第 2 章

分类

本章介绍分类的基本概念、解决分类的一般方法以及如何处理过拟合问题，重点详解决策树及其工作原理、决策树归纳法和处理拟合方法；贝叶斯决策及朴素贝叶斯分类器，同时讲解支持向量机中的最大边缘超平面、线性 SVM、非 SVM、核函数的基本内容，并列举分类在实际场景中的应用实例。

分类问题是一个普遍存在的问题，其应用具有普遍性。分类反映同类事物共同性质的特征和不同事物之间的差异型特征，如医学中根据核磁共振扫描的结果区分肿瘤是恶性还是良性等。本章将对分类的相关知识，如决策树、分类器、贝叶斯、支持向量机等内容进行讲解和研究。

2.1 分类概述

2.1.1 分类的基本概念

分类，是一种重要的数据分析形式。根据重要数据类的特征向量值及其他约束条件，可以将数据对象划分为不同的类型，通过进一步的分析挖掘事物的本质，建立分类函数或分类模型。分类的主要用途是“预测”，基于已知样本预测新样本的所属类型。

分类任务就是通过学习得到一个目标函数（分类模型）f_x，把每个属性集 x 映射到一个预先定义的类标号 y。分类模型可用于描述性建模和预测性建模。描述性建模可作为解释性工具，用于区分不同类中的对象；预测性建模可用于预测未知记录的类别[1]。

2.1.2 解决分类问题的一般方法

分类法是一种根据输入数据集建立分类模型的系统方法，包括决策树分类法、基于

规则的分类法、支持向量机分类法、朴素贝叶斯分类法、神经网络等分类法。另外，还有用于组合单一分类方法的集成学习算法，如 Bagging 和 Boosting 等。这些分类法使用分类算法确定分类模型，此模型能很好地拟合输入数据中类标号和属性集之间的联系。分类算法得到的分类模型不仅能拟合输入数据，同时能正确预测未知样本的类标号。

分类算法是解决分类问题的方法，是数据挖掘、机器学习和模式识别中一个重要的研究领域。解决分类问题的一般方法如下。

第一步，建立一个模型。这需要有一个训练样本数据集作为预先的数据集或概念集，通过分析属性/特征描述等构成的样本（也可以是实体等）建立模型，如图 2-1 所示。

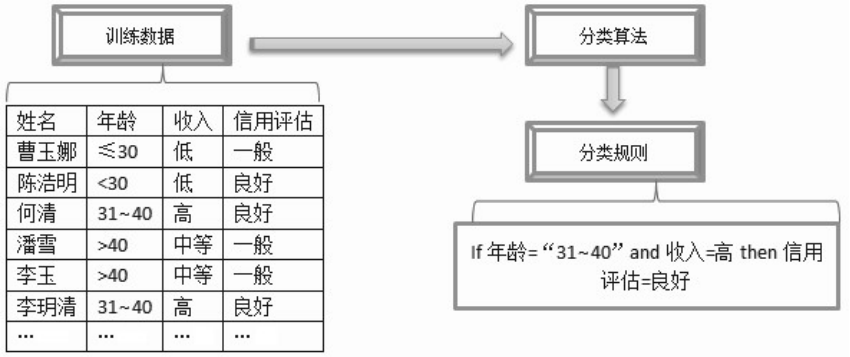

图 2-1　建立一个模型

用于建立模型的训练数据集由一组数据库记录或元组构成，而每个元组是一个由关键字段值（属性或特征）组成的特征向量。

每个训练样本都有一个预先定义的类别标记，它由一个被称为类标签的属性确定。可表示为$\{X_1,\cdots,X_n,C\}$；其中 X_n 表示字段值，C 表示类别。因样本数据的类别标记是已知，从训练样本集中提取出分类规则，用于对其他标号未知对象进行类标识。因此，分类又被称为有监督的学习。

