第5章 深度学习基础知识

本章学习目标

- 掌握深度学习的基础概念；
- 了解深度学习在机器学习中的地位和作用；
- 掌握组合不同算法构建机器学习算法的能力；
- 了解监督学习和无监督学习的主要算法。

从20世纪80年代开始，统计机器学习开始逐渐成为机器学习的主流发展方向，并使得人工智能从早期纯粹的模型和理论研究发展为可以解决现实生活问题的应用研究。随着计算机性能的提升和大数据时代的来临，机器学习在硬件的支持上取得了重大突破，并且催生了很多新的理论，机器学习便是其中一个重要的分支。深度学习中的很多理论和基础来自于统计机器学习，在掌握深度学习之前有必要对机器学习的基本原理有所理解。机器学习要解决的问题是，基于数据构建合理的统计模型，并利用该模型对数据进行分析和预测。本章将主要对与深度学习相关的基础知识进行讲解。

5.1 学习算法

让机器像人类一样学习关于世界的知识往往面临着诸多挑战。例如，在人工智能图像识别领域中便存在一种称作语义鸿沟的挑战：对人类来说，从图像中识别一个对象轻而易举；对计算机来说，图像识别却是一项极具挑战性的工作。因为在计算机的“视觉”中，图像是由大量的三维数组表示的，对于人来说一眼就能识别出图像中的对象，机器却需要将数百万个数字映射到一个标记来完成对该图像的识别。同一个物体不同的角度和光照在图像中的变化，如图5.1所示。

图5.1 不同的角度和光照下的同一件雕塑

在图 5.1 中，通过左图和中图的对比可以看出，同一个对象由于拍摄的角度不同，图像的形状发生了巨大的变化，如何让机器知道不同的角度拍摄出的图像其实是同一个对象显然是一项巨大的挑战。左图和右图相比，虽然拍摄角度相同，但光影效果不同，而光照会使像素值的大小产生巨大的变化，而在机器学习中，如何处理光照对图像识别的巨大影响同样是一项艰巨的任务。由此可见，机器的认知方式与人类存在着巨大的差异。

米切尔教授对机器学习的定义为："对于某类任务(Task，简称 T)和某项性能评价准则(Performance，简称 P)，如果一个计算机程序在 T 上，以 P 作为性能的度量，随着积累经验(Experience，简称 E)不断自我完善，那么称这个计算机程序从经验 E 中学习了。"对于计算机系统而言，通过运用数据及某种特定的方法(比如统计的方法或推理的方法)，来提升机器系统的性能，就是机器学习。任务 T、评价准则 P 和经验 E 这 3 个要素构成了机器学习的主题，但它们的定义非常宽泛，因此本书将通过有限的直观示例对这 3 个概念进行解释，帮助大家了解机器学习中的这 3 个要素。

5.1.1　任务 T

任务 T 一般定义为机器学习系统处理样本的过程。样本是指从希望计算机学习的特定对象或事件中收集到的已经量化的特征的集合。通常将样本表示成一个向量 $\boldsymbol{x}\in\mathbf{R}^n$，向量的每一个元素 x_i 是一个特征。例如，一幅图片的特征通常是指构成该图片的像素。很多初学者有时会将学习的过程误认为是任务 T。实际上，学习是为了获取完成任务的能力。例如，通过学习让计算机识别图片中的水果，任务 T 是指识别图片中的水果。

机器学习可以完成很多类型的任务，下面列举几个常见的机器学习任务。

- 回归：主要是预测数值型的数据，如图 5.2 所示。对于一组样本 x，通过函数 f 计算，有一组正确的输出 y，模型通过函数 f' 计算得出输出 y'，通过比较输出 y' 和输出 y 改进函数 f'，使其接近于真实函数 f。
- 分类：在这类任务中，计算机程序需要将实例数据划分到合适的分类中，如图 5.3 所示。例如，在图像识别领域，计算机可以像人类一样识别图中的不同类型的物体。

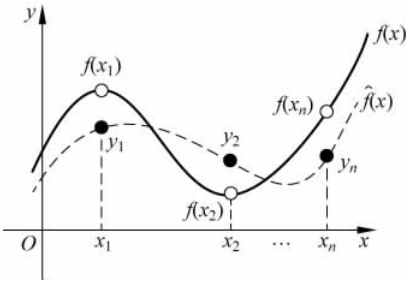


图 5.2　回归任务示意图

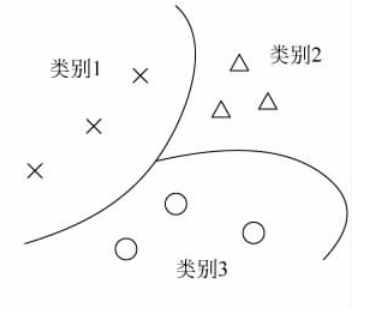


图 5.3　分类任务示意图

- 异常检测：这类任务主要是寻找输入样本中所包含的异常数据，如图 5.4 所示。计算机程序通过对一组事件或对象进行筛选，标记出异常或非典型的个体。若已知异常数据，则与有监督的分类类似；通常在不知道异常数据特征的情况下，采用密度估计的方法来剔除偏离密度中心的数据。例如，QQ 异地登录时，系统通过对用户

经常登录的 IP 地址建模，当系统检测到登录地址远离平时经常登录地址时，就会提示登录异常。

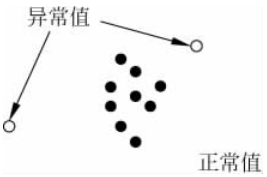


图 5.4　异常检测任务示意图

- 聚类：聚类属于模式识别问题，将不同数据归于相应的簇中，如图 5.5 所示。聚类与分类相似，但是聚类任务中只有输入，并且用簇代替了分类任务中的类别，这是一种无监督学习。
- 降维：在这类任务中，一般从高维度数据中提取关键信息，将高维度问题转换为易于计算的低维度问题求解，如图 5.6 所示。如果输入和输出均已知，则属于监督学习；若只有输入已知，则属于无监督学习。在将样本降维时，应保持原始输入样本的数据分布特征，以及数据间的相邻关系。

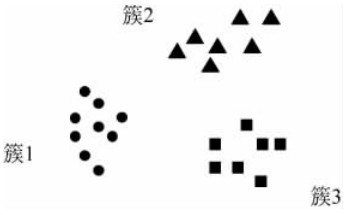


图 5.5　聚类任务示意图

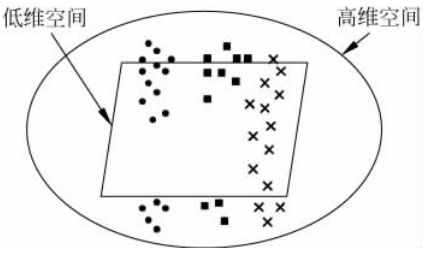


图 5.6　降维任务示意图

5.1.2　性能度量 P

性能度量(Performance Measure)反映了任务需求，通过为机器学习设定性能的度量来评估学习算法的效果，即对完成任务 T 的能力进行度量。一般情况下，性能度量 P 是根据所要完成的任务 T 来制定的，在对比不同模型的能力时，不同性能度量往往对评判结果产生重大影响，这意味着模型的“好坏”是相对的。判断模型“好坏”的标准不仅取决于算法和数据，还取决于任务需求，因此与系统理想表现相匹配的性能度量对训练模型是十分重要的。

