



# 人工智能

## 第八讲：监督学习



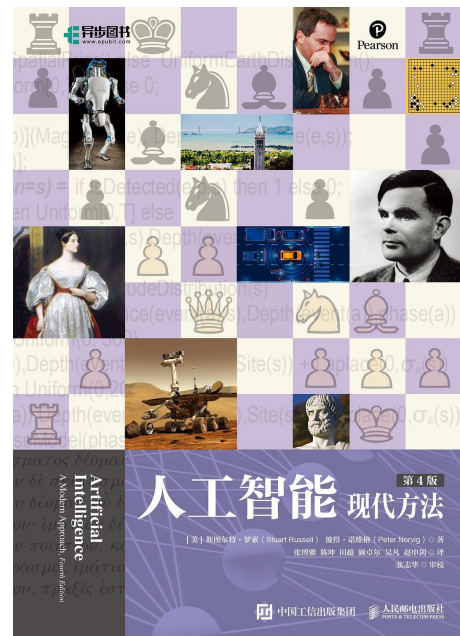
# 第8讲：监督学习

## • 章节概要

- 8.1 监督学习的主要任务：分类与回归
- 8.2 决策树分类
- 8.3 从二分类到多分类
- 8.4 线性回归
- 8.5 本章小结

## • 参考书目

- 《人工智能：现代方法（第4版）》（美）  
罗素,（美）诺维格, 人民邮电出版社,  
2022。Ch19.2,19.3,19.6
- 周志华, 《机器学习》, 清华大学出版社,  
2016。Ch3-Ch4





# 8.1 监督学习：分类与回归



根据样本数据的**标记 (label)** 特性，可将机器学习任务分为：

监督学习：样本特征 $x$ 均**有**对应的样本标记 $y$

无监督学习：样本特征 $x$ 均**没有**对应的样本标记 $y$

半监督学习：样本特征 $x$  **(大) 部分没有**对应的样本标记 $y$

强化学习：可近似理解为具有**延迟标记信息**

注解：样本特征 (feature)  $\mathbf{x} = (x_1; x_2; \dots; x_d)$ ，粗体表示向量，分号“;”表示列向量（ $d$ 个分量按列拼接），逗号“,”表示行向量（ $d$ 个分量按行拼接）



监督学习：样本特征 $x$ 均**有**对应的样本标记 $y$

当标记为**离散**变量时：分类（定性）

二分类：标记类型2种

例如，一张图片预测是不是人脸

多分类：标记类型 $>2$ 种

一张图片预测是猫、狗、还是老虎

姓名	职业	年收入	...	好顾客
张三	教师	6 万	...	否
李四	公务员	7 万	...	是
王五	学生	2 万	...	否
周六	企业家	15 万	...	是
董七	演员	13 万	...	否
钱八	教师	8 万	...	是

当标记为**连续**变量时：回归（定量）

例如，房价预测、GDP增速预测

人均 犯罪率	.....	环保 指标	房价 w/m <sup>2</sup>
0.00632	.....	0.5380	3.1
.....			
0.04741	.....	0.5730	2.2
特征 <sub>1</sub>	.....	特征 <sub>d</sub>	标记

样例/样本



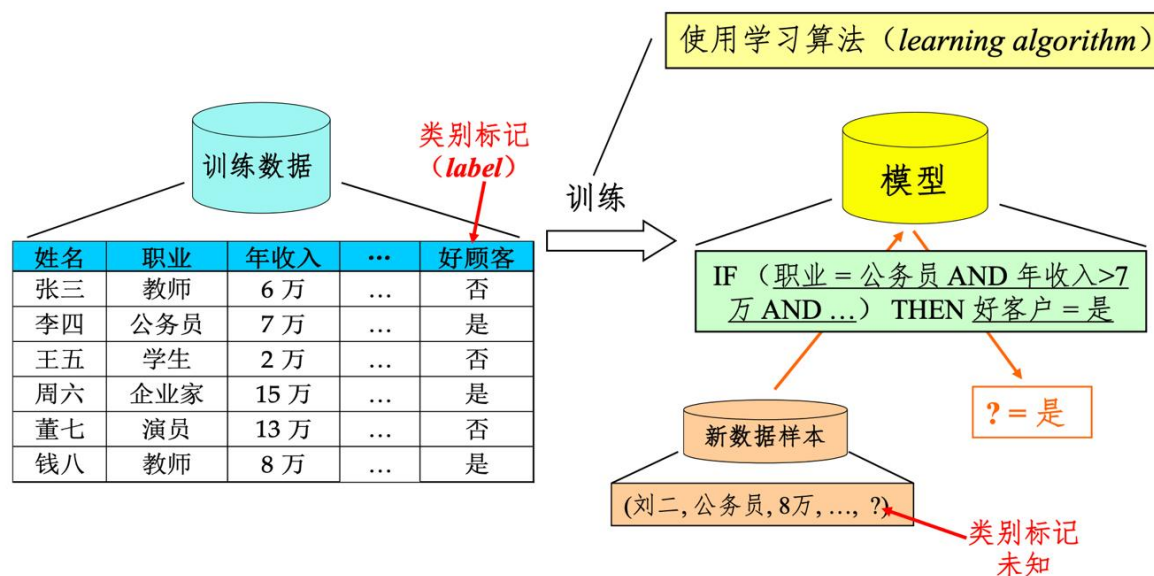
**监督学习**：样本特征 $x$ 均有**有**对应的样本标记 $y$

当标记为**离散**变量时：分类

当标记为**连续**变量时：回归

监督学习的目标：**预测**（即解决“**是什么**”的问题）

$(x, y)$      $(x, ?)$   
 $(x, y)$      $(x, ?)$   
 $(x, y)$      $(x, ?)$   
 $(x, y)$      $(x, ?)$



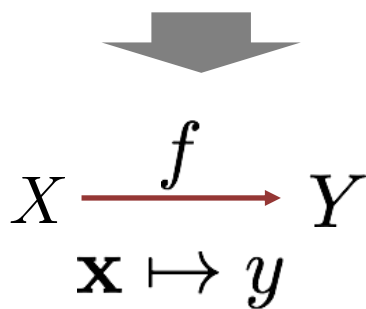
学到的模型能够较好地适用于未来的测试样本（泛化能力），而不仅仅是在已有的训练样本上表现良好



监督学习的目标：**预测**（即解决“**是什么**”的问题）

为了达到预测这一目标，可从训练数据集 $D$ 中**学习**特征空间 $X$ 到标记空间 $Y$ 的**映射 $f$**

训练集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$



对新样本特征  $\mathbf{x}_{n+1}$  进行标记预测： $f(\mathbf{x}_{n+1})$

**训练阶段：**从训练集中学习 $f$ ，作为对真实 $f^*$ 的近似/逼近

**测试阶段：**对真实标记 $y_{n+1}$ 进行预测



## 8.2 决策树分类





GPT-4 >

Context cleared

在人工智能课上，讲一个包含机器学习知识的，有关 决策树的笑话

有一天，一棵决策树和它的朋友们在聚会上玩游戏。他们决定玩一个猜数字游戏，让决策树来猜一个人选的数字。

决策树问：“数字是否大于50？”

朋友回答：“是的。”

决策树接着问：“数字是否小于75？”

朋友回答：“是的。”

