第1章　机器学习绪论

1.1　机器学习简介

1.1.1　机器学习简史

机器学习(ML)是人工智能(artificial intelligence,AI)的一个分支领域,它使得计算机系统能够通过学习数据模式,自动改善其性能。ML 算法使用大量的数据(输入数据)来训练模型,使得计算机能够预测、分类或者作出决策,而无须明确的编程规则。ML 在各个领域中得到了广泛的应用,包括自然语言处理、图像识别、语音识别、推荐系统、医疗诊断等领域。ML 的发展和 AI 的发展是分不开的,ML 是 AI 研究发展到一定阶段的必然产物。ML 是一门多领域交叉学科,涉及概率论、统计学等多门学科。它研究的是计算机怎样模拟或实现人类的学习行为,以获取新的知识或技能,重新组织已有的知识结构使之不断改善自身的性能。ML 的发展简史如下:

- 1943 年 McCulloch 和 Pitts 提出了神经网络(neural network,NN)层次结构模型,确立了神经网络的计算模型理论,从而为机器学习的发展奠定了基础。
- 1950 年 Turing 提出了著名的"图灵测试",使人工智能成为科学领域的一个重要研究课题。
- 1957 年 NN 第一次崛起,Rosenblatt 提出了感知器(perceptron)概念,用 Rosenblatt 算法对 perceptron 进行训练。并且首次用算法精确定义了自组织自学习的神经网络数学模型,设计出了第一个计算机 NN 算法,开启了 NN 研究活动的第一次兴起。
- 1958 年 Cox 给逻辑回归(logistic regression,LR)方法正式命名,用于解决美国人口普查任务。
- 1959 年 Samuel 设计了一个具有学习能力的跳棋程序,曾经战胜了美国保持 8 年不败的冠军。这个程序向人们初步展示了机器学习的能力,Samuel 将机器学习定义为无须明确编程即可为计算机提供能力的研究领域。
- 1960 年 Widrow 用 delta 学习法则来对 perceptron 进行训练,可以比 Rosenblatt 算法更有效地训练出良好的线性分类器(最小二乘法问题)。

- 1962 年 Hubel 和 Wiesel 发现了猫脑皮层中独特的神经网络结构可以有效降低学习的复杂性，从而提出著名的 Hubel-Wiese 生物视觉模型，该模型是卷积神经网络（CNN）的雏形，这之后提出的神经网络模型也均受此启迪。
- 1963 年 Vapnik 和 Chervonenkis 发明原始支持向量方法，即起决定性作用的样本为支持向量算法（support vector machine，SVM）。
- 1969 年 Minsky 和 Papert 出版了对机器学习研究有深远影响的著作 *Perceptron*，其中对于机器学习基本思想的论断——解决问题的算法能力和计算复杂性，影响深远且延续至今。文章中提出了著名的线性感知机无法解决异或问题，打击了 NN 社区，从此以后 NN 研究活动直到 20 世纪 80 年代都萎靡不振。
- 1980 年在美国卡内基梅隆大学举行了第一届机器学习国际研讨会，标志着机器学习研究在世界范围内兴起，该研讨会也是著名会议国际机器学习大会（ICML）的前身。
- 1981 年 Werbos 提出多层感知机，解决了线性模型无法解决的异或问题，第二次兴起了 NN 研究。
- 1984 年 Breiman 发表分类回归树（CART，一种决策树）算法。
- 1986 年 Quinlan 提出 ID3 算法（一种决策树）。
- 1986 年 Rumelhart、Hinton 和 Williams 联合在 *Nature* 杂志发表了著名的反向传播算法（BP 算法）。
- 1989 年 Yann 和 LeCun 提出了目前最为流行的卷积神经网络（CNN）计算模型，推导出基于 BP 算法的高效训练方法，并成功地应用于英文手写体识别。
- 1995 年 Vapnik 和 Cortes 发表软间隔支持向量机算法（SVM），开启了随后的机器学习领域 NN 和 SVM 两大社区的竞争。
- 1995 年到随后的 10 年，NN 研究发展缓慢，SVM 在大多数任务的表现上一直压制着 NN，并且 Hochreiter 的工作证明了 NN 存在一个严重缺陷——梯度爆炸和梯度消失问题。
- 1997 年 Freund 和 Schapire 提出了另一种可靠的机器学习方法——adaboost。
- 2001 年 Breiman 发表随机森林（random forest）方法，adaboost 在对过拟合问题和奇异数据容忍上存在缺陷，而随机森林在这两个问题上更具有鲁棒性。
- 2005 年，经过多年的发展，NN 众多研究发现被现代 NN“大牛”Hinton、LeCun、Bengio、Andrew Ng 和其他老一辈研究者整合，NN 随后开始被称为深度学习（deep learning），迎来了第三次崛起。

