第5章 机器学习

我们正置身于一个科技浪潮汹涌澎湃的时代，智能服务如春风化雨般融入生活的每一处角落，智能语音助手随时响应我们的需求，智能推荐系统精准推送心仪的商品，自动驾驶汽车在道路上稳健行驶，智能交通管理系统让城市拥堵状况得到显著缓解。这些令人瞩目的突破性进展已成为推动社会不断前进的核心动力，而这些辉煌成就的背后，机器学习技术无疑是最为坚实的支撑。机器学习赋予计算机模拟人类学习行为的能力，通过精心设计的算法，使得机器能够像人类一样，从海量的数据中自动探寻规律、汲取知识，在数据的海洋中不断探索、成长，逐渐掌握应对复杂任务的本领。在这个过程中，机器学习极大地提升了智能系统的自主性和智能化水平，让系统不再依赖预先设定的规则，而是能够根据实际情况灵活应对。机器学习为各行各业带来了前所未有的变革机遇，无论是医疗、金融，还是教育、制造业，都在其影响下发生着翻天覆地的变化，深刻重塑着我们的生活和工作方式，引领我们迈向更加智能、便捷的未来。

5.1 机器学习概述

5.1.1 何为机器学习

机器学习英文为“Machine Learning”，顾名思义，是指机器像人一样进行学习，习得类似于人的或超越人的能力。在计算机领域，Machine 一般指计算机，故机器学习可以理解为让计算机进行学习的技术。

但是，计算机是没有生命的，如何像人一样学习呢？为了回答该问题，我们先来看看人是如何学习的。

人类学习通常是先根据经验归纳规律，再依据规律给出解决新问题的方法。其中，经验可以是多次经历某事的过程。与人类似，机器学习通常基于许多历史数据训练出一个计算机程序，再利用所训练的计算机程序对新的数据进行预测。在此，计算机程序也称为模型，而训练模型的历史数据则类似于人学习的经验，指导模型归纳学习规律，最终所学规律会使模型对某类任务的性能得到提升。人类学习与机器学习的对比如图 5-1 所示。

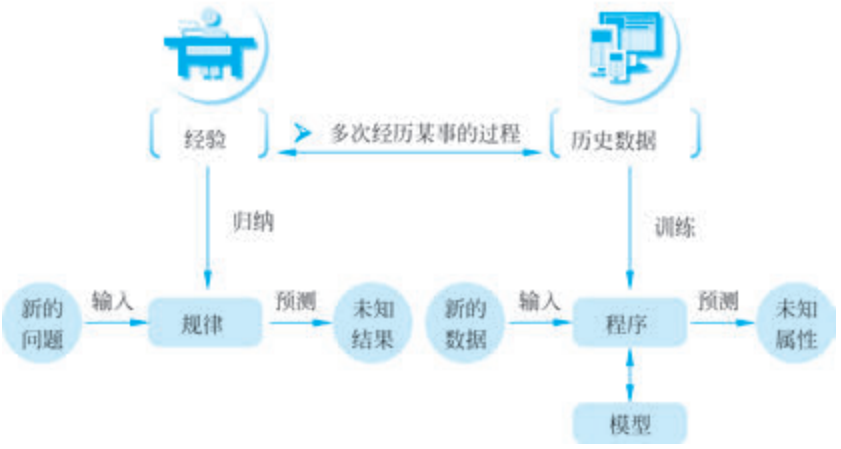


图 5-1 人类学习与机器学习的对比

为了给机器学习(图 5-2)下一个更正式的定义,我们引用全球机器学习之父、卡内基-梅隆大学计算机学院院长 Tom Mitchell 在《机器学习》一书中的定义:“对于某类任务(T)和性能度量(P),如果一个计算机程序在 T 上以 P 衡量的性能随着经验(E)而自我完善,那么称这个计算机程序在从 E 学习。”

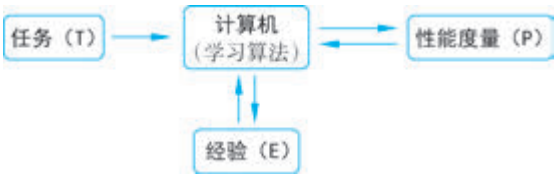


图 5-2 机器学习定义

任务(T):是这样的活动,计算机在其上学习以改进它的性能,如学习下棋、识别手写文字、自动驾驶。

经验(E):关于以往情况的记录,如和自己进行对弈、手写文字库、人类驾驶时录制的一系列图像和驾驶指令。

性能度量(P):对于反应或行动质量的度量,如对弈击败对手的百分比、分类的正确率、平均无差错形式的历程(差错由人裁定)。

通俗地说,机器学习就是研究计算机程序随着经验积累自动提高性能的学问。

5.1.2 为何用机器学习

机器像人一样学习,其意义何在呢?纵观人类发展史,人类制造和使用工具的原始动力就是为了提升人的生存能力和生活质量。机器学习利用计算机进行类似于人的学习,也是为了更好地服务于人。下面以两个简单的例子来进行说明。

首先,我们来看一个简单的问题:手写数字识别。

识别手写的数字是几,这对于人来说是极其容易的,那么为什么还要使用机器学习呢?这涉及简单工作的劳动量问题。比如,在没有机器学习以前,邮件上手写的邮编数字是由邮递员来识别的,邮递员根据识别结果对邮件进行分类运输。虽然识别手写邮编是一项很简单的任务,但是一个城市一天有几万封甚至几十上百万的邮件,需要邮递员耗费大量的时间来识别手写邮编数字。

有了机器学习技术,就可以让机器自动识别手写邮编数字,机器能不知疲倦地帮我们完成这项“单调乏味”的工作,进而帮助我们自动分拣邮件。更进一步,如果机器能掌握识别并理解文字的能力,就可以用机器来分拣快递了。

下面,再来看一个复杂一些的问题:用户喜好推荐。

当前,手机和移动互联网的广泛普及使得信息的传播速度极快。不同的地方都会发生不同的事情,不同的人也会关心不同的事情,例如,用户从事科技工作,他关心每天科技相关的发展动态;用户从事教育工作,他关心的是各种新的教学方法或教育政策。因此,如何根据用户喜好,给不同的用户推荐不同的文章有着非常重要的意义。现代化的城市动辄就有成百上千万人,每个人有着各自不同的喜好,而且喜好可能还会随时间发

生变化。比如，一名科技工作者改行从事教育工作，那么他的喜好就会发生变化。这种根据每个用户喜好推荐文章的工作由人来完成几乎是不可能的。

机器学习可以针对每个用户喜好提供精准推荐。比如，今日头条会收集每个用户搜索的关键词，或根据用户浏览的历史来推荐相关的文章，用户浏览某类文章越多，说明其对该类信息感兴趣，机器就可以给该用户推荐相关的文章。用户浏览的文章类别改变了，说明用户的喜好改变了。根据用户喜好推荐文章是非常复杂的一个问题，而机器学习使得该问题变得简单。

5.1.3 如何用机器学习

机器学习运用与人的学习运用类似：人的学习运用需要先总结规律，再利用规律解决新问题；机器学习运用需要先根据历史数据训练模型，再利用模型对新的数据进行预测。因此，机器学习运用主要分为训练和预测两个过程。

训练过程就是机器从历史数据中学习规律的过程，如根据坦克历史数据训练一个由炮管口径等特征来预测打击能力的模型。训练模型的历史数据称为样本集，通常由一个个样本组成，比如 Y 国已有所有坦克的信息作为样本集，每辆坦克的对应信息就是一个样本；所训练的模型可以从机器学习模型库中选择一种合适的模型，如图 5-3 所示。

