

---

## 分类问题视角下的 AdaBoost 算法

December 25, 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，直到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**。

# 模型结构图

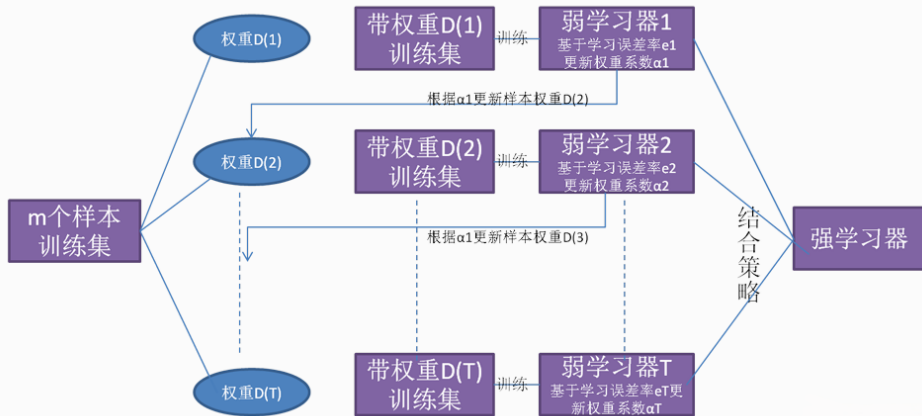


Figure 1. AdaBoost 模型结构图

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

第 1 个学习器直接由初始权重的样本训练即可

不失一般性地，假设现在训练已经进行到第  $t$  轮。 ( $t = 2, \dots, T$ )

记第  $t - 1$  个学习器在样本  $x$  上的预测结果为  $h_{t-1}(x)$ 。

设训练第  $t - 1$  个弱学习器所使用的样本权重为

$$D_t = \{\omega_{t-1,1}, \dots, \omega_{t-1,m}\}, \quad \sum_{i=1}^m \omega_{t-1,i} = 1.$$

则第  $t - 1$  轮弱学习器的加权错误率为

$$\varepsilon_{t-1} = \sum_{i=1}^m \omega_{t-1,i} \mathbf{1}\{h_{t-1}(x_i) \neq y_i\}.$$

第  $t - 1$  个弱学习器的投票权重为

$$\alpha_{t-1} = \frac{1}{2} \left( \frac{1 - \varepsilon_{t-1}}{\varepsilon_{t-1}} \right)$$

## 模型计算流程：样本权重更新

1. 计算系数

$$\beta_{t-1} = \frac{\varepsilon_{t-1}}{1 - \varepsilon_{t-1}}, \quad 0 < \beta_{t-1} < 1.$$

