Computer Vision: AdaBoost

D. Schlesinger – TUD/INF/KI/IS

Idee

Gegeben sei eine Menge schwacher (einfacher, schlechter) Klassifikatoren
Man bilde einen guten durch eine „geschickte“ Kombination der schwachen.

Vergleiche mit SVM – komplizierte Merkmalsräume, *ein* Klassifikator.

Ausgangspunkt:

- Die Menge der schwachen Klassifikatoren $\mathcal{H}$
  Beispiel: lineare Klassifikatoren für zwei Klassen $h \in \mathcal{H} : \mathcal{X} \to \{-1, +1\}$

$$h(x) = \operatorname{sign}\Big(\langle x, w\rangle + b\Big)$$

- Eine klassifizierte Lernstichprobe
  $\Big((x_1, k_1), (x_2, k_2) \dots (x_m, k_m)\Big)$, $x_i \in \mathcal{X}$, $k_i \in \{-1, +1\}$

Gesucht wird ein Klassifikator

$$f(x) = \operatorname{sign}\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big)$$

mit $h_t \in \mathcal{H}$, $\alpha_t \in \mathbb{R}$, der die Lernstichprobe richtig (oder am besten) separiert.

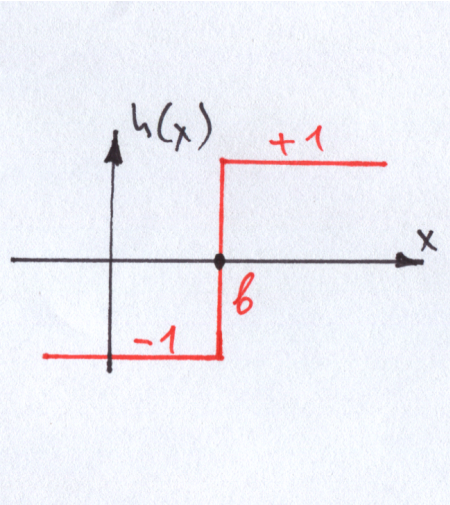
Fragen:

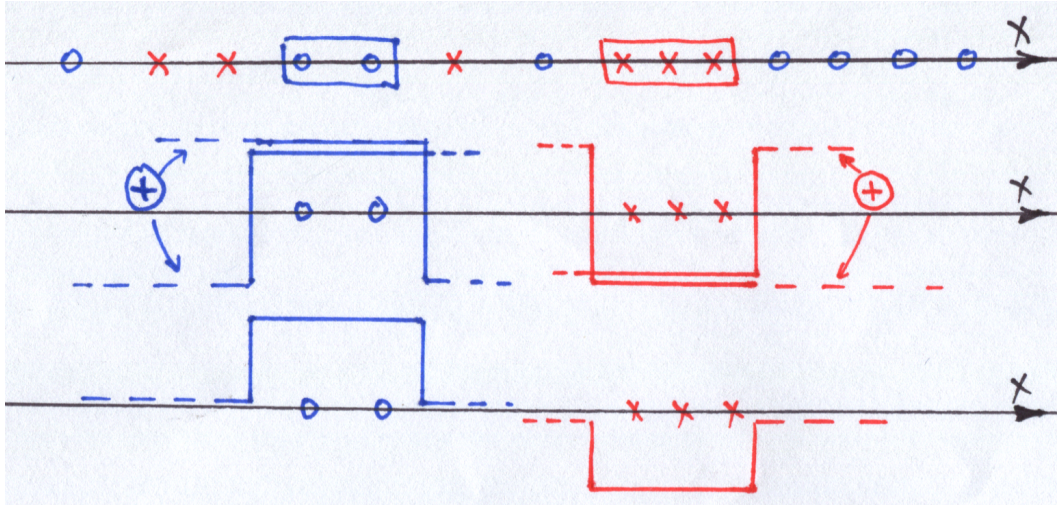
- Ist es überhaupt für eine beliebige Lernstichprobe möglich?
- Eigenschaften der Konvergenz, Generalisierbarkeit.

Mächtigkeit der Menge der Entscheidungsstrategien

Ja, es ist für eine beliebige (endliche) Lernstichprobe möglich
(wenn die Anzahl der zu verwendeten Klassifikatoren nicht eingeschränkt ist).

