第7.1节 Boosting方法

中国科学院大学 叶齐祥

qxye@ucas.ac.cn

内容

- 背景介绍
- Adaboost 算法
- Face detection例子

Adaboost的优点

- 高的分类精度
- 可以让多种分类器集成起来
  - 很多领域都很适用
- 程序上易于实现，运行速度高
- 不易产生overfitting

Adaboost的历史

- Bootstrapping
- Bagging
- Boosting (Schapire 1989)
- Adaboost (Schapire 1995)

Bagging - Aggregate Bootstrapping

- For i = 1 .. M
  - 从训练集合 D *提取* $n^* < n$ 个样本
  - 学习分类器 h_i
- 最终的分类器是 $h_1 \ldots h_i$, h_M的投票结果
- 增加了分类器的稳定性

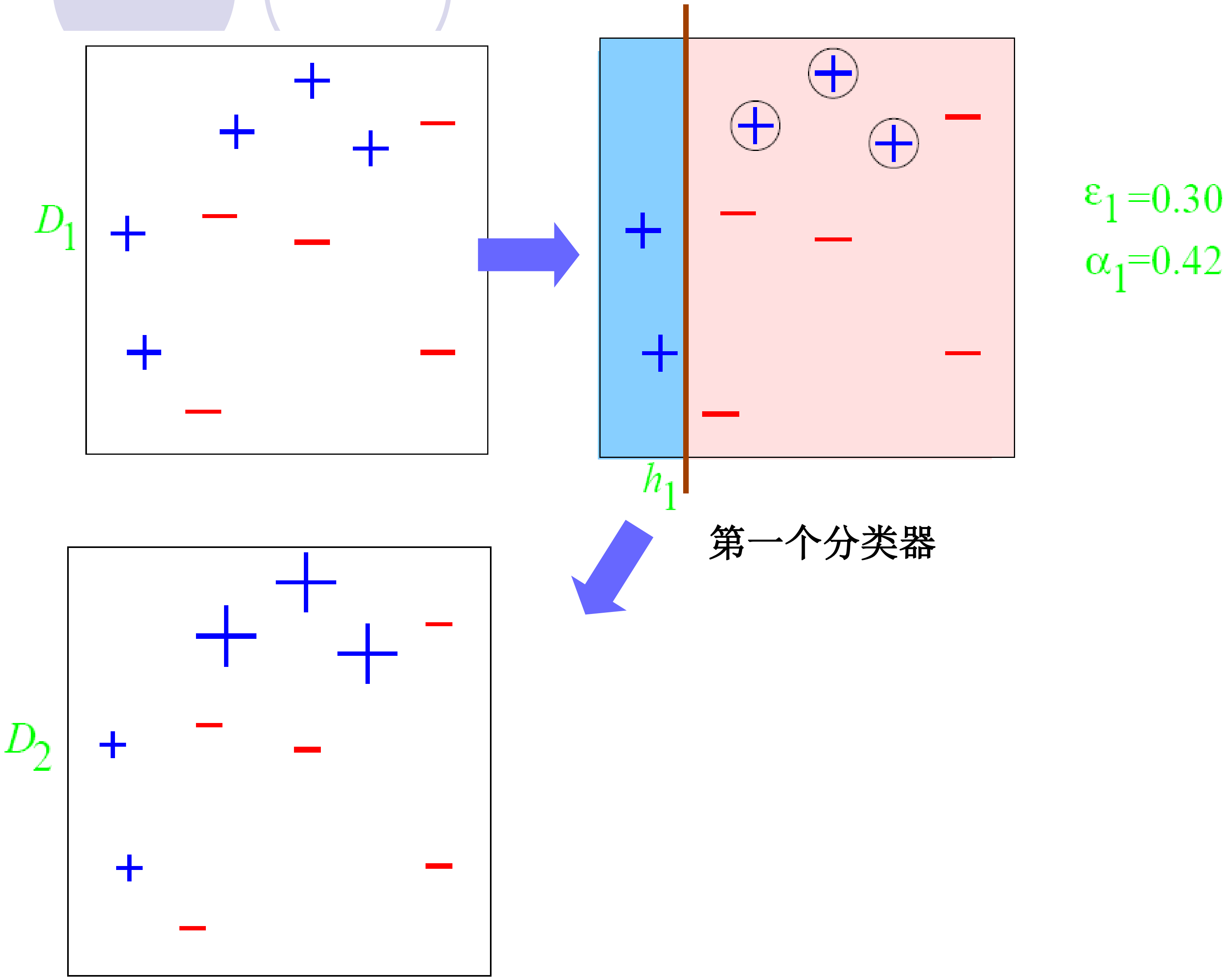
Boosting (Schapire 1989)