决策树再问：“数字是否是偶数？”

朋友回答：“不是。”

于是决策树很自信地说：“数字是71！”

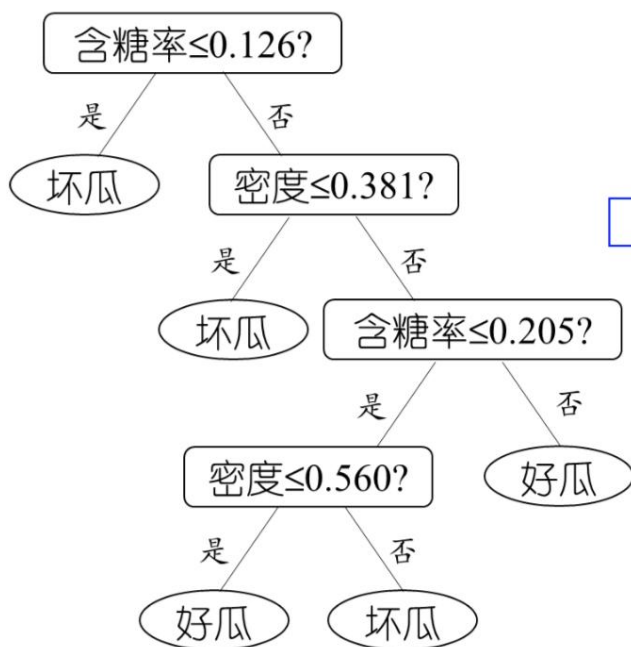
朋友惊讶地问：“哇！你怎么知道的？”

决策树得意地回答：“通过不断地分割问题空间，我可以迅速找到答案。这就是我的‘根’本之道！”

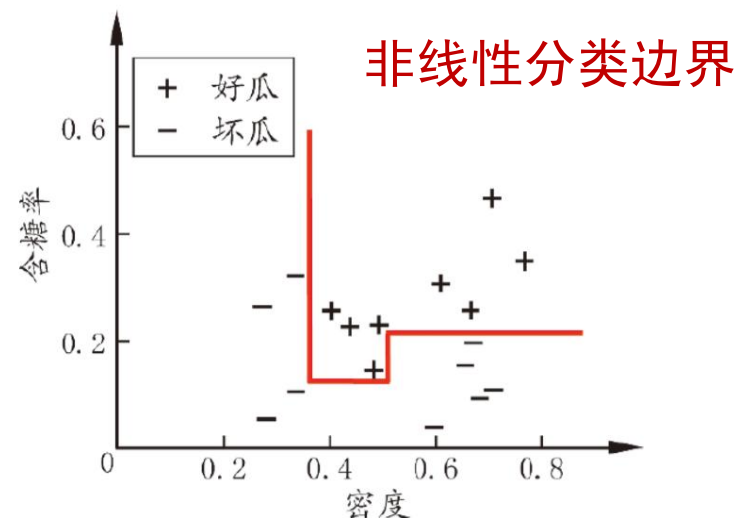


# 决策树

决策树：非线性建模特征空间 $X$ 到标记空间 $Y$ 之间的映射关系  
以二（元）分类任务为例，样本标记只可取两个值中的一者



决策树



决策树的决策边界与轴平行



# 训练集



例子来源：《机器学习》

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



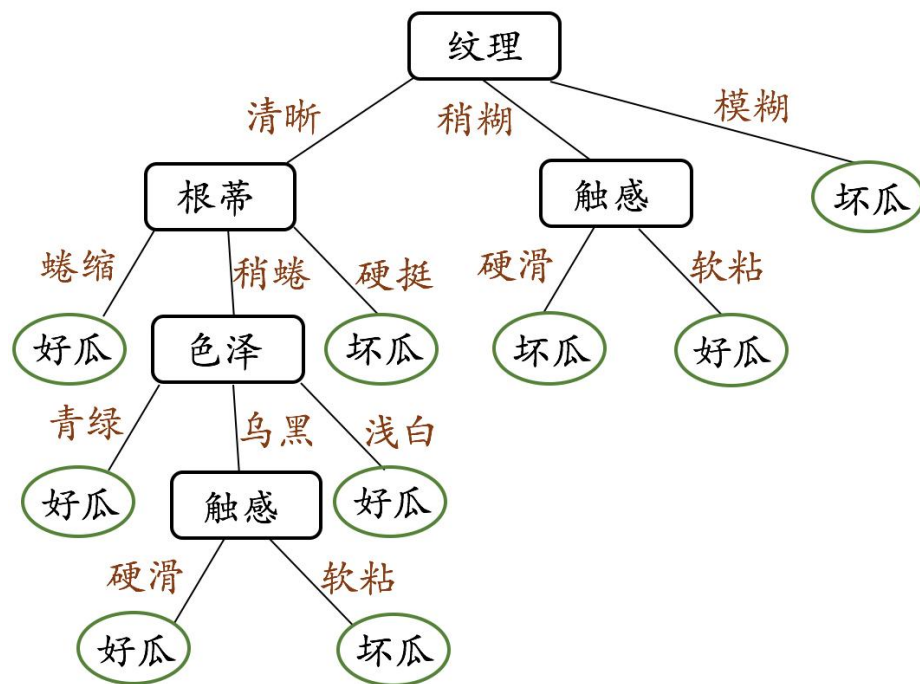
# 决策树

决策树（decision tree）：基于树结构进行决策

- 树的每个内部结点对应某个特征/属性上的判定
- 每个分支对应上述判定的一种可能结果（该属性的某个取值）
- 每个叶结点对应一个预测结果

**训练：**通过分析训练样本，确定内部结点所对应的特征/属性（划分特征/属性）

**预测：**将测试样本从根结点开始，沿着划分属性所构成的判定序列下行，直到叶结点





## 训练决策树的基本思想：分而治之

- 自根结点至叶结点进行递归
- 为树的中间结点找到一个划分属性

## 训练算法的三种终止条件：

- ① 当前结点包含的样本全属于同一类别，无需划分
- ② 当前属性集为空，或者是所有样本在所有属性上取值相同，无法划分
- ③ 当前结点包含的样本集合为空，不能划分



## 算法 1 决策树算法

输入:

训练集  $D = \{(\mathbf{x}_k, y_k)\}_{k=1}^m$ ;

属性集  $A = \{a_1, a_2, \dots, a_d\}$ .

