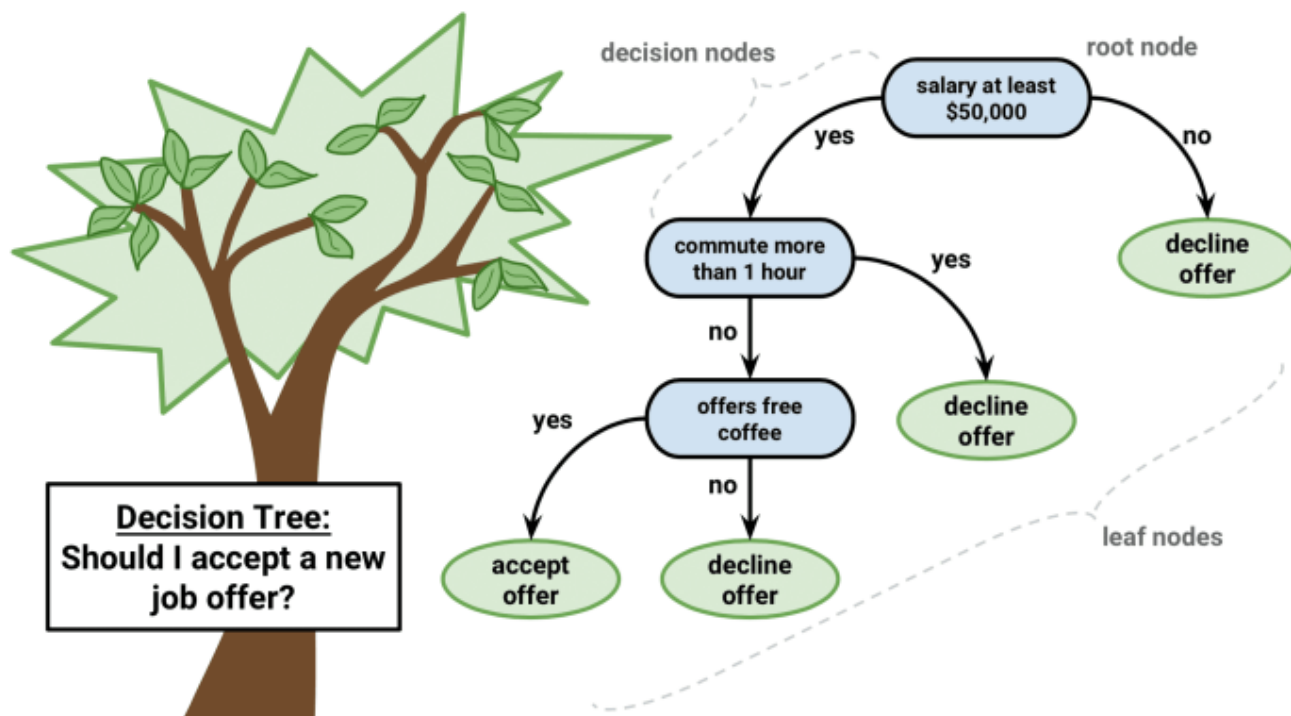


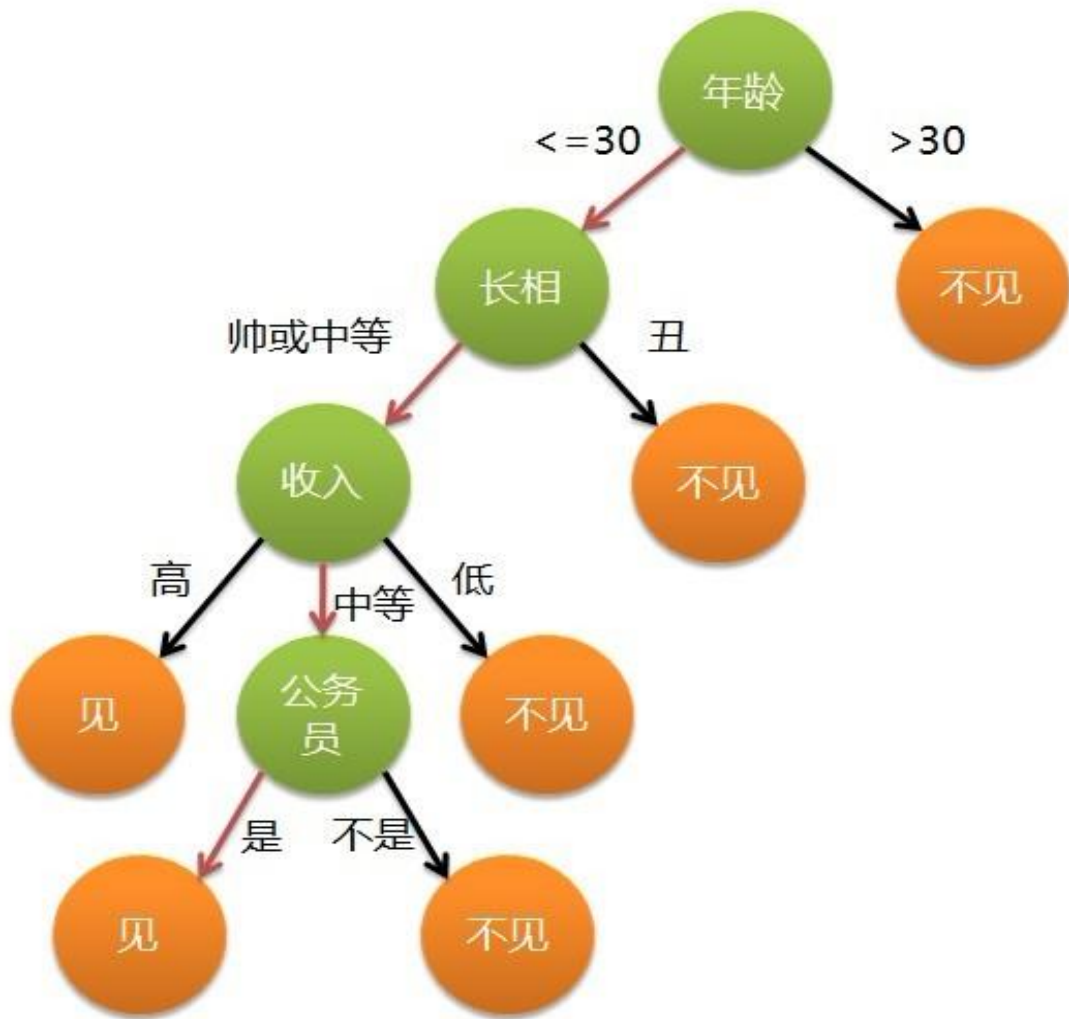
决策树



赵海涛

haitaozhao@ecust.edu.cn

开课一张图



父母：多大年纪了？

女儿：25。

父母：长的帅不帅？

女儿：挺帅的。

父母：收入高不？

女儿：中等情况。

父母：是公务员不？

女儿：是，在税务局上班呢。

父母：好，可以。

决策树

关于分类问题

名称	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
海龟	冷血	鳞片	否	半	否	是	否	爬行类
鸽子	恒温	羽毛	否	否	是	是	否	鸟类
鲸	恒温	毛发	是	是	否	否	否	哺乳类