- 从训练集合 D 提取$n_1<n$ 个样本,构建D_1
  - *训练弱分类器* h_1
- 从训练集合 D 提取$n_2<n$ 个被h_1错误分类的样本,构建D_2
  - *训练弱分类器* h_2
- 从训练集合 D 提取$n_3<n$ 个被h_1，h_2错误分类的样本,构建D_3
  - *训练弱分类器* h_3
- ……
- 最后的分类器是 h_1, h_2, h_3…的投票结果

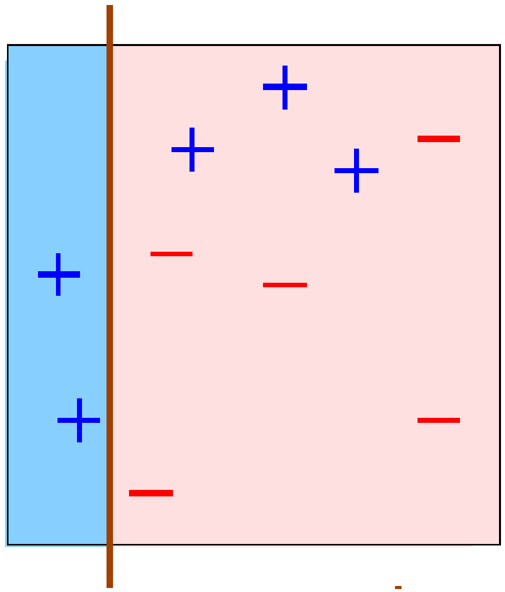
Adaboost - Adaptive Boosting

- 对样本不仅仅是采样,还有 加权(re-weight)
- 最终分类器是弱分类器的加权平均
  - 加权线性组合
- 弱分类器
  - -Weak Learner: 错误率< 50%的分类器

