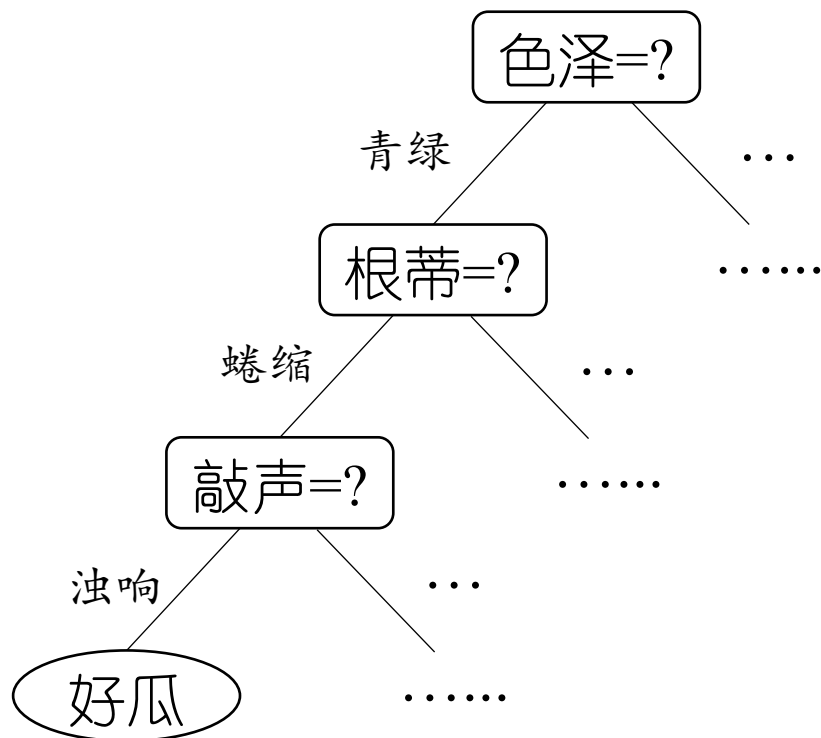

第四章：决策树

大纲

- 基本流程
- 划分选择
- 剪枝处理
- 连续与缺失值
- 多变量决策树

基本流程

决策树基于树结构来进行预测



基本流程

- 决策过程中提出的每个判定问题都是对某个属性的“测试”
- 决策过程的最终结论对应了我们所希望的判定结果
- 每个测试的结果或是导出最终结论，或者导出进一步的判定问题，其考虑范围是在上次决策结果的限定范围之内
- 从根结点到每个叶结点的路径对应了一个判定测试序列

决策树学习的目的是为了产生一棵泛化能力强，
即处理未见示例能力强的决策树

基本流程

Algorithm 1 决策树学习基本算法

输入:

- 训练集 $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$;
- 属性集 $A = \{a_1, \dots, a_d\}$.

过程: 函数 $\text{TreeGenerate}(D, A)$

```
1: 生成结点 node;  
2: if  $D$  中样本全属于同一类别  $C$  then  
3:   将 node 标记为  $C$  类叶结点; return  
4: end if  
5: if  $A = \emptyset$  OR  $D$  中样本在  $A$  上取值相同 then  
6:   将 node 标记为叶结点, 其类别标记为  $D$  中样本数最多的类; return  
7: end if  
8: 从  $A$  中选择最优划分属性  $a_*$ ;  
9: for  $a_*$  的每一个值  $a_*^v$  do  
10:   为 node 生成每一个分枝; 令  $D_v$  表示  $D$  中在  $a_*$  上取值为  $a_*^v$  的样本子集;  
11:   if  $D_v$  为空 then  
12:     将分枝结点标记为叶结点, 其类别标记为  $D$  中样本最多的类; return  
13:   else  
14:     以  $\text{TreeGenerate}(D_v, A - \{a_*\})$  为分枝结点  
15:   end if  
16: end for
```

输出: 以 node 为根结点的一棵决策树

(1) 当前结点包含的样本全部属于同一类别

(2) 当前属性集为空, 或所有样本在所有属性上取值相同

(3) 当前结点包含的样本集合为空

大纲

- 基本流程
- 划分选择
- 剪枝处理
- 连续与缺失值
- 多变量决策树

划分选择

- 决策树学习的关键在于如何选择最优划分属性。一般而言，随着划分过程不断进行，我们希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”(purity)越来越高
- 经典的属性划分方法：
 - 信息增益
 - 增益率
 - 基尼指数

划分选择-信息增益

- “信息熵”是度量样本集合纯度最常用的一种指标，假定当前样本集合 D 中第 k 类样本所占的比例为 $p_k (K = 1, 2, \dots, |\mathcal{Y}|)$ ，则 D 的信息熵定义为

$$\text{Ent}(D) = - \sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k$$

$\text{Ent}(D)$ 的值越小，则 D 的纯度越高

- 计算信息熵时约定：若 $p = 0$ ，则 $p \log_2 p = 0$
- $\text{Ent}(D)$ 的最小值为0，最大值为 $\log_2 |\mathcal{Y}|$

划分选择-信息增益

- 离散属性 a 有 V 个可能的取值 $\{a^1, a^2, \dots, a^V\}$ ，用 a 来进行划分，则会产生 V 个分支结点，其中第 v 个分支结点包含了 D 中所有在属性 a 上取值为 a^v 的样本，记为 D^v 。则可计算出用属性 a 对样本集 D 进行划分所获得的“信息增益”：

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v)$$

为分支结点权重，样本数越多的分支结点的影响越大

- 一般而言，信息增益越大，则意味着使用属性 a 来进行划分所获得的“纯度提升”越大
- ID3决策树学习算法[Quinlan, 1986]以信息增益为准则来选择划分属性

划分选择-信息增益

信息增益实例

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

该数据集包含17个训练样本, $|\mathcal{Y}| = 2$, 其中正例占 $p_1 = \frac{8}{17}$, 反例占 $p_2 = \frac{9}{17}$, 计算得到根结点的信息熵为

$$\text{Ent}(D) = - \sum_{k=1}^2 p_k \log_2 p_k = -(\frac{8}{17} \log_2 \frac{8}{17} + \frac{9}{17} \log_2 \frac{9}{17}) = 0.998$$

划分选择-信息增益

- 以属性“色泽”为例，其对应的3个数据子集分别为 D^1 (色泽=青绿), D^2 (色泽=乌黑), D^3 (色泽=浅白)
- 子集 D^1 包含编号为{1, 4, 6, 10, 13, 17}的6个样例，其中正例占 $p_1 = \frac{3}{6}$ ，反例占 $p_2 = \frac{3}{6}$

划分选择-信息增益

信息增益实例

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

划分选择-信息增益

- 以属性“色泽”为例，其对应的3个数据子集分别为 D^1 (色泽=青绿), D^2 (色泽=乌黑), D^3 (色泽=浅白)
- 子集 D^1 包含编号为{1, 4, 6, 10, 13, 17}的6个样例，其中正例占 $p_1 = \frac{3}{6}$ ，反例占 $p_2 = \frac{3}{6}$ ， D^2 、 D^3 同理，3个结点的信息熵为：

$$\text{Ent}(D^1) = -(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}) = 1.000$$

$$\text{Ent}(D^2) = -(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6}) = 0.918$$

$$\text{Ent}(D^3) = -(\frac{1}{5} \log_2 \frac{1}{5} + \frac{4}{5} \log_2 \frac{4}{5}) = 0.722$$

