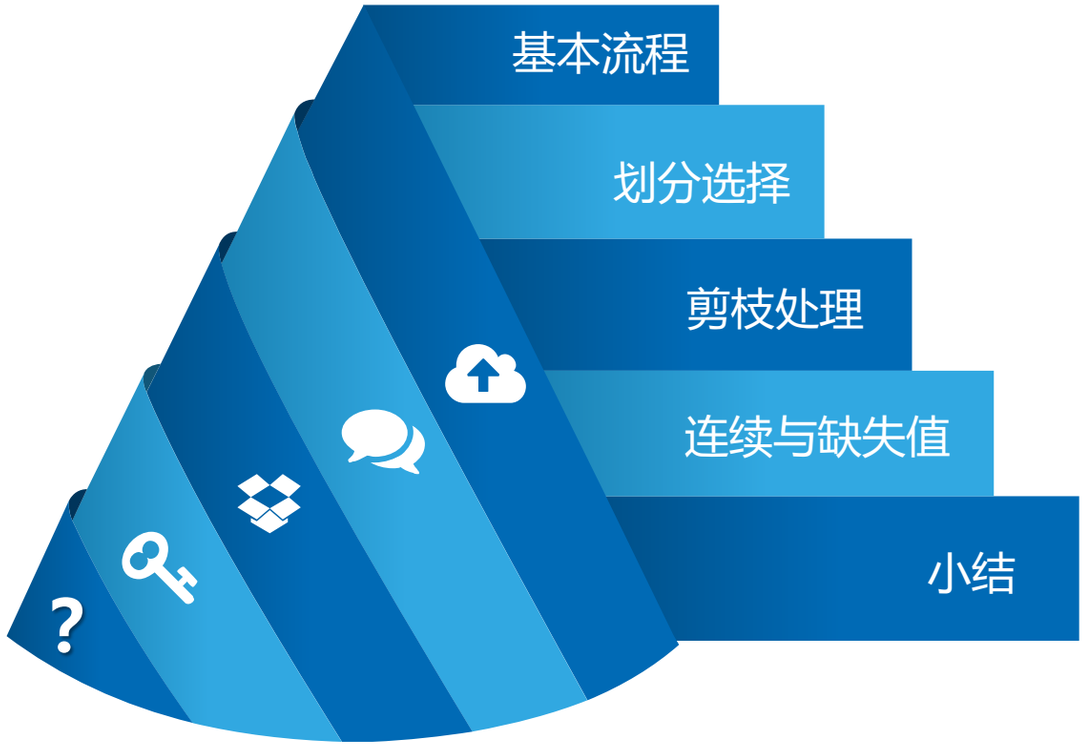
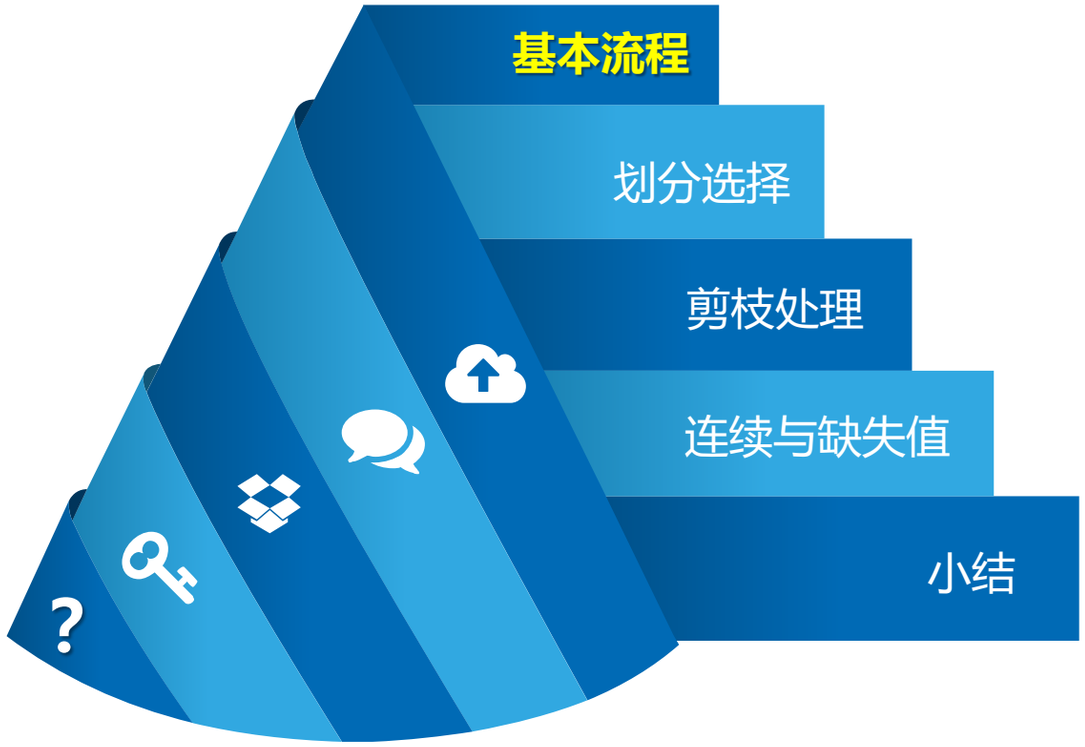
第6讲 决策树学习

周文晖

杭州电子科技大学



- **决策数基本流程**
  决策过程，基本流程，…
- **划分选择**
  信息熵，增益率，基尼指数，…
- **剪枝处理**
  预剪枝，后剪枝 …
- **连续与缺失值**
  连续值处理，缺失值处理 …
- **小结**
  决策树与深度学习…



决策数基本流程

决策过程，基本流程，…

划分选择

信息熵，增益率，基尼指数，…

剪枝处理

预剪枝，后剪枝 …

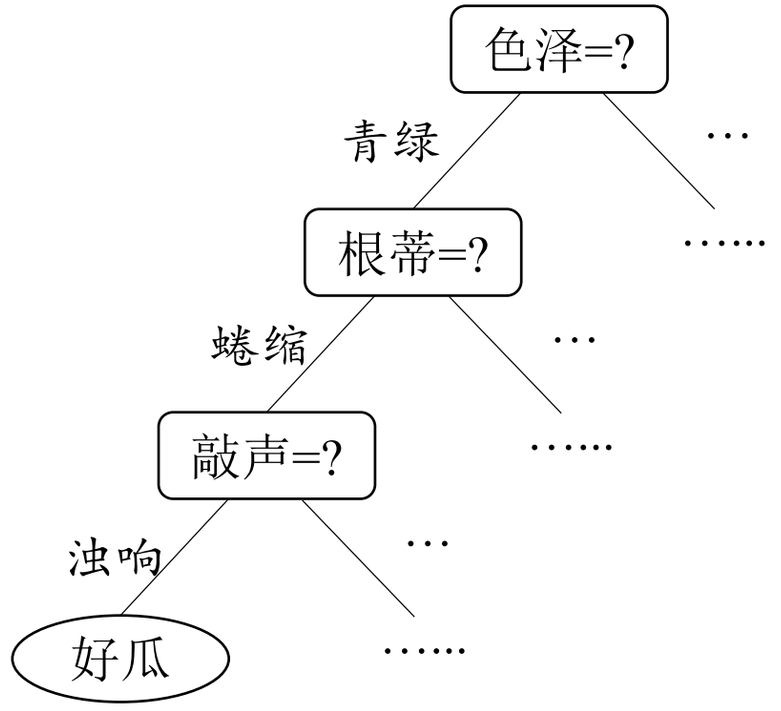
连续与缺失值

连续值处理，缺失值处理 …

小结

决策树与深度学习…

人类的决策过程



人类在做某些决策时会基于规则，这些规则是**人工总结或制定**的。

医生根据生理指标判定是否有病？

以及判断一个西瓜是否是好瓜？

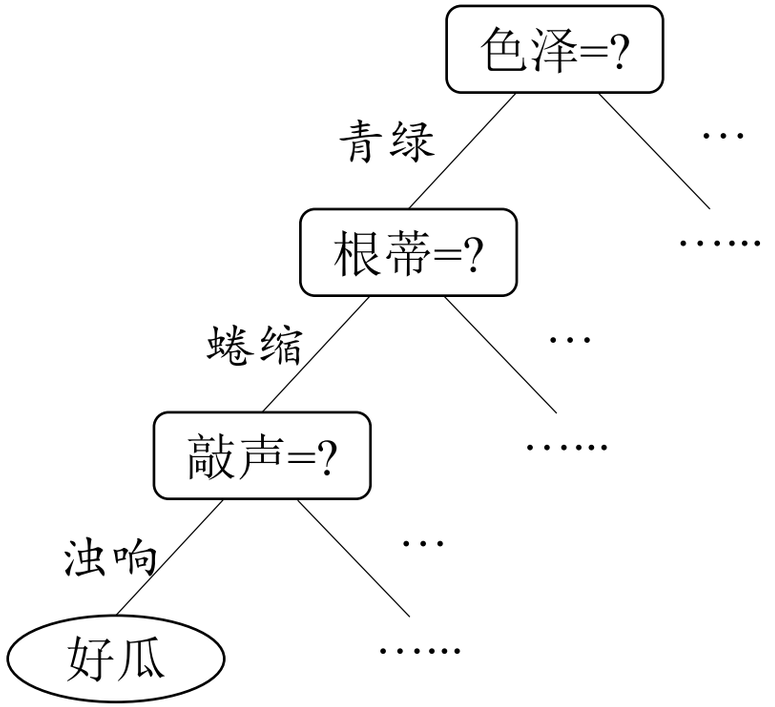
人类决策的特点：

1）决策过程中提出的每个判定问题都是对某个属性的“测试”；

2）决策过程的最终结论对应了我们所希望的判定结果；

3）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。

人类的决策过程→决策树（Decision Tree）



与人类决策过程类似，决策树也是这种基于规则的方法，它用一组嵌套的规则进行预测；

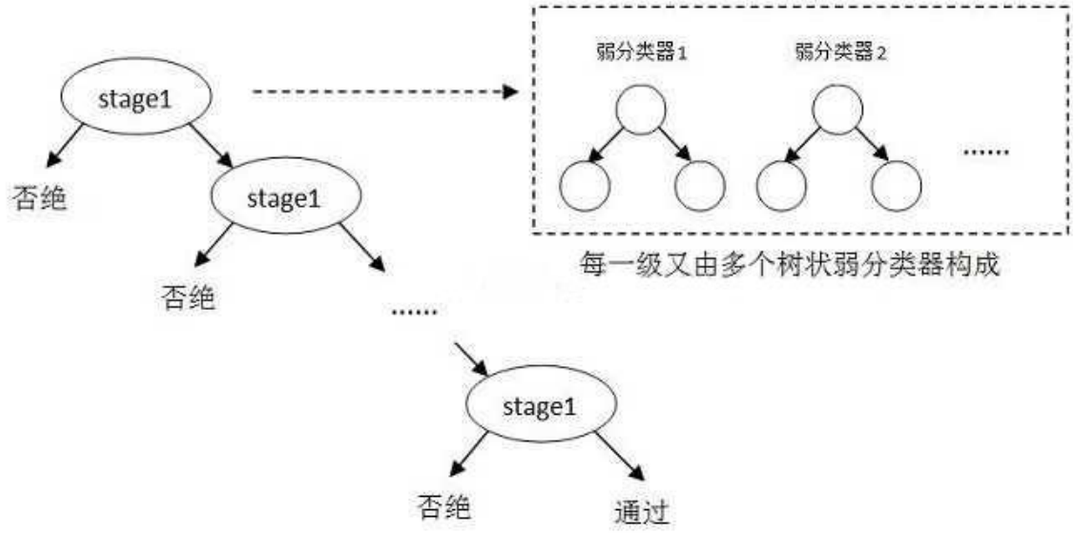
嵌套的规则形成一种基于树结构的预测：

- 由对象的若干属性、属性值和有关决策组成的一棵树；
- 决策树包含一个根结点、若干内部结点和叶结点；
- 决策树的结点为属性（一般为语言变量）；
- 树的分枝为相应的属性值（一般为语言值）；

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

决策树与集成学习（Ensemble Learning）

- 决策树是集成学习的基础学习器。
- 集成学习是一种将多个基础学习器（如决策树、支持向量机、随机森林等）组合在一起的学习方法，以提高模型的准确性和稳定性。
- 集成学习的基本思想：多个弱学习器（Weak Learners）的组合可以形成一个强学习器（Strong Learner）。
- 集成学习通常能够提供比单一模型更好的泛化能力，减少过拟合的风险。
- 常见集成学习框架：Bagging、Boosting、Stacking等。



