第 5 章

决策树分类

数据挖掘的分类指的是通过对事物特征的定量分析，形成能够进行分类预测的分类模型，利用该模型能够预测一个具体的事物所属的类别。决策树是一种十分常用的分类方法，决策树是带有判决规则的一种树，可以依据树中的判决规则来预测未知样本的类别。

5.1 相似性和相异性的度量

相似性和相异性是数据挖掘中两个非常重要的概念。两个对象之间的相似度是这两个对象相似程度的数值度量，通常在 0(不相似)和 1(完全相似)之间取值。两个对象之间的相异度是这两个对象差异程度的数值度量，两个对象越相似，它们的相异度就越低，通常用“距离”作为相异度的同义词。数据对象之间相似性和相异性的度量有很多种方法，如何选择度量方法依赖于对象的数据类型、数据的量值是否重要以及数据的稀疏性等。

5.1.1 数据对象之间的相异度

人们通常所说的相异度其实就是距离。距离越小，相异度越低，则对象越相似。度量对象间差异性的距离形式有闵可夫斯基距离、马哈拉诺比斯距离、汉明距离和杰卡德距离。

1. 闵可夫斯基距离

在 m 维欧几里得空间中，每个点是一个 m 维实数向量，两个点 $x_i=(x_{i1},x_{i2},\cdots,x_{im})$ 与 $y_j=(y_{j1},y_{j2},\cdots,y_{jm})$ 之间的闵可夫斯基距离 L_r 定义如下：

$$d(x_i,y_j)=\left(\sum_{k=1}^{m}|x_{ik}-y_{jk}|^r\right)^{1/r}$$

当 $r=2$ 时，又称为 L_2 范式距离、欧几里得距离，两个点 $x_i=(x_{i1},x_{i2},\cdots,x_{im})$ 与 $y_j=(y_{j1},y_{j2},\cdots,y_{jm})$ 之间的欧几里得距离 $d(x_i,y_j)$ 定义如下：

$$d(x_i,y_j)=\sqrt{\sum_{k=1}^{m}|x_{ik}-y_{jk}|^2}$$

另一个常用的距离是 L_1 范式距离，又称曼哈顿距离，两个点的曼哈顿距离为每维距离之和。之所以称为“曼哈顿距离”，是因为这里在两个点之间行进时必须要沿着网格线前进，就如同沿着城市(如曼哈顿)的街道行进一样。

另一个有趣的距离形式是 L_∞ 范式距离，即切比雪夫距离，也就是当 r 趋向无穷大时 L_r 范式的极限值。当 r 增大时，只有那个具有最大距离的维度才真正起作用，因此，通常 L_∞

范式距离定义为在所有维度下$|x_i-y_i|$中的最大值。

考虑二维欧几里得空间(即通常所说的平面)上的两个点(2,5)和(5,9)。它们的L_2范式距离为$\sqrt{(2-5)^2+(5-9)^2}=5$,L_1范式距离为$|2-5|+|5-9|=7$,而L_∞范式距离为$\max(|2-5|,|5-9|)=\max(3,4)=4$。

距离(如欧几里得距离)具有一些众所周知的性质。如果$d(x,y)$表示两个点x和y之间的距离,则如下性质成立。

(1) $d(x,y)\geqslant 0$(距离非负),当且仅当$x=y$时,$d(x,y)=0$(只有点到自身的距离为0,其他的距离都大于0)。

(2) $d(x,y)=d(y,x)$(距离具有对称性)。

(3) $d(x,y)\leqslant d(x,z)+d(z,y)$(三角不等式)。

2. 马哈拉诺比斯距离

马哈拉诺比斯距离为数据的协方差距离,它是一种有效地计算两个未知样本集的相似度的方法,与欧几里得距离不同的是它考虑各种特性之间的联系,并且是尺度无关的(独立于测量尺度)。向量$\boldsymbol{x}_i$与$\boldsymbol{y}_i$之间的马哈拉诺比斯距离定义如下:

$$d(\boldsymbol{x}_i,\boldsymbol{y}_j)=\sqrt{(\boldsymbol{x}_i-\boldsymbol{y}_j)^{\mathrm{T}}\boldsymbol{S}^{-1}(\boldsymbol{x}_i-\boldsymbol{y}_j)}$$

