第 27 章 梯度提升决策树

27.1 自适应提升决策树

算法 27.1.1: 自适应提升决策树（AdaBoost-DTree）

输入: 数据集 $\mathcal{D}=\{(x_n,y_n)\}_{n=1}^{N}$；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**

重加权数据 依据当前权重 $u^{(t)}$;

训练加权树

$$g_t=\texttt{DTree}\big(\mathcal{D},u^{(t)}\big);$$

计算投票权重

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right),\quad \text{其中}\varepsilon_t=\frac{\sum\limits_{n:\,y_n\neq g_t(x_n)}u_n^{(t)}}{\sum\limits_{n=1}^{N}u_n^{(t)}}.$$

返回最终分类器

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

命题 27.1.1 (加权决策树的黑箱化实现逻辑)

针对无法直接接入样本权重的 C&RT 决策树，可通过随机采样机制间接实现加权学习，具体模式如下：

1. Bagging-DTree 模式

样本权重 u_n 依托 **Bootstrap** 重采样传递：

- 重采样逻辑：样本 (x_n,y_n) 出现在重采样数据集 $\tilde{\mathcal{D}}_t$ 的概率，与权重 u_n 正相关；
- 执行流程：通过 Bootstrap 生成重采样集 $\tilde{\mathcal{D}}_t$ 后，调用 $\texttt{DTree}(\tilde{\mathcal{D}}_t)$ 训练基学习器 g_t。

2. AdaBoost-DTree 模式

第 t 轮样本权重 $u_n^{(t)}$ 借助比例采样（按 $u_n^{(t)}/\sum u_k^{(t)}$ 概率）传递：

- 黑箱兼容：无需修改 DTree 内部逻辑，仅通过采样生成加权分布 $\tilde{\mathcal{D}}_t$;
- 执行流程：$u^{(t)}\xrightarrow{\text{按权采样}}\tilde{\mathcal{D}}_t\xrightarrow{\text{DTree 训练}}g_t$（权重驱动采样构建数据集，再用决策树训练基学习器）。

结论 无论 Bagging 还是 AdaBoost 框架，只要 C&RT 决策树保持”黑箱”（不改动算法内部），均可通过随机采样策略桥接”权重信息”与”决策树训练”，让黑箱模型间接支持加权学习逻辑。

命题 27.1.2 (弱决策树与 AdaBoost-Stump 机制)

1. 过强树的失效风险

在 AdaBoost 框架中，若每棵决策树都以完全生长（不剪枝、不限高）的方式，在全部训练样本 $\{x_n\}$ 上学习，会引发关键问题：

由于完全生长的树可对训练样本实现"完美拟合"(若样本无重复标签)，此时训练误差 $\boldsymbol{E}_{\text{in}}(\boldsymbol{g}_t) = \mathbf{0}$，其加权误差率 ε_t（衡量样本权重下的分类错误）也会趋近于 0。代入 AdaBoost 为单树设计的投票权重公式：

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

当 $\varepsilon_t \to \mathbf{0}$ 时，$\boldsymbol{\alpha}_t \to \infty$。这意味着单个强树会垄断整个集成的投票权，彻底破坏 AdaBoost "多弱学习器协作互补"的核心逻辑，失去集成学习的意义。

2. 弱树的构造策略

为避免强树"独裁"，需主动构造弱决策树（单树性能弱于随机猜测，但多树可通过 AdaBoost 协作），核心手段包含两方面：

- 限制模型复杂度：
  - 对完全生长的树进行后剪枝（**post-pruning**），通过合并/删除节点精简结构；
  - 或直接限制树高（如强制树高 ≤ 1），从根源降低单树拟合能力。
- 控制训练数据：

  训练每棵树前，按样本权重 $\boldsymbol{u}^{(t)}$ 进行加权采样，让单树仅学习部分样本（而非全部数据）。此举进一步削弱单树对全局规律的拟合，强化"多树协作"的必要性。

3. 极限特例：**AdaBoost-Stump**

当树高被严格限制为 **1**（极端剪枝，仅保留一个分裂节点），且以 **0/1** 分类误差（最简单的"杂质度"指标，直接统计分类错误数）作为优化目标时，决策树会退化为决策桩（**decision stump**）——结构极简，仅通过一个特征的阈值划分样本。