AdeBoosting

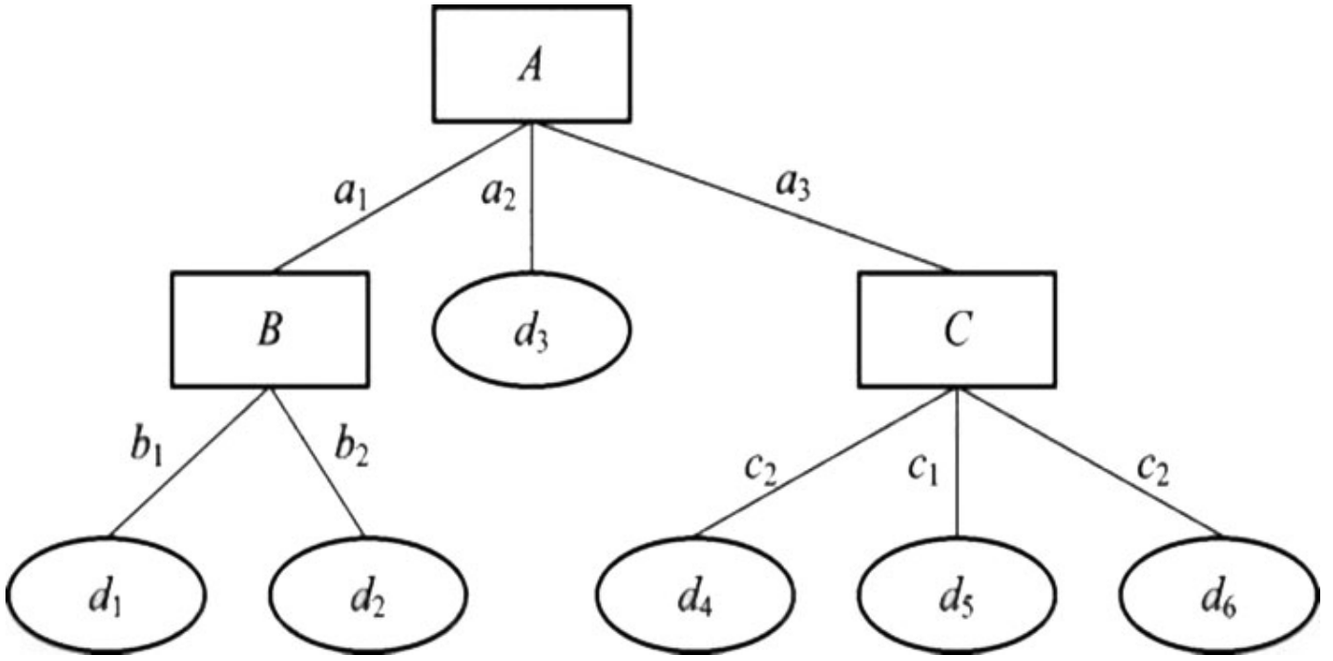
决策树示意图

结点 A, B, C 代表各个属性；

a_i, b_j, c_k 代表各属性的属性值；

叶结点 d_l 代表对应的决策结果；

内部结点则对应于一个属性测试；



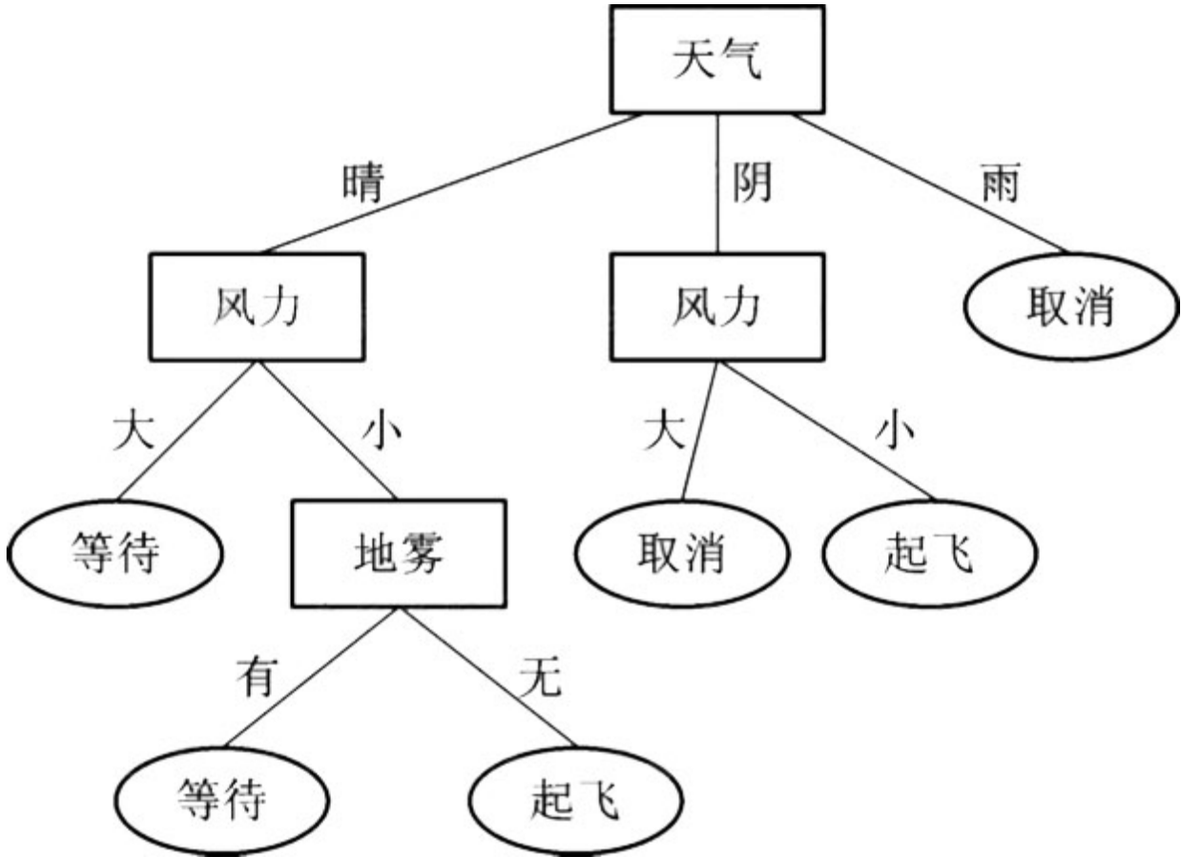
从根结点到每个叶结点的路径对应了一个判定测试序列。

决策树上从根节点到各叶子节点分枝路径上的诸“属性-值”对，和对应叶子节点的决策，构成一个产生式规则。如 A 到 d_2 的规则：$(A= a_1) \wedge (B= b_2) \Rightarrow d_2$

决策树举例

机场指挥台关于飞机起飞的简单决策树。

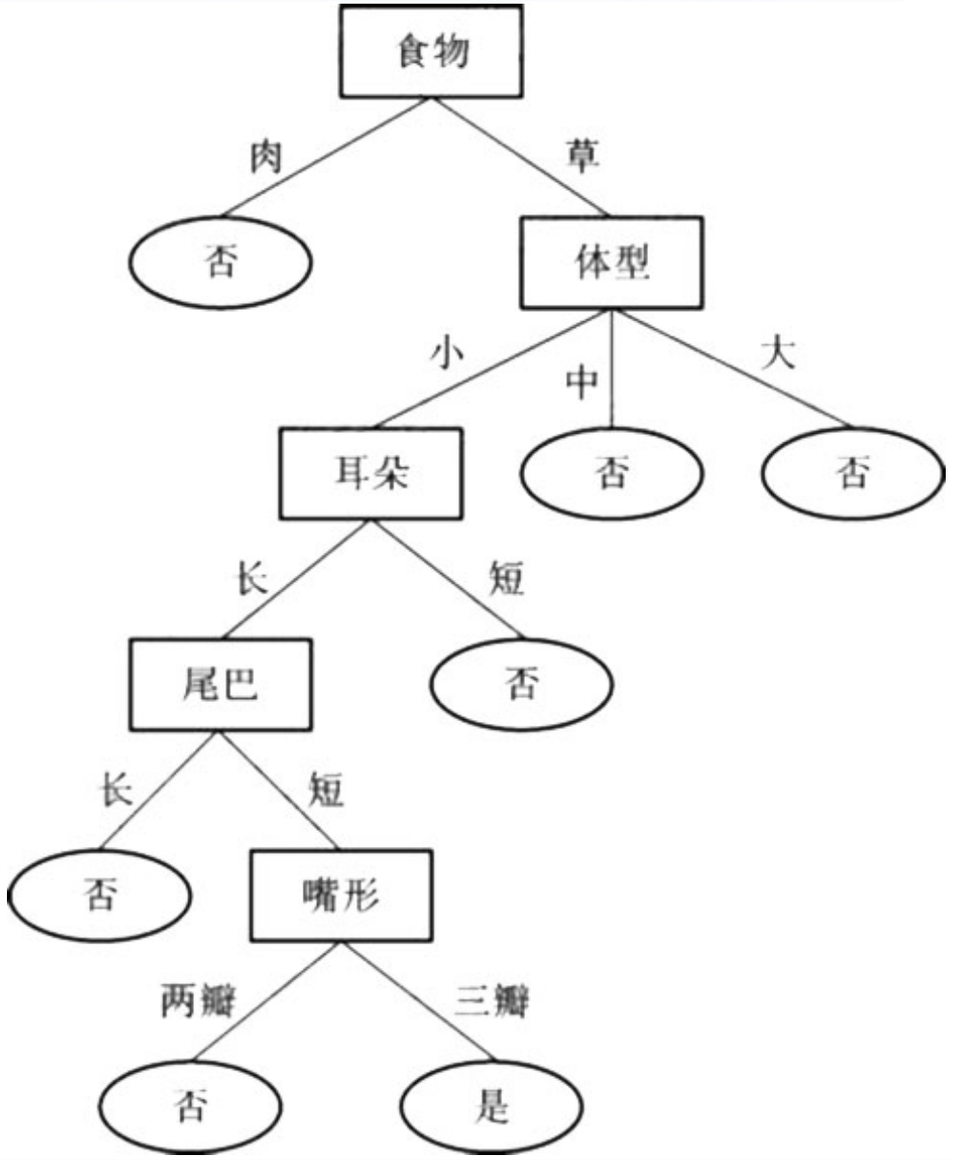
属性	属性值
天气	晴、阴、雨
风力	大、小
地雾	有、无



决策树举例2

描述“兔子”概念的决策树

属性	属性值
食物	食肉、食草
体型	大、中、小
耳朵	长、短
尾巴	长、短
嘴型	两瓣、三瓣



怎样学习决策树？

决策树学习是以实例为基础的归纳学习；

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；

其基本流程遵循简单且直观的“分而治之 (divide-and-conquer)”策略。

决策树的关键是选择最优划分属性，其基本思想：

- 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。

决策树基本流程

1、选取一个属性，以该属性为根节点，以该属性取值为根节点分枝，进行画树。

2、考察每个子类，若其中的实例结论完全相同，则以这个相同的结论作为相应分枝路径末端的叶子节点；

3、否则，选取一个非父节点的属性，按这个属性的不同取值对该子集进行分类，并以该属性作为节点，以这个属性的诸取值作为节点的分枝，继续进行画树。

4、如此继续，直到所分的子集全都满足：实例结论完全相同，而得到所有的叶子节点为止。

决策树学习举例

某保险公司的汽车驾驶保险类别划分的部分事例表。

以该表为一个实例集，用决策树学习来归纳该保险公司汽车驾驶保险类别的划分规则。

序号	实例			
	性别	年龄段	婚状	保险类别
1	女	<21	未	C
2	女	<21	已	C
3	男	<21	未	C
4	男	<21	已	B
5	女	≥21 且≤25	未	A
6	女	≥21 且≤25	已	A
7	男	≥21 且≤25	未	C
8	男	≥21 且≤25	已	B
9	女	>25	未	A
10	女	>25	已	A
11	男	>25	未	B
12	男	>25	已	B

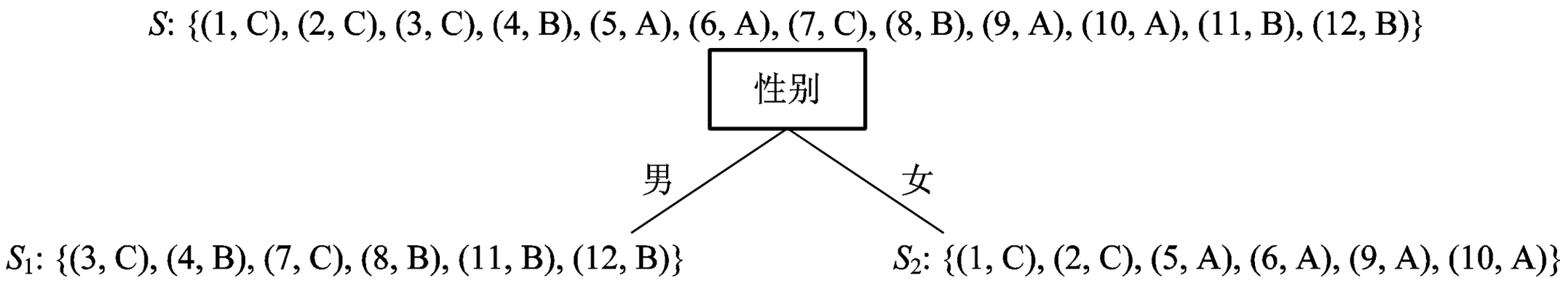
决策树学习举例2

按属性“性别”的不同取值将其分类。S应被分类为两个子集：

S_1= {(3,C), (4,B), (7,C), (8,B), (11,B), (12,B)}

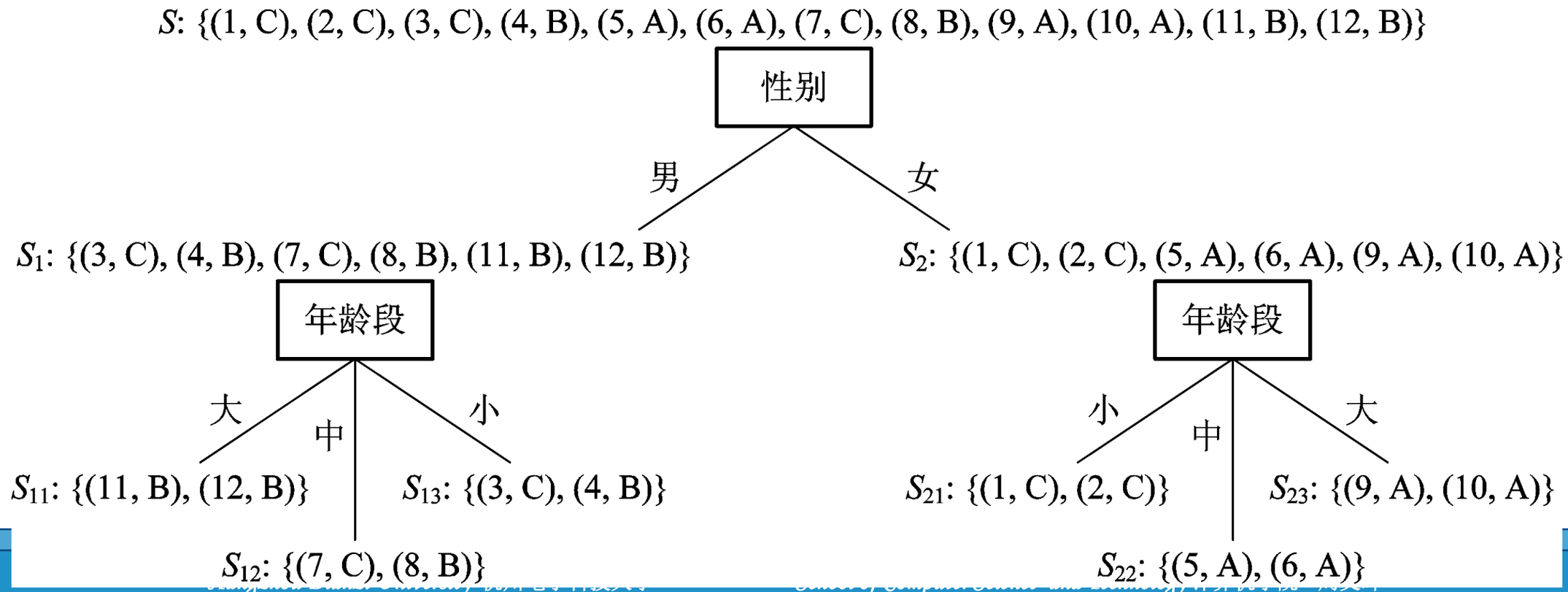
S_2= {(1,C), (2,C), (5,A), (6,A), (9, A), (10,A)}