1.1.2 机器学习主要流派

1. 行为主义派

行为主义派又称为进化主义或控制论学派，是一种基于“感知—行动”的行为智能模拟方法，行为主义最早起源于 20 世纪的一个心理学流派，行为主义认为行为是有机体用以适应环境变化的各种身体反应的组合，它的理论目标在于预见和控制行为。其核心在于控制论，认为人工智能源于控制论。控制论思想早在 20 世纪四五十年代就成为时代思潮的重要部分，影响了早期的 AI 工作者。维纳（Wiener）[1] 和麦卡洛克（McCulloch）[2] 等提出的控制论和自组织系统以及钱学森等提出的工程控制论和生物控制论，影响了许多领域。控制论

把神经系统的工作原理与信息理论、控制理论、逻辑以及计算机联系起来。早期的研究工作重点是模拟人在控制过程中的智能行为和作用，如对自寻优、自适应、自镇定、自组织和自学习等控制论系统的研究，并进行“控制论动物”的研制。到 20 世纪六七十年代，上述这些控制论系统的研究取得一定进展，播下智能控制和智能机器人的种子，并在 20 世纪 80 年代诞生了智能控制和智能机器人系统。行为主义是 20 世纪末才以 AI 新学派的面孔出现的，引起许多人的兴趣。这一学派的代表作首推布鲁克斯(Brooks)的六足行走机器人，它被看作是新一代的“控制论动物”，是一个基于感知—动作模式模拟昆虫行为的控制系统。

2. 连接主义派

连接主义派又称仿生学派或生理学派，是一种基于神经网络及网络间的连接机制与学习算法的智能模拟方法的学派，其原理主要为神经网络和神经网络间的连接机制及学习算法，该学派认为人工智能源于仿生学，特别是对人脑模型的研究。它的代表性成果是 1943 年由生理学家麦卡洛克和数理逻辑学家皮茨(Pitts)创立的脑模型[2]，即 MP 模型，开创了用电子装置模仿人脑结构和功能的新途径。它从神经元开始进而研究神经网络模型和脑模型，开辟了人工智能的又一发展道路。20 世纪六七十年代，连接主义，尤其是对以感知机(perceptron)为代表的脑模型的研究出现过热潮，由于受到当时的理论模型、生物原型和技术条件的限制，脑模型研究在 20 世纪 70 年代后期至 80 年代初期落入低潮。直到 Hopfield 教授在 1982 年和 1984 年发表两篇重要论文，提出用硬件模拟神经网络[3-4]以后，连接主义才又重新抬头。1986 年，鲁梅尔哈特(Rumelhart)等[5]提出多层网络中的反向传播(BP)算法。此后，连接主义势头大振，从模型到算法，从理论分析到工程实现，为神经网络计算机走向市场打下基础。现在，对人工神经网络(ANN)的研究热情仍然较高，但研究成果没有像预想得那样好。

3. 符号主义派

符号主义派是一种基于数理逻辑推理的智能模拟方法，又称为逻辑主义、心理学派或计算机学派，其原理主要为物理符号系统假设和有限合理性原理，长期以来在人工智能中处于主导地位。其核心侧重于数理逻辑。数理逻辑从 19 世纪末起得以迅速发展，到 20 世纪 30 年代开始用于描述智能行为。计算机出现后，又在计算机上实现了逻辑演绎系统。其有代表性的成果为启发式程序“逻辑理论家”(logic thorist，LT)，它证明了 38 条数学定理，表明了可以应用计算机研究人的思维过程，模拟人类智能活动。正是这些符号主义者，早在 1956 年就首先采用了“人工智能”这个术语。后来又发展了启发式算法、专家系统、知识工程理论与技术，并在 20 世纪 80 年代取得很大发展。符号主义曾长期一枝独秀，为人工智能的发展作出重要贡献，尤其是符号主义专家系统的成功开发与应用，为人工智能走向工程应用和实现理论与实践相结合提供了特别重要的意义。符号主义学派在人工智能领域的代表人物有纽厄尔(Newell)、西蒙(Simon)和尼尔逊(Nilsson)等。

