第5章 数据分类分析方法

分类(classification)是一种重要的数据分析形式,它是提取刻画重要数据类的模型,也是机器学习和数据挖掘领域中的一整套用于处理分类问题的方法。该类方法是有监督学习类型的,即:给定一个数据集,所有实例都由一组属性来描述,每个实例仅属于一个类别,在给定数据集上运行可以学习得到一个从属性值到类别的映射,进而可使用该映射去分类新的未知实例,这种映射又称为模型或分类器(classifer)。在数据挖掘社区遴选出的十大算法中六个都是这类方法,这也反映出此类方法在数据挖掘中被使用的广泛程度。最早这类算法只能处理标称类别数据,如今已扩展到支持数值、符号乃至混合型的数据类型。具体的应用领域也很广泛,如临床决策、生产制造、文档分析、生物信息学、空间数据建模(地理信息系统)等。

本章先引入分类的基本概念(5.1 节),然后介绍数据分类分析的基本技术,包括最常用的决策树分类器的构建方法(5.2 节),基于概率统计思想的贝叶斯分类算法(5.3 节),具有统计学习理论坚实基础的在所有知名的数据挖掘算法中最健壮、最准确的支持向量机(Support Vector Machine)算法(5.4 节),以及通过构建一组基于学习器进行集成学习的 Adaboost 算法(5.7 节),最后通过使用 Python 语言给出具体案例,使大家能够熟悉数据分类分析的整个过程。

5.1 基本概念和术语

本节给出分类分析相关的基本概念及其基本术语,为读者研究分类分析建立基础。5.1.1 节通过一个描述性模型介绍分类中的有关定义。5.1.2 节介绍分类的方法,并对相关的术语做出了解释。

5.1.1 什么是分类

分类(classfication)任务就是通过学习得到一个目标函数(target function)f,把每个属性集 x 映射到一个预先定义的类标号 y。

目标函数也称分类模型(classfication model)。分类模型可以用于以下目的。

描述性建模,分类模型可以作为解释性工具,用于区分不同类中的对象。例如,对于生物学家或者其他人,一个描述性模型有助于概括表 5-1 中的数据,并说明哪些特征决定一种脊椎动物是哺乳类、爬行类、鸟类、鱼类或者两栖类。

表 5-1 脊椎动物的数据集

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
人类	恒温	毛发	是	否	否	是	否	哺乳类
蟒蛇	冷血	鳞片	否	否	否	否	是	爬行类
鲑鱼	冷血	鳞片	否	是	否	否	否	鱼类
鲸	恒温	毛发	是	是	是	否	否	哺乳类
青蛙	冷血	无	否	半	否	是	是	两栖类
巨蜥	冷血	鳞片	否	否	否	是	否	爬行类
蝙蝠	恒温	毛发	是	否	是	是	是	哺乳类
鸽子	恒温	羽毛	否	是	是	是	否	鸟类
猫	恒温	软毛	是	否	否	是	否	哺乳类
豹纹鲨	冷血	鳞片	是	是	否	否	否	鱼类
海龟	冷血	鳞片	否	半	否	是	否	爬行类
企鹅	恒温	羽毛	否	半	否	是	否	鸟类
豪猪	恒温	刚毛	是	否	否	是	是	哺乳类
鳗	冷血	鳞片	否	是	否	否	否	鸟类
蝾螈	冷血	无	否	半	否	是	是	两栖类

预测性建模，分类模型还可以用于预测未知记录的类标号。如图 5-1 所示，分类模型可以看作一个黑箱，当给定未知记录的属性集上的值时，它自动地赋予未知样本类标号。例如，假设有一种叫作毒蜥的生物，其特征如表 5-2 所示。

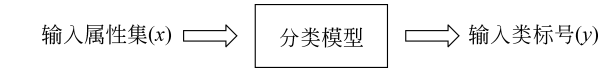

图 5-1 分类器的任务是根据输入属性集 x 确定类标号 y

表 5-2 毒蜥特征

名字	体温	表皮覆盖	胎生	水生动物	飞行动物	有腿	冬眠	类标号
毒蜥	冷血	鳞片	否	否	否	是	是	?