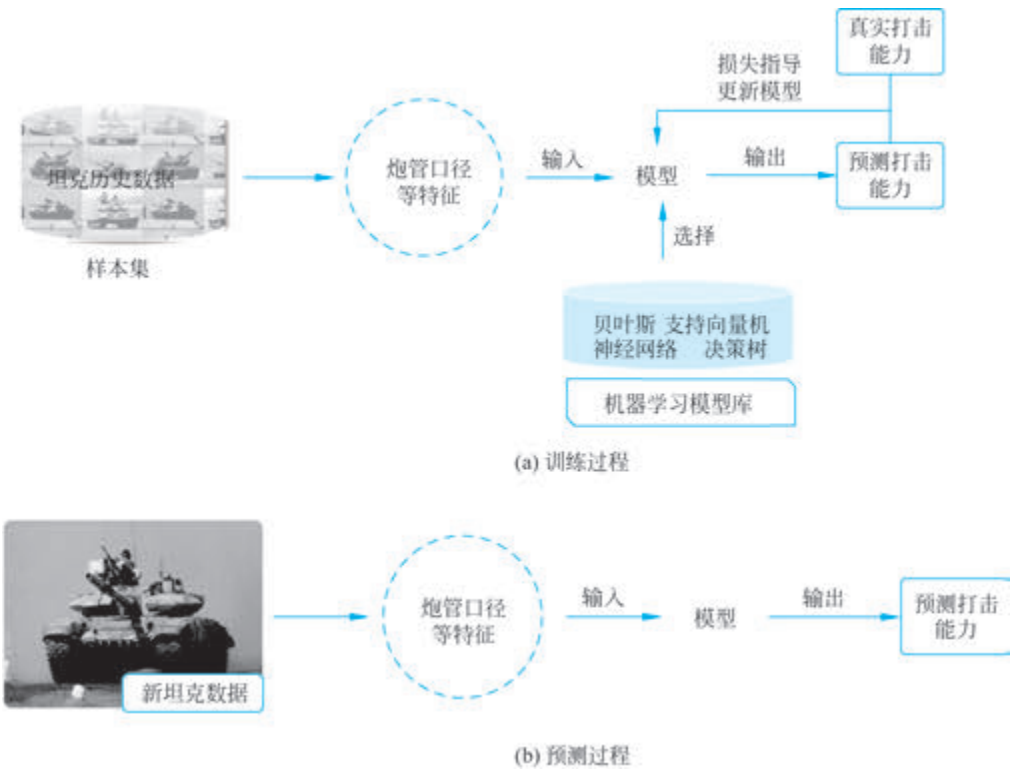


图 5-3 机器学习模型的训练与预测

预测过程是机器依据模型进行决策的过程，如Y国研制了一种新坦克，给定一辆新坦克的炮管口径等特征，模型即可预测输出该坦克的打击能力。

5.2 机器学习方法的分类

机器学习就是机器模仿人进行学习，所以搞清楚人的学习分类，就可以理解机器学习的分类。人的学习大致可以分为有人指导学习和无人指导学习。比如：我们从小学到大学的教育就是在老师的指导下进行的学习，这种有人指导的学习可以极大地提高学习的效率；而抗战时期，我军没有条件开设学校，很多指挥员都是在战争中学习战争，从战争中总结经验，这种学习就是无人指导的学习。

机器学习中，对应有数据标签、无数据标签和有部分标签的学习，可以分为有监督学习、无监督学习和半监督学习。其中，有监督学习是指有期望输出条件下的学习，其训练数据包括成对的输入与输出数据，模型训练好后，给定一个输入则模型会产生一个明确的预期输出；无监督学习是指无期望输出条件下的学习，其训练数据仅包括输入数据，没有输出数据，模型训练好后，通常会为输入数据生成潜在的规律，但输出是无法预期的；半监督学习是仅有一部分数据具有标签的学习方式，介于有监督学习和无监督学习之间。

此外，有些教材也将机器学习方法分为有监督学习、无监督学习和强化学习。强化学习作为智能体通过与环境交互学习最优策略的核心范式，在大模型时代通过与海量参数架构的深度融合，已成为突破预训练模型决策能力瓶颈、实现从语言生成到复杂任务规划跃迁的关键技术，尤其在机器人控制、多轮对话优化及动态环境适应等场景中展现出不可替代的价值。由于强化学习的重要性，第8章将对该学习范式进行单独重点介绍。

5.2.1 有监督学习

有监督学习很清楚自己想要什么，有着非常明确的期望输出。依据期望输出的数值类型，有监督学习主要包括回归和分类两大任务。其中，回归任务的期望输出是连续的数值，如预测房价、气温等；而分类任务的期望输出则是离散的数值，如预测动物的类别、人的性别等。

下面介绍房价预测回归任务（图5-4）和动物分类任务（图5-5）。

图5-4 房价预测回归任务

从直观上来说，房屋的价格与房子的面积总体上呈线性关系，即房价 y 等于面积 x 乘上一个系数 w，加上一个偏移量 b。当然，这不是绝对的，房价还受到地理位置、楼层等因素的影响，这些因素暂不考虑。因此，上述房价预测问题可以用一个线性模型进行建模，在给定大量房屋面积和房价数据的基础上，学习线性模型中的系数 w 和偏移量 b。学习时，预测输出与期望输出之间通常是有误差的，这个误差称为目标函数。目标函数可以指导模型学习，使得预测输出与期望输出误差不断减小。模型训练完，可以得到最佳的参数 w 和 b。

人类根据物种的差异，将自然界的动物分类为猫、狗、牛、羊等，每类动物对应可以表示为一个离散的数值，如猫编号 0、狗编号 1、牛编号 2、羊编号 3 等，如图 5-5 所示。对于一个新的动物，该任务预测输出的类别必须是上述 0、1、2、3 等离散编号中的一个，而不可以是一个没有意义的小数。

图 5-5　动物分类任务

分类任务与回归任务不同，其期望输出的数值类型是离散的，而非连续的。因此，其学习的目标函数通常也不同于回归任务。根据上述回归任务和分类任务可以发现，有监督机器学习为模型给出了期望输出，在期望输出的指导下模型可以进行高效的学习。

5.2.2　无监督学习

无监督学习，顾名思义，就是不受监督指导的学习。该学习方式没有期望输出进行指导，直接通过自我认知、自我归纳进行学习。无监督学习与有监督学习最大的区别是模型的训练数据不再提供期望输出信息，这也使得无监督学习通常以数据间的差异度或相似度来进行自我学习。聚类就是无监督学习的典型任务。

例如，一个从未接触过军事装备的人，给他一堆装甲车辆图像，虽然他叫不出装甲车辆的各种称谓，也无法区分轮式突击车、履带式坦克、轮式步战车、履带式步战车等。但是，根据装甲车的炮管粗细、是轮式还是履带式等特征，他可以将装甲车划分为多个不同的类簇，再为每个类簇编一个编号。当出现新的装甲车辆时，根据特性就可以为车辆编号归类。对于未知的事物，人类通过聚类可以揭示未知事物的类聚特性及规律，如图 5-6 所示。

聚类学习把彼此“相近”的特征聚在一个簇中，方便有针对性地研究类簇的共性特征。当然，还可以利用聚类发现远离各个类簇的孤立点，这种孤立点的检测也有着重要

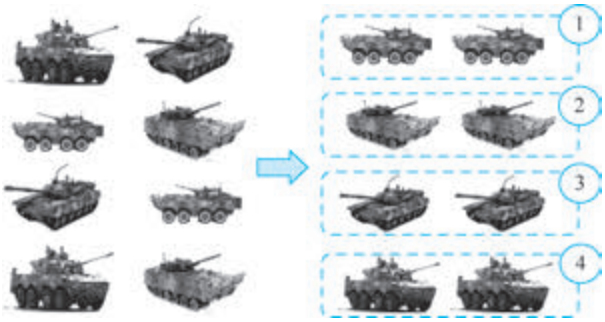


图 5-6　聚类的过程

意义。例如，一辆坦克绝大部分时间的工作状态都是正常的，故状态对应的特征描述点会呈现出类聚特性，而偶尔出现故障的状态，此时故障状态的特征描述点会显著地远离正常状态的特征描述点，通过聚类可以轻易发现这些点，实现对装备运行状态的异常检测。

5.2.3　半监督学习

半监督学习作为机器学习领域的一个重要分支，旨在利用少量标注数据和大量未标注数据进行模型训练，以提高模型的泛化能力。半监督学习可以看作监督学习和无监督学习的结合，它既能利用标签数据提高模型的准确性，也能通过无标签数据挖掘更多特征。

半监督学习的现实需求非常强烈，因为在现实应用中最容易收集到的往往是大量未标记的样本，而对这些样本打标签却是一项耗时耗力的工作。图像分类是半监督学习的典型任务。假设有一个包含 1000 个类别的图像数据集，每个类别均有 1000 个样本，但是其中只有 100 个样本有标注，其余均为未标注样本。此时，半监督学习首先使用这些少量的标注样本训练一个初步的分类器，并用这个分类器对未标注的数据进行预测，生成伪标签；随后将这些伪标签和原始的标注样本一起，用于训练新的分类器。通过多次迭代这个过程，分类器的性能会逐渐提高，如图 5-7 所示。

