

---

## 分类问题视角下的 AdaBoost 算法

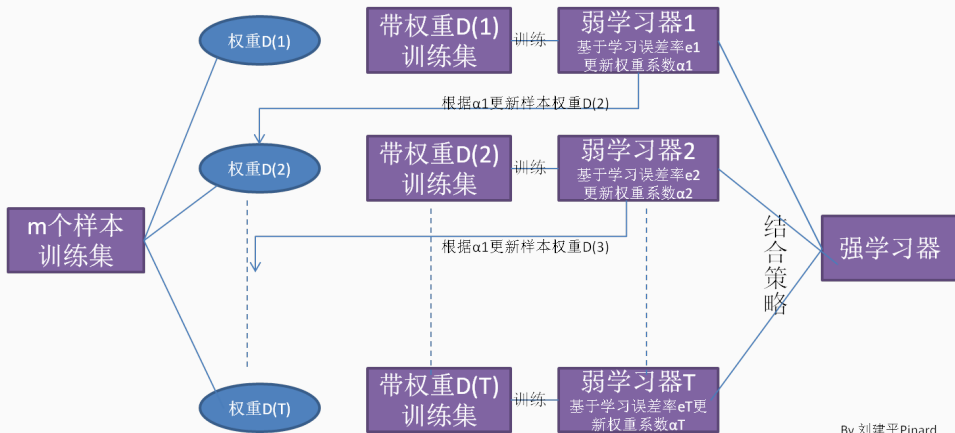
December 7, 2025

## AdaBoost 模型结构

---

- AdaBoost 是一种典型的提升（Boosting）型集成学习框架，其核心思想是：**通过多轮训练、聚焦难样本，把一群“弱学习器”提升为一个“强学习器”。**
- 在每一轮中，AdaBoost 都会为总模型增加一个新的学习器，指导模型的弱学习器个数达到预先指定的值。
- 训练新学习器时，根据上一轮的推理结果在同一训练集上**重新分配样本权重**，使新的弱学习器更加关注上一轮中被分错或“难学”的样本。
- 各轮得到的弱学习器本身能力都比较弱，但在最后通过加权组合（加权投票或加权求和），形成一个整体性能更高、泛化能力更强的强学习器。
- AdaBoost 不限定弱学习器的具体形式（如决策树桩、小深度决策树等），因此可以看作一个**通用的、可移植的集成学习框架。**

# AdaBoost 模型结构图



By 刘建平Pinard

Figure 1. AdaBoost 模型结构图

# AdaBoost 模型结构

---

## 计算流程

## 错误率计算规则

假设训练集包含  $m$  个样本  $\{(x_i, y_i), \dots, (x_m, y_m)\}$ 。

不失一般性地，假设现在训练已经进行到第  $t$  轮，将要训练第  $t$  个弱学习器。

记第  $t$  个学习器在样本  $x$  上的预测结果为  $h_t(x)$ 。

设训练第  $t$  个弱学习器所使用的样本权重为

$$D_t = \{\omega_{t1}, \dots, \omega_{tm}\}, \quad \sum_{i=1}^m \omega_{ti} = 1.$$

则第  $t$  轮弱学习器的加权错误率为

$$\varepsilon_t = \sum_{i=1}^m \omega_{ti} \mathbf{1}\{h_t(x_i) \neq y_i\}.$$

# 样本加权规则

第  $t$  轮 ( $t = 1, \dots, T$ ):

1. 计算系数

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}, \quad 0 < \beta_t < 1.$$

2. 先得到更新后的未归一化权重

$$\tilde{\omega}_{t+1,i} = \omega_{ti} \beta_t^{1 - \mathbf{1}\{h_t(x_i) \neq y_i\}}, \quad i = 1, \dots, m.$$

3. 再将其归一化, 得到下一轮的权重分布

$$\omega_{t+1,i} = \frac{\tilde{\omega}_{t+1,i}}{\sum_{j=1}^m \tilde{\omega}_{t+1,j}}.$$

第  $t$  轮 ( $t = 1, \dots, T$ ): 直观理解:

$$\begin{cases} h_t(x_i) = y_i \Rightarrow \tilde{\omega}_{t+1,i} = \omega_{ti} \beta_t & (\text{分对: 权重减小}) \\ h_t(x_i) \neq y_i \Rightarrow \tilde{\omega}_{t+1,i} = \omega_{ti} & (\text{分错: 权重不变, 归一化后相对增大}) \end{cases}$$