可以根据表 5-1 中的数据集建立的分类模型确定该生物所属的类。

假设销售经理希望预测一位给定的顾客一次购物期间将花多少钱，这个数据分析任务就是**数值预测**(numeric prediction)的一个例子，其中所构造的模型预测一个连续值函数或有序值，而不是类标号。这种模型是**预测器**(predictor)。**回归分析**(regression analysis)是数值预测最常用的统计方法，因此这两个术语常作为同义词使用，尽管还存在其他数值预测方法。分类和数值预测是预测问题的两种主要类型。

5.1.2 解决分类问题的一般方法

分类技术(或分类法)是一种根据输入数据集建立分类模型的系统方法。分类法包括决策树分类法、基于规则的分类法、神经网络、支持向量机和朴素贝叶斯分类法。这些技术都使用一种**学习算法**(learning algorithm)确定分类模型,该模型能够很好地拟合输入数据中类标号和属性集之间的联系。学习算法得到的模型不仅要很好地拟合输入数据,还要能够正确地预测未知样本的类标号。因此,训练算法的主要目标就是建立具有很好的泛化能力的模型,即建立能够准确地预测未知样本类标号的模型。

图 5-2 展示了解决分类问题的一般流程。首先,需要一个**训练集**(training set),它由类标号已知的记录组成。使用训练集建立分类模型,该模型随后将运用于**检验集**(test set),检验集由类标号未知的记录组成。

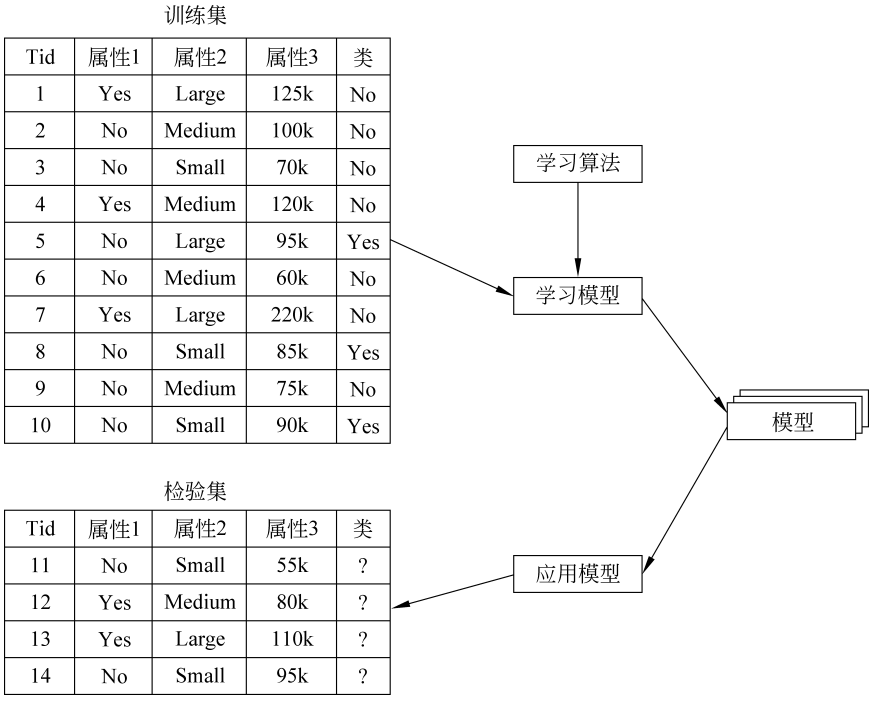

图 5-2 建立分类模型的一般流程

由于提供了每个训练元组的类标号,这一阶段也成为**监督学习**(supervised learning,即分类器的学习是在被告知每个训练元组属于哪个类的"监督"下进行的)。它不同于**无监督学习**(unsupervised learning,或聚类),每个训练元组的类标号是未知的,并且要学习的类的个数或集合也可能事先不知道。例如,如果没有用于训练集的数据,则可以使用聚类尝试确定"相似元组的组群"。

分类过程的学习模型也可以看作是学习一个映射或函数 $y=f(X)$,它可以预测给定元组 X 的类标号 y。在这种观点下,我们希望学习把数据类分开的映射或函数。在典型情况下,该映射用分类规则、决策树或数学公式的形式提供。

在应用模型阶段,使用模型进行分类。首先要评估分类器的预测准确率。如果使用训练集衡量分类器的准确率,则评估可能是乐观的,因为分类器趋向于**过分拟合**(overfitting)该数据(即在学习期间,它可能包含了训练数据中的某些特定的异常,这些异常不在一般数据集中出现)。因此,需要使用由检验元组和与它们相关联的类标号组成的**检验集**(test set)。它们独立于训练元组,即不使用它们构造分类器。

