机器学习
Machine Learning

北京航空航天大学计算机学院智能识别与图像处理实验室
IRIP Lab, School of Computer Science and Engineering, Beihang University
黄 迪　刘庆杰

2018年秋季学期
Fall 2018

课前回顾

概率图模型

- **概率图模型（Probabilistic Graphical Model）**

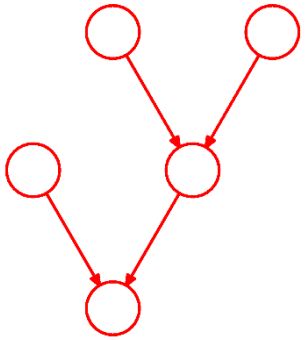
概率论

$$p(X) = \sum_{Y} p(X, Y)$$

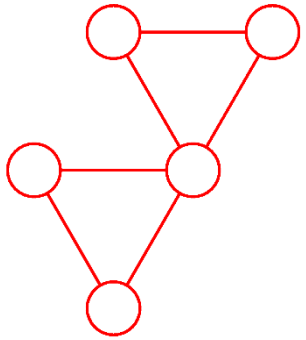
$$p(X, Y) = p(Y|X)p(X)$$

$$p(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

图论



有向图

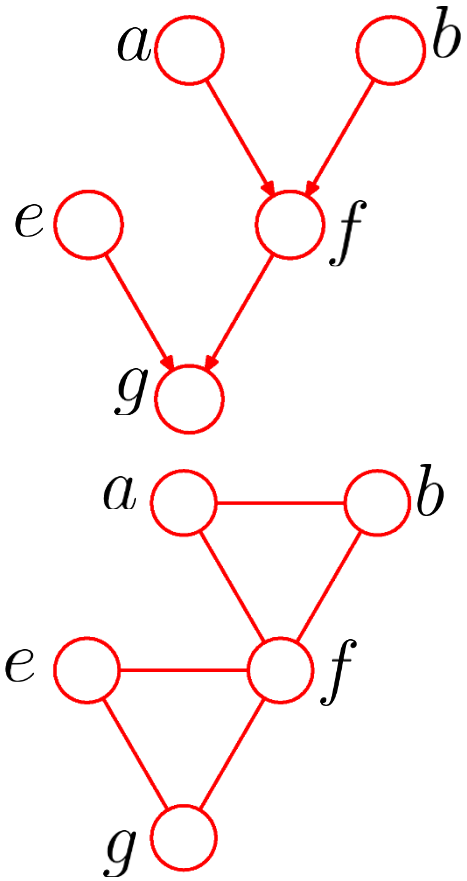


无向图

概率论 + 图论 = 概率图模型

概率图模型

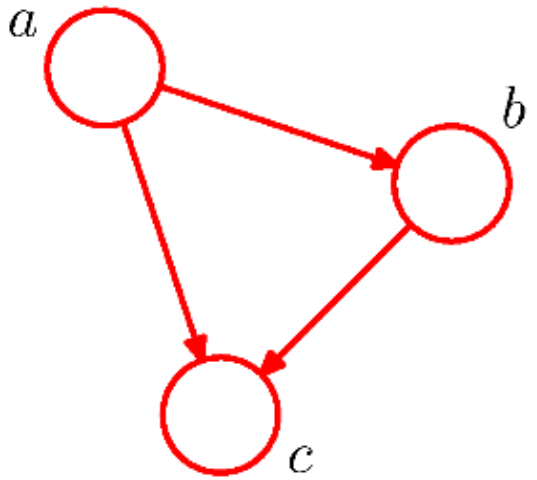
- **概率图模型（Probabilistic Graphical Model）**



- **结点：随机变量或一组随机变量**
- **连接弧：随机变量之间的关系**

概率图模型

- **贝叶斯网络（Bayesian Network）**
  - **有向无环图（Directed Acyclic Graph, DAG）**

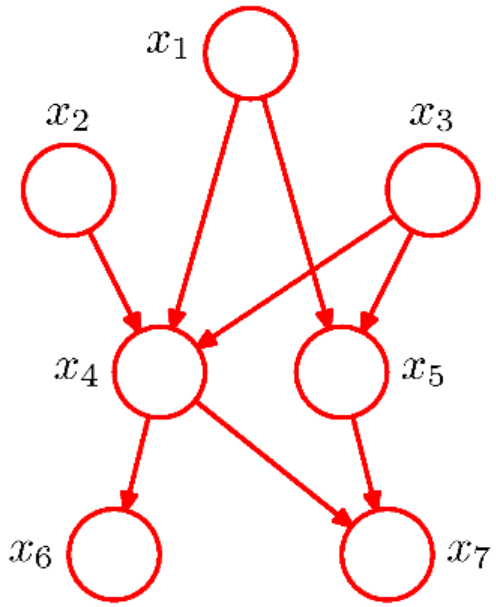


$$p(a,b,c) = p(c|a,b)p(a,b) = p(c|a,b)p(b|a)p(a)$$

$$p(x_1,,\ldots,x_K) = p(x_K|x_1,\ldots,x_{K-1})\ldots p(x_2|x_1)p(x_1)$$

概率图模型

- 贝叶斯网络（Bayesian Network）



$$p(x_1,\ldots,x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)$$
$$p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5)$$

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k)$$

条件独立
(Conditional Independence)

- 三个变量 a, b, c

$$p(a|b,c) = p(a|c)$$

称在**给定*c*的条件下，*a*与*b*条件独立**

$$\begin{aligned} p(a,b|c) &= p(a|b,c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned}$$

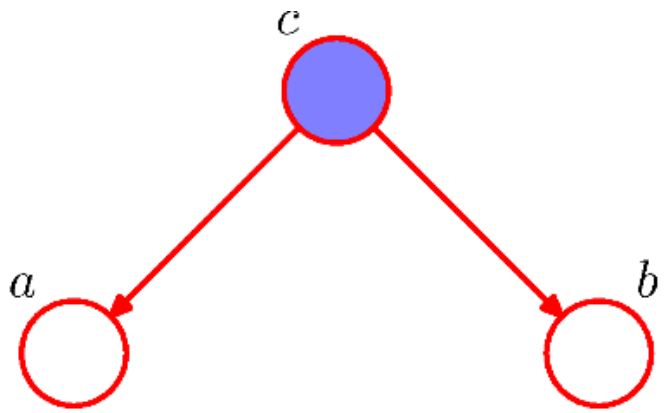
$$p(a,b|c) = p(a|c)p(b|c) \Rightarrow a \perp\!\!\!\perp b \mid c$$

条件独立
（Conditional Independence）

$$p(a,b|c) = \frac{p(a,b,c)}{p(c)}$$

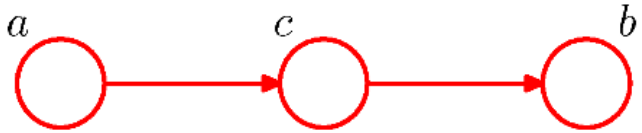
$$= p(a|c)p(b|c)$$

$$a \perp\!\!\!\perp b \mid c$$



尾尾相连（Tail-to-Tail）

条件独立
（Conditional Independence）

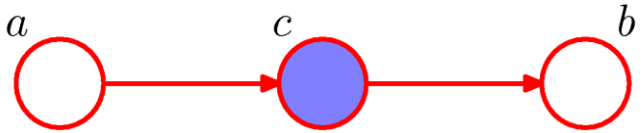


$$p(a,b,c) = p(a)p(c|a)p(b|c)$$

$$p(a,b) = p(a)\sum_c p(c|a)p(b|c)$$

$$= p(a)p(b|a)$$

$$a \not\perp\!\!\!\perp b \mid \emptyset$$



$$p(a,b|c) = \frac{p(a,b,c)}{p(c)}$$

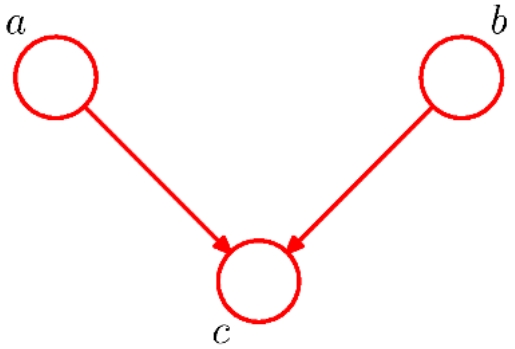
$$= \frac{p(a)p(c|a)p(b|c)}{p(c)}$$

$$= p(a|c)p(b|c)$$

$$a \perp\!\!\!\perp b \mid c$$

头尾相连（Head-to-Tail）

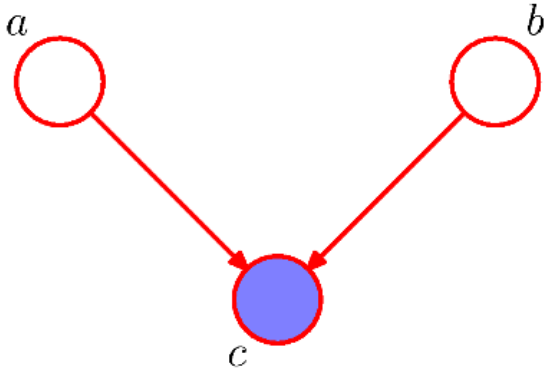
条件独立
（Conditional Independence）



$$p(a,b,c) = p(a)p(b)p(c|a,b)$$

$$p(a,b) = p(a)p(b)$$

$$a \perp\!\!\!\perp b \mid \emptyset$$



$$p(a,b|c) = \frac{p(a,b,c)}{p(c)}$$

$$= \frac{p(a)p(b)p(c|a,b)}{p(c)}$$

$$a \not\perp\!\!\!\perp b \mid c$$

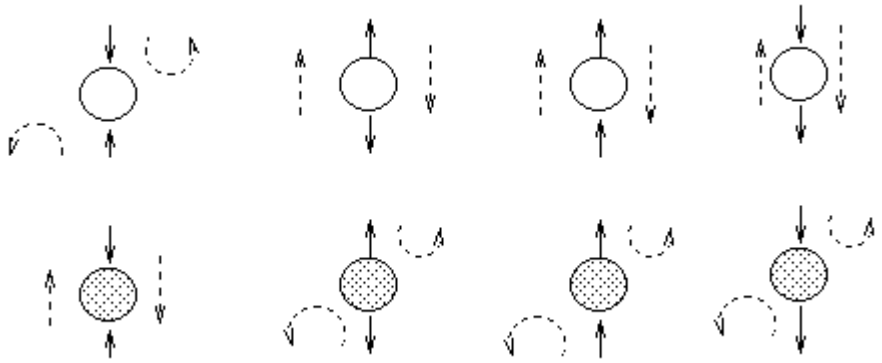
头头相连（Head-to-Head）

“D-分离”
(D-Separation)

