

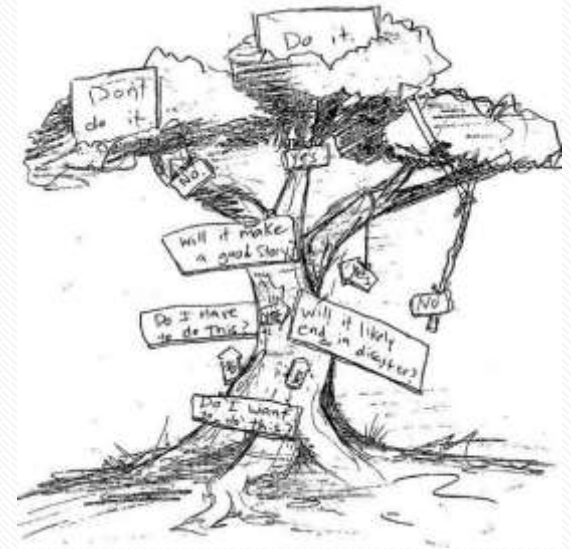
# **第三章：决策树**

## **Ch3: Decision Tree**

# 什么是决策树?

## 场景:

选专业、选大学、找工作、找对象、日常决定...



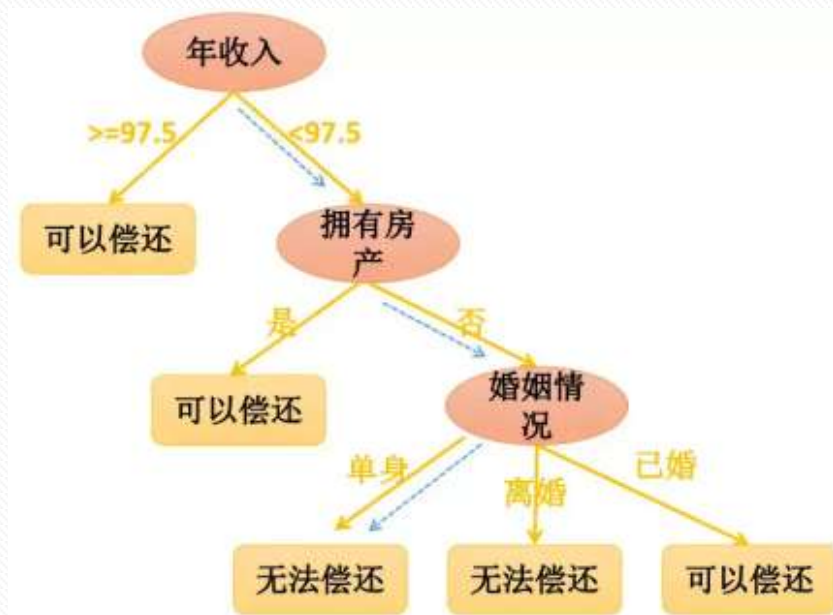
**决策树**是用于决策制定的最简单却高效的**分类**和**预测**的可视化工具的一种。

A decision tree is one of the simplest yet highly effective classification and prediction visual tools used for decision making.

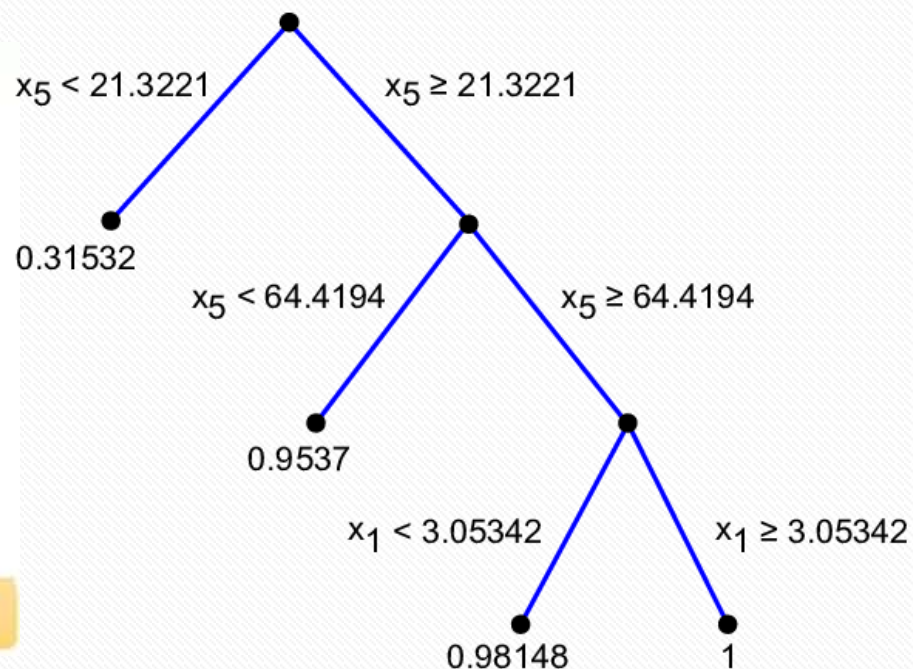
**To be or not to be. It is a decision!**