第二步，应用所建立的模型对测试数据进行分类，如图 2-2 所示。

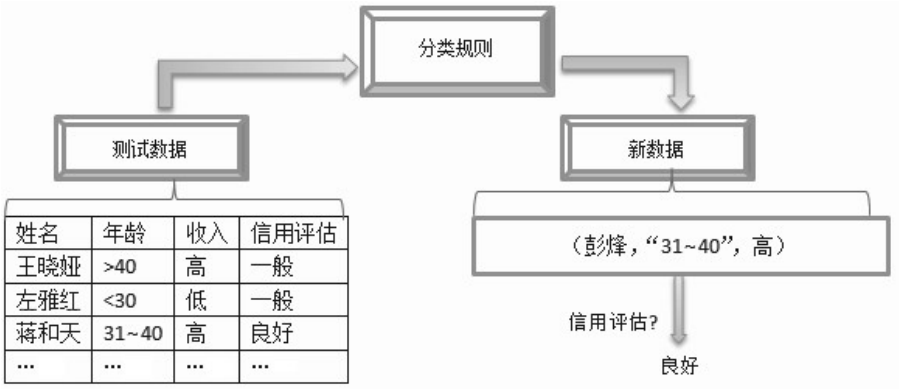

图 2-2　应用模型的分类

分类模型的性能根据模型正确和错误预测的检验记录计数进行评估。分类模型的性能可以用准确率和错误率来表示。例如：

$$准确率=\frac{正确预测数}{预测总数}；$$

$$错误率=\frac{错误预测数}{预测总数}。$$

2.1.3 分类模型的过拟合

分类模型的误差可分两类：训练误差和泛化误差。其中，训练误差又称再代入误差，是模型在训练数据集上的错误分类样本比例；泛化误差是模型在未知数据集的期望误差。所谓模型过拟合，是指模型的训练数据拟合度过高，其泛化误差可能比具有较高训练误差的模型高。具备较高训练数据拟合度的同时，能对未知样本数据准确分类的分类模型就是一个好模型。模型过拟合通常分为两种情况：噪声导致的过拟合、缺乏代表性样本导致的过拟合。

过拟合就是模型训练过程中过度拟合训练集，将训练样本中的噪声（错误的样本）学习进去，使得训练误差不断降低和模型复杂度不断提高，最终导致泛化误差升高的一种现象。在分类算法尤其是决策树中容易出现过拟合的问题，通过以下途径可以避免过度拟合。

（1）使用大量的数据。导致过拟合的根本原因是训练集和测试集的特征存在较大差异，导致原本完美拟合的模型无法对测试集产生较好的效果；通过使用大量的数据集，可能会增加训练集和测试集的特征相似性，这样会使模型的泛化性能较好。

（2）降维。通过减少维度选择或转换的方式，降低参与分类模型的维度数量，能有效地防止原有数据集中的“噪声”对模型的影响，从而达到避免过拟合的目的。

（3）正则化。正则化通过定义不同特征的参数来保证每个特征有一定的效用，不会使某一特征特别重要。

（4）使用组合方法。例如，随机森林、adaboost 不容易产生过拟合的问题。

2.2 决策树

决策树是一种简单而广泛使用的分类技术。基于决策树的分类方法也是最为典型的分类方法，是从实例集中构造决策树，再根据训练子集形成决策树[2]。

2.2.1 决策树的工作原理及构建

1. 决策树的工作原理

决策树的工作原理：通过提出一系列精心构思的关于检验记录属性的问题，解决分类问题。当一个问题得到答案，后续问题就随之而来，直到得到记录的类标号。这一系列的问题及可能的答案就构成决策树的形式。决策树是一种由节点和有向边构成的层次结构。分类问题的决策树，树中包含以下 3 种节点。

- 根节点：就是树的最顶端，最开始的那个节点。
- 内部节点：就是树中间的那些节点。
- 叶节点：就是树最底部的节点，也就是决策结果。

例如，对脊椎动物的分类只考虑两个类别：哺乳类动物和非哺乳类动物。如图 2-3 所示，假设发现一个新物种，怎么判断是哺乳类动物还是非哺乳类动物呢？此时，提出一系列有关物种特征的问题：新物种是冷血动物还是恒温动物？若是冷血，则它不是哺乳动物，反之可能是某种鸟或某种哺乳动物；若是恒温的，新物种是由雌性胎生繁殖吗？答案若为是，则可以肯定是哺乳动物。

