9. Klassifikatorunabhängige Prinzipien

9.1 Lernen

Begriffe:

D: Lernstichprobe

T: Teststichprobe

S: Gesamtstichprobe $\mathrm{S} = \mathrm{D} \cup \mathrm{T}$ mit $\mathrm{D} \cap \mathrm{T} = \varnothing$

Lernen

Bestimmung einer Funktion $\hat{\omega}\left[\mathbf{m} \mid \mathrm{D}\right] = \hat{\omega}\left(k\left(\mathbf{m} \mid \mathrm{D}\right)\right)$ auf der Basis der Stichprobe D, die die wahre Klassenzugehörigkeit des Merkmalsvektors $\mathbf{m}$ möglichst genau schätzt.

Überwachtes Lernen

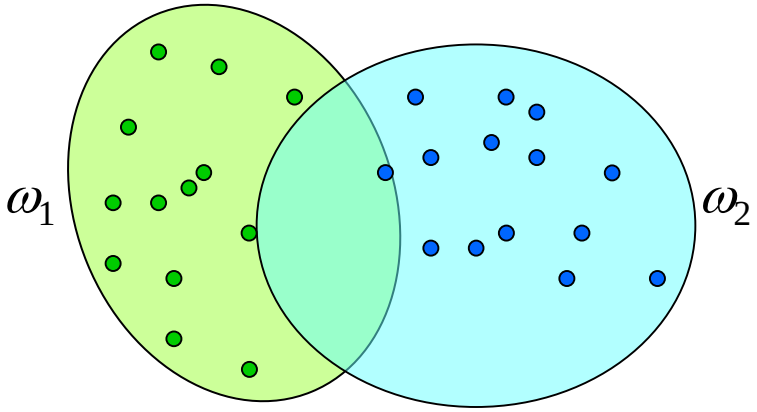
Lernen anhand von Beispielen bekannter Klassenzugehörigkeit

Unüberwachtes Lernen

- Lernen anhand von Beispielen unbekannter Klassenzugehörigkeit
- Wunsch dabei auch: Zugrunde liegenden Prozess verstehen

Probleme:

- Der Umfang der Lernstichprobe ist zu gering.
- Die Lernstichprobe ist nicht repräsentativ.



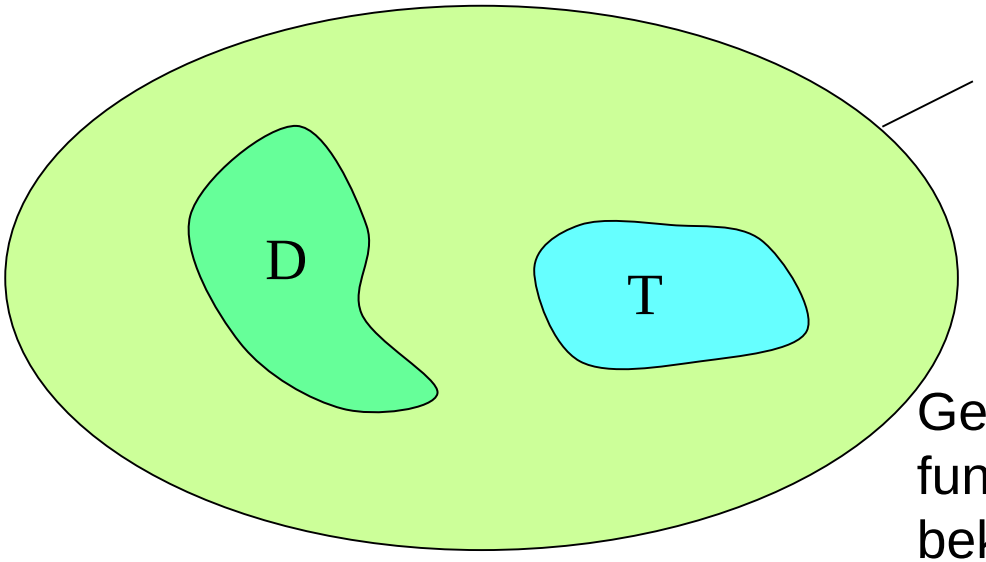
Bsp. für eine nicht repräsentative Lernstichprobe D

- Ein kleiner Lernfehler (Fehler auf der Lernstichprobe) ist noch kein Garant für einen kleinen Testfehler (Fehler auf der Teststichprobe); Overfitting-Problematik.