可得到以性别作为根节点的部分决策树。



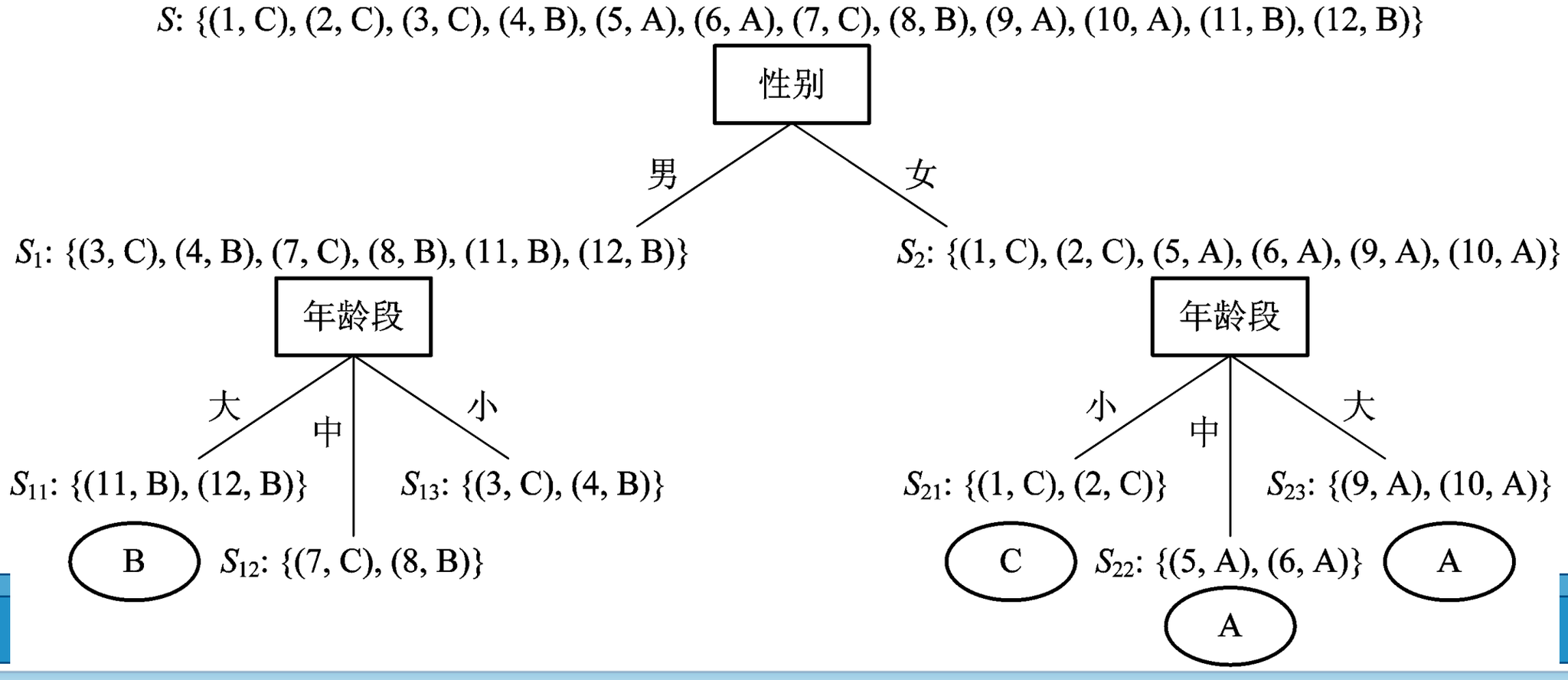
决策树学习举例3

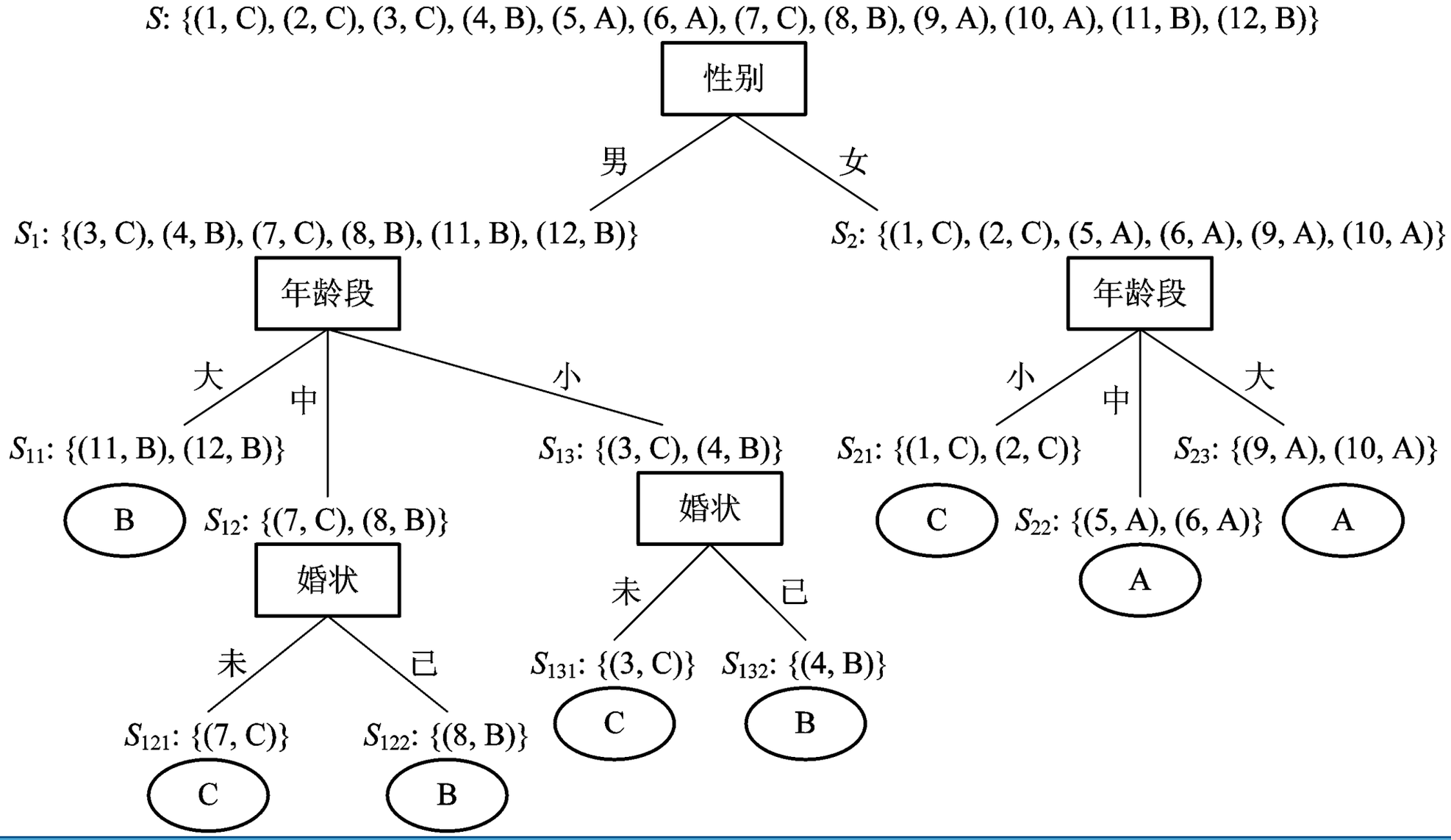
再对S_1和S_2按“年龄段”将其分类，分别得到子集S_{11}, S_{12}, S_{13}和S_{21}, S_{22}, S_{23}。于是可进一步得到含有两层节点的部分决策树。



