



# 人工智能：机器学习 I

饶洋辉

计算机学院,

中山大学

[raoyangh@mail.sysu.edu.cn](mailto:raoyangh@mail.sysu.edu.cn)

<http://cse.sysu.edu.cn/node/2471>

课件来源：中山大学陈川副教授等

# 机器学习

- “Learning is any process by which a system improves performance from experience.” - Herbert Simon
- Tom Mitchell (1997): 机器学习是指一个**电脑程序**从经验  $E$ （如**带标签数据**）中**学习**如何提高某些任务  $T$ （如**分类**）的性能指标  $P$ （如**精度和召回率**），即以 $P$ 作为指标，通过 $E$ 来提高 $T$ 的表现。

# 机器学习

- 机器学习就是让计算机能够像人那样自动获取新知识，并在实践中不断地完善自我和增强能力，使得系统在下一次执行同样任务或类似的任务时，会比现在做得更好或效率更高。
- 机器学习的研究一方面可以使机器能自动获取知识，赋予机器更多的智能；另一方面可以进一步揭示人类思维规律和学习奥秘，帮助人们提高学习效率。机器学习的研究还会对记忆存储模式、信息输入方式及计算机体系结构产生重大影响。

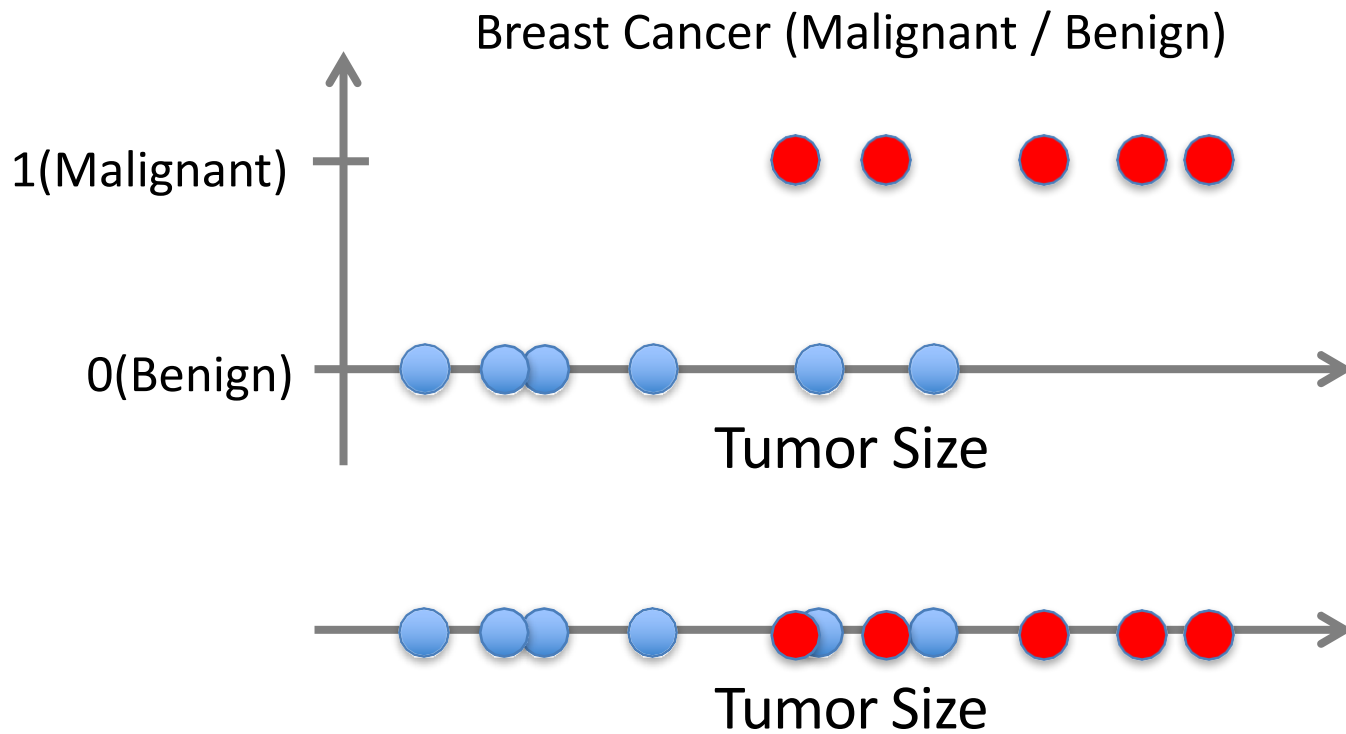
# 机器学习的类型

- 有监督学习
  - 分类、回归
  - 给定：输入特征 + 输出（标签）
- 无监督学习
  - 聚类、关联分析
  - 给定：输入特征
- 混合学习
  - 对有监督学习和无监督学习的融合
- 强化学习
  - 只给出对当前输出的一个评价，而非具体的标签