通常用以下几种方法来度量模型的性能度量：

- 错误率/精度(accuracy)；
- 准确率(precision)/召回率(recall)；
- P-R 曲线，F1 度量；
- ROC 曲线/AUC(最常用)；
- 代价曲线。

例如，分类任务通常以模型的准确率来作为性能评价的准则。准确率是指该模型输出正确结果的样本比例，反之，错误率也可以作为性能评价准则。一般将错误率称为 0-1 损失的期望。在一个特定的样本中，如果结果是正确的，那么 0-1 损失为 0，否则为 1。然而，对于密度估计这类任务来说，评价结果的准确率、错误率或者 0-1 损失是没有意义的。针对不同的模型需要使用不同的性能评价准则。

为了对算法在实际应用中的性能进行评估，通常会用测试数据来评估系统性能，这里的

测试数据与训练时使用的数据是相互分开的。

5.1.3　经验 E

根据学习过程中的不同经验，机器学习可以粗略分为无监督学习和监督学习。

本书中的大部分学习算法可以理解成在整个数据集中获取经验。数据集是指很多样本组成的集合，样本也被称作数据点(Data Point)。例如，电影推荐系统中往往需要用到的电影数据集，其中每部电影都对应一个样本，每个样本的特征属性可以是该电影的导演、演员、票价和上映时间等。

5.1.4　人工神经网络

芬兰计算机科学家 Teuvo Kohonen 教授对人工神经网络的定义为："人工神经网络，是一种由具有自适应性的简单单元构成的广泛并行互联的网络，它的组织结构能够模拟生物神经系统对真实世界所做出的交互反应。"

通常在机器学习中提到的"神经网络"，实际上是指"神经网络学习"。人工神经网络的学习方法中的连接主要通过编写一个初始模型，然后通过数据训练，不断改善模型中的参数，直到输出的结果符合预期，便实现了机器学习。在网络层次上模拟人的思维过程中的某些神经元的层级组合，用人脑的并行处理模式来表征认知过程。这种受神经科学启发的机器学习方法，被称人工神经网络(Artificial Neural Network，ANN)。

5.1.5　反向传播算法

在神经网络(甚至深度学习)参数训练中反向传播算法(Back Propagation，简称 BP 算法)具有非常重要的意义。1974 年，Paul Werbos 在他的博士论文中，首次提出了通过误差的反向传播来训练人工神经网络。Werbos 不仅是 BP 算法的开创者，还参与了早期的循环神经网络(Recurrent Neural Network，RNN)的开发。

BP 算法虽然称为反向传播，但事实上是一个典型的双向算法，其工作流程分为以下两个部分：

(1) 正向传播输入信号，输出分类信息(对于有监督学习而言，基本上都可归属于分类算法)。

(2) 反向传播误差信息，调整全网权值(通过微调网络参数，让下一轮的输出更加准确)。

本章只对反向传播算法进行初步的介绍，后续章节会进一步深入讲解有关内容。

5.1.6　M-P 神经元模型

除了人工神经网络方法，在人工智能领域还有一个"仿生派"，它就是 20 世纪 40 年代提出并一直沿用至今的"M-P 神经元模型"，即模仿生物的某些特性，复现这些对象的特征。发展到今天的神经网络与深度学习更接近于"仿生派"的理念——模拟大脑神经元的工作机理。

在这个模型中，神经元接收来自 n 个其他神经元传递过来的输入信号，这些信号通常通过神经元之间连接的权重(weight)大小来表示，神经元将接收到的输入值按照某种权重叠加起来，并将当前神经元的阈值进行比较，然后通过"激活函数"(activation function)向外表

达输出(这在概念上就叫感知机)。

5.1.7 激活函数

激活函数对于人工神经网络模型去学习、理解非常复杂和非线性的函数来说具有十分重要的作用,简单来说,激活函数的作用是在神经网络中引入非线性因素。一些复杂的事情相互之间往往存在着许多隐藏层的非线性问题,对这些非线性问题的处理将有助于了解复杂的数据。在神经网络中加入非线性激活函数可以在由输入到输出转化时生成非线性映射以适应复杂的模型。

一个没有激活函数的神经网络在大多数情况下会因为过于简单而无法用于解决复杂的实际问题。这是因为,没有激活函数的神经网络将只能进行线性变换,多层神经网络的输入叠加依然是线性变换,充其量是通过复杂的线性组合来逼近曲线。激活函数在神经网络中引入了非线性因素,在每层神经网络线性变换后,添加一个非线性激活函数对这种线性变换进行转换,使其变成非线性函数,以应用于复杂的实际应用中。

常见的激活函数主要有 Sigmoid 函数、Tanh 函数、ReLU 函数、Softmax 函数等,具体会在后续章节中进行介绍。

5.2 容量与拟合

5.2.1 机器学习中的泛化

在训练模型时,通常希望模型能够从训练集中学到适用于所有潜在样本的"普适规律",从而在处理未观测到的数据时取得良好效果,而不仅仅是在训练集上取得理想的效果。在未观测到的输入上取得良好效果的能力被称为泛化(generalization)。

一般情况下,当训练机器学习模型时,可以访问训练集,在训练集上计算度量误差,被称为训练误差(training error)。机器学习与优化的不同之处在于,优化只强调改善模型在训练数据集上的表现,即只关注降低模型的训练误差。而机器学习除了关注模型的训练误差,还关注模型的测试误差,即模型在处理测试数据时的误差期望。

训练误差与测试误差之间可直接观测到的联系之一是:随机模型训练误差的期望和该模型测试误差的期望是一样的。假设有概率分布 $P(x,y)$,从中重复采样生成训练集和测试集。对于某个固定的 w,训练集误差的期望恰好和测试集误差的期望一样,这是因为这两个期望的计算采用的都是相同的数据集生成过程。

在使用机器学习算法时,通常不会在数据采样前对参数进行固定,而是先在训练集上进行采样,然后优化参数降低模型的训练误差,最后在测试集上采样获取测试误差。在这个过程中,测试误差期望会大于或等于训练误差期望。以下是决定机器学习算法取得良好效果的两点主要因素:

(1) 低训练误差;

(2) 训练误差和测试误差的差距小。

然而当学习算法在训练集中取得"理想"的结果时,很可能已经把训练样本自身的一些特点当作了所有潜在样本都会具有的一般性质,这样就会导致泛化性能下降,这种现象被称

为“过拟合”(Overfitting)。与“过拟合”相对的是“欠拟合”(Underfitting)。“欠拟合”是指学习算法没有充分学习到训练样本中的一般性质。关于过拟合与欠拟含的直观类比如图 5.7 所示。

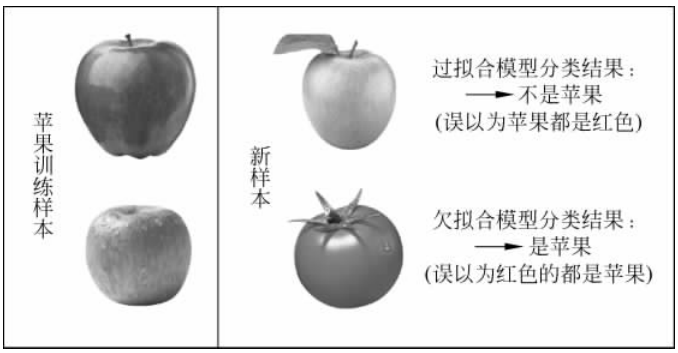


图 5.7　过拟合与欠拟含的直观类比

诸多因素可能导致模型的过拟合，其中最常见的情况是由于模型的学习能力过于强大，以至于把训练样本所包含的不太一般的特性都学到了，而欠拟合则通常是由于学习能力低下造成的。与过拟合相比，欠拟合问题更容易被克服。

