南昌大学
NANCHANG UNIVERSITY

统计机器学习

主讲人：彭振华

数学与计算机学院

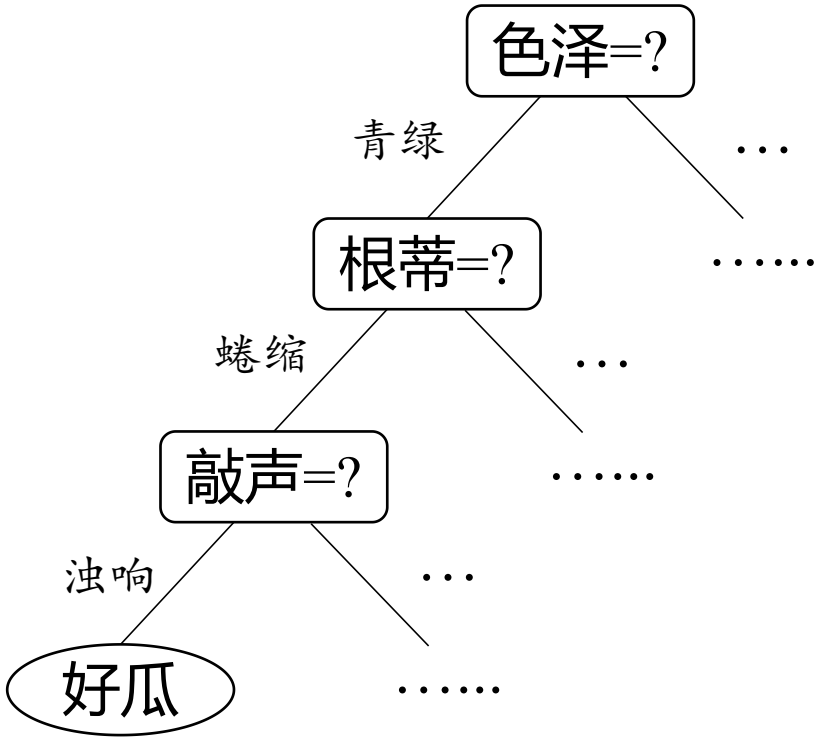
2026年

目录

CONTENTS

决　　策　　树

- 决策树基于树结构来进行预测



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

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

假定公司收集了右表数据，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？

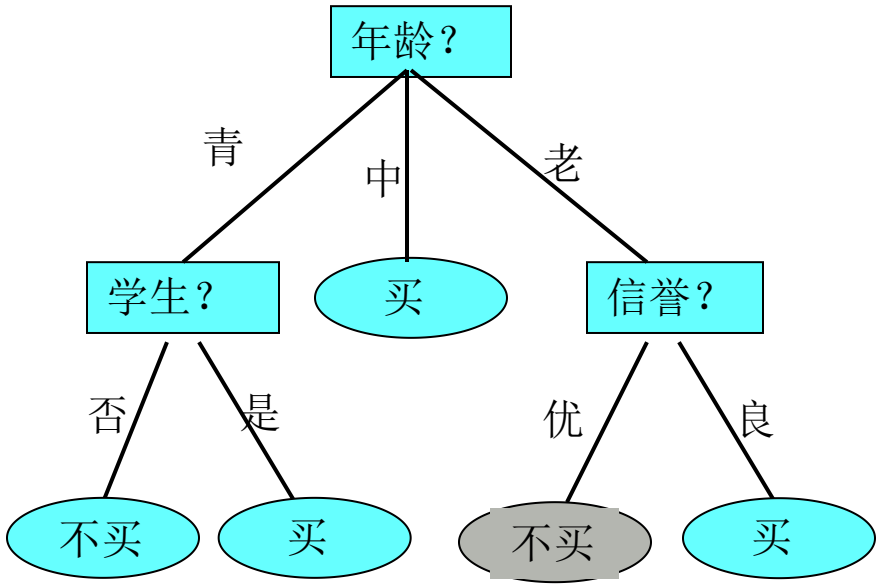
即：你能预测这位客人是属于“买”计算机的那一类，还是属于“不买”计算机的那一类？

又：你需要多少有关这位客人的信息才能回答这个问题？

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

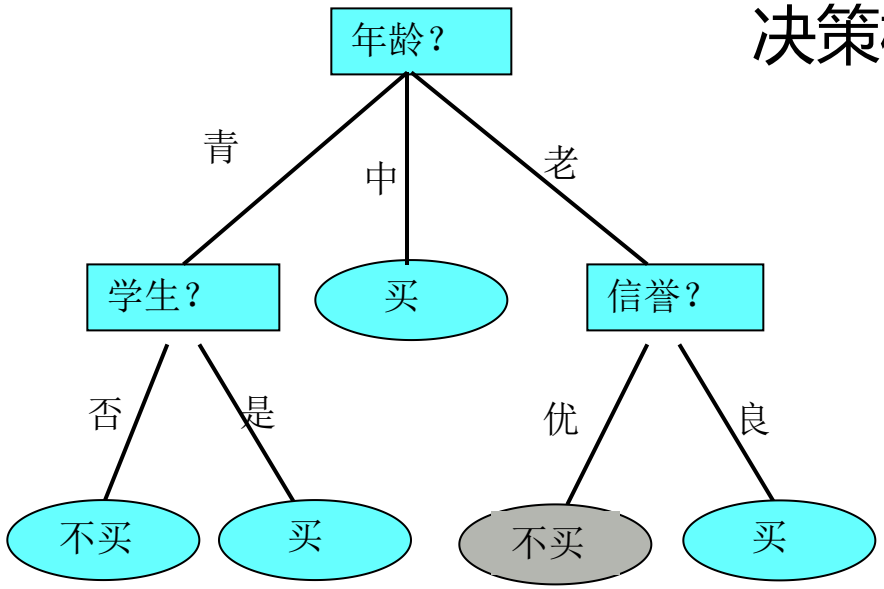
决　　策　　树

谁在买计算机？



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

决　　策　　树



决策树的基本组成部分：决策结点、分支和叶子。

决策树中最上面的结点称为根结点。是整个决策树的开始。每个分支是一个新的决策结点，或者是树的叶子。每个决策结点代表一个问题或者决策.通常对应待分类对象的属性。每个叶结点代表一种可能的分类结果

在沿着决策树从上到下的遍历过程中，在每个结点都有一个测试。对每个结点上问题的不同测试输出导致不同的分枝，最后会达到一个叶子结点。这一过程就是利用决策树进行分类的过程，利用若干个变量来判断属性的类别

