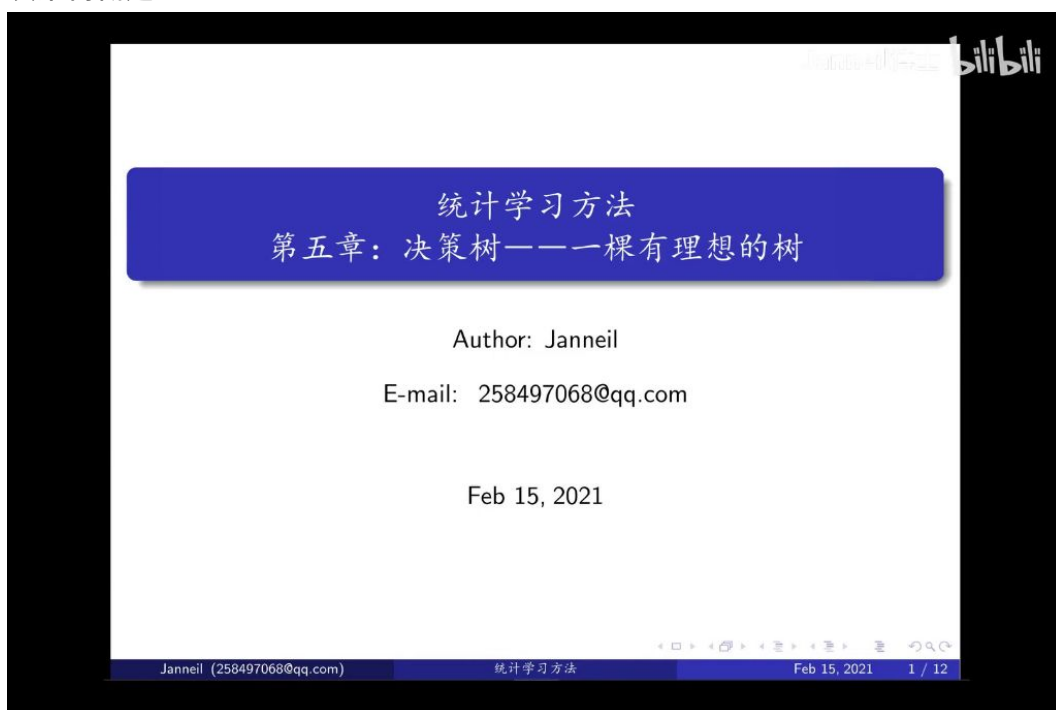
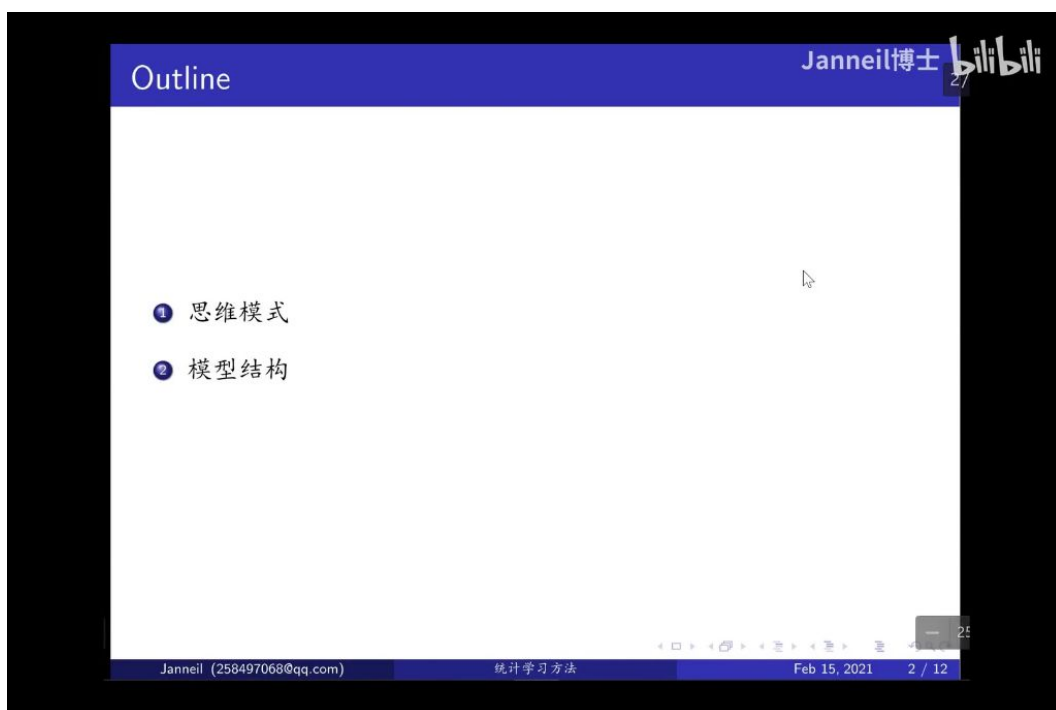


一、决策树概述 00:02



-
- 本质特征：决策树是一种有明确目标的树形结构，用于实现特定分类或决策目标
- 思维特点：采用归纳法思维模式，通过观察大量样本自底向上构建

1. 思维模式 00:30



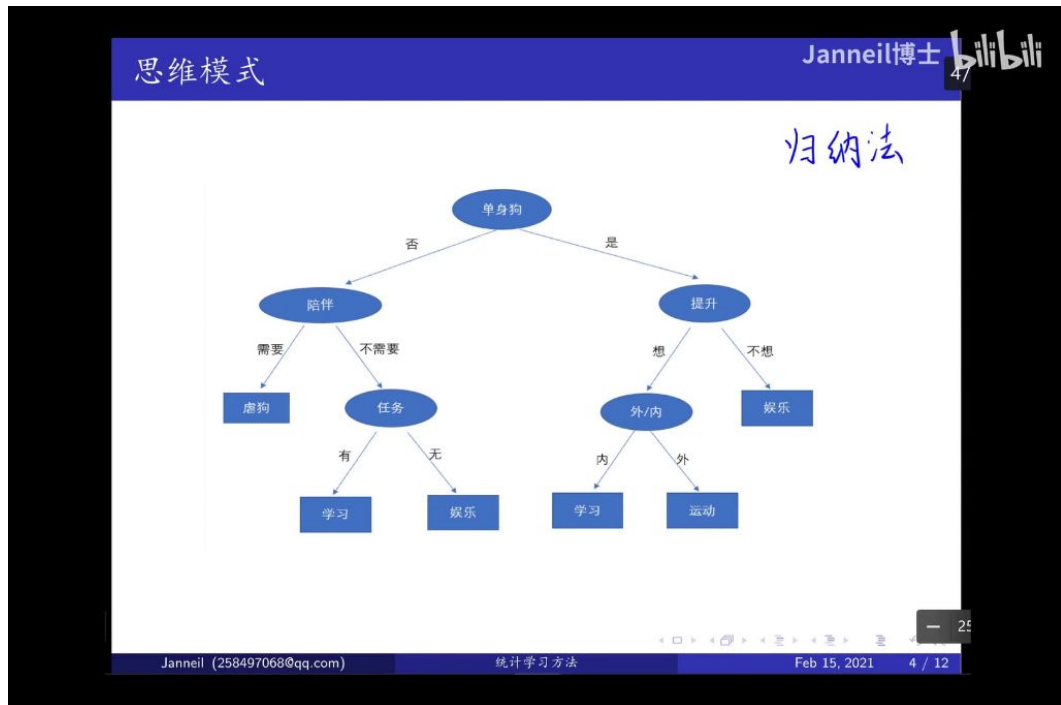
-
- 逻辑基础：基于归纳法而非演绎法，通过样本观察总结规律
- 构建过程：从具体实例出发，逐步抽象形成分类规则

2. 归纳法 00:47

- 方法论：通过观察大量样本（如情人节活动安排）总结规律
- 特点：自底向上的认知过程，与演绎法形成鲜明对比

3. 决策树实例 01:01

1) 情人节活动决策树 02:01



- 根节点：首先判断"是否单身"
- 分支逻辑：
 - 非单身：考虑"对方是否需要陪伴"→"是否有任务"→最终活动
 - 单身：考虑"是否想脱单"→"内修/外链"→具体活动
- 活动类型：包括学习、运动、娱乐等多种终端分类

2) 贷款决策树实例 05:26

- 特征选择：年收入、房产情况、婚姻状况三个关键特征
- 决策路径：
 - 年收入 ≥ 80 万→可偿还
 - 年收入 < 80 万→检查房产→无房产→检查婚姻状况
- 分类结果：最终判断贷款申请人是否具备偿还能力

4. 决策树特性 04:44

- 多叉性：决策树不一定是二叉树，节点可分多个叉（如学习可分为读书、看视频、实践）
- 主观性：实际构建的决策树可能带有主观色彩，需通过数据验证

二、决策树模型 06:43

1. 模型结构 06:49

决策树

分类决策树模型是一种描述对实例进行分类的树形结构。



-
- 组成要素：由节点和有向边构成
- 核心功能：描述对实例进行分类的树形结构

2. 节点类型 07:05

- 内部节点：表示特征或属性（圆形节点），用于数据分叉
- 叶节点：表示最终分类结果（方形节点）
- 根节点：最顶端的内部节点，决策起点

3. 有向边 08:14

- 表示形式：带箭头的线段
- 方向性：体现自上而下的决策路径

三、If-Then 规则 09:22

If-Then 规则

Janneil博士

97

决策树是通过一系列规则对数据进行分类的过程。

Janneil (258497068@qq.com)

统计学习方法

Feb 15, 2021 9 / 12

-
- 规则本质：决策树是if-then规则的集合
- 路径对应：每条从根到叶的路径对应一条分类规则
- 特性：
 - 互斥性：各分支条件互不重叠
 - 完备性：所有可能情况都被覆盖

四、决策树构建 12:13

1. 构建步骤 12:20

构建决策树

Janneil博士

107

步骤：

- 构建根结点；
- 选择最优特征，以此分割训练数据集；
- 若子集被基本正确分类，构建叶结点，否则，继续选择新的最优特征；
- 重复以上两步，直到所有训练数据子集被正确分类

Janneil (258497068@qq.com)

统计学习方法

Feb 15, 2021 10 / 12

-
- 构建根节点（包含全部训练样本）

- 选择最优特征进行数据分割
- 对子集递归应用步骤2，直到：
 - 子集被正确分类→创建叶节点
 - 无法正确分类→继续选择新特征
- 重复直至所有数据被分类

2. 关键问题 13:49

- **最优特征选择**：使用熵等指标评估特征重要性
- **分类标准**：不要求100%正确分类，需平衡拟合与泛化能力
- **过拟合防范**：通过剪枝等方法提高模型泛化性

五、条件概率分布 16:18

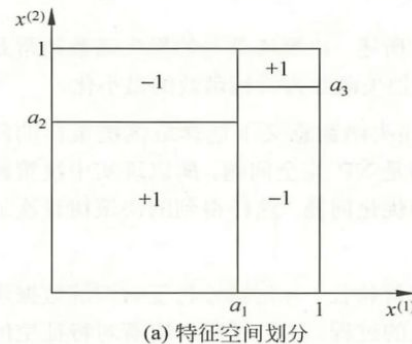
1. 概率模型分类 16:51



- **概率模型特征**：以条件概率形式表示，如朴素贝叶斯法和决策树
- **非概率模型**：包括感知机和k近邻法，不涉及条件概率分布
- **分类依据**：根据统计学习方法第一章的分类标准，模型可分为概率模型和非概率模型两类

2. 决策树条件概率 18:01

- 决策树：给定特征条件下类的 $P(Y|X)$
- 条件概率分布：特征空间的一个划分 (Partition)；
- 划分：单元 (Cell) 或区域 (Region) 互不相交。



- 数学表示：决策树表示给定特征 X 条件下类 Y 的条件概率分布 $P(Y|X)$
- 变量顺序意义： Y 在前表示分类目标， X 在后表示特征条件
- 本质理解：决策树本质上是将特征空间划分为多个区域，每个区域对应一个条件概率分布

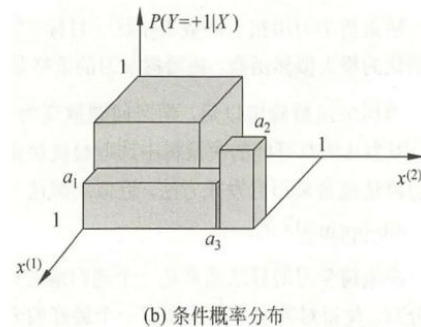
3. 特征空间划分 18:48

- 划分性质：特征空间的划分(Partition)要求各单元(Cell)或区域(Region)互不相交
- 空间定义：特征空间由特征向量组成，通常与输入空间相同
- 示例说明：以二维特征 $x = (x_1, x_2)$ 为例，当 $x_1 \in [0, 1]$ 且 $x_2 \in [0, 1]$ 时，构成单位正方形特征空间
- 与路径关系：每个单元对应决策树中的一条路径，体现if-then规则的互斥性和完备性

4. 单元分类标准 19:24

当某单元 c 的条件概率满足 $P(Y = +1|X = c) > 0.5$ 时，则认为该单元属于正类。

- 决策树：给定特征条件下类的 $P(Y|X)$;
- 条件概率分布：特征空间的一个划分 (Partition) ;
- 划分：单元 (Cell) 或区域 (Region) 互不相交。



- 判定规则：当单元 c 满足 $P(Y = +1|X = c) > 0.5$ 时归为正类
- 概率解释：正类概率大于负类概率即表示正类概率超过0.5
- 生活实例：乌云密布时判断下雨概率 > 0.5 ，类比于分类决策

5. 正类判定条件 22:08

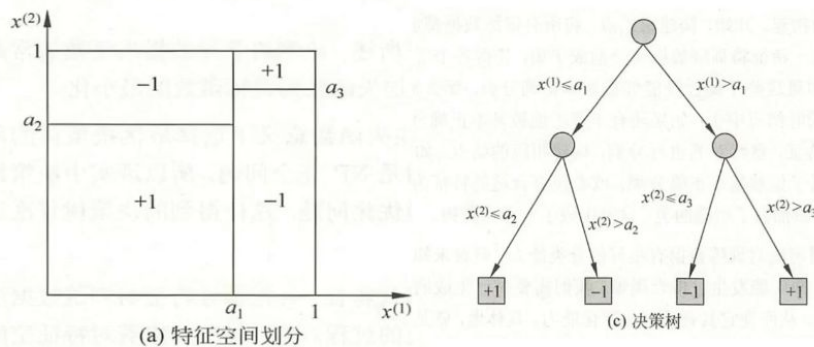
- 图形表示：立体图中 $y=+1$ 概率 > 0.5 的区域对应正类
- 反例情况：若改为 $y=-1$ 概率图，原高概率区域将变为低概率区域
- 概率关系： $P(Y = +1|X) + P(Y = -1|X) = 1$ ，因此比较大小即可判断类别

6. 多类情况处理 25:04

- 扩展规则：对于多分类问题，比较各类别条件概率大小
- 决策方法：将单元归为条件概率最大的那个类别
- 示例说明：若分为三类，则比较 $P(Y = 1|X = c)$ 、 $P(Y = 2|X = c)$ 、 $P(Y = 3|X = c)$ 的大小