5.2.2　过拟合

机器学习表现不佳的原因往往与过拟合或欠拟合数据有关。过拟合指的是模型对于训练数据拟合程度过当的情况。通俗来说，过拟合可以理解成一个模型通过学习获得了很强的应试能力，却无法将应试能力应用于考试以外的领域。若某个模型过度地学习了训练数据中的细节和噪声，以至于模型在新的数据上表现很差，则称过拟合发生了。这意味着训练数据中的噪声或者随机波动也被当作概念被模型学习了。而问题就在于这些概念不适用于新的数据，从而导致模型泛化性能变差。

过拟合更可能在无参数非线性模型中发生，因为学习目标函数的过程是易变的、具有弹性的。同样，许多无参数机器学习算法也包括限制约束模型学习概念多少的参数或者技巧。

例如，决策树就是一种无参数机器学习算法，非常有弹性并且容易受过拟合训练数据的影响。这种问题可以通过对学习后的树进行剪枝来解决，这种方法就是为了移除一些学习到的细节。

过拟合是机器学习面临的关键障碍，各类学习算法都必然带有一些针对过拟合的措施；然而，在机器学习领域过拟合是在所难免的，在训练模型中所能做的只是“缓解”或者减小其带来的风险。

5.2.3　欠拟合

欠拟合是指模型在训练和预测时的表现都不好的情况。一个欠拟合的机器学习模型不是一个良好的模型。欠拟合通常可以直观地在训练数据上表现出来而不被讨论，因为在给定一个评估模型表现的指标的情况下，欠拟合很容易被发现。矫正方法是继续学习并且试着更换机器学习算法。虽然如此，欠拟合与过拟合仍然形成了鲜明的对照。

理想的模型拟合状态应处于欠拟合和过拟合之间，这是优化模型的理想情况，但实际操作中很难平衡模型的欠拟合与过拟合状态。为了理解这个目标，可以观察正在学习训练数据机器学习算法的表现，把这个训练的过程划分为训练过程和测试过程。

随着算法的不断学习，模型在训练数据和测试数据上的误差会有所下降，但是，过长的学习时间可能会让模型在测试数据上的表现下变差。此时，模型可能已经处于过拟合状态，学习到了训练数据中的不恰当细节或噪声，在测试数据集上的错误率开始上升，即模型的泛化能力开始下降。因此，判断训练模型的临界点时，可以选择模型在测试集中的泛化误差刚开始上升时。此时模型在训练集和测试集上的表现都处于良好的状态。但是，这种把控停止训练时机的方法在实践中难以操作。因为在测试数据上运用这个方法时便意味着测试数据集对于模型的训练者来说并不是“未知的”，此时可能会受到人为影响而泄露测试数据的一些相关知识进而对模型的保真性产生影响。

5.2.4 没有免费的午餐定理

在现实任务中，面对同一种需求往往可以采用多种学习算法，而同一种算法也会因为参数配置的不同而生成不同的模型。在模型选择中往往希望用最合适的学习算法和参数配置来解决问题。模型选择该问题的最理想解决方案是对候选模型的泛化误差进行评估，然后选择泛化误差最小的模型。不过，模型的泛化误差通常是无法事先获取的，训练误差由于很容易受过拟合的影响，因此并不适合作为评估标准。

学习理论表明，通过机器学习算法能够从有限个训练集样本中学到泛化的能力。然而通过从一组有限的样本中推断出普适性的规则，显然是具有局限性的。在逻辑推断中，如果想用一个一般性的规则去描述集合中的所有元素，那么该推断中必须具有集合中每个元素的信息，这样的规则会显得十分复杂。在机器学习领域，一般通过概率法则来应对这一问题，从而找到一个对绝大多数样本适用的正确规则，进而避免使用纯逻辑推理得出一个确定性的规则。

上述方法也并不完善，因为它并不能完美解决由有限数据集推断一般性规则的严谨性问题。机器学习的没有免费的午餐定理(No Free Lunch theorem，NFL)表明，没有任何方法可以保证一种机器学习算法在任何情况下总是比其他算法的表现更优秀。即，不存在一个与具体应用无关的、普遍适用的“最优分类器”；学习算法必须要做出一个与问题领域有关的“假设”，分类器必须与问题域相适应。

NFL 定理的前提是，所有问题出现的机会相同，或所有问题同等重要。但实际情况中很难出现这样的情景，在现实中得到的数据、分布情况以及要解决的问题往往都是特定的，因此只需要具体问题具体分析就可以了，而不需要考虑该模型是否能够在解决除该问题以外的其他问题时同样优秀，这样可以让模型更加高效地学习，模型的效果也更好。

机器学习研究的目标不是找一个任何情况都适用或是绝对最优的学习算法，而是需要研究和创造更多的学习算法来应对不同的情况。

5.3 评估方法

模型的选择一般通过测试来对学习模型的泛化误差评估来决定。通过“测试集”(Testing Set)来对模型的学习效果在新样本中的表现进行测试，然后根据测试集上的“测试

误差”(Testing Error)评估模型的泛化能力。通常会假设测试样本也是从样本真实分布中独立同分布采样得到的,测试集应该与训练集相互独立(Independent)、同分布(Identically Distributeb),倘若测试集和训练集不是相互独立同分布的,那么就无法通过机器学习习得一个“合适”的模型,无法准确地对未观测数据进行预测。

需要注意避免在训练集中出现测试样本。不妨设想如下场景:如果将书本的课后习题作为考试的试题,那么考试成绩是否能够有效反映出学习效果呢?显然考试成绩无法真实地反映学生的学习效果,因为训练的题目和测试的题目是一模一样的,只需记忆答案就可以考出较理想的分数,而无法通过这样的测试了解到学生对所学知识的“泛化能力”。

得到泛化性能优秀的模型,就像在通过“1+1=2”学会了加法以后,“举一反三”学会求“2+2”或者“2+4”等其他加法运算的值。训练样本相当于平时学习的课后习题,测试则相当于单元考试。如果测试样本与训练样本相同,那么很可能造成对学习效果错误的评估。

接下来介绍几种常见的评估方法。

1. 留出法

“留出法”(Hold-out)直接将数据集 D 划分为两个互斥的集合 A 和集合 B。将集合 A 作为训练集 S,集合 B 作为测试集 T,即 $D=A\cup B, A\cap B=\varnothing$。在训练集 S 中训练模型,然后用测试集 T 来评估泛化误差。在划分的时候既要保证两个集合相互独立,也要尽可能地保证这两个数据集数据分布的一致性,避免在划分过程中引入额外的偏差而影响最终结果。