Im Folgenden sei der Fall von zwei Klassen, also $c = 2$ angenommen. Für die Klassifikation reicht dann eine einzige Entscheidungsfunktion $k(\mathbf{m}, \boldsymbol{\theta})$.

Zentrales Problem der statistischen Lerntheorie:
Wann führt ein kleiner Lernfehler auch zu einem kleinen Testfehler?

Generalisierung: die Fähigkeit des Klassifikators unbekannte Stichproben mit einem kleinen Fehler zu klassifizieren.



Stochastischer Prozess unbekannter Verbundverteilung $P(\mathbf{m},\omega)$

Lernfehler $e_{\text{Lern}}(\mathbf{m},\boldsymbol{\theta})$

Testfehler $e_{\text{Test}}(\mathbf{m},\boldsymbol{\theta})$

Gesucht ist eine Entscheidungsfunktion $k(\mathbf{m},\boldsymbol{\theta})$, die durch den unbekannten Vektor $\boldsymbol{\theta}$ parametriert wird und die den Erwartungswert des Testfehlers $\varepsilon(\boldsymbol{\theta}) = \mathrm{E}\{e_{\text{Test}}(\mathbf{m},\boldsymbol{\theta})\}$ minimiert.

Generalisierungsfähigkeit ist umso höher, desto geringer der mittlere Testfehler ist.

Vapnik-Chervonenkis Lerntheorie: Mit der Wahrscheinlichkeit 1-η gilt die folgende Abschätzung für den Testfehler:

$$\varepsilon(\boldsymbol{\theta}) \leq e_{\text{Lern}}(\boldsymbol{\theta}) + \Phi(v, N, \eta)$$

VC-Konfidenz:

$$\Phi(v, N, \eta) = \sqrt{\frac{v\left(\log\frac{2N}{v} + 1\right) - \log\left(\frac{\eta}{4}\right)}{N}}$$

N: Zahl der Beispiele in der Lernstichprobe

v: VC-Dimension der Funktionenmenge $\mathrm{K} = \{k(\mathbf{m}, \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$

Voraussetzungen:

- Allen Daten liegt die gleiche Verteilung zugrunde.
- Daten wurden unabhängig erzeugt.

