

# 人工智能 ——样例学习 I



饶洋辉

数据科学与计算机学院,

中山大学

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

# 分类

- 预测离散型变量
  - 首先基于一个包含 $x$ 值，以及离散型的真实 $y$ 值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。
- 典型应用
  - 精准营销 (电子商务)
  - 信用审批 (银行/金融)
  - 医学诊断 (健康医疗)
  - 欺诈/入侵检测 (互联网)

# 评测指标

- 准确率
- 速度
  - 构建分类模型的时间（训练速度）
  - 使用分类模型的时间（预测速度）
- 鲁棒性
  - 模型在处理噪音和缺失值方面的能力
- 可扩展性
  - 模型用在更大规模的数据集上的能力
- 可解释性

# 评测指标





# 评测指标



# k-近邻分类

- 随着人们生活水平不断的提高,红酒越来越受到人们的喜爱。红酒的产量越来越大,然而红酒品质鉴定的手段还是仅靠品酒师的人工品尝打分来判定红酒质量的好坏,显然这种鉴定方式难以满足当今市场的需求。现在有不少学者运用一些机器学习的算法来对红酒质量进行预测研究,使得红酒品质鉴定的速度得到大幅提升并且有着较高的准确率。

# k-近邻分类

- 对于红酒品质的分类，可以基于红酒的理化指标(例如：酒精的浓度、pH值、糖的含量、非挥发性酸含量、挥发性酸含量、柠檬酸含量等)作为特征，建立分类模型，然后对红酒品质进行预测。本案例中，我们将使用UCI数据库中的 Wine Quality Data Set 数据集，利用k-近邻分类算法来进行红酒品质的分类。





# k-近邻分类

- 我们使用一份包含1599个样本的关于葡萄牙的Vinho Verde葡萄酒数据集。 每个样本包含12个变量，其中最后一个变量quality为预测变量。

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

- k-近邻分类算法：所有属性（变量）同等重要。
- k指的是选取的和待预测样本距离最近的训练样本数。



# k-近邻分类

- 0、1、2、3为训练集
- 4为测试集
- $k = 1, 2, 3, 4$
- 采用街区距离
- $d(4, 0) = 0.0$
- $d(4, 1) = 49.1$
- $d(4, 2) = 25.7$
- $d(4, 3) = 37.6$

变量名称	含义说明
fixed acidity	非挥发性酸含量
volatile acidity	挥发性酸含量
citric acid	柠檬酸
residual sugar	糖含量
chlorides	氯化物
free sulfur dioxide	游离二氧化硫
total sulfur dioxide	总二氧化硫
density	密度
pH	酸碱度
sulphates	硫酸盐
alcohol	酒精浓度
quality	品质， 为预测变量

