

① Lineare Modelle - Lineare Diskriminante Analyse (LDA)

Wir betrachten Modelle der Form

$$(x) \quad p(y=c|x, \theta) = \frac{p(x|y=c, \theta) \cdot p(y=c)}{\sum_k p(x|y=k, \theta) \cdot p(y=k)}$$

Label / Target Daten Modellparameter

A-posteri Wahrscheinlichkeit der Klasse c ,
gegeben Datenspunkt x und fixen Modell-
parametern θ

$p(x|y=c, \theta)$ Klassenbedingte Wahrscheinlichkeitsdichte,
gegeben Klasse c und fixen Modellparametern
 $p(y=c)$ A-priori Wahrscheinlichkeit der Klasse c

Zu gegebenem x (Datenspunkt) und fixem θ , erhalten wir als
Entscheidungsregel:

$$k^* = \arg \max_c p(y=c|x, \theta)$$

Beispiel: Wir betrachten den Fall $x \in \mathbb{R}^d$, und $y \in \{0, 1\}$.
 Datenpunkt Label

Annahme 1: $p(x|y=c, \theta)$ lässt sich mittels einer
multivariaten Normalverteilung beschreiben.

$$P(x | y=c, \Theta_c) = (2\pi)^{-d/2} \cdot \sqrt{\det(\Sigma_c)} \cdot \exp\left(-\frac{1}{2} \cdot (x - \mu_c)^T \Sigma_c^{-1} (x - \mu_c)\right)$$

(MVN) \leftarrow Multivariate Normalverteilung mit Kovarianz Matrix
 Σ_c und Lokationsparameter $\mu_c \in \mathbb{R}^d$
 Σ_c $\xrightarrow{d \times d}$ Matrix

$$\text{mit } \Theta_c = \{\Sigma_c, \mu_c\}.$$

Annahme 2: $\underbrace{\Sigma_1 = \Sigma_0 = \Sigma}$

Kovarianz Matrizen der Klasse 1 und 0 sind gleich!

Wir sehen aus (x), dass $\xrightarrow{\text{proportional zu}}$

$$P(y=c | x, \Theta) \propto P(x | y=c, \Theta) \cdot P(y=c)$$

(da Nenner in (x) immer gleich)

Wir setzen $P(y=0 | x, \Theta) = P(y=1 | x, \Theta)$ um zu sehen wie die Entscheidungsgrenze aussieht. Definieren $\bar{\pi}_0 = P(y=0)$ und $\bar{\pi}_1 = P(y=1)$.

$$\bar{\pi}_0 \cdot \frac{1}{\sqrt{(2\pi)^d \cdot \det(\Sigma)}} \cdot \exp\left(-\frac{1}{2} \cdot (x - \mu_0)^T \Sigma^{-1} (x - \mu_0)\right) = \frac{1}{\sqrt{(2\pi)^d \cdot \det(\Sigma)}} \cdot \exp\left(-\frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\right)$$

$$\stackrel{\log}{\Leftrightarrow} \log(\bar{\pi}_0) - \frac{1}{2} \cdot (x - \mu_0)^T \Sigma^{-1} (x - \mu_0) = \log(\bar{\pi}_1) - \frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1)$$

$$\Leftrightarrow \log(\bar{\pi}_0) - \frac{1}{2} \cdot x^T \Sigma^{-1} x + \frac{1}{2} x^T \Sigma^{-1} \mu_0 + \frac{1}{2} \mu_0^T \Sigma^{-1} x - \frac{1}{2} \mu_0^T \Sigma^{-1} \mu_0 =$$

$$\log(\bar{\pi}_1) - \frac{1}{2} \cdot x^T \Sigma^{-1} x + \frac{1}{2} \cdot x^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_1^T \Sigma^{-1} x - \frac{1}{2} \cdot \mu_1^T \Sigma^{-1} \mu_1$$

$$\Leftrightarrow -\frac{1}{2} \cancel{x^T \Sigma^{-1} x} + \mu_0^T \Sigma^{-1} x - \frac{1}{2} \mu_0^T \Sigma^{-1} \mu_0 + \log(\pi_0) = \\ -\frac{1}{2} \cancel{x^T \Sigma^{-1} x} + \mu_1^T \Sigma^{-1} x - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \log(\pi_1)$$

$$\Leftrightarrow (\mu_1^T \Sigma^{-1} x - \mu_0^T \Sigma^{-1} x) - \frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_0^T \Sigma^{-1} \mu_0 + \log\left(\frac{\pi_1}{\pi_0}\right) = 0$$

$$\Leftrightarrow (\Sigma^{-1}(\mu_1 - \mu_0))^T x + \frac{1}{2}(\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 + \mu_1) + \log\left(\frac{\pi_1}{\pi_0}\right) = 0$$

$$\stackrel{!}{\Leftrightarrow} \underbrace{2 \cdot (\Sigma^{-1}(\mu_1 - \mu_0))^T x}_{\text{abhängig von } x!!!} + \underbrace{(\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 + \mu_1)}_{\text{Skalar check!!!}} + 2 \cdot \log\left(\frac{\pi_1}{\pi_0}\right) = 0$$

skalar

\Rightarrow Wir erhalten einen Ausdruck der Form $Q^T x + b = 0$ mit

$$Q = \Sigma^{-1}(\mu_1 - \mu_0) \cdot 2$$

$$b = (\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 + \mu_1) + 2 \cdot \log\left(\frac{\pi_1}{\pi_0}\right)$$

lineare Entscheidungsgrenze!

(daher der Name LDA)

Linear

wichtig: Wir haben $\Sigma_1 = \Sigma_0 = \Sigma$ angenommen.
(Annahme 2).

Betrachten wir kurz

$$P(y=1 | x, \theta) = \frac{P(x | y=1, \theta) \cdot \underbrace{P(y=1)}_{\substack{\text{Normalisierung} \\ \text{const.}}}}{\pi_1}$$

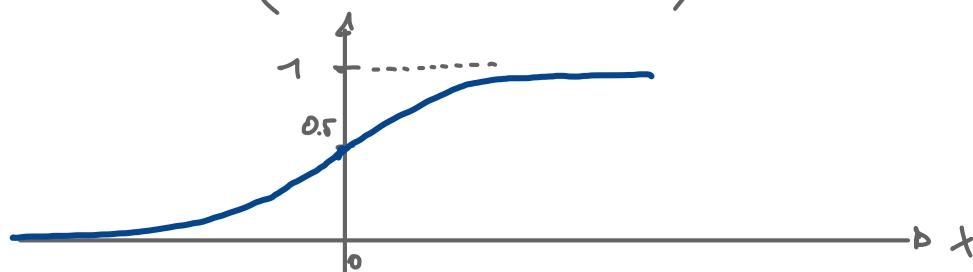
Einsetzen der MVN + Log., ergibt

$$\begin{aligned} \log P(y=1 | x, \theta) &= \log(\pi_1) - \frac{1}{2} \cdot (x - \mu_1)^T \Sigma^{-1} (x - \mu_1) + \text{const.} \\ &= -\frac{1}{2} \cdot \left[x^T \Sigma^{-1} x - 2\mu_1^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} \mu_1 \right] + \log(\pi_1) + \text{const.} \\ &= \underbrace{\log(\pi_1)}_{N_1} - \underbrace{\frac{1}{2} \cdot \mu_1^T \Sigma^{-1} \mu_1}_{x^T \Sigma^{-1} \mu_1} + \underbrace{\mu_1^T \Sigma^{-1} x}_{x^T \beta_1} + \underbrace{\text{const} - \frac{1}{2} x^T \Sigma^{-1} x}_{\approx} \\ &= \mu_1 + x^T \beta_1 + \underbrace{\Sigma}_{\text{unabhängig von der Klasse}} \end{aligned}$$

$$\begin{aligned} P(y=1 | x, \theta) &= \frac{\exp(x^T \beta_1 + \mu_1)}{\exp(x^T \beta_0 + \mu_0) + \exp(x^T \beta_1 + \mu_1)} \\ &= \frac{1}{1 + \exp((\beta_1 - \beta_0)^T x + (\mu_1 - \mu_0))} \end{aligned}$$

Worum? Mit $\sigma(x) := \frac{1}{1 + \exp(-x)}$ (Sigmoid-Funktion) erhalten wir

$$P(y=1 | x, \theta) = \sigma((\beta_1 - \beta_0)^T x + (\mu_1 - \mu_0)) \quad (\times)$$



Betrachten wir $\pi_1 - \pi_0$:

$$\begin{aligned}\pi_1 - \pi_0 &= \log(\pi_1) - \frac{1}{2} \cdot \mu_1^\top \Sigma^{-1} \mu_1 - \log(\pi_0) + \frac{1}{2} \cdot \mu_0^\top \Sigma^{-1} \mu_0 \\ &= \log\left(\frac{\pi_1}{\pi_0}\right) - \frac{1}{2} \cdot (\mu_1 - \mu_0)^\top \Sigma^{-1} (\mu_1 + \mu_0)\end{aligned}$$

Betrachten wir $\beta_1 - \beta_0$:

$$\beta_1 - \beta_0 = \sum \mu_1 - \sum \mu_0 = \sum (\mu_1 - \mu_0) =: w$$

Ansetz: $x_0 = \frac{1}{2} \cdot (\mu_1 + \mu_0) - (\mu_1 - \mu_0) \cdot \frac{\log\left(\frac{\pi_1}{\pi_0}\right)}{(\mu_1 - \mu_0)^\top \Sigma^{-1} (\mu_1 + \mu_0)}$

Worum? $w^\top x_0 = (\mu_1 - \mu_0)^\top \Sigma^{-1} (\mu_1 + \mu_0) \cdot \frac{1}{2} - (\mu_1 - \mu_0)^\top \cancel{\Sigma^{-1}} (\mu_1 - \mu_0) \cdot \frac{\log\left(\frac{\pi_1}{\pi_0}\right)}{(\mu_1 - \mu_0)^\top \cancel{\Sigma^{-1}} (\mu_1 - \mu_0)}$
 \Rightarrow das ist genau $-(\pi_1 - \pi_0)$ raus oben.

$$w^\top x_0 = -(\pi_1 - \pi_0)$$

Einsetzen in (x) ergibt:

$$\begin{aligned}p(y=1 | x, \theta) &= \sigma\left(\underbrace{(\beta_1 - \beta_0)^\top x}_{w^\top} + (\pi_1 - \pi_0)\right) \\ &= \sigma(w^\top x - w^\top x_0) \\ &= \sigma(w^\top(x - x_0))\end{aligned}$$

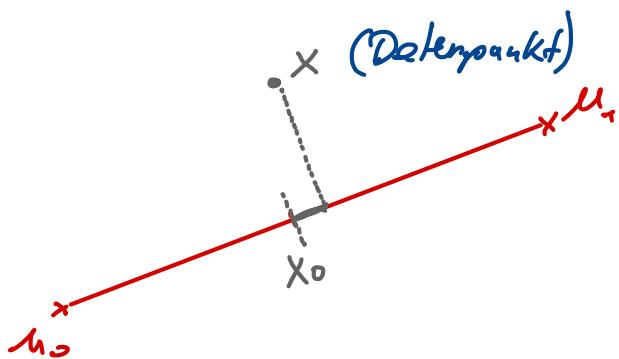
\uparrow
Datenpunkt

Beobachtung 1: Ist also $w^\top x > w^\top x_0$, also $w^\top(x - x_0)$ positiv, dann
ist $\sigma(w^\top(x - x_0)) > 0.5 \Rightarrow p(y=1 | x, \theta) > p(y=0 | x, \theta)$
 \Rightarrow wir weisen x der Klasse 1 zu.