为了保证数据分布的一致性,通常采用“分层采样”(Stratified Sampling)的方式来对数据进行采样。假设的数据中有 m 个正样本、n 个负样本,训练集 S 占数据集 D 的比例为 p,测试集 T 占训练集 D 的比例为 $1-p$,可以通过在 m 个正样本中采 $m\times p$ 个样本作为训练集中的正样本,而通过在 n 个负样本中采 $n\times p$ 个样本作为训练集中的负样本,其余的作为测试集中的样本。若训练集 S 和测试集 T 中样本类别比例差别很大,则会由于训练集数据与测试集数据分布的差异而产生误差。

数据集的错误率求值公式为:

$$\text{错误率} = (\text{错误样本数} / \text{样本总数}) \times 100\%$$

训练精度求值公式为:

$$\text{精度} = 1 - \text{错误率}$$

以照片中的人脸识别为例,正样本为人脸的图片,负样本为人脸周围的环境,负样本的选取往往与任务和场景有关,不能选取与任务毫无关联的内容作为负样本。假设数据集 D 中含有 200 个样本,其中训练集 S 包含 120 个样本,另外 80 个样本划分为测试集 T。模型在被训练集 S 训练后,如果在测试集 T 上有 24 个样本分类错误,那么其错误率为:(错误样本数/样本总数)×100%=20%,相应地,精度为 1−20%=80%。

值得注意的是,即使在给定“训练集/测试集”的样本比例后,仍有多种方法对初始数据集 D 进行分割。例如在上述例子中,将数据集 D 中的样本排序,然后把排序靠前的 120 个正例放到训练集中,把排序靠后的 80 个正例放到测试集中,不同的划分比例将导致不同的训练集与测试集的比值,相应的模型评估的结果也会有差别。因此,单次使用留出法得到的估计结果往往不能准确反映实际结果,在使用留出法时,一般要采用若干次随机划分、重复进行实验评估后取平均值作为留出法的评估结果。例如,进行 100 次随机划分,每次产生一个训练/测试集用于实验评估,100 次后就得到 100 个结果,而留出法返回的则是这 100 个

结果的平均。留出法将数据集划分成训练集和测试集：若令训练集 S 包含绝大多数样本，则训练出的模型可能更接近于用初始数据集 D 训练出的模型，但由于测试集 T 比较小，评估结果可能不够稳定准确；若令测试集 T 包含较多样本，则会增加训练集 S 与初始数据集 D 差异，被评估的模型与用初始数据集 D 训练出的模型相比可能有较大差别，从而降低了评估结果的保真性(fidelity)，这个问题难以调和。通常将样本中的 60%～80%用于训练，剩余样本用于测试。

2. 交叉验证法

交叉验证法先将数据集 D 通过分层采样分成 N 个大小相似的互斥子集，每个子集尽可能保证数据相互独立和分布一致性，每次用 $N-1$ 个子集的并集作为训练子集，剩下的一个子集作为测试集，通过这样的方法得到 N 组训练/测试集。在 N 次训练和测试后返回这 N 次测试结果的均值。

在进行交叉验证时，其结果的稳定性和保真性在很大程度上受到了 N 的取值的影响。如果在 N 次测试后，测试集的误差很小，则说明模型可能存在问题。一个小规模的测试集意味着平均测试误差估计的统计不确定性，使得很难判断算法 A 是否比算法 B 在给定的任务上做得更好。

当数据集样本非常大时，交叉验证的稳定性和保真性相对更可靠。在数据集样本过少时，可以通过替代方法允许使用所有的样本估计平均测试误差，但计算量有所增加。这些过程是基于在原始数据上随机采样或分离出的不同数据集上进行重复训练和测试的想法。最常见的是 k-折交叉验证过程，例如，将数据集分成 k 个不重合的子集。测试误差可以估计为 k 计算后的平均测试误差。在第 i 次测试时，数据的第 i 个子集用于测试集，其他的数据用于训练集。由此带来的一个问题是不存在平均误差方差的无偏估计，但是通常会使用近似的方法来解决。

5.4 偏差与方差

在机器学习中，不仅需要知道通过实验估计学习算法的泛化性能的方法，也要知道一个学习算法为什么具有这样的性能。在机器学习领域，通常采用“偏差-方差分解”(bias-variance decomposition)的方法来对学习算法泛化能力进行解释。偏差与方差的关系如图 5.8 所示。

偏差(bias)：用于描述根据训练样本拟合出的模型输出的期望预测与样本真实结果的差距，即算法的样本拟合状态。在偏差上想要取得良好的效果，就像射手射出的箭尽想要可能命中靶心区域，对应图 5.8 中的低偏差(low bias)部分。想要降低偏差，就需要构建复杂化的模型——模型参数增加，而过多的模型参数容易引起过拟合。出现过拟合的情况对应图 5.8 中的高方差(high varience)部分，这种情况就像射手在射箭时太想射中靶心而用力过猛导致手发抖，最终因为射箭时受到较大扰动而导致着箭点分布过于分散。

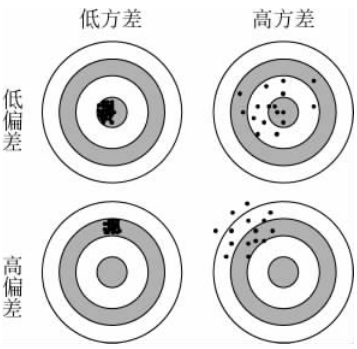


图 5.8 偏差-方差分解的类比

方差(varience)：用于描述根据训练样本得到的模型在测试集中的表现的变化，即刻画了数据扰动所造成的影响，在方差中取得良好的表现对应图 5.8 中的低方差(low warience)，这就需要简化模型，降低过多的参数带来的过拟合的可能性，但这样也容易产生欠拟合，出现欠拟合的状态类似于图 5.8 中的高偏差部分，相当于着箭点虽然分布密集但是却偏离了靶心区域。

泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。为了保证模型能够取得良好的泛化性能，需要使偏差尽可能小，以充分拟合数据，并且使方差较小，以减少数据扰动产生的影响。

偏差与方差的取舍往往是有冲突的，这称为偏差-方差窘境(bias-variance dilemma)。假设给定一个学习任务，在训练不足时，模型的拟合能力不足，训练数据的扰动不足以使模型产生显著变化，此时偏差将主导泛化错误率；随着训练程度的加深，模型的拟合能力逐渐增强，训练数据发生的扰动渐渐能被模型学到，方差逐渐主导了泛化错误率；在训练程度充足时，模型的拟合能力已非常强，训练数据发生的轻微扰动都会导致模型发生显著变化，此时若模型学习到了训练数据中的非全局特性，便会发生过拟合。具体如图 5.9 所示。

