第3章

机 器 学 习

本章学习目标

- 掌握机器学习的概念，理解机器学习的三要素
- 掌握线性回归模型基本原理，了解基于最小二乘法的闭式解求解方法，掌握基于梯度下降法的优化求解方法，理解岭回归与 Lasso 回归等正则化方法，了解线性模型的可扩展性，包括广义线性模型和线性基函数回归
- 掌握基于逻辑回归的线性分类方法
- 掌握支持向量机基本原理，包括最大分类间隔和核函数法，理解求解不等式约束条件下二次函数极值问题的拉格朗日乘子法及 KKT 条件
- 了解集成学习方法，包括决策树、Bagging、Boosting 和梯度提升决策树
- 掌握模型拟合能力的分析方法、模型的评价指标与评估方法

本章首先引入机器学习基本概念，然后重点围绕机器学习的模型、策略、算法三要素展开相关方法原理的介绍，包括线性回归模型、线性分类模型、支持向量机以及集成学习方法等。

3.1 机器学习基本概念

3.1.1 基本定义

人类借助媒体，以视觉、听觉、触觉等多种方式来感知物理世界，并通过语言及行为与外界环境进行交互，不断提高认知水平。随着人类社会进入信息时代，人工智能技术逐步兴起，具有广泛的应用前景，可为教育、医疗、制造、零售、金融、自动驾驶等领域提供智能化解决方案，如智慧校园、疾病诊断、生产监控、商品推荐、风险管理、自动避障等。这些应用都离不开机器学习技术。

机器学习技术采用计算机算法对输入数据与预期输出之间的映射规律进行建模，并利用得到的模型对新的输入数据进行预测，例如，根据输入的水果图像，输出水果的类别，或者，根据某地区的降雨量等数据，预测水果产量。输入与输出的数据可以为标量或向量，类别标签等符号类数据可以转换为标量或向量。输入的数据往往是对原始数据进行特征提取后的特征向量。不失一般性，假设输入数据为向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$，预期的输出为 y，

机器学习的建模过程是学习映射函数 $y=f(\boldsymbol{x})$，如图 3-1 所示。

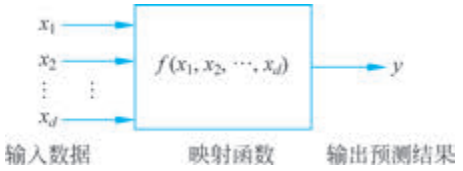


图 3-1　机器学习的建模过程

机器学习的三要素包括模型、策略和算法。

模型对应于上述函数 $f(\boldsymbol{x})$。函数中的计算过程通常可以用矩阵及向量形式表示，例如，$y=f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$。函数 $f(\boldsymbol{x})$ 的自变量为向量 $\boldsymbol{x}$，y 为输出值，$\boldsymbol{w}$ 为 $f(\boldsymbol{x})$ 中可学习的参数。

策略指的是在训练过程中用于指导模型参数学习的损失函数，如回归问题中常用的均方误差函数 $\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$，其中，$y_i$ 为第 i 个样本中因变量的真值，$\hat{y}_i$ 为预测值。

算法是指为达到模型参数学习最优的目标所采用的优化方法，如优化问题求解中常用的梯度下降法。

机器学习任务根据训练集中样本是否具有真值(ground-truth)标签以及训练策略分为三大类型：

(1) 监督学习：训练集样本标签已知，训练集通常可以表示为成对的特征向量—标签数据的集合，记为 $\{(\boldsymbol{x}_i,y_i)\}$，$i=1,\cdots,N$，其中 N 为样本总数。监督学习的任务是利用训练集数据拟合函数 $y=f(\boldsymbol{x})$ 得到对应的模型参数。常见的监督学习方法包括神经网络、支持向量机等。

(2) 非监督学习：训练集样本标签未知，即仅有数据 $\{\boldsymbol{x}_i\}$，$i=1,\cdots,N$，其中 N 为样本总数。非监督学习的任务是对无标签的训练集数据进行分析与转换，挖掘数据蕴含的规律。常见的非监督学习方法包括主成分分析、聚类等。

(3) 强化学习：智能体在与环境的交互中，根据行动的回报，做出下一步行动的决策。

大部分机器学习方法属于监督学习，需要利用具有标签真值的训练集数据对模型进行训练。非监督学习方法也有较大的应用潜力，例如，从互联网获取的文本、图像、视频、语音等原始数据通常缺乏标签真值，人工标注费时费力，而采用非监督学习方法可以有效利用这些原始数据。近年来兴起的自监督学习就是一种非监督学习方法。早期的自监督学习一般是利用已有的模型对原始数据样本进行类别标签预测，得到伪标签用于监督学习。目前应用较为广泛的自监督学习技术，采用不需要类别标签的机器学习代理任务，如深度学习技术中采用图像重构任务来进行预训练，可在图像上施加掩码(即遮蔽图像部分区域)再进行图像重构，由此通过掩码图像预测等代理任务对模型进行预训练，从而有效缓解因模型参数量过大而导致标记样本不足的问题。

大模型技术是人工智能与机器学习领域的前沿技术，其训练方法综合了非监督学习、监督学习和强化学习。OpenAI 公司于 2022 年底推出了模型参数量高达 4 亿级别的 ChatGPT。ChatGPT 的训练方法结合了预训练(pretraining)、微调(fine-tuning)和强化学

习：首先通过海量文本进行预测下一个单词标记的非监督预训练，学习语言的基本规律；然后使用标注数据进行监督微调，使其适应特定任务；最后通过强化学习从人类反馈中优化模型，使其生成更符合人类偏好和价值观的高质量回复。这种综合训练的方法不仅使ChatGPT具备了强大的语言理解和生成能力，还推动了其在多任务、多领域的应用，成为迈向通用人工智能的重要一步。

监督学习的任务进一步根据 y 的取值不同大致分为回归和分类两种类型：

(1) **回归**：y 的取值一般为连续变化的实数；

(2) **分类**：y 的取值对应于类别集合$\{\omega_i\}$，$i=1,\cdots,C$，其中 C 为类别总数。y 的取值可以采用类别标号，通常为离散值，如 $y\in\{1,2,\cdots,C\}$。

对于二分类任务，即类别集合包含正类和负类$\{\omega_1,\omega_2\}$，y 一般取 0 或 1，即 $y\in\{0,1\}$，其中类别标号 1 代表正类 ω_1，类别标号 0 代表负类 ω_2；有时也取-1 和 1，即 $y\in\{-1,1\}$，其中类别标号 1 代表正类 ω_1，类别标号-1 代表负类 ω_2。

二分类决策的判决函数可以表示为

$$f(x)\begin{cases}\geqslant 0, & x\in\omega_1\\ <0, & x\in\omega_2\end{cases} \tag{3-1}$$

方程 $f(x)=0$ 对应于数据空间中的决策面，将属于 ω_1 类别的数据和 ω_2 类别的数据区分开来。

机器学习中的分类任务即模式识别技术。自 20 世纪 70 年代发展起来的传统模式识别技术主要采用人工设计特征提取方法的特征工程，包括贝叶斯分类器和隐含马尔可夫模型等概率模型，较为成功地应用于文字识别、人脸识别和语音识别。以 2012 年卷积神经网络AlexNet 在 ImageNet 图像识别大赛中取得成功为标志，深度学习技术迅速兴起，通过引入模拟人类认知的感受野、记忆、注意力等机制，在图像识别、语音识别、机器翻译等任务上显著超过传统方法，推动了人工智能领域的发展。

近年来，基于注意力机制和预训练技术的大语言模型发展日新月异。这些大语言模型不仅在自然语言处理任务中表现出色，还展现出诸如上下文学习、复杂推理等涌现能力，推动了通用人工智能的研究进程。随着数据、模型和算力的不断提升，大语言模型的应用范围逐渐拓展到支持视觉、语音等多模态场景。大模型训练的前沿技术不断迭代，例如，DeepSeek-V3 采用了基于矩阵低秩分解的多头潜在注意力机制(multi-head latent attention，MLA)和基于混合专家模型(mixture of experts，MoE)的模型稀疏化技术，显著降低了计算成本。这些新技术与机器学习领域的传统方法有着密切的联系。

对于监督学习中的分类任务来讲，机器学习的建模方法可以分为生成式模型(generative model)和鉴别式模型(discriminative model)。生成式模型通过建模学习数据的联合概率分布 $P(X,Y)$，能够生成新的数据样本并推断类别，如朴素贝叶斯分类器(naive Bayes classifier)等；而鉴别式模型则直接学习条件分布 $P(Y|X)$，直接利用输入数据预测输出标签，如逻辑回归(logistic regression)和支持向量机(support vector machine，SVM)。生成式模型通过对数据概率分布的生成过程进行建模实现分类，鉴别式模型则直接优化求解分类决策面。

需要注意的是，这里所说的生成式模型是指一种基本的机器学习建模方法，与当前迅速发展的人工智能内容生成应用技术不同。人工智能内容生成是深度学习技术在媒体创作领域的应用，可根据用户需要生成文本、图像、音频等高质量内容，这些合成数据也可以作为机器学习中的训练样本。

机器学习流程包括训练阶段和测试阶段。在训练阶段，利用训练集样本学习映射函数 $y=f(\boldsymbol{x})$。在测试阶段，对于输入样本数据 $\boldsymbol{x}$，根据训练得到的函数模型输出预测值 $\hat{y}$。

机器学习所使用的数据集可分为训练集、验证集和测试集。训练集指的是在训练阶段用于训练模型的数据集；验证集指的是在训练阶段用于检验模型训练状态、收敛情况的数据集，验证集通常可以用于模型超参数的选取；测试集指的是在训练完成后，在测试阶段用于评估模型的实际能力的数据集。

3.1.2　线性回归模型

1. 误差平方和最小原则

假设用某种测量仪器尽可能准确地测量一个未知量 u，进行 n 次读数，由于存在各种实验误差，可能得到略有不同的结果 $z_1,z_2,\cdots,z_n$。

考虑一个问题：哪一个数值最能代表“真实”值或“最优”值呢？通常选择一个使误差平方和最小的值作为代表。也就是说，选取合适的 u，使 $(u-z_1)^2+(u-z_2)^2+\cdots+(u-z_n)^2$ 最小。可以证明，这个“最优值”正是这些数的算术平均值，当 $u=\bar{z}$（即 u 为 z_i，$i=1,2,\cdots,n$ 的算术平均值），误差平方和最小。

误差平方和最小原则在很多实际测量中都能提供最优估计，而由此诞生的最小二乘法不仅用于简单的均值估计，在更复杂的情形中也有广泛应用，成为参数估计与数据回归分析的基础。

2. 线性回归基本模型

线性模型是机器学习中最为基础的建模方法。对于输入数据 $\{x_i\}$，$i=1,\cdots,d$，输出变量 y 与输入数据的线性映射函数为

$$y=f(\boldsymbol{x})=\sum_{i=1}^{d}w_i x_i+b \tag{3-2}$$

式中，$\{w_i\}$，$i=1,\cdots,d$ 为线性加权系数；b 为偏置量。若 $d=1$，则为一元线性回归；若 $d>1$，则为多元线性回归。

采用向量形式表示该线性映射，输入向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$，权重向量 $\boldsymbol{w}=(w_1,w_2,\cdots,w_d)^{\mathrm{T}}$，则有

$$y=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{3-3}$$

还可以采用增广向量形式表示该线性映射，输入向量 $\boldsymbol{x}=(x_0,x_1,x_2,\cdots,x_d)^{\mathrm{T}}$，其中，常量 $x_0=1$，权重向量 $\boldsymbol{w}=(w_0,w_1,w_2,\cdots,w_d)^{\mathrm{T}}$，其中，$w_0=b$，为对应偏置量的参数。线性映射关系可用增广向量形式表示为

$$y=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \tag{3-4}$$

在线性模型的训练阶段，利用训练集数据 $\{(\boldsymbol{x}_j,y_j)\}$，$j=1,2,\cdots,N$ 对模型参数进行估

计。训练集的数据通常需要进行预处理以及特征提取，得到特征向量。$\boldsymbol{x}_i=(x_{i0},x_{i1},x_{i2},\cdots,x_{id})^{\mathrm{T}}$ 是第 i 个样本的特征向量。y_i 是第 i 个样本的目标值，通常包含噪声 ε，满足

$$y_i=f(\boldsymbol{x}_i)+\varepsilon=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+\varepsilon \tag{3-5}$$

如图 3-2 所示线性回归的几何解释，y 是观测数据，$\hat{y}$ 是估计数据，t 是最优估计。线性回归的目标是让线性模型的输出 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ 尽可能接近最优估计 t，从而使得噪声 ε 尽可能小。

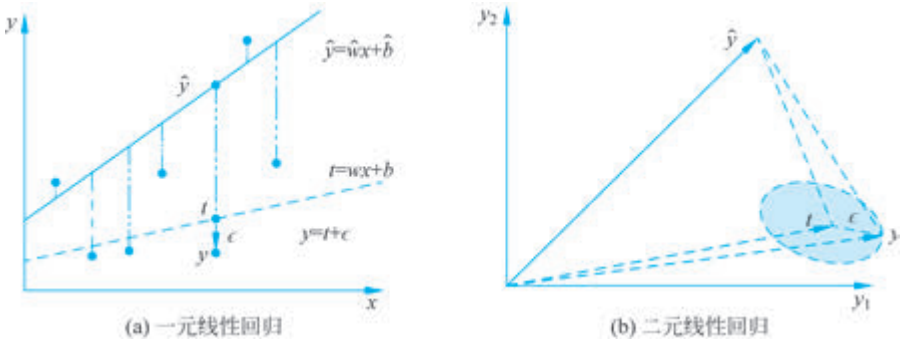


图 3-2　线性回归的几何解释

为了实现优化目标，考虑最大似然估计，使得似然函数 $p(y\mid\boldsymbol{x})$ 最大，也即最小化负对数似然 $-\log p(y\mid\boldsymbol{x})$。假设噪声 ε 服从正态分布 $\varepsilon\sim N(0,\sigma^2)$，因此

$$p(\varepsilon)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{\varepsilon^2}{2\sigma^2}} \tag{3-6}$$

似然函数满足

$$p(y\mid\boldsymbol{x})=\prod_{i=1}^{N}p(y_i\mid\boldsymbol{x}_i)=\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{|y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i|^2}{2\sigma^2}} \tag{3-7}$$

相应地，负对数似然为

$$-\log p(y\mid\boldsymbol{x})=\sum_{i=1}^{N}\frac{|y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i|^2}{2\sigma^2}+\frac{N}{2}\log(2\pi\sigma^2) \tag{3-8}$$

忽略常数项，为使得负对数似然最小化，需要最小化求式(3-8)中的第一项，即

$$\sum_{i=1}^{N}|y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i|^2=\sum_{i=1}^{N}[y_i-f(\boldsymbol{x}_i)]^2 \tag{3-9}$$

记

$$L(\boldsymbol{w})=\sum_{i=1}^{N}[f(\boldsymbol{x}_i)-y_i]^2=\sum_{i=1}^{N}\left(w_0+\sum_{j=1}^{d}w_jx_{ij}-y_i\right)^2 \tag{3-10}$$

通过选择系数 $\boldsymbol{w}=(w_0,w_1,\cdots,w_d)^{\mathrm{T}}$，以最小化误差平方和。最小化上述误差平方和(即损失函数)的估计方法通常称为最小二乘法，在噪声为正态分布时，该估计方法与最大似然估计等价。

