机器学习—决策树

Decision Tree

授课对象：计算机科学与技术专业 二年级

课程名称：人工智能（专业必修）

节选内容：第六章 机器学习

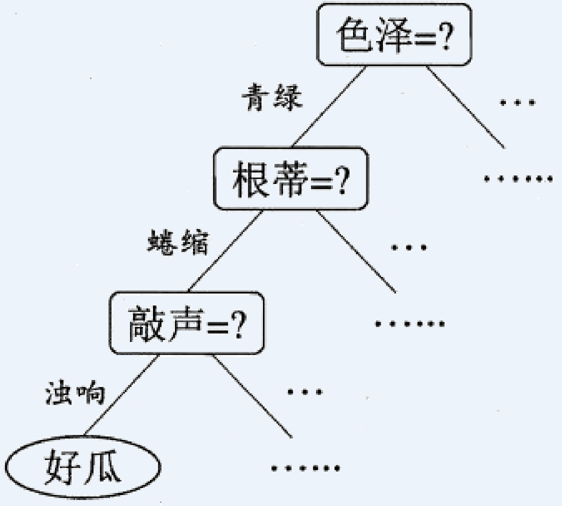
课程学分：3学分

什么是决策树?

- 一种树状结构的分类模型
  - 每个“内部结点”对应于某个属性上的“测试”(test)
  - 每个分支对应于该测试的一种可能结果（即该属性的某个取值)
  - 每个“叶结点”对应于一个“预测结果”

- **学习过程**：通过对训练样本的分析来确定“划分属性” （即内部结点所对应的属性)

- **预测过程**：将测试示例从根结点开始，沿着划分属性所构成的“判定测试序列”下行，直到叶结点



西瓜问题的一颗决策树

什么是决策树?

年龄	收入	学生?	信用等级?	是否买电脑
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

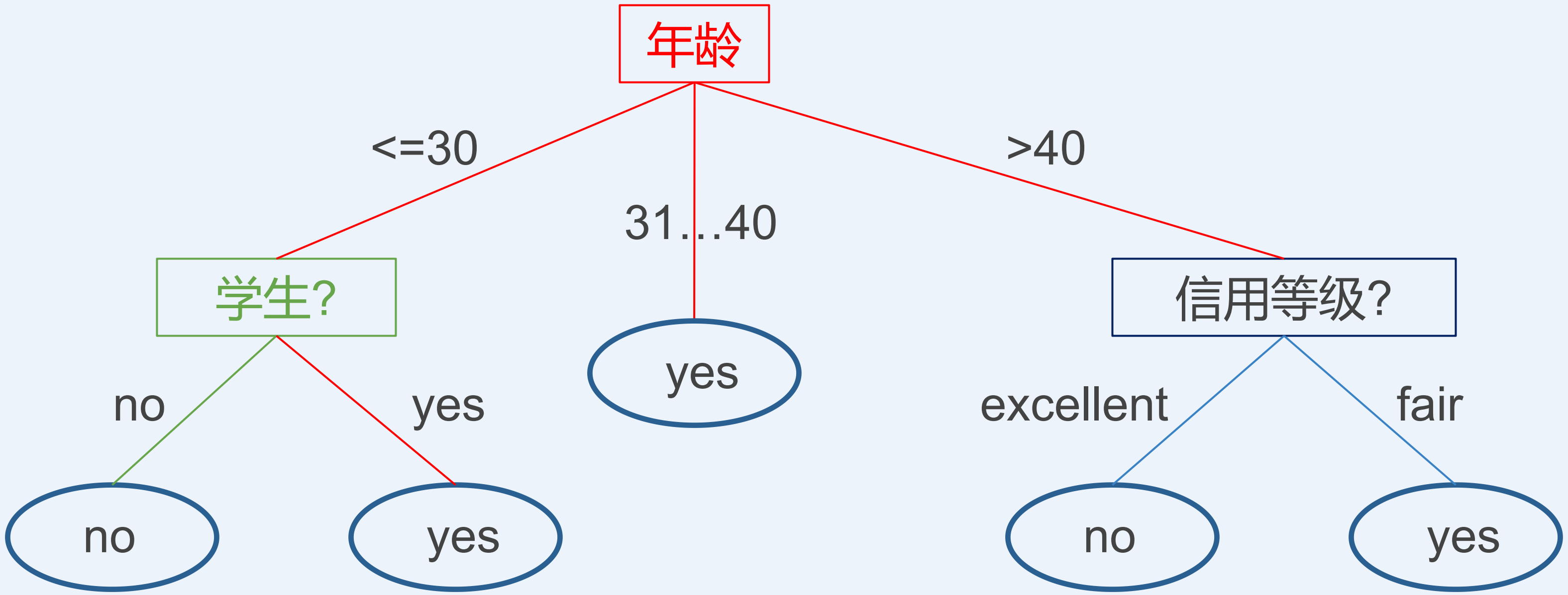
什么是决策树?