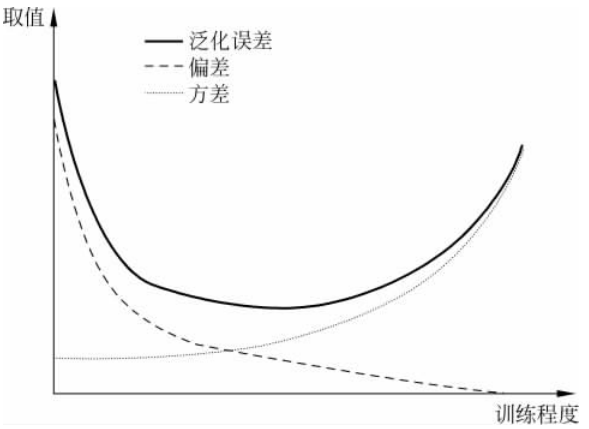


图 5.9　泛化误差与偏差、方差的关系

5.5　监督学习算法

监督学习可以简单理解成给定一组输入数据集 x 和输出数据集 y，让模型习得如何关联输入和输出。通常输出 y 很难自动收集，必须由人来进行“管理”。

线性模型(Linear Model)是机器学习中的一类算法的统称，其形式化定义为：通过给定的样本数据集 D，线性模型试图学习到对于任意的输入特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，模型的预测输出 $f(\boldsymbol{x})$ 都能够表示为输入特征向量 $\boldsymbol{x}$ 的线性函数，即满足：

$$f(\boldsymbol{x})=w_1x_1+w_2x_2+\cdots+w_nx_n+b$$

上式可以用矩阵表达式进行表示，具体如下：

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$$

其中 $\boldsymbol{w}=(w_1,w_2,\cdots,w_n)^{\mathrm{T}}$ 和 b 为模型的参数(parameter)，参数是控制系统行为的值，在确定了 $\boldsymbol{w}$ 和 b 的值后，模型便可以确定了。$\boldsymbol{w}$ 可以看作是一组决定每个特征如何影响预测结

果的权重(weight),由于权重 $\boldsymbol{w}$ 可以比较直观地表达各属性在预测结果中的重要性,因此线性模型具有良好的可解释性(comprehensibility)。如果特征 x_i 对应的权重 w_i 值为正,则特征值增加,预测值 $f(\boldsymbol{x})$相应增加;如果特征 x_i 对应的权重 w_i 值为负,则特征值减少,预测值 $f(\boldsymbol{x})$相应减少。特征权重的值越大,对预测值 $f(\boldsymbol{x})$的影响就越大;特征权重的值为零,说明它对预测值 $f(\boldsymbol{x})$没有影响。

线性模型属于非常基础的机器学习模型结构,主要应用于分类、回归等学习任务中,很多非线性模型也是基于对线性输出结果的非线性变换和层级叠加等操作后得到的。常见的线性模型主要有线性回归、单层感知机和 Logistic 回归。本节将只对线性回归和 Logistic 回归进行讲解,有关单层感知机的内容将在后续章节中单独讲解。

5.5.1 线性回归

回归(Regression)是监督学习任务的一种,形式化定义为:设定由 m 个训练样本数据构成的数据集 D:

$$D=\{(\boldsymbol{x}^1,y^1),(\boldsymbol{x}^2,y^2),\cdots,(\boldsymbol{x}^m,y^m)\}$$

其中,$\boldsymbol{x}^i=(x_1^i,x_2^i,\cdots,x_n^i)$表示第 i 个训练数据的输入特征向量,$y^i\in\mathbf{R}$,回归分析的任务是通过训练数据集 D 学习到一个模型 T,使得模型 T 能够尽量拟合训练数据集 D,并且对于新的输入数据 $\boldsymbol{x}$,应用模型 T 能够得到预测结果 $f(\boldsymbol{x})$。回归与分类是监督学习的两种形式,它们的概念很接近,唯一的区别在于:回归的预测值是一个连续的实数,而分类任务的预测值是离散的类别数据。

线性回归是回归学习的一种策略,模型试图通过对训练集 D 的学习,使得输入 $\boldsymbol{x}$ 和预测的输出 $f(\boldsymbol{x})$之间具有 $f(\boldsymbol{x})=w_1x_1+w_2x_2+\cdots+w_nx_n+b$ 的线性关系。要确定模型 T,需要确定参数 w 和 b。求解参数首先需要定义一种衡量标准,用于衡量模型的预测值 $f(\boldsymbol{x})$与准确值 y 之间的差距,即对损失函数进行定义。在回归任务中,常用的损失函数是均方误差,表达式如下所示:

$$L(w,b)=\frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}^i)-y^i)^2$$

通过均方误差最小化来求解模型的方法也被称为最小二乘法(Ordinary Least Square Method,OLS)。均方误差有着良好的几何意义,它对应常用的欧几里得距离,即“欧氏距离”(Euclidean distance)。最小二乘法的思想是尝试寻找一条直线,使得训练数据集 D 中的所有样本点到超平面的欧氏距离之和最小,如图 5.10 所示。

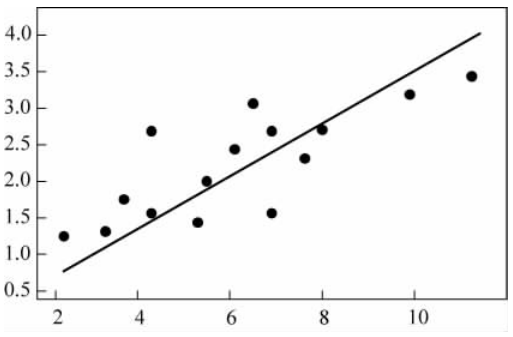


图 5.10 最小二乘法

求解参数 w 和 b 使均方误差最小化的过程称为线性回归模型的最小二乘“参数估计”(parameter estimation)。可以直接利用解析法来求解均方误差中的最小化问题,即根据极值存在的必要条件,对损失函数的参数 w 和 b 进行求导得到参数方程组,令参数方程组等于 0,将最优化问题转化为求解方程组问题。之所以采用解析法来求解,是因为线性回归模型相对简单,在数据量不大且满足一定条件的情况下采用解析法来求解会更加高效,但是在其他情况下采用解析法求解最优化问题是不可行的。

首先,将参数 w 和 b 合并成向量 $\boldsymbol{\theta}$,向量 $\boldsymbol{\theta}$ 满足如下表达式:

$$\boldsymbol{\theta} = (w_1, w_2, \cdots, w_n, b)^{\mathrm{T}}$$

$\boldsymbol{\theta}$ 是一个 $(n+1)$ 维向量,数据集 D 第 i 个训练数据的输入特征向量为:

$$\boldsymbol{X}^i = (x_1^i, x_2^i, x_3^i, \cdots, x_n^i, 1)^{\mathrm{T}}$$

数据集 D 的输入特征向量的矩阵表示形式如下:

$$\boldsymbol{X} = \begin{pmatrix} (\boldsymbol{X}^1)^{\mathrm{T}} \\ (\boldsymbol{X}^2)^{\mathrm{T}} \\ \vdots \\ (\boldsymbol{X}^m)^{\mathrm{T}} \end{pmatrix}$$

$\boldsymbol{X}$ 是一个大小为 $m \times (n+1)$ 维的矩阵,训练数据的预测输出值满足 $f(\boldsymbol{X}) = \boldsymbol{\theta X}$。设,输出数据 $\boldsymbol{Y}$ 满足 $\boldsymbol{Y} = (y^1, y^2, \cdots, y^m)^{\mathrm{T}}$。经过新的符号定义,均方误差函数的矩阵表达式如下所示:

$$\boldsymbol{L}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta X} - \boldsymbol{Y})^{\mathrm{T}}(\boldsymbol{\theta X} - \boldsymbol{Y})$$

