

第5讲：决策树

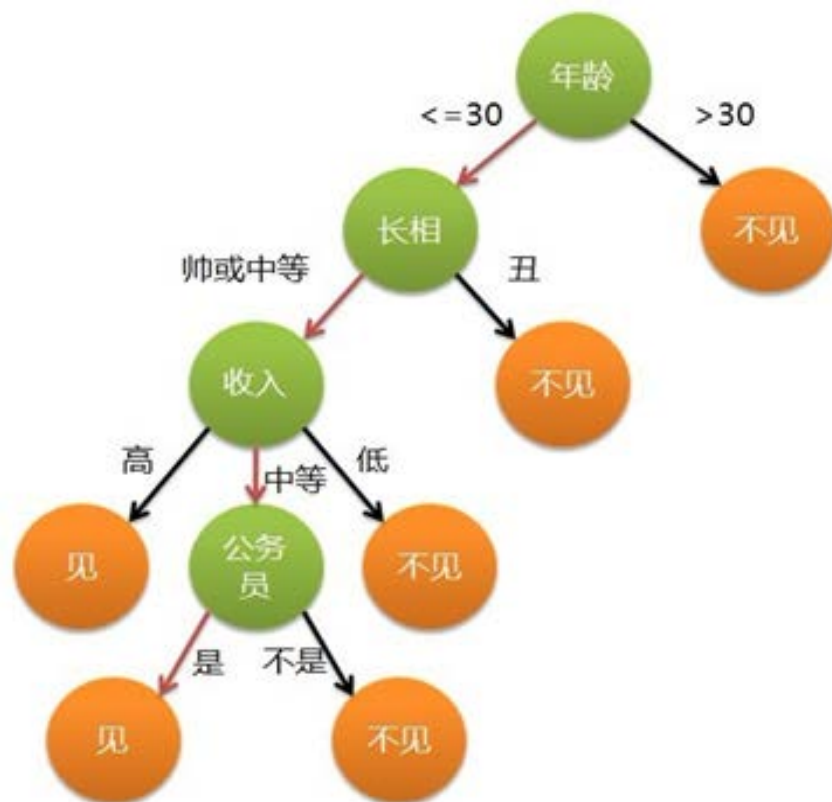
Lecture 5: Decision Tree

张永飞

2018年10月17日

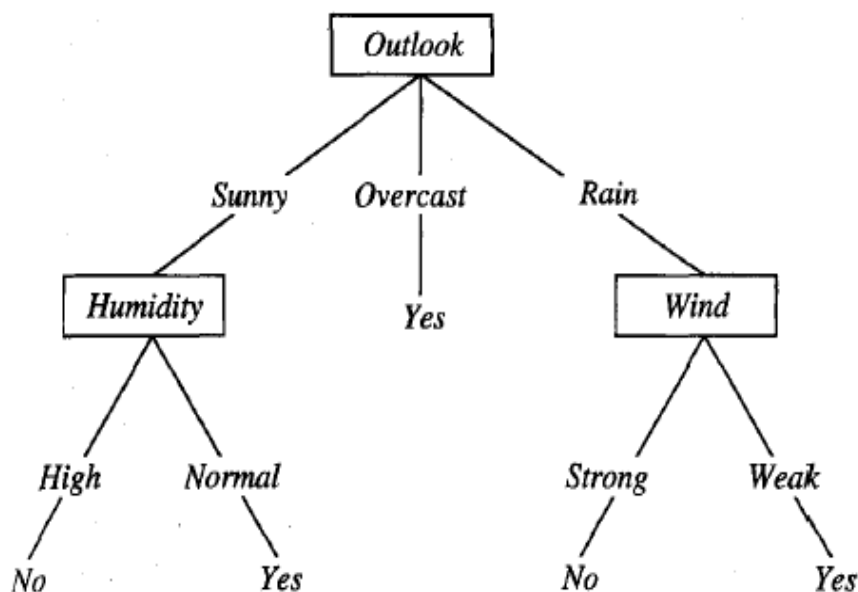
引例

● 引例1：相亲



30岁以下、不丑、高收入或
中等收入公务员

● 引例2：天气是否适合打网球



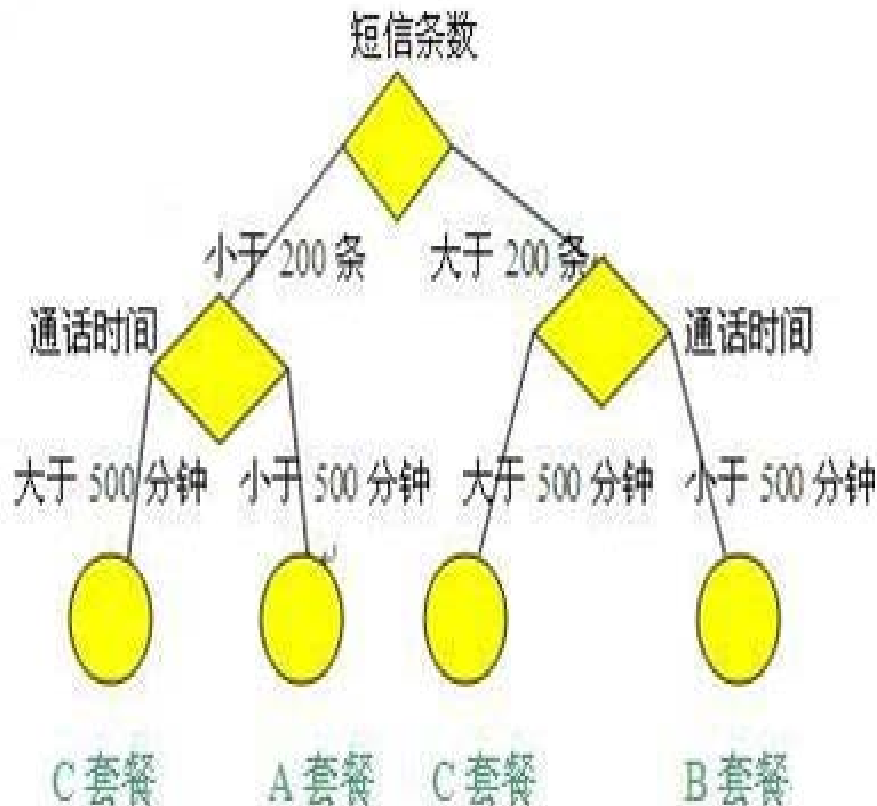
$(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal})$

$\vee (\text{Outlook} = \text{Overcast})$

$\vee (\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})$

引例

- 引例3:通信套餐选择/推荐
- 引例4: Akinator名人猜



Open a **FREE** Practice Account

Learn more

Online

Available on the App Store

Akinator on your iPhone and iPod Touch

Hello, I am Akinator. To start a game, click on **play**

The last ten games

You thought of	I guessed
The Scouts Mother (Te: You !!!)	The Scouts Mother (Te: You !!!)
Johnny Depp	Johnny Depp
Tasaki Takahiro (Exile)	Tasaki Takahiro (Exile)
Joseph Biden	Joseph Biden
Lt. Aldo Raine (Inglouri)	Lt. Aldo Raine (Inglouri)
Jun Manjome	Jun Manjome
Uni (Katekyo Hitman Reborn)	Uni (Katekyo Hitman Reborn)
Tamahome (Fushigi Yui)	Tamahome (Fushigi Yui)
Gisele Bündchen	Gisele Bündchen

858 people are currently playing (Max=1500)

39234215 games played 23156 today

引 例

● 适用问题的特征

- 实例由“属性-值”对表示
- 属性可以是连续值或离散值
- 具有离散的输出值
- 训练数据可以包含缺少属性值的实例（后面实例）

● 问题举例

- 根据症状或检查结果分类患者
- 根据起因或现象分类设备故障
- 根据拖欠支付的可能性分类贷款申请

● 分类问题

- 核心任务是把样例分类到各可能的离散值对应的类别

决策树算法

● 决策树 (Decision Tree)

■ 决策树是一种**树型结构**，由结点和有向边组成

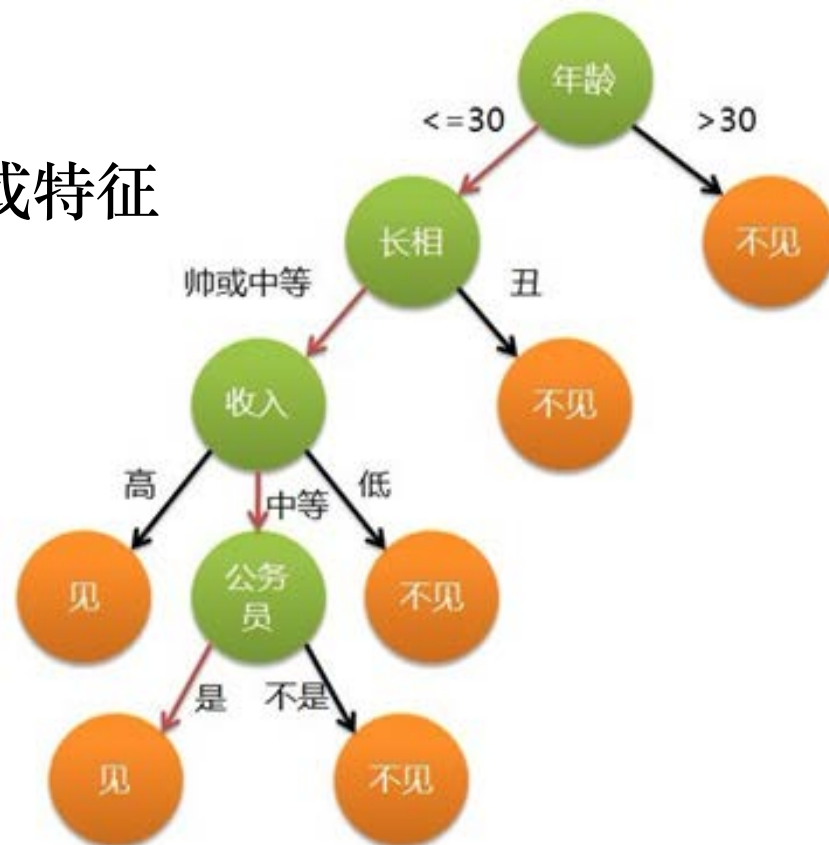
■ 节点

● **内部结点**表示一个属性或特征

● **叶结点**代表一种**类别**

■ 有向边/分支

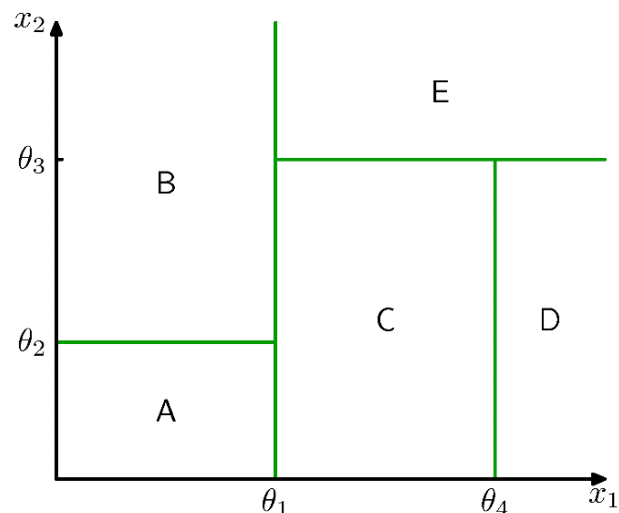
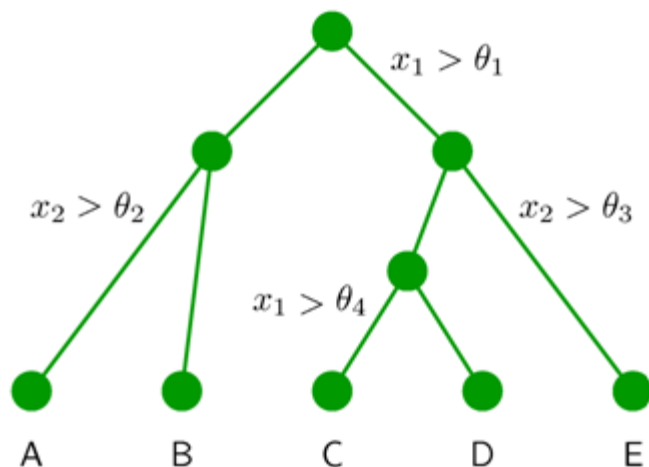
■ **分支**代表一个测试**输出**