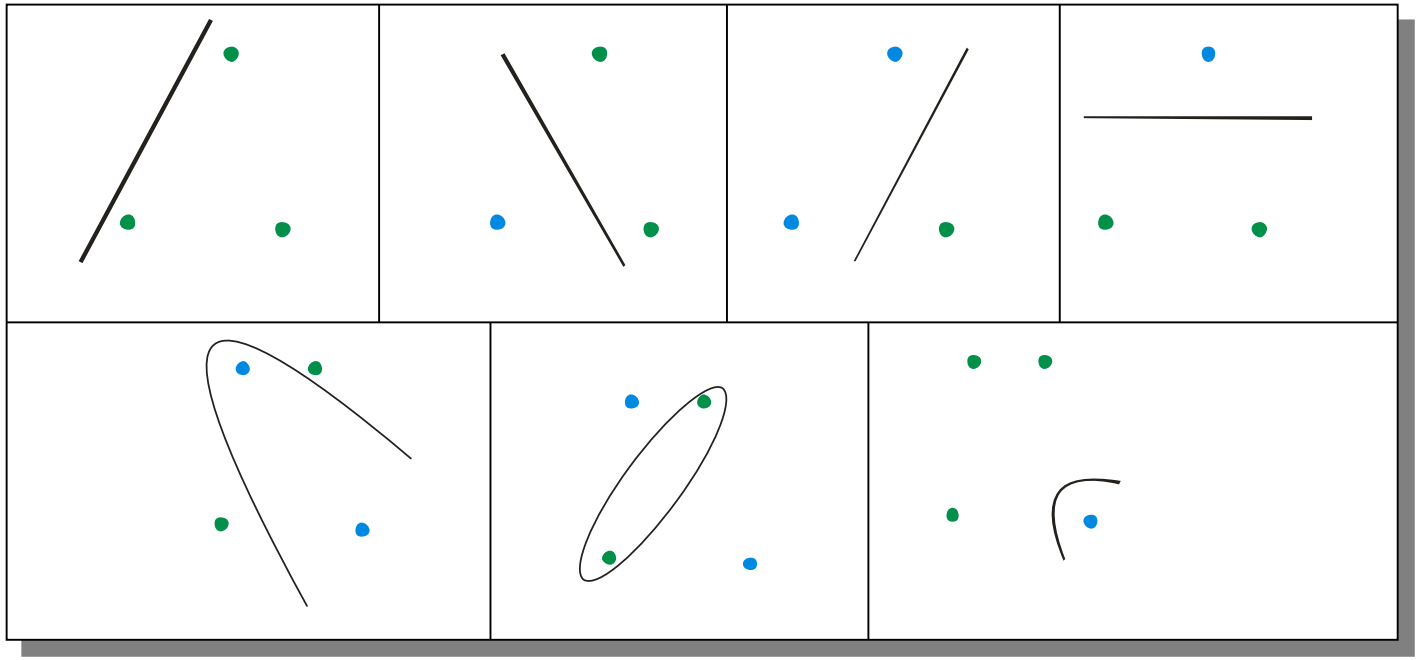
→ d.h.: i.i.d. Daten

Das Ergebnis hängt nicht von der zugrunde liegenden Verteilung ab.

Vapnik-Chervonenkis-Dimension (VC-Dimension) ν

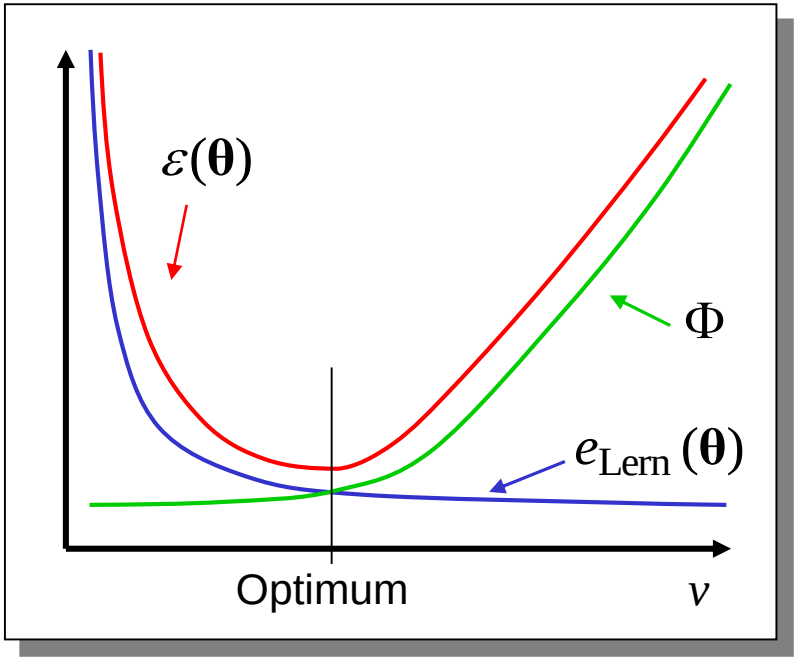
Eine gegebene Menge von N Stichproben kann für den Zweiklassen-Fall in 2^N mögliche Klassenzugehörigkeitskonstellationen annehmen. Die VC-Dimension ν einer Menge K von Funktionen ist definiert als die maximale Anzahl von Stichproben, die durch K bei allen möglichen Klassenzugehörigkeitskonstellationen separiert werden können.