Geometrisch: Wenn $\Sigma_1 = \Sigma_0 = \Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$ und $\pi_1 = \pi_0$, erhalten wir:

$$\omega = \Sigma^{-1}(\mu_1 - \mu_0), \text{ also } \mu_1 - \mu_0$$

$$x_0 = \frac{1}{2} \cdot (\mu_1 + \mu_0)$$



Würden wir $\pi_1 > \pi_0$ haben (d.h. nicht $\pi_1 = \pi_0$), dann hätten wir

$$p(y=1) > p(y=0)$$

und es reicht nicht mehr, über der "Kritte" zu liegen, um x der Klasse 0 zuzuweisen, sondern wir müssen näher an μ_0 rutschen.

Einschub: Maximum-Likelihood-Estimation (MLE)

Daten $\mathcal{D} = ((x_1, y_1), \dots, (x_N, y_N))$

$$\underbrace{p(y|x, \theta)}_{\text{Modell mit Parametern } \theta}$$

$p(\mathcal{D}|\theta)$... Likelihood

W-keit der Daten \mathcal{D} unter Modell mit Parametern θ

Wir möchten θ so wählen, dass \mathcal{D} die höchste Wahrscheinlichkeit hat (also den größten Likelihood).

$$\hat{\theta}_{\text{MLE}} = \underset{\Theta}{\operatorname{argmax}} \rho(\mathcal{D} | \Theta)$$

Annahme: Daten sind unabhängig und identisch verteilt (iid... independent and identically distributed).

$$\begin{aligned}\hat{\theta}_{\text{MLE}} &= \underset{\Theta}{\operatorname{argmax}} \prod_{i=1}^N \rho(y_i | x_i, \Theta) && \text{Likelihood} \\ \Leftrightarrow & \underset{\Theta}{\operatorname{argmax}} \sum_{i=1}^N \log \rho(y_i | x_i, \Theta) && \text{Log-Likelihood} \\ \Leftrightarrow & \underset{\Theta}{\operatorname{argmin}} - \sum_{i=1}^N \log \rho(y_i | x_i, \Theta) && \text{negative Log-Likelihood (NLL)}\end{aligned}$$

Beispiel: Categorical (Cat) Distribution (Kategoriale Verteilung)

Diskrete Wahrscheinlichkeitsverteilung mit einem Parameter pro Klasse.
($1, \dots, C$ Klassen) -

$$\text{Cat}(y | \Theta) = \prod_{c=1}^C \theta_c^{1_{y=c}}$$

$1_{y=c} = \begin{cases} 1, & \text{wenn } y=c \\ 0, & \text{sonst} \end{cases}$

$$\text{also } \rho(y=c | \Theta) = \theta_c \in [0, 1], \sum_c \theta_c = 1$$

Andere Schreibweise: wir könnten $y=c$ auch als Binär Vektor repräsentieren:

$$[0, 0, \dots, 0, \underset{\text{c-te Stelle}}{1}, 0, \dots]^T$$

Dann:

$$\text{Cat}(y | \Theta) = \prod_{c=1}^C \theta_c^{y_c}$$

Für einen Datensatz $((x_1, y_1), \dots, (x_N, y_N)) = \mathcal{D}$ wollen wir den Likelihood unter unserem Modell anschreiben. Für $p(x | y=c, \theta)$ haben wir bereits die MVN mit Parametern (μ_c, Σ_c) . Für $p(y=c)$ nehmen wir $\text{Cat}(y | \pi)$.

π -priori Wahrscheinlichkeiten

Somit haben wir Parameter $\{\mu_1, \dots, \mu_C, \Sigma_1, \dots, \Sigma_C, \pi_1, \dots, \pi_C\}$.

$$p(\mathcal{D} | \theta) = \prod_{n=1}^N \text{Cat}(y_n | \pi) \cdot \prod_{c=1}^C \text{MVN}(x_n | \mu_c, \Sigma_c)^{\mathbb{1}_{y_n=c}}$$

Logarithmieren:

$$\begin{aligned} \log p(\mathcal{D} | \theta) &= \sum_{n=1}^N \left[\sum_{c=1}^C \mathbb{1}_{y_n=c} \cdot \log(\pi_c) + \log \left(\prod_{n=1}^N \prod_{c=1}^C \text{MVN}(x_n | \mu_c, \Sigma_c)^{\frac{1}{\mathbb{1}_{y_n=c}}} \right) \right] \\ &= -\underbrace{\sum_{n=1}^N \sum_{c=1}^C \mathbb{1}_{y_n=c}}_{n:y_n=c} + \underbrace{\sum_{c=1}^C \sum_{n:y_n=c} \log \text{MVN}(x_n | \mu_c, \Sigma_c)}_{n:y_n=c} \end{aligned}$$

Beobachtung: beide Terme können separat maximiert werden (weil wir ja $\log p(\mathcal{D} | \theta)$)

Term 1:

$$\begin{aligned}
 & \sum_{n=1}^N \sum_{c=1}^C \frac{1}{y_{n=c}} \cdot \log(\pi_c) \\
 &= \sum_{c=1}^C \sum_{n=1}^N \frac{1}{y_{n=c}} \cdot \log(\pi_c) \\
 &= \sum_{c=1}^C \left[\log(\pi_c) \cdot \underbrace{\sum_{n=1}^N \frac{1}{y_{n=c}}}_{=: N_c \dots \text{Anzahl von } y \text{ mit } y=c} \right] \\
 &= \sum_{c=1}^C N_c \cdot \log(\pi_c)
 \end{aligned}$$

Wir versuchen $-\sum_{c=1}^C N_c \cdot \log(\pi_c)$ zu minimieren (unter der Nebenbedingung $\sum_c \pi_c = 1$).

Wir schreiben mit Hilfe von sogen. Lagrange-Multiplizierern (λ)

$$-\sum_{c=1}^C N_c \cdot \log(\pi_c) - \lambda \cdot \underbrace{\left(1 - \sum_c \pi_c\right)}_{\text{Nebenbedingung}} = \mathcal{L}(\pi, \lambda)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -\left(1 - \sum_c \pi_c\right) = 0 \Rightarrow -1 + \sum_c \pi_c = 0 \Rightarrow \sum_c \pi_c = 1$$

$$\frac{\partial \mathcal{L}}{\partial \pi_c} = \underbrace{-\frac{N_c}{\pi_c}}_c + \lambda = 0 \quad (\text{für } \pi_1, \dots, \pi_C)$$

$$\lambda = \frac{N_c}{\pi_c} \Rightarrow N \pi_c = N_c \quad (\text{Wir wissen } \sum_c N_c = N)$$

$$\text{d.h. } \sum_c \lambda \pi_c = \sum_c N_c = N \Rightarrow \boxed{\lambda = N}$$

Wir wissen $\widehat{\pi}_c = N_c \Rightarrow \widehat{\pi}_c = \frac{N_c}{N}$

MLE für π

Term 2:

$$\sum_{c=1}^C \sum_{n: y_n=c} \log MVN(x_n | \mu_c, \Sigma_c)$$

genau der log-Likelihood der MVN für Klasse c.

Eine Herleitung

$\hat{\cdot}$ Notation
= Schätzen

$$\hat{\mu}_{c, \text{mle}} = \frac{1}{N_c} \cdot \sum_{n: y_n=c} x_n$$

$$\hat{\Sigma}_{c, \text{mle}} = \frac{1}{N_c} \cdot \sum_{n: y_n=c} (x_n - \hat{\mu}_c) \cdot (x_n - \hat{\mu}_c)^T$$

Wenn wir wollen können wir $\sum_c = \sum$ (also gleich für alle Klassen)
setzen, und erhalten \rightarrow in dem Fall, lin. Entscheidungsgrenzen
(wie vorher)

$$\hat{\Sigma} = \frac{1}{N} \sum_{c=1}^C \sum_{n: y_n=c} (x_n - \hat{\mu}_c) \cdot (x_n - \hat{\mu}_c)^T$$

Oder: $\Sigma_c = \begin{pmatrix} \theta_{1c}^2 & \theta_{2c}^2 & \dots & \emptyset \\ \emptyset & \ddots & \ddots & \theta_{dc}^2 \end{pmatrix}$ also eine Diagonalmatrix pro Klasse

Haben wir \sum_c diagonal und $\sum_c = \sum$ (also gleich für alle Klassen), nennen wir das "diagonal LDA".

Ann: MVN mit μ und \sum :

$$\frac{1}{(2\pi)^{d/2} \cdot |\sum|} \cdot \exp\left(-\frac{1}{2} \cdot (x-\mu)^T \sum^{-1} (x-\mu)\right)$$

Mit Annahme von $\sum = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_d^2 \end{pmatrix}$ erhalten wir $\sum^{-1} = \begin{pmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_d^2 \end{pmatrix}$

und $|\sum| = \prod_{i=1}^d \sigma_i^2$. Weiters

$$-\frac{1}{2} (x-\mu)^T \sum^{-1} (x-\mu) = \left(-\frac{1}{2}\right) \cdot \left[\sum_{i=1}^d (x_i - \mu_i)^2 \cdot \frac{1}{\sigma_i^2} \right]$$

$$\begin{aligned} \Rightarrow \exp\left(-\frac{1}{2} (x-\mu)^T \sum^{-1} (x-\mu)\right) &= \exp\left(-\frac{1}{2} \cdot \sum_{i=1}^d \frac{(x_i - \mu_i)^2}{\sigma_i^2}\right) \\ &= \prod_{i=1}^d \exp\left(-\frac{1}{2} \frac{(x_i - \mu_i)^2}{\sigma_i^2}\right) \quad (\alpha) \end{aligned}$$

Bzpl. $\frac{1}{(2\pi)^{d/2} \cdot |\sum|}$ erhalten wir:

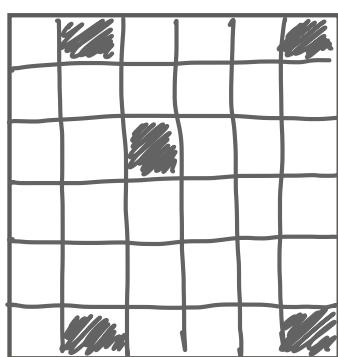
$$\frac{1}{(2\pi)^{d/2} \cdot |\sum|} = \frac{1}{(2\pi)^{d/2} \cdot \prod_{i=1}^d \sigma_i^2} = \prod_{i=1}^d \frac{1}{\sqrt{2\pi \sigma_i^2}} \quad (\beta)$$

Aus (α) und (β) erhalten wir schlussendlich

$$\prod_{i=1}^d \frac{1}{\sqrt{2\pi \sigma_i^2}} \cdot \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right)$$