# 决策树模型

决策树：分类决策树和回归决策树



分类决策树



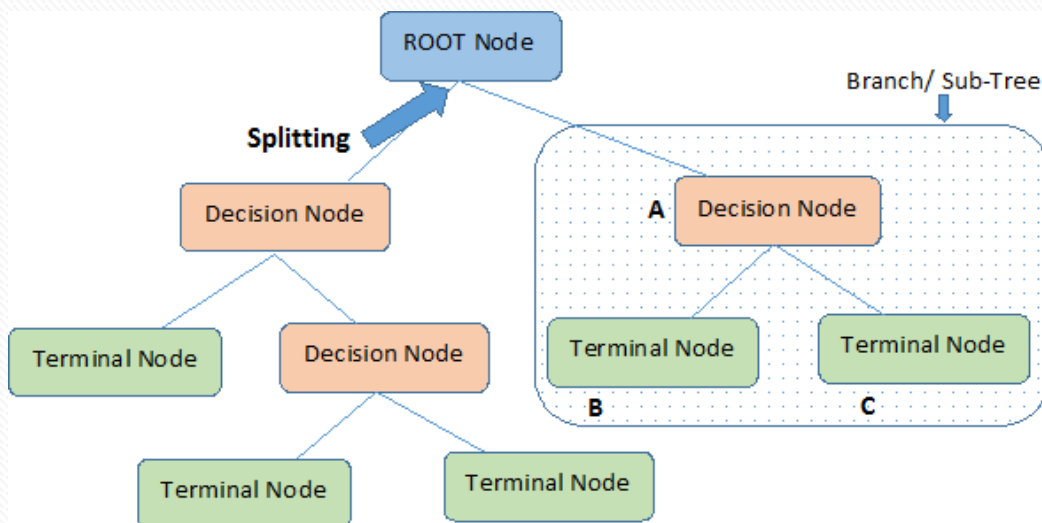
回归决策树

# 决策树模型

## 分类决策树

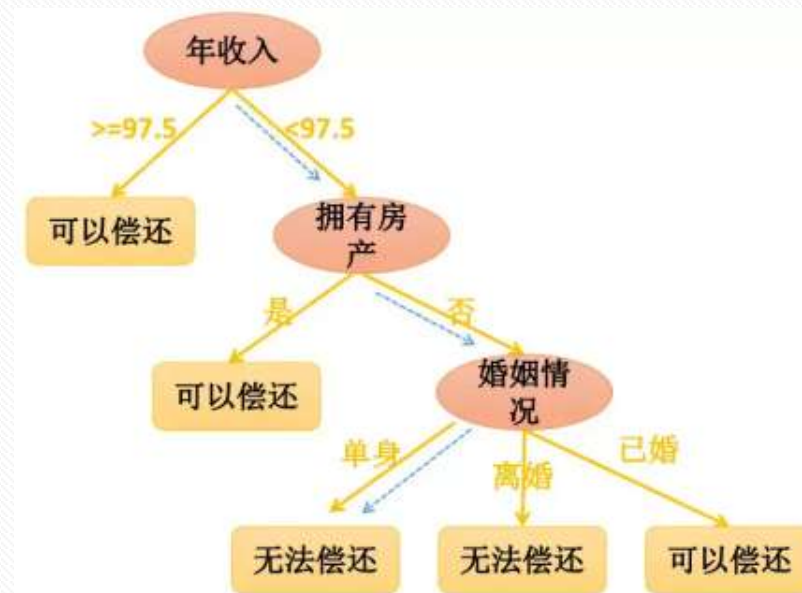
是一种描述对实例进行分类的**树形结构**，由**结点**(node)和**有向边**(directed edge)组成。

- 每个“**内部节点**”对应某个属性上的“测试”
- 每个**分支**对应于测试的一种可能结果（即该属性的某个取值）
- 每个“**叶节点**”对应于一个“预测结果”



**Note:-** A is parent node of B and C.

基本概念



具体例子

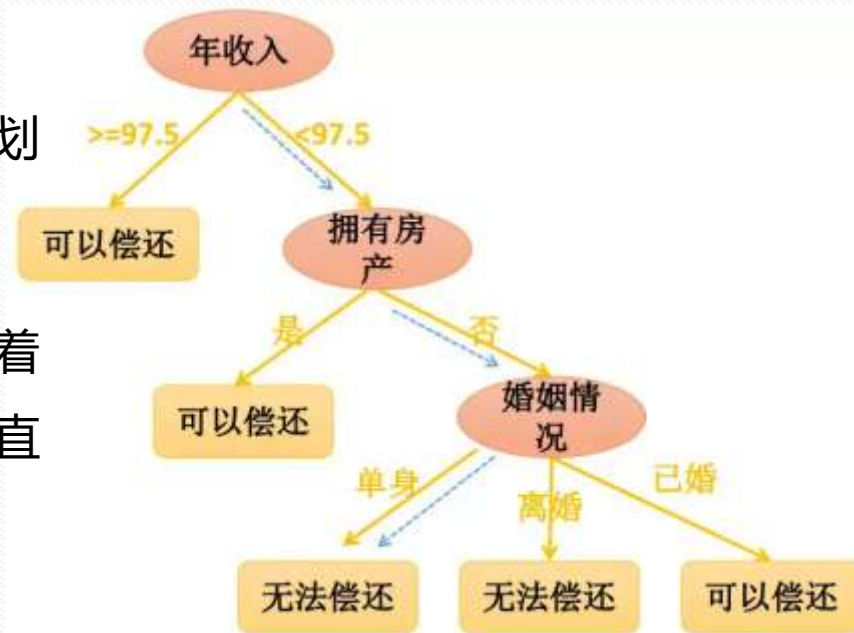
# 决策树模型

## 分类决策树

**学习过程：**通过训练样本的分析来确定“划分属性”

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

**互斥且完备：**每一个实例都被一条路径或一条规则所覆盖，而且只被一条规则所覆盖



银行贷款审批决策树

# 决策树模型

## 假设给定训练、数据集

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中,  $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)})^T$  为输入实例 (特征向量),  $n$  为特征个数  $y_i \in \{1, 2, \dots, K\}$  为类标记,  $i = 1, 2, \dots, N$ , 为样本容量。决策树学习的目标是根据给一定的训练数据集构建 一个决策树模型, 使它能够对实例进行正确的分类。

## 决策树学习算法

- 特征选择 (信息增益)
- 决策树生成 (ID3、C4.5)
- 决策树剪枝 (CART)

# 特征选择

## 概念1：信息量 (Information Quantity )

事件1：德国队获得世界杯冠军

事件2：中国队获得世界杯冠军

**信息量的大小和事件发生的概率成反相关**

信息奠基人香农 (Shannon, 1948) 认为 “信息是用来消除随机不确定性的东西”，也就是说衡量信息量大小就看这个信息消除不确定性的程度。事件E的信息量表达式为：

$$I(E) = -\log_2(p(E))$$

## 概念2：信息熵 ( Entropy ) C.E. Shannon (1948). A Mathematical Theory of Communication

信息量度量的是一个具体事件发生所带来的信息，而熵则是在结果出来之前对可能产生的信息量的期望——考虑该随机变量的所有可能取值，即所有可能发生事件所带来的信息量的期望。**是随机变量不确定性的度量。**