# 有监督学习：分类

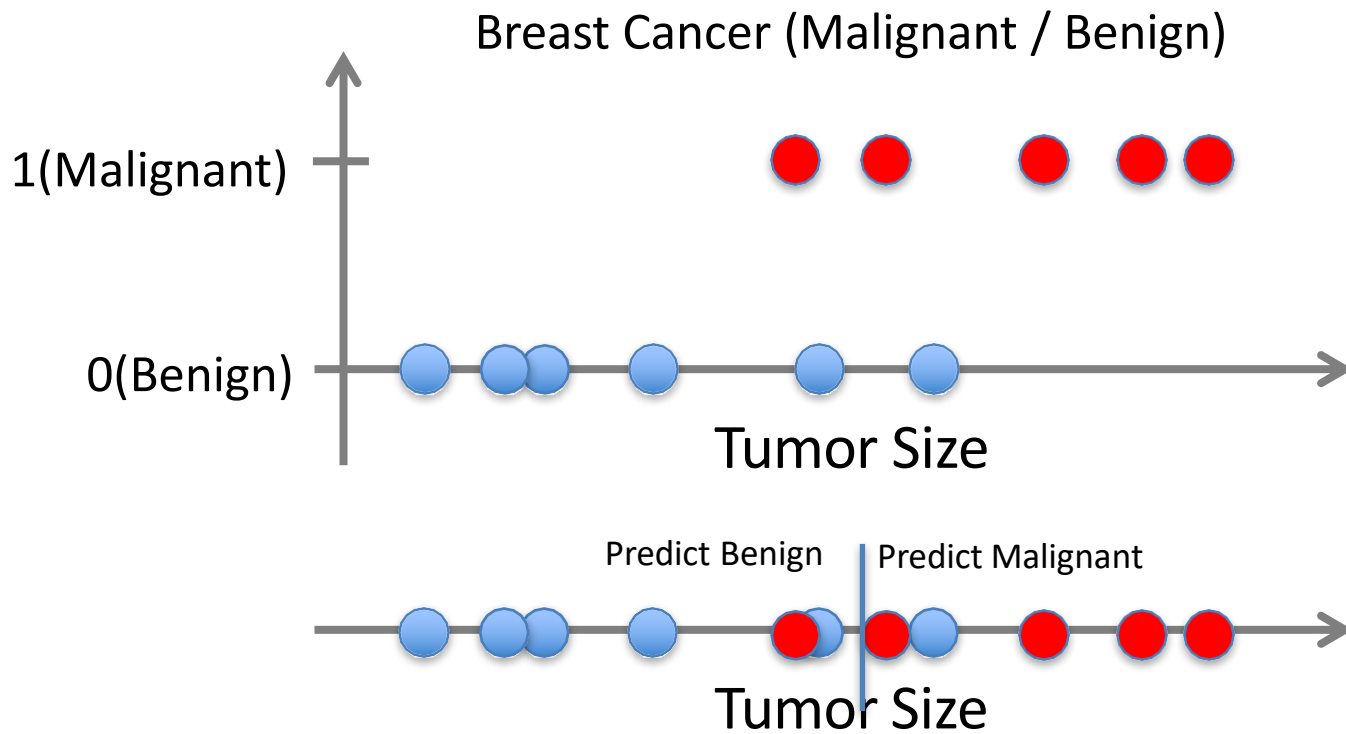
- 预测离散型变量
  - 首先基于一个包含 $x$ 值（输入特征），以及离散型的真实 $y$ 值（标签）的训练数据集构建分类模型；然后将该模型用来预测新的只包含 $x$ 值的测试数据集的 $y$ 值。
- 给定 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，学习一个输入 $x$ 、预测 $y$ 的函数 $f(x)$ 
  - $y$ 为离散型 == 分类