# 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

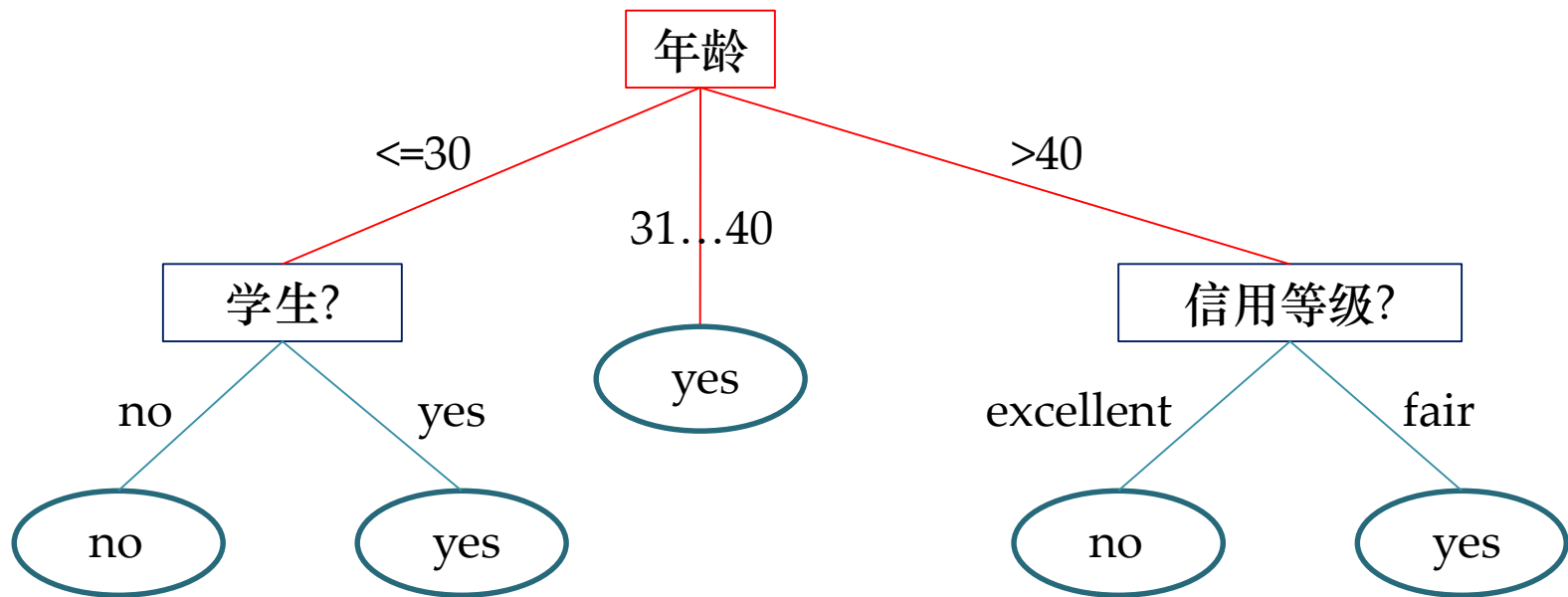
年龄	收入	学生?	信用等级?	是否买电脑
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	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（例如，信息增益，基尼系数等）。
  - (训练) 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

# 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 (用来衡量离散变量的不确定性，如抛硬币、掷骰子)

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

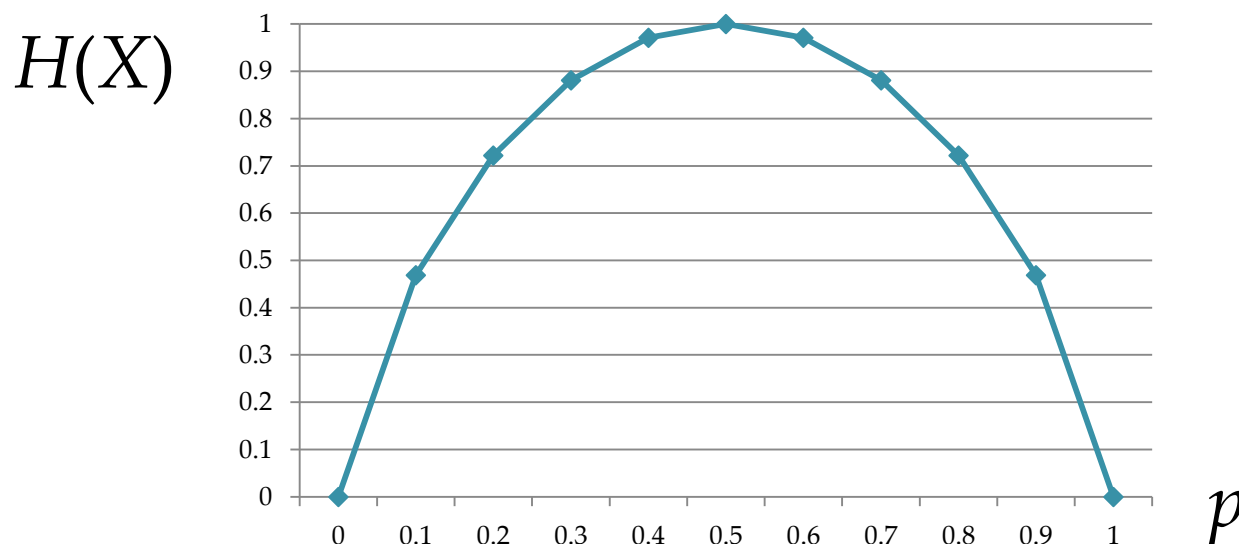
$$= \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