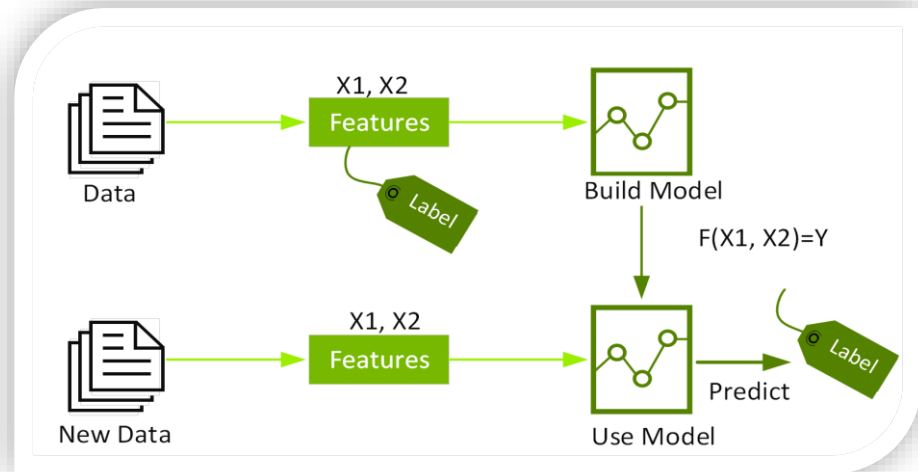
X

y

分类与回归

分类目标属性 y 是离散的，回归目标属性 y 是连续的

决策树-解决分类问题的一般方法



通过以上对分类问题一般方法的描述，可以看出分类问题一般包括两个步骤：

- 1、模型构建（**归纳**）
通过对训练集的归纳，建立分类模型。
- 2、预测应用（**推论**）
根据建立的分类模型，对测试集进行测试。

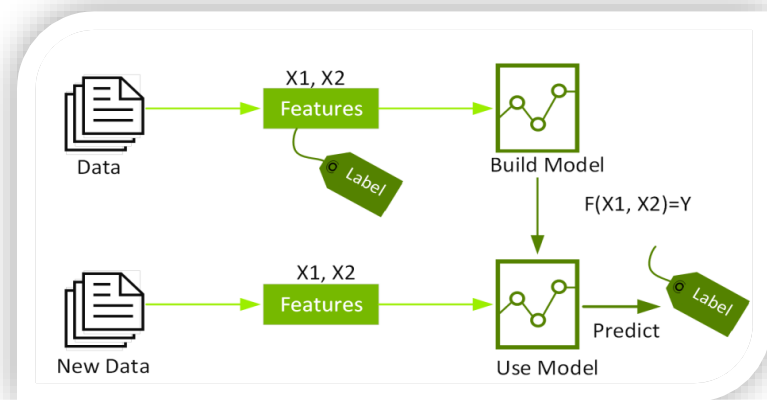
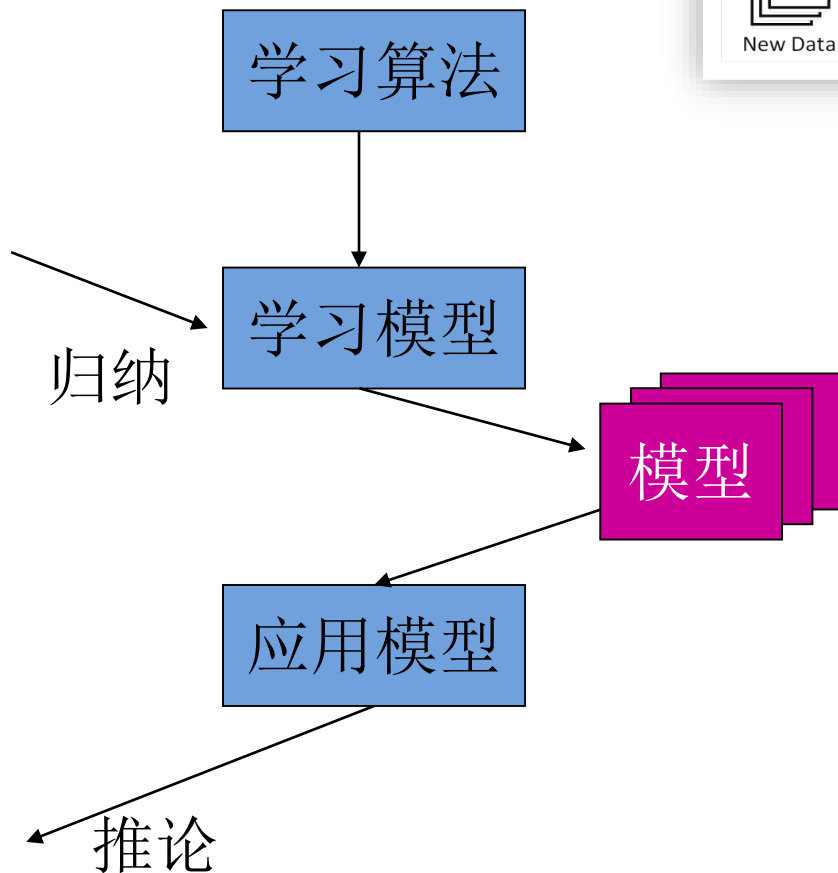
决策树-解决分类问题的一般方法

训练集（类标号已知）

TID	A1	A2	A3	类
1	Y	100	L	N
2	N	125	S	N
3	Y	400	L	Y
4	N	415	M	N

检验集（类标号未知）

TID	A1	A2	A3	类
1	Y	100	L	?
2	N	125	S	?
3	Y	400	L	?
4	N	415	M	?



决策树

- 决策树是一种典型的分类方法
 - 首先对数据进行处理，利用归纳算法生成可读的规则和决策树，
 - 然后使用决策对新数据进行分析。
- 本质上决策树是通过一系列规则对数据进行分类的过程。

决策树和归纳算法

- 归纳学习由于依赖于检验数据，因此又称为检验学习。
- 归纳学习存在一个**基本的假设**：
 - 任一假设如果能够在足够大的训练样本集中很好的逼近目标函数，则它也能在未见样本中很好地逼近目标函数。该假定是归纳学习的有效性的前提条件。

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, J.R. 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的决策树由二元逻辑问题生成, 每个树节点只有两个分枝。

决策树算法

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？

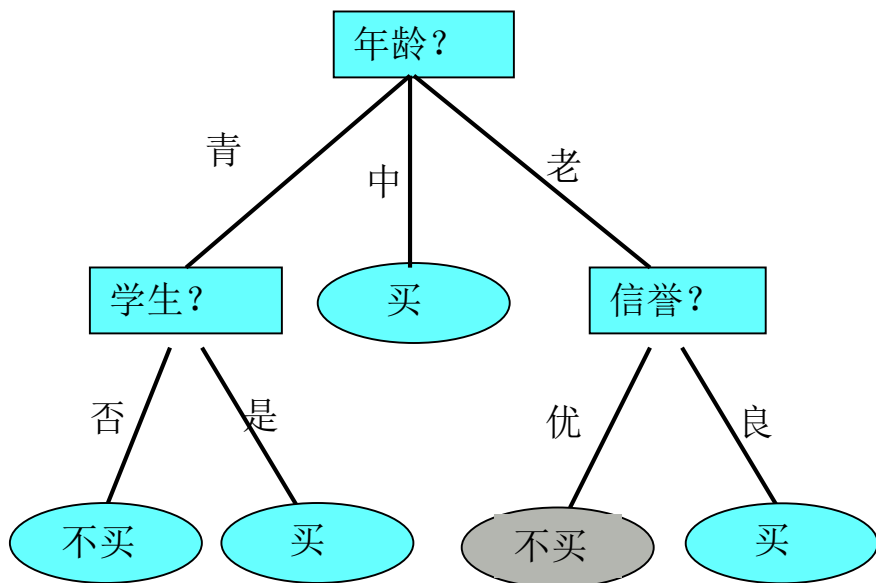
即：你能预测这位客人是属于“买”计算机的那一类，还是属于“不买”计算机的那一类？

