模式识别

主讲：崔林艳&邹征夏

单位：宇航学院

开课专业：飞行器控制与信息工程

第四章 非线性分类器

CONTENTS PAGE

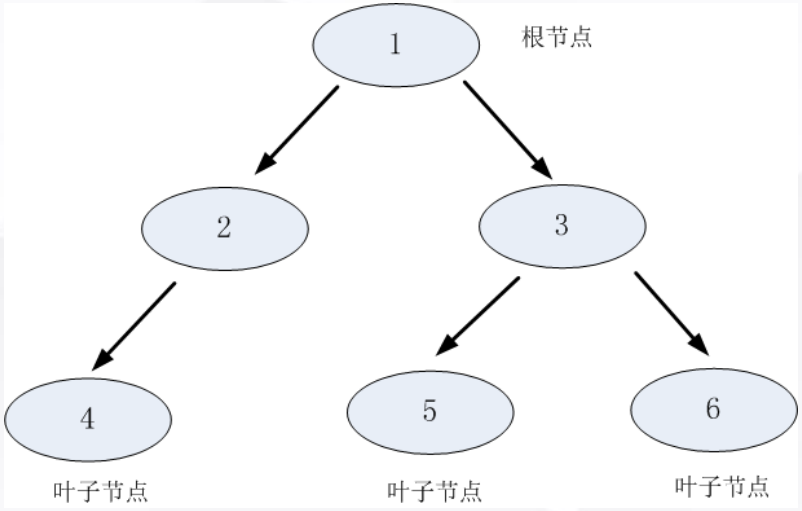
4.4 决策树

4.4 决策树

4.4.1 决策树概念

➢ 树的概念

- 树是由节点和边两种元素组成的结构，包括根节点、父节点、子节点和叶子节点等。
- 父节点和子节点是相对的，子节点由父节点根据某一规则分裂而来，然后子节点作为新的父亲节点继续分裂，直至不能分裂为止。
- 根节点：没有父节点的节点，即初始分裂节点。
- 叶子节点：没有子节点的节点。



4.4 决策树

4.4.1 决策树概念

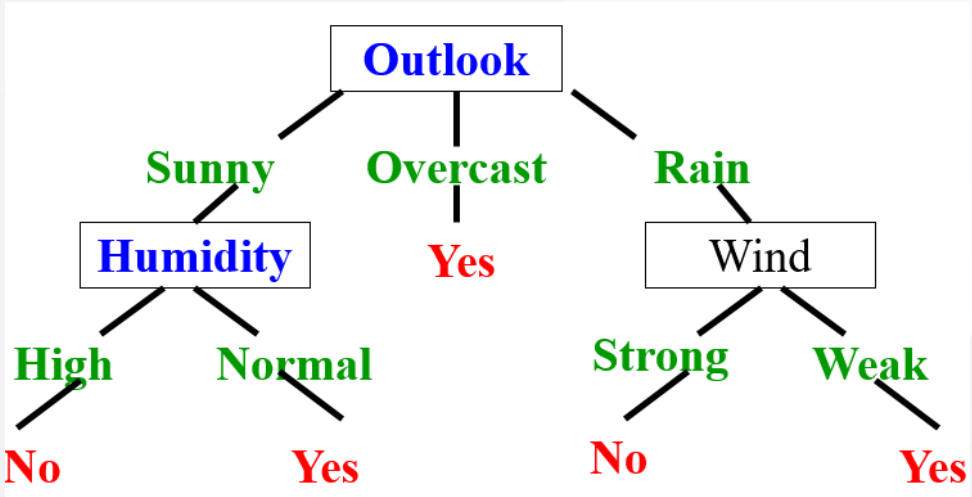
➢ 决策树的概念

- 决策树分类方法：利用一定的训练样本，从数据中“学习”出决策规则，自动构造决策树，并利用决策树实现分类。
- 决策树利用树结构进行决策，每一个非叶子节点是一个判断条件，每一个叶子节点是结论。
- 从根节点开始，经过多次判断得出结论。