决　　策　　树

- CLS（Concept Learning System）算法
  - CLS算法是早期的决策树学习算法。它是许多决策树学习算法的基础
- CLS基本思想
  - 从一棵空决策树开始，选择某一属性（分类属性）作为测试属性。该测试属性对应决策树中的决策结点。根据该属性的值的不同，可将训练样本分成相应的子集：
    - 如果该子集为空，或该子集中的样本属于同一个类，则该子集为叶结点，
    - 否则该子集对应于决策树的内部结点，即测试结点，需要选择一个新的分类属性对该子集进行划分，直到所有的子集都为空或者属于同一类。

决　策　树

- **步骤：**
  - 生成一颗空决策树和一张训练样本属性集;
  - 若训练样本集T 中所有的样本都属于同一类,则生成结点T ，并终止学习算法;否则
  - 根据某种策略从训练样本属性表中选择属性A 作为测试属性， 生成测试结点A
  - 若A的取值为v1,v2,…,vm, 则根据A 的取值的不同,将T 划分成 m个子集T1,T2,…,Tm;
  - 从训练样本属性表中删除属性A;
  - 转步骤2, 对每个子集递归调用CLS;

CLS算法问题：

- 在步骤3中，根据某种策略从训练样本属性表中选择属性A作为测试属性。没有规定采用何种测试属性。实践表明，测试属性集的组成以及测试属性的先后对决策树的学习具有举足轻重的影响。

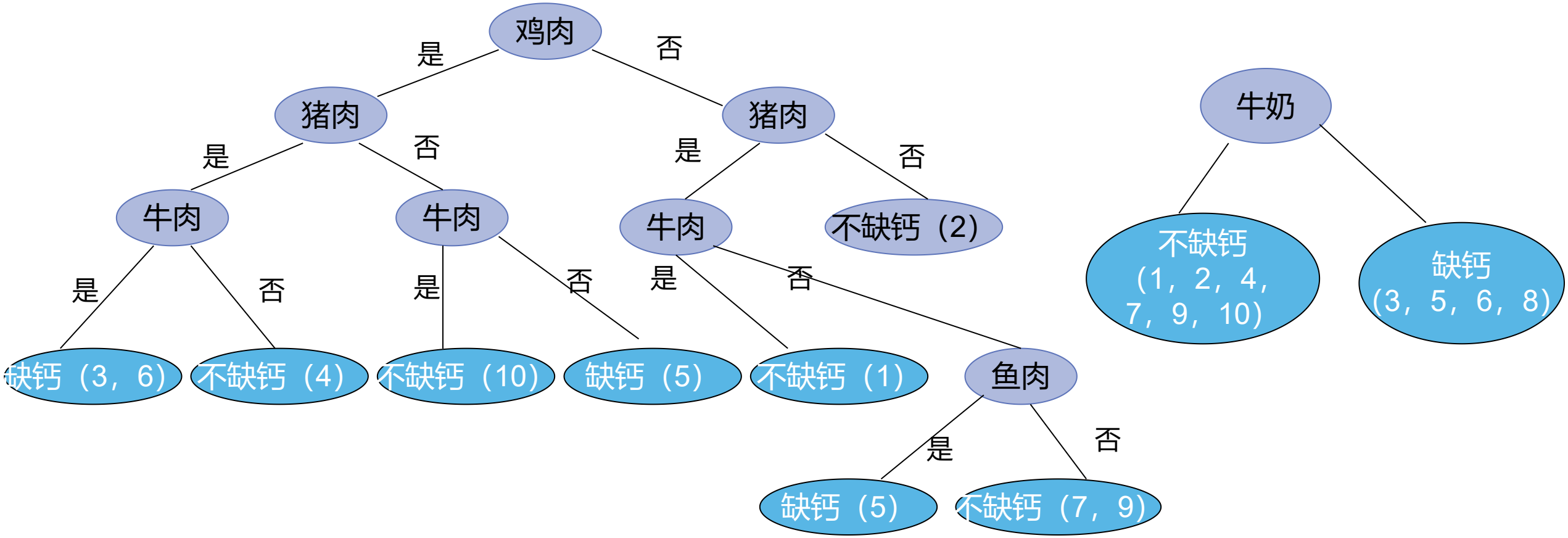
决　　策　　树

学生膳食结构和缺钙调查表

学生	鸡肉	猪肉	牛肉	羊肉	鱼肉	鸡蛋	青菜	番茄	牛奶	健康情况
1	0	1	1	0	1	0	1	0	1	不缺钙
2	0	0	0	0	1	1	1	1	1	不缺钙
3	1	1	1	1	1	0	1	0	0	缺钙
4	1	1	0	0	1	1	0	0	1	不缺钙
5	1	0	0	1	1	1	0	0	0	缺钙
6	1	1	1	0	0	1	0	1	0	缺钙
7	0	1	0	0	0	1	1	1	1	不缺钙
8	0	1	0	0	0	1	1	1	1	缺钙
9	0	1	0	0	0	1	1	1	1	不缺钙
10	1	0	1	1	1	1	0	1	1	不缺钙

决　　策　　树

采用不同的测试属性及其先后顺序将会生成不同的决策树



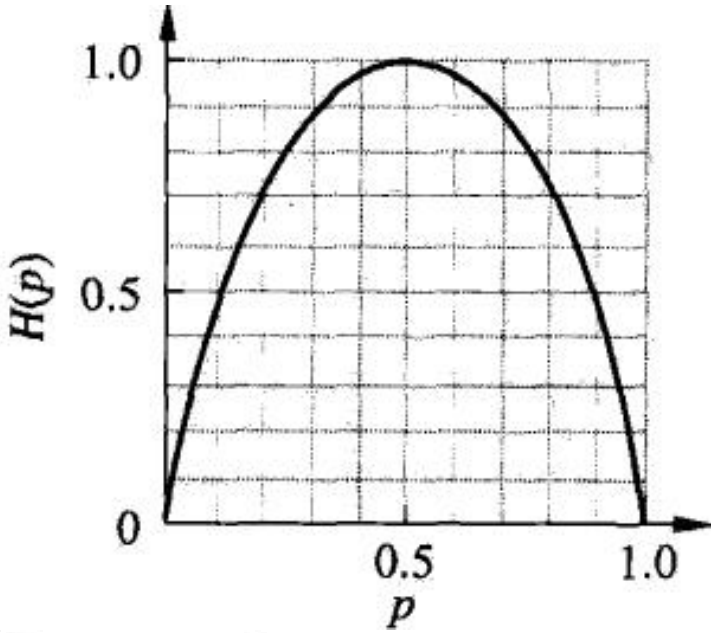
决　　策　　树

- ID3算法是一种经典的决策树学习算法，由Quinlan于1979年提出。
- ID3算法主要针对属性选择问题。是决策树学习方法中最具影响和最为典型的算法。
- 该方法使用信息增益度选择测试属性。
- 当获取信息时，将不确定的内容转为确定的内容，因此信息伴着不确定性。
- 从直觉上讲，小概率事件比大概率事件包含的信息量大。如果某件事情是“百年一见”则肯定比“习以为常”的事件包含的信息量大。

