

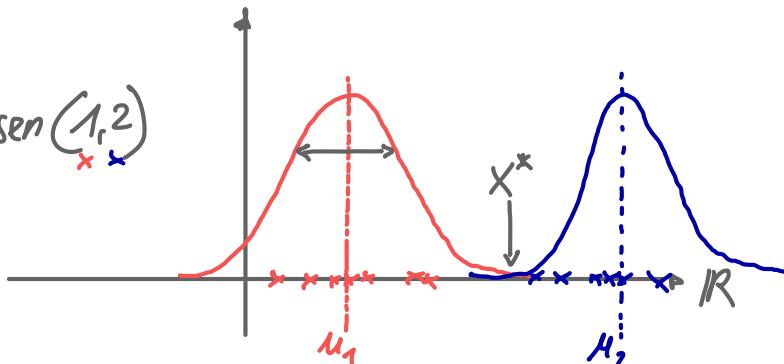
# MACHINE LEARNING

14.10.24

## LDA

$x \in \mathbb{R}$

zwei Klassen (1, 2)



$\bar{F}_1(x), \bar{F}_2(x)$  Verteilungsfunktionen (CDF)

$f_1(x) = \frac{\partial \bar{F}_1(x)}{\partial x}, f_2(x) = \frac{\partial \bar{F}_2(x)}{\partial x}$  Dichtefunktionen (PDF)

Annahme: 1-D Gaußverteilung für beide Klassen mit  $\mu_1 < \mu_2$

$$x \sim \begin{cases} N(\underline{\mu_1}, \sigma_1^2), & \text{wenn } x \text{ aus Klasse 1} \\ N(\underline{\mu_2}, \sigma_2^2), & \text{wenn } x \text{ aus Klasse 2} \end{cases}$$

$$\Pr(\text{error}) = \Pr(x > x^*, x \in \text{Klasse 1}) + \Pr(x < x^*, x \in \text{Klasse 2})$$

$$\Pr(A, B) = \Pr(A|B) \cdot \Pr(B)$$

$$= \Pr(x > x^* | x \in \text{Klasse 1}) \cdot \Pr(x \in \text{Klasse 1}) +$$

$$\Pr(x < x^* | x \in \text{Klasse 2}) \cdot \Pr(x \in \text{Klasse 2})$$

wir wollen

minimiere  $\Pr(\text{error})$   
 $x^*$

## Umschreiben mittels CDFs

$$\bullet \Pr(x < c \mid x \in \text{Klasse 1}) = F_1(c)$$

$$\Rightarrow \Pr(x > x^* \mid x \in \text{Klasse 1}) = 1 - F_1(x^*)$$

$$\bullet \Pr(x < x^* \mid x \in \text{Klasse 2}) = F_2(x^*)$$

Also,  $\Pr(\text{error}) = (1 - F_1(x^*)) \cdot \bar{n}_1 + F_2(x^*) \cdot \bar{n}_2$

a-priori W-keit  $\Pr(x \in \text{Klasse 1})$

Da  $f_1(x) = \frac{dF_1(x)}{dx}$  und  $f_2(x) = \frac{dF_2(x)}{dx}$ , erhalten wir

$$\frac{\Pr(\text{error})}{\Pr(x^*)} = -f_1(x^*) \cdot \bar{n}_1 + f_2(x^*) \cdot \bar{n}_2 = 0 \quad // 0 \text{ setzen}$$

$$\Rightarrow f_1(x^*) \cdot \bar{n}_1 = f_2(x^*) \cdot \bar{n}_2 \quad (\times)$$

Alternativ (mit Bayes):

$$\Pr(x \in \text{Klasse 1} \mid X=x) = \Pr(x \in \text{Klasse 2} \mid X=x)$$

$f_1(x)$  A-posteriori W-keit dass  $x$  zu Klasse 1 gehört!

$$\text{Boyes: } \Pr(X=x \mid x \in \text{Klasse 1}) \cdot \Pr(x \in \text{Klasse 1}) = \Pr(X=x \mid x \in \text{Klasse 2}) \cdot \Pr(x \in \text{Klasse 2})$$