分类器在给定检验集上的**准确率**(accuracy)是分类器正确分类的检验元组所占的百分比。每个检验元组的类标号与学习模型对该元组的类预测进行比较。如果认为分类器的准确率是可以接受的,那么就可以用它对类标号未知的数据元组进行分类(这种数据在机器学习中也称为"未知的"或"先前未见到的"数据)。

5.2 决策树算法

本节介绍决策树分类法,这是一种简单但广泛使用的分类技术。在5.2.1节通过案例对决策树归纳过程做了介绍。决策树的建立过程在5.2.2节给出。5.2.3节和5.2.4节分别给出属性测试条件的方法和选择最佳划分的度量的方法。5.2.5节给出决策树归纳算法。树剪枝的概念在5.2.6节介绍。5.2.7节对决策树归纳的特点做了总结。

5.2.1 决策树归纳

决策树归纳是从有类标号的训练元组中学习决策树。**决策树**(decision tree)是一种类似于流程图的树结构,其中,每个**内部结点**(internal node,即非树叶结点)表示在一个属性上的测试,每个分枝代表该测试的一个输出,而每个**树叶结点**(leaf node)(或终端结点)存放一个类标号。树的最顶层结点是**根结点**(root node)。叶结点用矩形表示,而内部结点和根结点用椭圆表示。有些决策树算法只产生二叉树,而另一些决策树算法可能产生非二叉的树。例如,在图5-3中,在根结点处,使用体温这个属性把冷血脊椎动物和恒温脊椎动物区别开来。因为所有的冷血脊椎动物都是非哺乳动物,所以用一个类标号为非哺乳动物的叶结点作为根结点的右孩子。如果脊椎动物的体温是恒温的,则接下来用胎生这个属性区分哺乳动物与其他恒温动物。

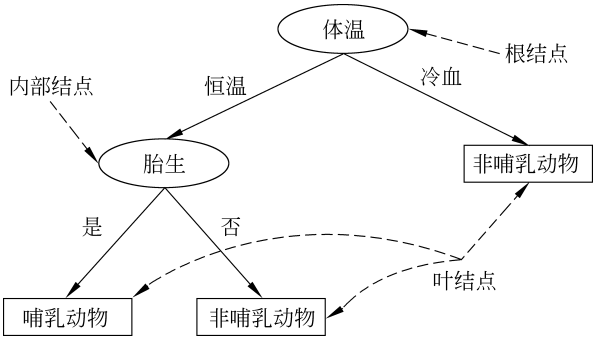

图5-3 哺乳动物分类问题的决策树

如何使用决策树分类？给定一个类标号未知的元组 X，在决策树上测试该元组的属性值。跟踪一条由根到叶结点的路径，该叶结点就存放着该元组的类预测。决策树容易转换成分类规则。

为什么决策树分类器如此流行？决策树分类器的构造不需要任何领域知识或参数设置，因此适合于探测式知识发现。决策树可以处理高维数据。获取的知识用树的形式表示是直观的，并且容易被人理解。决策树归纳的学习和分类步骤是简单和快速的。一般而言，决策树分类器具有很好的准确率。然而，成功的使用可能依赖手头的数据。决策树归纳算法已经成功地应用于许多领域的分类，如医学、制造和生产、金融分析、天文学和分子生物学。决策树是许多商业规则归纳系统的基础。

一旦构造了决策树，对检验记录进行分类就相当容易了。从树的根结点开始，将测试条件用于检验记录，根据测试结果选择适当的分枝。沿着该分枝或者到达另一个内部结点，使用新的测试条件，或者到达一个叶结点。到达叶结点之后，叶结点的类标号就被赋值给该检验记录。例如，图 5-4 显示应用决策树预测火烈鸟的类标号所经过的路径，路径终止于类标号为非哺乳动物的叶结点。

名字	体温	胎生	…	称号
火烈鸟	恒温	否	…	?

