# 第 24 章 自适应提升

## 24.1 提升的动机

**命题 24.1.1 (动机：AdaBoost 的比喻)**

- 学生（弱学习器）：简单的假设 $g_t$，例如垂直或水平直线。
- 班级（强学习器）：复杂的最终假设 $G$，例如图中的黑色曲线。
- 教师（**AdaBoost**）：一种巧妙的算法，通过对样本权重的动态调整，引导学生把注意力集中在当前最困难、最关键的例子上，从而逐步提升整体性能。

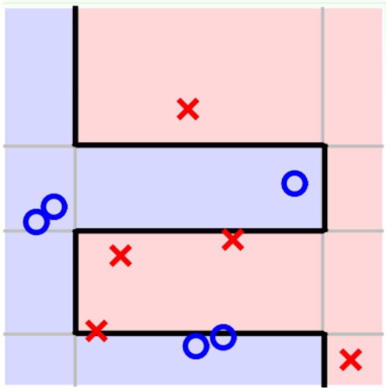

图 24.1.1: AdaBoost 的示意图

## 24.2 重加权带来多样性

**命题 24.2.1 (Bootstrap 等价于样本重加权)**

设原始训练集为 $\mathcal{D}=\{(x_n,y_n)\}_{n=1}^N$。

**1.** 重加权视角

一次 Bootstrap 抽样可视为给每个样本 $(x_n,y_n)$ 赋予一个非负权重 $u_n$，表示该样本在此次抽样中被选中的次数。例如，若抽样结果为

$$\mathcal{D}_t=\{(x_1,y_1),(x_1,y_1),(x_2,y_2),(x_4,y_4)\},$$

则对应权重为

$$u_1=2,\quad u_2=1,\quad u_3=0,\quad u_4=1.$$

**2.** 加权经验误差

任意假设 $h$ 的加权误差为

$$E_{in}^u(h)=\sum_{n=1}^N u_n\cdot\mathbb{I}[y_n\neq h(x_n)].$$

**3.** 加权基学习算法

- 加权 SVM（通过求解对偶二次规划问题 $E_{in}^{u} \propto C\sum_{n=1}^{N} u_n \hat{\text{err}_{SVM}}$）：将软间隔约束调整为
$$0 \le \alpha_n \le Cu_n.$$
- 加权逻辑回归（通过 SGD 求解 $E_{in}^{u} \propto u_n \text{err}_{CE}$）：在 SGD 步骤中，按权重 $u_n$ 对样本进行概率采样，等价于最小化加权交叉熵
$$\sum_{n=1}^{N} u_n \cdot \ell_{\text{CE}}\big(y_n, h(x_n)\big).$$

结论　Bootstrap 抽样等价于样本重加权学习，这一视角统一了 Bagging 与加权基学习算法，并可自然推广至类别加权学习。

**命题 24.2.2 (通过重加权提升 Bagging 多样性)**

设第 $t$ 轮基学习器为 $g_t$，样本权重为 $u_n^{(t)}$，定义加权误差
$$\epsilon_t = \frac{\sum_{n=1}^{N} u_n^{(t)}\, \mathbb{I}[y_n \neq g_t(x_n)]}{\sum_{n=1}^{N} u_n^{(t)}}.$$

目标　构造下一轮权重 $u_n^{(t+1)}$，使新基学习器 $g_{t+1}$ 与 $g_t$ 足够多样。

最优重加权策略

为实现多样性，需使 $g_t$ 在新权重 $u_n^{(t+1)}$ 下的加权正确率与错误率相等（即 $\sum_{\text{正确}} u_n^{(t+1)} = \sum_{\text{错误}} u_n^{(t+1)}$），令
$$u_n^{(t+1)} = u_n^{(t)} \cdot \begin{cases} \sqrt{\dfrac{\epsilon_t}{1-\epsilon_t}}, & y_n = g_t(x_n)\ (\text{正确样本}), \\ \sqrt{\dfrac{1-\epsilon_t}{\epsilon_t}}, & y_n \neq g_t(x_n)\ (\text{错误样本}). \end{cases}$$

等价地，可统一写为
$$u_n^{(t+1)} = u_n^{(t)} \cdot \exp\big(-\alpha_t\, y_n g_t(x_n)\big), \quad \alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right).$$

性质

- 加权正确率与错误率相等：$\sum_{\text{正确}} u_n^{(t+1)} = \sum_{\text{错误}} u_n^{(t+1)}$；
- $g_t$ 在新的权重下不再是加权误差最小的假设，从而强制 $g_{t+1}$ 与之不同，实现多样性。

**例题 24.1 选择题：AdaBoost 的样本权重更新**

在 AdaBoost 算法中，初始样本权重均为 $u_n^{(1)} = \frac{1}{4}$（共 4 个样本）。第一个分类器 $g_1$ 错误分类了样本 1，正确分类了其他 3 个样本。则更新后 $\frac{u_1^{(2)}}{u_2^{(2)}}$ 的值为：