决策树算法

● 决策树

- **基本思想**：采用自顶向下的**递归方法**，以信息熵为度量构造一棵熵值下降最快的树，到叶子节点处的熵值为零，此时每个叶节点中的实例都属于同一类
- 决策树可以看成是一个**if-then的规则集合**
- 一个决策树将特征空间划分为不相交的单元（cell）或区域(region)



决策树算法

- 决策树分类：分为两步

- 第1步：利用训练集建立(并精化)一棵决策树，建立决策树模型。这个过程实际上是一个从数据中获取知识，进行机器学习的过程
- 第2步：利用生成的决策树对输入数据进行分类。对输入的记录，从根结点依次测试记录的属性值，直到到达某个叶结点，从而找到该记录所在的类

决策树生成/学习算法

- step 1: 选取一个属性作为决策树的根结点，然后就这个属性所有的取值创建树的分支
- step 2: 用这棵树来对训练数据集进行分类：
 - 如果一个叶结点的所有实例都属于同一类，则以该类为标记标识此叶结点
 - 如果所有的叶结点都有类标记，则算法终止
- step 3: 否则，选取一个从该结点到根路径中没有出现过的属性为标记标识该结点，然后就这个属性所有的取值继续创建树的分支；重复算法步骤step 2

主要算法

- 决建立决策树的关键，即在当前状态下**选择哪个属性作为分类依据**

示例-高考报志愿：好学校？好专业？好位置？ ...

找对象：年龄？学历？兴趣爱好？身高？ ...

- **目标：**每个分支节点的样本尽可能属于同一类别，即节点的**“纯度” (purity)**越来越高；最具区分性的属性！
- 根据不同目标函数，建立决策树主要有以下**三种算法**
 - ID3： 信息增益
 - C4.5： 信息增益率
 - CART： 基尼指数

ID3算法

●ID3 (Iterative Dichotomiser 3)算法

- ID3算法是一种最经典的决策树学习算法，由Quinlan于1986年提出
- 基本思想：以信息熵为度量，用于决策树节点的属性选择，每次优先选取**信息增益最大的属性**，亦即使熵值变为最小的属性，以构造一颗**熵值下降最快**的决策树，到叶子节点处的熵值为0。此时，每个叶子节点对应的实例集中的实例属于同一类

信息论基础

- 信息量：具有确定概率事件的信息的定量度量

- 定义：
$$I(x) = -\log_2 p(x)$$

其中 $p(x)$ 为事件 x 发生的概率

- 信息熵：事件集合的信息量的平均值

- 定义：
$$\begin{aligned} H(x) &= \sum_i h(x_i) = \sum_i p(x_i) I(x_i) \\ &= -\sum_i p(x_i) \log_2 p(x_i) \end{aligned}$$

- 举例：明天下雪？明天雾霾？

ID3算法

● 信息熵 (entropy)

- 信息论与概率统计中，熵表示随机变量不确定性的
大小，是度量样本集合纯度最常用的一种指标
- 令离散随机变量 X 概率分布为 $p(X = x_i) = p_i$ ，则随
机变量 X 的熵定义为：

$$H(X) = -\sum_{i=1}^n p_i \log_2 p_i$$

信息按二进制位编
码，因此以2为底

- 若 X 为连续随机变量，则概率分布变成概率密度函数，
求和符号变成积分符号即可

ID3算法

● 信息熵 (entropy)

- 信息论与概率统计中，熵表示随机变量不确定性的
大小，是度量样本集合纯度最常用的一种指标
- 令离散随机变量 X 概率分布为 $p(X = x_i) = p_i$ ，则随
机变量 X 的熵定义为：

$$H(X) = -\sum_{i=1}^n p_i \log_2 p_i$$

- 若 X 为连续随机变量，则概率分布变成概率密度函数，
求和符号变成积分符号即可

ID3算法

● 信息熵

- 熵定义了一个函数(概率密度函数pdf)到一个值(信息熵)的映射
 $P(x) \rightarrow H$ (函数 \rightarrow 数值)

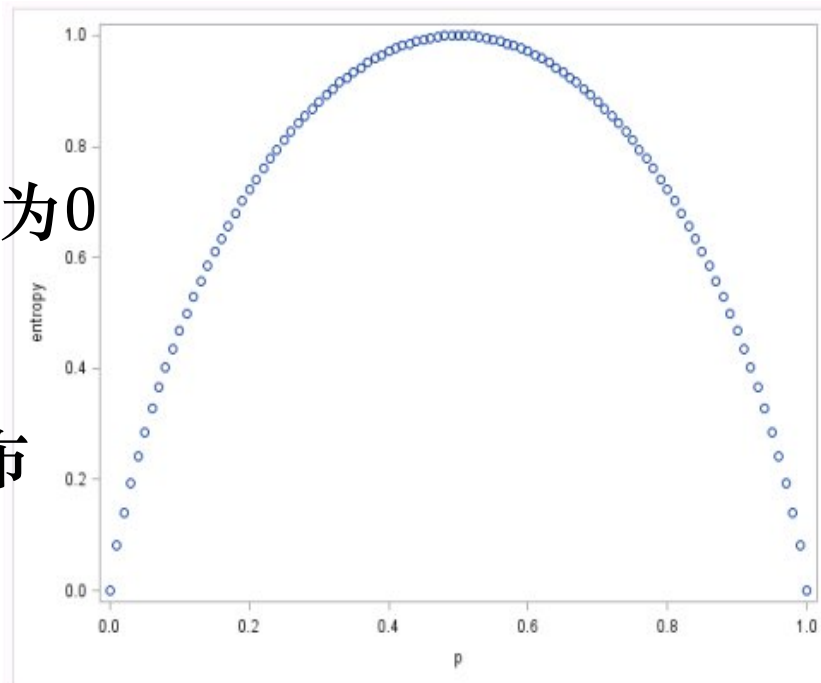
- 熵是随机变量**不确定性**的度量:

- 不确定性越大, 熵值越大

- 若随机变量退化成定值, 熵为0

示例: 明天下雪? 明天晴天?

- 均匀分布是“最不确定”的分布



ID3算法

- 经验(信息)熵

- 假设当前样本集合 D 中第 c ($c=1,2,...,C$) 类样本所占比例为 p_c ($c=1,2,...,C$) , 则 D 的经验信息熵(简称经验熵)定义为

$$\begin{aligned} H(D) &= -\sum_{c=1}^C p_c \log_2 p_c \\ &= -\sum_{c=1}^C \frac{D_c}{D} \log_2 \frac{D_c}{D} \end{aligned}$$

- 可以看出: $H(D)$ 的值越小, 则 D 的纯度越高

ID3算法

● 条件熵 (conditional entropy)

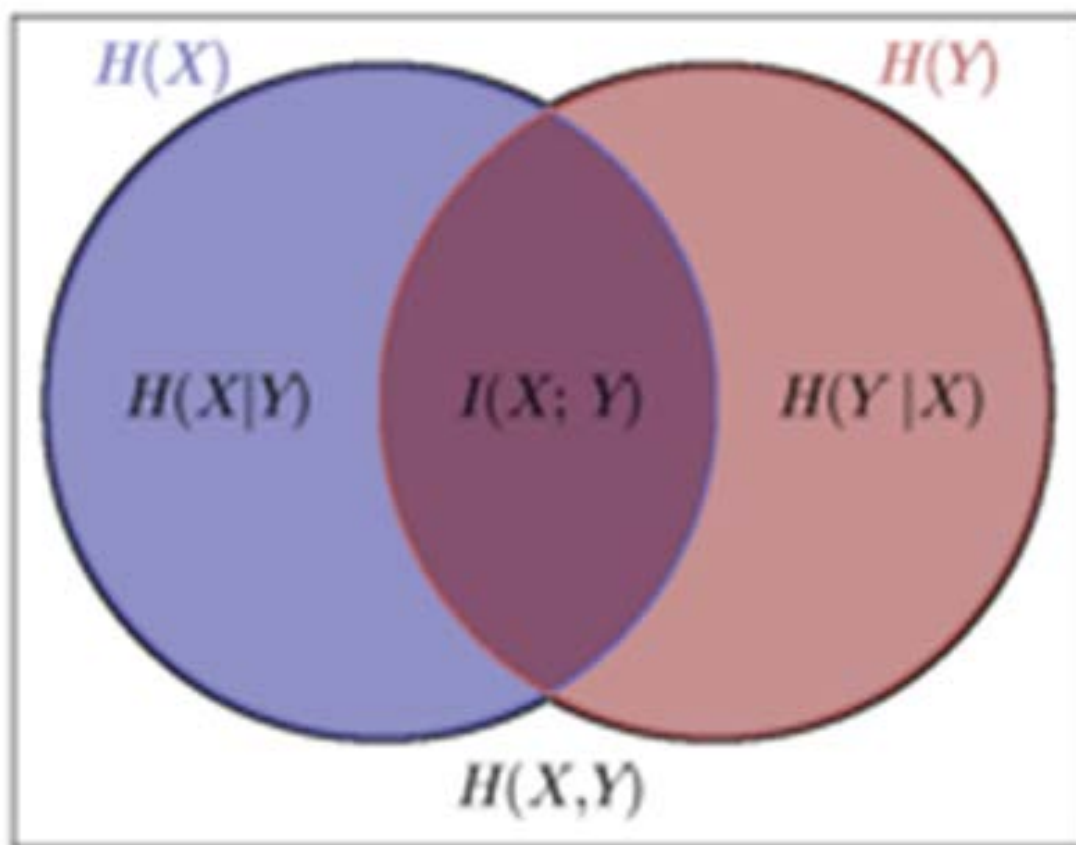
- 对随机变量 (X, Y) ，联合分布为： $p(X = x_i, Y = y_i) = p_{ij}$
条件熵 $H(Y|X)$ 表示在已知随机变量 X 的条件下，随机变量 Y 的不确定性，定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望：

