatrix} 8 \\ 2 \end{bmatrix} + \begin{bmatrix} 8 \\ 4 \end{bmatrix} + \begin{bmatrix} 10 \\ 2 \end{bmatrix} + \begin{bmatrix} 10 \\ 4 \end{bmatrix}\right) \cdot \frac{1}{4} = \begin{bmatrix} 9 \\ 3 \end{bmatrix}$$

Beispiel: Lösung

… für allgemeinen Fall

2. Schätzen der Kovarianzmatrizen

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}})^T$$

Beispiel: Lösung

… für allgemeinen Fall

2. Schätzen der Kovarianzmatrizen

$$\hat{\Sigma}_A = \frac{1}{4-1}\left\{\left(\begin{bmatrix} 2 \\ 6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 2 \\ 6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 4 \\ 6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 4 \\ 6 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 4 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 4 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 3 \end{bmatrix}\right)^T\right\} =$$

$$= \frac{1}{3}\left\{\begin{bmatrix} -1 \\ 3 \end{bmatrix}\begin{bmatrix} -1 & 3 \end{bmatrix} + \begin{bmatrix} -1 \\ -3 \end{bmatrix}\begin{bmatrix} -1 & -3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \end{bmatrix}\begin{bmatrix} 1 & 3 \end{bmatrix} + \begin{bmatrix} 1 \\ -3 \end{bmatrix}\begin{bmatrix} 1 & -3 \end{bmatrix}\right\} = \frac{1}{3}\left\{\begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix} + \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} + \begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix} + \begin{bmatrix} 1 & -3 \\ -3 & 9 \end{bmatrix}\right\} =$$

$$= \frac{1}{3}\begin{bmatrix} 4 & 0 \\ 0 & 36 \end{bmatrix} = \begin{bmatrix} 4/3 & 0 \\ 0 & 12 \end{bmatrix}$$

Beispiel: Lösung

… für allgemeinen Fall

2. Schätzen der Kovarianzmatrizen

$$\hat{\Sigma}_B = \frac{1}{4-1}\left\{\left(\begin{bmatrix} 8 \\ 2 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 8 \\ 2 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 8 \\ 4 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 8 \\ 4 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 10 \\ 2 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 10 \\ 2 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)^T + \left(\begin{bmatrix} 10 \\ 4 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)\left(\begin{bmatrix} 10 \\ 4 \end{bmatrix} - \begin{bmatrix} 9 \\ 3 \end{bmatrix}\right)^T\right\} =$$

$$= \frac{1}{3}\left\{\begin{bmatrix} -1 \\ -1 \end{bmatrix}\begin{bmatrix} -1 & -1 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -1 \end{bmatrix}\begin{bmatrix} 1 & -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix}\right\} =$$

$$= \frac{1}{3}\left\{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\right\} = \frac{1}{3}\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} =$$

$$= \begin{bmatrix} 4/3 & 0 \\ 0 & 4/3 \end{bmatrix}$$

Beispiel: Lösung

… für allgemeinen Fall

3. Determinanten ermitteln

$$|\boldsymbol{\Sigma}| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$|\boldsymbol{\Sigma}|_A = \begin{vmatrix} 4/3 & 0 \\ 0 & 12 \end{vmatrix} = 4/3 \cdot 12 = 16$$

$$|\boldsymbol{\Sigma}|_B = \begin{vmatrix} 4/3 & 0 \\ 0 & 4/3 \end{vmatrix} = 4/3 \cdot 4/3 = 16/9$$

Beispiel: Lösung

… für allgemeinen Fall

4. Inverse berechnen

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{-1}{}_A = \frac{1}{16}\begin{bmatrix} 12 & 0 \\ 0 & 4/3 \end{bmatrix} = \begin{bmatrix} 3/4 & 0 \\ 0 & 1/12 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{-1}{}_B = \frac{9}{16}\begin{bmatrix} 4/3 & 0 \\ 0 & 4/3 \end{bmatrix} = \begin{bmatrix} 3/4 & 0 \\ 0 & 3/4 \end{bmatrix}$$

Beispiel: Lösung

… für allgemeinen Fall

5. Diskriminantenfunktionen bestimmen

$$g_j(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_j)^T\Sigma_j^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_j) - \frac{1}{2}\ln|\Sigma_j| + \cancel{\ln P(\omega_j)}$$

$$g_A(\boldsymbol{x}) = -\frac{1}{2}\left\{\left(\begin{pmatrix}x_1\\x_2\end{pmatrix}-\begin{pmatrix}3\\3\end{pmatrix}\right)^T\begin{pmatrix}3/4 & 0\\0 & 1/12\end{pmatrix}\left(\begin{pmatrix}x_1\\x_2\end{pmatrix}-\begin{pmatrix}3\\3\end{pmatrix}\right)\right\} - \frac{1}{2}\ln(16)$$

$$((x_1-3)\ (x_2-3))\begin{pmatrix}3/4 & 0\\0 & 1/12\end{pmatrix}\begin{pmatrix}(x_1-3)\\(x_2-3)\end{pmatrix} =$$

$$\left(\left(\frac{3x_1}{4}-\frac{9}{4}\right)\ \left(\frac{x_2}{12}-\frac{1}{4}\right)\right)\begin{pmatrix}(x_1-3)\\(x_2-3)\end{pmatrix} =$$

$$\left(\frac{3x_1}{4}-\frac{9}{4}\right)(x_1-3)+\left(\frac{x_2}{12}-\frac{1}{4}\right)(x_2-3) =$$

$$3/4\,x_1^2 - 9/4\,x_1 - 9/4\,x_1 + 27/4 + 1/12\,x_2^2 - 1/4\,x_2 - 3/12\,x_2 + 3/4 =$$

$$3/4\,x_1^2 - 18/4\,x_1 + 1/12\,x_2^2 - 7/4\,x_2 + 30/4 = d_A^2$$

Beispiel: Lösung

… für allgemeinen Fall

5. Diskriminantenfunktionen bestimmen

$$g_A(\boldsymbol{x}) = -1/2\ d_A^2 - 1/2\ \ln(16) = -3/8\,x_1^2 + 18/8\,x_1 - 1/24\,x_2^2 + 7/8\,x_2 - 30/8 -$$
$$1/2\ \ln(16) = \boxed{-3/8\,x_1^2 + 9/4\,x_1 - 1/24\,x_2^2 + 7/4\,x_2 - 15/4 - 1/2\ \ln(16)}$$

$$g_B(\boldsymbol{x}) = -1/2\left\{\left(\begin{pmatrix}x_1\\x_2\end{pmatrix}-\begin{pmatrix}9\\3\end{pmatrix}\right)^T\begin{pmatrix}3/4 & 0\\0 & 3/4\end{pmatrix}\left(\begin{pmatrix}x_1\\x_2\end{pmatrix}-\begin{pmatrix}9\\3\end{pmatrix}\right)\right\} - 1/2\ln(16/9)$$

$$((x_1-9)\ (x_2-3))\begin{pmatrix}3/4 & 0\\0 & 3/4\end{pmatrix}\begin{pmatrix}(x_1-9)\\(x_2-3)\end{pmatrix} =$$

$$\left(\left(\frac{3x_1}{4}-\frac{27}{4}\right)\ \left(\frac{3x_2}{4}-\frac{9}{4}\right)\right)\begin{pmatrix}(x_1-9)\\(x_2-3)\end{pmatrix} =$$

$$\left(\frac{3x_1}{4}-\frac{27}{4}\right)(x_1-9)+\left(\frac{3x_2}{4}-\frac{9}{4}\right)(x_2-3) =$$

$$3/4\,x_1^2 - 27/4\,x_1 - 27/4\,x_1 + 243/4 + 3/4\,x_2^2 - 9/4\,x_2 - 9/4\,x_2 + 27/4 =$$

$$3/4\,x_1^2 - 54/4\,x_1 + 3/4\,x_2^2 - 18/4\,x_2 + 270/4 = d_B^2$$

Beispiel: Lösung

… für allgemeinen Fall

5. Diskriminantenfunktionen bestimmen

$$g_B(\boldsymbol{x}) = -1/2\ d_B^2 - 1/2\ \ln(16/9) = -3/8\,x_1^2 + 54/8\,x_1 + 3/8\,x_2^2 + 18/8\,x_2 + 270/8 -$$
$$1/2\ \ln(16/9) = \boxed{-3/8\,x_1^2 + 27/4\,x_1 - 3/8\,x_2^2 + 9/4\,x_2 - 135/4 - 1/2\ \ln(16/9)}$$