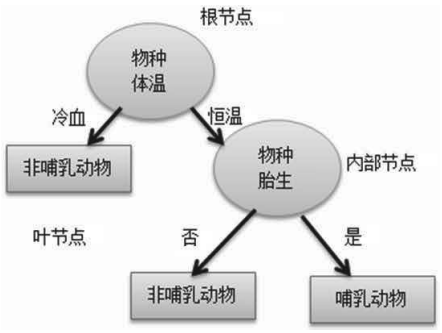

图 2-3　分类问题决策

决策树形成后，对检验记录进行分类就容易了。从树的根节点开始，将测试条件用于检验记录，根据测试结果选择适当的分支。沿着该分支或者到达另一个内部节点，使用新的测试条件，或者到达一个叶节点。到达叶节点之后，叶节点的类称号就被赋值给该检验记录了。

2．如何建立决策树

理论上在给定的属性集中，构造决策树的数目为指数级，尽管最优决策树会比其他决策树更准确，因搜索空间是指数规模的，找出最优决策树在计算上基本很难完成。因此需要开发一些有效的算法，在合理的时间内建立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，在选择划分数据的属性时，采取一系列局部最优决策建立决策树，Hunt 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。它们将分类领域从类别属性扩展到数值型属性。

2.2.2　决策树归纳算法

1．算法原理

算法 2.1 给出了称作 TreeGrowth 的决策树归纳算法的框架。该算法的输入是训练记录集 E 和属性集 F。算法递归地选择最优的属性来划分数据（步骤 7），并扩展树的叶节点（步骤 11 和步骤 12），直到满足结束条件（步骤 1）。

（1）函数 createdNode()为决策树建立新节点。决策树的节点可以是一个测试条件，记作 node.test_cond，也可以是一个类标号，记作 node.label。

（2）函数 find_best_split()确定应当选择哪个属性作为划分训练记录的测试条件。

（3）函数 Classify()为叶节点确定类标号。

（4）函数 stopping_cond()通过检查是否所有的记录都属于同一个类，或者都具有相同的属性值，决定是否终止决策树的增长。终止递归函数的另一种方法是，检测记录是否小于某个最小阈值。

算法 2.1　决策树归纳算法的框架

```
    TreeGrowth(E,F)
1: if stopping_cond(E,F)=true then
2:     leaf=createNode()
3:     leaf.label=Classify(E)
```

```
4:     return leaf
5: else
6:     root=createNode()
7:     root.test_cond=find_best_split(E,F)
8:     令 V={v|v 是 root.test_cond 的一个可能的输出}
9:   for  每个 vϵV do
10:     Ev={e|root.test_cond(e)=v  并且 eϵE}
11:     child=TreeGrowth(Ev,F)
12:     将 child 作为 root 的派生节点添加到树中，并将边（root→child）标记为 v
13:   end for
14: end if
15: return root
```

在建立决策树的过程中，容易出现决策树太大的现象，即过拟合现象，这就需要对决策树剪枝，以减小决策树的规模。修剪初始决策树的分支有助于提高决策树的泛化能力。

2．决策树归纳的学习算法须解决的两个问题

（1）训练记录的分裂。决策树增长过程中每步都要选择一个属性测试条件，将记录划分成较小的子集。算法则是为不同类型的属性指定测试条件提供方法，并且提供评估每种测试条件的客观度量。

（2）停止分裂过程。任何决策树都要有结束条件，以终止决策树的无限生长过程。一个可能的策略是分裂节点，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。尽管两个结束条件对于结束决策树归纳算法都是充分的，也可以使用其他算法合理地终止决策树的生长过程。