2. 先得到更新后的未归一化权重

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

3. 再将其归一化，得到第  $t$  轮的权重分布

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

4. 由得到的新样本权重训练集训练得到第  $t$  个弱学习器

第  $t$  轮：直观理解：

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

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。

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

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

通过求导计算得到  $Z_t$  的极小值：

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

从而得证：

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

## 总结与展望

---

## 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

---



## 核心目标:

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。

## 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

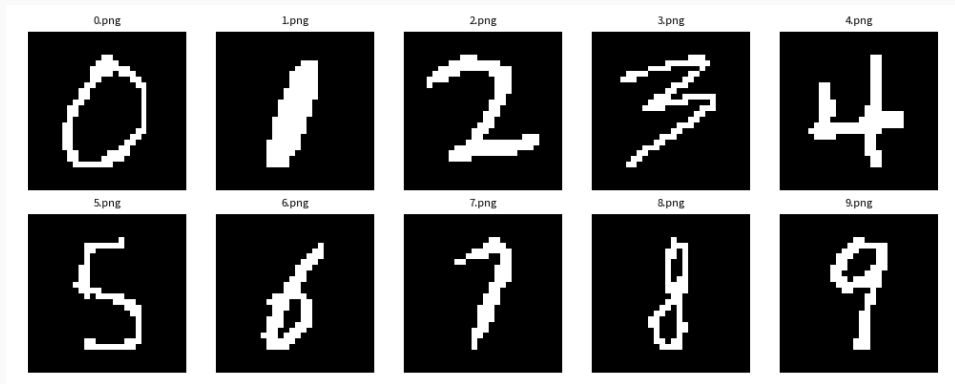


Figure 2.预处理后数据示意图

## 特征提取方法:

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

## 目标:

- 对比原始特征和提取特征的聚类效果, 证明提取特征的有效性
- 提升风格迁移泛化能力

## 聚类与可视化:

- 对不同风格数据集的特征进行聚类 and 可视化, 呈现分布差异

AdaLab 是一个面向研究的 AdaBoost 实验园地，通过端到端的实验框架支持 AdaBoost 算法的过拟合行为、泛化能力与鲁棒性特征的系统性研究。项目提供了 Python 实验框架与 CLI 工具，支持通过 JSON 配置文件驱动实验流程，实现零代码运行实验适用于算法分析、实验复现与对比研究。

- 端到端实验框架从训练、评估到可视化，所有步骤都可以通过 JSON 配置文件驱动。
- AdaBoost 核心机制可观测支持样本权重、噪声样本与干净样本权重、弱分类器  $\alpha$  等的精细监控。
- 鲁棒性研究的数据处理模块包括特征提取（HOG、Hu 不变矩）和多种图像扰动方式（噪声、破坏、变形等）。
- CLI 驱动，零代码运行提供统一的 adalab 命令行工具，适合批量实验管理与脚本化操作。
- 可选配套可视化模块独立的 adalab\_viz 包，用于结构化可视化实验结果。

# AdaLab 使用方法与常见运行模式

```
look 23
) adalab -h
usage: adalab [-h] --config CONFIG [--experiments-dir EXPERIMENTS_DIR]
              [--course-folder COURSE_FOLDER] [--viz | --viz-only]

AdaLab experiment runner (CLI)

options:
  -h, --help            show this help message and exit
  --config CONFIG        Path to json config file
  --experiments-dir EXPERIMENTS_DIR
                        Base directory that stores experiment runs (default:
                        experiments/)
  --course-folder COURSE_FOLDER
                        Course test folder used in evaluation (default:
                        ./data/test_images)
  --viz                  Train + eval + visualize after training (requires
                        use_monitor=true)
  --viz-only             Skip training; load existing experiment results then
                        eval + visualize
```

Figure 3.预处理后数据示意图

# 任务：手写数字识别

---

## 实验结果展示

参数	值
max_depth	3
max_features	0.3
criterion	entropy
random_state	42
n_estimators	500
learning_rate	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	0.9184	0.9196	0.9180	0.9182
课程数据集	0.6	0.4	0.6	0.4667

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	0.5208	0.5017	0.5113	0.5038
课程数据集	0.4	0.2333	0.4	0.2833

Table 3.模型在 MNIST 和课程数据集上的测试结果

参数	值
orientations	9
pixels_per_cell	[4, 4]
cells_per_block	[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)
MNIST	0.958	0.9581	0.9578	0.9579
课程数据集	0.6	0.525	0.6	0.54

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

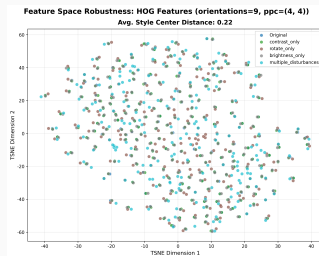
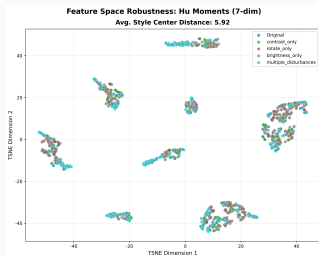
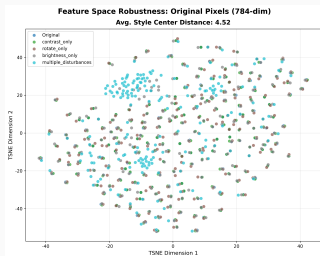
## 泛化能力:MNIST 数据量下的结论

测试集	准确率	平均 Recall
Contrast	0.958	0.960
Rotate	0.958	0.960
Brightness	0.958	0.960
Multiple Disturbances	0.916	0.922

**备注：**对于多个扰动测试集，尽管准确率有所下降（0.916），但大部分类别的 Recall 值较高，尤其是数字“0”、“6”和“8”的 Recall 均维持在较高水平。少数类别（如“1”和“7”）的 Recall 较低，表明其泛化能力受到了影响。

# 泛化能力: 特征提取有效性

不同特征提取方式的特征空间分布:



# 对不同风格测试集的泛化能力

## 泛化原理:

- 通关特征提取忽略与分类无关的特征因素

## 测试原理:

- 在测试集中引入特定扰动以评估模型的泛化能力

## 实现过程:

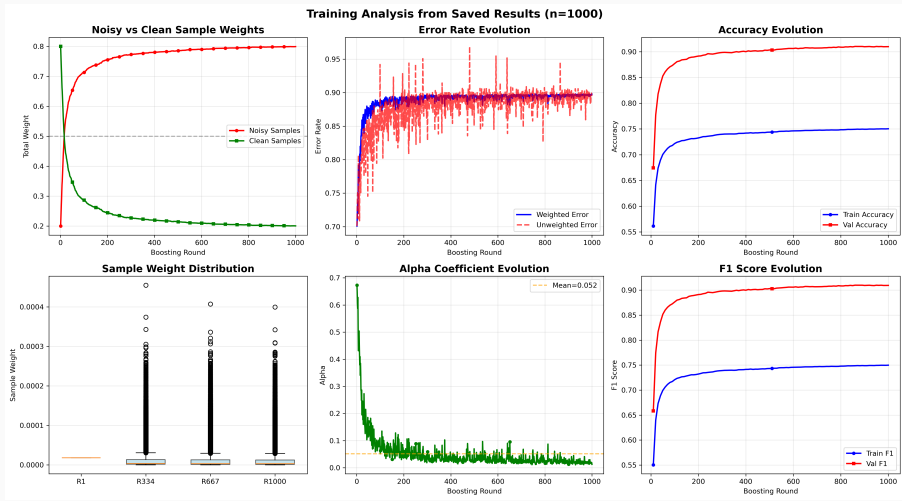
- 调整训练模型的风格参数实现不同风格

### 风格扰动研究分析与结论:

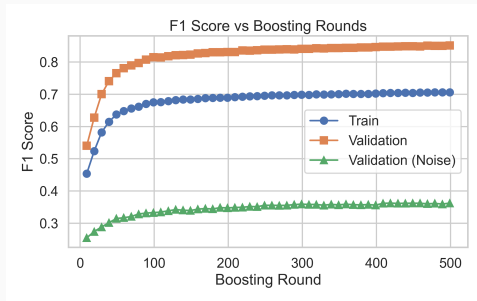
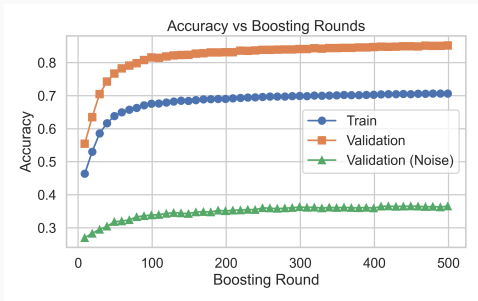
- 加入不同扰动后，特征提取的效果明显
- 合适的特征提取能够有效减少风格本身的影响
- 最终呈现的模型准确率与无扰动数据差距很小



# 鲁棒性：模型对噪声的适应能力



# 鲁棒性：模型对噪声的适应能力



### 噪声研究分析与结论

- 随着迭代次数增多，噪声权重增大导致后续学习器逐渐关注训练集的噪声部分
- 实际训练中，更关注噪声的这部分学习器在最终得到的模型中权重很小
- 训练集噪声对模型训练的影响相对稳定，且无噪声测试集仍呈现较高的准确率

## 误差来源与影响

- 样本数量少，识别结果方差大，准确率波动较大
- 人工噪声无法模拟真实的误差环境, 实验对比更多体现了模型的泛化能力

## 实验结果:

- **噪声适应能力较强**: 噪声研究显示, 训练模型能够有效识别噪声并在合理迭代次数下, 能够减少噪声对模型的影响, 总体分析得出训练模型对噪声的适应能力较强
- **泛化能力强**: 风格扰动对模型训练的干扰不明显, 数据能够直观展现模型的泛化能力强

## 综合评价:

- 训练模型虽然可以保证对纯净样本较高的准确率, 但准确率随噪声增大而下降的现象仍然明显, 训练时需要**尽量避免噪声干扰**
- 特征提取是加强泛化能力的关键, 需要**选择合适的特征提取方式**强化模型的泛化能力

谢谢大家！