**机器学习简介**

01

本章对机器学习做总体介绍。

1.1节和1.2节分别介绍机器学习的一些基本概念和发展历史。1.3节介绍机器学习任务的分类，1.4节叙述机器学习项目的一般步骤，1.5节对模型评估进一步展开讨论。

## **标题2**  1.1 什么是机器学习

机器学习是研究如何让机器模拟人类的学习行为，是实现人工智能的重要手段。人工智能的研究范围很宽，从表面上研究看，其可以被理解为机器的智能化，即让机器能像人一样思考问题、解决问题。人类所使用的学习方法有两种：演绎法和归纳法。这两种方法分别对应两种人工智能系统：专家系统和机器学习系统。演绎法从已知的规则和事实出发，推导新的规则和新的事实，对应专家系统。早期的人工智能系统大多是专家系统，亦被称为规则系统。归纳法则是通过对样本数据进行不断归纳，进而总结出规律和事实，对应机器学习系统。由于大数据时代的到来，数据的获取将会更容易，机器学习也会变得更重要。

1952年，IBM科学家塞缪尔利用最基本的人工智能形式——Alpha-Beta 剪枝算法，开发了一个跳棋程序。随着时间的增长，通过与塞缪尔下棋，程序下棋下得越来越好。因此塞缪尔创造了“机器学习”一词，并将其定义为“可以提供计算能力而无需显式编程的研究领域”。

著名的机器学习研究者米歇尔（Mitchell）在“*Machine Learning*”书中对机器学习给出定义：对于某类任务和性能度量，若一个计算机程序在任务中以度量的性能随着经验而自我改善，则我们称该程序在从经验学习[1]。

其中经验大多以数据的形式给出，（经验），需要做出什么样的决策（任务），以及如何评价结果（性能指标）。

**例**1-1**：玩跳棋**

经验：玩很多盘跳棋游戏的经验。

任务：下棋。

性能指标：程序赢得比赛的概率。

**例1-**2**：垃圾邮件分类**

经验：收集到的邮件，包含垃圾邮件和正常邮件。

任务：标记每一封邮件是否为垃圾邮件（分类）。

性能度量：准确率。

毕夏普（Bishop）则更多地从工程角度来研究这个领域，其在“*Pattern Recognition and Machine Learning*”一书的前言中写道：模式识别起源于工程学，而机器学习产生于计算机科学。然而这些领域可以看作是同一领域的两个方面[3]。

1952年，IBM科学家塞缪尔利用最基本的人工智能形式——Alpha-Beta 剪枝算法，开发了一个跳棋程序。随着时间的增长，通过与塞缪尔下棋，程序下棋下得越来越好。因此塞缪尔创造了“机器学习”一词，并将其定义为“可以提供计算能力而无需显式编程的研究领域”。

1957年，罗森布拉特（Rosenblatt）基于神经感知科学背景提出了感知机，非常类似于今天的机器学习模型。这在当时是一个非常令人兴奋的发现，比赫布的想法更实用。3年后，维德罗（Widrow）提出Delta学习规则用于线性神经网络的训练，后来被称为最小均方（Least Mean Square，LMS）算法。这些工作引发了联结主义的一次高潮。

1969年，明斯基（Minsky）和佩普特（Papert）证明单层神经网络不能解决异或问题。异或是一个基本逻辑问题，如果这个问题都解决不了，那神经网络的计算能力实在有限。这导致神经网络研究的式微，直到20世纪80年代。

值得一提的是，1951年菲克斯（Fix）和霍奇斯（Hodges）非公开提出了最近邻（The Nearest Neighbor，KNN）算法，后来康弗（Cover）和哈特（Hart）在1967年正式证明了KNN的一些性质（如KNN的分类错误率不大于最佳贝叶斯分类器的错误率的2倍）。KNN算法的核心思想是，如果一个样本在特征空间中的个最相邻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。KNN易于理解和实现，在有些情况下表现也非常好。

**第二阶段**是在1960年代中叶到1970年代中叶，被称为机器学习的冷静时期，研究目标是模拟人类的概念学习过程，并采用逻辑结构或图结构作为机器内部描述。机器采用符号来描述概念，并提出关于学习概念的各种假设，因此被称为符号主义。符号主义学派以认知主义为基础构成了学习的初步框架：认知就是通过对有意义的表示符号进行推导计算。本阶段的代表性工作有温斯顿（Winston）的结构学习系统和罗思（Roth）等的归纳学习系统。虽然这类学习系统取得较大成功，但只能学习单一概念，未能投入实际应用。