设X是一个取有限个值的离散随机变量，其概率分布为：

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

随机变量X的熵定义为：

$$H(X) = -\sum_{i=1}^n p_i \log p_i \quad H(p) = -\sum_{i=1}^n p_i \log p_i \quad \text{只与X分布有关}$$

# 特征选择

## 举例：抛硬币

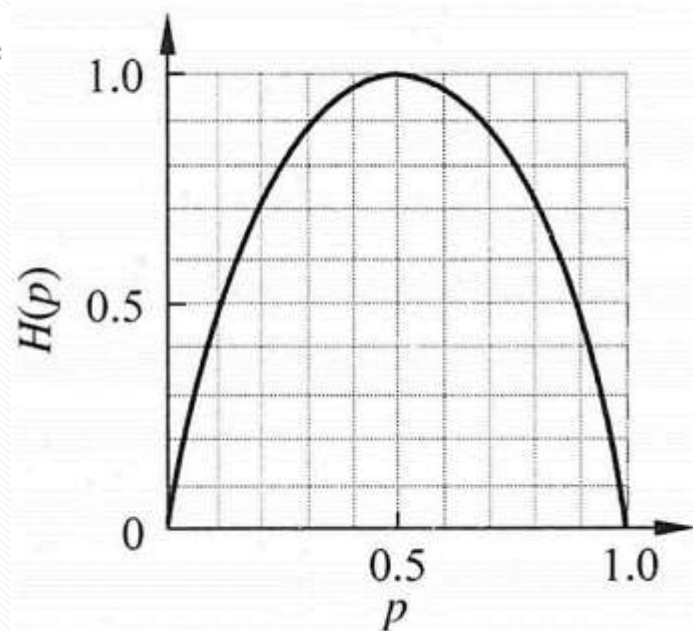
随机变量 $X$ 为1, 0分布：

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

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

- 当  $p = 0$  或者  $p = 1$ ， $H(p) = 0$ ，随机变量完全没有不确定性；
- 当  $p = 0.5$ ， $H(p) = 1$ ，随机变量不确定性最大。

约定： $0 \log 0 = 0$



伯努利分布时熵与概率的关系

可以证明： $0 \leq H(p) \leq \log n$



# 特征选择

## 概念3：条件熵 (Conditional Entropy)

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：

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

or

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

当熵和条件熵中的概率由**数据估计**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) 。

# 特征选择

## 概念4：信息增益 (Information Gain )

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,

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

**信息增益**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 $n$ 个不同取值 $\{a_1, a_2, \dots, a_n\}$ 根据特征 $A$ 取值将 $D$ 划分为 $n$ 个子集 $D_1 \dots D_n, |D_i|$ 为 $D_i$ 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

**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**增益最大**的特征为**最优特征**

# 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**，是最具影响和最为典型的决策树学习算法

**输入：**训练数据集 $D$ ，特征集 $A$ ，阈值 $\epsilon$ ；

**输出：**决策树 $T$

- 1) 若 $D$ 中所有实例属于同类 $C_k$ ，则 $T$ 为单结点树，将类 $C_k$ 作为该节点类标记，返回 $T$ ；
- 2) 若 $A=\emptyset$ ，则 $T$ 为单结点树，并将 $D$ 中实例数最大类 $C_k$ 作为该节点的类标记，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 $\epsilon$ ，则 $T$ 为单节点树，并将 $D$ 中实例数最大的类 $C_k$ 作为该节点的类标记，返回 $T$ ；
- 5) 否则，对 $A_g$ 的每种可能值 $a_i$ ，依 $A_g=a_i$ 将 $D$ 分割为若干非空子集 $D_i$ ，将 $D_i$ 中实例数最大的类作为标记，构建子结点，由结点及其子树构成树 $T$ ，返回 $T$ ；
- 6) 对子节点 $i$ ，以 $D_i$ 为训练集，以 $A-\{A_g\}$ 为特征集，递归调用1~5，得到子树 $T_i$ ，返回 $T_i$ ；

# 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。

**概念5：信息增益比 (Information Gain Ratio)：**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 $A$ 的熵之比值**。

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

其中，

$$H_A(D) = - \sum_{i=1}^n \frac{|D_i|}{|D|} \log_2 \frac{|D_i|}{|D|}$$

$n$  是特征  $A$  取值的个数

## 2.2.2 Gain ratio criterion

For some years the selection of a test in ID3 was made on the basis of the gain criterion. Although it gave quite good results, this criterion has a serious deficiency—it has a strong bias in favor of tests with many outcomes. We can see this by considering a hypothetical medical diagnosis task in which one of the attributes contains a patient identification. Since every such identification is intended to be unique, partitioning any set of training cases on the values of this attribute will lead to a large number of subsets, each containing just one case. Since all of these one-case subsets necessarily contain cases of a single class,  $info_X(T) = 0$ , so information gain from using this attribute to partition the set of training cases is maximal. From the point of view of prediction, however, such a division is quite useless.