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

# 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

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

- $H(D | A = \text{"年龄"}) = 0.694$

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

# 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**属性的基尼指数，信息增益？



# 连续型属性的处理

- 应该如何计算具有**连续值**属性的基尼指数，信息增益？
  - 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

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

# 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)。

# 过拟合

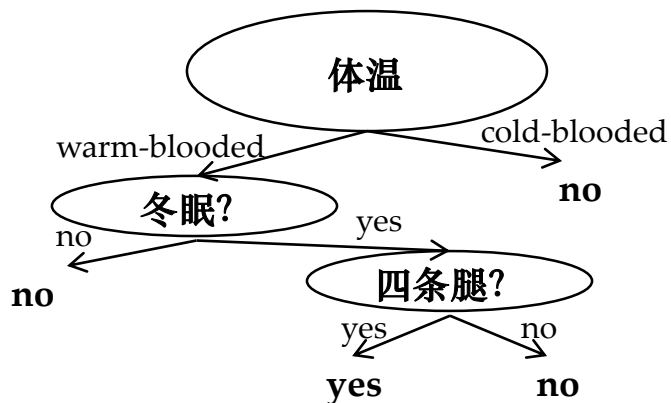
- 过拟合问题

- 一颗决策树可能对训练样本过拟合，导致在测试样本上没有良好的泛化表现。（即，在训练集上有良好表现但在新样本上准确率很低）

# 过拟合

- 训练样本错误为0, 将所有不冬眠的温血动物分类为非哺乳动物。

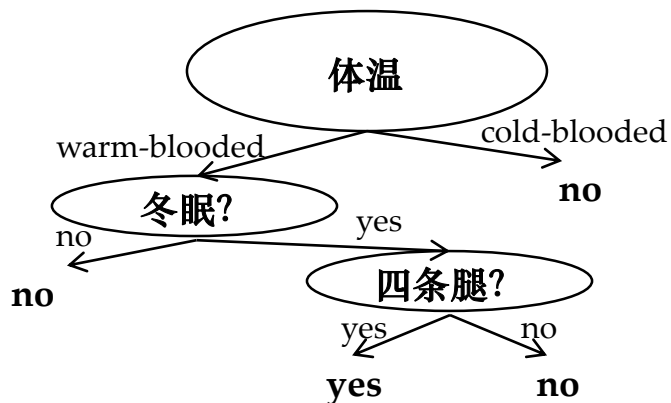
名称	体温	四条腿?	冬眠?	哺乳动物?
salamander	cold-blooded	yes	yes	no
guppy	cold-blooded	no	no	no
eagle	warm-blooded	no	no	no
poorwill	warm-blooded	no	yes	no
platypus	warm-blooded	yes	yes	yes



# 过拟合

- 训练样本错误为0, 将所有不冬眠的温血动物分类为非哺乳动物。

名称	体温	四条腿?	冬眠?	哺乳动物?
salamander	cold-blooded	yes	yes	no
guppy	cold-blooded	no	no	no
eagle	warm-blooded	no	no	no
poorwill	warm-blooded	no	yes	no
platypus	warm-blooded	yes	yes	yes



Humans, elephants and dolphins都分错了

# 过拟合

- 基于少量训练样本做出分类决策的模型都很容易过拟合。
  - 解决办法：加入更多样本训练模型

名称	体温	四条腿?	冬眠?	哺乳动物?
salamander	cold-blooded	yes	yes	no
guppy	cold-blooded	no	no	no
eagle	warm-blooded	no	no	no
poorwill	warm-blooded	no	yes	no
platypus	warm-blooded	yes	yes	yes
human	warm-blooded	no	no	yes
dolphin	warm-blooded	no	no	yes
elephant	warm-blooded	yes	no	yes