# 有监督学习：分类



Based on example by Andrew Ng

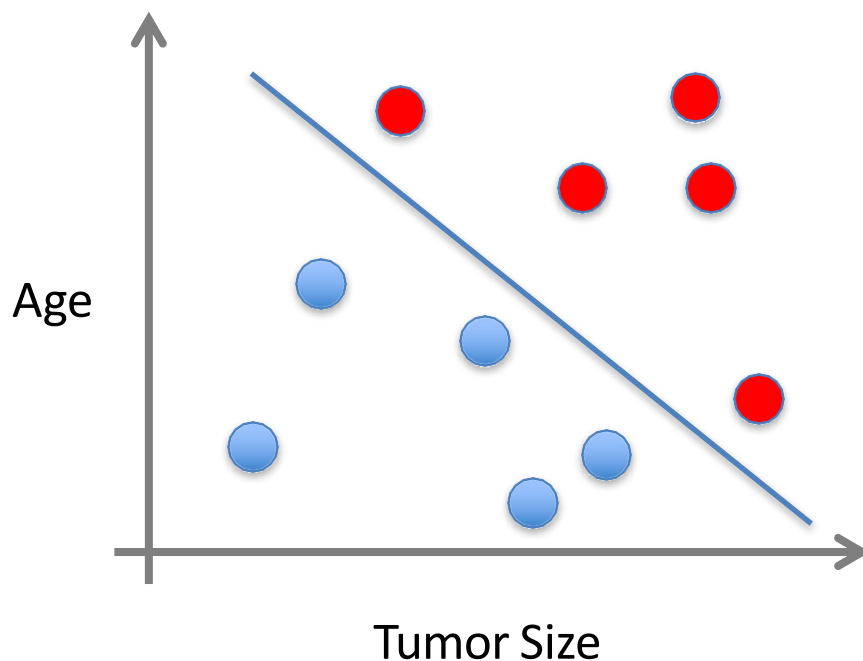
# 有监督学习：分类



Based on example by Andrew Ng

# 有监督学习：分类

- $x$  可以是多维的，即每个样本的输入特征都是一个向量。



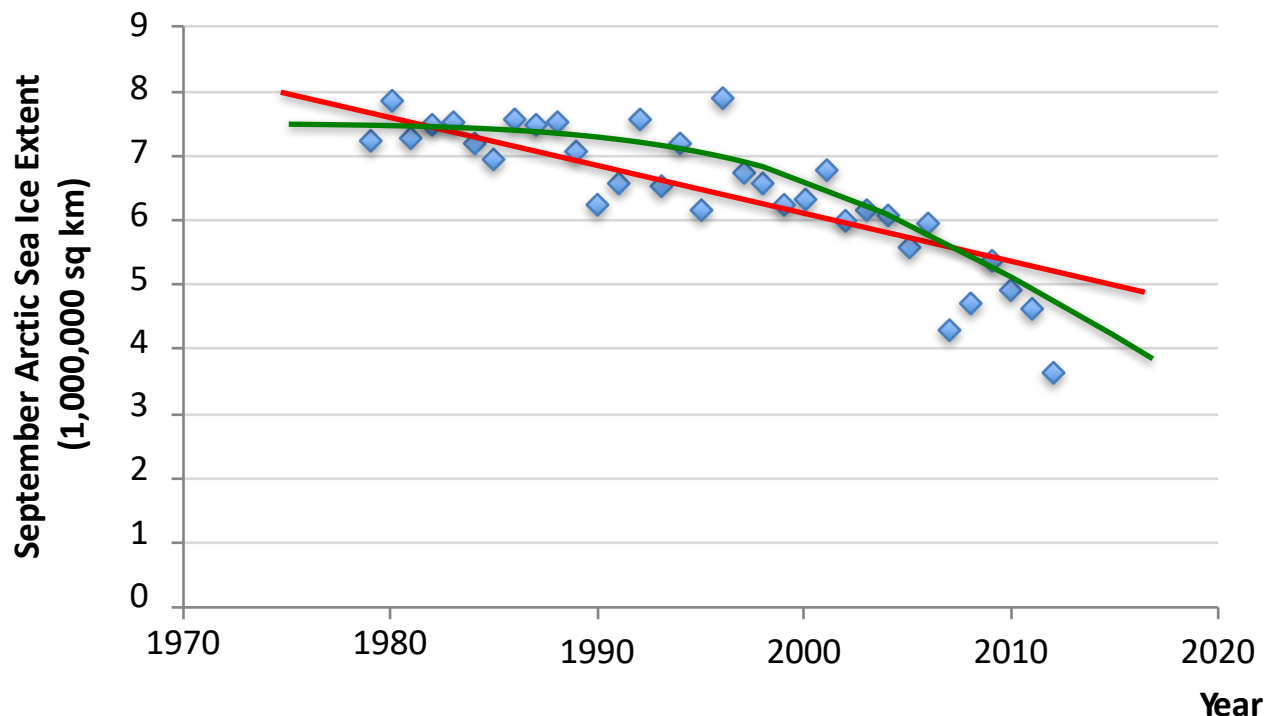
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- ...

Based on example by Andrew Ng

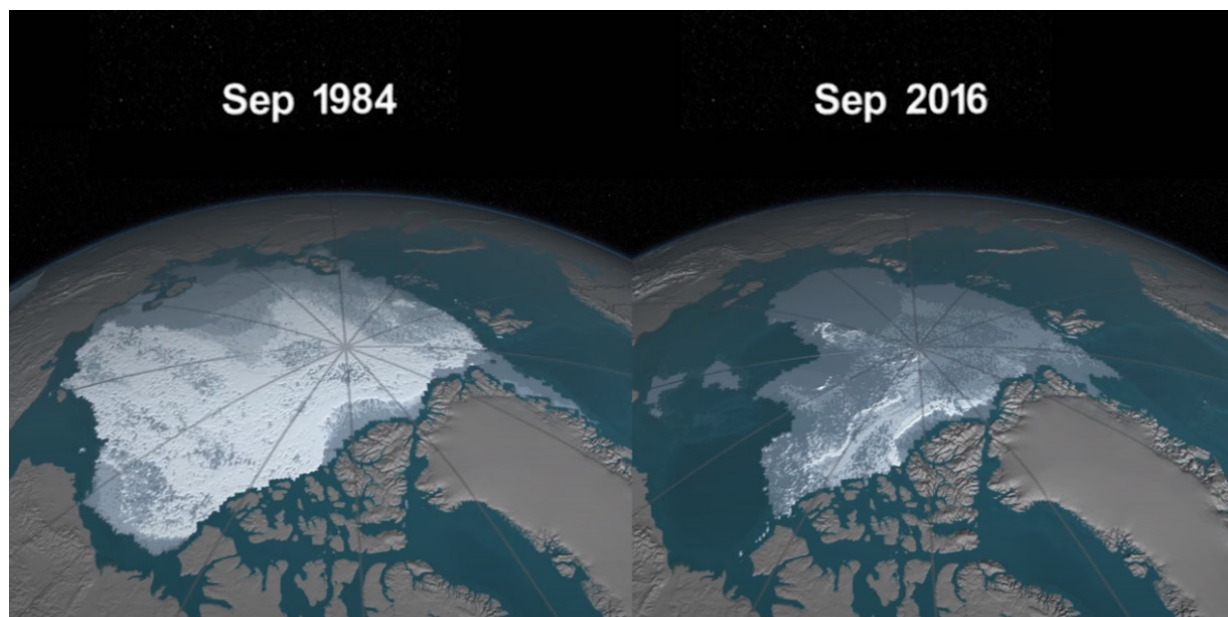


# 有监督学习：回归

- 给定 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , 学习一个输入 $x$ 、预测 $y$ 的函数 $f(x)$ 
  - $y$ 为连续型 == 回归