4.4 决策树

4.4.1 决策树概念

下图绘出了一棵学习到的典型决策树。这棵决策树根据天气情况分类“星期六上午是否适合打网球”。



可以将学习到的决策树表示为多个IF－THEN 规则。左图的决策树可以表示成以下的规则集：

Rule 1： IF Outlook=Sunny AND Humidity=High THEN No

Rule 2： IF Outlook=Sunny AND Humidity=Normal THEN Yes

Rule 3： IF Outlook=Overcast THEN Yes

Rule 4： IF Outlook=Rain AND Wind=Strong THEN No

Rule 5： IF Outlook=Rain AND Wind=Strong THEN Yes

4.4 决策树

4.4.2 二叉树

最简单的决策树模型

- 每个非终止节点上采用线性分类器进行决策
- 每个非终止节点有两个分支，故称为二叉树
- 二叉树将多峰问题或多类问题转换为多级的线性问题

4.4 决策树

4.4.2 二叉树

- 二叉树算法步骤

① 采用线性判别函数g，决策面正负侧判别

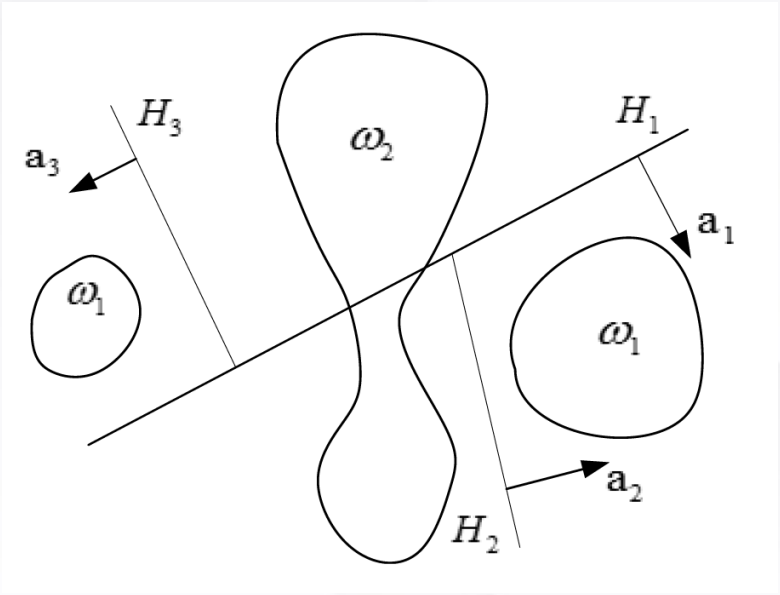
② 设计线性分类器，将特征空间一分为二，成为两个子空间

③ 对每个子空间，继续进行一分为二的操作，直至每个子空间中只含一类样本

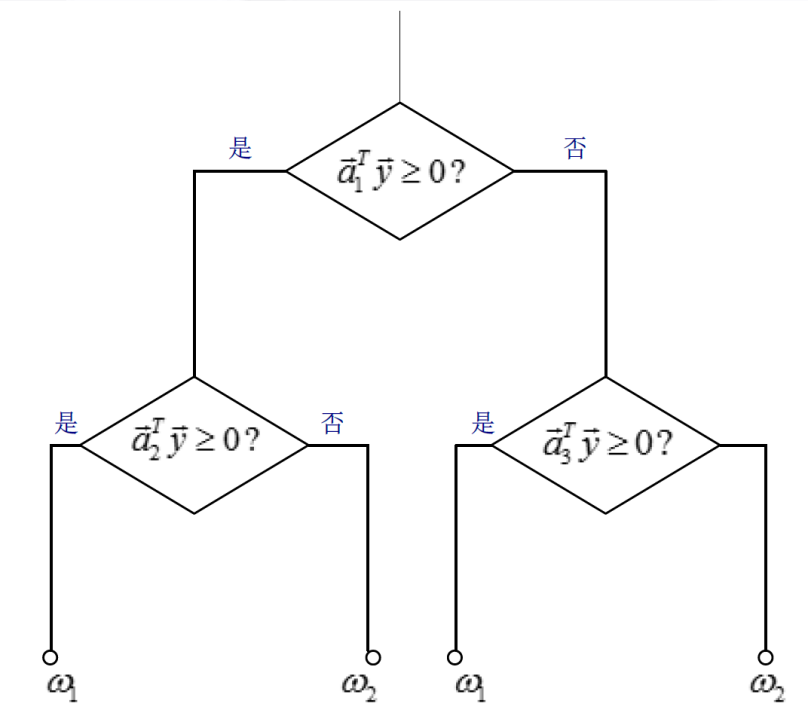
4.4 决策树

4.4.2 二叉树

- 二叉树实例（连续特征空间）



(a) 二叉树在特征空间中的表示



(b) 二叉树

4.4.2 二叉树

- 二叉树算法特点及问题
  - 分段线性分类器
  - 二叉树各节点（分段）可以优化设计
  - 解决两类多峰问题
  - 分类器设计相对复杂
  - 第一个权向量影响后续所有权向量