年龄	收入	学生?	信用等级?	是否买电脑
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

什么是决策树?

年龄	收入	学生?	信用等级?	是否买电脑
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

什么是决策树?



什么是决策树?

- 建模阶段
  - Tree construction (建树)
    - 首先，所有训练样本都位于根节点位置
    - 基于一定的指标(如：信息增益，基尼系数等)选择属性
    - 根据选择的属性，递归地划分训练样本
  - Tree pruning (剪枝)
    - 识别并删除异常值和噪声影响较大的分支
- 预测阶段
  - 使用构建的树模型预测未知样本

决策树算法

输入： 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$;

属性集 $A=\{a_1,a_2,\ldots,a_d\}$.

过程： 函数 TreeGenerate$(D,\ A)$

生成结点 node;

if D 中样本全属于同一类别 C **then**

将 node 标记为 C 类叶结点; **return**

end if

递归返回，情形(1)

if $A=\varnothing$ **OR** D 中样本在 A 上取值相同 **then**

将 node 标记为叶结点，其类别标记为 D 中样本数最多的类; **return**

end if

递归返回，情形(2)

利用当前结点的后验分布

从 A 中选择最优划分属性 a_*;

for a_* 的每一个值 a_*^v **do**

为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;

if D_v 为空 **then**

将分支结点标记为叶结点，其类别标记为 D 中样本最多的类; **return**

else

以 TreeGenerate$(D_v,\ A\setminus\{a_*\})$ 为分支结点

end if

end for

递归返回，情形(3)

将父结点的样本分布作为当前结点的先验分布

决策树算法的核心

输出： 以 node 为根结点的一棵决策树

决策树算法

- 基础算法(一种贪心算法)
  - 根据分治的思想，用自顶向下的方法，递归建树
  - 属性应当是离散的（如果是连续型数据，需要先进行离散化）
  - 首先，所有训练样本都位于根节点位置
  - 基于统计学指标或启发式的方法来对属性进行选择（如信息增益、基尼系数等）
  - 根据选择的属性，递归地划分训练样本
- 停止划分的条件
  - 被分到同一个节点内的所有样本都是同一个类别（相同label/class）
  - 所有的属性都已经被用于之前的划分，没有属性可以继续划分——采用多数投票的方法决定该叶子节点的列表
  - 无训练数据

决策树算法

- 与决策树相关的重要算法包括：
  - CLS, ID3，C4.5，CART
- 算法的发展过程
  - Hunt, Marin和Stone于1966年研制的CLS学习系统，用于学习单个概念。
  - 1979年, Quinlan给出ID3算法，并在1983年和1986年对ID3进行了总结和简化，使其成为决策树学习算法的典型。
  - Schlimmer和Fisher于1986年对ID3进行改造，在每个可能的决策树节点创建缓冲区，使决策树可以递增式生成，得到ID4算法。
  - 1988年，Utgoff在ID4基础上提出了ID5学习算法，进一步提高了效率。
  - 1993年，Quinlan 进一步发展了ID3算法，改进成C4.5算法。
  - 另一类决策树算法为CART，与C4.5不同的是，CART由二元逻辑问题生成，每个树节点只有两个分枝，分别包括学习实例的正例与反例。

CLS算法

- CLS（Concept Learning System）算法
  - CLS是早期的决策树学习算法。它是许多决策算法的基础
- CLS的基本思想
  - 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。