其中,$\boldsymbol{S}$为协方差矩阵,若$\boldsymbol{S}$为单位矩阵,则马哈拉诺比斯距离变为欧几里得距离。

3. 汉明距离

两个等长字符串之间的汉明距离是两个字符串对应位置的不同字符的个数。换句话说,它就是将一个字符串变换成另外一个字符串所需要替换的字符个数。举例如下。

'1011101'与'1001001'之间的汉明距离是2。

'2143896'与'2233796'之间的汉明距离是3。

'toned'与'roses'之间的汉明距离是3。

4. 杰卡德距离

杰卡德距离(Jaccard Distance)用于衡量两个集合的差异性,它是杰卡德相似度的补集,被定义为1减去Jaccard相似度。Jaccard相似度用来度量两个集合之间的相似性,它被定义为两个集合交集的元素个数除以并集的元素个数,即集合A和B的相似度$\mathrm{sim}(A,B)$为

$$\mathrm{sim}(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

集合A、B的杰卡德距离$d_J(A,B)$为

$$d_J(A,B)=1-\mathrm{sim}(A,B)$$

5. 非度量的相异度

有些相异度不满足一个或多个距离性质,如集合差。

设有两个集合A和B,A和B的集合差$A-B$定义为由所有属于A且不属于B的元素组成的集合。例如,如果$A=\{1,2,3,4\}$,而$B=\{2,3,4\}$,则$A-B=\{1\}$,而$B-A=$空集。若将两个集合A和B之间的距离定义为$d(A,B)=\mathrm{size}(A-B)$,其中size是一个函数,它返回集合元素的个数。该距离是大于或等于零的整数值,但不满足非负性的第二部分,也不满足对称性,同时还不满足三角不等式。然而,如果将相异度修改为$d(A,B)=\mathrm{size}(A-B)+\mathrm{size}(B-A)$,则这些性质都可以成立。

5.1.2　数据对象之间的相似度

对象(或向量)之间的相似性可用距离和相似系数来度量。距离常用来度量对象之间的相似性,距离越小相似性越大。相似系数常用来度量向量之间的相似性,相似系数越大,相似性越大。

将距离用于相似度大小度量时,距离的三角不等式(或类似的性质)通常不成立,但是对称性和非负性通常成立。更明确地说,如果 $s(x,y)$ 是数据点 x 和 y 之间的相似度,则相似度具有如下典型性质。

(1) 仅当 $x=y$ 时 $s(x,y)=1$。($0\leqslant s\leqslant 1$)

(2) 对于所有 x 和 y,$s(x,y)=s(y,x)$。(对称性)

对于相似度,没有三角不等式性质。然而,有时可以将相似度简单地变换成一种度量距离,余弦相似性度量和 Jaccard 相似性度量就是这样的两个例子。

令 x_i、x_j 是 m 维空间中的两个向量,r_{ij} 是 x_i 和 x_j 之间的相似系数,r_{ij} 通常满足以下条件。

(1) $r_{ij}=1\Leftrightarrow x_i=x_j$。

(2) $\forall x_i、x_j, r_{ij}\in[0,1]$。

(3) $\forall x_i、x_j, r_{ij}=r_{ji}$。

常用的相似系数度量方法有相关系数法、夹角余弦法。

1. 二元数据的相似性度量

两个仅包含二元属性的对象之间的相似性度量也称为相似系数,并且通常取值为 0～1,值为 1 表明两个对象完全相似,而值为 0 表明两个对象一点也不相似。

设 x 和 y 是两个对象,都由 n 个二元属性组成。这样的两个对象(即两个二元向量)的比较可生成如下四个量(频率)。

$f_{00}=x$ 取 0 并且 y 取 0 的属性个数;

$f_{01}=x$ 取 0 并且 y 取 1 的属性个数;