Beispiel: Lösung

… für allgemeinen Fall

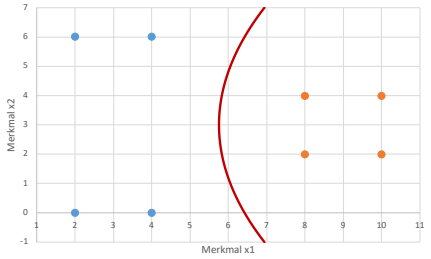
6. Entscheidungsgrenze berechnen

$$g_A(\boldsymbol{x}) = g_B(\boldsymbol{x}) \qquad 0 = g_A(\boldsymbol{x}) - g_B(\boldsymbol{x})$$

$$0 = -3/8\,x_1^2 + 9/4\,x_1 - 1/24\,x_2^2 + 7/4\,x_2 - 15/4 - 1/2\ \ln(16) + 3/8\,x_1^2 - 27/4\,x_1 + 3/8\,x_2^2 -$$
$$9/4\,x_2 + 135/4 + 1/2\ \ln(16/9) = -18/4\,x_1 + 8/24\,x_2^2 - 8/4\,x_2 + 120/4 -$$
$$1/2\ \ln(16) + 1/2\ \ln(16/9) = -9/2\,x_1 + 1/3\,x_2^2 - 2\,x_2 - 1/2\ \ln(16) +$$
$$1/2\ \ln(16/9) + 30 = \boxed{-9/2\,x_1 + 1/3\,x_2^2 - 2\,x_2 + 28{,}9014}$$

Beispiel: Ergebnis

… für allgemeinen Fall



Beispiel: Lösung

… für Spezialfall 1

2. Diskriminantenfunktionen ermitteln

$$g_j(\boldsymbol{x}) = -\frac{1}{2\sigma^2}(-2\boldsymbol{\mu}_j^T\boldsymbol{x} + \boldsymbol{\mu}_j^T\boldsymbol{\mu}_j) + \cancel{\ln P(\omega_j)}$$

$$g_A(\boldsymbol{x}) = -1/2\ \left(-2\,(3\ 3)\begin{pmatrix}x_1\\x_2\end{pmatrix} + (3\ 3)\begin{pmatrix}3\\3\end{pmatrix}\right) =$$

$$-1/2\ \left((-6\ -6)\begin{pmatrix}x_1\\x_2\end{pmatrix} + 18\right) =$$

$$-1/2\ (-6x_1 - 6x_2 + 18) =$$

$$\boxed{3x_1 + 3x_2 - 9}$$

Beispiel: Lösung

… für Spezialfall 1

2. Diskriminantenfunktionen ermitteln

$$g_B(\boldsymbol{x}) = -1/2\ \left(-2\,(9\ 3)\begin{pmatrix}x_1\\x_2\end{pmatrix} + (9\ 3)\begin{pmatrix}9\\3\end{pmatrix}\right) =$$

$$-1/2\ \left((-18\ -6)\begin{pmatrix}x_1\\x_2\end{pmatrix} + 90\right) =$$

$$-1/2\ (-18x_1 - 6x_2 + 90) =$$

$$\boxed{9x_1 + 3x_2 - 45}$$

Beispiel: Lösung

… für Spezialfall 1

3. Entscheidungsgrenze berechnen

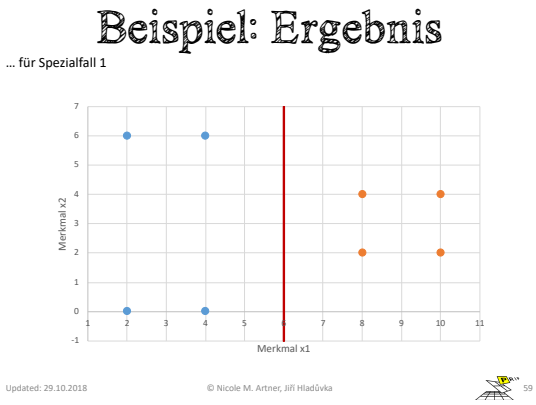
$$g_A(\boldsymbol{x}) = g_B(\boldsymbol{x}) \qquad 0 = g_A(\boldsymbol{x}) - g_B(\boldsymbol{x})$$

$$3x_1 + 3x_2 - 9 = 9x_1 + 3x_2 - 45$$

$$-6x_1 - 9 = -45$$

$$-6x_1 = -36$$

$$\boxed{x_1 = 6}$$



4.1.2 Fluch der Dimensionalität

Curse of dimensionality Der Fluch der Dimensionalität ist das exponentielle Wachstum der Möglichkeiten als eine Funktion der Dimensionalität. Wenn die Dimensionalität wächst

- dann ist eine **exponentiell wachsende Menge an Daten** notwendig, um den Merkmalsraum abzudecken und Training zu ermöglichen;
- dann wächst auch der **Rechenaufwand**;

(Dazu gibs noch mehr aber bin mir nicht sicher ob es Prüfungsrelevant ist.)

4.1.3 Merkmalsselektion

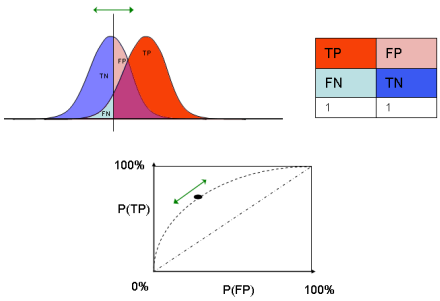
Grundregel Wähle Merkmale, sodass im Merkmalsraum

- der Abstand zwischen unterschiedlichen Klassen groß ist
- und die Varianz innerhalb der Klassen klein

Ziel ist die beste Auswahl einer Teilmenge aus allen Merkmalen zu bestimmen. Eine Entscheidungskriterien für Merkmale Trennbarkeit der Klassen, d.h. unabhängige Klassifikatoren, und die Perfomance des Klassifikators (Fehlerrate).

Entscheidungskriterium

- ROC – Receiver Operating Characteristic



- FDR – Fishre's discriminant ratio Je größer der FDR Wert eines Merkmales ist desto besser.

$$FDR = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \qquad \text{für 2 Klassen} \tag{53}$$

$$FDR_1 = \sum_i^M \sum_{j \neq i}^M \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \qquad \text{für M Klassen} \tag{54}$$

Auswahlprozess z.B. Sequential Backwards Selection oder Sequential Forward Selection

5 Merkmalsreduktion. Perceptron

5.1 Merkmalsreduktion

Man spricht von einer Merkmalsreduktion, wenn man die Dimensionalität und die damit verbundenen Probleme reduziert in dem man Merkmale geschickt kombiniert.
Mögliche Verfahren sind

- PCA – Principal Component Analysis, auch Hauptkomponentenanalyse genannt. Dies erzeugt unkorrelierte Merkmale.
- ICA – Independant Component Analysis. Erzeugt unabhängige Merkmale

5.1.1 PCA - Hauptkomponentenanalyse

$$\boldsymbol{x} = \boldsymbol{U}\boldsymbol{y} \qquad \Leftrightarrow \qquad \boldsymbol{y} = \boldsymbol{U}^{-1}\boldsymbol{x}$$

Verwendung einer linearen Transformation um reduzierte Basis $\boldsymbol{U}$ zu finden.

$$\boldsymbol{x} = \sum_{i=1}^{p} y_i \boldsymbol{u}_i \qquad \text{fehlerfreie Rekonstruktion}$$

$$\hat{\boldsymbol{x}}(m) = \sum_{i=1}^{m} y_i \boldsymbol{u}_i, \quad \boldsymbol{x} = \hat{\boldsymbol{x}}(m) + \varepsilon^2 \qquad \varepsilon\text{: Rekonstruktionsfehler} \tag{55}$$

Die Hauptkomponentenanalyse (PCA) findet eine reduzierte Basis $\boldsymbol{U}$ die den Rekonstruktionsfehler ε^2 minimiert, weil sie als Basisvektoren die Eigenvektoren (EV) $\boldsymbol{u}_i$ mit den n größten Eigenwerten (EW) λ_i der Kovarianzmatrix $\hat{\boldsymbol{\Sigma}}$ des ursprünglichen Merkmalsvektors $\boldsymbol{x}$ verwendet.

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \qquad \Rightarrow \qquad \boldsymbol{A}\boldsymbol{v} \text{ ergibt vielfachens } \lambda \text{ von } \boldsymbol{v} \text{ daher ist } \boldsymbol{v} \text{ EV von } \boldsymbol{A} \tag{56}$$

Die Koeffizienten y_i werden folgendermaßen berechnet>

$$y_i = \boldsymbol{u}_i^T \boldsymbol{x} \qquad \boldsymbol{y} = \boldsymbol{U}^T \boldsymbol{x} \tag{57}$$

Anwendungen Diese Methode kann zur Gesichts- oder Objekterkennung verwendet werden. Dabei müssen nur die Eigenwerte gespeichert werden, da die Eigenvektoren für alle Daten gelich bleiben.

Implementierungsablauf Als erster muss der Mittelwertvektor, aller Daten, geschätzt werden.

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x} \tag{58}$$

Meist werden danach differenz Vektoren gebildet

$$\boldsymbol{\Phi}_k = \boldsymbol{x}_k - \hat{\boldsymbol{\mu}}, \tag{59}$$

womit des weiteren die Kovarianzmatrix abgeschätzt wird.