- 属性“色泽”的信息增益为

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

划分选择-信息增益

□ 类似的，其他属性的信息增益为

$$\text{Gain}(D, \text{根蒂}) = 0.143$$

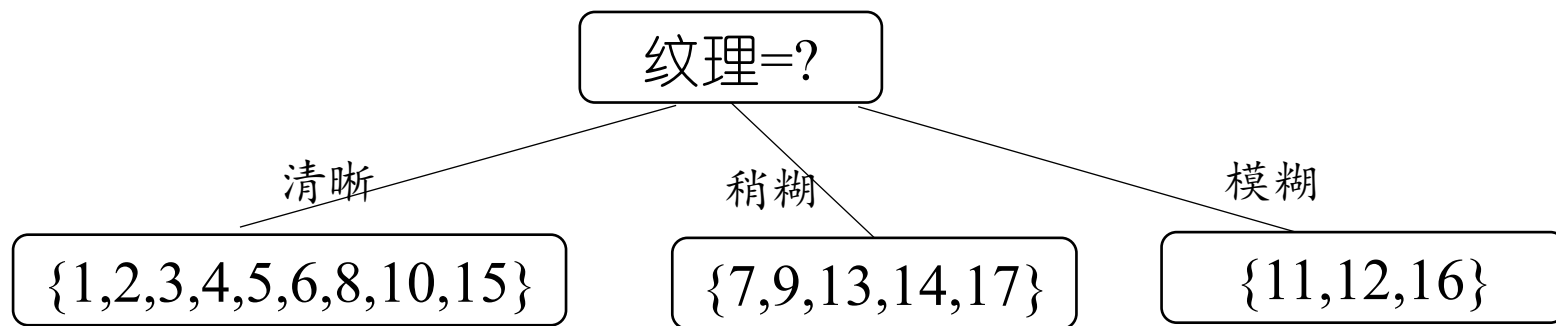
$$\text{Gain}(D, \text{敲声}) = 0.141$$

$$\text{Gain}(D, \text{纹理}) = 0.381$$

$$\text{Gain}(D, \text{脐部}) = 0.289$$

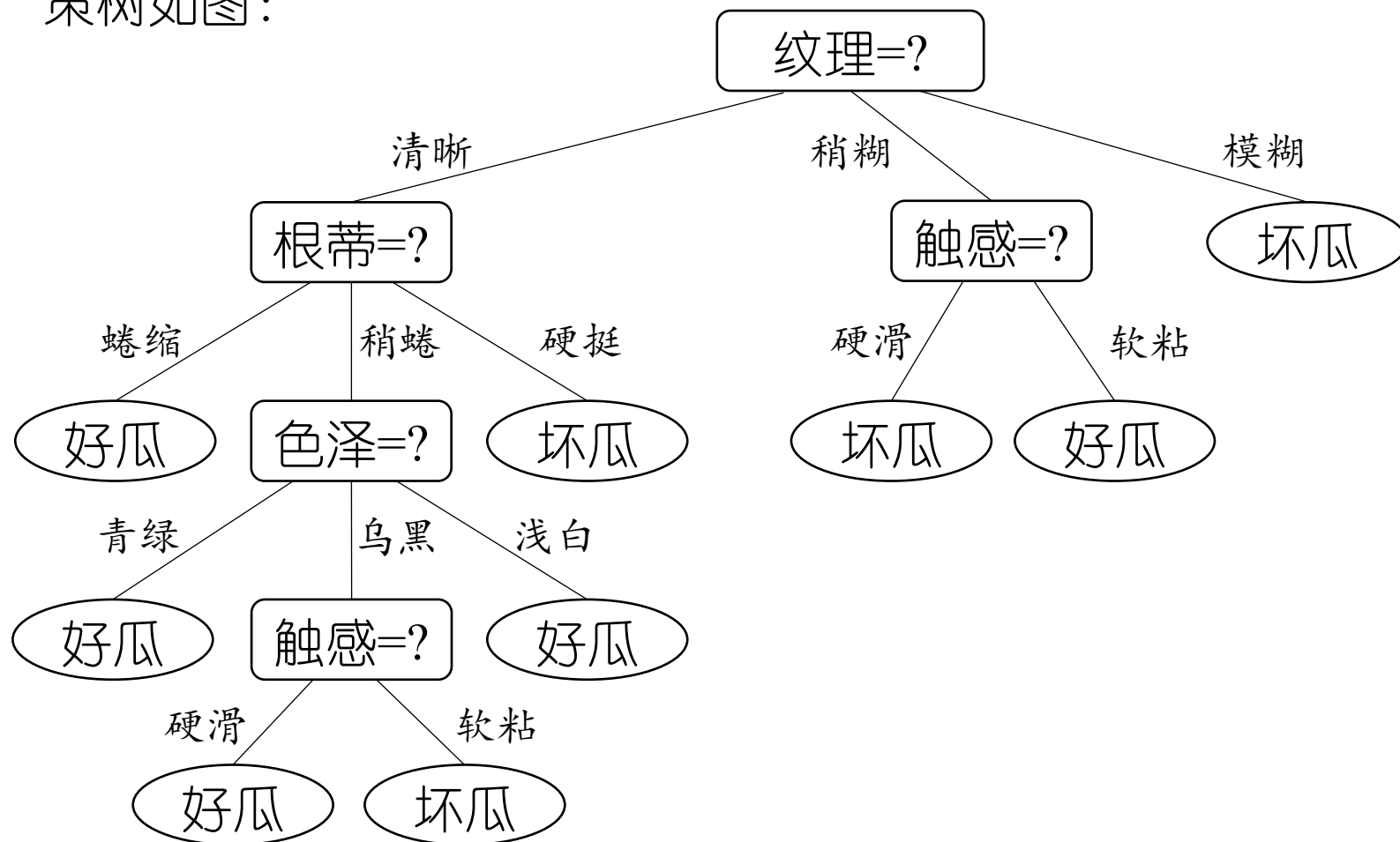
$$\text{Gain}(D, \text{触感}) = 0.006$$

□ 显然，属性“纹理”的信息增益最大，其被选为划分属性



划分选择-信息增益

- 决策树学习算法将对每个分支结点做进一步划分，最终得到的决策树如图：



划分选择-信息增益

存在的问题

- 若把“编号”也作为一个候选划分属性，则其信息增益一般远大于其他属性。显然，这样的决策树不具有泛化能力，无法对新样本进行有效预测

信息增益对可取值数目较多的属性有所偏好

划分选择-增益率

□ 增益率定义：

$$\text{Gain_ratio}(D, a) = \frac{\text{Gain}(D, a)}{\text{IV}(a)}$$

其中

$$\text{IV}(a) = - \sum_{v=1}^V \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

称为属性 a 的“固有值” [Quinlan, 1993]，属性 a 的可能取值数目越多（即 V 越大），则 $\text{IV}(a)$ 的值通常就越大

□ 存在的问题

增益率准则对可取值数目较少的属性有所偏好

□ C4.5 [Quinlan, 1993]使用了一个启发式：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的

划分选择-基尼指数

- 数据集 D 的纯度可用“基尼值”来度量

$$\text{Gini}(D) = \sum_{k=1}^{|Y|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|Y|} p_k^2$$

反映了从 D 中随机抽取两个样本，其类别标记不一致的概率

$\text{Gini}(D)$ 越小，数据集 D 的纯度越高

- 属性 a 的基尼指数定义为：

$$\text{Gini_index}(D, a) = \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Gini}(D^v)$$

- 应选择那个使划分后基尼指数最小的属性作为最优划分属性，即

$$a_* = \underset{a \in A}{\text{argmin}} \text{Gini_index}(D, a)$$

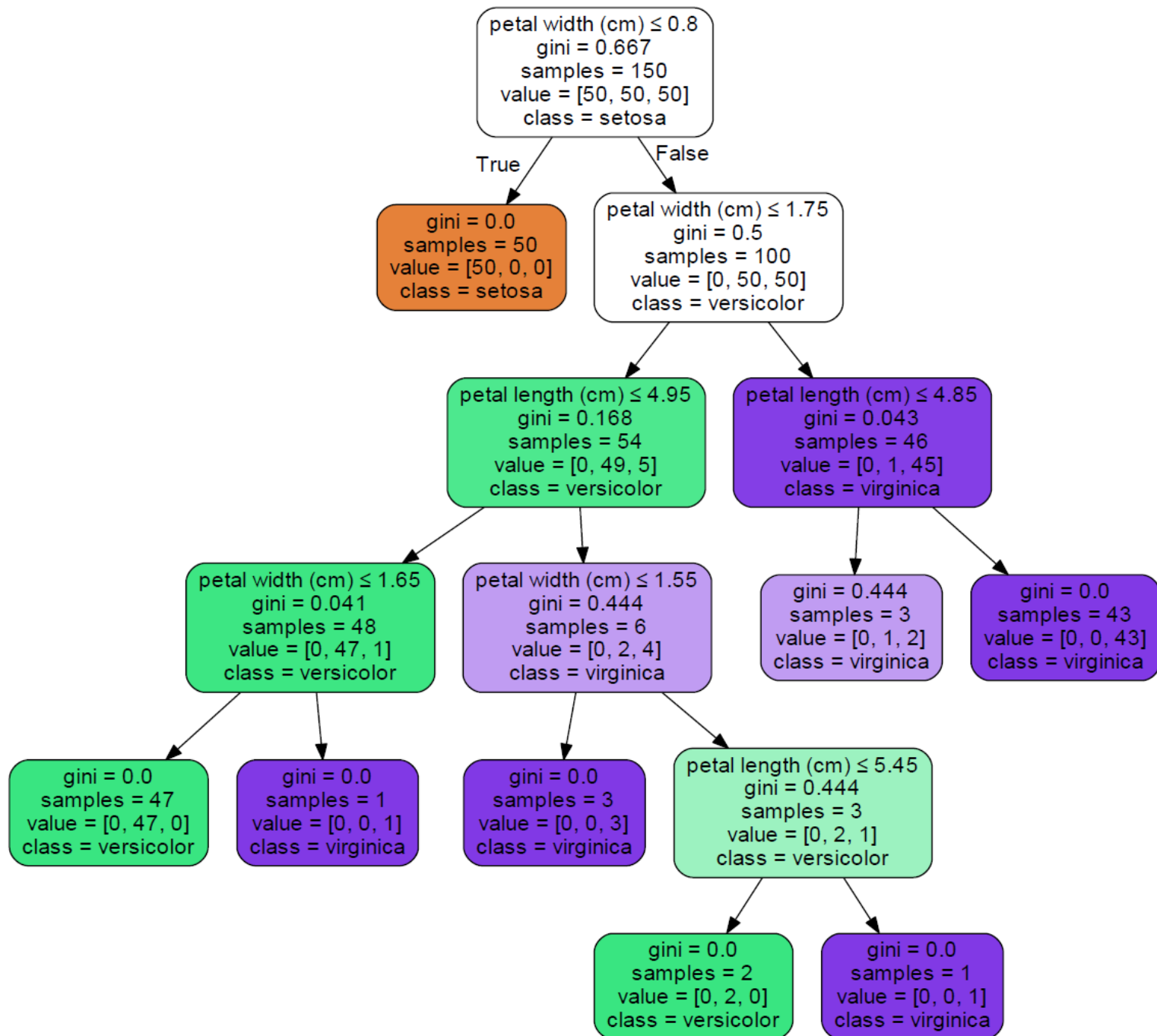
- CART [Breiman et al., 1984]采用“基尼指数”来选择划分属性

❑ CART (Classification and Regression Tree)

❑ Breiman等人（1984年）提出

❑ CART决策树是二叉树

- 分类：基尼指数最小化准则
- 回归：平方误差最小化准则



大纲

- 基本流程
- 划分选择
- 剪枝处理
- 连续与缺失值
- 多变量决策树

剪枝处理

□ 为什么剪枝

- “剪枝”是决策树学习算法对付“过拟合”的主要手段
- 可通过“剪枝”来一定程度避免因决策分支过多，以致于把训练集自身的一些特点当做所有数据都具有的一般性质而导致的过拟合