为便于推导，采用矩阵形式的表示对 $L(\boldsymbol{w})$ 进行优化求解。优化求解可采用闭式解或数值求解方法。

设 $\boldsymbol{X}$ 为 $N\times(d+1)$维矩阵，其中每一行是一个输入向量(第一位是 1)，并设 $\boldsymbol{y}$ 为训练集中的 N 维输出向量。则可以写出误差平方和为

$$L(\boldsymbol{w})=(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}) \tag{3-11}$$

这是一个关于 $d+1$ 个参数的二次函数。误差平方和 $L(\boldsymbol{w})$对 $\boldsymbol{w}$ 求导可得

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}}=2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}) \tag{3-12}$$

假设 $\boldsymbol{X}$ 具有满列秩，因此 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 是正定矩阵，根据矩阵求导规则有

$$\frac{\partial \|\boldsymbol{A}\|^{2}}{\partial \boldsymbol{A}}=2\boldsymbol{A} \tag{3-13}$$

$$\frac{\partial \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}}=\boldsymbol{A}^{\mathrm{T}} \tag{3-14}$$

计算损失函数对参数的导数，并令导数为零得

$$\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})=0 \tag{3-15}$$

从而得到唯一解为

$$\hat{\boldsymbol{w}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3-16}$$

在测试阶段，输入一个样本的增广向量 $\boldsymbol{x}$，预测值为 $\hat{y}=\hat{f}(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\hat{\boldsymbol{w}}$

对于 N 个输入样本对应的矩阵 $\boldsymbol{X}$，预测值为

$$\hat{\boldsymbol{y}}=\boldsymbol{X}\hat{\boldsymbol{w}}=\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3-17}$$

其中 $\hat{y}_i=\hat{f}(\boldsymbol{x}_i)$。

3. 多输入多输出的线性回归矩阵形式表示

设有多个输出变量 $y_1,y_2,\cdots,y_K$，根据多个输入变量 $x_1,x_2,\cdots,x_d$ 来预测这些输出。每个输出对应一个线性回归模型。对于 N 个训练样本，可以将模型写成矩阵形式：

$$\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{W} \tag{3-18}$$

式中，$\boldsymbol{Y}$ 是维度为 $N\times K$ 的响应矩阵，其中第 i 行、第 k 列的元素为 y_{ik}；$\boldsymbol{X}$ 是维度为 $N\times(d+1)$的输入矩阵增广形式，第 i 行元素为第 i 个样本的数据增广向量$(x_{i0},x_{i1},x_{i2},\cdots,x_{id})$，其中 $x_{i0}=1$；$\boldsymbol{W}$ 是维度为$(d+1)\times K$ 的参数矩阵。损失函数为

$$L(\boldsymbol{W})=\sum_{k=1}^{K}\sum_{i=1}^{N}(f_k(x_i)-y_{ik})^{2}=\mathrm{tr}\left[(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{W}-\boldsymbol{Y})\right] \tag{3-19}$$

最小二乘法求得的闭式解为

$$\hat{\boldsymbol{W}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} \tag{3-20}$$

二维输入变量的线性回归模型如图 3-3 所示。

4. 数值求解方法

训练阶段需要求解使损失函数 L 极小化的权重向量 $\boldsymbol{w}$ 和偏置量 b。模型参数迭代求解过程为：在初始化模型参数后，对于迭代过程输入的样本计算损失函数，再求得损失函数 L 对于模型参数的偏导数，然后利用梯度下降法调整模型参数。基于梯度下降法的优化求解算法具体计算步骤如算法 3-1 所示。

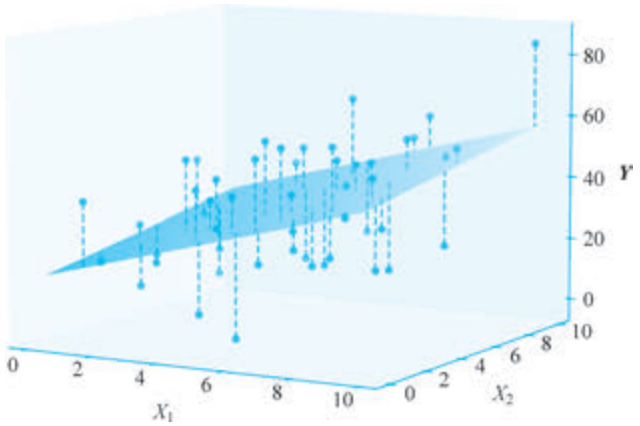


图 3-3　二维输入变量的线性回归模型

算法 3-1　基于梯度下降法的优化求解算法

输入：样本数据 $D=\{(\boldsymbol{x}_i,y_i)\}$，$i=1,2,\cdots,N$；学习率 η

输出：权重向量 $\boldsymbol{w}$，偏置量 b

算法步骤：

(1) 初始化：初始化权重向量 $\boldsymbol{w}_0$，偏置量 b_0

(2) 迭代 $t=1,2,\cdots,T$：

对于每次迭代输入的样本，计算损失函数 L

计算损失函数对权重向量和偏置量的导数 $\frac{\partial L}{\partial \boldsymbol{w}}$、$\frac{\partial L}{\partial b}$

更新权重向量和偏置量

$$\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1}-\eta\frac{\partial L}{\partial \boldsymbol{w}}$$

$$b_t \leftarrow b_{t-1}-\eta\frac{\partial L}{\partial b}$$

所求解的参数也可记为 $\theta=\{\boldsymbol{w},b\}$，基于梯度下降法的优化求解方法如图 3-4 所示。

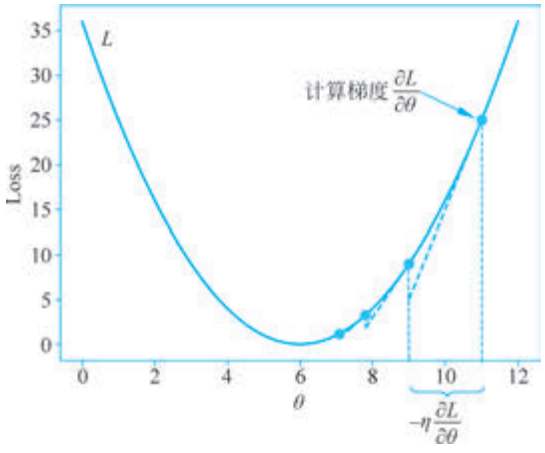


图 3-4　基于梯度下降法的优化求解方法

【例 3-1】　某省多个地区的夏季平均降雨量、苹果产量数据如表 3-1 所示。建立苹果产量与夏季平均降雨量的一元线性回归关系，并采用数值方法求解。

表 3-1　某省多个地区的夏季平均降雨量与苹果产量数据

地　区	降雨量/mm	苹果产量/(吨/公顷)
1	268	16
2	352	19
3	536	32
4	172	10
5	384	27

解： 记降雨量为 x，苹果产量为 y，一元线性回归关系为 $\hat{y}=wx+b$，需要求解的模型参数为 w 和 b。

步骤 1：初始化模型参数 $w_1=0.1, b_1=0$

步骤 2：对于第 $t=1$ 次迭代，用当前模型参数进行预测

$$\hat{y}_1=w_t x_1+b_t=0.1\times 268+0=26.8$$

$$\hat{y}_2=w_t x_2+b_t=0.1\times 352+0=35.2$$

$$\hat{y}_3=w_t x_3+b_t=0.1\times 536+0=53.6$$

$$\hat{y}_4=w_t x_4+b_t=0.1\times 172+0=17.2$$

$$\hat{y}_5=w_t x_5+b_t=0.1\times 384+0=38.4$$

步骤 3：计算预测值与真值之间的 MSE(均方误差)损失函数为

$$L=\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)^2$$

$$=\frac{1}{5}[(26.8-16)^2+(35.2-19)^2+(53.6-32)^2+(17.2-10)^2+(38.4-27)^2]$$

$$=205.488$$

步骤 4：误差反向传播

由 $L=\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)^2$ 和 $\hat{y}_i=w_t x_i+b_t$ 有

$$\frac{\partial L}{\partial \hat{y}_i}=\frac{2}{n}(\hat{y}_i-y_i)$$

$$\frac{\partial \hat{y}_i}{\partial w_t}=x_i,\quad \frac{\partial \hat{y}_i}{\partial b_t}=1$$

按照复合函数求导的链式法则，计算误差对模型参数的梯度

$$\frac{\partial L}{\partial w_t}=\sum_{i=1}^{n}\frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial w_t}=\sum_{i=1}^{n}\frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial w_t}=\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)x_i$$

$$=\frac{2}{5}[(26.8-16)\times 268+(35.2-19)\times 352+(53.6-32)\times 536+$$

$$(17.2-10)\times 172+(38.4-27)\times 384]$$

$$=10316.16$$

$$\frac{\partial L}{\partial b_t}=\sum_{i=1}^{n}\frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial b_t}=\sum_{i=1}^{n}\frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial b_t}=\frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i-y_i)$$

$$=\frac{2}{5}[(26.8-16)+(35.2-19)+(53.6-32)+(17.2-10)+(38.4-27)]$$

$$=26.88$$

步骤 5：利用梯度下降法更新模型参数

假设学习率 $\eta=5\times10^{-6}$，按梯度下降法更新后的模型参数为

$$w_{t+1}=w_t-\eta\frac{\partial L}{\partial w_t}=0.1-5\times10^{-6}\times10,316.16=0.0484192$$

$$b_{t+1}=b_t-\eta\frac{\partial L}{\partial b_t}=0-5\times10^{-6}\times26.88=-0.0001344$$

步骤 6：令 $t=2,3,\cdots,N_t$，重复步骤 2～5，多次迭代更新。将最后一次迭代的模型参数作为模型训练的结果。

为了求解最终的模型参数 w 和 b，需要进行多次迭代，直至达到预设的迭代次数或收敛条件。经过 10 次迭代后，模型参数已经基本稳定，得到

$$w_{10}=0.0609281,\quad b_{10}=-0.0001075$$

而根据闭式解计算公式，得到

$$w=\frac{\sum_{i=1}^{n}(y_i-\bar{y})(x_i-\bar{x})}{\sum_{i=1}^{n}x_i^2-\frac{1}{n}\left(\sum_{i=1}^{n}x_i\right)^2}=0.0623538$$

$$b=\bar{y}-w\bar{x}=-0.5499477$$

图 3-5 分别展示了【例 3-1】中训练前、1 次迭代更新后、10 次迭代更新后以及闭式解的一元线性模型的可视化结果。可以看到预测得到的曲线和闭式解曲线非常接近。

5. 正则化方法

线性回归模型可以从机器学习三要素角度进行多种扩展，从目标函数来看，可以采用正则化技术防止模型过拟合。常见的线性回归正则化方法包括岭回归（ridge regression）和 Lasso（least absolute shrinkage and selection operator）回归，分别通过引入 L2 正则化和 L1 正则化来防止模型过拟合。岭回归通过惩罚模型权重参数的平方和（L2 范数）来缩小系数值，但不会将系数完全压缩为零。

在岭回归目标函数中增加模型参数的 L2 范数（记为 $\|\cdot\|_2$ 或 $\|\cdot\|$）的平方，即

$$L(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w})=(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})+\lambda\|\boldsymbol{w}\|_2^2 \tag{3-21}$$

式中，$\|\boldsymbol{w}\|_2=\sqrt{\sum_{i=0}^{d}w_i^2}$。

Lasso 回归通过惩罚系数的绝对值之和（L1 范数）可以将某些系数压缩为零，从而实现特征选择，适用于高维数据中的稀疏建模。岭回归和 Lasso 回归都能提升模型的泛化能力，但 Lasso 回归更适合于特征选择任务。

Lasso 回归目标函数中增加模型参数的 L1 范数（记为 $\|\cdot\|_1$），即

$$L(\boldsymbol{X},\boldsymbol{y},\boldsymbol{w})=(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})+\lambda\|\boldsymbol{w}\|_1 \tag{3-22}$$

式中，$\|\boldsymbol{w}\|_1=\sum_{i=0}^{d}|w_i|$；$\lambda$ 为控制正则化项作用大小的加权系数。

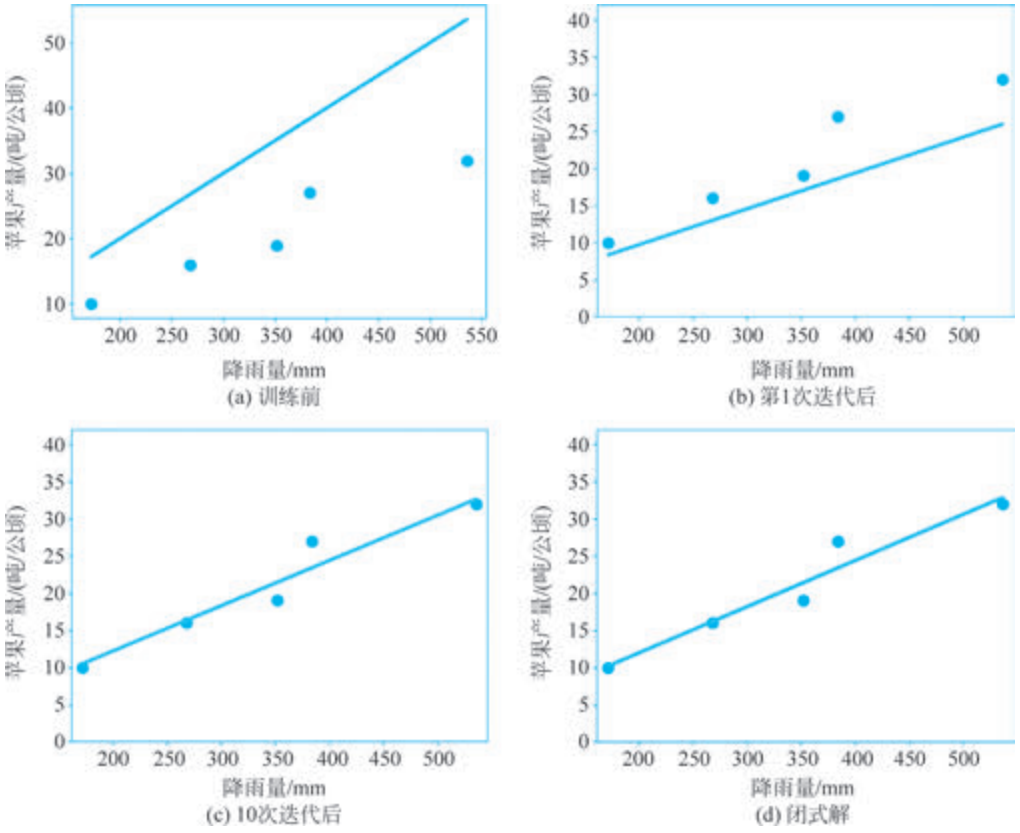


图 3-5　一元线性模型训练过程的可视化结果

【例 3-2】　尝试推导如何在岭回归中引入核函数。假设输入的特征空间为 $\mathcal{X}=\mathbb{R}^{(d+1)\times 1}$，输出为 $\mathcal{Y}=\mathbb{R}$，采样得到的数据集为 $\mathcal{D}=\{(\boldsymbol{x}_1,y_1),\cdots,(\boldsymbol{x}_n,y_n)\}$。记 $\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n]^{\mathrm{T}}\in\mathbb{R}^{n\times(d+1)}$，$\boldsymbol{y}=[y_1,\cdots,y_n]^{\mathrm{T}}\in\mathbb{R}^{n\times 1}$，则岭回归目标函数可以表示为

$$L=\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|^2+\lambda\|\boldsymbol{w}\|_2^2$$