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{k=1}^{N} \boldsymbol{\Phi}\boldsymbol{\Phi}^T \tag{60}$$

Von der Kovarianzmatrix können nun die Eigenwerte λ_i bestimmt werden.

$$\det\left(\hat{\boldsymbol{\Sigma}} - \lambda \boldsymbol{I}\right) = 0 \tag{61}$$

Mittel der Eigenwertgleichung und dem bestimmten Eigenwerten λ_i können die gesuchten Eigenvektoren $\boldsymbol{u}_i$ berechnet werden.

$$\hat{\boldsymbol{\Sigma}}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \quad \Rightarrow \quad \left(\hat{\boldsymbol{\Sigma}} - \lambda_i \boldsymbol{I}\right)\boldsymbol{u}_i = \vec{0} \tag{62}$$

Die Transformationsmatrix $\boldsymbol{U}$ ist durch die Eigenvektoren $\boldsymbol{u}_i$ gegeben.

$$\boldsymbol{U} = \begin{pmatrix} \vdots & & \vdots \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_N \\ \vdots & & \vdots \end{pmatrix} \tag{63}$$

Example 4: Berechnung der Eigenwerte, Eigenvektoren

$x_{1,k}$	2.5	0.5	2.2	1.9	3.1	2.3	2	1	1.5	1.1
$x_{2,k}$	2.4	0.7	2.9	2.2	3.0	2.7	1.6	1.1	1.6	0.9

Der dazugehörige Mittelwertvektor ist

$$\hat{\boldsymbol{\mu}} = \begin{pmatrix} 1.81 \\ 1.91 \end{pmatrix}$$

$\Phi_{1,k}$	0.69	−1.31	0.39	0.09	1.29	0.49	0.19	−0.81	−0.31	−0.71
$\Phi_{2,k}$	0.49	−1.21	0.99	0.29	1.09	0.79	−0.31	−0.81	−0.31	−1.01

Damit kann die Kovarianzmatrix bestimmt werden.

$$\hat{\boldsymbol{\sigma}} = \begin{pmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{pmatrix}$$

Bestimmung der Eigenwerte:

$$\det\left(\hat{\boldsymbol{\Sigma}} - \lambda \boldsymbol{I}\right) = \det\begin{pmatrix} 0.6166 - \lambda & 0.6154 \\ 0.6154 & 0.7166 - \lambda \end{pmatrix}$$
$$= \lambda^2 - 1.3332\lambda + 0.0632 = 0$$

$$\lambda_1 = 1.284 \quad \text{und} \quad \lambda_2 = 0.0492$$

Bestimmung der Eigenvektoren:

$$\lambda_1: \quad \begin{pmatrix} 0.6166 - 1.284 & 0.6154 \\ 0.6154 & 0.7166 - 1.284 \end{pmatrix} \boldsymbol{u}_1 = \vec{0} \quad \Rightarrow \quad \boldsymbol{u}_1 = \begin{pmatrix} 0.6779 \\ 0.7352 \end{pmatrix}$$

$$\lambda_2: \quad \begin{pmatrix} 0.6166 - 0.0492 & 0.6154 \\ 0.6154 & 0.7166 - 0.0492 \end{pmatrix} \boldsymbol{u}_2 = \vec{0} \quad \Rightarrow \quad \boldsymbol{u}_2 = \begin{pmatrix} -0.7352 \\ 0.6779 \end{pmatrix}$$

Transformationsmatrix:

$$\boldsymbol{U} = \begin{pmatrix} 0.6779 & -0.7352 \\ 0.7352 & 0.6779 \end{pmatrix}$$

Bei dieser Transformationsmatrix ist es sehr leicht zu erkennen, dass es ich um eine orthogonale Matrix handelt, d.h. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ bzw. $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$. Nun können die trasnformierten Merkmalsvektoren berechnet werden.

$$\boldsymbol{y}_i = \boldsymbol{U}^T \boldsymbol{x}_i \qquad \text{z.B.} \quad \boldsymbol{y}_1 = \begin{pmatrix} 0.6779 & -0.7352 \\ 0.7352 & 0.6779 \end{pmatrix} \begin{pmatrix} 2.5 \\ 2.4 \end{pmatrix} = \begin{pmatrix} 3.4592 \\ -0.211 \end{pmatrix}$$

$y_{1,k}$	3.4592	0.8536	3.6235	2.9055	4.3071	3.5442	2.5321	1.4866	2.1932	1.4074
$y_{2,k}$	−0.211	0.1069	0.3485	0.0945	0.1394	−0.2454	−0.3858	0.0105	−0.0182	−0.1986

5.2 Perceptron

Das Perceptron ist ein Trainingsalgorithmus, um für binäre Klassifikationsprobleme (zwei Klassen) eine lineare Entscheidungsgrenze zu finden.
Der Absolutbetrag der Diskriminantenfunktion $|g|$ ist ein Maß für das Vertrauen in die geschätzte Klassenzugehörigkeit eines Merkmalsvektors $\boldsymbol{x}$.

$$g(\boldsymbol{x}) = \sum_{i=1}^{d} w_i x_i - \theta = \boldsymbol{w}^T \boldsymbol{x} - \theta \tag{64}$$

$\boldsymbol{x} \in \mathcal{R}^d$: Merkmalsvektor $\boldsymbol{w} \in \mathcal{R}^d$: Gewichtsvektor $\theta \in \mathcal{R}$: Bias/Schwellwert (Threshold)
Die lineare Diskriminantenfunktion kann auch als Signumfunktion angeschrieben werden.

$$o(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T \boldsymbol{x} - \theta) = \begin{cases} 1 & \text{falls } \boldsymbol{w}^T \boldsymbol{x} \geq \theta \\ -1 & \text{falls } \boldsymbol{w}^T \boldsymbol{x} < \theta \end{cases} \tag{65}$$

5.2.1 Homogene Koordinaten

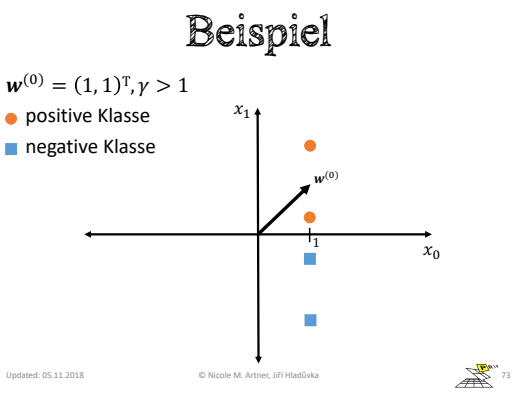
Bei homogene Koordinaten wird das Bias direkt in den Gewichtvektor gegeben.

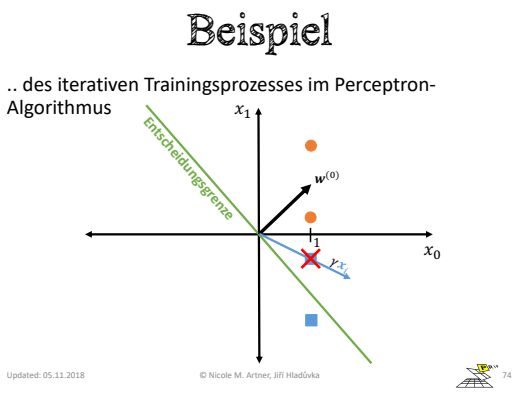
$$\check{\boldsymbol{x}} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \qquad \check{\boldsymbol{w}} = \begin{bmatrix} -\theta \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \qquad \Rightarrow \qquad g(\boldsymbol{x}) = \check{\boldsymbol{w}}^T \check{\boldsymbol{x}} \tag{66}$$

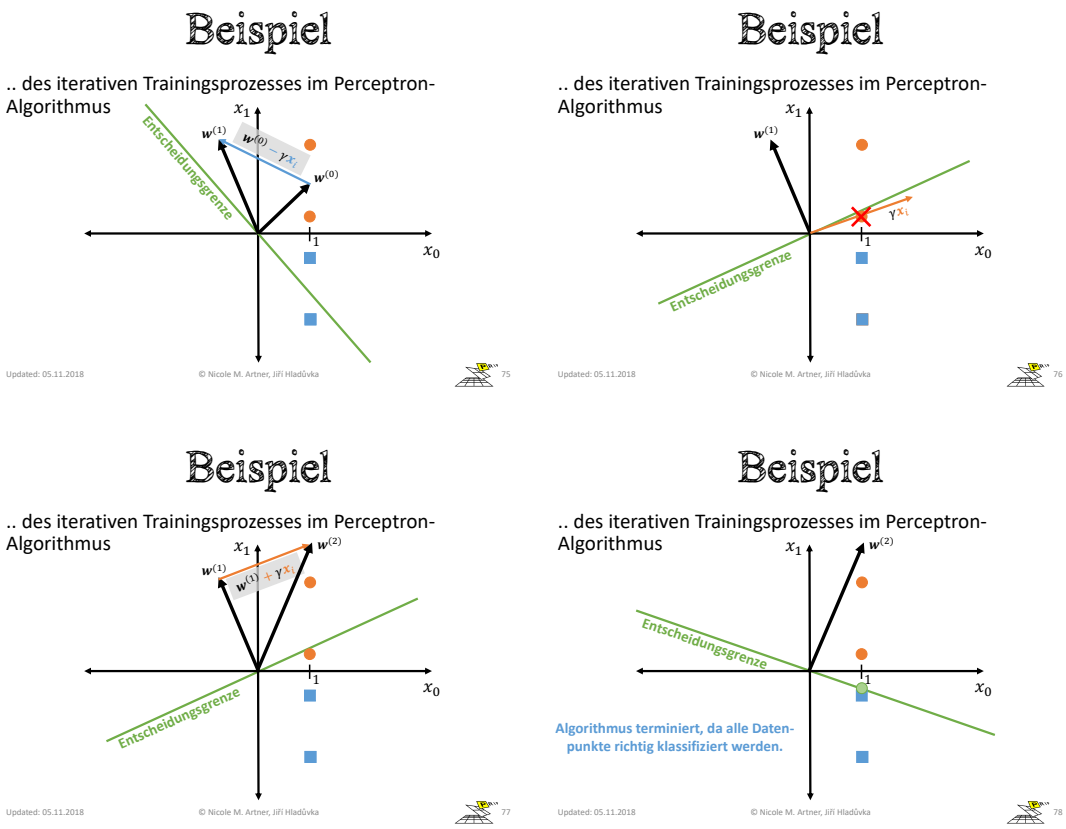
5.3 Perceptron-Algorithmus

Ziel Finde einen Gewichtsvektor $\boldsymbol{w}$, sodass $o(\boldsymbol{x}_i) = \operatorname{sgn}(\boldsymbol{w}^T \boldsymbol{x}_i) = y_i$ ist.