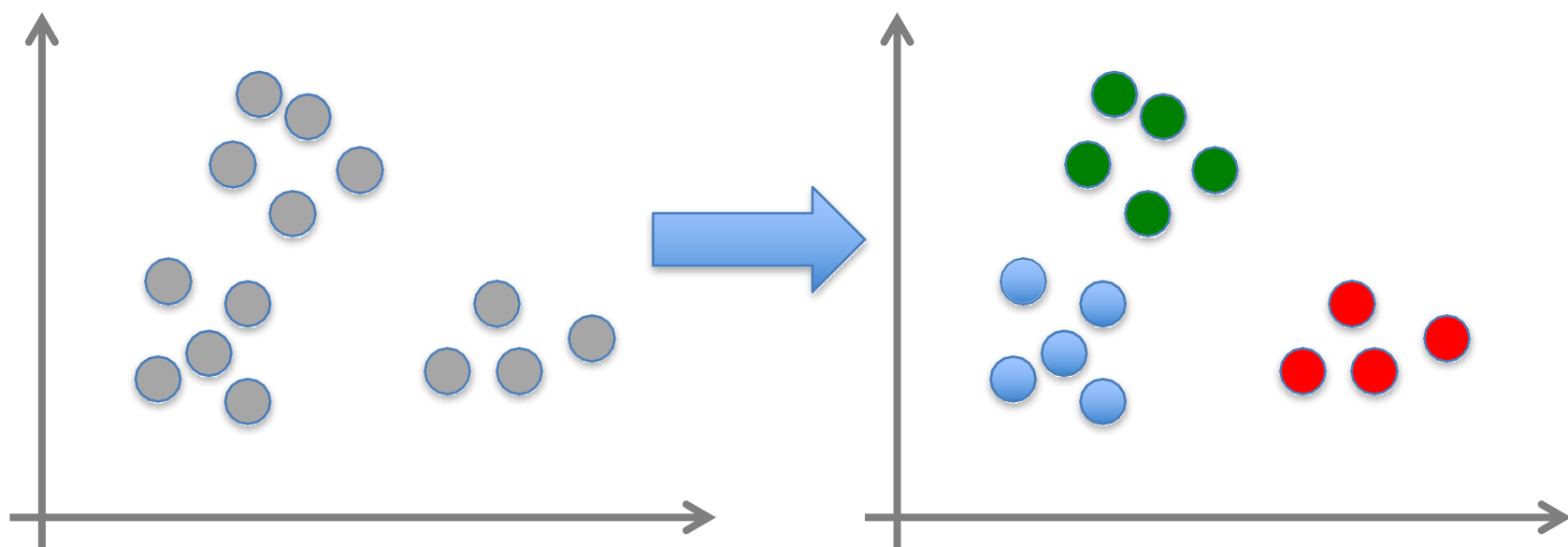


# 有监督学习：回归



# 无监督学习

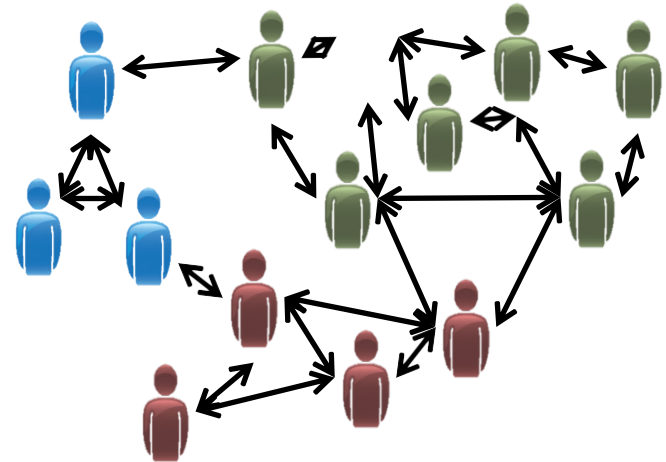
- 给定 $x_1, x_2, \dots, x_n$ ，无监督学习旨在输出 $x$ 中的隐含结构。
- 无监督学习中的聚类示例：



# 无监督学习



Organize computing clusters



Social network analysis



Market segmentation



Astronomical data analysis

# 混合学习

- 有监督学习具有分类精细、准确的优点，但学习过程复杂。
- 无监督学习具有分类灵活、算法简练的优点，但学习过程较慢。
- 混合学习过程一般事先用无监督学习抽取输入数据的特征，然后将这种内部表示提供给有监督学习进行处理，以达到输入输出的某种映射。由于对输入数据进行了预处理，将会使有监督学习以及整个学习过程加快。



# 强化学习

- 强化学习 (Reinforcement Learning) 就是智能系统从环境到行为进行映射的学习，其目的是使强化信号（回报函数值）最大。
- 强化学习中由环境提供的强化信号会对产生动作的好坏作一种评价，而不是告诉强化学习系统 (Reinforcement Learning System, RLS) 如何去产生正确的动作。由于外部环境提供的信息很少，RLS必须靠自身的经历进行学习。据此，RLS在行动-评价的环境中获得知识，并改进行动方案以适应环境。

# 机器学习的发展史

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

# 机器学习的发展史

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning



# 机器学习的发展史

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision
- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.