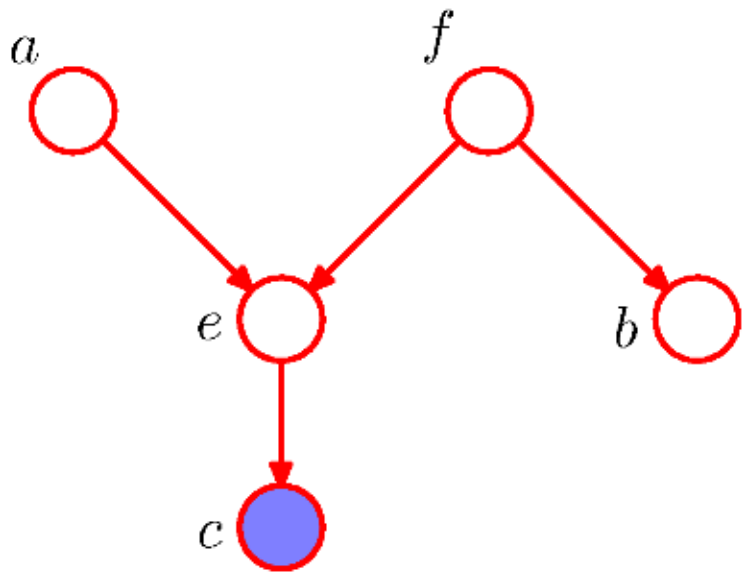
$$A \perp\!\!\!\perp B \mid C$$

看A与B相连的每条路径有没有都被阻隔（blocked）

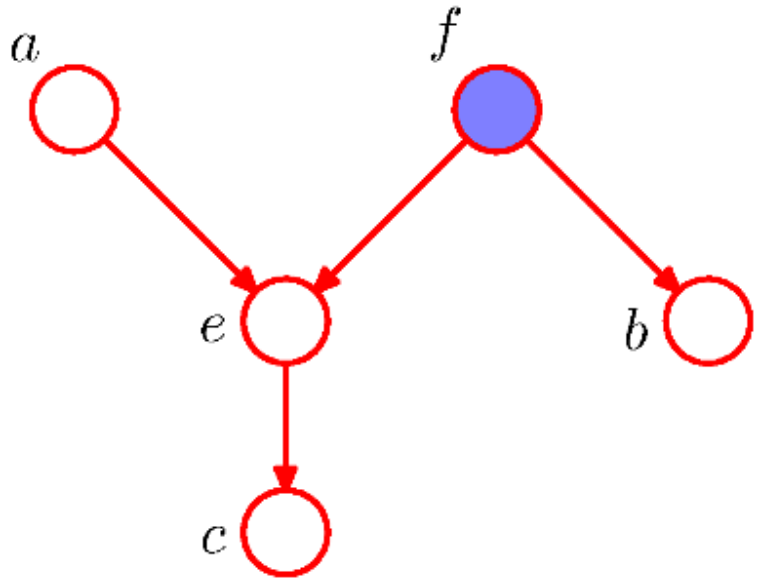
- 满足“头尾相连”或“尾尾相连”的节点都在C中；
- 满足“头头相连”的节点和它的任何后裔节点都不在C中



“D-分离”
（D-Separation）



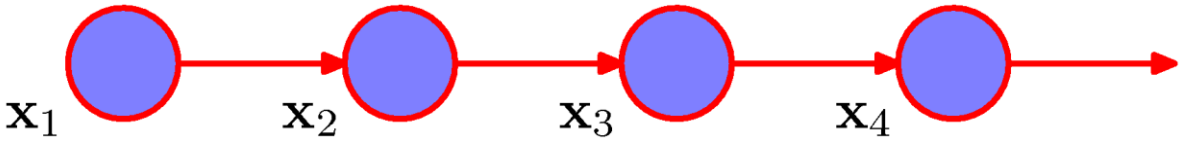
$a \not\perp\!\!\!\perp b \mid c$



$a \perp\!\!\!\perp b \mid f$

隐马尔可夫模型
(Hidden Markov Models)

- 马尔可夫链



$$p(\mathbf{x}_n|\mathbf{x}_1,\ldots,\mathbf{x}_{n-1}) = p(\mathbf{x}_n|\mathbf{x}_{n-1})$$

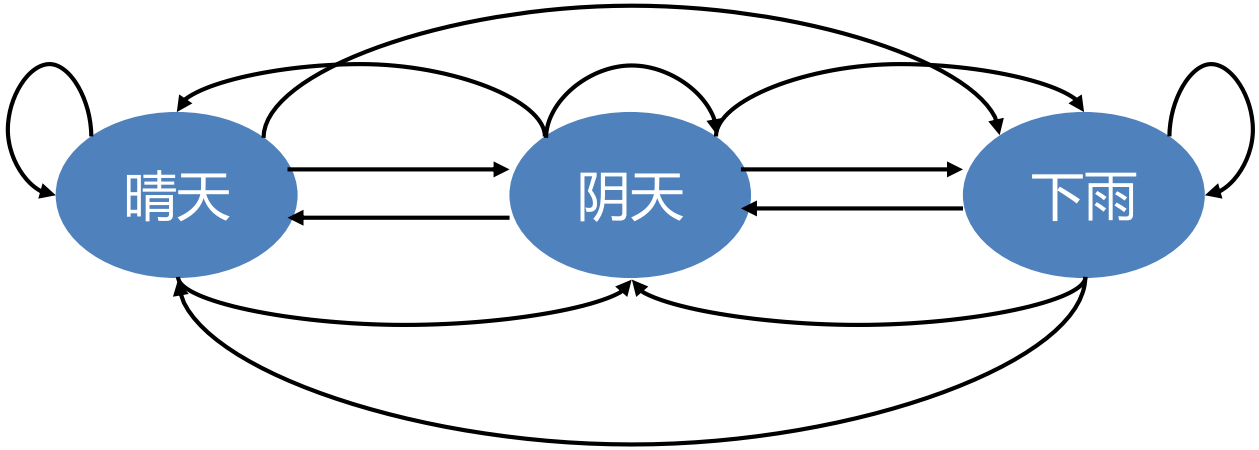
$$p(\mathbf{x}_1,\ldots,\mathbf{x}_n) = p(\mathbf{x}_1)\prod_{n=2}^{N} p(\mathbf{x}_n|\mathbf{x}_{n-1})$$

如果一个过程的“将来”仅依赖“现在”而不依赖“过去”，则此过程具有马尔可夫性，或称此过程为马尔可夫过程。

$$X(t+1) = f(X(t))$$

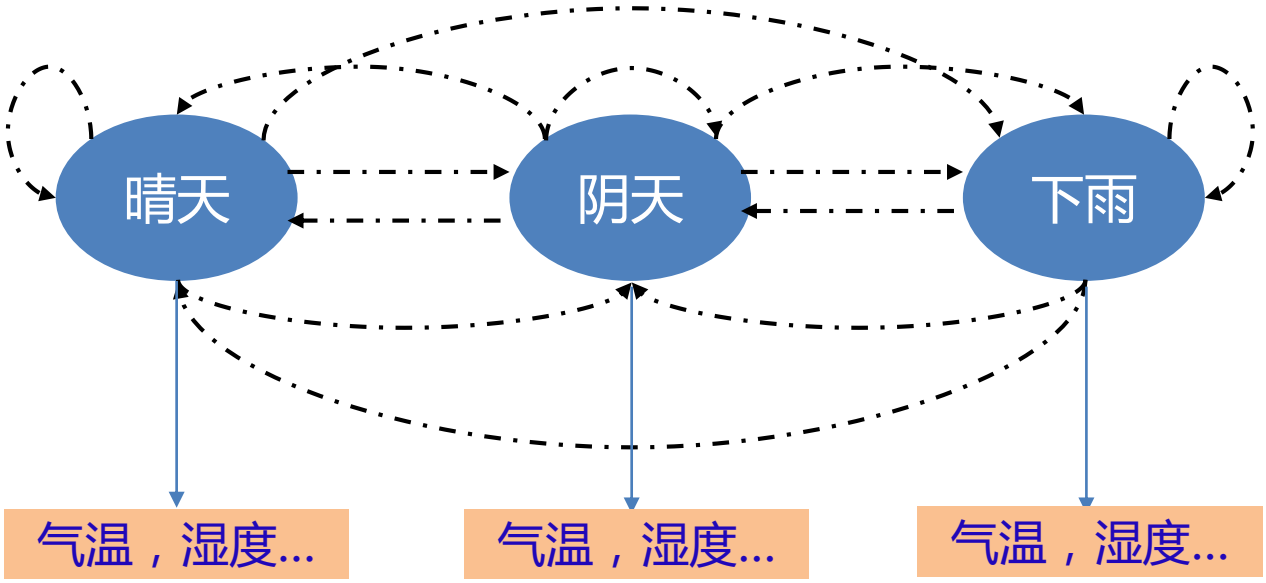
隐马尔可夫模型
(Hidden Markov Models)

- **时间和状态都离散的马尔可夫过程称为马尔可夫链**



隐马尔可夫模型
(Hidden Markov Models)

- 状态序列（State Sequence）不可见的过程为隐马尔可夫过程



- 只能得到对状态的观测序列（Observation Sequence）

隐马尔可夫模型
(Hidden Markov Models)