1.2 人工智能与机器学习的关系

1.2.1 什么是人工智能

AI 的概念是在 1956 年提出的。ML 包含于 AI。AI 是一个更广泛的概念，即让机器能

够以我们认为“智能”的方式执行任务。AI 是一种具体的结果，而 ML 是我们达到 AI 的一个途径。ML 是 AI 的一个子集，它可以被定义为 AI 的一个分支，或者也可以认为是 AI 的一些具体应用，它的目的是让计算机能够更好地访问和利用数据，并且在没有人工干预的情况下从中检索出事件发展的内在规律。AI 是指计算机系统能够模拟、理解、学习和执行人类智能任务的能力。这包括了通过计算机程序实现的语音识别、图像识别、自然语言处理、推理、决策制定等一系列智能行为。AI 的目标是使计算机系统能够以与人类智能相似的方式作出思考、学习、解决问题，并且在某些任务上甚至能够超过人类的智能水平。

AI 可以分为弱人工智能和强人工智能两种类型。

(1) 弱人工智能(narrow AI)：是指在特定任务或领域内表现出色的人工智能系统。这些系统可以处理特定的任务，但在其他领域或任务上的表现不如人类。例如，语音助手(如 Siri、Alexa)和自动驾驶汽车属于弱人工智能，它们在特定任务上具有智能，但没有人类的智能水平。

(2) 强人工智能，也称通用人工智能(artificial general intelligence，AGI)：是指能够理解、学习、适应各种任务，并且在各个领域都能够像人类一样灵活运用智能的人工智能系统。强人工智能具有类似人类的智能水平，能够在多个领域进行通用性智能表现。目前，强人工智能尚未实现，是 AI 领域的一个追求目标。

1.2.2 机器学习、人工智能的关系

ML：从算法的角度出发，在数据挖掘中提供详细的学习模型，从数据中发掘知识。机器学习更加强调的是算法，即从数据中获取有用知识的算法，需要一定的训练数据集，构建过往经验的“知识”，在进行数据挖掘过程中提供详细的学习模型。机器学习还可以解决其他问题，并不仅限于数据挖掘，其应用范围更加广泛。

AI 的主要目的是使机器拥有像人一样的推理分析和解决问题的能力，为了达到这一目的，必须要有足够的数据和先进的算法，并且还要对仿生学和认知心理学有一定的了解，能够让机器模仿人的行动。

数据挖掘、ML 和 AI 之间相互联系紧密，却又不可区分，一步一步随着技术的不断演进而发展。但是各自有不同的目的，因此从目的出发，能够对这三者进行更好的区分，如图 1.1 所示。

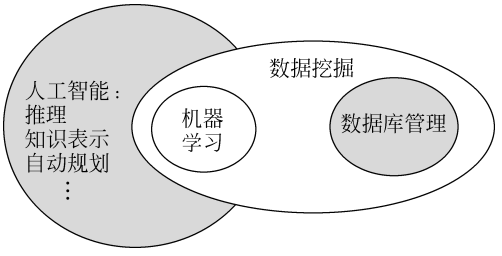

图 1.1 机器学习、人工智能、数据挖掘的关系

1.3 典型机器学习应用领域

ML 被广泛应用于餐饮业、农业、金融、医学等各个领域。在餐饮业，可以根据用户评论改进食谱、利用神经网络预测食品设施的需求、利用 Keras 进行食品分类、使用深度学习将

图片翻译成食谱、按餐厅人数作回归分析等；在农业领域，可以进行农产品价格预测，利用卫星图像进行农业地块分割，利用图像的深度学习框架对农作物病虫害进行识别、分析灌溉数据和预测病虫害发生的可能性等；在金融领域，可以利用自动化功能工程来预测贷款偿还方式，帮助银行建立检查客户是否有资格获得给定贷款的系统、信用卡审批系统，深度学习模型预测交易金额和未来交易天数、预测信用卡客户流失率、信用评分等；在计算机视觉领域，可以进行面部识别、运动追踪、物体检测等；在计算机生物学领域，可以进行 DNA 测序、脑肿瘤检测、药物发现等；在自然语言处理领域，可以进行语音识别、文本分析、机器翻译、问答系统、情感分析等；在汽车、航空航天和制造中，进行预测性维护。ML 的应用领域如图 1.2 所示。

