第 3 章 线性模型

3.1 线性回归

我们在第一章中已经介绍了多项式拟合 sin 函数的例子，它实际上是线性回归模型的一个特例。尽管线性模型方法早在计算机诞生之前就已经相对发展成熟，但在当今计算机技术高度发达的时代，它依然被广泛采用，并值得我们深入学习与研究。线性模型结构简单，具有良好的分析性质和清晰的几何解释，让我们能够直观地理解输入是如何影响输出的，同时也为理解更复杂的高级模型（如：支持向量机和人工神经网络）奠定基础。在实际的应用中，线性模型有时甚至能优于更复杂的非线性模型，特别是在训练数据有限或信噪比较低的情况下。

最简单的线性回归模型是输入变量的线性组合形式：

$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0 + x_1 w_1 + \cdots + x_D w_D = w_0 + \sum_{j=1}^{D} x_j w_j \tag{3.1}$$

其中输入 $\boldsymbol{x} = (x_1, \ldots, x_D)^\top$，$x_j$ 是 $\boldsymbol{x}$ 的第 j 个特征，$\boldsymbol{w} = (w_0, \ldots, w_D)^\top$ 是模型的可调参数，而可调参数中的 w_0 称为偏置或截距。

对于输入 $\boldsymbol{x}$，我们可以施加各种非线性变换，以增强模型的建模能力。因此，我们可以将上述最简单的线性回归模型扩展为对输入 $\boldsymbol{x}$ 经过变换后的形式：

$$f(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{j=1}^{M-1} \phi_j(\boldsymbol{x}) w_j \tag{3.2}$$

其中 $\phi_j(\boldsymbol{x})$ 是施加于输入 $\boldsymbol{x}$ 的某种变换，通常被称为**基函数**（Basis Function）。在上述公式中，下标 j 最大值为 $M-1$，因此模型的参数数量为 M。由于输入可以经过非线性变换，因此严格来说，线性模型并不一定是原始输入 $\boldsymbol{x}$ 的线性形式，但必定为可调参数 $\boldsymbol{w}$ 的线性形式。为了简化表示，我们可以定义一个额外的基函数 $\phi_0(\boldsymbol{x}) = 1$，从而将公式3.2简化为：

$$f(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} \phi_j(\boldsymbol{x}) w_j = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} \tag{3.3}$$

其中 $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^\top$，而 $\boldsymbol{w} = (w_0, \ldots, w_{M-1})^\top$。对输入 $\boldsymbol{x}$ 的变换可以包括：

- 恒等变换 $\phi(\boldsymbol{x}) = \boldsymbol{x}$，即将输入映射到自身。
- 幂函数 $\phi(\boldsymbol{x}) = x_j^i$，即对输入 $\boldsymbol{x}$ 的第 j 个特征取 i 次幂，其形式包括 $\sqrt{x_j}$、x_j^2、x_j^3 等。
- 特征之间的交互 $\phi(\boldsymbol{x}) = x_i x_j$，即输入 $\boldsymbol{x}$ 的第 i 维特征和第 j 维特征的乘积。
- 对数函数 $\phi(\boldsymbol{x}) = \log_b(x_j)$，即对输入 $\boldsymbol{x}$ 的第 j 个特征取以 b 为底的对数。
- 高斯基函数 $\phi(\boldsymbol{x}) = \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma^2}\right)$，该函数的值反映了第 j 个特征与其定义域中位置 μ_j 之间的距离。离 μ_j 越近，函数值越大；而离 μ_j 越远，则值越小。该距离的尺度由参数 σ 控制。需要注意的是，该函数并不一定具有概率意义，因为它没有包含高斯分布的归一化系数。函数值会与相应的可调参数 w_j 相乘，从而将归一化系数的影响整合到可调参数中。

- 傅里叶基函数（Fourier Basis），它将函数表示为不同频率正弦波的叠加。每个基函数对应一个特定的频率，特别适用于信号的分解和分析。

接下来，我们讨论如何根据给定包含 N 个样本的训练数据集 $\mathcal{D}=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^N$ 来确定可调参数 $\boldsymbol{w}$ 的值。回顾第一章的内容（见公式1.18），我们已知在假设高斯噪声的情况下，线性模型的最大化似然函数等价于最小化以下平方误差损失函数：

$$\mathrm{RSS}(\boldsymbol{w}) \triangleq \frac{1}{2}\sum_{i=1}^{N}\left(y_i-\boldsymbol{\phi}(\boldsymbol{x}_i)^\top\boldsymbol{w}\right)^2 \tag{3.4}$$

其中 RSS（Residual Sum of Squares）是残差平方和，即所有样本的预测值与实际值之间误差的平方和。与第一章中所定义的残差平方和略有不同，在公式3.4中，我们额外引入了一个 $\frac{1}{2}$ 作为系数。这是一种常用的处理技巧，旨在简化后续推导，使公式更加整洁。具体而言，该 $\frac{1}{2}$ 将与对二次项求导后出现的 2 相互抵消，从而在不改变最优解的情况下，避免出现不必要的系数。

为了确定参数的取值，我们将公式3.4对 $\boldsymbol{w}$ 求偏导，得到如下表达式：

$$\nabla\mathrm{RSS}(\boldsymbol{w}) = -\sum_{i=1}^{N}\left(y_i-\boldsymbol{\phi}(\boldsymbol{x}_i)^\top\boldsymbol{w}\right)\boldsymbol{\phi}(\boldsymbol{x}_i) \tag{3.5}$$

将该导数设为零，并注意到 $\boldsymbol{\phi}(\boldsymbol{x}_i)^\top\boldsymbol{w}$ 是一个标量，因此可以与 $\boldsymbol{\phi}(\boldsymbol{x}_i)$ 交换乘法顺序，我们得到：

$$-\sum_{i=1}^{N}y_i\boldsymbol{\phi}(\boldsymbol{x}_i)+\left(\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^\top\right)\boldsymbol{w}=0 \tag{3.6}$$

将含有参数 $\boldsymbol{w}$ 的项与不含 $\boldsymbol{w}$ 的项分别移至等式的两边，可得：

$$\left(\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_i)\boldsymbol{\phi}(\boldsymbol{x}_i)^\top\right)\boldsymbol{w}=\sum_{i=1}^{N}y_i\boldsymbol{\phi}(\boldsymbol{x}_i) \quad \Rightarrow$$

$$\sum_{i=1}^{N}\begin{bmatrix}\phi_0(\boldsymbol{x}_i)\phi_0(\boldsymbol{x}_i) & \phi_0(\boldsymbol{x}_i)\phi_1(\boldsymbol{x}_i) & \cdots & \phi_0(\boldsymbol{x}_i)\phi_{M-1}(\boldsymbol{x}_i)\\ \phi_1(\boldsymbol{x}_i)\phi_0(\boldsymbol{x}_i) & \phi_1(\boldsymbol{x}_i)\phi_1(\boldsymbol{x}_i) & \cdots & \phi_1(\boldsymbol{x}_i)\phi_{M-1}(\boldsymbol{x}_i)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_{M-1}(\boldsymbol{x}_i)\phi_0(\boldsymbol{x}_i) & \phi_{M-1}(\boldsymbol{x}_i)\phi_1(\boldsymbol{x}_i) & \cdots & \phi_{M-1}(\boldsymbol{x}_i)\phi_{M-1}(\boldsymbol{x}_i)\end{bmatrix}\boldsymbol{w}=\sum_{i=1}^{N}\begin{bmatrix}y_i\phi_0(\boldsymbol{x}_i)\\ y_i\phi_1(\boldsymbol{x}_i)\\ \vdots\\ y_i\phi_{M-1}(\boldsymbol{x}_i)\end{bmatrix} \quad \Rightarrow$$

$$\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{w}=\boldsymbol{\Phi}^\top\mathbf{Y} \quad \Rightarrow$$

$$\boldsymbol{w}=(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{Y}$$

其中 $\mathbf{Y}=(y_1,y_2,\ldots,y_N)^\top$ 表示所有训练样本的实际观测值构成的列向量，而矩阵 $\boldsymbol{\Phi}$ 定义如下：

$$\boldsymbol{\Phi}=\begin{bmatrix}\phi_0(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_1) & \cdots & \phi_{M-1}(\boldsymbol{x}_1)\\ \phi_0(\boldsymbol{x}_2) & \phi_1(\boldsymbol{x}_2) & \cdots & \phi_{M-1}(\boldsymbol{x}_2)\\ \vdots & \vdots & \ddots & \vdots\\ \phi_0(\boldsymbol{x}_N) & \phi_1(\boldsymbol{x}_N) & \cdots & \phi_{M-1}(\boldsymbol{x}_N)\end{bmatrix} \tag{3.7}$$