- **如何度量信息量的大小?**

- “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 $(K=1,2,...,|\mathcal{Y}|)$ ，则 D 的信息熵定义为

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



- $\mathrm{Ent}(D)$ 的值越小，则 D 的纯度越高
- 计算信息熵时约定：若 $p=0$，则 $p\log_2 p=0$
- $\mathrm{Ent}(D)$ 的最小值为 0，最大值为 $\log_2|\mathcal{Y}|$
- 当X为1,0分布时：$P(X=1)=p$，$P(X=0)=1-p$，$0\leqslant p\leqslant 1$
- 熵：

$$H(p)=-p\log_2 p-(1-p)\log_2(1-p)$$

决　　策　　树

- 设有随机变量(X,Y),其联合概率分布为：

$$P(X=x_i,Y=y_j)=p_{ij}\,,\quad i=1,2,\cdots,n\,;\quad j=1,2,\cdots,m$$

- **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：

$$H(Y\mid X)=\sum_{i=1}^{n}p_iH(Y\mid X=x_i)$$

- 当熵和条件熵中的概率由数据估计（特别是极大似然估计）得到时，所对应的熵与条件熵分别称为**经验熵**（empirical entropy)和**经验条件熵**（empirical conditional entropy ）

决　　策　　树

- **定义5.2 (信息增益)**: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

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

- (Information gain)表示得知特征X的信息而使得类Y的信息的不确定性减少的程度.
- 一般地，熵H(Y)与条件熵H(Y|X)之差称为互信息（mutual information)
- 决策树学习中的信息增益等价于训练数据集中类与特征的互信息.

决　　策　　树

- 设训练数据集为D
- |D|表示其样本容量，即样本个数
- 设有K个类C_k, k = 1,2，...K，
- $|C_k|$为属于类C_k的样本个数
- 特征A有n个不同的 取值{a_1,a_2...a_n}根据特征A的取值将D划分为n个子集D_1.....D_n
- $|D_i|$为 D_i的样本个数
- 记子集D_i中属于类C_k的样本集合为D_{ik}
- $|D_{ik}|$为D_{ik}的样本个数

决　　策　　树

- 输入：训练数据集D和特征A；
- 输出：特征A对训练数据集D的信息增益g(D,A)
- 1、计算数据集D的经验熵H(D)

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

- 2、计算特征A对数据集D的经验条件熵H(D|A)

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

- 3、计算信息增益

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

决　策　树

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

解 首先计算经验熵 $H(D)$.

$$H(D)=-\frac{9}{15}\log_2\frac{9}{15}-\frac{6}{15}\log_2\frac{6}{15}=0.971$$

然后计算各特征对数据集 D 的信息增益. 分别以 A_1，A_2，A_3，A_4 表示年龄、有工作、有自己的房子和信贷情况 4 个特征，则

(1)

$$\begin{aligned}g(D,A_1)&=H(D)-\left[\frac{5}{15}H(D_1)+\frac{5}{15}H(D_2)+\frac{5}{15}H(D_3)\right]\\&=0.971-\left[\frac{5}{15}\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)\right.\\&\quad\left.+\frac{5}{15}\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)+\frac{5}{15}\left(-\frac{4}{5}\log_2\frac{4}{5}-\frac{1}{5}\log_2\frac{1}{5}\right)\right]\\&=0.971-0.888=0.083\end{aligned}$$

这里 D_1，D_2，D_3 分别是 D 中 A_1（年龄）取值为青年、中年和老年的样本子集. 类似地，

(2)

$$\begin{aligned}g(D,A_2)&=H(D)-\left[\frac{5}{15}H(D_1)+\frac{10}{15}H(D_2)\right]\\&=0.971-\left[\frac{5}{15}\times0+\frac{10}{15}\left(-\frac{4}{10}\log_2\frac{4}{10}-\frac{6}{10}\log_2\frac{6}{10}\right)\right]=0.324\end{aligned}$$

(3)

$$\begin{aligned}g(D,A_3)&=0.971-\left[\frac{6}{15}\times0+\frac{9}{15}\left(-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}\right)\right]\\&=0.971-0.551=0.420\end{aligned}$$

(4)

$$g(D,A_4)=0.971-0.608=0.363$$

最后，比较各特征的信息增益值. 由于特征 A_3（有自己的房子）的信息增益值最大，所以选择特征 A_3 作为最优特征. ■

解 利用例 5.2 的结果，由于特征 A_3（有自己的房子）的信息增益值最大，所以选择特征 A_3 作为根结点的特征．它将训练数据集 D 划分为两个子集 D_1（A_3 取值为“是”）和 D_2（A_3 取值为“否”）．由于 D_1 只有同一类的样本点，所以它成为一个叶结点，结点的类标记为“是”．

对 D_2 则需从特征 A_1（年龄），A_2（有工作）和 A_4（信贷情况）中选择新的特征．计算各个特征的信息增益：

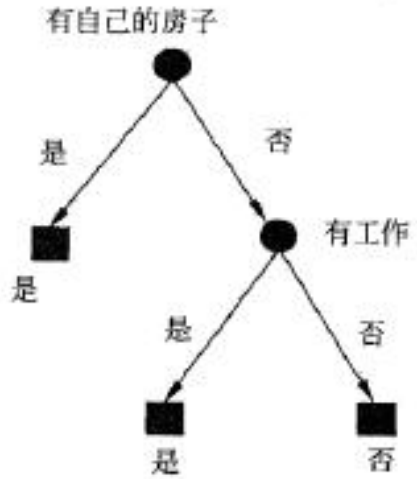
$$g(D_2, A_1) = H(D_2) - H(D_2 \mid A_1) = 0.918 - 0.667 = 0.251$$

$$g(D_2, A_2) = H(D_2) - H(D_2 \mid A_2) = 0.918$$

$$g(D_2, A_4) = H(D_2) - H(D_2 \mid A_4) = 0.474$$

选择信息增益最大的特征 A_2（有工作）作为结点的特征．由于 A_2 有两个可能取值，从这一结点引出两个子结点：一个对应“是”（有工作）的子结点，包含 3 个样本，它们属于同一类，所以这是一个叶结点，类标记为“是”；另一个是对应“否”（无工作）的子结点，包含 6 个样本，它们也属于同一类，所以这也是一个叶结点，类标记为“否”．