**第三阶段**是在1970年代中叶到1980年代中叶，称为复兴时期。在此期间，人们从学习单个概念扩展到学习多个概念，探索不同的学习策略和学习方法，且学习系统开始与各种应用结合起来，并取得很大成功。1980年，在美国的卡内基—梅隆大学（Carnegie Mellon University，CMU）召开了第一届机器学习国际研讨会，标志着机器学习作为一个独立的学科兴起。

神经网络在1980年代的复兴归功于物理学家霍普菲尔德（Hopfield）。1982年，霍普菲尔德提出了一种新的全连接反馈神经网络，既可以解决离散的模式识别问题，也可以给出一类组合优化问题的近似解（如旅行商优化问题）。1984年，霍普菲尔德用模拟集成电路实现了自己提出的模型。

霍普菲尔德模型的提出振奋了神经网络领域。联结主义运动再次兴起。1981年沃波斯（Werbos）基于反向传播（Back-Propagation，BP）算法，提出了多层感知器。1986年，鲁梅尔哈特、辛顿和威廉姆斯等成功将BP算法用于训练多层神经网络模型，BP算法迅速走红，掀起了神经网络的第二次高潮。

在另一个谱系中，昆兰（Quinlan）于1986年提出了一种著名的ID3决策树算法。这是符号主义流派的突破点。决策树能够解决很多实际生活应用问题，并且基于简洁的规则和清晰的推理，可解释性很强，这与黑盒下的神经网络模型恰恰相反。在ID3算法提出来以后，后续机器学习研究者探索了许多不同的选择或改进（如C4.5、CART算法），这些算法至今仍然活跃在机器学习领域。

**第四个阶段**是1980年代到2010年代，称为蓬勃发展时期。神经网络的复苏、符号学习方法、强化学习等并驾齐驱，百花齐放。学习方法学习系统开始走出实验室，进入实际应用领域。

1990年，勒坤（LeCun）采用BP神经网络实现对手写数字的识别，这是神经网络的第一个重大应用。直到20世纪90年代末，超过10%的美国支票都采用该技术进行自动识别。1998年，勒坤提出了LeNet-5的框架，即卷积神经网络（Convolutional Neural Network，CNN）的基本框架。CNN受视觉系统结构的启发，其基本思想是通过卷积层（局部连接、共享权重）和池化层（空间/时间下采样）减少参数数目以提高训练性能，在语音识别和图像处理方面显示出独特的优越性。

20世纪90年代，研究人员在使用神经网络进行序列建模的方面取得了重要进展。1990年埃尔曼（Elman）等提出的循环神经网络（Recurrent Neural Network，RNN）是一类具有短期记忆能力的神经网络。在循环神经网络中，神经元不但可以接受其它神经元的信息，也可以接受自身的信息，形成具有环路的网络结构。RNN这种环路结构能够利用过去时刻的信息，具有记忆功能，同时各个时间上参数共享，大幅减少了模型参数。但RNN存在梯度消失或者梯度爆炸问题，无法利用过去长时间的信息。霍克赖特（Hochreiter）和施密德胡伯（Schmidhuber）在1997年为RNN谱系引入了一个新的神经元，称为长短时记忆单元（Long Short-Term Memory，LSTM）来解决这些难题。LSTM建立门机制（输入门、输出门、遗忘门）来控制旧时刻输入与新时刻输入之间的折中，忘记不重要的，能缓解梯度消失、梯度爆炸问题，在较长的序列上比 RNN有更好的表现。

机器学习另一个非常重要的突破是科尔特斯（Cortes）和瓦普尼克（Vapnik）在1995年提出的支持向量机（Support Vector Machine，SVM）。SVM算法有坚实的理论基础和很好的实验结果，尤其在 2000 年左右提出了带核函数的支持向量机后，SVM在许多以前由神经网络占据的任务中获得了更好的效果，这使得人工神经网络再次进入寒冬。

19世纪90年代末神经网络精度不如传统机器学习算法的原因主要有两个：一是虽然训练神经网络的算法得到了改进，但在当时的计算资源下，要训练深层的神经网络仍然是比较困难的；二是当时的数据量较小，无法满足深层神经网络训练的需求。BP方法带来梯度消失问题和梯度爆炸问题，也限制了神经网络的深度和效果。