以下矩阵

$$\boldsymbol{\Phi}^\dagger \triangleq (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top \tag{3.8}$$

当设计矩阵（Design Matrix）$\boldsymbol{\Phi}$ 满列秩时，$\boldsymbol{\Phi}^{\dagger}$ 被称为 $\boldsymbol{\Phi}$ 的 Moore-Penrose 伪逆，它可以看作是矩阵求逆在非方阵情形下的一种推广。实际上，当 $\boldsymbol{\Phi}$ 是可逆的方阵时，由等式 $(\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi})^{-1} = \boldsymbol{\Phi}^{-1}\left(\boldsymbol{\Phi}^{\top}\right)^{-1}$ 和公式3.8可得 $\boldsymbol{\Phi}^{\dagger} \triangleq \boldsymbol{\Phi}^{-1}$。

现在，我们可以探讨并理解偏置项 w_0 的作用。如果将偏置 w_0 从其余参数中单独分离出来，则损失函数3.4可改写为：

$$\mathrm{RSS}(\boldsymbol{w}) \triangleq \frac{1}{2}\sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{M-1}\phi_j(\boldsymbol{x}_i)w_j\right)^2 \tag{3.9}$$

将上式单独对偏置 w_0 求导，并将导数设为零，我们可以得到：

$$w_0 = \overline{y} - \sum_{j=1}^{M-1}\overline{\phi_j}w_j \tag{3.10}$$

其中 $\overline{y}$ 和 $\overline{\phi_j}$ 分别为：

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N}y_i \tag{3.11}$$

$$\overline{\phi_j} = \frac{1}{N}\sum_{i=1}^{N}\phi_j(\boldsymbol{x}_i) \tag{3.12}$$

从公式3.10可以看出，偏置 w_0 等于拟合目标的平均值与基函数平均值加权和之间的差异。如果对所有目标值 y_i 加上或减去相同的常数，则偏置项 w_0 会相应地增加或减少。因此，第一章中提到的正则化方法（见公式1.13）通常只会作用于除偏置项 w_0 之外的其他可调参数。

通过最小化所有样本的预测值与实际观测值之间差异的平方和（RSS）来确定模型参数的方法，通常称为最小二乘法。接下来，我们探讨采用最小二乘法所得到线性回归参数解的直观解释及其几何意义。从上述推导出 $\boldsymbol{w} = (\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\top}\mathbf{Y}$ 的中间步骤里，我们已知 $\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}\boldsymbol{w} = \boldsymbol{\Phi}^{\top}\mathbf{Y}$，而该式还可以改写成：

$$\boldsymbol{\Phi}^{\top}(\mathbf{Y} - \boldsymbol{\Phi}\boldsymbol{w}) = 0 \tag{3.13}$$

定义 $\mathbf{F} = (f(\boldsymbol{x}_1, \boldsymbol{w}), f(\boldsymbol{x}_2, \boldsymbol{w}), \ldots, f(\boldsymbol{x}_N, \boldsymbol{w}))^{\top}$，即将所有样本的拟合结果组合成为一个列向量，则有：

$$\mathbf{F} = \boldsymbol{\Phi}\boldsymbol{w} = \left[\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{M-1}\right]\boldsymbol{w} = \boldsymbol{\varphi}_0 w_0 + \boldsymbol{\varphi}_1 w_1 + \cdots + \boldsymbol{\varphi}_{M-1}w_{M-1} \tag{3.14}$$

其中 $\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{M-1}$ 依次是矩阵 $\boldsymbol{\Phi}$（详见公式3.7）中的列向量。每一个列向量表示所有样本在对应基函数下的值，其中 $\boldsymbol{\varphi}_0$ 的元素全为 1（对应于偏置参数 w_0）。从上式可以看出，向量 $\mathbf{F}$ 是向量 $\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{M-1}$ 的线性组合，即 $\mathbf{F}$ 位于这些向量所张成的子空间中。根据公式3.14中对于 $\mathbf{F}$ 的定义，我们可以将公式3.13简化为：

$$\boldsymbol{\Phi}^{\top}(\mathbf{Y} - \mathbf{F}) = 0 \tag{3.15}$$

上式表明，所有样本的实际值 $\mathbf{Y}$ 与预测值 $\mathbf{F}$ 之间的误差向量 $(\mathbf{Y} - \mathbf{F})$ 与向量 $\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_{M-1}$ 所张成的子空间正交。为便于说明与简洁展示，图3.1仅显示了由两个基函数对应的向量 $\boldsymbol{\varphi}_1$ 和

φ_2 所张成的子空间 $\mathcal{S}$。从图中可以看出，向量 $\mathbf{F}$ 是 $\mathbf{Y}$ 在子空间 $\mathcal{S}$（图中为一平面）上的投影，而误差向量（用虚线表示）则与该子空间正交。多个样本点分别绘制的拟合示意图如图3.2所示，图中的平面是通过最小化每个样本实际观测值（图中红色圆圈）到该平面的垂直距离来确定。

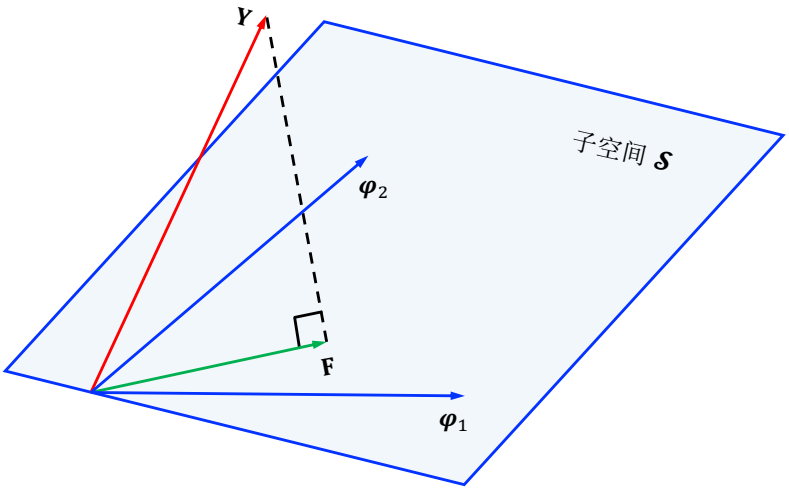


图 3.1: 最小二乘法下线性回归的几何解释。样本实际观测值 $\mathbf{Y}$ 垂直投影到由两个基函数对应的向量 φ_1 和 φ_2 所张成的平面 $\mathcal{S}$ 上，所得投影 $\mathbf{F}$ 即为线性回归模型的预测值。

如果将以下可调参数 $\boldsymbol{w}$ 的求解公式（Normal Equations）：

$$\boldsymbol{w} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{Y} \tag{3.16}$$

代入到公式3.14后，我们可以得到：

$$\mathbf{F} = \boldsymbol{\Phi}\boldsymbol{w} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{Y} = \mathbf{H}\mathbf{Y} \tag{3.17}$$

其中矩阵 $\mathbf{H} = \boldsymbol{\Phi}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top$ 用于计算 $\mathbf{Y}$ 在线性子空间上的正交投影，因而被称为投影矩阵。

对于包含大量样本的数据集，直接使用公式3.16来求解线性模型参数 $\boldsymbol{w}$ 的计算成本较高。此外，在某些应用中，观测数据以数据流的形式持续到达，此时往往需要在尚未获得全部数据之前就能够进行预测。针对上述两种情况，通常采用在线学习（Online Learning）或序贯学习（Sequential Learning）算法来逐步更新参数 $\boldsymbol{w}$。这类算法的一般思路是：先设定参数的初始值 $\boldsymbol{w}^{(0)}$，然后根据一个样本或一小批样本对模型参数进行持续更新。随机梯度下降法（Stochastic Gradient Descent）是一种常用的序贯学习算法，它通过以下更新公式对模型参数进行迭代调整：

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \eta \nabla \mathcal{L}(\boldsymbol{w}^{(t)}) \tag{3.18}$$

其中 t 表示迭代的时间步，η 是学习率（或称学习步长），而 $\mathcal{L}$ 表示损失函数。若每次使用多个样本来更新参数，其损失函数为多个样本的损失函数之和。由于损失函数对参数 $\boldsymbol{w}$ 的梯度指向损失函数增长最快的方向，而我们的目标是最小化损失函数，因此参数应沿着梯度的相反方向进行更新。为此，更新公式中学习率 η 前通常带有一个负号（学习率本身一般取小于 1 的正值）。对于公式3.4所定义的损失函数，其参数更新公式为：