Bsp.:

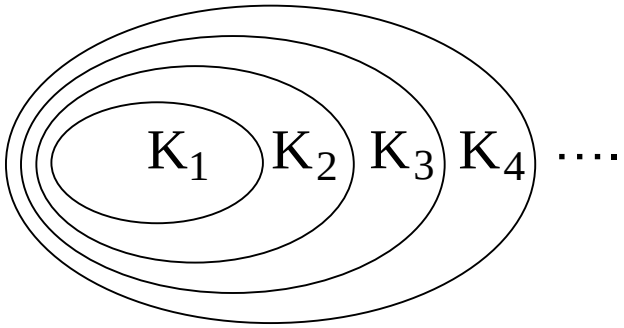


- Für K = { Hyperebenen in $\mathbb{R}^d$ } ist $\nu = d+1$.
- Bei Polynomen wächst ν mit dem Polynomgrad an.

Vapnik-Chervonenkis Lerntheorie



- $v \uparrow \Rightarrow \Phi(v, N, \eta) \uparrow$
- $v \uparrow \Rightarrow e_{\text{Lern}}(\theta) \downarrow$ (Overfitting)
- $v \uparrow$ bedeutet wachsende „Plastizität“ der Funktionenmenge K



$$v_1 < v_2 < v_3 < v_4 < \ldots.$$

Mit dieser Theorie kann ein Kompromiss zwischen Lernfehler und Generalisierungsfähigkeit kann gefunden werden.

Literatur: V. Vapnik: Statistical Learning Theory. Wiley 1998

Structural Risk Minimization (SRM) Prinzip

$$\varepsilon(\boldsymbol{\theta}) \le e_{\text{Lern}}(\boldsymbol{\theta}) + \Phi(\nu, N, \eta) = e_{\text{Lern}}(\boldsymbol{\theta}) + \sqrt{\frac{\nu\left(\log\frac{2N}{\nu}+1\right) - \log\left(\frac{\eta}{4}\right)}{N}}$$

Fall 1: $\frac{N}{\nu} \to \infty \quad \Rightarrow \quad \Phi \to 0 \quad \Rightarrow \quad \varepsilon \to e_{Lern}$

Es reicht den Lernfehler zu minimieren. Ein kleiner Lernfehler garantiert einen kleinen Testfehler. *Empirical Risk Minimization* (ERM) Prinzip.

Fall 2: $\frac{N}{\nu}$ ist klein. $\Rightarrow$ Φ wird wesentlich.

Ein kleiner Lernfehler garantiert nicht für einen kleinen Testfehler.
e_{Lern} und Φ müssen simultan minimiert werden.

Structural Risk Minimization **(SRM)** Prinzip (Vapnik).

Schätzung der Fehlerwahrscheinlichkeit

Voraussetzungen:

Zu untersuchender Klassifikator wurde mit der Lernstichprobe D trainiert:

Klassenanzahl: c

Teststichprobe T: $|\mathrm{T}| = N_{\mathrm{T}}$

Teilteststichprobe T_j: $|\mathrm{T}_j| = N_{\mathrm{T},j}$, enthält die Beispiele der Klasse ω_j

Alle Elemente der Teststichprobe sind unabhängig.

P_j: unbekannte Fehlerwahrscheinlichkeit für die Klasse ω_j

n_j: Anzahl der falsch klassifizierten Teststichprobenelemente aus T_j

$$\Pr\{n_j \text{ Elemente von } \mathrm{T}_j \text{ falsch klassifiziert}\} = \binom{N_{\mathrm{T},j}}{n_j} P_j^{n_j} (1-P_j)^{N_{\mathrm{T},j}-n_j}$$