这样生成一个如图 5.5 所示的决策树．该决策树只用了两个特征（有两个内部结点）．



决　　策　　树

- 以信息增益作为划分训练数据集的特征，存在偏向于选择取值较多的特征的问题
- 使用信息增益比可以对这一问题进行校正

- 定义5.3（**信息增益比**） 特征A对训练数据集D的信息增益比定义为信息增益与训练数据集D关于特征A的值的熵之比

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

$$H_A(D)=-\sum_{i=1}^{n}\frac{|D_i|}{|D|}\log_2\frac{|D_i|}{|D|}，n\ 是特征\ A\ 取值的个数。$$

决　策　树

- 1 决定分类属性；
- 2 对目前的数据表，建立一个节点N
- 3 如果数据库中的数据都属于同一个类，N就是树叶，在树叶上标出所属的类
- 4 如果数据表中没有其他属性可以考虑，则N也是树叶，按照少数服从多数的原则在树叶上标出所属类别
- 5 否则，根据平均信息期望值E或GAIN值选出一个最佳属性作为节点N测试属性
- 6 节点属性选定后，对于该属性中的每个值：
  - 从N生成一个分支，并将数据表中与该分支有关的数据收集形成分支节点的数据表，在表中删除节点属性那一栏如果分支数据表非空，则运用以上算法从该节点建立子树。
- **ID3算法的基本思想**是，以信息熵为度量，用于决策树节点的属性选择，每次优先选取信息量最多的属性，亦即能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。

决　　策　　树

- 分类树的生成：
- 基尼指数
- 分类问题中，假设有k个类，样本点属于k的概率Pk，则概率分布的基尼指数：

$$\text{Gini}(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$$

- 二分类问题：$\text{Gini}(p)=2p(1-p)$
- 对给定的样本集合D，基尼指数

$$\text{Gini}(D)=1-\sum_{k=1}^{K}\left(\frac{|C_k|}{|D|}\right)^2$$

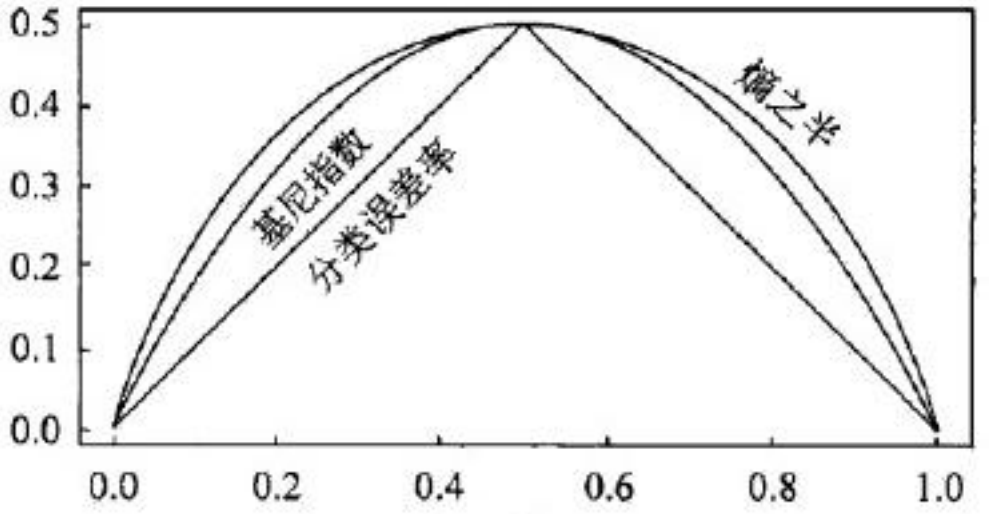
决　　策　　树

- 如果样本集合D根据特征A是否为a被分割成D1和D2，即

$$D_1 = \{(x,y) \in D \mid A(x) = a\}, \quad D_2 = D - D_1$$

- 则在特征A的条件下，集合D的基尼指数：

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



决　　策　　树

- CART生成算法
- 输入：训练数据集D，停止计算条件
- 输出：CART决策树
- 从根节点开始，递归对美国结点操作
- 1、设结点数据集为D，对每个特征A，对其每个值a，根据样本点对A=a的测试为是或否，将D分为D1，D2，计算A=a的基尼指数
- 2、在所有的特征A以及所有可能的切分点a中，选择基尼指数最小的特征和切分点，将数据集分配到两个子结点中。
- 3、对两个子结点递归调用1，2步骤
- 4、生成CART树

解 首先计算各特征的基尼指数，选择最优特征以及其最优切分点．仍采用例 5.2 的记号，分别以 A_1，A_2，A_3，A_4 表示年龄、有工作、有自己的房子和信贷情况 4 个特征，并以 1，2，3 表示年龄的值为青年、中年和老年，以 1，2 表示有工作和有自己的房子的值为是和否，以 1，2，3 表示信贷情况的值为非常好、好和一般．

求特征 A_1 的基尼指数：

$$\mathrm{Gini}(D, A_1=1)=\frac{5}{15}\left(2\times\frac{2}{5}\times\left(1-\frac{2}{5}\right)\right)+\frac{10}{15}\left(2\times\frac{7}{10}\times\left(1-\frac{7}{10}\right)\right)=0.44$$

$$\mathrm{Gini}(D, A_1=2)=0.48$$

$$\mathrm{Gini}(D, A_1=3)=0.44$$

由于 $\mathrm{Gini}(D, A_1=1)$ 和 $\mathrm{Gini}(D, A_1=3)$ 相等，且最小，所以 $A_1=1$ 和 $A_1=3$ 都可以选作 A_1 的最优切分点．

求特征 A_2 和 A_3 的基尼指数：

$$\mathrm{Gini}(D, A_2=1)=0.32$$

$$\mathrm{Gini}(D, A_3=1)=0.27$$

由于 A_2 和 A_3 只有一个切分点，所以它们就是最优切分点．

求特征 A_4 的基尼指数：

$$\mathrm{Gini}(D, A_4=1)=0.36$$

$$\mathrm{Gini}(D, A_4=2)=0.47$$

$$\mathrm{Gini}(D, A_4=3)=0.32$$

$\mathrm{Gini}(D, A_4=3)$ 最小，所以 $A_4=3$ 为 A_4 的最优切分点．

在 A_1，A_2，A_3，A_4 几个特征中，$\mathrm{Gini}(D, A_3=1)=0.27$ 最小，所以选择特征 A_3 为最优特征，$A_3=1$ 为其最优切分点．于是根结点生成两个子结点，一个是叶结点．对另一个结点继续使用以上方法在 A_1，A_2，A_4 中选择最优特征及其最优切分点，结果是 $A_2=1$．依此计算得知，所得结点都是叶结点． ■

决　　策　　树

- 理想的决策树有三种：
  - (1)叶子结点数最少；
  - (2)叶子结点深度最小；
  - (3)叶子结点数最少且叶子结点深度最小。
- 然而，洪家荣等人已经证明了要找到这种最优的决策树是NP难题。因此，决策树优化的目的就是要找到尽可能趋向于最优的决策树。
- **过度拟合**
- 决策树算法增长树的每一个分支的深度，直到恰好能对训练样例比较完美地分类。实际应用中，当数据中有噪声或训练样例的数量太少以至于不能产生目标函数的有代表性的采样时，该策略可能会遇到困难。
- 这个简单算法产生的树会过渡拟合训练样例（过拟合：Over Fitting）.