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \eta \left(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{w}^{(t)} \right) \boldsymbol{\phi}(\boldsymbol{x}_i) \tag{3.19}$$

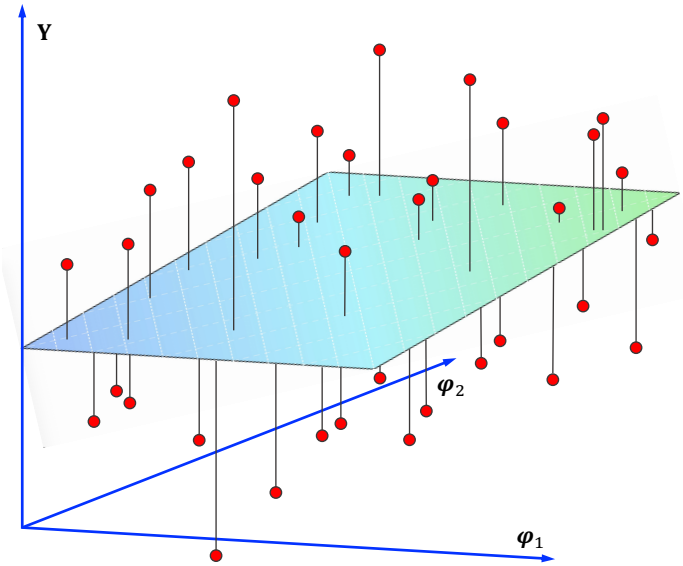


图 3.2: 最小二乘法线性回归示意图。最优线性回归函数（由图中的平面表示）通过最小化每个样本实际观测值（红色圆圈）与其在平面上的垂直投影之间的距离来确定。

此方法被称为最小均方算法（Least-mean-squares），简称 LMS 算法。

在前面的讨论中，我们关注的是每个样本对应一个拟合目标的情形。现在，我们将其扩展到每个样本对应多个输出值的情况。最简单的方法是为每一个输出目标分别选择不同的基函数，并为每个目标构建一个独立的线性回归模型，从而形成多个彼此独立的回归问题。另一种更为常见的做法是对所有输出目标使用一组相同的基函数，通过这组共享的基函数同时拟合多个目标值。假设有 $K > 1$ 个拟合目标，则我们有：

$$\boldsymbol{f}(\boldsymbol{x}, \mathbf{W}) = \boldsymbol{\phi}(\boldsymbol{x})^\top \mathbf{W} \tag{3.20}$$

其中 $\boldsymbol{f}$ 是一个 K 维的行向量，表示模型的 K 个输出值。而 $\mathbf{W}$ 则是 $M \times K$ 的参数矩阵，用于将基函数的输出映射到 K 个拟合目标上。仍然使用上述的最小二乘法，则可以通过以下公式来求解参数矩阵：

$$\mathbf{W} = (\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^\top \mathbf{T} = \boldsymbol{\Phi}^\dagger \mathbf{T} \tag{3.21}$$

其中 $\mathbf{T}$ 是一个 $N \times K$ 的矩阵，该矩阵的每一行对应一个样本的 K 个实际观测值。从公式3.21可以看出，尽管不同的拟合目标共享同一个 Moore-Penrose 伪逆 $\boldsymbol{\Phi}^\dagger$（只需计算一次），回归问题的解在不同拟合目标之间是解耦的，即使用相同的伪逆 $\boldsymbol{\Phi}^\dagger$ 独立地为每个拟合目标求解出对应的一组可调参数。

3.2 特征选择

基函数 $\phi(\boldsymbol{x})$ 可以理解为从输入 $\boldsymbol{x}$ 中提取某种特征的方式。如第3.1节所述，基函数的形式多样，常见的包括幂函数、对数函数和高斯基函数等。然而，并非所有通过基函数变换得到的

特征都是有用的。如果所选特征过少或不当，模型可能无法捕捉数据的内在规律，从而导致欠拟合；而如果所选特征过多，则容易造成模型对训练数据的过拟合，降低泛化能力。实际应用中通常采用“先多后筛”的策略，即初步构造较为丰富的特征集合，再通过特征选择或正则化等方法筛除冗余或无效特征，从而提升模型的拟合效果与泛化性能。本节将主要介绍线性回归模型的特征选择方法，而下一节则将探讨相应的正则化方法。

对于线性回归函数 $f(\boldsymbol{x}, \boldsymbol{w}) = w_0\phi_0(\boldsymbol{x}) + w_1\phi_1(\boldsymbol{x}) + \cdots + w_{M-1}\phi_{M-1}(\boldsymbol{x})$，若某个参数 w_j 的绝对值较小，甚至接近于零，通常认为相应的特征 $\phi_j(\boldsymbol{x})$ 对拟合结果的贡献较小，即该特征的重要性相对较低。然而，仅依据参数的绝对值来判断特征的重要性并不可靠，因为各个参数估计值的方差可能存在差异，直接比较参数的绝对值会导致误判。因此，在评估特征重要性时，还应考虑参数估计的标准差，从而判断其统计显著性。

在判断某个特征是否重要时，我们通常会假设其对应的参数服从以零为均值、σ^2 为方差的高斯分布。基于这一零假设分布（Null Distribution），我们可以计算参数的 z 值（z-score），并据此对特征的重要性进行排序和统计推断。z 值的计算方法也比较简单，即将参数估计值减去其假设均值（此处为零），再除以该参数估计值的标准差 σ（假设标准差 σ 已知）。

根据线性回归的参数估计公式 $\hat{\boldsymbol{w}} = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{Y}$（为了强调参数 $\boldsymbol{w}$ 的估计值带有随机性，此处使用 $\hat{\boldsymbol{w}}$ 来表示估计结果），我们可以推导出参数估计的协方差（具体推导过程详见后文）：

$$\mathrm{Cov}[\hat{\boldsymbol{w}}] = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\sigma_\epsilon^2 \tag{3.22}$$

其中 σ_ϵ^2 表示误差项 ϵ 的方差，而误差项定义如下：

$$y = \boldsymbol{\phi}(\boldsymbol{x})^\top\boldsymbol{w} + \epsilon \tag{3.23}$$

即 ϵ 表示真实观测值 y 与线性回归模型预测值之间的误差，并且假设其服从高斯分布 $\mathcal{N}(0, \sigma_\epsilon^2)$。根据公式3.23，所有样本的实际观测值 $\mathbf{Y}$ 可以表示为 $\boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\epsilon}$，其中列向量 $\boldsymbol{\epsilon}$ 的第 j 个分量 ϵ_j 对应第 j 个样本的误差。将 $\mathbf{Y} = \boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\epsilon}$ 代入参数估计公式，我们可以得到：

$$\begin{aligned}
\hat{\boldsymbol{w}} &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{w} + \boldsymbol{\epsilon}) \\
&= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}\boldsymbol{w} + (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon} \\
&= \boldsymbol{w} + (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon} && ((\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \mathbf{I})
\end{aligned}$$

对 $\hat{\boldsymbol{w}}$ 求期望，我们可得：

$$\begin{aligned}
\mathbb{E}[\hat{\boldsymbol{w}}] &= \mathbb{E}[\boldsymbol{w} + (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}] \\
&= \mathbb{E}[\boldsymbol{w}] + \mathbb{E}[(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}] && (\text{期望的线性性质}) \\
&= \boldsymbol{w} + (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbb{E}[\boldsymbol{\epsilon}] \\
&= \boldsymbol{w} && (\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0})
\end{aligned}$$

结合以上两式，我们可以推导出：

$$\begin{aligned}\hat{\boldsymbol{w}} - \mathbb{E}[\hat{\boldsymbol{w}}] &= \boldsymbol{w} + (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon} - \boldsymbol{w} \\ &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\end{aligned} \tag{3.24}$$

至此，我们可以推导公式3.22如下：

$$\begin{aligned}&\mathrm{Cov}[\hat{\boldsymbol{w}}] \\ &= \mathbb{E}\left[(\hat{\boldsymbol{w}} - \mathbb{E}[\hat{\boldsymbol{w}}])(\hat{\boldsymbol{w}} - \mathbb{E}[\hat{\boldsymbol{w}}])^\top\right] && \text{（协方差定义）} \\ &= \mathbb{E}\left[(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\left((\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\right)^\top\right] && \text{（利用公式3.24）} \\ &= \mathbb{E}\left[(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\right] && \text{（}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\text{ 为对称矩阵）} \\ &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right]\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} && \text{（}\boldsymbol{\Phi}\text{ 为常数矩阵）} \\ &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\sigma_\epsilon^2\mathbf{I}\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1} && \text{（}\mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right] = \sigma_\epsilon^2\mathbf{I}, \mathbb{E}\left[\epsilon_i^2\right] = \mathrm{Var}[\epsilon_i] = \sigma_\epsilon^2, \\ &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi}(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\sigma_\epsilon^2 && \mathbb{E}[\epsilon_i\epsilon_j] = \mathrm{Cov}[\epsilon_i, \epsilon_j] = 0\text{）} \\ &= (\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\sigma_\epsilon^2\end{aligned}$$

