



Queen Mary  
University of London

# LEHRPROBE ZUM THEMA BOOSTING

**DR.-ING. ALEXANDER DOCKHORN**

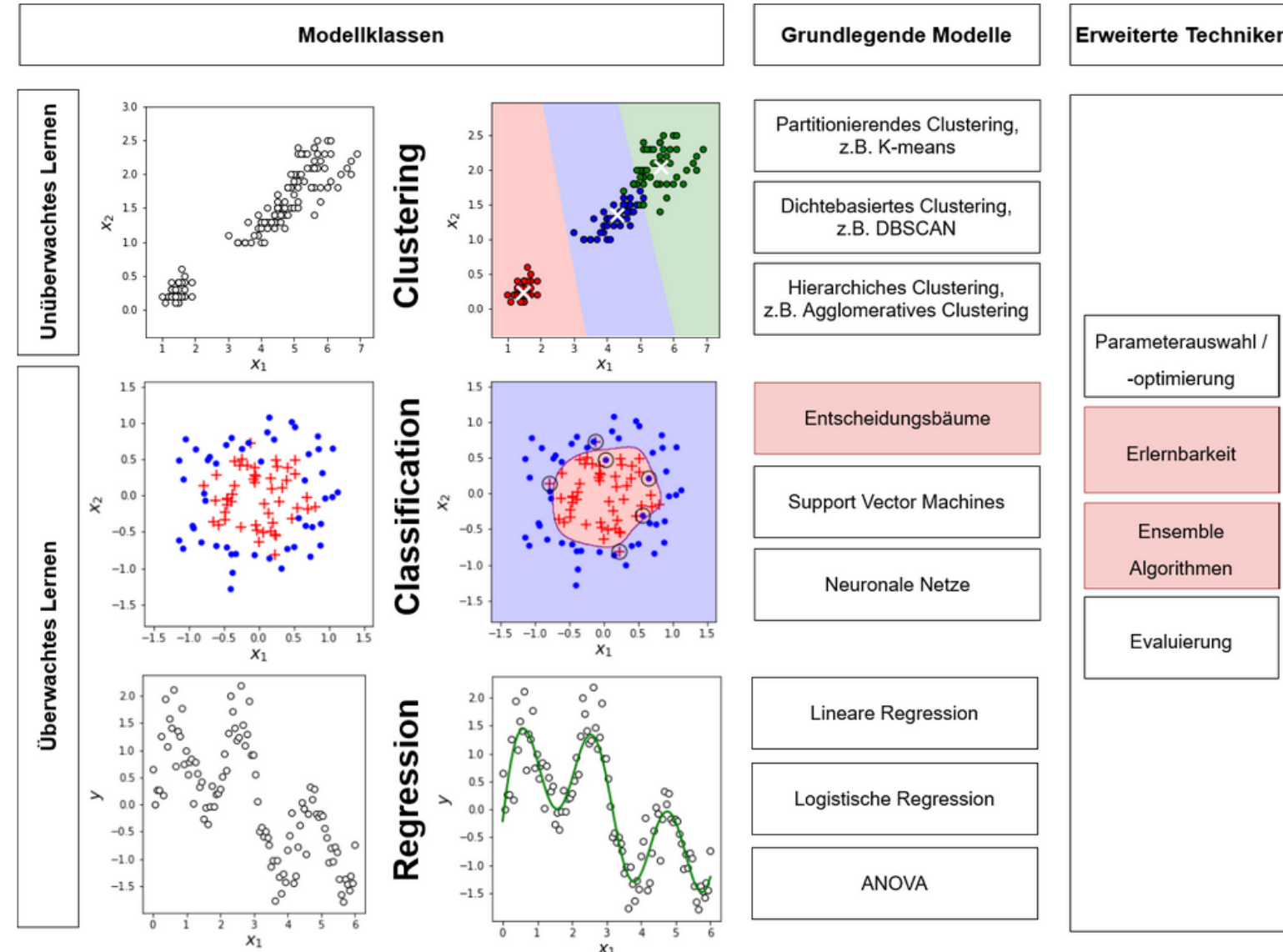
*Postdoctoral Research Associate,  
School of Electronic Engineering and Computer Science,  
Queen Mary University of London*

[a.dockhorn@qmul.ac.uk](mailto:a.dockhorn@qmul.ac.uk)