Schätzung der Fehlerwahrscheinlichkeit

ML-Schätzung der Fehlerwahrscheinlichkeit für Klasse ω_j → $\hat{P}_j = \frac{n_j}{N_{\mathrm{T},j}}$

$$\hat{P}_e = \sum_{j=1}^{c} P(\omega_j) \frac{n_j}{N_{\mathrm{T},j}}$$

P_e: Fehlerwahrscheinlichkeit des Klassifikators

Mit $\mathrm{E}\{n_j\} = N_{\mathrm{T},j} P_j$ folgt $\mathrm{E}\{\hat{P}_e\} = \sum_{j=1}^{c} P(\omega_j) P_j = P_e$: erwartungstreu

$$\mathrm{Var}\{\hat{P}_e\} = \sum_{j=1}^{c} P^2(\omega_j) \frac{P_j(1-P_j)}{N_{\mathrm{T},j}}$$

Probleme:

- Kleine Teilstichproben
- Vergleich unterschiedlicher Klassifikatoren mit Unterschieden in der Fehlerwahrscheinlichkeit in der Größenordnung oder kleiner als der stochastische Fehler $\sqrt{\mathrm{Var}\{\hat{P}_e\}}$

Schätzung der Fehlerwahrscheinlichkeit

Allgemeinere Vorgehensweise

Guyon I., Makhoul J., Schwarz R., Vapnik, V.: „What size test set gives good error rate estimates? “ IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20(1), pp52-64, 1998.

Ziel: Bestimmung von N_T, sodass die wahre Fehlerwahrscheinlichkeit P_e die geschätzte Fehlerwahrscheinlichkeit $\hat{P}_e$ mit Wahrscheinlichkeit $1 - a$ um nicht mehr als $\varepsilon(N_\mathrm{T}, a)$ überschreitet.

$$\Pr\left\{P_e \geq \hat{P}_e + \varepsilon(N_\mathrm{T}, a)\right\} \leq a \qquad 0 \leq a \leq 1$$

Mit $\varepsilon(N_\mathrm{T}, a) := \beta P_e$ und speziell mit $a = 0{,}05$ und $\beta = 0{,}2$ folgt:

$$N_\mathrm{T} \approx \frac{100}{P_e}$$

Ergebnis ist unabhängig von der Klassenzahl c.

9.2 Empirische Leistungsbestimmung von Klassifikatoren

Umgang mit kleinen Stichproben:

Cross-Validation (m-fache Cross-Validation)

$$S = S_1 \cup S_2 \cup \ldots \cup S_q \qquad S_i \cap S_j = \varnothing \text{ für } i \neq j \qquad |S_1| = |S_2| = \ldots = |S_q|$$

Lernstichprobe:

$$D := S \setminus S_j$$

Teststichprobe:

$$T := S_j \qquad j = 1, \ldots, q$$

Bemerkungen:

- Empirisches Verfahren zur Bewertung der Güte eines Klassifikators.
- Besonders geeignet bei geringem Stichprobenumfang.
- Der Klassifikator wird mit D trainiert und mit T getestet.
- Wiederholung bis jede der q Teilstichproben Teststichprobe war.
- Gesamter Testfehler = Mittelwert der q Testfehler
- Typische Werte: $q = 5, 10$

9.2 Empirische Leistungsbestimmung von Klassifikatoren

Umgang mit kleinen Stichproben:

Leave-one-out

$$S = S_1 \cup S_2 \cup ... \cup S_N \qquad |S_1| = |S_2| = ... = |S_N| = 1$$

Lernstichprobe:

$$D := S \setminus S_j$$

Teststichprobe:

$$T := S_j \qquad j = 1,\ldots,N$$

Bemerkungen:

- Sonderfall von Cross-Validation, mit $q = N$ (Anzahl der Stichproben).
- Anwendung genau wie Cross-Validation
- Nachteil: Großer Aufwand

9.3 Boosting

Gegeben:

Merkmalsvektoren: $\mathbf{m}_i \in \mathrm{D}, \quad i = 1,\ldots,N$

Indikatorvariablen: $z_i := \begin{cases} 1 & \text{für } \omega(\mathbf{m}_i) = \omega_1 \\ -1 & \text{für } \omega(\mathbf{m}_i) = \omega_2 \end{cases}$