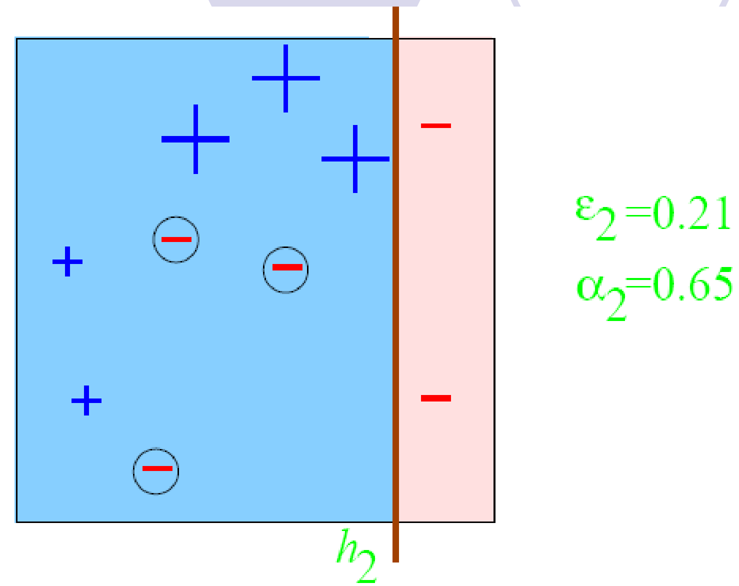
Adaboost – 示意图



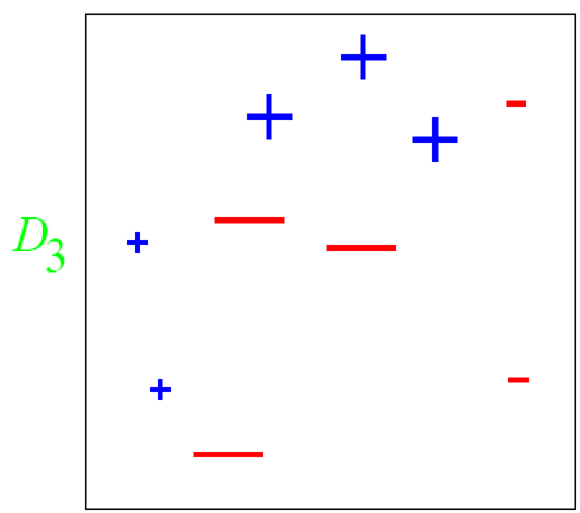
Adaboost –示意图



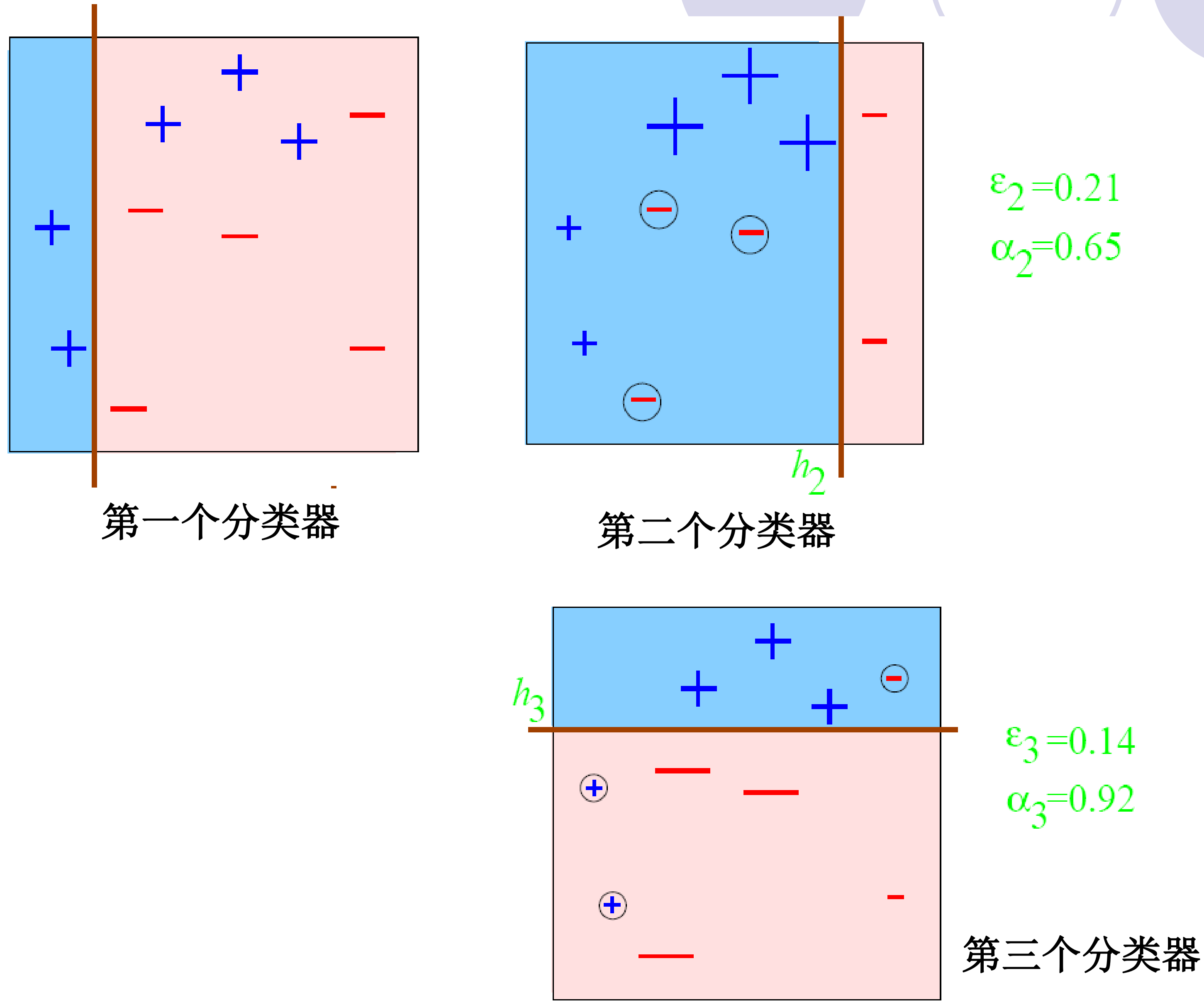
第一个分类器



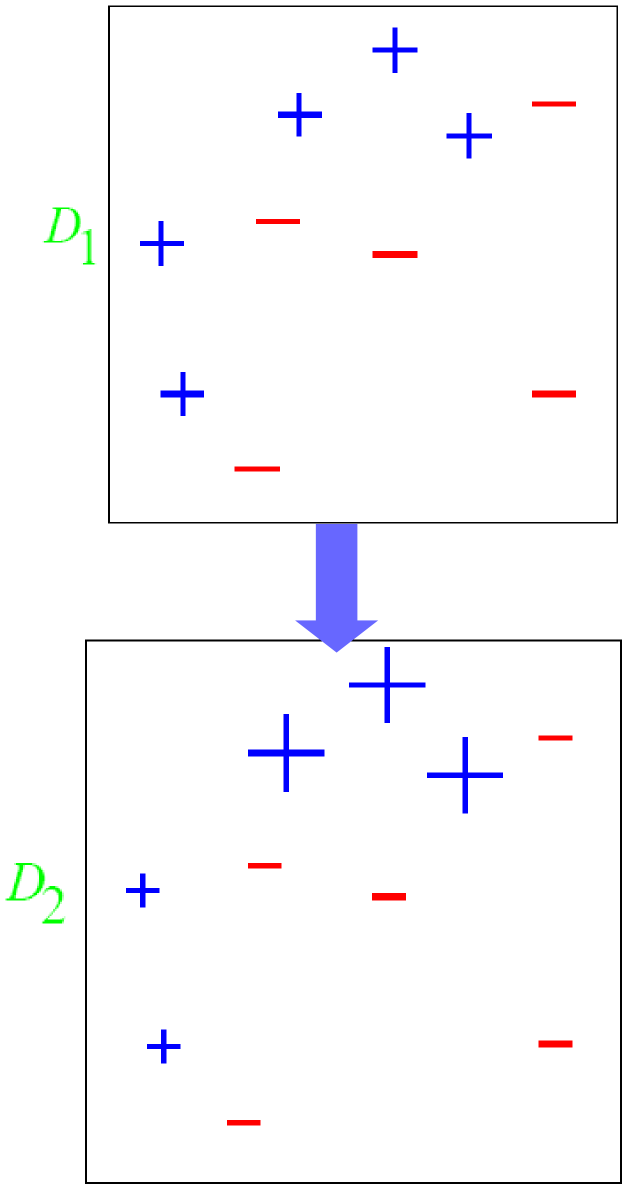
第二个分类器

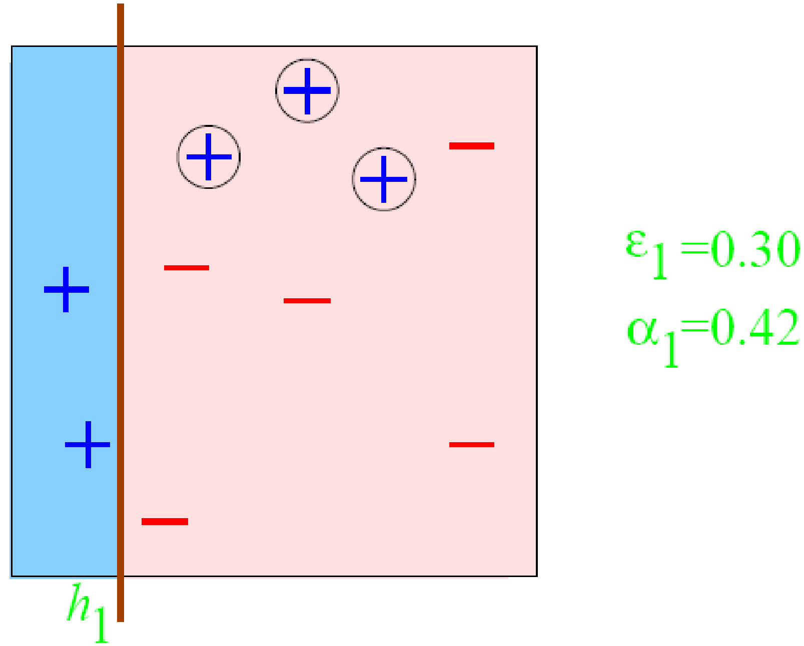


Adaboost –示意图



Adaboost –示意图





第一个分类器

在下一个循环中，

增大被分错的样本权值

Adaboost – 算法

- 对于给定的训练样本集合 $X = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- $y_i \in \{-1, +1\}$ 表示这些样本的真实标号

- For (t = 1, …, T)
  - 在 $\{1, \ldots, n\}$ 构造 D_t
  - 找到弱分类器 $h_t : X \to \{-1, +1\}$

    使得分类误差：$\varepsilon_t = \mathrm{P}_{D_t}\left[h_t(x_i) \neq y_i\right]$最小

- 输出最终的假设（分类器）

$$H(x) = sign\left(\sum_t \alpha_t \cdot h_t(x)\right)$$

Adaboost – 算法

- 改变训练过程中样本集合的重要性（权值）D_t
- $D_1(i) = 1.0 / n$
- 给定 D_t 与 h_t 令 $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$

$$= \frac{D_t(i)}{Z_t} \cdot \exp(-\alpha_t \, y_i \, h_t(x_i))$$

  - 其中 Z_t 是归一化常数

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$$

- 输出最终的假设（分类器）

$$H_{\text{final}}(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$$

Adaboost – 算法

- 权值分析

$$f(x) = \sum_t \alpha_t h_t(x) \Rightarrow H_{\text{final}}(x) = \text{sign}(f(x))$$