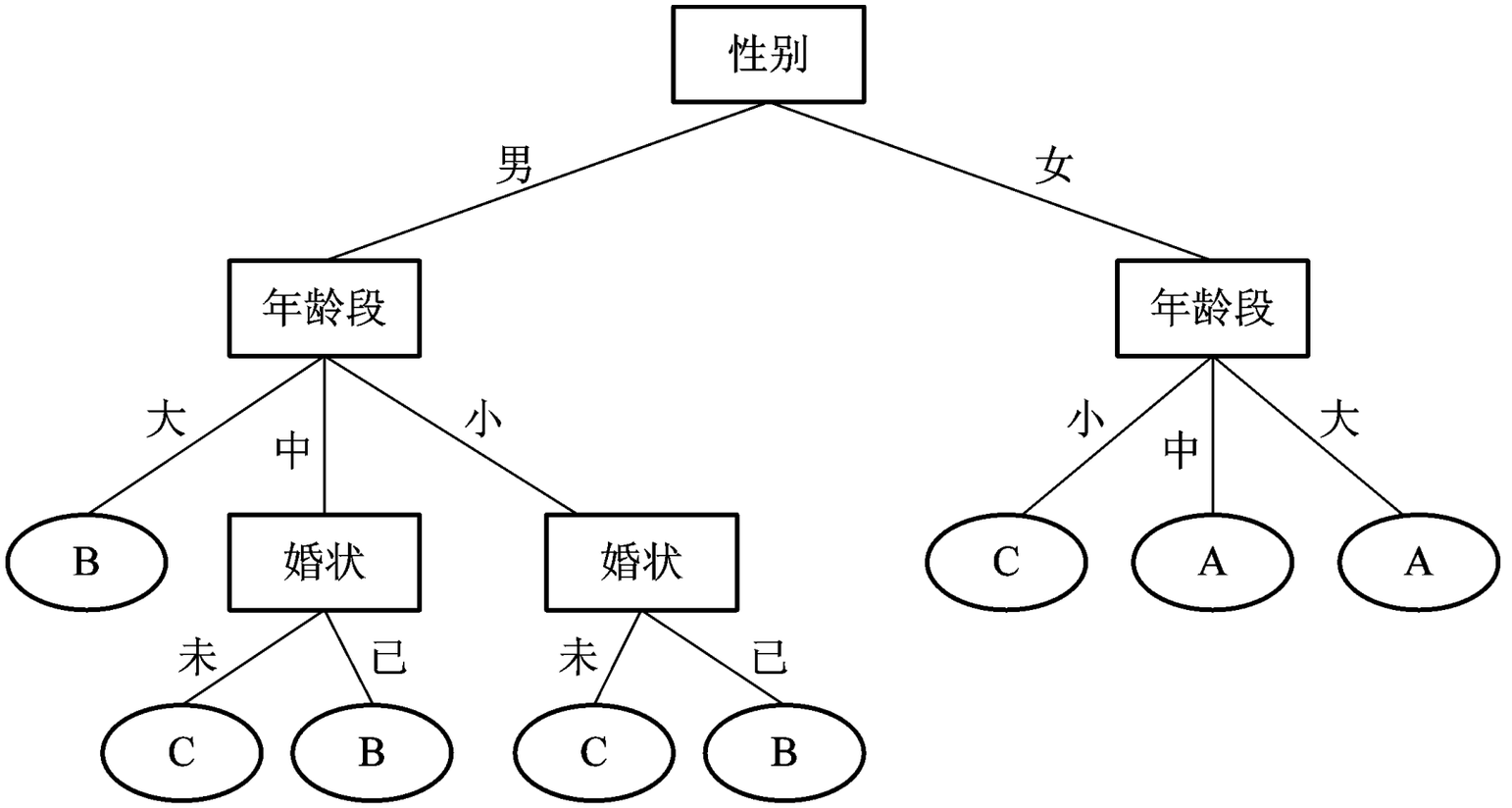
决策树学习举例4

除S_{12}和S_{13}外，其余子集中各实例的保险类别已完全相同，不需再对其进行分类，每个子集中相同保险类别值可作为相应分枝的叶子节点。

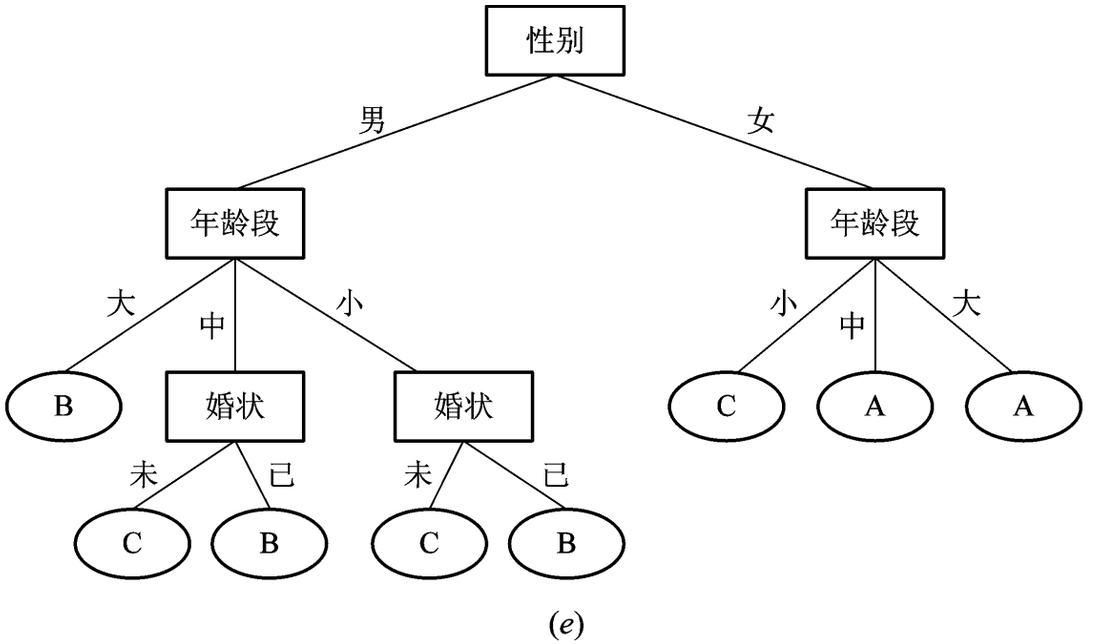



S: {(1, C), (2, C), (3, C), (4, B), (5, A), (6, A), (7, C), (8, B), (9, A), (10, A), (11, B), (12, B)}
性别
男
女
S1: {(3, C), (4, B), (7, C), (8, B), (11, B), (12, B)}
S2: {(1, C), (2, C), (5, A), (6, A), (9, A), (10, A)}
年龄段
大
中
小
S11: {(11, B), (12, B)}
B
S12: {(7, C), (8, B)}
婚状
未
已
S121: {(7, C)}
C
S122: {(8, B)}
B
S13: {(3, C), (4, B)}
婚状
未
已
S131: {(3, C)}
C
S132: {(4, B)}
B
年龄段
小
中
大
S21: {(1, C), (2, C)}
C
S22: {(5, A), (6, A)}
A
S23: {(9, A), (10, A)}
A

最终决策树生成图



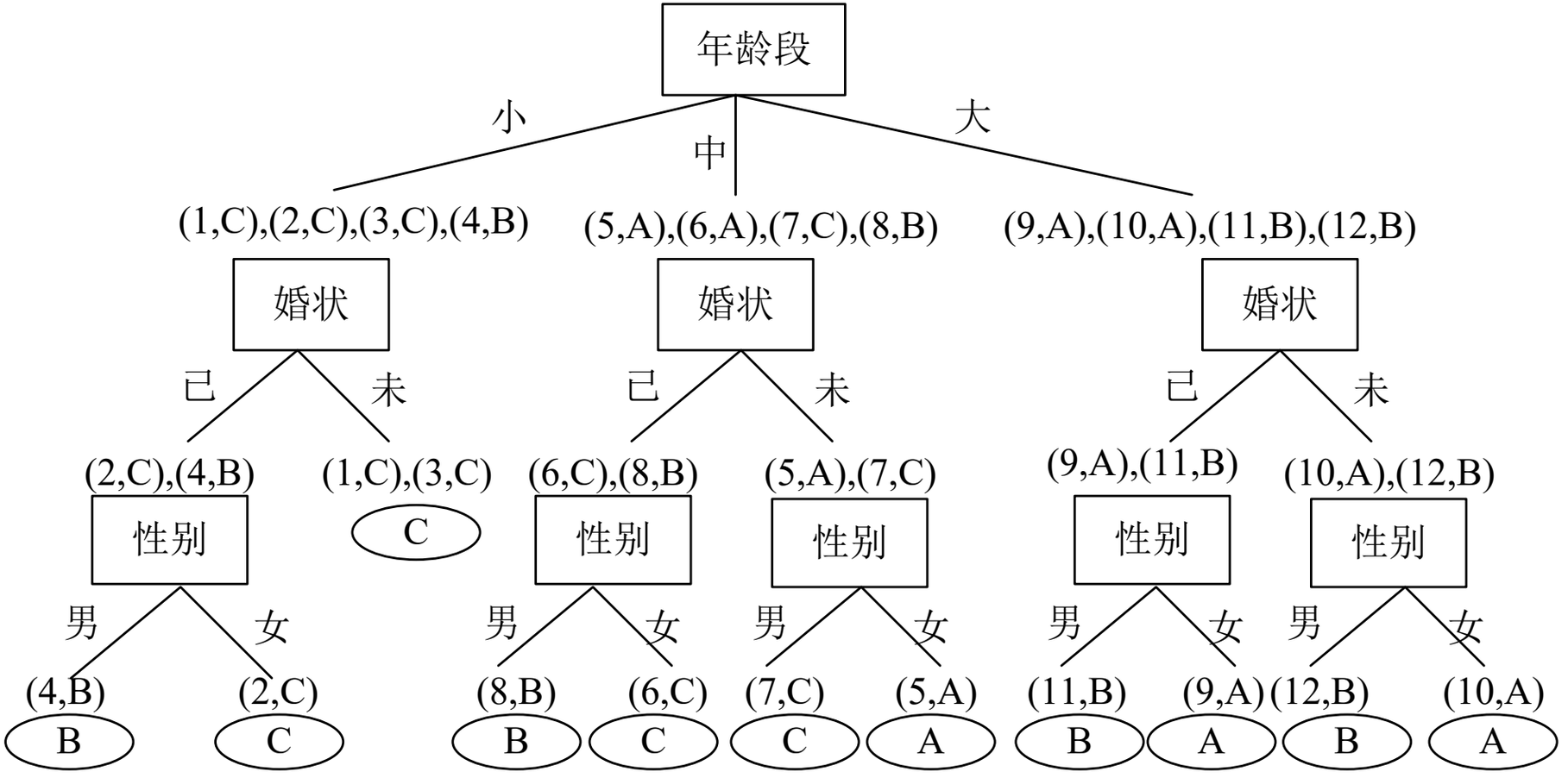
由决策树所得的规则集



(e)

- 女性且年龄在25岁以上，则给予A类保险；
- 女性且年龄在21岁到25岁之间，则给予A类保险；
- 女性且年龄在21岁以下，则给予C类保险；
- 男性且年龄在25岁以上，则给予B类保险；
- 男性且年龄在21岁到25岁之间且未婚，则给予C类保险；
- 男性且年龄在21岁到25岁之间且已婚，则给予B类保险；
- 男性且年龄在21岁以下且未婚，则给予C类保险；
- 男性且年龄在21岁以下且已婚，则给予B类保险。

若按另外属性顺序分类


年龄段
小
中
大
(1,C),(2,C),(3,C),(4,B)
(5,A),(6,A),(7,C),(8,B)
(9,A),(10,A),(11,B),(12,B)
婚状
婚状
婚状
已
未
已
未
已
未
(2,C),(4,B)
(1,C),(3,C)
(6,C),(8,B)
(5,A),(7,C)
(9,A),(11,B)
(10,A),(12,B)
C
性别
性别
性别
性别
性别
男
女
男
女
男
女
男
女
(4,B)
(2,C)
(8,B)
(6,C)
(7,C)
(5,A)
(11,B)
(9,A)
(12,B)
(10,A)
B
C
B
C
C
A
B
A
B
A

决策树基本流程

(1) 当前结点包含的样本全部属于同一类别，无需进一步划分。

(2) 当前属性集为空，或所有样本在所有属性上取值相同，无法划分。

(3) 当前结点包含的样本集合为空，不能划分。

Algorithm 1 决策树学习基本算法

输入:

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

过程: 函数 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_*; **决策树学习的关键**
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 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

- **决策数基本流程**
  决策过程，基本流程，…
- **划分选择**
  信息熵，增益率，基尼指数，…
- **剪枝处理**
  预剪枝，后剪枝 …
- **连续与缺失值**
  连续值处理，缺失值处理 …
- **小结**
  决策树与深度学习…

决策树属性的最优划分

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，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。

纯度指标用样本集中每类样本出现的概率值构造。

每类出现的概率通过训练样本集中每类样本数 N_k 除以样本总数 N 得到：

$$p_k = \frac{N_k}{N}$$

■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**”（information entropy）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 中有 $|Y|$ 类样本，第 k 类样本所占比例为 p_k，则 D 的信息熵定义为

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

$E(D)$ 值越小，则 D 的纯度越高。

▽ 计算信息熵时约定：若 $p_k = 0$，则 $p_k \log_2 p_k = 0$

▽ 当样本只属于某一类时，熵最小，最小值为 0；

▽ 当样本均匀分布于所有类时，熵最大，最大值为 $\log_2 m$

划分选择-信息增益

信息增益（information gain）：在信息论中也称为互信息，表示已知一个随机变量的信息后，使得另一个随机变量的不确定性减少的程度。

离散属性 a 有 V 个可能的取值 $\{a^1, a^2, \ldots, a^V\}$，用 a 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”：

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

这里可理解为在属性 a 的取值已知后，样本类别这个随机变量的不确定性减小的程度。

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。

划分选择-信息增益

例：数据集包含17个训练样本，类别为2，其中正例占 p_1=8/17，反例占 p_2=9/17。

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

根结点的信息熵为：

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

属性集合{色泽、根蒂、敲声、纹理、脐部、触感}，计算每个属性的信息增益。

以属性“色泽”为例，计算“色泽”的信息增益：

根据“色泽”取值{青绿、乌黑、浅白}，得到三个子集D1(色泽=青绿), D2(色泽=乌黑), D3(色泽=浅白).

划分选择-信息增益

子集D1：包含编号为{1,4,6,10,13,17}的6个样例；

正例占 p_1=3/6，反例占 p_2=3/6；

子集D2：包含编号为{2,3,7,8,9,15}的6个样例；

正例占 p_1=4/6，反例占 p_2=2/6；

子集D3：包含编号为{5,11,12,14,16}的5个样例；

正例占 p_1=1/5，反例占 p_2=4/5；

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

3个结点的信息熵为：

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

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

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

属性“色泽”的信息增益为

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

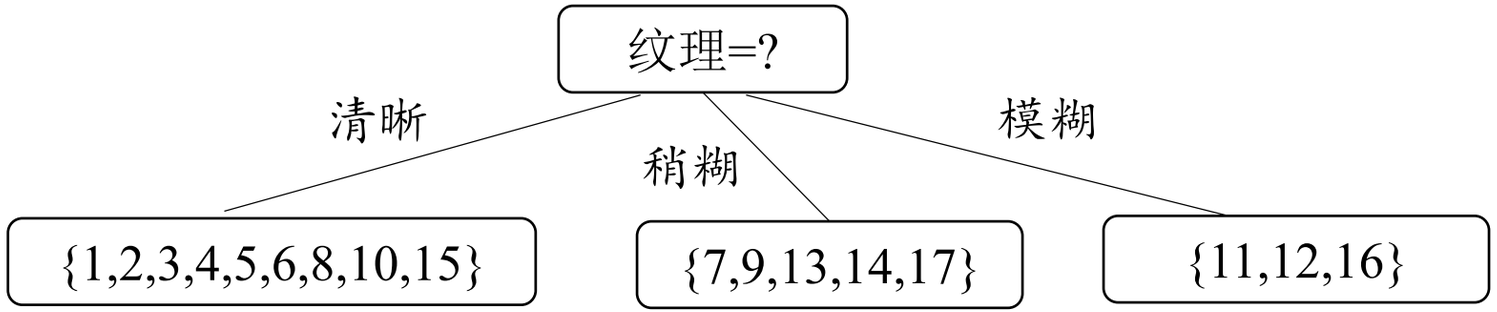
划分选择-信息增益

类似的，其他属性的信息增益为

$$\text{Gain}(D, \text{根蒂}) = 0.143 \qquad \text{Gain}(D, \text{敲声}) = 0.141 \qquad \text{Gain}(D, \text{触感}) = 0.006$$