式中，$\boldsymbol{w}\in\mathbb{R}^{(d+1)\times 1}$，$\lambda>0$。

（1）证明：对于岭回归，使得 L 最小时的 $\boldsymbol{w}^*$ 满足 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}^*+\lambda\boldsymbol{w}^*=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$，且 $\boldsymbol{w}^*=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$。

（2）证明：$\boldsymbol{w}^*$ 可以由采样点的特征向量线性组合表示，即 $\boldsymbol{w}^*=\boldsymbol{X}^{\mathrm{T}}\alpha$，其中，$\alpha=(\lambda\boldsymbol{I}+\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{y}$。

（3）设核函数为 $k(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$，$\phi(\boldsymbol{x})$表示 $\boldsymbol{x}$ 映射到某个便于求解问题的空间中，$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$表示把 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 映射到该空间中后计算的内积。例如，多项式核函数为 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(1+\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^p$，高斯核函数为 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$，其中 $\gamma=\dfrac{1}{2\sigma^2}$，$p$ 为阶数。将输入数据之间核函数矩阵记为

$$K(\boldsymbol{x})=\begin{bmatrix}k(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1,\boldsymbol{x}_n)\\ \vdots & & \vdots\\ k(\boldsymbol{x}_n,\boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n,\boldsymbol{x}_n)\end{bmatrix}$$

对于一个新输入的数据 $\boldsymbol{x}$，$\boldsymbol{x}$ 与原有数据 $[\boldsymbol{x}_1 \quad \cdots \quad \boldsymbol{x}_n]$ 的核函数可以表示为向量

$$k_x = [k(\boldsymbol{x},\boldsymbol{x}_1) \quad \cdots \quad k(\boldsymbol{x},\boldsymbol{x}_n)]$$

在采用岭回归的情形下，引入核函数，推导具有核函数的模型表示形式 $f(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}^*$。

解：

(1)

$$\frac{\partial L}{\partial \boldsymbol{w}} = 2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y})+2\lambda\boldsymbol{w}=0 \Rightarrow \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}^* + \lambda\boldsymbol{w}^* = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

$$\Rightarrow (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})\boldsymbol{w} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \Rightarrow \boldsymbol{w}^* = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

上面矩阵可取逆，是因为 $\lambda>0$，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I}$ 可逆。

(2)

由(1)可知

$$\boldsymbol{w}^* = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

由于

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}+\lambda\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}+\lambda\boldsymbol{I}) = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})\boldsymbol{X}^{\mathrm{T}}$$

因此

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}+\lambda\boldsymbol{I})^{-1}$$

代入 $\boldsymbol{w}^*$ 的表达式中得到

$$\boldsymbol{w}^* = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}+\lambda\boldsymbol{I})^{-1}\boldsymbol{y}$$

所以满足 $\boldsymbol{w}^* = \boldsymbol{X}^{\mathrm{T}}\alpha$，其中

$$\alpha = (\lambda\boldsymbol{I}+\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{y}$$

(3)

$$\boldsymbol{w}^* = \boldsymbol{X}^{\mathrm{T}}\alpha = \boldsymbol{X}^{\mathrm{T}}(\lambda\boldsymbol{I}+\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{y}$$

对于一个新输入的数据 $\boldsymbol{x}$，模型预测值 $f(\boldsymbol{x})=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}^* = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}(\lambda\boldsymbol{I}+\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{y}$

在引入核函数后，用 k_x 代替 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}$，用 $K(\boldsymbol{X})$ 代替 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$，可得

$$f(\boldsymbol{x}) = k_x[\lambda\boldsymbol{I}+K(\boldsymbol{X})]^{-1}\boldsymbol{y}$$

从这个例子可以看出，通过引入核函数，可以实现更复杂的非线性建模，但不需要把数据的非线性映射函数显式表示出来。

6. 线性回归的扩展

广义线性模型(generalized linear models，GLM)由 Nelder 和 Wedderburn 于 1972 年提出，扩展了普通线性回归模型，使其能够处理因变量与自变量之间呈非线性关系的问题。

广义线性模型是线性模型的推广，其一般形式为

$$g(y) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{3-23}$$

其中，函数 $g(\cdot)$ 称为连接函数(link function)，该函数单调可微，且存在逆函数 $g^{-1}(\cdot)$，使得

$$y = g^{-1}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b) \tag{3-24}$$

特别的，对于一般线性模型，其连接函数满足 $g(y)=y$。

对于对数线性回归，其连接函数满足 $g(y)=\ln y$，此时

$$y = e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b} \tag{3-25}$$

对于对数概率回归(又称为逻辑回归),其连接函数 $g(y)=\ln\frac{y}{1-y}$,此时

$$y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}} \tag{3-26}$$

式中,$y=p(\boldsymbol{x})=f(\boldsymbol{x})=\frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$ 为事件发生的概率,$\frac{y}{1-y}$ 为事件胜算的概率(odds),$g(y)=\ln\frac{y}{1-y}$ 为事件胜算的对数概率。

此外,还可以引入基函数向量 $\boldsymbol{\phi}(\boldsymbol{x})$,增加回归模型的非线性描述能力,此时回归模型为

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b \tag{3-27}$$

其中,基函数向量包括一组非线性映射函数(基函数)

$$\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_K(\boldsymbol{x})] \tag{3-28}$$

每个基函数将输入向量 $\boldsymbol{x}$ 通过非线性函数映射为一个标量值,最终使得输出与特征向量的关系是非线性的,但输出与权重向量 $\boldsymbol{w}$ 的关系仍然是线性的,因此称这种模型为线性基函数回归模型。基函数的类型有很多,常用的有多项式基函数、高斯函数、正余弦函数集等。与基本线性回归相比,只要用 $\phi(\boldsymbol{x})$ 替代 $\boldsymbol{x}$,其他是一致的。

3.1.3　线性分类模型

在利用线性模型实现二分类任务时,相当于利用一个超平面将特征空间分成对应于两个类别的决策区域。该超平面的法线方向为权重向量 $\boldsymbol{w}$,位置由偏置量 b 决定。线性分类器的学习过程主要是求解权重向量 $\boldsymbol{w}$ 和偏置量 b 的过程。

1. 决策面

在映射函数为线性函数的情况下,决策面 $f(\boldsymbol{x})=0$ 退化为高维空间的超平面,并具备如下性质:

假设两个不同的样本点 $\boldsymbol{x}_i$,$\boldsymbol{x}_j$ 都在判定面上,则满足

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_j + b \rightarrow \boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_i - \boldsymbol{x}_j) = 0 \tag{3-29}$$

这表明权重向量为超平面法向量,与超平面上的任意向量正交。

图 3-6 展示了二元输入 $\boldsymbol{x}=(\boldsymbol{x}_1,\boldsymbol{x}_2)^{\mathrm{T}}$ 情况下决策面示意图,此时决策面 $f(\boldsymbol{x})=0$ 如图中实线所示。权重向量 $\boldsymbol{w}$ 为决策面法向量,b 为偏置量。输入向量 $\boldsymbol{x}$ 向权重向量 $\boldsymbol{w}$ 上的投影结果 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ 除以法向量的模长 $\|\boldsymbol{w}\|$,减去负偏置量 $-b$ 除以法向量的模长 $\|\boldsymbol{w}\|$,得到输入向量到决策面的距离 $\frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|}=\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}{\|\boldsymbol{w}\|}$。

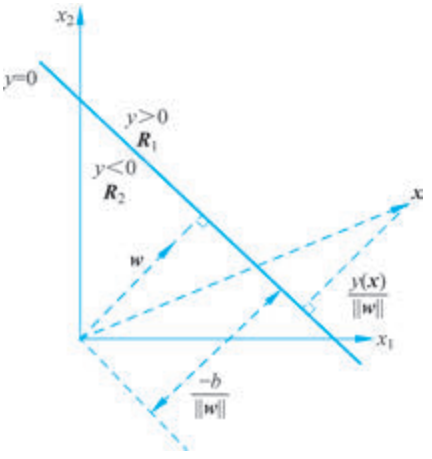


图 3-6　二元输入下的决策面示意图

2. 线性可分与线性不可分

假设我们有特征空间中的 N 个样本数据 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$，其分别属于两个不同的类别 ω_1 和 ω_2。如果在特征空间上存在一个判定超平面，可将上述所有样本数据正确地划分至期望的类别，则称这些样本数据是线性可分的；否则是线性不可分的。数学上表示，假设对于一组样本 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$，存在权重向量 $\boldsymbol{w}$ 和偏置量 b，使得对于所有属于 ω_1 类的样本均满足 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b>0$，对于所有属于 ω_2 类的样本均满足 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b<0$，则称样本是线性可分的；反之，如果不存在满足上述条件的权重向量 $\boldsymbol{w}$ 和偏置量 b，则认为样本是线性不可分的。通常而言，在线性可分的情况下，分类错误率为 0 的权重向量 $\boldsymbol{w}$ 和偏置量 b 并不唯一。

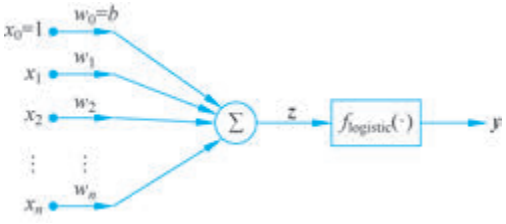


图 3-7　Logistic 回归模型

Logistic 回归（中译逻辑回归、罗吉斯回归、对数几率回归）是统计机器学习中的经典方法，在二分类问题中应用广泛，其模型如图 3-7 所示。Logistic 回归是一种线性分类器，即分类决策面为线性函数，其主要思想是：利用训练集数据对分类边界进行拟合，以此实现分类。因此，Logistic 回归是旨在使用优化方法寻找最优的拟合参数，以解决二分类问题的机器学习方法。

Logistic 回归的建模方式如下，首先对数据进行线性加权计算，再引入 Sigmoid 函数将其变换成二分类问题的概率。

Sigmoid 函数的表达式为

$$\sigma(\boldsymbol{x})=\frac{1}{1+\mathrm{e}^{-\boldsymbol{x}}} \tag{3-30}$$

Sigmoid 函数的导数为

$$\frac{\mathrm{d}\sigma(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}}=\frac{\mathrm{e}^{-\boldsymbol{x}}}{(1+\mathrm{e}^{-\boldsymbol{x}})^2}=\sigma(\boldsymbol{x})[1-\sigma(\boldsymbol{x})]$$

Sigmoid 函数及其导数如图 3-8 所示。Sigmoid 函数的导数最大值为 0.25。

Logistic 回归模型与人工神经元模型一致，包括两步计算。

第一步为线性加权求和：

$$z=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{3-31}$$

第二步利用非线性激活函数得到分类输出结果：

$$f_{\text{logistic}}(z)=\sigma(z)=\frac{1}{1+\mathrm{e}^{-z}} \tag{3-32}$$

Logistic 回归的损失函数采用二分类交叉熵：

$$L=-y\log f(\boldsymbol{x})-(1-y)\log[1-f(\boldsymbol{x})] \tag{3-33}$$

Logistic 回归问题可以用数值优化计算方式求解，得到二分类交叉熵损失的极小值点对应的模型参数。

【例 3-3】　考虑逻辑回归问题：

$$\min f(\boldsymbol{w})=\min\left\{-\sum_{i=1}^{n}\left[y_i\log\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)+(1-y_i)\log(1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i))\right]\right\}$$

设 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]^{\mathrm{T}}\in\mathbb{R}^{n\times d+1}$ 为样本矩阵，$\boldsymbol{Y}=[y_1,y_2,\cdots,y_n]^{\mathrm{T}}\in\mathbb{R}^n$ 为标签向量，$\boldsymbol{w}\in$

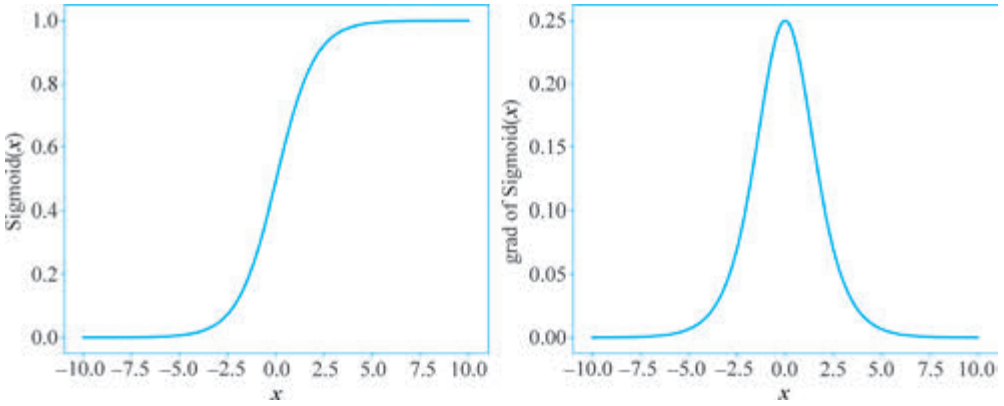


图 3-8　Sigmoid 函数及其导数

$\mathbb{R}^{d+1}$ 为模型参数向量，σ 是 Sigmoid 函数，$\sigma(z)=\dfrac{1}{1+\mathrm{e}^{-z}}$。

推导

$$\frac{\mathrm{d}f(\boldsymbol{w})}{\mathrm{d}\boldsymbol{w}}=\boldsymbol{X}^{\mathrm{T}}[\sigma(\boldsymbol{X}\boldsymbol{w})-\boldsymbol{Y}]$$

式中，$\sigma(\boldsymbol{X}\boldsymbol{w})$ 表示对 $\boldsymbol{X}\boldsymbol{w}$ 中的每个元素应用 σ 函数。

解：通过对 $f(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的导数进行计算，可以得到

$$\begin{aligned}\frac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}}&=-\sum_{i=1}^{n}\left[\frac{y_i}{\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)}-\frac{1-y_i}{1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)}\right]\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)[1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)]\boldsymbol{x}_i\\&=-\sum_{i=1}^{n}\{y_i[1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)]-(1-y_i)\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)\}\boldsymbol{x}_i\\&=-\sum_{i=1}^{n}[y_i-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)]\boldsymbol{x}_i\\&=\sum_{i=1}^{n}[\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)-y_i]\boldsymbol{x}_i\\&=\boldsymbol{X}^{\mathrm{T}}[\sigma(\boldsymbol{X}\boldsymbol{w})-\boldsymbol{Y}]\end{aligned}$$

3.2　支持向量机

支持向量机是在 20 世纪 90 年代兴起的一种统计学习方法，以最大化间隔为核心思想，在小样本条件下有更好的泛化能力。在上述基于线性判别函数的机器学习方法中，优化准则为经验风险最小化，即利用训练样本上的估计风险来代替样本期望风险。在样本不足的情况下，利用训练集上的经验风险最小化进行训练容易造成分类器的过拟合。在支持向量理论中，为保证分类器能获取较好的推广性能，其以结构化风险最小化取代经验风险最小化作为优化准则。结构化风险包括两部分，一部分为经验风险，另一部分为由训练样本数量和模型复杂度决定的风险。其中模型复杂度用 VC 维（Vapnik-Chervonenkis dimension）表示。对于二分类任务，考虑一个指示函数集，如果存在 h 个样本能够被函数集中的函数按