4.4 决策树

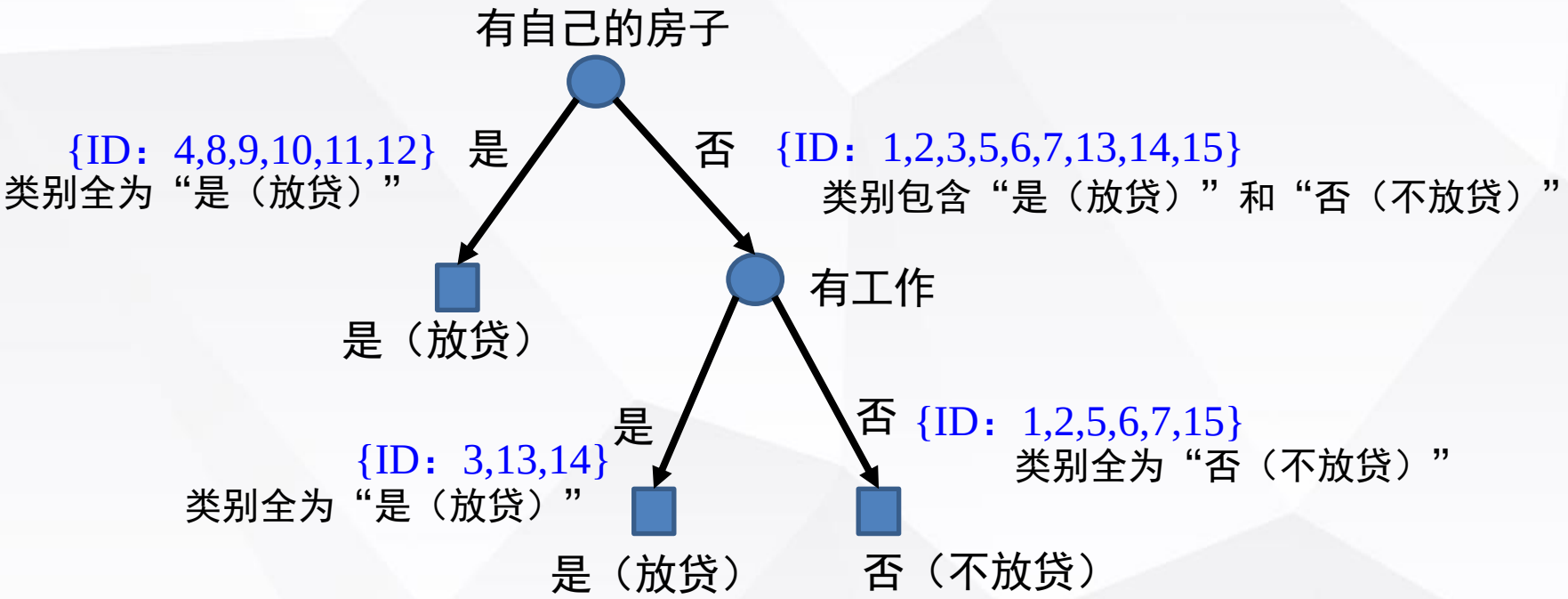
实例：右表为由若干客户（样本）组成的贷款申请训练数据，每一个客户都有四个属性（特征）：年龄、是否工作、是否有自己的房子和信用情况。现通过所给的训练数据学习一个贷款申请的决策树，当新的客户提出贷款申请时，根据申请人的属性利用决策树决定是否批准贷款申请。

尝试构建二叉树

ID	年龄	是否工作	是否有自己的房子	信用情况	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

4.4 决策树

- 假设选择属性顺序分别为：是否有自己的房子、是否有工作。。。



4.4 决策树

- 假设选择属性顺序分别为：年龄、是否工作、。。。

自行计算！

选择属性顺序不同，获得不同的二叉树！

4.4.3 决策树构建方法

- 决策树构建一般步骤

步骤1：属性（特征）选择

- 从众多的属性中选择一个属性作为当前节点分裂的标准

步骤2：决策树生成

- 根据选择的特征评估标准，从上至下递归地生成子节点，直到数据集不可分则停止决策树生长

步骤3：剪枝

- 缩小树结构规模、缓解过拟合

4.4 决策树

实例：右表为由若干客户（样本）组成的贷款申请训练数据，每一个客户都有四个属性（特征）：年龄、是否工作、是否有自己的房子和信用情况。现通过所给的训练数据学习一个贷款申请的决策树，当新的客户提出贷款申请时，根据申请人的属性利用决策树决定是否批准贷款申请。

尝试构建一棵决策树

ID	年龄	是否工作	是否有自己的房子	信用情况	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

4.4 决策树

➢ 实例分析：

上述贷款数据表中存在4种属性，最少可以得到4个可能的决策树，分别由4个不同属性的根节点构成，比如

选择不同的属性可以产生不同的决策树

问题引入：如何确定属性选择的准则以更好地分类？

4.4.3 决策树构建方法

➢ 属性选择定性准则

决策树学习关键在于如何选择最优划分属性（特征）。

- 一般而言，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。

- 属性选择的准则：使用某属性对数据集划分之后，各数据子集的纯度要比划分前的数据集的纯度高，也就是不确定性要比划分前数据集的不确定性低。

4.4.3 决策树构建方法

➢ 属性选择定量准则（信息熵）

- 熵：在信息论中，熵（entropy）是随机变量不确定性的度量，熵越大，随机变量的不确定性越大。
- “信息熵”是度量样本集合纯度最常用的一种指标。

设 X 是一个取有限个值 $\{x_1,x_2,\ldots,x_n\}$ 的离散随机变量，其概率分布为：$P(X=x_i)=p_i, i=1,2,\cdots,n$