此时，

$$\text{AdaBoost-Stump} = \text{AdaBoost 框架} + (\text{树高} \leq 1 \text{ 的 C\&RT 决策树})$$

是 AdaBoost-DTree（基于决策树的 AdaBoost 通用框架）的典型特殊实例。其本质是用"决策桩"这类极致弱学习器，验证 AdaBoost 从"多弱协作"到"强集成"的核心逻辑。

例题 27.1 选择题：AdaBoost - DTree 中 α_t 的可能性分析

运行带采样的 AdaBoost - DTree 算法时，得到决策树 g_t，且 g_t 在采样数据集 $\tilde{\mathcal{D}}_t$ 上实现零误差。关于 g_t 对应的权重 α_t，以下哪种情况可能发生？

1) $\alpha_t < 0$
2) $\alpha_t = 0$
3) $\alpha_t > 0$
4) 以上所有情况

解答 正确选项为 $\boxed{4}$。在 AdaBoost 中，弱学习器权重 α_t 由误差率 ϵ_t 决定，公式为 $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$。虽 g_t 在采样集 $\tilde{\mathcal{D}}_t$ 上零误差，但在带权数据集 $(\mathcal{D}, \mathbf{u}^{(t)})$ 中，加权误差不一定为 0。

- 若加权误差 $\epsilon_t \to 0$，则 $\alpha_t > 0$；
- 若加权误差 $\epsilon_t = 0.5$，则 $\alpha_t = 0$；

- 若加权误差 $\epsilon_t > 0.5$，则 $\alpha_t < 0$。

因此 α_t 可小于、等于、大于 0，所有情况都可能发生。 ■

27.2 AdaBoost 的优化视角

命题 27.2.1 (AdaBoost 的样本权重与投票得分)

在第 t 轮迭代中，样本权重的更新规则如下：

$$u_n^{(t+1)} = u_n^{(t)} \cdot \begin{cases} \exp(\alpha_t), & \text{若分类错误（即} y_n \neq g_t(x_n)\text{）} \\ \exp(-\alpha_t), & \text{若分类正确（即} y_n = g_t(x_n)\text{）} \end{cases}$$

其中，$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，$\varepsilon_t$ 为第 t 轮基分类器 g_t 的分类误差率。

多轮迭代后权重的累积更新从初始权重 $u_n^{(1)}$ 开始，经过 T 轮迭代，样本 (x_n, y_n) 的权重 $u_n^{(T+1)}$ 可通过逐轮指数更新累积推导：

$$u_n^{(T+1)} = u_n^{(1)} \cdot \prod_{t=1}^{T} \exp\bigl(-y_n \alpha_t g_t(x_n)\bigr)$$

若初始时所有样本权重均匀分布（即 $u_n^{(1)} = \frac{1}{N}$，N 为样本总数），则上式可简化为：

$$u_n^{(T+1)} = \frac{1}{N} \cdot \exp\left(-y_n \sum_{t=1}^{T} \alpha_t g_t(x_n)\right)$$

定义累积投票得分为 voting score$(x_n) = \sum\limits_{t=1}^{T} \alpha_t g_t(x_n)$，权重最终形式可统一关联为：

$$u_n^{(T+1)} \propto \exp\bigl(-y_n \cdot \text{voting score}(x_n)\bigr)$$

投票得分与最终输出　AdaBoost 最终的强分类器由基分类器加权投票决定：

$$G(x) = \operatorname{sign}\left(\underbrace{\sum_{t=1}^{T} \alpha_t g_t(x)}_{\text{投票得分（voting score）}} \right)$$

其中，$\sum\limits_{t=1}^{T} \alpha_t g_t(x)$ 是基分类器集合 $\{g_t\}$ 在样本 x 上的投票得分，符号函数 sign$(\cdot)$ 依据投票得分正负输出最终分类标签。 ♠

命题 27.2.2 (投票得分与间隔的 AdaBoost 视角)

将最终分类器写成线性混合模型

$$G(x) = \operatorname{sign}\Bigl(\underbrace{\sum_{t=1}^{T} \alpha_t g_t(x)}_{\text{投票得分（voting score）}} \Bigr) = \operatorname{sign}\Bigl(\underbrace{\sum_{i=1}^{d} \omega_i \phi_i(x)}_{\text{投票得分（voting score）}} \Bigr)$$