所有可能的 2^h 种形式分开，即函数集能够把 h 个样本打散，那么函数集的 VC 维是其所能打散的最大样本数目 h。

对于线性可分的二分类问题，分类界面的选择存在多种可能，如图 3-9 所示。支持向量机的主要设计目的是使得不同类别的样本分类间隔最大。假设样本-标签对集合为$\{(\boldsymbol{x}_i, y_i)\}, i=1,\cdots,N, y_i \in \{+1,-1\}$，其中 y_i 是 $\boldsymbol{x}_i$ 对应的类别标号。以应用于二类线性分类的支持向量机为例，根据分类决策过程，$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 在大于 0 时被认为属于类别 ω_1（标签：1），在小于 0 时被认为属于类别 ω_2（标签：-1）。因此 $H: \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$ 可视为分类界面。选取距离分类界面最近的正类样本和负类样本，分别位于平行于 H 的两个平面 H_1、H_2 上，这两个平面之间的距离叫作分类间隔（margin），如图 3-10 所示。若分类间隔越大，则分类正确的“安全”距离最大。基于上述考虑，支持向量学习是以最大化分类间隔为优化目标的机器学习方法。也就是说，支持向量机的目的是希望找到最优投影向量 $\boldsymbol{w}$ 和最优线性分类界面 H，使分类间隔尽量大。

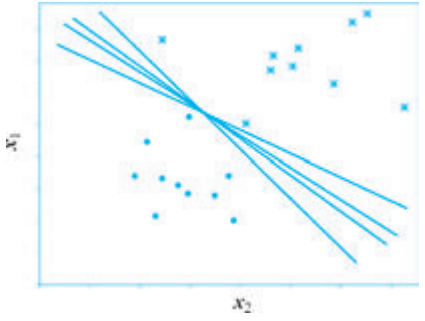


图 3-9　分类界面的选择

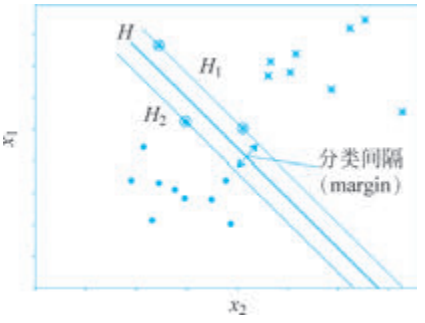


图 3-10　两类线性可分问题

3.2.1　线性可分问题

我们首先以二分类问题展开讨论，后续进一步将其推广至多分类问题。为简化讨论，首先假设样本集线性可分，如图 3-10 所示。在线性可分的条件下，存在权重向量 $\boldsymbol{w}$ 和偏置量 b 满足 $\forall i$：$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)>0$。

记正类、负类样本与理想决策面的最小距离分别为 D^+（$D^+>0$）、D^-（$D^->0$），此时分类判决条件为

$$\begin{cases} y_i=1, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b \geqslant D^+ \\ y_i=-1, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b \leqslant -D^- \end{cases} \tag{3-34}$$

将式(3-34)变形，使不等式右边为 $\dfrac{D^++D^-}{2}$，再对权重向量 $\boldsymbol{w}$ 和偏置量 b 进行调整：

$$\begin{aligned} \boldsymbol{w} &\leftarrow \frac{\boldsymbol{w}}{\dfrac{D^++D^-}{2}} \\ b &\leftarrow \frac{b}{\dfrac{D^++D^-}{2}}+\frac{D^--D^+}{D^++D^-} \end{aligned} \tag{3-35}$$

经过上述调整后，分类判决条件等价于 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1$，即

$$\begin{cases} y_i=1, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\geqslant 1 \\ y_i=-1, \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\leqslant -1 \end{cases} \tag{3-36}$$

将位于 H_1 分类界面上的样本记为 $\boldsymbol{x}^+$，对应的位于 H_2 分类界面上的样本记为 $\boldsymbol{x}^-$，则满足：

$$\begin{cases} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^+ +b=1 \\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^- +b=-1 \end{cases} \tag{3-37}$$

由此变换可得 $\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}^+-\boldsymbol{x}^-)=2$。进一步，记分类界面的法向量为 $\boldsymbol{n}$，则 $\boldsymbol{n}=\dfrac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$。分类界面 H_1 和 H_2 之间的分类间隔 M 为

$$M=(\boldsymbol{x}^+-\boldsymbol{x}^-)\cdot\boldsymbol{n}=(\boldsymbol{x}^+-\boldsymbol{x}^-)\cdot\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}=\frac{2}{\|\boldsymbol{w}\|} \tag{3-38}$$

由此可见，分类间隔 M 是权重向量 $\boldsymbol{w}$ 模值倒数的两倍。支持向量机算法的优化目标是使得分类间隔最大，因此使得分类间隔最大等价于使得权重向量的模值最小。基于上述分析，线性可分条件下的支持向量机可以建模成有约束条件的二次规划问题：

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2 \tag{3-39}$$

约束条件为

$$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1,\quad i=1,2,\cdots,N \tag{3-40}$$

为求解上述有约束条件下的二次规划问题，定义如下的 Lagrange 损失函数：

$$L(\boldsymbol{w},b,\boldsymbol{\alpha})\equiv\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{n}\alpha_i[1-y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)],\quad \alpha_i\geqslant 0, i=1,\cdots,n \tag{3-41}$$

式中，α_i 是对应于每个约束条件的 Lagrange 乘子。

该 Lagrange 损失函数包括 2 项，第 1 项的优化目标为最大化分类间隔，第 2 项的优化目标为最小化误分类点损失。如图 3-11 所示，位于 H_1 分类界面下方的误分类点到 H_1 分类界面的距离为 $\dfrac{1}{\|\boldsymbol{w}\|}[1-y(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)]$。误分类点损失可以用 $\max[0,1-y(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)]$ 来表示，该损失又称为铰链损失或合页损失(hinge loss)。

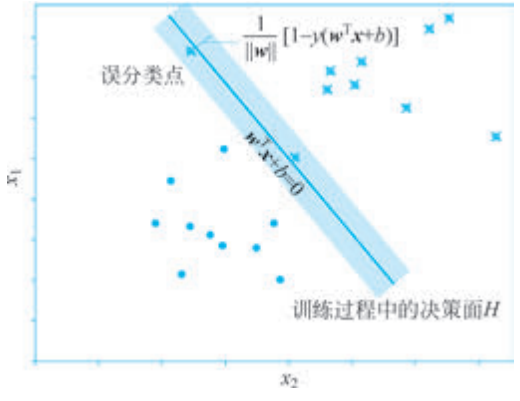


图 3-11　误分类点损失示意图

对于不等式约束条件下的二次函数极值问题，在满足 KKT(Karush-Kuhn-Tucker)条件下，可以交换求解顺序，将原问题 $\min\limits_{\boldsymbol{w},b}[\max\limits_{\boldsymbol{\alpha}}L(\boldsymbol{w},b,\boldsymbol{\alpha})]$ 变为求解对偶问题 $\max\limits_{\boldsymbol{\alpha}}\{\min\limits_{\boldsymbol{w},b}L(\boldsymbol{w},b,\boldsymbol{\alpha})\}=\max\limits_{\boldsymbol{\alpha}}Q(\boldsymbol{\alpha})$。

首先说明 KKT 条件。一般地，对于一个凸优化问题 $\min f(x), g_i(x)\leqslant 0, i=1,2,\cdots,m$，Lagrange 函数可以表示为

$$\mathcal{L}(x,\boldsymbol{\lambda})=f(x)+\sum_{i=1}^{m}\lambda_i g_i(x) \tag{3-42}$$

式中，$f(x)$ 是目标函数；$g_i(x)\leqslant 0$ 是约束条件；$\lambda_i\geqslant 0$ 是拉格朗日乘子。当最优解在不等式约束可行域中时，该凸优化问题相当于无约束优化问题；当最优解不在不等式约束可行域中时，该凸优化问题等价于等式约束条件优化问题。

凸优化问题的 KKT 条件可以总结如下：

(1) 原始优化问题约束条件：$g_i(x)\leqslant 0, i=1,2,\cdots,m$；

(2) 对偶问题约束条件：$\lambda_i\geqslant 0, i=1,2,\cdots,m$；

(3) 互补松弛条件：$\lambda_i g_i(x)=0, i=1,2,\cdots,m$。

那么对于式(3-41)的 Lagrange 损失函数，相应地，线性支持向量机的 KKT 条件包括以下几方面：

(1) $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1$，即 $1-y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\leqslant 0, i=1,2,\cdots,N$；

(2) $\alpha_i\geqslant 0, i=1,2,\cdots,N$；

(3) $\alpha_i[1-y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)]=0, i=1,2,\cdots,N$。

在满足上述 KKT 条件的情况下，对于对偶问题 $\max\limits_{\boldsymbol{\alpha}}[\min\limits_{\boldsymbol{w},b}L(\boldsymbol{w},b,\alpha)]$，先求解 $\min\limits_{\boldsymbol{w},b}L(\boldsymbol{w},b,\boldsymbol{\alpha})$，将 $L(\boldsymbol{w},b,\boldsymbol{\alpha})$ 分别对 $\boldsymbol{w}$、b 求导，并令导数为 0，得到：

$$\boldsymbol{w}=\sum_{i=1}^{N}\alpha_i y_i\boldsymbol{x}_i \tag{3-43}$$

$$\sum_{i=1}^{N}\alpha_i y_i=0 \tag{3-44}$$

代入式(3-41)，得到对偶问题：

$$\begin{cases}\max\limits_{\boldsymbol{\alpha}}Q(\boldsymbol{\alpha})=\sum\limits_{i=1}^{N}\alpha_i-\dfrac{1}{2}\sum\limits_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j\\ \text{约束条件：}\sum\limits_{i=1}^{N}\alpha_i y_i=0,\alpha_i\geqslant 0, i=1,\cdots,N\end{cases} \tag{3-45}$$

式中，$\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$ 表示样本数据 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 之间的内积。

该问题可以利用 1998 年 John C. Platt 提出的序列最小优化算法(sequential minimal optimization，SMO)迭代求解。在每次迭代中，选择两个拉格朗日乘子(如 α_1 和 α_2)作为待优化参数，其他参数固定不变，即 $\alpha_1 y_1+\alpha_2 y_2=-\sum\limits_{i=3}^{N}\alpha_i y_i$，然后进行优化求解。

记 $\boldsymbol{\alpha}^*$ 为上述优化问题的最优解，则权重向量 $\boldsymbol{w}^*$ 的最优解可以计算如下：

$$\boldsymbol{w}^*=\sum_{i=1}^{N}\alpha_i^* y_i\boldsymbol{x}_i=\sum_{j\in \mathrm{SV}}\alpha_j y_j\boldsymbol{x}_j \tag{3-46}$$

式中，SV 表示求解得到的支持向量集合，即 $\alpha_i>0$ 对应的样本点的集合。上述对偶问题是

在不等式约束条件下的二次函数极值问题。

又根据 KKT 条件可知，最优超平面需要满足的条件是：

$$\alpha_i [1 - y_i (\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b)] = 0 \tag{3-47}$$

结合上述分类界面的定义可知，式(3-47)存在如下两种情况：

(1) 样本$(\boldsymbol{x}_i, y_i)$在 H_1、H_2 界面上时，$y_i (\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b) = 1$，$\alpha_i > 0$；

(2) 样本$(\boldsymbol{x}_i, y_i)$在 H_1、H_2 界面外时，$y_i (\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b) > 1$，$\alpha_i = 0$。

我们将在临界分类界面 H_1、H_2 上的样本称为支持向量，其对应的 $\alpha_i > 0$。在得到权重向量的基础上，任选一支持向量，可以求得偏置量 b。在实际中，由于只有少量样本对应的 $\alpha_i > 0$，因此支持向量机分类的权重向量和分类界面通常由训练集中的很少一部分样本决定，从而使得支持向量机适合处理小样本分类问题。

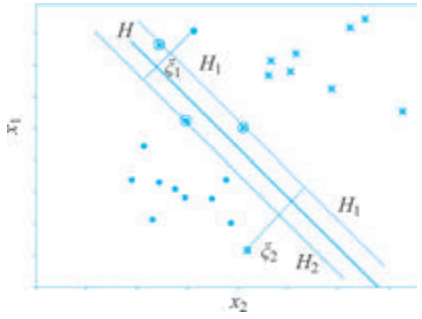


图 3-12　两类近似线性可分支持向量机问题

3.2.2　近似线性可分问题

线性可分是分类问题中非常理想的情况。面向实际应用的需要，我们进一步将上述理论推广至解决近似线性可分问题。如图 3-12 所示为两类近似线性可分支持向量机问题，此时，需要在约束条件中添加松弛变量 ξ_i，代表样本 $\boldsymbol{x}_i$ 跨过其实际类别分类界面的距离。将上述模型优化为

$$\begin{cases} \min \dfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum\limits_{i=1}^{n} \xi_i^q \\ \text{约束条件：} y_i (\boldsymbol{w}^{\mathrm{T}} x_i + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0 \end{cases} \tag{3-48}$$

式中，$q \geqslant 1$(通常 $q=1$ 或 $q=2$；$q=1$ 时为线性惩罚，$q=2$ 时为平方惩罚，对较大的偏离进行更重的惩罚；为简化后续推导，假设 $q=1$)；正则化参数 $C>0$，用于控制模型拟合能力。正则化参数允许模型有误差，用于控制允许部分样本位于间隔区间。C 越大，即模型容忍的误差越小，进入间隔区间的样本点越少，容易过拟合；C 越小，即模型容忍的误差越大，进入间隔区间的样本点越多，容易欠拟合。

式(3-48)的优化问题同样是线性约束条件下的二次规划问题。类似地，引入 Lagrange 函数：

$$\begin{cases} L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}) \equiv \dfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum\limits_{i=1}^{n} \xi_i + \sum\limits_{i=1}^{n} \alpha_i [1 - \xi_i - y_i (\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_i + b)] - \sum\limits_{i=1}^{n} \beta_i \xi_i \\ \text{约束条件：} \alpha_i \geqslant 0, \beta_i \geqslant 0 \end{cases} \tag{3-49}$$

式中，α_i、β_i 为对应于每个约束条件的 Lagrange 乘子。根据 KKT 条件和对偶理论，可以得到原优化问题的对偶问题为

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) = \sum\limits_{i=1}^{n} \alpha_i - \dfrac{1}{2} \sum\limits_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \\ \text{约束条件：} \sum\limits_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \cdots, n \end{cases} \tag{3-50}$$

相比于式(3-45)线性可分条件下的对偶优化问题，近似线性可分条件下的对偶问题求解需要满足 $\alpha_i \leqslant C$，即 α_i 有上界。上述对偶问题求解 α_i 确定之后，权重向量和偏置量的求解同线性可分情况相同。

在近似线性可分情况下，分类界面的参数 $\boldsymbol{w}$、b 由对应于 $\alpha_i > 0$ 的训练样本决定，这些样本称为支持向量，且满足 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 1 - \xi_i \leqslant 1$。

将式(3-48)进行变形，得到的优化问题为

$$\begin{cases} \min \dfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum\limits_{i=1}^{n} \xi_i \\ \text{约束条件：} \xi_i \geqslant \max(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b), 0) \end{cases} \tag{3-51}$$

此时，Lagrange 损失函数为

$$\begin{cases} L(\boldsymbol{w}, b, \boldsymbol{\gamma}, \boldsymbol{\xi}) \equiv \dfrac{1}{2} \| \boldsymbol{w} \|^2 + C \sum\limits_{i=1}^{n} \xi_i + \sum\limits_{i=1}^{n} \gamma_i [\max(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b), 0) - \xi_i] \\ \text{约束条件：} \gamma_i \geqslant 0 \end{cases} \tag{3-52}$$

由 $\dfrac{\partial L}{\partial \xi_i} = C - \gamma_i = 0$ 可知 $\gamma_i = C$，代入式(3-52)可得

$$L(\boldsymbol{w}, b) \equiv \frac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^{n} \max(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b), 0) \tag{3-53}$$