- **HMM的状态是不确定或不可见的，只有通过观测序列的随机过程才能表现出来**
- **观察到的事件与状态并不是一一对应，而是通过一组概率分布相联系**
- **HMM是一个双重随机过程**
  - 马尔可夫随机：状态之间的转移是随机的，且具有马尔可夫性，状态之间的转移用**转移概率**描述。
  - 一般随机过程：状态生成某种观测是随机的，用**观测概率**描述。

隐马尔可夫模型 (Hidden Markov Models)

- **HMM的模型用 (N, M, π, A, B) 五元组来表示，或简写为 $\lambda = (\pi, A, B)$**

参数	含义
N	状态数目
M	观测值数目
π	初始状态概率分布
A	与时间无关的状态转移概率矩阵
B	输出观测概率矩阵

- **HMM两个基本假设**
  - 齐次马尔可夫假设
  - 观测独立假设

隐马尔可夫模型
(Hidden Markov Models)

- **HMM的三个基本问题**
  - **概率计算问题**：给定观测序列$O=\{O_1,$ ，以及模型$\lambda=(\pi,A,B)$，如何计算$P(O|\lambda)$

    前向-后向算法
  - **预测问题(解码问题)**：给定观测序列O，及模型$\lambda=(\pi,A,B)$，如何选择一状态序列$(q_1,q_2,\ldots,q_T)$，使得S能够合理的解释观测序列

    维特比（Viterbi）算法
  - **学习问题**：给定观测序列$O=\{O_1,O_2,$ 模型参数$\lambda=(\pi,A,B)$，使得在该模型下观 概率最大

    Baum-Welch算法

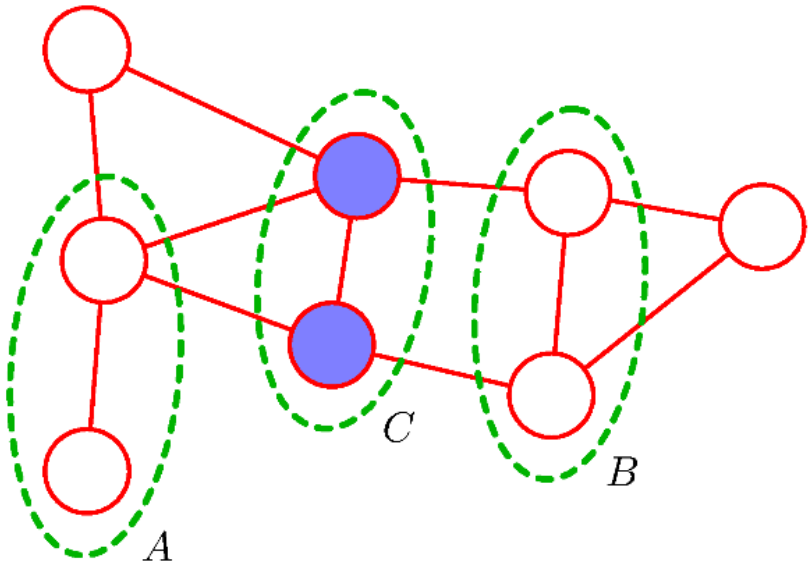
马尔科夫随机场
(Markov Random Fields)

- 马尔可夫随机场(Markov Network or Undirected Graphical Model)

  - 如果A，B之间每条路径存在至少 一个节点在C中

  或者

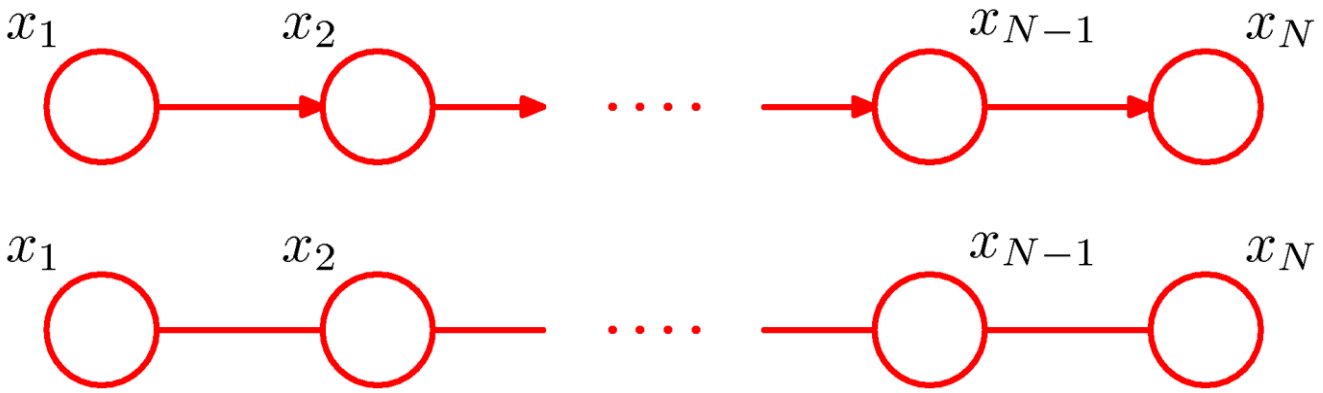
  - 如果去掉C中的所有的节点，A和B没有连通路径



$$A \perp\!\!\!\perp B \mid C$$

因式分解（Factorization）

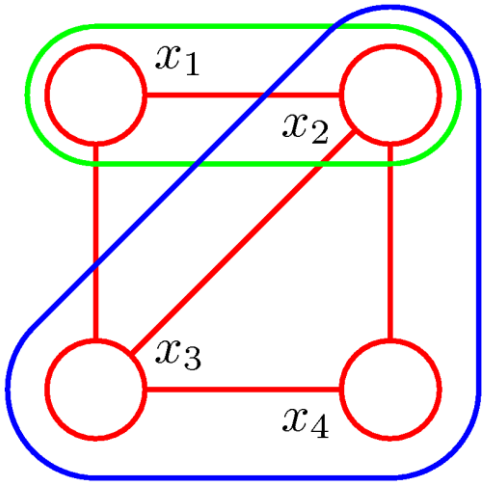
- 马尔可夫随机场



$$p(x_i, x_j|\mathbf{x}_{\setminus\{i,j\}}) = p(x_i|\mathbf{x}_{\setminus\{i,j\}})p(x_j|\mathbf{x}_{\setminus\{i,j\}})$$

因式分解（Factorization）

- 马尔可夫随机场



团 (*Clique*)

$\{x_1, x_2\}$

$\{x_2, x_3\}$

$\{x_3, x_4\}$

$\{x_4, x_2\}$

$\{x_1, x_3\}$

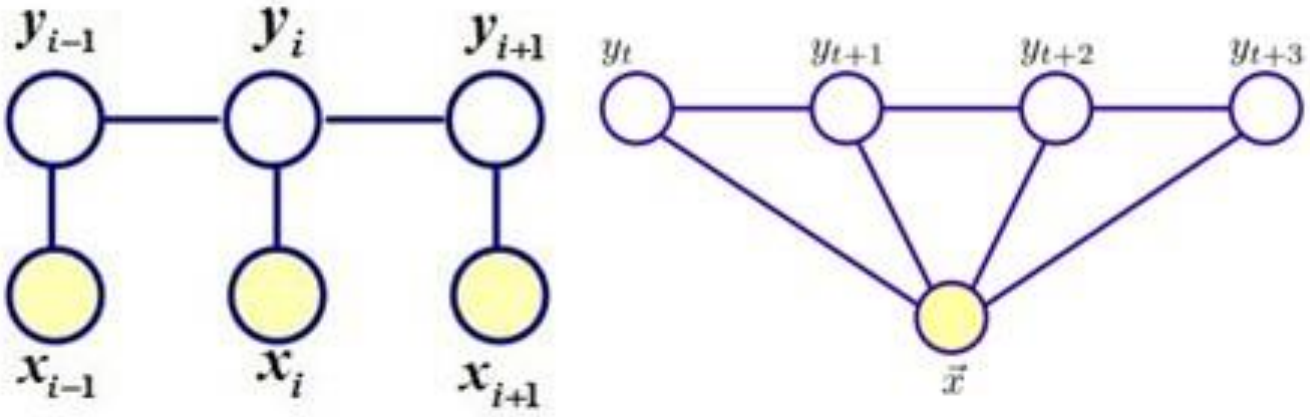
$\{x_1, x_2\, x_3\}$

$\{x_2\, x_3, x_4\}$

最大团 (Maximal Clique)

条件随机场
（Conditional Random Fields）

- **条件随机场(CRFs)模型最早由Lafferty等人于2001年ICML提出的。**
- **CRF是在给定随机变量*X*(或*X*的观测)条件下，随机变量*Y*的马尔可夫场。**