带符号的未归一化间隔　对训练样本 (x_n, y_n) 定义

$$\text{margin}_n = \frac{y_n \cdot (w^\top \phi(x_n) + b)}{||w||} \propto y_n \cdot \big(\text{voting score}\,\big),$$

目标：$\text{margin}_n > 0$ 且越大越好 $\Leftrightarrow$ $y_n\big(\text{voting score}\,\big)$ 越大越好 $\Leftrightarrow$ $\exp\big(-y_n\big(\text{voting score}\big)\big)$ 越小越好 $\Leftrightarrow$ $u_n^{(T+1)}$ 越小越好。

结论　AdaBoost 的权重更新使得

$$\sum_{n=1}^{N} u_n^{(t)} \propto \sum_{n=1}^{N} \exp\big(-y_n \cdot \text{voting score}(x_n)\big)$$

在迭代中单调下降，从而保证间隔持续增大，最终获得强分类器。

命题 27.2.3 (AdaBoost 的指数误差函数)

AdaBoost 的迭代过程单调递减

$$\sum_{n=1}^{N} \exp\Big(-y_n \sum_{t=1}^{T} \alpha_t g_t(x_n)\Big),$$

从而近似最小化该目标。

指数误差（算法误差度量）　给定线性得分 $s = \sum_{t=1}^{T} \alpha_t g_t(x)$，定义

$$\widehat{\text{err}}_{\text{ADA}}(s, y) = \exp(-ys),$$

它是 0/1 误差的凸上界：

$$\text{err}_{0/1}(s, y) = \mathbb{I}[ys \le 0] \le \exp(-ys).$$

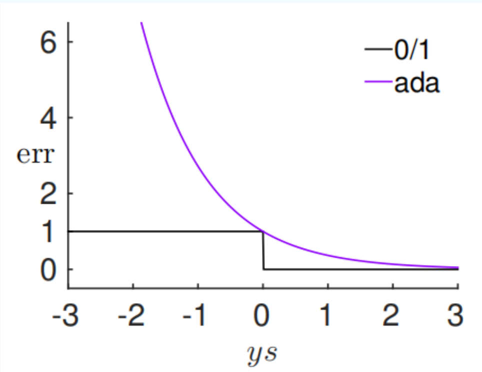

图 27.2.1: AdaBoost 的指数误差函数

结论　AdaBoost 通过最小化凸上界 $\exp(-ys)$ 来间接优化原本难以处理的 0/1 误差，实现高效的 boosting 过程。

命题 27.2.4 (AdaBoost 的梯度下降视角)

1. 函数空间梯度下降基础

梯度下降的核心是在迭代第 t 步时，通过一阶泰勒展开近似误差函数的局部更新：

$$\min_{\|\boldsymbol{v}\|=1} E_{\text{in}}(\boldsymbol{w}_t + \eta \boldsymbol{v}) \approx E_{\text{in}}(\boldsymbol{w}_t) + \eta \cdot \boldsymbol{v}^\top \nabla E_{\text{in}}(\boldsymbol{w}_t),$$

其中 $\boldsymbol{v}$ 为搜索方向，η 为步长，$\nabla E_{\text{in}}(\boldsymbol{w}_t)$ 为当前点梯度。

2. AdaBoost 误差的泰勒展开（函数空间梯度）

AdaBoost 的整体指数误差定义为：

$$E_{\mathrm{ADA}} = \frac{1}{N}\sum_{n=1}^{N}\exp\Big(-y_n\big(f_{t-1}(\boldsymbol{x}_n) + \eta\, h(\boldsymbol{x}_n)\big)\Big),$$

其中：

- $f_{t-1}(\boldsymbol{x}) = \sum_{\tau=1}^{t-1}\alpha_\tau g_\tau(\boldsymbol{x})$ 是前 $t-1$ 轮弱学习器的加权组合（函数空间“当前点”）；
- $h(\boldsymbol{x})$ 是第 t 轮待选弱学习器（搜索方向候选）；
- η 是步长（后续对应投票权重 α_t）。

引入样本权重 $u_n^{(t)} = \frac{1}{N}\exp\big(-y_n f_{t-1}(\boldsymbol{x}_n)\big)$（由历史模型 f_{t-1} 决定），对 η 做一阶泰勒展开（保留线性项 $\exp(-z)\approx 1-z$），误差简化为：

$$E_{\mathrm{ADA}} \approx \sum_{n=1}^{N} u_n^{(t)}\cdot\exp\big(-y_n\eta\, h(\boldsymbol{x}_n)\big) \approx \sum_{n=1}^{N} u_n^{(t)}\big(1 - y_n\eta\, h(\boldsymbol{x}_n)\big).$$