Idee Falls ein Trainingsvektor $\boldsymbol{x}_i$ falsch klassifiziert wurde, dann wird ein Vielfaches γ von $\boldsymbol{x}_i$ zu $\boldsymbol{w}$ addiert. Dadurch wird die Hyperebene auf den falsch klassifizierenden Vektor hinbewegt. Dieses Vielfache γ wird Lernrate gennant und muss positiv sein.
Der Perceptron-Algorithmus ist ein iterativer Prozess der die Parameter der linearen Diskriminantenfunktion, $\boldsymbol{w}$ und θ, aus einem Trainingsdatensatz bestimmt. Häufig wird der Gewichtsvektor als $\boldsymbol{w} = \vec{0}$ initialisiert.







Die Überprüfung welcher Klasse eine Datenpunkt zugehört mittels der Signumfunktion kann vereinfacht werden, indem direkt geprüft wird ob der Wert größergleich oder kleiner Null ist.

$$\operatorname{sgn}\left(\boldsymbol{w}^T \boldsymbol{x}_i\right) = y_i \tag{67}$$

Da wir annehmen, dass y_i nur die Werte 1 und -1 annehmen können, folgt aus vorheriger Formel

$$\operatorname{sgn}\left(\boldsymbol{w}^T \boldsymbol{x}_i\right) y_i = 1 \qquad \Rightarrow \qquad \left(\boldsymbol{w}^T \boldsymbol{x}_i\right) y_i > 0 \qquad \text{bzw.} \qquad \boldsymbol{w}^T \left(\boldsymbol{x}_i y_i\right) > 0 \tag{68}$$

Data: initialisiere $\boldsymbol{w}$ und γ
repeat
  for $i = 1$ **to** N **do**
    if $\boldsymbol{w}^T \left(\boldsymbol{x}_i y_i\right) \leq 0$ **then** `// falsche Klassifikation des Vektors` $\boldsymbol{x}_i$
      $\boldsymbol{w} \leftarrow \boldsymbol{w} + \gamma \left(\boldsymbol{x}_i y_i\right)$ `// Update Gewichtsvektor`
    end
  end
until *alle Merkmalsvektoren* $\boldsymbol{x}_i$ *richtig klassifiziert werden*;

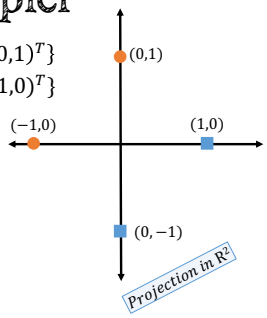
Der Gewichtvektor am Ende des Trainings $\boldsymbol{w}^{(p)}$ ist gegeben durch

$$\boldsymbol{w}_p = \sum_{i=1}^{N} \boldsymbol{x}_i \left(y_i \gamma k_i\right) = \gamma \sum_{i=1}^{N} \boldsymbol{x}_i \left(y_i k_i\right), \tag{69}$$

dieser lasst sich auch Auffassen als lineare Kombination aller falsch klassifizierten Merkmalsvektoren $\boldsymbol{x}_i$. k_i gibt dabei an wie oft der Merkmalsvektor $\boldsymbol{x}_i$ falsch klassifieziert wurde.

Beispiel

- Klasse $w_1 = \{(1,-1,0)^T, (1,0,1)^T\}$
- Klasse $w_2 = \{(1,0,-1)^T, (1,1,0)^T\}$



- $N = 4$
- $\boldsymbol{w} = \vec{\boldsymbol{0}} = (0,0,0)^T$
- $\gamma = 1$

Beispiel

- überprüfe für jeden Vektor $\boldsymbol{x}_i$ ob $\boldsymbol{w}^T(\boldsymbol{x}_i y_i) \leq 0$
- führe Update durch falls obige Bedingung zutrifft: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \gamma \boldsymbol{x}_i y_i$

Beginn Epoche 1:

$$\boldsymbol{w}^{(0)\,T}(\boldsymbol{x}_1 y_1) = [0 \quad 0 \quad 0]\begin{bmatrix}1\\-1\\0\end{bmatrix} \cdot 1 = 0$$

Update: $\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} + \gamma \boldsymbol{x}_1 y_1 = \begin{bmatrix}0\\0\\0\end{bmatrix} + \begin{bmatrix}1\\-1\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\0\end{bmatrix}$

$$\boldsymbol{w}^{(1)\,T}(\boldsymbol{x}_2 y_2) = [1 \quad -1 \quad 0]\begin{bmatrix}1\\0\\1\end{bmatrix} \cdot 1 = 1 > 0$$

Beispiel

$$\boldsymbol{w}^{(1)\,T}(\boldsymbol{x}_3 y_3) = [1 \quad -1 \quad 0]\begin{bmatrix}1\\0\\-1\end{bmatrix} \cdot -1 = -1$$

Update: $\boldsymbol{w}^{(2)} = \boldsymbol{w}^{(1)} + \gamma \boldsymbol{x}_3 y_3 = \begin{bmatrix}1\\-1\\0\end{bmatrix} + \begin{bmatrix}1\\0\\-1\end{bmatrix} \cdot -1 = \begin{bmatrix}0\\-1\\1\end{bmatrix}$

$$\boldsymbol{w}^{(2)\,T}(\boldsymbol{x}_4 y_4) = [0 \quad -1 \quad 1]\begin{bmatrix}1\\1\\0\end{bmatrix} \cdot -1 = 1 > 0$$

Ende Epoche 1

Beispiel

Beginn Epoche 2:

$$\boldsymbol{w}^{(2)\,T}(\boldsymbol{x}_1 y_1) = [0 \quad -1 \quad 1]\begin{bmatrix}1\\-1\\0\end{bmatrix} \cdot 1 = 1 > 0$$

$$\boldsymbol{w}^{(2)\,T}(\boldsymbol{x}_2 y_2) = [0 \quad -1 \quad 1]\begin{bmatrix}1\\0\\1\end{bmatrix} \cdot 1 = 1 > 0$$

$$\boldsymbol{w}^{(2)\,T}(\boldsymbol{x}_3 y_3) = [0 \quad -1 \quad 1]\begin{bmatrix}1\\0\\-1\end{bmatrix} \cdot -1 = 1 > 0$$

$$\boldsymbol{w}^{(2)\,T}(\boldsymbol{x}_4 y_4) = [0 \quad -1 \quad 1]\begin{bmatrix}1\\1\\0\end{bmatrix} \cdot -1 = 1 > 0$$

Ende Epoche 2: Algorithmus hat konvergiert

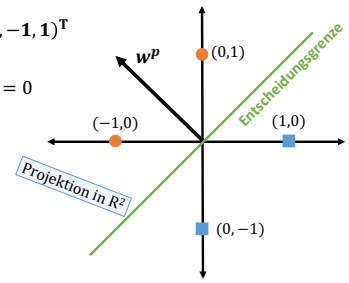
Beispiel

Ergebnis:

$\boldsymbol{w}^p = (\boldsymbol{0}, -\boldsymbol{1}, \boldsymbol{1})^{\mathbf{T}}$

$\updownarrow$

$\theta = 0$



5.4 Perceptron als Neuron

Ein einstufiges Perceptron kann nur linear separierbare Mengen, d.h. Mengen die durch eine Hyperebene trennbar sind, fehlerfrei klassifizieren.
Das `XOR`-Problem ist nicht linear separierbare, im gegensatz zum `UND`-Problem.

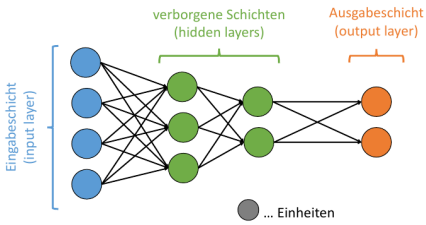
5.5 Generalisierungsfähigkeit des Perceptrons

Es ist anzunehmen, da es unendliche viele Möglichkeiten gibt eine Entscheidungsgrenze einzuziehen, dass die beste Option die ist, welche den Abstand der Trainingsdatenpunkte zu der Hyperebene maximiert. Diese ist eine beleibe Grundidee von SVM-Klassifikatoren (Support Vector Machine).