$$D_{\text{final}}(i) = \frac{1}{\text{n}} \cdot \frac{\exp\left(-y_i \sum_t \alpha_t h_t(x_i)\right)}{\prod_t Z_t}$$

$$= \frac{1}{\text{n}} \cdot \frac{e^{-y_i f(x_i)}}{\prod_t Z_t}$$

Adaboost – 算法

- 误差分析
  - 因为

$$H_{\text{final}}(x) \neq y \Rightarrow yf(x) \leq 0 \Rightarrow e^{-yf(x)} \geq 1$$

  - 所以

$$\begin{aligned}
\text{training error}(H_{\text{final}}) &= \frac{1}{\text{n}} \sum_i \begin{cases} 1 & \text{if } y_i \neq H_{\text{final}}(x_i) \\ 0 & \text{else} \end{cases} \\
&\leq \frac{1}{\text{n}} \sum_i e^{-y_i f(x_i)} \\
&= \sum_i D_{\text{final}}(i) \prod_t Z_t \\
&= \prod_t Z_t
\end{aligned}$$

Adaboost – 算法

- 误差分析

$$\epsilon_t = 1/2 - \gamma_t$$

$$\begin{aligned}\text{training error}(H_{\text{final}}) &\leq \prod_t \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right] \\ &= \prod_t \sqrt{1-4\gamma_t^2} \\ &\leq \exp\left(-2\sum_t \gamma_t^2\right)\end{aligned}$$

$$\forall t: \ \gamma_t \geq \gamma > 0$$

$$\text{training error}(H_{\text{final}}) \leq e^{-2\gamma^2 T}$$

Adaboost – 算法

- 多类问题 $y \in Y = \{1, \ldots, k\}$

$$h_t : X \to Y$$

  - 一种直观的方法
  - 样本权值更新

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

  - 分类假设

$$H_{\text{final}}(x) = \arg\max_{y \in Y} \sum_{t: h_t(x) = y} \alpha_t$$

$$\alpha_t = \tfrac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right) > 0$$

Adaboost – 算法

- **多类问题**
  - 将多类问题转化为二类问题的方法
  - 假如一个分类问题的类标号为五种 { a, b, c, d, e}
  - 则可以将任何一次分类转化为5个二类问题

$$x \ , \ \mathrm{c} \rightarrow \begin{cases} (x, \mathrm{a}) & , \ -1 \\ (x, \mathrm{b}) & , \ -1 \\ (x, \mathrm{c}) & , \ +1 \\ (x, \mathrm{d}) & , \ -1 \\ (x, \mathrm{e}) & , \ -1 \end{cases}$$

Adaboost – 算法

- 多类问题

$$h_t : X \times Y \to \{-1, +1\} (\text{or } \mathbb{R})$$

$$D_{t+1}(i, y) = \frac{D_t(i, y)}{Z_t} \cdot \exp(-\alpha_t\, v_i(y)\, h_t(x_i, y))$$

$$\text{where } v_i(y) = \begin{cases} +1 & \text{if } y_i = y \\ -1 & \text{if } y_i \neq y \end{cases}$$