过程: 函数 TreeGenerate( $D, A$ )

1: 生成结点 node

2: **if**  $D$  中样本全属于同一类别  $C$  **then**

3: 将 node 标记为  $C$  类叶结点; **return**

4: **end if**

递归返回,  
情形 (1)

5: **if**  $A = \emptyset$  或者  $D$  中样本在  $A$  上取值相同 **then**

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

7: **end if**

递归返回,  
情形 (2)

8: 从  $A$  中选择最优划分属性  $a_*$

9: **for**  $a_*$  的每一个值  $a_*^v$  **do**

10: 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

11: **if**  $D_v = \emptyset$  **then**

12: 将分支结点标记为叶结点, 其类别标记为  $D$  中样本数最多的类; **return**

13: **else**

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

15: **end if**

16: **end for**

17: 输出: 以 node 为根结点的一棵决策树

递归返回,  
情形 (3)



# 决策树

决策树算法的核心：如何从属性集A中选择最优的划分属性 $a_*$

随着划分过程不断进行，希望决策树的分支结点所包含的样本尽可能属于同一类别，即结点的“纯度”(purity)越来越高

- 使决策树得到关注并成为机器学习主流技术的算法: ID3  
1979年, 由J. R. Quinlan首先提出  
基于信息增益 (Information Gain)
- 最常用的决策树算法: C4.5  
1993年, 由J. R. Quinlan首先提出  
基于增益率 (Gain Ratio)
- 可用于回归的决策树算法: CART (Classification and Regression Tree) 1984年, 由L. Breiman等人首先提出  
基于基尼指数 (Gini Index)





# 决策树：ID3

ID3算法：基于信息增益（Information Gain）选择最优划分属性

信息增益以信息熵为基础，计算当前划分对信息熵造成的变化

信息熵可以用来度量样本集合的纯度

若当前样本集合D中第k类样本所占的比例为 $p_k$ ，则D的信息熵为：

$$H(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$$

计算信息熵时约定：若  $p = 0$ ，则  $p \log_2 p = 0$ 。  $H(D)$  的值越小，则  $D$  的纯度越高。

思考：  $H(D)$  何时最小，何时最大，分别对应哪种情形？





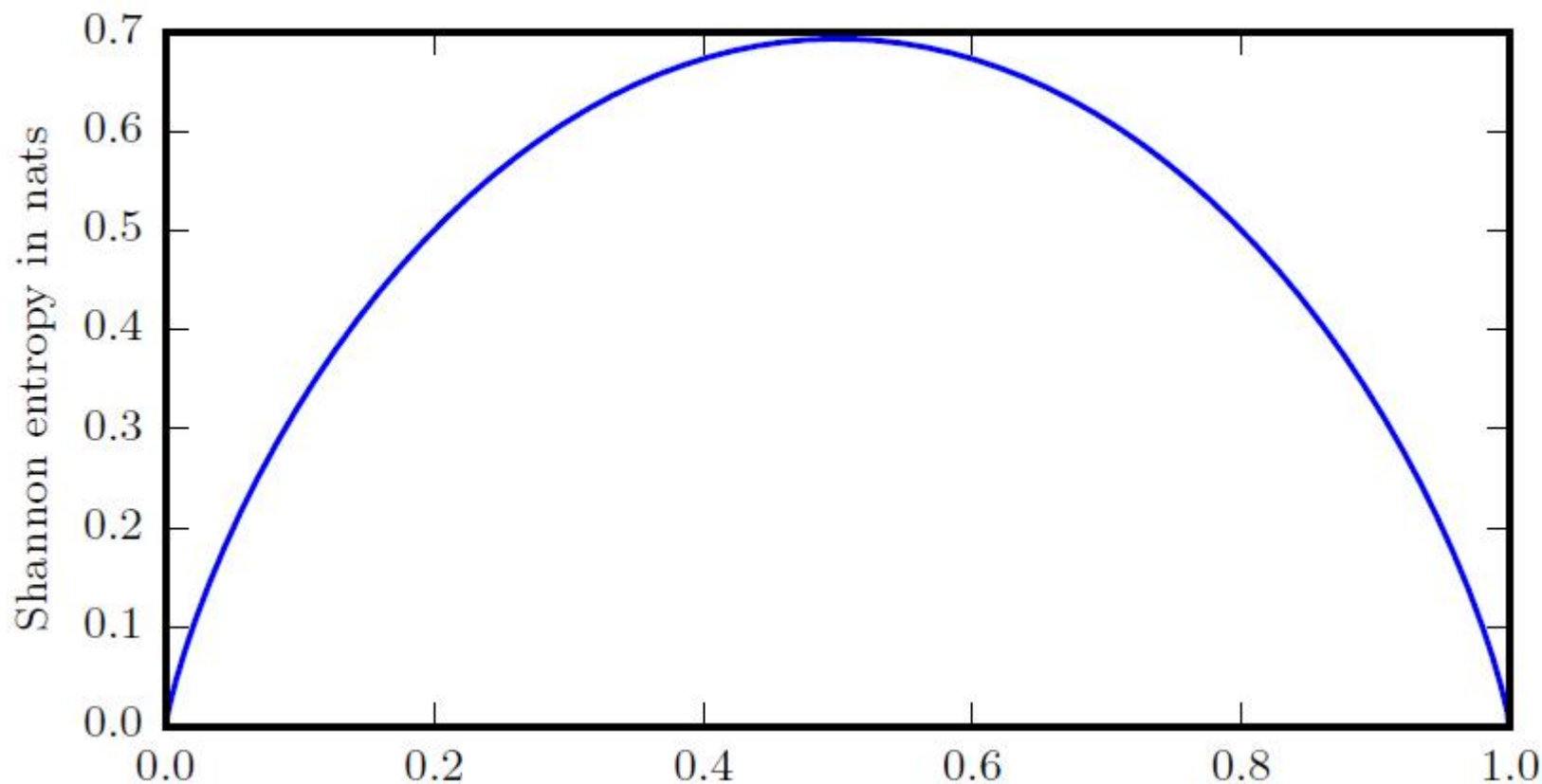
# 决策树：ID3

若当前样本集合 $D$ 中第 $k$ 类样本所占的比例为 $p_k$ ，则 $D$ 的信息熵为：

$$H(D) = - \sum_{k=1}^{|Y|} p_k \log_2 p_k$$

$$H(p) = -p \cdot \log p - (1 - p) \cdot \log(1 - p)$$

计算信息熵时约定：若  $p = 0$ ，则  $p \log_2 p = 0$ 。 $H(D)$  的值越小，则  $D$  的纯度越高。





# 决策树：ID3

信息增益以**信息熵**为基础，计算当前划分对信息熵造成的变化

如何计算信息增益？

离散属性  $a$  的取值： $\{a^1, a^2, \dots, a^V\}$

$D^v$ ： $D$  中在  $a$  上取值等于  $a^v$  的样本集合

以属性  $a$  对数据集  $D$  进行划分所获得的信息增益定义为：

$$G(D, a) = \underbrace{H(D)}_{\text{划分前的信息熵}} - \underbrace{\sum_{v=1}^V \left[ \frac{|D^v|}{|D|} \cdot H(D^v) \right]}_{\text{划分后的信息熵}}$$

第 $v$ 个分支的权重，样本越多越重要

选择使得当前样本集合**信息增益最大的属性** $a$ 作为当前的**划分属性**



## ID3举例

该数据集含 17 个训练样本， $|\mathcal{Y}| = 2$ ，  
正类占  $p_1 = 8/17$ ，负类占  $p_1 = 9/17$

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

例子来源:《机器学习》



# 决策树：ID3

## ID3举例

根结点的信息熵为：

$$H(D) = - \sum_{k=1}^2 p_k \log_2 p_k = - \left( \frac{8}{17} \log_2 \frac{8}{17} + \frac{9}{17} \log_2 \frac{9}{17} \right) = 0.998$$