$f_{10}=x$ 取 1 并且 y 取 0 的属性个数;

$f_{11}=x$ 取 1 并且 y 取 1 的属性个数。

一种常用的相似性系数是简单匹配系数(Simple Matching Coefficient, SMC),简单匹配系数定义如下:

$$\mathrm{SMC}=\frac{\text{值匹配的属性个数}}{\text{属性个数}}=\frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}}$$

该度量对出现和不出现都进行计数。因此,SMC 可以在一个仅包含是非题的测验中用来发现回答问题相似的学生。

假定 x 和 y 是两个数据对象,代表一个事务矩阵的两行(两个事务)。如果每个非对称的二元属性对应于商店的一种商品,则 1 表示该商品被购买,而 0 表示该商品未被购买。由于未被顾客购买的商品数远大于被其购买的商品数,因而像 SMC 这样的相似性度量将会判定所有的事务都是类似的。这样,常常使用 Jaccard 系数(Jaccard Coefficient)来处理仅包含非对称的二元属性的对象。Jaccard 系数通常用符号 J 表示,由如下等式定义:

$$J=\frac{\text{匹配的个数}}{\text{不涉及 0-0 匹配的属性个数}}=\frac{f_{11}}{f_{01}+f_{10}+f_{11}}$$

【例 5-1】 SMC 和 J 相似性系数。

为了解释 SMC 和 J 这两种相似性度量之间的差别，对如下二元向量计算 SMC 和 J。

x = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)；

y = (0, 0, 0, 0, 0, 0, 1, 0, 0, 1)；

x 取 0 并且 y 取 1 的属性个数，f_{01} = 2；

x 取 1 并且 y 取 0 的属性个数，f_{10} = 1；

x 取 0 并且 y 取 0 的属性个数，f_{00} = 7；

x 取 1 并且 y 取 1 的属性个数，f_{11} = 0。

$$\mathrm{SMC}=\frac{f_{11}+f_{00}}{f_{01}+f_{10}+f_{11}+f_{00}}=\frac{0+7}{2+1+0+7}=0.7$$

$$J=\frac{f_{11}}{f_{01}+f_{10}+f_{11}}=\frac{0}{2+1+0}=0$$

2. 相关系数法度量向量之间的相似性

令 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 是 m 维空间中的两个向量，令 $\bar{\boldsymbol{x}}_i=\frac{1}{m}\sum_{k=1}^{m}\boldsymbol{x}_{ik}$，$\bar{\boldsymbol{x}}_j=\frac{1}{m}\sum_{k=1}^{m}\boldsymbol{x}_{jk}$，$\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的相关系数 $r_{ij}=\dfrac{\sum_{k=1}^{m}(\boldsymbol{x}_{ik}-\bar{\boldsymbol{x}}_i)(\boldsymbol{x}_{jk}-\bar{\boldsymbol{x}}_j)}{\sqrt{\sum_{k=1}^{m}(\boldsymbol{x}_{ik}-\bar{\boldsymbol{x}}_i)^2}\times\sqrt{\sum_{k=1}^{m}(\boldsymbol{x}_{jk}-\bar{\boldsymbol{x}}_j)^2}}$，取值范围为[−1,1]，其中 0 表示不相关，1 表示正相关，−1 表示负相关。相关系数的绝对值越大，则表明 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 相关度越高。当 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 线性相关时，相关系数取值为 1(正线性相关) 或 −1(负线性相关)。

3. 余弦相似度

余弦相似度也称为余弦距离，是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。余弦距离在有维度的空间下才有意义，这些空间有欧几里得空间和离散欧几里得空间。在上述空间下，点可以表示方向，两个点的余弦距离实际上是点所代表的向量之间的夹角的余弦值。

给定向量 $\boldsymbol{x}$ 和 $\boldsymbol{y}$，其夹角 θ 的余弦 $\cos\theta$ 等于它们的内积除以两个向量的 L_2 范式距离(即它们到原点的欧几里得距离)乘积：

$$\cos\theta=\frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|\cdot\|\boldsymbol{y}\|}$$