□ 剪枝的基本策略

- 预剪枝
- 后剪枝

□ 判断决策树泛化性能是否提升的方法

- 留出法：预留一部分数据用作“验证集”以进行性能评估

剪枝处理

数据集

训练集

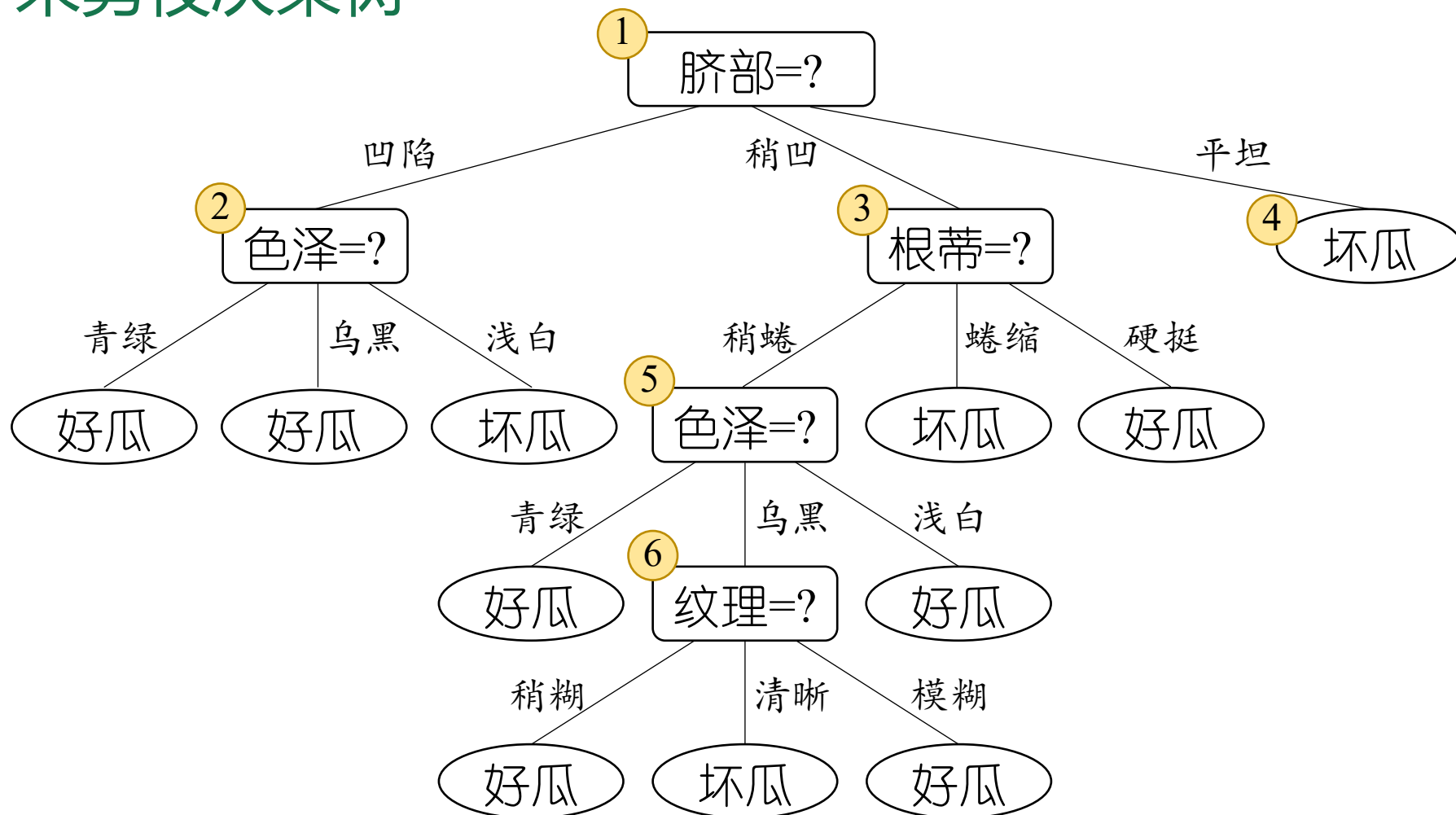
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

验证集

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

剪枝处理

未剪枝决策树



剪枝处理-预剪枝

- 决策树生成过程中，对每个结点在划分前先进行估计，若当前结点的划分不能带来决策树泛化性能提升，则停止划分并将当前结点记为叶结点，其类别标记为训练样例数最多的类别
- 针对上述数据集，基于信息增益准则，选取属性“脐部”划分训练集。分别计算划分前（即直接将该结点作为叶结点）及划分后的验证集精度，判断是否需要划分。若划分后能提高验证集精度，则划分，对划分后的属性，执行同样判断；否则，不划分

剪枝处理-预剪枝

验证集

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

结点1：若不划分，则将其标记为叶结点，类别标记为训练样例中最多的类别，即好瓜。验证集中，{4, 5, 8}被分类正确，得到验证集精度为 $\frac{3}{7} \times 100\% = 42.9\%$

1
脐部=?

验证集精度

← “脐部=?” 划分前: 42.9%

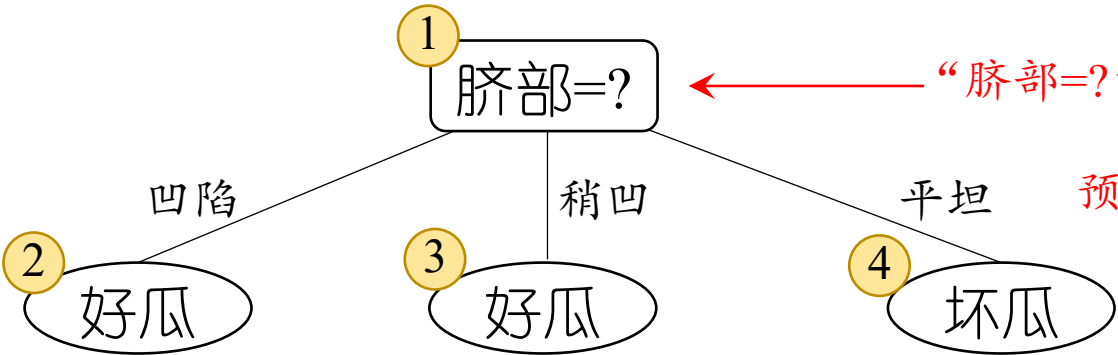
剪枝处理-预剪枝

验证集

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

结点1：若划分，根据结点②，③，④的训练样例，将这3个结点分别标记为“好瓜”、“好瓜”、“坏瓜”。此时，验证集中编号为{4, 5, 8, 11, 12}的样例被划分正确，验证集精度为 $\frac{5}{7} \times 100\% = 71.4\%$

验证集精度



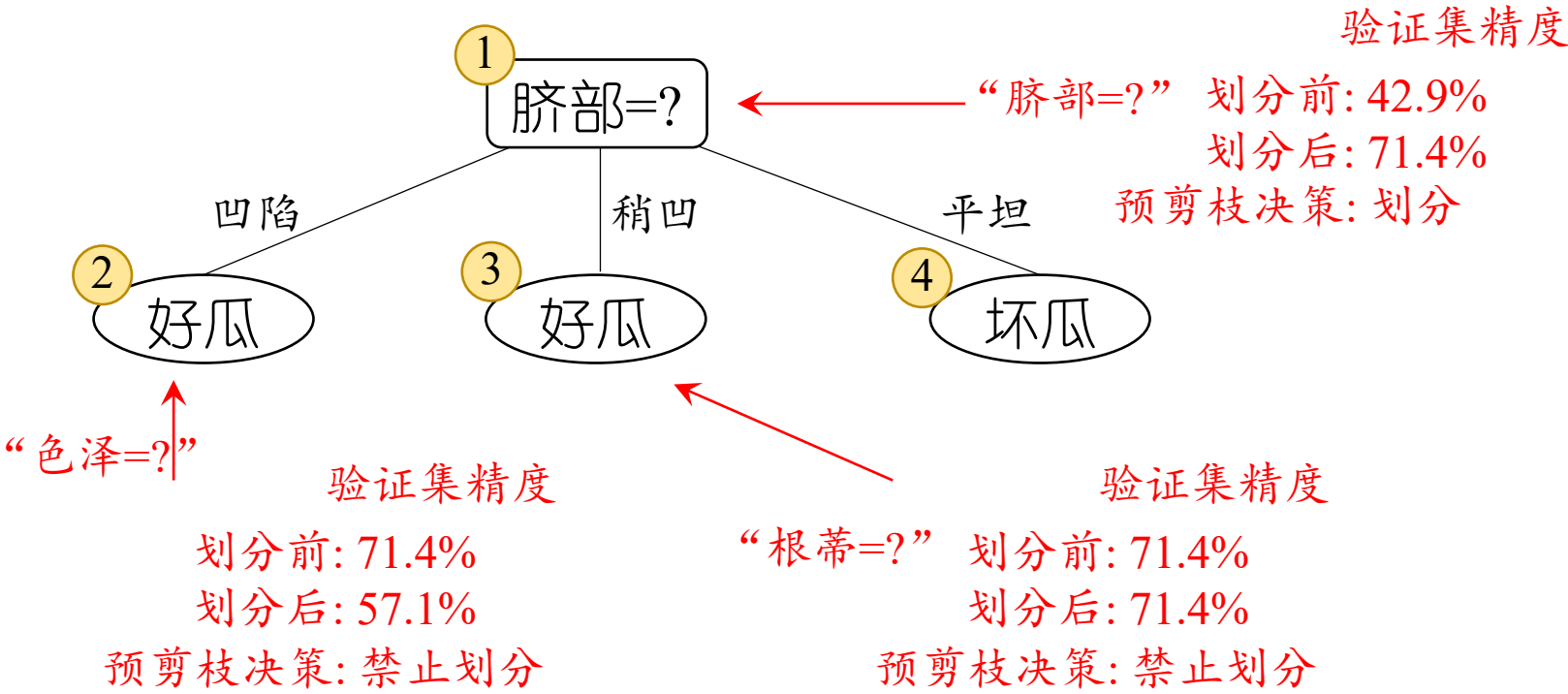
“脐部=?” 划分前: 42.9%
划分后: 71.4%
预剪枝决策: 划分

剪枝处理-预剪枝

验证集

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

对结点②, ③, ④ 分别进行剪枝判断, 结点②, ③都禁止划分, 结点④ 本身为叶子结点。最终得到仅有一层划分的决策树, 称为“决策树桩”