其中，$\max(1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b), 0)$ 即 3.2.1 节提到的铰链损失。损失函数 $L(\boldsymbol{w}, b)$ 常用于支持向量机的神经网络实现，可以视为权重系数的 L2 范数与带正则化参数的铰链损失之和。

记 $\hat{y}_i = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b$，式(3-53)可以写成

$$L(\boldsymbol{w}, b) \equiv \frac{1}{2} \| \boldsymbol{w} \|^2 + C \sum_{i=1}^{n} \max(1 - y_i \hat{y}_i, 0) \tag{3-54}$$

其误差反向传播过程的梯度为

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} \\ \dfrac{\partial L}{\partial \hat{y}_i} = C\, \mathbb{I}(1 - y_i \hat{y}_i)(-y_i) \end{cases} \tag{3-55}$$

其中，$\mathbb{I}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geqslant 0 \end{cases}$。

3.2.3 非线性分类问题

支持向量机可以处理线性可分和近似线性可分问题，但实际的数据通常是线性不可分的。为解决非线性分类问题，一个通常的解决思路是将样本的特征映射到高维空间，使之在高维空间中线性可分，如图 3-13 所示。假设映射函数形式为 $\Phi: \boldsymbol{x} \to \phi(\boldsymbol{x})$，则式(3-50)的对偶优化问题可以表示为

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) = \sum\limits_{i=1}^{n} \alpha_i - \dfrac{1}{2} \sum\limits_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^{\mathrm{T}} \phi(\boldsymbol{x}_j) \\ \text{约束条件：} \sum\limits_{i=1}^{n} \alpha_i y_i = 0, 0 \leqslant \alpha_i \leqslant C, i = 1, \cdots, n \end{cases} \tag{3-56}$$

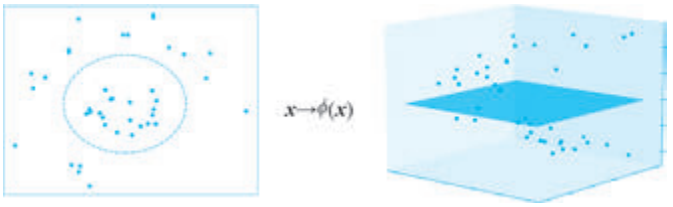


图 3-13　线性不可分问题：映射函数与高维空间

式中，$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$表示样本数据 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 在高维空间中的内积。引入核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x}_j)$，避免寻求低维到高维空间的显式映射函数。基于核函数的支持向量机的对偶优化问题可表示为

$$\begin{cases}\max\limits_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha})=\sum\limits_{i=1}^{n}\alpha_i-\dfrac{1}{2}\sum\limits_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)\\ \text{约束条件：}\sum\limits_{i=1}^{n}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant C,i=1,\cdots,n\end{cases}\tag{3-57}$$

结合式(3-46)的权重向量表达式，我们可以将非线性分类决策函数对应地表示为核函数的形式：

$$(\boldsymbol{w}^{*})^{\mathrm{T}}\phi(\boldsymbol{x})+b=\sum_{j\in \mathrm{SV}}\alpha_j y_j\phi(\boldsymbol{x}_j)^{\mathrm{T}}\phi(\boldsymbol{x})+b=\sum_{j\in \mathrm{SV}}\alpha_j y_j K(\boldsymbol{x}_j,\boldsymbol{x})+b\tag{3-58}$$

式中，SV 代表根据式(3-57)求解出的支持向量集合。

由上述推导可见，决策函数仅和核函数有关，不需要显式给出从输入空间到高维特征空间的映射函数，从而避免高维显式建模带来的计算复杂度，降低了模型的 VC 维，缓解了小样本条件下模型容易在训练集上过拟合的现象。通过引入核函数，支持向量机可以有效解决线性不可分问题。

常用核函数包括：

线性核函数：$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$

多项式核函数：$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(1+\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^{d}$，其中 d 为多项式阶数

高斯核函数：$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$

Sigmoid 核函数：$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\tanh(\beta_0\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j+\beta_1)$，其中 $\tanh(z)=\dfrac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

核函数对应的基函数不一定能显式写出，如高斯核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$ 对应的基函数 $\phi(\boldsymbol{x})$ 的维度为无穷维。为便于分析，假设 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 为标量，根据 e^{x} 在 $x=0$ 处的泰勒展开公式为

$$e^{x}=1+x+\frac{x^2}{2!}+\cdots+\frac{x^n}{n!}+\cdots\tag{3-59}$$

将高斯核函数进行展开得

$$K(x_i,x_j)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right)$$

$$=\exp\left(-\frac{x_i^2+x_j^2-2x_ix_j}{2\sigma^2}\right)=\exp\left(-\frac{x_i^2+x_j^2}{2\sigma^2}\right)\exp\left(\frac{x_ix_j}{\sigma^2}\right)$$

$$=\exp\left(-\frac{x_i^2+x_j^2}{2\sigma^2}\right)\left(1\times1+\frac{x_i}{\sigma}\times\frac{x_j}{\sigma}+\frac{1}{2!}\times\frac{x_i^2}{\sigma^2}\times\frac{x_j^2}{\sigma^2}+\cdots+\frac{1}{n!}\times\frac{x_i^n}{\sigma^n}\times\frac{x_j^n}{\sigma^n}+\cdots\right)$$

$$=\phi(x_i)^{\mathrm{T}}\phi(x_j) \tag{3-60}$$

其中

$$\phi(x)=\exp\left(-\frac{x^2}{2\sigma^2}\right)\left[1,\frac{x}{\sigma},\frac{1}{\sqrt{2!}}\times\frac{x^2}{\sigma^2},\cdots,\frac{1}{\sqrt{n!}}\times\frac{x^n}{\sigma^n},\cdots\right] \tag{3-61}$$

由此可见，高斯核函数对应的基函数的维度为无穷维。

【例 3-4】 异或(XOR)问题如表 3-2 所示，试利用非线性支持向量机求解。

表 3-2 异或(XOR)问题

输入向量 x	二 维 向 量	输出真值 y
$\boldsymbol{x}_1$	$(-1,-1)^{\mathrm{T}}$	-1
$\boldsymbol{x}_2$	$(-1,1)^{\mathrm{T}}$	1
$\boldsymbol{x}_3$	$(1,-1)^{\mathrm{T}}$	1
$\boldsymbol{x}_4$	$(1,1)^{\mathrm{T}}$	-1

解：采用 2 阶多项式核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(1+\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^2$，其中 $\boldsymbol{x}_i=(x_{i1},x_{i2})^{\mathrm{T}}$。因此有

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=1+x_{i1}^2x_{j1}^2+2x_{i1}x_{i2}x_{j1}x_{j2}+x_{i2}^2x_{j2}^2+2x_{i1}x_{j1}+2x_{i2}x_{j2}$$

核函数的矩阵形式，即核矩阵为

$$K=\begin{bmatrix}K(\boldsymbol{x}_1,\boldsymbol{x}_1) & K(\boldsymbol{x}_1,\boldsymbol{x}_2) & K(\boldsymbol{x}_1,\boldsymbol{x}_3) & K(\boldsymbol{x}_1,\boldsymbol{x}_4)\\ K(\boldsymbol{x}_2,\boldsymbol{x}_1) & K(\boldsymbol{x}_2,\boldsymbol{x}_2) & K(\boldsymbol{x}_2,\boldsymbol{x}_3) & K(\boldsymbol{x}_2,\boldsymbol{x}_4)\\ K(\boldsymbol{x}_3,\boldsymbol{x}_1) & K(\boldsymbol{x}_3,\boldsymbol{x}_2) & K(\boldsymbol{x}_3,\boldsymbol{x}_3) & K(\boldsymbol{x}_3,\boldsymbol{x}_4)\\ K(\boldsymbol{x}_4,\boldsymbol{x}_1) & K(\boldsymbol{x}_4,\boldsymbol{x}_2) & K(\boldsymbol{x}_4,\boldsymbol{x}_3) & K(\boldsymbol{x}_4,\boldsymbol{x}_4)\end{bmatrix}=\begin{bmatrix}9&1&1&1\\1&9&1&1\\1&1&9&1\\1&1&1&9\end{bmatrix}$$

事实上，引入核函数并不需要得到基函数 $\phi(\boldsymbol{x})$的显式表示。为了便于理解，在本例中，也可以给出根据所选取的核函数推导出的基函数 $\phi(\boldsymbol{x})$，也就是二维输入向量在六维特征空间中的映射，即

$$\phi(\boldsymbol{x}_i)=\left[1,x_{i1}^2,\sqrt{2}x_{i1}x_{i2},x_{i2}^2,\sqrt{2}x_{i1},\sqrt{2}x_{i2}\right]^{\mathrm{T}}$$

此时，可以分别计算出每个样本映射到六维特征空间中的向量

$$\phi(\boldsymbol{x}_1)=\left[1,1,\sqrt{2},1,-\sqrt{2},-\sqrt{2}\right]^{\mathrm{T}}$$

$$\phi(\boldsymbol{x}_2)=\left[1,1,-\sqrt{2},1,-\sqrt{2},\sqrt{2}\right]^{\mathrm{T}}$$

$$\phi(\boldsymbol{x}_3)=\left[1,1,-\sqrt{2},1,\sqrt{2},-\sqrt{2}\right]^{\mathrm{T}}$$

$$\phi(\boldsymbol{x}_4)=\left[1,1,\sqrt{2},1,\sqrt{2},\sqrt{2}\right]^{\mathrm{T}}$$

结合式(3-57)写出目标函数

$$\theta(\boldsymbol{\alpha})=\sum_{i=1}^{4}\alpha_i-\frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4}\alpha_i\alpha_jy_iy_jK(\boldsymbol{x}_i,\boldsymbol{x}_j)$$

代入样本点数据，并依次令$\frac{\partial\theta(\boldsymbol{\alpha})}{\partial\alpha_1}=0$，$\frac{\partial\theta(\boldsymbol{\alpha})}{\partial\alpha_2}=0$，$\frac{\partial\theta(\boldsymbol{\alpha})}{\partial\alpha_3}=0$，$\frac{\partial\theta(\boldsymbol{\alpha})}{\partial\alpha_4}=0$，得到方程组

$$\begin{cases}9\alpha_1-\alpha_2-\alpha_3+\alpha_4=1\\-\alpha_1+9\alpha_2+\alpha_3-\alpha_4=1\\-\alpha_1+\alpha_2+9\alpha_3-\alpha_4=1\\\alpha_1-\alpha_2-\alpha_3+9\alpha_4=1\end{cases}$$

求解得到拉格朗日系数的最优解为

$$\alpha_1=\alpha_2=\alpha_3=\alpha_4=\frac{1}{8}$$

由于每个拉格朗日系数均大于0，因此本例中的四个样本都是支持向量。

根据权重向量$\boldsymbol{w}$的计算公式$\boldsymbol{w}=\sum_{i=1}^{N}\alpha_i y_i\phi(\boldsymbol{x}_i)$，得到权重向量$\boldsymbol{w}=\left[0,0,-\frac{1}{\sqrt{2}},0,0,0\right]^{\mathrm{T}}$。然后，取一个支持向量$\boldsymbol{x}=\boldsymbol{x}_1$，$\phi(\boldsymbol{x}_1)=\left[1,1,\sqrt{2},1,-\sqrt{2},-\sqrt{2}\right]^{\mathrm{T}}$，代入$y=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b$，计算求得偏置量$b=0$；或者，利用具有核函数的决策函数$y=\sum_{j\in\mathrm{SV}}\alpha_j y_j K(\boldsymbol{x}_j,\boldsymbol{x})+b$，取一个支持向量$\boldsymbol{x}=\boldsymbol{x}_1$，计算求得偏置量$b=0$。

由于高维空间中的决策面方程为$g(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b=0$，对于输入向量$\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$，代入$\phi(\boldsymbol{x})=[1,x_1^2,\sqrt{2}x_1x_2,x_2^2,\sqrt{2}x_1,\sqrt{2}x_2]^{\mathrm{T}}$和$\boldsymbol{w}=\left[0,0,-\frac{1}{\sqrt{2}},0,0,0\right]^{\mathrm{T}}$，计算得到决策面$g(\boldsymbol{x})=-x_1x_2=0$。图3-14分别给出了原始数据空间和六维空间中$\sqrt{2}x_1x_2$、$\sqrt{2}x_1$二维张成子空间中的决策面，分别对应非线性分类和线性分类的结果。

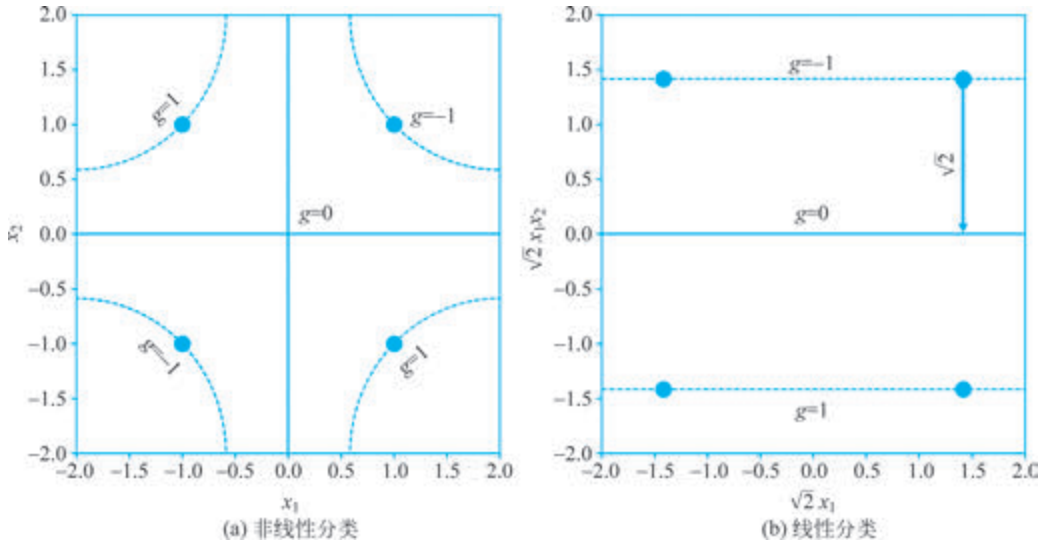


图3-14　原始数据空间和六维空间中$\sqrt{2}x_1x_2$、$\sqrt{2}x_1$二维张成子空间中的决策面

3.2.4　基于支持向量机的多分类任务

上述支持向量机算法是基于二分类问题开展推导和讨论的。对于多分类任务，需要将

多分类问题转换成二分类问题。一般而言，有两种转换策略，分别称为一对一和一对多。一对一策略指的是在任意两类样本之间设计并学习一个两类分类器。假设共有 K 类，通过在 K 类中任选两类进行组合，总共需要学习 C_K^2 个两类分类器。在测试时，对每一个未知样本都使用所有的两类分类器进行分类并对样本所属类别进行投票，然后选取得票最多的类别作为该样本的预测类别。一对多策略指的是对每个类别进行设计并学习一个是否属于该类别的两类分类器。在训练时，该分类器将所对应类别的样本作为正样本，其他所有类别的样本作为负样本进行训练。在测试时，选取计算获取的间隔最大的分类器所属类别作为未知样本的类别。相比于一对一策略，一对多策略可节省计算资源的开销。

3.3 集成学习方法

3.3.1 决策树

决策树是一种无须对数据统计分布进行假设的非参数方法，能够处理包含连续特征和离散特征的数据，不需要对数据进行特殊的预处理。决策树通过构建树状结构对基于数据的决策过程进行建模。