弗罗因德（Freund）和沙皮勒（Schapire）在1997年提出了著名的集成机器学习模型—提升（Boosting）算法，即利用多个弱分类器组合成强分类器的（Adaptive Boosting，Adaboost）算法。AdaBoost算法通过提高那些被前一轮弱分类器错误分类样本的权重，使得这些被错误分类样本受到下一轮弱分类器的更大关注。AdaBoost算法利用多个弱分类器解决复杂问题，在很多不同任务（如人脸检测等）上表现优秀。布莱曼（Breiman）在2001年探索了另一种集成模型—随机森林。随机森林集成了多棵决策树，其中每一棵决策树都根据样本的随机子集构建，每一个结点都是从特征的随机子集中选择。随机森林在抗过拟合方面有理论和实践证明，在多种任务上取得了成功。傅利曼（Friedman）1999年提出了基于决策树的梯度提升（Gradient Boosting Decision Tree，GBDT）算法，将模型学习视为一个数值优化问题，采用梯度下降的方式，每次加入一个弱学习器模拟负梯度，这是集成学习的又一大成。2014年陈天奇领衔开发的一套梯度提升算法的快速实现——XGBoost。由于其性能好、速度快，迅速成为各大数据科学竞赛的神器。2016年微软的分布式机器学习工具包（Distributed Machine Learning Toolkit，DMTK）开源了更快的轻量级提升算法工具——LightGBM。

**第五个阶段**是21世纪初年代到现在，是深度学习时代。深度学习指那些拥有多个处理层的计算模型，多个处理层可以学习具有多层次抽象的数据表示。近些年来，由于有更多数据且机器的计算能力大幅提高，人们可以训练更复杂的模型。这些复杂模型在很多复杂任务上取得了显著的改善，包括语音识别、计算机视觉、自然语言处理和许多其它领域，例如药物发现和基因组学等。

在深度学习的早期，自编码器（AutoEncoder，AE）和受限玻尔兹曼机（Restricted Boltzmann Machine，RBM）被广泛研究。2011年微软首次将深度学习应用在语音识别上，语音识别错误率降低约30%，是语音识别领域十多年来最大的突破性进展。2012年，深度学习技术在图像识别领域取得惊人的效果，克里泽夫斯基（Krizhevsky）等基于卷积神经网络（Convolutional Neural Network， CNN）在ImageNet图像识别比赛一举夺得冠军，将错误率从26%降低到15%，从此CNN吸引到了众多研究者的注意。2015年何恺明等提出了深度残差网络（Deep Residual Network，ResNet），将网络的层数推广到152层，在当时的ImageNet竞赛上取得了最好的成绩。从此，深度残差网络在各个计算机视觉任务上被广泛使用。

很多技术被用于提高神经网络的训练过程。线性整流函数（Rectified Linear Unit，ReLU）作为神经元的激活函数，极大加快收敛速度且从解决了梯度消失问题。丢弃法（Dropout）在每次训练中，每个隐藏单元以概率随机地被从网络中省掉，可以防止过拟合。批量归一化（Batch Normalization，BN）对每一网络层的输入进行批标准化，能保证输入分布的统一，并在一定程度上能替代丢弃法。注意力（Attention）机制使得深度学习模型能够只集中关注输入数据中最为重要的一部分，使模型做出更加准确的判断，同时不会对模型的计算和存储带来更大的开销。

另外，图形处理器（Graphics Processing Unit，GPU）被广泛用于加速模型计算。大数据、大模型、大计算是深度学习的3大支柱。由于标注数据难以获得，人们开始探索从无标注数据中学习。生成式对抗网络（Generative Adversarial Networks，GAN）和对偶学习（Dual Learning）提供了一种利用未标注数据进行端到端学习的有效方式。

深度强化学习将强化学习和深度学习结合在一起，用强化学习来定义问题和优化目标，用深度学习来解决策略和值函数的建模问题，然后使用误差反向传播算法来优化目标函数。深度强化学习在一定程度上具备解决复杂问题的通用智能，并在很多任务上都取得了很大的成功。例如，2013年慕尼黑（Mnih）等人提出了深度 Q 网络（DeepQ-Network，DQN）模型，采用深度神经网络拟合动作价值函数即Q函数，或者直接拟合策略函数。深度强化学习最有影响力的新闻莫过于2016年AlphaGo打败了世界冠军李世石，并在2017年又战胜了排名世界第一的世界围棋冠军柯洁。

