

ensemble (昂桑宝) learning

做实际任务必用

弱学习器：精准度仅比随即猜测高一点点

根据个体学习器生成方式，集成学习分成两大类

一、Boosting族

是一族可将弱学习器提升为强学习器的算法

因为个体学习器之间存在强依赖关系，可串行从生成的序列化方法

序列化方法：每个方法是基于上一个方法产生的

- 先从原始数据集训练出一个基学习器（可能用决策树）
- 根据基学习器表现，调整训练样本分布（把做的不好的样本权重调大），再训练下一个基学习器（不断辅助修正之前的）
- 检查当前基分类器是否比随机猜测好，不满足则终止
- 直到训练出事先指定的T个基学习器，再加权结合

AdaBoost算法

基学习器线性组合

$$H(\mathbf{x}) = \sum_{t=1}^T \alpha_t h_t(\mathbf{x})$$

<p>初始化样本权重分布。 基于分布 \mathcal{D}_t 从数据集 D 中训练出分类器 h_t。 估计 h_t 的误差。 确定分类器 h_t 的权重。 更新样本分布，其中 Z_t 是规范化因子，以确保 \mathcal{D}_{t+1} 是一个分布。</p>	<p>输入：训练集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$; 基学习算法 \mathcal{L}; 训练轮数 T。</p> <p>过程：</p> <pre>1: $\mathcal{D}_1(\mathbf{x}) = 1/m$. 2: for $t = 1, 2, \dots, T$ do 3: $h_t = \mathcal{L}(D, \mathcal{D}_t)$; 4: $\epsilon_t = P_{\mathbf{x} \sim \mathcal{D}_t}(h_t(\mathbf{x}) \neq f(\mathbf{x}))$; 5: if $\epsilon_t > 0.5$ then break 6: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$; 7: $\mathcal{D}_{t+1}(\mathbf{x}) = \frac{\mathcal{D}_t(\mathbf{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\mathbf{x}) = f(\mathbf{x}) \\ \exp(\alpha_t), & \text{if } h_t(\mathbf{x}) \neq f(\mathbf{x}) \end{cases}$ $= \frac{\mathcal{D}_t(\mathbf{x}) \exp(-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))}{Z_t}$ 8: end for</pre> <p>输出： $H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$</p>
--	---

图 8.3 AdaBoost算法

- 替代0/1损失函数作为优化目标

最小化指数损失函数(exponential loss function) [Friedman et al., 2000]

$$\ell_{\text{exp}}(H \mid \mathcal{D}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[e^{-f(\mathbf{x})H(\mathbf{x})}] .$$

记前 t 个弱学习器的集成为 $H_t(\mathbf{x}) = \sum_{i=1}^t \alpha_i h_i(\mathbf{x})$ ，该阶段优化目标为：

$$\ell_{\text{exp},t} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[e^{-y(\mathbf{x})H_t(\mathbf{x})} \right]. \quad (3)$$

sign()函数是符号函数

证最优错误率

二、Bagging

Bootstrap (鞋带) AGGREGATING

个体学习器之间没有强依赖关系、可同时生成的并行化方法

三、结合策略

结合方法：投票法、平均法、学习法

多样性的度量目前并不能反映集成学习的效果

不同任务上需要的多样性不同

增强多样性策略：

- 数据样本扰动：只对不稳定分类器有用（暂时两个：决策树，神经网络）

稳定：SVM，朴素贝叶斯，线性分类器