# 决策树：ID3

## ID3举例

以属性色泽为例，其对应的 3 个数据子集分别为： $D^1$  (色泽为青绿)， $D^2$  (色泽为乌黑)， $D^3$  (色泽为浅白)。

子集  $D^1$  包含编号为 {1, 4, 6, 10, 13, 17} 这 6 个样本，其中正类占 3/6，负类占 3/6， $D^2$  和  $D^3$  同理，从而 3 个结点的信息熵为：

$$H(D^1) = -(\frac{3}{6} \log_2 \frac{3}{6} + \frac{3}{6} \log_2 \frac{3}{6}) = 1.000$$

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

$$H(D^3) = -(\frac{1}{5} \log_2 \frac{1}{5} + \frac{4}{5} \log_2 \frac{4}{5}) = 0.722$$

属性色泽的信息增益为：

$$\begin{aligned} G(D, \text{色泽}) &= H(D) - \sum_{v=1}^3 \frac{|D^v|}{|D|} \cdot H(D^v) \\ &= 0.998 - (\frac{6}{17} \times 1.000 + \frac{6}{17} \times 0.918 + \frac{5}{17} \times 0.722) \\ &= 0.109 \end{aligned}$$

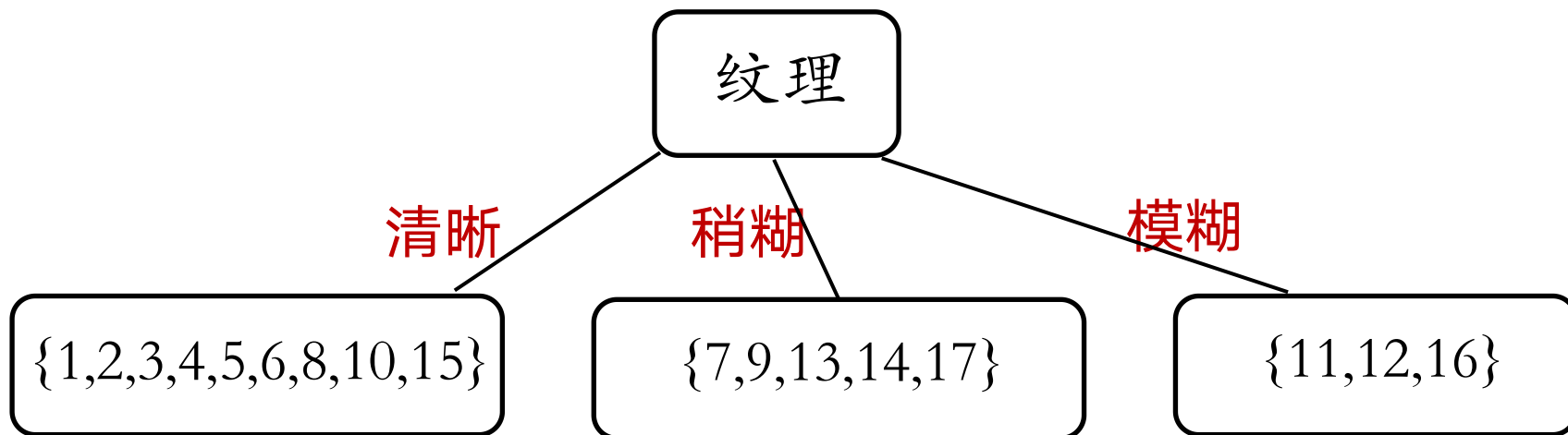


## ID3举例

同理可得其它属性的信息增益：

$$G(D, \text{根蒂}) = 0.143, G(D, \text{敲声}) = 0.141, G(D, \text{纹理}) = 0.381, \\ G(D, \text{脐部}) = 0.289, G(D, \text{触感}) = 0.006.$$

由于纹理这一属性的信息增益最大，从而被选为划分属性  
第一次划分后所建的树：

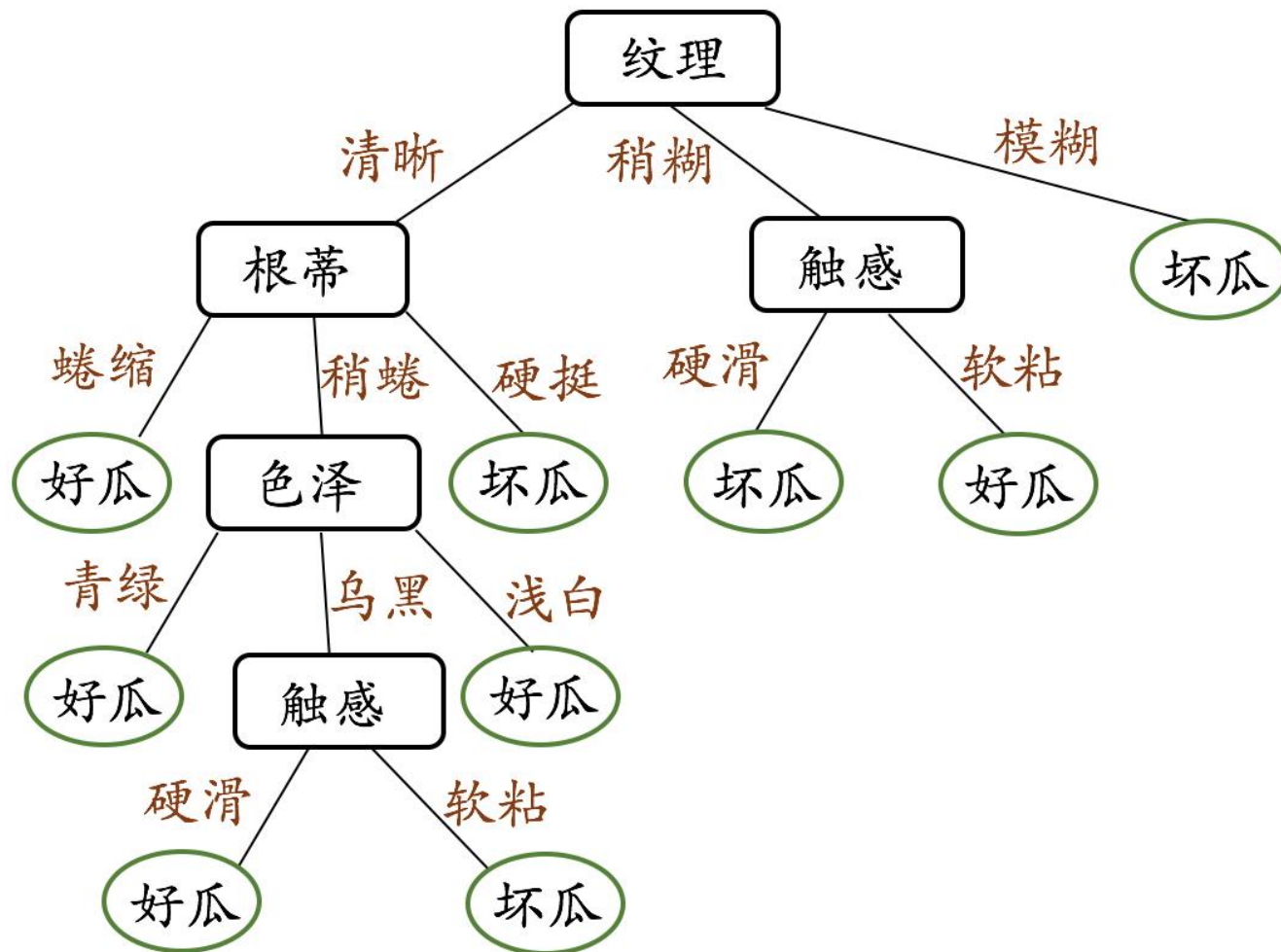




# 决策树：ID3

## ID3举例

对分支结点进一步划分，最终可得决策树








# 决策树：ID3

基于信息增益选择划分属性的缺点：

偏向于选择可取值数目较多的属性（比如会将样本编号作为一个划分属性来选择，这明显不合理）

$$G(D, a) = H(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \cdot H(D^v)$$