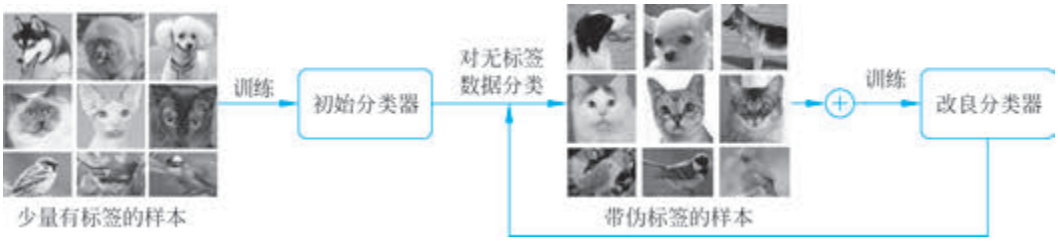


图 5-7　半监督学习的过程

5.3　分类方法

人对事物的区分主要依托于不同事物的属性差异来决定。根据人类流传的知识，人可以对已知事物很好地进行分类。比如，有人拍摄到一种未见过的新鸟，他会去鸟类百

科或百度等工具查询，对比其与不同鸟的羽毛、色泽、嘴巴、爪子等特征，如果发现这个鸟与某个已知类别的鸟非常相似，就会认定是该类别的鸟。这种简单的分辨类别的方法其实就体现了分类中的最近邻方法思想。

5.3.1 k 近邻方法

k 近邻(k-Nearest Neighbors，kNN)分类算法是一种基础且简单的机器学习算法，主要用于分类任务。该算法的核心思想是：若一个样本在特征空间中的 k 个最邻近的样本中的大多数属于某一个类别，则该样本也属于这个类别。具体来说，当有一个新的数据点需要分类时，算法会找到数据集中与该点最邻近的 k 个点，并统计这 k 个点中每个类别的频率。最终，新数据点被分配到出现频率最高的类别中。

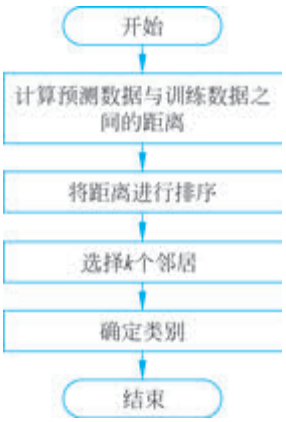


图 5-8 k 近邻分类算法流程图

k 近邻分类算法流程(图 5-8)如下：

(1) 计算预测数据与训练数据之间的距离。对于每个待分类的数据点，计算其与训练集中所有数据点的距离。常用的距离计算方法有欧几里得距离和曼哈顿距离等。

(2) 将距离进行排序。根据计算出的距离，将训练集中的数据点按距离从小到大排序。

(3) 选择 k 个邻居。选择距离最小的前 k 个数据点作为待分类样本的邻居。

(4) 确定类别。根据这 k 个邻居的类别，使用“多数表决法”来确定待分类样本的类别。即选择出现次数最多的类别作为预测结果。

1. 距离计算

k 近邻中距离是度量不同样本间差异的指标，不同的距离计算方法会表现出不同的差异。常用的就是欧几里得距离，即

$$d(x_i,x_j)=\sqrt{\sum_{k=1}^{n}(x_i^k-x_j^k)^2} \tag{5-1}$$

式中：x_i、x_j 为样本特征向量；n 为特征维度。

除了欧几里得距离，还包括闵可夫斯基距离、曼哈顿距离、切比雪夫距离等。

闵可夫斯基距离，即

$$d(x_i,x_j)=\left(\sum_{k=1}^{n}|x_i^k-x_j^k|^p\right)^{1/p} \tag{5-2}$$

闵可夫斯基距离是一种概括性描述距离：当 $p=2$ 时，闵可夫斯基距离就是欧几里得距离；当 $p=1$ 时，闵可夫斯基距离就是曼哈顿距离，即

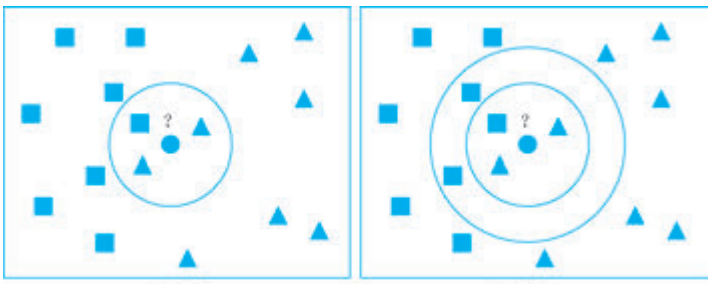
$$d(x_i,x_j)=\sum_{k=1}^{n}|x_i^k-x_j^k| \tag{5-3}$$

当 $p\to\infty$ 时，闵可夫斯基距离就是切比雪夫距离，即

$$d(x_i, x_j) = \lim_{p\to\infty}\left(\sum_{k=1}^{n} | x_i^k - x_j^k |^p\right)^{1/p} \tag{5-4}$$

2. k 的选择

k 值的大小对算法预测结果会有比较大的影响，如图 5-9 所示。



(a) k=3　　(b) k=5

图 5-9　$k=3$ 和 $k=5$ 的差异

当 $k=3$ 时，距离圆形样本最近的 3 个实例中（圆圈内），有两个是三角形（正类）、一个是正方形（负类），则该样本属于三角形（正类）；当 $k=5$ 时，则会判定圆形实例属于正方形类别。

一般来说，k 值太小会导致预测的标签比较容易受到样本的影响，容易过拟合（overfitting）；k 值太大会导致预测标签比较稳定，可能过平滑，容易欠拟合（underfitting）。实际应用中可以选择不同的 k 值，通过验证来决定 k 值大小。

k 值通常使用交叉验证来获取，将样本数据按照一定比例拆分出训练用的数据和验证用的数据，比如 6∶4 拆分出部分训练数据和验证数据，从选取一个较小的 k 值开始，不断增加 k 的值，然后计算验证集合的方差，最终找到一个比较合适的 k 值。

k 近邻方法不需要标签，使得 kNN 算法在处理分类问题时非常灵活和易于实现。当 $k=1$ 时，也称为最近邻分类方法。

5.1.3 节介绍了机器学习算法的训练过程和预测过程。kNN 算法的本质是训练过程中将所有训练样本的输入和输出标签都存储起来，测试过程中计算测试样本与每个训练样本的距离，选取与测试样本距离最近的前 k 个训练样本，然后对 k 个训练样本的标签进行投票，票数最多的类别即为测试样本所归类。可以发现，训练过程需要将所有的训练样本及其输出标签存储起来，当样本规模很大时，存在空间成本很大；测试过程中每个测试样本都需要与所有的训练样本进行比较，计算的运行时间成本很大。当样本规模较小时，k 近邻方法不失为一种有效的分类方法。

5.3.2　鸢尾花分类

接下来，我们在经典的鸢尾花卉（iris）数据集上利用 k 近邻实现分类，如图 5-10 所示。

iris 数据集每个样本 x 包含了花萼长度（sepal length）、花萼宽度（sepal width）、花瓣长度（petal length）、花瓣宽度（petal width）四个特征。样本标签 y 共有三类，分别是

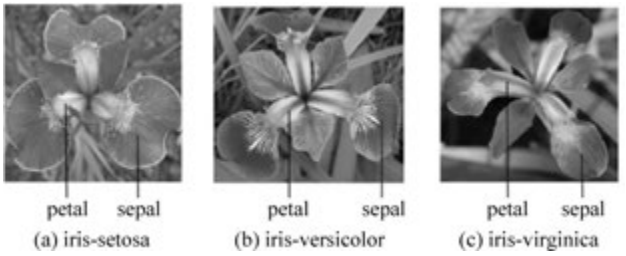


图 5-10　鸢尾花卉数据集的 3 个类别(setosa、versicolor 和 virginica)及花萼(sepal)和花瓣(petal)

setosa、versicolor 和 virginica。iris 数据集总共包含 150 个样本，每个类别由 50 个样本，整体构成一个 150 行 5 列的二维表，表 5-1 展示了 10 个样本。