Idee:

- Kombination mehrerer „schwacher“ Klassifikatoren (nur geringfügig besser als eine Zufallsentscheidung (Raten)) zu einem starken.
- Ergebnis der Klassifikation $k(\mathbf{m})$ basiert auf der gewichteten Summe der M „schwachen“ Klassifikatoren $k_j(\mathbf{m})$.

$$k(\mathbf{m}) = \operatorname{sign}\left(\sum_{j=1}^{M} \alpha_j k_j(\mathbf{m})\right)$$

- Die Gewichte α_j werden im Training so bestimmt, dass die „schwachen“ Klassifikatoren $k_j(\mathbf{m})$ nach ihrer Güte gewichtet werden.

Der Algorithmus AdaBoost.M1

Initialisierung: $w_i = \frac{1}{N}, i = 1,\ldots,N$

For $j = 1,\ldots,M$ do:

- Trainiere Klassifikator $k_j(\mathbf{m})$ unter Berücksichtigung von $\{w_i\}$
- Berechne den Lernfehler:

$$\varepsilon_j = \frac{\sum_{i=1}^{N} w_i \mathrm{I}\left(z_i \neq k_j(\mathbf{m}_i)\right)}{\sum_{i=1}^{N} w_i}, \quad \text{mit} \quad \mathrm{I}\left(z_i \neq k_j(\mathbf{m}_i)\right) = \begin{cases} 1 & z_i \neq k_j(\mathbf{m}_i) \\ 0 & z_i = k_j(\mathbf{m}_i) \end{cases}$$

- Berechne:

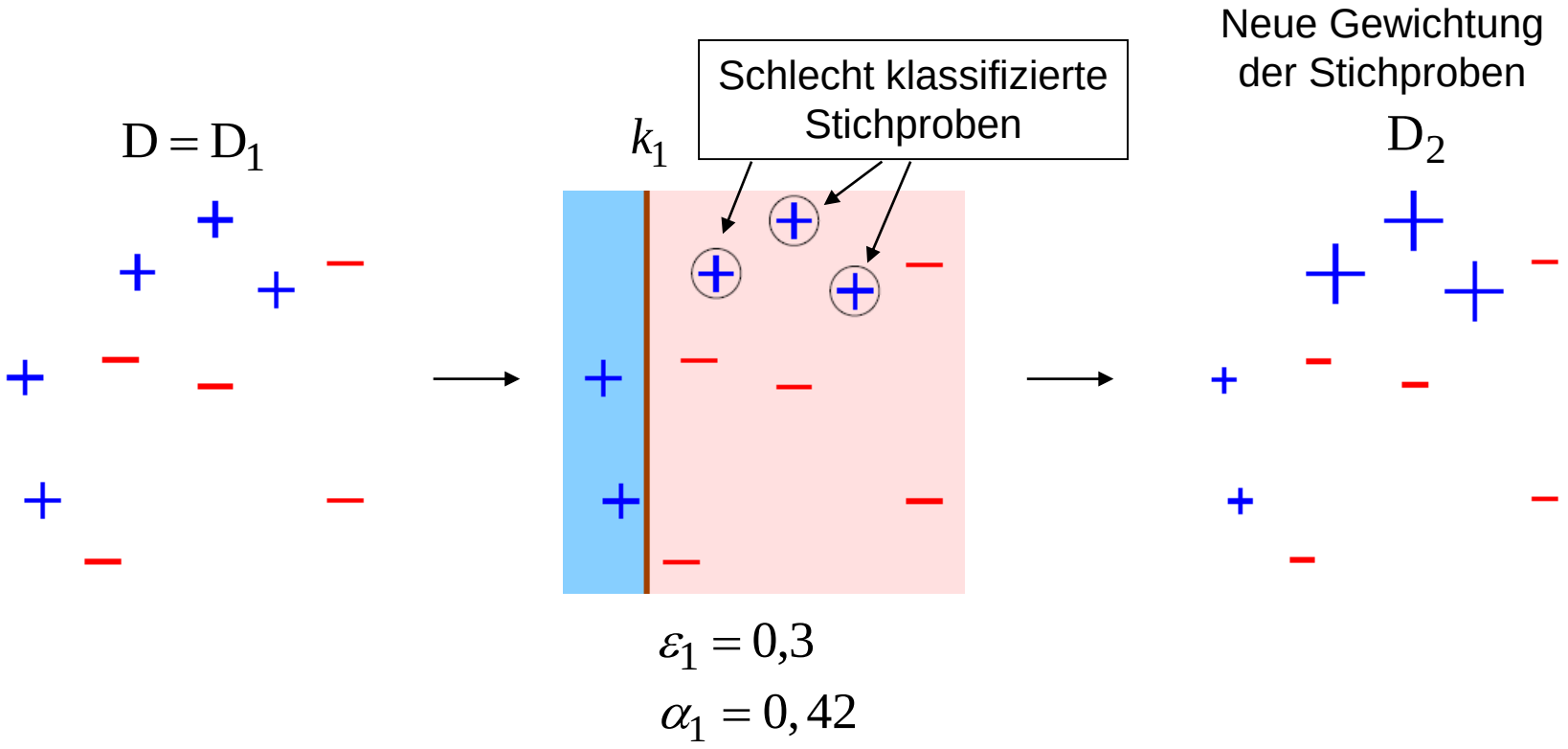
$$\alpha_j = \ln\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right)$$