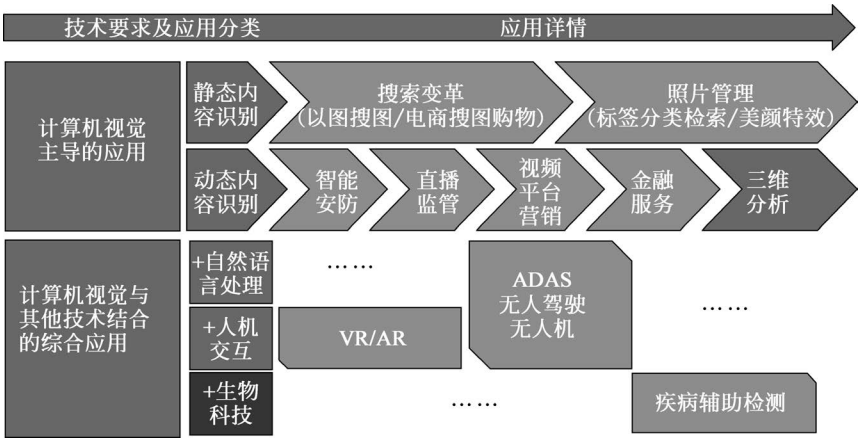

图 1.2　机器学习的应用领域

1.4 机器学习算法

ML 算法是用于从数据中学习规律和模式，以便作出预测、分类、决策等任务的数学模型和方法。这些算法可以根据学习方式和目标任务的不同进行分类。基本的 ML 算法或模型包含线性回归、支持向量机(SVM)、K 近邻(KNN)、逻辑回归、决策树、k-均值、随机森林、朴素贝叶斯、降维、梯度增强。ML 算法大致可以分为三类。

监督学习算法(supervised algorithms)：在监督学习训练过程中，可以由训练数据集学到或建立一个模式(函数/learning model)，并以此模式推测新的实例。该算法要求特定的输入输出，首先需要决定使用哪种数据作为范例。例如，我们可以选择使用一个包含猫和狗图像的数据集作为我们的范例数据。数据集中的每个图像都有一个对应的类别标签，指示该图像是猫还是狗。主要算法包括神经网络、支持向量机、K 近邻算法、朴素贝叶斯法、决策树等。

无监督学习算法(unsupervised algorithms)：这类算法没有特定的目标输出，算法将数据集分为不同的组。

强化学习算法(reinforcement algorithms)：强化学习普适性强，主要基于决策进行训练，算法根据输出结果(决策)的成功或错误来训练自己，大量经验训练优化后的算法将能够

给出较好的预测。类似有机体在环境给予的奖励或惩罚的刺激下，逐步形成对刺激的预期，产生能获得最大利益的习惯性行为。在运筹学和控制论的语境下，强化学习被称作“近似动态规划”(approximate dynamic programming，ADP)。

1.4.1 线性回归

线性回归是一种回归分析方法，它将某一现象或类归纳为线性模型。通过连续的输入变量，预测出一个值；通过拟合最佳直线来建立线性模型拟合自变量和因变量之间的关系，而这条最佳直线就叫作回归线，并且可以用 $Y= aX+b$ 表示，如图 1.3 所示。

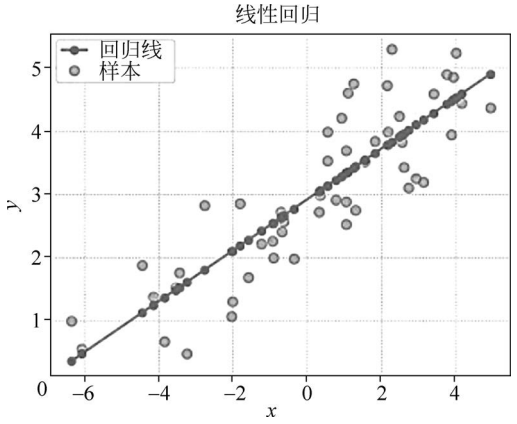

图 1.3 线性回归拟合例子

用线性等式来表示一元线性回归(Y 为因变量，X 为自变量，a 为斜率，b 为截距)，一元线性回归的特点是只有一个自变量。对于多元线性回归存在多个自变量，找最佳拟合直线时，可以拟合到多项或者曲线回归，其表达式如下：