3. 弱学习器：梯度方向的求解（加权 0/1 误差）

优化弱学习器 h 需最小化近似误差的线性项：

$$\min_h \sum_{n=1}^{N} u_n^{(t)}\big(-y_n h(\boldsymbol{x}_n)\big).$$

由于二分类中 $y_n, h(\boldsymbol{x}_n)\in\{-1,+1\}$，符号分析可得：

- 若 $y_n = h(\boldsymbol{x}_n)$，则 $-y_n h(\boldsymbol{x}_n) = -1$；
- 若 $y_n \neq h(\boldsymbol{x}_n)$，则 $-y_n h(\boldsymbol{x}_n) = +1$。

因此，目标等价于加权 **0/1** 误差最小化：

$$\sum_{n=1}^{N} u_n^{(t)}\big(-y_n h(\boldsymbol{x}_n)\big) = \sum_{n=1}^{N} u_n^{(t)}\cdot\mathbb{I}\big[y_n \neq h(\boldsymbol{x}_n)\big]\cdot 2 - \sum_{n=1}^{N} u_n^{(t)} = -\sum_{n=1}^{N} u_n^{(t)} + 2E_{in}^{u^{(t)}}\cdot N,$$

这正是 AdaBoost 中弱学习算法 $\mathcal{A}$ 的目标——找到 g_t 最小化加权 0/1 误差，即 g_t 提供了函数空间梯度下降的近似方向。

4. 步长 α_t：最速下降的解析解

确定 g_t 后，误差简化为二分类形式（区分 g_t 正确/错误样本）：

$$E_{\mathrm{ADA}}(\eta) = \sum_{\substack{n\\ y_n = g_t(\boldsymbol{x}_n)}} u_n^{(t)} e^{-\eta} + \sum_{\substack{n\\ y_n \neq g_t(\boldsymbol{x}_n)}} u_n^{(t)} e^{\eta} = \Big(\sum_{n=1}^{N} u_n^{(t)}\Big)\big((1-\varepsilon_t)e^{-\eta} + \varepsilon_t e^{\eta}\big),$$

其中 $\varepsilon_t = \sum_{\substack{n\\ y_n\neq g_t(\boldsymbol{x}_n)}} u_n^{(t)}$ 是 g_t 的加权分类误差。

对 η 求导并令导数为 0（最速下降条件）：

$$\frac{\mathrm{d}E_{\mathrm{ADA}}}{\mathrm{d}\eta} = -(1-\varepsilon_t)e^{-\eta} + \varepsilon_t e^{\eta} = 0,$$

解得最优步长：

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

即 AdaBoost 的投票权重 α_t 是函数空间梯度下降的最优步长。

结论　AdaBoost 等价于函数空间的最速下降算法，迭代中：

1. 弱学习器 $\mathcal{A}$ 寻找梯度方向（最小化加权 0/1 误差的 g_t）；

2. 投票权重 α_t 是解析求解的最优步长，驱动模型沿梯度方向更新。
简言之，AdaBoost 通过“弱学习器定方向 + 解析步长定更新”，在函数空间逐步优化指数误差，最终构建强分类器。

27.3 梯度提升

命题 27.3.1 (Gradient Boosting：任意误差函数的推广)

AdaBoost（仅二元输出假设）

$$\min_{\eta}\min_{h}\frac{1}{N}\sum_{n=1}^{N}\exp\Big(-y_n\big(f_{t-1}(x_n)+\eta\,h(x_n)\big)\Big),\quad 其中 h(x_n)\in\{-1,+1\}.$$

Gradient Boosting（任意假设，实值输出） 将指数误差替换为任意可微损失 $\ell(\cdot,\cdot)$：

$$\min_{\eta}\min_{h}\frac{1}{N}\sum_{n=1}^{N}\ell\big(f_{t-1}(x_n)+\eta\,h(x_n),\ y_n\big),\quad 其中 h(x_n)\in\mathbb{R}.$$

关键差异

- AdaBoost 限定 h 为二元输出，损失固定为指数；
- Gradient Boosting 允许 h 为实值输出，并可选用任意损失（平方误差、Huber、logistic 等），从而自然扩展到回归、软分类等任务。