**= 0**

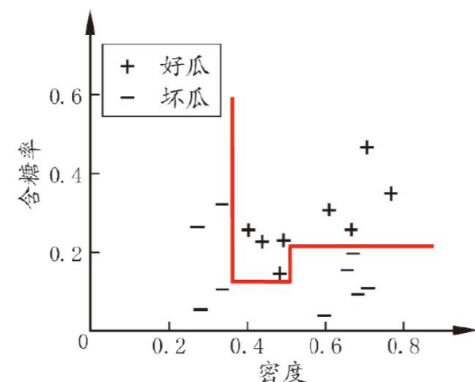
泛化能力低，可能无法对新样本进行有效预测！



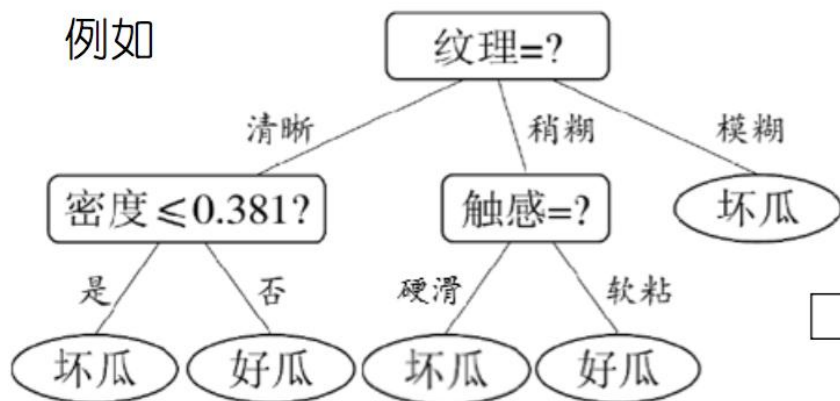


## 从树到规则

- 一棵决策树对应于一个“规则集”
- 每个从根结点到叶结点的分支路径对应于一条规则



例如



- IF (纹理=清晰)  $\wedge$  (密度 $\leq 0.381$ ) THEN 坏瓜
- IF (纹理=清晰)  $\wedge$  (密度 $> 0.381$ ) THEN 好瓜
- IF (纹理=稍糊)  $\wedge$  (触感=硬滑) THEN 坏瓜
- IF (纹理=稍糊)  $\wedge$  (触感=软粘) THEN 好瓜
- IF (纹理=模糊) THEN 坏瓜

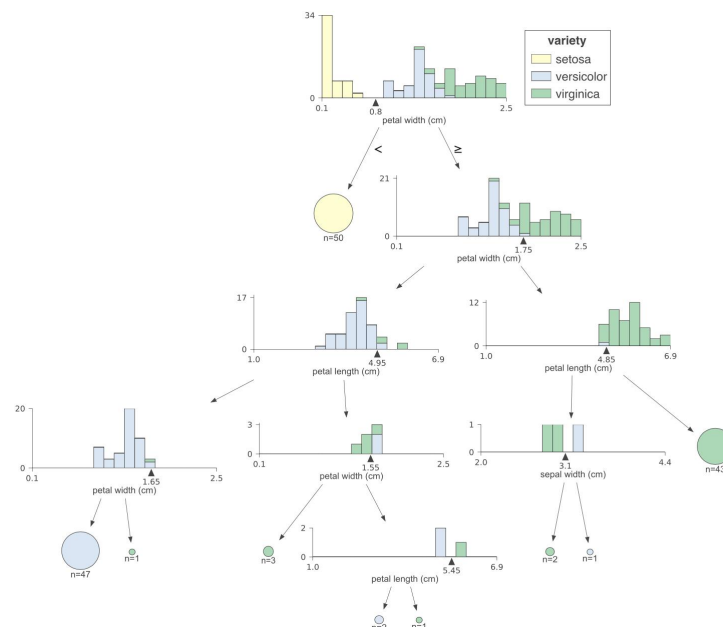
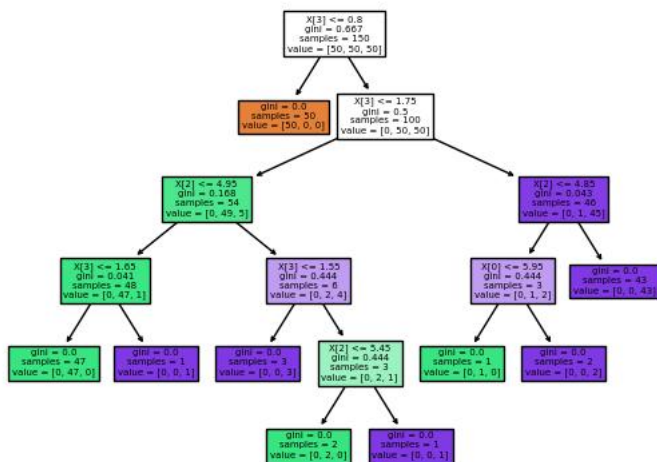
好处:

- 改善可理解性
- 进一步提升泛化能力

由于转化过程中通常会进行前件合并、删减等操作，最终规则集的泛化性能可能优于原决策树

## 决策树相关工具包

- Scikit-learn
  - <https://scikit-learn.org/stable/modules/tree.html>
- 决策树可视化
  - <https://explained.ai/decision-tree-viz/index.html>