表 5-1　iris 数据集的 setosa 类部分样本

序号	sepal length	sepal width	petal length	petal width	species
0	5.1	3.5	1.4	0.2	iris-setosa
1	4.9	3.0	1.4	0.2	iris-setosa
2	4.7	3.2	1.3	0.2	iris-setosa
3	4.6	3.1	1.5	0.2	iris-setosa
4	5.0	3.6	1.4	0.2	iris-setosa
5	5.4	3.9	1.7	0.4	iris-setosa
6	4.6	3.4	1.4	0.3	iris-setosa
7	5.0	3.4	1.5	0.2	iris-setosa
8	4.4	2.9	1.4	0.2	iris-setosa
9	4.9	3.1	1.5	0.1	iris-setosa

为了可视性，只选择 sepal length 和 petal length 两个特征，在二维平面上作图，如图 5-11 所示。


彩图 5-11

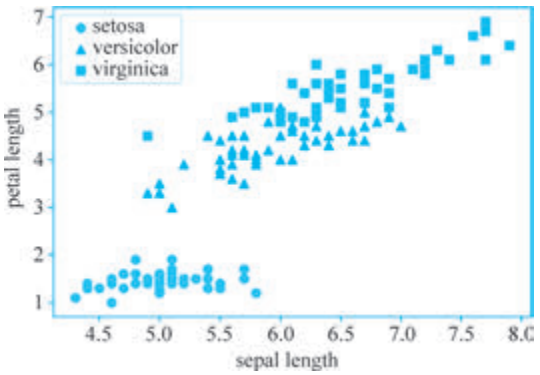


图 5-11　iris 数据集特征可视化

由图 5-11 可见，三个类别之间是有较明显区别的，各类别数据在图上呈现出明显的差异性和聚集性。setosa 类的花萼和花瓣长度在三个类别中整体是最小的，其聚集在图的左下方，virginica 类在三个类别之中平均的花萼和花瓣长度最长，versicolor 类的花萼和花瓣长度大小整体位于其他两种之间，在图像上整体居中。

接下来将每个类别的所有样本分成训练集(training set)、验证集(validation set)和测试集(testing set),各占所有样本的比例分别为60%、20%、20%。

在验证集上进行k-fold交叉验证。k-fold交叉验证是一种评估模型性能的常用方法,其将数据集均分为k个子集,依次将其中$k-1$个子集作为训练集,剩余1个子集作为测试集,重复k次训练和测试,最终取k次评估指标的平均值作为模型性能的估计。该方法能更充分利用数据,减少数据划分导致的评估偏差,提升结果可靠性。根据验证结果(图5-12),选择最佳的k值。

从图5-12可知,随着k的增大,在验证集上的准确率逐渐下降,当$k=3$时,验证集的准确率最高。此例中,由于总体样本数据量不够多,验证结果并不明显。但是,使用k-fold交叉验证来选择最佳k值是最常用的方法之一。

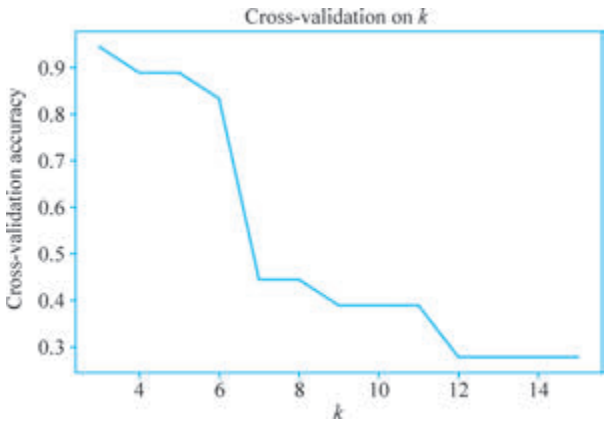


图5-12 验证集交叉验证结果可视化

选择完合适的k值之后,就可以对测试集进行预测。最终结果显示,测试集预测准确率为100%。最后把预测结果绘图表示。仍然只选择sepal length和petal length两个特征,在二维平面上作图,如图5-13所示。从图中可以看到,当$k=3$时,在此数据集上预测结果最佳。

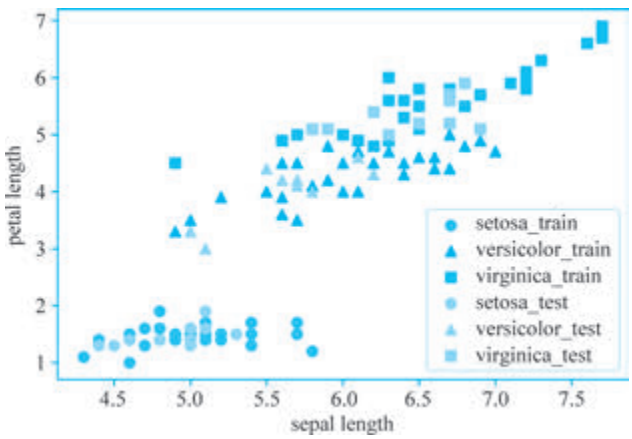


图5-13 iris数据集预测结果可视化

彩图5-13

k 近邻算法是一种最简单最直观的分类算法，它的训练过程保留了全部样本的所有特征，把所有信息都记下来，没有经过处理和提取。而其他机器学习算法包括神经网络则是在训练过程中提取最重要、最有代表性的特征。在这一点上，kNN 算法还非常不够“智能”。但是，kNN 算法作为机器学习的基础算法，还是值得我们学习。

5.3.3 决策树方法

决策树是一种经典的预测模型，它反映的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，每个分叉路径表示某个可能的属性值，每个叶节点对应从根节点到该叶节点所经历的路径所表示的对象的值。

决策树构造就是每次选择一个好的特征以及分裂点作为当前节点的分类条件。那么，决策树的构造过程中，应该先后选择哪些属性作为分类属性？即判定哪个属性是当前最佳的分类属性的标准或方法是什么？

定义 5-1 熵 $E(S)$。

为了精确地定义信息增益，先定义信息论中广泛使用的一个度量标准——熵，熵刻画了任意数据集的纯度。给定包含关于某个目标概念的正负样例的数据集 S，那么 S 相对这个布尔型分类的熵为

$$E(S)=-p^{+}\log_2 p^{+}-p^{-}\log_2 p^{-} \tag{5-5}$$

式中：p^{+} 代表正样例；p^{-} 代表负样例。上式中，定义 $0\log_2 0=0$。

假设 S 是有 14 个样例的一个集合，它包括 9 个正例和 5 个负例（采用记号$[9^{+},5^{-}]$表示），那么 S 相对于这个布尔样例的熵为

$$E([9^{+},5^{-}])=-\frac{9}{14}\log_2\frac{9}{14}-\frac{5}{14}\log_2\frac{5}{14}\approx 0.94 \tag{5-6}$$

若 S 的所有成员属于同一类，则 $E(S)=0$；若 S 的正、负样例数量相等，则 $E(S)=1$；若 S 的正、负样例数量不等，则熵介于 0～1 之间。

熵确定了要编码集合 S 中任意成员的分类所需要的最少二进制位数。更一般地，如果目标属性具有 c 个不同的值，那么 S 相对于 c 个状态的分类的熵定义为

$$E(S)=\sum_{i=1}^{c}-p_i\log_2 p_i \tag{5-7}$$

式中：p_i 为子集合中不同类别（二分类即正样例和负样例）的样例的比例。

定义 5-2 信息增益 $G(S,A)$。

有了熵作为衡量训练样例集合纯度的标准，就可以定义属性分类训练数据的效力的度量标准——信息增益。简单地说，一个属性的信息增益就是使用这个属性分割样例而导致的期望熵降低（或者说，样本按照某属性划分时造成熵减少的期望）。更精确地讲，一个属性 A 相对样例集合 S 的信息增益 $G(S,A)$ 定义为

$$G(S,A)=E(S)-\sum_{v\in V(A)}\frac{|S_v|}{|S|}E(S_v) \tag{5-8}$$

式中：$V(A)$ 为属性 A 所有可能值的集合；S_v 为 S 中属性 A 的值为 v 的子集。

换句话来讲，$G(S,A)$是由于给定属性 A 的值而得到的关于目标函数值的信息。当对 S 的一个任意成员的目标值编码时，$G(S,A)$的值是在知道属性 A 的值后可以节省的二进制位数。