# 决策树

- 一种树状结构的分类模型
- 中间节点：表示基于某个属性进行训练数据集的划分，中间节点中指明该属性的取值
- 分支：用于展示某种划分方式的输出
- 叶子节点：表示按照当前分支得到的训练数据的类分布（class distribution）

# 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

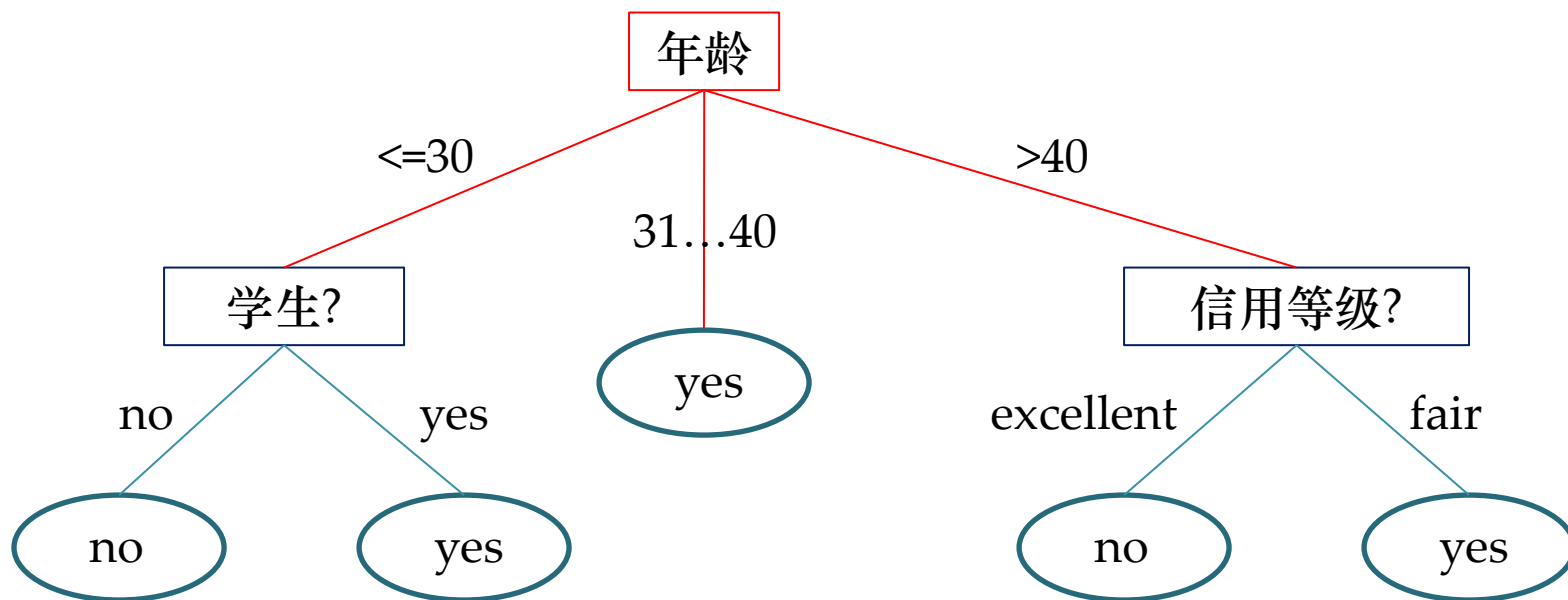
# 决策树

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

# 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

# 决策树



# 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

# 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



# 算法

- 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

- 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。

# 决策树

## 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。

# 决策树

## CLS算法

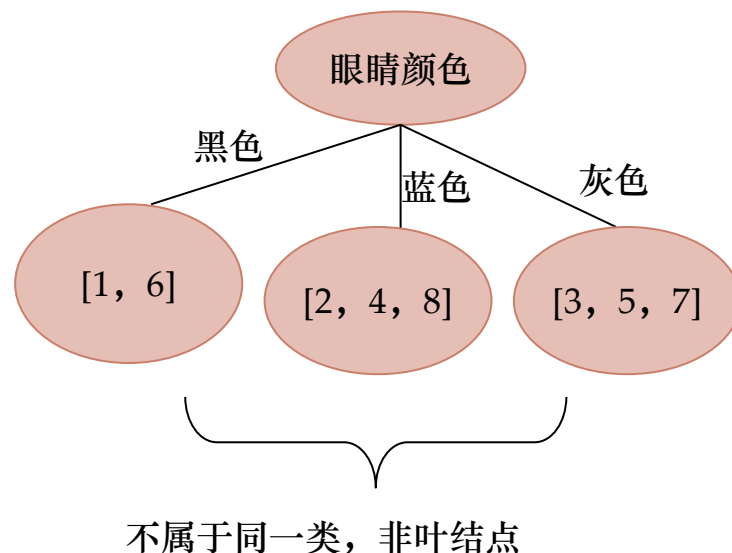
人员	眼睛颜色	头发颜色	所属人种
1	黑色	黑色	黄种人
2	蓝色	金色	白种人
3	灰色	金色	白种人
4	蓝色	红色	白种人
5	灰色	红色	白种人
6	黑色	金色	混血
7	灰色	黑色	混血
8	蓝色	黑色	混血

