

## Boosting

### 基本概念

Boosting算法是一种把若干个分类器整合为一个分类器的方法

- 将多个弱分类器组合成为强分类器
- 三个臭皮匠顶一个诸葛亮

前继方法

- Bootstrapping
- Bagging

### Adaboost算法

#### 算法概述

输入: 训练样例

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

输出: 由M个弱分类器构成的最终分类器G(x)

步骤:

初始化权值分布

$$D_1 = (w_{11}, \dots, w_{1i}, \dots, w_{1n}), w_{1i} = \frac{1}{n}$$

对于m=1,2,...,M:

使用带权值的实例集合D<sub>m</sub>训练模型，得到弱分类器:

$$G_m(x) : x \rightarrow y$$

计算G<sub>m</sub>(x)在训练集上的误差率

$$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^n w_{mi} I(G_m(x_i) \neq y_i)$$

计算 $G_m(x)$ 的系数

$$a_m = \frac{1}{2} \ln \frac{1 - e_m}{e_m}$$

这个地方用模型的整体误差来衡量弱分类器在最终分类器中的权重。

更新训练样例的权值分布，为下一轮迭代做准备

$$D_{m+1} = (w_{m+1,1}, \dots, w_{m+1,i}, \dots, w_{m+1,n})$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} \exp(-a_m y_i G_m(x_i))$$

$Z_m$ 是规范化因子:

$$Z_m = \sum_{i=1}^n w_{mi} \exp(-a_m y_i G_m(x_i))$$

$\exp(-a_m y_i G_m(x_i))$ 这个部分，当分类正确时，整体 $<1$ ；错误时，整体 $>1$ 。意思是，当样例分类错误，我们加大它的权重，以便在后面的迭代中更受重视。相应的，降低分类正确的样例的权重。进行了M轮迭代之后，产出了M个弱分类器，将他们组合起来:

$$f(x) = \sum_{i=1}^m a_m G_m(x)$$

## 算法特点

- adaboost是一种有很高精度的分类器
- 可以使用各种方法构建子分类器，adaboost算法提供的是框架
- 当使用简单分类器时，计算出的结果是可以理解的。而且弱分类器构造极其简单
- 简单，不用做特征筛选
- 不用担心overfitting!

## 参考资料

- [百度技术博客](#)

- [技术博客](#)