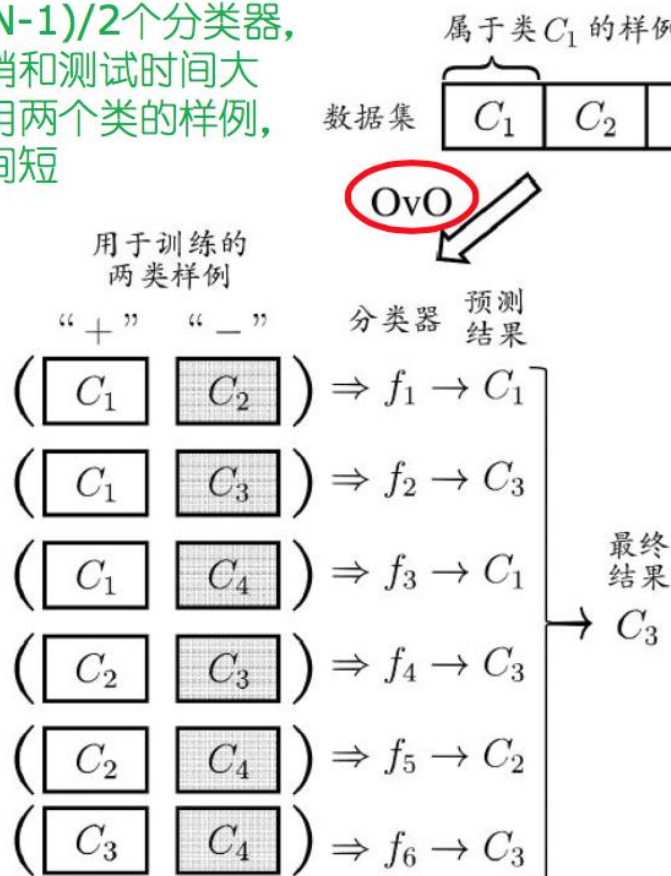
## 8.3 从二分类到多分类



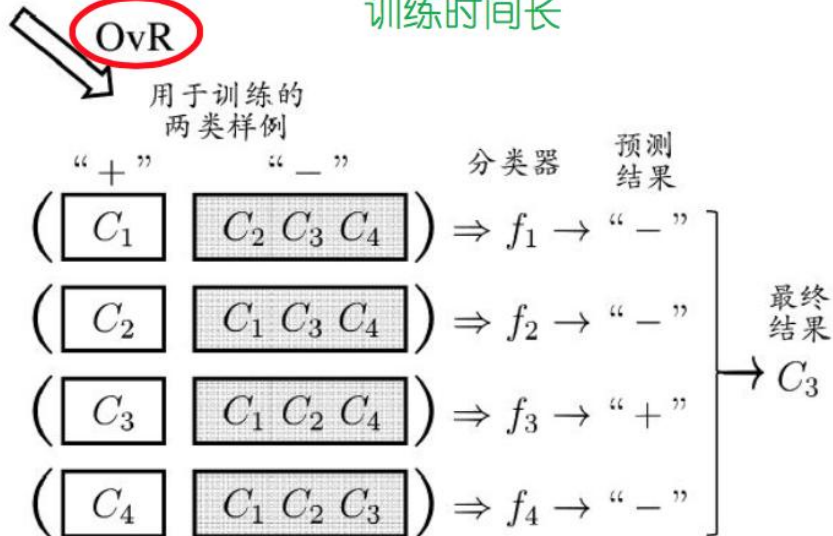
多（元）分类任务：样本标记可取多个值（>2）中的一者

拆解法：将一个多分类任务拆分为若干个二分类任务求解

- 训练 $N(N-1)/2$ 个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短



- 训练 $N$ 个分类器，存储开销和测试时间小
- 训练用到全部训练样例，训练时间长



预测性能取决于具体数据分布，多数情况下两者差不多

注解：OvO指One vs. One，即一对一；OvR指One vs. Rest，即一对多



## 8.4 线性回归

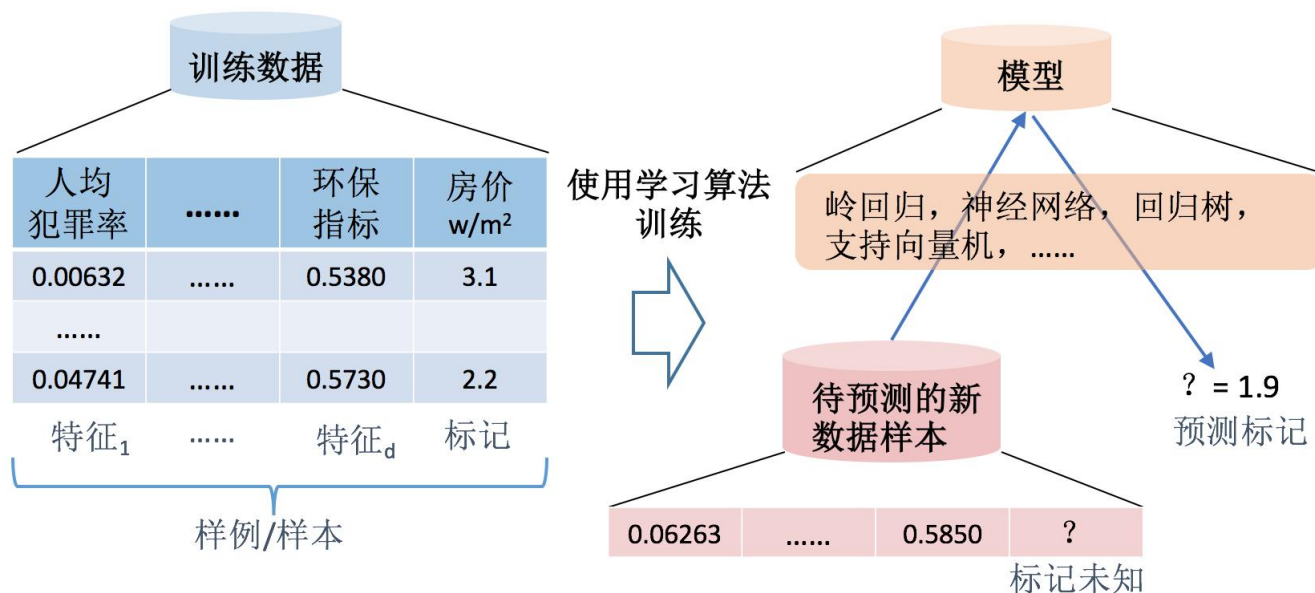


## 建立回归模型：

$$y = f(x_1, x_2, \dots, x_d) + \varepsilon.$$

$\varepsilon$  为随机误差项，表示由于人们的认识以及其它客观原因的局限而没有考虑的各种偶然因素。

某地来年各区平均房价预测任务





建立回归（regression）模型：

$$y = f(x_1, x_2, \dots, x_d) + \varepsilon.$$

$\varepsilon$  为随机误差项，表示由于人们的认识以及其它客观原因的局限而没有考虑的各种偶然因素。

若考虑映射  $f$  为线性函数，**线性回归（linear regression）** 模型：

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d + \varepsilon,$$

$$\text{即 } \mathbb{E}[y|\mathbf{x}] = f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d.$$

线性模型试图学得一个通过特征的线性组合来进行预测的线性函数  $f$

一个例子  $f_{\text{好瓜}}(\mathbf{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$

- 综合考虑色泽、根蒂和敲声来判断西瓜好不好
- 其中根蒂的系数最大，表明根蒂对判别好坏最重要；而敲声的系数比色泽大，说明敲声比色泽更重要





## 如何训练线性回归模型？

### 最小二乘 (least square) 法

学习参数  $\mathbf{w} = (w_0; w_1; w_2; \dots; w_d)$   
使得其在训练集上的均方误差最小

学习问题  $\rightarrow$  优化问题

训练集  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$

$\mathbf{x} = (x_1; x_2; \dots; x_d)$

$$\hat{\mathbf{w}} = (\hat{w}_0; \hat{w}_1; \hat{w}_2; \dots; \hat{w}_d) = \underset{w_0, w_1, w_2, \dots, w_d}{\operatorname{argmin}} \mathcal{J}(w_0, w_1, w_2, \dots, w_d).$$

$$\mathcal{J}(w_0, w_1, w_2, \dots, w_d) = \sum_{i=1}^n (y_i - w_0 - w_1 x_{i1} - w_2 x_{i2} - \dots - w_d x_{id})^2$$