Normalisierung  $\quad \quad$  Normalisierung

gleich

$$\Rightarrow f_1(x) \cdot \bar{n}_1 = f_2(x) \cdot \bar{n}_2$$

Einschub: Multivariate Normalverteilung ;  $x \in \mathbb{R}^d$

$$x \sim N(\mu, \Sigma) \quad \underbrace{\mu \in \mathbb{R}^d}_{\text{Koordinaten}}, \underbrace{\Sigma \in \mathbb{R}^{d \times d}}_{\text{Kovarianzmatrix}}$$

Dichtefunktion:

$$f(x) = \frac{1}{\sqrt{(2\pi)^d \cdot |\Sigma|}} \cdot e^{-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}}$$

Determinante von  $\Sigma$

Wir nehmen an:  $f_1, f_2$  multivariate Normalverteilungen mit  $(\mu_1, \Sigma), (\mu_2, \Sigma)$

$$\frac{1}{\sqrt{(2\pi)^d \cdot |\Sigma|}} \cdot e^{-\frac{(x-\mu_1)^T \Sigma^{-1} (x-\mu_1)}{2}} \cdot \pi_1 = \frac{1}{\sqrt{(2\pi)^d \cdot |\Sigma|}} \cdot e^{-\frac{(x-\mu_2)^T \Sigma^{-1} (x-\mu_2)}{2}} \cdot \pi_2$$

$$\log \Rightarrow -\frac{1}{2} \cdot \underbrace{(x-\mu_1)^T \Sigma^{-1} (x-\mu_1)} + \ln(\pi_1) = -\frac{1}{2} \cdot \underbrace{(x-\mu_2)^T \Sigma^{-1} (x-\mu_2)} + \ln(\pi_2)$$

$$x^T \Sigma^{-1} x - \underbrace{x^T \Sigma^{-1} \mu_1 - \mu_1^T \Sigma^{-1} x}_{-2 \mu_1^T \Sigma^{-1} x} + \mu_1^T \Sigma^{-1} \mu_1$$

$$(x-\mu)^T = x^T - \mu^T$$

$$\Rightarrow -\frac{1}{2} \cdot x^T \Sigma^{-1} x + \mu_1^T \Sigma^{-1} x - \frac{1}{2} \cdot \mu_1^T \Sigma^{-1} \mu_1 + \ln(\pi_1) = -\frac{1}{2} x^T \Sigma^{-1} x + \mu_2^T \Sigma^{-1} x - \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \ln(\pi_2)$$

Zusammenfassen ( $\times 2$ , links  $\rightarrow$  rechts):

$$2 \cdot \ln\left(\frac{\pi_2}{\pi_1}\right) + 2 \cdot \left(\Sigma^{-1}(\mu_2 - \mu_1)\right)^T x + (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) = 0$$

$$\Leftrightarrow \underbrace{2 \cdot \left(\Sigma^{-1}(\mu_2 - \mu_1)\right)^T}_{Q^T} x + \underbrace{(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)}_b + 2 \cdot \ln\left(\frac{\pi_2}{\pi_1}\right) = 0$$

$\Rightarrow$  Lineare Entscheidungsregeln!  
(da von der Form  $Q^T x + b$ )

Entscheidungsregel:

Daf:  $f(x) := 2 \cdot \left( \Sigma^{-1} (\mu_2 - \mu_1) \right)^T x + (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_1 - \mu_2) + 2 \ln \left( \frac{P_2}{P_1} \right)$

$\hat{C}(x)$  =  $\begin{cases} \text{Klasse 1, wenn } f(x) < 0 \\ \text{Klasse 2, wenn } f(x) > 0 \end{cases}$   
Prediction

Einschub: MLE - Bernoulli Verteilung

Zufallsvariable  $y = 0$  TAILS  
 $y = 1$  HEAD



Annahme:  $P(y=1) = \theta$   $(0 \leq \theta \leq 1)$   
 $P(y=0) = 1-\theta$

Wir schreiben  $y \sim \text{Ber}(\theta)$

$\mathcal{D} = \{y_n : n=1, \dots, N\}$   $\hookrightarrow \text{Ber}(y|\theta) = \theta^{\mathbb{1}_{y=1}} \cdot (1-\theta)^{\mathbb{1}_{y=0}}$