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

决策树算法

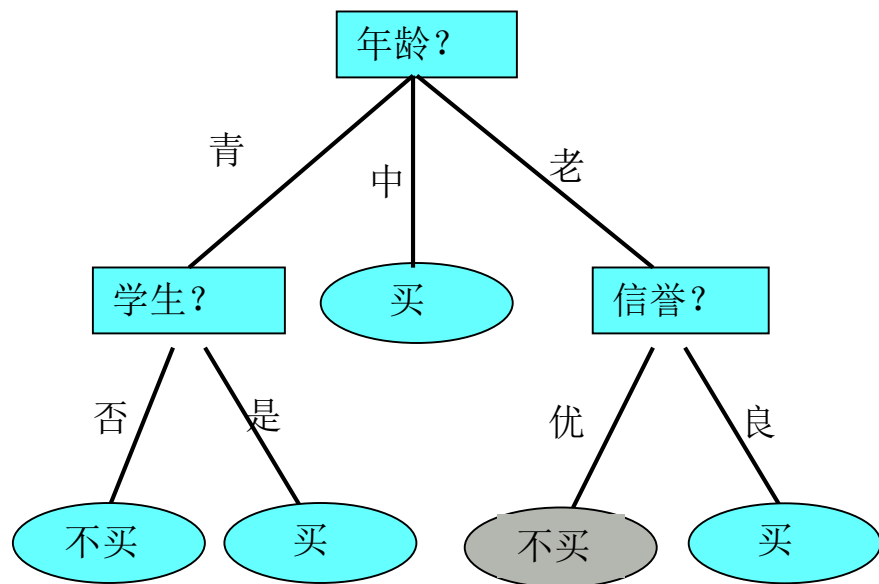
谁在买计算机？



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

决策树算法

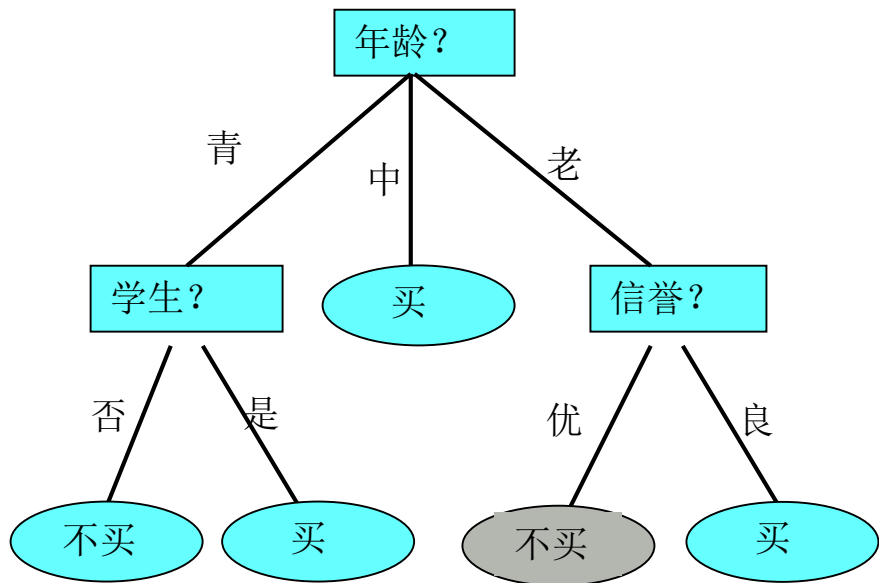
谁在买计算机？



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

决策树的表示

决策树的基本组成部分：决策结点、分支和叶子。



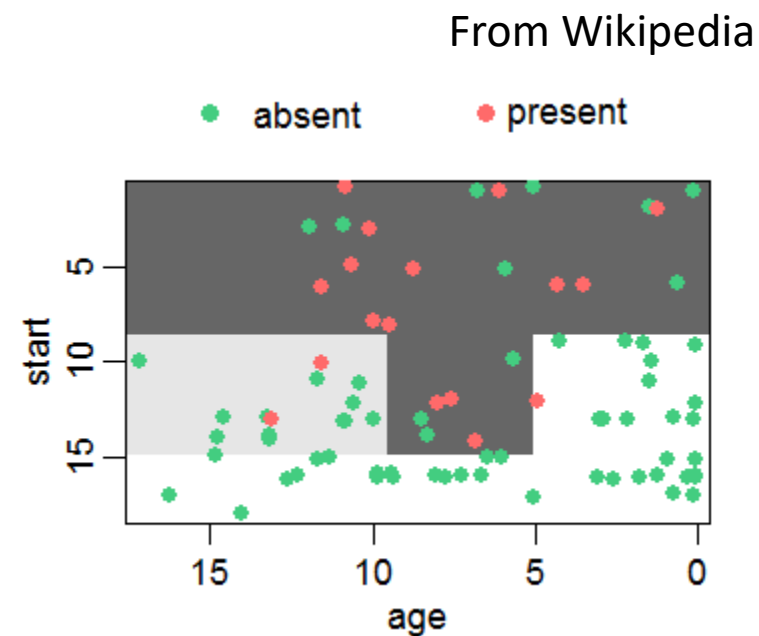
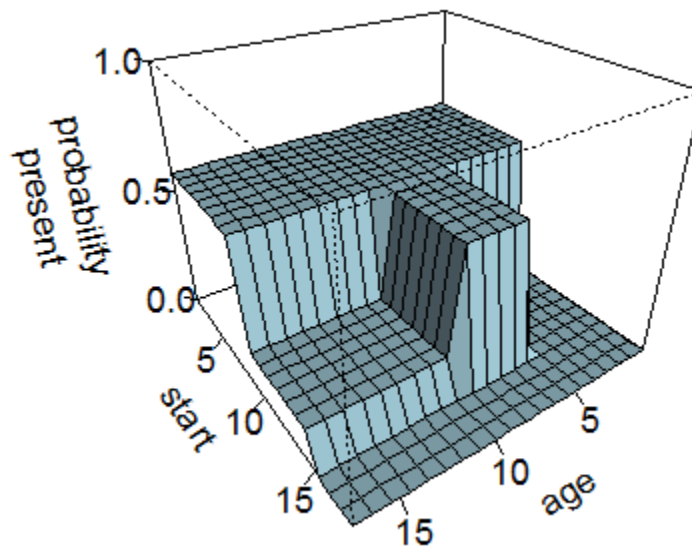
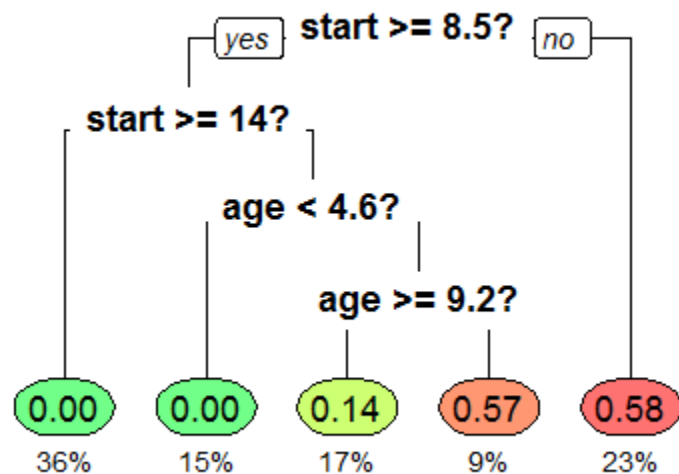
在沿着决策树从上到下的遍历过程中，在每个结点都有一个测试。对每个结点上问题的不同测试输出导致不同的分枝，最后会达到一个叶子结点。这一过程就是利用决策树进行分类的过程，利用若干个变量来判断属性的类别。