$\cos\theta$ 范围为[−1,1]，若 $\cos\theta=0$，即两向量正交时，表示完全不相似。

相比距离度量，余弦相似度更加注重两个向量在方向上的差异，而非距离或长度上。

【例 5-2】 假设新闻 a 和新闻 b 对应的向量分别是 $\boldsymbol{x}(x_1,x_2,\cdots,x_{100})$ 和 $\boldsymbol{y}(y_1,y_2,\cdots,y_{100})$，则新闻 a 和新闻 b 的余弦相似度 $\cos\theta=\dfrac{x_1y_1+x_2y_2+\cdots+x_{100}y_{100}}{\sqrt{x_1^2+x_2^2+\cdots+x_{100}^2}\sqrt{y_1^2+y_2^2+\cdots+y_{100}^2}}$。

当两条新闻向量夹角等于 0°时，这两条新闻完全重复；当夹角接近于 0°时，两条新闻相似；夹角越大，两条新闻越不相关。

4. 编辑距离

编辑距离只适用于比较两个字符串之间的相似性。字符串 $x=x_1x_2\cdots x_n$ 与 $y=$

$y_1y_2\cdots y_m$ 的编辑距离指的是要用最少的字符操作数目将字符串 x 转换为字符串 y。这里所说的字符操作包括：一个字符替换成另一个字符，插入一个字符，删除一个字符。将字符串 x 变换为字符串 y 所用的最少字符操作数称为字符串 x 到 y 的编辑距离，表示为 $d(x,y)$。一般来说字符串编辑距离越小，两个串的相似度越大。

【例 5-3】 两个字符串 x=eeba 和 y=abac 的编辑距离为 3。将 x 转换为 y，需要进行如下操作。

(1) 将 x 中的第一个 e 替换成 a。

(2) 删除 x 中的第二个 e。

(3) 在 x 中最后添加一个 c。

编辑距离具有下面几个性质。

(1) 两个字符串的最小编辑距离是两个字符串的长度差。

(2) 两个字符串的最大编辑距离是两字符串中较长字符串的长度。

(3) 只有两个相等的字符串的编辑距离才会为 0。

(4) 编辑距离满足三角不等式，即 $d(x,z)\leqslant d(x,y)+d(y,z)$。

5.2　分类概述

5.2.1　分类的基本概念

从对与错、好与坏的简单分类，到复杂的生物学中的界、门、纲、目、科、属、种，人类对客观世界的认识离不开分类，通过将有共性的事物归到一类，以区别不同的事物，使得对大量的繁杂事物条理化和系统化。

数据挖掘的分类指的是通过对事物特征的定量分析，形成能够进行分类预测的分类模型(分类函数、分类器)，利用该模型能够预测一个具体的事物所属的类别。注意，分类的类别取值必须是离散的，分类模型做出的分类预测不是归纳出的新类，而是预先定义好的目标类。因此，分类也称为有监督学习，与之相对应的是无监督学习，例如聚类。分类与聚类的最大区别在于，分类数据中的一部分数据的类别是已知的，而聚类数据中所有数据的类别是未知的。

现实商业活动中的许多问题都能抽象成分类问题，在当前的市场营销行为中很重要的一个特点是强调目标客户细分，例如银行贷款员需要分析贷款申请者数据，搞清楚哪些贷款申请者是“安全的”，哪些贷款申请者是“不安全的”。其他场景如推荐系统、垃圾邮件过滤、信用卡分级等，都能转化为分类问题。

分类任务的输入数据是记录的集合，记录也称为样本、样例、实例、对象、数据点，用元组 (x,y) 表示，其中 x 是对象特征属性的集合，而 y 是一个特殊的属性，称为类别属性、分类属性或目标属性，指出样例的类别是什么。表 5-1 列出一个动物样本数据集，用来将动物分为两类：爬行类和鸟类。属性集指明动物的性质，如翅膀数量、脚的只数、是否产蛋、是否有毛等。尽管表 5-1 中的属性主要是离散的，但是属性集也可以包含连续特征。另一方面，类别属性必须是离散属性，这是区别分类与回归的关键特征。回归是一种预测模型，其目标属性 y 是连续的。