7. 路径单元对应 27:00

- 一条路径对应于划分中的一个单元;
- 决策树的条件概率分布由各个单元给定条件下类的条件概率分布组成



Janneil (258497068@qq.com)

统计学习方法

Feb 24, 2021

8 / 13

- 对应关系：决策树每条路径对应特征空间的一个划分单元
- 示例解析：路径" $x^{(1)} \leq a_1$ 且 $x^{(2)} \leq a_2$ "对应区域 R_2
- 整体构成：决策树的条件概率分布由各单元的条件概率分布组成
- 双重理解：决策树既可视作if-then规则集合，也可视为条件概率分布的表示形式

六、决策树学习 29:22

1. 训练数据集 29:27

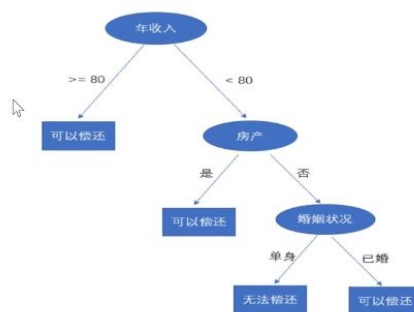
决策树学习

已知：训练集：

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中 $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T$, $y_i \in \{1, 2, \dots, K\}$, $i = 1, 2, \dots, N$;

目的：构造决策树，并对实例正确分类



Janneil (258497068@qq.com)

统计学习方法

Feb 24, 2021

10 / 13

- 数据表示：训练集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中每个实例 x_i 包含 n 个特征，输出 y_i 属于 K 个类别
- 学习目标：构造决策树模型，对新实例进行正确分类。例如贷款申请案例中，通过年收入、房产、婚姻状况等特征预测能否偿还贷款。

2. 贷款申请实例 30:37

- **实例分析：**年收入75万、无房产、单身 → 决策路径：年收入<80 → 无房产 → 单身 → 最终分类为"无法偿还"
- **关键问题：**
 - 同一训练集可生成多种决策树
 - 特征选择顺序影响树结构复杂度（如优先选择"有工作"特征可使树更简单）

3. 学习策略 33:16

决策树学习

- **本质：**从训练数据集中归纳出一组分类规则，与训练数据集不相矛盾
- **假设空间：**由无穷多个条件概率模型组成
- **一棵好的决策树：**与训练数据矛盾较小，同时具有很好的泛化能力
- **策略：**最小化损失函数
- **特征选择：**递归选择最优特征
- **生成：**对应特征空间的划分，直到所有训练子集被基本正确分类
- **剪枝：**避免过拟合，具有更好的泛化能力

Janneil (258497068@qq.com) 统计学习方法 Feb 24, 2021 11 / 13

- **本质：**从训练数据归纳分类规则，要求与训练数据不相矛盾
- **优化目标：**最小化损失函数，平衡拟合能力与泛化能力
 - **拟合能力：**对训练数据的分类准确度
 - **泛化能力：**对未知数据的预测准确度

4. 假设空间 34:09

- **模型特点：**由无穷多个条件概率模型组成，对应不同的特征选择顺序和树结构

5. 特征选择 35:04

- **递归选择：**通过信息增益等准则递归选择最优划分特征

6. 停止条件 35:41

- **终止标准：**当所有训练子集被基本正确分类时停止，允许少量错误以避免过拟合

7. 剪枝方法 36:16

- **预剪枝：**类似果树修剪，在生长过程中控制树高（如限制深度）
- **后剪枝：**类似园艺修剪，在树生成后修剪特定分支
- **目的：**提高模型泛化能力，防止过拟合

七、特征选择 37:46

1. 选择问题 37:52

- **核心问题：**如何确定特征选择顺序使决策树最优

2. 最优特征选择 38:08

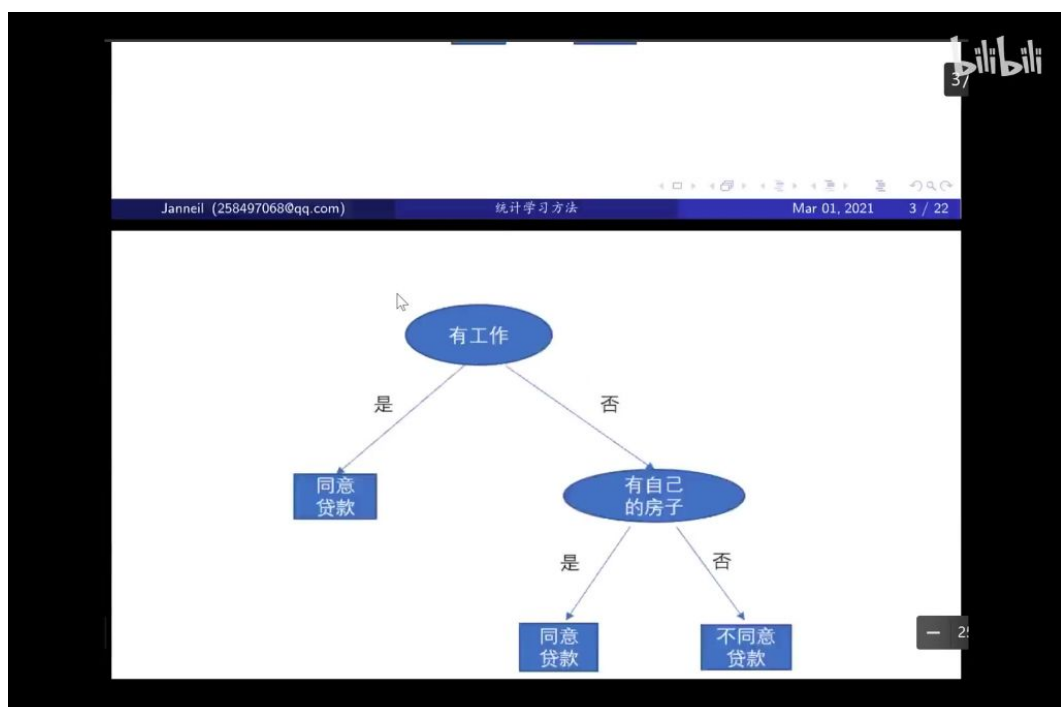
- 评价标准：需要量化指标（如信息增益）评估特征重要性

3. 贷款数据实例 39:50

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

- 数据集：15个样本，4个特征（年龄、工作、房产、信贷），2个类别（是否同意贷款）
 - 决策树对比：
 - 以"年龄"为根节点：生成较复杂的三分支树
 - 以"有工作"为根节点：生成较简单的二分支树
- ## 4. 特征选择必要性 41:51



- **关键发现：**不同特征作为根节点会导致决策树复杂度显著差异，因此需要科学的选择方法

八、信息增益 42:03

1. 熵定义 42:15

信息增益：熵
Janneil博士 Bilibili

- 熵表示是随机变量不确定性，

$$H(X) = - \sum_{i=1}^n p_i \log p_i$$

或

$$H(p) = - \sum_{i=1}^n p_i \log p_i$$

- 随机变量的取值等概率分布的时候，相应的熵最大。

$$0 \leq H(p) \leq \log n$$



Janneil (258497068@qq.com)
统计学习方法
Mar 01, 2021 7 / 22

- 数学表达： $H(X) = - \sum_{i=1}^n p_i \log p_i$ ，表示随机变量的不确定性
- 取值范围： $0 \leq H(p) \leq \log n$ ，当分布均匀时熵最大

2. 熵性质 43:02

- **最大熵：**当随机变量取值等概率时熵最大（如公平骰子的熵为 $\log_2 6$ ）
- **最小熵：**当结果确定时熵为0（如特异功能控制骰子点数）

3. 熵计算实例 45:01

- **二值变量：**设 $P(X=1)=p$ ，则 $H(p) = -p \log p - (1-p) \log (1-p)$
- **极值证明：**通过求导可得 $p=0.5$ 时熵最大为1

4. 条件熵 48:05

- **定义：**已知随机变量X时Y的不确定性， $H(Y|X) = \sum p_i H(Y|X=x_i)$

5. 经验熵 49:05

- **区别：**基于训练数据计算的熵称为经验熵，区别于真实分布的熵

6. 信息增益计算 50:54

- **计算步骤：**
 - 计算数据集的经验熵 $H(D)$
 - 计算特征A条件下的经验条件熵 $H(D|A)$
 - 信息增益 $g(D,A) = H(D) - H(D|A)$
- **选择准则：**信息增益越大表示特征带来的确定性提升越多，应优先选择

九、例题解说 51:58

1. 贷款数据表 52:00

信息增益：熵

Janneil博士

- 熵表示是随机变量不确定性,

$$H(X) = - \sum_{i=1}^n p_i \log p_i$$

或

$$H(p) = - \sum_{i=1}^n p_i \log p_i$$

- 随机变量的取值等概率分布的时候, 相应的熵最大。

$$0 \leq H(p) \leq \log n$$

Janneil (258497068@qq.com)
统计学习方法
Mar 01, 2021 8 / 22

- 熵的定义: $H(X) = - \sum_{i=1}^n p_i \log p_i$, 表示随机变量的不确定性
- 最大熵条件: 当随机变量取值等概率分布时熵最大, 范围满足 $0 \leq H(p) \leq \log n$
- 初始熵计算:
 - 数据集包含15个样本 (6个不同意贷款, 9个同意贷款)
 - 经验熵 $H(D) = - \frac{6}{15} \log_2 \frac{6}{15} - \frac{9}{15} \log_2 \frac{9}{15} = 0.971$

2. 条件熵计算 53:46

例题解说

Janneil博士

经验条件熵 $H(D|A)$:

$$H(D|A) = \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i) = - \sum_{i=1}^n \frac{|D_i|}{|D|} \sum_{k=1}^K \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$$

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$D A_i$	信贷情况	个数	是否同意贷款	
			否	是
D_1	非常好	4	0	4
D_2	好	6	2	4
D_3	一般	5	4	1

Janneil (258497068@qq.com)
统计学习方法
Mar 01, 2021 16 / 22

- 条件熵公式: $H(D|A) = \sum_{i=1}^n \frac{|D_i|}{|D|} H(D_i)$
- 年龄特征计算:
 - 划分为青年(5)、中年(5)、老年(5)三个子集
 - 青年子集: 3否/2是 $\rightarrow H(D_1) = -\frac{3}{5} \log \frac{3}{5} - \frac{2}{5} \log \frac{2}{5}$
 - 中年子集: 2否/3是 $\rightarrow$ 类似计算
 - 老年子集: 1否/4是 $\rightarrow$ 类似计算
 - 加权和得 $H(D|\text{年龄}) = 0.888$
- 有工作特征:
 - 划分为有工作(5)、无工作(10)
 - 有工作子集全为"是" $\rightarrow H(D_1) = 0$
 - 无工作子集: 6否/4是 $\rightarrow$ 计算得 $H(D_2)$
 - 最终 $H(D|\text{有工作}) = 0.647$
- 有房子特征:
 - 有房子(6)全为"是" $\rightarrow H(D_1) = 0$
 - 无房子(9): 3否/6是 $\rightarrow$ 计算 $H(D_2)$
 - 最终 $H(D|\text{有房子}) = 0.551$
- 信贷情况特征:
 - 划分为非常好(4)、好(6)、一般(5)
 - 非常好子集全为"是" $\rightarrow H(D_1) = 0$
 - 好子集: 2否/4是 $\rightarrow$ 计算 $H(D_2)$
 - 一般子集: 4否/1是 $\rightarrow$ 计算 $H(D_3)$
 - 最终 $H(D|\text{信贷}) = 0.608$

3. 信息增益结果 59:41

例题解说
Janneil博士 bilibili

信息增益:

$$g(D, A) = H(D) - H(D|A)$$

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

特征	经验熵	经验条件熵	信息增益
年龄	0.971	0.888	0.083
有工作	0.971	0.647	0.324
有自己的房子	0.971	0.551	0.420
信贷情况	0.971	0.608	0.363

Janneil (258497068@qq.com)
统计学习方法 Mar 01, 2021 17 / 22

- 计算公式: $g(D, A) = H(D) - H(D|A)$
- 各特征增益:
 - 年龄: $0.971 - 0.888 = 0.083$
 - 有工作: $0.971 - 0.647 = 0.324$

- 有房子: $0.971 - 0.551 = 0.420$
- 信贷: $0.971 - 0.608 = 0.363$
- **最优特征**: 有自己房子 (增益最大0.420)

4. 信贷情况分析 01:05:02

- **取值个数影响**: 信贷情况有3个取值 (非常好/好/一般), 有工作仅2个取值
- **潜在偏差**: 信息增益可能偏好取值多的特征 (信贷0.363 > 有工作0.324)

十、信息增益比 01:09:59

1. 取值个数影响 01:10:05

- **问题发现**: 信息增益倾向选择取值多的特征 (如信贷情况)
- **实例对比**: 信贷情况增益(0.363) > 有工作增益(0.324)

2. 惩罚项引入 01:11:52

- **解决思路**: 引入特征熵 $H_A(D)$ 作为惩罚项
- **类比解释**: 类似夏普比率 (单位风险收益)

3. 特征熵计算 01:13:31

例题解说

D 关于特征 A 的熵:

$$H(D) = - \sum_{k=1}^K \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$$

Janneil博士

197

表 6.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

特征	D_1	D_2	D_3
年龄	5	5	5
有工作	5	10	-
有自己的房子	6	9	-
信贷情况	4	6	5

Janneil (258497068@qq.com)

统计学习方法

Mar 01, 2021 19 / 23

● 年龄特征熵:

- 三个等概率子集 $\rightarrow H_{\text{年龄}}(D) = - \frac{5}{15} \log \frac{5}{15} \times 3 = 1.585$

● 有工作特征熵:

- 两个子集(5,10) $\rightarrow H_{\text{有工作}}(D) = 0.918$

● 有房子特征熵:

- 两个子集(6,9) $\rightarrow H_{\text{有房子}}(D) = 0.971$

● 信贷特征熵:

- 三个子集(4,6,5) $\rightarrow H_{\text{信贷}}(D) = 1.566$

4. 信息增益比计算 01:16:31

信息增益比:

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)}$$

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

特征	信息增益	$H_A(D)$	信息增益比
年龄	0.083	1.585	0.052
有工作	0.324	0.918	0.353
有自己的房子	0.420	0.971	0.432
信贷情况	0.363	1.566	0.232

- 计算公式: $g_R(D, A) = \frac{g(D, A)}{H_A(D)}$
- 计算结果:
 - 年龄: $0.083/1.585=0.052$
 - 有工作: $0.324/0.918=0.353$
 - 有房子: $0.420/0.971=0.432$
 - 信贷: $0.363/1.566=0.232$