假定 S 是一套有关天气的训练样例，描述它的属性可能是具有 Weak 和 Strong 两个值的 Wind。假定 S 包含 14 个样例$[9^+,5^-]$，其中正例中的 6 个和负例中的 2 个有 Wind=Weak，其他的有 Wind=Strong。由于按照属性 Wind 分类 14 个样例的信息增益可以计算如下：

$$V(\text{Wind})=\{\text{Weak},\text{Strong}\}$$

$$S=[9^+,5^-],\quad S_{\text{Weak}}=[6^+,2^-],\quad S_{\text{Strong}}=[3^+,3^-]$$

$$\begin{aligned}G(S,\text{Wind})&=E(S)-\sum_{v\hat{I}\{\text{Weak},\text{Strong}\}}\frac{|S_v|}{|S|}E(S_v)\\&=E(S)-\frac{8}{14}E(S_{\text{Weak}})-\frac{6}{14}E(S_{\text{Strong}})\\&=0.94-\frac{8}{14}\times\left|-\log_2\frac{6}{8}-\frac{2}{8}\log_2\frac{2}{8}\right|-\frac{6}{14}\times\\&\quad\left|-\frac{3}{6}\log_2\frac{3}{6}-\frac{3}{6}\log_2\frac{3}{6}\right|\\&=0.94-\frac{8}{14}\times0.811-\frac{6}{14}\times1\approx0.048\end{aligned}\tag{5-9}$$

定义 5-3 信息增益率 $G_R(S,A)$。

信息增益率度量是用信息增益 $G(S,A)$度量和分裂信息 SplitInformation(S,A)度量共同定义的(分裂信息用来衡量属性分裂数据的广度和均匀)，即

$$G_R(S,A)=\frac{G(S,A)}{\text{SplitInformation}(S,A)}\tag{5-10}$$

式中：

$$\text{SplitInformation}(S,A)=-\sum_{i=1}^{c}\frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}\tag{5-11}$$

式中：S_1 到 S_c 是 c 个值的属性 A 分割 S 而形成的 c 个样例子集。注意分裂信息实际上就是 S 关于属性 A 的各值的熵。

值得注意的是，分裂信息项阻碍选择值为均匀分布的属性。例如，考虑含有 n 个样例的一个集合被属性 A 彻底分割(分成 n 组，即一个样例一组)，这时分裂信息的值为 $\log_2 n$。一个布尔属性 B 分割同样的 n 个实例，如果恰好平分两半，那么分裂信息是 1。如果属性 A 和 B 产生同样的信息增益，那么根据增益比率度量，明显 B 得分更高。

信息增益率代替信息增益来选择属性产生的一个实际问题是，当某个 S_i 接近 S 时，分母可能为 0 或非常小。如果某个属性对于 S 的所有样例有几乎同样的值，这时要么导致增益比率未定义，要么是增益比率非常大。为了避免选择这种属性，可以采用一些启发式规则，比如先计算每个属性的增益，再仅对增益高于平均值的属性应用增益比率测试。

1. ID3 算法

迭代二叉树 3 代(Iterative Dichotomiser 3,ID3)算法是由 Ross Quinlan 发明的用于决策树的算法,这个算法建立在奥卡姆剃刀(Occam's Razor)的基础上,小型的决策树优于大型的决策树。尽管如此,该算法也不总是生成最小的树形结构,而是一个启发式算法。

奥卡姆剃刀是由 14 世纪逻辑学家、圣方济各会修士奥卡姆的威廉(William of Occam)提出,他在《箴言书注》2 卷 15 题说"切勿浪费较多东西,去做'用较少的东西,同样可以做好的事情'。简单点说,便是 Be Simple"。

从信息论知识中可知,期望信息越小,信息增益越大,从而纯度越高。ID3 算法的核心思想就是以信息增益度量属性选择,选择分裂后信息增益最大的属性进行分裂。该算法采用自顶向下的贪婪搜索遍历可能的决策树空间。

ID3 的思想:自顶向下的贪婪搜索遍历可能的决策树空间构造决策树;从"哪个属性将在树的根节点被测试"开始;使用统计测试来确定每个实例属性单独分类训练样例的能力,分类能力最好的属性作为树的根节点测试(如何定义或者评判一个属性是分类能力最好的呢?这便是上文介绍的信息增益或信息增益率);然后为根节点属性的每个可能值产生一个分支,并把训练样例排列到适当的分支(也就是说,样例的该属性值对应的分支)之下;重复这个过程,用每个分支节点关联的训练样例来选取在该点被测试的最佳属性。

ID3 算法的核心问题是选取在树的每个节点要测试的属性。希望选择最有利于分类实例的属性,信息增益是用来衡量给定的属性区分训练样例的能力,而 ID3 算法在增长树的每一步使用信息增益从候选属性中选择属性。

2. C4.5 算法

C4.5 算法是决策树核心算法,也是 ID3 算法的改进算法,C4.5 算法的改进:一是用信息增益率来选择属性。ID3 算法使用的是熵,也就是熵的变化值,而 C4.5 算法使用的是信息增益率。在树构造过程中进行剪枝,构造决策树时,只有少量元素的节点不要考虑,否则容易导致 overfitting。二是可以对非离散数据和不完整数据进行处理。

针对上述第一点,一般来说增益率就是用来取平衡的,与方差起的作用差不多。比如,有两个跑步的人,一个起点速度为 10m/s,10s 后速度为 20m/s;另一个人起点速度为 1m/s,1s 后速度为 2m/s。如果仅仅算差值,那么两个差距就很大了。如果使用速度增加率(加速度,即都是为 $1m/s^2$)来衡量,两个人就是一样的加速度。因此,C4.5 算法改进了 ID3 算法用信息增益选择属性时偏向选择取值多的属性的不足之处。

5.3.4 西瓜分类

下面以对西瓜分类为例,利用决策树进行西瓜分类问题的建模。为了实现定量建模,首先收集了如表 5-2 所示的西瓜数据集,记作 S。该数据集共有 17 条数据,将利用这些数据学习一棵没有被剖开的西瓜是不是好瓜的决策树。

表 5-2 西瓜数据集样例

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

1. 根节点

在决策树学习开始时，根节点包含 S 中所有的数据，其中正例占 $p_1=\frac{8}{17}$，反例占 $p_2=\frac{9}{17}$，于是可计算出该节点的信息熵为

$$E(S)=E([8^+,9^-])=-\frac{8}{17}\log_2\frac{8}{17}-\frac{9}{17}\log_2\frac{9}{17}\approx 0.998 \tag{5-12}$$

2. 属性分类的划分

计算当前属性集合{色泽，根蒂，敲声，纹理，脐部，触感}中每个属性的信息增益。以属性“根蒂”为例，该属性具有{卷缩，稍卷缩，硬挺}三个可能取值，若使用该属性对 S 进行划分，则可得到三个子集，分别记作 S_1（根蒂＝卷缩），S_2（根蒂＝稍卷缩），S_3（根蒂＝硬挺）。子集 S_1 包含编号为{1,2,3,4,5,12,16,17}的 8 个样例，其中正例占 $p_1=\frac{5}{8}$，反例占 $p_2=\frac{3}{8}$；子集 S_2 包含编号为{6,7,8,9,13,14,15}的 7 个样例，其中正例占 $p_1=\frac{3}{7}$，反例占 $p_2=\frac{4}{7}$；子集 S_3 包含编号为{10,11}的 2 个样例，均为反例。于是，基于“根蒂”划分之后所获得的 3 个分支节点计算信息熵为

$$E(S_1)=E([5^+,3^-])=-\frac{5}{8}\log_2\frac{5}{8}-\frac{3}{8}\log_2\frac{3}{8}\approx 0.954 \tag{5-13}$$

$$E(S_2)=E([3^+,4^-])=-\frac{3}{7}\log_2\frac{3}{7}-\frac{4}{7}\log_2\frac{4}{7}\approx 0.985 \tag{5-14}$$

$$E(S_3)=E([0^+,2^-])=-\frac{2}{2}\log_2\frac{2}{2}=0 \tag{5-15}$$

有了信息熵，便可以计算出属性“根蒂”的信息增益为

$$\begin{aligned}G(S,\text{根蒂})&=E(S)-\sum_{v=1}^{3}\frac{|S_v|}{|S|}E(S_v)\\&=0.998-\left(\frac{8}{17}\times 0.954+\frac{7}{17}\times 0.985+\frac{2}{17}\times 0\right)\\&\approx 0.143\end{aligned} \tag{5-16}$$