# 决策树

## CLS算法

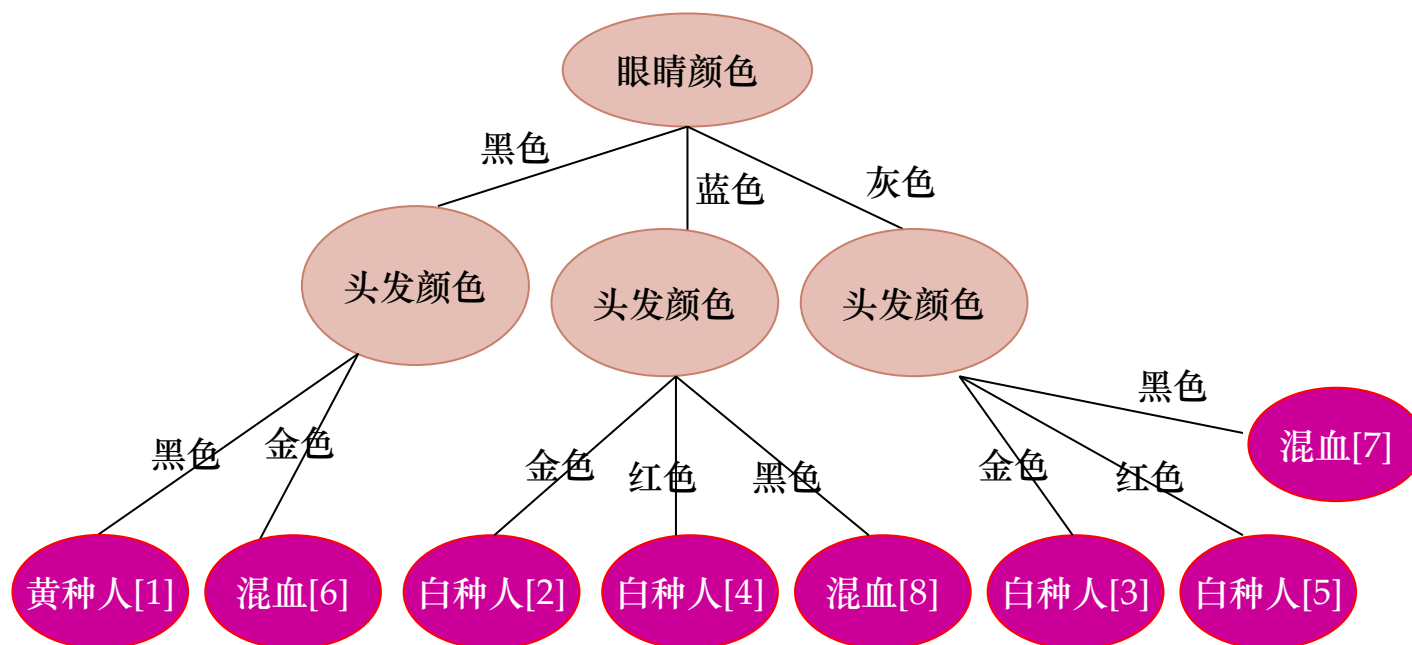
### 决策树的构建

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 $v_1, v_2, \dots, v_m$ ，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 $A$ ;
  - 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

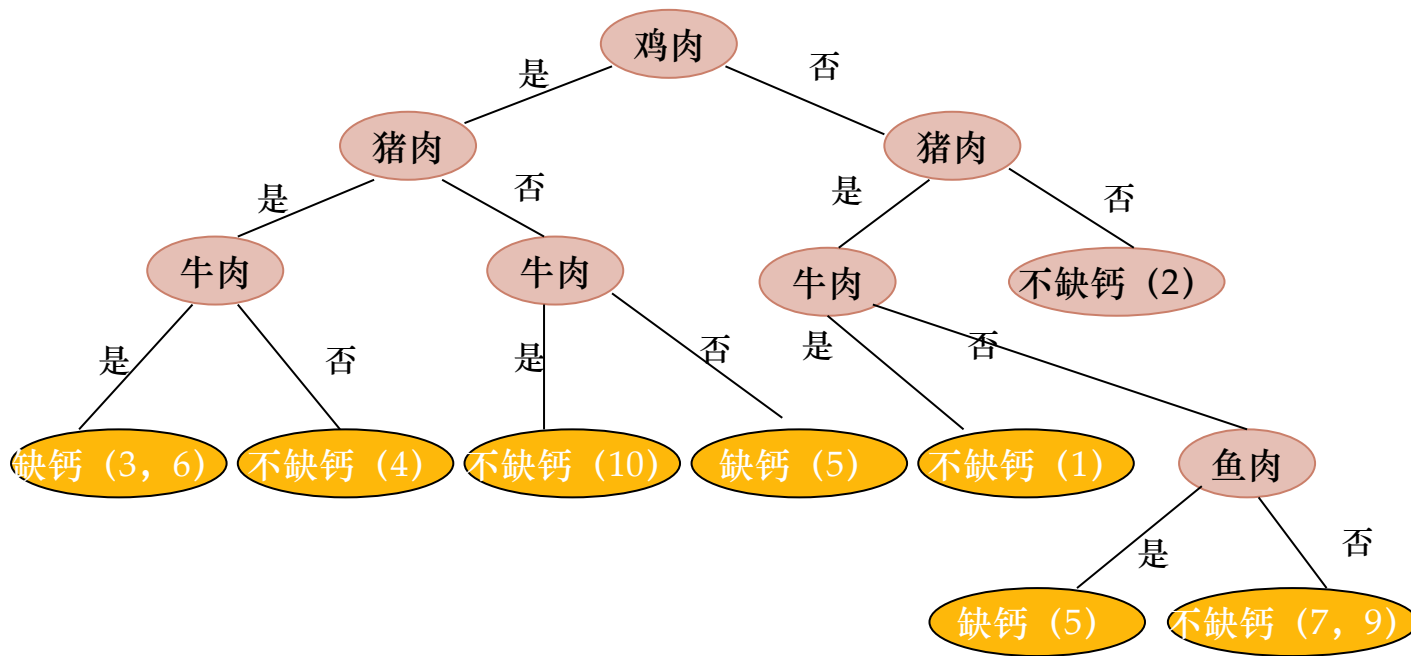
学生膳食结构和缺钙调查表

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	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