$P(\mathcal{D}|\theta) = \prod_{n=1}^N \theta^{\mathbb{1}_{y_n=1}} \cdot (1-\theta)^{\mathbb{1}_{y_n=0}}$

$$\left( \mathbb{1}_{y=1} = \begin{cases} 1, & \text{wenn } y=1 \\ 0, & \text{sonst} \end{cases} \right)$$

$$\ln(p(D|\theta)) = \sum_{n=1}^N \mathbb{1}_{y_n=1} \cdot \ln(\theta) + \mathbb{1}_{y_n=0} \cdot \ln(1-\theta)$$

$$= \underbrace{N_1 \cdot \ln(\theta)}_{\hookrightarrow \text{Häufigkeit von } y=1} + N_0 \cdot \ln(1-\theta)$$

$$\underline{\text{NLL}(\theta)} = -\ln(p(D|\theta)) = -N_1 \cdot \ln(\theta) - N_0 \cdot \ln(1-\theta)$$

Negative Log-Likelihood

$$\frac{\partial \text{NLL}(\theta)}{\partial \theta} = \frac{-N_1}{\theta} + \frac{N_0}{1-\theta} \quad // 0 \text{ setzen}$$

$$\Rightarrow -\frac{N_1}{\theta} + \frac{N_0}{1-\theta} = 0 \quad \Rightarrow \quad 0$$

Beispiel (MLE): Kategoriale Verteilung (categorical distribution)

(z.B.: C-Seiten Würfel: Wir modellieren den Ausgang eines Wurfs als Zufallsvariable  $y_n \in \{1, \dots, C\}$ )

$$y_n \sim \text{Cat}(\theta) \quad 0 \leq \theta_c \leq 1, \sum_{c=1}^C \theta_c = 1$$

$$\Pr(Y=c) = \theta_c$$

Wahrscheinlichkeitsmesse-Funktion:

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

Auf Basis von  $\mathcal{D} = \{y_1, \dots, y_N\}$  würden wir gerne den MLE von  $\theta$  bestimmen.

$$p(\mathcal{D}|\theta) = \prod_{n=1}^N \prod_{c=1}^C \theta_c^{\mathbb{1}_{y_n=c}} \quad | \log$$

$$\ln p(\mathcal{D}|\theta) = \sum_n \sum_c \mathbb{1}_{y_n=c} \cdot \ln(\theta_c)$$

$$\Rightarrow \text{NLL}(\theta) = - \sum_n \sum_c \mathbb{1}_{y_n=c} \cdot \ln(\theta_c)$$

$$= - \sum_c N_c \cdot \ln(\theta_c) \quad \text{mit}$$

$$N_c = \sum_n \mathbb{1}_{y_n=c}$$

Hinweis: Minimieren von  $\text{NLL}(\theta)$  unter Nebenbedingung  $\sum_c \theta_c = 1$ .

Lagrange Multiplizierer (2D); Funktion  $f(x,y)$ , Nebenbedingung  $g(x,y)=0$

1. Lagrangepunkt:  $L(x,y,\lambda) = f(x,y) - \lambda \cdot g(x,y)$

## 2. Wir suchen

$$\nabla_{x,y,d} L(x, y, d) = 0 \quad \text{unter } \nabla f \neq 0$$

Anm.:  $\begin{pmatrix} \frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} - d \cdot \frac{\partial g}{\partial x} \\ \frac{\partial L}{\partial y} = \frac{\partial f}{\partial y} - d \cdot \frac{\partial g}{\partial y} \end{pmatrix} = 0 \quad (\text{also } \nabla f = d \nabla g)$

und  $\frac{\partial L}{\partial d} = 0 \rightarrow \underbrace{g(x, y)}_0 = 0$

unsere Nebenbedingung

## 3. Gleichungssystem lösen

Hinweis: Zu sehen ob die gefundenen Extrema Minima od. Maxima sind, ist nicht immer unmittelbar klar (siehe Optimierungs-Lit.).