$$Y=a_1X_1+a_2X_2+a_3X_3+\cdots+a_nX_n+b$$

线性回归不像逻辑回归是用于分类，其基本思想是用梯度下降法对最小二乘法形式的误差函数进行优化，当然也可以用正规方程(normal equation)直接求得参数的解，结果为

$$\hat{\boldsymbol{w}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}$$

而在局部加权线性回归(LWLR)中，参数的计算表达式为

$$\hat{\boldsymbol{w}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Y}$$

由此可见，LWLR 与 LR 不同，LWLR 是一个非参数模型，因为每次进行回归计算都要遍历训练样本至少一次。

1.4.2 逻辑回归

逻辑回归是一种分类算法，通过将数据拟合到一个逻辑函数中来预测事件发生的概率。对于 0～1 的概率值，当概率大于 0.5，预测为 1，概率小于 0.5，则预测为 0。它属于判别式模型，有很多正则化模型的方法(L0、L1、L2 等)。如果需要一个概率架构(比如，简单地调节分类阈值，指明不确定性，或者是要获得置信区间)，或者希望以后将更多的训练数据快速整合到模型中去，可以使用逻辑回归。逻辑回归中使用的 Sigmoid 函数为

$$f(x)=\frac{1}{1+e^{-x}}$$

逻辑回归的优点：实现简单，广泛应用于工业问题上；分类时计算量非常小，速度很快，占用存储资源低；便利的观测样本概率分数。对逻辑回归而言，多重共线性并不是问题，它可以结合 L2 正则化来解决该问题。

逻辑回归的缺点：当特征空间很大时，逻辑回归的性能不是很好；容易出现欠拟合，准确度一般不太高；只能处理两分类问题（在此基础上衍生出来的归一化指数函数 Softmax 可以用于多分类），且必须线性可分。对于非线性特征，还需要进行转换。

1.4.3 决策树

决策树算法通常用于分类问题。它同时适用于分类变量和连续因变量，在这个算法中，将总体分成两个或更多的同类群。根据属性或者自变量分成尽可能不同的组别，再根据一些特征（feature）进行分类，每个节点提一个问题，通过判断，将数据分为两类，再继续提问。这些问题是根据已有数据学习出来的，再投入新数据时，就可以根据这棵树上的问题，将数据划分到合适的叶子上。经典的决策树是 ID3，其中有两个重要的概念：熵和信息增益。

熵：是描述系统混乱的程度（在模型中，熵越小，混乱程度越小，模型越好）。

信息增益：是描述属性（非叶子节点）对模型的贡献。信息增益越大，它对模型的贡献越大。

决策树算法优点：计算复杂度不高，输出结果易于理解，对中间值的缺失不敏感，可以处理不相关特征数据。缺点：可能会产生过度匹配问题。决策树的基本结构如图 1.4 所示。

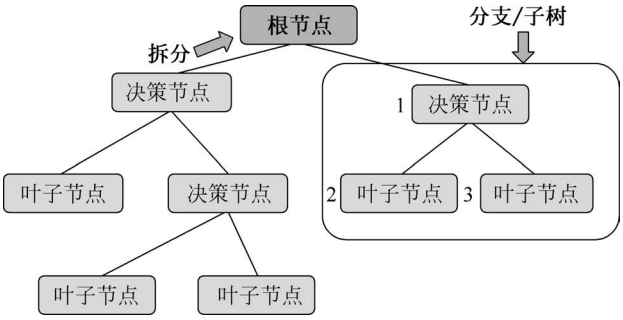

图 1.4 决策树基本结构

1.4.4 支持向量机

支持向量机（SVM）是一种分类方法。在这个算法中，将每个数据在 N 维空间中用点标出（N 是所有的特征总数），每个特征的值是一个坐标的值。SVM 算法的实质是找出一个能够将某个值最大化的超平面，这个值就是超平面离所有训练样本的最小距离。这个最小距离用 SVM 术语来说，叫作间隔（margin）。假设给定一些分属于两类的 2 维点，这些点可以通过直线分割，找到一条最优的分割线。支持向量机将向量映射到一个更高维的空间里，在这个空间里建立一个最大间隔超平面。在分开数据的超平面的两边建有两个互相平