则随机变量 X 的熵定义为：

$$H(X)=-\sum_{i=1}^{n} p_i \log_2 p_i$$

上式中的对数一般以2为底，并规定若 $p_i=0$ 时，定义 $0\log_2 0=0$ 。

- 经验熵：当熵中的概率 p_i 由数据估计（特别是最大似然估计）得到时，所对应的熵称为经验熵(empirical entropy)。

4.4.3 决策树构建方法

➢ 经验熵计算

定义贷款申请数据表中的数据为训练数据集D，则训练数据集D的经验熵为$H(D)$，$|D|$表示样本个数。设有K个类$C_k, k=1,\cdots,K$，$|C_k|$为属于类C_k的样本个数，此时经验熵公式可以写为

$$H(D)=-\sum\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|}$$

熵中的概率p_i由数据估计得到

示例：放贷分类结果只有两类(K=2)，即放贷和不放贷。根据表中的数据统计可知，在15个数据中，9个数据的结果为放贷，6个数据的结果为不放贷。所以数据集D的经验熵$H(D)$为：

$$H(D)=-\left(\frac{9}{15}\log_2\frac{9}{15}+\frac{6}{15}\log_2\frac{6}{15}\right)=0.971$$

经过计算可知，数据集的经验熵的值为0.971。

4.4.3 决策树构建方法

➢ 条件熵计算

X（随机变量）给定的条件下Y（随机变量）的条件熵 $H(Y|X)$ 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：

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

上述条件熵物理含义：在已知随机变量X的条件下随机变量Y的不确定性

其中，$p_i = P(X=x_i)$，当条件熵中的概率由数据估计（特别是极大似然估计）得到时，为经验条件熵。

4.4.3 决策树构建方法

① ID3（Interactive Dichotomizer-3）

② C4.5

③ CART（classification and regression tree）

4.4 决策树

① ID3（Interactive Dichotomizer-3）

➢ 信息增益

- 在熵和条件熵的基础上，定义信息增益。
- 属性A对训练数据集D的信息增益$Gain(D,A)$定义为：

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

$H(D)$：集合D的经验熵

$H(D|A)$：属性A给定条件下D的经验条件熵

- 信息增益物理含义：表示已知属性A的信息的条件下，使得数据集D的信息不确定性减少的程度。

4.4 决策树

① ID3（Interactive Dichotomizer-3）

➢ ID3算法基本思想

ID3算法在决策树各个节点上利用信息增益准则选择属性，递归地构建决策树（自顶向下构造决策树）。

① 从根节点开始，对该节点计算所有可能的属性的信息增益，选择信息增益最大的属性作为节点的属性。

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

此时属性A对应的条件熵$H(D|A)$最小。即通过该属性对数据集D的划分，使得决策属性的不确定性最低。

② 由该属性的不同取值建立子节点，再对子节点递归地调用以上方法；直到所有属性的信息增益均很小或没有属性可以选择为止，最后得到一个决策树。

4.4 决策树

ID3算法

输入：训练数据集D，属性集A，阈值ε ；

输出：决策树T。

Step1： 若D中所有样本属于同一类，则T为单节点树，并将类作为该节点的类标记，输出T；

Step2： 若$A=\emptyset$，则T为单节点树，并将D中样本数最大的类作为该节点的类标记，输出T；

Step3： 若$A\neq\emptyset$则计算A中各属性对D的信息增益，选择信息增益最大的属性A_{ig}；

Step4： 如果A_{ig}信息增益小于阈值ε，则T为单节点树，并将D中样本数最大的类作为该节点的类标记，输出T；

Step5： 否则，对A_{ig}的每一种可能值a_j，按照$A_{ig}=a_j$ 将D分割为若干非空子集D_j，将D_j中样本数最大的类作为标记，构建子节点，由节点及其子树构成树T，输出T；

Step6： 对第j个子节点，以D_j为训练集，以A-$\{A_{ig}\}$为特征集合，递归调用Step1~step5，得到子树T_j，输出T；

4.4 决策树

实例：右表为由若干客户（样本）组成的贷款申请训练数据，每一个客户都有四个属性（特征）：年龄、是否工作、是否有自己的房子和信用情况。现通过所给的训练数据学习一个贷款申请的决策树，当新的客户提出贷款申请时，根据申请人的属性利用决策树决定是否批准贷款申请。

尝试用ID3算法构建一棵决策树

ID	年龄	是否工作	是否有自己的房子	信用情况	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

4.4 决策树

步骤1：从根节点开始，对节点计算所有可能的属性的信息增益，选择信息增益最大的特征作为节点的属性。

① 分别计算各个属性（特征）对数据集D的信息增益。分别记年龄、是否工作、是否有自己的房子和信用情况分别为A_1，A_2，A_3，A_4。

以A_3为例计算信息增益$Gain(D,A_3)$：

$$H(D)=-\sum\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|}$$

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

$$\begin{aligned}Gain(D,A_3)&=H(D)-H(D|A_3)\\&=0.971-\left[\frac{6}{15}H(D_1)+\frac{9}{15}H(D_2)\right]\\&=0.971-\left[\frac{6}{15}\times0+\frac{9}{15}\times\left(-\frac{3}{9}\log_2\frac{3}{9}-\frac{6}{9}\log_2\frac{6}{9}\right)\right]\\&=0.971-0.551=0.420\end{aligned}$$

其中D_1，D_2分别表示数据集中属性A_3取值分别为有自己的房子和没有自己的房子的样本子集。D_1 中的6个数据的分类结果均为放贷，所以$H(D_1)=0$；D_2 中的3个数据的分类结果为放贷，6个数据分类结果为不放贷.