The bias inherent in the gain criterion can be rectified by a kind of normalization in which the apparent gain attributable to tests with many outcomes is adjusted. Consider the information content of a message pertaining to a case that indicates not the class to which the case belongs, but the outcome of the test. By analogy with the definition of  $info(S)$ , we have

$$split\ info(X) = - \sum \frac{|T_i|}{|T|} \times \log_2 \left( \frac{|T_i|}{|T|} \right)$$

# 决策树C4.5算法

- C4.5算法和ID3算法类似，C4.5由J. Ross Quinlan在ID3的基础上提出；
- C4.5算法用信息增益比来选择特征。

**输入：**训练数据集 $D$ ，特征集 $A$ ，阈值 $\epsilon$ ；

**输出：**决策树 $T$

- 1) 若 $D$ 中所有实例属于同类 $C_k$ ，则 $T$ 为单结点树，将类 $C_k$ 作为该节点类标记，返回 $T$ ；
- 2) 若 $A=\emptyset$ ，则 $T$ 为单结点树，并将 $D$ 中实例数最大类 $C_k$ 作为该节点的类标记，返回 $T$ ；
- 3) 否则，计算 $A$ 中各特征对 $D$ 的**信息增益比**，选择**信息增益比**最大的特征 $A_g$ ；
- 4) 如果 $A_g$ 的**信息增益比**小于阈值 $\epsilon$ ，则 $T$ 为单节点树，并将 $D$ 中实例数最大的类 $C_k$ 作为该节点的类标记，返回 $T$ ；
- 5) 否则，对 $A_g$ 的每种可能值 $a_i$ ，依 $A_g=a_i$ 将 $D$ 分割为若干非空子集 $D_i$ ，将 $D_i$ 中实例数最大的类作为标记，构建子结点，由结点及其子树构成树 $T$ ，返回 $T$ ；
- 6) 对子节点 $i$ ，以 $D_i$ 为训练集，以 $A-\{A_g\}$ 为特征集，递归调用1~5，得到子树 $T_i$ ，返回 $T_i$ ；



# 决策树ID3算法实例

某公司收集了右表数据，  
那么对于任意给定的潜在客户（测试样例），  
你能预测这位客户“买”  
还是“不买”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ID3算法实例

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步计算数据集的熵

**决策属性** “买计算机？”  
该属性分两类：买/不买

$$|C_1|=641 \quad (\text{买})$$

$$|C_2|=383 \quad (\text{不买})$$

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

$$P_1=641/1024=0.6260$$

$$P_2=383/1024=0.3740$$

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

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



# 决策树ID3算法实例

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

然后计算条件属性的信息增益

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。

# 决策树ID3算法实例

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步计算年龄的信息增益

年龄共分三个组：

青年、中年、老年

**青年**买与不买比例为128/256

$$|D_{11}|(\text{买})=128$$

$$|D_{12}|(\text{不买})=256$$

$$|D_1|=384$$

$$P_1=128/384$$

$$P_2=256/384$$

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

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

# 决策树ID3算法实例

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步计算年龄的信息增益

年龄共分三个组：

青年、中年、老年

**中年**买与不买比例为256/0

$$|D_{21}|(\text{买})=256$$

$$|D_{22}|(\text{不买})=0$$

$$|D_2|=256$$

$$P_1=256/256$$

$$P_2=0/256$$

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

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

# 决策树ID3算法实例

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步计算年龄的信息增益

年龄共分三个组：

青年、中年、老年

**老年**买与不买比例为257/127

$$|D_{31}|(\text{买})=257$$

$$|D_{32}|(\text{不买})=127$$

$$|D_3|=384$$

$$P_1=257/384$$

$$P_2=127/384$$

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

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

# 决策树ID3算法实例

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  $384/1024=0.375$

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（经验条件熵）

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

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

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

# 决策树ID3算法实例

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3-1步计算收入的信息增益

收入共分三个组：

高、中、低

**高收入**买与不买比例为160/128

$$|D_{11}|(\text{买})=160$$

$$|D_{12}|(\text{不买})=128$$

$$|D_1|=288$$

$$P_1=160/288$$

$$P_2=128/288$$

$$\begin{aligned}
 H(D_1) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0.9911
 \end{aligned}$$

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

# 决策树ID3算法实例

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3-2步计算收入的信息增益

收入共分三个组：

高、中、低

**中收入**买与不买比例为289/191

$$|D_{21}|(\text{买})=289$$

$$|D_{22}|(\text{不买})=191$$

$$|D_2|=480$$

$$P_1=289/480$$

$$P_2=191/480$$

$$\begin{aligned}
 H(D_2) &= -P_1 \log_2 P_1 - P_2 \log_2 P_2 \\
 &= -(P_1 \log_2 P_1 + P_2 \log_2 P_2) \\
 &= 0.9697
 \end{aligned}$$