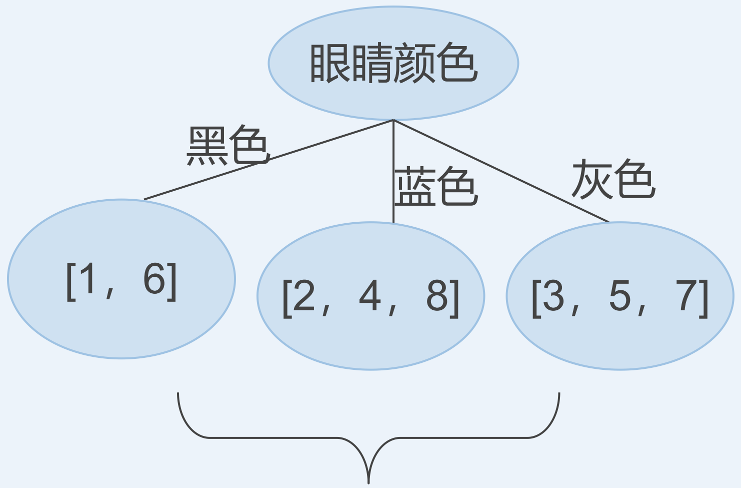
CLS算法

人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血

CLS算法

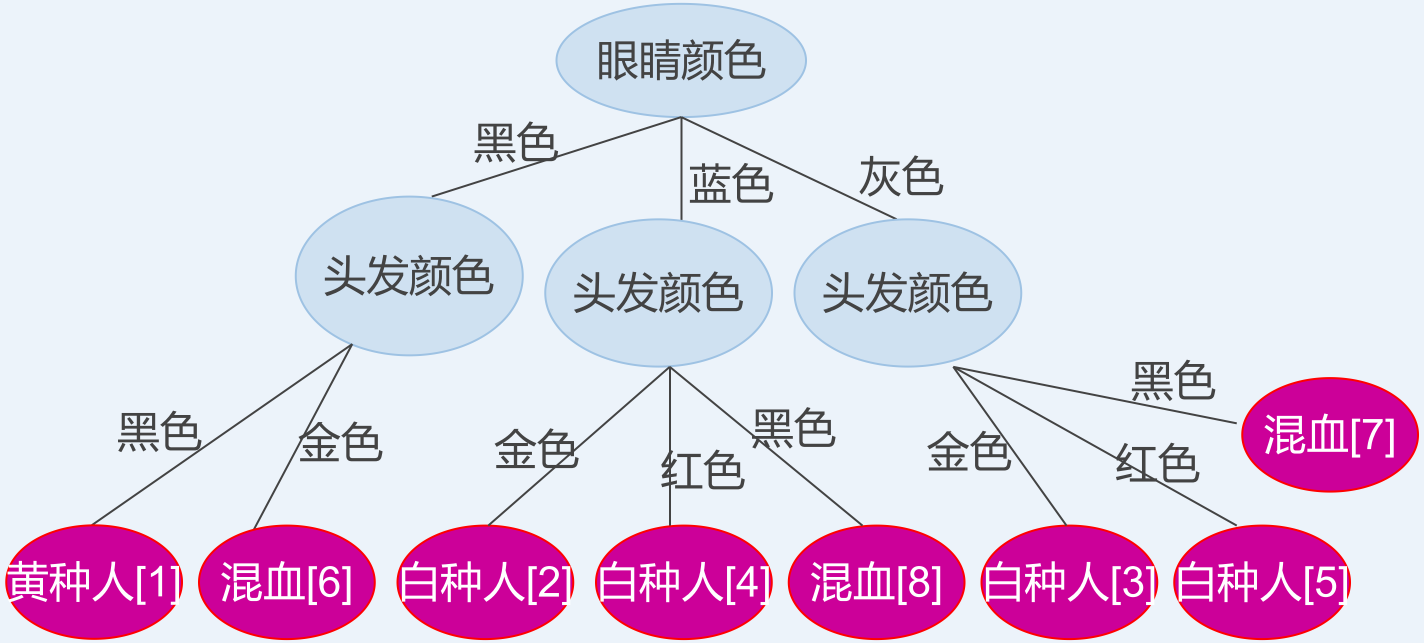
人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血

决策树的构建



不属于同一类，非叶结点

CLS算法



眼睛颜色
黑色
蓝色
灰色
头发颜色
头发颜色
头发颜色
黑色
金色
金色
红色
黑色
金色
红色
黑色
黄种人[1]
混血[6]
白种人[2]
白种人[4]
混血[8]
白种人[3]
白种人[5]
混血[7]