决策树用非叶子节点表示特征。在分类任务中，用叶子节点表示类别标签，称为分类树；在回归任务中，叶子节点上的预测值为连续值，称为回归树。

以分类任务为例，在节点划分时，我们希望决策树的分支节点所包含的样本尽可能属于同一类别，即节点“纯度”越来越高，从而实现从根节点到叶子节点的快速检索以及决策结果的快速获取。决策树的构建过程包括两个步骤：

1. 逐层确定节点特征以及分支规则

在决策树的每个非叶子节点，需要选取特征并确定基于特征的划分规则。通过对划分规则的判定将训练样本划分为子集。为确定每个节点的最优划分，需要对节点训练样本的分类不纯度或不确定度进行度量，即我们希望选取的节点划分规则可以使得其两个子节点的样本纯度最高。记节点的不纯度为 $I(\text{impurity})$。常用的不纯度度量包括如下三种：

熵不纯度(entropy)：

$$I(n) = -\sum_j p_j \log_2 p_j \tag{3-62}$$

基尼不纯度(gini)：

$$I(n) = 1 - \sum_j p_j^2 \tag{3-63}$$

误分类不纯度：

$$I(n) = 1 - \max_j p_j \tag{3-64}$$

为选取最优规则，需要分别对父节点和划分后的子节点的不纯度分别进行度量，并选取不纯度落差(信息增益)最大的划分规则作为节点的划分规则。

2. 确定终止节点，并为终止节点赋予类别标签

在单个节点的不纯度低于一定阈值时，可以认为该节点的样本属于某个类别的概率已经非常高，从而终止该节点的划分，将其设为叶子节点，并根据先验概率最高的类别赋予类别标签。

决策树能够提供直观的机器学习建模的解释和可视化。决策树在应用中的不足主要体现在两个方面：一是容易过拟合，决策树通过多层多节点判定形成非线性的判别函数，当划分得太细时，决策树会对训练集数据产生过拟合；二是决策树的分类具有不稳定性，容易受到训练数据不均衡的影响。

过拟合问题可以通过剪枝来解决。剪枝方法可分为前剪枝和后剪枝，前剪枝是指在构造树的过程中对节点进行剪枝；后剪枝是指构造出完整的决策树之后再对那些子树进行剪枝。此外，也可以考虑引入集成学习方法，组合多个分类器以提升性能。

3.3.2　Bagging 方法

随机森林算法是一种特征选择分类/回归算法，其根据集成学习(ensemble learning)中的 Bagging 思路构建。如图 3-15 所示，随机森林由多棵决策树或者回归树构成，每棵树均为独立训练获取。根据 Bagging 的思路，每棵决策树/回归树并不利用所有的训练样本进行训练，而是通过随机选取训练样本的子集进行训练，且该随机选取为放回抽样，因此单个训练样本可能被用于随机森林中的多棵决策树/回归树的训练。传统的树状分类/回归算法在训练结束之后需要引入“剪枝”过程以去除效果较差的节点，而对随机树进行剪枝可能增加树之间的相关性，从而影响随机森林的泛化性能。其算法步骤如下：

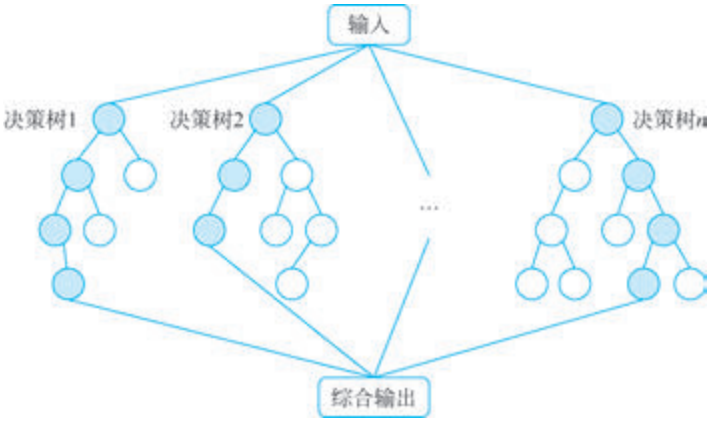


图 3-15　随机森林示意

(1) 假设有 N 个训练样本构成的样本集，从整个样本集中随机选取 n 个样本作为训练单棵随机树的子样本集。

(2) 利用该子样本集训练未剪枝的分类树或者回归树；对于每棵树的每个节点，随机从所有 M 个特征中抽取 m 个特征($m \ll M$)，并且从中选择分类效果或者回归误差最小的特征作为该树节点的特征并获取二值分类器。

(3) 重复步骤(1)和(2)直至训练随机树数目达到指定要求。

(4) 对于测试数据，融合所有分类或回归树的预测结果。对于分类树，选择得票数最多的样本类别；而对于回归树，则是取所有回归结果的平均。

理论上可以证明在单个分类器/回归函数不稳定时，Bagging 方法是有效的。对于支持向量机等算法，使用不同子样本集或不同特征获取的分类器相关性强，因此使用 Bagging 方法并不能有效提升性能。对于决策树/回归树，在每个节点通过选取不同的特征，样本集会

有不同的划分，因此不同子样本集获取的分类树/回归树之间的相关性较弱，此时利用 Bagging 方法可以有效提升分类或回归性能。Liaw 等证明了随机森林的泛化误差(generalization error)随着树的数目增加而减少并渐近逼近 0；相反，若树之间的相关系数增加，则泛化性能变差。

随机森林算法是机器学习中应用广泛的一种算法。其优点在于可以获取很好的分类或回归性能，并且能处理输入变量较多的情况，适用于以集合形式定义的弱特征并且容易进行多特征融合。由于每棵随机树和树的每个分支并不使用所有的特征，随机森林算法可以处理样本数据缺失问题。相对于其他组合算法如 Boosting 算法，随机森林算法的训练速度较快。随机森林算法也存在一些缺点，在噪声较大、树设置过深或训练样本不足等情况下容易过拟合、不适用于标签带噪声的训练样本。

3.3.3 Boosting 方法

1. Boosting 原理

Boosting 方法是另一种典型的集成学习方法。Boosting 通过对训练样本的重采样依次学习多个分类器，并将其组合在一起形成分类性能强的分类器。在此以经典的 AdaBoost (adaptive boosting)算法为例介绍 Boosting 算法的主要思路。给定训练集 $\{(\boldsymbol{x}_i, y_i)\}$，$i=1,\cdots,n$，$y_i \in \{+1,-1\}$ 和弱学习分类器 $h: X \to \{+1,-1\}$。在初始化阶段所有训练样本的权重相同，使用弱学习分类器在训练集上赋予不同的权重，利用权重信息训练 T 轮弱分类器；在每轮训练后，Boosting 对正确分类的样本降低权重和对错误分类的样本增加权重，从而使后续训练的弱分类器集中对分类难度大(高权重)的样本进行学习；通过迭代训练获取 T 个弱分类器 $h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_T(\boldsymbol{x})$，根据每个弱分类器的训练集错误率确定加权系数，组合成最终的强分类器 $H(\boldsymbol{x})$。具体的流程如算法 3-2 所示。由于样本的权重可视作概率分布，因此 Boosting 主要通过重采样在原训练集的基础上生成一个新的训练集，然后使用该训练集训练弱分类器。

算法 3-2　AdaBoost 算法

给定训练集 $L=\{(\boldsymbol{x}_i, y_i)\}$，$i=1,2,\cdots,n$，$y_i \in \{+1,-1\}$ 是样本类别标号，$\boldsymbol{x}_i \in X$ 是样本特征

样本的初始权重 $D_1(i)=\dfrac{1}{n}$，$i=1,2,\cdots,n$

迭代次数 $t=1,2,\cdots,T$：

在样本集的 D_t 上训练弱分类器，得到 $h_t: X \to \{+1,-1\}$

(1) 计算弱分类器的错误率 $\varepsilon_t = \sum_i D_t(i) \dfrac{(1-y_i h_t(\boldsymbol{x}_i))}{2}$

(2) 选择参数 $\alpha_t = \dfrac{1}{2}\log\left(\dfrac{1-\varepsilon_t}{\varepsilon_t}\right)$

(3) 更新样本的权重 $D_{t+1}(i)=\dfrac{D_t(i)\exp[-\alpha_t y_i h_t(\boldsymbol{x}_i)]}{Z_t}$，其中 $Z_t = \sum_i D_t(i)\exp[-\alpha_t y_i h_t(\boldsymbol{x}_i)]$ 是规一化因子

输出最后的强分类器：$H(\boldsymbol{x}) = \operatorname{sign}\left[\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right]$

Freund、Schapire 等分析并证明了 AdaBoost 算法训练得到的强分类器 $H(\boldsymbol{x})$是一个以最小化训练集错误为目标函数的优化过程的结果。此外，Friedman 从 Logistic 回归的角度证明了可以利用得到的强分类器 $H(\boldsymbol{x})$估计模式的后验概率，如下所示：

$$\frac{e^{H(\boldsymbol{x})}}{e^{H(\boldsymbol{x})}+e^{-H(\boldsymbol{x})}} \mapsto P(y=1 \mid \boldsymbol{x}), \frac{e^{-H(\boldsymbol{x})}}{e^{H(\boldsymbol{x})}+e^{-H(\boldsymbol{x})}} \mapsto P(y=-1 \mid \boldsymbol{x}) \tag{3-65}$$

2. 用于人脸检测的级联 AdaBoost 分类器

Viola 和 Jones 提出的基于级联 AdaBoost 分类器和类 Haar(Haar-like)特征的方法已成功应用于人脸检测任务，该方法的优势在于：①利用可快速计算的类 Haar 特征提取人脸鉴别性模式；②结合 AdaBoost 算法和基于单特征的弱分类器构造区分目标/非目标区域的强分类器；③针对目标检测稀疏分类的特点，利用级联结构 Bootstrap 背景窗口图像以提升检测性能并提高检测速度。

该方法首先将训练样本图像大小归一化为 24×24 像素，然后提取类 Haar 特征。类 Haar 特征使用灰度图像上的相邻矩形区域进行定义。图 3-16(a)为该方法所使用的四种类 Haar 特征，其中 A、B 分别检测水平和垂直方向的边缘特征，C 用于检测线状特征，而 D 用于检测对角线特征。类 Haar 特征的特征值通过白色区域的像素值之和减去黑色区域的像素值之和计算提取。为提高特征计算的速度，引入“积分图像”(integral image)。假设图像中(m,n)点的像素值为 $i(m,n)$，则积分图像在该点的像素值 $ii(m,n)$定义为

$$ii(m,n)=\sum_{m'\leqslant m,n'\leqslant n} i(m',n') \tag{3-66}$$

如图 3-16(b)所示，D 矩形区域的像素值之和可通过四个顶点的积分图像像素值加减获取，为

$$ii(\text{point4})-ii(\text{point3})-ii(\text{point2})+ii(\text{point1}) \tag{3-67}$$

通过改变矩形框的大小、形状与位置，可获取不同的类 Haar 特征。

由于类 Haar 特征集数量庞大，Viola 和 Jones 提出使用 AdaBoost 算法进行特征选择并构造区分人脸/非人脸的强分类器。设$(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_L,y_L)$为样本图像，其中 $y_l=+1,-1$ 为样本标签，表示图像是人脸图像还是非人脸图像；$\boldsymbol{x}_l$ 则为样本特征向量；L 为训练样本个数。利用算法 3-2 中的 AdaBoost 分类算法迭代特征提取并构建强分类性能。实验表明，AdaBoost 算法在训练过程中能够使得后续弱分类器集中处理已有分类器分错的样本，如图 3-16(c)，从而有效地实现人脸与背景的区分。

虽然单个 AdaBoost 分类器可以获取很低的正负样本分类错误率，但仍不能满足高精度和实时人脸检测的要求。这主要是因为图像的背景窗口数目要远高于目标窗口数目。为解决该问题，Viola 和 Jones 进一步引入级联分类器结构，如图 3-16(d)所示，其将一系列 AdaBoost 强分类器进行串联其中 T 代表是人脸图像，F 代表非人脸图像。假设第 j 个强分类器的检测率为 d_j，虚警率为 f_j，则串联后分类器的检测率和虚警率分别为所有分类器检测率和虚警率的乘积，即

$$d=\prod_j d_j, \quad f=\prod_j f_j \tag{3-68}$$

在利用级联结构的分类器进行检测时，只有通过该结构中所有分类器的窗口才会被认为是目标窗口。因此，在训练级联分类器时，需利用可通过前端分类器的正负样本训练后端

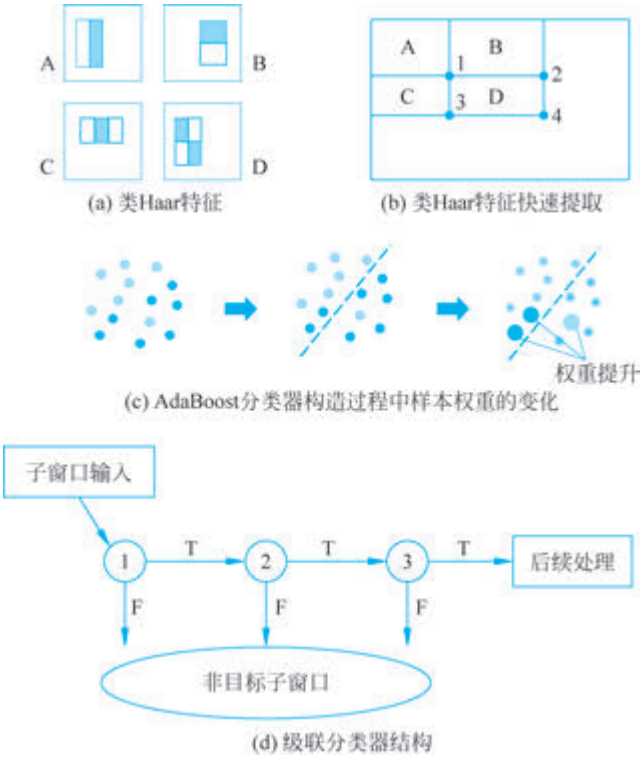


图 3-16　基于级联 AdaBoost 分类器的人脸检测

分类器。在该结构中，前层的分类器可使用较少的特征剔除大量的背景窗口，尽可能在保证高检测率的前提下降低虚警率；而后层的分类器则侧重于处理复杂的背景负样本，完成前层未排除的背景窗口图像与人脸图像之间的分类。由于大部分的负样本图像在前层都可以被排除，因此可大大提高检测速度。

3.3.4　梯度提升决策树

梯度提升决策树(gradient boosting decision tree，GBDT)通过迭代构造一组弱的学习器，并把多棵决策树的预测结果累加起来作为最终的预测输出。每次迭代都以当前模型预测为基准，构造下一个弱分类器去拟合预测值与样本真值之间的残差。GBDT 算法所采用的弱分类器为决策树。

用于回归任务时，GBDT 采用回归树，损失函数为误差平方和，前面加上系数 1/2，用于抵消平方项求导得到的系数。对于第 i 个样本，损失函数为

$$L_i = \frac{1}{2}[y_i - F(\boldsymbol{x}_i)]^2 \tag{3-69}$$

损失函数对预测值的负梯度为

$$r_i = -\frac{\partial L}{\partial F(\boldsymbol{x}_i)} = y_i - F(\boldsymbol{x}_i) \tag{3-70}$$