这意味这模型会不断给难以分类的样本点增加权重, 而这些难分类的点可能是数据中的噪声/离群点, 可能导致模型在中后期过拟合。



## 收敛性

---

设:

- 第  $t$  轮弱学习器的加权错误率为  $\varepsilon_t$ ;
- 最终强分类器为  $h_f$ , 其在训练分布  $D$  下的错误率为

$$\varepsilon = \Pr_{i \sim D}[h_f(x_i) \neq y_i].$$

## 训练误差上界

$$\varepsilon \leq 2^T \prod_{t=1}^T \sqrt{\varepsilon_t(1 - \varepsilon_t)}.$$

引入第  $t$  轮的“优势” (edge) : 比随机猜测强的部分

$$\gamma_t = \frac{1}{2} - \varepsilon_t,$$

则上界可以改写为

$$\varepsilon \leq \prod_{t=1}^T \sqrt{1 - 4\gamma_t^2} = \exp\left(-\sum_{t=1}^T \text{KL}\left(\frac{1}{2} \parallel \frac{1}{2} - \gamma_t\right)\right) \leq \exp\left(-2\sum_{t=1}^T \gamma_t^2\right).$$

**特殊情形：**若所有弱学习器的错误率都相同， $\varepsilon_t = \frac{1}{2} - \gamma$  ( $\gamma > 0$ )，则

$$\varepsilon \leq (1 - 4\gamma^2)^{T/2} = \exp(-T \cdot \text{KL}(\frac{1}{2} \parallel \frac{1}{2} - \gamma)) \leq \exp(-2T\gamma^2).$$

**结论：**只要每一轮的弱学习器都略好于随机猜测 ( $\gamma_t > 0$ )，AdaBoost 在训练集上的错误率会随轮数  $T$  **指数级下降**。

# 收敛性

---

## 收敛性证明

# 收敛性证明

引理：训练误差界受限于归一化因子之积。

$$\frac{1}{m} \sum_{i=1}^m \exp(-y_i f(x_i)) = \prod_{t=1}^T Z_t.$$

其中  $Z_t = \sum_{i=1}^m \omega_{ti} \exp(-\alpha_t y_i h_t(x_i))$ 。

证明思路：

1. 递推关系：  $\omega_{t+1,i} = \frac{\omega_{ti} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
2. 展开得到：  $\omega_{T+1,i} = \frac{1}{m \prod_{t=1}^T Z_t} \exp(-y_i \sum_{t=1}^T \alpha_t h_t(x_i))$
3. 由  $\sum \omega_{T+1,i} = 1$ ，可得上述等式。
4. 由于  $\mathbf{1}\{H(x) \neq y\} \leq \exp(-yH(x))$ ，故误差  $\leq \prod Z_t$ 。

## 收敛性证明 (续)

计算  $Z_t$  的极小值:

$$Z_t = \sum_{y_i=h_t(x_i)} \omega_{ti} e^{-\alpha_t} + \sum_{y_i \neq h_t(x_i)} \omega_{ti} e^{\alpha_t} = (1 - \varepsilon_t) e^{-\alpha_t} + \varepsilon_t e^{\alpha_t}$$

对  $\alpha_t$  求导并令其为 0, 可得最优权重:

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t}$$

代回  $Z_t$  表达式:

$$Z_t = 2\sqrt{\varepsilon_t(1 - \varepsilon_t)} = \sqrt{1 - 4\gamma_t^2}$$

从而得证:

$$\varepsilon_{\text{train}} \leq \prod_{t=1}^T \sqrt{1 - 4\gamma_t^2} \leq \exp(-2 \sum_{t=1}^T \gamma_t^2)$$

## 泛化误差

---



- **现象：**在许多实验中观察到，即使训练误差已经降为 0，继续增加弱分类器数量  $T$ ，测试误差不仅没有上升（过拟合），反而进一步下降。
- **传统解释失效：**传统的 VC 维理论认为模型复杂度随  $T$  增加，泛化界应变差。
- **间隔理论（Margin Theory）：**AdaBoost 在训练误差为 0 后，继续训练会使得样本的**分类间隔（Margin）**不断增大。

$$\text{margin}(x, y) = \frac{y \sum_t \alpha_t h_t(x)}{\sum_t \alpha_t}$$

- 更大的间隔意味着更强的鲁棒性和更好的泛化能力。

## 理论进阶

---

## 统计解释：指数损失最小化

AdaBoost 的另一种解释是：**前向分步加法模型** (Forward Stagewise Additive Modeling) 在**指数损失函数**下的优化过程。