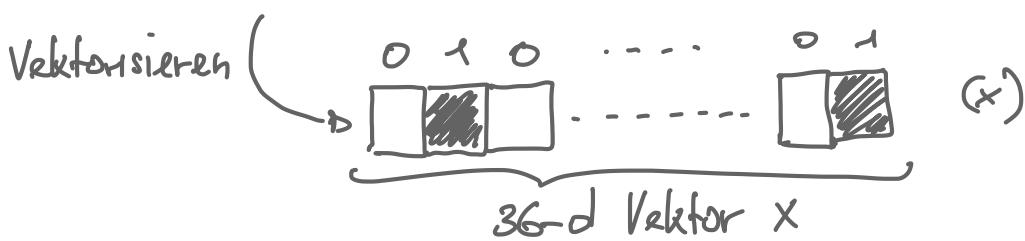
Produkt von 1-D Normalverteilungen $N(x_i | \mu_i, \sigma_i^2)$

Beispiel (zu Normalverteilung als nicht-passendes Modell)



6x6 pixel "Bild"

Pixel sind entweder 0 oder 1
(also "Binärbild")



Hier passt die Normalverteilungsannahme nicht. Wir könnten aber jede einzelne Dimension in (x) als Bernoulli-verteilt modellieren, d.h.

$$\underbrace{\text{Ber}(y_i | \theta)}_{\theta^{\prod_{y_i=1}} \cdot (1-\theta)^{\prod_{y_i=0}}}$$

MLE (für Bernoulli Verteilung): Daten $\underbrace{y_1, \dots, y_n}_{D} \quad y_i \in \{0,1\}$

$$p(D|\theta) = \prod_{i=1}^N \theta^{\prod_{y_i=1}} \cdot (1-\theta)^{\prod_{y_i=0}} \quad // \text{Likelihood}$$

$$\begin{aligned} \log p(D|\theta) &= \sum_{i=1}^N \left(\underbrace{\prod_{y_i=1} \log(\theta)}_{\hookrightarrow \text{Anzahl der } y_i \text{ mit } y_i=1} + \underbrace{\prod_{y_i=0} \log(1-\theta)}_{\hookrightarrow \text{Anzahl der } y_i \text{ mit } y_i=0} \right) \quad // \text{Log-Likelihood} \\ &= N_1 \cdot \log(\theta) + N_0 \cdot \log(1-\theta) \end{aligned}$$

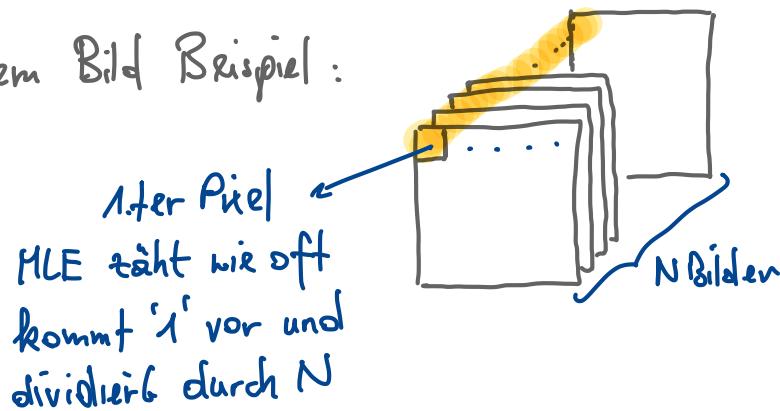
$$\Rightarrow \text{Negative Log-Likelihood (NLL)} : \text{NLL}(\theta) = - (N_1 \cdot \log(\theta) + N_0 \cdot \log(1-\theta))$$

$$\frac{\partial \text{NLL}(\theta)}{\partial \theta} = -\frac{N_1}{\theta} + \frac{N_0}{1-\theta}$$

\Rightarrow nach 0 setzen und auflösen nach θ erhalten wir:

$$\hat{\theta}_{\text{MLE}} = \frac{N_1}{N_0 + N_1} = \frac{N_1}{N}$$

D.h. in unserem Bild Beispiel:



Im Allgemeinen (also nicht auf Normalverteilung beschränkt) haben wir ein Modell der Form:

$$P(y=c | X, \Theta) = \frac{P(y=c) \cdot \prod_{e=1}^d P(x_e | y=c, \theta_{ec})}{\sum_{c'=1}^C P(y=c') \cdot \prod_{e=1}^d P(x_e | y=c', \theta_{ec'})}$$

wir nennen dies einen "Naive Bayes Classifier".

z.B. Parameter der Bernoulli - Verteilung für Koordinate e und Klasse c .

② Logistische Regression (LR)

Im Gegensatz zu LDA, ist LR ein diskriminativer Ansatz!
 wir versuchen $p(y=c|x)$ direkt zu modellieren.

1. Binärer Fall ($y \in \{0,1\}$): Da $y \in \{0,1\}$, bietet es sich an $p(y|x)$ als Bernoulli-Verteilung zu modellieren, wobei wir den Parameter der Bernoulli-Verteilung abhängig vom Input (also andere x) machen, z.B.

$$\text{Ber}(y | f(x; \alpha))$$

Funktion von x mit Parameter α
 (wir wollen $f(x; \alpha) \in [0,1]$)

Nehmen wir an $x \in \mathbb{R}^d$, $y \in \{0,1\}$. In der LR haben wir folgendes Modell

$$p(y|x, \theta) = \text{Ber}(y | \sigma(w^T x + b))$$

inkludierbar, jetzt w und b !

Setzen ob nun $\alpha := w^T x + b$. Wir haben

$$p(y=1|x, \theta) = \sigma(\alpha) = \frac{1}{1+e^{-\alpha}}$$

$$p(y=0|x, \theta) = 1 - p(y=1|x, \theta) = 1 - \frac{1}{1+e^{-\alpha}} \\ = \frac{1+e^{-\alpha}-1}{1+e^{-\alpha}} = \frac{e^{-\alpha}}{1+e^{-\alpha}} = \frac{1}{1+e^{\alpha}} = \sigma(-\alpha)$$

(Ann.: wir können uns b subsummiert in w denken, also
 $w = [b, w_1, \dots, w_d]^T$ und $x = [1, x_1, \dots, x_d]^T \Rightarrow$ in \mathbb{R}^{d+1} eigentlich)

Wie bekommen wir w ? MLE

$$NLL(w) = -\frac{1}{N} \sum_{n=1}^N \log \text{Ber}(y_n | \underbrace{\sigma(w^T x_n)}_{=: M_n})$$

$$(x_1, y_1), \dots, (x_N, y_N); x \in \mathbb{R}^d; y_n \in \{0, 1\}$$

Einsetzen:

$$\begin{aligned} NLL(w) &= -\frac{1}{N} \cdot \sum_{n=1}^N \log \left(M_n^{y_n} \cdot (1-M_n)^{1-y_n} \right) \\ &= -\frac{1}{N} \cdot \sum_{n=1}^N \left[y_n \cdot \log(M_n) + (1-y_n) \cdot \log(1-M_n) \right] \\ &\quad \xrightarrow{\sigma(w^T x_n)} \end{aligned}$$

Wir haben $w \in \mathbb{R}^d$ (ad. \mathbb{R}^{d+1}). Wir versuchen den Gradienten von $NLL(w)$ (also $\nabla_w NLL(w)$) Null zu setzen und nach w auflösen.

$$\hookrightarrow \nabla_w NLL(w) = \begin{pmatrix} \frac{\partial NLL(w)}{\partial w_1} \\ \vdots \\ \frac{\partial NLL(w)}{\partial w_d} \end{pmatrix}$$

Wir wissen

$M_n = \sigma(w^T x_n)$
$a_n = w^T x_n$

also $M_n = \sigma(a_n)$

$$1) \frac{\partial M_n}{\partial a_n} = \sigma(a_n) \cdot (1-\sigma(a_n))$$

$$2) \frac{\partial M_n}{\partial w_d} = \frac{\partial}{\partial w_d} \sigma(\underbrace{w^T x_n}_{a_n}) = \underbrace{\frac{\partial}{\partial a_n} \sigma(a_n)}_{\text{siehe (1), also } \sigma'(a_n)} \cdot \frac{\partial}{\partial w_d} a_n$$

$$\text{siehe (1), also } \sigma'(a_n) \cdot (1-\sigma(a_n))$$

$$\Rightarrow \frac{\partial \mu_n}{\partial w_d} = \sigma(\mu_n) \cdot (1 - \sigma(\mu_n)) \cdot x_{nd}$$

$\hookrightarrow \frac{\partial}{\partial w_d} \sigma(\mu_n)$

$$= \mu_n \cdot (1 - \mu_n) \cdot x_{nd}$$

Wir haben also $\nabla_w \log(\mu_n) = \frac{1}{\mu_n} \cdot \nabla_w \mu_n$

$$= \begin{pmatrix} \frac{1}{\mu_n} \cdot \frac{\partial}{\partial w_1} \mu_n \\ \vdots \\ \frac{1}{\mu_n} \cdot \frac{\partial}{\partial w_d} \mu_n \end{pmatrix} = \begin{pmatrix} \frac{1}{\mu_n} \cdot \mu_n \cdot (1 - \mu_n) x_{n1} \\ \vdots \\ \frac{1}{\mu_n} \cdot \mu_n \cdot (1 - \mu_n) \cdot x_{nd} \end{pmatrix}$$

$$= \begin{pmatrix} (1 - \mu_n) \cdot x_{n1} \\ \vdots \\ (1 - \mu_n) \cdot x_{nd} \end{pmatrix} = (1 - \mu_n) \cdot \begin{pmatrix} x_{n1} \\ \vdots \\ x_{nd} \end{pmatrix}$$

$(1 - \mu_n) x_n$

Auf gleiche Art u. Weise bekommen wir

$$\nabla_w \log(1 - \mu_n) = -\mu_n \cdot \begin{pmatrix} x_{n1} \\ \vdots \\ x_{nd} \end{pmatrix} = -\mu_n \cdot x_n$$

$$\Rightarrow \nabla_w \text{NLL}(\omega) = -\frac{1}{N} \cdot \sum_{n=1}^N \left[y_n \cdot (1 - \mu_n) \cdot x_n - (1 - y_n) \mu_n \cdot x_n \right]$$

$$= -\frac{1}{N} \cdot \sum_{n=1}^N \left[y_n x_n - y_n \mu_n x_n - \mu_n x_n + y_n \mu_n x_n \right]$$

$$= +\frac{1}{N} \cdot \sum_{n=1}^N (\mu_n - y_n) \cdot x_n$$

Datenpunkt
Label $\in \{0, 1\}$ (gewünscht)
 $\Rightarrow \sigma(w^\top x_n)$

Ziel ist ja $\nabla_w \text{NLL}(\omega) = 0$ nach w zu lösen.