CLS算法

- 步骤：
  - 生成一颗空决策树和一个训练样本属性表；
  - 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v_1，v_2，…，v_m， 则根据 A 的取值的不同，将 T 划分成m个子集T_1，T_2，…，Tm；
  - 从训练样本属性表中删除属性A；
  - 对每个子集递归调用CLS

CLS算法有什么问题？

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。

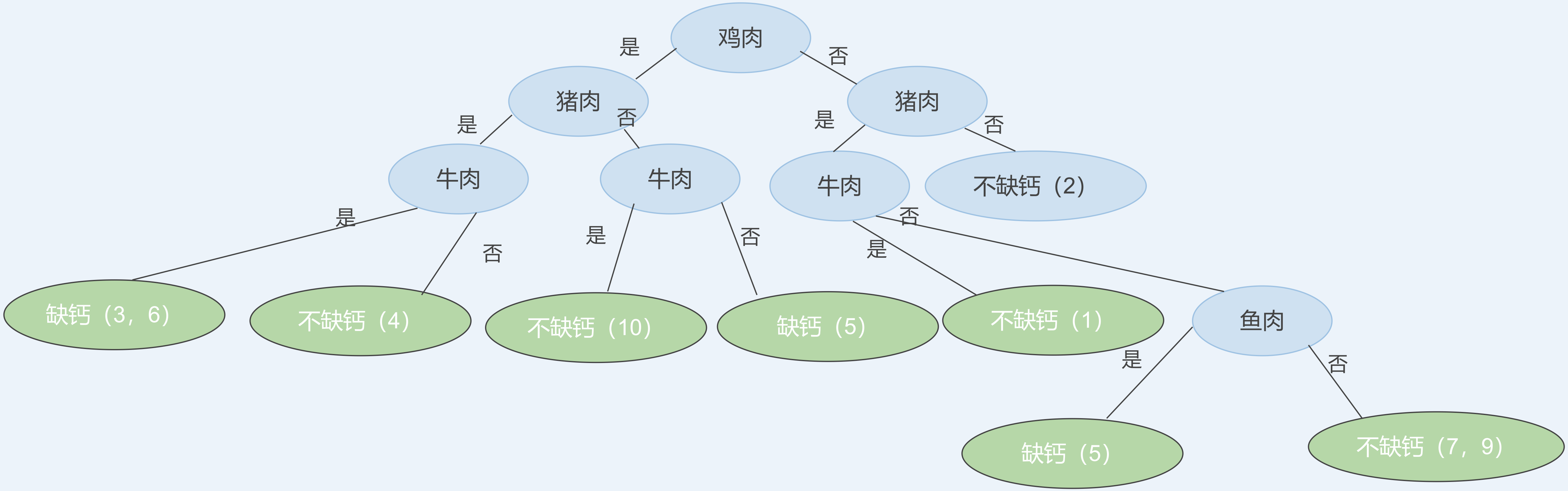
CLS算法

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	0	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

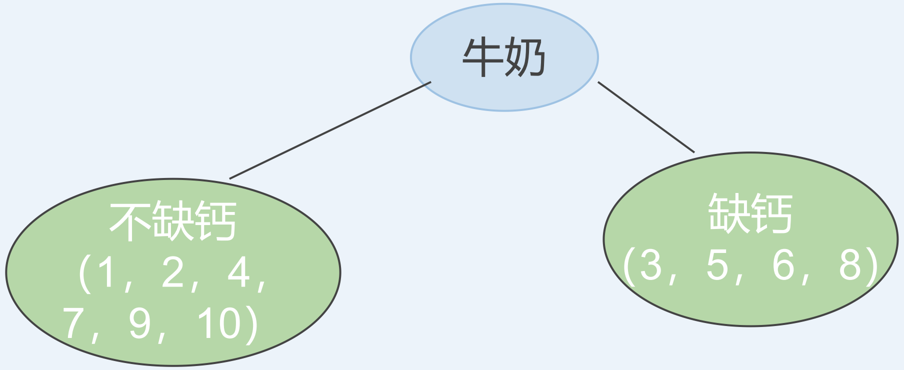
CLS算法

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！



CLS算法

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!



ID3算法

- 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?