3．决策树归纳的特点

（1）决策树归纳无须假设类和其他属性服从某一概率分布，即是一种构建分类模型的非参数方法。

（2）找到最佳的决策树，即决策树获得的不是全局最优，而是每个节点的局部最优决策。

（3）建立决策树后，未知样本分类很快，而已开发构建的决策树技术的计算成本不高，即使训练集很大，也能快速建立模型。

（4）决策树相对其他分类算法更简便。特别是小型决策树的准确率较高。

（5）决策树算法对于噪声干扰有较强的抗干扰性，在运用此算法时注意避免过拟合后抗干扰性更强。

（6）冗余属性不会对决策树的准确率造成不利的影响。

（7）大多数决策树算法都采用“自顶向下”的递归划分方法，因此沿着决策树向下，记录会越来越少。解决该问题的一种可行的方法是，当样本数小于某个特定阈值时停止分裂。

2.2.3　处理决策树中的过拟合

下面介绍两种决策树归纳上避免过拟合的策略。

（1）先剪枝（提前终止）：决策树增长算法在产生完全拟合整个训练数据集的完全增长的决策树之前就停止决策树的生长，这就需要采用更具限制性的结束条件。例如，当估计的泛化误差的改进低于某个确定的阈值时，就停止扩展叶节点，其优点在于避免产生过拟合训练数据的过于复杂的子树。提前终止过程中，选取正确阈值的难度很大。若阈值太高，将产生拟合不足的模型；若阈值太低，就不能充分地解决过拟合的问题。

（2）后剪枝（过程修剪）：初始决策树按照最大规模生长，再剪枝。按照由下向上的方式修剪完全增长的决策树。修剪有两种做法：第一种，用新的叶节点替换子树，该叶节点的类标号由子树下记录中的多数类确定；第二种，用子树中最常使用的分支代替子树。当模型不能再改进时终止剪枝步骤。

与先剪枝相比，后剪枝能获得更好的结果，后剪枝是根据完全增长的决策树做出的剪枝决策，先剪枝则可能过早终止决策树的生长。当然，运用后剪枝时会浪费之前完全增长决策树的部分计算。

2.3　贝叶斯决策与分类器

2.2 节介绍了决策树归纳这种简单有效的分类技术，本节将讲解构建分类模型的其他技术——最简单的基于规则的分类器。

2.3.1　规则分类器

基于规则的分类器是使用一组“if…then…”规则来对记录进行分类的技术。表 2-1 所列的例子中给出脊椎动物分类问题基于规则的分类器产生的一个模型。此模型的规则用析取范式 $R=(r_1 \vee r_2 \vee \cdots \vee r_k)$表示，其中 R 称作规则集，而 r_i 是分类规则或析取项。

表 2-1　脊椎动物分类问题的规则集举例

r_1：（胎生=否）∧（飞行动物=是）→鸟类
r_2：（胎生=否）∧（水生动物=是）→鱼类
r_3：（胎生=是）∧（体温=恒温）→哺乳动物
r_4：（胎生=否）∧（飞行动物=否）→爬行类
r_5：（水生动物=半）→两栖类

每一个分类规则可以表示为

$$r_i:(\text{条件}_i) \rightarrow y_i$$

规则左边称为规则前件或前提。它是属性测试的合取：

$$\text{条件}_i=(A_1 \text{ op } v_1) \wedge (A_2 \text{ op } v_2) \cdots \wedge (A_k \text{ op } v_k)$$