利用极值存在的必要条件,对上式的参数 $\boldsymbol{\theta}$ 求导。

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \frac{1}{2}(\boldsymbol{\theta X} - \boldsymbol{Y})^{\mathrm{T}}(\boldsymbol{\theta X} - \boldsymbol{Y}) \\
&= \frac{1}{2}\nabla_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta X}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\theta Y}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) \\
&= \frac{1}{2}\nabla_{\boldsymbol{\theta}}\mathrm{Tr}(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta X}^{\mathrm{T}}\boldsymbol{X} - \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\theta Y}^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}) \\
&= \frac{1}{2}\nabla_{\boldsymbol{\theta}}(\mathrm{Tr}(\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{\theta}) - 2\mathrm{Tr}(\boldsymbol{\theta Y}^{\mathrm{T}}\boldsymbol{X})) \\
&= \frac{1}{2}(2\boldsymbol{\theta X}^{\mathrm{T}}\boldsymbol{X} - 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}) \\
&= \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{\theta X} - \boldsymbol{Y})
\end{aligned}$$

当 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 为满秩矩阵(full-rank matrix)或正定矩阵(positive definite matrix)时,可令上式的值为 0,从而得到如下表达式:

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

上式即为解析法的求解公式,其中 $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 是矩阵 $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})$ 的逆矩阵。对于新的测试数据,其预测输出为:

$$f(\boldsymbol{X}) = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{X}$$

然而,现实任务中 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 不一定是满秩矩阵,在面对大量变量时,其测试数据的数量很可能超过训练样本的数量,这会导致 $\boldsymbol{X}$ 的列多于行数,使得 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 不为满秩矩阵,此时由于因

变量过多，可能存在多个解使得均方误差值最小化，此时需要根据学习算法的归纳偏好决定，常见的方法为引入正则化项。

在此提出两种可行的解决欠拟合问题的方法：

(1) 通过挖掘数据中不同特征之间的组合，来获取更多的特征从而避免欠拟合现象。但是，这样做会造成模型的复杂化，降低学习效率，并且这对特征的选取提出了更高的要求。

(2) 通过对线性回归进行局部加权。局部加权线性回归(Locally Weighted Linear Regression，LWR)可以看作对线性回归的一种改良。线性回归采用直线来拟合所有的训练数据，当训练数据不存在线性分布关系时，线性模型得到的结果容易出现欠拟合的现象。LWR 的原理相当简单，只需对原始线性回归损失函数添加一个非负的权重值，新的损失函数如下所示：

$$L(w,b)=\frac{1}{2}\sum_{i=1}^{m}w^i(f(\boldsymbol{X}^i)-\boldsymbol{Y}^i)^2$$

上式中添加的 w^i 即为函数的权重值，它根据要预测的点与数据集中的点的距离来为数据集中的点赋权值。某点距离要预测的点越远，其权重值越小(最小值为 0)，反之则权重值越大(最大值为 1)。权重函数的表达式为：$w^i=\exp\left(-\frac{(\boldsymbol{X}^i-\boldsymbol{X})^2}{2k^2}\right)$。该函数称为指数衰减函数，其中 k 为超参数，它的值决定了权值随距离下降的速率，该函数形式上类似高斯分布，但并没有任何高斯分布的意义。参照最小二乘法的推导过程求得该函数的回归系数，如下所示：

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

$\boldsymbol{W}$ 为对角矩阵，满足 $W_{i,i}=w^i=\exp\left(-\frac{(\boldsymbol{X}^i-\boldsymbol{X})^2}{2k^2}\right)$。

当训练数据较多时，LWR 能够取得较好的训练效果。在线性回归模型中，当通过训练得到参数 $\boldsymbol{\theta}$ 的最优解后，只保留参数 $\boldsymbol{\theta}$ 的最优解就可以得到新数据的预测输出，因此训练完成后训练数据可以被舍弃，但是，由于使用 LWR 算法训练数据时，不仅需要学习线性回归的参数，还需要学习波长参数，对于每一个预测点，都要重新依据整个数据集计算出一个线性回归模型，这使得算法代价极高。LWR 需要保留全部训练数据，因此相比之下 LWR 需要占用更大的空间。

之前讲到过机器学习的问题往往可以转化成数值最优化问题。有关最优化的算法策略，将会在本章后面进一步讲解。

5.5.2 Logistic 回归

5.5.1 节介绍了通过线性模型来进行回归学习的方法，本节将介绍通过线性模型进行分类学习的方法。Logistic 回归也被称为广义线性回归模型，属于广义线性模型，它与线性回归模型的形式基本相同，通常应用于信用评分模型，判定某个人的违约概率等。

Logistic 回归与线性回归的主要区别为：在线性回归模型中输出一般是连续的，但是对于 Logistic 回归，输入可以是连续也可以是离散的，并且 Logistic 回归的输出通常是离散的，即输出值是有限的。

Logistic 回归与线性回归模型的形式基本上相同。以二元分类为例，输出结果只有 0

或 1，即 $y\in\{0,1\}$。通过线性模型来进行分类学习，其基本思路是在空间中构造一个合理的超平面，将区域内的两类结果进行分隔。多重线性回归直接将 $ax+b$ 作为因变量，即 $y=ax+b$，而 Logistic 回归则通过 Sigmoid 函数将 $ax+b$ 对应到一个隐状态 p，其中 $p=S(ax+b)$，然后根据 p 与 $1-p$ 的大小决定因变量的值，这一点类似于阶跃函数（但阶跃函数具有不连续、不可导的特点，因此采用 Sigmoid 函数）。Sigmoid 函数如下所示：

$$f(x)=\frac{1}{1+e^{-(\boldsymbol{w}^{T}x+b)}}$$

通过 Sigmoid 函数可以将输入数据压缩到区间[0,1]内，得到的结果不是二值输出而是概率值，即当一个 x 发生时，y 被分到 0 或 1 的概率。事实上，最终得到的 y 的值是在[0,1]这个区间上的某个值，然后根据事先设定的一个阈值，通常是 0.5，当 $y>0.5$ 时，就将这个 x 归为 1 类；当 $y<0.5$ 时，将 x 归为 0 类。这个阈值是可以调整的。输入数据分别属于 0 类或者 1 类的概率如下所示。

$$P(y=1\mid x)=f(x)=\frac{e^{\boldsymbol{w}^{T}x+b}}{1+e^{\boldsymbol{w}^{T}x+b}}$$

$$P(y=0\mid x)=1-f(x)=\frac{1}{1+e^{\boldsymbol{w}^{T}x+b}}$$

Sigmoid 函数的导数形式如下所示：

$$\nabla f(x)=f(x)(1-f(x))$$

想要确定 Logistic 回归模型，需要求得其参数 $\boldsymbol{w}$ 和参数 b。Logistic 回归一般使用对数最大似然作为损失函数。

$$L(\boldsymbol{w},b)=\ln\left(\sum_{i=1}^{m}P(y_i\mid x_i;\boldsymbol{w},b)\right)$$

上式中 m 表示训练样本的个数，i 表示第 i 个样本。由于在二元分类问题中 y_i 只存在 0 和 1 两个值，因此 $P(y_i|x_i;\boldsymbol{w},b)$可以通过如下形式进行表示。

$$P(y_i\mid x_i;\boldsymbol{w},b)=(f(x_i))^{y_i}(1-f(x_i))^{1-y_i}$$

将上式代入 Logistic 回归模型的损失函数可得：

$$L(\boldsymbol{w},b)=-\sum_{i=1}^{m}(y_i\ln(f(x_i))+(1-y_i)\ln(1-f(x_i)))$$

通过上述方法将 Logistic 回归问题转换成了最小化上述损失函数的最优化问题。

5.5.3　支持向量机

支持向量机（Support Vector Machine，SVM）由 Corinna Cortes 和 Vapnik 于 1993 年提出，并于 1995 年发表，是机器学习中极具代表性的算法。接下来通过一个示例来理解 SVM。假设桌子上有两种不同颜色的球，如图 5.11 所示。