6 Neuronale Netze

6.1 Grundlagen neuronaler Netze

Was ist ein neuronales Netz?



- Sie bestehen aus künstlichen Neuronen
- haben mindestens eine verborgene Schicht
- sind nicht intelligent
- können nicht wirklich denken
- können begrenzte Mustererkennungsprobleme lösen
- sind nicht zwingend besser als andere Mustererkennungsalgorithmen
- menschliches Gehirn besteht aus 10.000.000.000 Neuronen

`XOR`-Problem Eine Verknüpfung von Perceptronen, also ein neuronales Netz, macht es möglich mehrere Entscheidungsgrenzen zu ziehen. Damit kann das `XOR`-Problem gelöst werden.

Anzahl der Einheiten Einheiten werden oft auch als Neuronen bezeichnet.

- Die **Anzahl der Einheiten** ist gleich der Anzahl der Elemente des Merkmalsvektors
- Die Anzahl der verborgenen Einheiten kann nicht so einfach festgelegt werden. Sie beruht sehr stark auf Erfahrungswerte oder Experimente.
- Die Anzahl der Ausgabeeinheiten ist gleich der Anzahl der Werte im Ausgabevektor $\mathbf{o}$, was in den meisten Fällen gleich der Anzahl der Klassen ist. In den meisten Fällen auch nur deswegen, da bei einer einfache binären Klassifizierung nur eine Ausgabeeinheit benötig wird.

6.1.1 Aktivierungsfunktion

Anstatt der Sigmnumfunktion können auch andere Aktivierungsfunktion verwendet werden.

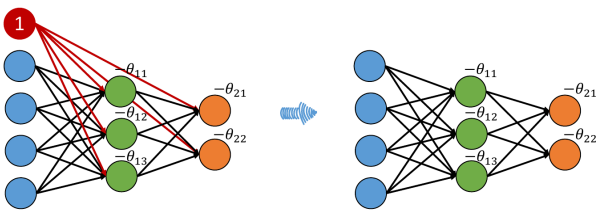
Sigmoid-Funktion

$$g(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{70}$$

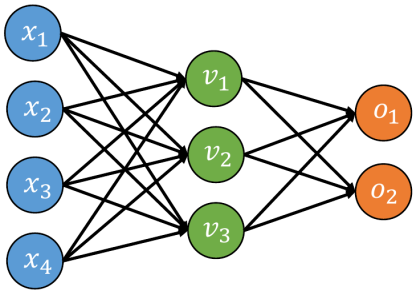
Tangens Huperbelfunktion

$$g(x) = \tanh(x) \tag{71}$$

Bias Die zusätzliche Einheit und ihre Verbindungen werden oft nicht dargestellt, um die Darstellung des neuronalen Netzes zu vereinfachen.



Ausgabe des Netzes – Forward Pass



1. Berechnung des Aktivierungswerts der verborgenen Schicht

$$v_j = g\left(\sum_{i=0}^{N_i} w_{ji}x_i\right) = g\left(\sum_{i=1}^{N_i} w_{ji}x_i - \theta_{1j}\right) \tag{72}$$

2. Berechnung der Aktivierungswerte an der Ausgabeschicht

$$o_k = g\left(\sum_{j=0}^{N_j} w_{kj}v_j\right) = g\left(\sum_{j=0}^{N_j} w_{kj}v_j - \theta_{2k}\right) \tag{73}$$

Oder direkte Berechnung

$$o_k = g\left(\sum_{j=0}^{N_j} w_{kj}g\left(\sum_{i=0}^{N_i} w_{ji}x_i\right)\right) \tag{74}$$

Terminologie

1. **Multilayer-Perceptron (MLP):** Neuronale Netze werden oft als MLP bezeichnet. Dieser Name ist aber irreführend, da die Neuronen in einem neuronalen Netz andere Aktivierungsfunktionen verwenden als das Perceptron.
2. **Backpropagation-Netz** nennt man neuronale Netze die mit dem Backpropagation-Algorithmus trainiert wurden. Dieser Algorithmus ist ein überwachtes Lernverfahren das aus drei Schritten besteht (mehr dazu im nächsten Abschnitt).

6.2 Training neuronaler Netze

Grundidee

- für Parameter (Gewichte, Bias) werden Zufallswerte angenommen
- Backpropagation-Algorithmus ändert die Parameter mit Hilfe eines Trainingsdatensatzes (Supervised Learning - überwachtes Lernen)
- Trainingsdatensatz besteht aus:
  - N Eingabevektoren $\boldsymbol{x}_n$ und den zugehörigen
  - N Ausgabevektoren $\boldsymbol{t}_n$ (target vectors)
  - idealerweise ein möglichst großer Datensatz

Folgende Schritte werden wiederholt durchlaufen (iterativ) bis das Netz ausreichend trainiert ist, d.h. der Fehler klein genug ist:

1. Forward Pass
2. Bestimmung des Fehlers
3. Backward Pass

Das Training läuft üblicherweise in mehreren Epochen (Zyklen) ab, wobei eine Epoche bedeutet, dass alle Eingabevektoren $\boldsymbol{x}_n$ abgearbeitet wurden.

Forward Pass Für einen beliebigen Eingabevektor $\boldsymbol{x}_n$ aus dem Trainingsdatensatz wird der Ausgabevektor $\boldsymbol{o}_n$ ermittelt. Seine Elemente o_k ergeben sich wie folgt:

$$o_k = g\left(\sum_{j=0}^{N_j} w_{kj} g\left(\sum_{i=0}^{N_i} w_{ji} x_i\right)\right) \tag{75}$$

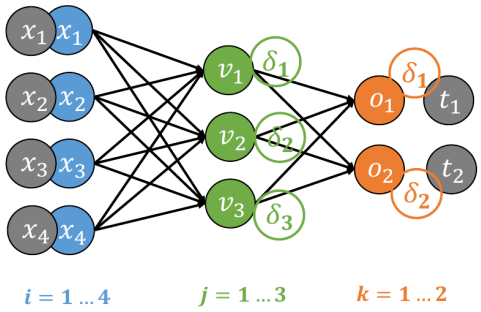
Bestimmung des Fehlers Der vom neuronalen Netz gelieferte Ausgabevektor $\boldsymbol{o}_n$ wird mit der Zielausgabe $\boldsymbol{t}_n$ verglichen:

$$J(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{t}_n - \boldsymbol{o}_n||^2 = \frac{1}{2}\sum_{k=1}^{N_k}(t_k - o_k)^2 \tag{76}$$

Das Ziel des Trainings ist es $J(\boldsymbol{w})$ (Sum of Squares Error) zu minimieren. $J(\boldsymbol{w})$ wird auch als Kostenfunktion bezeichnet.

Backward Pass Der Backward Pass erfolgt in umgekehrter Reihenfolge wie der Forward Pass. Beginnend mit der Ausgabeschicht werden die Gewichte/Biaswerte abhängig von ihrem Einfluß auf den Fehler verändert. Diese Anpassung kann z.B. mit Gradient Descent durchgeführt werden.

Korrekturen basierend auf Fehler



Aktivierungsfunktion ist gleich der Identität.
Output-Layer:

$$\delta_k = t_k - o_k \qquad \Delta w_{kj} = \delta_k \boldsymbol{v}_j \tag{77}$$

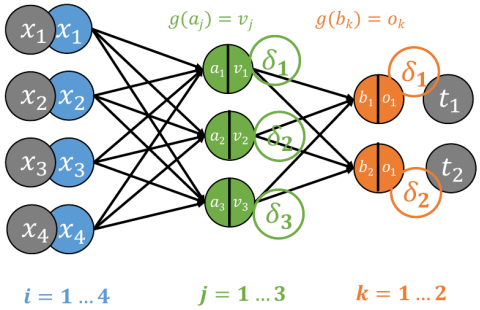
Hidden-Layer:

$$\delta_j = \sum_k \delta_k w_{kj} \qquad \Delta w_{ji} = \delta_j \boldsymbol{x}_i \tag{78}$$

(Vergleiche Indizes mit Bild:)

$$\nabla \boldsymbol{J} = \begin{bmatrix} \cdots & \Delta w_{ij} & \cdots & \cdots & \Delta w_{kj} & \cdots \end{bmatrix}^T \tag{79}$$

Korrekturen basierend auf Fehler mit Aktivierungsfunktion



Aktivierungsfunktion g kann beliebige differenzierbare Funktion sein, wie z.B. tanh, Sigmoid, aber auch die Identität, welche das Ergebnis von vorhin liefert.
Output-Layer:

$$\delta_k = g'(b_k)\,(t_k - o_k) \qquad \Delta w_{kj} = \delta_k \boldsymbol{v}_j \tag{80}$$

Hidden-Layer:

$$\delta_j = g'(\boldsymbol{a}_j) \sum_k \delta_k w_{kj} \qquad \Delta w_{ji} = \delta_j \boldsymbol{x}_i \tag{81}$$

(Vergleiche Indizes mit Bild:)

$$\nabla \boldsymbol{J} = \begin{bmatrix} \cdots & \Delta w_{ij} & \cdots & \cdots & \Delta w_{kj} & \cdots \end{bmatrix}^T \tag{82}$$

6.3 Wichtige Hinweise und Tipps

Reihenfole der Trainingsdaten Da die Gefahr besteht, dass das Backpropagation-Netz die "Reihenfolge" der Trainingsdaten lernt, sollte die Reihenfolge in jedem Zyklus verändert werden.