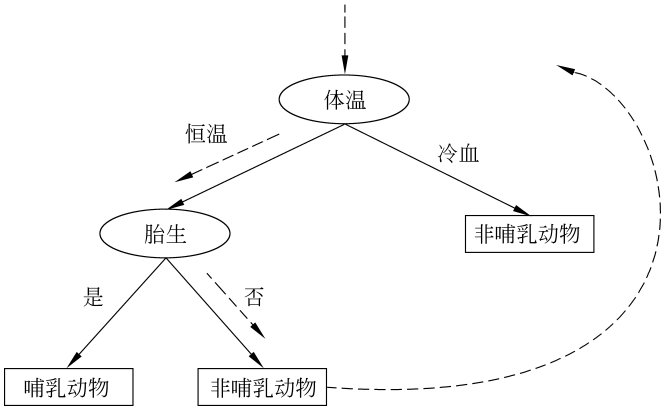

图 5-4　对一种未标记的脊椎动物分类(虚线表示在未标记的脊椎动物上使用各种属性测试条件的结果，该脊椎动物最终被指派到非哺乳动物类。)

5.2.2　如何建立决策树

原则上讲，对于给定的属性集，可以构造的决策树的数目达指数级。尽管某些决策树比其他决策树更准确，但是由于搜索空间是指数规模的，找出最佳决策树在计算上是不可行的。尽管如此，人们还是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最优决策树。这些算法通常都采用贪心策略(非回溯的)，在选择划分数据的属性时，采取一系列局部最优决策构造决策树，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、C4.5 和 CART。

在 Hunt 算法中，通常将训练记录相继划分成较纯的子集，以递归方式建立决策树。设 D_t 是与结点 t 相关联的训练记录集，而 $y=\{y_1,y_2,\cdots,y_c\}$ 是类标号，Hunt 算法的递归定义如下。

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

(2) 如果 D_t 中包含属于多个类的记录，则选择一个**属性测试条件**(attribute test condition)，将记录划分成较小的子集。对于测试条件的每个输出，创建一个子女结点，并根据测试结果将 D_t 中的记录发布到子女结点中。然后，对于每个子女结点，递归地调用该算法。

为了解释该算法如何执行，考虑如下问题：预测贷款申请者是会按时归还贷款，还是会拖欠贷款。对于这个问题，训练数据集可以通过考察以前贷款者的贷款记录来构造。在表 5-3 所示的例子中，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 5-3 训练数据集：预测拖欠银行贷款的贷款者

Tid	有房者(二元)	婚姻状况(分类)	年收入(连续)	拖欠贷款者(类)
1	是	单身	125k	否
2	否	已婚	100k	否
3	否	单身	70k	否
4	是	已婚	120k	否
5	否	离异	95k	是
6	否	已婚	60k	否
7	是	离异	220k	否
8	否	单身	85k	是
9	否	已婚	75k	否
10	否	单身	90k	是

该分类问题的初始决策树只有一个结点，类标号为“拖欠贷款者＝否”(见图 5-5(a))，意味着大多数贷款者都按时归还贷款。然而，该树需要进一步地细化，因为根结点包含两个类的记录。根据“有房者”测试条件，这些记录被划分为较小的子集，如图 5-5(b)所示。选取属性测试条件的理由稍后讨论，目前，假定此处这样选是划分数据的最优选择。接下来，对根结点的每个子女递归地调用 Hunt 算法。从表 5-3 给出的训练数据集可以看出，有房的贷款者都按时偿还了贷款，因此，根结点的左子女为叶结点，标记为“拖欠贷款者＝否”(见图 5-5(b))。对于右子女，需要继续递归调用 Hunt 算法，直到所有的记录都属于同一个类为止。每次递归调用所形成的决策树显示在图 5-5(c)和图 5-5(d)中。

如果属性值的每种组合都在训练数据中出现，并且每种组合都具有唯一的类标号，则 Hunt 算法是有效的，但是对于大多数实际情况，这些假设太苛刻，因此，需要附加的条件来处理以下情况。