# EINBETTUNG DIESES VORTRAGS IN EINE LEHRVERANSTALTUNG





## EINBETTUNG DIESES VORTRAGS IN EINE LEHRVERANSTALTUNG





## WELCHER JOKER IST BESSER?



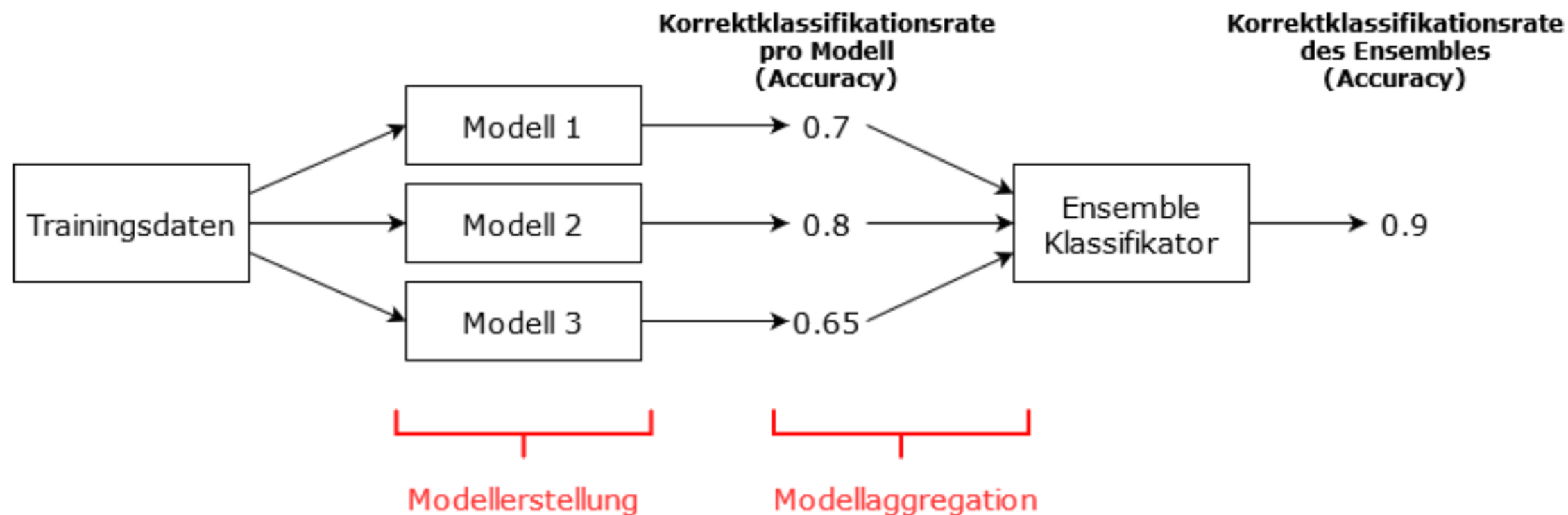
Telefonjoker: 65% richtig

Publikumsjoker: 90% richtig

## GILT DAS AUCH FÜR KLASSIFIKATOREN?

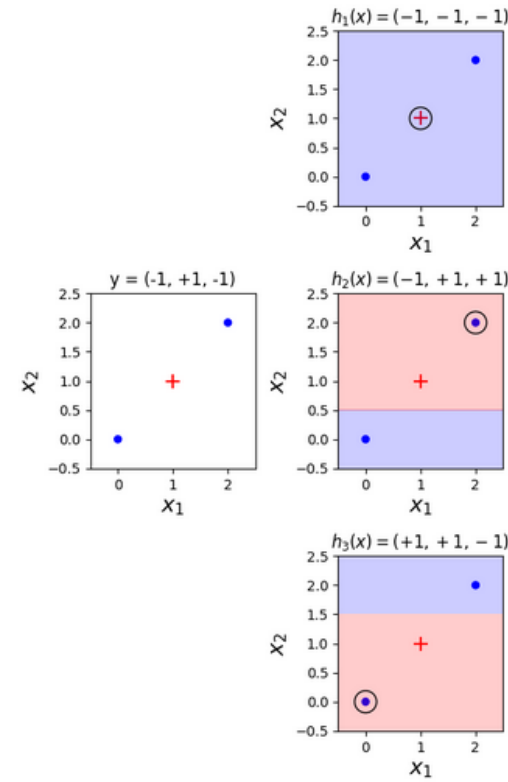


## EINFÜHRUNG IN ENSEMBLE ALGORITHMEN

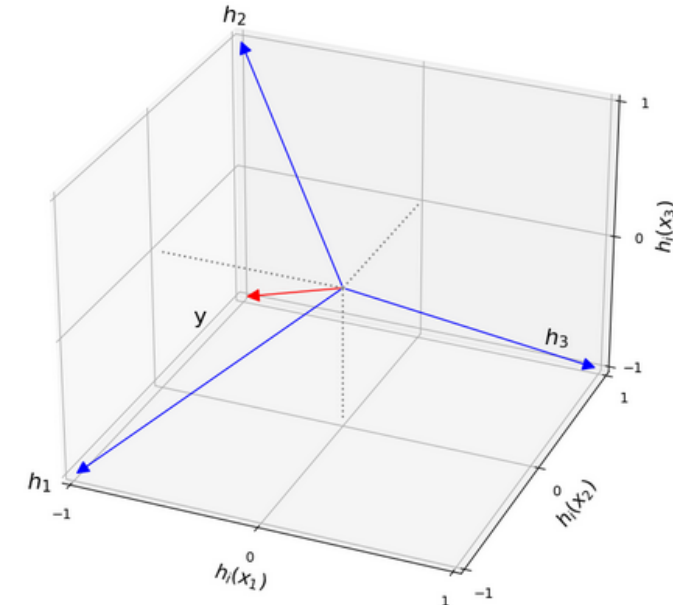




# BOOSTING - EIN ENSEMBLE META-ALGORITHMUS



Klassifikatoren im Funktionsraum