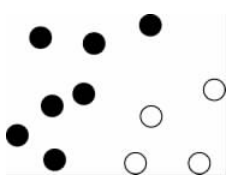


图 5.11　桌子上有两种不同颜色的球

现在用一根线绳将这两种颜色的球分隔开，如图 5.12 所示。

此时在桌上放入了更多的这两种颜色的球，可以看出，有一个球被错误地划分了，如图 5.13 所示。

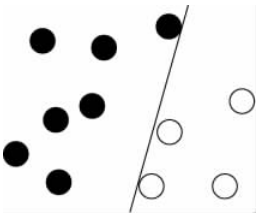


图 5.12　用一根直绳将两种不同颜色的球区分隔开

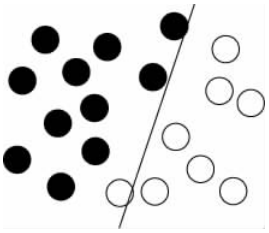


图 5.13　球变多后原来位置的直绳无法再将两类球完全分隔开

支持向量机的作用可以看作是将这根线绳摆放在桌面上最合适的位置，从而保证这两类球之间的间距最大。桌子上加入了更多的球时，支持向量机的作用就是将线绳的位置重新调整为最优分界线的位置，如图 5.14 所示。

有时会出现在平面上无法用直线将两种颜色的球分开的情况，如图 5.15 所示。

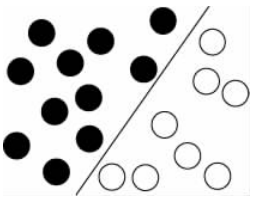


图 5.14　球变多后重新调整直绳位置将两类球完全分隔开

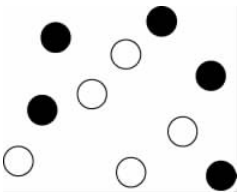


图 5.15　此时显然无法在平面内通过一根直绳将两类球完全分隔开

这种情况下 SVM 的做法类似于将桌子上的球抛起，让它们处于立体空间中，然后在立体空间中找到最合适的位置，用一个平面将两种球分隔开，如图 5.16 所示。

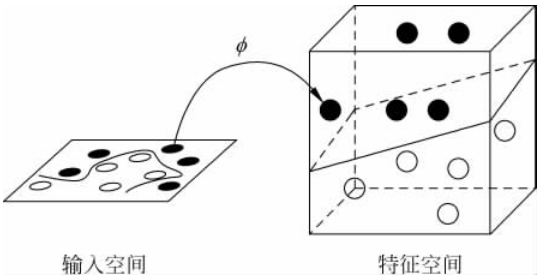


图 5.16　将球由平面抛起在立体空间中找到合适的位置将两类球分隔开

在立体空间中对两种球的分隔，在平面图上看上去是使用一条曲线将两种球分开了。

在支持向量机的概念中，上述的球称为数据，对球进行分隔的线称为分类器，最大间隙称为最优间隔，将球抛起至立体空间中称为核函数，在立体空间中对球进行分隔的平面称为超平面。本书仅需对 SVM 有一个初步了解即可，如果想要深入理解 SVM 的有关内容可以参考机器学习的有关书籍。

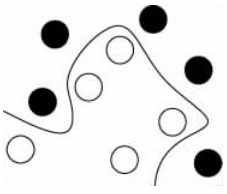


图 5.17　在平面上看到的结果

5.6　无监督学习算法

在实际生活中可能会遇到并不是为了找出某个答案而学习的情形，这种学习过程类似于无监督学习算法。例如，一个人出于爱好去欣赏古典音乐，此时，并不一定有一个明确的目的。在听了大量的古典音乐以后，可能会在这些乐曲中发现一些共性特征，比如编曲风格、演奏方式、旋律和节奏等，在随后的生活中再听到以前没听过的古典音乐时便可以通过其中的某些特征判断出音乐属于哪个流派，其创作时期以及作者。

实际上，在机器学习领域，无监督学习算法与监督学习算法之间并没有严格的区分规范和定义。在无监督学习中，样本数据的标签信息是未知的，无监督学习的目标是从数据中挖掘出数据集中包含的有用特征和规律，从而应用于未知的数据的分析和预测中的过程。

5.6.1　*K*-均值聚类

K-均值聚类也称作 *K*-means 聚类，是目前最流行的经典聚类方法之一，*K*-均值聚类算法将训练集分成 *K* 个靠近彼此的不同样本聚类，希望找出每一个样本点归属于哪个类，使得各聚类中每一个点到其对应聚类集合的中心距离的平方和最小。如图 5.18 所示。

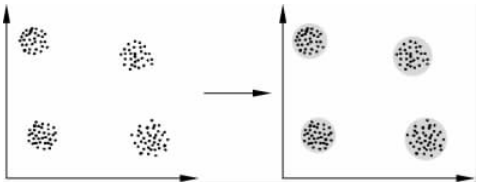


图 5.18　*K*-均值聚类

K-均值聚类先指定 *K* 个不同的聚类中心，并采用一定规则初始化它们的位置，然后迭代交换以下两个步骤直到收敛。

(1) 簇分配：将每个训练样本分配到对应的最近的中心点所代表的聚类。

(2) 移动中心：将对应聚类中心更新为归属于该中心的所有训练样本的均值处。

5.6.2　主成分分析

主成分分析(Principal Components Analysis，PCA)是一种数据降维技术，用于数据预处理，它与线性代数紧密联系。在机器学习中，数据往往以张量的形式表示，算法的复杂度与数据的维数有着密切关系，甚至与维数呈指数级关联，在处理成千上万甚至几十万维的情况时，资源消耗将会是巨大的，因此必须对数据进行降维。PCA 通过正交变换将一组可能存在相关性的变量转换为一组线性不相关的变量，转换后的这组变量被称为主成分。

一般情况下，在数据挖掘和机器学习中，数据被表示为向量。例如，某个淘宝店 2012 年全年的流量及交易情况可被看成一组记录的集合，其中每一天的数据是一条记录，格式

如下：

PCA 的思想是通过在大量 n 维原始数据中搜索出 k 个最能代表原始数据的 n 维正交向量($k \leqslant n$)，将这 k 个正交向量投影到一个小得多空间上，从而对原始数据进行降维。通过 PCA 对原始数据进行压缩后，往往可以去除原始数据中的部分噪声，揭示一些难以觉察的特征，并且大幅降低计算量。