Beispiel mit $x \in \mathbb{R}$, d.h. $h(x) = \text{sign}(\langle x, w \rangle + b) = \pm\text{sign}(x - b)$

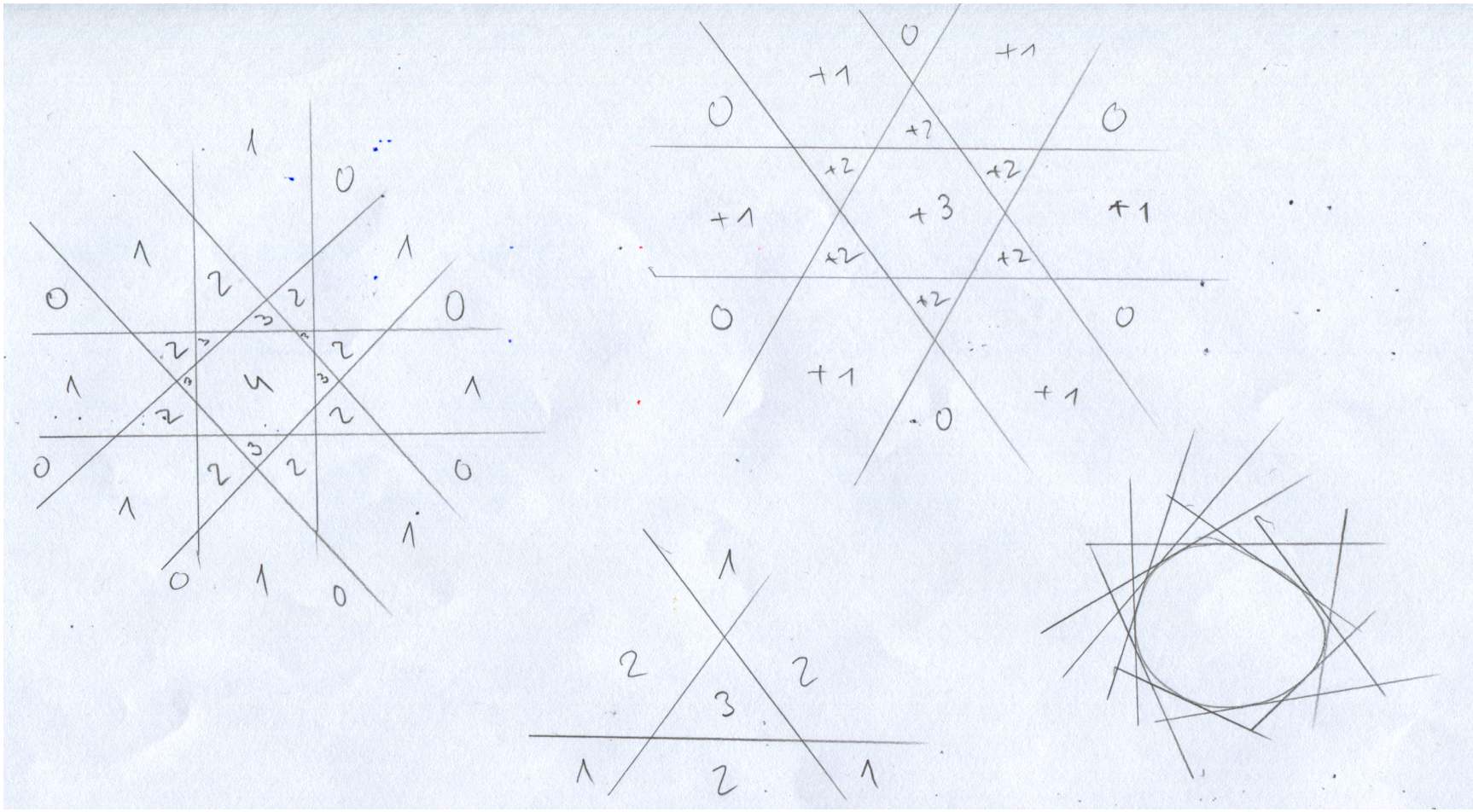




Der Klassifikator wird aus einfachen Klassifikationen gebildet, die jeweils für ein bestimmtes Muster die richtige Klasse liefern und „neutral“ für alle anderen sind.