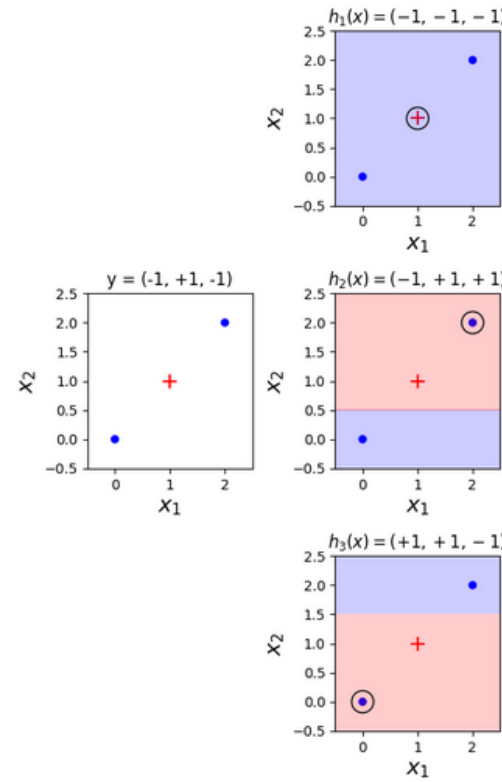


# BOOSTING - EIN ENSEMBLE META-ALGORITHMUS

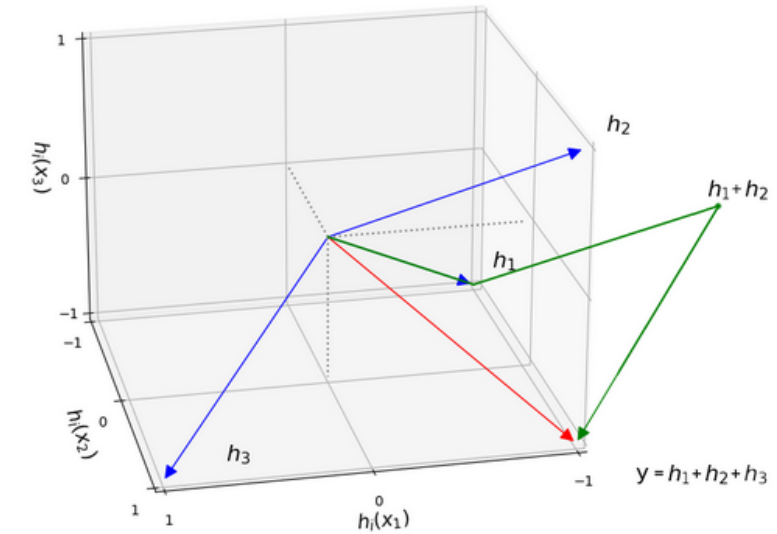
## ENSEMBLE

= AGGREGATION MEHRERER KLASSIFIKATOREN

$$H_t(x) = \sum_{i=1}^t \alpha_i h_i(x)$$



Klassifikatoren im Funktionsraum





# BOOSTING - EIN ENSEMBLE META-ALGORITHMUS

## ENSEMBLE

= AGGREGATION MEHRERER KLASSIFIKATOREN

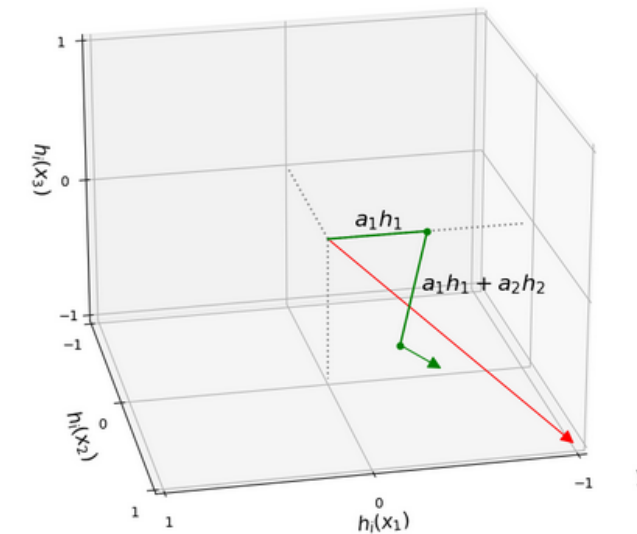
$$H_t(x) = \sum_{i=1}^t \alpha_i h_i(x)$$

## BOOSTING

= ITERATIVE ENSEMBLE ALGORITHMEN

⇒ Gradientenabstieg im Funktionsraum:

Klassifikatoren im Funktionsraum







# BOOSTING - EIN ENSEMBLE META-ALGORITHMUS

## ENSEMBLE

= AGGREGATION MEHRERER KLASSIFIKATOREN

$$H_t(x) = \sum_{i=1}^t \alpha_i h_i(x)$$

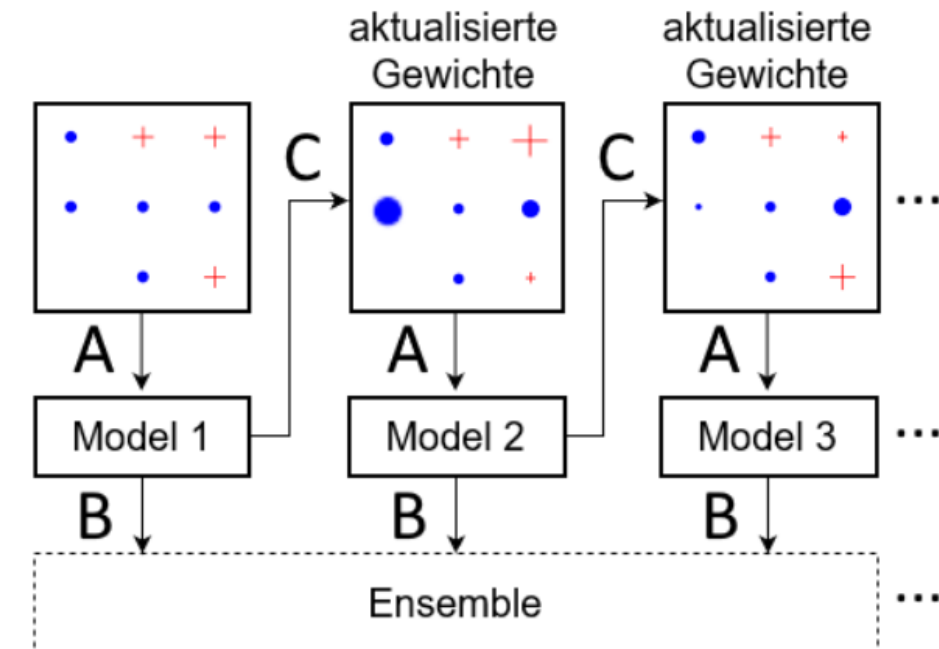
## BOOSTING

= ITERATIVE ENSEMBLE ALGORITHMEN

⇒ **Gradientenabstieg im Funktionsraum:**

- A) Wahl des optimalen Klassifikators
- B) Wahl des optimalen Klassifikatorgewichts
- C) Anpassung der Trainingsdatengewichte

## PRINZIP VON BOOSTING ALGORITHMEN





## ADAPTIVE BOOSTING (ADABOOST)

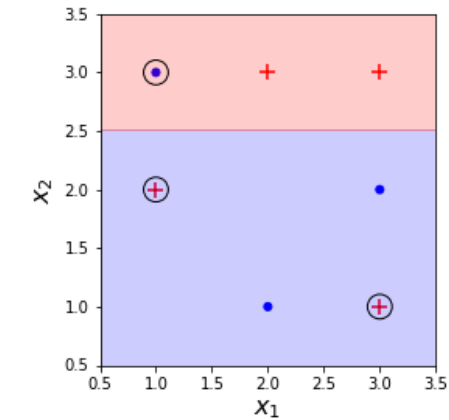
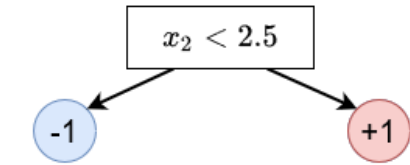
- **Basisklassifikator:** Entscheidungsstümpfe
- Zu jedem Zeitpunkt  $t$  wird dem Ensemble  $H$  ein Entscheidungsstumpf  $h_t$  hinzugefügt.

- Wähle den Klassifikator, welcher den Gesamtfehler bestmöglich reduziert:

$$h_{t+1} = \underset{h_{t+1} \in \mathbb{H}}{\operatorname{argmin}} \mathcal{L}(H_{t+1}) = \underset{h_{t+1} \in \mathbb{H}}{\operatorname{argmin}} \mathcal{L}(H_t + \alpha_{t+1} h_{t+1})$$

- exponentieller Fehler über die gewichteten Trainingsdaten  $D$ :

$$\mathcal{L}(H_t | y) = \sum_i^{|D|} w_i^t e^{-\frac{1}{2} y_i H_t(x_i)}$$





## A) WAHL DES OPTIMALEN KLASSIFIKATORS

$$\mathcal{L}(H_{t+1} | y) = \sum_i^{|D|} w_i^t e^{-\frac{1}{2} y_i \alpha_{t+1} h_{t+1}(x_i)}$$

$$\left| \sum_{\text{alle}} = \sum_{\text{falsch}} + \sum_{\text{richtig}} \right.$$

$$= \underbrace{\sum_{i: h_{t+1}(x_i) \neq y_i}^{|D|} w_i^t e^{-\frac{1}{2} y_i \alpha_{t+1} h_{t+1}(x_i)}}_{\text{falsch klassifizierte Punkte}} + \underbrace{\sum_{i: h_{t+1}(x_i) = y_i}^{|D|} w_i^t e^{-\frac{1}{2} y_i \alpha_{t+1} h_{t+1}(x_i)}}_{\text{richtig klassifizierte Punkte}}$$

$$\left| y_i, h(x_i) \in \{-1, +1\} \right.$$

$$= \sum_{i: h_{t+1}(x_i) \neq y_i}^{|D|} w_i^t e^{\frac{\alpha_{t+1}}{2}} + \sum_{i: h_{t+1}(x_i) = y_i}^{|D|} w_i^t e^{-\frac{\alpha_{t+1}}{2}}$$

$$\left| I(x \neq y) = \begin{cases} 1 & , x \neq y \\ 0 & , \text{sonst} \end{cases} \right.$$

$$= e^{\frac{\alpha_{t+1}}{2}} \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) + \underbrace{e^{-\frac{\alpha_{t+1}}{2}} \sum_i^{|D|} w_i^t}_{\text{alle Punkte}} - \underbrace{e^{-\frac{\alpha_{t+1}}{2}} \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)}_{\text{falsch klassifizierte Punkte}}$$

$$\left| \sum_{\text{richtig}} = \sum_{\text{alle}} - \sum_{\text{falsch}} \right.$$

$$= (e^{\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}}) \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) + e^{-\frac{\alpha_{t+1}}{2}} \sum_i^{|D|} w_i^t$$