尽管深度学习在欧氏空间中的数据方面取得了巨大的成功，但在许多实际的应用场景中的数据是从非欧式空间生成的，如在电子商务中，一个基于图的学习系统能够利用用户和产品之间的交互来做出非常准确的推荐。人们研究用于处理图数据的神经网络结构，一个新的研究热点—图神经网络（Graph Neural Networks，GNN）应运而生。

## **标题2**  1.3 机器学习任务的类型

机器学习研究如何构造能从经验学习，从而改善性能的计算机系统。在计算机系统中，经验通常以数据形式存在。根据数据中监督信息的给定方式，机器学习任务可分为3大类：监督学习、无监督学习、强化学习。

（1）**监督学习：**监督学习的训练数据中，每个样本既包含该样本的输入属性，也含对应的“正确答案”。例如，对一个猫狗图片分类任务，给定很多图片，有的标记为猫，有的标记为狗。 监督学习的目标是学习输入特征与标签之间的映射：。

（2）无**监督学习：**无监督学习旨在寻找训练数据中蕴含的结构，每个训练样本只包含描述该样本的特征，没有对应的标注信息。例如，要对图片进行分类，给了很多图片，却没有给图片的标签，机器学习算法需要自己对图片进行聚类，区分不同类型的图片。

（3）**强化学习：**强化学习中，智能体与环境交互，监督信息通过环境反馈获得。算法要根据当前的环境状态，确定一个动作来执行，然后进入下一个状态。如此反复，目标是让得到的收益最大化。例如要对图片分类，强化学习学习算法不断观察图片，尝试对图片分类，获得外界反馈（如分类正确给正的积分等，分类错误给负的积分），根据外界反馈不断改进分类。

除了这3大类，还存在一些介于3者之间的机器学习任务，如训练数据即包含部分标注数据，也包含一些非标注数据的半监督学习；标注信息不准或不全的弱监督学习；以及将其他领域的监督信息迁移到新领域的迁移学习等。本书主要关注监督学习和无监督学习。

### **1.3.1** 监督学习

监督学习从标注数据中学习模型，模型对输入和输出之间的关系进行建模，从而对新的输入进行预测。

#### 1．输入空间、特征空间和输出空间

称为一个样本（Sample）或者实例（Instance），其中为该样本的输入，为该样本的标签（Label）、输出（Output）或响应（Response）。输入与输出所有可能取值的集合分别称为输入空间X和输出空间Y。输入空间X和输出空间Y可以是有限元素的集合，也可以是整个欧氏空间。输入空间与输出空间可以是同一个空间，也可以是不同的空间，但通常输出空间远小于输入空间。

在监督学习中，输入和输出均可视为输入空间和输出空间上的随机变量/向量的取值。随机变量大些字母表示，习惯上用表示输入变量、用表示输出变量。随机变量的取值用小些字母表示，表示输入变量的取值、用表示输出变量的取值。变量可以是标量或向量，书中用粗斜体字母表示向量和矩阵，斜体字母表示标量。

原始输入可能与特定模型的假设不符，此时需要将原始输入做预处理或特征工程，将原始输入映射到特征空间。模型定义在特征空间，书中除非特别说明，我们假设输入向量已经是特征向量。

输入变量和输出变量可以是连续的，也可以是离散的。根据输出变量的不同类型，人们将监督学习问题进行细分。若输出变量为连续变量，输出空间Y，为回归问题；若输出变量为有限个离散值，Y，其中为可能取值的数目，为分类问题。

#### 2．假设空间

监督学习的目的是学习一个有输入到输出的映射，这个映射由模型来表示。所有模型的集合称为假设空间（Hypothesis Space）F。监督学习的模型可以是概率模型或非概率模型，由条件概率或决策函数表示。对具体的输入进行预测时，记为或。

学习的目的是在假设空间中找到一个最好的模型。这个优化问题通常通过定一个目标函数，求目标函数的极小值实现。监督学习的目标函数通常包含两部分：训练集上的损失之和和正则项：

|  |  |
| --- | --- |
|  | （1-1） |

其中为训练样本的数据，为损失函数，为模型的参数，为正则项，为正则参数。由于是待调节的超参数，我们将吸收进，即，目标函数变为。为书写简洁，记为，得到目标函数为：

|  |  |
| --- | --- |
|  | （1-2） |

损失函数表示模型预测值和真实标签之间的差异带来的损失。训练集上的平均损失被称为经验风险，表示模型与训练数据的拟合程度。损失函数的定义和具体的任务类型以及模型假设有关。本书中部分模型的损失函数如表1-1所示，各损失函数的含义请见后续章节。