Skalierung der Einheiten

- Merkmale liegen oft in unterschiedlichen Einheiten vor
- damit alle Merkmale gleichen Einfluss haben sollte eine Standardisierung (Mittelwert = 0, Varianz = 1) durchgeführt werden
- zusätzlicher Vorteil: Gradient-Descent-Verfahren konvergieren schneller auf skalierten Merkmalen

Verborgene Einheiten Zu wenig verborgenen Einheiten kann dazu führen, dass Trainingsdaten nciht ausreichend gelernt werde (= underfitting). Wo hingegen zu viele zu einer schlechten Generalisierung führen können (= overfitting).

Lernrate Die Lernrate sollte auf jedenfall nicht zu groß angesetzt werden, aber auch nicht zu klein, da es sehr lange dauern kann bis das Gradient-Descent-Verfahren konvergiert. Meist ist ein $\gamma = 0.1$ geeignet.

6.3.1 Vorteile Neuronaler Netze

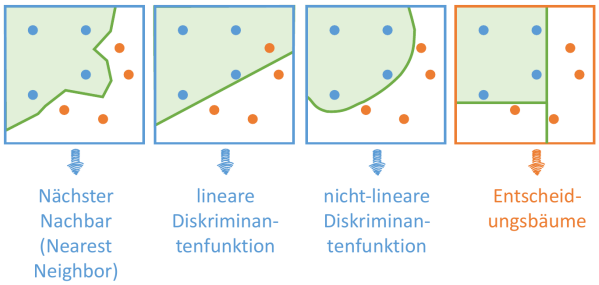
- einfacher Trainingsalgorithmus
- ermöglicht Implementierung komplexer Entscheidungsgrenzen
- schnelle Hardwareimplementierung möglich, druch Parallelisierung

6.3.2 Nachteile Neuronaler Netze

- Training kann langwierig sein (falsch gewählte Lernrate, etc.)
- Training benötigt viele Parameter die teilweise schwer wählbar sind
  - Anzahl der Schichten
  - Anzahl der verborgenen Einheiten
  - Lernrate
- Neurale Netze mit vielen verborgenen Einheiten neigen zum "Overfitting"
  - um dem entgegen zu wirken, sollte der Trainingsdatensatz groß sein
- "Black-Box"
  - es ist schwierig nachzuvollziehen wie ein Netz ein Problem löst

7 Entscheidungsbüme

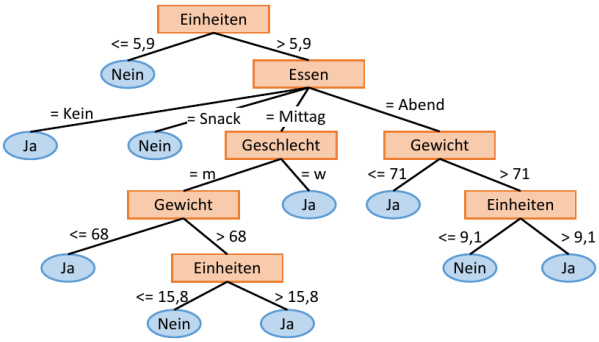
7.1 Was sind Entscheidungsbäume?



Nominelle Merkmale

- Beschreibung eines Musters durch Eigenschaften und ihre Ausprägungen
- Katze
  - Eigenschaften: Fell, Augen, Konstitution
  - Ausprägungen (Merkmalsvektor): orange, grün, fett
- Obst
  - Eigenschaften: Geschmack, Farbe, Form, Größe
  - Ausprägungen (Merkmalsvektor): sauer, rot, rund, klein

Entscheidungsbäume sind aufgebaut nach intuitive Klassifiktion durch eine Abfolge von Fragen. Dabei ist jede nächste Frage abhängig von der vorhergeneden. Die Antwort auf eine Frage muss jedoch eine konkrete Ausprägungen haben, wie z.B. Farbe (rot, blau,dots) oder ja/nein.
Ein Beispiel eines entscheidungsbaumes ist der Besoffenheitstest nach Padraig Cunningham



Klassifiktion

- Klassifikation eines Musters beginnt immer bei **Frage des Wurzelknotens**
- abhängig von Antwort (Ausprägung) folgt man entsprechender **Verbindung zu einem Folgeknoten**
- es muss eine Entscheidung getroffen werden, d.h. es darf nur einer Verbind- ung gefolgt werden
- dann wird Entscheidung für “Frage” des Folgeknoten gefällt, usw. . .
- bis man bei einem Blattknoten ankommt , also es keine weiteren Fragen mehr gibt
- Blattknoten ist einer Klasse zugeordnet, diese wird einem Muster zugewiesen

Vorteile

- einfach zu **interpretieren**
- logische Strukturen
- Interpretation der Klassifiktion durch die Klassen selbst, z.B. Apfel=(grün UND mittel) oder (rot UND mittel)
- in der Regel schnelle Klassifikation
- einfache Integration von Expertenwissen

7.2 Erstellen von Entscheidungsbäumen

Training Auf einem Trainingsdatensatz $\mathcal{D}$, welches gelabelt ist, wird eine ein Entscheidungsbaum mittels einem geeigeneten Satz an Fragen und Antworten trainiert. Dabei handelt es sich um supervised learning.

Ziel des Trainings Das Ziel ist es **automatisch** einen Baum zu erzeugen. Dabei werden die Trainingsdaten in immer kleineren Untermengen aufgeteilt. Im Idealfall sollte jede dieser Untermengen rein sein, was so viel bedeutet, dass die beinhalteten Trainingsdaten nur von einer Klasse stammen.

Herausforderung Ein Problem ist jedoch, dass normalerweise Untermengen nicht rein sind, sondern aus Mischungen von verschiedenen Klassen bestehen. Wird versucht reine Untermengen zu erzeugen, so wird der Baum meist zu groß oder er wird zu stark an die Trainingsdaten angepasst (= overfitting). Daher gibt man sich meist mit einem kleinen Fehler zufrieden.
Eine Methode die Hilft Bäume zu generieren ist CART (= Classification And regression Trees).

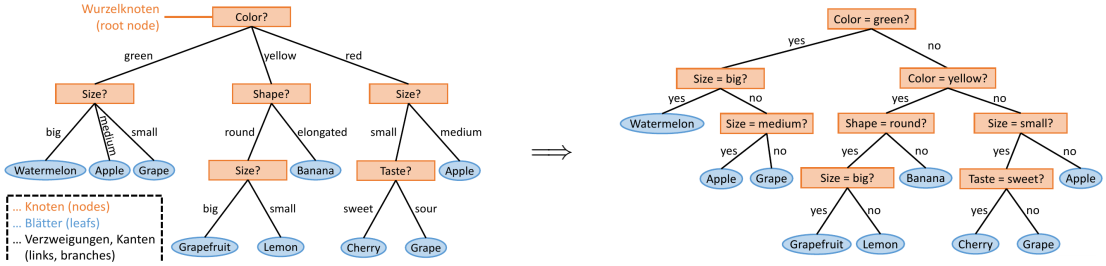
7.3 CART im Detail

CART (= Classification And Regression Trees) ist eine iterative Methode um Entscheidungsbäume zu erstellen/wachsen zu lassen. Abhängig von den Daten in einem Knoten wird: (1) der Knoten als Blatt deklariert und eine Klasse gewählt oder (2) ein Merkmal (Eigenschaft) gefunden, um die Daten in kleinere Untermengen zu teilen.
Dabei werden 6 grundlegende Fragen aufgeworfen

- Wie viele **Verzweigungen** sollen von einem Knoten ausgehen?
  Wie viele Entscheidungsausgänge/Teilungen gibt es?
- Welches **Kriterium** soll an einem Knoten getestet werden?
- Wann soll ein Knoten **in ein Blatt umgewandelt** werden?
- Wie kann man einen **Baum verkleinern** oder vereinfachen, wenn er zu "groß" ist?
- Welcher **Klasse** soll ein Blatt zugeordnet werden, wenn es nicht rein ist?
- Wie geht man mit **fehlenden Daten** um?

7.3.1 Verzweigung

Jeder Entscheidungsausgang an einem Knoten wird als Verzweigung (split) bezeichnet, weil die Trainingsdaten aufgespalten/aufgeteilt werden.
Die Anzahl von Verzweigungen (Kanten) ausgehend von einem Knoten wird häufig als Verteilungsfaktor (branching factor) oder Verteilungsverhältnis (branching ratio) bezeichnet.
Jeder Baum kann als binärer Entscheidungsbaum dargestellt werden, d.h. Verteilungsfaktor $B = 2$. Solch ein binärer Baum führt zum gleichen Klassifikationsergebnis wie der ursprüngliche Baum mit beliebigem Verteilungsfaktor B an den Knoten. Binäre Entscheidungsbäume sind jedoch einfacher zu trainieren. Aus diesem Grund verwendet CART auch solche.