Bei uns (im Fall der kategorialen Verteilung):

$$\begin{aligned} L(\theta, d) &= -NLL(\theta) - d \cdot \left(1 - \sum_c \theta_c\right) \\ &= -\sum_c N_c \cdot \ln(\theta_c) - d \cdot \left(1 - \sum_c \theta_c\right) \end{aligned}$$

$$\frac{\partial L(\theta, d)}{\partial \theta_c} = -\frac{N_c}{\theta_c} + d \stackrel{\text{null setzen}}{\Rightarrow} N_c = d \cdot \theta_c \quad (\times)$$

$$\frac{\partial L(\theta, d)}{\partial d} = -\left(1 - \sum_c \theta_c\right) \stackrel{-\sim}{\Rightarrow} \sum_c \theta_c = 1 \quad (\times)$$

Wir wissen  $\sum_c N_c = N \stackrel{(\times)}{\Rightarrow} \underbrace{\sum_c N_c}_N = d \cdot \underbrace{\sum_c \theta_c}_1 = d = N$

Also 
$$\hat{\theta}_c = \frac{N_c}{d} = \frac{N_c}{N}$$

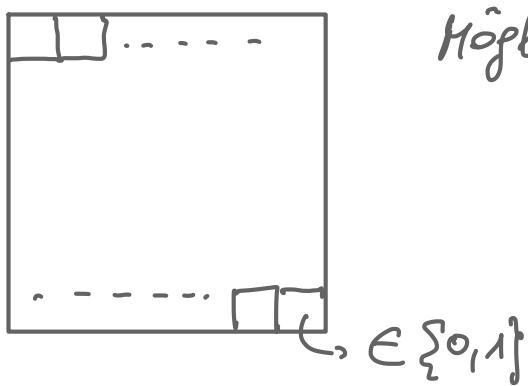
Betrachten wir einen Spezialfall:

$$P(y=c|x, \{\theta_j\}) = \frac{P(x|y=c, \theta) \cdot P(y=c|\pi)}{\text{Normierung}} ; x \in \mathbb{R}^d$$

$\Rightarrow \frac{1}{\prod_{j=1}^d} \rho(x_j | y=c, \theta_{jc})$

Ein solches Modell nennt man NAIVE BAYES Klassifizierer.

Beispiel: Binäre Bilddaten



Möglicher Name: "Bernoulli Naive Bayes"

# LOGISTISCHE REGRESSION (LR)

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

## 1. Binärer Fall (also $y \in \{0, 1\}$ )

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

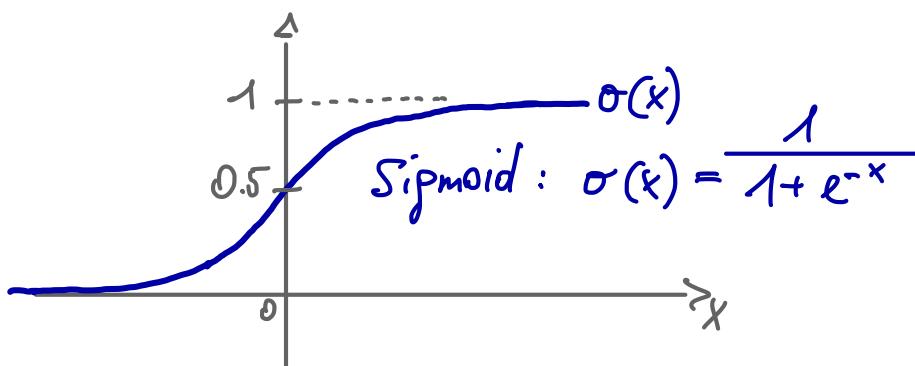
$$\text{Ber}(y | f(x; \alpha))$$

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

Nehmen wir an, dass  $x \in \mathbb{R}^d$  und  $y \in \{0, 1\}$ ; in der LR haben wir folgendes Modell:

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

inkludiert Parameter  $w \in \mathbb{R}^d$  und  $b \in \mathbb{R}$



Setzen wir  $Q = \omega^T x + b$ ; wir haben also konkret

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

$$P(y=0|x, \theta) = 1 - P(y=1|x, \theta)$$

$$= 1 - \frac{1}{1+e^{-a}} = \frac{e^{-a}}{1+e^{-a}} = \frac{1}{1+e^a} = \sigma(-a)$$

Anm.: Setzen wir  $P(y=1|x, \theta) = q$  und betrachten

$$\log \left( \underbrace{\frac{q}{1-q}}_{\text{"odds"}, \text{log-odds}} \right)$$

$$\log \left( \frac{q}{1-q} \right) = \log \left( \frac{e^a}{1+e^a} \cdot \frac{1+e^a}{1} \right) = \log(e^a) = a (= \omega^T x + b)$$

Anmerkung:  $P(y|x, \{w, b\}) = \text{Ber}(y | \theta(w^T x + b))$ ,  $x \in \mathbb{R}^d$ ,  $w \in \mathbb{R}^d$   
 $b \in \mathbb{R}$

Nehmen wir nur

$$\hat{w} = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{pmatrix} \quad \hat{x}, \hat{w} \in \mathbb{R}^{d+1}$$

$$\Rightarrow \langle w, x \rangle + b = \langle \hat{w}, \hat{x} \rangle$$

Im Allgemeinen werden wir immer nur  $\langle w, x \rangle$  ab jetzt schreiben.  
 entweder als  $\in \mathbb{R}^d$ , oder  $\in \mathbb{R}^{d+1}$

$$\begin{aligned} NLL(w) &= -\frac{1}{N} \log P(\mathcal{D}|w) \\ &= -\frac{1}{N} \cdot \log \left( \prod_{n=1}^N \text{Ber}(y_n | \theta(w^T x_n)) \right) \end{aligned}$$

Wir setzen:  $\mu_n = \theta(w^T x_n)$

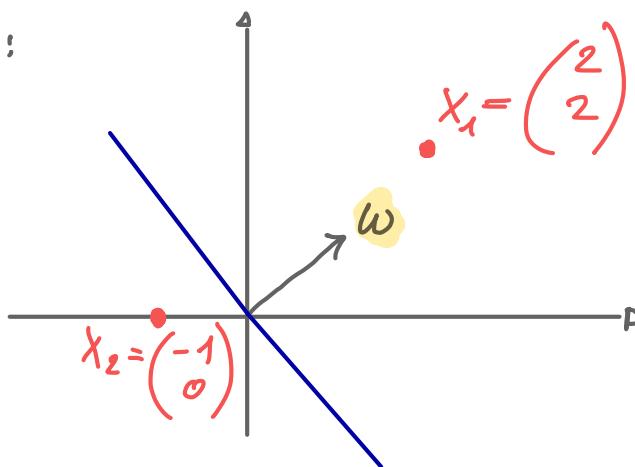
$L_D$

$$= -\frac{1}{N} \cdot \log \left( \prod_{n=1}^N \text{Ber}(y_n | \mu_n) \right) \in [0,1]$$

$$= -\frac{1}{N} \cdot \sum_{n=1}^N \log (\text{Ber}(y_n | \mu_n))$$

Anmerkung:

$$\omega = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



$$\omega^T x_1 = \left\langle \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\rangle = 4$$

$$\omega^T x_2 = \left\langle \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\rangle = -1$$

$$NLL(\omega) = -\frac{1}{N} \cdot \sum_{n=1}^N \log (\mu_n^{y_n} \cdot (1-\mu_n)^{1-y_n})$$

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

$\hookrightarrow \sigma(\omega^T x_n)$

Wie minimieren wir  $NLL(\omega)$ ?

$$\nabla_{\omega} NLL(\omega) = 0$$

$$\nabla_{\omega} NLL(\omega) = \begin{pmatrix} \frac{\partial NLL(\omega)}{\partial \omega_1} \\ \vdots \\ \frac{\partial NLL(\omega)}{\partial \omega_d} \end{pmatrix}$$