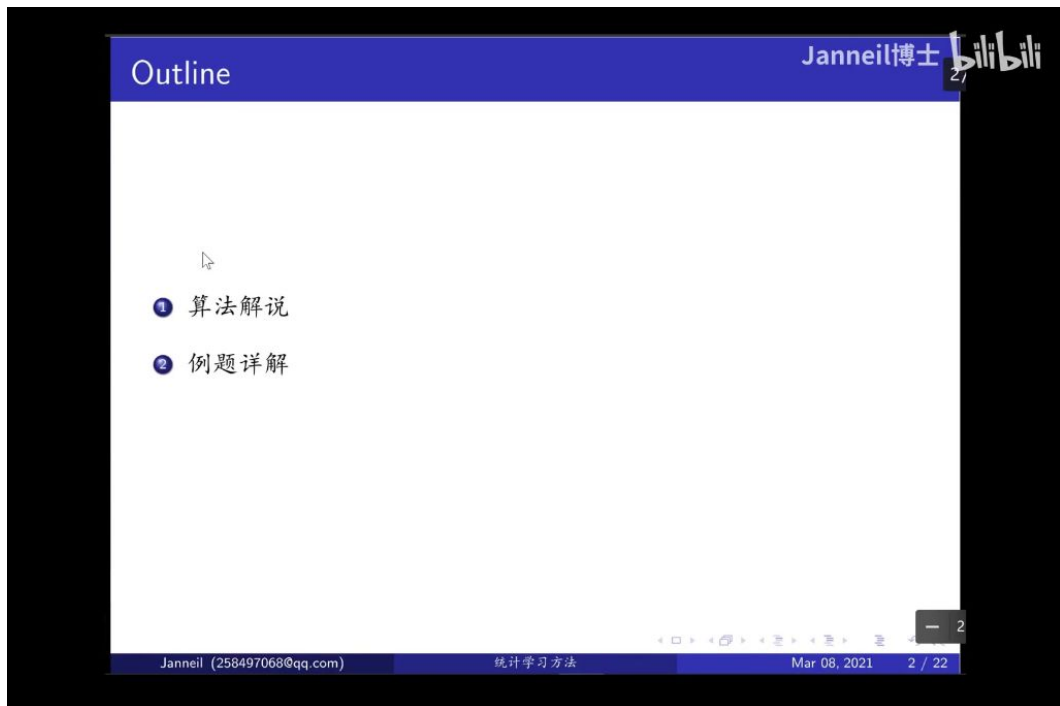
5. 特征选择比较 01:20:09

- 指标对比:
 - 信息增益: 有房子(0.420) > 信贷(0.363) > 有工作(0.324)
 - 信息增益比: 有房子(0.432) > 有工作(0.353) > 信贷(0.232)
- 指标特点:
 - 信息增益: 倾向取值多的特征
 - 信息增益比: 倾向取值少的特征
- 实际选择: 本例中两种方法均选择"有房子"为最优特征

十一、决策树算法 01:20:47

1. 决策树生成算法 01:20:50

1) 决策树生成步骤 01:20:52



-
- **基础概念**：决策树是通过递归地选择最优特征对数据集进行划分的树形结构
- **核心步骤**：特征选择→树生成→剪枝（本节课重点讲解前两个步骤）

2) CLS算法 01:21:09

- **历史地位**：1966年由Hunt提出的最早决策树算法，是其他算法的基础
- **主要缺陷**：未明确给出特征选择方法，导致实际应用受限
- **算法演进**：后续被ID3、C4.5等改进算法取代

3) ID3算法 01:21:47

- **提出时间**：1986年由Quinlan提出
- **核心改进**：明确使用信息增益作为特征选择标准
- **典型应用**：适合处理离散型特征数据

4) C4.5算法 01:22:10

- **提出时间**：1993年由Quinlan提出
- **算法地位**：2006年被国际数据挖掘大会票选为十大算法之首
- **主要改进**：将信息增益改为信息增益比，解决特征选择偏差问题

2. ID3算法详解 01:22:54

1) 算法输入输出 01:23:01

ID3算法

Janneil博士 bilibili

输入：训练数据集 D 、特征集 A 、阈值 ϵ

输出：决策树 T

- 判断 T 是否需要选择特征生成决策树；
 - 若 D 中所有实例属于同一类，则 T 为单结点树，记录实例类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 若 D 中所有实例无任何特征 ($A = \emptyset$)，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
- 否则，计算 A 中各特征的信息增益，并选择信息增益最大的特征 A_g ；
 - 若 A_g 的信息增益小于 ϵ ，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 否则，按照 A_g 的每个可能值 a_i ，将 D 分为若干非空子集 D_i ，将 D_i 中实例个数最多的类别作为标记，构建子结点，以结点和其子节点构成 T ，并返回 T ；
- 第 i 个子结点，以 D_i 为训练集， $A - A_g$ 为特征集合，递归地调用以上步骤，得到子树 T_i 并返回。

Janneil (258497068@qq.com)

统计学习方法

Mar 08, 2021

4 / 22

● 输入要素：

- 训练数据集 D
- 特征集 A
- 阈值 ϵ (人为设定)

● 输出结果：决策树 T

2) 算法步骤解析 01:23:49

● 递归终止条件：

- 所有实例属于同一类 C_k 时，生成单节点树
- 特征集 $A = \emptyset$ 时，选择实例最多的类 C_k 作为节点标记

● 核心操作：计算各特征信息增益，选择最大增益特征 A_g

3) 单节点树条件 01:24:53

● 两种情况：

- 实例完全同质（如10天天气数据都是晴天）
- 无可利用特征（如只有天气类别无温湿度等特征数据）

● 标记方法：选择出现频率最高的类别 C_k 作为节点标记

4) 特征选择过程 01:28:28

● 阈值判断：当 A_g 的信息增益 $< \epsilon$ 时仍生成单节点树

● 子树生成：

- 按 A_g 的取值 a_i 将 D 划分为子集 D_i
- 以 D_i 为训练集， $A - A_g$ 为特征集递归调用

● 终止条件：所有特征信息增益 $< \epsilon$ 或特征用完

3. C4.5算法详解 01:34:53

1) 信息增益比 01:35:03

- 改进原理：用信息增益比替代信息增益，解决偏向多值特征的问题
- 处理能力：可同时处理离散型和连续型特征
- 计算特点：相比ID3需要更多排序和扫描操作

2) 算法优缺点 01:35:43

- 优势:

- 继承ID3易解释性
- 特征选择更合理
- 支持连续变量处理

- 缺陷:

- 信息增益比计算复杂
- 大数据量时效率低下
- 需要多次数据扫描和排序

4. 例题: 贷款申请决策树 01:37:03

1) 数据集说明 01:37:47

例题详解

Janneil博士 Bilibili

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$D_2|A_1$

	年龄	个数	是否同意贷款	
			是	否
D_{21}	青年	4	1	3
D_{22}	中年	2	0	2
D_{23}	老年	3	2	1

Janneil (258497068@qq.com) 统计学习方法 Mar 08, 2021 11 / 17

-

- 数据特征: 包含15个样本, 9个同意贷款, 6个不同意贷款

- 特征集合: 包含4个特征 (年龄、有工作、有自己的房子、信贷情况)

- 初始判断:

- 实例不属于同一类 (既有同意又有不同意贷款)
- 特征集非空 ($A \neq \emptyset$)
- 不满足单结点树生成条件

2) 决策树绘制 01:39:07

C4.5 算法

Janneil博士 Bilibili

输入：训练数据集 D 、特征集 A 、阈值 ϵ

输出：决策树 T

- 判断 T 是否需要选择特征生成决策树
 - 若 D 中所有实例属于同一类，则 T 为单结点树，记录实例类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 若 D 中所有实例无任何特征 ($A = \emptyset$)，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
- 否则，计算 A 中各特征的信息增益比，并选择信息增益比最大的特征 A_g ：
 - 若 A_g 的信息增益比小于 ϵ ，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 否则，按照 A_g 的每个可能值 a_i ，将 D 分为若干非空子集 D_i ，将 D_i 中实例个数最多的类别作为标记，构建子结点，以结点和其子节点构成 T ，并返回 T ；
- 第 i 个子结点，以 D_i 为训练集， $A - A_g$ 为特征集合，递归地调用以上步骤，得到子树 T_i 并返回。

25

Janneil (258497068@qq.com)

统计学习方法

Mar 08, 2021

7 / 17

- 根节点选择：信息增益最大的特征"有自己的房子"
 - 当"有自己的房子=是"时，全部样本同意贷款
 - 当"有自己的房子=否"时，样本包含6同意3不同意
 - 递归构建：对"无自己房子"的子集 D_2 继续划分
- 3) 信息增益计算 01:41:23

例题详解

Janneil博士 Bilibili

表 5.1 贷款申请样本数据集

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$D_2 A_1$	年龄	个数	是否同意贷款	
			是	否
D_{21}	青年	4	1	3
D_{22}	中年	2	0	2
D_{23}	老年	3	2	1

$$g(D_2, A_1) = H(D_2) - H(D_2|A_1)$$

26

Janneil (258497068@qq.com)

统计学习方法

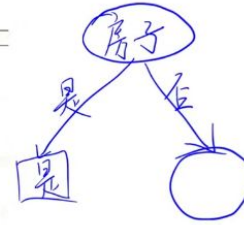
Mar 08, 2021

11 / 17

- 计算公式： $g(D_2, A_1) = H(D_2) - H(D_2|A_1)$
 - 特征选择：
 - 剩余特征集： $\{A_1(\text{年龄}), A_2(\text{有工作}), A_4(\text{信贷情况})\}$
 - 计算各特征的信息增益
- 4) 经验熵计算 01:44:19

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	是	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否



$$H(D_2) = -\frac{6}{9} \log_2 \frac{6}{9} - \frac{3}{9} \log_2 \frac{3}{9} = 0.918$$

$D A_3$	有自己的房子	个数	是否同意贷款	
			是	否
D_1	是	6	6	0
D_2	否	9	6	3

$\{A_1, A_2, A_4\}$

- 计算公式: $H(D_2) = -\frac{6}{9} \log_2 \frac{6}{9} - \frac{3}{9} \log_2 \frac{3}{9}$
- 计算结果: 0.918
- 子集分布:
 - 同意贷款: 6个
 - 不同意贷款: 3个
- 5) 条件熵计算 01:45:49
- 年龄特征
 - 划分情况:
 - 青年: 4个 (1同意3不同意)
 - 中年: 2个 (0同意2不同意)
 - 老年: 3个 (2同意1不同意)
 - 计算公式: $H(D_2|A_1) = \sum \frac{|D_{2i}|}{9} H(D_{2i})$
 - 计算结果: 0.667
- 工作特征 01:50:48

The slide displays a dataset of 15 samples with features: 序号 (Serial Number), 年龄 (Age), 工作 (Job), 房子 (House), 信用 (Credit), and 是否同意贷款 (Loan Approval). The samples are grouped into two categories: '有工作' (Has Job) and '无工作' (No Job).

序号	年龄	工作	房子	信用	是否同意贷款
7	中年	是	是	好	是
8	中年	是	是	非常好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	否
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$D_2 A_2$	有工作	个数	是否同意贷款	
			是	否
D_{21}	是	3	3	0
D_{22}	否	6	0	6

Handwritten decision tree diagram:

```

graph TD
    Root(( )) -->|是| JobYes[有工作]
    Root -->|否| JobNo[无工作]
    JobYes -->|是| LoanYes[是]
    JobYes -->|否| LoanYes
    JobNo -->|是| LoanNo[否]
    JobNo -->|否| LoanNo
  
```

-
- 划分情况：
 - 有工作：3个（全部同意）
 - 无工作：6个（全部不同意）
- 特殊性质：条件熵为0（子集纯度100%）
- 信贷特征
 - 划分情况：
 - 非常好：1个（1同意0不同意）
 - 好：4个（2同意2不同意）
 - 一般：4个（0同意4不同意）
 - 计算结果：0.444
- 6) 决策树生成
 - 特征选择：
 - 选择信息增益最大的"有工作"特征
 - 当"有工作=是"时全部同意贷款
 - 当"有工作=否"时全部不同意贷款
 - 终止条件：所有子节点纯度达到100%
 - 最终结构：
 - 第一层：有自己的房子
 - 第二层：有工作
 - 共使用2个特征完成分类
 - 算法对比：
 - ID3算法使用信息增益
 - C4.5算法使用信息增益比（建议课后练习）

十二、决策树剪枝 01:53:52

1. 剪枝目的 01:53:56

统计学习方法

第五章：决策树——剪枝

Author: Janneil

E-mail: 258497068@qq.com

Mar 12, 2021

Janneil (258497068@qq.com)

统计学习方法

Mar 12, 2021

1 / 8

- 类比解释：决策树剪枝与植树节移植树苗的原理相似，关键在于通过适当修剪使结构更合理
- 核心目的：处理决策树的过拟合问题，通过人工干预使树形结构更适应决策需求（类比仿生学中飞机机翼借鉴鸟类但加入人工改进）

2. 优秀决策树标准 01:56:51

剪枝

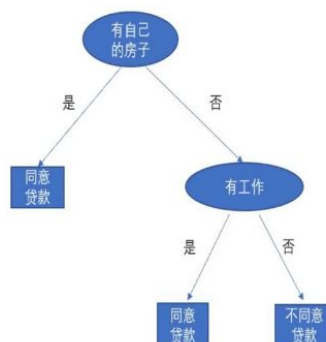
Janneil博士

优秀的决策树：

在具有好的拟合和泛化能力的同时，

- 深度小；
- 叶结点少；
- 深度小并且叶结点少

深度：所有结点的最大层次数



Janneil (258497068@qq.com)

统计学习方法

Mar 12, 2021

3 / 8

- 能力维度：
 - 拟合能力：对训练数据的解释能力（对应训练误差）
 - 泛化能力：对未知数据的预测能力（对应测试误差）
- 结构维度：
 - 深度：所有结点的最大层次数（类比楼房层数）
 - 叶结点数：终端节点数量（反映树的宽度）

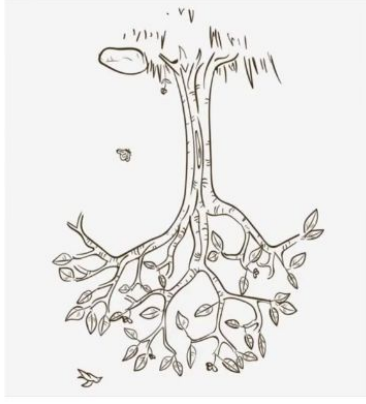
- 理想状态：同时满足深度小且叶结点少 ($depth \downarrow \cap leaves \downarrow$)
- 过拟合表现：训练误差低但测试误差高，通常伴随深度过大或叶结点过多
- 欠拟合表现：训练误差和测试误差都较高，结构过于简单

3. 剪枝分类 01:54:26

1) 预剪枝

剪枝
Janneil博士 bilibili

- 剪枝：处理决策树的过拟合问题
- 预剪枝：生成过程中，对每个结点划分前进行估计，若当前结点的划分不能提升泛化能力，则停止划分，记当前结点为叶结点；
- 后剪枝：生成一棵完整的决策树之后，自底而上地对内部结点考察，若此内部结点变为叶结点，可以提升泛化能力，则做此替换。