# 过拟合

名称	体温	胎生?	四条腿?	冬眠?	哺乳动物?
porcupine	warm-blooded	yes	yes	yes	yes
cat	warm-blooded	yes	yes	no	yes
bat	warm-blooded	yes	no	yes	no
whale	warm-blooded	yes	no	no	no
salamander	cold-blooded	no	yes	yes	no
komodo dragon	cold-blooded	no	yes	no	no
python	cold-blooded	no	no	yes	no
salmon	cold-blooded	no	no	no	no
eagle	warm-blooded	no	no	no	no
guppy	cold-blooded	yes	no	no	no

Training set

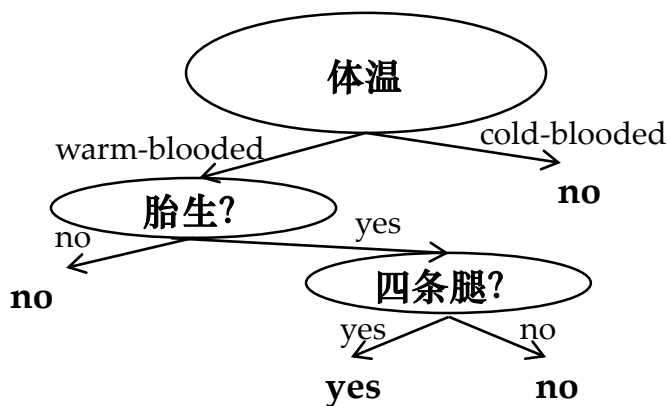


# 过拟合

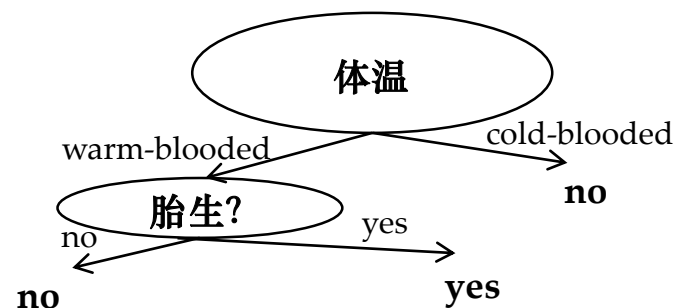
名称	体温	胎生?	四条腿?	冬眠?	哺乳动物?
human	warm-blooded	yes	no	no	yes
pigeon	warm-blooded	no	no	no	no
elephant	warm-blooded	yes	yes	no	yes
leopard shark	cold-blooded	yes	no	no	no
turtle	cold-blooded	no	yes	no	no
penguin	warm-blooded	no	no	no	no
eel	cold-blooded	no	no	no	no
dolphin	warm-blooded	yes	no	no	yes
spiny anteater	warm-blooded	no	yes	yes	yes
gila monster	cold-blooded	no	yes	yes	no