使用 z 值来评估参数对应特征的重要性时，要求方差 σ_ϵ^2 是已知的。但在实际的应用中，我们只能通过样本数据来估计该方差，具体估计方法如下公式所示：

$$\hat{\sigma}_\epsilon^2 = \frac{1}{N-M}\sum_{i=1}^{N}\left(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^\top\hat{\boldsymbol{w}}\right)^2 \tag{3.25}$$

上式计算误差的方差时，括号内省略了减去为零的均值项。以上公式中分母取 $N-M$ 而非 N，是为了使 $\hat{\sigma}_\epsilon^2$ 成为 σ_ϵ^2 的无偏估计，即满足 $\mathbb{E}(\hat{\sigma}_\epsilon^2) = \sigma_\epsilon^2$。其中，$M$ 表示线性回归模型中需要估计的参数个数，等于基函数的数量加上偏置项。在进行参数估计时会消耗一定的自由度。例如，用样本数据估计一个均值会占用一个自由度。类似地，在拟合线性模型时，我们需要估计 M 个参数，因此应从总样本数 N 中扣除 M 个自由度，以修正模型拟合所带来的自由度损失。

根据卡方分布的定义，若有 η 个相互独立且服从标准高斯分布 $\mathcal{N}(0,1)$ 的随机变量，则它们的平方和服从自由度为 η 的**卡方分布**，记为 $\chi^2(\eta)$。此外，从方差为 σ^2 的高斯分布中独立采样得到的 N 个样本，其样本方差 $\hat{\sigma}^2$ 满足如下关系 [20]：

$$\frac{(N-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2(N-1) \tag{3.26}$$

结合公式3.25和3.26，我们可以得到：

$$\begin{aligned}&(N-M)\hat{\sigma}_\epsilon^2 = \sum_{i=1}^{N}\left(y_i - \boldsymbol{\phi}(\boldsymbol{x}_i)^\top\hat{\boldsymbol{w}}\right)^2 = \sum_{i=1}^{N}(\epsilon_i)^2 \quad \Rightarrow \\ &\frac{(N-M)\hat{\sigma}_\epsilon^2}{\sigma_\epsilon^2} \sim \chi^2(N-M) \text{（假设每个 } \epsilon_i \text{ 都服从高斯分布 } \mathcal{N}(0,\sigma_\epsilon^2)\text{）}\end{aligned} \tag{3.27}$$

为了检验线性回归模型中某个参数 $\hat{w}_j$ 是否显著不等于零，并且已知 $\hat{\boldsymbol{w}}$ 的协方差为 $(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\sigma_\epsilon^2$

(公式3.22)，我们可以通过如下方式计算其 z 值：

$$z_j = \frac{\hat{w}_j - 0}{\sqrt{\sigma_\epsilon^2 \lambda_j}} = \frac{\hat{w}_j}{\sigma_\epsilon \sqrt{\lambda_j}} \tag{3.28}$$

其中 λ_j 是矩阵 $(\mathbf{\Phi}^\top \mathbf{\Phi})^{-1}$ 的第 j 个对角元素，这是因为参数 $\hat{w}_j$ 的方差对应于协方差矩阵中的第 j 个对角元素。然而，公式3.28假设 σ_ϵ^2 是已知的，但我们只能通过公式3.25来对其进行估计。因此，只能对 $\hat{w}_j$ 进行基于学生氏分布的 t 检验。

设有两个相互独立的随机变量 y 和 z，其中 y 服从自由度为 η 的卡方分布 $\chi^2(\eta)$，而 z 服从标准高斯分布 $\mathcal{N}(0,1)$。若定义随机变量 x 为：

$$x = \frac{z}{\sqrt{y/\eta}} \tag{3.29}$$

则该随机变量 x 服从自由度为 η 的学生氏分布（Student's t-Distribution）[20]，该分布的概率密度函数见公式2.119。

由于公式3.28中的随机变量 z_j 服从标准高斯分布 $\mathcal{N}(0,1)$，并结合公式3.27所示的卡方分布，我们可以根据学生氏分布的定义公式3.29来计算关于参数 $\hat{w}_j$ 的 t 值（t-value）：

$$t_j = \frac{z_j}{\sqrt{\frac{(N-M)\hat{\sigma}_\epsilon^2}{\sigma_\epsilon^2}/(N-M)}} = \frac{\hat{w}_j}{\sigma_\epsilon \sqrt{\lambda_j}} \cdot \frac{\sigma_\epsilon}{\hat{\sigma}_\epsilon} = \frac{\hat{w}_j}{\hat{\sigma}_\epsilon \sqrt{\lambda_j}} \tag{3.30}$$

因此，统计量 t_j 服从自由度为 $N-M$ 的学生氏分布。在假设检验中，t 值与 z 值的主要区别在于，t 值使用样本方差来估计真实方差，并且使用学生氏分布而非高斯分布来进行假设检验。根据第二章的分析，学生氏分布可以看作具有相同均值但不同精度的无穷多个高斯分布的叠加。通常认为，当自由度足够大时，学生氏分布逐渐接近标准高斯分布。

虽然参数 $\hat{w}_j$ 的 t 值绝对值可以作为衡量对应特征重要性的一项参考指标，但仅依据该绝对值的大小进行特征筛选却并不可靠。这是因为，从 M 个候选特征中选择 K 个最优特征是一个组合优化问题。例如，假设特征子集 $\{\phi_1, \phi_2, \phi_3\}$ 和 $\{\phi_4, \phi_5\}$ 都可以构成模型的最优特征组合。然而，当将所有特征一并纳入线性模型中进行拟合时，可能会发现特征 ϕ_3 和 ϕ_5 的 t 值绝对值最小。按照 t 值筛选的逻辑，这两个特征可能会被删除，但所保留的特征集合却无法还原任何一个最优特征子集，从而导致整体模型性能的下降。

一种较为可靠的做法是将样本集合划分为训练集和验证集，先在训练集上拟合模型参数，然后在验证集上评估和选择模型。由于选择最优特征子集本质上是一个组合优化问题，其搜索空间规模为 2^M（每一特征都有保留或剔除两种可能）。因此当候选特征数量 M 较大时，采用穷举搜索的方法在计算上是不可行的。为降低计算代价，通常采用贪婪搜索策略（Greedy Algorithm），如前向逐步选择（Forward Stepwise Selection）和后向逐步选择（Backward Stepwise Selection）等方法，作为对穷举搜索的近似替代。

前向逐步选择方法从仅包含偏置项的初始模型开始。在每一轮迭代中，方法将当前模型与尚未入选的各个特征逐一组合，构造出多个仅新增一个特征的候选模型。然后，依据这些候选模型在验证集上的预测性能，选择其中最优的一个特征加入模型。该过程不断重复，每次仅引入一个新特征，直到引入更多特征不再提升验证集上的预测性能为止。后向逐步选择方法则从

包含全部特征的完整模型开始。在每一步中，移除当前模型中一个对预测性能贡献最小的特征，直到进一步剔除特征将导致在验证集上性能下降为止。无论采用前向还是后向选择方法，在每一步从多个候选模型中做出选择，以及确定最终特征子集的大小 K，都需要在预测误差的偏差与方差之间权衡。此外，对于模型复杂度（即特征数量）的选择，通常还包含一定程度的主观判断。实践中，常采用奥卡姆剃刀原理，即在预测性能相近的情况下，优先选择参数更少的模型。

3.3 正则化方法

在上述特征选择方法中，每个特征只能被“保留”或“剔除”。这种“有或无”的离散取舍方式，容易导致模型在不同样本数据集上表现出较大的预测方差。其原因在于，这种非连续的选择机制可能会对样本数据的微小变动较为敏感。在特征选择的过程中，某一特征可能因为样本数据的轻微变化从被选中变为被剔除（或反之），从而导致模型结构发生较大改变。这种不稳定性会使得模型在不同数据上的预测结果差异较大，即出现所谓的“高方差”现象。为了缓解这一问题，我们希望采用一种更为平滑的特征选择策略，使得特征的重要性可以在“保留”与“剔除”之间以连续的方式进行调节，从而提升模型结构的稳定性和预测性能。