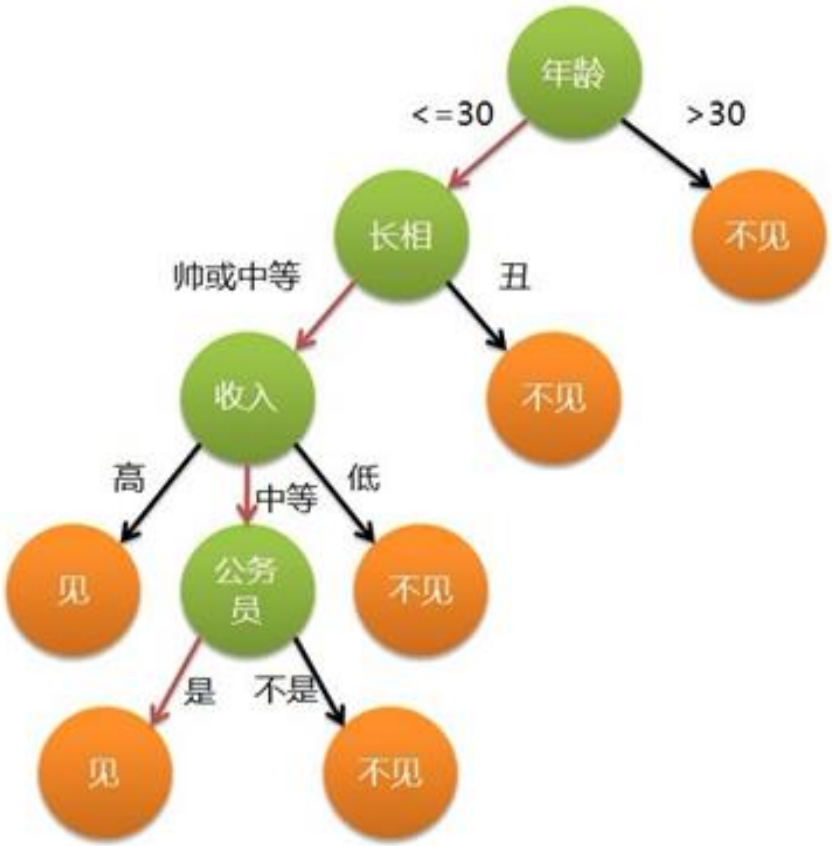


第十二章：决策树

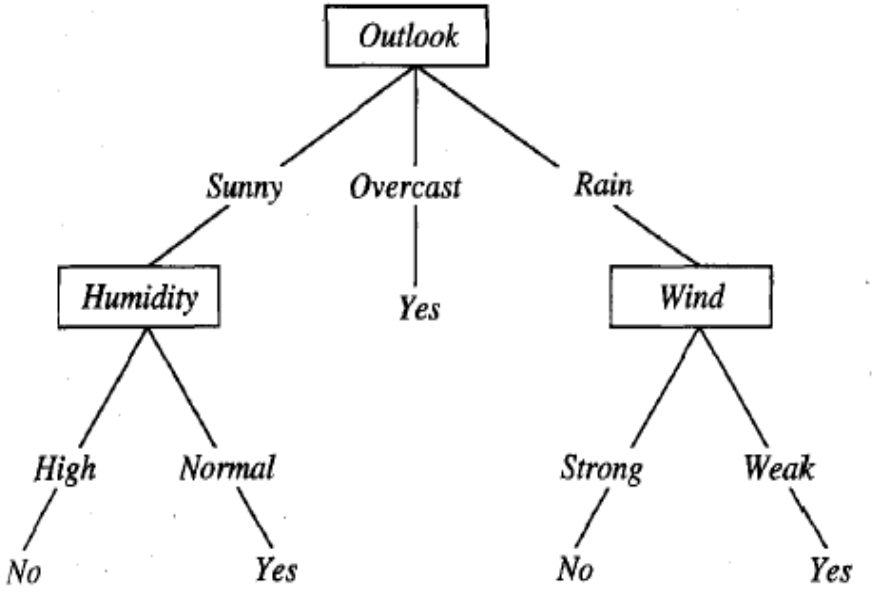
Chapter 12: Decision Tree

引例

- 引例1：相亲



- 引例2：天气是否适合打网球



$$(Outlook = Sunny \wedge Humidity = Normal)$$
$$\vee \quad (Outlook = Overcast)$$
$$\vee \quad (Outlook = Rain \wedge Wind = Weak)$$

P38《机器学习》, T. M. Mitchell著，曾华军译

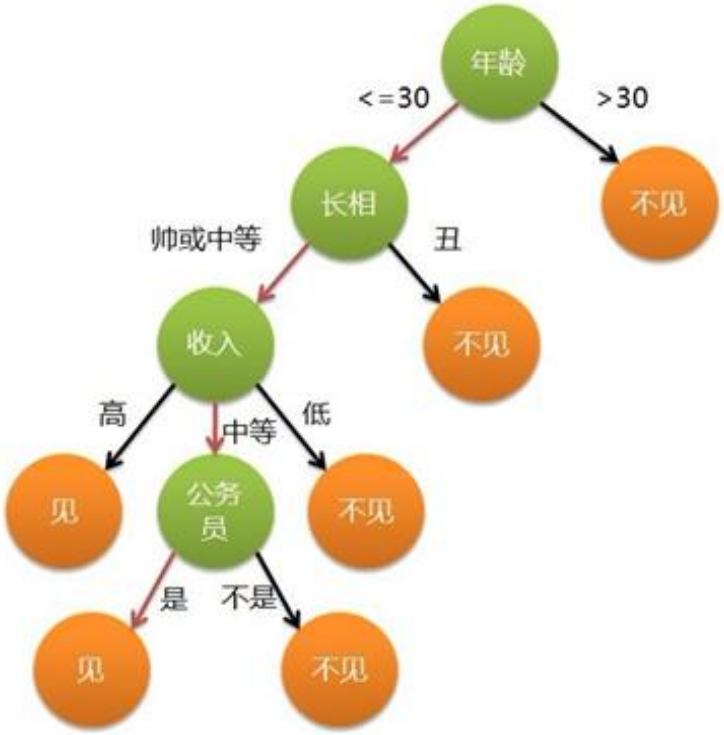
问题背景

- **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**
  - 把样例分类到各可能的离散值对应的类别
- **问题特征**
  - 实例由“属性-值”对表示，

    训练数据可以包含缺少属性值的实例
  - 属性可以是连续值或离散值
  - 具有离散的输出值

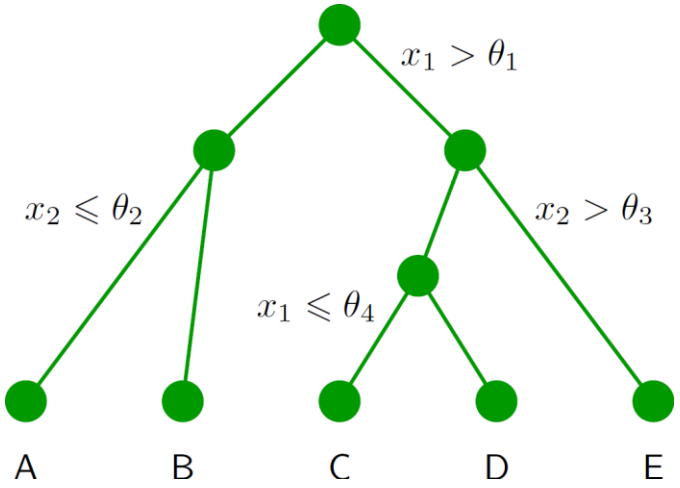
决策树定义

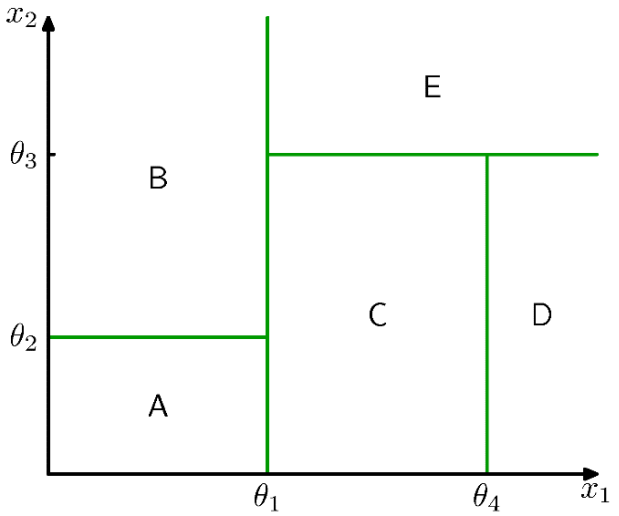
- 决策树(Decision Tree)
  - 决策树是一种树型结构，由结点和有向边组成
  - 结点
    - 内部结点表示一个属性或特征
    - 叶结点代表一种类别
  - 有向边/分支
    - 分支代表一个测试输出



决策树算法

- 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，到叶子结点处的熵值为零，此时每个叶结点中的实例都属于同一类
    - 决策树可以看成一个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)





算法流程

- **基本流程分为两步**
  - **第1步：训练，从数据中获取知识进行学习**

利用训练集建立(并精化)一棵决策树，构建决策树模型.

  - **第2步：测试，利用生成的模型对输入数据进行分类**

对测试样本，从根结点依次测试记录的属性值，直至到达某个叶结点，找到该样本所在的类别.

算法流程

- 构建过程的基本流程
  - Step1：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.
  - Step2：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.
  - Step3：否则，选取一个从该结点到根路径中没有出现过的属性作为标记标识该结点，然后就这个属性的所有取值继续创建树的分支；重复算法步骤2.

主要算法

- 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？富？帅？　会C++？会图像处理？深度学习？

- **目标**：每个分支结点的样本尽可能属于同一类别，即结点的“**纯度**”**(Purity)**越来越高
- 根据不同的目标函数，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)迭代二分器算法
  - 由J. R. Quinlan于1979年提出
  - 一种最经典的决策树学习算法

基本思想：以信息熵为度量，用于决策树结点的属性选择，每次优先选取信息增益最大的属性，即能使熵值最小的属性，构造一棵熵值下降最快的决策树。到叶子结点的熵值为0，此时对应实例集中的实例属于同一类别。

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：

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

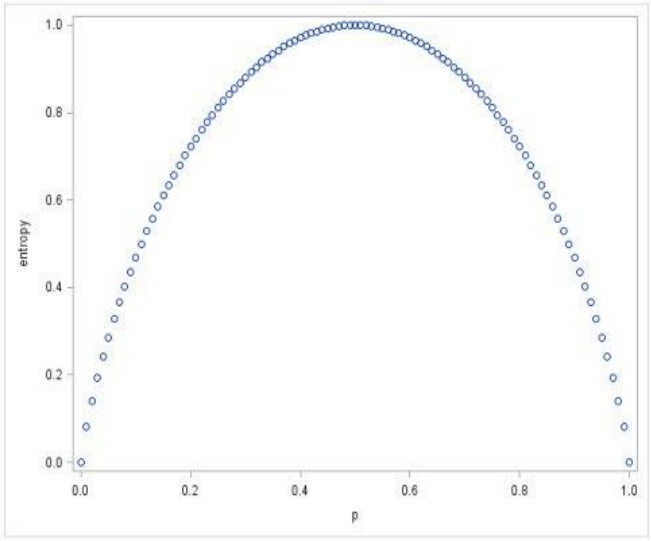
信息按二进制位编码，因此以2为底

  - 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)
  - 熵定义了一个函数(概率密度函数)到一个值(信息熵)的映射　　P(*x*)→H (函数→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，则*D*的经验信息熵(简称经验熵)定义为:

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

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

  - *H*(*D*)的值越小，则*D*的纯度越高.