- 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。

决策树与条件概率分布

- 决策树表示给定特征条件下类的条件概率分布。
- 条件概率分布定义在特征空间的一个划分(partition)上。将特征空间划分为互不相交的单元(cell)或区域(region),并在每个单元定义一个类的概率分布就构成了一个条件概率分布。
- 决策树的一条路径对应于划分中的一个单元。
- 决策树所表示的条件概率分布由各个单元给定条件下类的条件概率分布组成。

决策树与条件概率分布



- 决策树学习是由训练数据集估计条件概率模型。基于特征空间划分的类的条件概率模型有无穷多个。
- 我们选择的条件概率模型应该不仅对训练数据有很好的拟合，而且对未知数据有很好的预测。

决策树CLS算法

- 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。

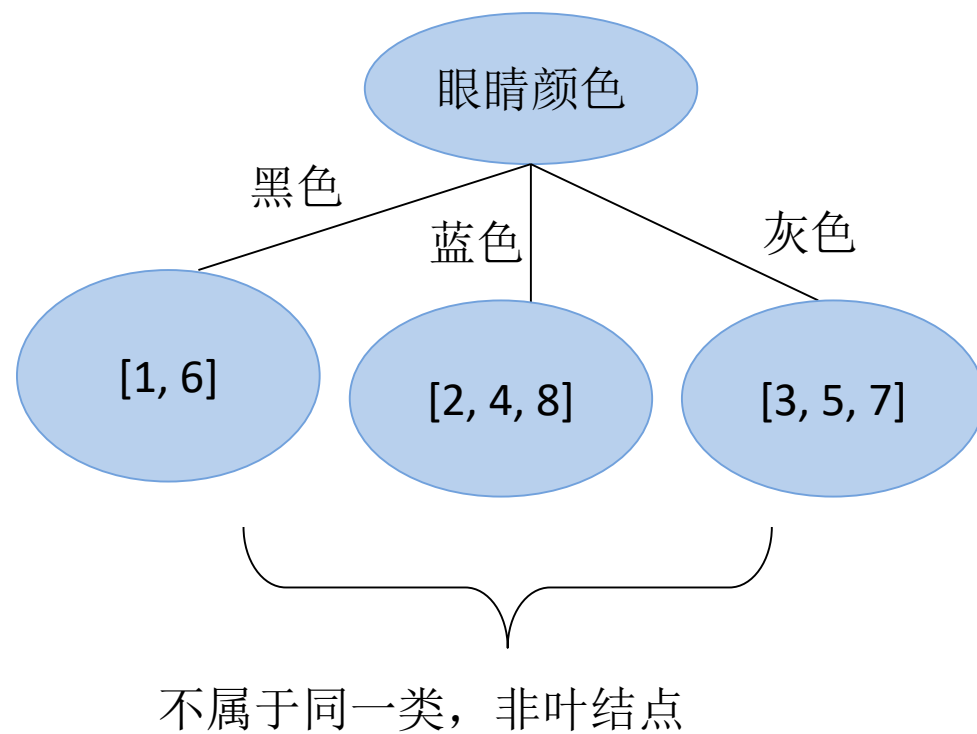
决策树CLS算法

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

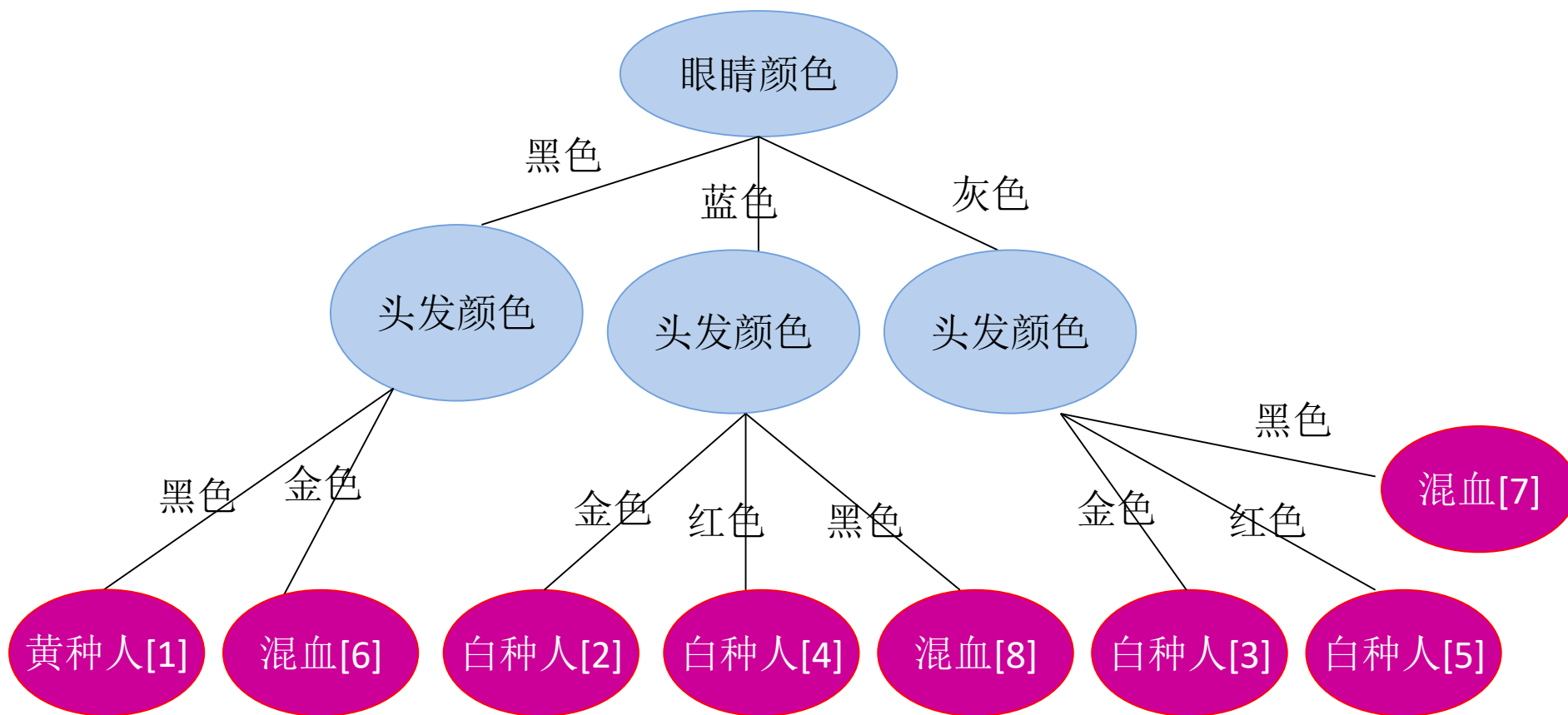
决策树CLS算法

决策树的构建

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CLS算法



决策树CLS算法

- 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 v_1, v_2, \dots, 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, \dots, T_m ;
 - 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;