即为模型预测的残差。也就是说，每次迭代过程，就是构造一个回归树拟合当前的残差。通过多次迭代，使得累加得到的集成模型的性能得到提升。

GBDT 计算流程如算法 3-3 所示。

算法 3-3　梯度提升决策树

输入：训练集 $D=\{(\boldsymbol{x}_i,y_i)\},i=1,2,\cdots,N$

输出：$F_T(\boldsymbol{x})=\sum_{t=0}^{T}f_t(\boldsymbol{x})$

算法步骤：

初始化：初始化第一个回归树模型 $F_0(\boldsymbol{x})=f_0(\boldsymbol{x})$，使得 $\sum_{i=1}^{N}L(y_i,f_0(\boldsymbol{x}_i))$ 最小化

迭代 $t=1,2,\cdots,T$

对每个样本计算负梯度 $r_{i,t}=-\dfrac{\partial L(y_i,F_{t-1}(\boldsymbol{x}_i))}{\partial F_{t-1}(\boldsymbol{x}_i)}$

以 $r_{i,t}$ 为拟合目标建立一个新的回归树 $f_t(\boldsymbol{x})$

更新 $F_t(\boldsymbol{x})=F_{t-1}(\boldsymbol{x})+f_t(\boldsymbol{x})$

用于多分类任务时，GBDT 针对每个类都独立训练一个分类回归树（classification and regression tree，CART），用于判决样本是否属于该类的二分类任务。

CART 方法使用基尼不纯度选择特征，将连续的特征离散化来构建二叉树。具体方法为：样本的连续特征 A 有 m 个值从小到大排列，CART 取相邻两数值的平均数做划分点，一共有 $m-1$ 个划分点；分别计算以这 $m-1$ 个点作为二分类判决阈值时的基尼不纯度；选择基尼不纯度最小的点为该连续特征的二分类判决阈值。

3.4　模型选择与评估

3.4.1　模型的拟合能力

机器学习本质的目标是从数据中发现规律，从而使模型对从未见过的数据也能做出正确的预测。所选择模型的拟合能力至关重要。

1. 模型的容量与选择

模型的拟合能力与模型的容量（model capacity）密切相关，模型的容量通常是指模型能够学习和表示数据复杂模式的能力，它直接影响模型的拟合能力和泛化性能。模型的容量通常与以下几个因素相关：参数数量、模型复杂度和特征多样性。其中，参数数量越多，模型的容量越高，例如，神经网络中层数和每层神经元数量的增加会显著提升模型容量；模型越复杂，通常拥有更高的容量，例如，Transformer 相较于简单的多层感知器拥有更高的模型容量；最后，模型能够使用的特征数量和质量也会影响其容量。

2. 经验误差与泛化误差

为评估模型的性能，通常将模型的预测标签与样本标签真值之间的差异定义为误差（error）。模型在训练集上的误差称为训练误差（training error），也称为经验误差（empirical error）；相应的，在测试集上的误差称为测试误差（testing error），也称为泛化误差（generalization error）。泛化是人类思维的一种重要方式，即将已有的概念或规律推广应用于新的情境。在机器学习中，通常希望得到泛化误差小的模型，也就是在实际样本上泛化性能好的模型。

3. 过拟合、欠拟合与理想模型

为提升模型的泛化能力，需要尽可能地从训练集样本中探索数据本身包含的“普遍规

律”，以在面对未知样本时获得更好的表现。在某些情况下，特别是模型本身复杂度比较高而训练集数据不足的时候，模型对较小规模的训练集数据过度学习，往往会导致模型泛化性能下降。这种现象称为过拟合。过拟合主要表现为：模型在训练集上的经验误差较小，但是在测试集上的泛化误差较大。这通常意味着模型过于复杂，以至于捕捉到了训练集数据中的噪声或随机波动，而不是数据的真实规律。

欠拟合是指模型在训练数据上表现不佳，即模型未能捕捉到数据的潜在规律或模式。欠拟合现象出现的主要原因是模型的学习能力不足，这通常意味着模型的复杂度太低，即函数 $f(\boldsymbol{x})$ 对数据的拟合能力不足，以至于模型无法捕捉到数据中的真实关系，导致在训练数据和测试数据上表现都不佳。

理想模型应该拥有良好的泛化能力和适当的模型容量。良好的泛化能力通常表现为：模型不仅在训练数据上表现优秀，更重要的是能够在新的、未知的数据上保持稳定和准确的性能。适当的模型容量能够确保模型在欠拟合和过拟合之间找到平衡，既能够捕捉数据中的真实规律，又不会受噪声的影响。

图 3-17 展示了二分类模型过拟合、欠拟合与理想模型的示例。其中曲线 1 在训练数据上表现很好，完美地区分了训练数据中的每一个样本，但显然该模型过于关注训练数据中的噪声和随机波动，产生了过拟合现象，并没有捕捉到数据的真实规律。而曲线 2 模型复杂度过低，未能捕捉到数据的潜在规律，产生了欠拟合现象，在训练数据上表现不佳。只有曲线 3 模型捕捉到了训练集潜在的总体规律，拥有良好的泛化能力和适当的模型容量。我们当然希望自己训练的模型能够达到理想模型的效果，但这并非易事，当遇到过拟合、欠拟合现象时应该如何解决，也是十分值得我们关注的。

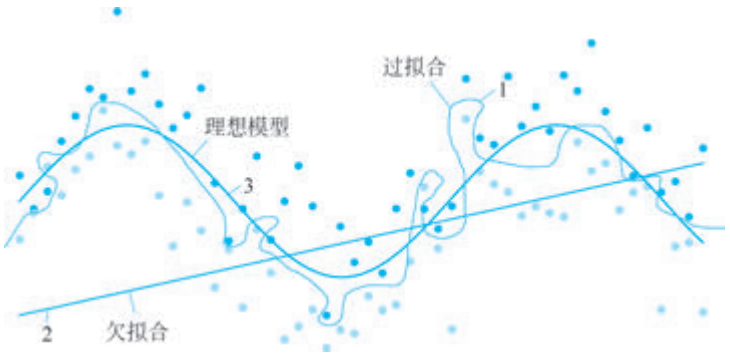


图 3-17　过拟合、欠拟合与理想模型的示例

当出现过拟合时，原因可能有：模型复杂度高、训练数据量少、训练数据噪声多。分析原因后我们可以进行针对性的解决，如简化模型，减少模型的参数数量，选择更简单的模型，或是在模型的损失函数中加入正则化项，如 L1 正则化或 L2 正则化，以限制模型的复杂度；增加数据量，收集更多的训练数据，以支撑复杂模型的训练；数据预处理，清洗训练数据，去除噪声和异常值；在训练过程中监控验证集上的表现，当训练集损失下降而验证集损失开始上升时及时停止训练，如图 3-18 所示，通过早停避免模型进入过拟合区间。

当出现欠拟合时，原因可能有：模型复杂度低、特征选择不当、训练时间不足。分析原因后可以针对性地解决，如复杂化模型，增加模型的参数数量，选择更复杂的模型；通过特征工程，选择更多的特征或进行特征变换，以更好地描述数据中的关系；增加训练时间，当

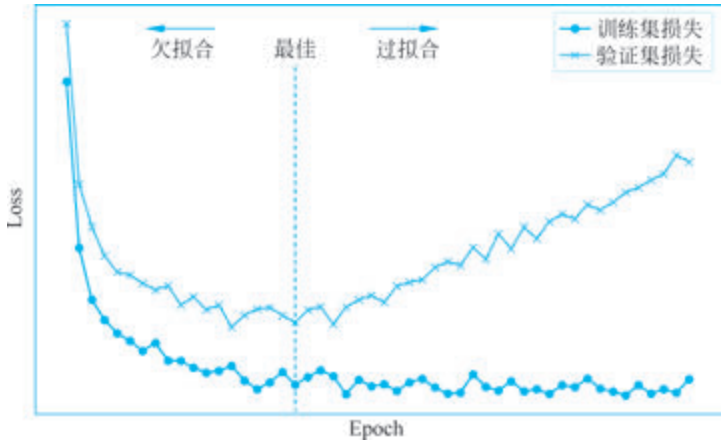


图 3-18　训练过程中根据验证集损失的变化决定是否提前停止训练

发现训练集和验证集损失还有下降趋势时，延长模型的训练时间，以充分学习数据中的关系；调整学习率，确保模型能够收敛到最优解。

4. 偏差与方差

在训练阶段，机器学习的目的是使训练集上的经验误差减小。通过对模型的期望误差进行偏差和方差分解，可对模型的性能进行进一步分析。

为便于分析，考虑 y 为标量的情形，由于观测数据受随机噪声的影响，模型的真实函数形式为 $y=\boldsymbol{X}\boldsymbol{w}+\epsilon$，$\boldsymbol{X}\in\mathbb{R}^{1\times(d+1)}$，$\boldsymbol{w}\in\mathbb{R}^{(d+1)\times 1}$，$\epsilon\sim N(0,\sigma^2)$，令 $t=\boldsymbol{X}\boldsymbol{w}$，相当于观测数据的理论真实值，则有 $y=t+\epsilon$。

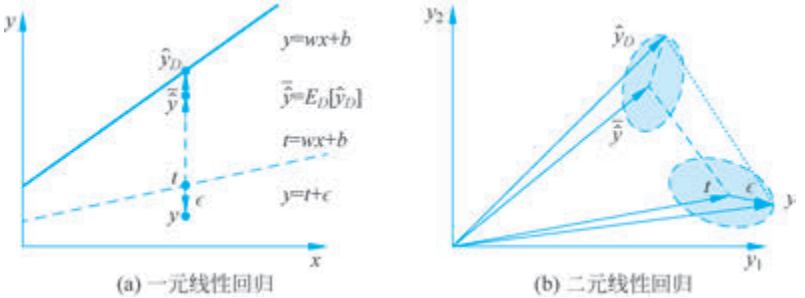
对于在训练集上某次采样得到的子集 D，训练得到的映射函数为 $\hat{y}_D=\boldsymbol{X}\hat{\boldsymbol{w}}$。多次采样得到多个子集，分别训练得到多个模型，对输入数据进行预测，多个模型的期望预测为 $\bar{\hat{y}}=E_D[\hat{y}_D]$。

对模型期望误差 $E_D[(\hat{y}_D-y)^2]$ 进行分析，可以利用如下关系：

$$\hat{y}_D-y=\hat{y}_D-\bar{\hat{y}}+\bar{\hat{y}}-t+t-y \tag{3-71}$$

式中，$\hat{y}_D-\bar{\hat{y}}$ 是均值为 0、方差为 $E_D[(\hat{y}_D-\bar{\hat{y}})^2]$ 的随机变量；由于 $y=t+\epsilon$，故 $t-y$ 是均值为 0、方差为 σ^2 的随机变量，并且 $\hat{y}_D-\bar{\hat{y}}$ 与 $t-y$ 相互独立；$\bar{\hat{y}}-t$ 为标量 $\bar{\hat{y}}$ 和标量 t 之差。

图 3-19 分别给出了一元线性回归和二元线性回归时，式(3-71)所示的误差分解示意图。点线部分为 $\hat{y}_D-\bar{\hat{y}}$，虚线部分为 $\bar{\hat{y}}-t$，实线部分为 $t-y$。



(a) 一元线性回归　　(b) 二元线性回归

图 3-19　线性回归模型的误差分解示意图

由于多个独立的随机变量之和的方差，等于各随机变量方差之和，并且式(3-71)中的随机变量 $\hat{y}_D-\bar{\hat{y}}$ 和 $t-y$ 的均值为 0，可对模型期望误差进行分解，得

$$\begin{aligned}E_D[(\hat{y}_D-y)^2]&=E_D[(\hat{y}_D-\bar{\hat{y}}+\bar{\hat{y}}-t+t-y)^2]\\&=E_D[(\hat{y}_D-\bar{\hat{y}})^2]+E_D[(\bar{\hat{y}}-t)^2]+E_D[(t-y)^2]\\&=E_D[(\hat{y}_D-\bar{\hat{y}})^2]+(\bar{\hat{y}}-t)^2+\sigma^2\end{aligned}\tag{3-72}$$

也就是说，模型期望误差包括三部分，即预测值相对于理论真实值的偏差的平方 $\mathrm{Bias}^2[f(x)]=(\bar{\hat{y}}-t)^2$、模型预测值的方差 $\mathrm{Var}[f(x)]=E_D[(\hat{y}_D-\bar{\hat{y}})^2]$ 和噪声方差 σ^2。

模型偏差指的是由训练集采用不同子集学习获取到的所有模型的预测平均值和真实结果之间的差异，度量了模型的期望预测与真实结果的偏离程度，刻画了模型本身的拟合能力。模型偏差大通常是采用复杂程度不足的函数形式所造成的，如真实模型是某个二次函数，而采用的模型是一次函数。模型偏差大在实际机器学习过程中表现为训练集上的经验误差较大。

模型方差指的是由训练集采用不同子集学习获取到的所有模型的预测的方差，刻画了数据扰动对模型所造成的影响。方差通常是由于模型的复杂度相对于训练样本数量过高而导致的。在实际情况下，模型方差大通常体现在测试误差相对于训练误差的增量上，即训练误差可能较小，但是测试误差却较大。

噪声项 $\sigma^2=E(\varepsilon^2)$ 是在当前任务上任何模型所能达到的期望误差的下界，刻画了学习问题本身的难度。

在机器学习算法实现过程中，控制模型偏差和方差是有冲突的，因此我们需要做偏差-方差的权衡(bias-variance trade-off)。给定一个任务，假设我们可以控制学习算法的训练程度。在训练不足时，模型的拟合能力不够，训练数据的变化不足以使模型的泛化误差有明显差异，此时偏差项占主导；随着训练程度的加深，模型的拟合能力逐渐加强；在训练充足后，模型的拟合能力足够强，此时偏差项减小，方差项占主导，若模型的拟合能力过强，则训练数据的轻微扰动即可能造成学习得到的模型预测发生明显变化，从而发生过拟合。

【例 3-5】 考虑函数 $f(x)=x^2$，在区间 $x\in[0,2]$ 上均匀采样 $N(N=5)$ 个点：$x=\{0,0.5,1,1.5,2\}$。观测值 $y=f(x)+\varepsilon$ 受到噪声 ε 的影响，其中 $\varepsilon\sim N(0,0.1)$。使用线性回归模型 $\hat{f}(x)=ax+b$ 来拟合数据。对于均匀采样的 5 个点，存在两个不同的观测数据集 D_1，D_2，采用最小二乘法求解得到的模型参数分别为 $a_1=1$，$b_1=0.5$ 以及 $a_2=2$，$b_2=0$。试计算：

(1) 模型偏差的平方 $\mathrm{Bias}^2=\dfrac{1}{N}\sum_{i=1}^{N}\{\mathbb{E}_D[\hat{f}(x_i)]-f(x_i)\}^2$

(2) 模型的方差 $\mathrm{Var}=\dfrac{1}{N}\sum_{i=1}^{N}\mathbb{E}_D[(\hat{f}(x_i)-\mathbb{E}_D[\hat{f}(x_i)])^2]$

(3) 模型的期望预测误差 $\dfrac{1}{N}\sum_{i=1}^{N}\mathbb{E}_D[(y_i-\hat{f}(x_i))^2]$

(4) 如果将模型复杂度提高(如三次函数拟合)，偏差和方差会如何变化？试给出定性分析。

解：

（1）对于两个观测数据集上的线性回归模型，可以分别列出如下预测结果：

x	0	0.5	1	1.5	2
$f(x)$	0	0.25	1	2.25	4
$\hat{f}_1(x)$	0.5	1	1.5	2	2.5
$\hat{f}_2(x)$	0	1	2	3	4
$E_D[\hat{f}(x)]$	0.25	1	1.75	2.5	3.25
$\hat{f}_1(x)-E_D[\hat{f}(x)]$	0.25	0	−0.25	−0.5	−0.75
$\hat{f}_2(x)-E_D[\hat{f}(x)]$	−0.25	0	0.25	0.5	0.75