- Aktualisiere:

$$w_i := w_i \exp\left(\alpha_j \, \mathrm{I}\left(z_i \neq k_j(\mathbf{m}_i)\right)\right), \quad i = 1,\ldots,N$$

T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning

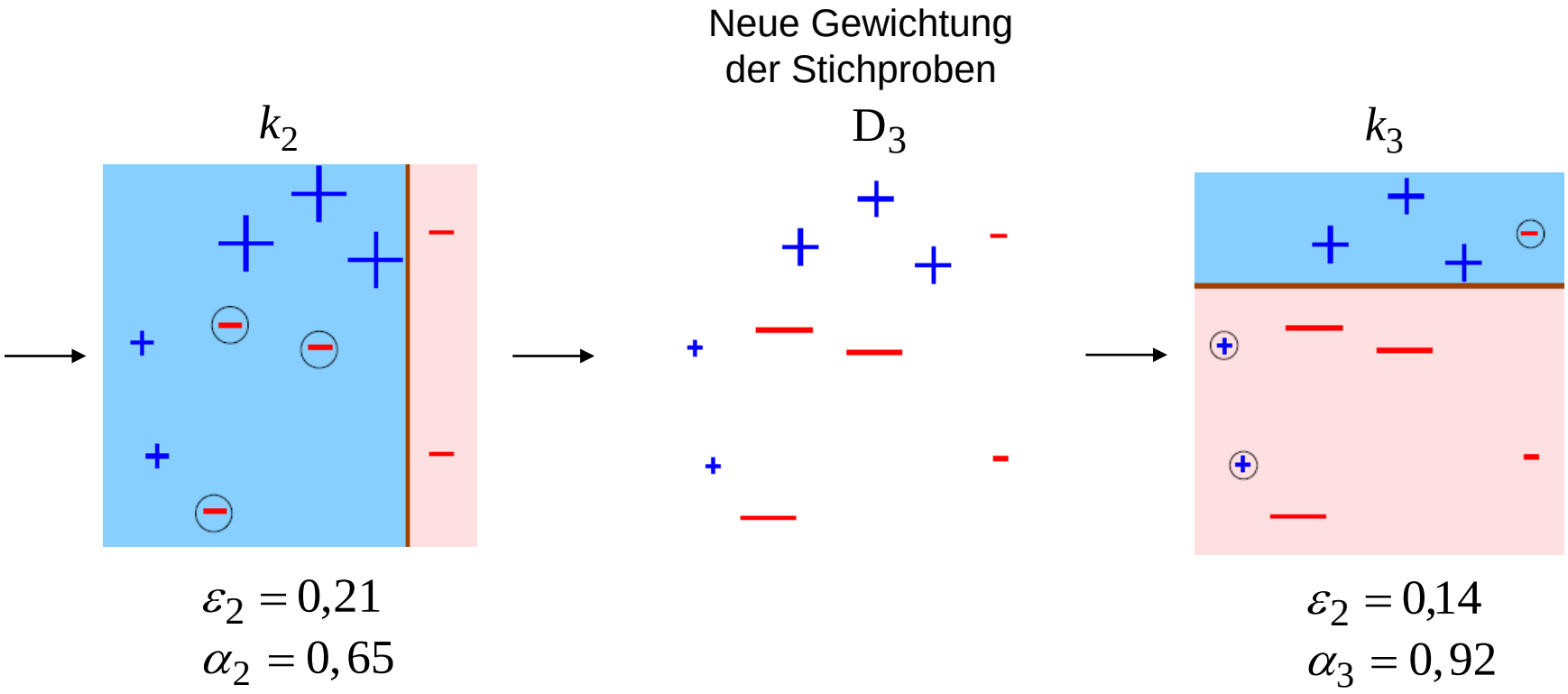
Beispiel: AdaBoost für $c = 2$



Anmerkung: Die „schwachen“ Klassifikatoren sind in diesem Beispiel „achsparallele“ Entscheidungsfunktionen $\text{sign}(m_1 - a)$ und $\text{sign}(m_2 - b)$.

Schapire, E. R., A Boosting Tutorial, (Toy Example)

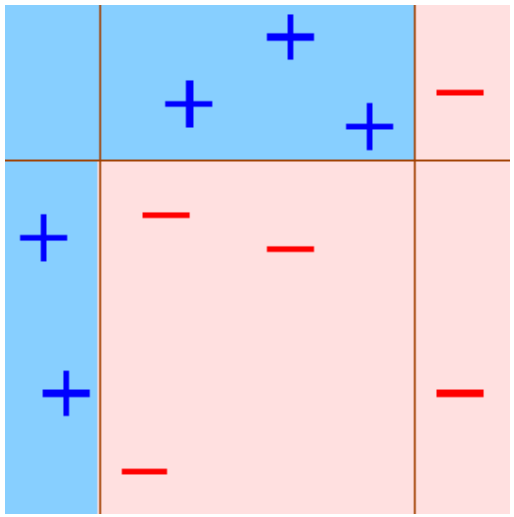
Beispiel: AdaBoost für $c = 2$



Schapire, E. R., A Boosting Tutorial, (Toy Example)

Beispiel: AdaBoost für c=2

$$k(\mathbf{m}) = \text{sign}\left(0,42 \; k_1 + 0,65 \; k_2 + 0,92 \; k_3 \right)$$



Schapire, E. R., A Boosting Tutorial, (Toy Example)

9.3 Boosting

Bemerkungen:

- Allgemein anwendbar auf unterschiedliche Klassifikatoren (Metaverfahren)
- Sehr leistungsfähiger Ansatz
- Gute Generalisierungsfähigkeit
- Neigt nicht zu Overfitting
- Gewichtung bevorzugt bessere Klassifikatoren
- Designfreiheitsgrade: schwache Klassifikatoren und Parameter M.
- Kein Vorwissen bezüglich der Merkmalsverteilungen notwendig
- Braucht große Stichproben

Prüfung zur Vorlesung Mustererkennung

Termin: 08.09.17, 11:00 bis 13:00 im Hörsaal am Fasanengarten (HSaF)

Hilfsmittel wie: Taschenrechner, Kommunikationssysteme, Aufzeichnungen, Bücher usw. sind **nicht erlaubt**.

Weitere Details zur Anmeldung siehe IES-Homepage.