Definieren

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & & \vdots \\ x_{N1} & \dots & \dots & x_{Nd} \end{pmatrix}$$

$N \times d$ Matrix,
"Design Matrix"

$$\Rightarrow \nabla_w \text{NLL}(\omega) = X^T (\mu - y) \cdot \frac{1}{N}$$

$$\begin{pmatrix} \theta(w^T x_1) \\ \vdots \\ \theta(w^T x_N) \end{pmatrix} \quad \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

Warum?

$$\frac{1}{N} \cdot \begin{pmatrix} x_{11} & \dots & x_{N1} \\ \vdots & & \vdots \\ x_{1d} & \dots & x_{Nd} \end{pmatrix} \begin{pmatrix} \mu_1 - y_1 \\ \vdots \\ \mu_N - y_N \end{pmatrix} = \begin{pmatrix} x_{11} \cdot (\mu_1 - y_1) + x_{21} \cdot (\mu_2 - y_2) + \dots \\ x_{12} \cdot (\mu_1 - y_1) + x_{22} \cdot (\mu_2 - y_2) + \dots \\ \vdots \\ x_{1d} \cdot (\mu_1 - y_1) + \dots \end{pmatrix}$$

X^T x_1 x_2

$$\nabla_w \text{NLL}(\omega) = X^T (\mu - y) \cdot \frac{1}{N}$$

Betrachten die Hesse-Matrix $H(\omega)$

$$\left(\begin{array}{c} \frac{\partial \text{NLL}(\omega)}{\partial \omega_n \partial \omega_1} \dots \\ \frac{\partial \text{NLL}(\omega)}{\partial \omega_1 \partial \omega_2} \\ \vdots \\ \frac{\partial \text{NLL}(\omega)}{\partial \omega_n \partial \omega_n} \end{array} \right) \quad \text{d} \times \text{d} \text{ Matrix}$$

Wir wissen

$$\frac{\partial}{\partial \omega_j} \text{NLL}(\omega) = \frac{1}{N} \cdot \sum_{n=1}^N (\mu_n - y_n) \cdot x_{nj} \sigma(\omega^T x_n)$$

$$\begin{aligned} \frac{\partial}{\partial \omega_j \partial \omega_k} \text{NLL}(\omega) &= \frac{1}{N} \cdot \sum_{n=1}^N x_{nj} \frac{\partial}{\partial \omega_k} \mu_n \quad \text{kennen wir schon von vorher} \\ &= \frac{1}{N} \cdot \sum_{n=1}^N x_{kj} \cdot x_{nk} \cdot (1-\mu_n) \cdot \mu_n \end{aligned}$$

Mit $z_j = (x_{1j}, \dots, x_{Nj})^T$ und $z_k = (x_{1k}, \dots, x_{Nk})^T$, haben wir

$$\frac{\partial}{\partial \omega_j \partial \omega_k} \text{NLL}(\omega) = z_j^T B z_k$$

$$\left(\begin{array}{ccc} \mu_1 \cdot (1-\mu_1) & & \\ & \ddots & \\ & & \mu_N \cdot (1-\mu_N) \end{array} \right) \quad N \times N \text{ Matrix}$$

Also

$$\begin{aligned} H(\omega) &= \nabla^2 \text{NLL}(\omega) \\ &= \frac{1}{N} \cdot X^T B X \end{aligned}$$

\Rightarrow der B nur positive Einträge hat können wir B als $B^{\frac{1}{2}} \cdot B^{\frac{1}{2}}$ schreiben.

$$\Rightarrow H(\omega) = \frac{1}{N} \cdot X^T B^{\frac{1}{2}} \cdot B^{\frac{1}{2}} \cdot X$$

$$= \frac{1}{N} \cdot (B^{\frac{1}{2}} X)^T \cdot (B^{\frac{1}{2}} X) \quad \text{also in der Form } A^T A, A = (B^{\frac{1}{2}} X)$$

Wir erinnern: eine reelle $N \times N$ Matrix C ist positiv semi-definit
wenn für alle $b \in \mathbb{R}^N$ gilt:

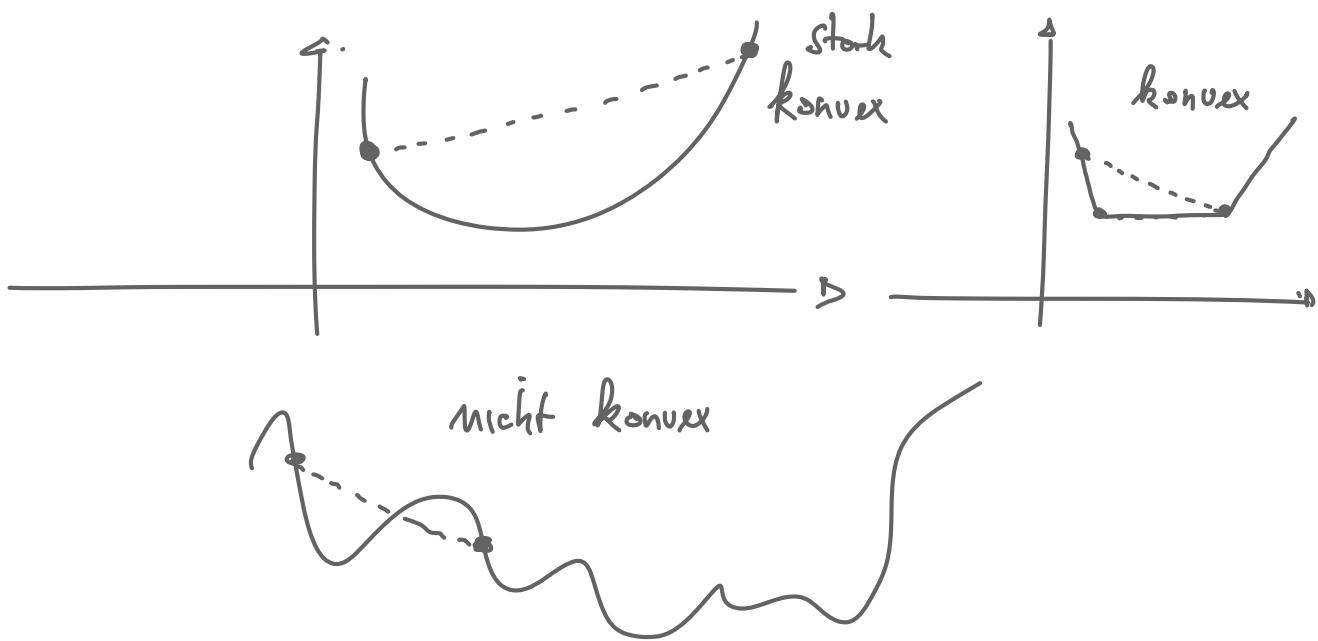
$$b^T C b \geq 0 \quad (\text{also nicht-negativ})$$

Bei uns: $b^T H(\omega) b = (b^T A^T A b) \cdot \frac{1}{N}$

$$= (Ab)^T \cdot (Ab) \cdot \frac{1}{N} = \frac{1}{N} \cdot \|Ab\|^2 \geq 0 \quad (\text{für alle } b \in \mathbb{R}^N)$$

→ $H(\omega)$ ist positiv semi-definit!

Eine 2-mal differenzierbare Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ist konvex
falls $\nabla^2 f$ positiv semi-definit ist.



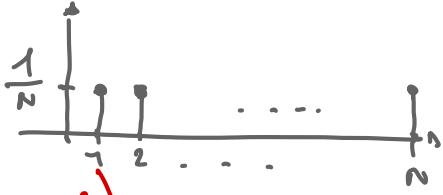
Einschub: Stochastischer Gradientenabstieg (Stochastic Gradient Descent - SGD)

Wir hatten $\nabla_w NLL(\omega) = \frac{1}{N} \cdot \sum_{n=1}^N (u_n - y_n) \cdot x_n$

Update Regel für Gradientenabstieg (in Iteration t):

$$\omega^{(t+1)} = \omega^{(t)} - \eta_t \cdot \nabla_w NLL(\omega) \quad \omega^{(0)} \dots \text{Startpunkt}$$

η ... Schrittweite / Lernrate



SGD Variante:

$$\left. \begin{array}{l} \tilde{n} \sim \text{Punktverteilung } \{\tilde{1}, \dots, \tilde{N}\} \\ w^{(t+1)} = w^{(t)} - \eta_t \cdot (u_{\tilde{n}} - y_{\tilde{n}}) \cdot x_{\tilde{n}} \end{array} \right\} T-\text{mal}$$

Ein Term an Stelle \tilde{n} von der Summe $(\nabla_w \text{NLL}(w))$ oben.

2. Mehrklassen-Fall LR $(y \in \{1, \dots, C\})$

Anzahl an Klassen

Wir versuchen $p(y|x, \theta)$ zu modellieren. Wie vorher $x \in \mathbb{R}^d$:

$$p(y|x, \theta) = \underset{\{W, b\}}{\underset{||}{=}} \text{Cat}(y | \text{softmax}(W^T x + b)) \rightarrow \mathbb{R}^C$$

$d \times C$ Matrix W
 $\Rightarrow W^T x + b \in \mathbb{R}^C$

softmax: $\mathbb{R}^C \rightarrow [0, 1]^C$

$$a \mapsto \text{softmax}(a) = \left[\underbrace{\frac{e^{a_1}}{\sum_{c'} e^{a_{c'}}}, \dots, \frac{e^{a_C}}{\sum_{c'} e^{a_{c'}}}}_{\in (0, 1)} \right]$$

$[a_1, \dots, a_C]^T$

- Eigenschaften:
- 1) $0 < \text{softmax}(a)_i < 1$ für alle $i \in \{1, \dots, C\}$
 - 2) $\sum_i \text{softmax}(a)_i = 1$

Zuerst schreiben wir $y = c$ als

$$y = [0, 0, \dots, \underset{c\text{-ten Stelle}}{\uparrow} 1, \dots, 0]$$

und dann

$$\text{Cat}(y | \theta) = \prod_{c=1}^C \theta_c^{y_c}$$

Allg.

Mit dieser Schreibweise erhalten wir

$$\text{NLL}(\mathbf{w}) = -\frac{1}{N} \cdot \log \left(\prod_{n=1}^N \prod_{c=1}^C \alpha_{nc}^{y_{nc}} \right) \text{ mit}$$

$$\alpha_{nc} = \text{softmax} \left(\mathbf{w}^T \mathbf{x}_n \right)_c \quad (\text{ohne } +b)$$

$$\left(\alpha_n = \text{softmax} \left(\mathbf{w}^T \mathbf{x}_n \right) = \left[\dots, \frac{e^{(\mathbf{w}^T \mathbf{x}_n)_c}}{\dots}, \dots \right] \right)$$

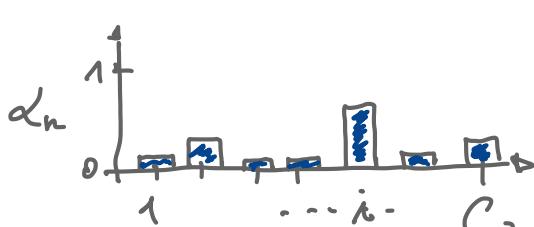
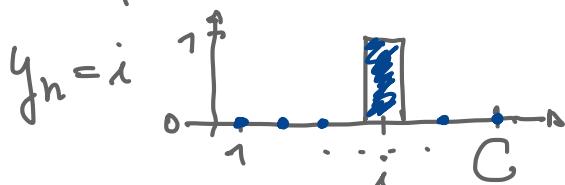
c-te Komponente

$$\Rightarrow \text{NLL}(\mathbf{w}) = -\frac{1}{N} \cdot \sum_{n=1}^N \sum_{c=1}^C y_{nc} \cdot \log(\alpha_{nc})$$

$$= +\frac{1}{N} \cdot \sum_{n=1}^N \text{CrossEntropy} (y_n, \alpha_n)$$

\hookrightarrow Kreuzentropie

Ahme: für ein fixes n :



Konkret (ohne Herleitung)

$$\nabla_W \text{NLL}(W) = \frac{1}{N} \cdot \sum_{n=1}^N x_n \cdot \underbrace{(d_n - y_n)}_{\substack{\text{Vektor } C \times 1 \\ \text{Vektor } C \times 1}}^T$$