Janneil (258497068@qq.com)
统计学习方法
Mar 12, 2021 5 / 8

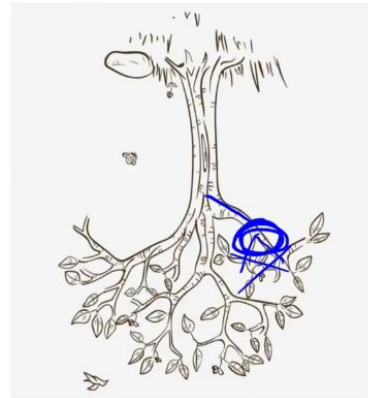
- 时机：决策树生成过程中（类比果树结果前的修剪）
- 执行逻辑：
 - 对每个结点划分前进行泛化能力估计
 - 若划分不能提升泛化能力，则停止划分并将当前结点标记为叶结点
- 特点：类似“防患于未然”的修剪策略

2) 后剪枝

- 剪枝：处理决策树的过拟合问题



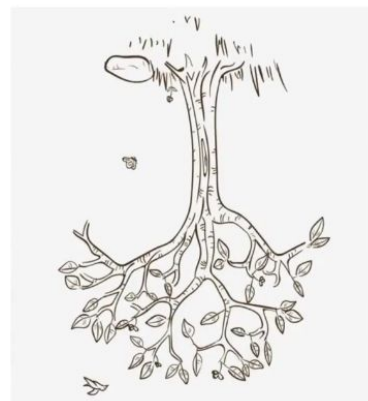
- 后剪枝：生成一棵完整的决策树之后，自底而上地对内部结点考察，若此内部结点变为叶结点，可以提升泛化能力，则做此替换。



- 时机：完整决策树生成后（类比电影《西虹市首富》中果树造型修剪）
 - 执行逻辑：
 - 自底向上考察内部结点
 - 若将内部结点变为叶结点能提升泛化能力，则进行替换
 - 特点：类似“事后优化”的修剪策略
4. 剪枝与过拟合关系 02:04:35

- 剪枝：处理决策树的过拟合问题

- 预剪枝：生成过程中，对每个结点划分前进行估计，若当前结点的划分不能提升泛化能力，则停止划分，记当前结点为叶结点；
- 后剪枝：生成一棵完整的决策树之后，自底而上地对内部结点考察，若此内部结点变为叶结点，可以提升泛化能力，则做此替换。



- 过拟合成因：决策树生成时过度追求训练数据拟合，导致结构复杂化
- 剪枝作用：
 - 通过限制树的深度和叶结点数量控制模型复杂度
 - 在拟合能力和泛化能力之间取得平衡
- 实现方式：

- 预剪枝：提前阻止不必要的分支生长
- 后剪枝：对已生成的复杂分支进行修剪

十三、预剪枝详解 02:08:28

1. 预剪枝方法 02:08:36

1) 限定决策树深度 02:08:43

预剪枝

Janneil博士

- 限定决策树的深度
- 设定一个阈值
- 设置某个指标，比较结点划分前后的泛化能力

Janneil (258497068@qq.com) 统计学习方法 Mar 26, 2021 2 / 18

- 控制原理：通过限制树的高度来简化模型结构，类比四层楼房比99层楼房结构更简单
- 实现方式：在训练时预设最大深度参数，当达到指定层数时停止分裂
- 效果影响：深度越小模型越简单，可能欠拟合；深度越大模型越复杂，可能过拟合
- 西瓜数据集案例

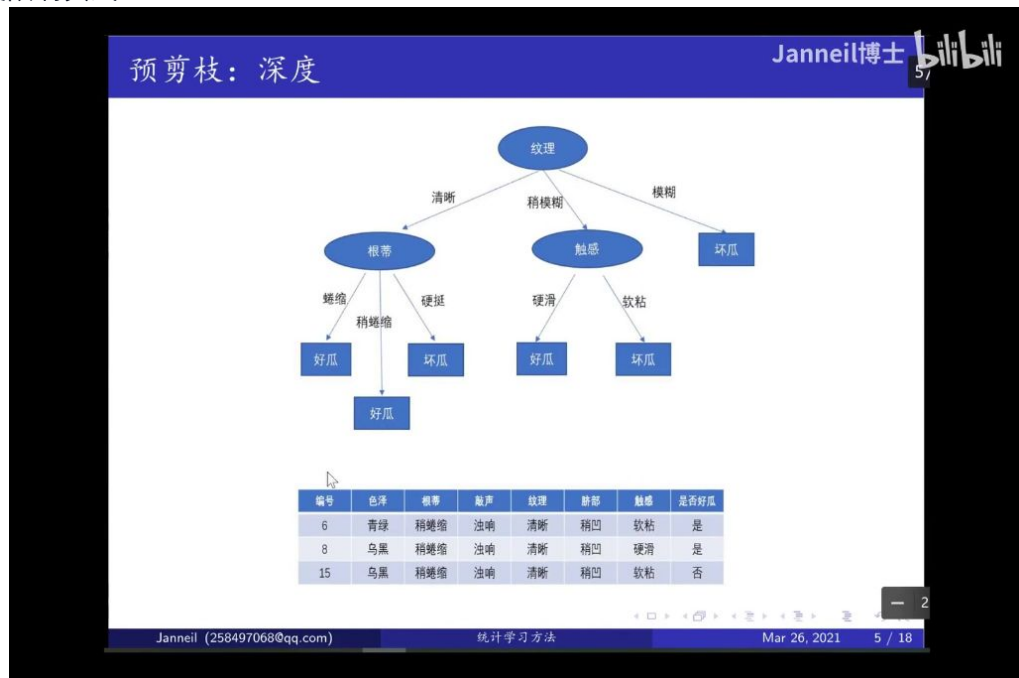
预剪枝：挑西瓜

Janneil博士

编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷缩	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷缩	浊响	稍模糊	稍凹	软粘	是
8	乌黑	稍蜷缩	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷缩	沉闷	稍模糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷缩	浊响	稍模糊	凹陷	硬滑	否
14	浅白	稍蜷缩	沉闷	稍模糊	凹陷	硬滑	否
15	乌黑	稍蜷缩	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍模糊	稍凹	硬滑	否

Janneil (258497068@qq.com) 统计学习方法 Mar 26, 2021 3 / 18

- **数据特征**：包含17个样本，6个特征变量（色泽、根蒂等）和1个二分类输出变量（是否好瓜）
- **决策树构建**：通过信息增益生成的完整决策树深度为4层（从第0层根节点到第4层叶节点）
- **深度限制实践**



- **操作示例**：当限制深度为2时，需剪除第3层及以下节点
 - **叶节点处理**：对于纹理清晰且根蒂稍蜷缩的样本（编号6,8,15），按多数表决确定为"好瓜"类别
 - **对比说明**：相比完整深度4的树，深度2的树保留了主要决策路径但结构更简洁
- 2) 设定阈值 02:17:44

预剪枝：阈值

Janneil博士

输入：训练数据集 D 、特征集 A 、阈值 ϵ
输出：决策树 T

- 判断 T 是否需要选择特征生成决策树
 - 若 D 中所有实例属于同一类，则 T 为单结点树，记录实例类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 若 D 中所有实例无任何特征 ($A = \emptyset$)，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
- 否则，计算 A 中各特征的信息增益，并选择信息增益最大的特征 A_g ：
 - 若 A_g 的信息增益小于 ϵ ，则 T 为单结点树，记录 D 中实例个数最多类别 C_k ，以此作为该结点的类标记，并返回 T ；
 - 否则，按照 A_g 的每个可能值 a_i ，将 D 分为若干非空子集 D_i ，将 D_i 中实例个数最多的类别作为标记，构建子结点，以结点和其子节点构成 T ，并返回 T ；
- 第 i 个子结点，以 D_i 为训练集， $A - A_g$ 为特征集合，递归地调用以上步骤，得到子树 T_i 并返回。

Janneil (258497068@qq.com) 统计学习方法 Mar 26, 2021 6 / 18

- **算法原理**：递归过程中比较特征信息增益与预设阈值 ϵ

- **阈值作用**：当所有特征的信息增益都小于 ϵ 时停止分裂，形成单节点树
- **递归应用**：阈值检查不仅作用于根节点，也应用于所有子树的生成过程
- 阈值设置案例

预剪枝：阈值

Janneil博士 Bilibili

阈值 $\epsilon = 0.4$

特征	信息增益
色泽	0.109
根蒂	0.143
敲声	0.141
纹理	0.381
脐部	0.289
触感	0.006

坏瓜

Janneil (258497068@qq.com) 统计学习方法 Mar 26, 2021 7 / 18

- 数据验证：西瓜数据集中所有特征的最大信息增益为0.381（纹理特征）
- 阈值实验：当设置 $\epsilon = 0.4$ 时，所有特征信息增益均小于阈值
- 结果分析：生成单节点树，根据样本分布（8好瓜/9坏瓜）判定为"坏瓜"类别
- 参数建议：可通过尝试不同阈值（如0.2、0.1）观察模型复杂度变化
- 注意事项
 - 阈值选择：需通过交叉验证确定最佳值，过大导致欠拟合，过小导致过拟合
 - 特征影响：不同特征的信息增益值差异较大（如触感仅0.006），需标准化处理时可考虑增益率
 - 对比深度限制：阈值法更侧重信息量过滤，深度限制更关注结构复杂度

3) 基于测试集误差率的预剪枝 02:21:00

预剪枝：基于测试集上的误差率

Janneil博士 Bilibili

编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷缩	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷缩	浊响	稍模糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷缩	沉闷	稍模糊	凹陷	硬滑	否
15	乌黑	稍蜷缩	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍模糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷缩	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷缩	沉闷	稍模糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷缩	浊响	稍模糊	凹陷	硬滑	否

测试集上的误差率：测试集中错误分类的实例占比；
测试集上的准确率：测试集中正确分类的实例占比。

Janneil (258497068@qq.com)

统计学习方法

Mar 26, 2021

8 / 18

基本概念

- 评估指标：使用测试集上的误差率作为预剪枝的判断标准，该指标反映模型泛化能力
- 误差率定义：测试集中错误分类的实例占比，计算公式为 $\frac{\text{错误样本数}}{\text{总样本数}}$
- 准确率定义：测试集中正确分类的实例占比，与误差率互补（准确率=1-误差率）
- 指标意义：误差率越低代表泛化能力越强，误差率越高则泛化能力越弱

预剪枝实施步骤

预剪枝：根部

Janneil博士 Bilibili

训练集	个数
好瓜	
坏瓜	

好瓜

测试集	个数
好瓜	
坏瓜	

坏瓜

Janneil (258497068@qq.com)

统计学习方法

Mar 26, 2021

9 / 18

- 数据划分：
 - 将原始数据集分为训练集（构建决策树）和测试集（评估泛化能力）
 - 示例中训练集包含5个好瓜和5个坏瓜，测试集包含3个好瓜和4个坏瓜
- 基准模型建立：

- 构建单节点决策树（根节点）
- 当类别分布相同时，可任意选择好瓜或坏瓜作为预测结果
- 误差率计算：
 - 若单节点决策树预测为好瓜：
 - 测试集误判数：4个（所有坏瓜样本）
 - 误差率： $\frac{4}{7}$
 - 若预测为坏瓜：
 - 测试集误判数：3个（所有好瓜样本）
 - 误差率： $\frac{3}{7}$
- 节点划分验证

预剪枝：第一层

Janneil博士 107

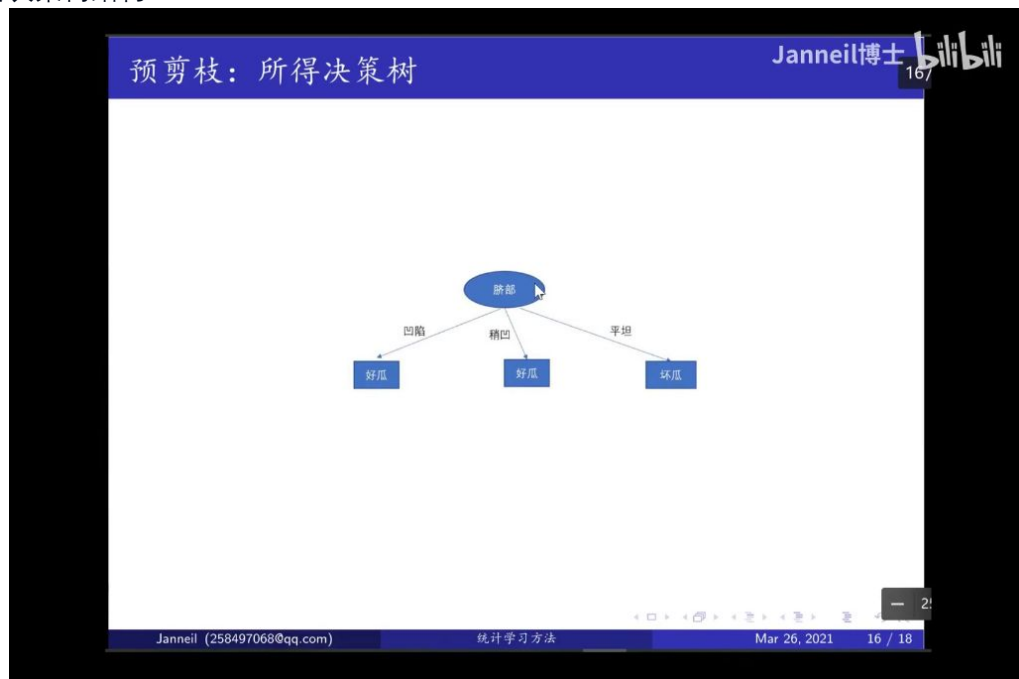
编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
14	浅白	稍蜷缩	沉闷	稍模糊	凹陷	硬滑	否
6	青绿	稍蜷缩	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷缩	浊响	稍模糊	稍凹	软粘	是
15	乌黑	稍蜷缩	浊响	清晰	稍凹	软粘	否
17	青绿	蜷缩	沉闷	稍模糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否

训练集

Janneil (258497068@qq.com) 统计学习方法 Mar 26, 2021 10 / 18

- 首次划分：
 - 根据信息增益选择"脐部"作为划分特征
 - 生成二层决策树：
 - 脐部=凹陷 → 好瓜
 - 脐部=稍凹 → 好瓜
 - 脐部=平坦 → 坏瓜
 - 新误差率： $\frac{2}{7}$ （比单节点树的 $\frac{4}{7}$ 或 $\frac{3}{7}$ 更低）
- 二次划分验证：
 - 对"凹陷"节点考虑按"色泽"进一步划分：
 - 浅白 → 坏瓜
 - 青绿 → 好瓜
 - 乌黑 → 好瓜
 - 新误差率升至 $\frac{3}{7}$ ，说明划分导致泛化能力下降
- 其他节点验证：
 - 对"稍凹"节点考虑按"根蒂"划分：
 - 误差率保持 $\frac{2}{7}$ ，但模型复杂度增加
 - 根据奥卡姆剃刀原理，选择不划分

- "平坦"节点无需划分（训练集中均为坏瓜）
- 最终决策树结构



-
- 层级结构：两层决策树
 - 根节点：脐部特征
 - 叶节点：
 - 凹陷/稍凹 → 好瓜
 - 平坦 → 坏瓜
- 性能指标：保持最低测试误差率 $\frac{2}{7}$
- 关键结论：当划分后误差率升高或不变时，应停止该节点的进一步划分