$$H_{\text{final}}(x) = \arg\max_{y \in Y} \sum_t \alpha_t h_t(x, y)$$

- 可以证明

$$\text{training error}(H_{\text{final}}) \leq \frac{k}{2} \cdot \Pi\, Z_t$$

Adaboost – 算法

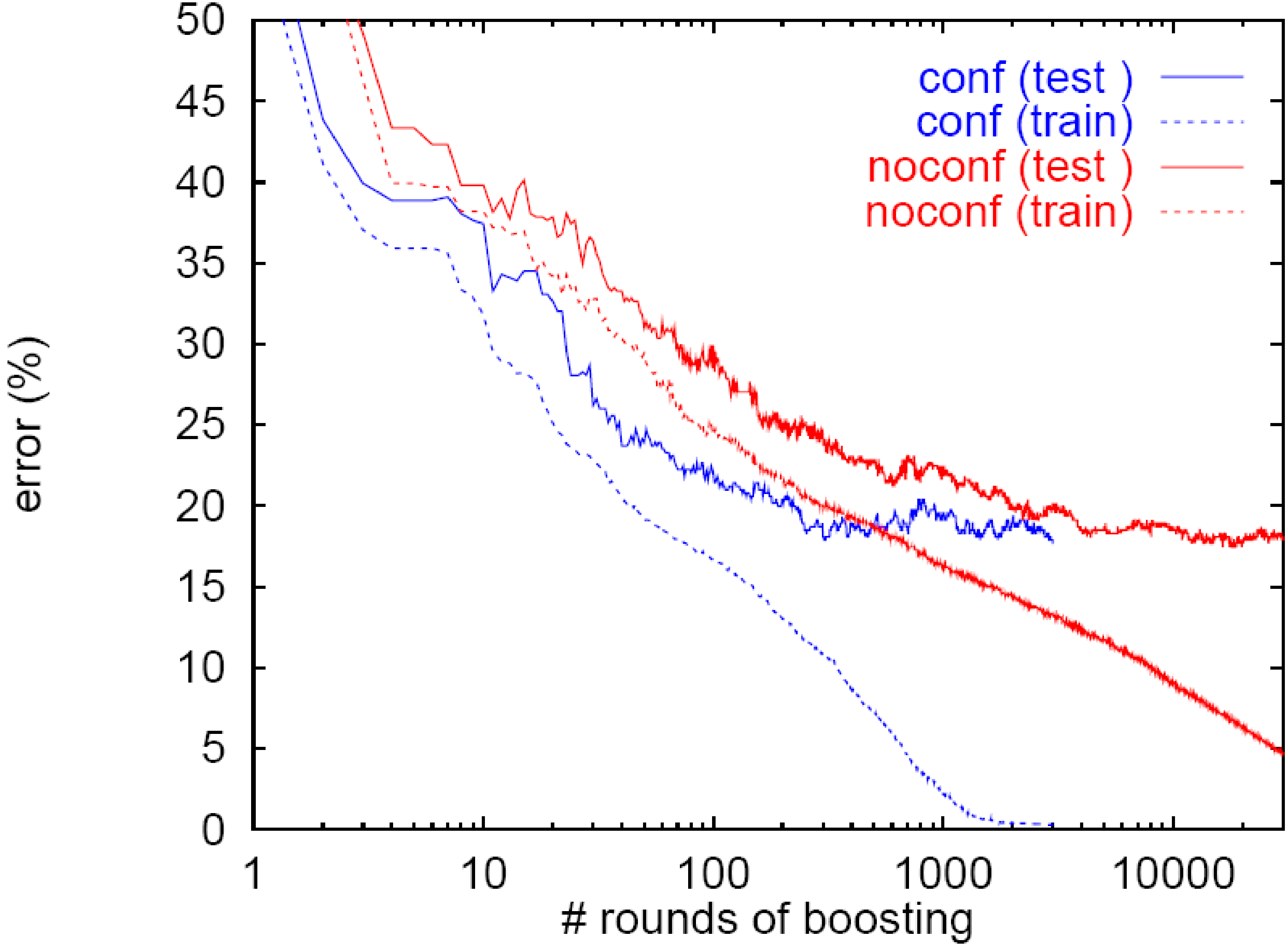
- **多类问题-类别特别多的情况**
  - 可以为每一个分类编码，编码来自少数的二值分类器

	π_1	π_2	π_3	π_4
a	$-$	$+$	$-$	$+$
b	$-$	$+$	$+$	$-$
c	$+$	$-$	$-$	$+$
d	$+$	$-$	$+$	$+$
e	$-$	$+$	$-$	$-$

$$x \ , \ \mathrm{c} \rightarrow \begin{cases} (x,\pi_1) \ , \ +1 \\ (x,\pi_2) \ , \ -1 \\ (x,\pi_3) \ , \ -1 \\ (x,\pi_4) \ , \ +1 \end{cases}$$

  - 在分类新样本时，先在二值分类器上进行，然后看分类结果的”编码”跟那一个类别“最近”