此外，无论我们采用何种特征选择方法，当某些特征（即基函数在各训练样本上的取值）之间存在线性相关或近似线性相关时，通常都会导致模型中对应参数的绝对值异常增大。在线性回归模型中，参数 $\boldsymbol{w}$ 的最小二乘解可表示为：$(\boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{Y}$，其中 $\boldsymbol{\Phi}$ 的第 j 列表示第 j 个基函数在所有训练样本上的取值。为便于直观说明和理解，这里我们假设 $\boldsymbol{\Phi}$ 是可逆的，此时最小二乘解可以简化为：$\boldsymbol{w}=\boldsymbol{\Phi}^{-1}\mathbf{Y}$。从几何的角度来看，矩阵的特征值（Eigenvalues）描述了其在对应特征向量方向上的缩放程度。若矩阵 $\boldsymbol{\Phi}$ 中两个列向量近似线性相关，则矩阵在某些方向上存在信息冗余，使得其在这些方向上的作用变得非常微弱，表现为对应的特征值 λ 非常小（甚至接近于零）。由于矩阵与其逆矩阵相乘应为单位矩阵 $\mathbf{I}$，当某个方向上 $\boldsymbol{\Phi}$ 的特征值 λ 非常小时，其逆矩阵 $\boldsymbol{\Phi}^{-1}$ 在该方向上的特征值将是 $\frac{1}{\lambda}$，数值非常大。这种情况会放大对 $\mathbf{Y}$ 中微小扰动的响应，导致最终求得的参数 $\boldsymbol{w}$ 出现剧烈波动。换言之，线性相关或近似线性相关的特征会严重影响模型求解过程的数值稳定性，进而引发“病态解”的问题。下面我们通过一个简明的例子对这一现象进行具体说明。

考虑如下 2×2 的矩阵：

$$\boldsymbol{\Phi}=\begin{bmatrix}1 & 1\\ 1 & 1.0001\end{bmatrix}$$

可以看出，以上矩阵的两个列向量近似线性相关。如果该矩阵右下角的元素从 1.0001 变为 1，则第二列将与第一列完全相同，此时矩阵不再满秩，列向量之间变得完全线性相关。对该矩阵进

行特征值分解，我们可以得到：

$$\boldsymbol{\Phi} = \underbrace{\begin{bmatrix} 0.7070891 & -0.7071245 \\ 0.7071245 & 0.7070891 \end{bmatrix}}_{\text{特征向量矩阵}} \underbrace{\begin{bmatrix} 2.00005 & 0 \\ 0 & 4.999875 \times 10^{-5} \end{bmatrix}}_{\text{特征值对角矩阵}} \underbrace{\begin{bmatrix} 0.7070891 & -0.7071245 \\ 0.7071245 & 0.7070891 \end{bmatrix}^{-1}}_{\text{特征向量矩阵的逆}}$$

以上等式右侧中间矩阵对角线上的元素就是矩阵 $\boldsymbol{\Phi}$ 的特征值：

$$\lambda_1 \approx 2.00005, \lambda_2 \approx 4.999875 \times 10^{-5}$$

虽然矩阵 $\boldsymbol{\Phi}$ 是可逆的（即不存在为零的特征值），但由于其中一个特征值 λ_2 非常小，导致其逆矩阵在对应方向上的特征值会非常大（因为逆矩阵特征值是原矩阵特征值的倒数）。通过计算，可以得到如下 $\boldsymbol{\Phi}$ 的逆矩阵：

$$\boldsymbol{\Phi}^{-1} = \begin{bmatrix} 10001 & -10000 \\ -10000 & 10000 \end{bmatrix}$$

对矩阵 $\boldsymbol{\Phi}^{-1}$ 进行特征值分解，可得：

$$\boldsymbol{\Phi}^{-1} = \begin{bmatrix} 0.7070891 & 0.7071245 \\ -0.7071245 & 0.7070891 \end{bmatrix} \begin{bmatrix} 20000.5 = \frac{1}{\lambda_2} & 0 \\ 0 & 0.4999875 = \frac{1}{\lambda_1} \end{bmatrix} \begin{bmatrix} 0.7070891 & 0.7071245 \\ -0.7071245 & 0.7070891 \end{bmatrix}^{-1}$$

其中矩阵 $\boldsymbol{\Phi}^{-1}$ 的两个特征值正好是原矩阵 $\boldsymbol{\Phi}$ 特征值的倒数。设我们要求解方程组 $\boldsymbol{\Phi}\boldsymbol{w} = \mathbf{Y}$，并且 $\mathbf{Y} = (2, 2.0001)^\top$，那么其解为：

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\mathbf{Y} = \begin{bmatrix} 10001 & -10000 \\ -10000 & 10000 \end{bmatrix} \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

若我们对 $\mathbf{Y}$ 施加微小扰动，将其第一个分量从 2 改为 2.0001，即 $\mathbf{Y} = (2.0001, 2.0001)^\top$。重新计算相应的解，可以得到：

$$\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\mathbf{Y} = \begin{bmatrix} 10001 & -10000 \\ -10000 & 10000 \end{bmatrix} \begin{bmatrix} 2.0001 \\ 2.0001 \end{bmatrix} = \begin{bmatrix} 2.0001 \\ 0 \end{bmatrix}$$

通过比较目标向量 $\mathbf{Y}$ 扰动前后参数向量 $\boldsymbol{w}$ 的取值，我们可以发现，尽管仅在目标向量 $\mathbf{Y}$ 的一个分量上施加了微小的扰动（幅度为 0.0001），参数向量的变化却极为显著。虽然第一个参数仅增长了约两倍，而而第二个参数则由 1 变为 0。这种对微小扰动极为敏感的现象，源于矩阵 $\boldsymbol{\Phi}$ 中存在近似线性相关的列向量，导致其特征值中有非常小的值。在矩阵求逆时，这些小特征值会被倒数放大，从而使得逆矩阵的相应方向的特征值变得非常大，进而放大了目标向量 $\mathbf{Y}$ 的微小扰动，最终导致线性方程组的解对输入数据的微小变化高度敏感。

为了缓解因以离散方式选取特征可能带来的预测结果方差较大，以及由于特征之间存在近似线性相关性而导致的参数估计过程中的数值不稳定性，正则化被认为是一种有效的应对方法。此外，正如第一章中使用线性模型拟合 sin 函数的例子所示，模型特征过多时容易出现过拟合现象，正则化也被认为是防止过拟合的主要策略之一。

本节我们主要讨论对参数 $\boldsymbol{w}$ 加范数（Norm）约束的正则化方法。在线性回归模型中对参数施加 ℓ_2 范数（即欧几里得范数）约束的方法被称为**岭回归**（Ridge Regression）。岭回归的优化

问题可以表示为：

$$\hat{\boldsymbol{w}}^{\text{ridge}} = \arg\min_{\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{M-1}\phi_j(\boldsymbol{x}_i)w_j\right)^2$$
$$\text{s.t.} \sum_{j=1}^{M-1} w_j^2 \leq t \tag{3.31}$$

其中“s.t.”是“受限于”的意思（英文 Subject to 的缩写），表示后续条件为优化问题的约束。在上述公式中，约束条件要求除偏置 w_0 之外的其余参数的 ℓ_2 范数必须小于等于给定阈值 t。我们还可以将岭回归问题写成如下等价的拉格朗日形式：

$$\hat{\boldsymbol{w}}^{\text{ridge}} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2}\sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{M-1}\phi_j(\boldsymbol{x}_i)w_j\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{M-1} w_j^2\right) \tag{3.32}$$

公式3.32中正则化系数 λ 的作用与约束形式公式3.31中的阈值 t 是等价的，两者可以一一对应地相互转换。正如第3.1节所述，正则化项仅作用于除偏置 w_0 之外的模型参数。这是因为偏置 w_0 表示拟合目标均值 $\bar{y}$ 与基函数的加权平均之间的差异（详见公式3.10）。具体而言，当所有拟合目标值 y_i 被整体平移（即加上同一个常数 c）时，偏置 w_0 也会相应地增加 c。为了不限制偏置项对拟合目标均值变化的适应能力，正则化约束不应施加于偏置 w_0。