利用同样的方法可以计算出其他属性的信息增益分别为

$$\begin{aligned}&G(S,\text{色泽})=0.109,\quad G(S,\text{敲声})=0.141\\&G(S,\text{纹理})=0.381,\quad G(S,\text{脐部})=0.289\\&G(S,\text{触感})=0.006\end{aligned} \tag{5-17}$$

从计算结果可以发现，属性“纹理”的信息增益最大，于是将“纹理”选作划分属性。图 5-14 给出了基于“纹理”对根节点进行划分的结果，各分支节点所包含的样例子集显示在节点中。

图 5-14　基于“纹理”属性对根节点划分

3. 递归计算子分支

利用决策树学习算法对每个分支节点进一步划分。以图 5-14 中的第一个分支节点（“纹理＝清晰”）为例，该节点包含的样例集合 S_1 中有编号为{1,2,3,4,5,6,8,10,15}的 9 个样例，可用属性集合为{色泽，根蒂，敲声，脐部，触感}。基于 S_1 计算出各属性的信息增益为

$$\begin{aligned}&G(S_1,\text{色泽})=0.043,\quad G(S_1,\text{根蒂})=0.458\\&G(S_1,\text{敲声})=0.331,\quad G(S_1,\text{脐部})=0.458\\&G(S_1,\text{触感})=0.458\end{aligned} \tag{5-18}$$

“根蒂”“脐部”“触感”三个属性均取得了最大的信息增益，可任选其一作为划分属性。类似地，对每个分支节点进行上述操作，最终可得到如图 5-15 所示的决策树。

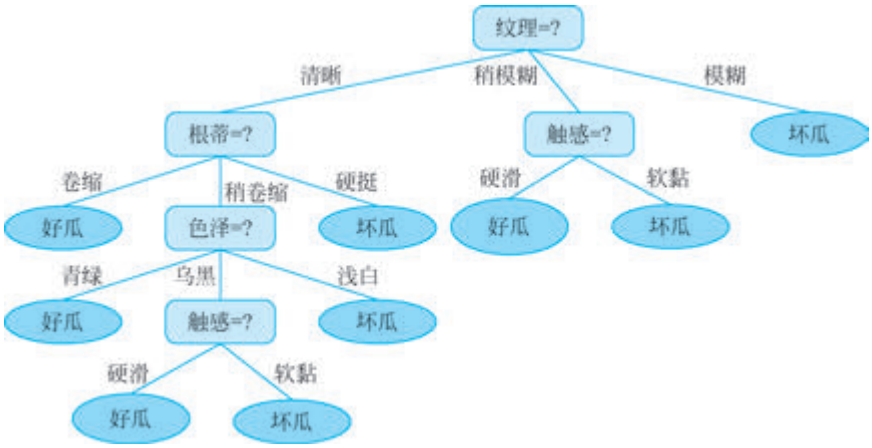


图 5-15　基于信息增益生成的决策树

5.4 聚类方法

自古至今，人类认知事物的方式多以聚类完成。比如，在没有苹果和梨的概念之前，人类会大量采摘这两种水果，通过对比它们的颜色、形状等，将采摘的水果分为不同的类，比如红色圆形一类、黄色椭圆形一类，再为每个类别赋予不同命名，红色圆形一类命名为苹果，黄色椭圆形一类命名为梨，这就产生了苹果和梨的概念。

在没有前人经验指导的条件下，聚类是人类认知世界的重要手段。对于拥有计算机的现代人类，如何利用计算机对给定数据进行聚类，发现新的规律或新的事物呢？下面学习聚类中的典型算法 k 均值聚类算法。

5.4.1　k 均值聚类算法

见名知意，k 均值由 k 和均值两部分组成，其中 k 表示类簇的个数，即我们期望将数据划分成几个类簇，means 表示类簇的均值。简言之，k 均值就是一种通过迭代计算类簇均值将数据点划分为 k 个类的聚类算法。

k 均值聚类算法计算流程图如图 5-16 所示。

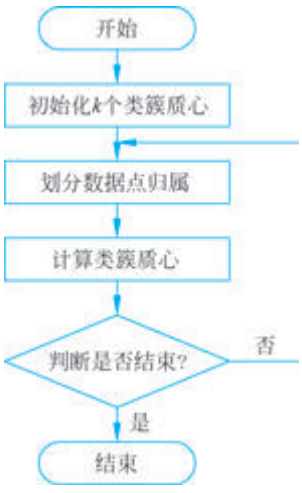


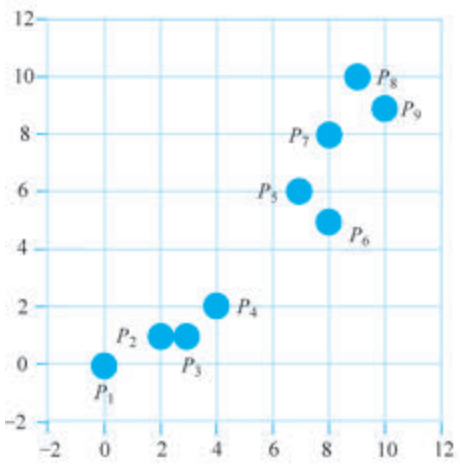
图 5-16　k 均值算法计算流程图

给定一个数据集，首先初始化 k 个类簇质心，最简单的方法就是随机从数据集中选择 k 个数据点作为 k 个类簇的初始质心；然后划分数据点归属，通过计算所有数据点到 k 个类簇质心的距离，将每个数据点划分给与其距离最近的类簇质心，形成 k 个类簇；接着计算类簇质心，对于每个类簇，计算该类簇内所有数据点的均值作为新的类簇质心；最后判断是否结束，通过判断所有类簇质心是否发生变化或达到预设最大迭代次数，若发生变化或达到预设最大次数，则终止算法，返回各数据点所属类编号，否则，重复划分数

据点归属、计算类簇质心过程。

从上述过程可以发现，k 均值聚类算法核心思想是迭代划分数据点归属、计算类簇质心。该算法需要预设待聚类的类别个数 k，且需要计算各个类簇数据点的均值，即类簇质心。这也是算法命名的由来。下面让结合示例进一步学习 k 均值聚类算法。

假设给定 9 个数据点，期望聚类的类别为 2 个，对应坐标信息如图 5-17 所示。



	X	Y
P_1	0	0
P_2	2	1
P_3	3	1
P_4	4	2
P_5	7	6
P_6	8	5
P_7	8	8
P_8	9	10
P_9	10	7

图 5-17　示例数据

根据 k 均值聚类算法，首先随机选取 2 个数据点作为质心，如选择 P_1 作为类 C_1 的质心，P_2 作为类 C_2 的质心。然后，执行迭代过程：

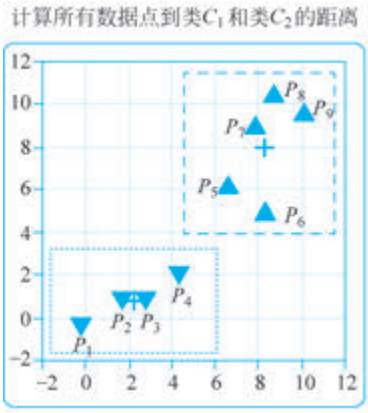
第一次迭代，计算所有数据点到类 C_1 和类 C_2 的距离，根据距离将数据点划分到类 C_1 或类 C_2 中，在此 P_1 距离类 C_1 最近，$P_2 \sim P_9$ 距离类 C_2 最近，由此得到了所有数据点的类别归属；然后计算各类数据点的均值作为新的质心，计算得到类 $C_1=(0.00, 0.00)$，类 $C_2=(6.38, 5.00)$，此时质心相对初始质心发生了改变，故继续循环。