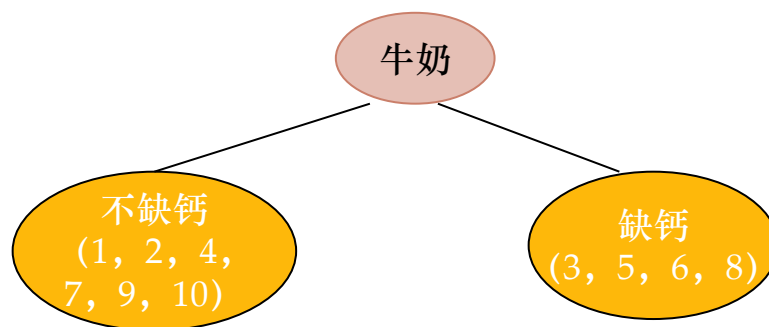
## CLS算法

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？**

# 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)

# 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)

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

# 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)

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

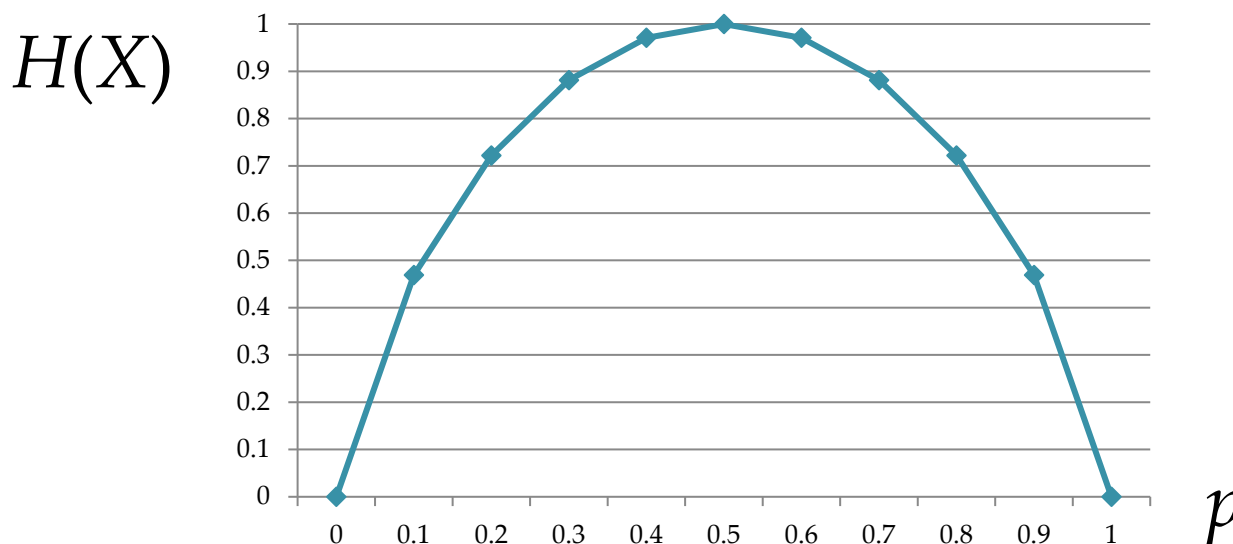
$$= \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

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

# 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:

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

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

- Class label: 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
- $H(D)=0.940$  14个训练样本中，9个买了电脑

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

- $H(D | A = \text{"年龄"}) = 0.694$

$$\begin{aligned} H(D | A = \text{"年龄"}) &= \frac{5}{14} \times \left( -\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\ &+ \frac{4}{14} \times \left( -\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) + \frac{5}{14} \times \left( -\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \end{aligned}$$

# 基于信息增益的ID3模型

计算 $D$  (类标签) 和 $A$  (每个属性)的互信息

- $H(D)=0.940$
- $H(D|A=\text{“年龄”})=0.694$

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

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

# 基于信息增益的ID3模型

- 类标签: 是否买电脑="yes/no"
- 对  $D$  和 每个属性  $A$  , 计算互信息 (mutual information)

- $H(D)=0.940$

- $H(D|A=\text{"年龄"})=0.694$

- $g(D,A=\text{"年龄"})=0.246$

- $g(D,A=\text{"收入"})=0.029$

- $g(D,A=\text{"学生?"})=0.151$

- $g(D,A=\text{"信用等级?"})=0.048$

**“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**

$$g(D, A)$$

$$= H(D)$$

$$- H(D | A)$$



# 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	$\leq 30$	high	no	fair	no
u2	$\leq 30$	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	$\leq 30$	medium	no	fair	no
u9	$\leq 30$	low	yes	fair	yes
u10	$> 40$	medium	yes	fair	yes
u11	$\leq 30$	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

- $\text{GainRatio}_A(D) = \text{Gain}_A(D) / \text{SplitInfo}_A(D)$

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

- $\text{GainRatio}_{A=\text{"income"}}(D) = ?$

# 基于增益率的C4.5模型

- $\text{GainRatio}_A(D) = \text{Gain}_A(D) / \text{SplitInfo}_A(D)$

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

- $\text{GainRatio}_{A=\text{"income"}}(D) = ?$

$$\text{SplitInfo}_{A=\text{"income"}}(D)$$

$$\begin{aligned} &= -\frac{4}{14} \times \log_2 \left( \frac{4}{14} \right) - \frac{6}{14} \times \log_2 \left( \frac{6}{14} \right) - \frac{4}{14} \times \log_2 \left( \frac{4}{14} \right) \\ &= 0.926 \end{aligned}$$