Erinnerung:  $q_n = \omega^T x_n$

$$m_n = \sigma(q_n) = \sigma(\omega^T x_n)$$

Wir betrachten:  $\frac{\partial m_n}{\partial \omega_d} = \sigma(q_n) \cdot (1 - \sigma(q_n))$

$$\begin{aligned} \frac{\partial m_n}{\partial \omega_d} &= \frac{\partial}{\partial \omega_d} \sigma(\omega^T x_n) = \underbrace{\frac{\partial}{\partial q_n} \sigma(q_n)}_{= \sigma'(q_n)} \cdot \underbrace{\frac{\partial}{\partial \omega_d} q_n}_{= x_{nd}} \\ &= \underbrace{\sigma(q_n) \cdot (1 - \sigma(q_n))}_{= \sigma'(q_n)} \cdot \underbrace{x_{nd}}_{\text{d-te Koordinate von } x_n} \end{aligned}$$

Wir erhalten also  $\frac{\partial}{\partial \omega_d} m_n = \underbrace{\sigma(q_n)}_{= m_n} \cdot \underbrace{(1 - \sigma(q_n))}_{= 1 - m_n} \cdot x_{nd}$   
 $= m_n \cdot (1 - m_n) \cdot x_{nd}$

Jetzt können wir den Gradienten  $\nabla_{\omega} \log(m_n)$

$$\nabla_{\omega} \log(m_n) = \frac{1}{m_n} \cdot \nabla_{\omega} m_n$$

$$= \begin{pmatrix} \frac{1}{m_n} \cdot \frac{\partial}{\partial \omega_1} m_n \\ \vdots \\ \frac{1}{m_n} \cdot \frac{\partial}{\partial \omega_d} m_n \end{pmatrix}$$



$$\Rightarrow \nabla_w \log(\mu_n) = \frac{1}{\mu_n} \cancel{\mu_n \cdot (1-\mu_n)} \cdot \begin{pmatrix} x_{n1} \\ x_{nd} \end{pmatrix}$$

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

(x)

Auf gleiche Art u. Weise erhalten wir

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

(xx)

Für den Gradienten  $\nabla_w NLL(w)$  folgt

$$\begin{aligned}
 \nabla_w NLL(w) &= -\frac{1}{N} \cdot \sum_{n=1}^N \left[ y_n \cdot \underbrace{(1-\mu_n) \cdot \vec{x}_n}_{(x)} - (1-y_n) \cdot \underbrace{\mu_n \vec{x}_n}_{(xx)} \right] \\
 &= -\frac{1}{N} \cdot \sum_{n=1}^N \left[ y_n \vec{x}_n - y_n \cancel{\mu_n \vec{x}_n} - \mu_n \vec{x}_n + \cancel{y_n \mu_n \vec{x}_n} \right] \\
 &= -\frac{1}{N} \cdot \sum_{n=1}^N \left[ y_n \vec{x}_n - \mu_n \vec{x}_n \right] \\
 &= \frac{1}{N} \cdot \sum_{n=1}^N (\mu_n - y_n) \vec{x}_n
 \end{aligned}$$

$\rightarrow \in \{0, 1\}$  Label

$$\Theta(w^T x_n) = \frac{1}{1+e^{-w^T x_n}}$$

In Matrizenbeschreibung: wir definieren

$$X = \begin{pmatrix} x_1 & \dots & x_d \\ \vdots & & \vdots \\ x_N & \dots & x_{Nd} \end{pmatrix} \in \mathbb{R}^{N \times d}$$

("Design Matrix")

Datenpunkte als Zeilenvektoren

Also,

$$\nabla_{\omega} \text{NLL}(\omega) = X^T (\vec{\mu} - \vec{y}) \frac{1}{N},$$

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

$$\vec{\mu} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix}, \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}$$