4.4 决策树

步骤1：从根节点开始，对节点计算所有可能的属性的信息增益，选择信息增益最大的特征作为节点的属性。

① 分别计算各个属性（特征）对数据集D的信息增益。分别记年龄、是否工作、是否有自己的房子和信用情况分别为A_1，A_2，A_3，A_4。

以A_4为例计算信息增益$Gain(D,A_4)$：

$$H(D)=-\sum\frac{|C_k|}{|D|}\log\frac{|C_k|}{|D|}$$

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

$$Gain(D,A_4)=H(D)-H(D\mid A_4)$$

$$=0.971-\left[\frac{4}{15}H(D_1)+\frac{6}{15}H(D_2)+\frac{5}{15}H(D_3)\right]$$

$$=0.971-\left[\frac{4}{15}\times 0+\frac{6}{15}\times\left(-\frac{4}{6}\log_2\frac{4}{6}-\frac{2}{6}\log_2\frac{2}{6}\right)+\frac{5}{15}\times\left(-\frac{1}{5}\log_2\frac{1}{5}-\frac{4}{5}\log_2\frac{4}{5}\right)\right]$$

$$=0.971-0.608=0.363$$

其中D_1，D_2，D_3分别表示数据集中属性A_4取值分别为非常好、好和一般的样本子集。

4.4 决策树

步骤1：从根节点开始，对节点计算所有可能的属性的信息增益，选择信息增益最大的特征作为节点的属性。

① 分别计算各个属性（特征）对数据集D的信息增益。分别记年龄、是否工作、是否有自己的房子和信用情况分别为A_1，A_2，A_3，A_4。

同理，计算出信息增益$Gain(D,A_1)$和$Gain(D,A_4)$：

$$\begin{aligned} Gain(D,A_1) &= H(D) - H(D|A_1) \\ &= 0.971 - 0.888 = 0.083 \end{aligned}$$

$$\begin{aligned} Gain(D,A_2) &= H(D) - H(D|A_2) \\ &= 0.971 - 0.647 = 0.324 \end{aligned}$$

4.4 决策树

步骤1：从根节点(root node)开始，对节点计算所有可能的属性的信息增益，选择信息增益最大的特征作为节点的属性。

② 选取信息增益最大的特征属性。

$$Gain(D, A_1) = H(D) - H(D \mid A_1) = 0.083$$

$$Gain(D, A_2) = H(D) - H(D \mid A_2) = 0.324$$

$$\boxed{Gain(D, A_3) = H(D) - H(D \mid A_3) = 0.420}$$

$$Gain(D, A_4) = H(D) - H(D \mid A_4) = 0.363$$

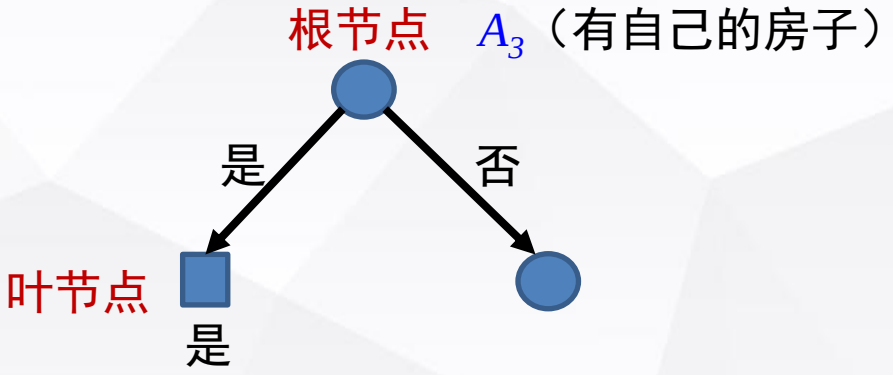
A_3属性（是否有自己的房子）的信息增益最大，将其作为决策树的根节点特征。

4.4 决策树

步骤1：从根节点(root node)开始，对节点计算所有可能的属性的信息增益，选择信息增益最大的特征作为节点的属性。

③ 利用信息增益最大的属性对数据集进行划分。

- 利用A_3属性（是否有自己的房子）取值将数据集划分为两个子集：样本子集D_1（有自己的房子）和样本子集D_2（没有自己的房子）。
- 样本子集D_1只有同一类的样本点（均为“是”），所以它成为一个叶节点，节点的类标记为“是”。
- 样本子集D_2有两类样本点，需要继续执行后续操作。