**Resultat:** wähle den Klassifikator zur Minimierung des gewichteten Fehlers



## B) WAHL DES OPTIMALEN KLASSIFIKATORGEWICHTS

$$\mathcal{L}(H_{t+1}|y) = (e^{\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}}) \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) - e^{-\frac{\alpha_{t+1}}{2}} \underbrace{\sum_i^{|D|} w_i^t}_{=1}$$

$$\frac{d\mathcal{L}(H_{t+1}|y)}{d\alpha} = \frac{1}{2}(e^{\frac{\alpha_{t+1}}{2}} + e^{-\frac{\alpha_{t+1}}{2}}) \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i) - \frac{1}{2}e^{-\frac{\alpha_{t+1}}{2}} \quad \Bigg| \quad \varepsilon = \sum_i^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)$$

$$e^{\frac{\alpha_{t+1}}{2}} \varepsilon + e^{-\frac{\alpha_{t+1}}{2}} \varepsilon - e^{-\frac{\alpha_{t+1}}{2}} = 0 \quad \Bigg| \quad - (e^{-\frac{\alpha_{t+1}}{2}} \varepsilon - e^{-\frac{\alpha_{t+1}}{2}})$$

$$e^{\frac{\alpha_{t+1}}{2}} \varepsilon = e^{-\frac{\alpha_{t+1}}{2}} - e^{-\frac{\alpha_{t+1}}{2}} \varepsilon$$

$$e^{\frac{\alpha_{t+1}}{2}} \varepsilon = e^{-\frac{\alpha_{t+1}}{2}} (1 - \varepsilon) \quad \Bigg| \quad \ln; \quad \ln(xy) = \ln(x) + \ln(y)$$

$$\frac{\alpha_{t+1}}{2} + \ln(\varepsilon) = -\frac{\alpha_{t+1}}{2} + \ln(1 - \varepsilon) \quad \Bigg| \quad + \frac{\alpha_{t+1}}{2} - \ln \varepsilon$$

$$\alpha_{t+1} = \ln(1 - \varepsilon) - \ln(\varepsilon) \quad \Bigg| \quad \ln(x) - \ln(y) = \ln\left(\frac{x}{y}\right)$$

$$\alpha_{t+1} = \ln\left(\frac{1 - \varepsilon}{\varepsilon}\right)$$

**Resultat:** die Schrittweite ist abhängig von dem gewichteten Fehler des Klassifikators



## AGGREGATION GELEARNTER KLASSIFIKATOREN

Ausgabe des Ensembles:  $H_t(x) = \sum_{i=1}^t \alpha_i h_i(x)$ ;      Ausgabe als Label:  $H_t(x) = \text{sign}\left(\sum_{i=1}^t \alpha_i h_i(x)\right)$

	$h_1(x); \alpha_1 = 1.1$			$h_2(x); \alpha_2 = 1.6$			$h_3(x); \alpha_3 = 1.1$			$h_4(x); \alpha_4 = 1.2$		
Klassifikator $h_t$	-1.1	-1.1	1.1	1.6	1.6	1.6	-1.1	1.1	1.1	-1.2	-1.2	-1.2
	-1.1	-1.1	1.1	-1.6	-1.6	-1.6	-1.1	1.1	1.1	-1.2	-1.2	-1.2
	-1.1	-1.1	1.1	-1.6	-1.6	-1.6	-1.1	1.1	1.1	1.2	1.2	1.2
Ensemble $H_t$	-1.1	-1.1	1.1	0.5	0.5	2.7	-0.6	1.6	3.8	-1.8	0.4	2.6
	-1.1	-1.1	1.1	-2.7	-2.7	-0.5	-3.8	-1.6	0.6	-5.0	-2.8	-0.6
	-1.1	-1.1	1.1	-2.7	-2.7	-0.5	-3.8	-1.6	0.6	-2.6	-0.4	1.8



## C) ANPASSUNG DER TRAININGSDATENGEWICHTE

$$\begin{aligned}\mathcal{L}(H_{t+1}|y) &= \sum_i^{|D|} e^{-\frac{1}{2}y_i H_{t+1}(x_i)} & | \quad H_{t+1}(x_i) &= \textcolor{red}{H_t(x_i)} + \alpha_{t+1} \textcolor{blue}{h_{t+1}(x_i)} \\ &= \sum_i^{|D|} e^{-\frac{1}{2}y_i \textcolor{red}{H_t(x_i)} - \frac{1}{2}y_i \alpha_{t+1} \textcolor{blue}{h_{t+1}(x_i)}} & | \quad e^{a+b} &= e^a e^b \\ &= \sum_i^{|D|} \textcolor{red}{e^{-\frac{1}{2}y_i H_t(x_i)}} \textcolor{blue}{e^{-\frac{1}{2}y_i \alpha_{t+1} h_{t+1}(x_i)}} & | \quad w_i^t &= e^{-\frac{1}{2}y_i H_t(x_i)} \\ &= \sum_i^{|D|} \textcolor{red}{w_i^t} \textcolor{blue}{e^{-\frac{1}{2}y_i \alpha_{t+1} h_{t+1}(x_i)}} & | \quad \Rightarrow w_i^0 &= \frac{1}{|D|}; \quad w_i^{t+1} = w_i^t \cdot e^{-\frac{1}{2}\alpha_t y_i h_t(x_i)}\end{aligned}$$