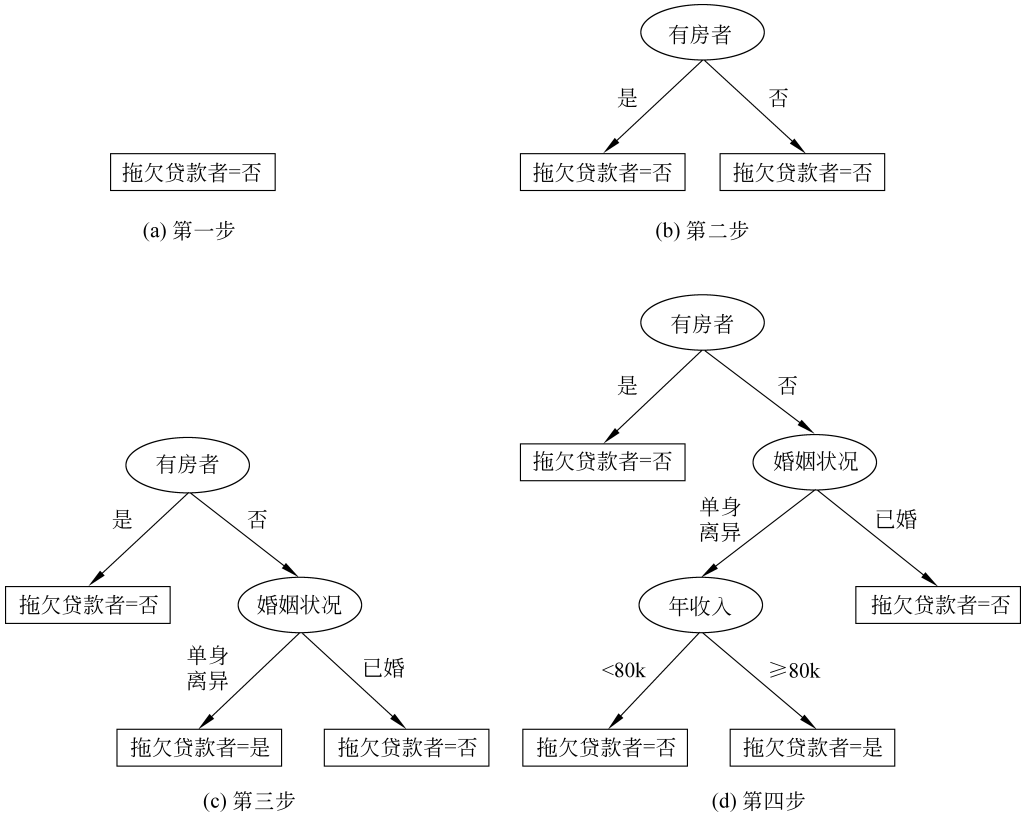

图 5-5　Hunt 算法构造决策树

(1) 算法的第二步(如图 5-5(b)所示)所创建的子女结点可能为空,即不存在与这些结点相关联的记录。如果没有一个训练记录包含与这样的结点相关联的属性值组合,这种情形就可能发生。这时,该结点成为叶结点,类标号为其父结点上训练记录中的多数类。

(2) 在第二步,如果与相关联的所有记录都具有相同的属性值(目标属性除外),则不可能进一步划分这些记录。在这种情况下,该结点为叶结点,其标号为与该结点相关联的训练记录中的多数类。

决策树归纳的学习算法必须解决下面两个问题。

(1) **如何分裂训练记录**? 树增长过程的每个递归步都必须选择一个属性测试条件,将记录划分成较小的子集。为了实现这个步骤,算法必须提供为不同类型的属性指定测试条件的方法,并且提供每个测试条件的客观度量。

(2) **如何停止分裂过程**? 需要有结束条件,以终止决策树的生长过程。一个可能的策略是分裂结点,直到所有的记录都属于同一个类,或者所有的记录都具有相同的属性值。尽管每个结束条件对于结束决策树归纳算法都是充分的,但是还可以使用其他的标准提前终止树的生长过程。

5.2.3 表示属性测试条件的方法

决策树归纳算法必须为不同类型的属性提供表示属性测试条件和其对应输出的方法。

(1) **二元属性**：二元属性的测试条件产生两个可能的输出，如图 5-6 所示。

(2) **标称属性**：由于标称属性有多个属性值，它的测试条件可以用两种方法表示，如图 5-7 所示。对于多路划分(见图 5-7(a))，其输出数取决于该属性不同属性值的个数。例如，如果属性婚姻状况有 3 个不同的属性值——单身、已婚、离异，则它的测试条件就会产生一个三路划分。另外，某些决策树算法(如 CART)只产生二元划分，他们考虑创建 k 个属性值的二元划分的所有 $2^{k-1}-1$ 种方法。图 5-7(b)显示了把婚姻状况的属性值划分为两个子集的 3 种不同的分组方法。