Testing set

# 过拟合



训练集上错误个数 0  
测试集上错误率为 30%



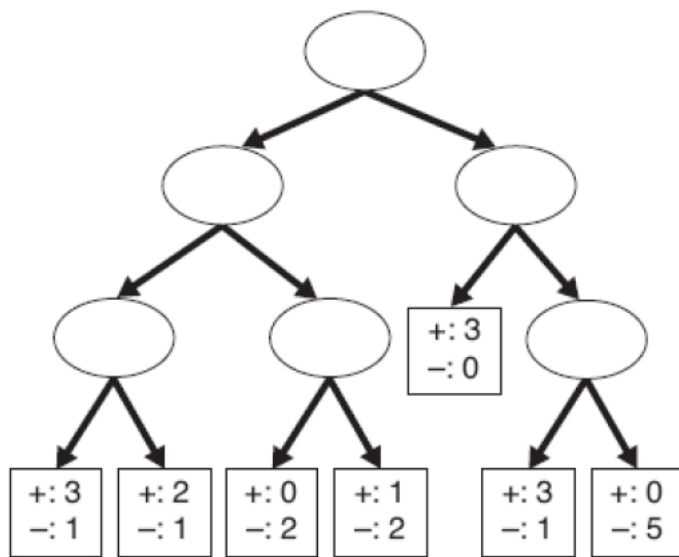
训练集上错误率 20%  
测试集上错误率 10%

# 过拟合

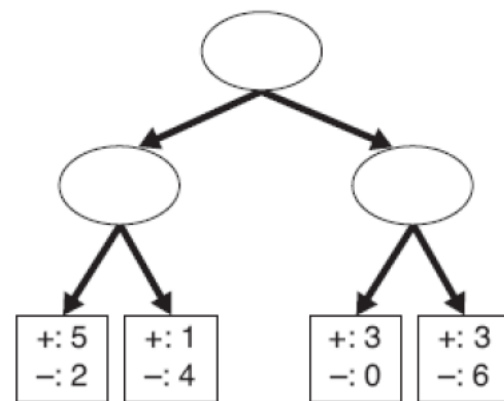
- 训练集的误差可以通过增加模型复杂度减少
  - 当树变得很大，训练集错误率可以持续降低
- 但是，测试集错误率（泛化错误率）会很大，因为模型会拟合一些训练样本的噪声点
  - 太多分支，某些分支会反映出因为噪声和离群值而产生的异常。

# 融入模型复杂度

- 泛化错误率用以下两个值的和估计：
  - 训练集错误数
  - 模型复杂度的惩罚项
- $T_L$  的训练集错误数为  $e(T_L)=4/24=0.167$
- $T_R$  的训练集错误数为  $e(T_R)=6/24=0.25$



Decision Tree,  $T_L$



Decision Tree,  $T_R$

# 融入模型复杂度

- 在决策树中，定义以下变量
  - $L$ : 叶子节点个数
  - $n_l$ : 第  $l$  个叶子节点
  - $m(n_l)$ : 被  $n_l$  准确分类的训练样本数
  - $r(n_l)$ : 被  $n_l$  错误分类的训练样本数
  - $\zeta(n_l)$ :  $n_l$  的惩罚项
- 决策树的错误率  $e_c$  可以估计如下:

$$e_c = \frac{\sum_{l=1}^L (r(n_l) + \zeta(n_l))}{\sum_{l=1}^L m(n_l)}$$

# 融入模型复杂度

- 考虑之前的两棵决策树  $T_L$  和  $T_R$ .
- 假设对每个叶子节点的惩罚项为 0.5

- $T_L$  的错误率估计为

$$e_c(T_L) = \frac{4 + 7 \times 0.5}{24} = \frac{7.5}{24} = 0.3125$$

- $T_R$  的错误率估计为

$$e_c(T_R) = \frac{6 + 4 \times 0.5}{24} = \frac{8}{24} = 0.3333$$

# 融入模型复杂度

- 基于此惩罚项, 可认为 $T_L$  优于  $T_R$ .
- 对一个二叉树而言, 惩罚项为 0.5 意味着当一个节点分裂为两个节点后能够提升至少一个分类准确数, 那就必须分裂。
- 因为增加一个节点, 相当于在总体错误加上 0.5, 比把一个样本错误分类的代价小。



# 融入模型复杂度

- 假设对于所有叶子节点的惩罚项都为1
- $T_L$  的错误估计为 0.458.
- $T_R$  的错误估计为 0.417.
- 基于这个惩罚项,  $T_R$  要优于  $T_L$ .
- 惩罚项为 1 意味着当分裂一个节点能够使多于一个的训练样本被正确分类时, 该节点才应该被分裂。

# 决策树剪枝

- 避免过拟合的两个方法
  - 预剪枝：提前停止建树-如果分裂节点会导致树的分类准确率低于阈值
    - 难点在于选择一个合适的阈值。
  - 后剪枝：从一个“建满了”的树上剪枝-获得一系列进一步剪枝了的树
    - 用不同于训练数据的数据集去决定最好的剪枝树是哪个。

# 决策树的选取方法

- 分出训练集和测试集
- 使用交叉认证，比如，k折交叉认证
  - 把数据集分成k部分
  - 随机选取k-1部分用于训练，在剩下那部分上测试
  - 重复k次

# 总结

- 每次处理一个属性
- 连续随机值应该分成离散随机值。