为了便于分析岭回归的实际作用，并加深对其机制的理解，我们对特征值（或基函数的值）进行中心化处理，即通过平移使每一特征的均值为零。具体地，我们首先按照公式3.12计算每个特征的平均值 $\overline{\phi_j}$，然后将每个样本的特征值 $\phi_j(\boldsymbol{x}_i)$ 替换为中心化后的值 $\phi_j(\boldsymbol{x}_i) - \overline{\phi_j}$（即减均值）。每一个特征在中心化后的均值都为零，因此根据公式3.10，偏置 w_0 的值可以先被确定为拟合目标均值 $\bar{y}$。为了简化后续推导，我们将偏置项 w_0 从参数向量 $\boldsymbol{w}$ 移除，仅保留与基函数相关的参数。同时，我们用 $\dot{\boldsymbol{\Phi}}$ 表示中心化后的矩阵 $\boldsymbol{\Phi}$。需要注意的是，除了进行中心化处理外，矩阵 $\dot{\boldsymbol{\Phi}}$ 相较于原始矩阵 $\boldsymbol{\Phi}$ 还去除了首列全为 1 的向量（该列对应偏置项 w_0）。

将岭回归问题的拉格朗日形式（公式3.32）表示为矩阵形式：

$$\frac{1}{2}\left(\dot{\mathbf{Y}} - \dot{\boldsymbol{\Phi}}\boldsymbol{w}\right)^{\top}\left(\dot{\mathbf{Y}} - \dot{\boldsymbol{\Phi}}\boldsymbol{w}\right) + \frac{\lambda}{2}\boldsymbol{w}^{\top}\boldsymbol{w} \tag{3.33}$$

其中 $\dot{\mathbf{Y}}$ 表示对原拟合目标向量 $\mathbf{Y}$ 进行中心化处理之后的结果，即每个元素均减去了拟合目标均值 $\bar{y}$。将上式3.33对参数 $\boldsymbol{w}$ 求导，并令导数为零，可得岭回归的闭式解：

$$\hat{\boldsymbol{w}}^{\text{ridge}} = \left(\dot{\boldsymbol{\Phi}}^{\top}\dot{\boldsymbol{\Phi}} + \lambda\mathbf{I}\right)^{-1}\dot{\boldsymbol{\Phi}}^{\top}\dot{\mathbf{Y}} \tag{3.34}$$

该闭式解的推导过程如下：

$$\begin{aligned}
-\dot{\boldsymbol{\Phi}}^{\top}\left(\dot{\mathbf{Y}} - \dot{\boldsymbol{\Phi}}\boldsymbol{w}\right) + \lambda\boldsymbol{w} = 0 \quad &\Rightarrow \\
\dot{\boldsymbol{\Phi}}^{\top}\dot{\boldsymbol{\Phi}}\boldsymbol{w} + \lambda\boldsymbol{w} = \dot{\boldsymbol{\Phi}}^{\top}\dot{\mathbf{Y}} \quad &\Rightarrow \\
\left(\dot{\boldsymbol{\Phi}}^{\top}\dot{\boldsymbol{\Phi}} + \lambda\mathbf{I}\right)\boldsymbol{w} = \dot{\boldsymbol{\Phi}}^{\top}\dot{\mathbf{Y}} \quad &\Rightarrow \\
\boldsymbol{w} = \left(\dot{\boldsymbol{\Phi}}^{\top}\dot{\boldsymbol{\Phi}} + \lambda\mathbf{I}\right)^{-1}\dot{\boldsymbol{\Phi}}^{\top}\dot{\mathbf{Y}} \quad &
\end{aligned}$$

从岭回归的闭式解公式3.34可以看出，引入了 ℓ_2 正则项之后，求解得到的参数向量 $\hat{\boldsymbol{w}}^{\text{ridge}}$ 仍然是拟合目标 $\dot{\mathbf{Y}}$ 的线性组合。然而，在计算矩阵 $\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}$ 的逆之前，先在其每个对角线元素上都加上一个正数 λ。这一操作确保了即使原矩阵 $\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}$ 因不满秩而不可逆，修正后的矩阵 $(\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}+\lambda\mathbf{I})$ 也必定是正定且可逆的。这是因为，对于任意非零向量 $\boldsymbol{v}$ 及 $\lambda>0$，我们有：

$$\boldsymbol{v}^\top\left(\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}+\lambda\mathbf{I}\right)\boldsymbol{v}=\|\dot{\boldsymbol{\Phi}}\boldsymbol{v}\|^2+\lambda\|\boldsymbol{v}\|^2>0 \tag{3.35}$$

此外，即使 $\dot{\boldsymbol{\Phi}}$ 中存在近似线性相关的列向量，ℓ_2 正则项亦能在一定程度上破坏这种线性相关性（更准确地说，正则项可以降低求解过程中所用矩阵的条件数），从而有效缓解模型参数求解过程中的数值不稳定性。

根据第一章第1.5节中的讨论，岭回归还可从贝叶斯分析的视角加以理解。具体而言，我们假设拟合目标 y_i 服从高斯分布 $\mathcal{N}(f(\boldsymbol{x}_i,\boldsymbol{w}),\beta^{-1})$，并且参数 $\boldsymbol{w}$ 服从高斯先验分布 $\mathcal{N}(\mathbf{0},\alpha^{-1}\mathbf{I})$。在这些假设下，通过最大化后验概率估计，可以推导出岭回归问题的拉格朗日形式（公式3.32）。其中正则化系数 λ 等于 α/β（见公式1.21），而岭回归的参数估计值对应于后验分布的均值。

对中心化后的矩阵 $\dot{\boldsymbol{\Phi}}$ 进行奇异值分解（Singular Value Decomposition）有助于我们更深入地理解岭回归的本质。奇异值分解在许多统计方法的分析中都具有重要作用，是一种非常有用的工具。对于形为 $N\times(M-1)$ 的矩阵 $\dot{\boldsymbol{\Phi}}$，其奇异值分解形式如下：

$$\begin{aligned}\dot{\boldsymbol{\Phi}}&=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \\ &=\begin{bmatrix}\\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_{M-1}\\ \\ \end{bmatrix}\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_{M-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_1^\top\\ \vdots\\ \boldsymbol{v}_{M-1}^\top\end{bmatrix}\\ &=\sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^\top+\cdots+\sigma_{M-1}\boldsymbol{u}_{M-1}\boldsymbol{v}_{M-1}^\top\end{aligned} \tag{3.36}$$

其中 $\mathbf{U}$ 和 $\mathbf{V}$ 分别是维度为 $N\times M-1$ 和 $M-1\times M-1$ 的正交矩阵（即列向量两两正交），$\boldsymbol{\Sigma}$ 是一个对角矩阵，并且除对角线外的元素均为零。矩阵 $\boldsymbol{\Sigma}$ 的对角线元素为非负的奇异值，并且按照从大到小的顺序排列 $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_{M-1}$。若一个或多个对角元素 σ_j 等于零，则矩阵 $\dot{\boldsymbol{\Phi}}$ 是奇异的（即不可逆）。奇异值分解常被用于数据压缩与降维。在对矩阵 $\dot{\boldsymbol{\Phi}}$ 所表示的数据进行压缩或降维时，通常会舍弃若干个最小的奇异值及其对应的 $\mathbf{U}$ 和 $\mathbf{V}$ 中的列向量，即仅保留如下公式中加紫色背景的主要成分：

$$\begin{aligned}\dot{\boldsymbol{\Phi}}&=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\\ &=\begin{bmatrix}\\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_K & \cdots & \boldsymbol{u}_{M-1}\\ \\ \end{bmatrix}\begin{bmatrix}\sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_K & & \\ & & & \ddots & \\ & & & & \sigma_{M-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_1^\top\\ \vdots\\ \boldsymbol{v}_K^\top\\ \vdots\\ \boldsymbol{v}_{M-1}^\top\end{bmatrix}\\ &=\sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^\top+\cdots+\sigma_K\boldsymbol{u}_K\boldsymbol{v}_K^\top+\cdots+\sigma_{M-1}\boldsymbol{u}_{M-1}\boldsymbol{v}_{M-1}^\top\end{aligned}$$

其中 $\{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_K\}$ 表示被保留下来的 K 个主要的左奇异向量，$\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_K\}$ 表示对应的 K 个右奇异向量，而 $\{\sigma_1,\ldots,\sigma_K\}$ 则为对应的奇异值。这些奇异值的大小反映了各奇异向量在表示原始数据时的重要性，也可以理解为在降维后重构原始数据时，各奇异向量所占的相对权重。