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

# 决策树ID3算法实例

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3-3步计算收入的信息增益

收入共分三个组：

高、中、低

**低收入**买与不买比例为192/64

$$|D_{31}|(\text{买})=192$$

$$|D_{32}|(\text{不买})=64$$

$$|D_3|=256$$

$$P_1=192/256$$

$$P_2=64/256$$

$$H(D_3)=-P_1\log_2 P_1 - P_2\log_2 P_2$$

$$=-(P_1\log_2 P_1 + P_2\log_2 P_2)$$

$$=0.8113$$

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



# 决策树ID3算法实例

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3-4步计算收入的信息增益

收入共分三个组：

高、中、低

所占比例

高收入组  $288/1024=0.281$

中收入组  $480/1024=0.469$

低收入组  $256/1024=0.250$

计算收入的平均信息期望（经验条件熵）

$$E(\text{收入}) = 0.281 * 0.9911 + 0.469 * 0.9697 + 0.250 * 0.8113 = 0.9361$$

$G$ （收入信息增益）

$$= 0.9537 - 0.9361$$

$$= 0.0176 \quad (2)$$

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

# 决策树ID3算法实例

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

$$E(\text{学生}) = 0.7811$$

$$G(\text{学生}) = 0.9537 - 0.7811$$

$$= 0.1726 \quad (3)$$

# 决策树ID3算法实例

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

$$E(\text{信誉}) = 0.9048$$

$$G(\text{信誉}) = 0.9537 - 0.9048 \\ = 0.0453 \quad (4)$$

# 决策树ID3算法实例

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步确定最优的特征

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

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

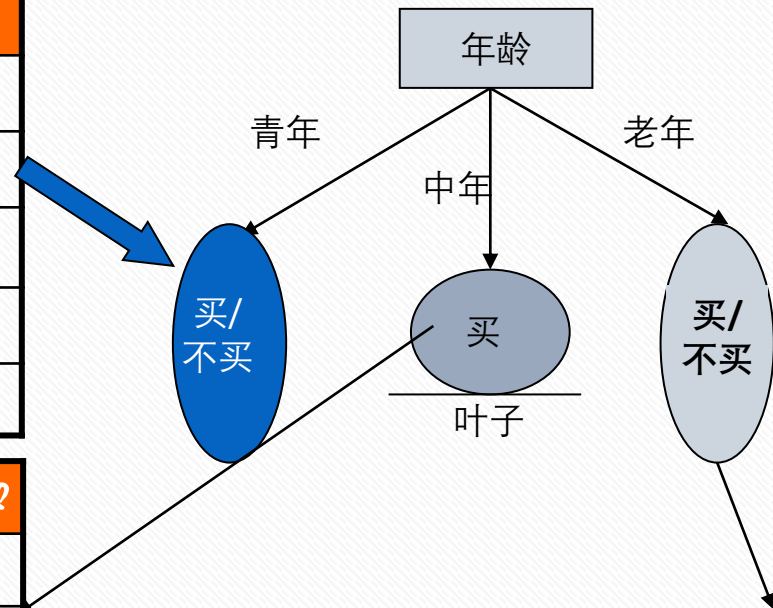
$$\text{年龄信息增益} = 0.9537 - 0.7811 = 0.1726 \quad (3)$$

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



# 决策树ID3算法实例

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？
128	中	高	否	良	买
64	中	低	是	优	买
32	中	中	否	优	买
32	中	高	是	良	买

计数	年龄	收入	学生	信誉	归类：买计算机？
60	老	中	否	良	买
64	老	低	是	良	买
64	老	低	是	优	不买
132	老	中	是	良	买
63	老	中	否	优	不买
1	老	中	否	优	买

# 决策树ID3算法实例

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

以青年组子集为训练集，以A-{年龄}为特征集，开始递归

## 第1步计算决策属性的信息增益

青年买与不买比例为128/256

$$|C1|(\text{买})=128$$

$$|C2|(\text{不买})=256$$

$$|D|=384$$

$$P1=128/384=0.3333$$

$$P2=256/384=0.6666$$

$$\begin{aligned} H(D) &= -P1\log_2 P1 - P2\log_2 P2 \\ &= (0.3333\log_2 0.3333 + 0.6666\log_2 0.6666) \\ &= 0.9183 \end{aligned}$$

# 决策树ID3算法实例

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

## 第2步计算收入属性的信息增益

收入分

高、中、低三个组：

$$H(D1)=0$$

比例：

$$128/384=0.3333$$

$$H(D2)=0.9183$$

$$\text{比例: } 192/384=0.5$$

$$H(D3)=0$$

比例：

$$64/384=0.1667$$

平均信息期望（加权总和）：

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

$$\text{Gain}(\text{收入}) = H(D) - E(\text{收入}) = \mathbf{0.9183} - 0.4592 = 0.4591$$

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

# 决策树ID3算法实例

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

## 第3步计算学生属性的信息增益

学生分

否、是两个组：

$$H(D_1)=0$$

$$\text{比例: } 256/384=0.6666$$

$$H(D_2)=0$$

$$\text{比例: } 128/384=0.3333$$

平均信息期望（加权总和）：

$$E(\text{学生}) = 0.6666 * 0 + 0.3333 * 0$$

$$\text{Gain}(\text{学生}) = H(D) - E(\text{学生}) = \mathbf{0.9183} - 0 = 0.9183$$

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



# 决策树ID3算法实例

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

## 第4步计算信誉属性的信息增益

学生分：

良、优两个组：

$$H(D1)=0.8113$$

比例：

$$256/384=0.6666$$

$$H(D2)=1$$

比例：

$$128/384=0.3333$$

平均信息期望（加权总和）：

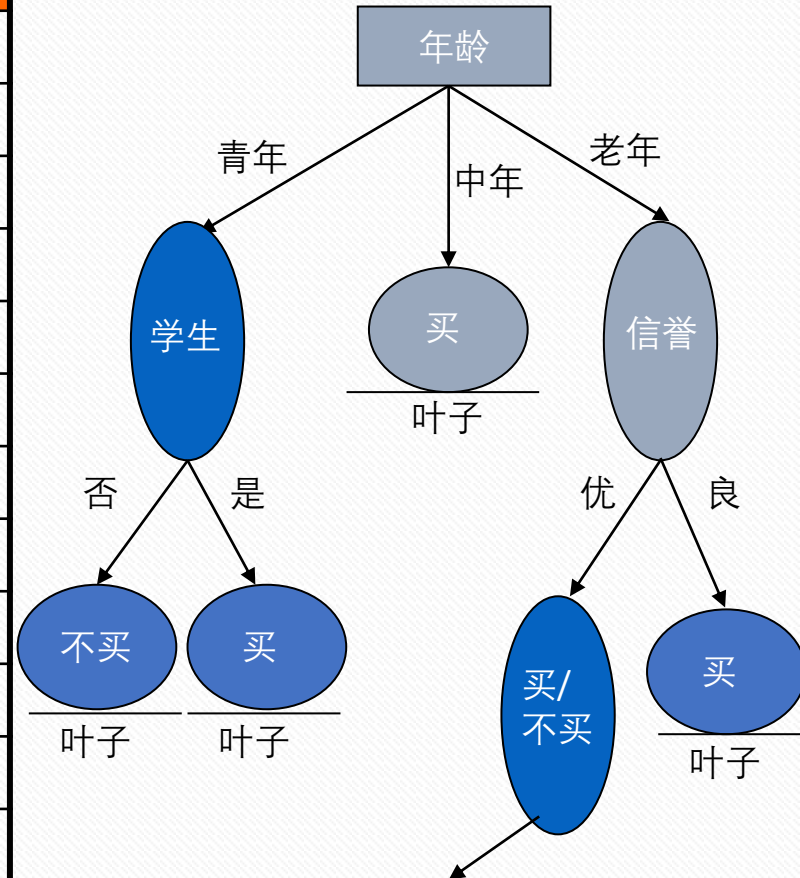
$$E(\text{信誉}) = 0.6666 * 0.8113 + 0.3333 * 1 = 0.8741$$

$$\text{Gain}(\text{信誉}) = H(D) - E(\text{信誉}) = 0.9183 - 0.8741 = 0.044$$

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

# 决策树ID3算法实例

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-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