剪枝处理-预剪枝

预剪枝的优缺点

□ 优点

- 降低过拟合风险
- 显著减少训练时间和测试时间开销

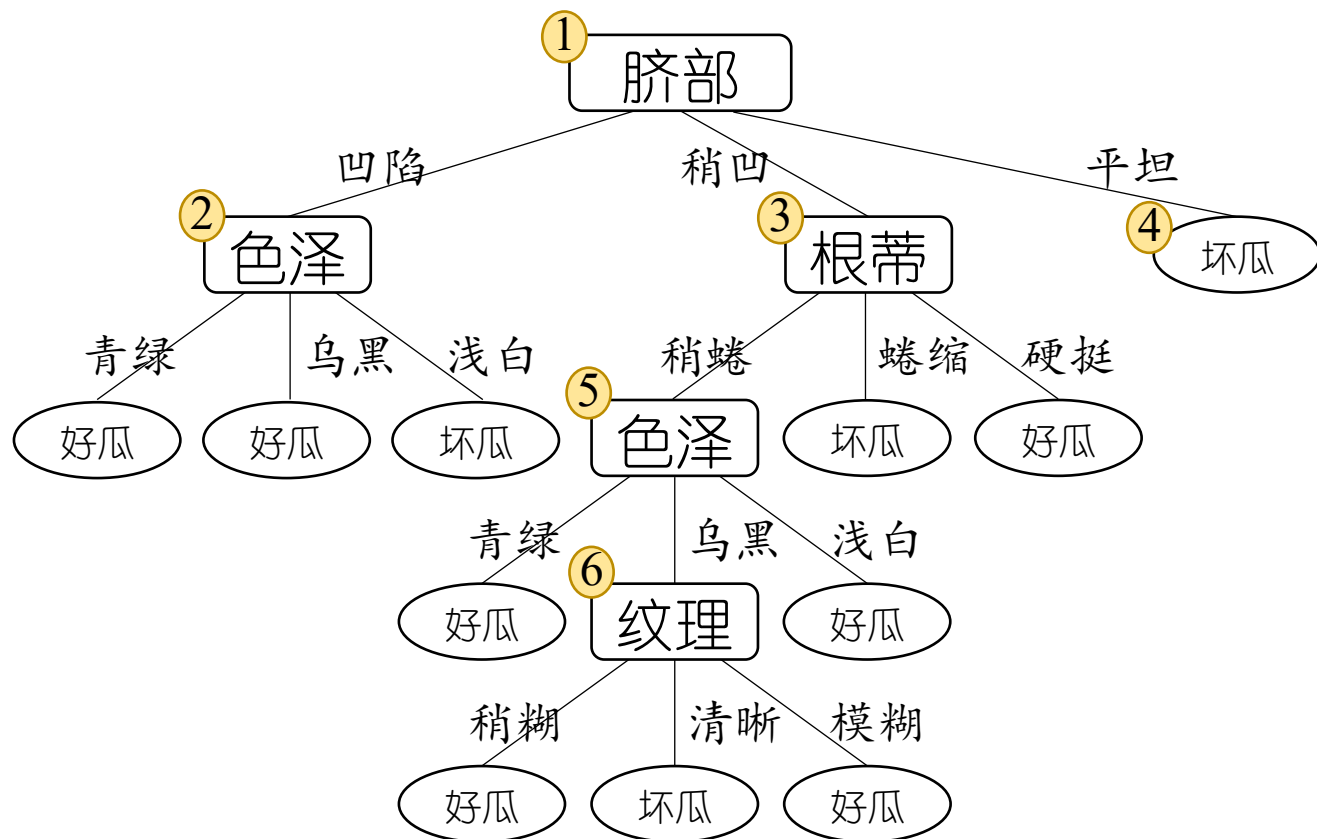
□ 缺点

- 欠拟合风险：有些分支的当前划分虽然不能提升泛化性能，但在其基础上进行的后续划分却有可能导致性能显著提高。预剪枝基于“贪心”本质禁止这些分支展开，带来了欠拟合风险

剪枝处理-后剪枝

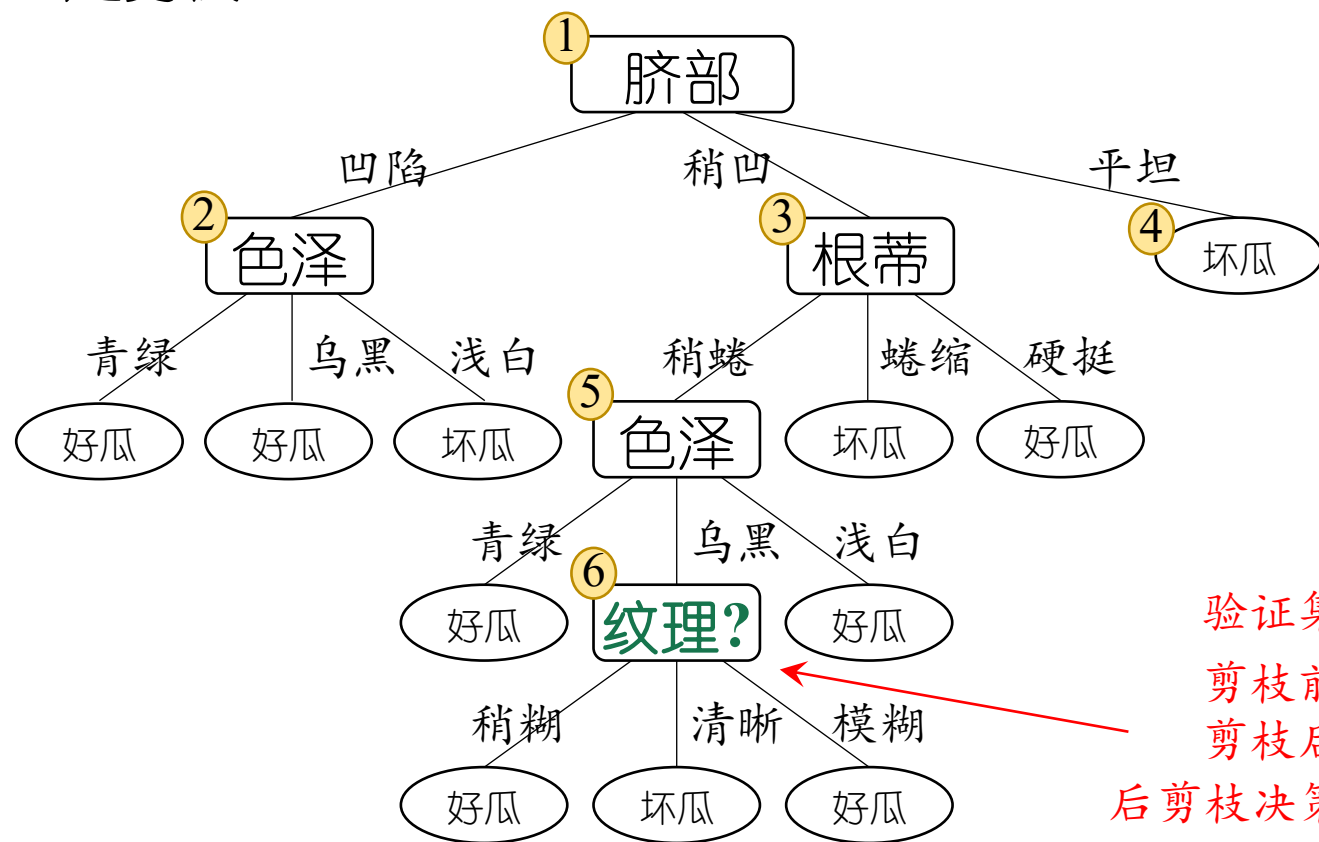
- 先从训练集生成一棵完整的决策树，然后自底向上地对非叶结点进行考察，若将该结点对应的子树替换为叶结点能带来决策树泛化性能提升，则将该子树替换为叶结点

首先生成一棵完整的决策树，该决策树的验证集精度为42.9%



剪枝处理-后剪枝

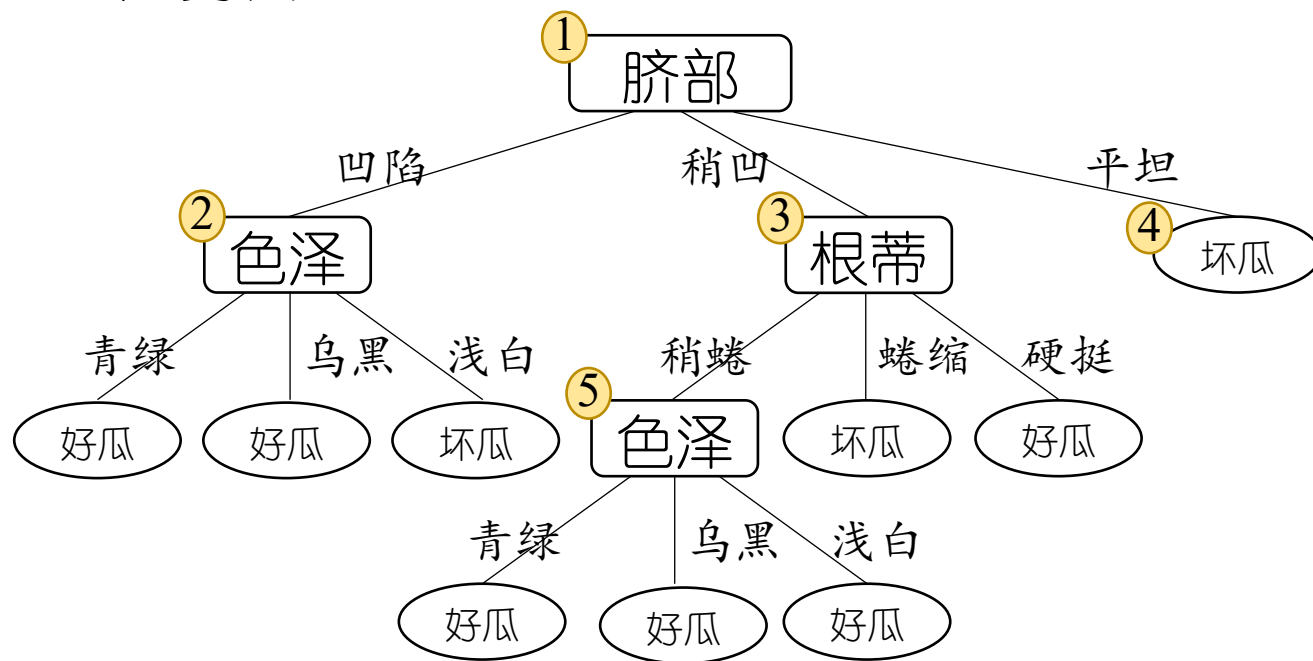
- 首先考虑结点⑥，若将其替换为叶结点，根据落在其上的训练样本{7, 15}将其标记为“好瓜”，得到验证集精度提高至57.1%，则决定剪枝