表 5-1 动物样本数据集

动物	翅膀数量	脚的只数	是否产蛋	是否有毛	动物类别
狗	0	4	否	是	爬行类
猪	0	4	否	是	爬行类
牛	0	4	否	是	爬行类
麻雀	2	2	是	是	鸟类
鸽子	2	2	是	是	鸟类
天鹅	2	2	是	是	鸟类

分类的形式化定义是：分类就是通过学习样本数据集得到一个目标函数 f，把每个特征属性集 x 映射到一个预先定义的类标号 y。目标函数也称为分类模型。

5.2.2 分类的一般流程

分类技术是一种根据输入数据集建立分类模型的技术。常用的分类模型主要有决策树分类、贝叶斯分类、人工神经网络分类方法、k-近邻分类、支持向量机分类等。这些技术都是通过一种学习算法确定相应的分类模型，该模型能够很好地拟合输入数据中类标号和特征属性集之间的联系。使用学习算法学习样本数据得到的模型不仅要能很好地拟合样本数据，还要能够正确地预测未知样本的类标号。因此，训练分类算法的主要目标就是建立具有强泛化能力的分类模型，即建立能够准确地预测未知样本类标号的分类模型。

求解分类问题的一般流程如图 5-1 所示。首先，得到一个样本数据集（也称训练数据集），它由类标号已知的记录组成。然后选择分类学习算法学习训练数据集建立分类模型，也称使用训练数据集训练分类模型从而得到一个训练好的分类模型，随后使用检验数据集评估训练好的分类模型的性能以及调整模型参数，检验数据集是由类标号已知的记录组成，最后将符合要求的分类模型用于未知样本的分类。

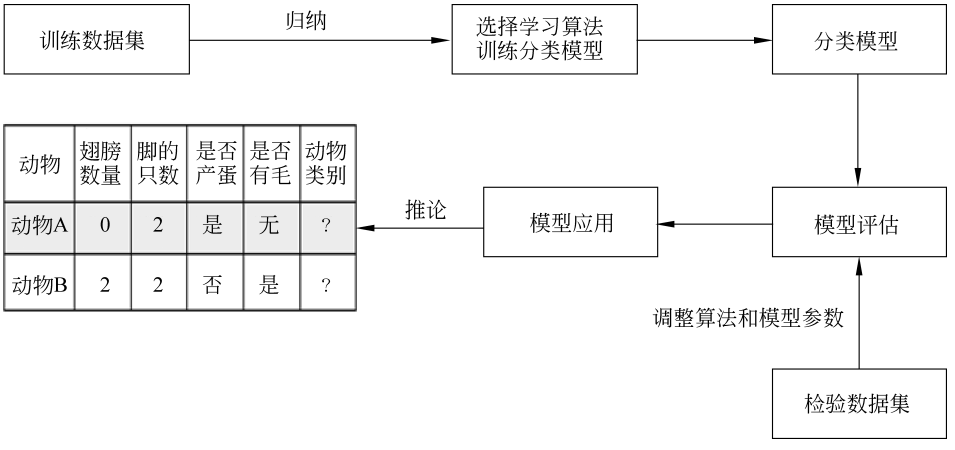

图 5-1 求解分类问题的一般流程

分类模型的性能根据模型正确和错误预测的检验记录数进行评估，对于二分类问题，记

f_{00}表示实际类标号为 0 被正确预测为类 0 的记录数，f_{01}代表原本属于类 0 但被误认为类 1 的记录数，f_{11}表示实际类标号为 1 被正确预测为类 1 的记录数，f_{10}代表原本属于类 1 但被误认为类 0 的记录数。则被分类模型正确预测的样本总数是 $f_{00}+f_{11}$，而被错误预测的样本总数是 $f_{01}+f_{10}$，因而分类模型的性能可以用准确率表示，具体如下。

$$\text{准确率}=\frac{\text{正确预测数}}{\text{预测总数}}=\frac{f_{00}+f_{11}}{f_{00}+f_{01}+f_{11}+f_{10}}$$