决策树CLS算法

- 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。

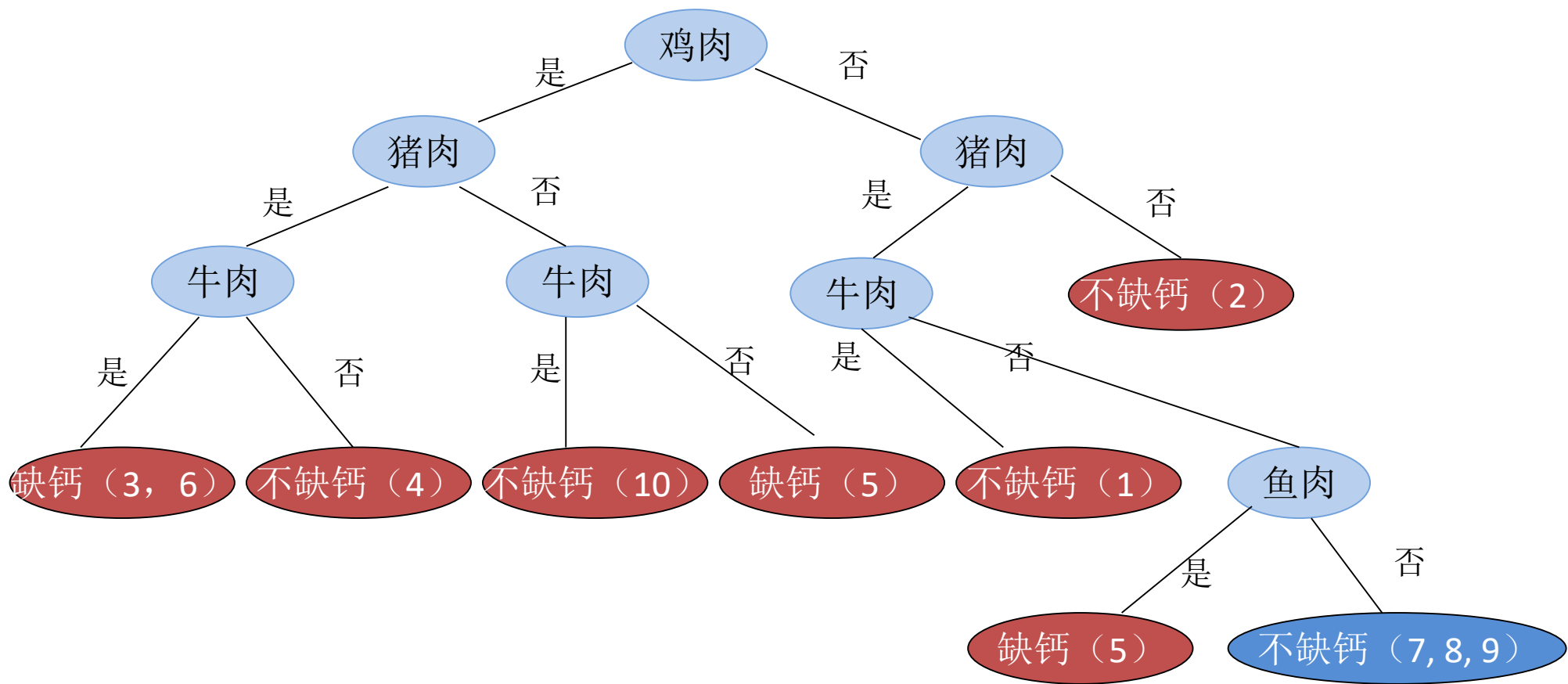
决策树CLS算法

学生膳食结构和缺钙调查表

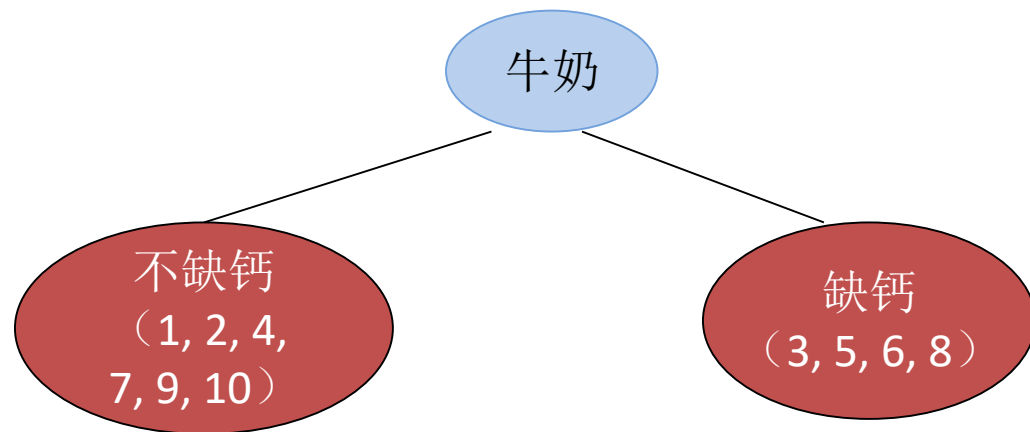
学生	鸡肉	猪肉	牛肉	羊肉	鱼肉	鸡蛋	青菜	番茄	牛奶	健康情况
1	0	1	1	0	1	0	1	0	1	不缺钙
2	0	0	0	0	1	1	1	1	1	不缺钙
3	1	1	1	1	1	0	1	0	0	缺钙
4	1	1	0	0	1	1	0	0	1	不缺钙
5	1	0	0	1	1	1	0	0	0	缺钙
6	1	1	1	0	0	1	0	1	0	缺钙
7	0	1	0	0	0	1	1	1	1	不缺钙
8	0	1	0	0	0	1	1	1	0	缺钙
9	0	1	0	0	0	1	1	1	1	不缺钙
10	1	0	1	1	1	1	0	1	1	不缺钙