表1-1 机器学习模型中常用的损失函数

|  |  |  |  |
| --- | --- | --- | --- |
| 问题 | 模型 | 损失函数名称 | 损失函数 |
| 回归 | 线性回归/决策树/  神经网络 | L2损失 |  |
| 线性回归/决策树/  神经网络 | Huber损失 |  |
| 支持向量回归 | 不敏感损失 |  |
| 分类 | Logistic回归/  神经网络 | 交叉熵损失 |  |
| 支持向量分类 | 合页损失 |  |
| 决策树 | 基尼指数 |  |

当训练样本数目足够大时，训练样本足以代表总体，模型与训练数据拟合得好意味着模型能学到总体数据蕴含的规律。但实际应用中，我们很难确定训练样本数目是否足够多。因为复杂模型需要更多训练数据来训练，我们总是能找到足够复杂的模型，使之完美拟合训练数据。例如，10000个训练样本对包含100个参数的线性模型已经足够多，但对上百兆参数的深度模型而言还是太少。下面我们通过一个多项式拟合得例子来直观地说明这一现象。

例1-3：多项式拟合

假设数据产生的过程为，输入在均匀采样 10个点，。我们用这10个样本训练阶多项式模型。阶多项式模型为：

其中为多项式模型的个参数。

假设多项式拟合的目标函数只包含训练集上的平方误差损失（L2损失），即

图1-1给出了曲线、10个训练样本点以及拟合的多项式曲线。当时，多项式曲线为一条曲线，与曲线比较接近，与训练数据点也拟合效果较好。当时，多项式曲线通过10个训练数据点，与训练数据完美拟合，但9阶多项式曲线与曲线相差甚远。此时最佳的模型应该是3阶多项式，因为模型既与训练数据拟合得较好，也比较平滑（模型简单）；而9阶多项式虽然与训练数据拟合得最好，但曲线波动厉害（模型复杂）。需要指出的是，当训练样本数目时，9阶多项式比3阶多项式更接近曲线。我们也另外采样了200个测试数据，多项式模型在测试数据上的平均损失函数在图中给出。从图中可以看出，当样本数时，最佳模型为3阶多项式；当时，最佳模型为5阶多项式。

|  |  |
| --- | --- |
|  |  |
| （a）训练样本数目为10，3阶多项式拟合 | （d）训练样本数目为100，3阶多项式拟合 |
|  |  |
| （b）训练样本数目为10，9阶多项式拟合 | （e）训练样本数目为100，9阶多项式拟合 |
|  |  |
| （c）训练样本数目为10，各阶多项式拟合的训练误差和测试误差 | （f）训练样本数目为100，各阶多项式拟合的训练误差和测试误差 |

图1-1 多项式拟合示例。其中为均匀分布，，服从正态分布

从上述多项式拟合的例子中可以看出，最佳模型不仅与训练数据有关，也与模型复杂度有关，模型复杂度与问题的复杂度相匹配时才是最好的。若样本数目较少，复杂模型会和训练数据拟合得太好，但此时复杂模型在测试数据上性能并不好，我们称之为过拟合（Over-Fitting）。当样本数较多时，训练集上的误差与训练误差基本相等，即训练样本足以代表总体样本，与训练数据拟合得的模型就是好的模型。若模型和训练数据拟合得不好，在训练集和测试集上训练都不好，我们称之为欠拟合（Under-Fitting）。事实上上述多项式拟合并不是个例。一般而言，机器学习模型复杂度与预测误差之间的关系如1-2所示。随着模型复杂度的增加，训练误差总是减小，但测试误差是先减少后增加，呈U形。U形底部对应的模型为最佳模型，左侧为欠拟合区域，右侧为过拟合区域。

因此最佳模型不仅要和训练数据拟合得好（训练集上的误差小），还要复杂度小，所以我们引入正则项对模型复杂度施加惩罚。模型越复杂，的值越大；反之，模型越简单/越平滑，的值越小。模型学习的目标函数中正则参数控制训练损失和正则项之间的折中。