**Resultat:** das Gewicht eines Datenpunktes ist proportional zum Fehler des aktuellen Ensembles



## FORMELVERZEICHNIS ADABOOST

Initialisieren der Gewichte

$$w_i^0 = \frac{1}{|D|}$$

Lernen einen Klassifikator zur Minimierung des Fehlers

$$\operatorname{argmin}_{h_{t+1} \in \mathbb{H}} \mathcal{L}(H_{t+1}) = \operatorname{argmin}_{h_{t+1} \in \mathbb{H}} \sum_{i=1}^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)$$

Bestimmung der Gewichtung des Klassifikators

$$a_t = \ln \left( \frac{1 - \varepsilon}{\varepsilon} \right); \quad \varepsilon = \sum_{i=1}^{|D|} w_i^t I(h_{t+1}(x_i) \neq y_i)$$

Aktualisieren und Normalisieren der Gewichte

$$w_i^{t+1} = w_i^t \cdot e^{-\frac{1}{2} \alpha_t y_i h_t(x_i)} \quad \sum_{i=1}^n w_i^{t+1} = 1$$

Klassifiziere Punkte anhand des Ensembles

$$H_t(x) = \operatorname{sign} \left( \sum_{i=1}^t \alpha_i h_i(x) \right)$$



## ADABOOST BEISPIEL

Initialisierung der Gewichte:

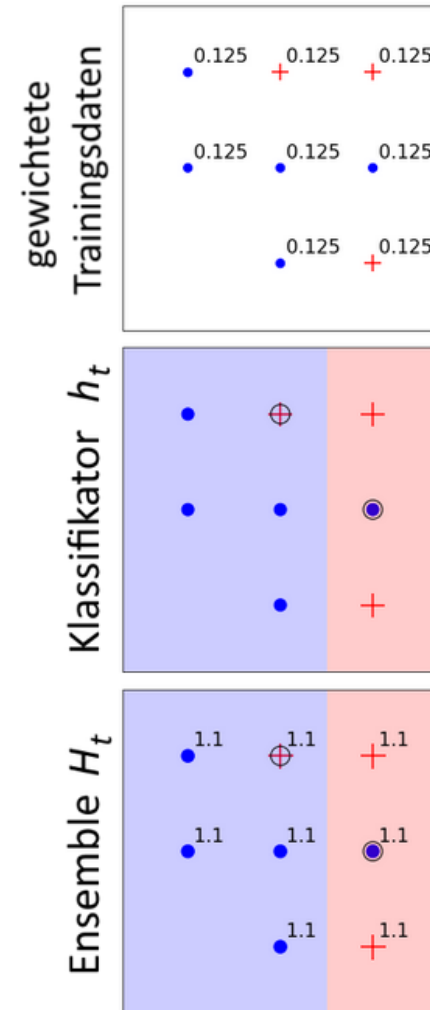
$$w_i^0 = \frac{1}{n} = \frac{1}{8} = 0.125$$

Fehler des Klassifikators:

$$\varepsilon_1 = \sum_{i=1}^{|D|} w_i^0 I[h_1(x_i) \neq y_i] = 0.25$$

Schrittweite:

$$a_1 = \ln\left(\frac{1 - \varepsilon_1}{\varepsilon_1}\right) = \ln\left(\frac{1 - 0.25}{0.25}\right) = \ln(3) \approx 1.1$$







## ADABOOST BEISPIEL

Gewichtsänderung:

$$w_i^1 = w_i^0 \cdot e^{-\frac{1}{2} \alpha_1 y_i h_1(x_i)}$$

korrekt klassifiziert:

$$w_i^1 = w_i^0 \cdot e^{-\frac{1}{2} \alpha_1} = 0.125 \cdot e^{-0.55} \approx 0.072$$

falsch klassifiziert:

$$w_i^1 = w_i^0 \cdot e^{\frac{1}{2} \alpha_1} = 0.125 \cdot e^{+0.55} \approx 0.217$$

Normalisierungsfaktor:

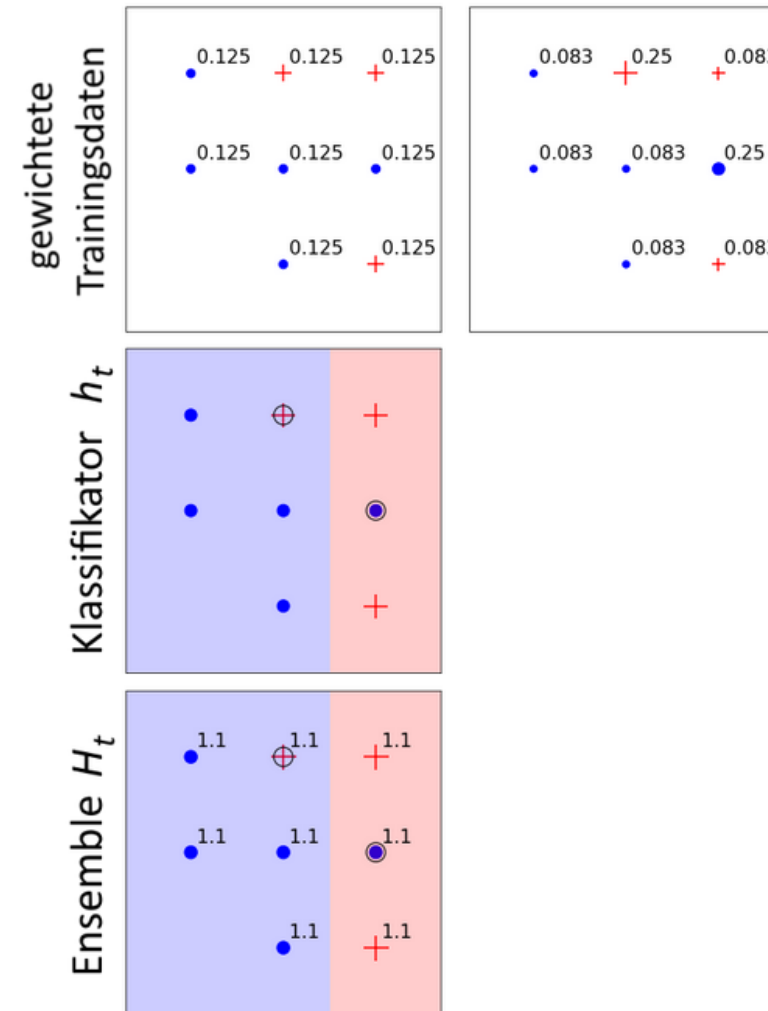
$$\sum_{i=1}^{|D|} w_i^1 = 0.217 \cdot 2 + 0.072 \cdot 6 = 0.866$$