7.3.2 Kriterium

Ein Kriterium besteht aus einer oder mehreren Eigenschaften (Merkmalen) und einer Regel (Heuristik) wie die Daten dadurch aufgeteilt werden. Ziel ist es dabei einen so flachen/einfachen Baum, also wenige Knoten, wie mögle zu erzeugen. Die Kriterien kann man in zwei verschiedene Arten unterteilen.

1. nominell: Frabe? (rot, blau, ...) oder Farbe = rot? (ja, nein)
2. numersisch: $x \leq 0.5$ (ja, nein)

Entropie Die Entropie ist ein Maß mit der die Unreinheit bestimmt werden kann.

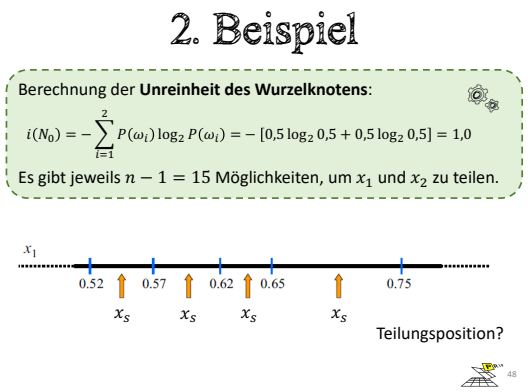
$$i(N) = -\sum_j P(\omega_j) \log_2 (P(\omega_j)) \tag{83}$$

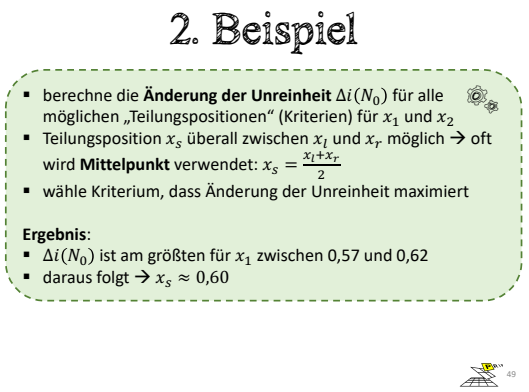
$P(\omega_j)$ gibt dabei die Häufigkeit der Klasse ω_j im Knoten N an.

Kriteriumsauswahl Das Ziel bei der Kriteriumsauswahl ist es, die Unreinheit $i(N)$ eines Knotens N zu minimieren bzw. so klein wie möglich zu halten. D.h. die Änderung der Unreinheit $\Delta i(N)$ zu maximieren. Für eine binäre Verzweigung ist

$$\Delta i(N) = i(N) - P_L i(N_L) - (1 - P_L) i(N_R),$$

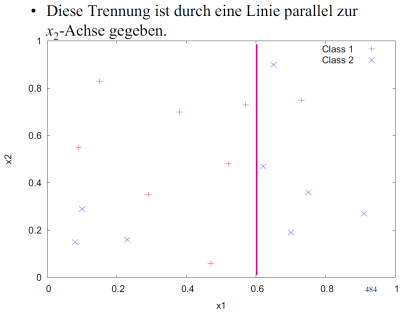
wobei N_L und N_R die resulteirenden linken und rechten Teilmengen sind. P_L ist der Anteil der Muster der dem linken Knoten zugeordnet wird.





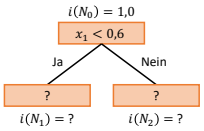
2. Beispiel

- Diese Trennung ist durch eine Linie parallel zur x_2-Achse gegeben.



2. Beispiel

... Ergebnis nach 1. Verzweigung

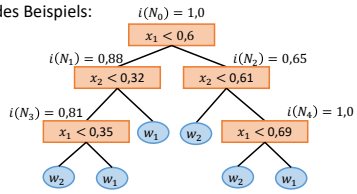


w_1		w_2	
x_1	x_2	x_1	x_2
0,15	0,83	0,10	0,29
0,09	0,55	0,08	0,15
0,29	0,35	0,23	0,16
0,38	0,70	0,70	0,19
0,52	0,48	0,62	0,47
0,57	0,73	0,91	0,27
0,73	0,75	0,65	0,90
0,47	0,06	0,75	0,36

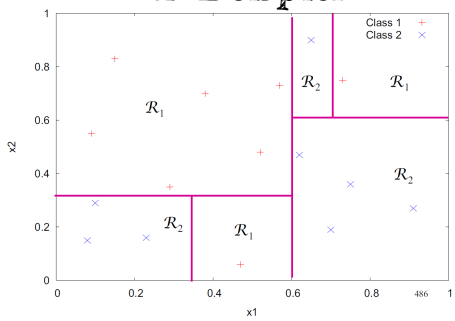
2. Beispiel

- die Knoten N_1 und N_2 sind nicht rein
- bestimme neue Kriterien für weitere Verzweigungen
- bis die Unreinheit aller Blätter 0 wird

Ergebnis des Beispiels:



2. Beispiel



7.3.3 Stoppen des Trainings

Wann das Training gestoppt werden sollte, kann durch verschiedene Abbruchkriterien angegeben werden.

- **Unabhängige Validierungsdaten** (Testdaten)
  - Training wird abgebrochen, wenn Fehler auf Validierungsdaten klein genug ist (Schwellwert)
- **Änderung der Unreinheit**
  - Knoten wird zu einem Blatt, wenn $\Delta i(N) < \beta$ ist
  - β schwierig festzulegen
- Anzahl der Muster
  - Knoten wird zu einem Blatt, wenn Anzahl der zugehörigen Muster unter einen Schwellwert fällt
  - z.B.: 5% der Trainingsdaten

7.3.4 Pruning

Poblem Training wird auf Grund von lokalen Abbruchkriterien gestoppt. Es fehlt der sogenannte Weitblick. Dieses Phänomen wird als “Horizon Effect” bezeichnet.

Lösung Eine Lösung dafuer ist das Pruning. Dabei wird das Training gestoppt, wenn Unreinheit aller Knoten gleich null ist. Dabei wird für paare von Knoten, welche einen gemeinsamen Elternknoten besitzen, geprüft welchen Einfluss diese auf die Unreinheit haben. Ist dieser klein, also wird die Unreinheit nur minimal erhöht, so werden die Knoten entfernt und der Elternknoten wird zu einem Blatt

Info:

- Pruning beginnt meist mit den Blatt-Knoten
- effiziente Algorithmen können einen komplexen Teilbaum (subtree) direkt mit einem Blatt ersetzen
- Pruning verhindert den „Horizon Effect“
- komplexes Problem:
  - Abbruchkriterien sind vorzuziehen
  - es dauert zu lange bis alle Blätter eine Unreinheit von Null aufweisen
- einfaches Problem:
  - Pruning wird bevorzugt
  - Rechenaufwand vernachlässigbar

7.3.5 Klassenzugehörigkeit

Problem Für welche Klasse soll man sich entscheiden, wenn ein Blatt nicht rein ist?

Lösung Das Blatt gehört zu jener Klasse die am häufigsten im Knoten vorkommt. D.h. die meisten Muster des Trainingsdatensatzes in dem Blatt waren dieser Klasse zugeordnet.

7.3.6 Fehlende Daten

Training

- naive Lösung: löschen der unvollständigen Muster (nur wenn ausreichend Trainingsdaten zur Verfügung stehen)
- “schlauere” Lösung:
  - Berechnung von $i(N)$ und $\Delta i(N)$ auf den vorhandenen Informationen
  - z.B.: In Knoten N gibt es drei Muster mit zwei Merkmalen, wobei in einem Muster das zweite Merkmal fehlt. Um die beste Teilung zu finden, ermittelt man alle drei möglichen Teilungen basierend auf dem ersten Merkmal und nur zwei Teilungen für das zweite Merkmal.

7.4 Möglichkeiten und Einschränkungen

Es können beliebige Entscheidungsgrenzen apporximiert werden, unter der Bedingung, dass der Entscheidungsbaum groß genug ist.
Verfahren wie CART funktionieren grundsätzlich am besten, wenn die Merkmale und Entscheidungskriterien sinvoll gewählt sind.
Entscheidungsgrenzen sind meist dann suboptimal gewählt, wenn sie paralell zu einder der Achsen sind, daher kann bei der Bestimmung direkt vorausgesetzt werden, dass dies nicht der Fall sein darf. (z.B. $-1.2x_1 + x_2 < 0.1$?)

8 Markov Modelle

8.1 Was sind Markov Modelle?

Markov erkennung Modelle für die (Markov Klassifikation Models) von werden Mustern in eingesetzt, der Muster die sich aus einer Sequenz von “Sub-Mustern” zusammensetzen. Wie z.B. das Wetter einer Sommerwoche (sonne-sonne-sonne-bewölkt-sonne-sonne-sonne) oder einer typischen Aprilwoche (regen-regen-beölkt-regen-regen-bewölkt-regen-regen)

State: Sequenz von Symbolen
Transition: Übergang zwischen zwei States, welche mit einer gewissen Wahrscheinlichkeit verknüpft sind.