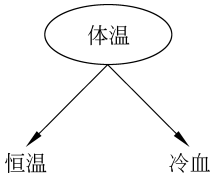

图 5-6 二元属性的测试条件

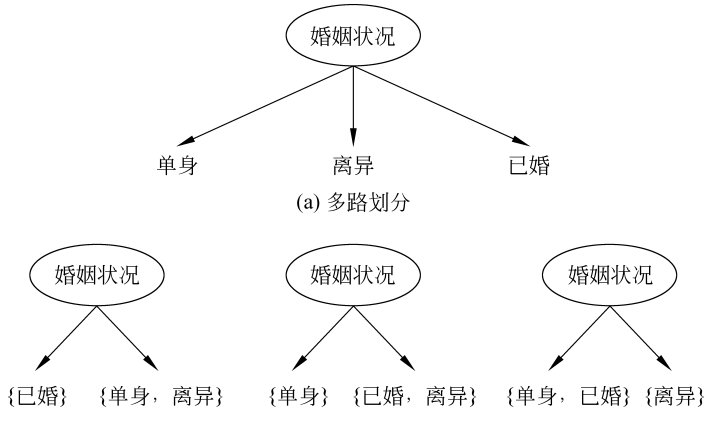

图 5-7 标称属性的测试条件

序值属性：序值属性也可以产生二元或多路划分，只要不违背序数属性值的有序性，就可以对属性值进行分组。图 5-8 显示了按照属性衬衣尺码划分训练记录的不同的方法。图 5-8(a)和图 5-8(b)中的分组保持了属性值间的序关系，而图 5-8(c)所示的分组则违反这一性质，因为它把小号和大号分为一组，把中号和加大号放在另一组。

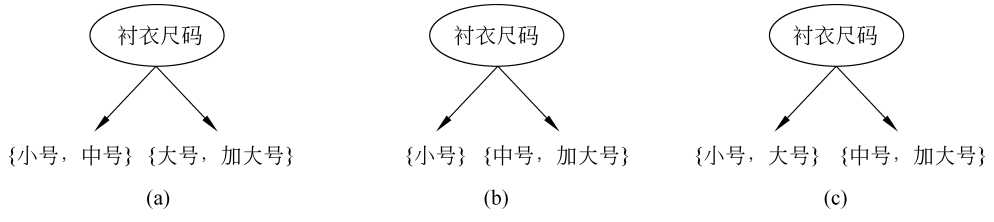

图 5-8 序值属性分组的不同方式

连续属性：对于连续属性来说，测试条件可以是具有二元输出的比较测试($A<v$)或($A\geqslant v$)，也可以是具有形如 $v_i\leqslant A<v_{i+1}(i=1,2,\cdots,n)$输出的范围查询。图 5-9 显示了这些方法的差别。对于二元划分，决策树算法必须考虑所有可能的划分点 v，并从中选择产生最佳划分的点。

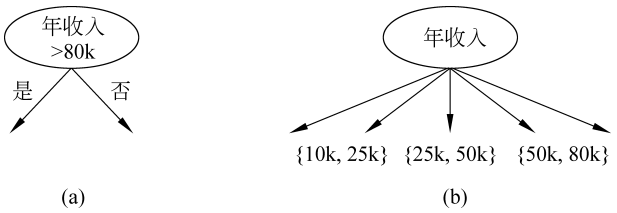

图 5-9　连续属性的测试条件

对于多路划分，算法必须考虑所有可能的连续值区间。可以采用离散化的策略，离散化之后，每个离散化区间赋予一个新的序数值，只要保持有序性，相邻的值还可以聚集成较宽的区间。

5.2.4　选择最佳划分的度量

有很多度量可以用来确定划分记录的最佳方法，这些度量用划分前和划分后记录的类分布定义。

设 $p(i|t)$表示给定结点 t 中属于类 i 的记录所占的比例，有时，省略结点 t，直接用 p_i 表示该比例。在二元分类问题中，任意结点的类分布都可以记作(p_0，p_1)，其中 $p_1=1-p_0$。例如，考虑图 5-10 中的测试条件，划分前的类分布是(0.5，0.5)，因为来自每个类的记录数相等。如果使用性别属性划分数据，则子女结点的类分布分别为(0.6，0.4)和(0.4，0.6)，虽然划分后两个类的分布不再平衡，但是子女结点仍然包含两个类的记录；按照第二个属性车型进行划分，将得到纯度更高的划分。