korrekt klassifiziert (normalisiertes Gewicht):

$$w_i'^1 = \frac{0.072}{0.866} \approx 0.083$$

falsch klassifiziert (normalisiertes Gewicht):

$$w_i'^1 = \frac{0.217}{0.866} \approx 0.25$$





## ADABOOST BEISPIEL

Gewichtsänderung:

$$w_i^1 = w_i^0 \cdot e^{-\frac{1}{2} \alpha_1 y_i h_1(x_i)}$$

korrekt klassifiziert:

$$w_i^1 = w_i^0 \cdot e^{-\frac{1}{2} \alpha_1} = 0.125 \cdot e^{-0.55} \approx 0.072$$

falsch klassifiziert:

$$w_i^1 = w_i^0 \cdot e^{\frac{1}{2} \alpha_1} = 0.125 \cdot e^{+0.55} \approx 0.217$$

Normalisierungsfaktor:

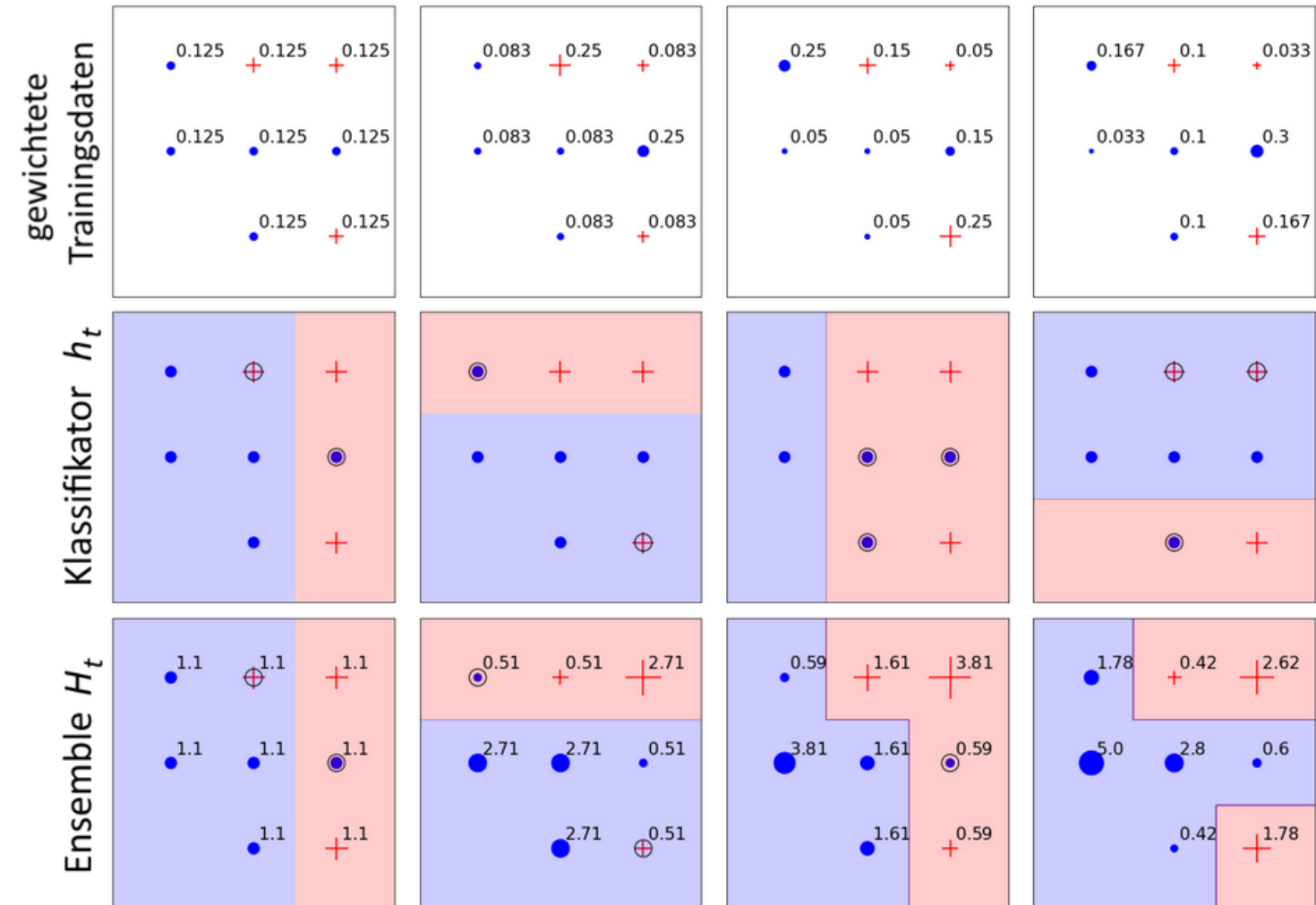
$$\sum_{i=1}^{|D|} w_i^1 = 0.217 \cdot 2 + 0.072 \cdot 6 = 0.866$$