ID3算法

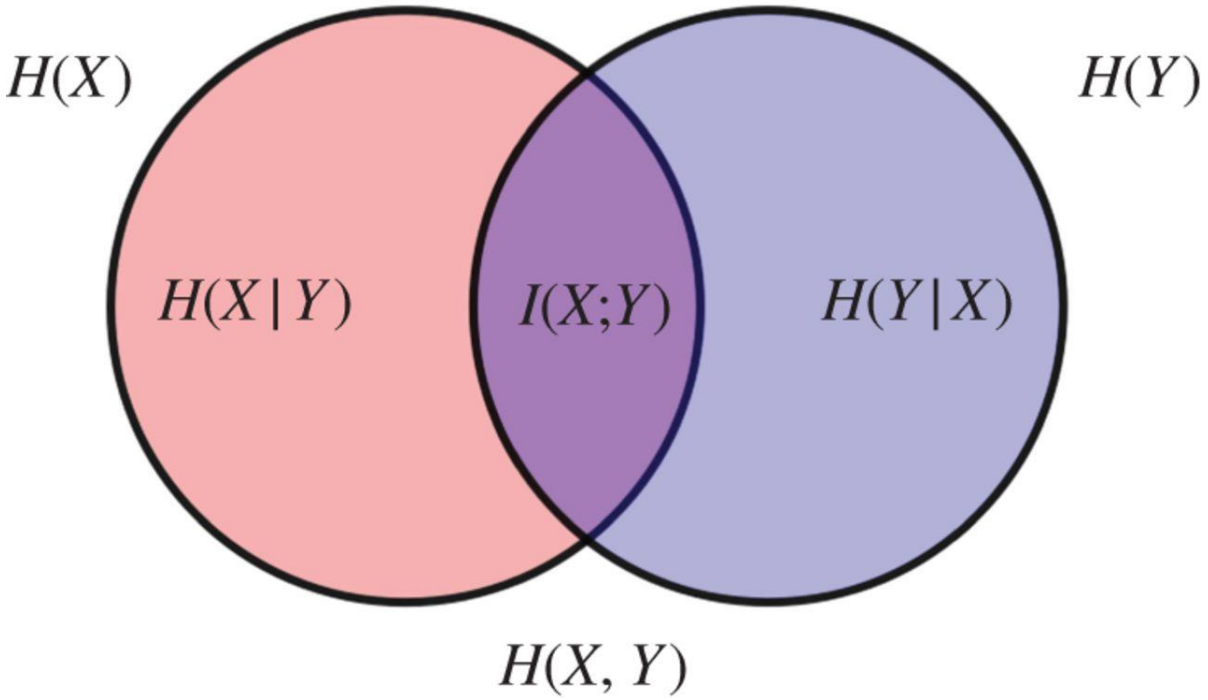
- **条件熵(Conditional Entropy)**
  - **对随机变量(*X*, *Y*)，联合分布为** $p(X = x_i, Y = y_i) = p_{ij}$ **条件熵*H*(*Y*|*X*)表示在已知随机变量*X*的条件下，随机变量*Y*的不确定性，定义为在*X*给定条件下*Y*的条件概率分布的熵对*X*的数学期望：**

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

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

ID3算法

- 条件熵(Conditional Entropy)



Venn图

ID3算法

- **条件熵-推导**

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

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$\{a_1, a_2, \ldots, a_N\}$，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

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

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

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

  - 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：**

$$G(D,a) = H(D) - H(D \mid a)$$

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

  - **ID3算法即是以此信息增益为准则，对每次递归的结点属性进行选择的.**

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$A=\Phi$，则T为单结点树，并将D中实例数最大类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，分割D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，$A-\{A_g\}$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(1)

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(1)

- 计算信息熵-以属性色泽为例

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

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

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

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

参照教材《机器学习》-周志华P75-77示例

ID3算法-示例(1)

- 计算信息增益-以属性色泽为例

$$G(D,\text{色泽})=H(D)-\sum_{n=1}^{3}\frac{|D^n|}{|D|}H(D^n)$$

$$=0.998-\left(\frac{6}{17}\times 1.000+\frac{6}{17}\times 0.918+\frac{5}{17}\times 0.722\right)$$

$$=0.109$$

$G(D,$ 色泽$) = 0.109$

$G(D,$ 根蒂$) = 0.143$

$G(D,$ 敲声$) = 0.141$

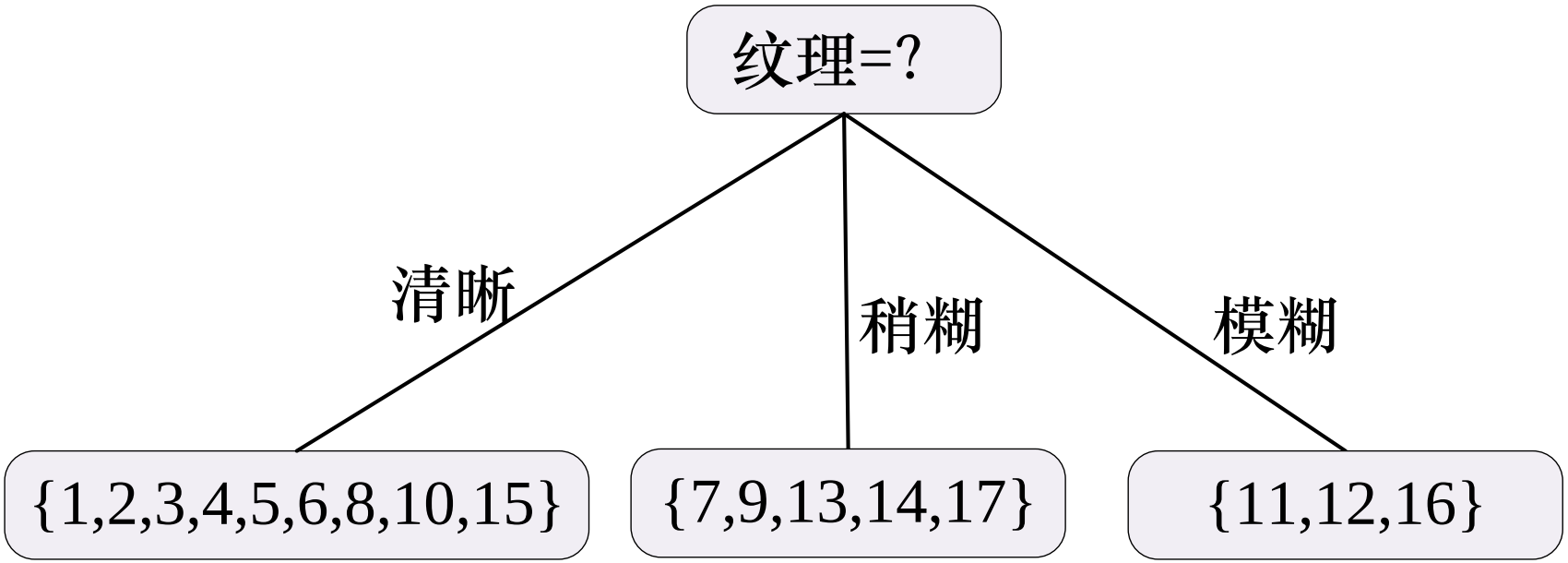
$G(D,$ 纹理$) = 0.381$

$G(D,$ 脐部$) = 0.289$

$G(D,$ 触感$) = 0.006$

ID3算法-示例(1)

- 基于属性“纹理”对根节点进行划分



ID3算法-示例(1)