同样，分类模型的性能也可以用错误率来表示，具体如下。

$$\text{错误率}=\frac{\text{错误预测数}}{\text{预测总数}}=\frac{f_{01}+f_{10}}{f_{00}+f_{01}+f_{11}+f_{10}}$$

大多数分类算法都在寻求这样一些分类模型，当把它们应用于检验集时具有最高的准确率，或者等价地，具有最低的错误率。

5.3　决策树分类概述

在现实生活中，经常会遇到各种选择，如人们要去室外打羽毛球，一般会根据“天气”“温度”“湿度”“刮风”这几个条件来判断，最后得到结果：去打羽毛球还是不去打羽毛球？如果把判断背后的逻辑整理成一个结构图，它实际上是一个树状图，这就是决策树。

5.3.1　决策树的工作原理

1. 决策树的概念

决策树简单来说就是带有判决规则的一种树，可以依据树中的判决规则来预测未知样本的类别和值。决策树就是通过树结构来表示各种可能的决策路径，以及每个路径的结果。一棵决策树一般包含一个根结点、若干个内部结点和若干个叶子结点。

(1) 叶子结点对应于决策结果。

(2) 每个内部结点对应于一个属性测试，每个内部结点包含的样本集合根据属性测试的结果被划分到它的子结点中。

(3) 根结点包含全部训练样本。

(4) 从根结点到每个叶子结点的路径对应了一条决策规则。

一棵预测顾客是否会购买计算机的决策树如图 5-2 所示，其中内部结点用矩形表示，叶子结点用椭圆表示，分支表示属性测试的结果。为了对未知的顾客判断其是否会购买计算

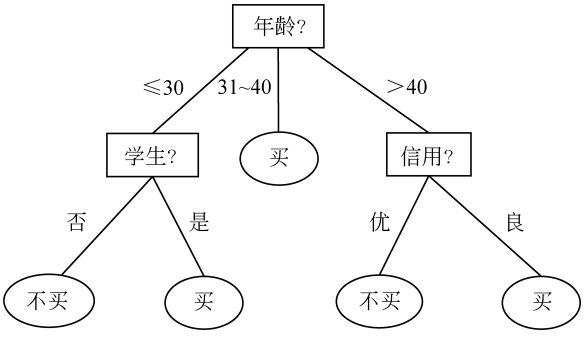

图 5-2　是否会购买计算机的决策树

机，将顾客的属性值在决策树上进行判断、选取相应的分支，直到到达叶子结点，从而得到顾客所属的类别。决策树中，从根结点到叶子结点的一条路径就对应着一条合取规则，对应着一条分类规则，对应着样本的一个分类。

在是否会购买计算机的决策树中包含三种结点：根结点，它没有入边，有 3 条出边；内部结点(子结点，非叶子结点)，有一条入边和两条或多条出边；叶子结点，只有一条入边，但没有出边。在这个例子中，用来进行类别决策的属性为年龄、学生、信用。

在沿着决策树从上到下的遍历过程中，在每个内部结点都有一个测试，不同的测试结果引出不同的分枝，最后会到达一个叶子结点，这一过程就是利用决策树进行分类的过程。

决策树分类方法，实际上是通过对训练样本的学习，建立分类规则，再依据分类规则，实现对新样本的分类。决策树分类方法属于有指导(监督)的分类方法。训练样本有两类属性：划分属性和类别属性。一旦构造了决策树，对新样本进行分类就相当容易。从树的根结点开始，将测试条件用于新样本，根据测试结果选择适当的分支，沿着该分支或者到达另一个内部结点，使用新的测试条件继续上述过程，直到到达一个叶子结点，也就是得到新样本所属的类别。

2. 构建决策树

决策树算法作为一种分类算法，目标就是将具有 m 维特征的 n 个样本分到 c 个类别中去。相当于做一个映射 $c=f(n)$，将样本经过一种变换赋予一种类别标签。决策树为了达到这一目的，将分类的过程表示成一棵树，每次通过选择一个特征来进行分叉。不同的决策树算法选择不同的特征选择方案进行分叉，如 ID3 决策树使用信息增益最大选择划分特征，C4.5 决策树用信息增益率最大选择划分特征，CART 用基尼指数最小选择划分特征。