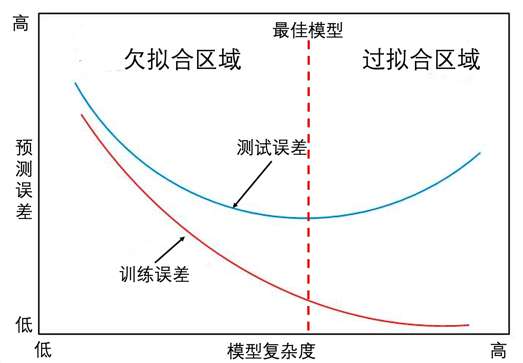


图1-2 模型复杂度与预测误差

常用的正则项由L1正则和L2正则。L1正则为参数的L1范数，即各元素绝对值之和：

|  |  |
| --- | --- |
|  | （1-2） |

L1正则亦为L0正则（非零元素的数目）的近似，因此我们对L1正则和L0正则不加区分。

L1正则为参数的L2范数的平方，即各元素平方之和：

|  |  |
| --- | --- |
|  | （1-3） |

目标函数最小既要求模型和训练数据拟合得好，又要求模型尽可能简单。这体现了机器学习的基本准则：奥卡姆剃刀（Occam's Razor）原理，即简单有效。

目标函数（损失函数、正则项和正则参数）确定后，通过优化技术可求得最佳的模型参数。常用的优化技术包括梯度下降法、牛顿法及其改进算法。正则参数的确定在1.5节模型评估中介绍。

### **1.3.2** 无监督学习

无监督学习从无标注的训练数据为中寻找数据的统计规律或隐含的结构。将个维的样本组成的数据矩阵，每一列为一个样本，每一行对应一个特征：

|  |  |
| --- | --- |
|  | （1-4） |

假设X为输入空间，Z为隐含结构空间，要学习的模型为函数，或条件概率分布。无监督学习在假设空间F中找出在给定评价指标下的最优模型。无监督学习是困难的，因为数据没有标注，要挖掘数据隐藏的规律通常需要大量的数据。

无监督学习的基本思想是对训练数据（数据矩阵）进行某种压缩，既要使得压缩得到的结果损失最小，也需考虑模型的复杂度。

无监督学习可分为：

• **降维** 降维挖掘数据矩阵的横向结构，将高维空间中的向量转换为低维空间中的向量。虽然原始数据的维度很高，其蕴含的本质维度可能很低。例如，一幅尺寸为人脸图像的维度为4096，但假设一组图像是同一个人脸在不同位置点光源下的图像，则其本质维度只有3维：点光源的位置。

• **聚类** 聚类挖掘数据矩阵的纵向结构，将相似样本聚成簇。例如，互联网公司有大量客户，聚类算法根据客户信息将客户进行整理归类。

• **概率密度估计** 概率密度估计同时挖掘数据的纵向和横向结构，假设数据由含有隐含结构的概率模型生成得到，从数据中学习该概率模型。但高维数据的概率密度估计是一个很难的问题。

## **标题2** 1.4 机器学习项目的一般步骤

一般而言，完成一个机器学习任务的一般步骤如下。

#### 1．明确任务，收集数据

我们首先要明确可以获得什么样的数据，目标是什么，是否可以归为标准的机器学习任务，如是否为分类、回归还是聚类。如果我们可以控制数据收集，应确保获取的数据有代表性，否则容易过拟合。对分类问题，数据偏斜不能过于严重。

#### 2．数据预处理和特征工程

拿到数据后，我们对数据进行探索式分析，以确定后续怎样进行特征变换和选用哪些机器学习模型。对数据进行探索式分析包括：

• 特征的数据类型（连续值、离散值、文本、时间、地理位置等）；

• 特征是否有缺失值；

• 特征的分布（高斯分布、均匀分布、指数分布等）；

• 特征与标签之间的关系；

• 特征与特征之间的相关性。

原始数据通常有噪声，需要进行数据清洗：

• 处理或删除异常值；

• 填充缺失值（零、均值、中位数等）或删除它们所在的行（或列）。

原始数据可能不符合机器学习算法的要求，这样我们需要将其转换为算法可接受的格式。在机器学习中，这个过程被称为特征工程。特征工程包括从原始数据中进行特征构建、特征提取和特征选择。特征工程直接关系到系统的性能，特征工程做得好，有时能甚至能使简单模型的效果比复杂模型效果好。但特征工程与业务场景高度相关，本书不做重点描述，读者可参考Scikit-Learn的相关模块：、和。不过在后续章节的案例分析中，我们会给出具体案例中特征工程的做法。