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

- (X, Y) 发生所包含的信息熵，减去 Y 单独发生所包含的信息熵——在 Y 发生的前提下， X 发生“新”带来的信息熵

ID3算法

- 条件熵-示意图



Venn图

ID3算法

● 条件熵-推导

$$\begin{aligned} H(Y | X) &= H(X, Y) - H(Y) \\ &= -\sum_{x,y} p(x, y) \log_2 p(x, y) + \sum_y p(y) \log_2 p(y) \\ &= -\sum_{x,y} p(x, y) \log_2 p(x, y) + \sum_y \left(\sum_x p(x, y) \right) \log_2 p(y) \\ &= -\sum_{x,y} p(x, y) \log_2 p(x, y) + \sum_{x,y} p(x, y) \log_2 p(y) \\ &= -\sum_{x,y} p(x, y) \log_2 \frac{p(x, y)}{p(y)} \\ &= -\sum_{x,y} p(x, y) \log_2 p(x | y) \end{aligned}$$

ID3算法

- 经验条件熵

- 假设当前样本集合 D 共有 C 类，每一类有 D_c 个样本，属性 a ($a \in A$)有不同的取值 $\{a_1, a_2, \dots, a_N\}$ ，每一类中属性为 i 的样本数为 D_c^n ，则 D 的经验条件熵定义为

$$H(D|a) = - \sum_{n,c} p(D_c, a_n) \log_2 p(D_c | a_n)$$

$$= - \sum_{n=1}^N \frac{|D^n|}{|D|} \sum_{c=1}^C \frac{|D_c^n|}{|D^n|} \log_2 \frac{|D_c^n|}{|D^n|}$$

$$= \sum_{n=1}^N \frac{|D^n|}{|D|} H(D^n)$$

- 即特征 a 的信息对样本 D 的信息的不确定性减少的程度

ID3算法

- 信息增益 (information gain) :
 - 特征 a 对训练数据集 D 的信息增益 $G(D, a)$ ，定义为集合 D 的经验熵 $H(D)$ 与特征 a 给定条件下 D 的经验条件熵 $H(D | a)$ 之差，即

$$\begin{aligned} G(D, a) &= H(D) - H(D | a) \\ &= H(D) - \sum_{n=1}^N \frac{|D^n|}{|D|} H(D^n) \end{aligned}$$

- ID3算法即是以此信息增益为准则，对每次递归的节点属性进行选择的

ID3算法

● 决策树的生成算法：

输入：训练数据集 D ，特征集 A ，阈值 ε

输出：决策树 T

- (1) 若 D 中所有实例属于同一类 C_k ，则 T 为单结点树，并将类 C_k 作为该结点的类标记，返回 T ；
- (2) 若 $A=\emptyset$ ，则 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点类标记，返回 T ；
- (3) 否则，计算 A 中各特征对 D 的信息增益，选择信息增益最大的特征 A_g
- (4) 如果 A_g 的信息增益小于阈值 ε ，则置 T 为单结点树，并将 D 中样本数最大的类 C_k 作为该结点的类标记，返回 T
- (5) 否则，对 A_g 的每一个可能值 a_i ，依 $A_g=a_i$ ，将 D 分割为若干非空子集 D_i ，将 D_i 中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树 T ，返回 T
- (6) 对第 i 个子结点，以 D_i 为训练集，以 $A-\{A_g\}$ 为特征集，递归的调用第 (1) ~ (5) 步，得到子树 T_i ，返回 T_i 。

ID3算法-示例

编号	色泽	根蒂	敲声	纹理	脐部	触感	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	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	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	浊响	稍糊	凹陷	硬滑	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	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	稍糊	稍凹	硬滑	否

ID3算法-示例

● 计算信息熵-以属性色泽为例

$$H(D) = -\sum_{c=1}^C p_c \log_2 p_c = -\left(\frac{8}{17} \log_2 \frac{8}{17} + \frac{9}{17} \log_2 \frac{9}{17}\right) = 0.998$$

$$H(D^{\text{青绿}}) = -\left(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}\right) = 1.000$$

$$H(D^{\text{乌黑}}) = -\left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6}\right) = 0.918$$

$$H(D^{\text{浅白}}) = -\left(\frac{1}{5} \log_2 \frac{1}{5} + \frac{4}{5} \log_2 \frac{4}{5}\right) = 0.722$$

■ 参照教材《机器学习》-周志华P75-77示例

ID3算法-示例

● 计算信息增益-以属性色泽为例

$$\begin{aligned} G(D, \text{色泽}) &= H(D) - \sum_{n=1}^3 \frac{|D^n|}{|D|} H(D^n) \\ &= 0.998 - \left(\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.722 \right) \\ &= 0.109 \end{aligned}$$