离差的平方和



勒让德  
法国数学家



高斯  
德国数学家



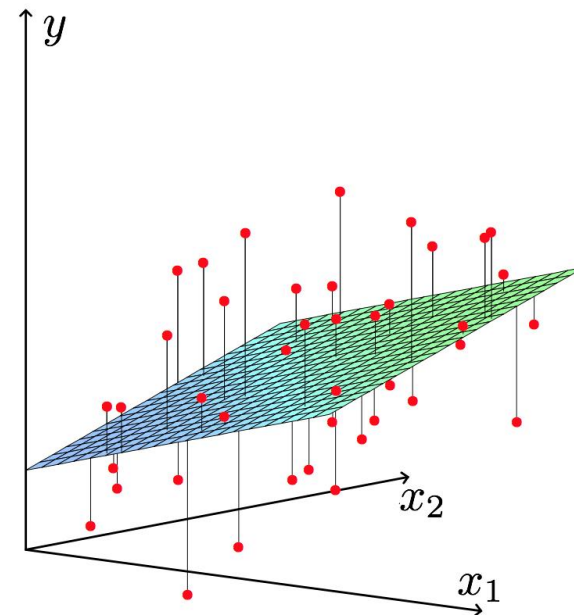


## 如何训练线性回归模型？

### 最小二乘（least square）法

学习参数  $\mathbf{w} = (w_0; w_1; w_2; \dots; w_d)$   
使得其在训练集上的均方误差最小

学习问题  $\rightarrow$  优化问题



$$\hat{\mathbf{w}} = (\hat{w}_0; \hat{w}_1; \hat{w}_2; \dots; \hat{w}_d) = \underset{w_0, w_1, w_2, \dots, w_d}{\operatorname{argmin}} \mathcal{J}(w_0, w_1, w_2, \dots, w_d).$$

$$\mathcal{J}(w_0, w_1, w_2, \dots, w_d) = \sum_{i=1}^n (y_i - w_0 - w_1 x_{i1} - w_2 x_{i2} - \dots - w_d x_{id})^2$$

离差的平方和

求解上述最优化问题，得到最小二乘估计作为学到的模型参数，用于后续的预测



求解优化问题：

$$\hat{\mathbf{w}} = (\hat{w}_0; \hat{w}_1; \hat{w}_2; \dots; \hat{w}_d) = \underset{w_0, w_1, w_2, \dots, w_d}{\operatorname{argmin}} \mathcal{J}(w_0, w_1, w_2, \dots, w_d).$$

$$\mathcal{J}(w_0, w_1, w_2, \dots, w_d) = \sum_{i=1}^n (y_i - w_0 - w_1 x_{i1} - w_2 x_{i2} - \dots - w_d x_{id})^2.$$

无约束优化，非负二次函数，最小值存在；根据微积分和凸优化中的求极值原理，最优解满足方程组：

$$\left\{ \begin{array}{l} \frac{\partial \mathcal{J}}{\partial w_0} \Big|_{w_0=\hat{w}_0} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \dots - \hat{w}_d x_{id}) = 0; \\ \frac{\partial \mathcal{J}}{\partial w_1} \Big|_{w_1=\hat{w}_1} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \dots - \hat{w}_d x_{id}) x_{i1} = 0; \\ \frac{\partial \mathcal{J}}{\partial w_2} \Big|_{w_2=\hat{w}_2} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \dots - \hat{w}_d x_{id}) x_{i2} = 0; \\ \dots \\ \frac{\partial \mathcal{J}}{\partial w_d} \Big|_{w_d=\hat{w}_d} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \dots - \hat{w}_d x_{id}) x_{id} = 0. \end{array} \right.$$

我们只须求解该方程组便可得到  $\hat{\mathbf{w}}$ 。



## 线性回归模型的矩阵表达：

$$\mathbb{E}[y|\mathbf{x}] = f(\mathbf{x}) = \mathbf{w}^\top \tilde{\mathbf{x}}$$

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_d \end{pmatrix}_{(d+1) \times 1}, \quad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \dots \\ w_d \end{pmatrix}_{(d+1) \times 1}$$

使用矩阵表示线性回归模型和对模型进行求解，矩阵表示方便简洁且易于成批操作和运算

要优化的目标函数： $\mathcal{J}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w})$

$$\mathbf{X} = \left( 1, \mathbf{x}_1^\top; 1, \mathbf{x}_2^\top; \dots; 1, \mathbf{x}_n^\top \right) = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1d} \\ 1 & x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix}_{n \times (d+1)}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}_{n \times 1}$$



$$\left\{ \begin{array}{l} \frac{\partial \mathcal{J}}{\partial w_0} \Big|_{w_0=\hat{w}_0} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \cdots - \hat{w}_d x_{id}) = 0; \\ \frac{\partial \mathcal{J}}{\partial w_1} \Big|_{w_1=\hat{w}_1} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \cdots - \hat{w}_d x_{id}) x_{i1} = 0; \\ \frac{\partial \mathcal{J}}{\partial w_2} \Big|_{w_2=\hat{w}_2} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \cdots - \hat{w}_d x_{id}) x_{i2} = 0; \\ \dots \\ \frac{\partial \mathcal{J}}{\partial w_d} \Big|_{w_d=\hat{w}_d} = -2 \sum_{i=1}^n (y_i - \hat{w}_0 - \hat{w}_1 x_{i1} - \hat{w}_2 x_{i2} - \cdots - \hat{w}_d x_{id}) x_{id} = 0. \end{array} \right.$$



$$\mathbf{X}^\top (\mathbf{y} - \mathbf{X} \hat{\mathbf{w}}) = \mathbf{0}$$



$$\mathbf{X}^\top \mathbf{X} \hat{\mathbf{w}} = \mathbf{X}^\top \mathbf{y}$$

$$\mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}$$



$$\mathbf{X}^\top \mathbf{X} \hat{\mathbf{w}} = \mathbf{X}^\top \mathbf{y}$$

若  $\mathbf{X}^\top \mathbf{X}$  可逆（逆矩阵：满足和原矩阵相乘为单位矩阵的矩阵）

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y},$$

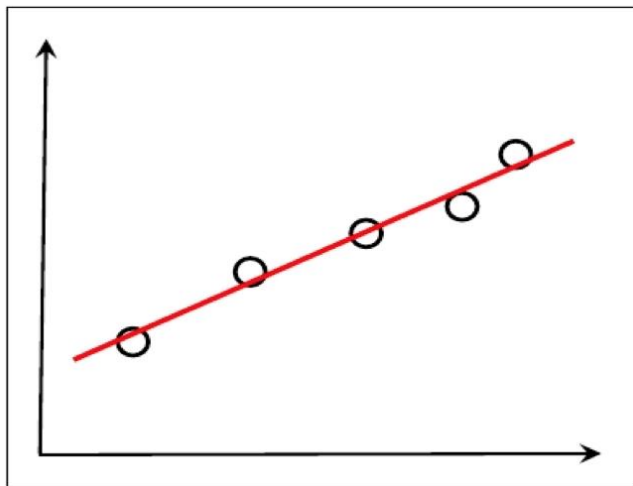
即是线性回归的最小二乘解。在学到  $f$  后，对于新的测试样本  $\mathbf{x}$  可得到其预测标记  $y = f(\mathbf{x}) = \hat{\mathbf{w}}^\top \tilde{\mathbf{x}}$