行的超平面，分隔超平面使两个平行超平面的距离最大化。假定平行超平面间的距离或差距越大，分类器的总误差就越小。SVM 由训练算法、测试算法、使用算法三个部分组成。

训练算法：SVM 的大部分时间都源自训练，该过程主要实现两个参数的调优。

测试算法：十分简单的计算过程就可以实现。

使用算法：几乎所有分类问题都可以使用 SVM，值得一提的是，SVM 本身是一个二类分类器，对多类问题应用 SVM 需要对代码做一些修改。

1.4.5 线性支持向量机

求解线性支持向量机的过程是凸二次规划问题。所谓凸二次规划问题，就是目标函数是凸的二次可微函数。而求解凸二次规划问题可以利用对偶算法，即引入拉格朗日算子，利用拉格朗日对偶性将原始问题的最优解问题转化为拉格朗日对偶问题，这样就将求 w^*、bw^*、b 的原始问题的极小问题转化为求：

$$\alpha \geqslant 0, \quad \sum_{i=1}^{m} \alpha_i \text{lable}^{(i)} = 0$$

对偶问题的极大问题，即求出 α^*，再通过卡罗需-库恩-塔克（KKT）条件求出对应的参数 w^*、bw^*、b，从而找到这样的间隔最大化超平面，进而利用该平面完成样本分类。目标函数如下：

$$\max_{\alpha} \left[\sum_{i=1}^{m} \alpha - \frac{1}{2} \sum_{ij=1}^{m} \text{lable}^{(i)} \text{lable}^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} > \right]$$

1.4.6 非线性支持向量机

当数据集不是线性可分时，不能通过前面的线性模型来对数据集进行分类。此时，必须想办法使这些样本特征符合线性模型，才能通过线性模型对这些样本进行分类。这就要用到核函数，核函数的功能就是将低维的特征空间映射到高维的特征空间，而在高维的特征空间中，这些样本经过转化后，变成了线性可分的情况，这样，在高维空间中，能够利用线性模型来解决数据集分类问题。

核函数的优点：泛化错误率低，计算开销不大，结果易被解释。缺点：对参数调节和核函数的选择敏感，原始分类器不加修改仅适用于处理二类问题。

1.4.7 随机森林

随机森林是一种利用多棵树对样本进行训练和预测的分类器[6]。它由多棵分类回归树（classification and regression tree，CART）组成。对于每棵树，训练集是从总的训练集中进行有放回采样的，这意味着总训练集中的一些样本可能会在某棵树的训练集中出现多次，也可能从未出现在其他树的训练集中。在训练每棵树的节点时，使用的特征是按照一定比例从所有特征中进行无放回随机抽取的。

随机森林的优点：在数据集上表现良好，在许多数据集上相对其他算法具有显著优势；能够处理高维度（特征较多）的数据，无须进行特征选择；可以评估特征的重要性；在创建随机森林时，使用的是无偏估计来评估泛化误差；训练速度快，易于并行化；在训练过程中，可以检测特征之间的相互影响；实现相对简单；对于不平衡的数据集，可以平衡误差；

可以应用于具有特征缺失的数据集，并且仍然具有良好的性能。缺点：随机森林已经被证明在某些具有较大噪声的分类或回归问题上容易出现过拟合；对于具有不同取值的属性，取值划分较多的属性会对随机森林产生较大影响，因此在这种数据上生成的属性权重不可靠。

1.4.8 k-均值算法

k-均值算法是一种常用的无监督学习算法，用于将数据集划分成 k 个不同的类别[7]。k-均值算法的基本思想是：将数据集中的每个样本分配到距离其最近的 k 个质心所代表的类别中，然后重新计算每个类别的质心，不断重复以上过程，直到类别不再发生变化或达到预定的迭代次数。k-均值算法的实现过程包括以下几个步骤。

(1) 随机选取 k 个样本作为初始质心。

(2) 计算每个样本与 k 个质心之间的距离，将每个样本分配到距离其最近的质心所代表的类别中。

(3) 重新计算每个类别的质心，将其设置为该类别中所有样本的平均值。

(4) 不断重复以上过程，直到类别不再发生变化或达到预定的迭代次数。

k-均值算法的优点包括实现简单和计算速度快等，同时也存在对初始质心的敏感性和需要事先确定类别数量 k 的缺点。在实际应用中，k-均值算法经常用于图像分割、用户行为分析、市场细分等领域。