■ $G(D, \text{色泽}) = 0.109$

■ $G(D, \text{敲声}) = 0.141$

■ $G(D, \text{脐部}) = 0.289$

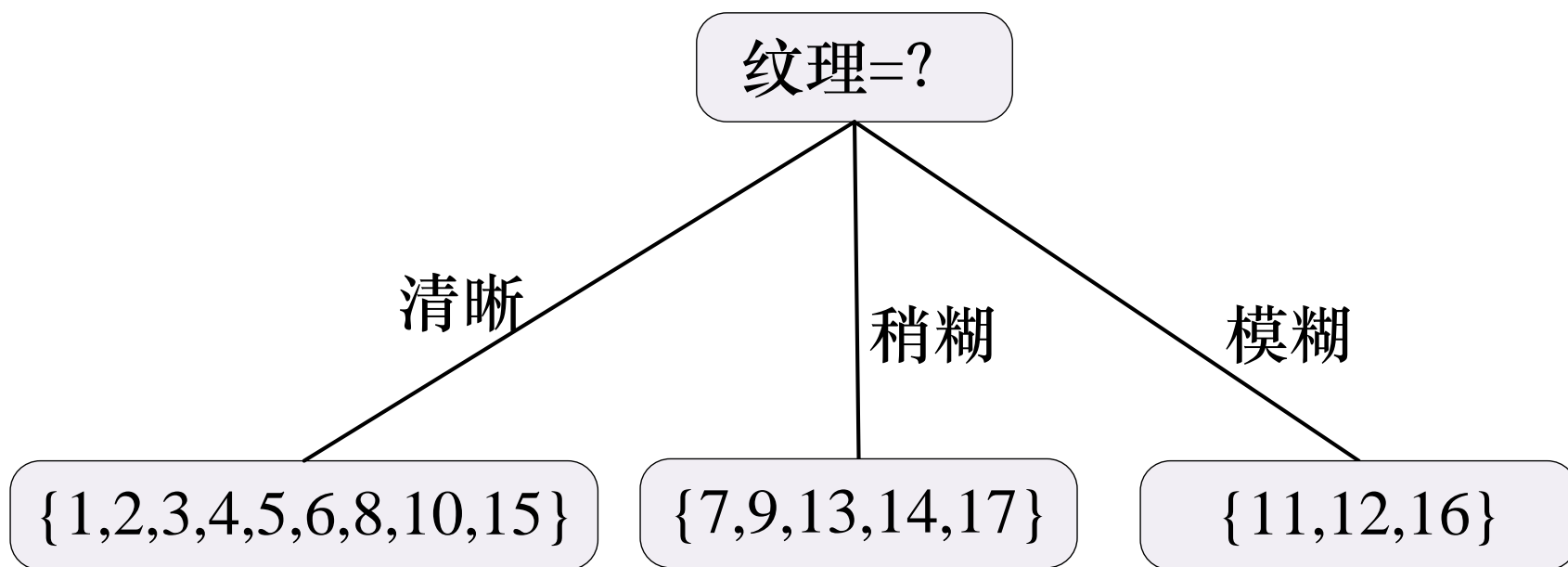
$G(D, \text{根蒂}) = 0.143$

$G(D, \text{纹理}) = 0.381$

$G(D, \text{触感}) = 0.006$

ID3算法-示例

- 基于属性“纹理”对根节点进行划分



ID3算法-示例

- 继续进行划分-以“纹理=清晰”分支为例

- “纹理=清晰”分支：

样本 {1,2,3,4,5,6,8,10,15}

- 计算信息增益

- $G(D_{\text{清晰}}, \text{色泽}) = 0.043$

- $G(D_{\text{清晰}}, \text{敲声}) = 0.331$

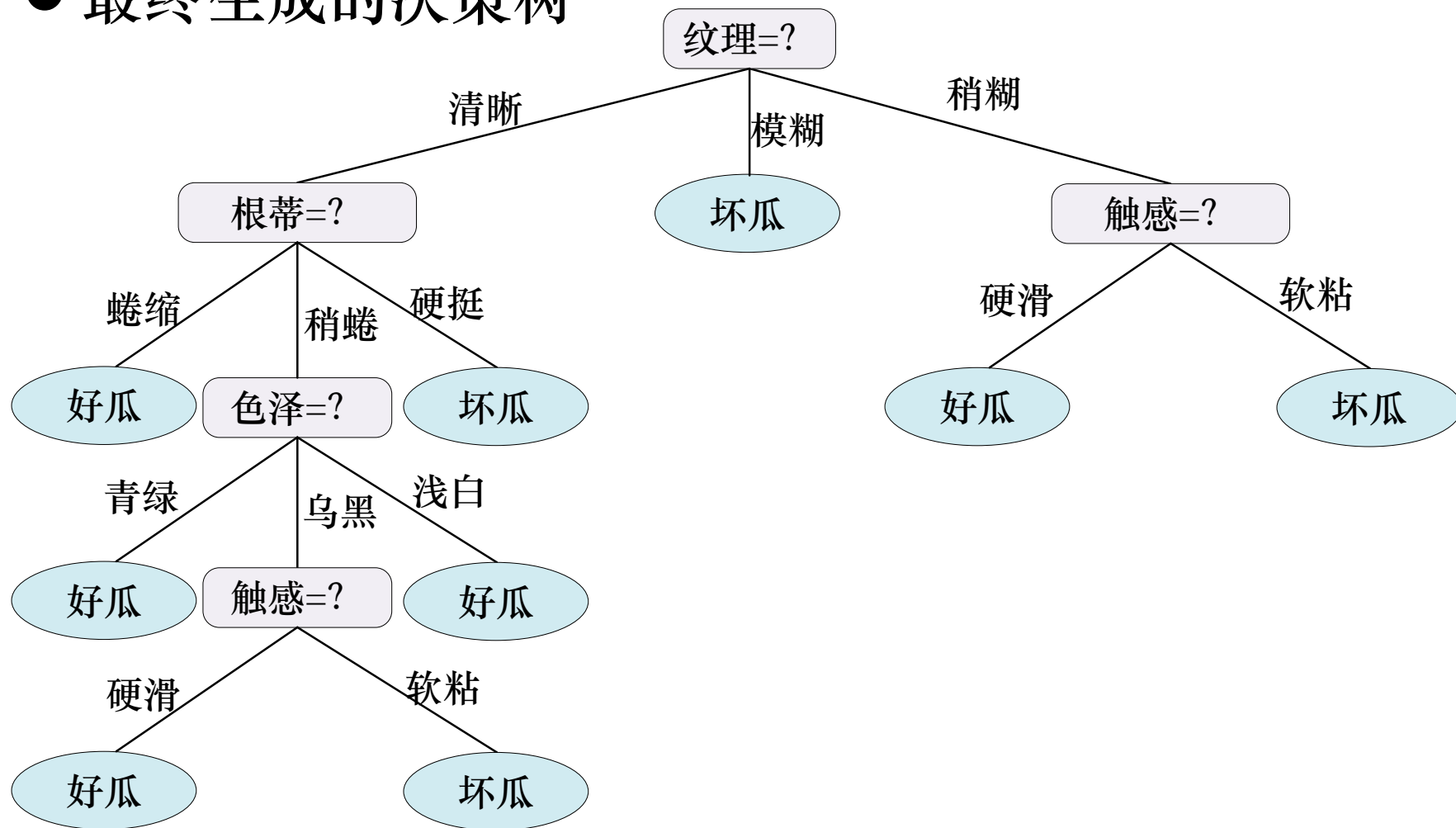
- $G(D_{\text{清晰}}, \text{触感}) = 0.458$

$$G(D_{\text{清晰}}, \text{根蒂}) = 0.458$$

$$G(D_{\text{清晰}}, \text{脐部}) = 0.458$$

ID3算法-示例

● 最终生成的决策树



算法特点

- 最大优点是，它可以自学习：在学习的过程中，不需要使用者了解过多背景知识，只需要对训练实例进行较好的标注，就能够进行学习
- 决策树的分类模型是树状结构，简单直观，比较符合人类的理解方式
- 可将决策树中到达每个叶节点的路径转换为IF—THEN形式的分类规则，这种形式更有利于理解
- 从一类无序、无规则的事物(概念)中推理出决策树表示的分类规则
- 显然，属于有监督学习

ID3算法的问题

- 信息增益偏好取值多的属性(分散, 极限趋近于均匀分布)
 - 属性筛选度量标准
- 可能会受噪声或小样本影响, 易出现过拟合问题
 - 剪枝处理
- 无法处理连续值的属性
 - 连续值处理
- 无法处理属性值不完整的训练数据
 - 缺失值处理
- 无法处理不同代价的属性
 - 不同代价属性的处理
- 针对这些问题的改进, ID3被扩展成C4.5(同时处理前4个问题)等算法

属性筛选度量标准

- 信息增益的问题：

$$G(D, a) = H(D) - \sum_{n=1}^N \frac{|D^n|}{|D|} H(D^n)$$

- 信息增益准则对可取值数目 N 较多的属性有所偏好
- 以序号为划分属性，信息增益最大！
- 取值更多的属性更容易使得数据更“纯”，其信息增益更大，决策树会首先挑选这个属性作为树的顶/节点；结果训练出来的形状是一棵庞大且深度很浅的树，这样的划分是极为不合理的

属性筛选度量标准

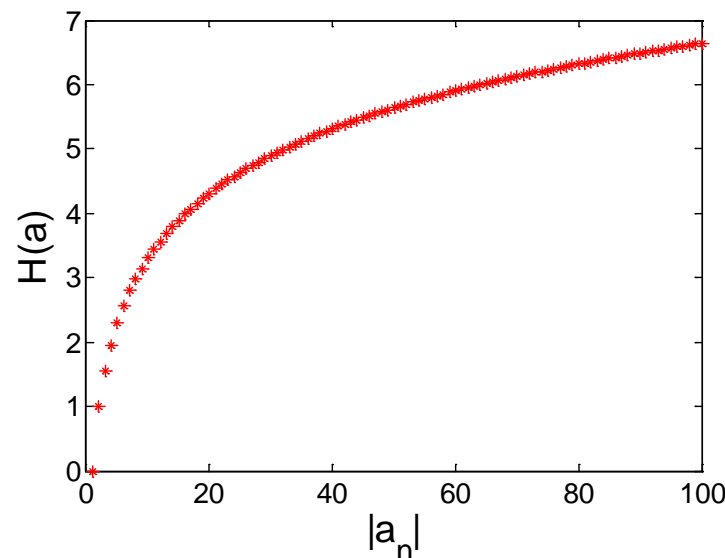
- 增益率 (gain ratio) :

$$G_{ratio}(D, a) = \frac{G(D, a)}{H(a)}$$

其中

$$H(a) = -\sum_{n=1}^N \frac{|D_n|}{|D|} \log_2 \frac{|D_n|}{|D|}$$

称为属性 a 的固有值



- N 越大, $H(a)$ 通常也越大; 因此, 采用增益率, 可缓解信息增益准则对可取值数目较多的属性的偏好
- C4.5算法[Quinlan 1993]就基于信息增益 (初选) 和增益率 (精选) 替代了ID3 算法的信息增益

属性筛选度量标准

- 基尼指数 (Gini index) :

$$Gini(D) = \sum_{c=1}^C \sum_{c' \neq c} p_c p_{c'} = 1 - \sum_{c=1}^C p_c^2 = 1 - \sum_{c=1}^C \left(\frac{|D_c|}{|D|} \right)^2$$

- 直观反映了从数据集中随机抽取两个样本，其类别不一致的概率；因此，Gini(D)越小，则数据集D的纯度越高

- 属性A的基尼指数:

$$Gini(D, a) = \sum_{n=1}^N \frac{|D^n|}{|D|} Gini(D^n)$$

- 最优属性选择:

$$a^* = \arg \min_{a \in A} Gini(D, a)$$

- CART算法(Breiman 1984)就采用基尼指数替代了ID3 算法的信息增益

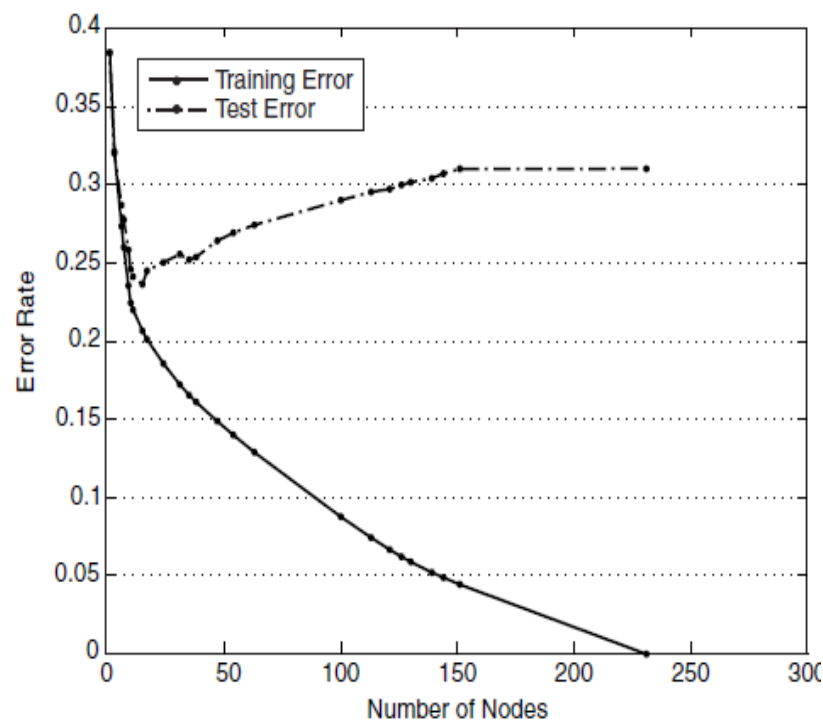
剪枝处理 (Pruning)

● 问题-过拟合

- 决策树对训练数据有很好的分类能力，但对未知的测试数据未必有好的分类能力，泛化能力弱，即可能发生过拟合现象

● 可能原因

- 训练数据有噪声，对训练数据拟合的同时也对噪音进行拟合，影响了分类效果
- 叶节点样本太少，易出现耦合的规律性，使一些属性恰巧可以很好地分类，但却与实际的目标函数并无关系



剪枝处理 (Pruning)

- 解决办法

- 剪枝是决策树学习算法中对付“过拟合”的主要手段

- 基本策略

- 预剪枝策略 (pre-pruning) : 决策树生成过程中, 对每个节点在划分前进行估计, 若划分不能带来决策树泛化性能提升, 则停止划分, 并将该节点设为叶节点

- 后剪枝策略 (post-pruning) : 先利用训练集生成决策树, 自底向上对非叶节点进行考察, 若将该叶节点对应子树替换为叶节点能带来泛化性能提升, 则将该子树替换为叶节点

剪枝处理 (Pruning)

训练样本

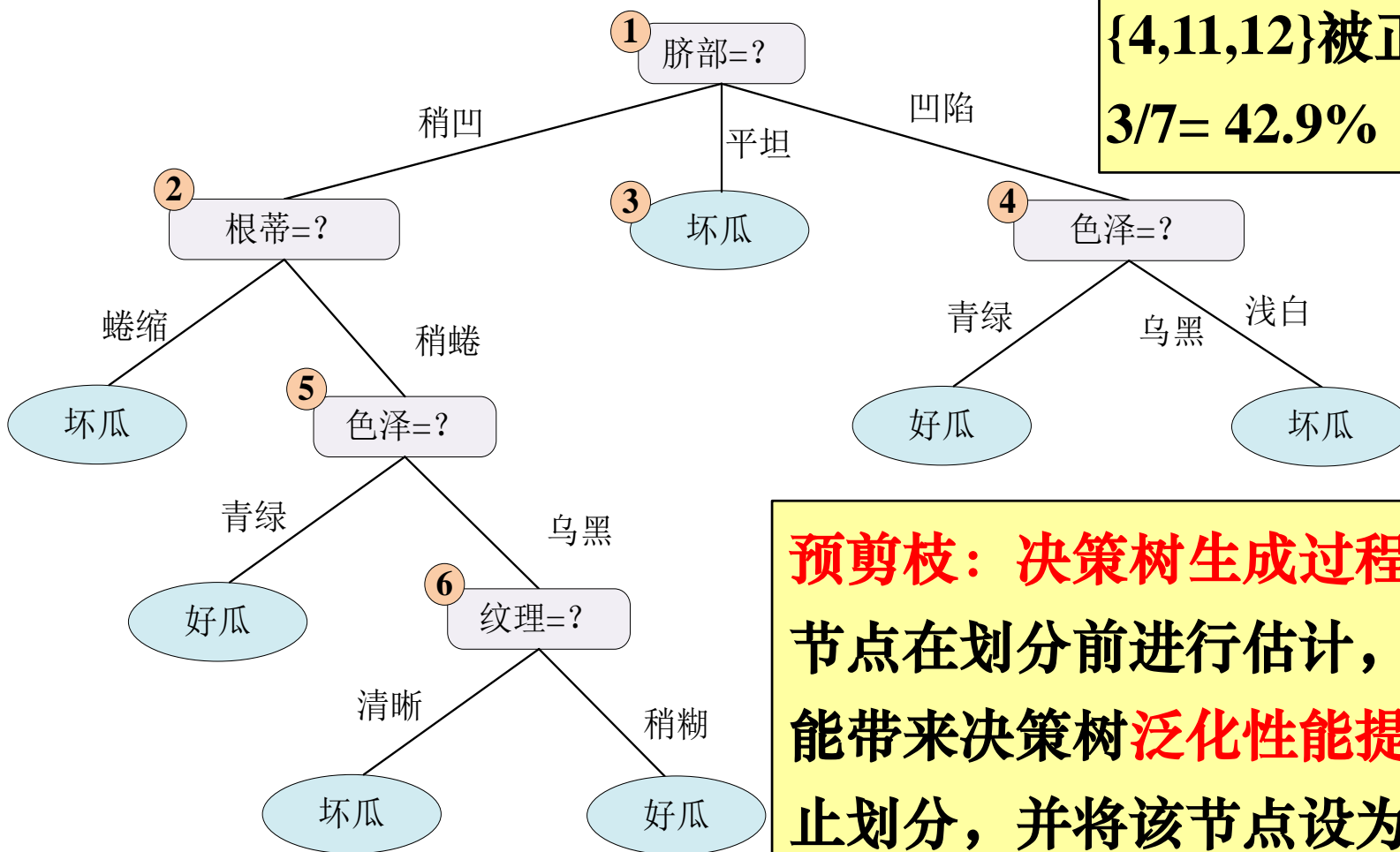
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