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



决策树CLS算法

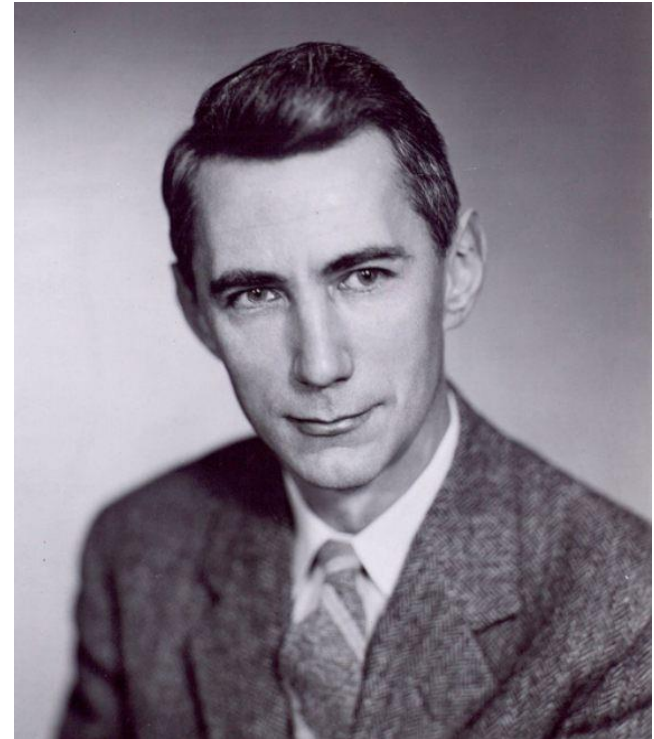


决策树ID3算法

- 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？

信息的度量

- 克劳德·艾尔伍德·香农（英语：Claude Elwood Shannon，1916年4月30日－2001年2月24日），[美国数学家](#)、[电子工程师](#)和[密码学家](#)，被誉为[信息论](#)的创始人
- 1948年，划时代的“通信的一个数学理论”分成两部分，在7月和10月的Bell System Technical Journal发表。文章系统论述了信息的定义，怎样数量化信息，怎样更好地对信息进行编码。在这些研究中，概率理论是香农使用的重要工具。香农同时提出了信息熵的概念，用于衡量信息的不确定性



信息熵

- 熵(entropy): 信息量大小的度量, 即表示**随机变量不确定性的度量**。
- 熵的**通俗解释**: 事件 a_i 的信息量 $I(a_i)$ 可如下度量:

$$I(a_i) = p(a_i) \log_2 \frac{1}{p(a_i)}$$

- 其中 $p(a_i)$ 表示事件 a_i 发生的概率。
- 假设有 n 个互不相容的事件 $a_1, a_2, a_3, \dots, a_n$, 它们中有且仅有一个发生, 则其平均的信息量(熵)可如下度量:

$$I(a_1, a_2, \dots, a_n) = \sum_{i=1}^n I(a_i) = \sum_{i=1}^n p(a_i) \log_2 \frac{1}{p(a_i)}$$

随机变量的熵

- 熵的**理论解释**:
- 设 X 是一个取有限个值的离散随机变量, 其概率分布为:

$$P(X = x_i) = p_i, \quad i = 1, 2, \dots, n$$

- 则随机变量的熵定义为:

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

- 对数以2为底或以e为底(自然对数), 这时熵的单位分别称作比特(bit)或纳特(nat), 熵只依赖于 X 的分布, 与 X 的取值无关。

$$H(p) = - \sum_{i=1}^n p_i \log p_i$$

随机变量的熵

- 熵的**理论解释**:
- 熵越大, 随机变量的不确定性越大:

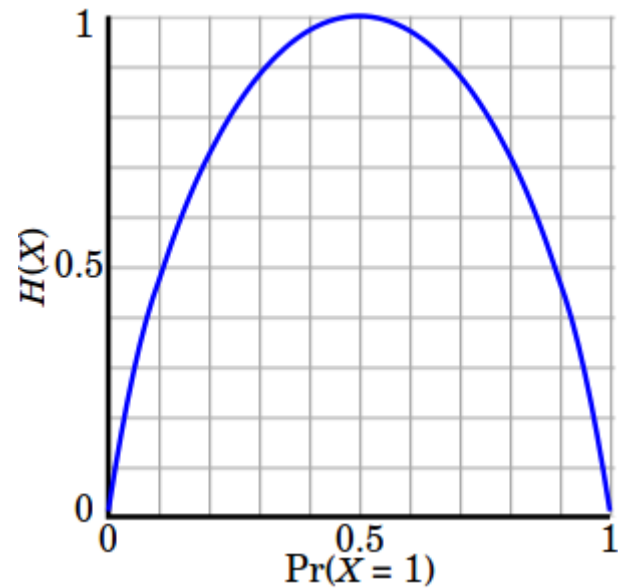
$$0 \leq H(p) \leq \log n$$

- 当 X 为1, 0分布时:

$$P(X = 1) = p, \quad P(X = 0) = 1 - p, \quad 0 \leq p \leq 1$$

- 熵:

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



条件熵

- 设有随机变量 (X, Y) ,其联合概率分布为:

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

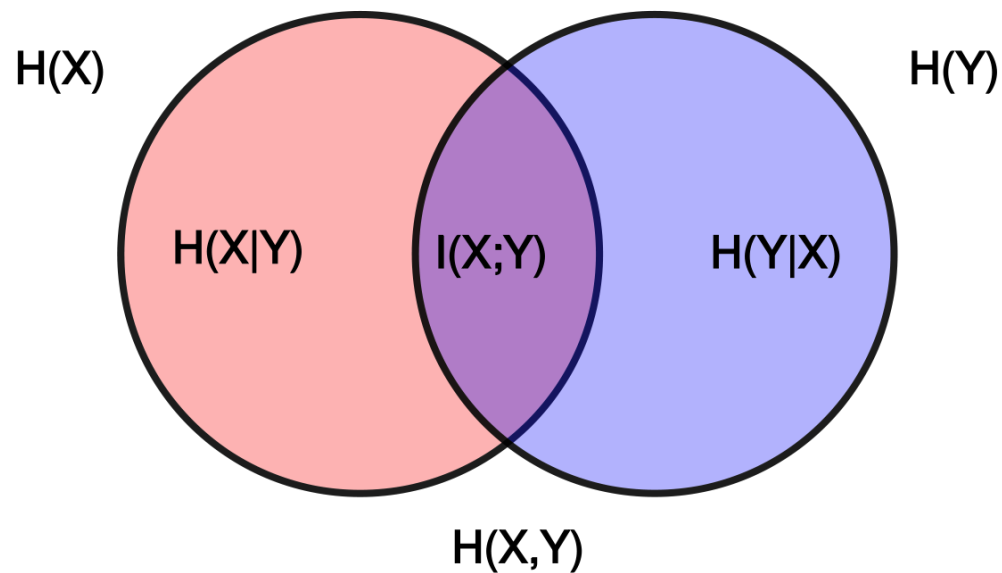
- 条件熵 $H(Y|X)$: 表示在已知随机变量 X 的条件下随机变量 Y 的不确定性, 定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望:

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

- 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)

条件熵

$$\begin{aligned} H(Y|X) &\equiv \sum_{x \in \mathcal{X}} p(x) H(Y|X=x) \\ &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x) \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x) \\ &= - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log p(y|x) \\ &= - \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)} \\ &= \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log \frac{p(x)}{p(x, y)} \end{aligned}$$



信息增益

- 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。

信息增益的算法

- 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 $C_k, k = 1, 2, \dots, K$,
- $|C_k|$ 为属于类 C_k 的样本个数
- 特征 A 有 n 个不同的取值 $\{a_1, a_2 \dots a_n\}$ 根据特征 A 的取值将 D 划分为 n 个子集 D_1, D_2, \dots, 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

信息增益的算法

- 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

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

信息增益比

- 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

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

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

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

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1步计算决策属性的熵

决策属性“买计算机？”。
该属性分两类：买/不买

$$|C_1|(\text{买}) = 641$$

$$|C_2|(\text{不买}) = 383$$

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

$$P_1 = 641/1024 = 0.6260$$

$$P_2 = 383/1024 = 0.3740$$

$$\begin{aligned} H(D) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\ &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\ &= 0.9537 \end{aligned}$$