结论 Gradient Boosting 通过“函数空间梯度下降 + 任意损失”框架，成为 AdaBoost 在任意误差函数下的通用推广。

命题 27.3.2 (Gradient Boost 回归的残差拟合机制)

1. 目标设定：平方误差优化框架

在第 t 轮迭代中，当前加法模型为：

$$f_{t-1}(\boldsymbol{x})=\sum_{\tau=1}^{t-1}\alpha_\tau g_\tau(\boldsymbol{x})$$

对样本 $\boldsymbol{x}_n$，记前 $t-1$ 轮模型的预测值为 $s_n=f_{t-1}(\boldsymbol{x}_n)$，需最小化的平方损失目标为：

$$\min_{\eta,h}\ \frac{1}{N}\sum_{n=1}^{N}\mathrm{err}\Big(s_n+\eta\cdot h(\boldsymbol{x}_n),\ y_n\Big),\quad 其中 \mathrm{err}(s,y)=(s-y)^2$$

2. 一阶泰勒近似：简化优化路径

对平方损失 $\mathrm{err}(s,y)$ 做一阶泰勒展开（保留线性项与常数项），损失近似为：

$$\frac{1}{N}\sum_{n=1}^{N}\mathrm{err}\Big(s_n+\eta\cdot h(\boldsymbol{x}_n),\ y_n\Big)\approx\underbrace{\frac{1}{N}\sum_{n=1}^{N}\mathrm{err}(s_n,y_n)}_{常数项}+\frac{\eta}{N}\sum_{n=1}^{N}2h(\boldsymbol{x}_n)\cdot(s_n-y_n)$$

若不约束 h，理论上 $h(\boldsymbol{x}_n)=-\infty\cdot(s_n-y_n)$ 可使线性项最小化，但此时 h 的幅度无界，导致模型发散。

3. 正则化约束：残差拟合的本质

为避免 h 幅度失控，引入正则化后等价优化目标：

$$\min_h \frac{1}{N}\sum_{n=1}^{N}\left(2h(\boldsymbol{x}_n)(s_n-y_n)+(h(\boldsymbol{x}_n))^2\right)=\min_h \frac{1}{N}\sum_{n=1}^{N}\Big(\underbrace{-(y_n-s_n)^2}_{\text{常数项}}+\big(h(\boldsymbol{x}_n)-(y_n-s_n)\big)^2\Big)$$

定义残差 $r_n=y_n-s_n$，则弱学习器 h 的训练目标为：

$$g_t \leftarrow \arg\min_h \quad \frac{1}{N}\sum_{n=1}^{N}\big(h(\boldsymbol{x}_n)-r_n\big)^2$$

即 g_t 需拟合当前残差 r_n，弥补前 $t-1$ 轮模型的预测偏差。

4. 步长优化：单变量线性回归解

得到弱学习器 g_t 后，优化其在加法模型中的贡献步长 $\alpha_t=\eta$，需最小化：

$$\min_\eta \quad \frac{1}{N}\sum_{n=1}^{N}\Big(s_n+\eta\cdot g_t(\boldsymbol{x}_n)-y_n\Big)^2$$

这等价于单变量线性回归问题（仅优化 η），解析解为：

$$\alpha_t=\eta=\frac{\sum_{n=1}^{N}(y_n-s_n)\cdot g_t(\boldsymbol{x}_n)}{\sum_{n=1}^{N}\big(g_t(\boldsymbol{x}_n)\big)^2}$$

5. 总结　Gradient Boost 回归的迭代过程：

- 残差拟合：用当前残差 $r_n=y_n-s_n$ 训练弱回归器 g_t（$s_n=\sum_{\tau=1}^{t-1}\alpha_\tau g_\tau(\boldsymbol{x}_n)$）；
- 步长优化：通过单变量线性回归计算 g_t 的最优贡献权重 α_t；
- 迭代更新 $f_t(\boldsymbol{x})=f_{t-1}(\boldsymbol{x})+\alpha_t g_t(\boldsymbol{x})$，逐步降低平方损失。

算法 27.3.2: 梯度提升决策树 Gradient Boosted Decision Tree (GBDT)