测试样本

编号	色泽	根蒂	敲声	纹理	脐部	触感	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	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	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	浊响	稍糊	凹陷	硬滑	否

预剪枝算法

● ID3算法生成的决策树



● 泛化性能:
{4,11,12}被正确划分
 $3/7 = 42.9\%$

预剪枝: 决策树生成过程中, 对各节点在划分前进行估计, 若划分不能带来决策树泛化性能提升, 则停止划分, 并将该节点设为叶节点

预剪枝算法

● 第一步：评估节点1

■ 属性选择：基于信息增益准则，选择属性“脐部”

■ 不划分：

● 标记为训练样例数最多的类别，如“好瓜”

● 泛化性能：{4,5,8}被正确分类， $3/7 = 42.9\%$

■ 划 分：

● 节点2：稍凹{6,7,15,17} “好瓜”

● 节点3：平坦{10,16} “坏瓜”

● 节点4：凹陷{1,2,3,14} “好瓜”

● 泛化性能：{4,5,8,11,12}被正确分类， $5/7 = 71.4\%$

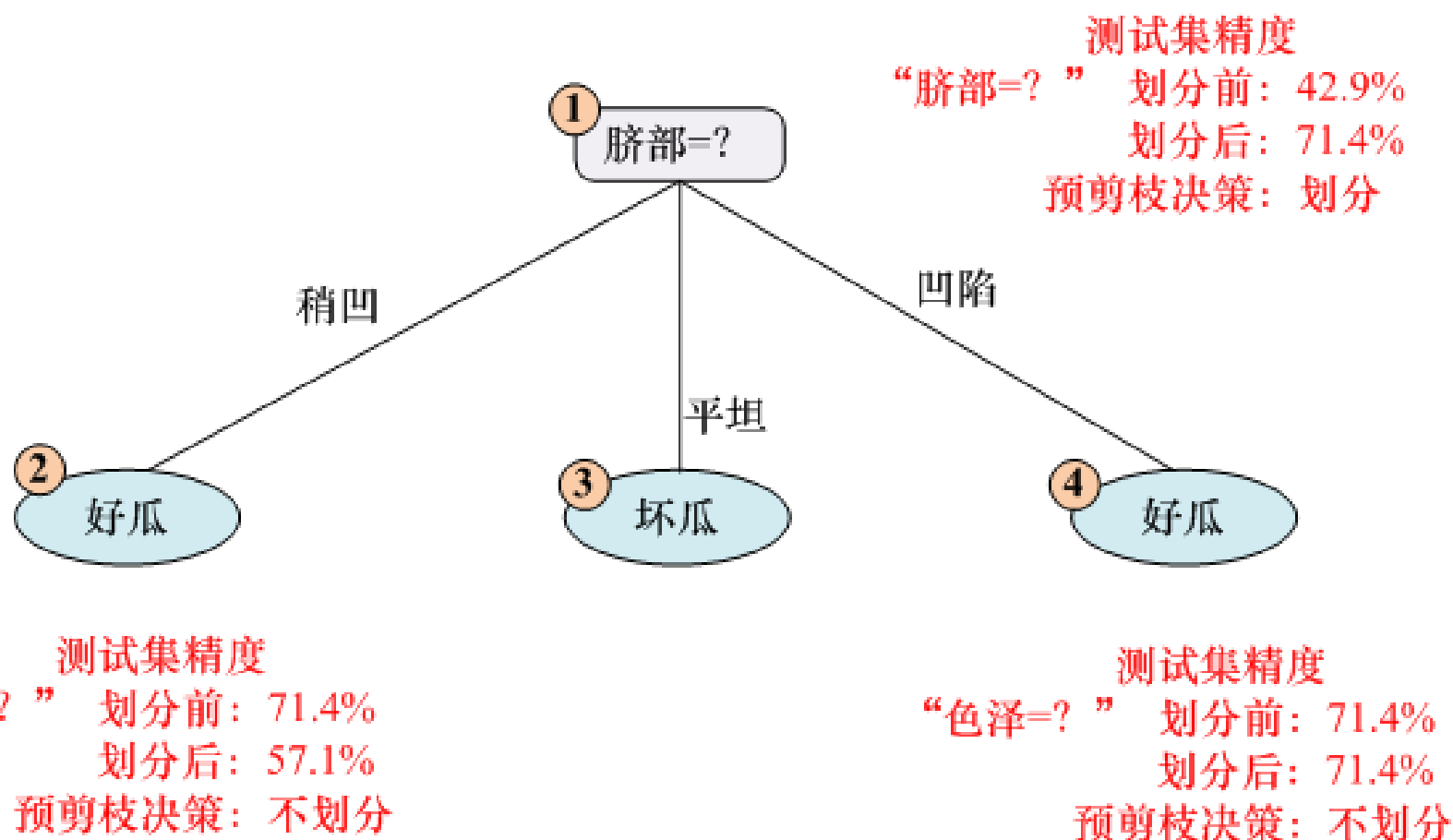
■ 评估结果/预剪枝决策： 划分

预剪枝算法

- 第二步：评估节点2：训练样本{6,7,15,17}
 - 属性选择：基于信息增益准则，选择属性“根蒂”
 - 不划分：{4,5,8,11,12}被正确分类 $5/7 = 71.4\%$
 - 划 分：{4,5,8,11,12}被正确分类 $5/7 = 71.4\%$
 - 评估结果/预剪枝决策：不划分
- 第三步：评估节点4：训练样本{1, 2, 3, 14}
 - 属性选择：基于信息增益准则，选择属性“色泽”
 - 不划分：{4,5,8,11,12}被正确分类 $5/7 = 71.4\%$
 - 划 分：{4,~~5~~,8,11,12}被正确分类 $4/7 = 57.1\%$
 - 评估结果/预剪枝决策：不划分

预剪枝算法

● 最终生成的决策树



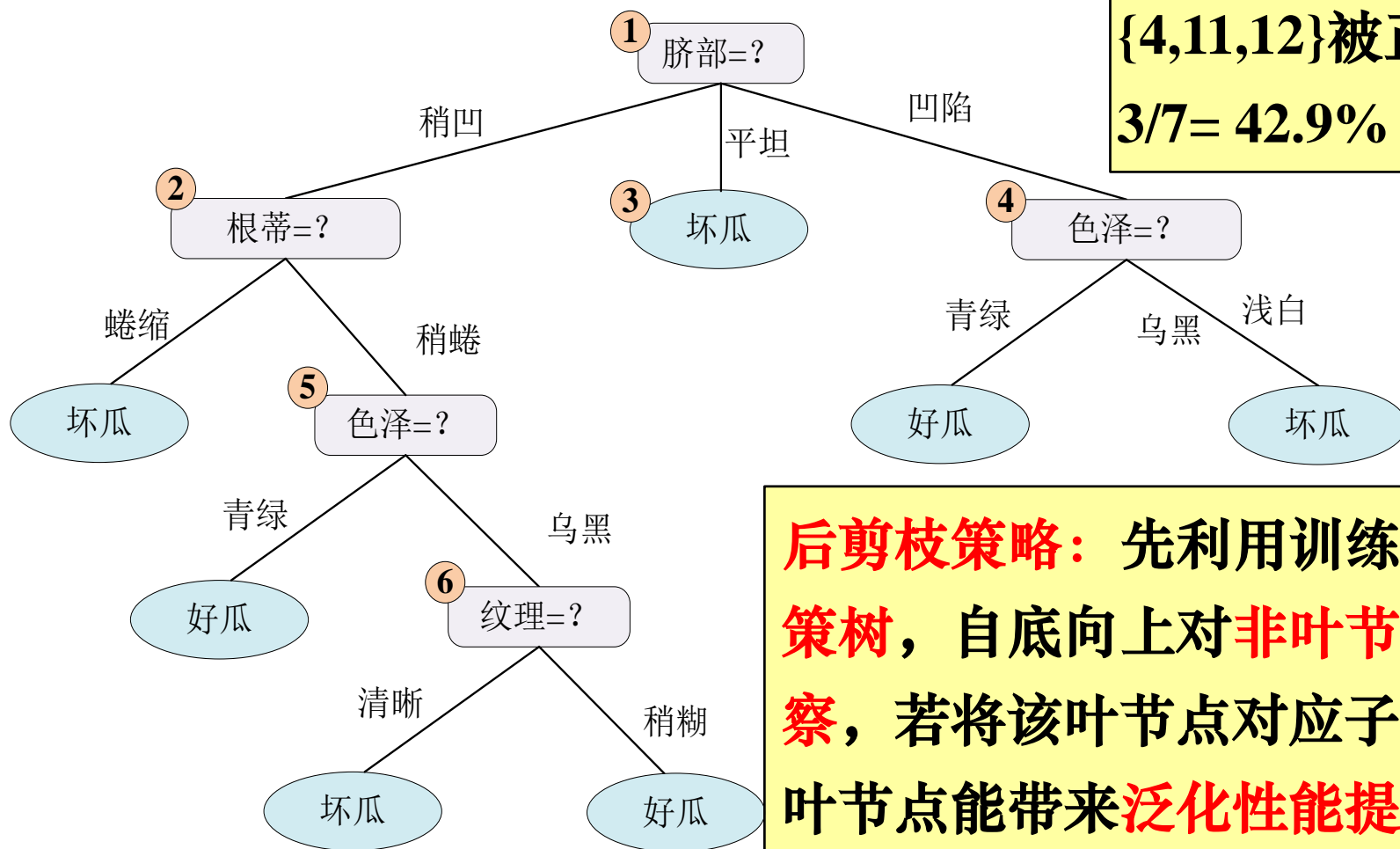
预剪枝算法

● 结论

- **优势：**预剪枝“剪掉了”很多没必要展开的分支，降低了过拟合的风险，并且显著减少了决策树的训练时间开销和测试时间开销
- **劣势：**有些分支的当前划分有可能不能提高甚至降低泛化性能，但后续划分有可能提高泛化性能；预剪枝禁止这些后续分支的展开，可能会导致欠拟合