基于上述奇异值分解（公式3.36），最小二乘法的拟合结果可以表示为：

$$\begin{aligned}\dot{\boldsymbol{\Phi}}\hat{\boldsymbol{w}} &= \dot{\boldsymbol{\Phi}}\left(\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}\right)^{-1}\dot{\boldsymbol{\Phi}}^\top\dot{\mathbf{Y}}\\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\left(\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^\top\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^{-1}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^\top\dot{\mathbf{Y}}\\ &= \mathbf{U}\mathbf{U}^\top\dot{\mathbf{Y}}\ \text{（}\mathbf{U}\text{ 和 }\mathbf{V}\text{ 为列向量两两正交的正交矩阵）}\end{aligned} \tag{3.37}$$

以上推导过程中需要注意的是：由于矩阵 $\mathbf{V}$ 是方阵，则有 $\mathbf{V}^{-1}=\mathbf{V}^\top$，因此 $\mathbf{V}^\top\mathbf{V}=\mathbf{V}\mathbf{V}^\top=\mathbf{I}$。然而，$\mathbf{U}$ 并不是一个方阵，尽管它的列向量是正交的，因此有 $\mathbf{U}^\top\mathbf{U}=\mathbf{I}$，但是 $\mathbf{U}\mathbf{U}^\top\neq\mathbf{I}$。

我们知道，如果线性空间 $\mathcal{S}$ 中的整个向量组 $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_M\}$ 线性无关，并且线性空间 $\mathcal{S}$ 中的任一向量都可以表示为这组向量的线性组合，则称该向量组 $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_M\}$ 为 $\mathcal{S}$ 的一组基 [73]。若 $\boldsymbol{r}$ 是 $\mathcal{S}$ 中任一向量，则有：

$$\boldsymbol{r}=z_1\boldsymbol{e}_1+\cdots+z_M\boldsymbol{e}_M \tag{3.38}$$

其中系数 $(z_1,\cdots,z_M)$ 是向量 $\boldsymbol{r}$ 在基 $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_M\}$ 下的坐标。如果向量组 $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_M\}$ 还满足彼此正交且每个向量的 ℓ_2 范数为 1:

$$\boldsymbol{e}_i^\top\boldsymbol{e}_j=\begin{cases}1 & \text{若 } i=j\\ 0 & \text{若 } i\neq j\end{cases} \tag{3.39}$$

则称该向量组为 $\mathcal{S}$ 的一个标准正交基（Orthonormal Basis）。

从公式3.37可以看出，等式左侧的 $\dot{\boldsymbol{\Phi}}\hat{\boldsymbol{w}}$ 是矩阵 $\dot{\boldsymbol{\Phi}}$ 列向量的线性组合，因此必然落在由这些列向量所张成的列空间中。而等式右侧 $\mathbf{U}\mathbf{U}^\top\dot{\mathbf{Y}}$ 中的 $\mathbf{U}^\top\dot{\mathbf{Y}}$ 可以看作是 $\dot{\mathbf{Y}}$ 在由 $\mathbf{U}$ 的列向量所组成的标准正交基下的坐标。结合公式3.17，我们知道最小二乘解本质上是将 $\dot{\mathbf{Y}}$ 投影到由 $\dot{\boldsymbol{\Phi}}$ 的列向量所张成的线性子空间，而矩阵 $\mathbf{U}$ 的列向量正好构成了该列空间的一组标准正交基。

现在让我们来看岭回归的拟合结果，它可以表示为：

$$\begin{aligned}\dot{\boldsymbol{\Phi}}\hat{\boldsymbol{w}}^{\text{ridge}} &= \dot{\boldsymbol{\Phi}}\left(\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}+\lambda\mathbf{I}\right)^{-1}\dot{\boldsymbol{\Phi}}^\top\dot{\mathbf{Y}}\\ &= \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\left(\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^\top\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top+\lambda\mathbf{I}\right)^{-1}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^\top\dot{\mathbf{Y}}\\ &= \mathbf{U}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2+\lambda\mathbf{I})^{-1}\boldsymbol{\Sigma}\mathbf{U}^\top\dot{\mathbf{Y}}\ \text{（}\mathbf{U}\text{ 和 }\mathbf{V}\text{ 为列向量两两正交的正交矩阵）}\\ &= \sum_{j=1}^{M-1}\boldsymbol{u}_j\frac{\sigma_j^2}{\sigma_j^2+\lambda}\boldsymbol{u}_j^\top\dot{\mathbf{Y}}\ \text{（对角矩阵求逆只需将对角线上的元素取倒数）}\end{aligned} \tag{3.40}$$

通过对比公式3.37和公式3.40可知，岭回归本质上仍然是在求解拟合目标 $\dot{\mathbf{Y}}$ 在标准正交基 $\mathbf{U}$ 下的坐标。然而，与普通最小二乘法不同的是，该坐标在岭回归中会被一个缩放因子 $\sigma_j^2/(\sigma_j^2+\lambda)$ 所缩放。因为正则化系数 λ 一定为正，所以该缩放因子始终小于 1。当奇异值的平方 σ_j^2 较大时，

此缩放因子更趋近于 1，这意味着在对应方向 $\boldsymbol{u}_j$ 上相对压缩较小，保留了该方面的大部分原始信息，而当 σ_j^2 较小时，此缩放因子更趋近于 0，表示该方向上的分量被显著抑制。

接下来，我们探讨为什么依据 σ_j^2 的大小对不同方向施加不同程度的缩放是合理且有效的。在此之前，我们需要先明确 σ_j^2 所代表的含义。我们可以将 $\dot{\boldsymbol{\Phi}}$ 的每一列看作一个随机变量在 N 个样本上的观测值。由于该矩阵 $\dot{\boldsymbol{\Phi}}$ 已经经过中心化处理，即每一特征的均值为零，那么这 $M-1$ 个随机变量的协方差矩阵 $\mathbf{S}$ 可以表示为：

$$\mathbf{S} = \frac{1}{N}\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}} = \frac{1}{N}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\right)^\top\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \frac{1}{N}\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top \tag{3.41}$$

由上式可知，矩阵 $\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}$ 的特征值分解形式为 $\mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top$，其中 $\mathbf{V}^\top = \mathbf{V}^{-1}$。我们知道，对于一个单位向量，任意向量在其方向上的投影可以通过与该单位向量的点积计算得到。因此，我们从 $\mathbf{V}$ 提取出对应最大特征值的特征向量 $\boldsymbol{v}_1$，并计算 $\boldsymbol{z}_1 = \dot{\boldsymbol{\Phi}}\boldsymbol{v}_1$，则向量 $\boldsymbol{z}_1$ 中的每一个分量表示某个样本的特征表示（即中心化后的基函数取值向量）在方向 $\boldsymbol{v}_1$ 上的投影，也就是该样本在向量 $\boldsymbol{v}_1$ 方向上的特征表达（即提取这个方向上的特征）。我们可以计算 $\boldsymbol{z}_1$ 的方差如下（此处将向量 $\boldsymbol{z}_1$ 中的每个分量视为一个样本观测结果）：

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{z}_1] &= \mathrm{Var}[\dot{\boldsymbol{\Phi}}\boldsymbol{v}_1] \\
&= \mathrm{Var}[\boldsymbol{u}_1\sigma_1] && (\dot{\boldsymbol{\Phi}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \text{ 两边右乘 } \mathbf{V} \text{ 可得 } \dot{\boldsymbol{\Phi}}\mathbf{V} = \mathbf{U}\boldsymbol{\Sigma}) \\
&= \sigma_1^2\mathrm{Var}[\boldsymbol{u}_1] && (\text{矩阵 } \dot{\boldsymbol{\Phi}} \text{ 已经中心化处理，因此 } \mathbb{E}[\boldsymbol{z}_1] = 0, \\
&= \frac{\sigma_1^2}{N} && \mathrm{Var}[\boldsymbol{u}_1] = \frac{1}{N}\boldsymbol{u}_1^\top\boldsymbol{u}_1, \text{ 并且 } \boldsymbol{u}_1 \text{ 为单位向量})
\end{aligned} \tag{3.42}$$