验证集精度
剪枝前: 42.9%
剪枝后: 57.1%
后剪枝决策: 剪枝

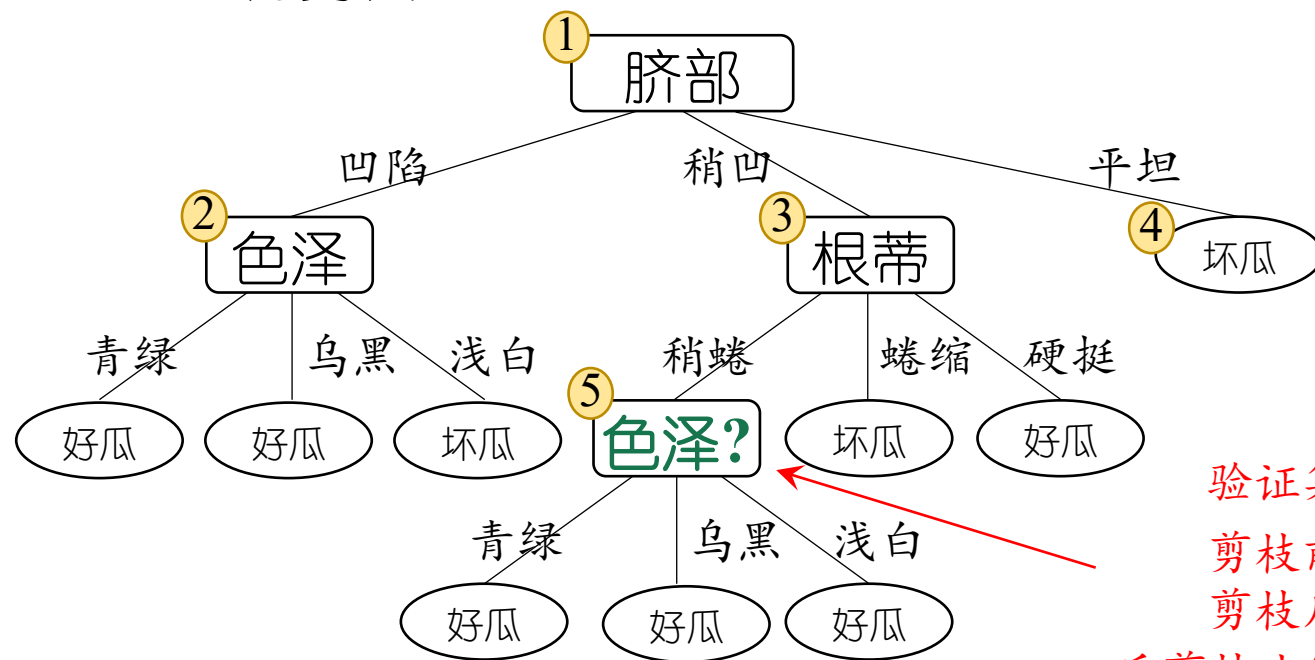
剪枝处理-后剪枝

- 首先考虑结点⑥，若将其替换为叶结点，根据落在其上的训练样本{7, 15}将其标记为“好瓜”，得到验证集精度提高至57.1%，则决定剪枝



剪枝处理-后剪枝

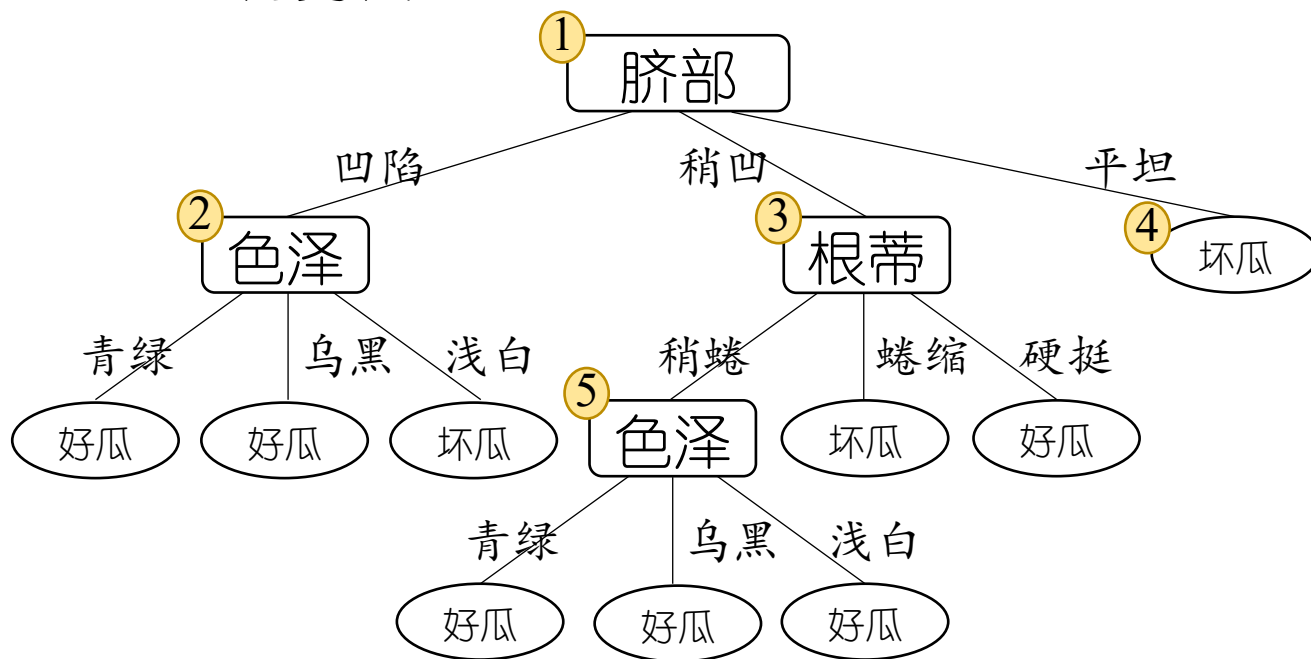
- 然后考虑结点⑤，若将其替换为叶结点，根据落在其上的训练样本{6, 7, 15}将其标记为“好瓜”，得到验证集精度仍为57.1%，可以不进行剪枝