1) 4
2) 3
3) $\frac{1}{3}$
4) $\frac{1}{4}$

**解答**　正确选项为 $\boxed{2}$。

1. 计算错误率：分类器 $g_1$ 错误分类 1 个样本，总样本数为 4，因此错误率：$\epsilon_1 = \frac{1}{4}$。

2. 根据 AdaBoost 公式，分类器权重 $\alpha_1$ 为：
$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right) = \frac{1}{2}\ln 3$$
3. 权重更新公式：
   - 错误样本：$u_1^{(2)} = u_1^{(1)} \cdot \exp\left(-\alpha_1 \cdot (-1) \cdot (+1)\right) = \frac{1}{4} \cdot \exp\left(\alpha_1\right)$
   - 正确样本：$u_2^{(2)} = u_2^{(1)} \cdot \exp\left(-\alpha_1 \cdot (+1) \cdot (+1)\right) = \frac{1}{4} \cdot \exp\left(-\alpha_1\right)$
4. 权重比：$\frac{u_1^{(2)}}{u_2^{(2)}} = \frac{\exp(\alpha_1)}{\exp(-\alpha_1)} = \exp\left(2\alpha_1\right) = \exp\left(\ln 3\right) = 3$。 ■

## 24.3 自适应提升算法

**命题 24.3.1 (最优重加权因子：提升多样性的缩放机制)**

设当前基学习器 $g_t$ 的加权错误率为
$$\epsilon_t = \frac{\sum\limits_{n=1}^{N} u_n^{(t)}\, \mathbb{I}[y_n \neq g_t(x_n)]}{\sum\limits_{n=1}^{N} u_n^{(t)}}.$$
最优缩放因子定义缩放因子
$$\diamondsuit_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}},$$
并对样本权重进行如下缩放：
$$u_n^{(t+1)} = u_n^{(t)} \cdot \begin{cases} \frac{1}{\diamondsuit_t}, & y_n = g_t(x_n)\ (\text{正确样本}), \\ \diamondsuit_t, & y_n \neq g_t(x_n)\ (\text{错误样本}). \end{cases}$$
物理意义

- 错误样本权重 放大 $\diamondsuit_t$ 倍；
- 正确样本权重 缩小 $\frac{1}{\diamondsuit_t}$ 倍；
- 当 $\epsilon_t \leq \frac{1}{2}$ 时，$\diamondsuit_t \geq 1$，即错误样本权重被放大，正确样本权重被缩小，使下一轮基学习器聚焦于先前错误区域，获得多样性。

等价性　该缩放机制等价于“最优重加权”(optimal re-weighting)，即：
   - 错误样本的权重更新与 $(1-\epsilon_t)$ 成正比；
   - 正确样本的权重更新与 $\epsilon_t$ 成正比。

结论　通过“放大错误、缩小正确”的缩放因子，Bagging 可进化出更丰富的假设空间，提升集成性能。

**算法 24.3.1:** 自适应提升算法 (AdaBoost)

**输入:** 训练集 $\mathcal{D}=\{(x_n,y_n)\}_{n=1}^{N}$；基学习算法 $\mathcal{A}$；迭代轮数 $T$

**输出:** 最终分类器 $G(x)$

**初始化权重**

$$u_n^{(1)}=\frac{1}{N},\quad n=1,\dots,N.$$

**for** $t=1$ ***to*** $T$ **do**

**训练弱分类器**

调用 $\mathcal{A}$ 在加权数据 $(\mathcal{D},u^{(t)})$ 上学习，得到

$$g_t=\mathcal{A}(\mathcal{D},u^{(t)}),$$

其中 $\mathcal{A}$ 最小化 $u^{(t)}$-加权 0/1 误差。

**计算加权错误率**

$$\varepsilon_t=\frac{\sum\limits_{n:\,y_n\neq g_t(x_n)} u_n^{(t)}}{\sum\limits_{n=1}^{N} u_n^{(t)}}.$$

**更新样本权重**

$$u_n^{(t+1)}=u_n^{(t)}\cdot\begin{cases}\sqrt{\dfrac{1-\varepsilon_t}{\varepsilon_t}}, & y_n\neq g_t(x_n),\\[2ex] \sqrt{\dfrac{\varepsilon_t}{1-\varepsilon_t}}, & y_n=g_t(x_n).\end{cases}$$

再归一化使 $\sum_n u_n^{(t+1)}=1$。

**计算投票权重**

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right).$$

**返回最终分类器**

$$G(x)=\operatorname{sign}\Big(\sum_{t=1}^{T}\alpha_t\,g_t(x)\Big).$$

**角色形象化**

- 弱学习器（weak base learning algorithm）$\mathcal{A}$：**Student**；
- 重加权机制（optimal re-weighting factor$\diamondsuit_t=\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$）：**Teacher**；
- 线性聚合权重（linear aggregation）$\alpha_t$：**Class** 。

**命题 24.3.2 (AdaBoost 的理论保证)**

由 VC 界可得

$$E_{\text{out}}(G)\leq E_{\text{in}}(G)+O\left(\sqrt{\underbrace{O\big(d_{\text{vc}}(\mathcal{H})\cdot T\log T\big)}_{d_{\text{vc}}\text{ of all possible }G}\cdot\frac{\log N}{N}}\right),$$