#### 3．训练模型

首先根据数据的特点和要解决的问题选择合适的模型，需要考虑的因素包括：要解决的问题是分类还是回归，样本数、特征维度、对内存的消耗程度、时间复杂度要求等。

模型类型确定好后，我们根据训练数据，采用优化算法得到最佳的模型参数。这里特指给定模型超参数的情况下，根据训练数据对模型参数的训练。

#### 4．模型评估与超参数调优

模型训练好后，我们对模型的性能进行评估。模型评估在验证集上进行，根据验证集上不同超参数对应模型的性能，对超参数进行调优。验证集可以是一个独立与训练集的数据集，也可以采用交差验证的方式，循环地从训练数据中分出一部分数据作为验证集。

模型评估判断模型是过拟合还是欠拟合，并且通过增加训练的数据量、降低模型复杂度来降低过拟合的风险，提高特征的数量和质量、增加模型复杂来防止欠拟合。通过分析误差产生的原因，还可提出针对性的模型迭代方案，进一步提升系统性能。

#### 5．模型融合（可选）

工程上，主要提升算法准确度的方法是分别在模型的前端（数据预处理和特征工程）与后端（模型融合）上下功夫。一般来说，模型融合后都能使得效果有一定提升。第7章介绍的集成学习可对模型进行融合。

#### 6．模型应用

最后是模型应用，包括系统启动、监控和维护。需要将准备好的生成环境数据载入机器学习模型，并定期检查系统的性能，定期根据新数据更新模型。

## **标题2** 1.5 模型评估

模型评估对模型的性能或优劣通过评价指标来衡量。不同的机器学习任务有不同的评价指标，每个指标的着重点不尽相同。2.4节和3.5节分别介绍了回归任务和分类任务中常用的评价指标。模型评估需要在验证集上完成，1.5.1节讨论验证集的划分。模型评估的目的是为了选择合适的模型，具体实现时，通常假设空间F为某个函数族，不同的模型体现为同一个函数族的超参数不同，因此模型评估的目的是为了超参数调优。1.5.2节介绍超参数的搜索方法。

### **1.5.1** 交叉验证

机器学习算法不仅关心模型与过去的训练数据拟合得有多好，更关心学习好的模型在未来的测试数据上的预测性能。模型在新的测试数据上的性能称为泛化（Generalization）性能。通常给定一个任务时只给了训练集，没有专门的验证集。因此我们需要从训练集中分离一部分样本作为验证集。

当训练样本较多时，可以直接从训练集中留出一部分样本作为验证集，称为留出法（Hold-Out），如图1-3（a）所示。Scikit-Learn中可以通过调用函数实现。

当训练样本没那么多时，可以采用交叉验证方式，将训练数据分成样本数目大致相等的份（折），每折样本轮流作为验证集，其余的折数据为训练集。图1-3（b）给出了5折交叉验证中验证集划分的示意图。Scikit-Learn的类实现了折交叉验证。对分类任务，我们还希望每折数据中各个类别的样本分布同总体训练数据的分布相同，因此采用分层（Stratified）交叉验证，即先将数据按类别分层，然后对每层的数据进行交叉验证。Scikit-Learn的类实现了分层的折交叉验证。

|  |  |
| --- | --- |
|  |  |
| （a）留出法 | （b）5折交叉验证 |

图1-3 验证集

令在留出第折的数据上训练的模型在第折的验证数据上的训练为，则折交叉验证得到的测试误差的估计为：。

另外，记得找到最佳超参数后，在给定超参数为最佳值的情况下，用全体训练数据再次训练模型。这个模型才是最终对测试样本做预测的模型。

在与时间有关的应用中，验证集的划分最好是按照时间拆分，比如评估的时候选取一个时间点，用在这个时间点之前的数据做训练，预测在这个时间点之后。这样更接近真实应用场景中我们根据历史数据预测未来的情形。

例1-4: 在鸢尾花分类数据集上寻找最佳的K近邻分类模型超参数

K近邻（K-Nearest Neighbors，KNN）模型是机器学习中最基本的分类算法之一。KNN模型计算测试样本和每个训练样本之间的距离，记录离其最近的K个邻居，并用这K个邻居的标签的投票结果作为该测试样本的标签。KNN算法也可以用于回归问题，唯一的区别是使用最近邻的平均值，而不是从最近邻投票。

在KNN模型中，参数K是一个很重要的参数。K越大，决策边界越平滑，但过于平滑的决策边界可能和问题的复杂度不匹配。K的选择通过模型在验证集的性能比较实现。

本例中我们以鸢尾花分类（Iris）数据集为例探讨参数K的取值对KNN的影响。Iris数据集是含150个样本，分为3类（Setosa，Versicolour，Virginica），每类50个数据。每个样本有4个属性：花萼长度、花萼宽度、花瓣长度、花瓣宽度。