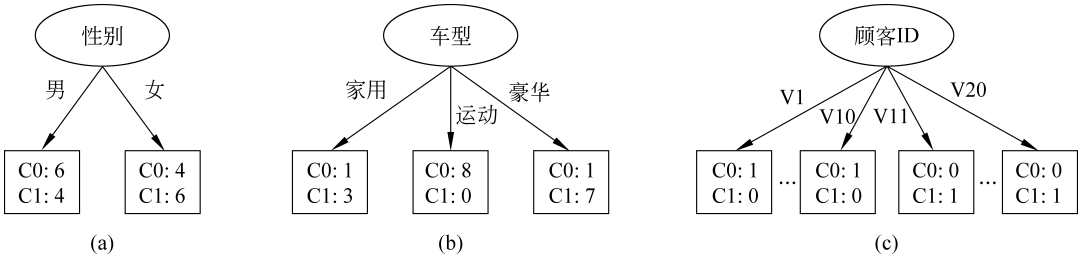

图 5-10　多路划分与二元划分

一般而言，随着决策树的建立，我们希望其分支结点包含的样本尽可能地属于同一个类别，即结点的“纯度”越来越高。通常根据划分后子女结点不纯性的程度作为选择最佳划分的度量。不纯的程度越低，类分布就越倾斜。例如，类分布为(0，1)的结点具有零不纯性，而均衡分布(0.5，0.5)的结点具有最高的不纯性。不纯性度量的例子包括：

$$\text{Entropy}(t)=-\sum_{i=0}^{c-1}p(i\mid t)\log_2 p(i\mid t) \tag{5-1}$$

$$\text{Gini}(t)=1-\sum_{i=0}^{c-1}[p(i\mid t)]^2 \tag{5-2}$$

$$\text{Classification error}(t)=1-\max_i[p(i\mid t)] \tag{5-3}$$

其中 c 是类的个数，并且在计算熵时，$O(\log_2 0)=0$。

图 5-11 显示了二元分类问题不纯性度量值的比较，p 表示属于其中一个类的记录所占的比例。从图中可以看出，3 种方法都在类分布均衡时（当 $p=0.5$ 时）达到最大值，而当所有记录都属于同一个类时（p 等于 1 或 0）达到最小值。下面给出 3 种不纯性度量方法的计算实例。

结点 N_1	计数
类=0	0
类=1	6

Gini=$1-(0/6)^2-(6/6)^2=0$

Entropy=$-(0/6)\log_2(0/6)-(6/6)\log_2(6/6)=0$

Error=$1-\max[0/6,6/6]=0$

结点 N_2	计数
类=0	1
类=1	5

Gini=$1-(1/6)^2-(5/6)^2=0.278$

Entropy=$-(1/6)\log_2(1/6)-(5/6)\log_2(5/6)=0.650$

Error=$1-\max[1/6,5/6]=0.167$

结点 N_3	计数
类=0	3
类=1	3

Gini=$1-(3/6)^2-(3/6)^2=0.5$

Entropy=$-(3/6)\log_2(3/6)-(3/6)\log_2(3/6)=1$

Error=$1-\max[3/6,3/6]=0.5$

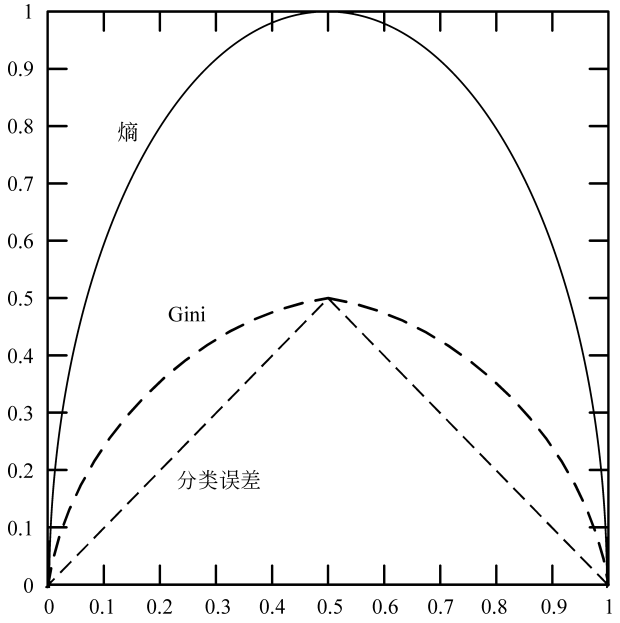

图 5-11 二元分类问题不纯性度量值的比较