\downarrow

Vektor $d \times 1$
 $d \times C$ Matrix

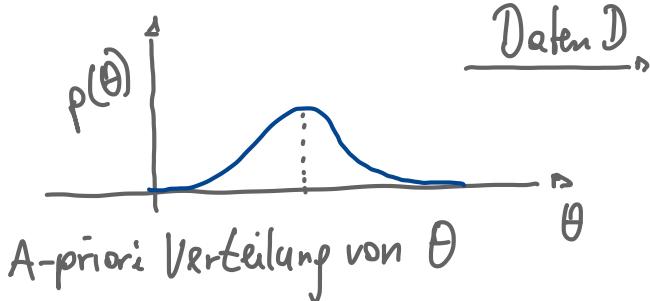
können wir für SGD
benutzen um $\text{NLL}(W)$
zu minimieren!

$$(W^{(t+1)} = W^{(t)} - \eta_t \cdot x_n (d_n - y_n)^T)$$

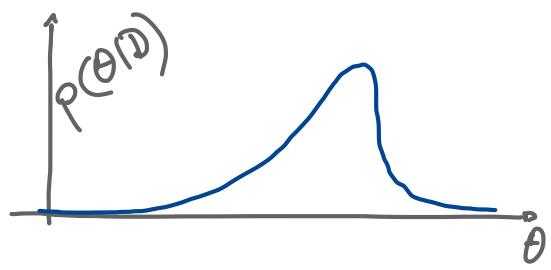
(θ)

Bisher haben wir immer MLE benutzt um Parameter zu schätzen,
d.h. wir haben θ so gewählt das $p(D|\theta)$ maximiert wird!
 \downarrow
Daten

Sagen wir nun, wir hätten "Vorwissen", in der Form $p(\theta)$ zur
Verfügung.



Daten D ,



$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}} \ p(\theta | D)$$

MAP... Maximum A-posteriori Schätzer

Wir haben

$$p(\theta | D) = \frac{P(\theta) \cdot P(D|\theta)}{\text{Normalisierungsterm}}$$

(Anw. von Bayes Regel)

↓
Vorwissen

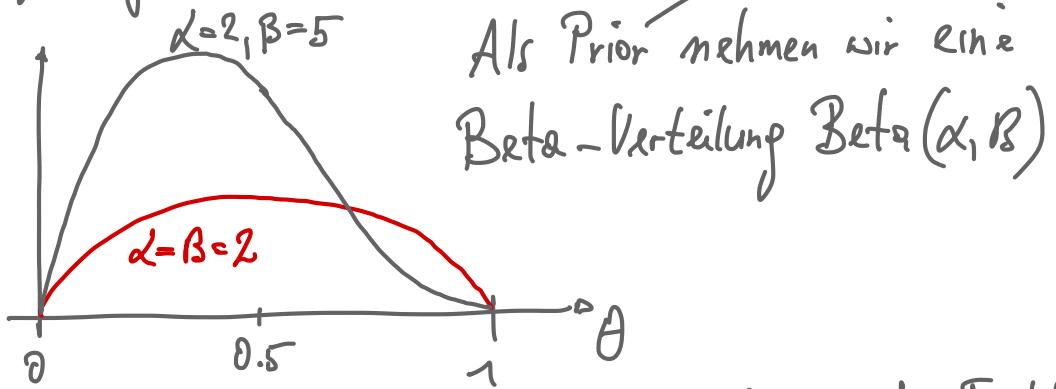
Likelihood

$$\Rightarrow \hat{\theta}_{MAP} = \operatorname{argmax}_{\theta} p(\theta | D)$$

$$= \operatorname{argmax}_{\theta} p(\theta) \cdot p(D|\theta)$$

$$\Leftrightarrow \hat{\theta}_{MAP} = \operatorname{argmin}_{\theta} - \underbrace{\log p(\theta)}_{\text{noch log und } \times (-1)} - \underbrace{\log p(D|\theta)}_{\text{hatten wir bereits bei MLE}}$$

Beispiel (Münzwurf): $y \sim \text{Ber}(\theta)$, $\theta \in (0,1) \rightarrow p(\theta)$



$$p(\theta) = \frac{1}{B(\alpha, \beta)} \cdot \theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}$$

($B(\alpha, \beta)$... Beta Funktion
mit $\alpha > 0, \beta > 0$)

Mit Daten y_1, \dots, y_N ($y_i \in \{0, 1\}$), sei N_1 die Anzahl an y_i mit $y_i=1$ und N_0 die Anzahl an y_i mit $y_i=0$ ($N_0 + N_1 = N$).

Wir haben: $- \log p(\theta | D) = \underbrace{-\log p(\theta)}_{(\log) \text{ Beta Prior}} - \underbrace{\log p(D|\theta)}_{(\log) \text{ Bernoulli}}$

Einsetzen ergibt: $-\log p(\theta | D) = -[N_1 \cdot \log(\theta) + N_0 \cdot \log(1-\theta)]$

$-[(\alpha-1) \cdot \log(\theta) + (\beta-1) \cdot \log(1-\theta)] + \text{const.}$

Minimieren von $-\log p(\theta | D)$ bzgl. θ ergibt:

$$\hat{\theta}_{MAP} = \frac{N_1 + \alpha - 1}{N_0 + N_1 + \alpha + \beta - 2}$$

$\left(\frac{N_1}{N} \text{ wäre MLE} \right)$

Mit $\underbrace{\alpha = \beta = 2}$:
Vorwissen

$$\hat{\theta}_{MAP} = \frac{N_1 + 1}{N_0 + N_1 + 2}$$

"Add-one Smoothing"

Wenn $N=0$ (kein einziger Wurf) $\hat{\theta}_{MAP} = \frac{1}{2}$

Konkreteres Beispiel (LR im binären Fall, also $y \in \{0, 1\}$). Wir haben $x \in \mathbb{R}^d$ und $w \in \mathbb{R}^d$ als Parameter (Lassen + b Wsp).

Wir nehmen als $p(w)$ folgendes:

$$p(w) = N(w | 0, s^2 I) \xrightarrow{\substack{\text{d} \times d \text{ Matrix} \\ (\text{Kovarianz})}} \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \xrightarrow{\substack{\downarrow [0, 0, 0, \dots]^T}}$$

mit $s > 0$.

$$N(w | 0, s^2 I) = (2\pi)^{-d/2} \cdot \det(s^2 I)^{-\frac{1}{2}} \cdot \exp \left\{ w^T \begin{pmatrix} s^2 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & s^2 \end{pmatrix} w - \frac{1}{2} \right\}$$

$$\exp \left\{ -\frac{1}{2s^2} \cdot w^T w \right\} =$$

$$\exp \left\{ -\frac{1}{2s^2} \cdot \|w\|^2 \right\}$$

Um \hat{w}_{MAP} zu bestimmen, minimieren wir

$$\hat{\theta}_{MAP} = \underset{w}{\operatorname{arg\,min}} \text{ NLL}(w) + \frac{1}{2s^2} \cdot \|w\|^2$$

$$\hat{\theta}_{MAP} = \underset{w}{\operatorname{argmin}} \text{NLL}(w) + \frac{1}{2s^2} \cdot \|w\|^2$$

$\underbrace{\phantom{\hat{\theta}_{MAP} = \underset{w}{\operatorname{argmin}} \text{NLL}(w) + \frac{1}{2s^2} \cdot \|w\|^2}}_{=: \lambda}$

Regularisierungsterm

10.11.25

Lineare Regression

Wir wenden uns kurz dem Fall $y \in \mathbb{R}$ zu.

$$p(y|x, \theta) = N(y \mid \underbrace{w^T x + w_0}_{\text{Mittelwert}}, \sigma^2) \quad \text{mit } \theta = \{w, w_0, \sigma^2\}$$

Varianz

Wir können natürlich w_0 in w subsummieren. Der $\text{NLL}(w)$ sieht folgendermaßen aus (gegebenen Daten $(x_1, y_1), \dots, (x_N, y_N)$ mit $x_i \in \mathbb{R}^d$ und $y \in \mathbb{R}$):

$$\begin{aligned} \text{NLL}(\theta) &= \text{NLL}(w, \sigma^2) = - \sum_{n=1}^N \log \left[\left(\frac{1}{2\pi\sigma^2} \right)^{\frac{1}{2}} \cdot e^{-\frac{1}{2\sigma^2} (y_n - w^T x_n)^2} \right] \\ &= - \sum_{n=1}^N \left[\frac{1}{2} \cdot \log \left(\frac{1}{2\pi\sigma^2} \right) - \frac{1}{2\sigma^2} \cdot (y_n - w^T x_n)^2 \right] \\ &= - \underbrace{\frac{N}{2} \cdot \log \left(\frac{1}{2\pi\sigma^2} \right)}_{\text{additive Konstante}} + \underbrace{\frac{1}{2\sigma^2} \cdot \sum_{n=1}^N (y_n - w^T x_n)^2}_{\text{Summe der Fehlerquadratsumme}} \end{aligned}$$

Fixieren wir σ^2 !

(a) ohne additive Konstante u. ohne $\frac{1}{2\sigma^2}$ (weil fix)

$$\sum_{n=1}^N (y_n - \omega^T k_n)^2 =: \text{RSS}(\omega)$$

Residual =: r_n

"Residual Sum of Squares"

D.h. wir versuchen

$$\sum_{n=1}^N r_n^2$$

bzgl. ω zu minimieren.

(b) wie (a) aber mit $\frac{1}{N}$ skaliert:

$$\frac{1}{N} \sum_{n=1}^N r_n^2 =: \text{MSE}(\omega)$$

"Mean Squared Error"

d.h. wir könnten auch den MSE bzgl. ω minimieren.

(c) gleich wie (b) aber mit



$$\sqrt{\frac{1}{N} \sum_{n=1}^N r_n^2} =: \text{RMSE}(\omega)$$

"Root Mean Squared Error"

— · —

Nehmen wir (a):

$$\hat{\omega}_{MLE} = \arg \min_{\omega} \text{RSS}(\omega)$$

D.h. wir versuchen $\nabla_{\omega} \text{RSS}(\omega) = 0$ nach ω auflösen.

Mit

$$X = \begin{pmatrix} X_{11} & \dots & X_{1d} \\ X_{21} & \dots & X_{2d} \\ \vdots & & \\ X_{N1} & \dots & X_{Nd} \end{pmatrix}, \quad w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix}, \quad Xw = \begin{pmatrix} \langle w, x_1 \rangle \\ \vdots \\ \langle w, x_N \rangle \end{pmatrix} = \begin{pmatrix} w^T x_1 \\ \vdots \\ w^T x_N \end{pmatrix}$$

$N \times d$ matrix $d \times 1$ Vektor $N \times 1$ Vektor

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

Können wir RSS(w) als

$$\begin{aligned} \text{RSS}(w) &= \|Xw - y\|^2 \\ &= (Xw - y)^T (Xw - y) \\ &= w^T X^T X w - 2w^T X^T y + y^T y \end{aligned}$$

Anm.: $\frac{\partial}{\partial X} X^T A X = (A + A^T) X$ (siehe "Matrix Cook Book")

Also $\nabla_w \text{RSS}(w) = (X^T X + X^T X) w - 2X^T y$

$\underbrace{d \times N}_{\text{d}} \quad \underbrace{N \times 1}_{\text{1}}$

Null setzen: $(X^T X + X^T X) w - 2X^T y = 0$

$$\Leftrightarrow X^T X w = X^T y$$

$$\Leftrightarrow w = (X^T X)^{-1} X^T y$$

Also wäre unser

$$\hat{w}_{\text{true}} = (X^T X)^{-1} X^T y$$

Hat X vollen Rang, erhalten wir mit \hat{w}_{MLE} ein eindeutiges globales Minimum von $\text{RSS}(\omega)$. $[(X^T X)^{-1} X^T \dots \text{Pseudo-Inverse von } X]$.