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

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步计算条件属性的熵

条件属性共有4个：
年龄、收入、学生、信誉。
分别计算不同属性的信息增益。

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

$$|D_{11}|(\text{买}) = 128$$

$$|D_{12}|(\text{不买}) = 256$$

$$|D_1| = 384$$

$$P_1 = 128/384$$

$$P_2 = 256/384$$

$$\begin{aligned} H(D_1) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\ &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\ &= 0.9183 \end{aligned}$$

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

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

$$|D_{21}|(\text{买}) = 256$$

$$|D_{22}|(\text{不买}) = 0$$

$$|D_2| = 256$$

$$P_1 = 256/256$$

$$P_2 = 0/256$$

$$\begin{aligned}
 H(D_2) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0
 \end{aligned}$$

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

$$|D_{31}|(\text{买}) = 125$$

$$|D_{31}|(\text{不买}) = 127$$

$$|D_3| = S1 + S2 = 252$$

$$P_1 = 125/252$$

$$P_2 = 127/252$$

$$\begin{aligned}
 H(D_3) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0.9157
 \end{aligned}$$

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步计算年龄的熵

年龄共分三个组：

青年、中年、老年
所占比例

青年组 $384/1024=0.375$

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

$$E(\text{年龄}) = 0.375 \times 0.9183 + 0.25 \times 0 + 0.375 \times 0.9157 = 0.6877$$

$$\text{年龄信息增益} = 0.9537 - 0.6877 = 0.2660 \quad (1)$$

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 计算收入的熵

收入共分三个组：
高、中、低

$$E(\text{收入}) = 0.9361$$

$$\begin{aligned} \text{收入信息增益} &= 0.9537 - 0.9361 \\ &= 0.0176 \text{ (2)} \end{aligned}$$

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步计算学生的熵

学生共分二个组：
学生、非学生

$$E(\text{学生}) = 0.7811$$

$$\begin{aligned} \text{学生信息增益} &= 0.9537 - 0.7811 \\ &= 0.1726 \quad (3) \end{aligned}$$

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步计算信誉的熵

信誉分二个组：
良好，优秀

$$E(\text{信誉}) = 0.9048$$

$$\begin{aligned}\text{信誉信息增益} &= 0.9537 - 0.9048 \\ &= 0.0453 \quad (4)\end{aligned}$$

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步计算选择节点

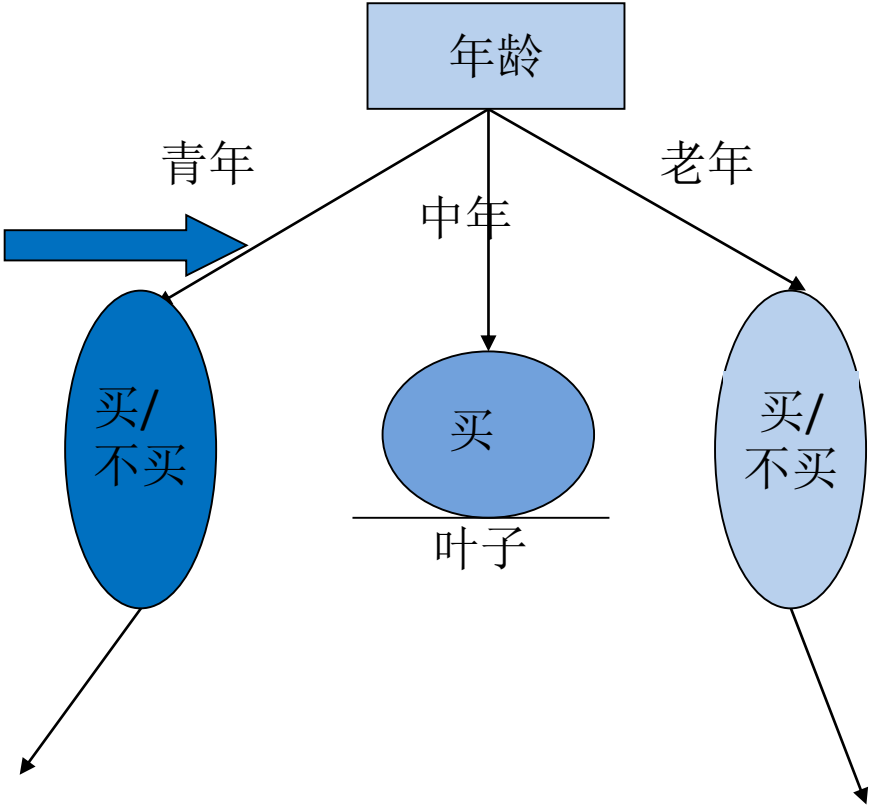
$$\text{年龄信息增益} = 0.9537 - 0.6877 \\ = 0.2660 \quad (1)$$

$$\text{收入信息增益} = 0.9537 - 0.9361 \\ = 0.0176 \quad (2)$$

$$\text{学生信息增益} = 0.9537 - 0.7811 \\ = 0.1726 \quad (3)$$

$$\text{信誉信息增益} = 0.9537 - 0.9048 \\ = 0.0453 \quad (4)$$

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



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

$$|C_1|(\text{买}) = 128$$

$$|C_2|(\text{不买}) = 256$$

$$|D| = 384$$

$$P_1 = 128/384$$

$$P_2 = 256/384$$

$$\begin{aligned}
 H(D) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0.9183
 \end{aligned}$$