十四、后剪枝详解 02:34:13

1. 后剪枝方法 02:34:21

- 降低错误剪枝 (REP)
- 悲观错误剪枝 (PEP)
- 最小误差剪枝 (MEP)
- 基于错误的剪枝 (EBP)
- 代价-复杂度剪枝 (CCP)

- 方法分类：降低错误剪枝(REP)、悲观错误剪枝(PEP)、最小误差剪枝(MEP)、基于错误的剪枝(EBP)、代价-复杂度剪枝(CCP)
- 1) 原理说明 02:35:07

后剪枝：降低错误剪枝

原理：自下而上，使用测试集来剪枝。对每个结点，计算剪枝前和剪枝后的误判个数，若是剪枝有利于减少误判（包括相等的情况），则减掉该结点所在分枝。

编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷缩	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷缩	浊响	稍模糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷缩	沉闷	稍模糊	凹陷	硬滑	否
15	乌黑	稍蜷缩	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍模糊	稍凹	硬滑	否

编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷缩	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷缩	沉闷	稍模糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷缩	浊响	稍模糊	凹陷	硬滑	否

- 基本流程：自下而上遍历决策树，使用测试集评估剪枝效果
- 决策标准：
 - 比较指标：计算剪枝前后误判个数
 - 剪枝条件：当剪枝后误判数 $\leq$ 剪枝前时执行剪枝（包含相等情况）
 - 理论依据：奥卡姆剃刀原理，在效果相同时选择结构更简单的模型
- 与预剪枝区别：
 - 方向差异：后剪枝自下而上，预剪枝自上而下
 - 评估指标：后剪枝使用误判个数，预剪枝使用误差率（两者本质等价）

- **数据集说明：**西瓜数据集共17个样本，其中10个训练样本，7个测试样本

2. 例题:完整决策树 02:37:10

1) 第四层剪枝分析 02:37:50

- **决策树特征：**深度为4的完整决策树
- **剪枝操作：**
 - **目标节点：**第四层子树（脐部稍凹、根蒂稍蜷缩、色泽乌黑）
 - **替代方案：**用标记为"好瓜"的叶节点替代该子树
 - **判断依据：**比较子树和叶节点在测试集上的误判个数
 - **训练集参考：**该节点对应训练样本中1个好瓜和1个坏瓜样本

十五、决策树后剪枝 02:38:46

1. 误判个数计算 02:38:49

后剪枝：第二层

Janneil博士 bilibili

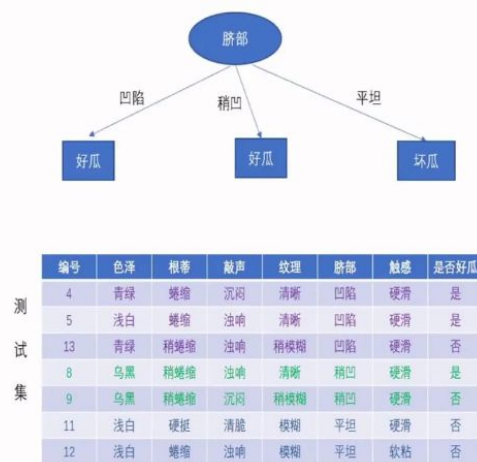
编号	色泽	根蒂	敲声	纹理	脐部	触感	是否好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
13	青绿	稍蜷缩	浊响	稍模糊	凹陷	硬滑	否
8	乌黑	稍蜷缩	沉闷	清晰	稍凹	硬滑	是
9	乌黑	稍蜷缩	沉闷	稍模糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否

- **子树误判示例：**当使用某棵子树判断时，脐部稍凹、根蒂稍蜷缩、色泽乌黑的样本中，纹理清晰的误判为坏瓜，稍模糊的误判为好瓜，共2个误判。
- **叶节点替换效果：**若用"好瓜"叶节点替换该子树，误判个数降为1个（ $1 < 2$ ），说明剪枝有效。

2. 子树替换策略 02:39:39

- **替换条件：**当叶节点的误判数小于子树的误判数时，应进行替换。
- **深度变化：**通过替换可使决策树深度从4层简化为3层。

3. 减支决策判断 02:40:30



- 训练集参考：脐部稍凹、根部稍蜷缩的3个训练样本中，2好瓜1坏瓜，替换叶节点应选择"好瓜"。
- 误判对比：子树在测试集误判1个；叶节点同样误判1个，此时根据奥卡姆剃刀原理选择剪枝。

4. 误判个数比较 02:41:41

- 多层剪枝示例：
 - 左子树：脐部凹陷样本中，色泽判断误判2个；替换为"好瓜"叶节点后误判降为1个
 - 右子树：脐部稍凹样本中，根蒂判断误判1个；替换叶节点后误判仍为1个

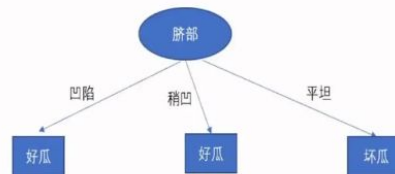
5. 决策树简化过程 02:42:16

- 最终简化：尝试将2层树简化为单节点树时，误判从2个增至4个，因此不应继续剪枝。

6. 奥卡姆剃刀原理 02:43:49

- 应用原则：当剪枝前后误判数相同时，优先选择更简单的模型（即剪枝后的叶节点）。

7. 后剪枝结果分析 02:46:02



- 计算复杂性是线性的
- 操作简单，容易理解
- 受测试集影响大，如果测试集比训练集小很多，会限制分类的精度

- 优点：
 - 计算复杂度为线性 $O(n)$
 - 操作简单直观，每个节点只需访问一次
 - 缺点：
 - 易受测试集规模影响
 - 测试集远小于训练集时可能导致欠拟合
8. 悲观错误剪枝 02:47:11

原理：根据剪枝前后的错误率来决定是否剪枝。和REP 不同之处在于，PEP 只需要训练集即可，不需要验证集，并且PEP是自上而下剪枝的。



- 核心区别：与REP方法相比，PEP只需训练集，且采用自上而下剪枝方式
 - 修正原理：通过+0.5将离散二项分布修正为连续正态分布
9. 悲观剪枝步骤 02:48:54

1) 减值前计算 02:48:55

- 叶子误差率: $error_{leaf_i} = \frac{e_i + 0.5}{n_t}$, 其中 e_i 为误判数
- 总修正误差: $error_T = \frac{\sum e_i + 0.5L}{n_t}$, L 为叶节点数
- 期望与标准差:
 - 期望: $E = n_t \times error_T$
 - 标准差: $\sigma = \sqrt{n_t \times error_T \times (1 - error_T)}$

2) 减值后计算 02:54:15

- 叶节点误差: $error_{leaf} = \frac{e + 0.5}{n_t}$
- 期望值: $E_{leaf} = e + 0.5$

3) 剪枝决策 02:55:19

- 判定条件: 当 $E_{leaf} \leq (E + \sigma)$ 时执行剪枝
- 悲观特性: 采用误判上限 (期望+标准差) 确保决策保守性

10. 悲观剪枝例题 02:55:51

Janneil博士 bilibili

后剪枝：悲观错误剪枝(PEP)

步骤:

- 计算剪枝前目标子树每个叶子结点的误差, 并进行连续修正:

$$Error(Leaf_i) = \frac{error(Leaf_i) + 0.5}{N(T)}$$
- 计算剪枝前目标子树的修正误差:

$$Error(T) = \sum_{i=1}^L Error(Leaf_i) = \frac{\sum_{i=1}^L error(Leaf_i) + 0.5 * L}{N(T)}$$
- 计算剪枝前目标子树误判个数的期望值:

$$E(T) = N(T) \times Error(T) = \sum_{i=1}^L error(Leaf_i) + 0.5 * L$$
- 计算剪枝前目标子树误判个数的标准差:

$$std(T) = \sqrt{N(T) \times Error(T) \times (1 - Error(T))}$$
- 计算剪枝前误判上限 (即悲观误差):

$$E(T) + std(T)$$
- 计算剪枝后该结点的修正误差:

$$Error(Leaf) = \frac{error(Leaf) + 0.5}{N(T)}$$
- 计算剪枝后该结点误判个数的期望值:

$$E(Leaf) = error(Leaf) + 0.5$$
- 比较剪枝前后的误判个数, 如果满足以下不等式, 则剪枝; 否则, 不剪枝。

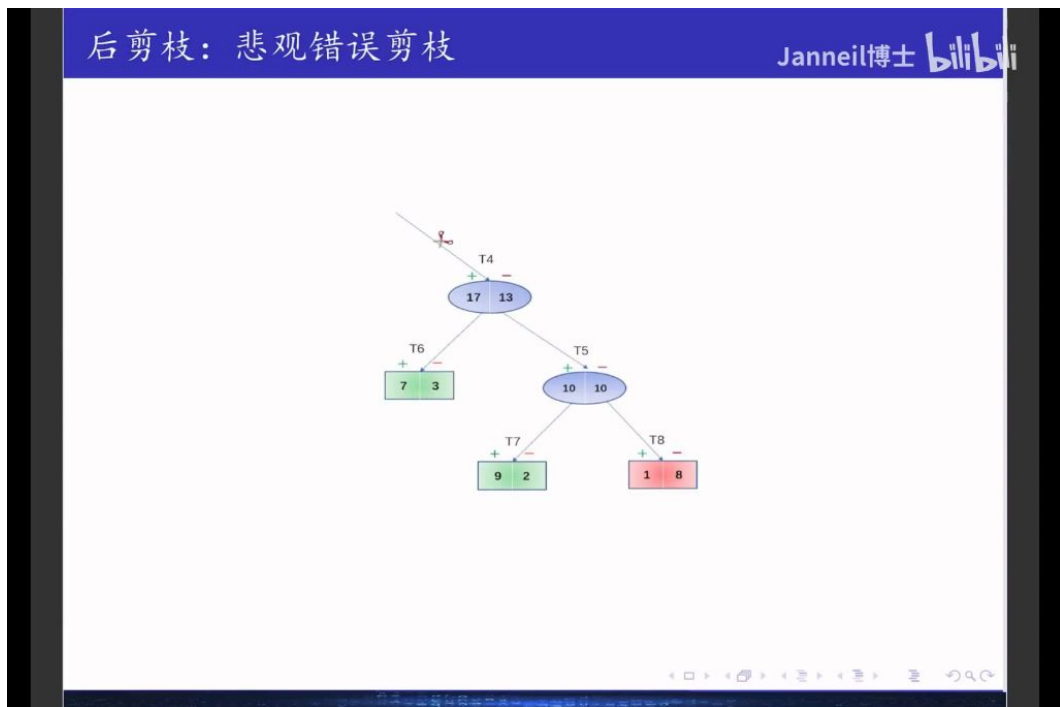
$$E(Leaf) < E(T) + std(T)$$

● 计算步骤:

- 叶子节点修正误差: $Error(Leaf_i) = \frac{error(Leaf_i) + 0.5}{N(T)}$
- 子树修正误差: $Error(T) = \frac{\sum_{i=1}^L error(Leaf_i) + 0.5 * L}{N(T)}$
- 误判期望值: $E(T) = \sum_{i=1}^L error(Leaf_i) + 0.5 * L$
- 误判标准差: $std(T) = \sqrt{N(T) \times Error(T) \times (1 - Error(T))}$
- 误判上限: $E(T) + std(T)$
- 剪枝后修正误差: $Error(Leaf) = \frac{error(Leaf) + 0.5}{N(T)}$

- 剪枝后误判期望: $E(Leaf) = error(Leaf) + 0.5$
- 剪枝条件: 当 $E(Leaf) < E(T) + std(T)$ 时执行剪枝

1) 例题1: 子树剪枝判断



● 题目解析:

- 样本分布: t4节点含30样本 (17正/13负), t6节点7正3负 (误判3), t7节点10正0负 (误判0), t8节点1正8负 (误判1)

○ 剪枝前计算:

■ 修正误差: $\frac{3 + 0 + 1 + 0.5 * 3}{30} = \frac{7.5}{30}$

■ 误判期望: $30 \times \frac{7.5}{30} = 7.5$

■ 标准差: $\sqrt{30 \times \frac{7.5}{30} \times (1 - \frac{7.5}{30})} \approx 2.37$

■ 误判上限: $7.5 + 2.37 = 9.87$

○ 剪枝后计算:

■ 修正误差: $\frac{13 + 0.5}{30} = \frac{13.5}{30}$

■ 误判期望: 13.5

- 结论: $13.5 > 9.87$, 不满足剪枝条件

2) 例题2: 修改样本数的剪枝判断

● 题目解析:

- 样本变化: 总样本27个, 误判总数7个 (3个叶节点)

○ 剪枝前计算:

■ 修正误差: $\frac{7 + 0.5 * 3}{27} = \frac{8.5}{27}$

■ 误判期望: 8.5

■ 标准差: $\sqrt{27 \times \frac{8.5}{27} \times (1 - \frac{8.5}{27})} \approx 2.41$

- 误判上限: $8.5+2.41=10.91$
- 剪枝后计算:
 - 修正误差: $\frac{10 + 0.5}{27} = \frac{10.5}{27}$
 - 误判期望: 10.5
- 结论: $10.5 < 10.91$, 满足剪枝条件

11. 西瓜数据集应用 03:07:10

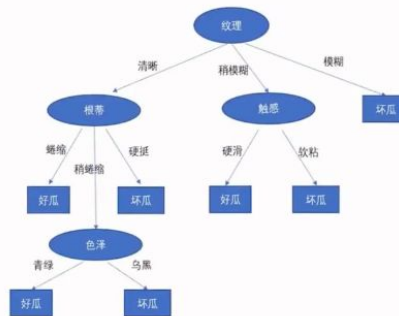
1) 西瓜数据集的决策树分类 03:07:14

- 数据结构:
 - 根节点 (纹理): 17样本 (8好/9坏)
 - 分支节点 (根蒂):
 - 更替处: 9样本 (7好/2坏)
 - 全缩: 5好0坏 (叶节点)
 - 硬挺: 0好1坏 (叶节点)

2) 例题3: 西瓜数据集剪枝判断 03:08:35

- 题目解析:
 - 剪枝前计算:
 - 叶节点数: 4个
 - 总误判: 1个
 - 修正误差: $\frac{1 + 0.5 * 4}{9} = \frac{3}{9} = \frac{1}{3}$
 - 误判期望: $9 \times \frac{1}{3} = 3$
 - 标准差: $\sqrt{9 \times \frac{1}{3} \times \frac{2}{3}} = \sqrt{2} \approx 1.414$
 - 误判上限: $3+1.414=4.414$
 - 剪枝后计算:
 - 修正误差: $\frac{2 + 0.5}{9} = \frac{2.5}{9}$
 - 误判期望: 2.5
 - 结论: $2.5 < 4.414$, 满足剪枝条件
- 关键记忆点:
 - 连续修正时, 每个叶节点加0.5
 - 比较标准是剪枝后期望与剪枝前上限
 - 实际应用中需注意样本归类的多数原则
 - 标准差计算反映误差的波动范围