- **加法模型：**  $f(x) = \sum_{t=1}^T \alpha_t h_t(x)$
- **损失函数：**  $L(y, f(x)) = \exp(-yf(x))$

**性质：**指数损失是 0/1 损失的一致上界，且处处可微。

$$\mathbf{1}\{y \neq f(x)\} \leq \exp(-yf(x))$$

最小化指数损失等价于最小化分类错误率的上界。这为 AdaBoost 的权重更新规则提供了统计学依据。

## 多分类扩展：SAMME 算法

原始 AdaBoost (AdaBoost.M1) 主要针对二分类问题。对于 MNIST 手写数字识别 (10 分类)，需要进行扩展。

**SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function)** 算法调整了权重更新公式：

$$\alpha_t = \ln \frac{1 - \varepsilon_t}{\varepsilon_t} + \ln(K - 1)$$

其中  $K$  是类别数量 (MNIST 中  $K = 10$ )。

**收敛条件：**只要  $\varepsilon_t < 1 - \frac{1}{K}$  (即弱分类器优于随机猜测)，则  $\alpha_t > 0$ ，算法即可收敛。

## 偏差-方差权衡 (Bias-Variance Tradeoff)

从误差分解的视角对比 Bagging (如随机森林) 与 Boosting (如 AdaBoost):

- **Bagging (并行)**: 通过对训练集重采样训练多个强学习器并取平均。主要降低**方差 (Variance)**, 适合高方差模型 (如完全生长的决策树)。
- **Boosting (串行)**: 通过迭代修正前一轮的错误。主要降低**偏差 (Bias)**, 能将高偏差的**弱学习器** (如决策树桩) 提升为强学习器。随着迭代次数增加, 方差也可能略有下降, 但过度迭代会导致过拟合 (方差上升)。

## 正则化：Shrinkage 策略

为了进一步防止过拟合，AdaBoost 常引入**学习率 (Learning Rate)** 参数  $\nu$  ( $0 < \nu \leq 1$ ):

$$f_t(x) = f_{t-1}(x) + \nu \cdot \alpha_t h_t(x)$$

- **作用：**限制每个弱分类器的贡献，迫使学习过程更加缓慢和稳健。
- **代价：**较小的  $\nu$  通常意味着需要更多的迭代次数  $T$  才能达到同样的训练误差。
- 在实践中，这是调节模型泛化能力最重要的参数之一。

## 总结与展望

---

# AdaBoost 优缺点总结

## 优点：

- **泛化能力强：**在许多问题上不易过拟合（Margin 理论）。
- **参数少：**原始算法几乎无需调参。
- **通用性：**可与任何弱学习器结合。

## 缺点：

- **对噪声敏感：**异常值权重会被过度放大（本次实验重点验证）。
- **串行训练：**难以并行化，训练速度较慢。

**经典应用：** Viola-Jones 人脸检测框架（基于 Haar 特征 + AdaBoost 级联）。



## 任务：手写数字识别

---

训练一个 AdaBoost 分类器, 对手写数字图片进行分类。  
这是一个多分类（十类）问题。

# 任务：手写数字识别

---

方法

## 数据集:

- 使用 MNIST 数据集，按照 8：2 切分训练集和测试集，
- 在 MNIST 测试集和课程提供的手写图片两组数据上分别测试。

## 数据预处理:

- 所有图片转化为黑底白字
- 按照包含该数字的最小正方形进行切割
- 使用 `cv2.resize` 方法将图片缩放至 20x20
- 将数字图片嵌入到 28x28 的纯黑色背景