4.4 决策树

- 样本子集D_1（有自己的房子）

  ID：4,8,9,10,11,12

- 样本子集D_2（没有自己的房子）

  ID：1,2,3,5,6,7,13,14,15

- 样本子集D_2有两类样本点，需要继续执行后续操作。

ID	年龄	是否工作	是否有自己的房子	信用情况	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

步骤2：对子节点（样本子集D_2）递归地调用以上方法

① 分别计算剩余属性（A_1，A_2，A_4）对样本子集D_2的信息增益。

$$Gain(D_2, A_1) = H(D_2) - H(D_2 | A_1) = 0.251$$

$$Gain(D_2, A_2) = H(D_2) - H(D_2 | A_2) = 0.918$$

$$Gain(D_2, A_4) = H(D_2) - H(D_2 | A_4) = 0.474$$

由计算结果可知，A_2属性（是否工作）的信息增益最大，因此将其作为节点的特征。

步骤2：对子节点（样本子集D_2）递归地调用以上方法

② 利用信息增益最大的属性对数据集进行划分。

- 利用A_2属性（是否有工作）取值将数据集划分为两个子集：样本子集D_3（有工作）和样本子集D_4（无工作）。
- 样本子集D_3只有同一类的样本点（均为“是”），所以是一个叶节点，节点的类标记为“是”。
- 样本子集D_4也只有一类样本点（均为“否”），所以是一个叶节点，节点的类标记为“否”。

4.4 决策树

- 样本子集D_3（有工作）

  ID：3，13，14

- 样本子集D_4（无工作）

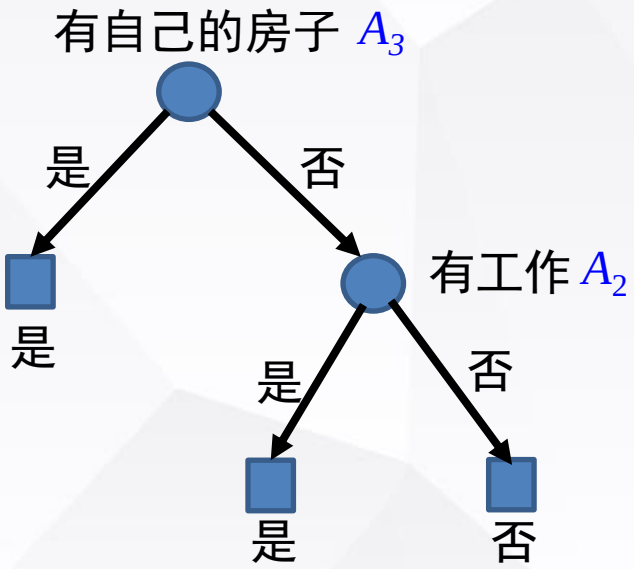
  ID：1，2，5，6，7，15

- 样本子集D_3和D_4也均只包含一类样本点。

ID	年龄	是否工作	是否有自己的房子	信用情况	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

4.4 决策树

此时，决策树构建完成，该树只用了两个属性，生成的决策树：



① ID3（Interactive Dichotomizer-3）

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

$$H(D)=-\sum\frac{|C_k|}{|D|}\log_2\frac{|C_k|}{|D|}$$

➢ 算法（信息增益准则）问题分析

a） 信息增益会偏向取值较多的属性

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

$$=H(D)-\left[p_1\cdot H(D_1)+p_2\cdot H(D_2)+...+p_n\cdot H(D_n)\right]$$

若属性A有n个取值，随着n的增多，其划分出的子集D_i越多，每个子集中包含的样本数越少（不确定性也减少），$H(D_i)$越小，则条件熵$H(D|A)$越小。此时，信息增益越大。

b） 信息增益值的大小是相对于训练数据集而言的，并没有绝对意义。在训练数据集经验熵$H(D)$大的时候，信息增益值会偏大，反之信息增益值会偏小。

② C4.5算法

➢ 引入信息增益比

属性A对训练数据集D的信息增益比

$$GainRatio(D,A)=\frac{Gain(D,A)}{H_A(D)}$$

$Gain(D,A)$：属性A对训练数据集D的信息增益

$H_A(D)$：以属性A对训练数据集D进行划分的经验熵

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

其中特征A有n种取值，$|D_i|$表示A的第i种取值对应的样本个数

C4.5算法在生成决策树过程中采用信息增益比来选择属性，是ID3算法的改进。

4.4 决策树

作业

右表为由若干客户（样本）组成的贷款申请训练数据，每一个客户都有四个属性（特征）：年龄、是否工作、是否有自己的房子和信用情况。现通过所给的训练数据学习一个贷款申请的决策树，当新的客户提出贷款申请时，根据申请人的属性利用决策树决定是否批准贷款申请。

