# 主要内容

### Boosting方法

Boosting 本意----通过增压,加大发动机功率引申—提升分类器性能

AdaBoost 最为广泛, Adaptive Boosting



## AdaBoost — Adaptive Boosting



Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119–139, 1997.

2003年, Schapire 和 Freund 被授予"the Godel Prize"

-- one of the most prestigious awards in theoretical computer science

# 基于AdaBoost算法的强分类器训练



輸入: (1) 训练样本集
$$\mathcal{D} = \{(x_i, y_i) | i = 1, ..., N\}$$

$$y_i \in \{-1, +1\}$$

$$其中 \begin{cases} y_i = -1, 训练样本x_i 为负样本 \\ y_i = +1, 训练样本x_i 为正样本 \end{cases}$$

- (2)弱分类器的学习算法L
- (3)弱分类器的数目M

注意:此处的弱分类器模型中常用的就是"决策树树状",也就是只使用了一个特征生成的决策树.

输出:一个由M个弱分类器构成的强分类器

#### 训练过程:



### A. 初始化训练样本 $x_i$ 权重 $\mathcal{D}(i)$ i=1,...,N

$$(1)$$
若正负样本数目一致,则 $\mathcal{D}_1(i)$ - $\frac{1}{N}$ 

(2)若正负样本数目分别为
$$N_+, N_-$$
,则
$$\begin{cases} E样本 \mathcal{D}_1(i) = \frac{1}{2N_+} \\ \text{负样本} \mathcal{D}_1(i) = \frac{1}{2N_-} \end{cases}$$

**B**. for 
$$m = 1, ..., M$$

- (1) 训练弱分类器 $f_m(x) = L(\mathcal{D}, \mathcal{D}_m) \in \{-1, +1\}$
- (2)估计弱分类器 $f_m(x)$ 的分类错误率 $e_m$

如: 
$$e_m = \frac{1}{2} \sum_{i=1}^{N} \mathcal{D}_m(i) \cdot \left| f_m(x_i) - y_i \right|$$
 [注:  $e_m < 0.5$ ]

$$\boldsymbol{B}$$
. for  $\boldsymbol{m} = 1, ..., \boldsymbol{M}$  (续前)



(3)估计弱分类器
$$f_m(x)$$
的权重 $c_m = \log \frac{1 - e_m}{e_m}$ 

(4)基于弱分类器 $f_m(x)$ 调整各样本权重,并归一化

归一化: 
$$\mathcal{D}_{m+1}(i) \leftarrow \frac{\mathcal{D}_{m+1}(i)}{\sum\limits_{j=1}^{N} \mathcal{D}_{m+1}(j)}$$
  $i=1,...,N$ 

#### C. 强分类器

$$H(x) = \operatorname{sgn}\left[\sum_{m=1}^{M} c_{m} f_{m}(x)\right]$$

**Input:** Data set 
$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\};$$
 Base learning algorithm  $L$ ; Number of learning rounds  $T$ .

#### Process:

- 1.  $\mathcal{D}_1(i) = 1/m$ . % Initialize the weight distribution
- 2. **for**  $t = 1, \dots, T$ :
- 3.  $h_t = L(D, \mathcal{D}_t)$ ; % Train a learner  $h_t$  from D using distribution  $\mathcal{D}_t$
- 4.  $\epsilon_t = \Pr_{x \sim \mathcal{D}_t, y} I[h_t(x) \neq y];$  % Measure the error of  $h_t$
- 5. if  $\epsilon_t > 0.5$  then break
- 6.  $\alpha_t = \frac{1}{2} \ln \left( \frac{1 \epsilon_t}{\epsilon_t} \right)$ ; % Determine the weight of  $h_t$
- 7.  $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) \text{ if } h_t(x_i) = y_i \\ \exp(\alpha_t) \text{ if } h_t(x_i) \neq y_i \end{cases}$

 $\frac{\mathcal{D}_t(i)\exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$  % Update the distribution, where

% enables  $\mathcal{D}_{t+1}$  to be distribution

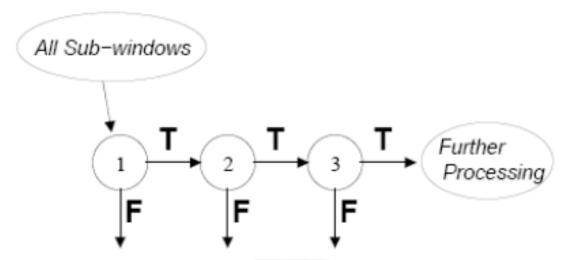
end

**Output:**  $H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)$ 



# 典型应用: real-time face detection Wilfers College of Hobse Normal University





Reject Sub-window