- **数据清理:** 删除/减少噪音, 补填空缺值
- **数据转化:**
  - 数据归一化(normalization)
  - 数据归纳, 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)
- **相关性分析:**
  - 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-实际使用

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
。 。 。					

# 决策树剪枝

**剪枝：**学习时过多地考虑如何提高对训练数据的正确分类，从而构建出过于复杂的决策树，造成**过拟合**现象（训练误差低，泛化误差高，称为过度拟合）。解决办法是考虑决策树的复杂度，对已生成的树进行简化，这一过程称为剪枝（pruning）。

决策树的剪枝往往通过**极小化**决策树整体的**损失函数**来实现

**决策树的损失函数：**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， $H_t(T)$ 为叶结点上的经验熵，则损失函数为：

$$C_{\alpha}(T) = C(T) + \alpha |T| \longrightarrow \text{模型复杂度}$$
$$\begin{aligned} C(T) &= \sum_{t=1}^{|T|} N_t H_t(T) = \sum_{t=1}^{|T|} N_t \left[ - \sum_{k=1}^K \frac{N_{tk}}{N_t} \log \frac{N_{tk}}{N_t} \right] \\ &= - \sum_{t=1}^{|T|} \sum_{k=1}^K N_{tk} \log \frac{N_{tk}}{N_t} \end{aligned}$$

训练误差  $\alpha \geq 0$

决策树生成只考虑了通过提高信息增益（或信息增益比）对训练数据进行更好的拟合。而“决策树剪枝”通过优化损失函数，还考虑了减小模型复杂度。

# 决策树剪枝算法

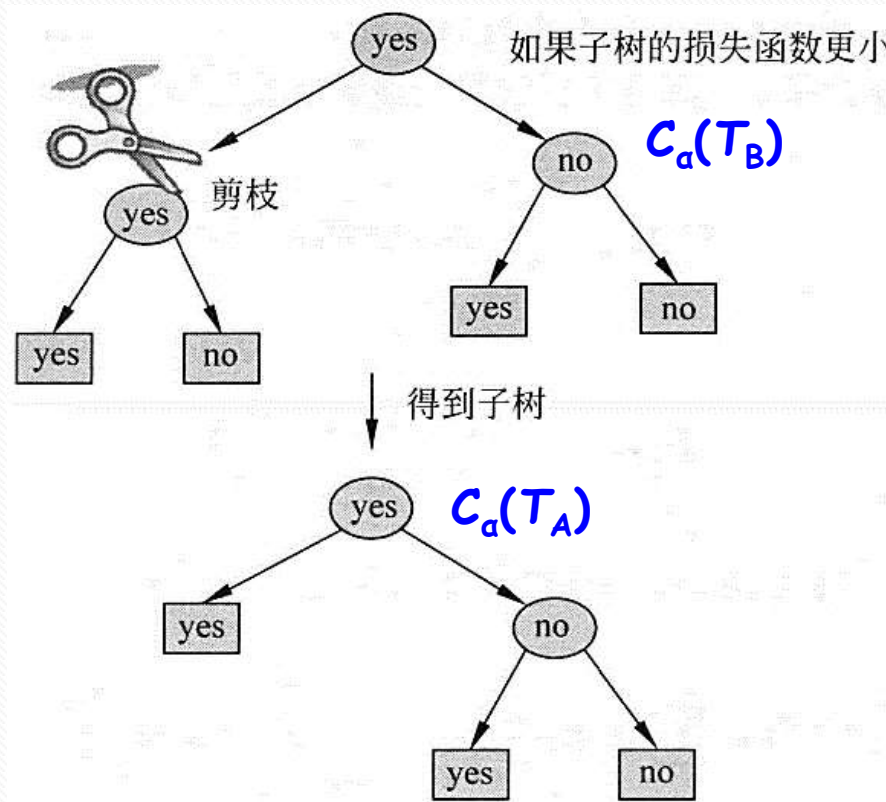
- (1) 计算每个结点的经验熵
- (2) 递归地从树的叶结点向上回缩，设一组叶结点回缩到其父结点之前与之后的整体树为  $T_B$  与  $T_A$ ，如果：

$$C_\alpha(T_A) \leq C_\alpha(T_B)$$

$$C_\alpha(T) = C(T) + \alpha|T|$$

则剪枝，将父结点变成新的叶结点。

- (3) 返回 (2)，直至不能继续，得到损失函数最小的子树  $T_\alpha$ 。



# CART算法

**Classification And Regression Tree**, 即**分类回归树算法**, 简称**CART算法**, 它是决策树的一种实现, 是一种二分递归分割技术, 把当前样本划分为两个子样本, 使得生成的每个非叶子结点都有两个分支, 因此CART算法生成的决策树是结构简洁的二叉树。由于CART算法构成的是一个**二叉树**, 它在每一步的决策时只能是“是”或者“否”, 即使一个feature有多个取值, 也是把数据分为两部分。

在CART算法中主要分为两个步骤:

- (1) 将样本递归划分进行建树过程
- (2) 用验证数据进行剪枝

CART分类树算法使用**基尼系数来代替信息增益比**, 基尼系数代表了模型的不纯度, 基尼系数越小, 不纯度越低, 特征越好。这和**信息增益 (比)**相反。



# CART算法

假设 $K$ 个类别，第 $k$ 个类别的概率为 $p_k$ ，概率分布的基尼系数表达式：

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

如果是二分类问题，样本属于1类的概率为 $p$ ，概率分布的基尼系数表达式为：

$$Gini(p) = 2p(1 - p)$$

对于样本 $D$ ，个数为 $|D|$ ，假设 $K$ 个类别，第 $k$ 个类别的数量为 $|C_k|$ ，则样本 $D$ 的基尼系数表达式：

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

对于样本 $D$ ，个数为 $|D|$ ，根据特征 $A$ 的某个值 $a$ ，把 $D$ 分成 $|D_1|$ 和 $|D_2|$ ，则在特征 $A$ 的条件下，样本 $D$ 的基尼系数表达式为：