用C4.5算法构建一棵决策树并写出计算过程。

ID	年龄	是否工作	是否有自己的房子	信用情况	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

③ CART（classification and regression tree）算法

CART算法使用基尼指数（Gini index）来选择最优属性，同时决定该属性的最优二值划分点。假设分类问题中，设有K个类，样本属于第k类的概率为 p_k，则概率分布的基尼指数定义为：

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

对于二分类问题，设属于第一类的概率为p，则$Gini(p)=2p(1-p)$。

对于给定的样本集合D，假设有K个类$C_k(k=1,2,\ldots,K)$，$|C_k|$为属于类C_k的样本个数，$|D|$表示总样本个数，其基尼指数为：

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

相比熵模型，二次运算比对数简单很多。

③ CART（classification and regression tree）算法

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

假设样本集合D根据属性A是否取某一可能值a被划分为D_1和D_2两部分，则在属性A的条件下，样本集合D的基尼指数定义为：

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

- 基尼指数$Gini(D)$：表示了集合D的不确定性；
- 基尼指数$Gini(D,A)$：表示了经$A=a$分割后集合D的不确定性。
- 基尼指数值越大，样本集合的不确定性也就越大，跟熵相似。

CART算法每次仅对某个特征的值进行二分，而不是多分。此时，CART算法建立起来的是二叉树，而不是多叉树。

4.4 决策树

CART算法

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

输入：训练数据集D，阈值；

输出：CART决策树T。

根据训练数据集D，从根节点开始，递归地对每个节点进行以下操作，构建二叉树：

Step1：计算现有属性对该数据集D的基尼指数。即对每一个属性A，对其可能取的每个值a，根据样本是否满足$A=a$将D划分为D_1和D_2部分，计算$A=a$时的基尼指数；

Step2：在所有可能的属性A以及它们所有可能的划分点a中，选择基尼指数最小的属性及其对应可能的划分点作为最优属性和最优划分点。根据最优属性和最优划分点，从现节点生成两个子节点，将训练数据集依属性分配到两个子节点中去；

Step3：对两个子节点递归地调用Step1、2，直至满足条件（节点中的样本个数小于预定阈值，或样本集基尼指数小于预定阈值，或者没有更多属性）；

Step4：生成CART决策树。

③ CART（classification and regression tree）算法

➢ 与ID3、C4.5算法区别

- ID3、C4.5：属性A被选取建立决策树节点，如果它有3个取值A_1, A_2, A_3，会在决策树上建立一个三叉点，决策树是多叉树。

- CART：采用不停的二分。

a) 把属性A分成$\{A_1\}$和$\{A_2, A_3\}$、$\{A_2\}$和$\{A_1, A_3\}$、$\{A_3\}$和$\{A_1, A_2\}$三种情况，找到基尼系数最小的组合，比如$\{A_2\}$和$\{A_1, A_3\}$