构建决策树的过程，就是通过学习样本数据集获得分类知识的过程，得到一种逼近离散值的目标函数的过程。决策树学习本质上是从训练数据集中归纳出一组分类规则，学习到的一组分类规则被表示为一棵决策树。决策树学习是以样本为基础的归纳学习，它采用自顶向下递归的方式来生长决策树，随着树的生长，完成对训练样本集的不断细分，最终都被细分到每个叶子结点上。决策树是一种树状结构，其中每个内部结点表示一个属性上的测试，每个分支代表一个测试输出，每个叶子结点代表一种类别。构建决策树的具体步骤如下。

(1) 选择最好的属性作为测试属性并创建树的根结点，开始时，所有的训练样本都在根结点。

(2) 为测试属性每个可能的取值产生一个分支。

(3) 根据属性的每个可能值将训练样本划分到相应的分支形成子结点。

(4) 对每个子结点，重复上面的过程，直到所有的结点都是叶子结点。

构建决策树的流程可用下述伪代码表示。

输入：训练样本集 $S=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$和划分属性集 $A=\{a_1,a_2,\cdots,a_m\}$。
输出：以 node 为根结点的一个决策树。
//根据给定训练样本集 S 和划分属性集 A 构建决策树
函数 GenerateDTree(S, A){
1: 生成结点 node;
2: if S 中训练样本全属于同一类别 c_j then

```
3:  将 node 标记为 c_j 类叶子结点; return
4:  end if
5:  if A =∅ or S 中样本在 A 上取值相同 then
6:      将 node 标记为叶子结点,其类别标记为 S 中样本数最多的类; return
7:  end if
8:  从 A 中选择最优化分属性 a*
9:  for a* 的每一值 a[i] do
10: 为 node 生成一个分支结点 node_i; 令 S_i 表示 S 中在 a* 上取值为 a[i]的样本子集;
11:     if S_i=∅ then
12:         将分支结点 node_i 标记为叶子结点,其类别为 S 中样本最多的类; return
13:     else
14:         调用 GenerateDTree(S_i, A-{a*})递归创建分支结点;
15:     end if
16: end for}
```

在上述决策树算法中,有如下三种情形会导致函数递归返回。

(1) 当前结点包含的样本全部属于同一类别,不需要再划分。

(2) 当前划分属性集为空,或者所有样本在当前所有划分属性上取值相同,无法划分。

(3) 当前结点包含的样本集为空,不能划分。

构建决策树的过程就是选择什么属性作为结点的过程,原则上讲,对于给定的属性集,若可以构造的不同决策树的数目达到指数级,则找出最佳决策树在计算上通常是不可行的。于是,人们开发了一些有效的算法,能够在合理的时间内构造出具有一定准确率的次优决策树。这些算法通常都是采用贪心策略,在选择划分记录的属性时,采用一系列局部最优决策来构造决策树,Hunt 算法就是一种这样的算法。Hunt 算法是许多决策树算法的基础,包括 ID3 算法、C4.5 决策树算法和 CART 算法。

3. Hunt 算法

Hunt 算法是一种采用局部最优策略的决策树构建算法,通过将训练记录相继划分为较纯的子集,以递归的方式建立决策树。设 D_t 是与结点 t 相关联的训练样本集,$y=\{y_1, y_2, \cdots, y_c\}$ 是类标号,Hunt 算法递归构建决策树的过程如下。

(1) 如果 D_t 中所有记录都属于同一个类,则 t 是叶子结点,用 y_t 标记。

(2) 如果 D_t 中包含多个类的样本,则选择一个属性将相关联的训练样本集划分成较小的子集。对于属性的每个取值,创建一个子结点,并根据属性的不同取值将 D_t 中的样本分布到相应的子结点中。然后,对于每个子结点,递归地调用该算法。