12. 悲观剪枝特点 03:12:16



- 数据集要求：不需要分离剪枝数据集，适合实例较少的问题（当样本量较小时无法分割训练集和验证集）
 - 误差计算：采用连续修正值计算错误率，使方法适用性更强
 - 剪枝策略：采用自上而下的特殊剪枝策略（与多数后剪枝方法不同）
 - 效率优势：相比减少错误剪枝方法，PEP（Pessimistic Error Pruning）效率更高
 - 潜在问题：可能过度剪枝，修剪掉本应保留的分支
13. 最小误差剪枝 03:13:44

- 不需要分离剪枝数据集，有利于实例较少的问题；
- 误差使用了连续修正值，使得适用性更强
- 由于自上而下的剪枝策略，PEP效率更高；
- 可能会修剪掉不应剪掉的枝条

- 核心原理：根据剪枝前后的最小分类错误概率（ P_{error} ）决定是否剪枝
- 方法特点：
 - 缩写含义：MEP = Minimum Error Pruning（最小误差剪枝）
 - 数据集要求：仅需训练集，无需验证集（与PEP相同）

- 剪枝方向：常规自下而上策略（与PEP的自上而下形成对比）
- 实施流程：
 - 生成完整决策树
 - 从叶节点开始向上逐层考察
 - 计算每个分支剪枝前后的分类错误概率差
 - 根据最小误差原则决定剪枝操作

14. 最小误差原理 03:15:57

1) 结点T处类别概率公式

后剪枝：最小误差剪枝(MEP) Janneil博士 bilibili

原理：根据剪枝前后的最小分类错误概率来决定是否剪枝。自下而上剪枝，只需要训练集即可。



- 概率计算式：在结点T处属于类别k的概率为 $P_k(T) = \frac{n_k(T) + Pr_k(T) \times m}{N(T) + m}$
- 符号说明：
 - $n_k(T)$: 结点T处属于类别k的样本数
 - $N(T)$: 结点T处总样本数
 - $Pr_k(T)$: 结点T处类别k的先验概率
 - m : 先验概率影响因子

2) 结点T处预测错误率公式 03:16:06

- 错误率定义: $Error(T) = \min\{1 - P_k(T)\} = \min\left\{\frac{N(T) - n_k(T) + m(1 - Pr_k(T))}{N(T) + m}\right\}$
- 计算原理：取不属于各类别概率的最小值作为预测错误率

3) 结点T处类别概率计算方法 03:16:38

- 常规方法：直接使用样本比例 $P_k(T) = \frac{n_k(T)}{N(T)}$
- 贝叶斯改进：引入先验概率 $Pr_k(T)$ 和影响因子 m 进行修正
- 符号解释：
 - $Pr_k(T)$ 中的 "Pr" 来自先验概率英文 "prior" 的缩写
 - 当 $m = 0$ 时退化为常规方法

- 极端情况1: 当 $m \rightarrow 0$ 时, $P_k(T) \rightarrow \frac{n_k(T)}{N(T)}$, 先验概率无影响
- 极端情况2: 当 $m \rightarrow \infty$ 时, $P_k(T) \rightarrow Pr_k(T)$, 完全由先验概率决定
- 实际意义: m 控制先验概率对后验概率的影响程度

- **错误率本质:** 是 m 的函数, 通过优化 m 来最小化错误率
- **简化条件:** 当假设各类别先验概率相等($Pr_k(T) = \frac{1}{K}$)且 $m = K$ 时
- **简化公式:** $Error(T) = \frac{N(T) - n_k(T) + K - 1}{N(T) + K}$

后剪枝：最小误差剪枝(MEP)

Janneil博士 bilibili

- 在结点 T 处，属于类别 k 的概率

$$P_k(T) = \frac{n_k(T) + Pr_k(T) \times m}{N(T) + m}$$

- 在结点 T 处，预测错误率

$$Error(T) = \min\{1 - P_k(T)\} = \min\left\{\frac{N(T) - n_k(T) + m(1 - Pr_k(T))}{N(T) + m}\right\}$$

Navigation icons: back, forward, search, etc.

- 解题步骤:
 - 确定类别数 K 和影响因子 m
 - 计算各结点处各类别的样本数 $n_k(T)$
 - 代入简化公式计算各结点预测错误率
 - 比较剪枝前后错误率变化
 - 选择使整体错误率最小的剪枝方案
- 注意事项: 实际应用中 m 需要通过优化确定, 不能直接指定

15. 最小误差剪枝(MEP)步骤 03:24:20

$$Error(T) = \frac{N(T) - n_k(T) + K - 1}{N(T) + K}$$

步骤:

- 计算剪枝前目标子树每个叶子结点的预测错误率:

$$Error(Leaf_i) = \frac{N(Leaf_i) - n_k(Leaf_i) + K - 1}{N(Leaf_i) + K}$$

- 计算剪枝前目标子树的预测错误率:

$$Error(Leaf) = \sum_{i=1}^L w_i Error(Leaf_i) = \sum_{i=1}^L \frac{N(Leaf_i)}{N(T)} Error(Leaf_i)$$

- 计算剪枝后目标子树的预测错误率:

$$Error(T) = \frac{N(T) - n_k(T) + K - 1}{N(T) + K}$$

- 比较剪枝前后的预测错误率, 如果满足以下不等式, 则剪枝; 否则, 不剪枝。

$$Error(T) < Error(Leaf)$$

- 1) 计算剪枝前目标子树每个叶子结点的预测错误率

- 叶子节点错误率公式: $Error(Leaf_i) = \frac{N(Leaf_i) - n_k(Leaf_i) + K - 1}{N(Leaf_i) + K}$

- 分母为当前叶子节点样本总数N加上类别总数K
- 分子包含: 非主导类别样本数($N - n_k$)和类别数修正项($K - 1$)
- 计算示例: 若某叶子有11个样本($N = 11$), 其中9个属于正类($n_k = 9$), $K = 2$ 类, 则错误率为 $\frac{2 + 1}{11 + 2} = \frac{3}{13}$

- 2) 计算剪枝前目标子树的预测错误率 03:25:23

- 加权求和法: $Error(Leaf) = \sum_{i=1}^L \frac{N(Leaf_i)}{N(T)} Error(Leaf_i)$

- 权重 w_i 为叶子节点样本占比 $\frac{N(Leaf_i)}{N(T)}$
- 需先计算每个叶子的错误率再按样本量加权
- 示例: 若某子树含两个叶子 (11个和9个样本), 错误率分别为 $\frac{3}{13}$ 和 $\frac{2}{11}$, 则整体错误率为 $\frac{11}{20} \times \frac{3}{13} + \frac{9}{20} \times \frac{2}{11} \approx 0.2087$

- 3) 计算剪枝后目标子树的预测错误率 03:26:10

- 剪枝后错误率公式: $Error(T) = \frac{N(T) - n_k(T) + K - 1}{N(T) + K}$

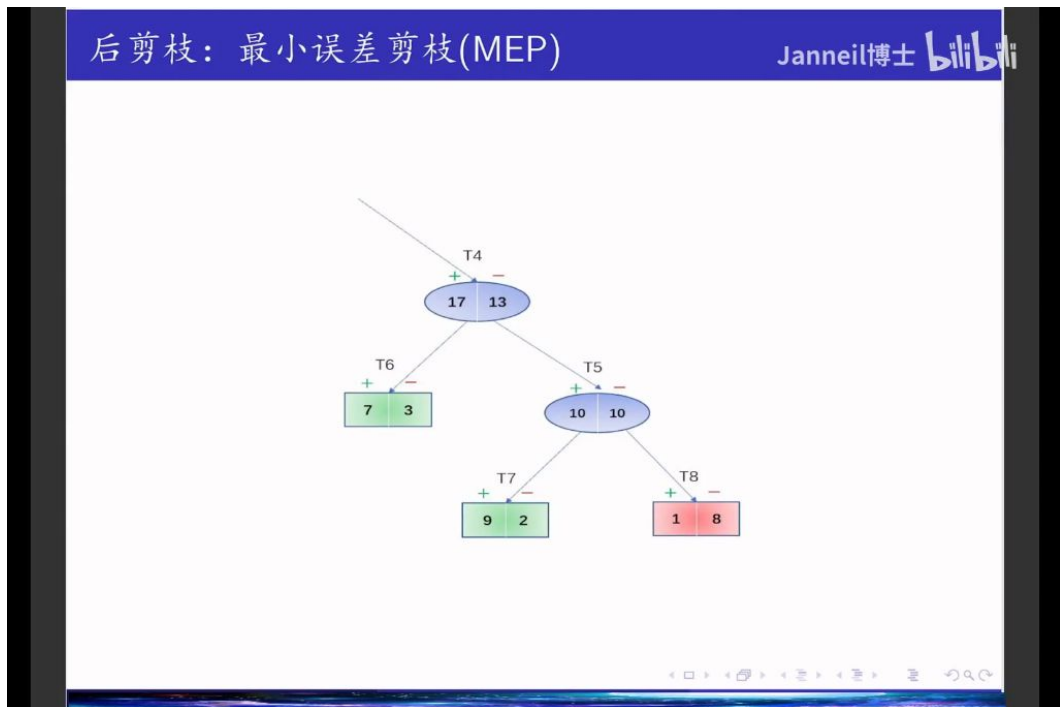
- 将整个子树视为单一叶子节点计算
- 示例: 若合并后节点含20个样本 (10正10负), $K = 2$, 则错误率为 $\frac{10 + 1}{20 + 2} = 0.5$

- 4) 比较剪枝前后的预测错误率 03:26:53

- 剪枝条件: 当且仅当 $Error(T) < Error(Leaf)$ 时执行剪枝
- 比较时需保留四位小数确保精度

- 若剪枝后错误率升高（如0.2087→0.5）则保留原结构
- 决策依据是错误率最小化原则

16. 最小误差剪枝例题 03:27:28



● 题目解析

- 案例背景：含T5-T8节点的二分类决策树（正类绿色/负类红色）
- 关键数据：

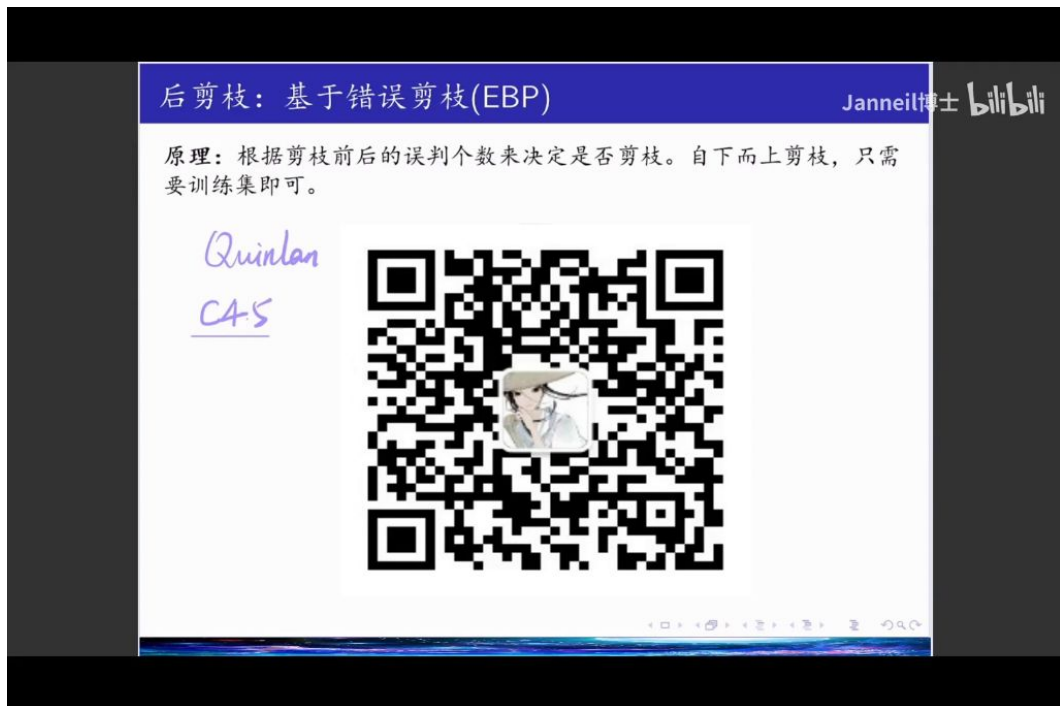
- T7节点：9正2负，错误率 $\frac{3}{13}$
- T8节点：1正8负，错误率 $\frac{2}{11}$
- 子树总样本20个（T7:11, T8:9）

- 计算过程：

- 剪枝前加权错误率： $\frac{11}{20} \times \frac{3}{13} + \frac{9}{20} \times \frac{2}{11} = 0.2087$
- 剪枝后（归为正类）错误率： $\frac{10+1}{20+2} = 0.5$

- **决策结果：**0.2087 < 0.5，故不剪枝
- **易错点：**需注意归并后节点的类别判定（本例正负样本数相同，默认归为正类）

17. 基于错误剪枝 03:33:23



-
- 提出者: 由Quinlan在1992年出版的《C4.5: Programs for Machine Learning》中提出
- 特点: 属于后剪枝方法，只需训练集即可完成剪枝过程

18. 错误剪枝原理 03:34:08

1) 错误剪枝的简写与含义 03:34:14

- **EBP全称:** Error Based Pruning
- **核心思想:** 基于剪枝前后的误判个数来决定是否剪枝

2) 基于错误减值方法的原理 03:34:40

- **决策依据:** 采用误判个数的上界而非期望值
- **剪枝方向:** 自下而上进行剪枝
- **数据需求:** 仅需训练集，无需额外验证集

3) 误判率上界的概念与计算 03:35:36

后剪枝：基于错误剪枝(EBP)

Janneil博士 bilibili

置信水平 α 下每个节点的误判率上界：

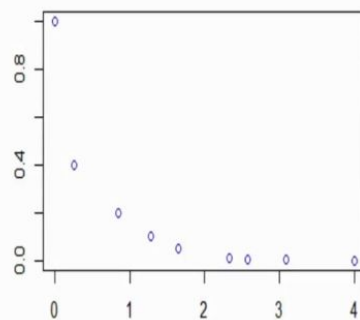
$$U_{\alpha}(e(T), N(T)) = \frac{e(T) + 0.5 + \frac{q_{\alpha}^2}{2} + q_{\alpha} \sqrt{\frac{(e(T)+0.5)(N(T)-e(T)-0.5)}{N(T)} + \frac{q_{\alpha}^2}{4}}}{N(T) + q_{\alpha}^2}$$

- 公式组成：
 - $e(T)$: 节点T的误判个数
 - $N(T)$: 节点T的样本总数
 - q_{α} : 置信水平 α 对应的上分位点
 - 0.5: 二项分布到正态分布的连续性修正项
- 计算过程: 通过复杂公式计算在给定置信水平下的误判率上界
- 4) 置信水平与上分位点 03:40:33

后剪枝：基于错误剪枝(EBP)

Janneil博士 bilibili

置信水平	0	0.001	0.005	0.01	0.05	0.10	0.20	0.40	1.00
上分位点	4.00	3.09	2.58	2.33	1.65	1.28	0.84	0.25	0.00



- 定义: $P(x \geq q_{\alpha}) = \alpha$, 其中 x 服从标准正态分布
- 特殊值:
 - $\alpha = 0.20$ 对应 $q_{\alpha} = 0.84$
 - $\alpha = 0.40$ 对应 $q_{\alpha} = 0.25$