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

输出: 最终回归器 $G(x)$

初始化残差

$$s_n^{(0)}=0,\quad n=1,\dots,N.$$

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

1. 拟合残差 使用平方误差回归算法 $\mathcal{A}$（如采样 + 剪枝的 C&RT）训练弱学习器

$$g_t=\mathcal{A}\big(\{(x_n,\,y_n-s_n^{(t-1)})\}_{n=1}^N\big).$$

2. 计算步长 对 $\{(g_t(x_n),\,y_n-s_n^{(t-1)})\}_{n=1}^N$ 做单变量线性回归，得

$$\alpha_t=\frac{\sum_{n=1}^N\big(y_n-s_n^{(t-1)}\big)\,g_t(x_n)}{\sum_{n=1}^N g_t(x_n)^2}.$$

3. 更新残差

$$s_n^{(t)}\leftarrow s_n^{(t-1)}+\alpha_t\,g_t(x_n),\quad n=1,\dots,N.$$

返回最终模型

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

27.4 集成模型小结

命题 27.4.1 (Blending 映射)

设已获得多样基模型 $\{g_t\}_{t=1}^T$，按聚合方式可将 Blending 映射为三类：

- **Uniform**：简单投票/平均

$$G(x)=\frac{1}{T}\sum_{t=1}^T g_t(x).$$

- **Non-uniform**：对 g_t 做线性加权

$$G(x)=\sum_{t=1}^T\alpha_t\,g_t(x),\quad \alpha_t\in\mathbb{R}.$$

- **Conditional**：在 g_t 变换后的特征上建立非线性模型

$$G(x)=\varphi\big(g_1(x),\dots,g_T(x)\big),\quad \varphi\text{ 非线性}.$$

实用准则

- 均匀聚合用于稳定性；
- 非均匀或条件聚合需谨慎设计，以控制复杂度。

命题 27.4.2 (Aggregation-Learning 模型映射)

设在学习过程中同时获得多样基模型 $\{g_t\}_{t=1}^T$，各类方法的多样性来源与聚合方式如下：

- **Bagging**：通过 自助采样产生多样 g_t，采用 均匀投票。
- **AdaBoost**：通过 重加权样本产生多样 g_t，采用 线性投票。
- 决策树：通过 数据分裂产生多样 g_t，采用 条件投票（分支路径）。
- **Gradient Boost**：通过 残差拟合产生多样 g_t，采用 线性投票，并以 最速下降更新。

结论　Boosting 类算法（AdaBoost、Gradient Boost 等）因兼具多样性与高效聚合，成为最受欢迎的集成方法。

命题 27.4.3 (Aggregation-of-Aggregation 模型映射)

三大基础集成策略及其常见组合：

- **Bagging**
  - Random Forest：Bagging + 完全生长的决策树（强基模型）。
  - Randomized Bagging：在 Bagging 中进一步引入随机性。
- **AdaBoost**
  - AdaBoost-DTree (强)：AdaBoost + 深度较大的决策树（强基模型）。
  - AdaBoost-DTree (弱)：AdaBoost + 决策树桩（弱基模型）。
- **Gradient Boost**
  - GBDT：Gradient Boost + 决策树桩（弱基模型）。

结论　Bagging、AdaBoost 与 Gradient Boost 及其树增强版本均为实践中高频使用的集成方法。

27.5 总结

笔记 [梯度提升决策树]

- 自适应提升决策树：通过抽样与剪枝构造“弱”树。
- AdaBoost 的优化视角：在指数误差上执行函数梯度下降。
- 梯度提升：迭代地沿最陡残差方向拟合。
- 集成模型小结：有的专治欠拟合，有的专治过拟合。

笔记 [总体结论] 梯度提升决策树作为集成学习的重要分支，通过迭代优化弱决策树的协作机制，实现了从”弱模型集成”到”强模型”的跨越。其核心在于利用权重调整（AdaBoost）或残差拟合（Gradient Boosting）构建具有多样性的基模型，并通过线性加权整合输出，在降低偏差、提升预测精度上表现卓越。这一框架既保留了决策树处理非线性数据的优势，又通过梯度下降等优化逻辑增强了模型的灵活性（支持多任务与任意损失函数），但也可能因迭代深度过深导致过拟合或计算成本上升。因此，需通过控制树的复杂度（如剪枝、限制深度）和迭代轮数，平衡性能与效率，是解决复杂预测问题的高效集成学习工具。