ID3算法

如何衡量属性的重要性（importance）？

年龄	收入	学生?	信用等级?	是否买电脑
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

信息论

- 假设要为投掷一个8面骰子的结果进行编码. 需要多少个比特？（bit）

$$3bits = \log_2 8 = -\sum_{i=1}^{8} \frac{1}{8} \log_2 \frac{1}{8} = -\sum_{i=1}^{8} p(i) \log_2 p(i) = H(X)$$

- 如果我们希望将投掷这个8面骰子的结果通过某种方式发送给别人，最有效的方式就是将这一信息进行二进制编码 【000 – 111】

信息论

- Entropy (熵)
  - represent the expectation of uncertainty for a random variable (可以用来衡量离散变量的不确定性，如抛硬币、掷骰子)

$$H(X) = -\sum_{x \in X} p(x)\log_2 p(x)$$

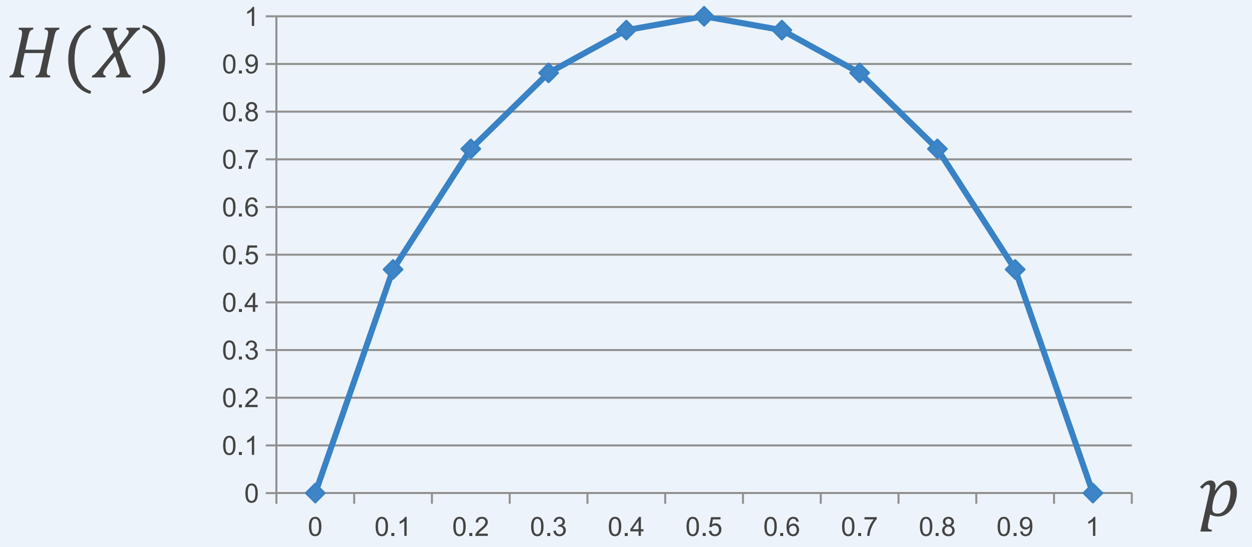
$$= \sum_{x \in X} p(x)\log_2 \frac{1}{p(x)}$$

$$= E\left(\log_2 \frac{1}{p(X)}\right)$$

信息论

- $P(X=1) = p,\ P(X=0) = 1-p$
- 假设抛一枚硬币，正面朝上的概率为p，反面朝上的概率为$1-p$，则抛这枚硬币所得结果的不确定性（熵值）是p的下述函数：

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



信息论

- 条件/联合熵

条件熵：

$$H(Y|X)=\sum_{x\in X}p(x)H(Y|X=x)$$
$$=\sum_{x\in X}p(x)\left[-\sum_{y\in Y}p(y|x)\log_2 p\,(y|x)\right]$$
$$=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2 p\,(y|x)$$
$$=-\sum_{x\in X}\sum_{y\in Y}p(x)p(y|x)\log_2 p\,(y|x)$$

联合熵：

$$H(X,Y)=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log_2 p\,(x,y)$$

信息论

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

两个离散变量X和Y的联合熵（即，联合出现的不确定性）

= X的熵 + 给定X，出现Y的条件熵

= X的不确定性 + 给定X，出现Y的不确定性

信息论

Mutual information (互信息)