$$\text{Gain}(D, \text{纹理}) = 0.381 \qquad \text{Gain}(D, \text{脐部}) = 0.289$$

显然，属性“纹理”的信息增益最大，其被选为划分属性



然后，决策树学习算法将对每个分支结点做进一步划分，以上图第一个分支结点（纹理=清晰）为例。

划分选择-信息增益

“纹理=清晰”分支结点包含的样例集合D1有编号为{1,2,3,4,5,6,8,10,15}的9个样例；

属性集合{色泽、根蒂、敲声、脐部、触感}，基于D1样例集合计算各属性的信息增益：

Gain(D1, 色泽)=0.043　　Gain(D1, 根蒂)=0.458　　Gain(D1, 敲声)=0.331

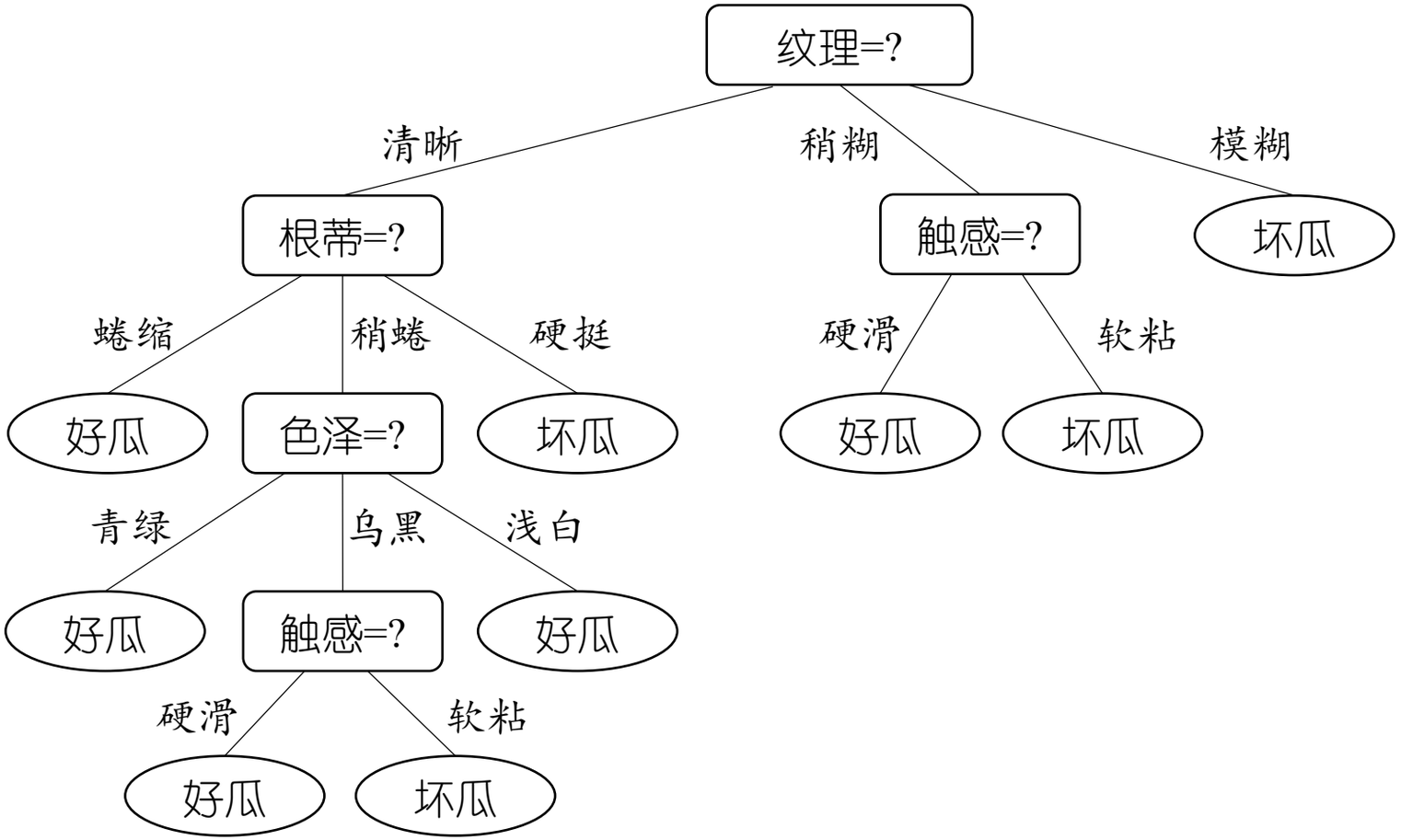
Gain(D1, 脐部)=0.458　　Gain(D1, 触感)=0.458

“根蒂”、“脐部”、“触感”三个属性均取得了最大的信息增益，可任选其一作为划分属性。

类似的，对每个分支进行上述操作，最终得到决策树。

划分选择-信息增益

最终的决策树



划分选择-信息增益的局限

信息增益公式：$Gain(D,a)=E(D)-\sum_{v=1}^{V}\frac{|D^v|}{|D|}E(D^v)$

信息增益对可取值数目较多的属性有所偏好

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

例：若把上例中的“编号”也作为一个候选划分属性，编号将产生17个分支，每个分支只包含一个样本；

导致这些分支结点的纯度已达到最大，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。

划分选择——增益率

增益率（Gain rate）：为避免信息增益偏好所带来的不利影响，采用增益率来选择最优划分属性。增益率定义为：

$$\text{Gain\_ratio}(D,a)=\frac{\text{Gain}(D,a)}{\text{IV}(a)} \qquad \text{其中} \quad \text{IV}(a)=-\sum_{v=1}^{V}\frac{|D^v|}{|D|}\log_2\frac{|D^v|}{|D|}$$

IV(a) 称为属性 a 的“固有值”[Quinlan, 1993]，与信息增益相反，属性 a 的可取值数目越多，IV(a) 的值通常就越大，导致增益率下降。

增益率准则对可取值数目较少的属性有所偏好

[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（Gini index）

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

数据集 D 的纯度可用“基尼值”来度量

$$\begin{aligned}\mathrm{Gini}(D) &= \sum_{k=1}^{|\mathcal{Y}|}\sum_{k'\neq k} p_k p_{k'} = \sum_{k=1}^{|\mathcal{Y}|} p_k \sum_{k'\neq k} p_{k'} = \sum_{k=1}^{|\mathcal{Y}|} p_k (1-p_k) = \sum_{k=1}^{|\mathcal{Y}|} p_k - p_k^2 \\ &= \sum_{k=1}^{|\mathcal{Y}|} p_k - \sum_{k=1}^{|\mathcal{Y}|} p_k^2 = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2\end{aligned}$$

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。

Gini(D) 越小，数据集 D 的纯度越高

属性 a 的基尼指数定义为：$\mathrm{Gini\_index}(D,a) = \sum_{v=1}^{V} \frac{|D^v|}{|D|}\mathrm{Gini}(D^v)$

- 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 $a_* = \underset{a\in A}{\mathrm{argmin}}\ \mathrm{Gini\_index}(D,a)$

划分选择——CART决策树

CART （ classification and regression tree）分类树是一颗二叉树，其构造方法：

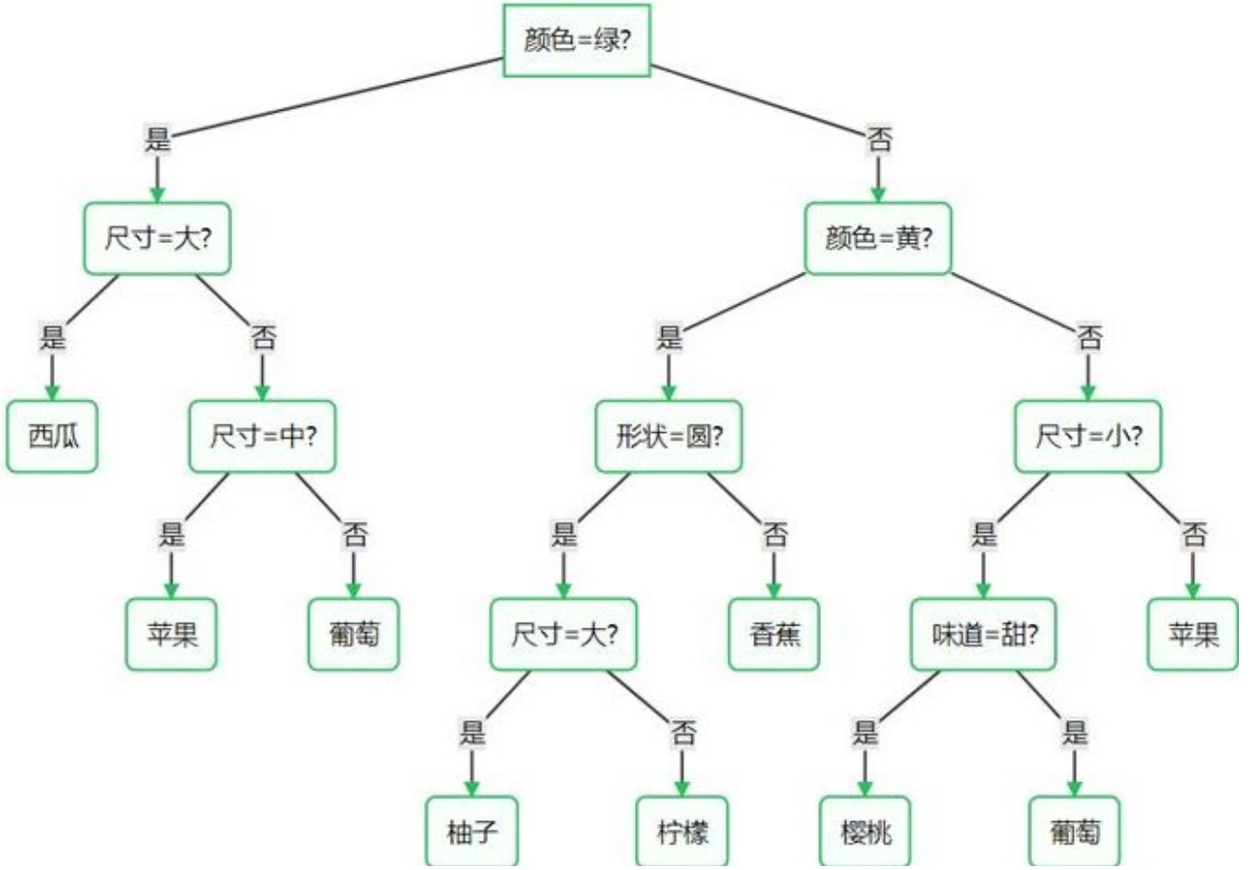
1、对每个属性a 的每个可能取值v，将数据集D 分为$a = v$ 和$a \neq v$ 两部分来计算基尼指数

$$\text{Gini\_index}(D, a)$$

$$= \frac{|D^{a=v}|}{|D|}\text{Gini}(D^{a=v}) + \frac{|D^{a\neq v}|}{|D|}\text{Gini}(D^{a\neq v})$$

2、然后，选择基尼指数最小的属性及其对应取值作为最优划分属性和最优划分点；

3、最后，重复以上两步，直至满足停止条件。



划分选择——CART决策树

以西瓜数据集为例，构造CART分类树，第一个最优划分属性和最优划分点的计算过程如下：

对于属性“色泽”的3个取值 {青绿、乌黑、浅白}，用属性值是否等于“青绿”对数据集D 进行划分，

则可得到2 个子集，分别记为 $D1$(色泽= 青绿)， $D2$(色泽≠青绿)

子集$D1$(色泽= 青绿)包含编号{1, 4, 6, 10, 13, 17}共6 个样例，正例占 $p1 = 3/6$，反例占$p2 = 3/6$；

子集$D2$ (色泽≠青绿)包含编号{2, 3, 5, 7, 8, 9, 11, 12, 14, 15, 16}共11 个样例，正例占$p1 = 5/11$，反例占$p2 = 6/11$;

用“色泽=青绿”划分之后得到基尼指数为

$$\text{Gini\_index}(D, \text{色泽} = \text{青绿}) = \frac{6}{17} \times \left(1 - (\frac{3}{6})^2 - (\frac{3}{6})^2\right) + \frac{11}{17} \times \left(1 - (\frac{5}{11})^2 - (\frac{6}{11})^2\right) = 0.497$$

属性“色泽”下其他值的基尼指数

$$\text{Gini\_index}(D, \text{色泽} = \text{乌黑}) = \frac{6}{17} \times \left(1 - (\frac{4}{6})^2 - (\frac{2}{6})^2\right) + \frac{11}{17} \times \left(1 - (\frac{4}{11})^2 - (\frac{7}{11})^2\right) = 0.456$$

$$\text{Gini\_index}(D, \text{色泽} = \text{浅白}) = \frac{5}{17} \times \left(1 - (\frac{1}{5})^2 - (\frac{4}{5})^2\right) + \frac{12}{17} \times \left(1 - (\frac{7}{12})^2 - (\frac{5}{12})^2\right) = 0.426$$

划分选择——CART决策树

类似的，可以计算出以下不同属性取不同值的基尼指数

$\text{Gini\_index}(D, \text{根蒂} = \text{蜷缩}) = 0.456$
$\text{Gini\_index}(D, \text{根蒂} = \text{稍蜷}) = 0.496$
$\text{Gini\_index}(D, \text{根蒂} = \text{硬挺}) = 0.439$
$\text{Gini\_index}(D, \text{敲声} = \text{浊响}) = 0.450$
$\text{Gini\_index}(D, \text{敲声} = \text{沉闷}) = 0.494$
$\text{Gini\_index}(D, \text{敲声} = \text{清脆}) = 0.439$

$\text{Gini\_index}(D, \text{纹理} = \text{清晰}) = 0.286$
$\text{Gini\_index}(D, \text{纹理} = \text{稍稀}) = 0.437$
$\text{Gini\_index}(D, \text{纹理} = \text{模糊}) = 0.403$
$\text{Gini\_index}(D, \text{脐部} = \text{凹陷}) = 0.415$
$\text{Gini\_index}(D, \text{脐部} = \text{稍凹}) = 0.497$
$\text{Gini\_index}(D, \text{脐部} = \text{平坦}) = 0.362$

$\text{Gini\_index}(D, \text{触感} = \text{硬挺}) = 0.494$
$\text{Gini\_index}(D, \text{触感} = \text{软粘}) = 0.494$

可知 Gini_index(D, 纹理= 清晰) = 0.286 最小，所以选择属性“纹理”为最优划分属性，并生成根结点；

接着以“纹理= 清晰”为最优划分点生成 $D1$(纹理= 清晰), $D2$(纹理 ≠清晰) 两个子结点；

对于两个子节点分别重复上述步骤继续生成下一层子节点，直至满足停止条件。

划分选择——CART决策树优缺点

优点

- **训练简单**：二叉树具有万能的表达能力，并且训练上非常简便。
- **优化简单**：二叉树作为一棵单调简单的树，在节点处的判别是一个一维优化的过程，更容易在节点处找到最优决策。
- **模型普适**：不同另外两种算法，CART既可用于分类，又能用于回归。