其中，(A_j, v_j)是属性名称和对应的值，op 是比较运算符$\{=,\neq,<,>,\leqslant,\geqslant\}$。每一属性测

试(A_j,op v_j)称为一个合取项。

规则右边称为规则后件，包含预测类 y_i。

如果规则 r 的前件与记录 x 的属性匹配，则称 r 覆盖 x。当 r 覆盖给定的记录时，称 r 被触发。两种脊椎动物——鹦鹉和棕熊的属性如表 2-2 所示。

表 2-2 鹦鹉与棕熊

名　称	体　温	表皮覆盖	胎　生	飞行动物	有　腿	冬　眠
鹦鹉	恒温	羽毛	否	是	是	否
棕熊	恒温	软毛	是	否	是	是

r_1 覆盖第一种脊椎动物，因为鹦鹉的属性满足它的前件。r_1 不覆盖第二种脊椎动物。因为棕熊是胎生的且不能飞，不符合 r_1 的前件。

基于规则的分类器产生的规则集有以下两个重要性质。

（1）互斥规则。如果规则集中不存在两条规则被同一条记录触发的情况，则称规则集中的规则是互斥的。

（2）穷举规则。如果对属性值的任一组合，规则集中都存在一条规则可以覆盖，则称规则集具有穷举覆盖。它确保每一条记录都至少被规则集里的一条规则覆盖。

2.3.2 贝叶斯定理在分类中的应用

本节介绍一种对属性集和类变量的概率关系建模的方法，例如，想通过一个人的饮食和锻炼的频率预测其是否有患心脏病的风险。

这里首先介绍贝叶斯定理，它是一种将类的先验知识和从数据中收集的新证据相结合的统计原理。

1．贝叶斯定理

假设 X、Y 是一对随机变量，联合概率 $P(X=x, Y=y)$是指 X 取值 x 且 Y 取值 y 的概率，条件概率是指一随机变量在另一随机变量取值已知的情况下取某一特定值的概率。例如，条件概率 $P(Y=y|X=x)$是指在变量 X 取值 x 的情况下，变量 Y 取值 y 的概率。

2．贝叶斯定理在分类中的应用

先从统计学的角度对分类问题进行形式化。设 X 表示属性集，Y 表示类变量。如果类变量和属性之间的关系不确定，那么可以将 X 和 Y 看成随机变量，用 $P(Y|X)$以概率的方式捕捉二者之间的关系，这个条件概率又称 Y 的后验概率，对应 $P(Y)$称为 Y 的先验概率。

在训练阶段，要根据从训练数据中收集的信息，对 X 和 Y 的每一种组合学习后验概率 $P(Y|X)$。知道这些概率后，通过找出使后验概率 $P(Y|X)$最大的类 Y 可以对测试记录 X 进行分类。例如用这种方法解决任务：预测一个贷款者是否会拖欠贷款，训练集的各项属性如表 2-3 所示。

表 2-3　训练集

序　号	二元变量	分类变量	连续变量	类　变　量
	有　房	婚姻状况	年　收　入	拖欠贷款
1	是	已婚	135 k	否
2	否	已婚	100 k	否
3	否	单身	70 k	否
4	是	已婚	120 k	否
5	否	离异	95 k	是
6	否	已婚	60 k	否
7	是	离异	225 k	否

假设给定一组测试记录有如下属性集：X=(有房=否∧婚姻状况=已婚∧年收入=120k)。要分类该记录，需要利用训练数据中的可用信息计算后验概率 $P(\text{Yes}|X)$和$P(\text{No}|X)$。如果 $P(\text{Yes}|X)>P(\text{No}|X)$，那么记录分类为 Yes，反之，分类为 No。

准确估计类标号和属性值的每一种可能组合的后验概率非常困难，因为即便属性数目不是很大，仍然需要很大的训练集。此时，贝叶斯定理可以用先验概率 $P(Y)$、类条件概率 $P(X|Y)$和证据 $P(X)$来表示后验概率：

$$P(Y|X)=\frac{P(X|Y)P(Y)}{P(X)} \tag{2-1}$$

在比较不同 Y 值的后验概率时，分母 $P(X)$总是常数，可以忽略。先验概率 $P(Y)$可以通过计算训练集中属于每个类的训练记录所占的比例很容易地估计。对类条件概率 $P(X|Y)$的估计，可以用朴素贝叶斯分类器和贝叶斯信念网络两种贝叶斯分类方法实现。

2.3.3　朴素贝叶斯在分类中的应用

1．条件独立

在研究朴素贝叶斯分类法如何工作之前，先介绍条件独立的概念。

引例：研究一个人的手臂长短与其阅读能力之间的关系。可能发现，手臂较长的人阅读能力较强，而这种关系可能就是年龄。小孩的手臂比成人的手臂短，同时也不具备成人的阅读能力。若年龄一定，此时手臂长度与阅读能力之间的关系消失。从而可以得出，在年龄一定时，手臂长短和阅读能力两组条件独立。