验证集精度
剪枝前: 57.1 %
剪枝后: 57.1 %
后剪枝决策: 不剪枝

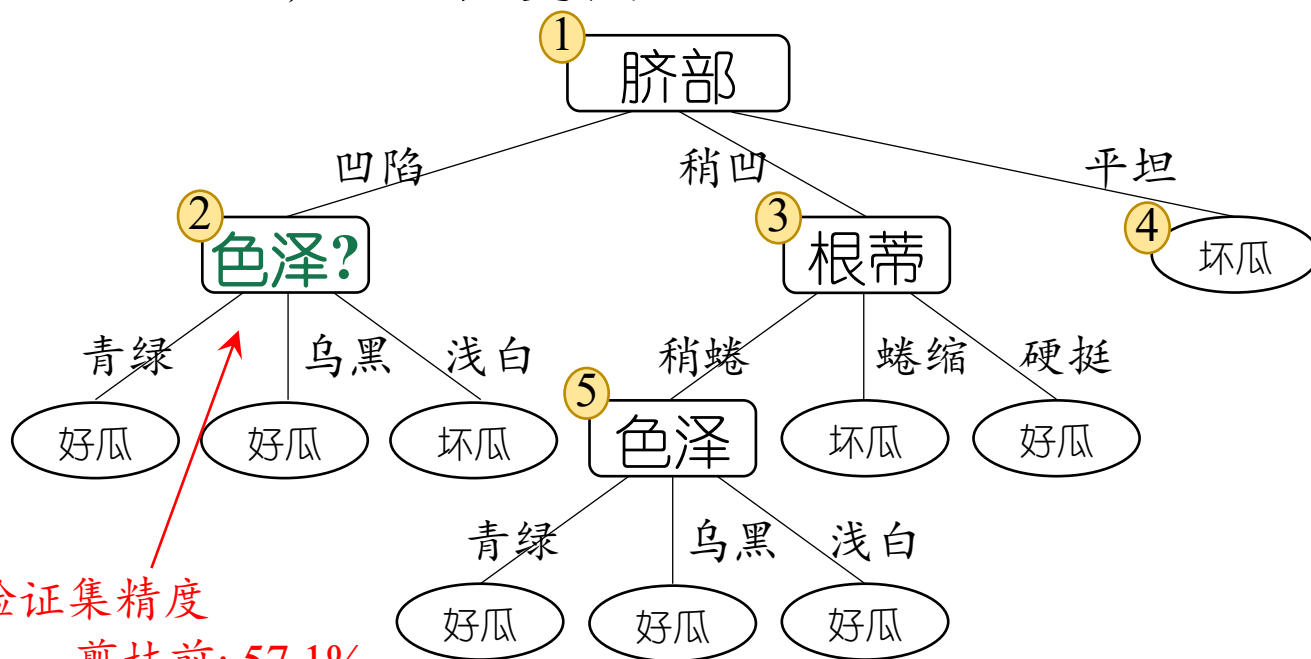
剪枝处理-后剪枝

- 然后考虑结点⑤，若将其替换为叶结点，根据落在其上的训练样本{6, 7, 15}将其标记为“好瓜”，得到验证集精度仍为57.1%，可以不进行剪枝



剪枝处理-后剪枝

- 对结点②，若将其替换为叶结点，根据落在其上的训练样本{1, 2, 3, 14}，将其标记为“好瓜”，得到验证集精度提升至71.4%，则决定剪枝



验证集精度

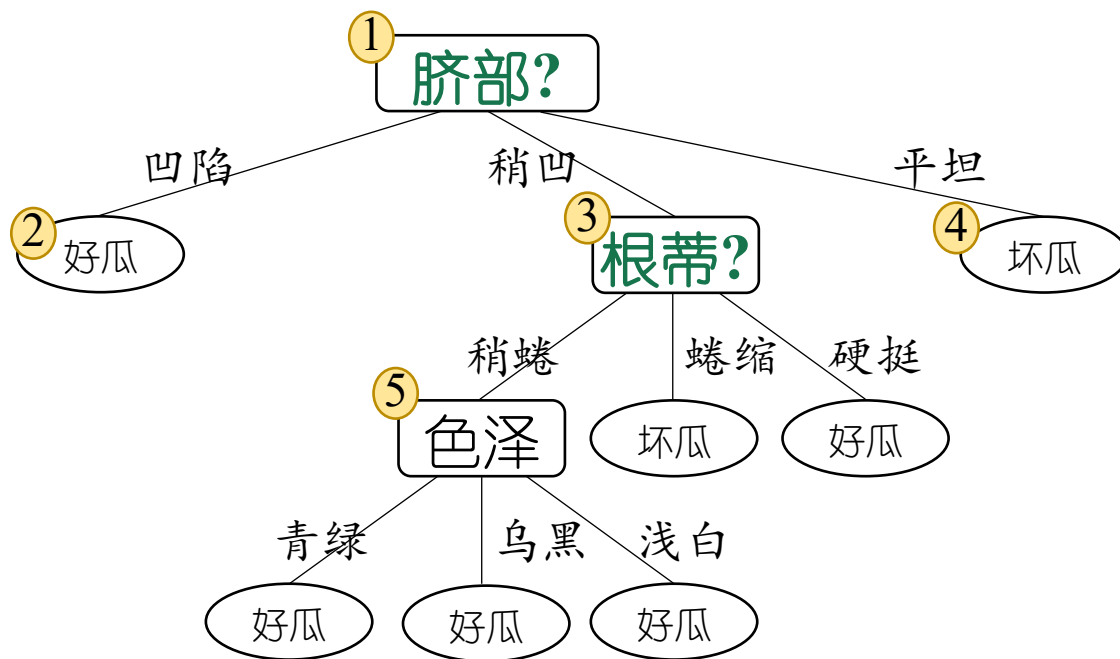
剪枝前: 57.1%

剪枝后: 71.4%

后剪枝决策: 剪枝

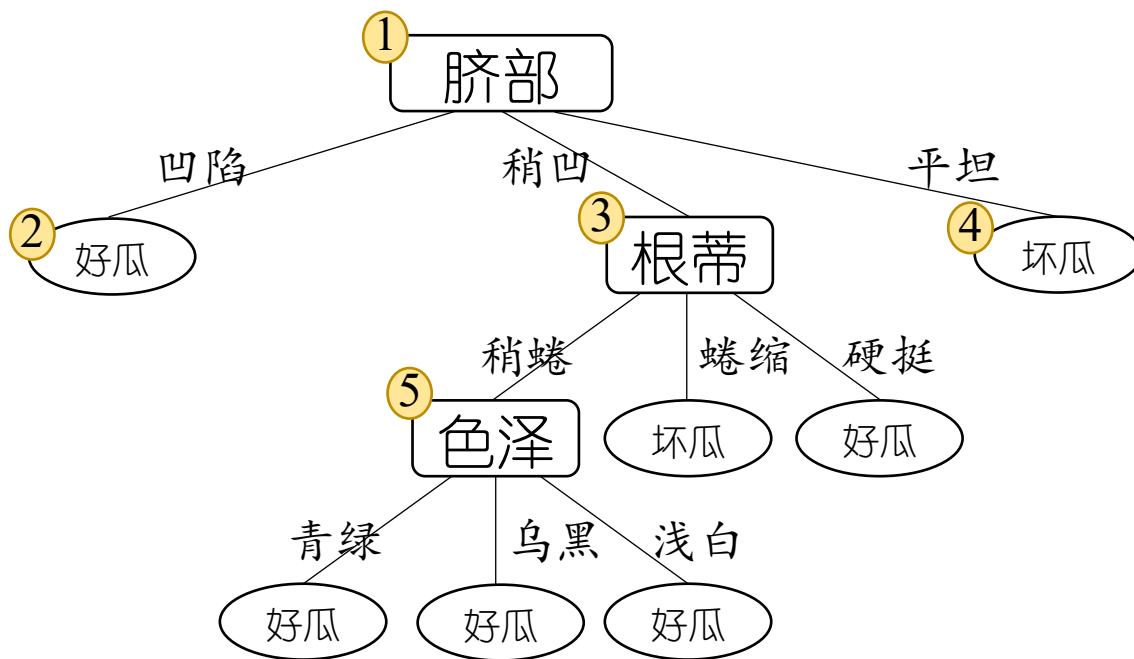
剪枝处理-后剪枝

- 对结点③和①，先后替换为叶结点，验证集精度均未提升，则分支得到保留



剪枝处理-后剪枝

□ 最终基于后剪枝策略得到的决策树如图所示



剪枝处理-后剪枝

后剪枝的优缺点

□ 优点

- 后剪枝比预剪枝保留了更多的分支，欠拟合风险小，泛化性能往往优于预剪枝决策树

□ 缺点

- 训练时间开销大：后剪枝过程是在生成完全决策树之后进行的，需要自底向上对所有非叶结点逐一考察

大纲

- 基本流程
- 划分选择
- 剪枝处理
- 连续与缺失值
- 多变量决策树

连续与缺失值 - 连续值处理

□ 连续属性离散化(二分法)

- 第一步：假定连续属性 a 在样本集 D 上出现 n 个不同的取值，从小到大排列，记为 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^- 包含那些在属性 a 上取值不大于 t 的样本， D_t^+ 包含那些在属性 a 上取值大于 t 的样本。考虑包含 $n - 1$ 个元素的候选划分点集合

$$T_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq n - 1 \right\}$$

即把区间 $[a^i, a^{i+1})$ 的中位点 $\frac{a^i + a^{i+1}}{2}$ 作为候选划分点

连续与缺失值 - 连续值处理

□ 连续属性离散化(二分法)

- 第二步：采用离散属性值方法，考察这些划分点，选取最优的划分点进行样本集合的划分

$$\begin{aligned}\text{Gain}(D, a) &= \max_{t \in T_a} \text{Gain}(D, a, t) \\ &= \max_{t \in T_a} \text{Ent}(D) - \sum_{\lambda \in \{-, +\}} \frac{|D_t^\lambda|}{|D|} \text{Ent}(D_t^\lambda)\end{aligned}$$

其中 $\text{Gain}(D, a, t)$ 是样本集 D 基于划分点 t 二分后的信息增益，于是，就可选择使 $\text{Gain}(D, a, t)$ 最大化的划分点

连续与缺失值 - 连续值处理

连续值处理实例

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	0.460	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	0.360	0.370	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

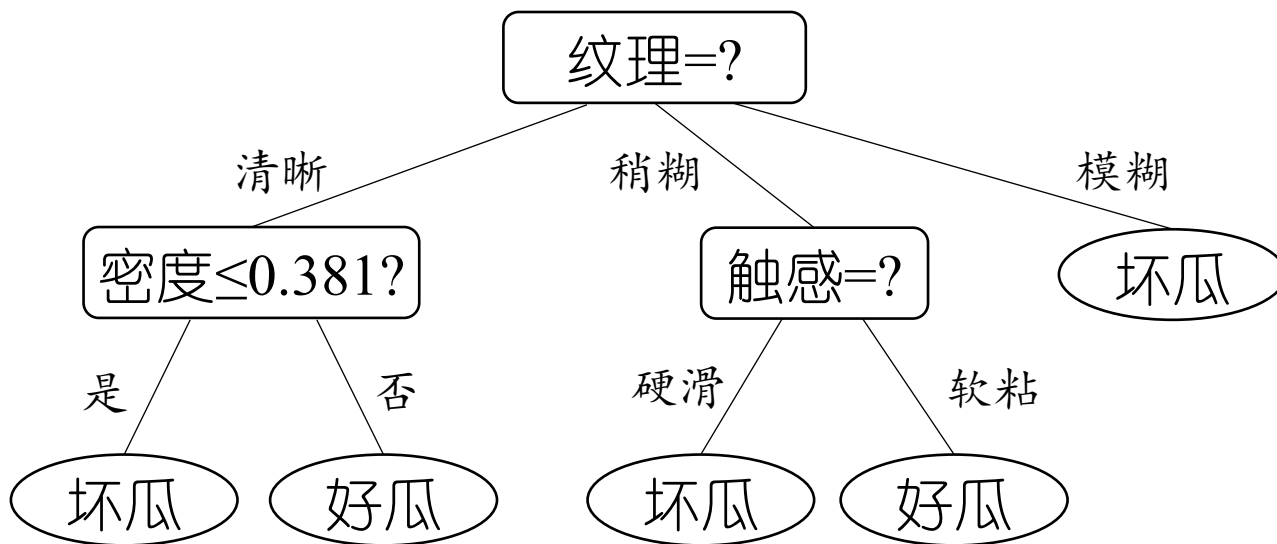
对属性“密度”，其候选划分点集合包含16个候选值：

$$T_{\text{密度}} = \{0.244, 0.294, 0.351, 0.381, 0.420, 0.459, 0.518, 0.574, 0.600, 0.621, 0.636, 0.648, 0.661, 0.681, 0.708, 0.746\}$$

可计算其信息增益为0.262，对应划分点为0.381

对属性“含糖量”进行同样处理

连续与缺失值 - 连续值处理



与离散属性不同，若当前结点划分属性为连续属性，该属性还可作为其后代结点的划分属性

连续与缺失值 - 缺失值处理

- 不完整样本，即样本的属性值缺失
- 仅使用无缺失的样本进行学习？

对数据信息极大的浪费

- 使用有缺失值的样本，需要解决哪些问题？

Q1：如何在属性缺失的情况下进行划分属性选择？

Q2：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？

连续与缺失值 - 缺失值处理

□ \tilde{D} 表示 D 中在属性 a 上没有缺失值的样本子集, \tilde{D}^v 表示 \tilde{D} 中在属性 a 上取值为 a^v 的样本子集, \tilde{D}_k 表示 \tilde{D} 中属于第 k 类的样本子集

为每个样本 \mathbf{x} 赋予一个权重 w_x , 并定义:

- 无缺失值样本所占的比例

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

- 无缺失值样本中第 k 类所占比例

$$\tilde{p}_k = \frac{\sum_{x \in \tilde{D}_k} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq k \leq |\mathcal{Y}|)$$

- 无缺失值样本中在属性 a 上取值 a^v 的样本所占比例

$$\tilde{r}_v = \frac{\sum_{x \in \tilde{D}^v} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq v \leq V)$$

Q1: 如何在属性缺失的情况下进行划分属性选择?