因为：$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

所以：$H(Y) - H(Y|X) = H(X) - H(X|Y) = I(X;Y)$

两个离散变量X和Y的互信息$I(X;Y)$衡量的是这两个变量之间的相关度

一个连续变量X的不确定性，用方差$Var(X)$来度量
一个离散变量X的不确定性，用熵$H(X)$来度量
两个连续变量X和Y的相关度，用协方差或相关系数来度量
两个离散变量X和Y的相关度，用互信息$I(X;Y)$来度量

基于信息增益的ID3模型

- 类标签: 是否买电脑=“yes/no”
- 用字母D表示类标签，字母A表示每个属性
- 计算D (类标签) 和A (每个属性)的互信息

14个训练样本中，9个买了电脑

$$H(D) = -\frac{9}{14}\log_2\frac{9}{14} - \left(1-\frac{9}{14}\right)\log_2\left(1-\frac{9}{14}\right) = 0.940$$

$$H(D|A=\text{"年龄"}) = \frac{5}{14}\times\left(-\frac{2}{5}\log_2\frac{2}{5}-\frac{3}{5}\log_2\frac{3}{5}\right)$$

$$+\frac{4}{14}\times\left(-\frac{4}{4}\log_2\frac{4}{4}-\frac{0}{4}\log_2\frac{0}{4}\right)+\frac{5}{14}\times\left(-\frac{3}{5}\log_2\frac{3}{5}-\frac{2}{5}\log_2\frac{2}{5}\right)=0.694$$

基于信息增益的ID3模型

- 类标签: 是否买电脑=“yes/no”
- 用字母D表示类标签，字母A表示每个属性
- 计算D (类标签) 和A (每个属性)的互信息
- $H(D) = 0.940$
- $H(D|A = \text{"年龄"}) = 0.694$

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

- $g(D, A = \text{“年龄”}) = 0.246$
- $g(D, A = \text{“收入”}) = 0.029$
- $g(D, A = \text{“学生?”}) = 0.151$
- $g(D, A = \text{“信用等级?”}) = 0.048$

“年龄”这个属性的条件熵最小（等价于信息增益最大），因而首先被选出作为根节点

基于信息增益的ID3模型

对于下述数据集，采用ID3算法会得到哪个属性最重要？

用户ID	年龄	收入	学生?	信用等级?	是否买电脑
u1	<=30	high	no	fair	no
u2	<=30	high	no	excellent	no
u3	31…40	high	no	fair	yes
u4	>40	medium	no	fair	yes
u5	>40	low	yes	fair	yes
u6	>40	low	yes	excellent	no
u7	31…40	low	yes	excellent	yes
u8	<=30	medium	no	fair	no
u9	<=30	low	yes	fair	yes
u10	>40	medium	yes	fair	yes
u11	<=30	medium	yes	excellent	yes
u12	31…40	medium	no	excellent	yes
u13	31…40	high	yes	fair	yes
u14	>40	medium	no	excellent	no

基于增益率的C4.5模型

- 信息增益（Information gain）的衡量容易偏向那些有大量值的属性
- C4.5 (ID3的一个改进版) 使用了增益率（Gain ratio）克服上述问题（对信息增益正则化）
- 每次选取最大增益率的属性进行划分

基于增益率的C4.5模型

- $GainRatio_A(D) = Gain_A(D)/SplitInfo_A(D)$

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- $GainRatio_{A=\text{“}income\text{”}}(D) = ?$

$$SplitInfo_{A=\text{“}income\text{”}}(D)$$
$$= -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right)$$
$$= 0.926$$

- $GainRatio_{A=\text{“}income\text{”}}(D) = 0.029/0.926 = 0.031$

基于Gini指数的CART模型

- 如果一个数据集D包含来自n个类的样本，那么基尼指数，$gini(D)$定义如下：

$$gini(D) = \sum_{j=1}^{n} p_j(1 - p_j) = 1 - \sum_{j=1}^{n} p_j^2$$

p_j 是类j在D中的相对频率。

- 如果$n = 2$，那么$gini(D) = 2p(1 - p)$

基于Gini指数的CART模型

- 如果一个数据集 D 被分成两个子集 D_1和 D_2 大小分别为 N_1 和N_2，数据包含来自 n 个类的样本，则基尼指数 $gini_{split}(D)$ 定义如下

$$gini_{split}(D) = \frac{N_1}{N} gini(D_1) + \frac{N_2}{N} gini(D_2)$$

- 具有最小 $gini_{split}(D)$的属性被选为分裂节点的属性 (对每个属性，需要遍历所有可能的分裂位置点).

