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

输入： (1) 训练样本集 $\mathcal{D}=\{(\boldsymbol{x}_i, y_i) \mid \boldsymbol{i}=1,\ldots,\boldsymbol{N}\}$

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

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

(2) **弱分类器**的学习算法 $\boldsymbol{L}$

(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** $\mathcal{D}_1(i)\quad i=1,\dots,N$

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

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

B. $\quad for\quad m=1,\dots,M$

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

$B.\quad for\ \ m=1,\ldots,M$ (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

调　整：$\mathcal{D}_{m+1}(i)=\mathcal{D}_m(i)\cdot\exp\left[c_m\cdot 1_{(f_m(x_i)\neq y_i)}\right]$

$$=\begin{cases}\mathcal{D}_m(i) & \text{若} f_m(x_i)=y_i\\ \mathcal{D}_m(i)\cdot\frac{1-e_m}{e_m} & \text{若} f_m(x_i)\neq y_i\end{cases}$$

归一化：$\mathcal{D}_{m+1}(i)\leftarrow\frac{\mathcal{D}_{m+1}(i)}{\sum_{j=1}^{N}\mathcal{D}_{m+1}(j)}\quad i=1,\ldots,N$

$C.$ **强分类器**

$$H(x)=\mathrm{sgn}\left[\sum_{m=1}^{M}c_m f_m(x)\right]$$

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;

Base learning algorithm L;

Number of learning rounds T.

Process:

1. $\mathcal{D}_1(i) = 1/m$. % Initialize the weight distribution
2. **for** $t = 1, \cdots, T$:
3. $h_t = L(D, \mathcal{D}_t)$; % Train a learner h_t from D using distribution $\mathcal{D}_t$
4. $\epsilon_t = \Pr_{x \sim \mathcal{D}_t, y}\, \boldsymbol{I}[h_t(\boldsymbol{x}) \neq y]$; % Measure the error of h_t
5. $\boxed{\textbf{if } \epsilon_t > 0.5 \textbf{ then break}}$
6. $\alpha_t = ½ \ln \left(\frac{1-\epsilon_t}{\epsilon_t}\right)$; % Determine the weight of h_t
7. $\mathcal{D}_{t+1}(i) = \frac{\mathcal{D}_t(i)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) = y_i \\ \exp(\alpha_t) & \text{if } h_t(\boldsymbol{x}_i) \neq y_i \end{cases}$

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

   % Z_t is a normalization factor which

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

8. **end**

$\boxed{\textbf{Output: } H(\boldsymbol{x}) = \text{sign}\left(\Sigma_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)}$

河北师范大学软件学院

Software College of Hebei Normal University

典型应用：*real-time face detection*