8.1.1 Sequentielle Daten

Markov Modelle werden zur Modellierung und Klassifiation von sequentiellen Daten eingesetzt. Solche Daten ergeben sich häufig aus zeitlicher Abfolge, wie z.B. Wetter Börsenkurse, ... Nicht zeitliche Abfolgen sind dagegen DNA Sequenzen oder Buchstaben eines Wortes.
Die Idee dahinter ist, dass zukünftige Prognosen aufgrund von bisherigen Daten geschätzt wird. Von wie vielen States so eine Progrnose abhängig ist, ist frei wählbar, jedoch wird meist dabei ein Markov Modell angenommen. Dies bedeutet, dass die Übergangswahrscheinlichkeit immer nur von dem jeweilig vorherigen State abhängt.

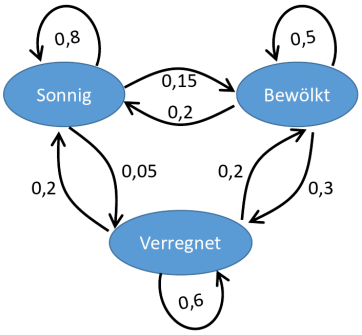
8.2 Grundlagen: "Markov Modell" (MM)

Markov Annahme (1. Ordnung):

$$P\left(w_n|w_{n-1}, w_{n-2}, \ldots, w_1\right) \approx P\left(w_n|w_{n-1}\right)$$

Wahrscheinlichkeit von w_n hängt nur von der Beobachtung w_{n-1} ab. Folglich: Bei einer Markov Annahme 2. Ordnung berücksichtigt man w_{n-1} und w_{n-2}.
Diese Übergangswahrscheinlichkeiten können in einer Matrix oder als Graph dargestellt werden.

		morgiges Wetter		
		Sonnig	Verregnet	Bewölkt
heutiges Wetter	Sonnig (S)	0,8	0,05	0,15
	Verregnet (R)	0,2	0,6	0,2
	Bewölkt (W)	0,2	0,3	0,5



Die Wahrscheinlichkeit, dass eine Sequenz $\{\mathrm{W}, \cdot, \mathrm{R}\}$ auftritt kann durch die Summation der einzelnen Sequenz ({W, W, R}, {W, R, R}, {W, S, R}) berechnet werden.

$$\begin{aligned} P(w_3 = R|w_1 = W) &= P(w_3 = R|w_2 = W)P(w_2 = W|w_1 = W) \\ &+ P(w_3 = R|w_2 = R)P(w_2 = R|w_1 = W) \\ &+ P(w_3 = R|w_2 = S)P(w_2 = S|w_1 = W) \\ &= 0.3 \cdot 0.5 + 0.6 \cdot 0.3 + 0.05 \cdot 0.2 = 0.34 \end{aligned} \tag{84}$$

8.3 Grundlagen: "Hidden Markov Modell" (HMM)

Was sind Hidden Markov Models? Am Beispiel vom Wetter: Angenommen man wird in einen Raum ohne Fenster eingeschlossen und man wird nach dem Wetter draußen befragt. Der einzige Hinweis den man hat, ist die An- oder Abwesenheit eines Regenschirms bei der Person die täglich das Essen bringt.

	Schirm (True)	Schirm (False)
Sonnig (S)	0,1	0,9
Verregnet (R)	0,8	0,2
Bewölkt (W)	0,3	0,7

In HMM ist es nicht möglich das Wetter ("States") zu beobachten, deshalb benötigt man das Bayes-Theorem:

$$P\left(w_1, \ldots, w_n\right) = \frac{P\left(u_1, \ldots, u_n|w_1, \ldots, w_n\right)}{P\left(u_1, \ldots, u_n\right)} \tag{85}$$

u_i: ist "wahr" (T) wenn Besucher Schirm dabei hat, sonst "falsch" (F)
$P(w_1, \ldots, w_n)$: wie beim Markov Modell $\prod_{i=2}^{N} P(w_i|w_{i-1})$
$P(u_1, \ldots, u_n)$: Prior für eine bestimmte Schirm-Sequenz
$P(u_1, \ldots, u_n|w_1, \ldots, w_n)$: schätzt man als $\prod_{i=1}^{N} P(u_i|w_i)$

8.4 Training von HMM

9 Evaluierung

10 Clustering

11 Prüfung

11.1 Prüfungsaufgaben

Rechenbeispiele

- Bayes-Theorem
- Perceptron
- PCA
- Neuronale Netze
- Kennzahlen
- Diskriminantenfunktionen
- etc.

Verständnisbeispiele

- überprüfen Verständnis der Theorie anhand konkreter Beispiele
- Fragen zur Theorie können im Zusammenhang mit den Beispielen gestellt werden

11.1.1 2019-01-28

1. Entscheidungsbäume: Es waren Punkte (der Klassen x und o) in einer graphischen Darstellung mit Entscheidungsgrenzen parallel zur x- und y-Achse gegeben. Man musste daraus einen Entscheidungsbaum zeichnen (es wurde extra erwähnt, die Nummern der einzelnen Punkte in die Blätter einzutragen und nicht nur x und o) und dann zu jeder Kante/Knoten die Entropie berechnen.
2. Bayes: Es waren zwei Klassen P(wj) und je 3 Merkmale P(x=i|wj) gegeben. Man musste ausrechnen: Die P(x=i), die P(wj|x=1), die P(error|x=i), den gesamt P(error).
3. Bayes-Theorem:
   Gegeben: 4 Grafiken
   Gesucht(Beschriftung der Graphiken): Likelihood, Likelihood x Prior, Evidence, Posterior, Mittelwert, Standardabweichung.
   Theorie: in welche Richtung bewegt sich die Entscheidungsgrenze wenn sich die Priors verändern
   Einzuzeichnen in den Graphiken: Die Entscheidungsgrenze, Fehlerwahrscheinlichkeit
4. Perceptron: Daten mussten zuerst homogenisiert werden. Dann Perceptron-Algorithmus anwenden und Ergebnis in Koordinatensystem einzeichnen. gegeben: 4 Punkte ((2;0) (0;2)(-2;0) (0;-2)) mit Labels 1,1,-1,-1
5. Eigenwerte/Eigenvektoren und U der Kovarianzmatrix (2 0; 0 2)
6. Es war ein Graph(x,y) mit Datenpunkten gegeben. 6 Kovarianzmatrizen, davon musste die richtige gewählt werden und danach argumentiert werden wieso diese richtig ist.

11.1.2 2019-06-03

Sehr ähnlich zur Prüfung vom 28.01.2019:

1. Es waren 6 Kovarianzmatrizen gegeben, und ein Bild und man musste begründen, welche davon warum zum Bild passt, und warum die anderen nicht.

2. Bayes berechnen, analog zu Folien (gegeben waren wie in den Folien die likelihoods und priors, berechnen musste man die evidence-Tabelle, posteriors und errors, und Gesamterror)

3. Es waren 4 Grafiken gegeben, die man richtig zu Beschreibungen mappen musste: Welche Grafik zeigt die Likelihoods, welche die gewichteten Likelihoods, welche den Bayes-Discriminator, welche die Evidence? Zusätzlich musste man die Standardabweichung und Mittelwert zu vorgegebenen Werten bei den Grafiken richtig eintragen, dann noch die Entscheidungsgrenze als Linie in der richtigen grafik einzeichnen und den P(error) als Fläche in der richtigen grafik durch schraffieren einzeichnen.

4. Perceptron: Man musste den PCA Algorithmus wie in den Folien durch-iterieren, 3 Epochen waren nötig, es war gegeben: $X = (2,0)^T, (0,2)^T, (-2,0)^T, (0,-2)^T$, Labels $= 1, 1, -1, -1$ und w soll mit 0, γ mit 1 initialisiert werden. Zu Beginn musste man also zudem noch $\boldsymbol{x}$ homogenisieren. Abhängig vom Ergebnis dieser Iterationen musste man in der zweiten Teilaufgabe in einem leeren Raster die Punkte aus $\boldsymbol{x}$, Achsen für die Grafik+Beschriftung, den errechneten Vektor $\boldsymbol{w}$ und die daraus resultierende Entscheidungsgrenze einzeichnen (analog zu Folien).

5. Entscheidungsbaum: Baum war gegeben, mit 2 Klassen x und o, 4 Knoten in denen entweder x<(irgendeinvalue) oder o<(irgendeinvalue) war, eine grafik in der die x und o abhängig von (x,y) eingezeichnet waren. Man musste die Entscheidungsgrenzen aufgrund des gegebenen Baums in die Grafik einzeichnen und danach die einzelnen Entropien zu den Knoten und Blättern berechnen.

6. Neuronales Netzwerk: Es war ein neuronales Netzwerk mit 3 Eingabeknoten, 2 hidden Nodes und 1 Ausgabeknoten gegeben, inklusive einer Tabelle mit w_{ij}, θ_{1j}, w_{1k}, θ_{1k} und o. Analog zu Folien: Berechne Ausgabe der hidden nodes, dann Ausgabe des Ausgabeknotens o.

Wenn man von allen Folien alle Übungsbeispiele 1 bis 2 mal nachrechnet, sollte man bei dieser Prüfung alle Praxisaufgaben leicht lösen können. Pro Foliensatz gibt es meist 1 bis 3 unterschiedliche Übungsaufgaben die auf mehreren Folien Step-by-Step erläutert werden. Die paar möglichen Theoriefragen aus den vorigen Prüfungen sollte man sich auch ansehen.