后剪枝算法

● ID3算法生成的决策树



● 泛化性能:

{4,11,12}被正确划分

$$3/7 = 42.9\%$$

后剪枝策略: 先利用训练集生成决策树, 自底向上对非叶节点进行考察, 若将该叶节点对应子树替换为叶节点能带来泛化性能提升, 替换

后剪枝算法

● 第一步：评估节点6

■ 剪枝前：

- 属性为“纹理”；样本为{7,15}

- 泛化性能：{4,11,12}被正确分类， $3/7 = 42.9\%$

■ 剪枝后：

- 把节点6替换为叶节点， “好/坏瓜”

- 泛化性能 {4, $8/9$, 11, 12}被正确分类， $4/7 = 57.1\%$

■ 评估结果/后剪枝决策： 剪枝

后剪枝算法

● 第二步：评估节点5

■ 剪枝前：

- 属性为“色泽”，样本{6,7,15}

- 泛化性能：同第一步

4/7 = 57.1%

■ 剪枝后：

- 把节点5替换为叶节点，

“好瓜”

- 泛化性能：{4, 8, 11, 12}被正确分类，

4/7 = 57.1%

■ 评估结果/后剪枝决策： 不剪枝

后剪枝算法

● 第三步：评估节点4

■ 剪枝前：

- 属性为“色泽”，样本{1,2,3,14}

- 泛化性能：同上一步

4/7 = 57.1%

■ 剪枝后：

- 把节点4替换为叶节点，

“好瓜”

- 泛化性能：{4, 5, 8, 11, 12}被正确分类，

5/7 = 71.4%

■ 评估结果/后剪枝决策：剪枝

后剪枝算法

● 第四步：评估节点2

■ 剪枝前：

- 属性为“根蒂”，样本{6,7,15,17}

- 泛化性能：同上一步

5/7 = 71.4%

■ 剪枝后：

- 把节点2替换为叶节点，

“好/坏瓜”

- 泛化性能：{4,5,**8/9**,11,12}被正确分类，**5/7 = 71.4%**

■ 评估结果/后剪枝决策：**不剪枝**

后剪枝算法

- 第五步：评估节点1

- 剪枝前：

- 泛化性能：同上一步

$5/7 = 71.4\%$

- 剪枝后：

- 把节点1替换为叶节点

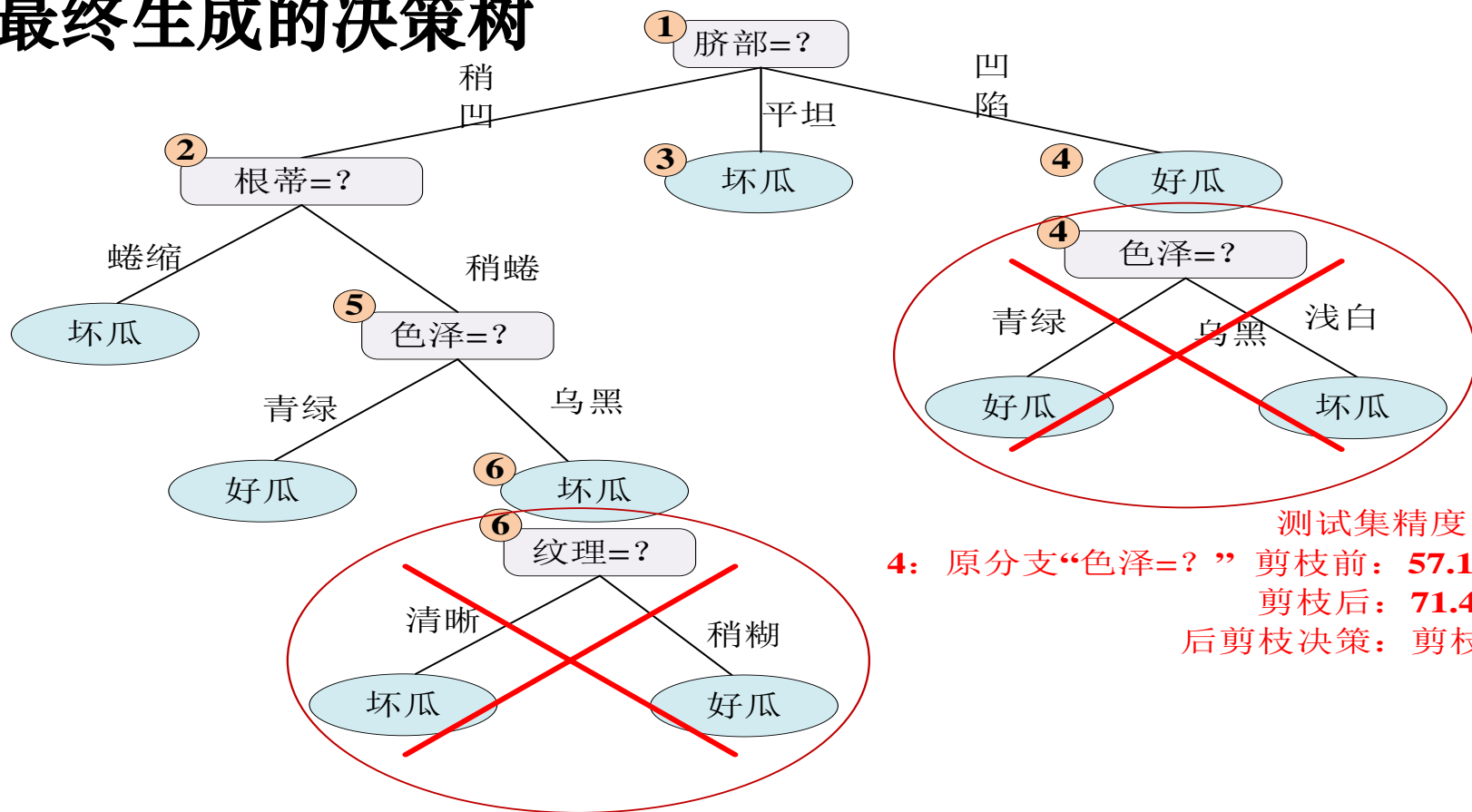
- 泛化性能：{4,5,8,11,12}被正确分类，

$5/7 = 71.4\%$

- 评估结果/后剪枝决策： **不剪枝**

后剪枝算法

● 最终生成的决策树



测试集精度

6: 原分支“纹理=? ” 剪枝前: 42.9%

剪枝后: 57.1%

后剪枝决策: 剪枝

测试集精度

4: 原分支“色泽=? ” 剪枝前: 57.1%

剪枝后: 71.4%

后剪枝决策: 剪枝

后剪枝算法

● 结论

- **优势：**测试了所有分支，比预剪枝决策树保留了更多分支，降低了欠拟合的风险，泛化性能一般优于预剪枝决策树
- **劣势：**后剪枝过程在生成完全决策树后进行，且要自底向上对所有非叶节点逐一评估；因此，决策树的训练时间开销要高于未剪枝决策树和预剪枝决策树

连续值处理

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.46	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.36	0.37	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

连续值处理

- **基本思想：** 采用二分法(bi-partition)进行离散化
 - 给定样本集 D 和连续属性 $a(a \in A)$ ，假定 a 在 D 上有 N 个不同取值，将这些值从大到小排序得 $\{a_1, a_2, \dots, a_N\}$
 - 基于划分点 t ，可将 D 分为子集 D_t^+ 和 D_t^- ，其中 D_t^+ (D_t^-) 包含了属性值 A 不小（大）于 t 的样本子集
 - t 在 $[a_n, a_{n+1})$ 上的任意取值的划分结果都相同
 - 候选划分点集合

$$T_a = \left\{ \frac{a_n + a_{n+1}}{2} \mid 1 \leq n \leq N - 1 \right\}$$

- C4.5算法[Quinlan 1993]就采用了二分法进行离散化

连续值处理

● 信息增益

$$\begin{aligned} G(D, a) &= \max_{t \in T_a} G(D, a, t) \\ &= \max_{t \in T_a} \left(H(D) - \sum_{\lambda \in \{+, -\}} \frac{|D_t^\lambda|}{|D|} H(D_t^\lambda) \right) \end{aligned}$$

其中， $G(D, A, t)$ 是样本集D基于划分点 t 二分后的信息增益，所以，我们需选择使 $G(D, A, t)$ 最大的划分点 t

连续值处理-示例

编号	色泽	根蒂	敲声	纹理	脐部	触感	密度	含糖率	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	0.697	0.46	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	0.774	0.376	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	0.634	0.264	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	0.608	0.318	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	0.556	0.215	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	0.403	0.237	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	0.481	0.149	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	0.437	0.211	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	0.666	0.091	否
10	青绿	硬挺	清脆	清晰	平坦	软粘	0.243	0.267	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	0.245	0.057	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	0.343	0.099	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	0.639	0.161	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	0.657	0.198	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	0.36	0.37	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	0.593	0.042	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	0.719	0.103	否

连续值处理-示例

- 计算候选划分点集合

$$T_a = \left\{ \frac{a_n + a_{n+1}}{2} \mid 1 \leq n \leq N-1 \right\}$$

- $T_{\text{密度}} = \{0.244, 0.294, 0.351, \dots, 0.708, 0.746\}$ **1*16**

- $T_{\text{含糖率}} = \{0.049, 0.074, 0.095, \dots, 0.373, 0.126\}$ **1*16**

■ 参照教材《机器学习》-周志华P84-85示例

连续值处理-示例

● 计算信息增益

$$G(D, a) = H(D) - \sum_{n=1}^N \frac{|D^n|}{|D|} H(D^n) \rightarrow G(D, a) = \max_{t \in T_a} \left(H(D) - \sum_{\lambda \in \{+, -\}} \frac{|D_t^\lambda|}{|D|} H(D_t^\lambda) \right)$$