连续与缺失值 - 缺失值处理

□ 基于上述定义，可得

$$\begin{aligned}\text{Gain}(D, a) &= \rho \times \text{Gain}(\tilde{D}, a) \\ &= \rho \times \left(\text{Ent}(\tilde{D}) - \sum_{v=1}^V \tilde{r}_v \text{Ent}(\tilde{D}^v) \right)\end{aligned}$$

其中

$$\text{Ent}(\tilde{D}) = - \sum_{k=1}^{|\mathcal{Y}|} \tilde{p}_k \log_2 \tilde{p}_k$$

□ 对于Q2

- 若样本 \mathbf{x} 在划分属性 a 上的取值已知，则将 \mathbf{x} 划入与其取值对应的子结点，且样本权值在子结点中保持为 w_x
- 若样本 \mathbf{x} 在划分属性 a 上的取值未知，则将 \mathbf{x} 同时划入所有子结点，且样本权值在与属性值 a^v 对应的子结点中调整为 $\tilde{r}_v \cdot w_x$ (直观来看，相当于让同一个样本以不同概率划入不同的子结点中去)

连续与缺失值 - 缺失值处理

缺失值处理实例

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

- 学习开始时，根结点包含样本集 D 中全部17个样例，各样例的权值均为1
- 以属性“色泽”为例，该属性上无缺失值的样例子集 \tilde{D} 包含14个样例， \tilde{D} 的信息熵为

$$\text{Ent}(\tilde{D}) = - \sum_{k=1}^2 \tilde{p}_k \log_2 \tilde{p}_k$$

$$= -\left(\frac{6}{14} \log_2 \frac{6}{14} + \frac{8}{14} \log_2 \frac{8}{14}\right) = 0.985$$

连续与缺失值 - 缺失值处理

- 令 $\tilde{D}^1, \tilde{D}^2, \tilde{D}^3$ 分别表示在属性“色泽”上取值为“青绿”“乌黑”以及“浅白”的样本子集，有

$$\text{Ent}(\tilde{D}^1) = -\left(\frac{2}{4} \log_2 \frac{2}{4} + \frac{2}{4} \log_2 \frac{2}{4}\right) = 1.000 \quad \text{Ent}(\tilde{D}^2) = -\left(\frac{4}{6} \log_2 \frac{4}{6} + \frac{2}{6} \log_2 \frac{2}{6}\right) = 0.918$$

$$\text{Ent}(\tilde{D}^3) = -\left(\frac{0}{4} \log_2 \frac{0}{4} + \frac{4}{4} \log_2 \frac{4}{4}\right) = 0.000$$

- 因此，样本子集 \tilde{D} 上属性“色泽”的信息增益为

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

- 于是，样本集 D 上属性“色泽”的信息增益为

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

连续与缺失值 - 缺失值处理

- 类似地可计算出所有属性在数据集上的信息增益

$$\text{Gain}(D, \text{色泽}) = 0.252 \quad \text{Gain}(D, \text{根蒂}) = 0.171$$

$$\text{Gain}(D, \text{敲声}) = 0.145 \quad \text{Gain}(D, \text{纹理}) = 0.424$$


$$\text{Gain}(D, \text{脐部}) = 0.289 \quad \text{Gain}(D, \text{触感}) = 0.006$$

 进入“纹理=清晰”分支

 进入“纹理=稍糊”分支

 进入“纹理=模糊”分支

样本权重在各子结点仍为1

 在属性“纹理”上出现缺失值，样本8和10同时进入3个分支，调整8和10在3分支权值分别为7/15，5/15，3/15

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

CART回归树

□ 训练样本集

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

□ 回归树模型

$$f(x) = \frac{1}{|R_m|} \sum_{x_i \in R_m} y_i, x \in R_m, m = 1, \dots, M$$

□ 切分变量与切分点

$$R_1(a, b) = \{A | a \leq b\} \quad R_2(a, b) = \{A | a > b\}$$

5 CART回归算法

□ 整体误差平方和

$$\sum_{x_i \in R_1} (y_i - \bar{y}_1)^2 + \sum_{x_i \in R_2} (y_i - \bar{y}_2)^2$$