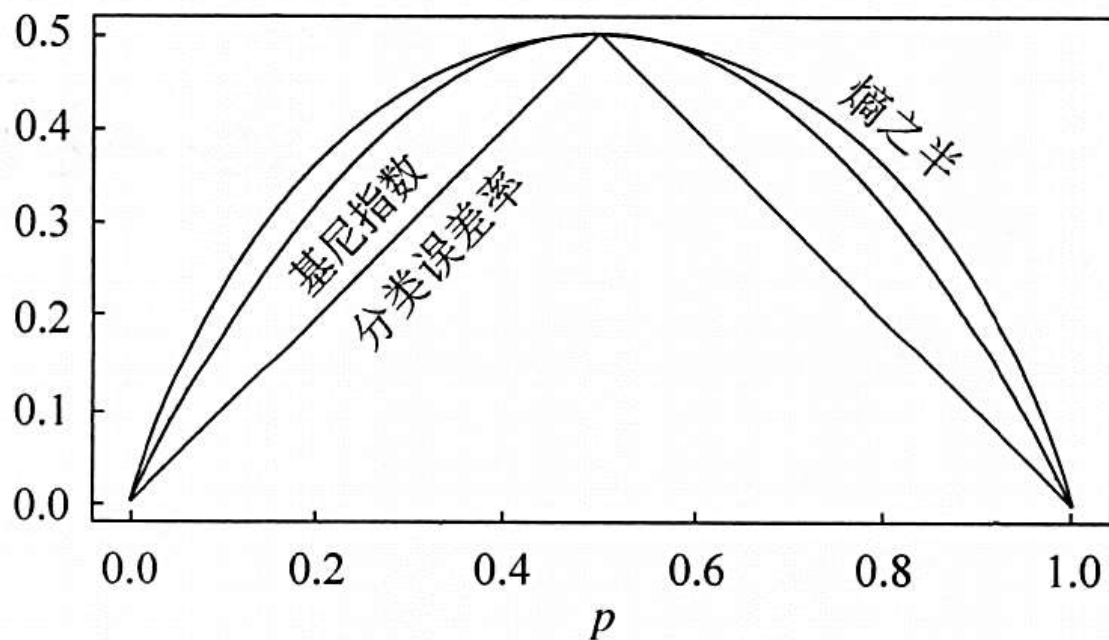
$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

比较基尼系数和熵模型的表达式，二次运算比对数简单很多。尤其是二分类问题，更加简单。**基尼系数可以作为熵模型的一个近似替代**



# CART算法

## 基尼系数和熵



$$Gini(p) = 2p(1-p)$$

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

$$Err = 1 - \max(p, 1-p)$$

# CART树生成算法

**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，递归对每个结点进行以下操作，构建**二叉树决策树**：

- 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 $D_1$ ， $D_2$ ，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

**停止计算条件：**

- 节点中样本个数小于预定阈值
- 样本集的基尼指数少于预定阈值
- 没有更多的特征

# CART树生成算法

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

$A_1$ =年龄 青=1, 中=2, 老=3  
 $A_2$ =有工作 有=1, 否=2  
 $A_3$ =有房子 有=1, 否=2  
 $A_4$ =信贷 一般=1, 好=2, 非常好=3

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

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

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

$$Gini(D, A_1 = 2) = 0.48$$

$$Gini(D, A_1 = 3) = 0.44$$

$$Gini(D, A_2 = 1) = 0.32$$

$$Gini(D, A_3 = 1) = 0.27$$

$$Gini(D, A_4 = 1) = 0.36$$

$$Gini(D, A_4 = 2) = 0.47$$

$$Gini(D, A_4 = 3) = 0.32$$

$A_3$ 为最优特征,  $A_3=1$ 为最优切分点, 根节点生成两个子节点, 一个是叶节点, 另一个节点在 $A_1, A_2, A_4$ 中选择最优特征和切分点。

# CART树剪枝算法

## CART树剪枝算法2步组成:

### 1) 剪枝, 形成子树序列

剪枝过程中, 计算子树的损失函数:

$$C_{\alpha}(T) = C(T) + \alpha|T|$$

其中,  $T$ 为任意子树,  $C(T)$ 为对训练数据的预测误差,  $|T|$ 为子树的叶结点个数,  $\alpha \geq 0$ 为正则系数,  $C_{\alpha}(T)$ 为参数 $\alpha$ 时子树 $T$ 的整体损失。

对固定的 $\alpha$ 一定存在损失函数最小的子树, 表示为 $T_{\alpha}$

- 当 $\alpha$ 大的时候, 最优子树 $T_{\alpha}$ 偏小
- $\alpha=0$ 时, 整体树最优
- $\alpha$ 趋近无穷大, 单结点树最优

# CART树剪枝算法

## 1) 剪枝, 形成子树序列

剪枝前以 $t$ 结点为根结点的子树的损失函数是:


$$C_{\alpha}(T_t) = C(T_t) + \alpha |T_t|$$

剪枝以后以 $t$ 为单结点树的损失函数是:

$$C_{\alpha}(t) = C(t) + \alpha$$

当 $\alpha=0$ 及 $\alpha$ 充分小, 有不等式:

$$C_{\alpha}(T_t) < C_{\alpha}(t)$$

当 $\alpha=0$ 继续增大,  $C_{\alpha}(T_t) = C_{\alpha}(t)$ .   $g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$

在 $T_0$ 中剪去 $g(t)$ 最小的 $T_+$ , 得到子树 $T_1$ , 同时将最小的 $g(t)$ 设为 $\alpha_1$ ,  $T_1$ 为区间 $[\alpha_1, \alpha_2)$ 的最优子树, 如此剪枝下去, 直到根节点, 不断增加 $\alpha$ 的值, 产生新的区间。