设 X、Y 和 Z 表示 3 个随机变量的集合。给定 Z、X 条件独立于 Y，若

$$P(X|Y,Z)=P(X|Z)$$

则 X和 Y 之间的条件独立可写为

$$P(X,Y|Z)=P(X|Z)\times P(Y|Z) \tag{2-2}$$

2．朴素贝叶斯分类器

分类测试记录时，朴素贝叶斯分类器对每个类 Y 计算后验概率：

$$P(Y|X)=\frac{P(Y)\prod_{i=1}^{d}P(x_i|Y)}{P(X)} \tag{2-3}$$

朴素贝叶斯分类法使用以下两种方法估计连续属性的类条件概率。

（1）将每一个连续的属性离散化，然后用相应的离散区间替换连续属性值。

（2）假设连续变量服从某种概率记录，然后使用训练数据估计分布的参数。

方法（2）更实用，因为它不需要很大的训练集就能获得较好的概率估计。

3．朴素贝叶斯分类器的特征

（1）在面对孤立的噪声点时，朴素贝叶斯分类器的性能受到的影响不大。

（2）面对无关属性时，朴素贝叶斯分类器的性能受到的影响同样不大。

（3）相关属性会降低朴素贝叶斯分类器的性能。

2.4 支持向量机

要了解 SVM，首先要了解最大边缘超平面的概念以及选择它的基本原理。其次，了解在线性可分的数据上怎样训练一个线性的 SVM，从而明确地找到这种最大边缘超平面。最后，了解如何将 SVM 方法扩展到非线性可分的数据上。

2.4.1 最大边缘超平面

支持向量机（support vector machine，SVM）于 1995 年由 Cortes 和 Vapnik 首先提出。SVM 已是一种倍受关注的分类技术，在小样本、非线性、高维模式识别中具备特有的优势，并能够推广应用到函数拟合等其他机器学习问题中。SVM 已成为最主要的模式识别方法之一，它可以在高维空间构造良好的预测模型，在 OCR、语言识别、图像识别等领域广泛应用。SVM 以扎实的统计学理论为基础，并在许多实际应用（如手写数字的识别、文本分类等）中展示了大有可为的实践效果。此外，SVM 可以很好地应用于高维数据中，避免了维灾难问题。这种方法具有一个独特的特点，它使用训练实例的一个子集来表示决策边界，该子集称为支持向量（Support Vector）。

一个数据集包含两个不同类的样本，分别用小黑方块和小圆圈表示。数据集是线性可分的，即能找到一个超平面，使得所有小黑方块位于这个超平面的一侧，所有小圆圈在它的另一侧。如图 2-4 所示，可以看到这种超平面存在无穷多个。通过检验样本的运行效果，分类器要从这些超平面中选一个作为它的决策边界。

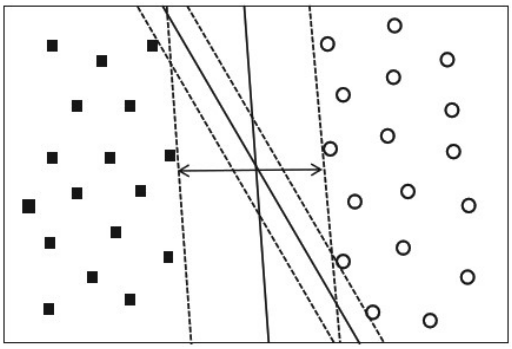

图 2-4 一个线性可分数据集可能决策边界

支持向量机方法是建立在统计学理论的 VC 维理论和结构风险（结构风险=经验风险+置信风险）最小原理基础上的。用有限样本信息在模型的复杂性和学习能力（无错误地识别任意样本的能力）之间寻求最佳折中，以期获得最好的推广能力（或称泛化能力）。