- 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

$G(D^{\text{清晰}}, \text{色泽}) = 0.043$

$G(D^{\text{清晰}}, \text{敲声}) = 0.331$

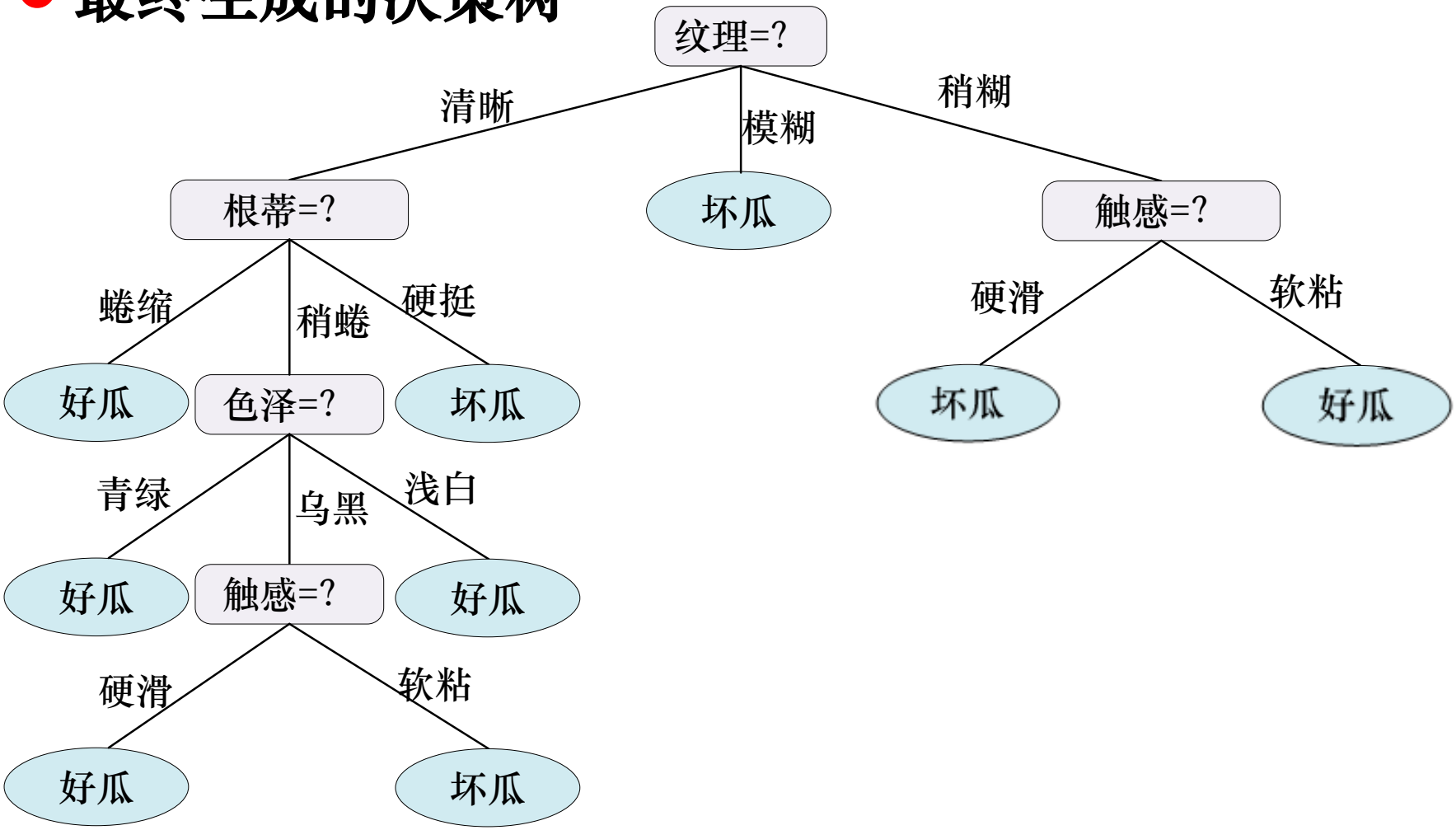
$G(D^{\text{清晰}}, \text{触感}) = 0.458$

$G(D^{\text{清晰}}, \text{根蒂}) = 0.458$

$G(D^{\text{清晰}}, \text{脐部}) = 0.458$

ID3算法-示例(1)

- 最终生成的决策树



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)，预测这位客人是属于“买”计算机的一类，还是属于“不买”计算机的一类?

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

$|C_1|$(买)=641

$|C_2|$ (不买)= 383

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

$P_1=641/1024=0.6260$

$P_2=383/1024=0.3740$

$$H(D) = -P_1\mathrm{Log}_2P_1 - P_2\mathrm{Log}_2P_2$$
$$= -(P_1\mathrm{Log}_2P_1 + P_2\mathrm{Log}_2P_2)$$
$$= 0.9537$$

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
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

$|D_{11}(买)|=128$
$|D_{12}(不买)|= 256$
$|D_1|=|D_{11}|+|D_{12}|=384$

$P_{11}=128/384$
$P_{12}=256/384$

H(青年|年龄)
$=-P_{11}\mathrm{Log}_2P_1-P_{12}\mathrm{Log}_2P_2$
$=-(P_{11}\mathrm{Log}_2P_{11}+P_{12}\mathrm{Log}_2P_{12})$
$=0.9183$

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

$|D_{21}(买)|= 256$
$|D_{22}(不买)|= 0$
$|D_2|=|D_{21}|+|D_{22}|=256$

$P_{21}=256/256$
$P_{22}=0/256$

$H(中年|年龄)$
$=-P_{21}\mathrm{Log}_2P_{21}-P_{22}\mathrm{Log}_2P_{22}$
$=-(P_{21}\mathrm{Log}_2P_{21}+P_{22}\mathrm{Log}_2P_{22})$
$=0$

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125/127

$|D_{31}(买)|= 125$
$|D_{32}(不买)|= 127$
$|D_3|=|D_{31}|+|D_{32}|=252$

$P_{31}=125/252$
$P_{32}=127/252$

$H(老年|年龄)$
$=-P_{31}\mathrm{Log}_2P_{31}-P_{32}\mathrm{Log}_2P_{32}$
$=-(P_{31}\mathrm{Log}_2P_{31}+P_{32}\mathrm{Log}_2P_{32})$
$=0.9157$

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:计算年龄的信息增益

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
H(年龄)=0.375*0.9183+
0.25*0+
0.375*0.9157
=0.6877
G(年龄信息增益)
=0.9537-0.6877
=0.2660 (1)

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:计算收入的信息增益

收入共分三个组:
高、中、低
H(收入)=0.9361
G(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:计算学生的信息增益

学生共分二个组：
学生、非学生
H(学生)=0.7811
G(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:计算信誉的信息增益

信誉分二个组:
良好、优秀
H(信誉)= 0.9048
G(信誉信息增益)=
0.9537-0.9048=0.0453 (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:选择结点

年龄信息增益=
0.9537-0.6877=0.2660 (1)

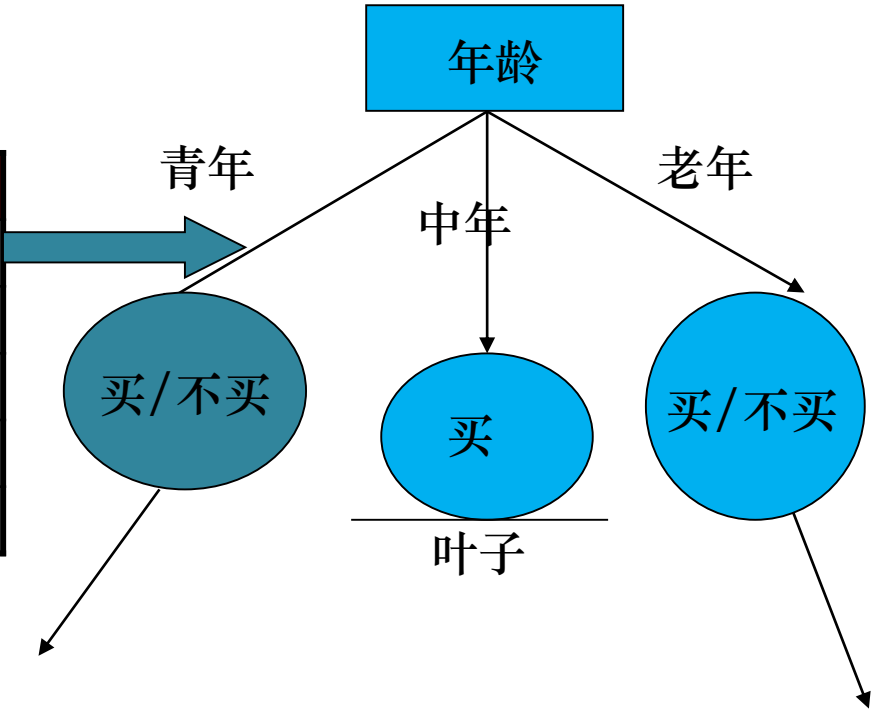
收入信息增益=
0.9537-0.9361=0.0176 (2)

学生信息增益=
0.9537-0.7811=0.1726 (3)

信誉信息增益=
0.9537-0.9048=0.0453 (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)

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

青年买与不买比例为128/256

$|D_{11}(买)|=128$
$|D_{12}(不买)|= 256$
$|D_1|=|D_{11}|+|D_{12}|=384$

$P_{11}=128/384$
$P_{12}=256/384$

H(青年|年龄)
$=-P_{11}\mathrm{Log}_2P_1-P_{12}\mathrm{Log}_2P_2$
$=-(P_{11}\mathrm{Log}_2P_{11}+P_{12}\mathrm{Log}_2P_{12})$
$=0.9183$

ID3算法-示例(2)

- 如果选择收入作为结点，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(高)=0
比例: 28/384=0.3333

H(中)=0.9183
比例: 192/384=0.5

H(低)=0
比例: 64/384=0.1667

经验条件熵(加权总和):

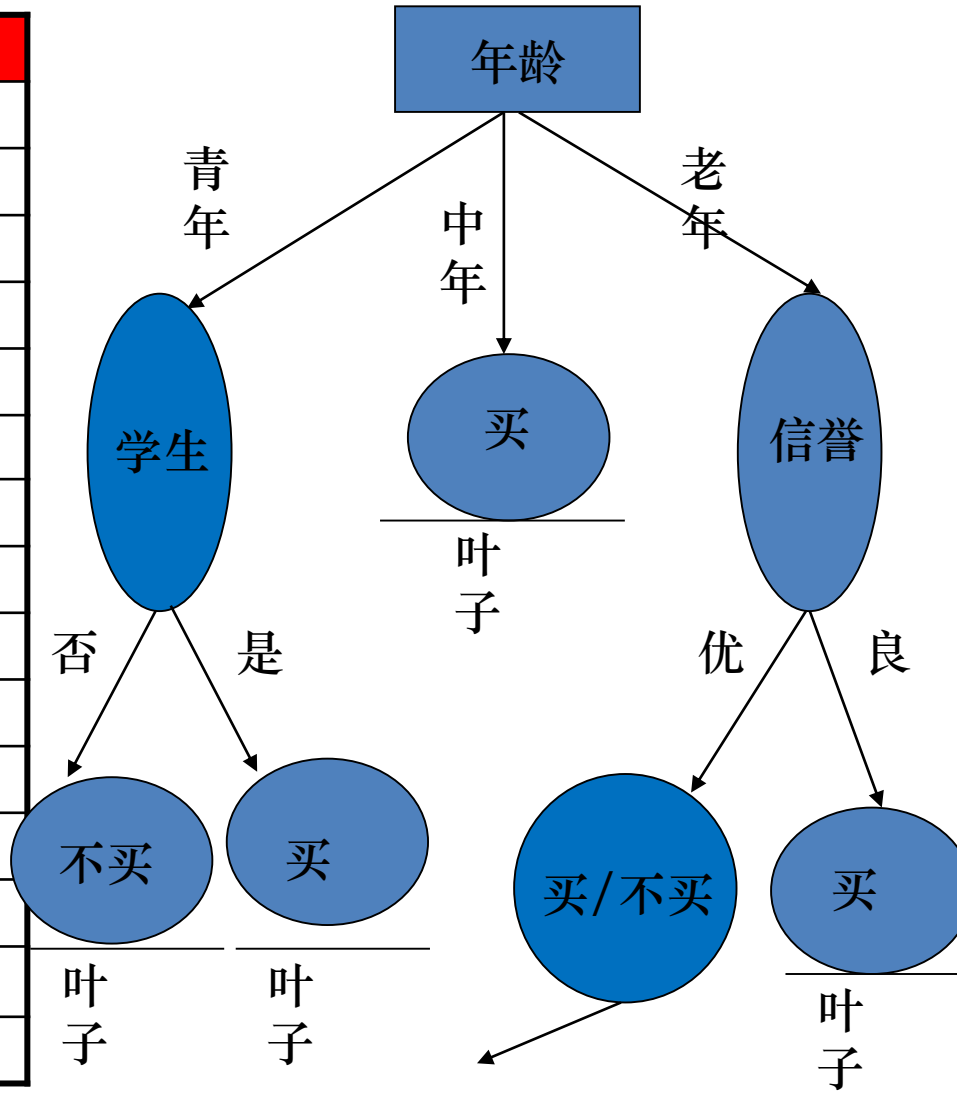
H(收入) = 0.3333 * 0 + 0.5 * 0.9183 + 0.1667 * 0 = 0.4592