其中 $d_{\text{vc}}(\mathcal{H})$ 为弱假设空间 $\mathcal{H}$ 的 VC 维，$T$ 为迭代次数。

两项可控

- 第一项为 $E_{\text{in}}(G)$：若每轮弱学习器满足 $\varepsilon_t \le \varepsilon < \frac{1}{2}$（略优于随机猜测），则经 $T = O(\log N)$ 轮迭代后，$E_{\text{in}}(G) = 0$。
- 第二项为复杂度：所有可能 $G$ 对应的整体 $d_{\text{vc}}$ 随迭代次数 $T$ "缓慢"增长。

**Boosting 视角总结** 若弱学习器 $\mathcal{A}$ 始终"略优于随机"（即 $\varepsilon_t \le \varepsilon < \frac{1}{2}$），则 AdaBoost 结合 $\mathcal{A}$ 可将其"强化"为强学习器，最终实现：

$$E_{\text{in}}(G) = 0 \quad \text{且} \quad E_{\text{out}}(G) \text{ 很小}$$

## 24.4 AdaBoost 实战

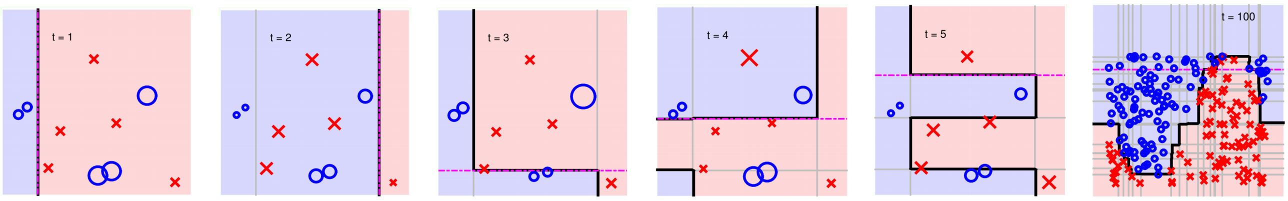


图 24.4.1: AdaBoost 实际应用

**命题 24.4.1 (AdaBoost-Stump 的实际应用：全球首个实时人脸检测系统)**

核心模型 采用 **AdaBoost-Stump**：在 $24 \times 24$ 像素图像中，从 162,336 种候选特征里，通过 AdaBoost-Stump 选出关键 patch，并进行线性聚合。

两大贡献

- 特征选择：AdaBoost-Stump 自动挑选最具判别力的 Haar-like 特征（决策桩），极大降低特征维度。
- 效率优化：对线性聚合器 $G$ 进行改造，可在早期阶段快速排除非人脸区域，实现实时检测。

结论 AdaBoost-Stump 同时具备高效的特征选择与聚合能力，成为首个可在实际场景中运行的实时人脸检测算法。

**例题 24.2 选择题：AdaBoost - Stump 特征使用数量**

现有一个规模为 9876 的数据集，其中样本的特征向量 $\mathbf{x}_n \in \mathbb{R}^{5566}$（即每个样本有 5566 维特征）。运行 AdaBoost - Stump 算法进行 1126 次迭代后，集成分类器 $G$ 实际有效使用的 $\mathbf{x}$ 中不同特征的数量范围是？

1) $0 \le$ 数量 $\le 1126$
2) $1126 <$ 数量 $\le 5566$
3) $5566 <$ 数量 $\le 9876$
4) $9876 <$ 数量

**解答** 正确选项为 $\boxed{1}$。AdaBoost - Stump（AdaBoost 树桩）作为弱分类器，每次迭代构建时仅依据一个特征做决策。进行 1126 次迭代，每次最多用 1 个不同特征（也可能重复选同个特征），所以有效使用的不同特征数最多 1126，最少 0，满足 $0 \le$ 数量 $\le 1126$。 ■

## 24.5 总结

笔记 [自适应提升]

- 提升的动机：将多个弱假设集成为强假设。
- 重加权带来多样性：放大错分样本权重，缩小正确样本权重。
- 自适应提升算法：理论保证“两人智慧胜一人”。
- AdaBoost 实战：AdaBoost-Stump 既实用又高效。

笔记 [总体结论] 自适应提升（AdaBoost）通过动态调整样本权重引导弱学习器聚焦关键样本，将多个弱假设集成为强假设。其核心机制是重加权带来多样性——放大错分样本权重、缩小正确样本权重，使新基学习器与先前模型形成差异。算法具备理论保证，若弱学习器略优于随机猜测，经迭代可实现训练误差为 0 且泛化误差较小。实战中，AdaBoost-Stump 展现出高效的特征选择与聚合能力（如实时人脸检测），但需注意迭代次数对复杂度的影响。作为集成学习的重要手段，它通过巧妙的加权与聚合策略，有效提升了模型性能，是增强学习效果的实用方法。