Anm.: Einsetzen v. \hat{w}_{MLE} als $\text{NLL}(\hat{w}_{\text{MLE}}, \sigma^2)$ und minimieren bzgl. σ^2

ergibt:

$$\hat{\sigma}_{\text{MLE}}^2 = \frac{1}{N} \cdot \| X \hat{w}_{\text{MLE}} - y \|^2$$

↪ eigentlich (unbiased) $\frac{1}{N-p}$
Freiheitsgrade

Alternativ sehen wir uns den MAP Schätzer für w an.

Wir nehmen

$$p(w) = N(w | 0, s^2 I_d)$$

$$\begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{pmatrix}$$

als Prior für w (mit $s^2 > 0$).

Die Negative Log-Posterior sieht folgendermaßen aus:

$$\frac{1}{2\sigma^2} \cdot (Xw - y)^T (Xw - y) + \frac{1}{2s^2} w^T w + \text{Konstanten} = : J(w)$$

Also:

$$J(w) = \frac{1}{2\sigma^2} \cdot \left[w^T X^T X w - 2w^T X^T y + y^T y \right] + \frac{1}{2s^2} \cdot w^T w$$

$$\Rightarrow \frac{\partial J(w)}{\partial w} = \frac{1}{2\sigma^2} \left[\underbrace{X^T X w}_{\substack{n \\ 1 \times d}} - \underbrace{X^T y}_{\substack{1 \times N}} \right] + \frac{1}{2s^2} \cdot X w^T \xrightarrow{1 \times d}$$

$\underbrace{\quad}_{\substack{N+1 \\ 1 \times d}}$

$$= \frac{1}{\sigma^2} \cdot \left[w^T X^T X - y^T X \right] + \frac{1}{s^2} w^T$$

Null setzen und nach w auflösen:

$$w^T \cdot \left[\underbrace{\frac{1}{\sigma^2} X^T X}_{d \times d} + \frac{1}{s^2} \cdot I_d \right] = y^T X \cdot \frac{1}{\sigma^2} \quad | \cdot \sigma^2$$

$$\begin{pmatrix} 1 & \dots & 0 \\ 0 & \ddots & \vdots \\ \vdots & \ddots & 1 \end{pmatrix}$$

$$\Leftrightarrow w^T \cdot \left[X^T X + \frac{\sigma^2}{s^2} I_d \right] = y^T X$$

$$\Rightarrow \widehat{w}_{\text{MAP}}^T = y^T X \left(X^T X + \underbrace{\frac{\sigma^2}{s^2} I_d}_{\text{Matrix}} \right)^{-1}$$

$$\left(\begin{array}{cccc} \frac{\sigma^2}{s^2} & & & \\ & \ddots & & \\ & & \frac{\sigma^2}{s^2} & \\ & & & \ddots & \frac{\sigma^2}{s^2} \end{array} \right) = : \lambda$$

Rechnet man lin. Reg. mit \widehat{w}_{MAP} neigt man dies "Regularized linear regression" (Regularisierte lineare Regression).

Bisher hatten wir (ohne Bias Terme)

$$(A) p(y|x, w) = \text{Ber}(y | \sigma(w^T x))$$

2-klassen LR (0/1)

$$(B) p(y|x, w) = \text{Cat}(y | \text{softmax}(w^T x))$$

C-klassen LR

$$(C) p(y|x, w, \sigma) = N(y | w^T x, \sigma^2)$$

Lin. Regression

Wir könnten auch $\phi: \mathbb{R}^d \rightarrow \mathbb{R}^e$ definieren (quasi als Transformation von $x \in \mathbb{R}^d$)

$$f(x; \theta) = W\phi(x) + b, \quad \theta = \{W, b\}$$

Statt wx in (B) mit w als $C \times e$ Matrix und $b \in \mathbb{R}^c$. Oder,

allgemeiner

$$f(x; \theta) = W\phi(x; \theta_1) + b, \quad \theta = \{W, b, \theta_1\}$$

Oder, noch allgemeiner

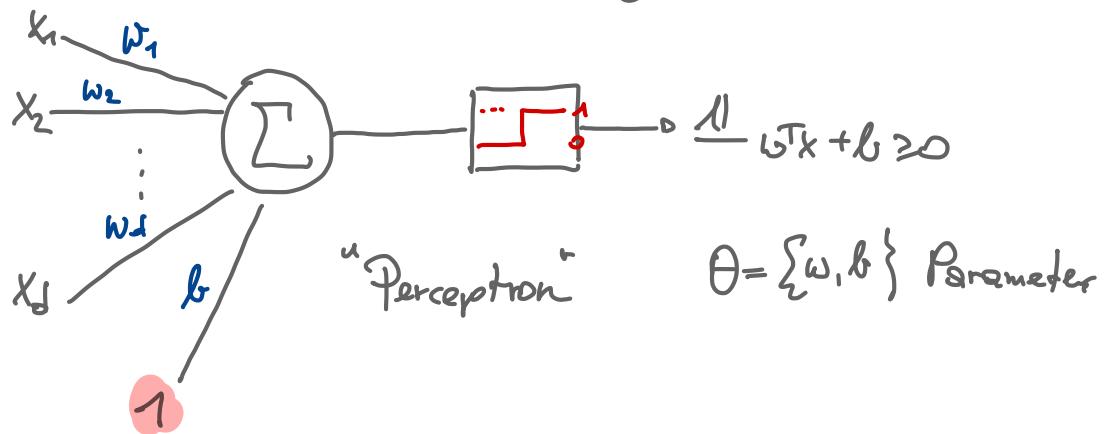
$$f(x; \theta) = W f_L(f_{L-1}(\dots f_1(x)) \dots) + b, \quad \text{mit}$$

$$f_e(x) = f(x; \theta_e)$$

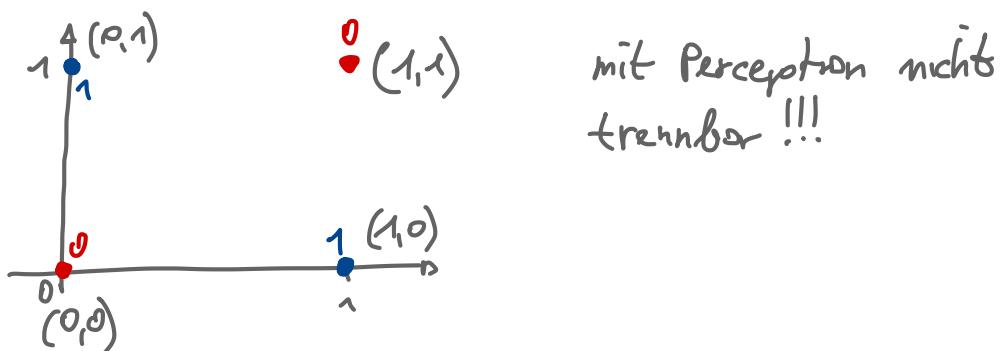
Einschub (XOR): Perceptron

$$f(x; \theta) = \frac{1}{1 + e^{-w^T x + b}} = \begin{cases} 1, & \text{wenn } w^T x + b \geq 0 \\ 0, & \text{sonst} \end{cases}$$

$x \in \mathbb{R}^d$
 $w \in \mathbb{R}^d$
 $b \in \mathbb{R}$



XOR:		x_1	x_2	$x_1 \oplus x_2$
0	0	0	0	•
0	1	1	1	•
1	0	1	1	•
1	1	0	0	•



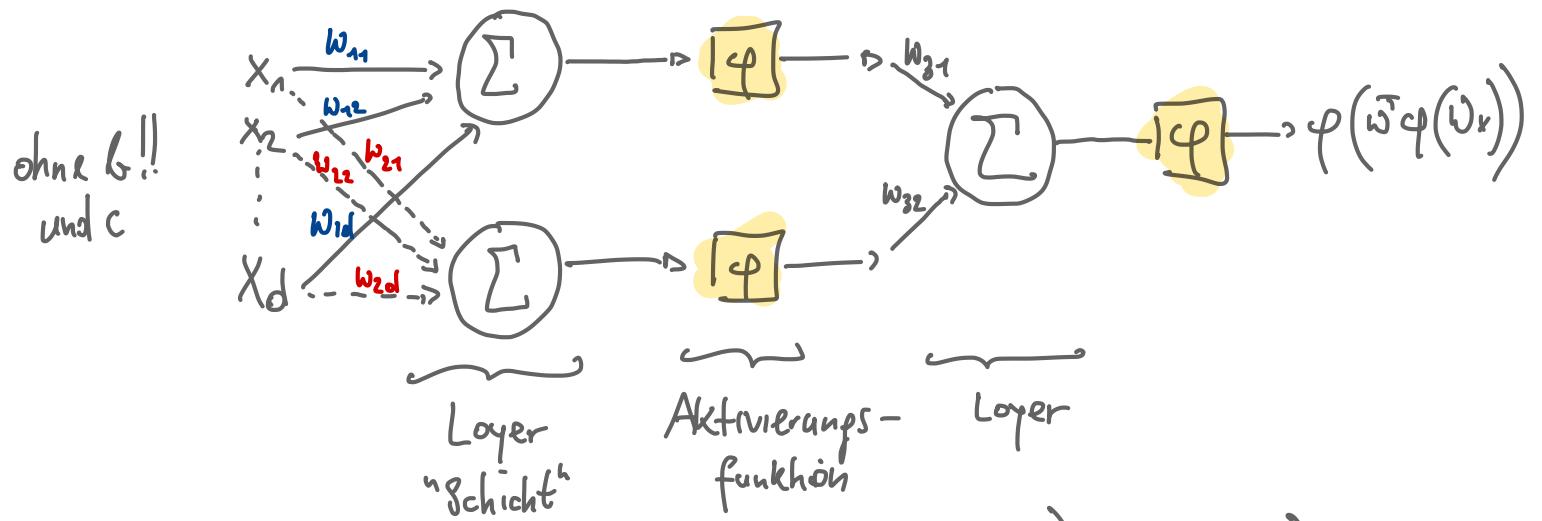
Aber, mit

$$f(x; \theta) = \varphi(w^T \varphi(wx + b) + c)$$

$$\varphi(x) = \begin{cases} 1, & \text{wenn } x \geq 0 \\ 0, & \text{sonst} \end{cases}$$

φ ... wird komponentenweise angewendet

"Multilayer - Perception (MLP)"



Im XOR Beispiel: $W = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, $b = \begin{pmatrix} -1.5 \\ -0.5 \end{pmatrix}$, $w = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, $c = -0.5$

Nehmen wir $x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$$wx = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

$$wx + b = \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \begin{pmatrix} -1.5 \\ -0.5 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix}$$

$$\varphi(wx + b) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

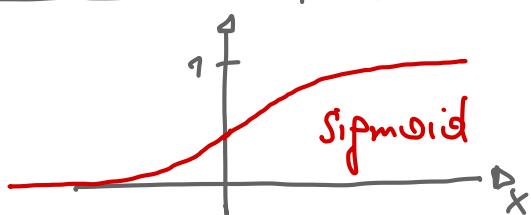
$$w^T \varphi(wx + b) + c = \langle \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \rangle = -0.5$$

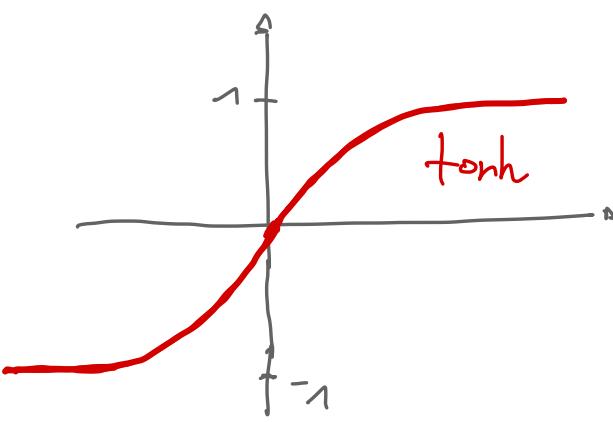
$$\Rightarrow \text{Output für } x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0$$

Anm.: produziert für alle XOR Inputs den korrekten Output.