Mächtigkeit der Menge der Entscheidungsstrategien

Beispiele mit $x \in \mathbb{R}^2$



AdaBoost – Algorithmus

Gegeben: $\Big((x_1, k_1), (x_2, k_2) \dots (x_m, k_m)\Big)$, $x_i \in \mathcal{X}$, $k_i \in \{-1, +1\}$

Initialisiere **Gewichte** für alle Beispiele mit $D^{(1)}(i) = 1/m$

Für $t = 1, \dots, T$

1. Wähle (lerne) einen schwachen Klassifikator $h_t \in \mathcal{H}$
   unter Berücksichtigung aktueller Gewichte $D^{(t)}$
2. Wähle α_t
3. Aktualisiere die Gewichte:

$$D^{(t+1)}(i) = \frac{D^{(t)}(i) \cdot \exp\Big(-\alpha_t y_i h_t(x_i)\Big)}{Z_t}$$

   mit der Normierungskonstante Z_t so, dass $\sum_i D^{(t+1)}(i) = 1$.

Der „starke“ Klassifikator ist:

$$f(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost – Algorithmus

1. Wähle (lerne) einen schwachen Klassifikator $h_t \in \mathcal{H}$.

$$h_t = \arg\min_{h \in \mathcal{H}} \epsilon(D^{(t)}, h) = \arg\min_{h \in \mathcal{H}} \sum_i D^{(t)}(i) \cdot \mathbb{I}\Big(y_i, h(x_i)\Big)$$

d.h. wähle den besten bezüglich aktueller $D(i)$ – (SVM)
Voraussetzung: $\epsilon(D^{(t)}, h) < 1/2$
– der beste h_t soll nicht schlechter sein, als eine zufällige Entscheidung.
Anderenfalls – Abbruch.

2. Wähle α_t. Das Ziel ist, $f(x)$ so zu konstruieren, dass sein Fehler $\epsilon(f) = \sum_i \mathbb{I}\Big(y_i, f(x_i)\Big)$ minimal ist.
Obere Schranke für den Fehler ist $\epsilon(f) \leq \prod_{t=1}^{T} Z_t$.
$\Rightarrow$ wähle α_t (gierig) so dass Z_t minimal ist.

$$Z_t = \sum_i D^{(t)}(i) \cdot \exp\Big(-\alpha_t y_i h_t(x_i)\Big) \to \min_{\alpha_t}$$

Die Aufgabe ist konvex und differenzierbar $\to$

$$\alpha_t = 1/2 \ln \left(\frac{1 - \epsilon(D^{(t)}, h_t)}{\epsilon(D^{(t)}, h_t)} \right)$$

AdaBoost – Algorithmus

3. Aktualisiere die Gewichte:

$$D^{(t+1)}(i) \sim D^{(t)}(i) \cdot \exp\Big(-\alpha_t y_i h_t(x_i)\Big)$$

Merke: $\alpha_t > 0$

$$\begin{aligned} y_i h_t(x_i) > 0 \text{ (richtig klassifiziert)} \quad &\Rightarrow \quad \exp\Big(-\alpha_t y_i h_t(x_i)\Big) < 1 \\ y_i h_t(x_i) < 0 \text{ (falsch klassifiziert)} \quad &\Rightarrow \quad \exp\Big(-\alpha_t y_i h_t(x_i)\Big) > 1 \end{aligned}$$

Die (aktuell) falsch klassifizierten Muster werden stärker gewichtet

$\Rightarrow$ der Klassifikator h_{t+1} in der nächsten Runde wird versuchen gerade diese richtig zu klassifizieren.

– Beispiel von Matas
– Beispiele von Freund: `http://cseweb.ucsd.edu/~yfreund/adaboost/`

AdaBoost – Zusammenfassung

Geschichte (Arbeiten):

1990 – Boost-by-majority algorithm (Freund)

1995 – AdaBoost (Freund & Schapire)

1997 – Generalized version of AdaBoost (Schapire & Singer) (**heute**)

2001 – AdaBoost in Face Detection (Viola & Jones)

Interessante Eigenschaften:

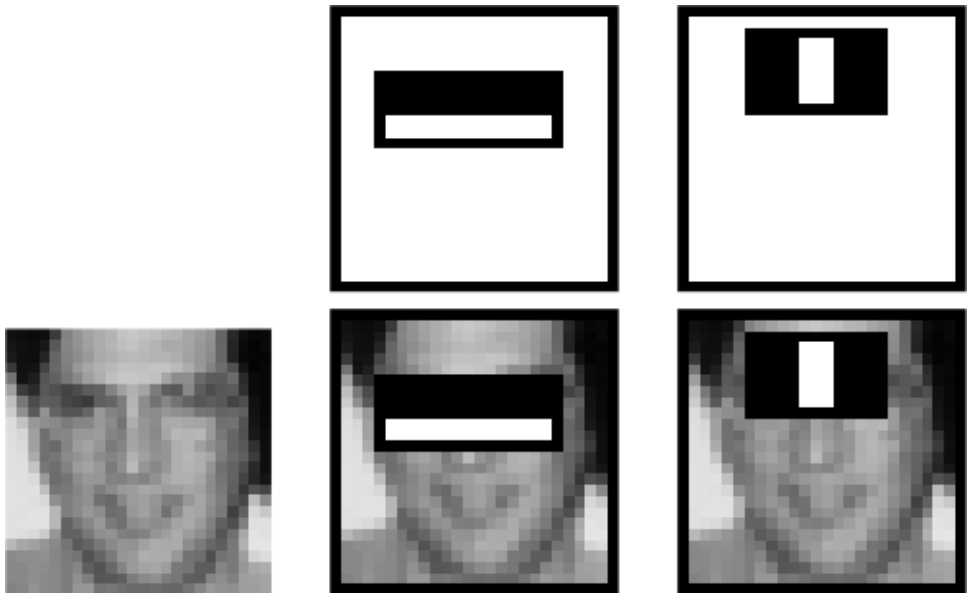
- AB ist eine einfache Kombination linearer Klassifikatoren – sehr einfach.
- AB konvergiert zum Logarithmus der Likelihood-Verhältnis.
- AB hat gute Generalisierbarkeit (?).
- AB ist ein Merkmalselektor.

Viola & Jones (CVPR 2001): Rapid Object Detection using a Boosted Cascade of Simple Features

Haar Merkmale – schnell berechenbar
24×24 Fenster $\times \ldots \rightarrow$ 180.000 Merkmalswerte pro Position!!!

Ein „schwacher“ Klassifikator
– Wert der Faltung mit Haar-Maske $\lesseqgtr$ (optimaler) Schwellwert

AdaBoost für das Lernen – Auswahl der Merkmale



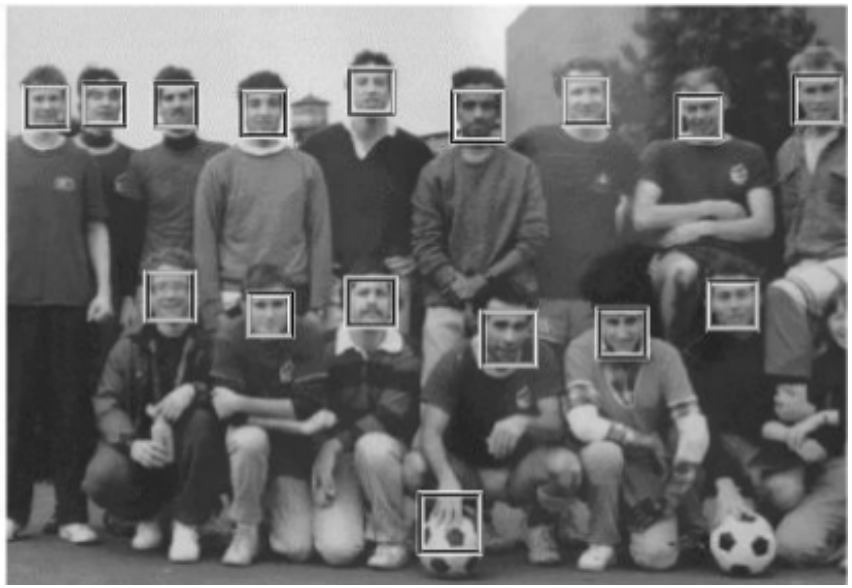
Die zwei beste Merkmale

Die besseren – 0.1 bis 0.3 Fehler, die später gewählten – 0.4 bis 0.5

Etwas extra noch...

Viola & Jones (CVPR 2001): Rapid Object Detection using a Boosted Cascade of Simple Features

Datenbank: 130 Bilder, 507 Gesichter



Gesamtfehlerrate – ca. 7%

Erkennung – Zusammenfassung