决　　策　　树

- 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

决　策　树

训练集

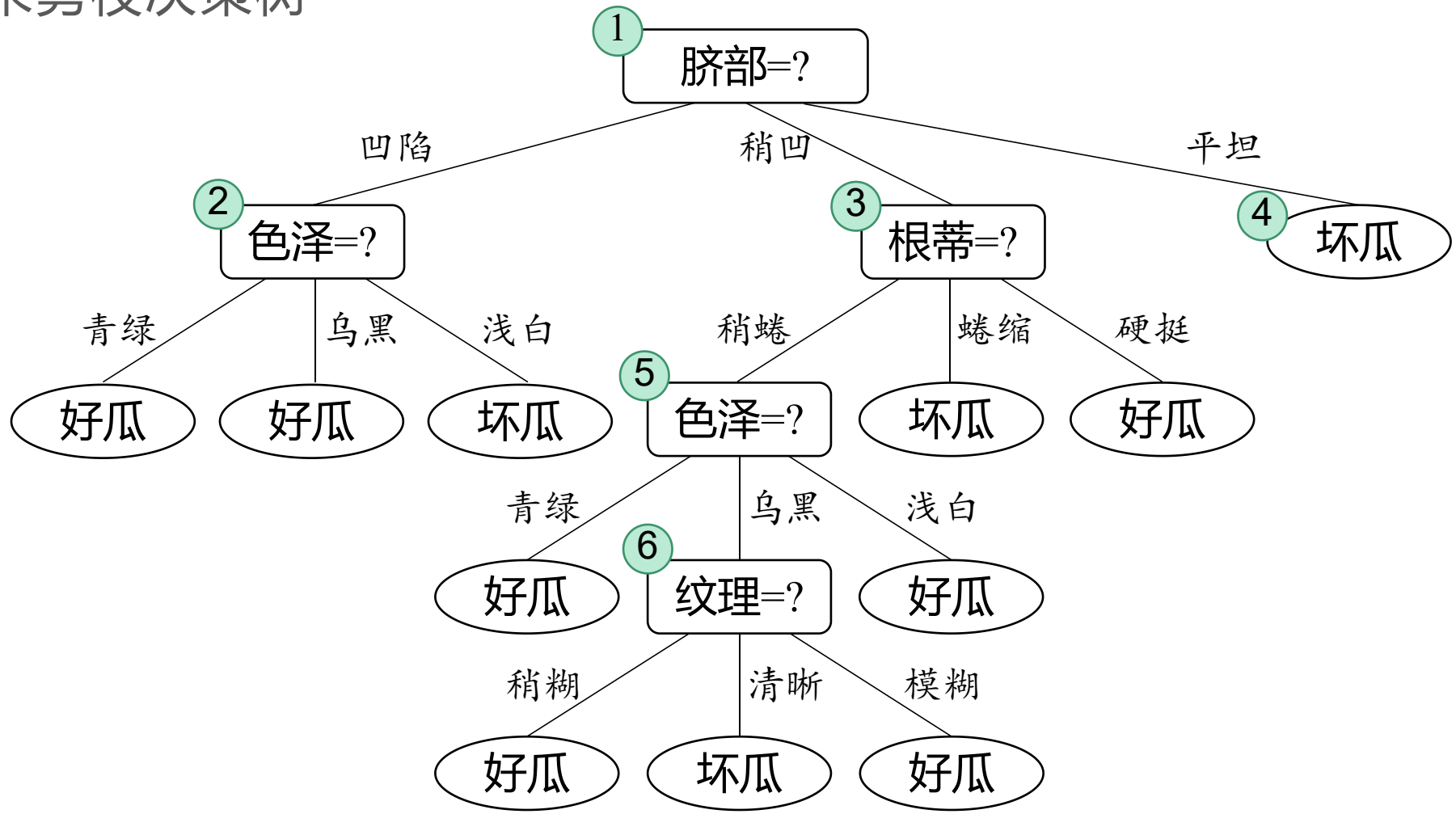
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

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

决　策　树

- 未剪枝决策树


1
脐部=?
凹陷
稍凹
平坦
2
色泽=?
3
根蒂=?
4
坏瓜
青绿
乌黑
浅白
稍蜷
蜷缩
硬挺
好瓜
好瓜
坏瓜
5
色泽=?
坏瓜
好瓜
青绿
乌黑
浅白
6
好瓜
纹理=?
好瓜
稍糊
清晰
模糊
好瓜
坏瓜
好瓜