为了演示 Hunt 算法构建决策树的过程,考虑一个预测拖欠银行贷款的贷款者数据集如表 5-2 所示,在表 5-2 所示的数据集中,每条记录都包含贷款者的个人信息,以及贷款者是否拖欠贷款的类标号。

表 5-2　银行贷款的贷款者数据集

样本序号	有房者	婚姻状况	年收入/元	拖欠贷款者
1	是	单身	125000	否
2	否	已婚	100000	否

续表

样本序号	有房者	婚姻状况	年收入/元	拖欠贷款者
3	否	单身	70000	否
4	是	已婚	120000	否
5	否	离异	95000	是
6	否	已婚	60000	否
7	是	离异	220000	否
8	否	单身	85000	是
9	否	已婚	75000	否
10	否	单身	90000	是

使用 Hunt 算法构建决策树时，决策树初始只有一个叶子结点，即类标号为“拖欠贷款者＝否”的叶子结点，如图 5-3(a)所示，表示大多数贷款者都没有拖欠贷款。之后，对该树进行细分，从表 5-2 所示的数据集可以看出最终决策树的根结点应包含两类记录：一类是“拖欠贷款者＝否”的记录集；一类是“拖欠贷款者＝是”的记录集。然后，将“有房者”作为划分属性，如图 5-3(b)所示，根据“有房者”属性的不同取值将数据集划分为两个较小的子集。接下来，对两个子集递归地调用 Hunt 算法。

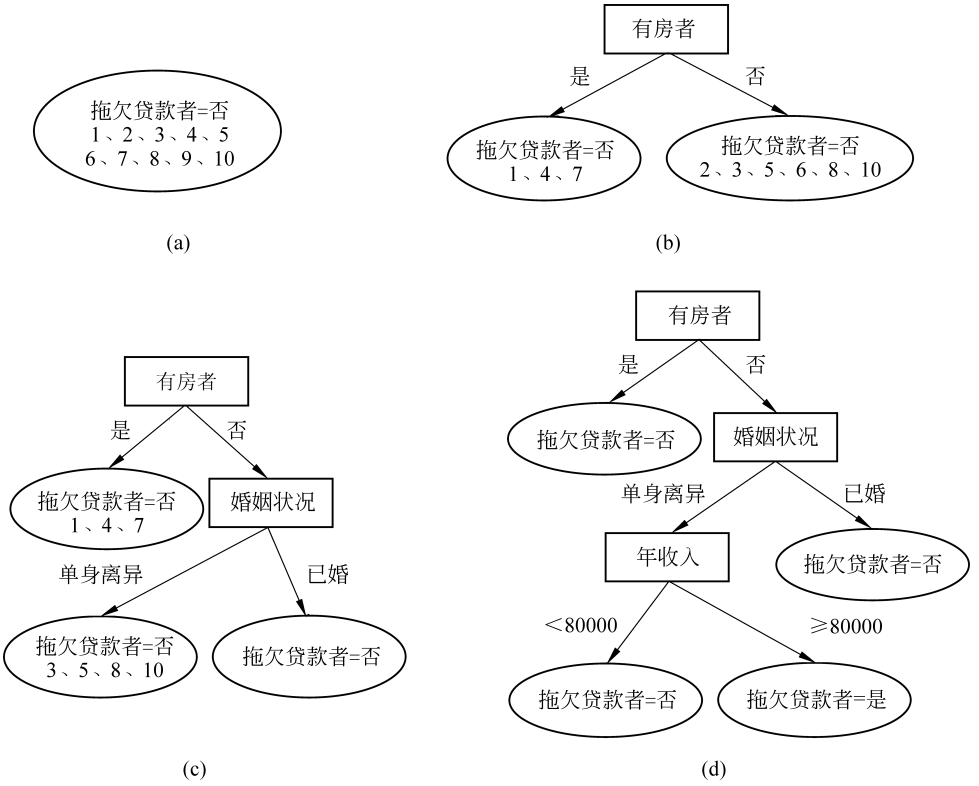

图 5-3　Hunt 算法构建决策树的过程