若  $\mathbf{X}^\top \mathbf{X}$  不可逆，可在优化目标中引入正则化（regularization）项。例如，岭回归，LASSO等

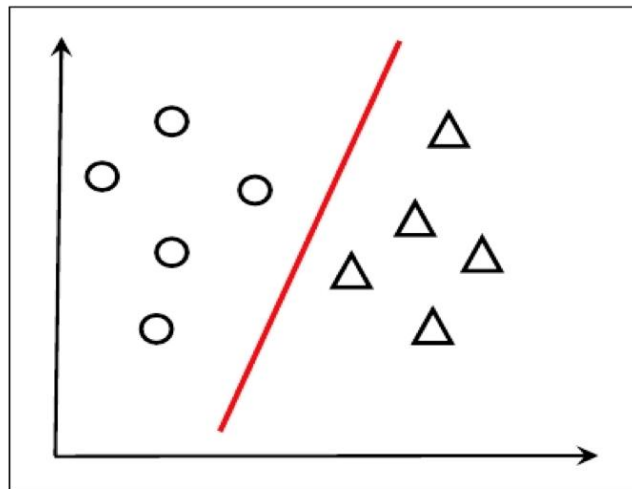


## 线性模型的优点和缺点：

简单、易于训练和测试、数学上易于优化、可解释性好、是复杂非线性模型的基础



线性回归



线性分类

线性模型的局限性：表示能力有限，难以直接拟合复杂的映射  
因此，需要引入更加复杂的非线性模型



## 8.5 本章小结



## 总结：

监督学习中的两大基本任务：分类与回归

### 分类：

- 决策树：选择划分属性      非线性模型

从二分类到多分类：拆解法，划归为二分类（一对一，一对多）

### 回归：

- 线性回归：最小二乘法      线性模型





# 谢谢



# 决策树：C4.5

## 选讲内容：C4.5 《机器学习》第4.2节

### 基于信息增益选择划分属性的缺点：

偏向于选择可取值数目较多的属性（比如会将样本编号作为一个划分属性来选择，这明显不合理）

### 增益率（Gain Ratio）：C4.5算法中使用

$$GR(D, a) = \frac{G(D, a)}{IV(a)}$$

$$\text{其中, } IV(a) = - \sum_{v=1}^V \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$$

增益率：偏向于选择可取值数目较少的属性

一般而言，属性a的可能取值数目越多，即V 越大，则  $IV(a)$  的值就越大

C4.5算法：不是直接选择增益率最大的划分属性，而是先从候选划分属性中找出信息增益高于平均水平的属性，再从中选取增益率最高的



# 决策树：剪枝

**选讲内容：为什么要对决策树剪枝？《机器学习》第4.3节**

剪枝方法和程度对决策树泛化性能的影响显著

在数据带噪时甚至可能将泛化性能提升25%

## Why?

剪枝 (pruning) 是决策树对付“过拟合”的主要手段！

为了尽可能正确分类训练样本，有可能造成分支过多 → 过拟合  
可通过主动去掉一些分支来降低过拟合的风险



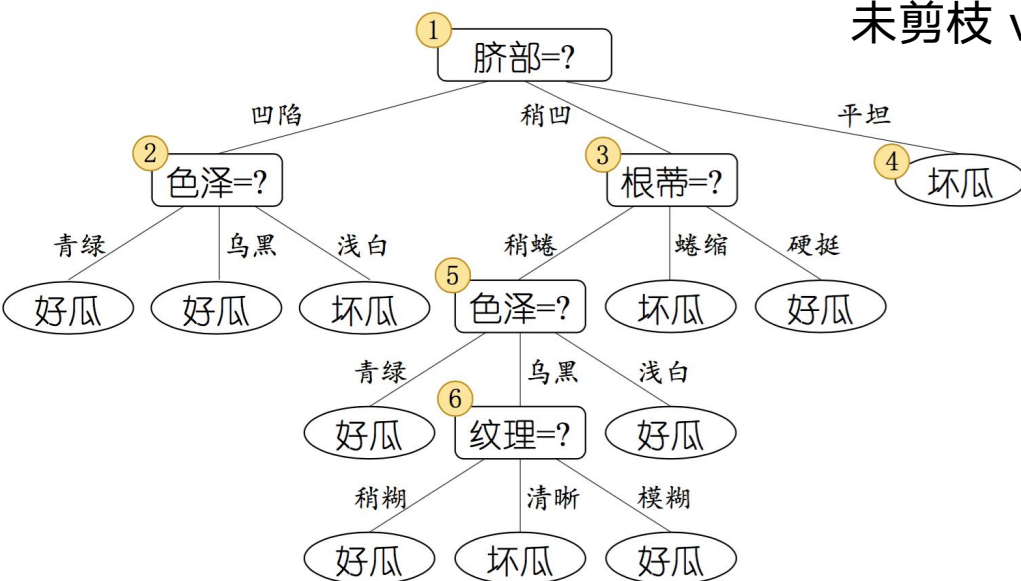
# 决策树：剪枝

## 选讲内容：剪枝的基本策略《机器学习》第4.3节

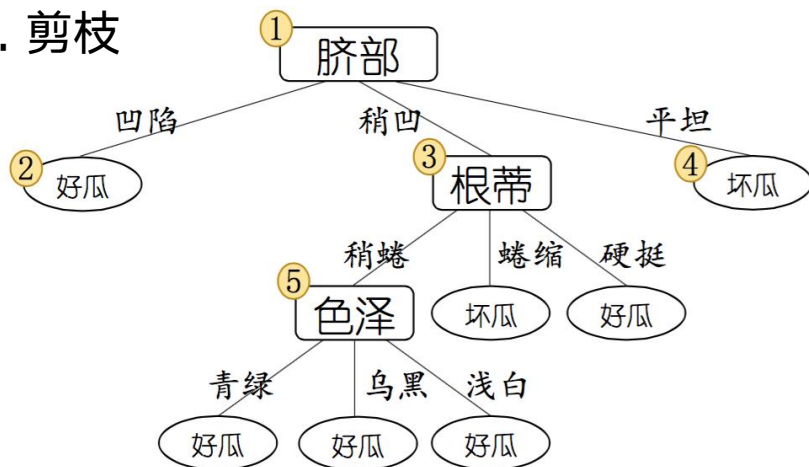
- 预剪枝 (pre-pruning): 提前终止某些分支的生长
- 后剪枝 (post-pruning): 生成一棵完全树, 再“回头”剪枝

泛化性能：后剪枝通常优于预剪枝

未剪枝 vs. 剪枝



未剪枝的决策树



已剪枝的决策树



# 决策树：连续值

选讲内容：当特征/属性为连续值时 《机器学习》第4.4节

基本思路：连续属性离散化

常见做法：二分法 (bi-partition)

连续属性 $a$ 的取值按从小到大排序，记为  $\{a^1, a^2, \dots, a^n\}$

划分点 $t$ 可将数据集 $D$ 分为：子集 $D^+$  ( $a > t$ 的样本) 和子集 $D^-$  ( $a \leq t$ 的样本)

划分点：
$$t_a = \left\{ \frac{a^i + a^{i+1}}{2} \mid 1 \leq i \leq n - 1 \right\}$$

- $n$  个属性值可形成  $n-1$  个候选划分
- 然后即可将它们当做  $n-1$  个离散属性值处理