第二次迭代，计算所有数据点到类 C_1 和类 C_2 的距离，根据距离将数据点划分到类 C_1 或类 C_2 类中，此时 P_1、P_2、P_3 距离类 C_1 最近，$P_4 \sim P_9$ 距离类 C_2 最近，由此得到所有数据点的类别归属；然后计算各类数据点的均值作为新的质心，计算得到类 $C_1=(1.67, 0.67)$，类 $C_2=(7.67, 6.33)$，此时质心相对初始质心发生了改变，故继续循环。

第三次迭代，计算所有数据点到类 C_1 和类 C_2 的距离，根据距离将数据点划分到类 C_1 或类 C_2 类中，此时 $P_1 \sim P_4$ 距离类 C_1 最近，$P_5 \sim P_9$ 距离类 C_2 最近，由此得到所有数据点的类别归属；然后计算各类数据点的均值作为新的质心，计算得到类 $C_1=(2.25, 1.00)$，类 $C_2=(8.40, 7.20)$，此时质心相对初始质心发生了改变，故继续循环。

第四次迭代，计算所有数据点到类 C_1 和类 C_2 的距离，根据距离将数据点划分到类 C_1 或类 C_2 类中，在此 $P_1 \sim P_4$ 距离类 C_1 最近，$P_5 \sim P_9$ 距离类 C_2 最近，由此得到了所有数据点的类别归属；然后计算各类数据点的均值作为新的质心，计算得到类 $C_1=(2.25, 1.00)$，类 $C_2=(8.40, 7.20)$，此时质心相对前一次迭代未发生改变，故终止循环。

至此，k 均值聚类算法结束，得到数据点的两类划分如图 5-18 所示。



	C_1数据点	C_2数据点	类质心	类质心是否改变
初始化			$C_1=P_1=(0.00,0.00)$ $C_2=P_2=(2.00,1.00)$	
第一次迭代	P_1	P_2、P_3、P_4、P_5、P_6、P_7、P_8、P_9	$C_1=(0.00,0.00)$ $C_2=(6.38,5.00)$	是
第二次迭代	P_1、P_2、P_3	P_4、P_5、P_6、P_7、P_8、P_9	$C_1=(1.67,0.67)$ $C_2=(7.67,6.33)$	是
第三次迭代	P_1、P_2、P_3、P_4	P_5、P_6、P_7、P_8、P_9	$C_1=(2.25,1.00)$ $C_2=(8.40,7.20)$	是
第四次迭代	P_1、P_2、P_3、P_4	P_5、P_6、P_7、P_8、P_9	$C_1=(2.25,1.00)$ $C_2=(8.40,7.20)$	否

图 5-18　聚类过程与结果

通过上述示例可以发现，k 均值聚类算法思想简单，实现容易；不足之处是需要手工输入类簇的数目，对初始质心敏感，同时对噪声和离群值也很敏感。算法不适合于发现差别很大的类簇或非球形的类簇，当两个类簇在差别极大或呈现非球形分布时，聚类结果会存在较大的问题。

5.4.2　k 均值超参数选择

在无法提前预知类别个数的条件下，如何设定类别个数？实际上，这涉及如何评价聚类结果好坏的问题。从聚类效果来说，期望聚类后同类数据点之间越近越好，不同类数据点之间越远越好。

下面介绍一种定量地评价不同聚类结果的好坏评价指标——轮廓系数。针对第 i 个数据点，对应轮廓系数定义为

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{5-19}$$

式中：$a(i)$为簇内距离，表示第 i 个数据点与其所在簇内其他数据点的平均距离，该值越小，说明该数据点与同类数据的距离越小，同类数据聚类越紧凑；$b(i)$为簇间距离，表示第 i 个数据点与其最近的簇内数据点的平均距离，该值越大，说明该数据点距离与其最近的簇越远，不同类簇分离得越远，不同类数据之间的分布越分散。

如图 5-19 所示，有 A、B、C 三个簇，簇 A 中数据点 i 到簇 B 内数据点的平均距离 $b(i,B)$，小于其到簇 C 内数据点的平均距离为 $b(i,C)$，故 $b(i)=b(i,B)$。

聚类总的轮廓系数定义为所有 N 个数据点的轮廓系数和平均，即

$$\mathrm{SC}=\frac{1}{N}\sum_{i=1}^{N}s(i) \tag{5-20}$$

轮廓系数综合考虑了簇内距离和簇间距离对聚类质量进行评价。如表 5-3 所示，轮廓系数为 $-1\sim1$，数值越大表明聚类结果越好，数值越小表明聚类结果越差。当 $b(i)=0$ 时，轮廓系

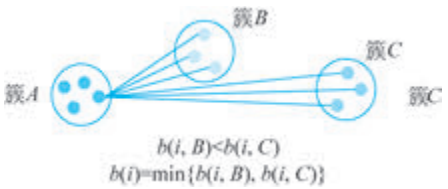


图 5-19　$b(i)$计算过程示意图

数为－1，此时不同的簇无法分开。当 $a(i)=0$ 时，轮廓系数数值为 1，此时簇退化为一个点。

表 5-3　轮廓系数评价聚类质量的参考值

轮 廓 系 数	对应效果描述
0.7～1.0	聚类结果具有良好的划分
0.5～0.7	聚类簇基本明确，但存在噪声点
0.25～0.5	聚类簇尚可辨识，很多样本难以确认所属簇
－1.0～0.25	无法分辨结果簇，近似于随机划分

案例

5.5 应用案例

文本聚类技术将无标签的文档按内容的相似性归为若干类别。文本聚类过程是对文档集合自动进行归类的过程，由于文档类别预先未知，需要根据数据的相关性来学习并得到文档的归类。文本聚类的目标是将文档划分为若干类别，同一类别中的文档内容相似度尽可能大，不同类别间的文档相似度尽可能小。

1. 给定类别条件下的聚类

案例数据来自环球军事网，主要涉及各类武器的文本简介，包括 8 个类别，共 430 个文档。期望将这些文档聚类为飞行器、舰船舰艇、枪械与单兵、坦克装甲车、火炮、导弹武器、太空装备和爆炸物 8 个类别。以上数据通过网络爬虫获取得到，存储于 txt 文档中，每个文档对应一个武器的文本简介。

由于聚类算法无法直接处理文本数据，采用文本预处理技术将每个文档内容转换为一个向量。具体实现时，首先采用“结巴”中文分词器对文本进行分词获得文档词；然后通过 TF-IDF 对文档词进行向量化表示；最后为了简化表示和可视化，通过 T-SNE 算法将特征至 2 维。这样每个文档就可以用 2 维平面上的一个点来表示，点与点之间的远近表达了文档和文档之间的相似程度，两点越近表示对应两个文档越相似。

文档预处理结束后，采用 k 均值聚类算法进行聚类。设置类别为 8 个，最大迭代次数为 100 次。可以得到如图 5-20 所示的聚类结果。

彩图

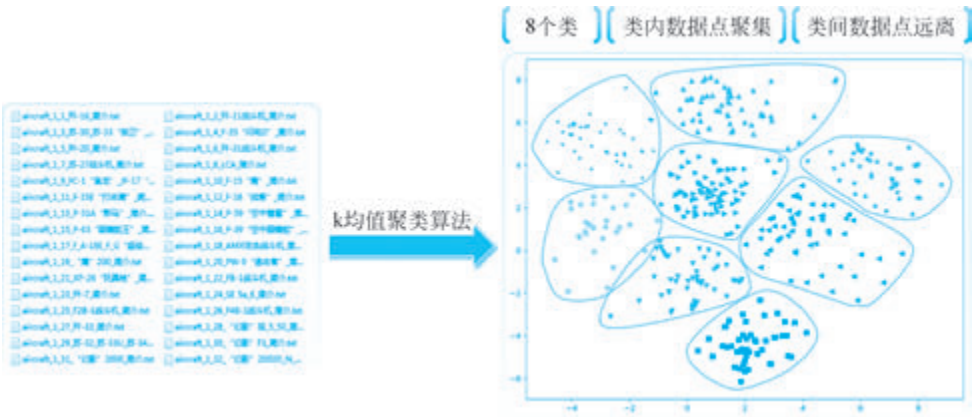


图 5-20　文档与聚类结果

图 5-20 中，不同颜色的点表示不同的类簇。由图 5-20 可以发现，数据点被归为 8 个类，且类内数据点相对聚集，类间数据点相对远。

此时，对每个类的所有文本内容进行词云显示，得到 8 个类的词云图，如图 5-21 所示。

图 5-21　类别数据词云可视化

由图 5-21 可以发现，类 1 包含 AK、56 毫米等词，表示枪械与单兵；类 2 包含大黄蜂、幻影等词，表示飞行器；类 4 包含 BMP、车顶水平等词，表示坦克装甲车辆；类 6 包含 USS、CV、CVN 等词，表示舰船舰艇；类 8 包含东风、烈火等词，表示导弹武器。