## 2) 交叉验证选取最优子树

在剪枝后子树序列 $\{T_0, T_1 \dots T_n\}$ 中通过交叉验证选取最优子树 $T_{\alpha}$ , 利用独立的验证数据集, 测试子树序列中各子树的平方误差或基尼指数, 最小的决策树就是最优决策树。

# CART树剪枝算法

## 算法 5.7 (CART 剪枝算法)

输入: CART 算法生成的决策树  $T_0$ ;

输出: 最优决策树  $T_\alpha$ 。

(1) 设  $k = 0$ ,  $T = T_0$ 。

(2) 设  $\alpha = +\infty$ 。

(3) 自下而上地对各内部结点  $t$  计算  $C(T_t)$ ,  $|T_t|$  以及

$$g(t) = \frac{C(t) - C(T_t)}{|T_t| - 1}$$

$$\alpha = \min(\alpha, g(t))$$

这里,  $T_t$  表示以  $t$  为根结点的子树,  $C(T_t)$  是对训练数据的预测误差,  $|T_t|$  是  $T_t$  的叶结点个数。

(4) 对  $g(t) = \alpha$  的内部结点  $t$  进行剪枝, 并对叶结点  $t$  以多数表决法决定其类, 得到树  $T$ 。

(5) 设  $k = k + 1$ ,  $\alpha_k = \alpha$ ,  $T_k = T$ 。

(6) 如果  $T_k$  不是由根结点及两个叶结点构成的树, 则回到步骤 (2); 否则令  $T_k = T_n$ 。

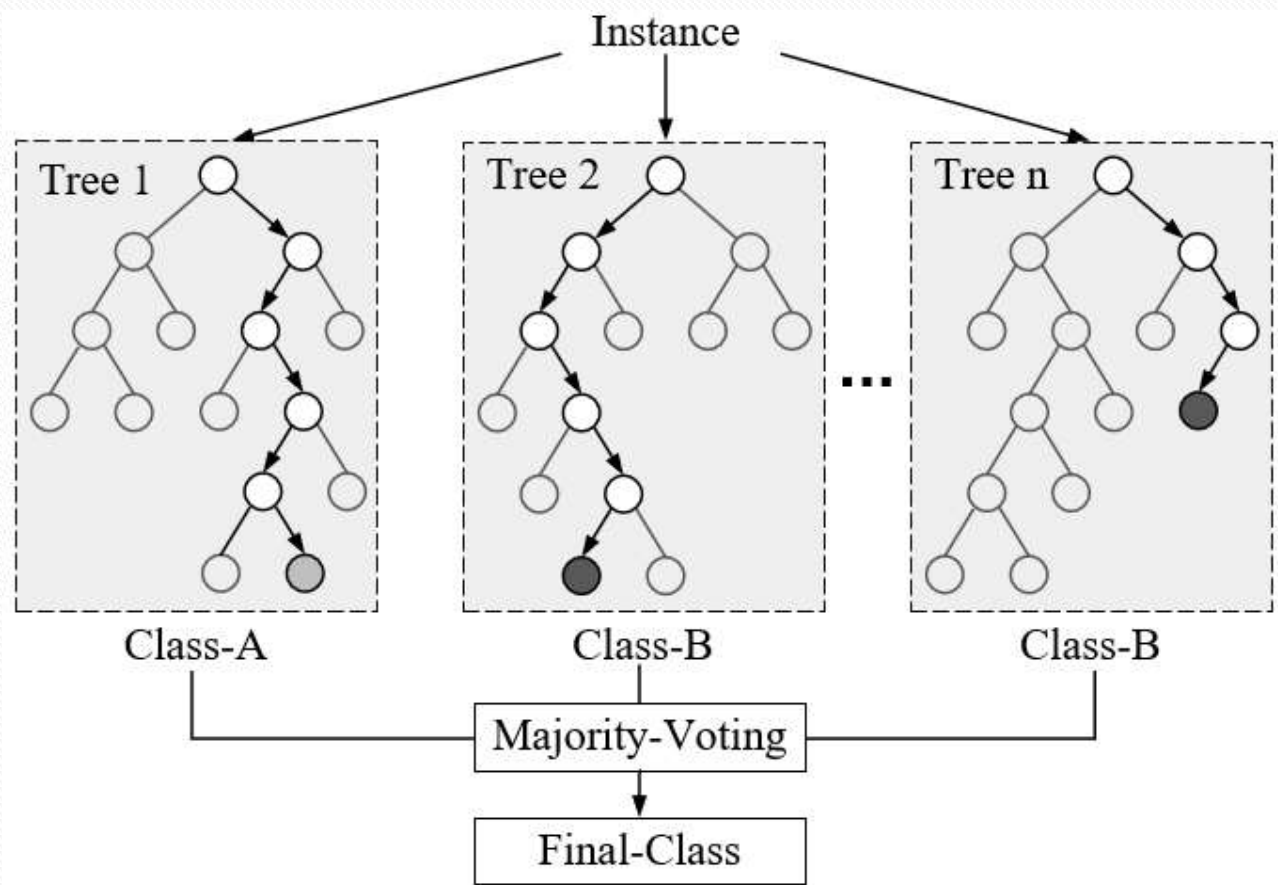
(7) 采用交叉验证法在子树序列  $T_0, T_1, \dots, T_n$  中选取最优子树  $T_\alpha$ 。





# 延伸：随机森林 (RF)

随机森林是一种利用**多棵树**对样本进行训练并预测的分类器。它的工作原理是通过对训练集进行随机采样，生成多棵决策树，然后根据个别树输出的类别的众数（**多数表决**）来决定最终的预测类别。每棵决策树都独立地拟合数据，但同时所有的树共享随机采样和特征选择的过程。





# 作业1

请根据某医院心血管科病人身体情况构建决策树模型。请利用ID3算法给出具体步骤。

患者ID	胸疼	性别	吸烟与否	是否运动	心脏病
1.	是	男	否	是	是
2.	是	男	是	否	是
3.	否	女	是	否	是
4.	否	男	否	是	否
5.	是	女	是	是	是
6.	否	男	是	是	否

**END**