- $\text{GainRatio}_{A=\text{"income"}}(D) = 0.029 / 0.926 = 0.031$

# 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  $\text{gini}_{\text{split}}(D)$  定义如下

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

- 具有最小  $\text{gini}_{\text{split}}(D)$  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”

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

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

# 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”

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

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

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



# 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？

# 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,  $a_i$  and  $a_{i+1}$  的中点是

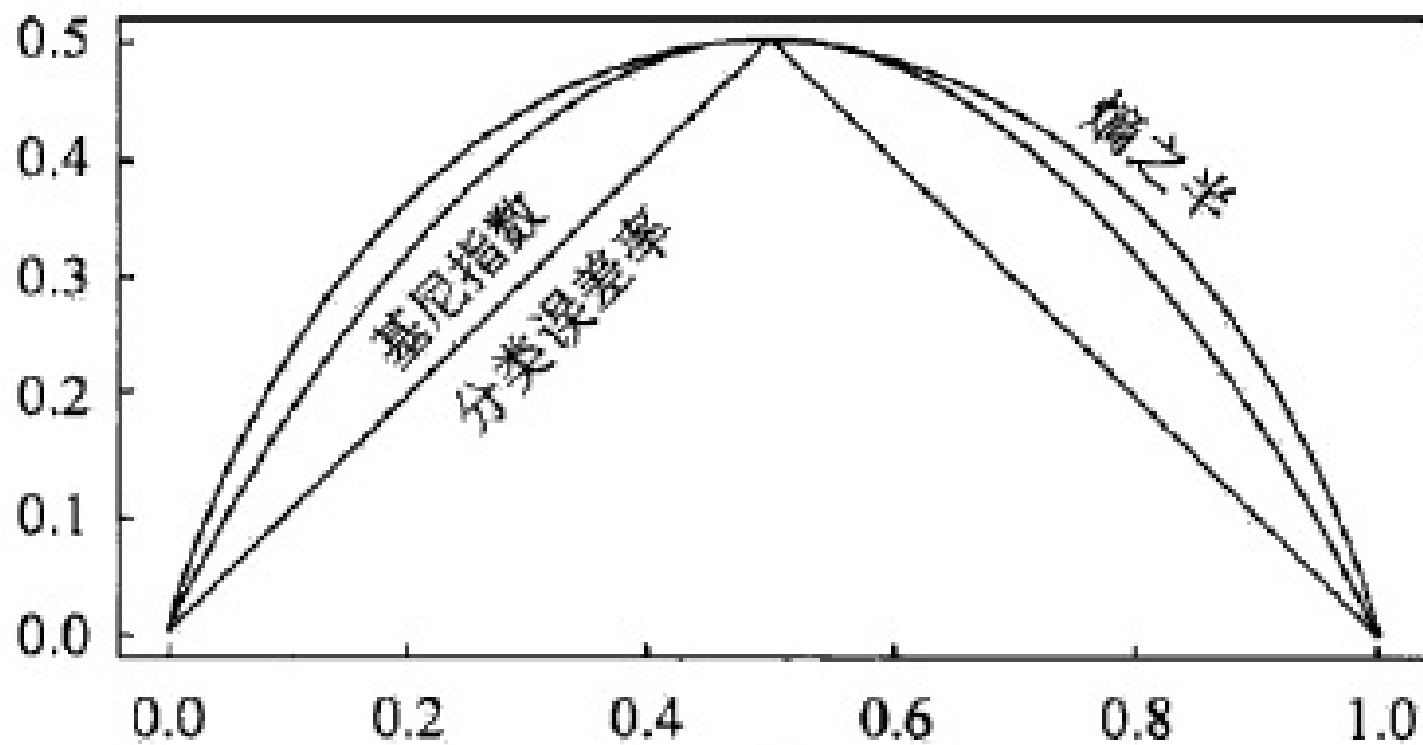
$$(a_i + a_{i+1}) / 2$$

# 基于Gini指数的CART模型

## CART（分类树）的生成

- CART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 $D_1$ ， $D_2$ ，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)。

# 回归树

## CART（回归树）的生成

- 回归树的生成
- 设 $y$ 是连续变量，给定训练数据集： $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
- 假设已将输入空间划分为 $M$ 个单元 $R_1, R_2, \dots, R_M$ ，并且每个单元 $R_m$ 上有一个固定的输出 $c_m$ ，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  $\sum_{x_i \in R_m} (y_i - f(x_i))^2$
- $R_m$ 上的 $c_m$ 的最优值： $\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$

# 回归树

## 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：

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

- 然后寻找最优切分变量和切分点：

$$\min_{j, s} \left[ \min_{c_1} \sum_{x_i \in R_1(j, s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j, s)} (y_i - c_2)^2 \right]$$

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

# 回归树

## CART（回归树）的生成

- 最小二乘回归树生成算法

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

$$\min_{j,s} \left[ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$

遍历变量  $j$ ，对固定的切分变量  $j$  扫描切分点  $s$ ，选择使式 (1) 达到最小值的对  $(j,s)$



# 回归树

## CART（回归树）的生成

- 最小二乘回归树生成算法

(2) 用选定的对  $(j, s)$  划分区域并决定相应的输出值:

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

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

(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