收入的信息增益

Gain(收入) = H(青年) - H(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



ID3算法

- 算法优点
  - 只需对训练实例进行较好地标注，就能进行学习，从一类无序、无规则事物(概念)中推理出分类规则.
  - 分类模型是树状结构，简单直观，可将决策树中到达每个叶结点的路径转换为IF—THEN形式的分类规则，比较符合人类的理解方式.

ID3算法

- ID3算法局限性
  - 信息增益偏好取值多的属性(极限趋近于均匀分布)
  - 可能会受噪声或小样本影响，易出现过拟合问题
  - 无法处理连续值的属性
  - 无法处理属性值不完整的训练数据
  - 无法处理不同代价的属性

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.
  - 取值更多的属性容易使得数据更“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/结点；结果训练出来的形状是一棵庞大且深度很浅的树，这样的划分极不合理.

属性筛选度量标准

- **信息增益率(Information 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；因此采用信息增益率，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算法就采用增益率替代了ID3算法的信息增益

属性筛选度量标准

- **基尼指数(Gini Index)**

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

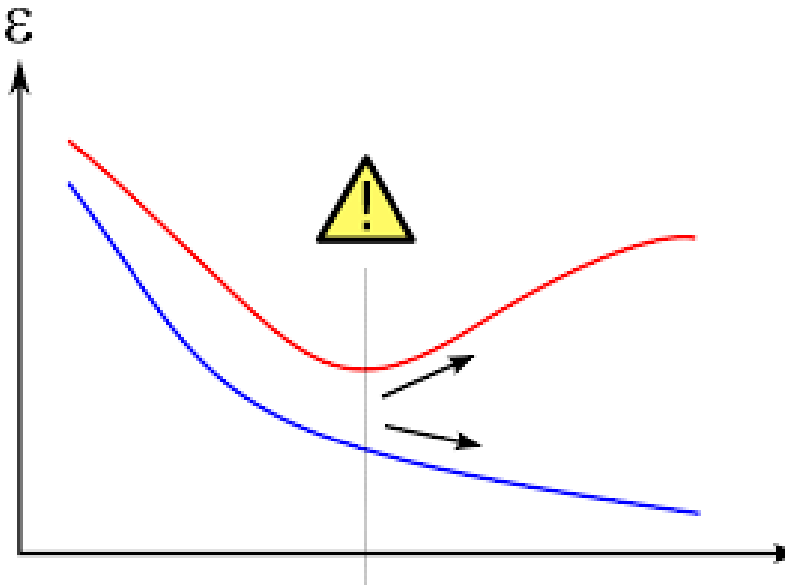
直观反映了从数据集中随机抽取两个样本，其类别不一致的概率；基尼指数越小，数据集的纯度越高.

- **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，同时拟合了数据和噪音，影响分类效果.
  - 叶结点样本太少，易出现耦合的规律性，使一些属性恰巧可很好地分类，但却与实际目标函数无关.



剪枝处理(Pruning)

- 针对过拟合问题
  - 剪枝是主要手段
- 基本策略
  - 预剪枝策略(Pre-pruning)：决策树生成过程中，对每个结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结点.
  - 后剪枝策略(Post-pruning)：先利用训练集生成决策树，自底向上对非叶结点进行考察，若将该结点对应子树替换为叶结点能带来泛化性能提升，则将该子树替换为叶结点.

剪枝处理(Pruning)

训练样本

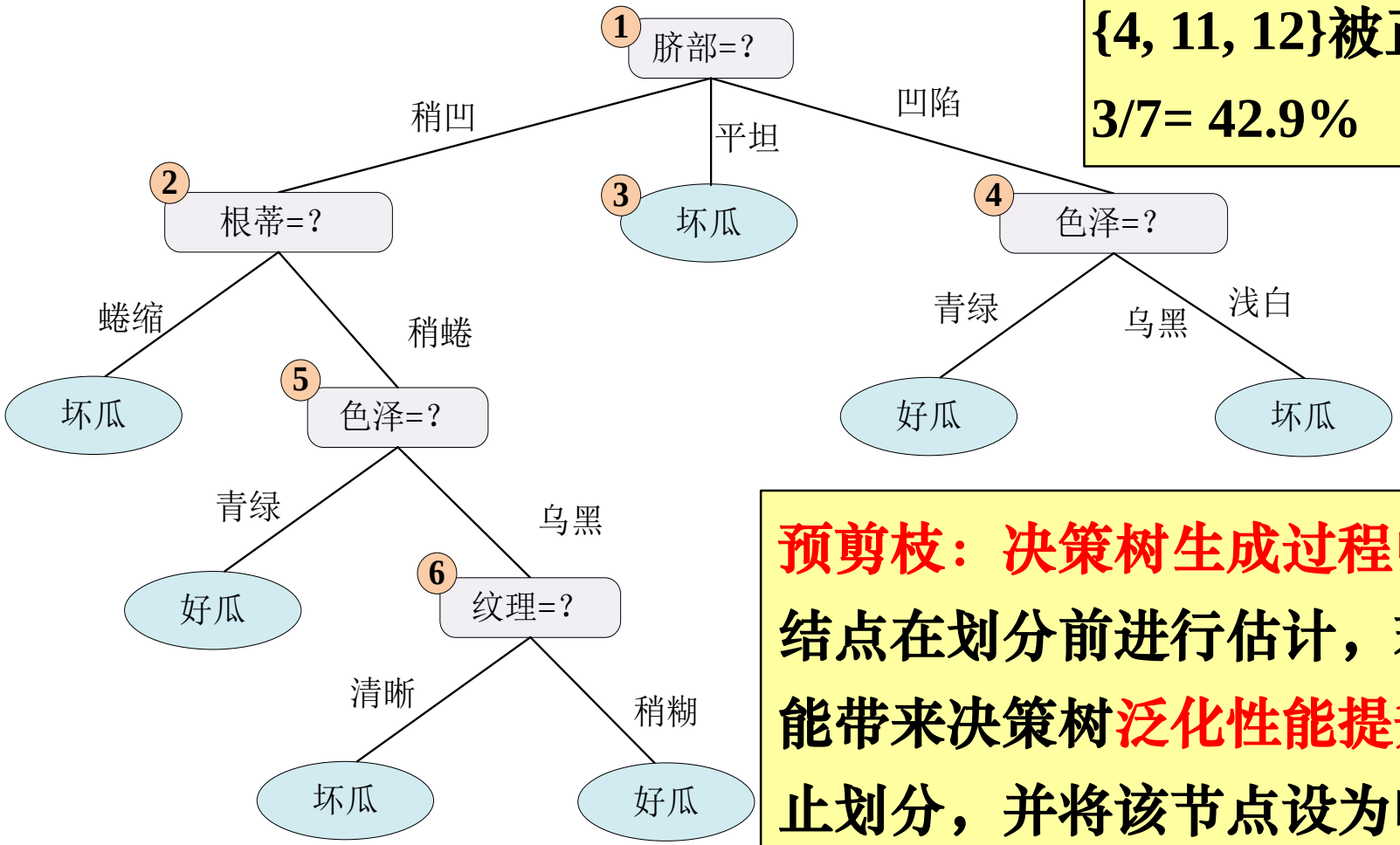
编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否

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，对各结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，并将该节点设为叶结点.