In weiterer Folge ersetzen wir $\varphi = 1$ durch eine differenzierbare Funktion $\varphi: \mathbb{R} \rightarrow \mathbb{R} \Rightarrow$ d.h. wir können Gradienten ausrechnen.

Beispiele für φ : (Aktivierungsfunktion)





Anderes Beispiel: $x \in \mathbb{R}^n, y \in \{0, 1\}$

$$p(y|x, \theta) = \text{Ber}(y|\theta(\alpha_3))$$

Im Vergleich zur LR:
 $\text{Ber}(y|\theta(w^T x + b))$

MLP

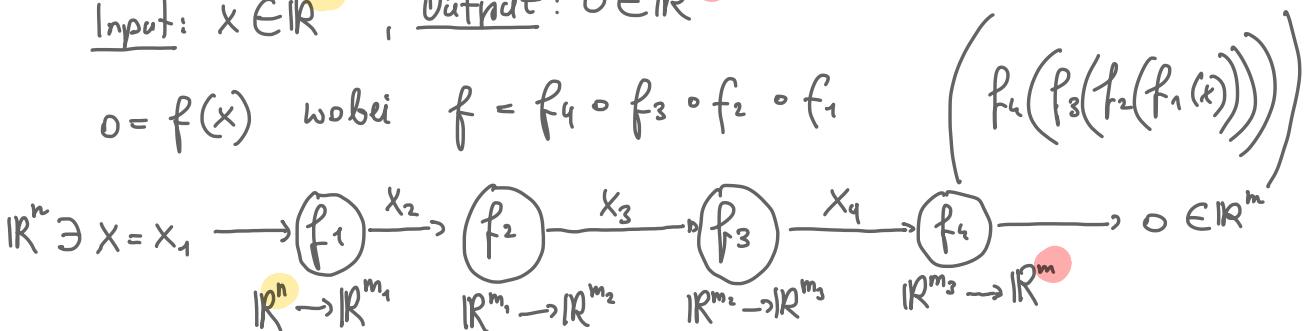
$$\left\{ \begin{array}{l} \alpha_3 = w^T z_2 + b_3 \\ z_2 = \sigma(w_2 z_1 + b_2) \\ z_1 = \sigma(w_1 x + b_1) \end{array} \right. \quad \theta = \{w_1, b_1, w_2, b_2, w, b_3\}$$

sigmoid (könnte auch tanh, o.d. ReLU sein)

Backpropagation:

Input: $x \in \mathbb{R}^n$, Output: $o \in \mathbb{R}^m$

$$o = f(x) \text{ wobei } f = f_4 \circ f_3 \circ f_2 \circ f_1$$



$$\frac{\partial o}{\partial x} = \frac{\partial o}{\partial x_4} \cdot \frac{\partial x_4}{\partial x_3} \cdot \frac{\partial x_3}{\partial x_2} \cdot \frac{\partial x_2}{\partial x_1} \quad // \text{Kettenregel}$$

$$= \underbrace{J_{f_4}(x_4)}_{\text{Jacobi-Matrix}} \cdot \underbrace{J_{f_3}(x_3)}_{-h-} \cdot \underbrace{J_{f_2}(x_2)}_{-i-} \cdot \underbrace{J_{f_1}(x)}_{-j-} \quad // \text{Produkt von Jacobi-Matrizen!}$$

Also ist $\frac{\partial o}{\partial x}$ eine $(m \times n)$ Matrix; bezeichnet als $J_f(x)$

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_m}{\partial x_1} & \dots & \dots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Partielle Ableitung der 1. ten Komponente
des Outputs von f (also o) nach der n -ten
Komponente vom Input abgeleitet.

m

n -Spalten

$$= \begin{pmatrix} \nabla f_1(x)^T \\ \nabla f_2(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{pmatrix}$$

Wollen wir die i-te Teile von J_f :

$$e_i^T J_f(x) = (0, 0, 0, \dots, \underset{i\text{-te Stelle}}{1}, 0, \dots, 0) \cdot \begin{pmatrix} m \times n \\ J_f(x) \end{pmatrix}$$

e_i^T i-te Einheitsvektor

Wollen wir die i-te Spalte von J_f :

$$J_f(x) \cdot e_i \rightarrow \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \rightarrow i\text{-te Stelle}$$

$$e_i^T J_f(x) \dots \text{Vektor-Jacobi-Produkt} \quad (\text{VJP})$$

$$J_f(x) \cdot e_i \dots \text{Jacobi-Vektor-Produkt} \quad (\text{JVP})$$

Fall: $n < m$

\downarrow Input-dim. \swarrow Output-dim.

in dem Fall ist es besser $J_f(x)$ anhand $j=1\dots n$ JVP's zu
rechnen.

Algorithmus ("Forward-Mode" Automatic Differentiation)

$$x_n = x$$

$$v_j = e_j \quad \text{for } j = 1, \dots, m$$

for $k = 1:K$ do

$$x_{k+1} = f_k(x_k) \quad // \text{Forward pass}$$

$$v_j = J f_k(x_k) v_j \quad \text{for } j = 1, \dots, m$$

end

Return $o = x_{K+1}, \underbrace{[J f(x)]_{:,j}}_{\text{j-te Spalte von } J_f \text{ mit } v_j \text{ befüllen}} = v_j \quad \text{for } j = 1, \dots, m$

Algorithmus ("Reverse Mode" Automatic Differentiation)

$$x_n = x$$

for $k = 1:K$ do $\left. \begin{array}{l} x_{k+1} = f_k(x_k) \\ \end{array} \right\} // \text{Forward pass}$

end

$$u_i = e_i \in \mathbb{R}^m \quad \text{for } i = 1, \dots, n \quad // \text{Einheitsvektoren}$$

for $k = K:1$ do $\left. \begin{array}{l} u_i^T = u_i \cdot J_k(x_k) \quad \text{for } i = 1, \dots, n \\ \end{array} \right\} // \text{Backward pass}$

end

Return $o = x_{K+1}, \underbrace{[J f(x)]_{i,:}}_{\text{i-te Zeile von } J_f \text{ mit } u_i^T \text{ befüllen}} = u_i^T \quad \text{for } i = 1, \dots, n$

Numerisches Beispiel

$$f: \mathbb{R}^3 \rightarrow \mathbb{R}^2, \quad \begin{pmatrix} x \\ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \end{pmatrix} \mapsto f(x) = \begin{pmatrix} x_1^2 + x_2 + x_3 \\ x_1 - x_2 + x_3 \end{pmatrix} \quad \begin{matrix} f_1 \\ f_2 \end{matrix}$$

Jacobi-Matrix ist 2×3 Matrix

$$J_f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \end{pmatrix} = \begin{pmatrix} 2x_1 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$$

an Stelle $x = \begin{pmatrix} 2 \\ 3 \\ 7 \end{pmatrix}$ auswerten: also

$$J_f(x) = \begin{pmatrix} 4 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$$

$$\text{Mit } u = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow u^T = (1 \ 0); \text{ also } u^T J_f(x) = (4 \ 1 \ 1)$$

$$\text{Mit } u = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow u^T = (0 \ 1); \text{ also } u^T J_f(x) = (1 \ -1 \ 1)$$

Beispiel (mit Parameter):

Daten: $(x_1, y_1), \dots, (x_n, y_n)$

Parameter

$$\mathcal{L}((x_i, y_i), \theta) = \frac{1}{2} \cdot \|y - W_2 \varphi(W_1 x)\|^2 \quad \theta = \{W_1, W_2\}$$

$$\mathcal{L} = f_4 \circ f_3 \circ f_2 \circ f_1$$

$$z_2 = f_1(x; W_1) = W_1 x \quad L = f_4(z_4, y) = \|y - z_4\|^2$$

$$z_3 = f_2(z_2; \{\}) = \varphi(z_2)$$

$$z_4 = f_3(z_3; W_2) = W_2 z_3$$

\{ ... keine Parameter (z.B. sigmoid)

Seien wir $\Theta_1 = \omega_1$ und $\Theta_2 = \omega_2$

$$\frac{\partial L}{\partial \Theta_2} = \frac{\partial L}{\partial z_4} \cdot \frac{\partial z_4}{\partial \Theta_2}$$

$$\frac{\partial L}{\partial \Theta_1} = \underbrace{\frac{\partial L}{\partial z_4}}_{\text{Jacobi-Matrizen}} \cdot \underbrace{\frac{\partial z_4}{\partial z_3}}_{\text{Jacobi-Matrizen}} \cdot \underbrace{\frac{\partial z_3}{\partial \Theta_1}}_{\text{Jacobi-Matrizen}}$$

Algorithmus ("Reverse-Mode" Automotic Diff. für MLP mit K Schichten/Loyern und skalar-wertigem Output). wie im Beispiel!!!

$$x_1 = x$$

```
[for k=1:k do
   $x_{k+1} = f_k(x_k)$ 
end]
```

$$u_{K+1} = 1 \quad // \text{da } m=1, \text{ also skalarwertiger Output}$$

```
[for k=k:-1:1 do
   $f_k = u_{k+1}^T \cdot \frac{\partial f_k(x_k; \theta_k)}{\partial \theta_k}$ 
   $u_k^T = u_{k+1}^T \cdot \frac{\partial f_k(x_k; \theta_k)}{\partial x_k}$ 
end]
```