■ $G(D, \text{密度}) = 0.262$

$T_{\text{密度}}^* = 0.381$

■ $G(D, \text{含糖率}) = 0.349$

$T_{\text{含糖率}}^* = 0.126$

■ 已知

■ $G(D, \text{色泽}) = 0.109$

$G(D, \text{根蒂}) = 0.143$

■ $G(D, \text{敲声}) = 0.141$

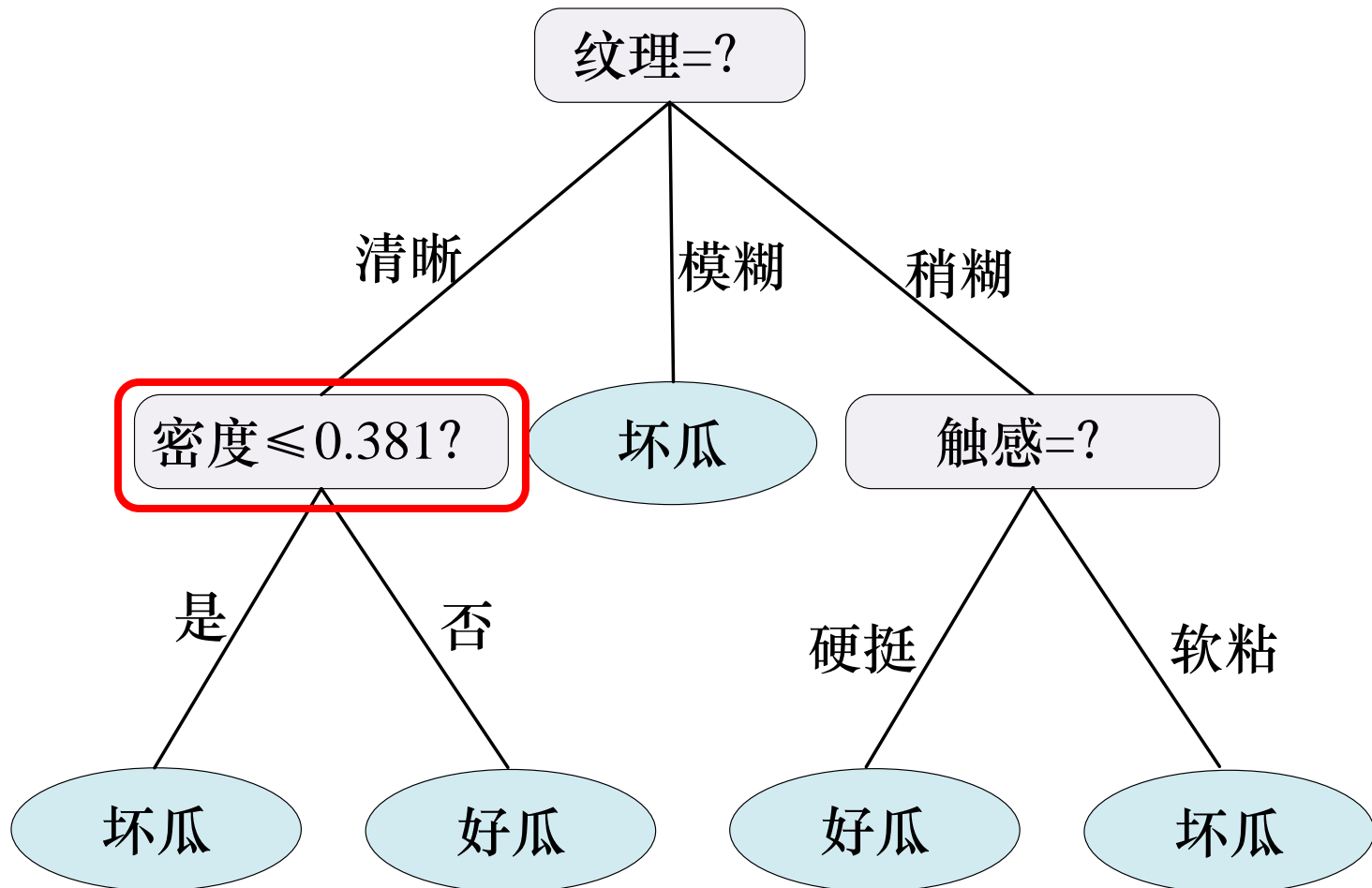
$G(D, \text{纹理}) = 0.381$

■ $G(D, \text{脐部}) = 0.289$

$G(D, \text{触感}) = 0.006$

连续值处理-示例

- 最终生成的决策树



缺失值处理

● 问题

- 前面假设：所有样本的属性完整
- 实际情况：存在不完整样本：即样本的某些属性缺失；特别是属性数目较多时
- 如果简单放弃不完整样本，会导致数据信息的浪费
- 实际中确实需要属性缺失情况下进行决策
- 例如：医疗领域，由于诊测成本、隐私保护等问题，只有部分诊断结果

缺失值处理

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

缺失值处理

- 需要解决的两个问题

- 如何在属性值缺失的情况下进行划分属性选择（计算信息增益）？
- 给定划分属性，若样本在该属性上的值缺失，如何对样本进行划分？

缺失值处理

- 问题1：信息增益计算

- 仅可利用没有属性缺失的样本

- 定义：

- \tilde{D} : D 中在属性 a 上没有缺失值的样本子集
- 属性 $a(a \in A)$ 有 N 个可取值 $\{a_1, a_2, \dots, a_N\}$
- \tilde{D}^n : \tilde{D} 中在属性 a 上取值为 a_n 的样本子集
- \tilde{D}_c : \tilde{D} 中属于第 $c(c \in C)$ 类的样本子集
- ω_x : 样本 x 的权重

缺失值处理

● 定义：

■ 无缺失值样本所占比例

$$\rho = \frac{\sum_{x \in \tilde{D}} \omega_x}{\sum_{x \in D} \omega_x}$$

■ 无缺失样本中第c类比例

$$\tilde{p}_c = \frac{\sum_{x \in \tilde{D}_c} \omega_x}{\sum_{x \in \tilde{D}} \omega_x}, (1 \leq c \leq C), \sum_{c=1}^C \tilde{p}_c = 1$$

■ 无缺失样本中属性a上取值为 a_n 的比例

$$\tilde{r}_n = \frac{\sum_{x \in \tilde{D}^n} \omega_x}{\sum_{x \in \tilde{D}} \omega_x}, (1 \leq n \leq N), \sum_{n=1}^N \tilde{r}_n = 1$$

缺失值处理

- 信息增益

$$G(D, a) = H(D) - \sum_{n=1}^N \frac{|D^n|}{|D|} H(D^n)$$



$$G(D, a) = \rho \times G(\tilde{D}, a) = \rho \times \left(H(\tilde{D}) - \sum_{n=1}^N \tilde{r}_n H(\tilde{D}_n) \right)$$

其中

$$H(\tilde{D}) = - \sum_{c=1}^C \tilde{p}_c \log_2 \tilde{p}_c$$

缺失值处理

- 问题2：含有缺失属性的样本的划分
- 原则：让样本根据属性情况以不同概率划分到不同子节点
 - 若样本 x 在属性 a 上的取值已知：则将 x 划入与其取值对应的子节点，且样本权值保持为 ω_x
 - 若样本 x 在属性 a 上的取值未知：则将 x 划入所有子节点，且其在与属性值对应的子节点中的权值根据属性 a 上已知的样本的比例调整为 $\tilde{r}_n \cdot \omega_x$
- C4.5算法[Quinlan 1993]就采用了上述方法处理缺失值

缺失值处理-示例

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

缺失值处理-示例

● 计算信息熵-以属性色泽为例

$$H(\tilde{D}) = -\sum_{c=1}^C \tilde{p}_c \log_2 \tilde{p}_c = -\left(\frac{6}{14} \log_2 \frac{6}{14} + \frac{8}{14} \log_2 \frac{8}{14}\right) = 0.985$$

$$H(\tilde{D}^{\text{青绿}}) = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1.000$$

$$H(\tilde{D}^{\text{乌黑}}) = -\left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6}\right) = 0.918$$

$$H(\tilde{D}^{\text{浅白}}) = -\left(\frac{0}{4} \log_2 \frac{0}{4} + \frac{4}{4} \log_2 \frac{4}{4}\right) = 0.000$$

■ 参照教材《机器学习》-周志华P86-88示例

缺失值处理-示例

● 计算信息增益-以属性色泽为例

$$\begin{aligned} G(\tilde{D}, \text{色泽}) &= H(\tilde{D}) - \sum_{n=1}^3 \tilde{r}_n H(\tilde{D}_n) \\ &= 0.985 - \left(\frac{4}{14} \times 1.000 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0.000 \right) \\ &= 0.306 \end{aligned}$$

$$G(D, \text{色泽}) = \rho \times G(\tilde{D}, \text{色泽}) = \frac{14}{17} \times 0.306 = 0.252$$

■ $G(D, \text{色泽}) = 0.252$

■ $G(D, \text{敲声}) = 0.145$

■ $G(D, \text{脐部}) = 0.289$

$G(D, \text{根蒂}) = 0.171$

$G(D, \text{纹理}) = 0.424$

$G(D, \text{触感}) = 0.006$

缺失值处理-示例

- 因此选择属性“纹理”用于对根结点划分

- 属性不缺失样本

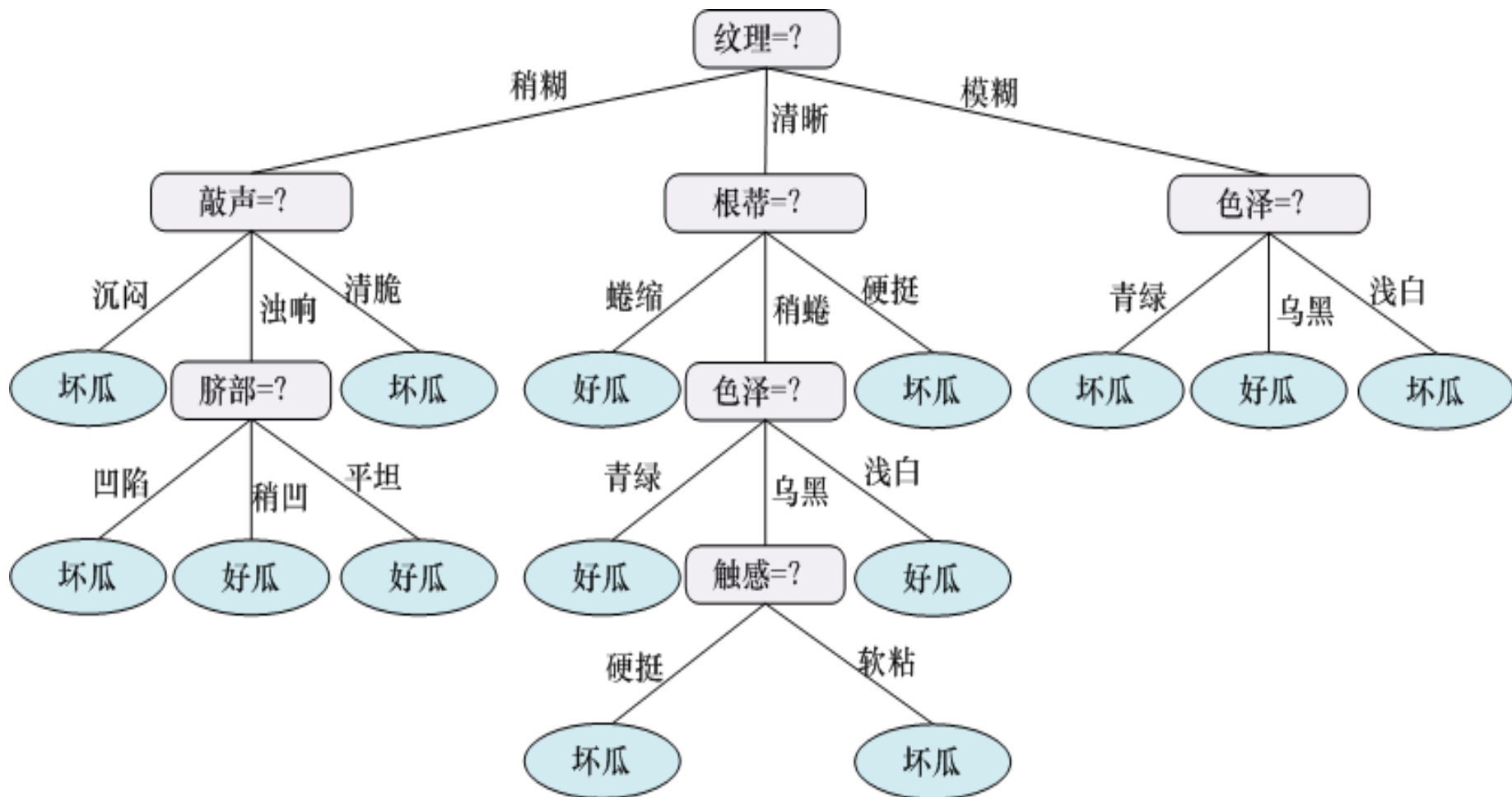
- “纹理=清晰”分支：样本 {1, 2, 3, 4, 5, 6, 15}
- “纹理=稍糊”分支：样本 {7, 9, 13, 14, 17}
- “纹理=模糊”分支：样本 {11, 12, 16}
- 且样本在各节点中的权重 ω 保持为1