缺点

- **模型不稳定**：CART算法最大的缺点在于，即便是很小的样本点变动，也会导致截然不同的模型。这个缺点可以由集成学习里的随机森林方法、gradientboost等方法补足。

- **决策数基本流程**
  决策过程，基本流程，…
- **划分选择**
  信息熵，增益率，基尼指数，…
- **剪枝处理**
  预剪枝，后剪枝 …
- **连续与缺失值**
  连续值处理，缺失值处理 …
- **小结**
  决策树与深度学习…

剪枝处理

- 为什么剪枝？
  - 如果决策树结构过于复杂，可能会导致过拟合问题，“剪枝”是解决“过拟合”的主要手段；
  - 通过“剪枝”在一定程度上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。
- 剪枝的关键问题是确定减掉哪些结点以及减掉后如何结点合并，其基本策略有：
  - 预剪枝：在树的训练过程中通过停止分裂对树的规模进行限制；
  - 后剪枝：先构造出一棵完整的树，然后通过某种规则消除掉部分节点，用叶子节点替代。
- 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

将数据集划分为训练集和验证集。

根据训练集，采用信息增益准则，首先选择“脐部”作为最优划分属性，最终生成完整得决策树。

为便于讨论，将部分结点编号。

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

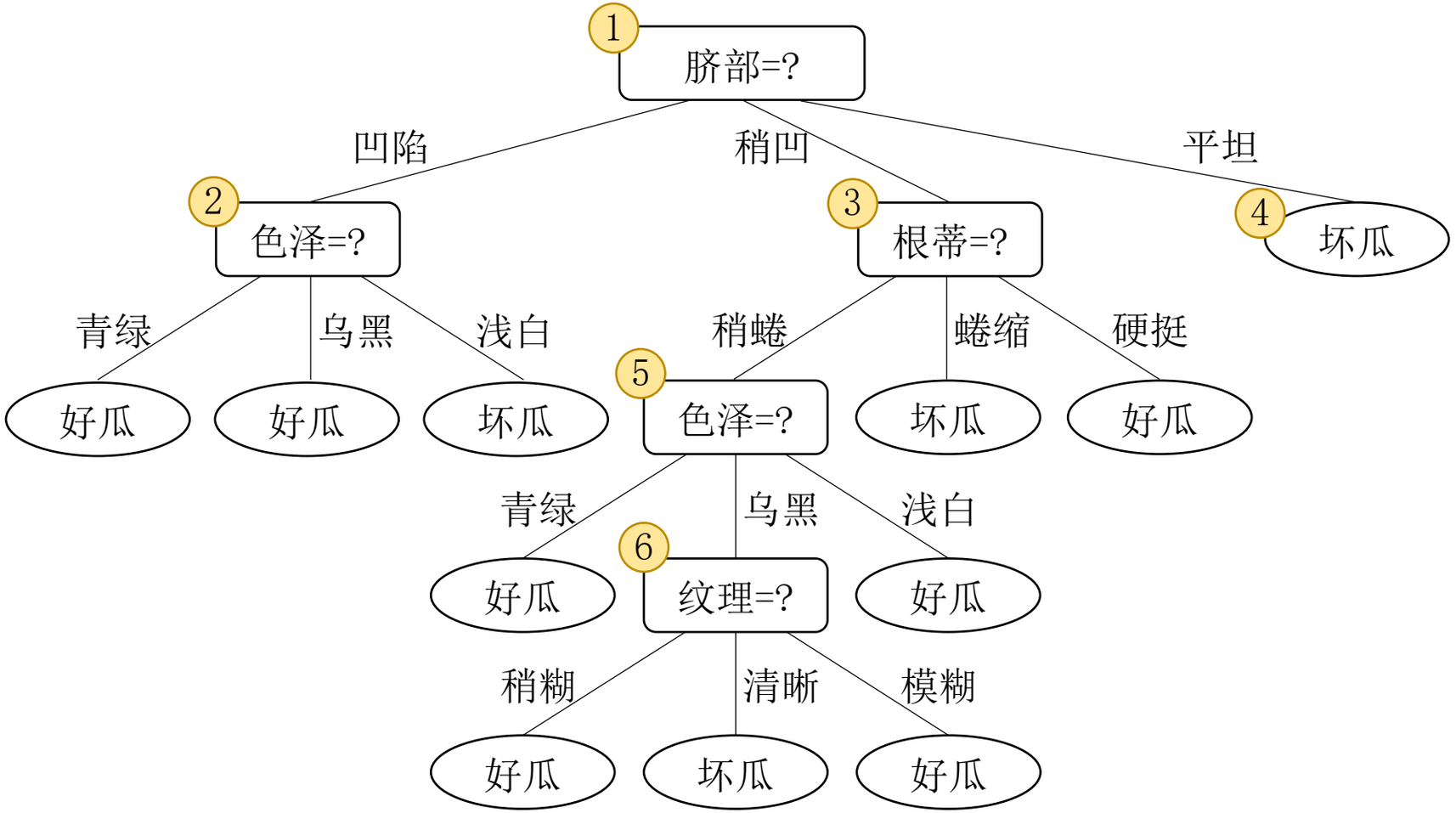
验证集

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

决策树生成过程中，若存在结点划分不能带来决策树泛化性能提升，则没必要进一步划分。



剪枝处理——预剪枝

预剪枝的思想：

决策树生成过程中，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。

例：对于结点1“脐部=？”，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。

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

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 $\frac{3}{7} \times 100\% = 42.9\%$

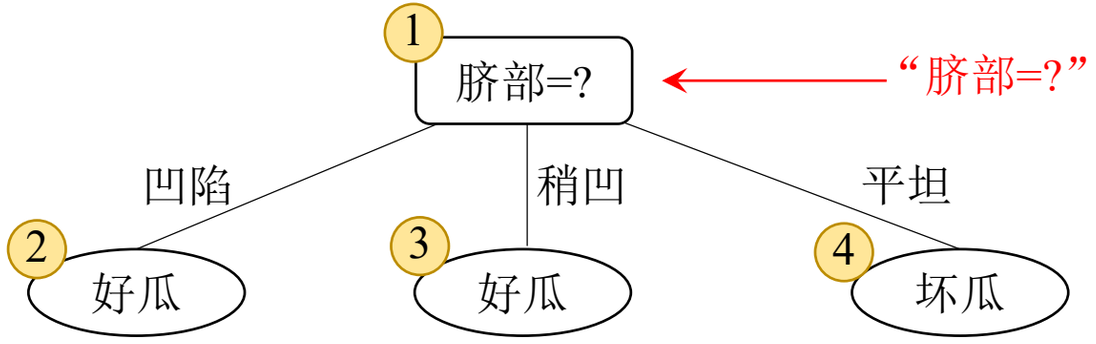
验证集精度

1 脐部=?

“脐部=?” 划分前：42.9%

剪枝处理——预剪枝

	编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
验证集	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



验证集精度　划分前: 42.9%

划分后: 71.4%

预剪枝决策: 划分

结点1：若划分，根据结点②，③，④的训练样例；

结点 ② ③ ④ 分别包含编号为{1,2,3,14}、{6,7,15,17}和{10,16}的训练样例，

根据最多类别划分准则，将这3个结点分别标记为“好瓜”、“好瓜”、“坏瓜”。

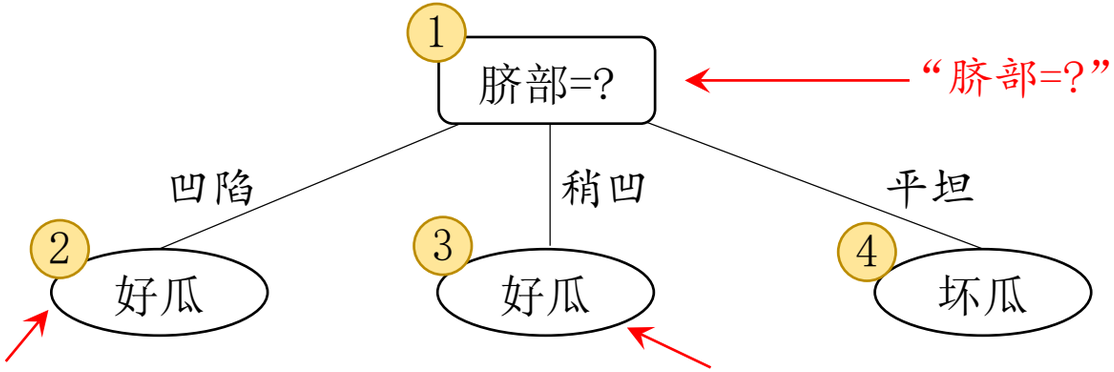
此时，验证集中编号为{4, 5, 8, 11, 12}的样例被划分正确，验证集精度为 $\frac{5}{7} \times 100\% = 71.4\%$

剪枝处理——预剪枝

	编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
验证集	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**”