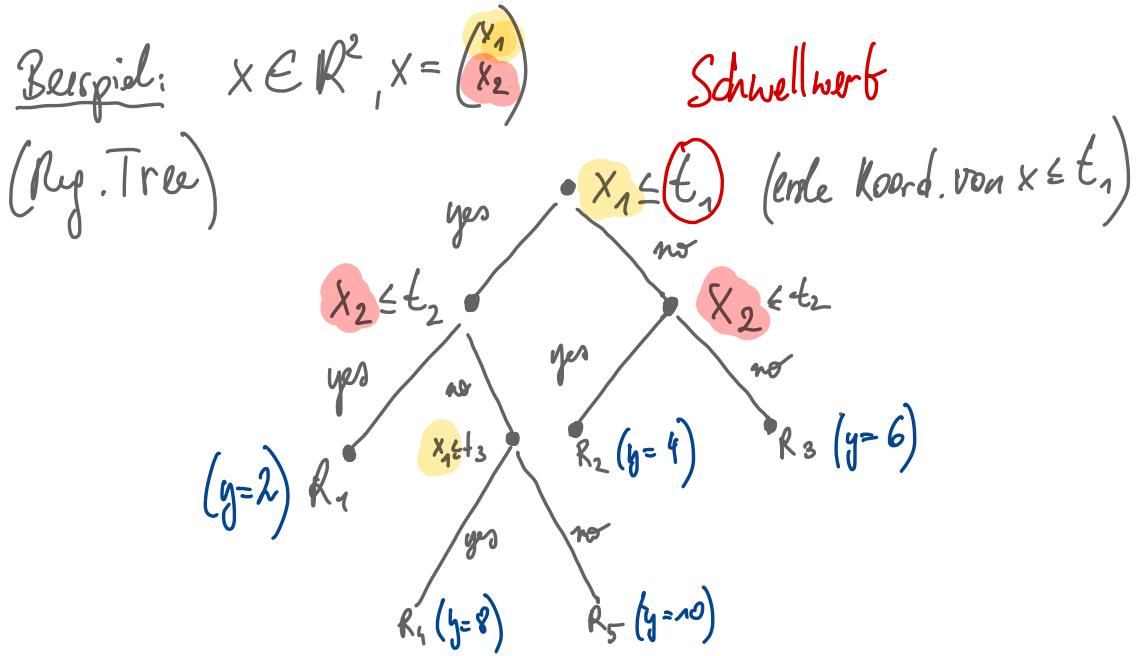
Return $x_{K+1}, \nabla_x L = u_1, \{ \nabla_{\theta_k} L = f_k \text{ for } k=1, \dots, K \}$

(\hookrightarrow nutzen wir in (stochastischem)

Gradientenabstieg

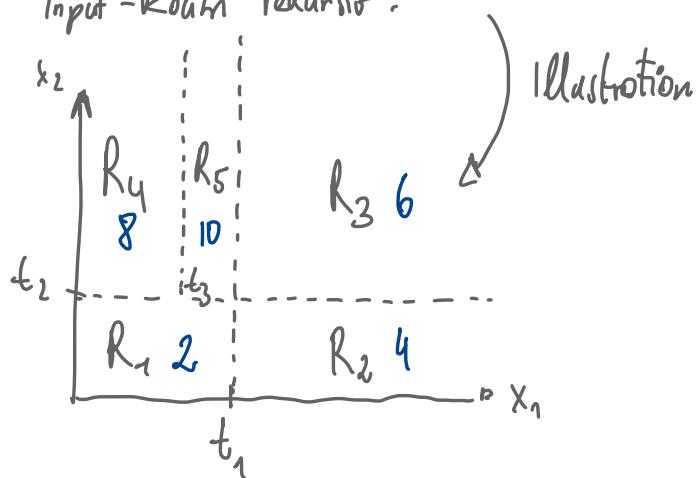
$$\left(\begin{array}{l} \theta_k^{(t+1)} = \theta_k^{(t)} - \eta \cdot \nabla_{\theta_k} L \\ k=1, \dots, K \end{array} \right)$$

TREES, FORESTS & BOOSTING



$$\text{z.B.: } R_1 = \left\{ x \in \mathbb{R}^2 : x_1 \leq t_1 \text{ and } (1) x_2 \leq t_2 \right\}$$

Grundidee: Classification and Reg. Trees partitionieren den Input-Raum rekursiv.



Formel haben wir:

$$f(x; \theta) = \sum_{j=1}^J w_j \cdot \mathbb{1}_{x \in R_j}$$

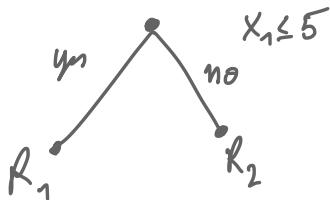
Regionen an Leaf-Nodes
(Blätter)

$$\theta = \left\{ (w_j, R_j) : j=1, \dots, J \right\} \text{ Parameter}$$

$$w_j = \frac{\sum_{n=1}^N y_n \cdot \mathbb{1}_{x_n \in R_j}}{\sum_{n=1}^N \mathbb{1}_{x_n \in R_j}}$$

Vorhergesagter Wert an
Blatt j.

Z.B. mit Daten $D = \{(1,2), 6\}, \{(9,6), 4\}, \{(4,10), 6\}\}$ und

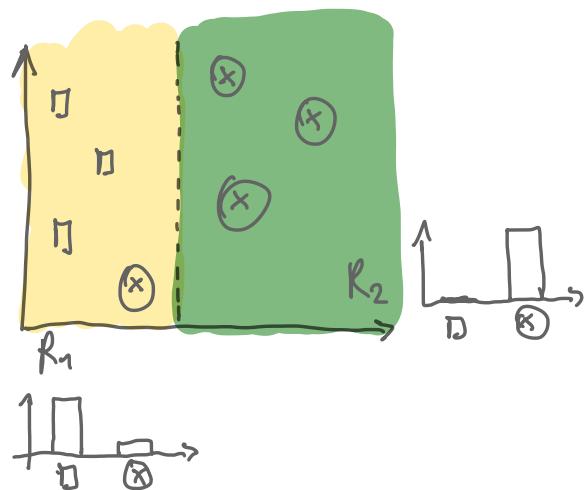


$$w_1 = \frac{6 \cdot \mathbb{1}_{x_1 \in R_1} + 4 \cdot \mathbb{1}_{x_2 \in R_1} + 6 \cdot \mathbb{1}_{x_3 \in R_1}}{\mathbb{1}_{x_1 \in R_1} + \mathbb{1}_{x_2 \in R_1} + \mathbb{1}_{x_3 \in R_1}} = \frac{12}{6} = 2$$

yes no yes

- | f.e. einen Reg. Tree mit Schwellwerten, wissen wir also wie wir zu
- Vorhersagen an den Blättern kommen.

In **Klass.-Problemen** beinhalten die Blatt-Knoten eine Verteilung über die Klassen-Labels (anstatt eines Mittels über die Targets (y)).



Wie finden wir die Parameter Θ ?

$$L(\theta) = \sum_{n=1}^N l(y_n, f(x_n; \theta))$$

nicht differenzierbar

$$= \sum_{j=1}^J \sum_{x_n \in R_j} l(y_n, w_j)$$

(Ann.: Finden einer optimalen Partitionierung des Input-Raums
ist NP-vollständig.)

Ansatz: Greedy-Strategie, also Baum-Knoten für Knoten aufbauen.

Sagen wir wir haben $x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{nd} \end{pmatrix}$ und betrachten Knoten i .
Sei

$$\mathcal{D}_i = \{(x_n, y_n) \in N_i\}$$

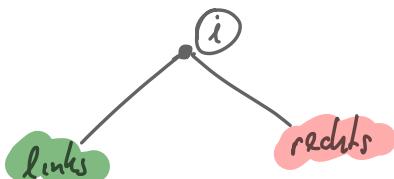
die Menge aller Daten die an Knoten i einkommen.

Fall 1: Merkmal j ist reellwertig.

z.B.: $j=1$ mit Werten

$$\{4.5, -12, 72, -12\} \quad \text{als Nullmenge zu verstehen}$$

Sortiert $T_1 = (-12, 4.5, 72)$. Dies sind alle möglichen Schwellwerte, um zu splitten.



$$\mathcal{D}_i^{links}(j, t) = \{(x_n, y_n) \in N_i, x_{nj} \leq t\}$$

$$\mathcal{D}_i^{rechts}(j, t) = \{(x_n, y_n) \in N_i, x_{nj} > t\}$$

Fall 2: Merkmal j ist **kategorisch**

z.B.: $j = 1$ mit K_1 möglichen Ausprägungen
 $\Rightarrow K_1$ mögliche Split-Werte (+)

$$D_i^{\text{links}}(j, t) = \left\{ (x_n, y_n) \in N_i : x_{nj} = t \right\}$$

$$D_i^{\text{rechts}}(j, t) = \left\{ (x_n, y_n) \in N_i : x_{nj} \neq t \right\}$$

Haben wir nun $D_i^{\text{links}}, D_i^{\text{rechts}}$ für alle möglichen j und t bestimmt
 am Knoten i , wählen wir das "beste" (j, t) wie folgt:

$$(j_{\text{best}}, t_{\text{best}}) = \underset{j \in \{1, \dots, d\}}{\text{argmax}} \underset{t \in T_j}{\text{argmin}} \left[\frac{|D_i^{\text{links}}(j, t)|}{|D_i|} \cdot \text{cost}(D_i^{\text{links}}(j, t)) + \frac{|D_i^{\text{rechts}}(j, t)|}{|D_i|} \cdot \text{cost}(D_i^{\text{rechts}}(j, t)) \right]$$

Wie wählen wir nun $\text{cost}(\cdot, \cdot)$?

Fall 1 (Regression)

$$\text{cost}(D_i) = \sum_{n: (x_n, y_n) \in D_i} (y_n - \bar{y})^2 \quad \text{mit} \quad \bar{y} = \left(\sum_{n: (x_n, y_n) \in D_i} y_n \right) \frac{1}{|D_i|}$$

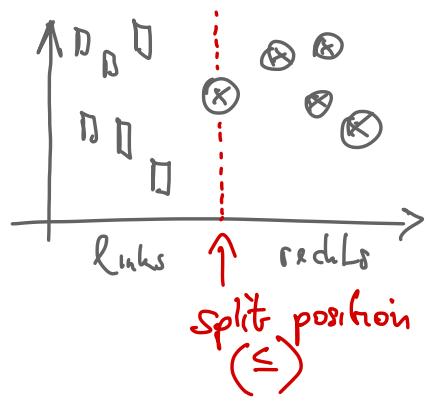
Fall 2 (Klassifikation)

$$\text{Berechnen } \hat{\pi}_{ic} = \frac{1}{|D_i|} \cdot \sum_{n: (x_n, y_n) \in D_i} \underbrace{\frac{1}{C} y_n}_\text{für alle } c \in \{1, \dots, C\} = \underbrace{\frac{1}{C} \sum_{c=1}^C \hat{\pi}_{ic}}_\text{alle Ausprägungen}$$

$$\text{Dann } Q_i = \sum_{c=1}^C \hat{\pi}_{ic} \cdot (1 - \hat{\pi}_{ic}) \quad \left\{ \begin{array}{l} \text{cost}(D_i) = Q_i \\ = 1 - \sum_{c=1}^C (\hat{\pi}_{ic})^2 \end{array} \right.$$

Dies nennt man den **Gini-Index**!

Peni - Beispiel:



hier: $C = 2$ \square and \otimes

im "linken" Teil:

$$\hat{\pi}_{in} = \frac{1}{7} \cdot 6 \quad \hat{\pi}_{in} = \frac{1}{7} \cdot 1$$

$$\Rightarrow \frac{6}{7} \cdot \left(1 - \frac{6}{7}\right) + \frac{1}{7} \cdot \left(1 - \frac{1}{7}\right) \approx 0.24$$

im "rechten" Teil: \circ