因此有

$$\begin{aligned}\text{Bias}^2 &= \frac{1}{N}\sum_{i=1}^{N}\{\mathbb{E}_D[\hat{f}(x_i)]-f(x_i)\}^2\\ &= \frac{1}{5}[(0.25-0)^2+(1-0.25)^2+(1.75-1)^2+(2.5-2.25)^2+(3.25-4)^2]\\ &= 0.3625\end{aligned}$$

（2）根据上述表格中的结果，可以计算

$$\begin{aligned}\text{Var} &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_D[(\hat{f}(x_i)-\mathbb{E}_D[\hat{f}(x_i)])^2]\\ &= \frac{1}{5}\left[\frac{0.25^2+(-0.25)^2}{2}+\frac{0^2+0^2}{2}+\frac{(-0.25)^2+0.25^2}{2}+\frac{(-0.5)^2+0.5^2}{2}+\frac{(-0.75)^2+0.75^2}{2}\right]\\ &= 0.1875\end{aligned}$$

（3）期望预测误差分解为偏差平方、方差与噪声方差之和，即

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_D[(y_i-\hat{f}(x_i))^2]=\text{Var}+\text{Bias}^2+\sigma^2=0.3625+0.1875+0.1=0.65$$

（4）若采用三次多项式拟合，由于真实函数为二次，三次模型表达能力更强，因此偏差会减小；然而，模型复杂度增加，对噪声更敏感，因此方差会增大。

3.4.2　分类任务评价指标

在现实任务中，我们往往有多种机器学习算法可以选择，对于同一种机器学习模型，使用不同的参数配置时也会有不同的结果。那么我们应该如何选择最优的模型呢，这就需要使用一定的评价指标对各个模型进行定量的评估。

对于分类任务，常见的评价指标有准确率（accuracy）、精确率（precision）、召回率（recall）、F1 分数（F1 score）、P-R 曲线和 AP 值、ROC 曲线和 AUC 值。

在两类分类问题中，模型在测试集上的结果可以分为以下四种情况：

（1）真正例（true positive，TP）：一个样本的真实类别为正，并且模型正确地预测为类别正；

（2）假负例（false negative，FN）：一个样本的真实类别为正，但模型错误地预测为类别负；

（3）假正例（false positive，FP）：一个样本的真实类别为负，但模型错误地预测为类别正；

（4）真负例（true negative，TN）：一个样本的真实类别为负，并且模型也正确地预测为类别负。

分类混淆矩阵如表 3-3 所示。

表 3-3　两类分类问题的分类混淆矩阵

真 实 类 别	预测为正类	预测为负类
正类	真正例（true positive，TP）	假负例（false negative，FN）
负类	假正例（false positive，FP）	真负例（true negative，TN）

通过统计以上四种情况的样本数量，我们可以计算得到如下的评价指标，从而对模型的分类效果进行定量评价。

1. 准确率（accuracy）：正确预测的样本数占总样本数的比例

$$\text{accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{3-73}$$

2. 精确率（precision）：被预测为正类的样本中实际为正类的比例

$$\text{precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{3-74}$$

3. 召回率（recall）：实际为正类的样本中被预测为正类的比例

$$\text{recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3-75}$$

4. F1 分数（F1 score）：精确率和召回率的调和平均值，综合考虑精确率和召回率

$$\text{F1 score}=\frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}} \tag{3-76}$$

F1 score 的取值范围通常为 0～1，1 表示完美的分类器，即模型能够完全正确地预测所有样本；0 表示模型在正类识别上表现很差，精确率或召回率为 0。

除了以上四个数值指标之外，通常还采用 P-R（precision-recall）曲线、ROC（receiver operator characteristic）曲线来评价模型性能。

5. P-R（precision-recall）曲线

P-R 曲线是一种用于评估二分类模型性能的方法，通过精确率和召回率来衡量模型的预测性能。P-R 曲线以召回率为横轴，精确率为纵轴，通过调整模型的分类阈值，得到不同的 precision-recall 值的组合对，绘图得到 P-R 曲线，如图 3-20 所示。曲线越靠近右上角，模型性能越好。如果一个模型的 P-R 曲线完全包住另一个模型的曲线，则前者性能更好，例如，图 3-20 中，模型 A 的性能比模型 C 好，模型 B 的性能也比模型 C 更好。AP（average precision）代表 P-R 曲线下的面积，AP 越大，说明模型的性能越好。

6. ROC（receiver operating characteristic）曲线

ROC 曲线也是一种用于评估二分类模型性能的方法，通过真正例率（true positive rate，TPR）和假正例率（false positive rate，FPR）来展示模型在不同阈值下的性能。

真正例率（TPR）：实际为正的样本中，被模型正确预测为正的比例。计算公式如下，与召回率相同：

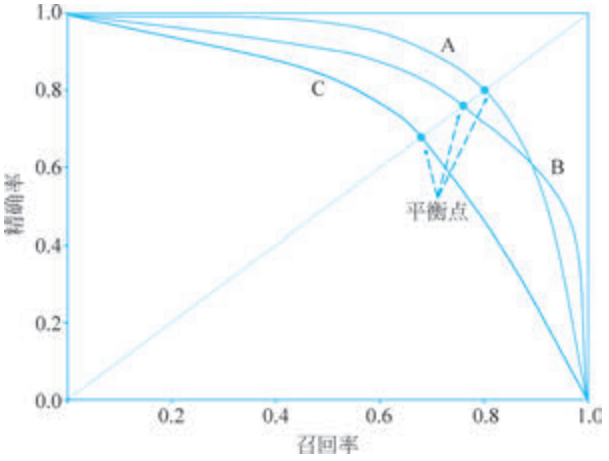


图 3-20　P-R 曲线示意图

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-77}$$

假正例率(FPR)：实际为负的样本中，被模型错误预测为正的比例。计算公式为

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \tag{3-78}$$

ROC 曲线以假正例率(FPR)为横轴，真正例率(TPR)为纵轴，通过调整模型的分类阈值，得到不同的 TPR-FPR 值的组合对，绘图得到 ROC 曲线，如图 3-21 所示。曲线越靠近左上角，模型性能越好，理想情况下，TPR 接近 1，FPR 接近 0。ROC 曲线下面积(area under the ROC curve，AUC)是一个常用的量化指标，值越接近 1，模型性能越好。AUC＝0.5 时表示模型采用随机分类策略。

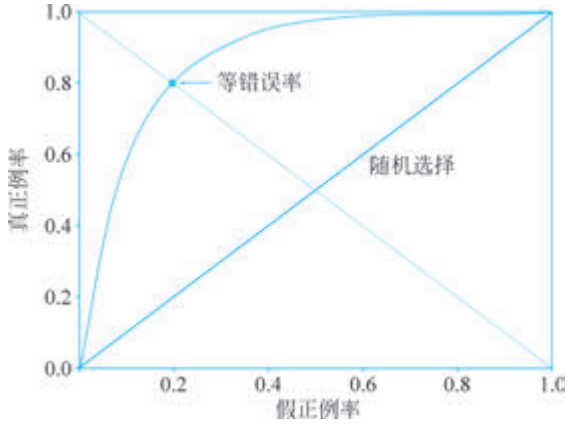


图 3-21　ROC 曲线示意图

【例 3-6】　根据某化验结果判断受试者是否患有肺炎，100 位真正肺炎患者被预测为肺炎的数目为 99，预测为正常的数目为 1；200 位正常受试者被预测为肺炎的数目为 5，预测为正常的数目为 195。根据混淆矩阵，计算分类表现的准确率、精确率、召回率和 F1 分数。

解：混淆矩阵为

真 正 类 别	预测为肺炎	预测为正常
肺炎	99	1
正常	5	195

则有 TP=99，FN=1，FP=5，TN=195。

根据公式可以分别计算出：

$$\text{accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}=\frac{294}{300}=0.9800$$

$$\text{precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}=\frac{99}{104}=0.9519$$

$$\text{recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}=\frac{99}{100}=0.9900$$

$$\text{F1 score}=2\times\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}=0.9706$$

3.4.3 回归任务评价指标

对于回归问题，常见的评价指标有均方误差(MSE)、均方根误差(RMSE)、平均绝对误差(MAE)和决定系数(R^2)。

假定测试集共有 n 个样本，每个样本的实际观测值为 $y_i(i=1,\cdots,n)$，通过回归模型预测的值为 $\hat{y}_i(i=1,\cdots,n)$，则回归问题中各评价指标计算方式如下：

1. 均方误差(MSE)

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{3-79}$$

MSE 代表预测值与真实值差的平方和的平均值，数值越小越好。

2. 均方根误差(RMSE)

$$\text{RMSE}=\sqrt{\text{MSE}}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{3-80}$$

RMSE 为 MSE 的平方根。

3. 平均绝对误差(MAE)

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{3-81}$$

MAE 代表预测值与真实值差的绝对值的平均值，数值越小越好。

4. 决定系数(R^2)

$$R^2=1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2} \tag{3-82}$$

决定系数(coefficient of determination)又称为拟合优度(goodness of fit)，常用于衡量模型对观测数据的拟合程度。R^2 的取值范围通常在 0 到 1 之间，越接近 1 表示模型拟合越好。在简单线性回归中，R^2 等于因变量与自变量之间相关系数 R 的平方。

3.4.4 模型的评估方法

机器学习模型的最终目标是能够对未知数据进行准确预测，因此我们需要对模型的泛化误差进行评估并做出选择。训练集是模型学习的“教材”，模型通过学习训练集中的数据特征和规律来更新自身。然而，过拟合现象的存在表明模型在训练集上的表现并不能完全正确反映其在真实世界数据中的表现。所以，我们需要测试集来模拟真实世界数据，以测试集上的测试误差作为泛化误差的近似，帮助我们评估模型在面对未见过的数据时的表现。

1. 留出法(hold-out method)

将数据集随机划分为互斥的训练集和测试集，模型在训练集上训练，在测试集上评估性能。分割的比例可以根据具体问题和数据量进行调整，常见的训练集、测试集样本数量比例是 7∶3 或 4∶1 左右。

此类评估方法的优点在于简单直观，计算效率高。但是测试结果依赖于数据划分方式，不同的划分方式可能导致结果不稳定；且由于划分了部分样本归为测试集，无法充分利用所有数据进行训练，特别是在数据量有限的情况下。

2. 交叉验证(cross-validation)

常见类型包括 K 折交叉验证(K-fold cross-validation)：将数据集划分为 k 个子集，每次用 $k-1$ 个子集作为训练集，剩下的 1 个子集作为测试集，重复 k 次后取平均性能。

更为特殊的方法为留一法(leave-one-out cross-validation)，每次只留一个样本作为测试集，比较适合小数据集，并不常用。

此类评估方法的优点在于充分利用数据，结果更稳定可靠，可以充分利用所有数据进行训练，特别是在数据量有限的情况下。但是计算成本相对较高，操作更加复杂。

习题

3.1 考虑一个线性模型 $y(\boldsymbol{x},w)=w_0+\sum_{i=1}^{D}w_i x_i$，并且具有一个最小二乘误差函数 $L_D[w]=\frac{1}{2}\sum_{n=1}^{N}[y(\boldsymbol{x}_n,w)-t_n]^2$，其中 t_n 为真值。现在假设对每个输入变量 x_i 添加了零均值、方差为 σ^2 的高斯噪声 ϵ_i，噪声独立添加到每个输入变量 x_i 上。利用 $E[\epsilon_i]=0$ 和 $E[\epsilon_i\,\epsilon_j]=\delta_{ij}\sigma^2$，证明在输入变量添加上述噪声 ϵ 的情况下，最小化 L_D 对噪声分布的期望 $\mathbb{E}_{\epsilon}[L_D(w)]$，等价于在输入变量不添加噪声的情况下，最小化 $L_D(w)$ 与正则化项之和，其中正则化项为 $\frac{N\sigma^2}{2}\sum_{i=1}^{D}w_i^2$。

3.2 考虑一个线性回归任务，目标是拟合模型 $y=w_1x+w_0$。有以下数据集：

x	y
1	2
2	2
3	6
4	8

采用均方误差损失函数，即 $L(w_0,w_1)=\frac{1}{n}\sum_{i=1}^{n}[y_i-(w_1x_i+w_0)]^2$。

(1) 使用最小二乘法求解 w_0 和 w_1。

(2) 对权值 w_1 引入 L2 正则化，选取参数 $\lambda=0.125$，即损失函数变为

$$L(w_0,w_1)=\frac{1}{n}\sum_{i=1}^{n}[y_i-(w_1x_i+w_0)]^2+\lambda w_1^2$$

求解此时的参数 w_0 和 w_1。

(3) 当 λ 继续增大时，会对 w_1 产生什么影响？

3.3 假设真实目标函数 $f(x)=x^2$，在区间 $x\in[0,2]$ 上均匀采样 5 个点：$x=\{0,0.5,1,1.5,2\}$。观测值 $y=f(x)+\varepsilon$ 受到噪声 ε 的影响，其中 $\varepsilon\sim N(0,0.1)$。使用线性回归模型 $\hat{f}(x)=ax+b$ 来拟合数据。采用多个数据集，通过最小二乘法拟合后，得到模型参数的期望 $E[a]=1$ 和 $E[b]=0.5$，并已知参数 a 和 b 的方差和协方差为

$$\mathrm{Var}(a)=0.05,\quad \mathrm{Var}(b)=0.1,\quad \mathrm{Cov}(a,b)=-0.02$$

试计算：

(1) 在 $x=\{0,0.5,1,1.5,2\}$ 处，模型的偏差 $\mathrm{Bias}[\hat{f}(x)]=E_D[\hat{f}(x)]-f(x)$；

(2) 在 $x=\{0,0.5,1,1.5,2\}$ 处，模型的预测方差 $\mathrm{Var}[\hat{f}(x)]=\mathrm{Var}(a)\cdot x^2+\mathrm{Var}(b)+2x\cdot\mathrm{Cov}(a,b)$；

(3) 在 $x=\{0,0.5,1,1.5,2\}$ 处，模型的期望预测误差 $E_D[(y-\hat{f}(x))^2]$；

(4) 如果将模型复杂度提高(如用二次函数拟合)，偏差和方差会如何变化？试给出定性分析。

3.4 已知正样本点 $x_1=(1,2)^{\mathrm{T}}$，$x_2=(2,3)^{\mathrm{T}}$，$x_3=(3,3)^{\mathrm{T}}$，负样本点 $x_4=(2,1)^{\mathrm{T}}$，$x_5=(3,2)^{\mathrm{T}}$，试求最大间隔分离超平面和分类决策函数，并绘图表示分离超平面、间隔边界及支持向量。

3.5 给定二维数据集包含以下三个样本点：$\boldsymbol{x}_1=(1,2)$，$\boldsymbol{x}_2=(0,1)$，$\boldsymbol{x}_3=(-1,0)$。考虑两种核函数，多项式核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j+1)^2$ 和高斯核函数(RBF 核)$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$，其中取 $\sigma=1$。

(1) 计算数据集在这两种核函数下得到的核矩阵。

(2) 解释为什么高斯核矩阵的对角线元素总是等于 1。

(3) 对于多项式核，验证 $K(x_1,x_2)=K(x_2,x_1)$，说明这体现了核函数的什么性质。