Adaboost – 算法



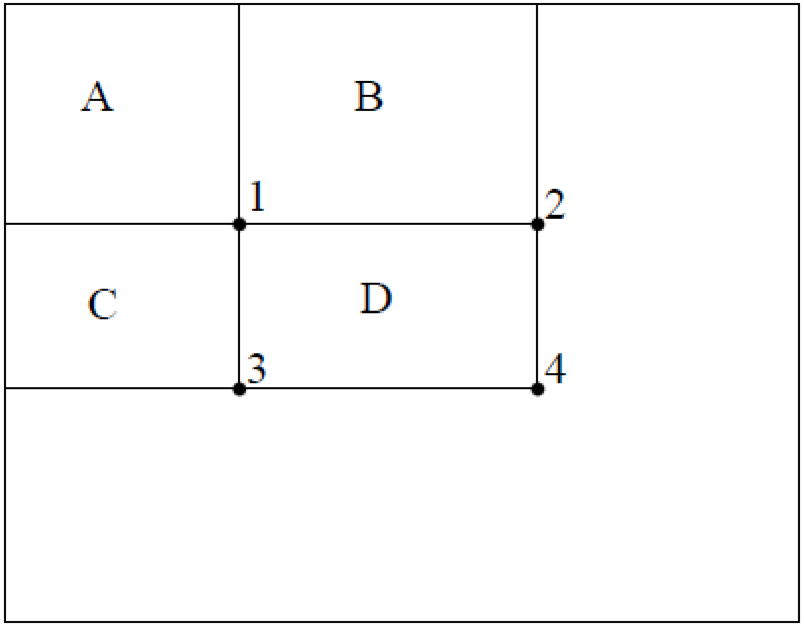
conf (test)
conf (train)
noconf (test)
noconf (train)
50
45
40
35
30
25
20
15
10
5
0
error (%)
1
10
100
1000
10000
rounds of boosting