决　　策　　树

- 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

决　策　树

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

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

验证集精度

“脐部=?”划分前: 42.9%

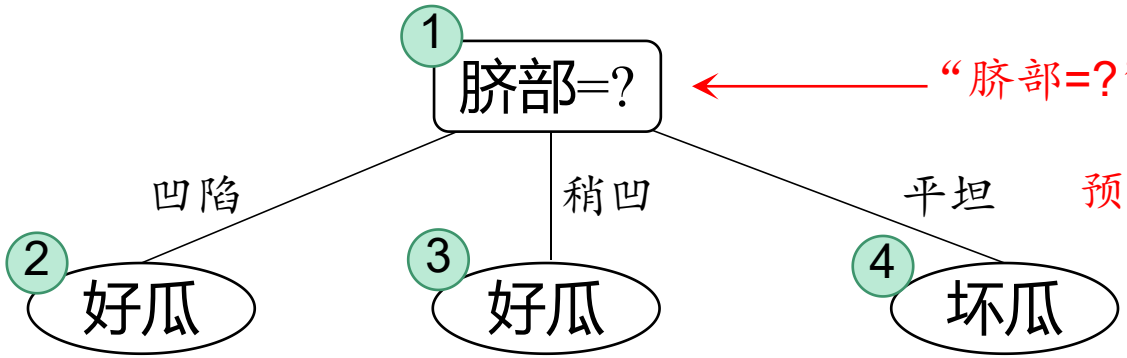
决　策　树

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%

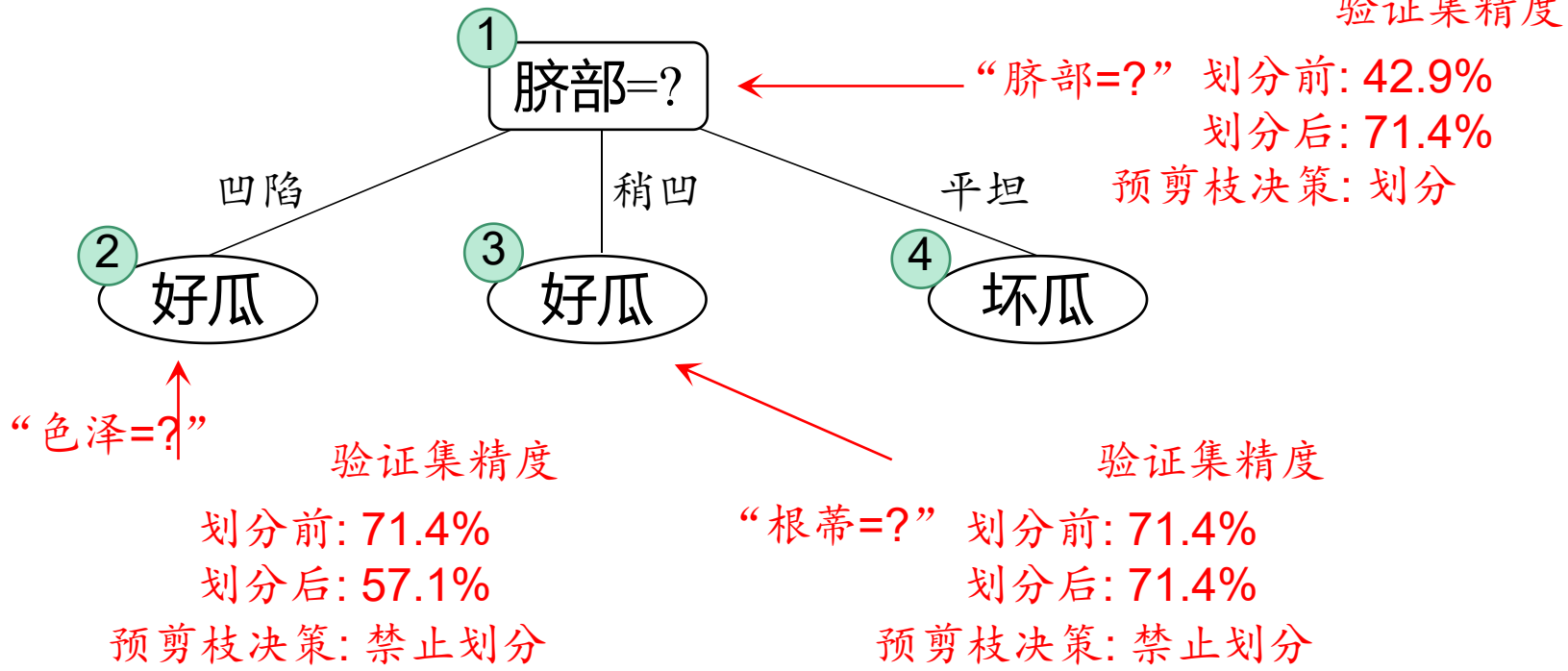