首先，对输入数据规范化，使得每个属性都假设数据样本进行了中心化，即将样本的均值变为 0。

设投影到 k 维空间后得到的新数据的坐标系为 $\boldsymbol{W}=\{\boldsymbol{w}^1,\boldsymbol{w}^2,\cdots,\boldsymbol{w}^k\}$，$\boldsymbol{w}^i$ 是标准正交基向量，满足 $\boldsymbol{w}^i \in \mathbf{R}^n$，$\boldsymbol{W}$ 是一个大小为 $n \times k$ 维的正交矩阵(此处如果 n 不等于 k，则该矩阵不是一个严格的正交矩阵)，满足下列条件：

$$\|\boldsymbol{w}^i\|_2 = 1, \quad \boldsymbol{w}^{i^{\mathrm{T}}}\boldsymbol{w}^j = 0(i \neq j)$$

PCA 由选择的解码函数决定，为了简化解码器，通过矩阵乘法将编码映射回 $\mathbf{R}^n$，即 $g(c)=\boldsymbol{W}$，其中 $\boldsymbol{W} \in \mathbf{R}^n$ 是解码矩阵。

先对原始数据零均值化，然后求协方差矩阵，接着对协方差矩阵求特征向量和特征值，这些特征向量组成了新的特征空间。设投影到 k 维空间后的新的坐标系为 $\{\boldsymbol{w}^1,\boldsymbol{w}^2,\cdots,\boldsymbol{w}^m\}$，是一个大小为 $n \times k$ 维的正交矩阵。由矩阵乘法的定义可以知道，投影到 k 维空间的点的坐标为 $\boldsymbol{Z}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$。通过该坐标系重构数据，将数据集 $\boldsymbol{Z}$ 从 k 维空间重新映射回 n 维空间，得到新的坐标点 $\boldsymbol{X}^*=\boldsymbol{W}\boldsymbol{Z}=\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$。

重构后的点 $\boldsymbol{X}^*$ 与原始数据点之间距离最小，即 PCA 可转化为求解最优问题：

$$\min_{\boldsymbol{W}} \|\boldsymbol{X}-\boldsymbol{X}^*\|_{\mathrm{F}}^2 = \min_{\boldsymbol{W}} \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\|_{\mathrm{F}}^2 (\text{约束条件为 } \boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}=\boldsymbol{I})$$

由于使用了相同的矩阵 $\boldsymbol{W}$ 对所有点进行解码，因此不能再孤立地看待每个点，必须采用最小化所有维数和所有点上的误差矩阵的 F 范数。根据 3.7 节中迹运算与 F 范数(对矩阵对应元素的平方和开方)的关系式，可得：

$$\begin{aligned}
\min_{\boldsymbol{W}} \|\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\|_{\mathrm{F}}^2 &= \min_{\boldsymbol{W}} \mathrm{Tr}((\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{X}-\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}})) \\
&= \min_{\boldsymbol{W}} \mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}) \\
&\quad (\text{循环改变迹运算中相乘矩阵的顺序不影响结果}) \\
&= \min_{\boldsymbol{W}} \mathrm{Tr}(-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}) \\
&\quad (\text{与 } \boldsymbol{W} \text{ 无关的项不影响} \min_{\boldsymbol{W}} \text{ 的值}) \\
&= \min_{\boldsymbol{W}} \mathrm{Tr}(-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}})
\end{aligned}$$

由约束条件 $\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}=\boldsymbol{I}$ 可得：

$$\begin{aligned}
&= \min_{\boldsymbol{W}} \mathrm{Tr}(-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}) \\
&= \min_{\boldsymbol{W}} \mathrm{Tr}(-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}) \\
&= \max_{\boldsymbol{W}} \mathrm{Tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}})
\end{aligned}$$

上述表达式的优化问题可以通过特征分解进行求解。具体地，最优解是 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 最大特征值对应的特征向量。一般情况下，在生成主成分的基时，矩阵 $\boldsymbol{W}$ 由几个最大的奇异值对应

的 I 个特征向量组成。该结论可以通过归纳法证明。

5.7　本章小结

通过本章的学习，需熟练掌握机器学习的主要基础知识，理解模型学习算法的过程和对模型的评估方法，了解偏差与方差在评估模型中的作用。

5.8　习　　题

1. 填空题

（1）在机器学习中，任务 T 是指机器学习系统处理_____的过程。

（2）通过为机器学习设定性能的_____来评估学习算法的效果。

（3）当模型过度地学习训练数据中的细节和噪声时，可能引起模型对训练数据的_____。

（4）BP 算法虽然称为反向传播，但事实上是一个典型的双向算法，包含了_____和_____两个流程。

（5）_____的作用是为神经网络中引入非线性因素。

2. 选择题

（1）在（　　）中取得良好效果的能力被称为泛化。

A. 已观测到的输入数据　　B. 已观测到的输出数据

C. 未观测到的输入数据　　D. 未观测到的输出数据

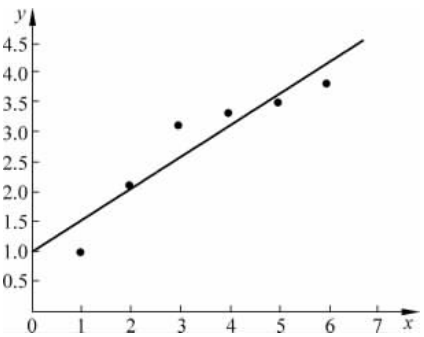
（2）在一个识别图片中的猫咪的模型训练中，模型习得（　　）的样本特性更可能属于过拟合情况。

A. 猫咪的毛　　B. 猫咪有两只眼睛

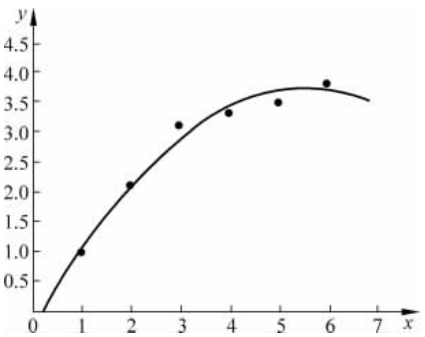
C. 猫咪有 4 条腿　　D. 猫咪有一条黑色的尾巴

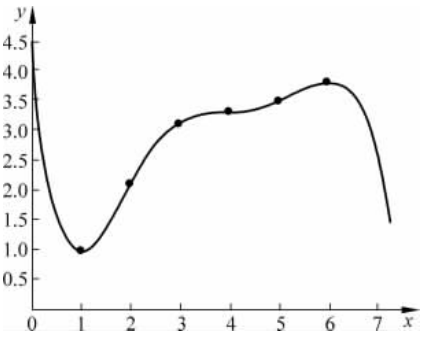
（3）下列图中，黑点表示训练数据，灰色直线表示学到的模型。下列 3 幅图中，反映模型欠拟合状态的是（　　）。

A.



B.



C. 

(4) 在模型训练中,若训练集 S 包含 200 个样本,测试集 T 包含 100 个样本。模型在被训练集 S 训练后,如果在测试集 T 上有 30 个样本错误,那么其错误率为(　　)。

A. 10%　　B. 15%

C. 30%　　D. 以上结果均不正确

(5) 在训练不足时,模型的拟合能力不足时,(　　)主导泛化错误率;随着训练程度的加深,模型的拟合能力逐渐增强,(　　)逐渐主导了泛化错误率。

A. 偏差　偏差　　B. 方差　偏差

C. 偏差　方差　　D. 方差　方差

3. 思考题

(1) 简述什么是过拟合、过拟合产生的原因及避免过拟合的方法。

(2) 简述偏差与方差在模型评估中的作用和区别。