上述结果表明，虽然在聚类的过程中没有使用到预先了解的内容标签，但是 k 均值聚类算法聚类后各个文档实际上按照一定的主题聚在了一起。

2. 未给定类别条件下的聚类

k 均值聚类算法需要预设类别个数 k，当 k 设置为 6 或 10 时，结果如图 5-22 所示。

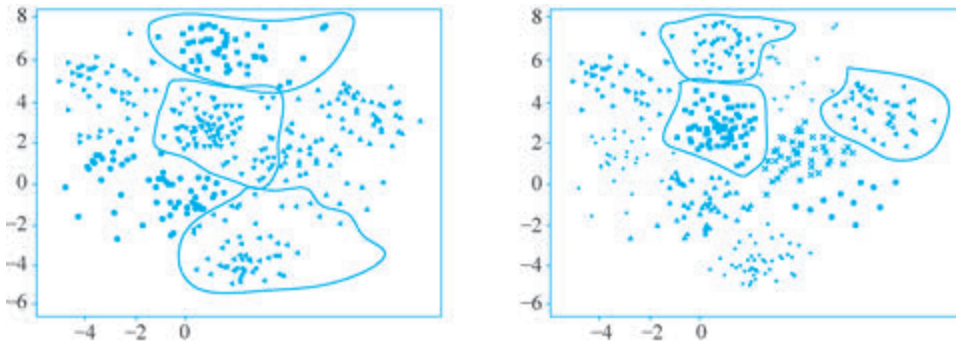


图 5-22　不同类别个数聚类结果

由图 5-22 可以发现，各类之间仍具有较好的类聚性。此时，如何来决定聚类 6 个类别好还是聚类 10 个类别好？根据轮廓系数评价指标，对 k 从 2～18 不同的值进行 k 均值聚类，计算每种聚类结果的轮廓系数，具体结果如图 5-23 所示。

从图 5-23 可以发现，当 $k=8$ 时，轮廓系数最大，表明此时聚类结果最佳。同时还可以发现，$k=10$ 相较 $k=6$ 的轮廓系数要大；$k=10$ 时的分类结果相较 $k=6$ 的结果，同类

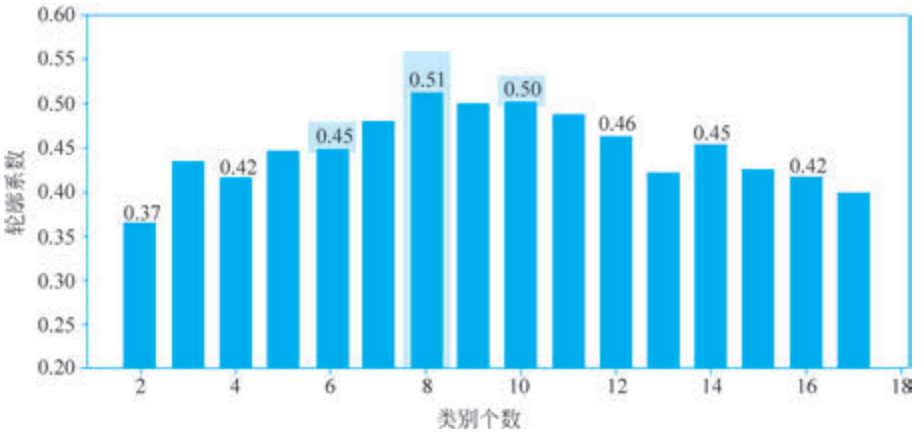


图 5-23　8 个类的轮廓系数

数据点更加聚集，不同类数据点在类边界上具有明显的可分性。

上述结果表明，通过轮廓系数的对比，可以选择最佳的聚类结果。

5.6 本章小结

机器学习作为人工智能领域的核心技术，宛如一把神奇的钥匙开启了计算机自主认知世界的大门。它巧妙地模拟人类的学习行为，让计算机能够从海量的数据中自动挖掘规律、优化性能，进而广泛应用于智能服务、交通、医疗等众多领域，深刻改变了我们的生活与社会的运行模式。

机器学习的核心逻辑在于借助历史数据（经验）精心训练模型，使其具备对新数据进行精准预测的能力，整个过程主要涵盖训练和预测两个关键阶段。依据学习方式的不同，机器学习可分为监督学习、无监督学习和半监督学习。监督学习，如回归、分类，依赖带标签的数据进行训练，在房价预测、图像识别等任务中表现出色；无监督学习，像聚类，通过挖掘数据内在相似性实现自主分类，在市场细分、异常检测等领域发挥着重要作用；半监督学习则结合少量标签与大量无标签数据，有效提升模型的泛化能力。

本章围绕这些核心内容展开，详细阐述了有监督学习分类方法中的决策树，以及无监督学习聚类方法中的 kNN 和 k 均值。这些算法是机器学习入门的基石，为我们打开了机器学习的大门。然而，机器学习的世界远不止于此，集成学习 Adboosting、支持向量机，以及密度聚类 DBSCAN 等更为复杂且强大的算法才是在实际应用中发挥作用的主力军，等待我们去深入探索与发掘。

习题

一、选择题

1. 机器学习的主要任务不包括以下哪一项？（　　）

A. 分类　　B. 聚类　　C. 回归　　D. 语义理解

2. 以下哪种算法不属于监督学习？（　　）

A. 决策树　　B. 支持向量机　　C. k 均值聚类　　D. 逻辑回归

3. 以下哪项任务更适合使用 k 均值聚类算法，而不是 kNN 分类算法？（　　）

A. 根据用户的历史购买记录，预测用户是否会购买某件商品。

B. 将客户按照消费习惯分成几个不同的群体，以便进行精准营销。

C. 根据病人的症状判断其患有哪种疾病。

D. 根据图片内容判断图片是猫还是狗。

4. 在 kNN 分类算法中，k 值的选择对分类结果的影响是什么？（　　）

A. k 值越大，分类结果越准确　　B. k 值越小，分类结果越准确

C. k 值越大，分类边界越平滑　　D. k 值越小，分类边界越复杂

5. 在 k 均值聚类算法中，“均值”指的是（　　）。

A. 聚类中心点（质心）是类簇样本点的算术平均值

B. 算法需要迭代多次，取平均结果

C. 算法对噪声和异常值不敏感

D. 算法最终会收敛到全局最优解

二、简答题

1. 简述机器学习的基本定义。

2. 简述机器学习的主要分类及其之间的主要区别。

3. 简述 k 近邻分类的基本思想、优缺点及主要适用条件。

4. 简述决策树分类的基本思想、优缺点及主要适用条件。

5. 简述 k 均值聚类的基本思想、优缺点及主要适用条件。

三、计算题

1. 假设有一个简单的二维数据集，用于分类任务。数据集包含训练样本：类别 A，点(1,1)、(2,2)、(3,1)；类别 B，点(6,5)、(7,7)、(8,6)。现在有一个新的待分类点(4,3)。使用 kNN 算法分别计算当 k 为 1 和 3 时的分类结果（假设距离度量使用欧几里得距离）。

2. 假设有一个二维数据集，包含样本点(1,1)、(2,2)、(6,5)、(7,7)和(8,6)。使用 k 均值聚类算法对这 5 个点进行聚类。初始时选择点 1 和点 3 作为初始聚类中心。

3. 假设有一个简单的数据集，用于预测是否下雨。数据集包含以下特征和标签：

风速（wind speed）	湿度（humidity）	是否下雨（rain）
高（high）	高（high）	是（yes）
高（high）	低（low）	否（no）
低（low）	高（high）	是（yes）
低（low）	低（low）	否（no）
高（high）	高（high）	是（yes）
低（low）	高（high）	是（yes）

使用 ID3 算法构建一个简单的决策树分类模型。