预剪枝决策: 划分

决　策　树

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”**



决　　策　　树

- 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

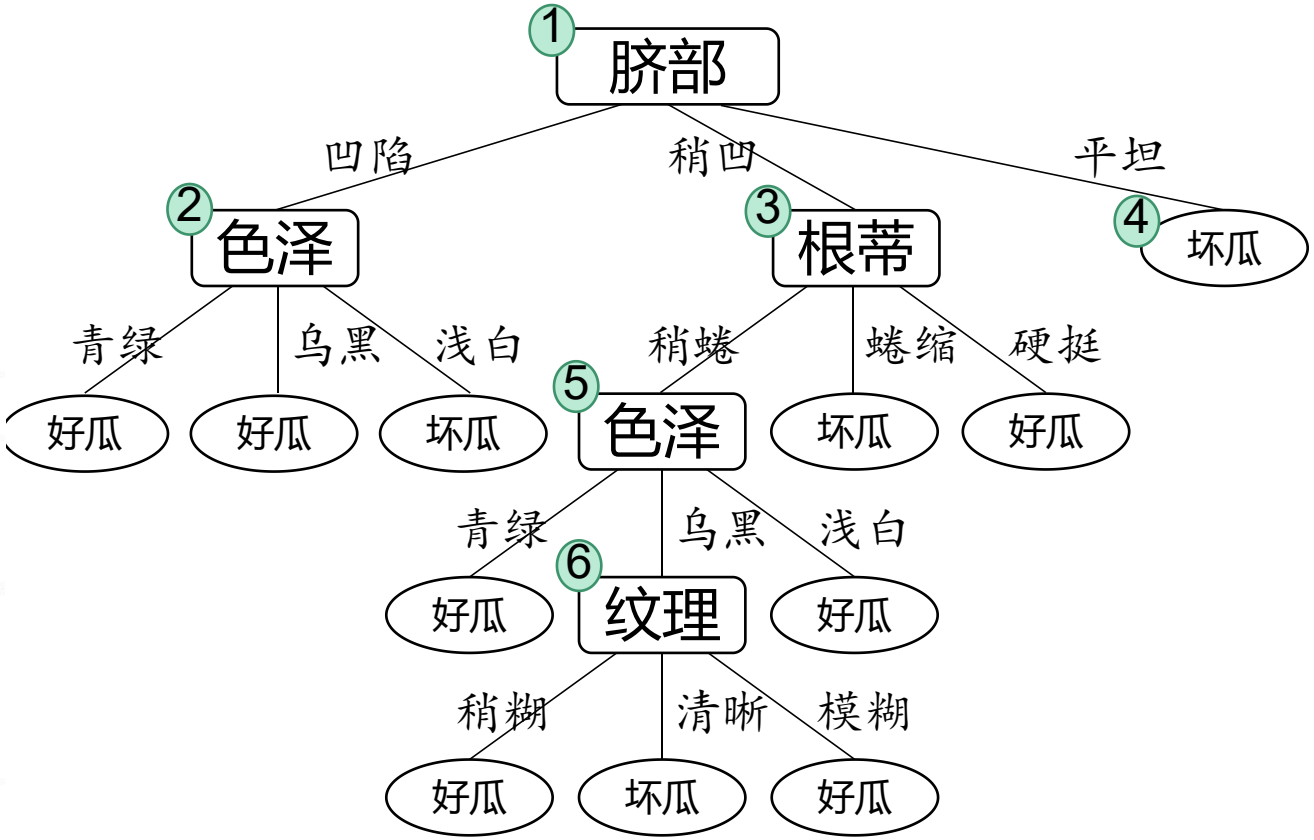
决　策　树

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%

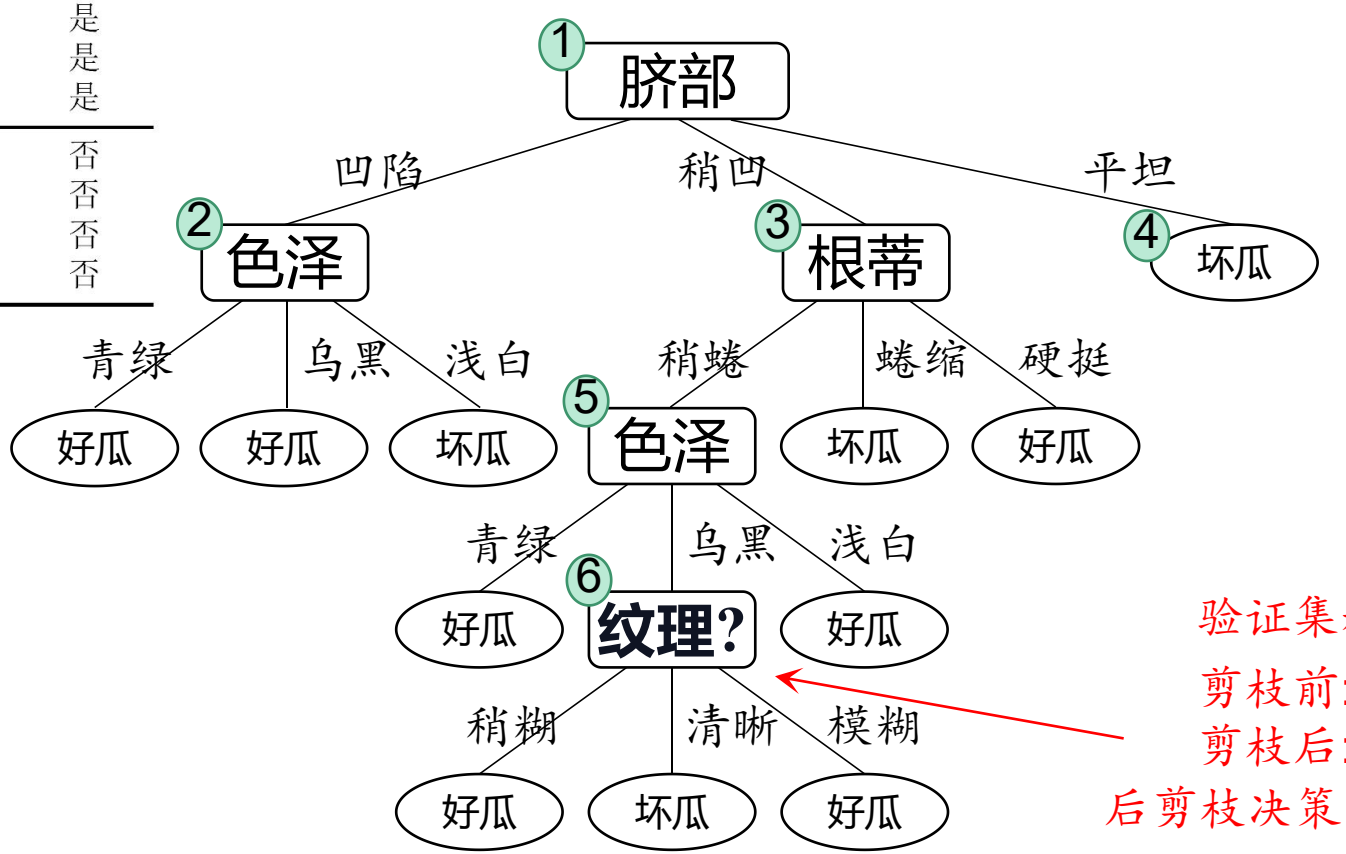
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否