- **Quinlan默认值:** 采用25%置信水平($\alpha = 0.25$)

5) 线性差值的方法 03:42:02

- **应用场景:** 当需要计算表中未列出的置信水平对应的分位点时
 - **计算示例:**
 - 已知 $\alpha = 0.20$ 对应0.84, $\alpha = 0.40$ 对应0.25
 - 求 $\alpha = 0.25$ 对应的 q_α :
 - 通过比例关系计算得到 $q_{0.25} = 0.6925$
 - **实际应用:** 在C4.5算法中默认使用25%置信水平, 对应 $q_\alpha = 0.6925$
- ## 19. 错误剪枝步骤 03:45:38

后剪枝：基于错误剪枝(EBP)
Janneil博士 bilibili

步骤:

- 计算剪枝前目标子树每个叶子节点的误判率上界:

$$U_{CF}(e(Leaf_i), N(Leaf_i)) = \frac{e(Leaf_i) + 0.5 + \frac{q_\alpha^2}{2} + q_\alpha \sqrt{\frac{(e(Leaf_i)+0.5)(N(Leaf_i)-e(Leaf_i)-0.5)}{N(Leaf_i)} + \frac{q_\alpha^2}{4}}}{N(Leaf_i) + q_\alpha^2}$$
- 计算剪枝前目标子树的误判个数上界:

$$E(Leaf) = \sum_{i=1}^L N(Leaf_i) U_{CF}(e(Leaf_i), N(Leaf_i))$$
- 计算剪枝后目标子树的误判率上界:

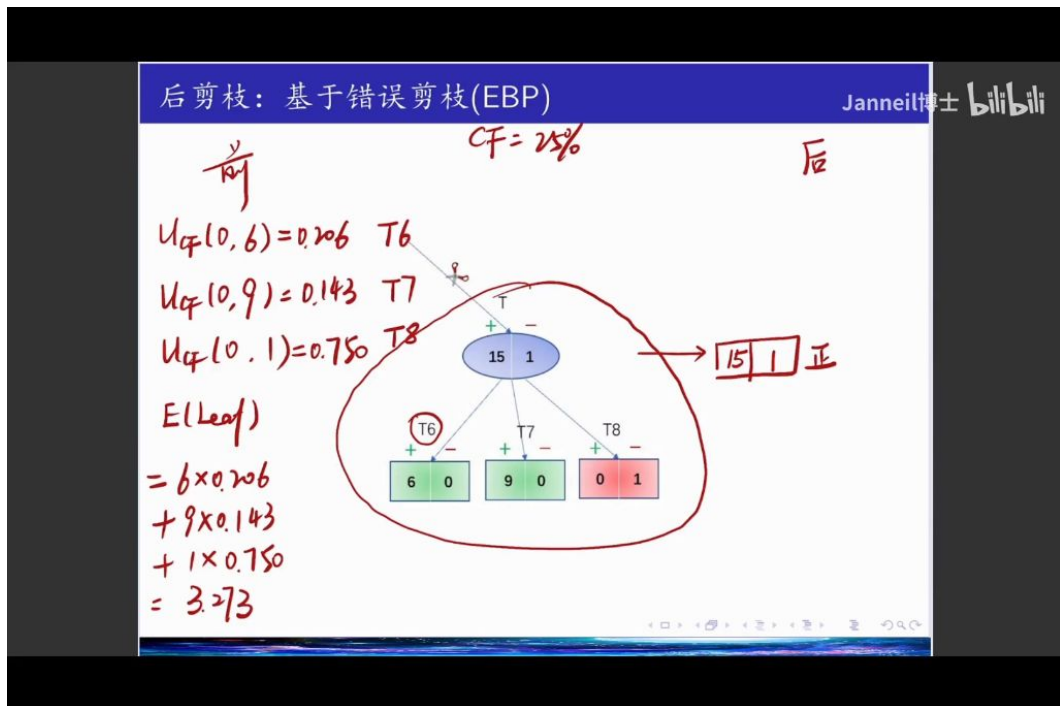
$$U_{CF}(e(T), N(T)) = \frac{e(T) + 0.5 + \frac{q_\alpha^2}{2} + q_\alpha \sqrt{\frac{(e(T)+0.5)(N(T)-e(T)-0.5)}{N(T)} + \frac{q_\alpha^2}{4}}}{N(T) + q_\alpha^2}$$
- 计算剪枝后目标子树的误判个数上界:

$$E(T) = N(T) U_{CF}(e(T), N(T))$$
- 比较剪枝前后的误判个数上界, 如果满足以下不等式, 则剪枝; 否则, 不剪枝。

$$E(T) < E(Leaf)$$

- **误判率上界计算:** 使用公式 $U_{CF}(e(leaf_i), N(leaf_i))$ 计算每个叶子节点的误判率上界, 其中 q_α 对应置信水平 (默认25%)
- **剪枝前误判数:** $E(Leaf) = \sum_{i=1}^L N(leaf_i) U_{CF}(e(leaf_i), N(leaf_i))$ 累加所有叶子节点的误判数上界
- **剪枝后计算:** 用相同公式计算替换为叶节点后的误判率 $U_{CF}(e(T), N(T))$ 和误判数 $E(T) = N(T) U_{CF}(e(T), N(T))$
- **剪枝条件:** 当且仅当 $E(T) < E(Leaf)$ 时执行剪枝

20. 错误剪枝例题 03:48:30



- 例题解析：
 - 剪枝前有3个叶节点（T6/T7/T8），样本数分别为6/9/1
 - 计算得各节点误判率上界：0.206/0.143/0.750
 - 总误判数上界：6×0.206+9×0.143+1×0.75=3.273
 - 剪枝后（16个样本，1个误判）误判率0.157，误判数上界16×0.157=2.512
 - 决策：因2.512<3.273，执行剪枝
21. CART算法介绍 03:52:47

统计学习方法

第五章：决策树——CART算法

Author: Janneil

E-mail: 258497068@qq.com

July 22, 2021

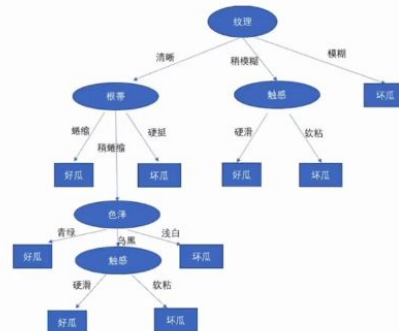
Janneil博士 bilibili

- 提出背景：Breiman于1984年提出，入选十大算法之一
 - 全称含义：Classification And Regression Tree（分类与回归树）
 - 核心功能：同时解决分类（离散输出）和回归（连续输出）问题
22. CART算法特点 03:53:29

CART算法

Janneil博士 bilibili

Breiman 1984年提出CART (Classification and Regression Tree) 算法, 同样需要一个用以选择特征的标准, 包括树的生成和剪枝。



- 二叉树结构: 所有决策节点均为二叉分裂 (是/否)
- 分裂标准: 需要专门的特征选择标准
- 完整流程: 包含树生成和剪枝两个阶段

23. 二叉树结构 03:54:33

1) 多叉树转换 03:56:40

- 转换原理: 任何多叉分裂可通过多次二叉分裂实现。例如纹理分类 (清晰/稍模糊/模糊) 可先分"是否清晰", 再将"不清晰"分"是否非常模糊"
- 结构约定: 左分支为"是", 右分支为"否"

2) 二叉树优势 03:58:04

- 简化复杂度: 用简单结构表示复杂决策逻辑
- 扩展性强: 通过二分类组合可处理多分类问题
- 计算机友好: 类似二进制逻辑 (0/1对应否/是), 便于程序实现

3) 与布尔逻辑类比 03:59:15

- 逻辑对应: 1→是/接通, 0→否/断开
- 运算体系: 基于简单布尔运算构建复杂逻辑结构
- 实际意义: 为计算机逻辑运算奠定数学基础

24. 基尼指数定义 04:00:08

- 核心概念: 基尼指数是CART算法中用于特征选择的标准, 与ID3算法的信息增益和C4.5的信息增益比类似, 都是用来度量不同特征的分类能力。
- 与经济指数的区别: 虽然名称相同, 但机器学习中的基尼指数与经济领域衡量收入差距的基尼指数不同, 它主要用于表示数据的不确定性。
- 不确定性度量: 基尼指数越大表示不确定性越高, 越小则确定性越强。

25. 基尼指数计算 04:02:21

1) 多分类问题与多项分布 04:02:23

- 概率分布: 在多分类问题中, 样本点属于第 k 类的概率为 p_k , 这构成一个多项分布。

- **错误分类概率:** 不属于第k类的概率为 $1 - p_k$ ，这部分概率反映了分类的不确定性。

2) 基尼指数的定义与计算 04:03:07

- **加权求和公式:** 基尼指数定义为 $Gini(p) = \sum_{k=1}^K p_k(1 - p_k)$ ，其中 p_k 是样本属于第k类的概率。
- **化简形式:** 可以展开为 $Gini(p) = 1 - \sum_{k=1}^K p_k^2$ ，因为 $\sum_{k=1}^K p_k = 1$ 。
- **不确定性解释:** 这个值越大，表示样本的分类不确定性越高。

3) 二分类问题下的基尼指数 04:05:20

- **简化公式:** 对于二分类问题，若样本属于第一类的概率为 p ，则基尼指数简化为 $Gini(p) = 2p(1 - p)$ 。
- **对称性:** 公式中 p 和 $1 - p$ 的对称性反映了二分类问题的特性。

4) 基尼指数的估计值计算 04:06:16

- **概率估计:** 在实际应用中， p_k 可以用样本中属于第k类的比例来估计，即 $\hat{p}_k = \frac{|C_k|}{n}$ ，其中 $|C_k|$ 是第k类的样本数， n 是总样本数。
- **经验基尼指数:** 将估计值代入理论公式，得到样本集的基尼指数 $Gini(D) = 1 - \sum_{k=1}^K \left(\frac{|C_k|}{|D|}\right)^2$ 。

26. 特征选择标准 04:07:47

- **特征条件下的基尼指数:** 给定特征 a ，可以将样本集 D 划分为 D_1 和 D_2 两部分，计算每部分的基尼指数后加权求和： $Gini(D, a) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$ 。
- **二叉树特性:** CART算法生成的是二叉树，因此特征划分总是将样本集分成两部分。
- **选择标准:** 选择使 $Gini(D, a)$ 最小的特征作为划分标准，这与信息增益最大化是类似的思路。

十六、分类树 04:09:19

1. 基尼指数计算 04:09:25

特征A 条件下，样本集 D 的基尼指数为：

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

- 定义：基尼指数用于衡量数据集的不纯度，计算公式为 $Gini(D) = 2p(1-p)$ ，其中 p 为某类样本所占比例
 - 计算示例：10个桃子（5好吃5不好吃）的基尼指数为 $2 \times 0.5 \times 0.5 = 0.5$
 - 二分类简化：对于二分类问题可直接使用 $2p(1-p)$ 公式计算，避免复杂的求和运算
2. 特征选择 04:10:29

- 划分原理：根据特征值将数据集 D 划分为 D_1 和 D_2 两个子集

加权计算：特征A条件下的基尼指数计算公式为：

$$\frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

3. 最优特征选择 04:11:23

- 选择标准：比较不同特征的基尼指数，选择使基尼指数最小的特征
- 数学表达： $\min_A Gini(D, A)$ ，即寻找使基尼指数最小化的特征A

4. 桃子分类案例 04:11:34

- 甜度特征：
 - 划分点：甜度 > 0.2
 - D_1 (> 0.2)：5好吃1不好吃，基尼指数 $\frac{10}{36}$
 - D_2 (≤ 0.2)：0好吃4不好吃，基尼指数0
 - 加权结果： $\frac{6}{10} \times \frac{10}{36} + \frac{4}{10} \times 0 \approx 0.17$
- 硬度特征：
 - 硬桃子：2好吃3不好吃
 - 软桃子：3好吃2不好吃
 - 两组基尼指数均为 $\frac{12}{25}$
 - 加权结果： $\frac{5}{10} \times \frac{12}{25} + \frac{5}{10} \times \frac{12}{25} = 0.48$

5. 特征划分效果比较 04:12:50

- 比较结果：甜度特征 (0.17) < 硬度特征 (0.48)
 - 结论：甜度特征更有利于分类，因其基尼指数更小，分类更彻底
6. 基尼指数图形解释 04:14:53

- 曲线特性： $Gini = 2p(1 - p)$ 是二次曲线
 - 极值点：在 $p=0.5$ 时取得最大值0.5
 - 边界情况：当 p 趋近0或1时，基尼指数趋近0，表示确定性高
7. 甜度特征分析 04:15:22

- 优势体现：在 D_2 中 $p=0$ （完全确定）
- 分类效果：能完全区分出不好吃的桃子（甜度 ≤ 0.2 时）

十七、CART分类树算法 04:21:26

1. 算法输入输出 04:21:31

- 输入：
 - 训练数据集D
 - 特征集A
 - 停止条件阈值 ϵ
- 输出：CART决策树T

2. 算法步骤 04:21:45

- 特征选择：
 - 对每个特征 A_g ，计算所有可能取值 a_g 的基尼指数
 - 选择使基尼指数最小的 a_g 作为最优切分点
- 特征比较：选择所有特征中基尼指数最小的作为最优特征
- 节点分裂：根据最优特征和切分点生成两个子节点
- 递归构建：对子节点重复上述过程

3. 停止条件 04:22:21

- 样本数量：节点样本数小于预定阈值（如 $\epsilon=5$ ）
- 纯度阈值：样本集基尼指数小于预定值（基本属于同一类）
- 特征耗尽：没有更多可用特征
- 默认情况：在示例中主要使用特征耗尽作为停止条件

十八、例题解说：贷款申请案例 04:27:23

1. 数据集介绍 04:27:32

CART分类树：例题解说

Janneil博士 bilibili

特征A条件下，样本集D的基尼指数为：

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

表 5.1 贷款申请样本数据集

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

D A ₁	年龄	个数	是否同意贷款	
			否	是
D ₁	青年	5	3	2
D ₂	中年	5	2	3
D ₃	老年	5	1	4

- 样本构成：包含15个贷款申请样本，分类结果为"是"（同意贷款）和"否"（不同意贷款）
 - 特征集合：包含年龄、有工作、有自己的房子、信贷情况4个特征
 - 目标变量：最终分类结果（是否给予贷款）
2. 年龄特征分析 04:28:03

Handwritten calculations for Gini index:

$$Gini(D_1) = 2 \times \frac{3}{5} \times \frac{2}{5} = \frac{12}{25}$$

$$w_1 = \frac{5}{15} = \frac{1}{3}$$

$$Gini(D_2) = 2 \times \frac{2}{5} \times \frac{3}{5} = \frac{12}{25}$$

$$w_2 = \frac{5}{15} = \frac{1}{3}$$

$$Gini(D_3) = 2 \times \frac{1}{5} \times \frac{4}{5} = \frac{16}{25}$$

$$w_3 = \frac{5}{15} = \frac{1}{3}$$

$$Gini(D, A) = \frac{1}{3} \times \frac{12}{25} + \frac{1}{3} \times \frac{12}{25} + \frac{1}{3} \times \frac{16}{25} = \frac{42}{100} = 0.42$$