1.4.9 PCA 算法

主成分分析(principal component analysis，PCA)算法是一种常见的数据降维算法，主要用于高维数据的分析和可视化[8]。其核心思想是将高维数据转化为低维数据，同时尽可能地保留原始数据的信息。具体而言，PCA 算法通过线性变换将原始数据映射到一个新的坐标系中，使得数据在新坐标系下具有最大的方差，即尽可能分散在新坐标系的各个方向上。这些新坐标轴被称为主成分，其数量通常少于原始数据的维度。PCA 算法的步骤包括计算数据的协方差矩阵、求解协方差矩阵的特征值和特征向量、选取前 k 个最大特征值对应的特征向量作为主成分，最后将数据映射到主成分上。PCA 算法可以应用于数据压缩、数据可视化、降噪、特征提取等领域。在机器学习中，PCA 算法可以作为预处理步骤，用于减少特征数量和相关性，从而提高模型的精度和泛化能力。

1.4.10 关联规则学习算法

关联规则学习(Apriori)算法是一种用于挖掘频繁项集的算法，它可以从一个事务数据库中发现频繁出现的项集[9]。该算法的基本思想是利用频繁项集的性质，即如果一个项集是频繁出现的，则它的所有子集也必须是频繁出现的。Apriori 算法采用了一种迭代的方法，每次迭代都产生一些候选项集，并计算它们的支持度，然后根据最小支持度过滤掉不满足要求的候选项集，最终得到频繁项集。Apriori 算法的实现过程通常包括以下几个步骤。

(1) 扫描整个事务数据库，统计每个项集的支持度，得到 1-项集的集合 L1。

(2) 根据 L1 生成 2-项集的候选集 C2，计算其支持度，筛选出满足最小支持度要求的项集，得到 2-项集的集合 L2。

(3) 根据 L2 生成 3-项集的候选集 C3，计算其支持度，筛选出满足最小支持度要求的项

集，得到 3-项集的集合 L3。

（4）重复上述步骤，直到不能再生成满足最小支持度要求的项集为止。

Apriori 算法的优点是简单易实现，并且可以处理大规模数据集。然而，该算法也存在一些缺点，包括计算频繁项集的代价较高，以及可能会产生大量的候选项集。近年来，一些改进算法，如 FP-growth 算法[10]、Eclat 算法[11]等也被提出来，用于提高频繁项集挖掘的效率。

1.5 机器学习的一般流程

机器学习的一般流程通常包括以下步骤。

（1）确定目标。机器学习的第一步是确定要解决的问题，这包括对问题进行理解和定义，例如，确定问题是分类、回归还是聚类等。

（2）数据收集。收集与问题相关的数据是机器学习的第一步。数据可以有各种来源，如数据库、文件、传感器等。机器学习对于数据质量有着高要求，因为它对机器学习模型的性能至关重要。因此，确保数据的准确性和完整性非常重要。

（3）数据预处理。在将数据用于机器学习之前，通常需要对数据进行预处理。这包括数据清洗、处理缺失值、特征选择和转换等步骤。预处理的目的是准备适用于机器学习算法的数据集。

（4）特征工程。特征工程是一个重要环节，它涉及选择哪些特征用于训练模型，并对这些特征进行适当的变换。特征工程的目标是提取出最具信息量的特征，以改善模型的性能。

（5）数据划分。通常将数据集分为训练集、验证集和测试集。训练集用于训练模型的参数，验证集用于调整模型的超参量，测试集用于评估模型的性能。

（6）选择模型。选择适合问题的机器学习模型。常见的模型包括线性回归、决策树、随机森林、神经网络等。选择合适的模型取决于问题类型和数据特性。

（7）模型训练。使用训练集来训练所选模型。在训练过程中，模型会根据输入数据不断调整其参数，以拟合数据并学习关系。

（8）模型评估。使用验证集来评估模型的性能。常见的性能指标包括准确度、精确度、召回率、F1 值等，具体指标取决于问题类型。

（9）超参量调优。调整模型的超参量，以改善模型的性能。这可以通过交叉验证等技术来实现。

（10）模型部署。模型一旦满足性能要求，可以将训练好的模型部署到生产环境中，以用于实际应用。部署可以在云端、移动设备或嵌入式系统中进行。

（11）模型监测和维护。模型部署后，需要定期监测模型的性能，并根据需要进行维护和更新。这可能涉及重新训练模型，使之适应新数据。

这些步骤构成了机器学习的一般流程，但实际应用中，根据问题的复杂性和特性，可能需要进行一些调整和变化。机器学习是一个迭代的过程，需要不断改进和优化模型，以获得更好的结果。机器学习的一般流程如图 1.5 所示。