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

“色泽=?” 验证集精度
划分前：71.4%
划分后：57.1%
预剪枝决策：禁止划分

“根蒂=?” 验证集精度
划分前：71.4%
划分后：71.4%
预剪枝决策：禁止划分

剪枝处理——预剪枝优缺点

优点

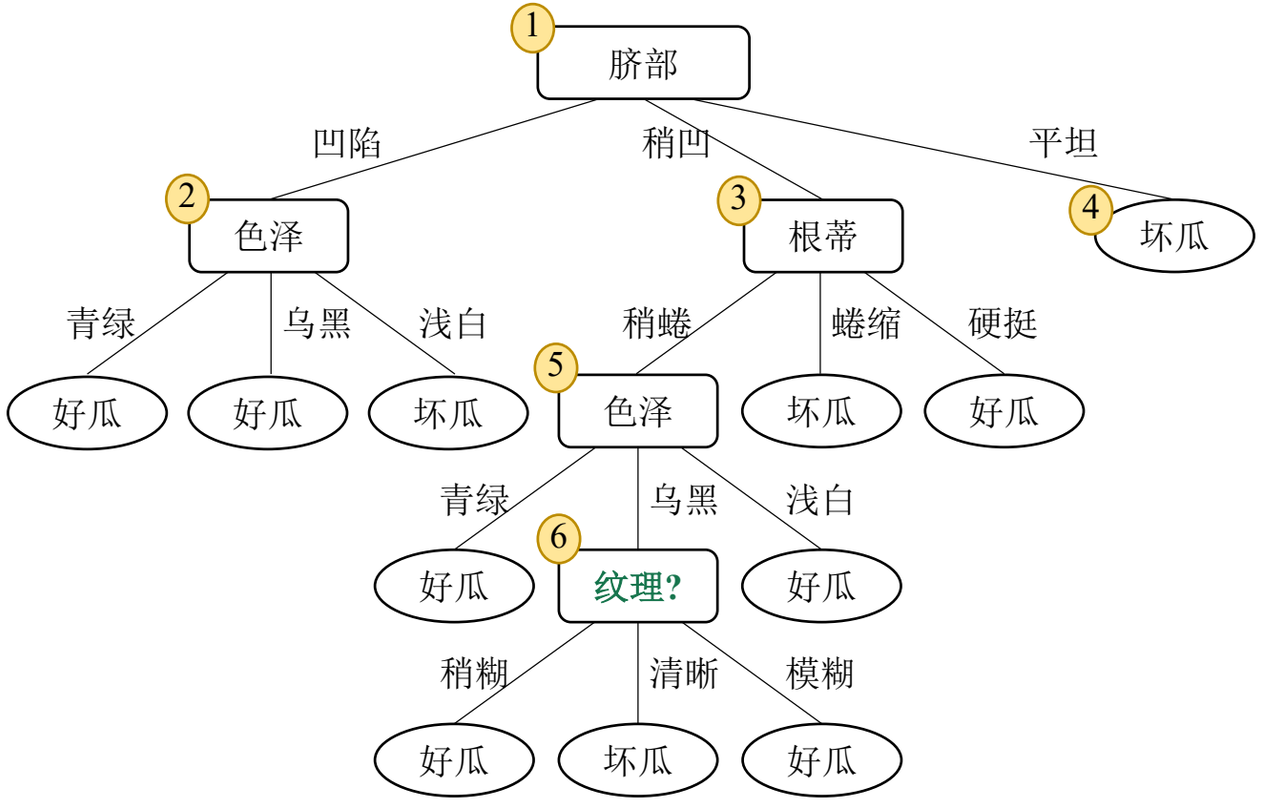
- 降低过拟合风险
- 显著减少训练时间和测试时间开销

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

后剪枝思想：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%。

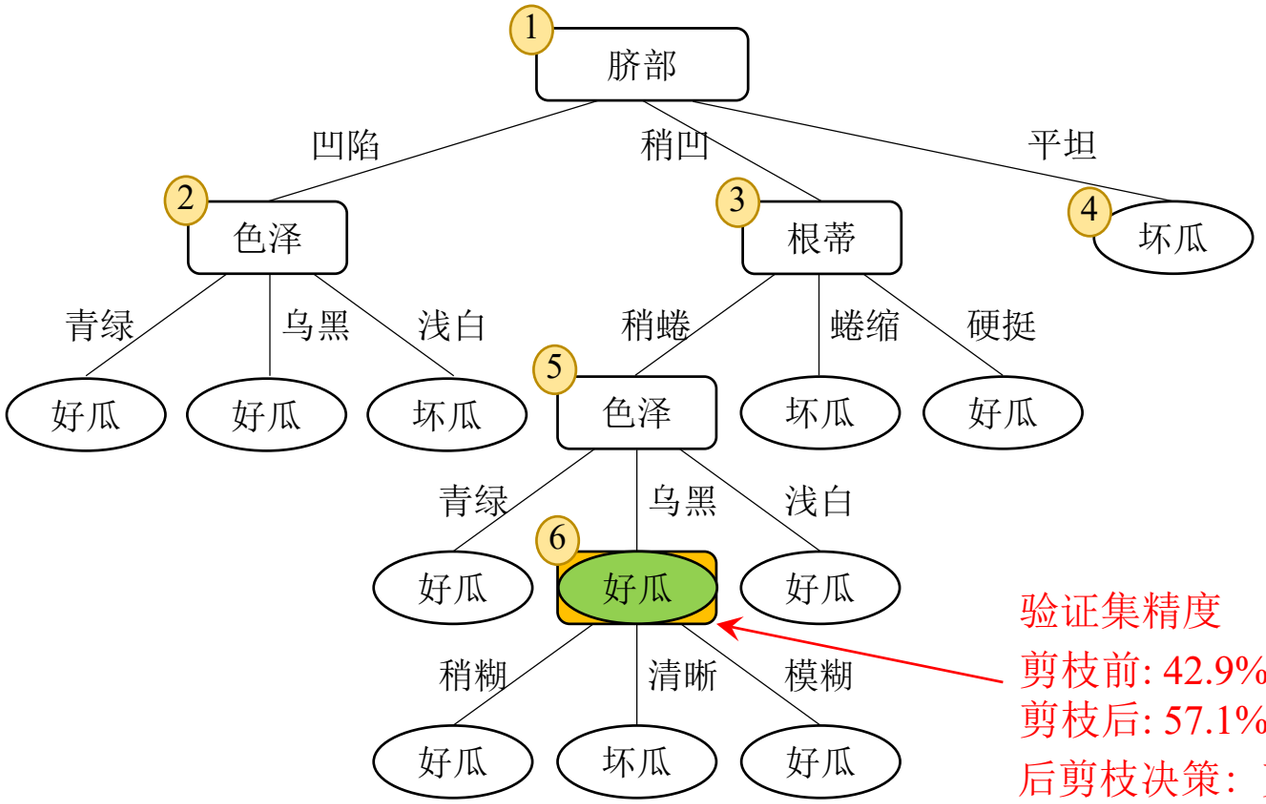
剪枝处理——后剪枝

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。



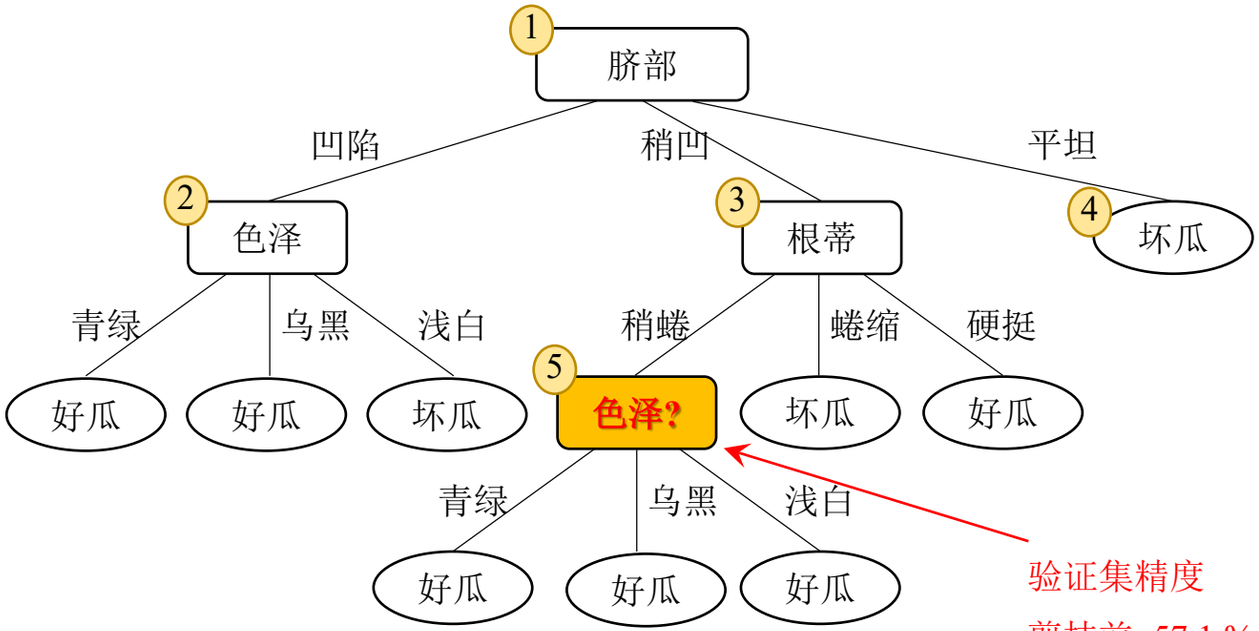
剪枝处理——后剪枝

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。



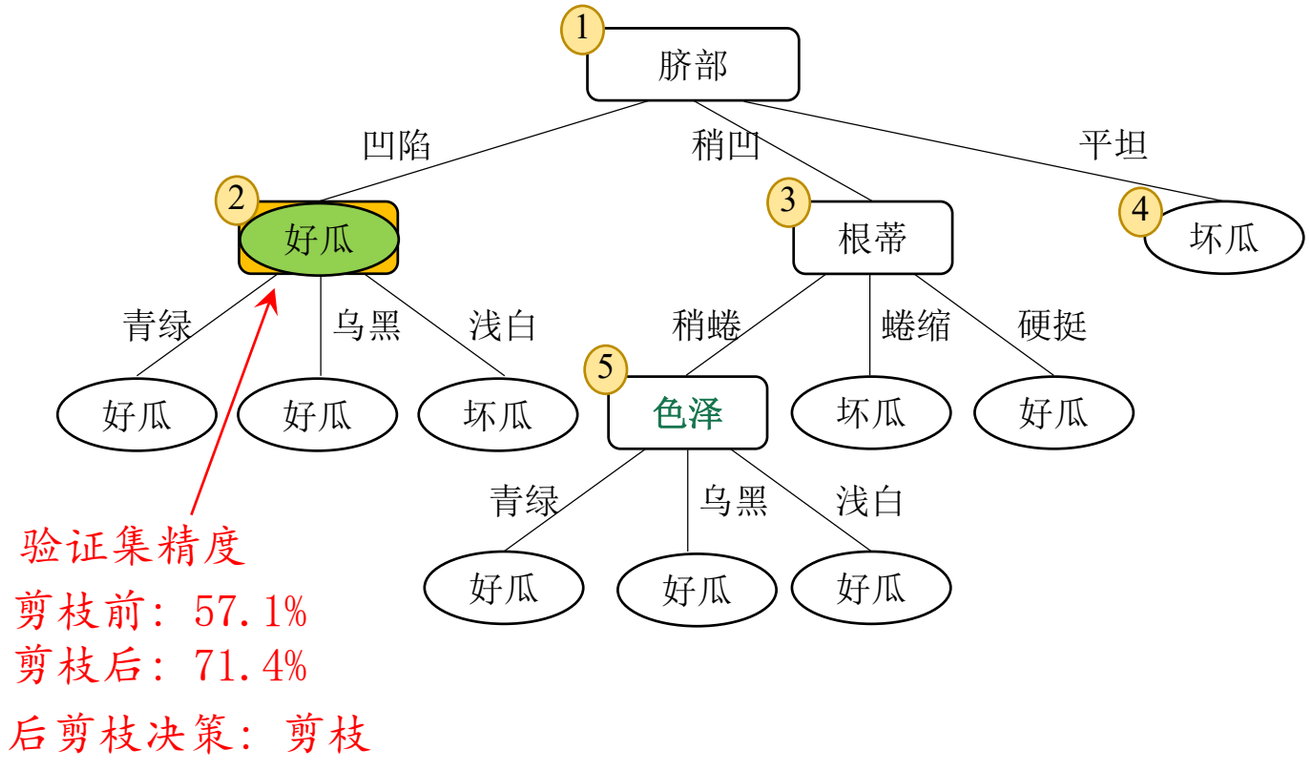
剪枝处理——后剪枝

考虑结点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



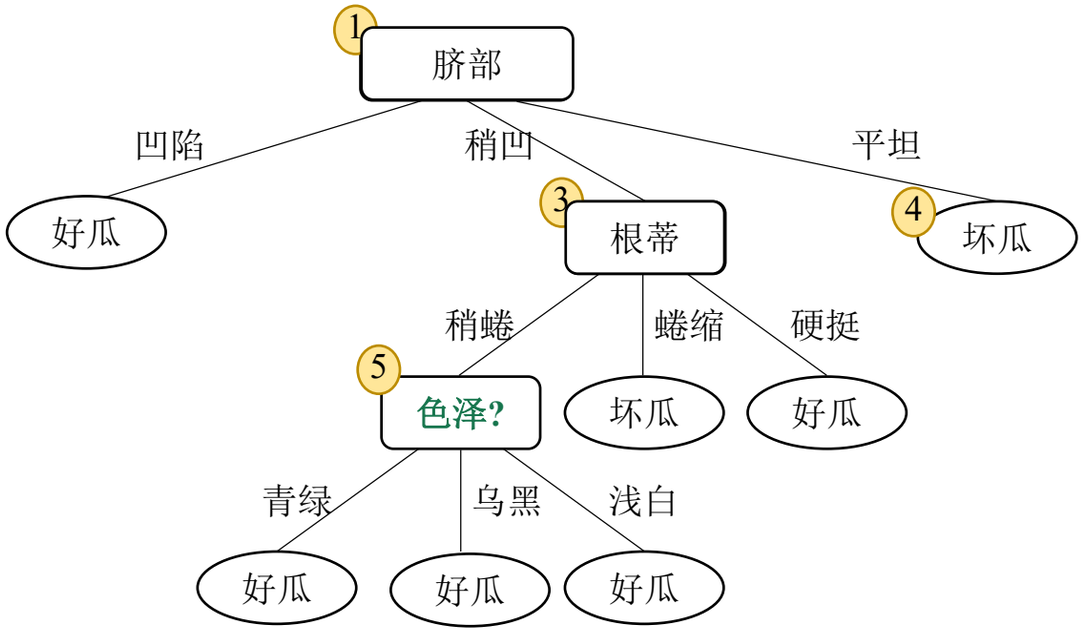
剪枝处理——后剪枝

考虑结点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

最终基于后剪枝策略得到的决策树如右图所示。



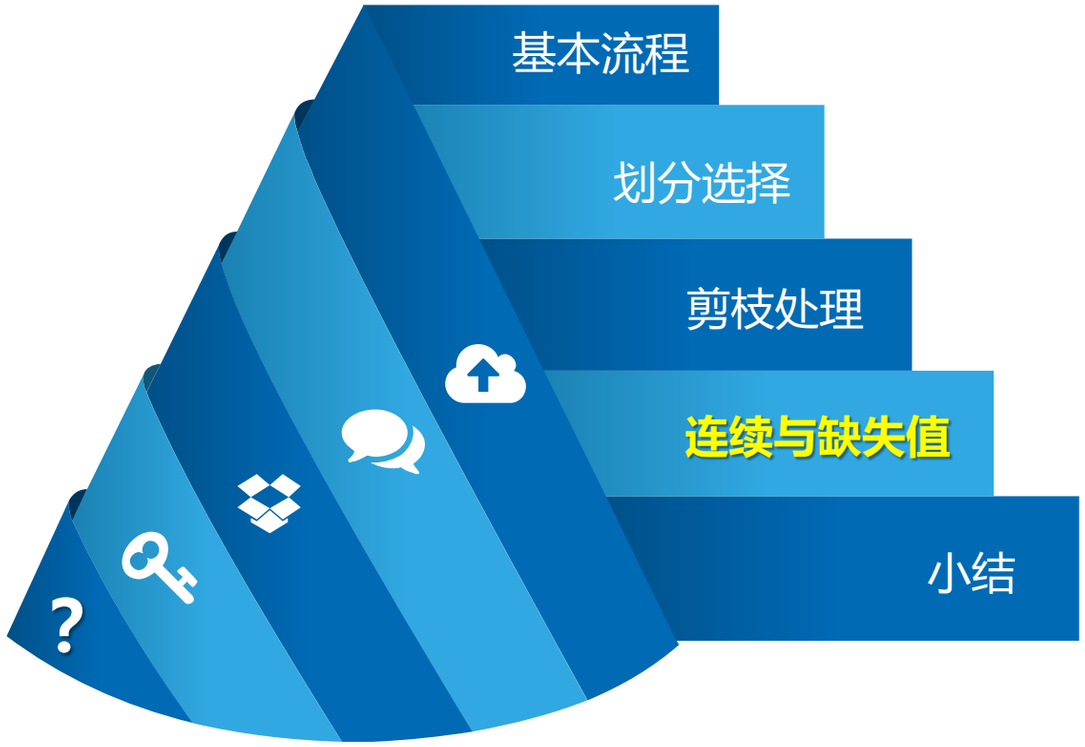
剪枝处理——后剪枝优缺点

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



- **决策数基本流程**
  决策过程，基本流程，…
- **划分选择**
  信息熵，增益率，基尼指数，…
- **剪枝处理**
  预剪枝，后剪枝 …
- **连续与缺失值**
  连续值处理，缺失值处理 …
- **小结**
  决策树与深度学习…

连续与缺失值 – 连续值处理

到目前为止，我们讨论的都是基于离散属性的决策树生成方法；

在许多学习任务中，常遇到连续属性，其可取值数目不再有限，因此不能根据连续属性的可取值对结点进行划分。

如在西瓜数据集中可以增加两项连续属性“密度”和“含糖量”

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

连续与缺失值 – 连续值处理

基本思想：对连续属性采用离散化技术，最简单的策略是二分法（bi-partition）

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 $a^1, a^2, \ldots a^n$，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

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

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

$$\begin{aligned}\mathrm{Gain}(D,a) &= \max_{t\in T_a} \mathrm{Gain}(D,a,t) \\ &= \max_{t\in T_a} \mathrm{Ent}(D) - \sum_{\lambda\in\{-,+\}} \frac{|D_t^\lambda|}{|D|}\mathrm{Ent}(D_t^\lambda)\end{aligned} \tag{4.8}$$