基于Gini指数的CART模型

- 在“是否买电脑”中，D有9个样本 “是” 5个样本 “否”

$$gini(D) = 1 - (\frac{9}{14})^2 - (\frac{5}{14})^2 = 0.459$$

- 属性“收入”将D分成：10个在D_1: $\{\text{medium}, \text{high}\}$ 以及4个在D_2：$\{\text{low}\}$

$$gini_{income\in\{\text{medium,high}\}}(D) = \frac{10}{14}gini(D_1) + \frac{4}{14}gini(D_2)$$
$$= \frac{10}{14}\left(1 - (\frac{6}{10})^2 - (\frac{4}{10})^2\right) + \frac{4}{14}\left(1 - (\frac{1}{4})^2 - (\frac{3}{4})^2\right)$$
$$= 0.450 = gini_{income\in\{\text{low}\}}(D)$$

连续型属性的处理

- 应该如何计算具有连续值属性的信息增益或基尼指数?
  - 给定A的v个值, 那么有$v-1$个可能的分裂位置。比如在A中，a_i和a_{i+1}的中点是

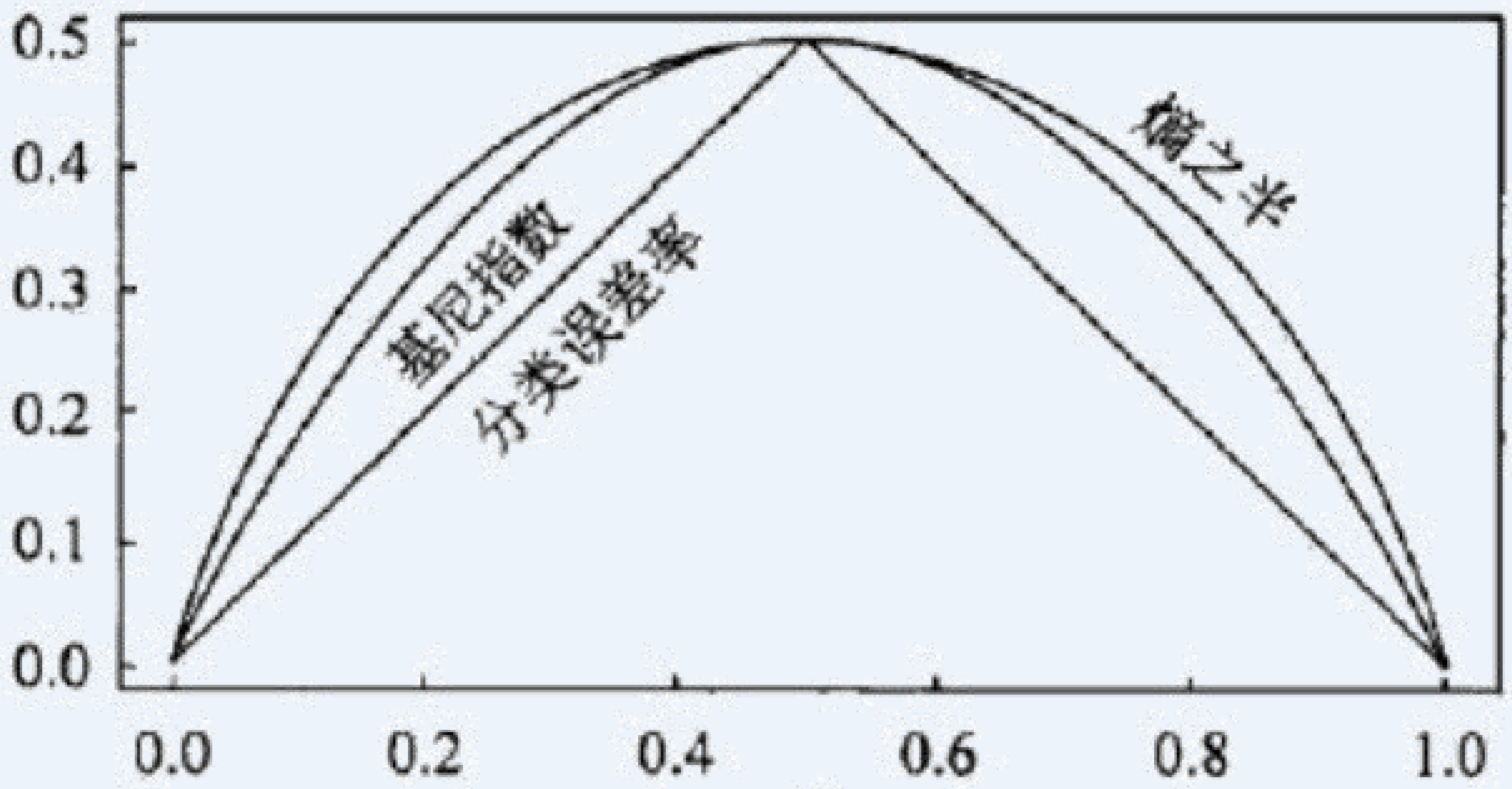
$$(a_i + a_{i+1})/2$$

基于Gini指数的CART模型

CART（分类树）的生成算法

- 输入：训练数据集D，停止计算条件
- 输出：CART分类树
- 从根节点开始，递归对每个结点操作
  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D_1，$D2$，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和2步骤
  4. 生成CART树

分类树



生成分类规则

- 将知识表示为 **IF-THEN** 形式的规则。
- 对每条从根节点到叶子节点的路径，创建一条规则。
- 从一个节点到下一层节点的一条分支上，每个属性-值对可以形成一个连接。
- 叶子节点代表预测的分类。
- 规则应当容易被人理解 (可解释性)。

CART（回归树）的生成

- 设y是连续变量，给定训练数据集：$D=\{(x_1,y_1),(x_2,y_2),...,(x_N,y_N)\}$

- 假设已将输入空间划分为M个单元R1,R2...Rm，并且每个单元Rm上有一个固定的输出cm，回归树表示为：

$$f(x)=\sum_{m=1}^{M}c_m I(x\in R_m)$$

- 平方误差来表示预测误差，用平方误差最小准则求解每个单元上的最优输出值:

$$\sum_{x_i\in R_m}\big(y_i-f(x_i)\big)^2$$