图1-4给出当取2维特征（花萼长度、花萼宽度）、K分别取3和14的时候，KNN的决策边界。可以看出，当K等于3时，训练样本基本被分类正确，单决策边界很不光滑；当K等于14时，分类正确率和决策边界的光滑性取得了较好的折中。

|  |  |
| --- | --- |
|  |  |
| （a）K=3 | （b）K=14 |

图1-4 Iris数据集上，K取不同值时，采用KNN模型的决策边界

我们从所有150个样本中随机抽取其中20%的数据作为验证集，其余样本为训练集。图1-4（a）给出K从1～40对应的KNN分类器的正确率，当K取9时验证集上的性能最好（0.9）。由于本数据集样本数较少，为了使得泛化误差估计更准确，我们采用15折交叉验证寻找最佳的K。不同K对应的交叉验证的正确率如图1-4（b）所示，当K取14时验证集上的性能最好（0.96）。

|  |  |
| --- | --- |
|  |  |
| （a）20%数据作为独立验证集 | （b）15折交叉验证 |

图1-5 Iris数据集上采用KNN模型寻找最佳的超参数K

### **1.5.2** 超参数调优

模型评估的目的是为了超参数调优。当模型超参数较少时，可采用网络搜索；当超参数更多时，为减少搜索时间，可采用随机搜索方式。二者的差异如图1-6所示。

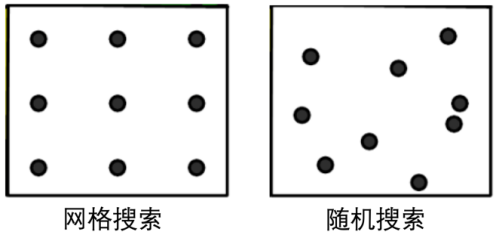


图1-6 超参数搜索方式

#### 1．网格搜索

如果有3个或更少的超参数时，常见的超参数搜索方法是网格搜索。对于每个超参数，使用者选择一个较小的有限值集去探索。这些超参数笛卡尔乘积得到一组组超参数。网格搜索使用每组超参数训练模型，挑选验证误差最小的一组超参数作为最佳模型。Scikit-Learn中函数了实现内嵌交叉验证的网格搜索，提供了在参数网格上穷举候选参数组合的方法。参数网格由参数指定，代码如下。

param\_grid = [

　　{'C': [1,10,100,1000], 'kernel': ['linear']},

　　{'C': [1,10,100,1000], 'gamma': [0.001,0.0001], 'kernel':['rbf']}

　　]

上面的参数指定了要搜索的两个网格（每个网格是一个字典）：第一个里面有4种参数组合，第二个里面有种参数组合。

#### 2．随机搜索

如果超参数较多，我们为每个超参数定义一个边缘分布，在这些边缘分布上进行搜索。Scikit-Learn中函数实现了内嵌交叉验证的随机搜索，依据某种分布对参数空间采样，随机得到一些候选参数组合方案。指定参数的采样范围和分布可以用一个字典完成。另外，计算预算（共要采样多少参数组合或者迭代做多少次）可以用参数来指定。针对每一个参数，既可以使用可能取值范围内的概率分布，也可以指定一个离散的取值列表（离散的列表将被均匀采样）。

例如：

{'C': scpiy.stats.expon(scale=100),

'gamma': scipy.stats.expon(scale=.1) }

在该例中，参数和均服从指数分布。

所以网格搜索是通过排列组合调整超参数，随机搜索是通过边缘分布调整超参数。通常随机搜索的运行时间却比网络搜索少得多，随机搜索得到的超参数组合的性能稍微差一点。另外超参数搜索可以并行进行，每组超参数的评估相互独立。

## **标题2** 1.6 小结

本章回顾了机器学习的发展历史，介绍了机器学习的定义和完成一个机器学习项目的基本步骤，介绍了监督学习和无监督学习中的一些基本概念，尤其是模型复杂度、过拟合、正则、模型评估、超参数调优等概念。这些概念可能略显抽象，读者可以在学习具体机器学习模型时再参照此处的综述加深理解。

## **标题2** 1.7 练习

1. 对例1-4中的Iris数据集，请用留一交叉验证（交叉验证的折数为训练样本数目）选择KNN模型中最佳的超参数K。
2. 对例2-1中的广告数据集，请用10折交叉验证选择KNN模型中最佳的超参数K。注意，这是一个回归问题，模型评价指标可用均方误差。