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

连续与缺失值 – 连续值处理

以西瓜数据库为例，对于“密度”这个连续属性

观测到的可能取值（从小到大排序）为：

{ 0.243, 0.245, 0.343, 0.360, 0.403, 0.437, 0.481, 0.556, 0.593,
0.608, 0.634, 0.639, 0.657, 0.666, 0.697, 0.719, 0.774 }

根据公式，候选划分点集合为：

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

$$T_a = \left\{ \frac{(0.243+0.245)}{2}, \frac{(0.245+0.343)}{2}, \frac{(0.343+0.360)}{2}, \frac{(0.360+0.403)}{2}, \frac{(0.403+0.437)}{2}, \frac{(0.437+0.481)}{2}, \frac{(0.481+0.556)}{2}, \frac{(0.556+0.593)}{2}, \right.$$
$$\left. \frac{(0.593+0.608)}{2}, \frac{(0.608+0.634)}{2}, \frac{(0.634+0.639)}{2}, \frac{(0.639+0.657)}{2}, \frac{(0.657+0.666)}{2}, \frac{(0.666+0.697)}{2}, \frac{(0.697+0.719)}{2}, \frac{(0.719+0.774)}{2} \right\}$$

即包含了16个候选划分点：

$T_{\text{密度}}$={ 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 }

根据公式4.8，**划分点为0.381时属性“密度”的信息增益最大**，为0.262。

连续与缺失值 – 连续值处理

类似的，对于属性“含糖率”，其候选划分点为

$T_{含糖量}$={ 0.049, 0.074, 0.095, 0.101, 0.126, 0.155, 0.179, 0.204, 0.213, 0.226, 0.250, 0.265, 0.292, 0.344, 0.373, 0.418 }

根据公式4.8，**划分点为0.126时属性“含糖率”的信息增益最大**，为0.349。

由此，各属性的信息增益为：

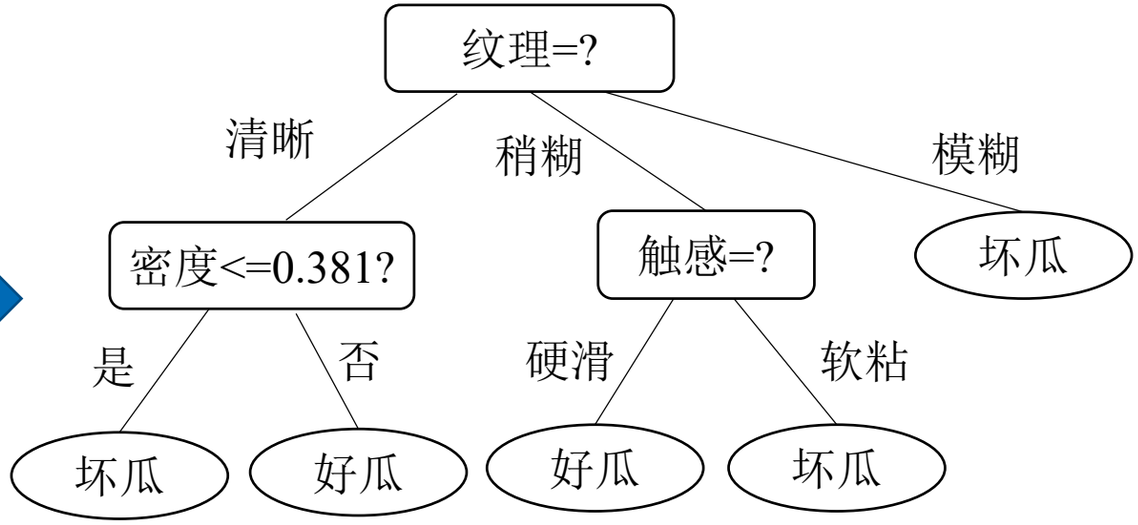
Gain(*D*, 色泽) = 0.109　　Gain(*D*, 纹理) = 0.381

Gain(*D*, 敲声) = 0.141　　Gain(*D*, 触感) = 0.006

Gain(*D*, 脐部) = 0.289　　Gain(*D*, 密度) = 0.262

Gain(*D*, 根蒂) = 0.143　　Gain(*D*, 含糖率) = 0.349

生成决策树



连续与缺失值 – 缺失值处理

现实任务中会遇到不完整样本，即样本的某些属性值缺失，如右图西瓜数据集。

如果简单放弃不完整样本，仅使用无缺失值的样本来学习，则是对数据信息的极大浪费。

使用有缺失值的样本，需要解决哪些问题？

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

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

针对Q1问题，解决思路是：（1）提取属性 a 上没有缺失的样本子集，计算信息增益

（2）最终信息增益根据缺失样本数比例进行加权。

$\tilde{D}$ 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 D 中属于第 k 类的样本子集

为每个样本 $\boldsymbol{x}$ 赋予一个权重 w_x ，并定义：

- 无缺失值样本所占比例 $\rho = \dfrac{\sum_{x \in \tilde{D}} w_x}{\sum_{x \in D} w_x}$
- 无缺失值样本中第 k 类所占比例 $\tilde{p}_k = \dfrac{\sum_{x \in \tilde{D}_k} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq k \leq |\mathcal{Y}|)$
- 无缺失值样本中在属性a上取值 a^v 的样本所占比例 $\tilde{r}_v = \dfrac{\sum_{x \in \tilde{D}^v} w_x}{\sum_{x \in \tilde{D}} w_x} \quad (1 \leq v \leq V)$

连续与缺失值 – 缺失值处理

基于上述定义，有

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

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

针对Q2问题，解决思路：

- 若样本 $\boldsymbol{x}$ 在划分属性 a 上的取值已知，则将 $\boldsymbol{x}$ 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 $\boldsymbol{w_x}$
- 若样本 $\boldsymbol{x}$ 在划分属性 a 上的取值未知，则将 $\boldsymbol{x}$ 同时划入所有的子结点，且样本权值在与属性值 a^v 对应的子结点中调整为 $\tilde{r}_v \cdot w_x$ (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 – 缺失值处理举例

- 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：

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

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

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

连续与缺失值 – 缺失值处理举例

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

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

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

因此，样本子集 $\tilde{D}$ 上属性“色泽”的信息增益为

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

- 于是，样本集 D 上属性“色泽”的信息增益为

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

连续与缺失值 – 缺失值处理举例

类似地可计算出所有属性在数据集上的信息增益

$$\text{Gain}(D, \text{色泽}) = 0.252$$
$$\text{Gain}(D, \text{敲声}) = 0.145$$
$$\text{Gain}(D, \text{根蒂}) = 0.171$$
$$\text{Gain}(D, \text{纹理}) = 0.424$$
$$\text{Gain}(D, \text{脐部}) = 0.289$$
$$\text{Gain}(D, \text{触感}) = 0.006$$

进入“纹理=清晰”分支

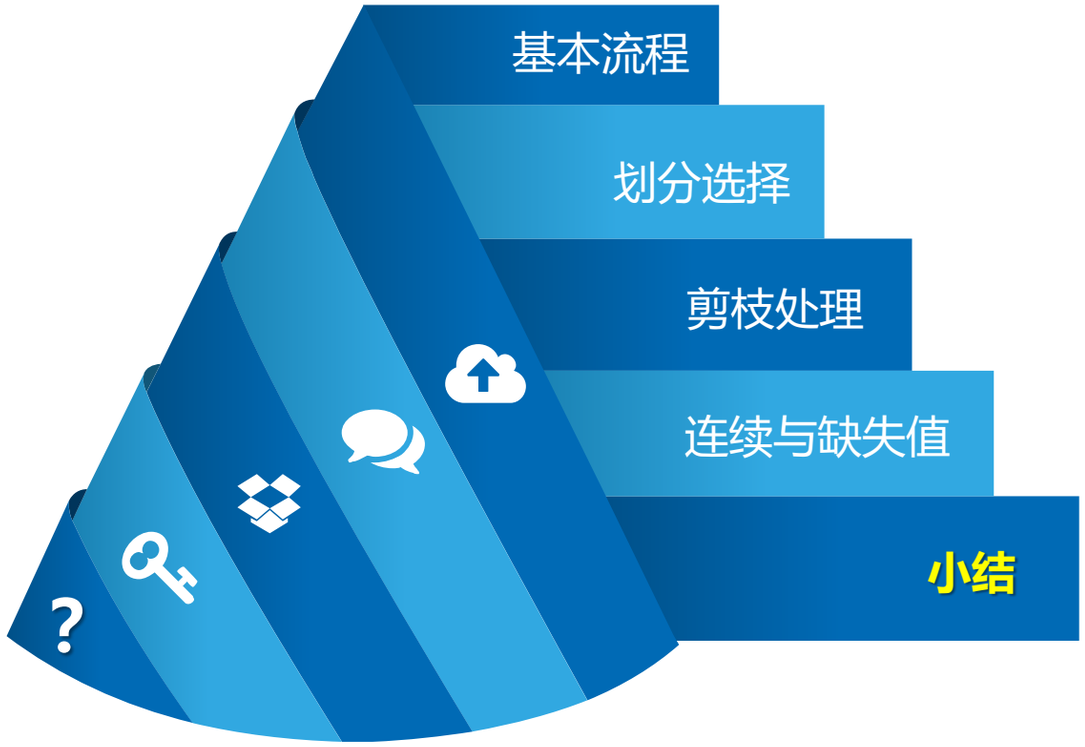
进入“纹理=稍糊”分支

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个分支权值分别为 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



- **决策数基本流程**
  决策过程，基本流程，…
- **划分选择**
  信息熵，增益率，基尼指数，…
- **剪枝处理**
  预剪枝，后剪枝 …
- **连续与缺失值**
  连续值处理，缺失值处理 …
- **小结**
  决策树与深度学习…

深度学习 v.s. 决策树

决策树是一种用于分类的经典机器学习方法，它易于理解且可解释性强，能够在中等规模数据上以低难度获得较好的模型。

但如果决策树遇上 ImageNet 这一级别的数据，其性能还是远远比不上神经网络。

深度学习在性能上优于决策树，但模型的可解释性一直是其痛点。

「准确率」和「可解释性」，「鱼」与「熊掌」要如何兼得？把二者结合会怎样？

深度学习 v.s. 决策树

神经支持决策树「Neural-backed decision trees」，在 ImageNet 上取得了 75.30% 的 top-1 分类准确率，在保留决策树可解释性的同时取得了当前神经网络才能达到的准确率，比其他基于决策树的图像分类方法高出了大约 14%。

•BAIR 博客地址：
https://bair.berkeley.edu/blog/2020/04/23/decisions/
•论文地址：https://arxiv.org/abs/2004.00221
•开源项目地址：https://github.com/alvinwan/neural-backed-decision-trees

NBDT: Neural-Backed Decision Trees

Alvin Wan[1], Lisa Dunlap[1⋆], Daniel Ho[1⋆], Jihan Yin[1], Scott Lee[1],
Henry Jin[1], Suzanne Petryk[1], Sarah Adel Bargal[2], Joseph E. Gonzalez[1]

UC Berkeley[1], Boston University[2]
{alvinwan, ldunlap, danielho, jihan_yin, scott.lee.3898,
henlinjin, spetryk, jegonzal}@berkeley.edu
sbargal@bu.edu

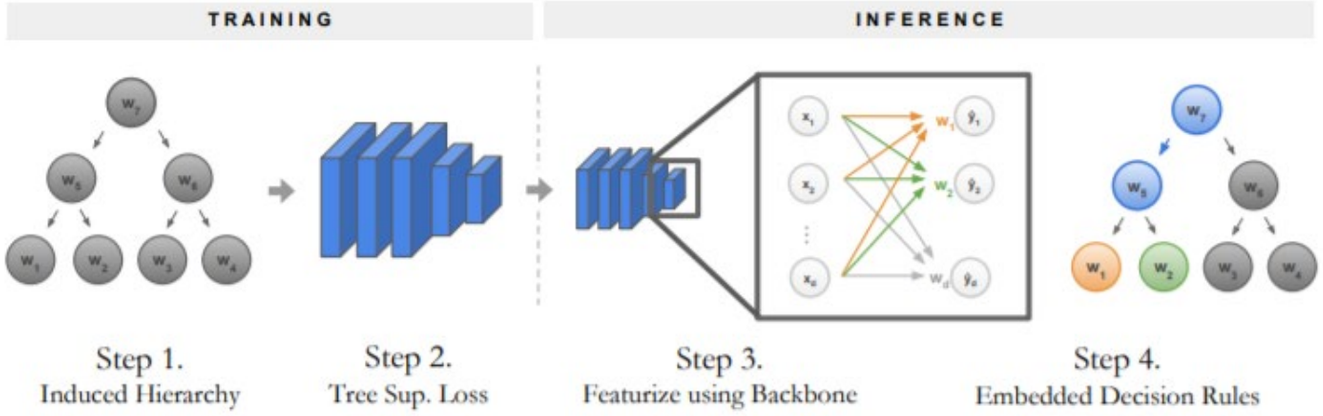


Fig. 1: Neural-Backed Decision Tree. In **step 1**, we use a pre-trained network's fully-connected layer weights to build a hierarchy (Sec 3.2). In **step 2**, we fine-tune the network with a custom loss (Sec. 3.3). In **step 3**, we featurize the sample using the neural network backbone. In **step 4**, we use the fully-connected layer's weights to run decision rules (Sec. 3.1). Illustrated above, the orange arrows in Step 3 are associated with tree's orange node in Step 4. Likewise, the green arrows map to the green node. The tree takes an inner product between the incoming sample and each of the orange w_1 and green w_2 vectors; the leaf with the higher inner product is predicted.

Any Questions?