korrekt klassifiziert (normalisiertes Gewicht):

$$w_i'^1 = \frac{0.072}{0.866} \approx 0.083$$

falsch klassifiziert (normalisiertes Gewicht):

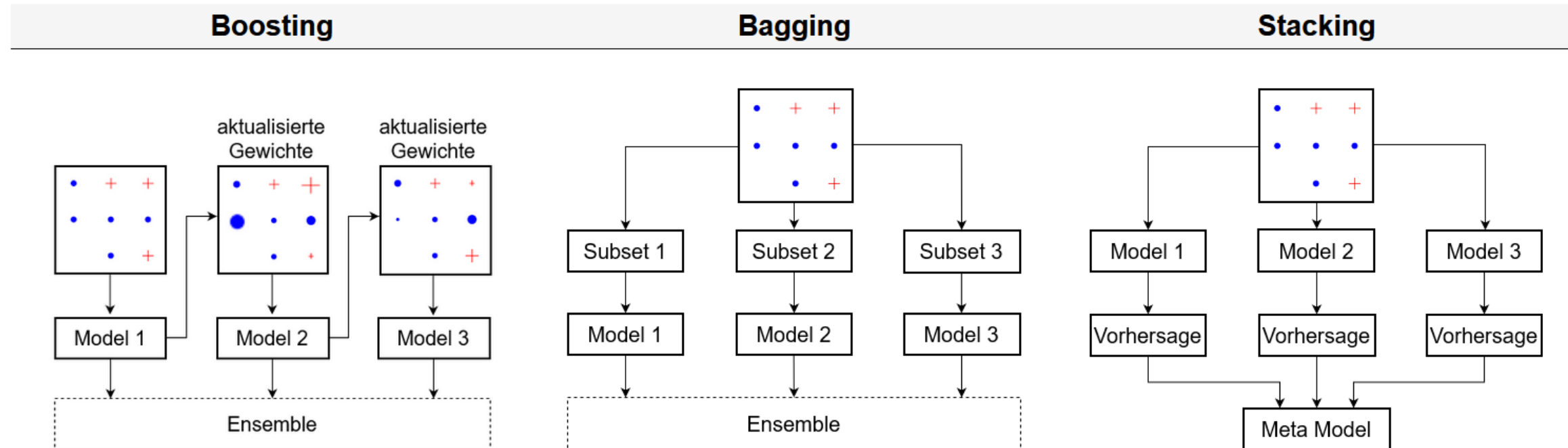
$$w_i'^1 = \frac{0.217}{0.866} \approx 0.25$$





## AUSBLICK

- weitere Varianten von Boosting, z.B. Gradient Boosting
- Vergleich mit alternativen Ensemble Verfahren:





## QUELLENVERZEICHNIS

- Surowiecki, J. (2004). The wisdom of crowds. Why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations. Abacus
  - das vorgestellte Beispiel und zahlreiche weitere Vergleiche von Gruppen- und Expertenentscheidungen
- Kearns, M. and Valiant, L.G. (1989). Cryptographic limitations on learning Boolean formulae and finite automata. Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (pp. 433-444). New York, NY: ACM Press.
  - <https://www.cis.upenn.edu/~mkearns/papers/cryptojacm.pdf>
  - Sind Weak und Strong Learnability unterschiedliche Klassen?
- Schapire, R. E. (1990). \*The strength of weak learnability\*. Machine learning, 5(2), 197-227.
  - <https://link.springer.com/content/pdf/10.1007/BF00116037.pdf>
  - erste Nachweis von effektiven Ensembles aus "Weak learnern"
- Freund, Y., & Schapire, R. E. (1997). \*A decision-theoretic generalization of on-line learning and an application to boosting\*. Journal of computer and system sciences, 55(1), 119-139.
  - [https://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic\\_generalization.pdf](https://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf)
  - Vorstellung des AdaBoost Algorithmus, Gewichtung eines Weak Learners basierend auf seiner Korrektheit
- Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean (2000); Boosting Algorithms as Gradient Descent, in S. A. Solla, T. K. Leen, and K.-R. Muller, editors, Advances in Neural Information Processing Systems 12, pp. 512-518, MIT Press
  - <https://papers.nips.cc/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf>
  - Boosting als Gradientenabstieg im Funktionsraum



## LERNMATERIALIEN

Weitere Materialien zu diesem Vortrag gibt es auf <https://adockhorn.github.io/Boosting/>:

- interaktive Inhalte zum Thema des Fortrags
- eine Formelübersicht
- Links und Quellen zu diesen und verwandten Themen