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

如果选择收入作为节点
分高、中、低

$$H(D_1) = 0$$

$$\text{比例: } 128/384=0.3333$$

$$H(D_2) = 0.9183$$

$$\text{比例: } 192/384=0.5$$

$$H(D_3) = 0$$

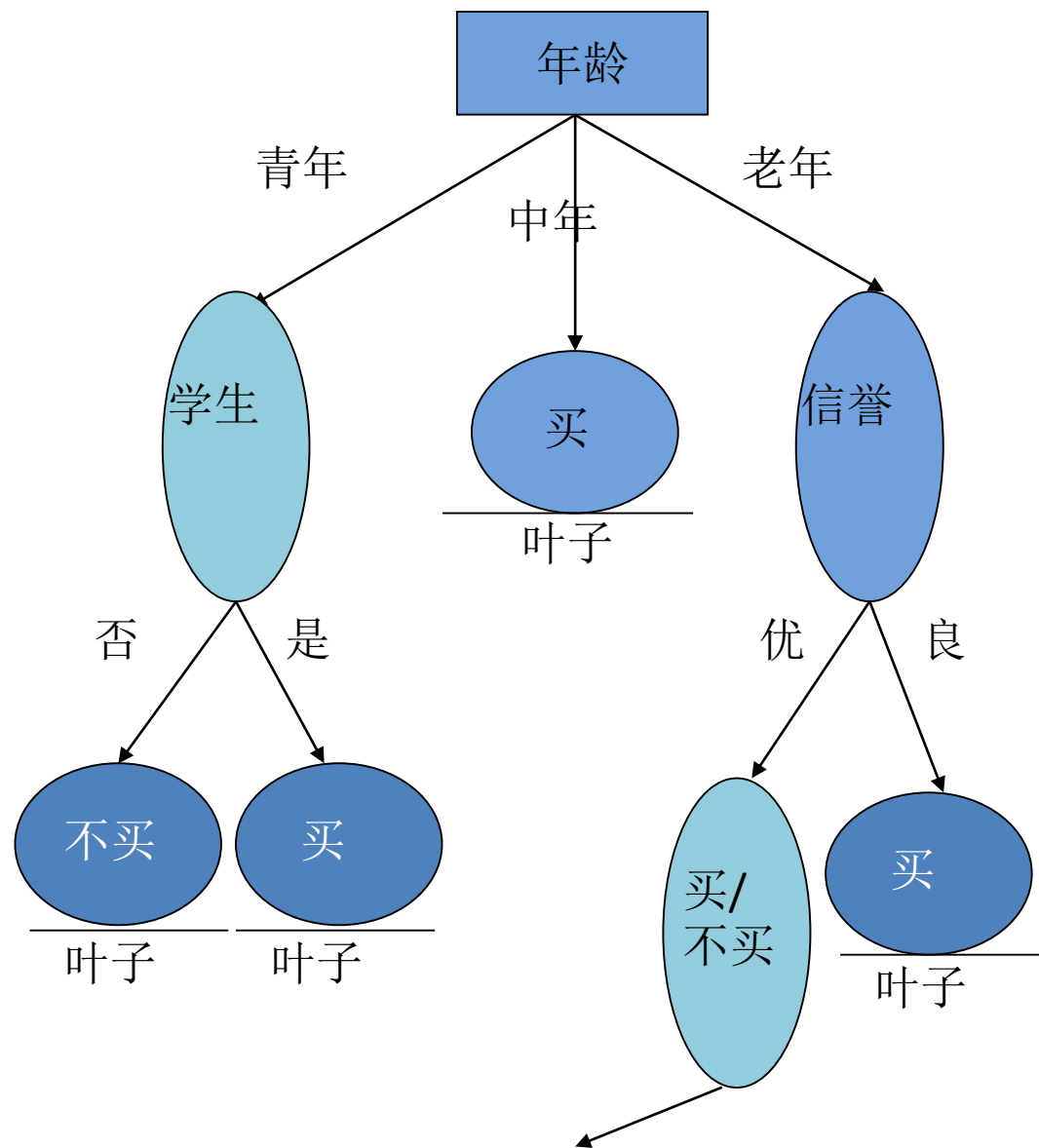
$$\text{比例: } 64/384=0.1667$$

平均信息期望（加权总和）：

$$E(\text{收入}) = 0.3333 \times 0 + 0.5 \times 0.9183 + 0.1667 \times 0 = 0.4592$$

$$\text{Gain}(\text{收入}) = I(128, 256) - E(\text{收入}) = 0.9183 - 0.4592 = 0.4591$$

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ID3算法-流程

- 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, 根据信息增益选出一个最佳属性作为节点 N 的测试属性
- 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。

决策树ID3算法-实际使用

原始表

姓名	年龄	收入	学生	信誉	电话	地址	邮编	买计算机
张三	23	4000	是	良	281-322-0328	2714 Ave. M	77388	买
李四	34	2800	否	优	713-239-7830	5606 Holly Cr	78766	买
王二	70	1900	否	优	281-242-3222	2000 Bell Blvd.	70244	不买
赵五	18	900	是	良	281-550-0544	100 Main Street	70244	买
刘兰	34	2500	否	优	713-239-7430	606 Holly Ct	78566	买
杨俊	27	8900	否	优	281-355-7990	233 Rice Blvd.	70388	不买
张毅	38	9500	否	优	281-556-0544	399 Sugar Rd.	78244	买
。 。 。	。 。							
。 。 。								

决策树ID3算法-实际使用

决策树的数据准备

整理后的数据表

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
。 。 。					

- ❖ Data cleaning
删除/减少noise,
补填missing values
- ❖ Data transformation
数据标准化（data normalization）
数据归纳（generalize data to higher-level concepts using concept hierarchies）
例如：年龄归纳为老、中、青三类
控制每个属性的可能值不超过七种（最好不超过五种）
- ❖ Relevance analysis
对于与问题无关的属性：删
对于属性的可能值大于七种
又不能归纳的属性：删

决策树ID3算法-小结

- 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

- 决策树的优点：
 - 1、推理过程容易理解，决策推理过程可以表示成**If Then**形式；
 - 2、推理过程完全依赖于属性变量的取值特点；
 - 3、可自动忽略目标变量没有贡献的属性变量，也为判断属性变量的重要性，减少变量的数目提供参考。

决策树面临的问题

- 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，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。

决策树面临的问题

- 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。
- 在以上情况发生时，这个简单的算法产生的树会过度拟合训练样例（过度拟合：Over Fitting）。

决策树面临的问题

- 对学习算法是否成功的真正测试是看它对于训练中未见到的数据的执行性能。训练过程应该包含训练样本和验证样本。验证样本用于测试训练后的性能。如果验证结果差，则需要考虑采用不同的结构重新进行训练，例如使用更大的样本集，或者改变从连续值到离散值得数据转换等。
- 通常应该建立一个验证过程，在训练最终完成后用来检测训练结果的泛化能力。

决策树面临的问题

- 决策树算法比较适合处理离散数值的属性。实际应用中属性是连续的或者离散的情况都比较常见。
- 在应用连续属性值时，在一个树结点可以将属性 A_i 的值划分为几个区间。然后信息增益的计算就可以采用和离散值处理一样的方法。原则上可以将 A_i 的属性划分为任意数目的空间。C4.5中采用的是二元分割（Binary Split）。需要找出一个合适的分割阈值。

参考C4.5算法 Top 10 algorithms in data mining Knowledge Information System 2008 14:1–37

决策树的剪枝

- 通过极小化决策树整体的损失函数或代价函数来实现。
- 设树 T 的叶结点个数为 $|T|$ ， t 是树 T 的叶结点，该叶结点有 N_t 个样本点，其中 k 类的样本点有 N_{tk} 个， $k = 1, 2, \dots, K$,
- $H_t(T)$ 为叶结点 t 上的经验熵， $\alpha \geq 0$ 为参数，损失函数：
$$C_\alpha(T) = \sum_{t=1}^{|T|} N_t H_t(T) + \alpha |T|$$
- 经验熵：
$$H_t(T) = - \sum_k \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t}$$
- 原式第一项：
$$C(T) = \sum_{t=1}^{|T|} N_t H_t(T) = - \sum_{t=1}^{|T|} \sum_{k=1}^K N_{tk} \log \frac{N_{tk}}{N_t}$$
- 则：
$$C_\alpha(T) = C(T) + \alpha |T|$$

决策树的剪枝

- 树的剪枝算法：
- 输入：生成算法产生的整个树 T ，参数 α ；
- 输出：修剪后的子树 T_α ；

(1) 计算每个结点的经验熵

(2) 递归地从树的叶节点向上回缩

设一组叶结点回缩到其父结点之前与之后的损失函数分别为：

$$C_\alpha(T_B) \text{ 与 } C_\alpha(T_A)$$

如果： $C_\alpha(T_A) \leq C_\alpha(T_B)$ ，则进行剪枝

(3) 返回 (2)，直至不能继续为止，得到损失函数最小的子树 T_α

谢谢各位同学！