b) 建立二叉树节点，一个节点是A_2对应的样本，另一个节点是$\{A_1, A_3\}$对应的样本。

c) 上述操作没有把属性A的取值完全分开，后面会对子节点继续选择属性A划分A_1和A_3。

与ID3、C4.5不同，在ID3或C4.5的一棵子树中，属性只会参与一次节点的建立。

4.4.4 过拟合与剪枝

➢ 过拟合（over-fitting）

- 也叫过学习（over-learning）
- 指一个算法在训练数据上表现很好，但在测试数据上或未来的新数据上的表现很差

➢ 欠拟合

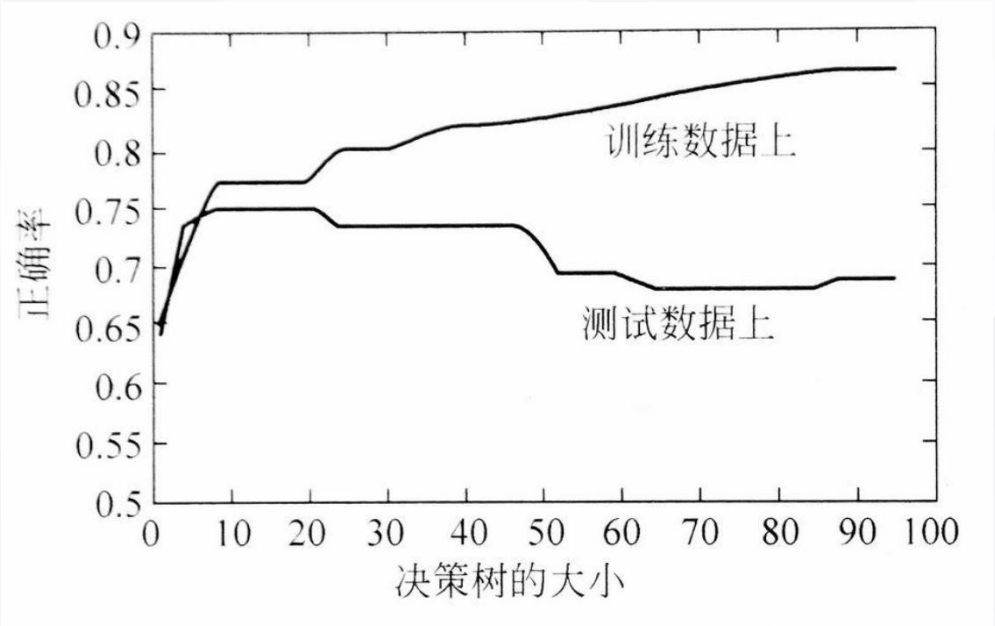
- 算法在训练数据上就没学好

一般来说，欠拟合容易解决，过拟合不可避免

4.4.4 过拟合与剪枝

➢ 决策树的过拟合

在有限样本情况下，决策树生长得很大（树枝很多或很深），则可能会抓住有限样本中由于采样的偶然性或噪声带来的假象，导致过拟合。



4.4.4 过拟合与剪枝

降低决策树过拟合风险的主要方法

- 控制决策树构建算法的终止条件
- 对决策树进行必要的剪枝

4.4.4 过拟合与剪枝

- 决策树过拟合解决方案-剪枝（pruning）

从已经生成的树上裁掉一些子树或叶节点，并将其根节点或父节点作为新的叶子节点，从而简化决策树模型。可分为先剪枝和后剪枝。

① 先剪枝：控制决策树的生长，在决策树构建过程中决定某节点是叶节点、还是继续分枝。

② 后剪枝：在决策树得到充分生长之后，再对其进行修剪。从叶节点开始，合并具有相同父节点的叶节点后不会导致纯度明显降低，则执行合并。

4.4.4 过拟合与剪枝

① 先剪枝方法分类

a) 数据划分法

将数据划分为训练样本和测试样本，基于训练样本对决策树进行生长，直至在测试样本上的分类错误率达到最小时停止生长。需要采用多次交叉验证(数据集多次划分)。

b) 阈值法

预设一个信息增益阈值，小于阈值的节点停止生长。

c) 信息增益的统计显著性分析

统计决策树已有节点的信息增益的分布，若决策树继续生长后信息增益的统计分布变化不显著，则决策树停止生长。

4.4.4 过拟合与剪枝

② 后剪枝方法分类

a) 减少分类错误率

对比剪枝前后的分类错误率的改变，由此判断是否合并分支

b) 最小代价与复杂性折中

折中考虑合并分枝后导致的错误率增加与复杂性减少，得到综合指标较优的决策树。可以定义代价函数进行优化。

c) MDL准则法（Minimal Description Length，最小描述长度）

核心思想为：最简单的树就是最好的树。为构建最简单的树，首先对决策树进行编码，再通过剪枝得到编码最短的决策树。

4.4.5 问题讨论

如何确定决策树生长的深度？

- 使用与训练样例截然不同的一套分离的样例，来评估通过后修剪方法从树上修剪节点的效用；
- 使用所有可用数据进行训练，但进行统计测试来估计扩展（或修剪）一个特定的节点是否有可能改善在训练集合外的样本上的性能；
- 使用一个明确的标准来衡量训练样例和决策树的复杂度，当这个编码长度最小时停止树增长。

4.4.5 问题讨论

决策树如何处理连续值属性？

- 动态地定义新的离散值属性，即把连续值属性的值域分割为离散的区间集合；
- 连续的属性可以分割成多个区间，而不是单一阈值的两个空间。

4.4.5 问题讨论

决策树如何处理缺失值的属性？假定（$x, C(x)$）是数据集D中的一个训练样例，并且其属性A的值$A(x)$未知。

- 1) 赋给它节点n的训练样例中该属性的最常见值；
- 2) 赋给它节点n的被分类为$C(x)$的训练样例中该属性的最常见值；
- 3) 为A的每个可能值赋予一个概率，而不是简单地将最常见的值赋给$A(x)$

西瓜数据集 2.0α

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜
1	–	蜷缩	浊响	清晰	凹陷	硬滑	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	–	是
3	乌黑	蜷缩	–	清晰	凹陷	硬滑	是
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是
5	–	蜷缩	浊响	清晰	凹陷	硬滑	是
6	青绿	稍蜷	浊响	清晰	–	软粘	是
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是
8	乌黑	稍蜷	浊响	–	稍凹	硬滑	是
9	乌黑	–	沉闷	稍糊	稍凹	硬滑	否
10	青绿	硬挺	清脆	–	平坦	软粘	否
11	浅白	硬挺	清脆	模糊	平坦	–	否
12	浅白	蜷缩	–	模糊	平坦	软粘	否
13	–	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	–	软粘	否
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否
17	青绿	–	沉闷	稍糊	稍凹	硬滑	否

4.4 决策树

4.4.6 小结

- 树和决策树概念
- 决策树构建流程
- 经典决策树算法

  ID3算法

  C4.5算法

  CART算法