- R_m上的c_m的最优值：$\hat{c}_m=ave(y_i|x_i\in R_m)$

CART（回归树）的生成

- 问题：如何对输入空间进行划分？
- 启发式：选择第j个变量$x^{(j)}$和它取的值s，作为切分变量和切分点，定义两个区域：

$$R_1(j,s)=\{x|x^{(j)}\leq s\}和R_2(j,s)=\{x|x^{(j)}>s\}$$

- 然后寻找最优切分变量和切分点：

$$\min_{j,s}[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_1(j,s)}(y_i-c_2)^2]$$

- 且：$\hat{c}_1=ave(y_i|x_i\in R_1(j,s))$和$\hat{c}_2=ave(y_i|x_i\in R_2(j,s))$
- 再对两个区域重复上述划分，直到满足停止条件。

CART（回归树）的生成

- 输入：训练数据集D；
- 输出：回归树$f(x)$
- 在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉决策树：
  1. 选择最优切分变量j与切分点s，求解：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_1(j,s)}(y_i-c_2)^2\right]$$

     遍历变量j，对固定的切分变量j扫描切分点s，选择使上式达到最小值

CART（回归树）的生成

- 输入：训练数据集D；
- 输出：回归树$f(x)$
- 在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉决策树：

2. 用选定的对$(j\ \ ,s\ \ )$划分区域并决定相应的输出值：

$R_1(j,s)=\{x \mid x^{(j)}\le s\}$，$R_2(j,s)=\{x \mid x^{(j)}> s\}$，

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)} y_i\text{，}\quad x\in R_m\text{，}\quad m=1,2$$

3. 继续对两个子区域调用步骤1、2，直至满足停止条件
4. 将输入空间划分为M个区域$R_1,R_2,\dots,R_M$，生成决策树：

$$f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$$

实验方法

- 分出训练集、（验证集）和测试集
- 使用交叉验证，比如，k折交叉验证
  - 把数据集分成k部分
  - 选取$(k-1)$部分用于训练，在剩下那部分上测试
  - 重复k次，使得每个部分都测试过一次

谢谢大家！