- 属性缺失样本

- 样本{8,10}:同时进入三个分支，权重 ω 分别为 $\frac{7}{15}, \frac{5}{15}, \frac{3}{15}$

缺失值处理-示例

● 最终生成的决策树



不同代价属性的处理

- 问题

- 不同的属性测量具有不同的代价

- 医疗诊断为例：

- 属性：体温、血压、血常规、活检

- 代价不同：所需时间、费用、友好性等

- 解决思路

- 在属性筛选度量标准中考虑属性的不同代价

- 优先选择低代价属性的决策树

- 必要时才依赖高代价属性

不同代价属性的处理

- 属性筛选度量标准1 [Tan et al., 1990, 1993]

$$G_{Cost}(D, a) = \frac{G(D, a)}{Cost(a)}$$

- 属性筛选度量标准2 [Nunez et al., 1988, 1991]

$$G_{Cost}(D, a) = \frac{2^{G(D, a)} - 1}{(Cost(a) + 1)^\omega}$$

其中 $Cost(a)$ 为属性 a 的代价

$\omega \in [0, 1]$ 为一常数，决定代价对于信息增益的相对重要性

延伸阅读

- **概念学习系统(Concept Learning System, CLS)**
 - 1966年由Hunt等人提出，奠定决策树算法发展的基础
- **CART(Classification And Regression Tree)算法：**
 - 即分类回归树算法，简称CART算法，是一种二分递归分割技术，1984年由Breiman等人提出
- **J. R. Quinlan**
 - 1986年：ID3算法；1993年：C4.5算法
- **多变量决策树**
 - OC1[Murthy et al., 1994]等

ID3算法-示例2

- 谁在买计算机?
- 问题：假定公司收集了左表数据，那么对于任意给定的客人(测试样例)，你能帮助公司将这位客人归类吗?
- 你能预测这位客人是属于“买”计算机的那一类，还是属于“不买”计算机的那一类?

计数	年龄	收入	学生	信誉	买计算机?
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

ID3算法-示例2

计数	年龄	收入	学生	信誉	买计算机?
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第1步：计算数据集的经验熵

决策属性“买计算机？”

该属性分两类：买/不买

$$|C_1|(\text{买})=641$$

$$|C_2|(\text{不买})=383$$

$$|D|=|C_1|+|C_2|=1024$$

$$P_1=641/1024=0.6260$$

$$P_2=383/1024=0.3740$$

$$\begin{aligned}
 H(D) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0.9537
 \end{aligned}$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	买计算机?
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2步:计算特征的信息增益

条件属性共有4个
分别是**年龄、收入、学生、信誉**

分别计算不同属性的信息增益

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-1步:计算年龄的经验条件熵

年龄共分三个组：

青年、中年、老年

青年买与不买比例为128/256

$$|D_{11}(\text{买})|=128$$

$$|D_{12}(\text{不买})|=256$$

$$|D_1|=|D_{11}|+|D_{12}|=384$$

$$P_{11}=128/384$$

$$P_{12}=256/384$$

$$H(\text{青年}|\text{年龄})$$

$$=-P_{11}\log_2 P_{11}-P_{12}\log_2 P_{12}$$

$$=-(P_{11}\log_2 P_{11}+P_{12}\log_2 P_{12})$$

$$=0.9183$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-2步:计算年龄的经验条件熵

年龄共分三个组：

青年、中年、老年

中年买与不买比例为256/0

$$|D_{21}(\text{买})| = 256$$

$$|D_{22}(\text{不买})| = 0$$

$$|D_2| = |D_{21}| + |D_{22}| = 256$$

$$P_{21} = 256/256$$

$$P_{22} = 0/256$$

$$H(\text{中年}|\text{年龄})$$

$$= -P_{21} \log_2 P_{21} - P_{22} \log_2 P_{22}$$

$$= -(P_{21} \log_2 P_{21} + P_{22} \log_2 P_{22})$$

$$= 0$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-3步:计算年龄的经验条件熵

年龄共分三个组：

青年、中年、老年

老年买与不买比例为257/127

$$|D_{31}(\text{买})| = 257$$

$$|D_{32}(\text{不买})| = 127$$

$$|D_3| = |D_{31}| + |D_{32}| = 384$$

$$P_{31} = 257/384$$

$$P_{32} = 127/384$$

$$H(\text{老年}|\text{年龄})$$

$$= -P_{31} \log_2 P_{31} - P_{32} \log_2 P_{32}$$

$$= -(P_{31} \log_2 P_{31} + P_{32} \log_2 P_{32})$$

$$= 0.9157$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第2-4步:计算年龄的经验条件熵

年龄共分三个组：
青年、中年、老年
所占比例

青年组 $384/1025=0.375$

中年组 $256/1024=0.25$

老年组 $384/1024=0.375$

计算年龄的平均信息期望

$$\begin{aligned} H(\text{年龄}) &= 0.375 * 0.9183 + \\ &\quad 0.25 * 0 + \\ &\quad 0.375 * 0.9157 \\ &= 0.6877 \end{aligned}$$

G(年龄信息增益)

$$\begin{aligned} &= 0.9537 - 0.6877 \\ &= 0.2660 \quad (1) \end{aligned}$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第3步:计算收入的熵

收入共分三个组:

高、中、低

$H(\text{收入})=0.9361$

$G(\text{收入信息增益})=$

$0.9537 - 0.9361 = 0.0176$ (2)

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第4步:计算学生的熵

学生共分二个组：

学生、非学生

$H(\text{学生})=0.7811$

$G(\text{学生信息增益})=$

$0.9537-0.7811=0.1726$ (3)

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第5步:计算信誉的熵

信誉分二个组:

良好、优秀

$$H(\text{信誉}) = 0.9048$$

$$G(\text{信誉信息增益}) = 0.9537 - 0.9048 = 0.0453 \quad (4)$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

第6步:计算选择节点

$$\text{年龄信息增益} = 0.9537 - 0.6877 = 0.2660 \quad (1)$$

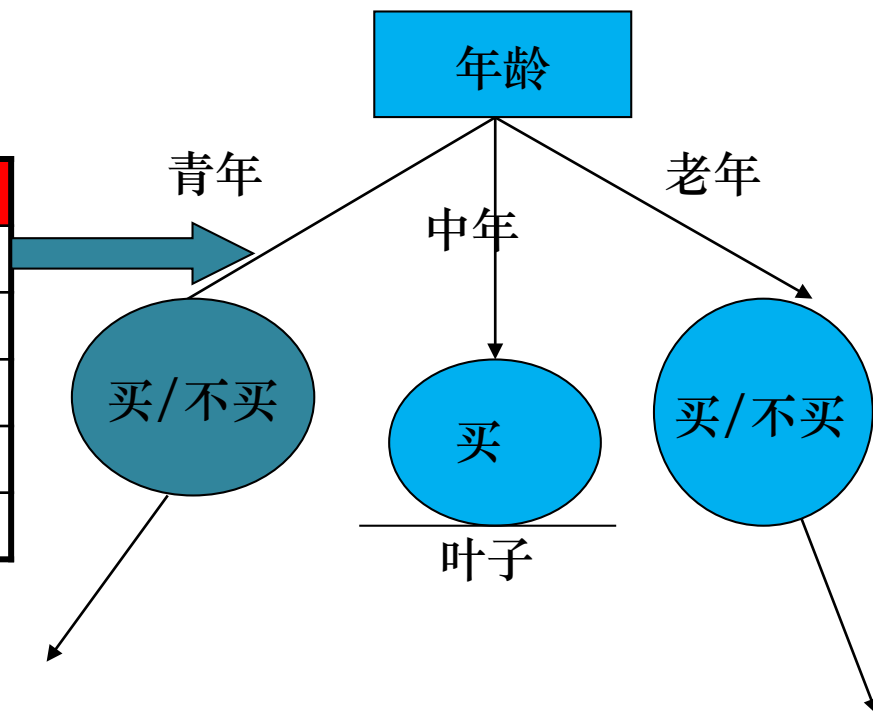
$$\text{收入信息增益} = 0.9537 - 0.9361 = 0.0176 \quad (2)$$

$$\text{学生信息增益} = 0.9537 - 0.7811 = 0.1726 \quad (3)$$

$$\text{信誉信息增益} = 0.9537 - 0.9048 = 0.0453 \quad (4)$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类: 买计算机?
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买



ID3算法-示例2

青年买与不买比例为128/256

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买

$$|D_{11}(\text{买})|=128$$

$$|D_{12}(\text{不买})|=256$$

$$|D_1|=|D_{11}|+|D_{12}|=384$$

$$P_{11}=128/384$$

$$P_{12}=256/384$$

$$H(\text{青年}|\text{年龄})$$

$$=-P_{11}\log_2 P_{11}-P_{12}\log_2 P_{12}$$

$$=-(P_{11}\log_2 P_{11}+P_{12}\log_2 P_{12})$$

$$=0.9183$$

ID3算法-示例2

如果选择收入作为节点，分高、中、低

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	青	中	否	良	不买
64	青	低	是	良	买
64	青	中	是	优	买

$$H(\text{高})=0$$

$$\text{比例: } 28/384=0.3333$$

$$H(\text{中})=0.9183$$

$$\text{比例: } 192/384=0.5$$

$$H(\text{低})=0$$

$$\text{比例: } 64/384=0.1667$$

经验条件熵(加权总和):

$$H(\text{收入}) = 0.3333 * 0 + 0.5 * 0.9183 + 0.1667 * 0 = 0.4592$$

收入的信息增益

$$\text{Gain}(\text{收入}) = H(\text{青年}) - H(\text{收入}) = 0.9183 - 0.4592 = 0.4591$$

ID3算法-示例2

计数	年龄	收入	学生	信誉	归类：买计算机？
64	青	高	否	良	不买
64	青	高	否	优	不买
128	中	高	否	良	买
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
64	中	低	是	优	买
128	青	中	否	良	不买
64	青	低	是	良	买
132	老	中	是	良	买
64	青	中	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