Final result circled: 0.44

Janneil (258497068@qq.com) 统计学习方法 July 22, 2021 9 / 14

- 划分方式：CART分类树是二叉树，需将年龄（青年/中年/老年）转化为二划分
- 基尼指数计算：

- 青年划分： $Gini(D, A) = \frac{5}{15} \times \frac{12}{25} + \frac{10}{15} \times \frac{42}{100} = 0.44$
- 中年划分： $Gini(D, A) = \frac{5}{15} \times \frac{12}{25} + \frac{10}{15} \times \frac{48}{100} = 0.48$

- 老年划分: $Gini(D, A) = 0.44$
 - 最优划分点: 青年或老年 (基尼指数均为0.44)
3. 有工作特征分析 04:34:58

CART分类树: 例题解说

Janneil博士

特征A 条件下, 样本集 D 的基尼指数为: :

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

表 5.1 贷款申请样本数据集

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

$D A_2$	有工作	个数	是否同意贷款	
			否	是
D_1	是	5	0	5
D_2	否	10	6	4

- 二分特性: 天然分为"有工作"和"无工作"两类
- 基尼指数计算:
 - 有工作部分: $Gini(D_1) = 0$ (全部同意贷款)
 - 无工作部分: $Gini(D_2) = 2 \times \frac{4}{10} \times \frac{6}{10} = \frac{48}{100}$
 - 加权结果: $\frac{5}{15} \times 0 + \frac{10}{15} \times \frac{48}{100} = 0.32$

4. 有自己的房子特征分析 04:38:50

CART分类

特征A条件下，样本集D

Janneil博士 bilibili

$Gini(D_1) = 0$

$Gini(D_2) = 2 \times \frac{6}{9} \times \frac{3}{9} = \frac{36}{81}$

$Gini(D, A_2) = 0 + \frac{9}{15} \times \frac{36}{81} = 0.27$

D A ₁	有自己的房子	个数	是否同意贷款	
			否	是
D ₁	是	6	0	6
D ₂	否	9	3	6

- 基尼指数计算：
 - 有房产部分： $Gini(D_1) = 0$ (全部同意贷款)
 - 无房产部分： $Gini(D_2) = 2 \times \frac{6}{9} \times \frac{3}{9} = \frac{36}{81}$
 - 加权结果： $\frac{9}{15} \times \frac{36}{81} = 0.27$

5. 信贷情况特征分析 04:40:20

特征A条件下，样本集D的基尼指数为：

年龄： $Gini(D, A_1) = 0.44$

$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$

$Gini(D, A) = \frac{5}{15} \times 0 + \frac{10}{15} \times \frac{36}{81} = 0.44$

$Gini(D_1) = 0$

$Gini(D_2) = 2 \times \frac{5}{10} \times \frac{5}{10} = \frac{10}{10} = 1$

$Gini(D, A) = \frac{5}{15} \times 0 + \frac{10}{15} \times 1 = \frac{10}{15} = \frac{2}{3} \approx 0.67$

$Gini(D, A) = 0.44$

D A ₁	年龄	个数	是否同意贷款	
			否	是
D ₁	青年	5	3	2
D ₂	中年	5	2	3
D ₃	老年	5	1	4

- 三分特性：分为"非常好"、"好"、"一般"三类
- 基尼指数结果：
 - 非常好划分：0.36
 - 好划分：0.47

- 一般划分：0.32
 - 最优划分点：一般（基尼指数0.32）
6. 最优特征选择与决策树生成 04:41:23

CART分类树：例题解说
Janneil博士 bilibili

特征A 条件下，样本集 D 的基尼指数为：

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

编号	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

- 特征比较：
 - 年龄：0.44
 - 有工作：0.32
 - 自有房产：0.27（最优）
 - 信贷情况：0.32
- 决策树结构：
 - 根节点：按"有自己的房子"划分
 - 是→同意贷款（6个样本）
 - 否→进入子节点
 - 子节点：按"有工作"划分
 - 是→同意贷款
 - 否→不同意贷款
- 终止条件：子节点基尼指数为0时停止分裂

十九、回归树 04:46:20

1. 回归树模型 04:46:28

- 结构组成：回归树作为树模型，包含根节点、内部节点和叶子节点，最终结果对应叶子节点
- 核心问题：需要解决如何将连续输出变量对应到离散的叶子节点上
- 解决方案：通过将连续数据分割为小区间实现，但需注意必须基于输入变量而非输出变量进行划分

2. 输入空间划分 04:47:21

- 划分原则：
 - 监督要求：必须根据输入变量而非输出变量划分，否则会退化为无监督问题

- **代表值计算:** 每个划分单元 R_m 对应一个代表值 c_m , 通常取该单元内样本输出的平均值
 - **数学表示:** 回归树可表示为分段函数 $f(x) = \sum_{m=1}^M c_m I(x \in R_m)$, 其中 $I()$ 为指示函数
3. 平方误差计算 04:48:46
- **误差度量:** 采用平方误差 $\sum (y_i - c_m)^2$ 作为划分质量的评价标准
 - **代表值确定:** 使平方误差最小的 c_m 即为该单元内所有样本输出 y_i 的平均值
4. 最优切分点选择 04:49:20
- 1) 回归树的表示与理解 04:49:27
- **函数表达:** 回归树本质是分段常数函数, 每个区域 R_m 对应固定输出值 c_m
 - **二叉树实现:** 虽然最终可能有多个叶子节点, 但通过递归二叉划分实现
- 2) 多叉树与二叉树的转换 04:51:17
- **转换原理:** 任何多叉树都可表示为二叉树形式
 - **递归划分:** 每次将当前区域二分, 未达停止条件则继续划分, 最终形成多个叶子节点
- 3) 回归树划分与代表值计算 04:53:52
- **划分过程:**
 - 第一次划分得到 $R_1^{(1)}$ 和 $R_2^{(1)}$
 - 对未达停止条件的区域继续划分 (如 $R_1^{(2)}, R_2^{(2)}$)
 - 最终形成多个终端区域 (如6个叶子节点)
 - **代表值计算:** 每个终端区域 R_m 的代表值 c_m 取该区域样本输出的平均值
- 4) 最优切分点与切分变量的选择 04:55:27
- **双优化问题:** 需要同时确定最优切分变量 j 和最优切分点 s
 - **选择步骤:**
 - 固定变量 j , 遍历所有可能切分点 s , 计算平方误差
 - 选择使平方误差最小的 s 作为该变量下的最优切分点
 - 比较所有变量的最小平方误差, 选择最小的对应变量作为最优切分变量
5. 桃子评分案例 04:57:22
- 1) 切分变量和切分点的选取过程 04:57:24

CART回归树：例题解说
Janneil博士 bilibili

甜度	0.05	0.15	0.25	0.35	0.45
评分	5.5	7.6	9.5	9.7	7.0

Janneil (258497068@qq.com)
统计学习本质
July 22, 2021 18 / 23

-
- 样本数据:
 - 甜度: [0.05,0.15,0.25,0.35,0.45]
 - 评分: [5.5,7.6,9.5,9.7,8.2]
- 切分点评估:
 - $s=0.1$: R_1 均值为5.5, R_2 均值为8.75, 总误差3.09
 - $s=0.2$: 总误差3.53
 - $s=0.3$: 总误差9.13
 - $s=0.4$: 总误差11.52
- 最优选择: $s=0.1$ 时误差最小(3.09), 确定为最优切分点

2) 例题1:桃子评分案例 04:58:15

- 初始划分: 按 $s=0.1$ 将数据分为两部分
 - R_1 : (0.05,5.5)
 - R_2 : 其余4个样本点
- 代表值计算:
 - $c_1 = 5.5$
 - $c_2 = 8.75$
- 树结构表示:
 - if $x \leq 0.1$ then 5.5
 - else 8.75
- 进一步划分: 可在 R_2 上继续划分形成更复杂的树结构

6. 回归树生成算法 05:03:32

1) 回归树生成算法的输入与输出 05:03:36

回归树
Janneil博士 bilibili

输入: 训练数据集 D , 停止条件
输出: CART决策树 T

- 从根节点出发, 进行操作, 构建二叉树;
- 结点处的训练数据集为 D , 计算变量的最优切分点, 并选择最优变量。

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$

- 在第 j 变量下, 对其可能取的每个值 s , 根据样本点分割成 R_1 和 R_2 两部分, 计算切分点为 s 时的平方误差。
- 选择平方误差最小的那个值作为该变量下的最优切分点。
- 计算每个变量下的最优切分点, 并比较在最优切分点下的每个变量的平方误差, 选择平方误差最小的那个变量, 即最优变量。
- 根据最优特征与最优切分点 (j, s) , 从现结点生成两个子结点, 将训练数据集依变量配到两个子结点中去, 得到相应的输出值。

$$R_1(j, s) = \{x | x^{(j)} \leq s\}, \quad R_2(j, s) = \{x | x^{(j)} > s\}$$

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m(j,s)} y_i, \quad x \in R_m, \quad m = 1, 2$$

- 继续对两个子区域调用上述步骤, 直至满足停止条件, 即生成CART决策树。

$$f(x) = \sum_{m=1}^M \hat{c}_m I(x \in R_m)$$

Janneil (258497068@qq.com)
统计学习方法
July 22, 2021 17 / 23

- 输入: 训练数据集 D , 停止条件
 - 输出: CART决策树 T
- 2) 回归树生成算法的具体步骤 05:04:09
- 初始化: 从根节点开始, 当前节点包含全部训练数据
 - 最优划分选择:
 - 对每个变量 j , 寻找最优切分点 s 使平方误差最小
 - 选择使误差最小的变量和切分点组合 (j, s)
 - 区域划分:
 - $R_1(j, s) = \{x | x^{(j)} \leq s\}$
 - $R_2(j, s) = \{x | x^{(j)} > s\}$
 - 代表值计算:
 - $c_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$
 - 递归构建: 对子节点重复上述过程直到满足停止条件
- 3) 例题1: 回归树生成算法的桃子例子 05:07:20

- 单变量简化: 甜度作为唯一输入变量, 简化了变量选择过程
- 递归构建:
 - 第一层划分: $s=0.1$
 - 第二层可在 R_2 上继续划分 (如 $s=0.3$)
 - 最终形成包含3个叶子节点的回归树
- 停止条件: 可根据需求设定, 如最小样本数或最大深度

二十、CART树剪枝 05:08:52

1. 代价复杂度剪枝 05:08:59

- 损失函数构成: 由两部分组成, 第一部分反映模型拟合代价 $C(T_t)$, 第二部分反映模型复杂度 $\alpha |T_t|$, 其中 α 是惩罚参数
- 参数作用:
 - 当 $\alpha = 0$ 时: 生成完整决策树, 只考虑拟合效果

- 当 $\alpha \rightarrow +\infty$ 时：生成单结点树，强调泛化能力但拟合效果差
 - **临界值计算**: 通过比较剪枝前后损失函数相等时的 α 值，确定剪枝临界点
2. 损失函数分析 05:09:28

- **子树表示**: 以 T_t 表示以 t 为根结点的子树， $|T_t|$ 表示叶结点个数
- **剪枝条件**:
 - 剪枝前损失: $C(T_t) + \alpha|T_t|$
 - 剪枝后损失: $C(t) + \alpha$
 - 当剪枝后损失 $\leq$ 剪枝前损失时执行剪枝
- **关键公式**: 临界值 $\alpha = \frac{C(t) - C(T_t)}{|T_t| - 1}$ ，称为 $g(t)$ 函数

3. 剪枝算法步骤 05:10:52

CART树的剪枝：算法
Janneil博士 bilibili

输入：CART算法生成的决策树。
输出：最优决策树 T_{α} 。

- (1) 设 $k=0, T=T_0$ 。
- (2) 设 $\alpha=+\infty$ 。
- (3) 自下而上地对各内部结点 t 计算 $C(T_t), |T_t|$ 以及

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}, \quad \alpha = \min(\alpha, g(t))$$

这里， T_t 表示以 t 为根结点的子树， $C(T_t)$ 是对训练数据的预测误差， $|T_t|$ 是 T_t 的叶结点个数。

- (4) 自上而下地访问内部结点 t ，如果有 $g(t) = \alpha$ ，进行剪枝，并对叶结点 t 以多数类法决定其类，得到树 T 。
- (5) 设 $k=k+1, \alpha_k = \alpha, T_k = T$ 。
- (6) 如果 T 不是由根结点单独构成的树，则回到步骤(4)。
- (7) 采用交叉验证法在子树序列 $T_0, T_1, \dots, T_n$ 中选取最优子树 T_{α} 。

Janneil (258497068@qq.com)
统计学习方法
July 22, 2021 21 / 23

- **初始化**:
 - 设 $k=0, T=T_0$ （完整树）
 - $\alpha = +\infty$
- **计算过程**:
 - 自下而上计算各内部结点 t 的 $C(T_t)$ 、 $|T_t|$ 和 $g(t)$
 - 更新 $\alpha = \min(\alpha, g(t))$
 - 对 $g(t) = \alpha$ 的结点进行剪枝
 - 递归执行直到得到树序列 $T_0, T_1, \dots, T_n$
- **嵌套特性**: 生成的树序列满足 T_{i+1} 是 T_i 的子树

4. 最优子树选择 05:18:51

- **选择方法**: 采用交叉验证法在子树序列中选择最优
- **评价标准**:
 - 分类问题：使用基尼指数
 - 回归问题：使用平方误差
- **算法输出**: 最优决策树 T_{α} ，与特定 α 值对应

- **本质联系：**虽然不同算法标准不同（信息增益、信息增益比、基尼指数、平方误差），但都源于对不确定性的度量

二十一、知识小结

知识点	核心内容	关键算法/指标	应用示例	难度系数
决策树基本概念	树形结构分类模型，包含根节点、内部节点和叶节点	信息增益/基尼指数	贷款审批决策树	★★
ID3算法	使用 信息增益 选择特征，倾向于选择取值多的特征	信息增益计算	情人节活动安排案例	★★★
C4.5算法	使用 信息增益比 改进ID3，解决取值偏向问题	信息增益比计算	贷款申请数据集	★★★★
CART分类树	二叉树结构，使用 基尼指数 作为分裂标准	基尼指数公式	桃子分类案例	★★★★
CART回归树	处理连续值预测，使用 平方误差 最小化	均方误差(MSE)	水蜜桃甜度预测	★★★★
预剪枝方法	在生长过程中提前停止：深度限制/样本数阈值/误差率检查	测试集误差率比较	西瓜数据集预剪枝	★★★
后剪枝方法	生成完整树后修剪：降低错误剪枝/悲观错误剪枝/代价复杂度剪枝	代价复杂度公式 α 计算	贷款决策树剪枝	★★★★
过拟合处理	平衡拟合能力与泛化能力，通过剪枝控制模型复杂度	损失函数=代价+ α *复杂度	果树修剪类比	★★★★
特征空间划分	输入空间被划分为互斥且完备的单元，每个单元对应一条路径	if-then规则集合	贷款特征空间划分	★★★
条件概率分布	决策树表示给定特征条件下类的条件概率分布	$P(Y X)$ 估计	下雨概率预测案例	★★★★