□ 从所有切分变量及其切分点中选取划分区域后整体误差平方和最小的作为当前结点最优切分变量和最优切分点

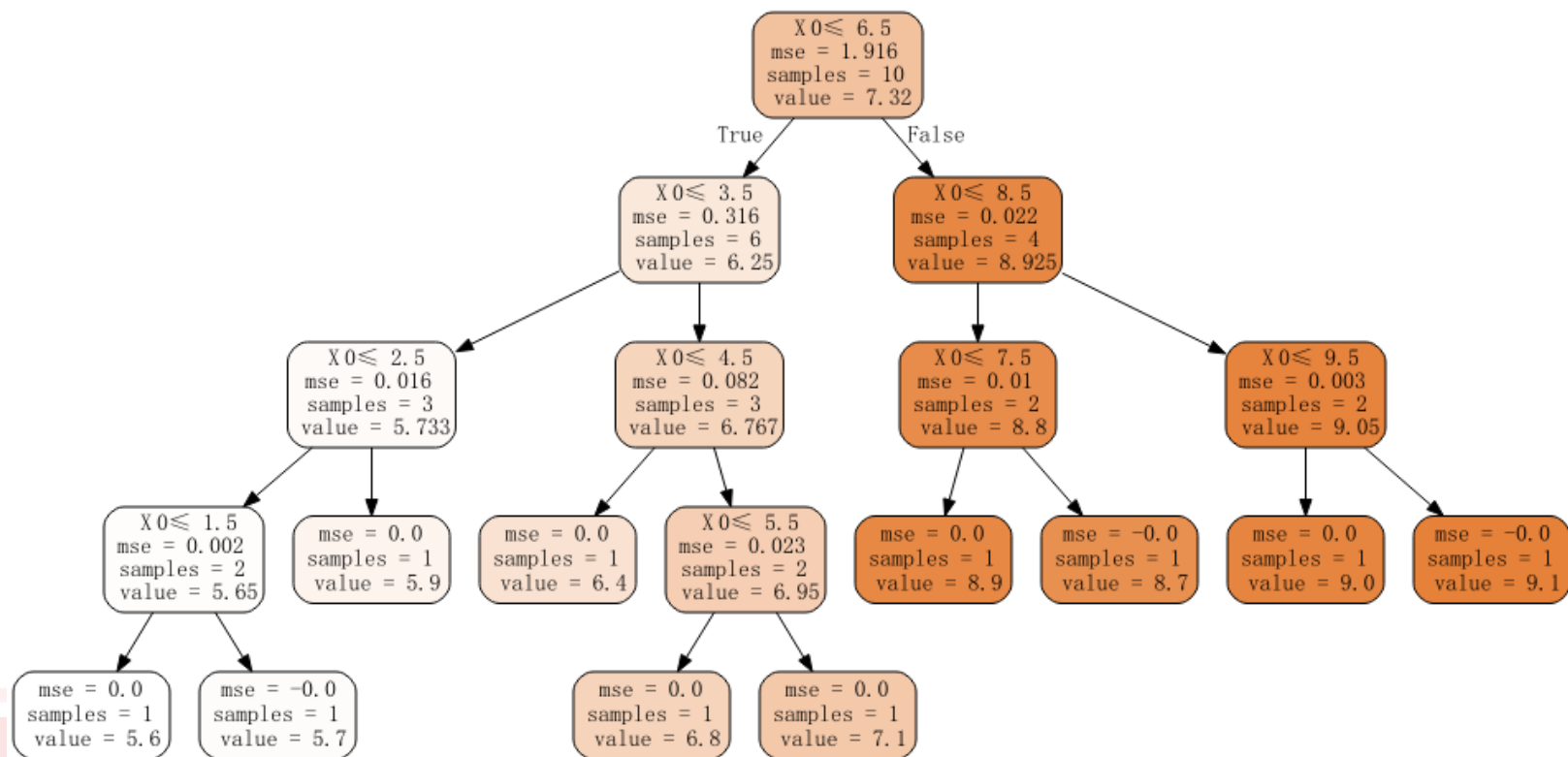
□ 从根结点开始递归生成**CART**回归决策树

□ 最小二乘回归树

5 CART回归树示例

□ 一维数据集

x	1	2	3	4	5	6	7	8	9	10
y	5.6	5.7	5.9	6.4	6.8	7.1	8.9	8.7	9	9.1

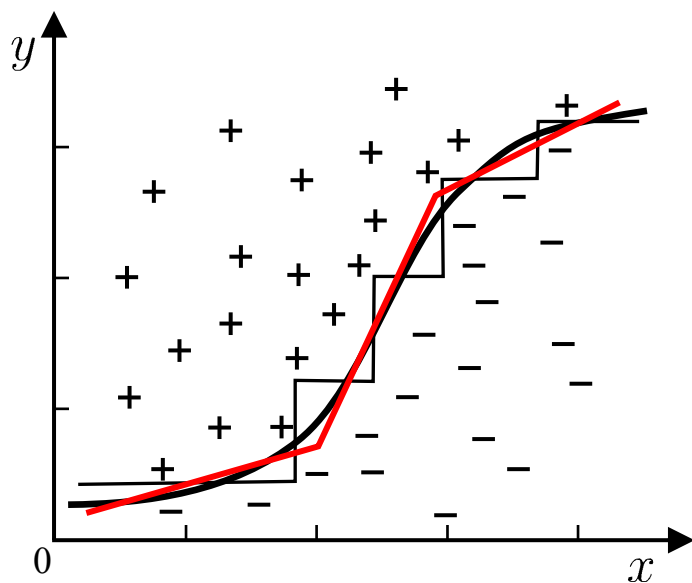


大纲

- 基本流程
- 划分选择
- 剪枝处理
- 连续与缺失值
- 多变量决策树

多变量决策树

- 单变量决策树分类边界:轴平行
- 多变量决策树



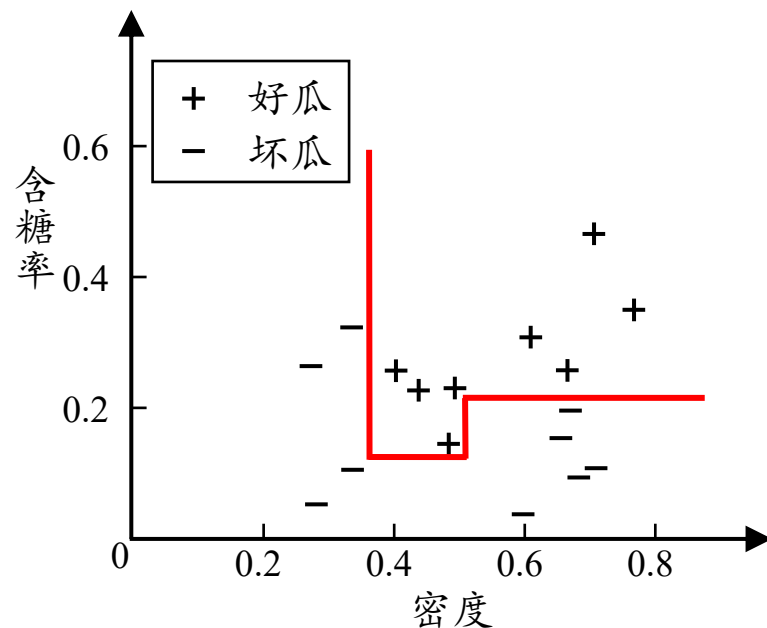
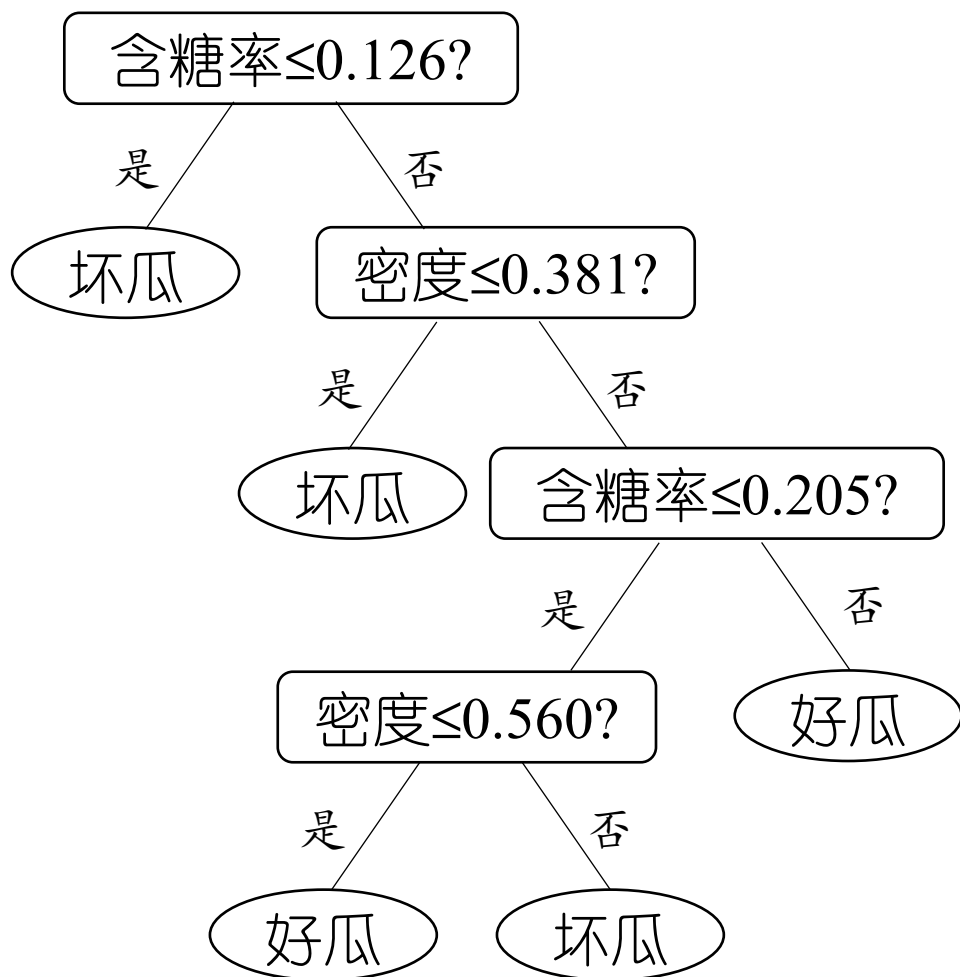
- 非叶节点不再是仅对某个属性, 而是对属性的线性组合



- 每个非叶结点是一个形如 $\sum_{i=1}^d w_i a_i = t$ 的线性分类器 w_i 其中 a_i 是属性 w_i 的权值, 和 可在该结点所含的样本集和属性集上学得

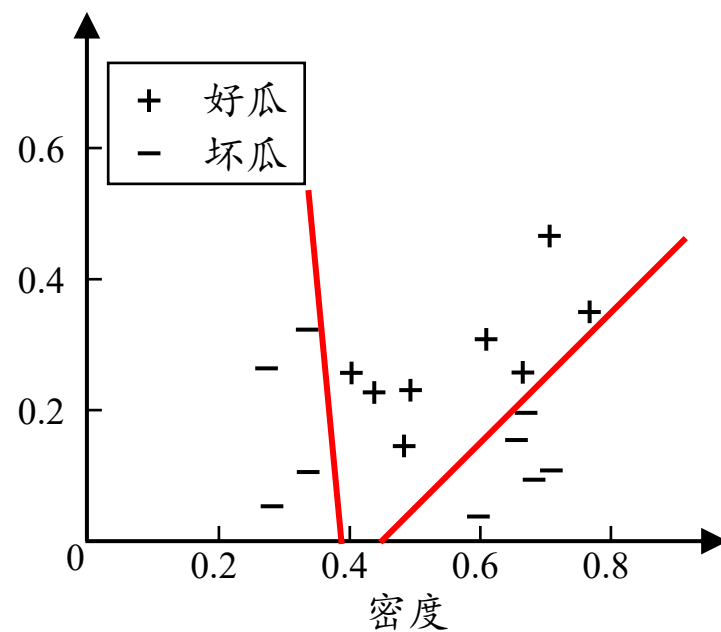
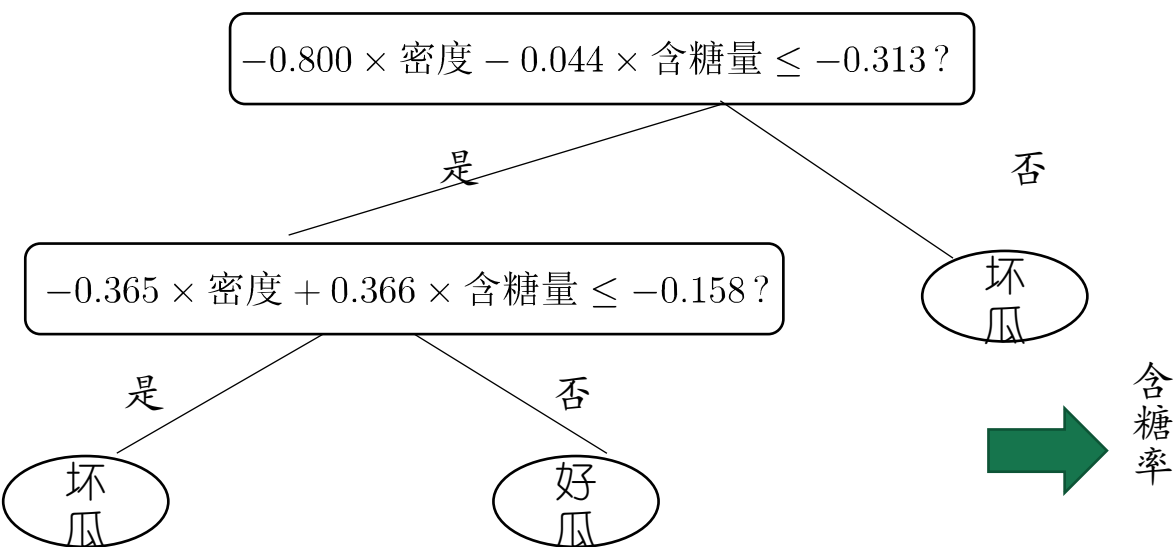
多变量决策树

□ 单变量决策树



多变量决策树

□ 多变量决策树



Some advantages of decision trees

- ❑ **Simple to understand and to interpret.** Trees can be visualised.
- ❑ **Requires little data preparation.** Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.
- ❑ The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- ❑ **Able to handle both numerical and categorical data.** Other techniques are usually specialised in analysing datasets that have only one type of variable. Able to handle multi-output problems.
- ❑ Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic.
- ❑ Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- ❑ Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

Some disadvantages of decision trees

- ❑ **Decision-tree learners can create over-complex trees that do not generalise the data well.** This is called overfitting. Mechanisms such as pruning (not currently supported), setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.
- ❑ **Decision trees can be unstable because small variations in the data might result in a completely different tree being generated.** This problem is mitigated by using decision trees within an ensemble.
- ❑ The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts.
- ❑ There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
- ❑ **Decision tree learners create biased trees if some classes dominate.** It is therefore recommended to balance the dataset prior to fitting with the decision tree.

成熟的决策树软件包

- ID3, C4.5, C5.0
<http://www.rulequest.com/Personal/>
- J48 (JAVA)
<http://www.cs.waikato.ac.nz/ml/weka/>

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