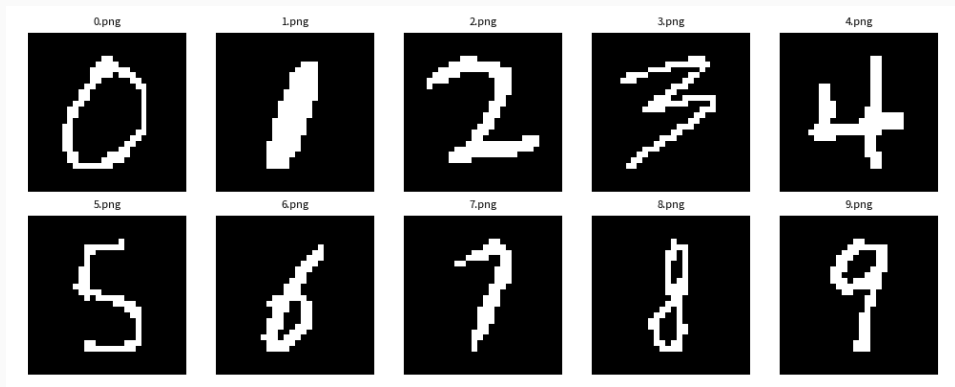


Figure 2.预处理后数据示意图

- 原始图片，reshape 为 (784,) 一维向量
- HOG
- Hu 不变矩

使用 'sklearn.ensemble.AdaBoostClassifier'实现的 AdaBoost 分类器

```
clf = AdaBoostClassifier(  
    estimator=base,  
    n_estimators=model_cfg["n_estimators"],  
    learning_rate=model_cfg["learning_rate"],  
    random_state=model_cfg["random_state"],  
)
```

**Figure 3.**sklearn AdaBoost 分类器实例构建

# 关键代码

```
(machinelearning) flyingbucket@H3C-R5300-G5-A30:~/adaboost$ python main.py --config configs/main_original.json
[Workflow] experiment dir created: experiments/main_original
[Data] Downloading MNIST...
[Data] No noise added.
[MODEL] Using original AdaBoost without BoostMonitor
[Workflow] Training Started...
[Workflow] Training Finished...
[Workflow] Uncompressed model joblib saved to : experiments/main_original/results/model.joblib
[Workflow] compressed and saved to : experiments/main_original/results/model.joblib.xz
[Workflow] compressed model joblib saved to : experiments/main_original/results/model.joblib.xz
[Data] Loading course dataset from: test_data

===Scores on test data of MNIST===

=== Evaluation ===
Accuracy:      0.8727
Precision_macro:0.8758
Recall_macro:  0.8719
F1_macro:      0.8723

===Scores on test data of corse data===

=== Evaluation ===
Accuracy:      0.4000
Precision_macro:0.2083
Recall_macro:  0.4000
F1_macro:      0.2567
```

Figure 4.实验主入口示例



# 任务：手写数字识别

---

## 实验

参数	值
<code>max_depth</code>	2
<code>max_features</code>	0.2
<code>criterion</code>	entropy
<code>random_state</code>	42
<code>n_estimators</code>	1000
<code>learning_rate</code>	0.5

Table 1.训练参数

数据集	准确率	精度 (宏平均)	召回率 (宏平均)	F1 值 (宏平均)
MNIST	0.8727	0.8758	0.8719	0.8723
课程数据集	0.4	0.2083	0.4	0.2567

Table 2.模型在 MNIST 和课程数据集上的测试结果

数据集	准确率	精度 (宏平均)	召回率 (宏平均)	F1 值 (宏平均)
MNIST	0.5051	0.4921	0.4961	0.4904
课程数据集	0.2	0.15	0.2	0.1667

**Table 3.**模型在 MNIST 和课程数据集上的测试结果

参数	值
<b>orientations</b>	9
<b>pixels_per_cell</b>	[2, 2]
<b>cells_per_block</b>	[2, 2]

Table 4.HOG 特征提取参数

数据集	准确率	精度 (宏平均)	召回率 (宏平均)	F1 值 (宏平均)
<b>MNIST</b>	0.9356	0.93596	0.93515	0.93536
<b>课程数据集</b>	0.7	0.5833	0.7	0.6167

Table 5.模型在 MNIST 和课程数据集上的测试结果

# 任务：手写数字识别

---

鲁棒性与误差分析

AdaBoost

## 实验结果与分析：

- **对噪声敏感：** AdaBoost 在干净数据上表现优异，但随着噪声比例增加，准确率下降明显快于 Bagging 类算法（如随机森林）。
- **原因探究：** AdaBoost 的核心机制是关注“难分样本”。高噪声样本往往被视为难分样本，权重被不断放大。
- **权重偏移：** 模型过度关注这些无法正确分类的噪声点（离群点），导致决策边界发生扭曲，从而降低了对正常样本的泛化能力。



谢谢大家！