Betrachten wir die Hesse Matrix:

$$\underbrace{\nabla_{\omega} \text{NLL}(\omega)}_{\text{mehrere Komponenten}} = \frac{1}{N} \cdot \sum_{n=1}^N (\mu_n - y_n) \vec{x}_n$$

$$\text{j-te Komponente: } \frac{\partial}{\partial \omega_j} \text{NLL}(\omega) = \frac{1}{N} \cdot \sum_{n=1}^N (\mu_n - y_n) \cdot x_{nj}$$

(Ignorieren wir  $\frac{1}{N}$ ):

wissen wir bereits

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

$$z_j = (x_{1j}, \dots, x_{Nj})^T$$

$$z_k = (x_{1k}, \dots, x_{Nk})^T$$

$$\mathcal{B} = \begin{pmatrix} \mu_1(1-\mu_1) & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \mu_N(1-\mu_N) \end{pmatrix} \quad (\text{im Buch } S)$$

Quadratische Form

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

Hesse Matrix

Da  $B = \begin{pmatrix} \mu_1 \cdot (1-\mu_1) & & & \\ & \ddots & & 0 \\ 0 & & \ddots & \mu_N \cdot (1-\mu_N) \end{pmatrix}$

nur positive  
Einträge hat ( $\neq 0$ )

können wir  $\nabla_w^2 NLL(\omega)$  schreiben als

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

$\Rightarrow$  positiv semi-definit (PSD) (siehe Buch)

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

Einschub: Lösen des Minimierungsproblems ( $NLL(\omega)$ ) mittels "Stochastic Gradient Descent" (SGD)

wir hatten  $\nabla_w NLL(\omega) = \frac{1}{N} \sum_{n=1}^N (\mu_n - y_n) \cdot \vec{x}_n$

Gradient descent:  $\omega^{(t+1)} = \omega^{(t)} - \eta_t \cdot \nabla_w NLL(\omega)$  (Lernrate (Schrittweite))

SGD Variante:

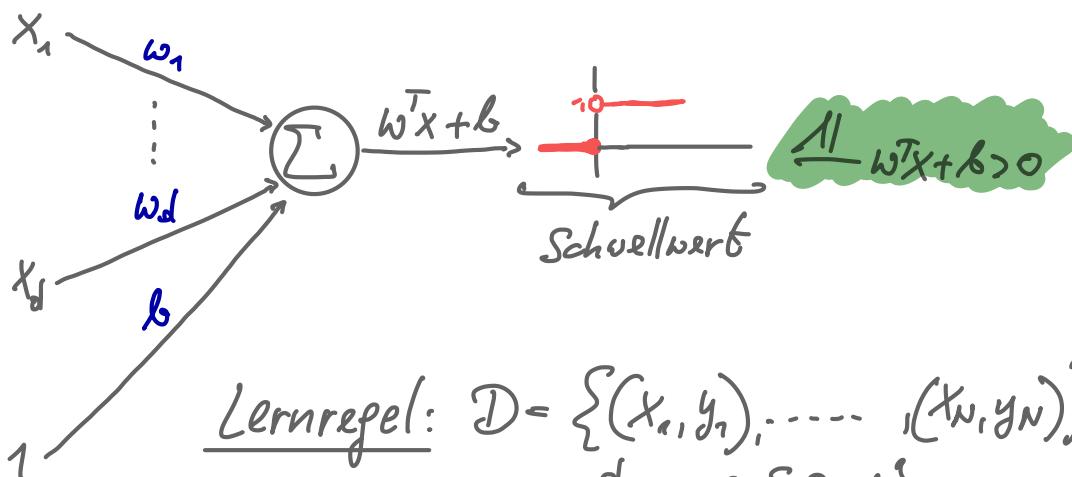
für  $t=0, \dots, T$   
Schritte

$$\tilde{n} \sim \text{Uniform}(\{1, \dots, N\})$$

$$\omega^{(t+1)} = \omega^{(t)} - \eta_t \cdot (\mu_{\tilde{n}} - y_{\tilde{n}}) \cdot \vec{x}_{\tilde{n}}$$