决　策　树

- 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

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



验证集精度

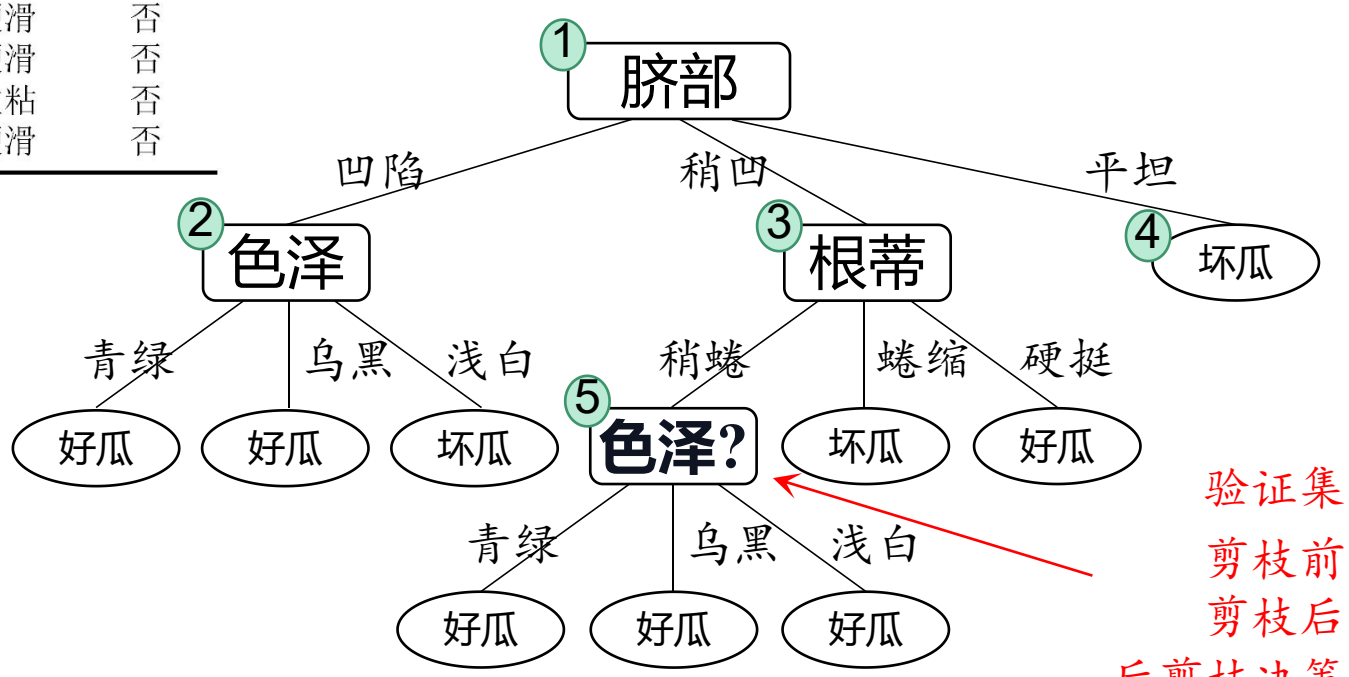
剪枝前: 42.9%

剪枝后: 57.1%

后剪枝决策: 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

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



验证集精度

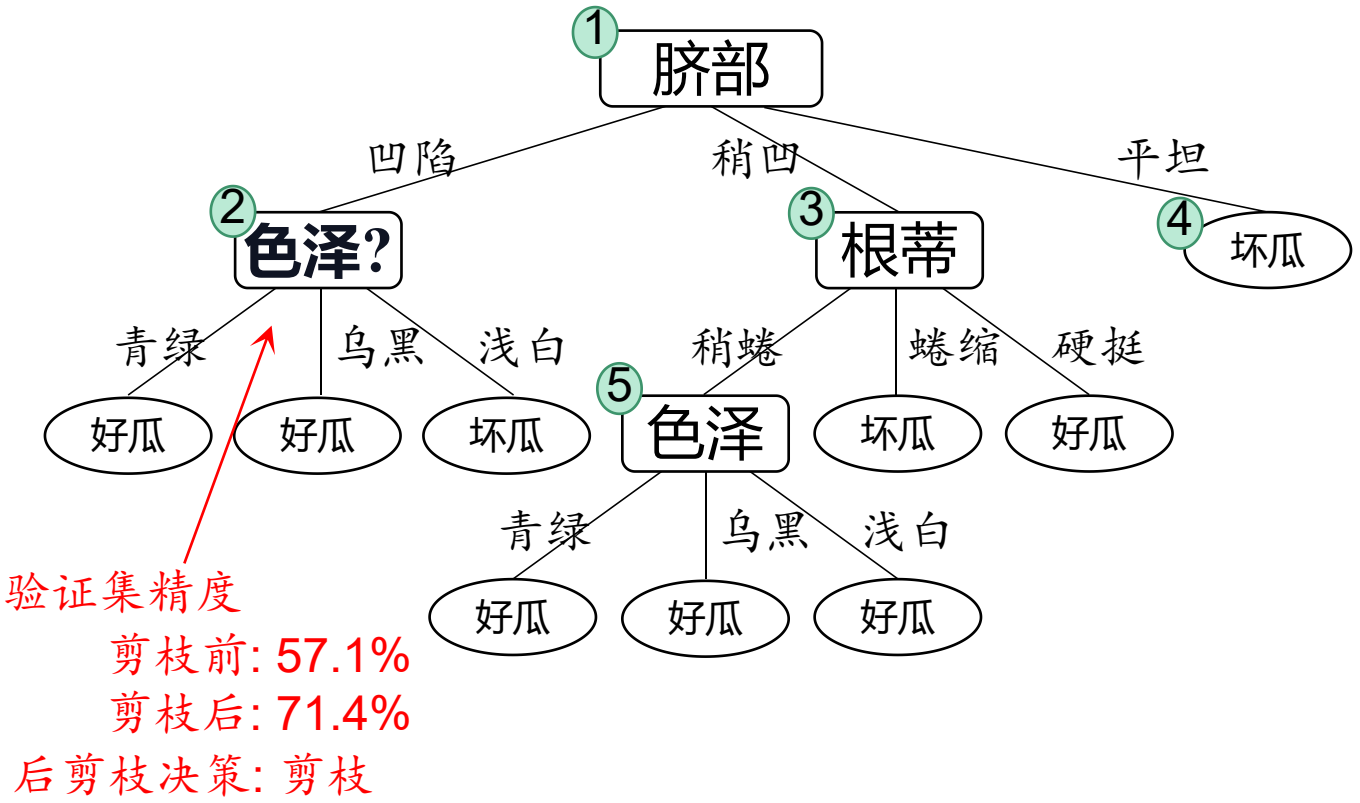
剪枝前: 57.1 %

剪枝后: 57.1%

后剪枝决策: 不剪枝

决　策　树

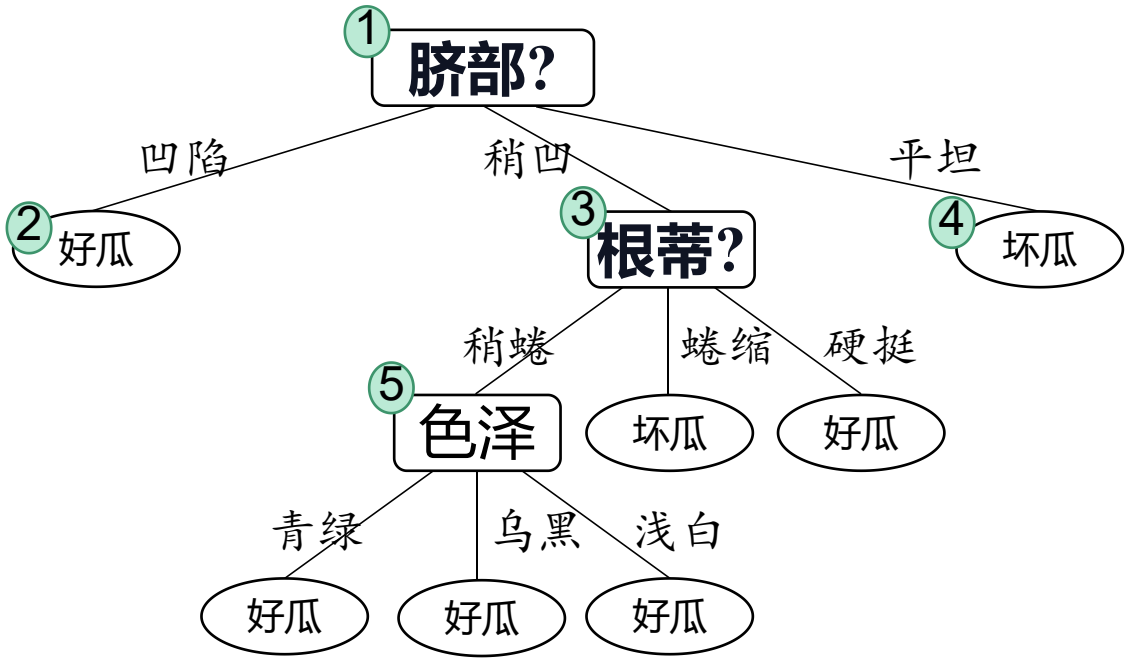
- 对结点②，若将其替换为叶结点，根据落在其上的训练样本 $\{1, 2, 3, 14\}$ ，将其标记为“好瓜”，得到验证集精度提升至 71.4% ，则决定剪枝



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

决　　策　　树

- 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. 最终基于后剪枝策略得到的决策树如图所示



- 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

感 谢 观 看

统 计 机 器 学 习

主讲人：彭振华

数学与计算机学院

2026年