预剪枝算法

- **第一步：评估结点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：
    - 结点2：稍凹{6, 7, 15, 17}　　“好瓜”
    - 结点3：平坦{10, 16}　　“好瓜”
    - 结点4：凹陷{1, 2, 3, 14}　　“坏瓜”
    - 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：评估结点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：评估结点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”

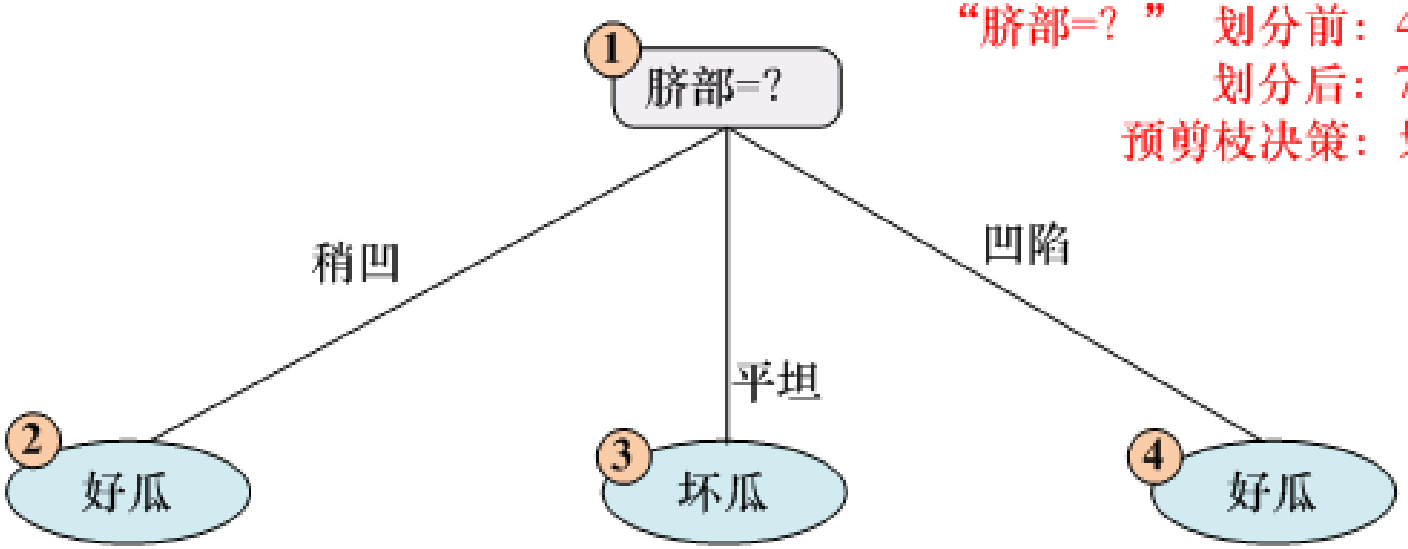
    不划分： {4, 5, 8, 11, 12}被正确分类 **5/7= 71.4%**

    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



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

测试集精度
“脐部=？” 划分前：71.4%
划分后：57.1%
预剪枝决策：不划分

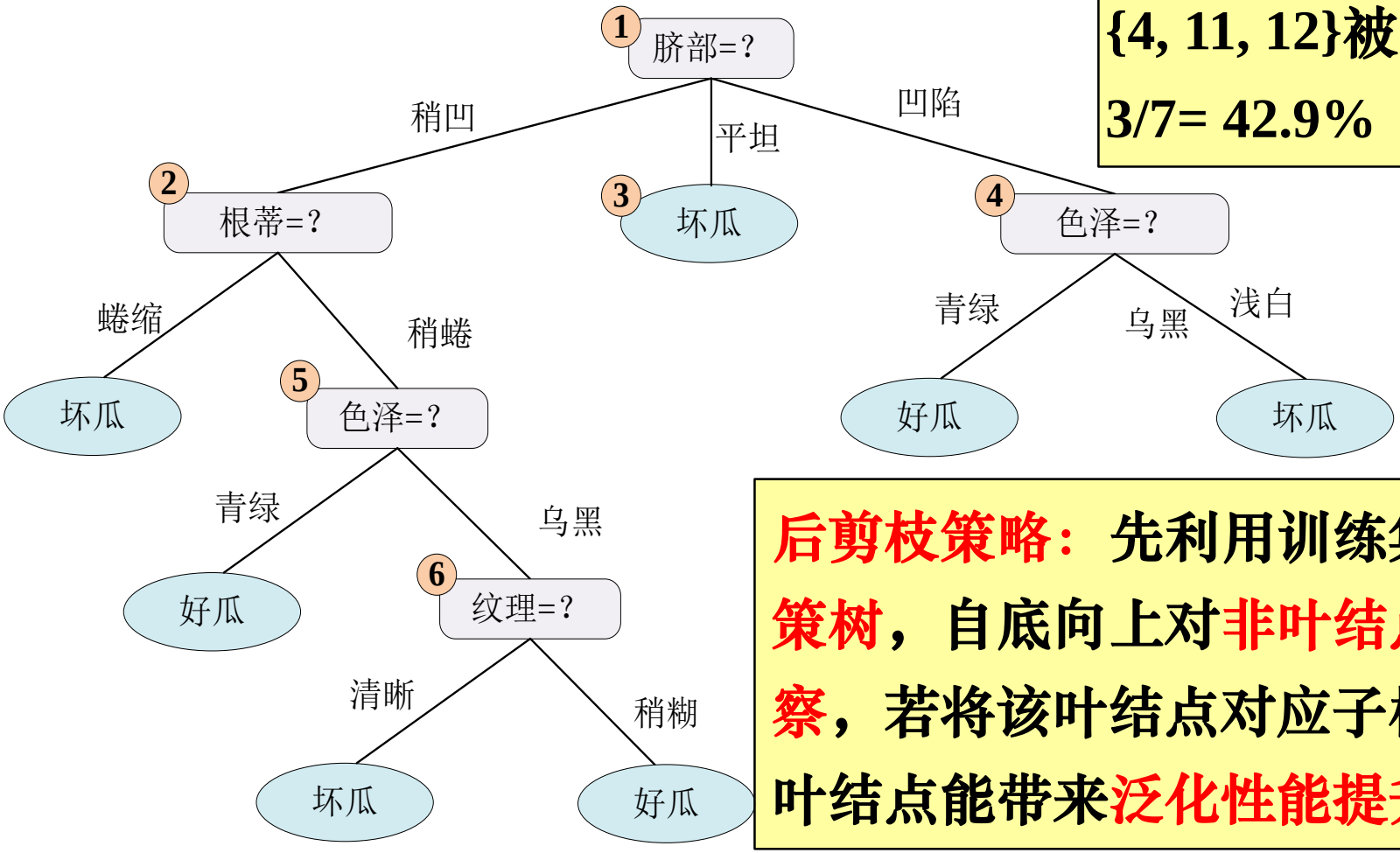
测试集精度
“色泽=？” 划分前：71.4%
划分后：71.4%
预剪枝决策：不划分

预剪枝算法

- **策略特点**
  - **优势**：“剪掉”很多没必要展开的分支，降低了过拟合风险，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，自底向上对非叶结点进行考察，若将该叶结点对应子树替换为叶结点能带来泛化性能提升，替换.

后剪枝算法

- **第一步：评估结点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替换为叶结点，“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：评估结点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替换为叶结点，“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：评估结点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：

  - 把结点4替换为叶结点，“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：评估结点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：

  - 把结点2替换为叶结点，“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：评估结点1**

  剪枝前：

  - 泛化性能：同上一步 **5/7= 71.4%**

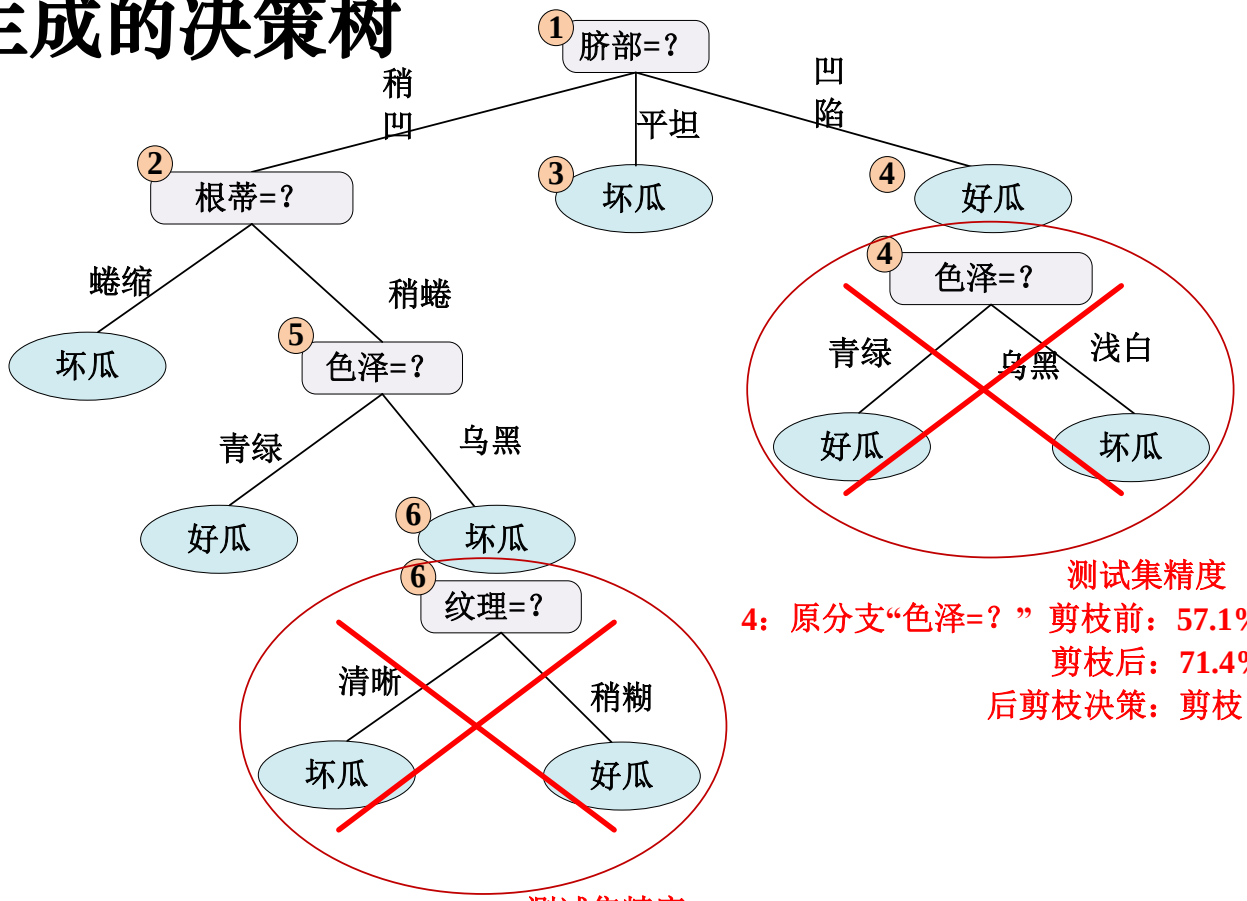
  剪枝后：

  - 把结点1替换为叶结点
  - 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

- **策略特点**
  - **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再进行，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.