$\Theta(w^T \vec{x}_n + b) \in [0, 1]$

# Einschub: Perceptron (Rosenblatt)



Lernregel:  $\mathcal{D} = \{(x_1, y_1), \dots, (x_N, y_N)\}$ ,  
 $x \in \mathbb{R}^d, y \in \{0, 1\}$

$$\hat{y}_n = \begin{cases} 1, & w^T x_n + b > 0 \\ 0, & \text{sonst} \end{cases}$$

$$\begin{cases} 1, & w^T x + b > 0 \\ 0, & \text{sonst} \end{cases}$$

1.  $y_n = 1, \hat{y}_n = 0$   
 $w^{(t+1)} = w^{(t)} + x_n$

2.  $y_n = 0, \hat{y}_n = 1$   
 $w^{(t+1)} = w^{(t)} - x_n$

3.  $y_n = \hat{y}_n$   
 $w^{(t+1)} = w^{(t)}$

Solangen bis kein Fehler!

Def.:  $\text{err}_n = (\hat{y}_n - y_n)$

$\Rightarrow$  Allg. Updateregel

$$w^{(t+1)} = w^{(t)} - \text{err}_n \cdot x_n$$

$$= w^{(t)} - (\hat{y}_n - y_n) \cdot x_n$$

$\in \{0, 1\}$

$\in \{0, 1\}$

Terminiert nur dann, wenn Punkte linear trennbar!

! ABER, Logistische Regression findet hingegen den Maximum-Likelihood Schätzer auch im nicht linear separierbaren Fall!

## Logistische Regression - Mehrklassen Fall

#klassen

wir versuchen  $p(y|x, \theta)$  zu modellieren (wobei  $y \in \{1, \dots, C\}$ )

Wie vorher:  $x \in \mathbb{R}^D$

$$p(y|x, \{W, b\}) = \text{Cat}\left(y \mid \text{softmax}\left(W^T x + b\right)\right)$$

$W \in \mathbb{R}^{D \times C}$

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

$$a \mapsto \text{softmax}(a) = \left[ \frac{e^{a_1}}{\sum_{c'} e^{a_{c'}}}, \dots, \frac{e^{a_C}}{\sum_{c'} e^{a_{c'}}} \right]^T$$

1.  $0 < \text{softmax}(a)_i < 1$  für alle  $i \in \{1, \dots, C\}$

2.  $\sum_c \text{softmax}(a)_c = 1$

## Maximum-Lik. Schätzung

Zuerst schreiben wir  $\text{Cat}(y|\theta)$  leicht um: wir haben  $y \in \{1, \dots, C\}$ , also ein Integer. Wir können aber auch  $y=c'$  folgendermaßen repräsentieren:

$$[0, 0, 0, \dots, 0, \underset{\text{Position } c'}{1}, 0, \dots, 0] \in \mathbb{R}^C$$

// One-hot encoding

$\hookrightarrow$  Position  $c'$

Wir schreiben:

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

=>

$$\begin{aligned} NLL(w) &= -\frac{1}{N} \log \left( \prod_{n=1}^N \prod_{c=1}^C \alpha_{nc}^{y_{nc}} \right) \quad (\text{ohne } b) \\ &= -\frac{1}{N} \cdot \sum_{n=1}^N \sum_{c=1}^C y_{nc} \cdot \log(\alpha_{nc}) \\ &= \frac{1}{N} \cdot \sum_{n=1}^N H_{CE}(\vec{y}_n, \vec{\alpha}_n) \end{aligned} \quad \text{mit } \alpha_{nc} = \text{softmax}(w^T x_n)_c$$

$\hookrightarrow \text{Cross-Entropy (Kreuzentropie)}$

Wir wollen  $\nabla_w NLL(w) = 0$  nach  $w$  lösen. Wir erhalten

$$\boxed{\nabla_w NLL(w) = \frac{1}{N} \cdot \sum_{n=1}^N \vec{x}_n \cdot (\vec{\alpha}_n - \vec{y}_n)^T \quad \hookrightarrow \in \mathbb{R}^{D \times 1}}$$

Wir können nun  $\nabla_w NLL(w)$  in SGD benutzen und so unser  $w$  updaten!

$$(\vec{\alpha}_n - \vec{y}_n)^T \in \mathbb{R}^{1 \times C}$$