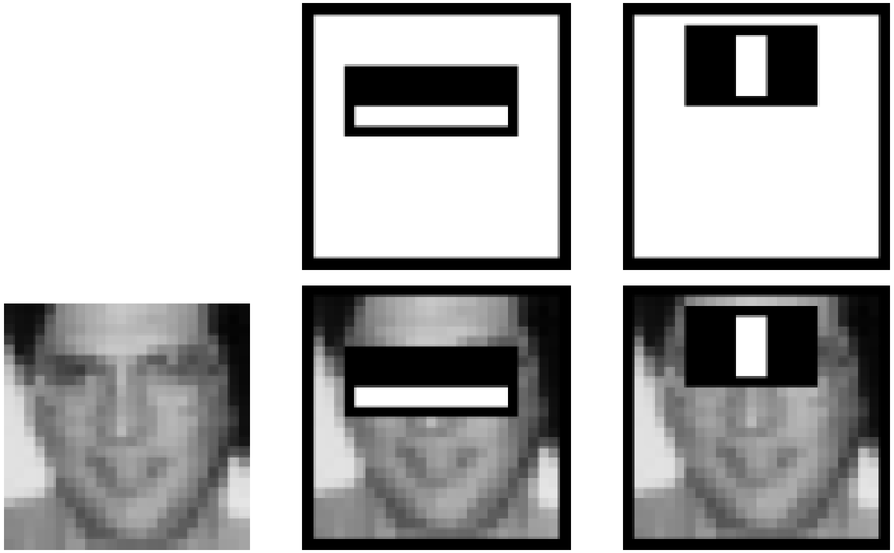
Adaboost – 应用

- 训练样本

Adaboost – 应用

- 特征计算



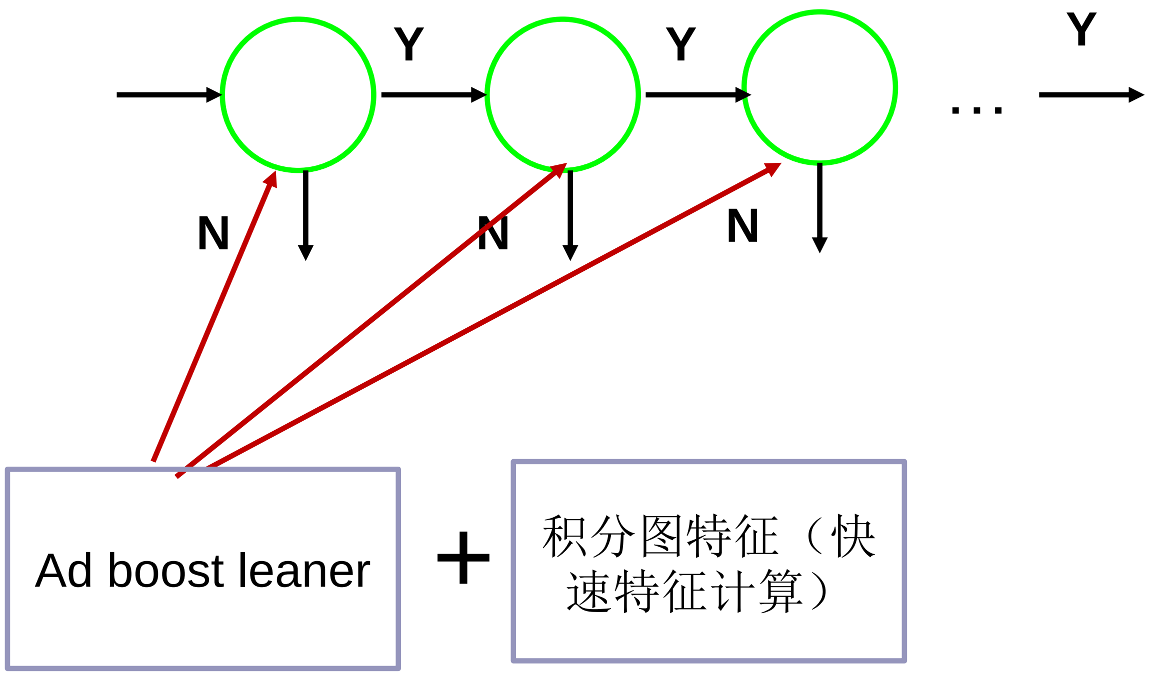


Adaboost – 应用

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i \left| h_j(x_i) - y_i \right|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

Adaboost – 应用

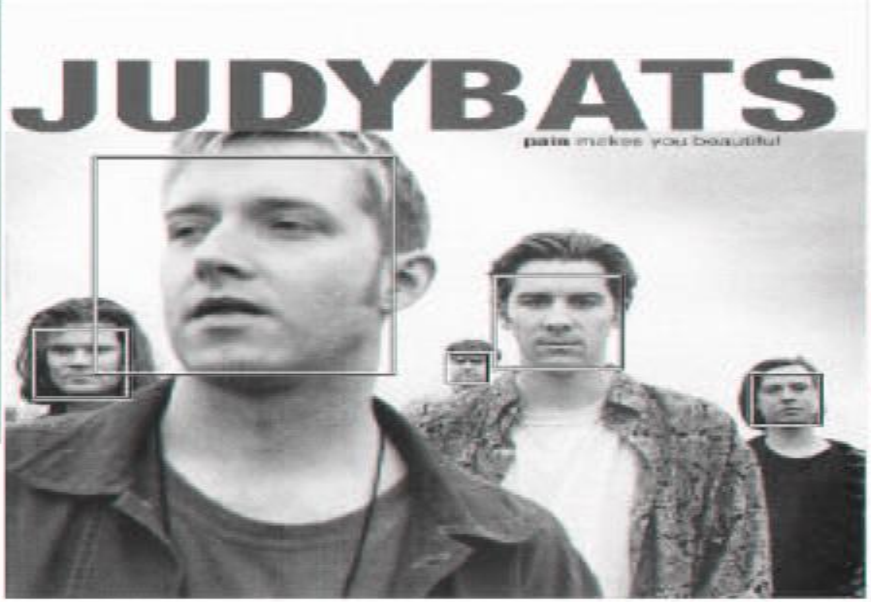
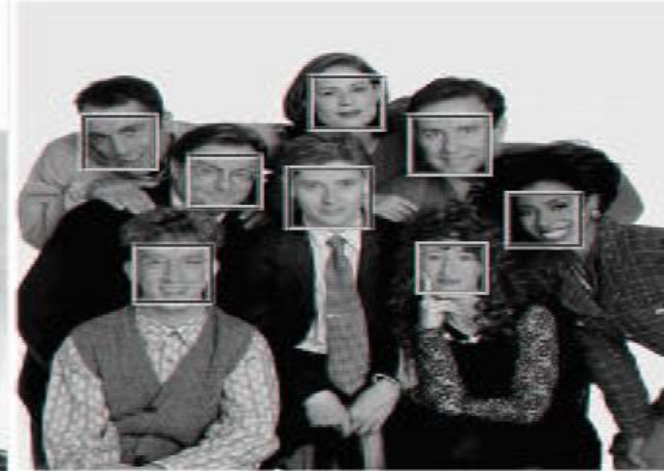
- Cascade 分类器



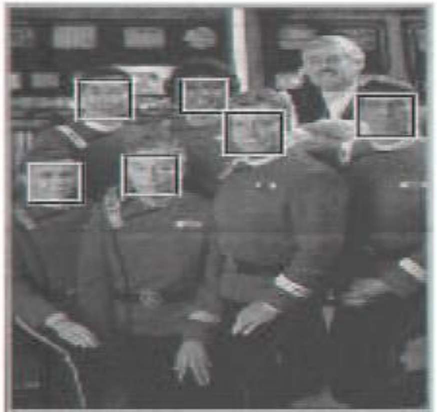
Adaboost – 应用

- 结果

Detector \ False detections	10	31	50	65	78	95	110	167	422
Viola-Jones	78.3%	85.2%	88.8%	89.8%	90.1%	90.8%	91.1%	91.8%	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	-	90.1%	89.9%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-	-	-



JUDYBATS
pain makes you beautiful





Thank you