按上述方法计算得到 $\boldsymbol{z}_1$ 被称为矩阵 $\dot{\boldsymbol{\Phi}}$ 的第一主成分，而 $\boldsymbol{u}_1$ 是归一化后（Normalized）的第一主成分。结合前文的分析可以看出，岭回归在求解拟合目标 $\dot{\mathbf{Y}}$ 在标准正交基 $\mathbf{U}$ 下的坐标时，由于第一主成分方向对应的样本方差 σ_1^2/N 最大，因此在该方向上的坐标被缩减得最小（公式3.40）。此后依次得到的主成分 $\boldsymbol{z}_j$，在保持与之前所有主成分正交的约束下，具有在剩余方向中最大的样本方差 σ_j^2/N，其对应方向上的坐标也会根据方差的大小被相应地缩减。总之，岭回归对样本方差较大的方向（即奇异值较大的方向）施加较小的缩减；而对于样本方差较小的方向（奇异值较小的方向），则施加更强的压缩。这一策略的合理性在于，在方差较大的方向上，不同样本在相应特征上的差异更加明显，因而沿这些方向观察样本的特征及其变化，有助于提取对模型拟合更有价值的结构信息。而方差较小的方向往往反映的是由噪声主导的微小变化（如图3.3的例子所示），若不加限制地利用，模型容易对噪声产生过度响应，进而导致过拟合。

类似岭回归所采用的 ℓ_2 正则化的思想核心在于实现偏差与方差之间的权衡。通过对参数的取值施加约束，允许模型在偏差上略有增加，从而换取对预测方差更大幅度的下降（参数 $\boldsymbol{w}$ 的估计偏差与方差最终会传递到模型预测结果的偏差与方差），最终使整体期望预测误差减小。从公式3.22已知，参数估计的协方差为 $(\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}})^{-1}\sigma_\epsilon^2$。岭回归通过在 $\dot{\boldsymbol{\Phi}}^\top\dot{\boldsymbol{\Phi}}$ 的对角线元素加上正则化系数 λ，直接增加了小特征值的数值，不仅缓解了求逆过程中的数值不稳定性，也抑制了由噪声引起的低方差主成分方向上的特征表达，从而提升了模型的泛化能力。此外，岭回归并未对特征进行离散的取舍，而是通过在不同主成分方向上施加不同程度的缩减，实现了本节开始

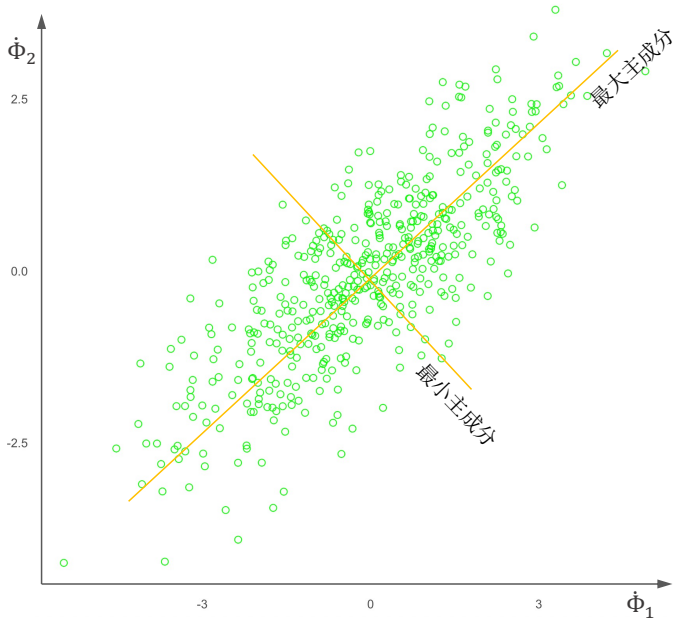


图 3.3: 二维数据的主成分方向示意图。最大主成分对应数据方差最大的方向，最小主成分则对应方差最小的方向。岭回归会先将拟合目标投影到这些主成分方向上，并对低方差方向的坐标施加比高方差方向更强的缩减。

时所期望的对特征利用更为平滑的连续调节方式。

在线性回归模型中对参数引入 ℓ_1 范数（即曼哈顿范数）约束的方法被称为**套索回归**或**拉索回归**（Lasso Regression）。套索回归的优化问题可以表示为：

$$\begin{aligned}\hat{\boldsymbol{w}}^{\text{lasso}} &= \arg\min_{\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{M-1}\phi_j(\boldsymbol{x}_i)w_j\right)^2 \\ &\text{s.t.} \sum_{j=1}^{M-1}|w_j| \leq t\end{aligned} \tag{3.43}$$

在上述公式中，约束条件要求除偏置项 w_0 外，其余参数的 ℓ_1 范数不超过给定阈值 t。相同地，套索回归问题也可以表示为如下等价的拉格朗日形式：

$$\hat{\boldsymbol{w}}^{\text{lasso}} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2}\sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{M-1}\phi_j(\boldsymbol{x}_i)w_j\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{M-1}|w_j|\right) \tag{3.44}$$

从贝叶斯分析的角度来看，套索回归与岭回归均假设拟合目标服从高斯分布 $\mathcal{N}(f(\boldsymbol{x},\boldsymbol{w}),\beta^{-1})$。两者的主要区别在于对参数 $\boldsymbol{w}$ 所采用的先验分布：岭回归假设参数 $\boldsymbol{w}$ 服从高斯先验分布，而套索回归则将该先验分布替换为**拉普拉斯分布**（Laplace Distribution）。拉普拉斯分布的概率密度函数为：

$$\text{Laplace}(x|\mu,\tau) = \frac{1}{2\tau}\exp\left(-\frac{|x-\mu|}{\tau}\right) \tag{3.45}$$

其中 μ 称为位置参数（Location Parameter），用于确定分布的中心位置；而 $\tau > 0$ 称为尺度参数

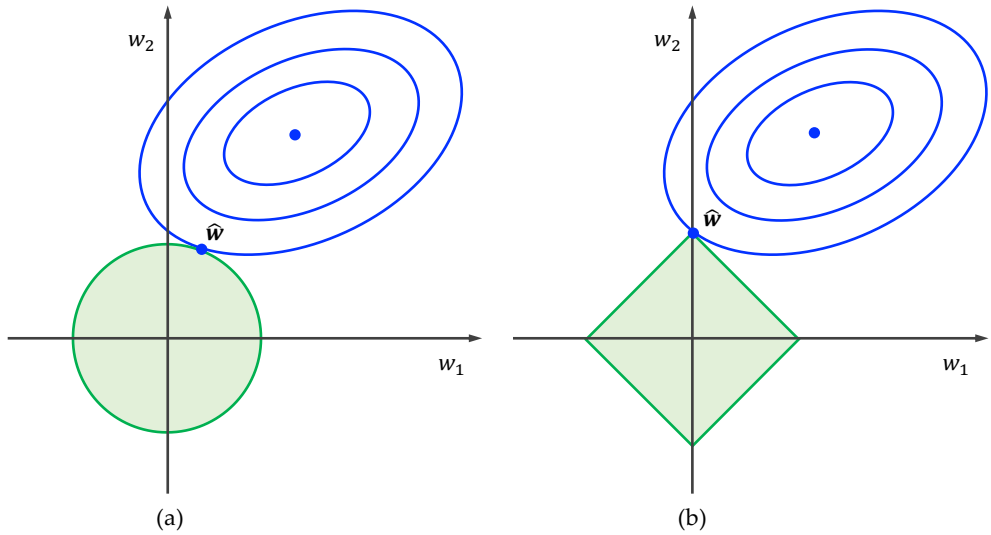


图 3.4: 岭回归与套索回归参数估计的示意图。蓝色等高线表示无正则化的误差函数，绿色区域表示正则所对应的约束范围。(a) 引入 ℓ_2 正则（$w_1^2 + w_2^2 \leq t^2$）的岭回归（Ridge）。(b) 引入 ℓ_1 正则（$|w_1| + |w_2| \leq t$）的套索回归（Lasso）。参数向量的最优解记为 $\hat{\boldsymbol{w}}$（蓝色等高线与绿色区域相交的点）。套索回归的正则化促使其解具有稀疏性，此处表现为 $\hat{w}_1 = 0$。

（Scale Parameter），用于控制分布的宽度。套索回归中，我们假设模型的每个参数 w_j 服从拉普拉斯先验分布 Laplace$(w_j|0,\tau)$。在这种假设下，所有参数的联合先验分布可以表示为：

$$\left(\frac{1}{2\tau}\right)^{M-1} \prod_{j=1}^{M-1} \exp\left(-\frac{|w_j|}{\tau}\right) \tag{3.46}$$

根据贝叶斯公式，参数 $\boldsymbol{w}$ 的后验分布正比于其先验分布和似然函数（即拟合目标所服从的高斯分布）的乘积。对该乘积取对数，并略去与参数 $\boldsymbol{w}$ 无关的常数项后，对于参数 $\boldsymbol{w}$ 的先验部分，我们可以得到如下表达式：

$$-\frac{1}{\tau} \sum_{j=1}^{M-1} |w_j| \tag{3.47}$$

由于最大化后验分布等价于最小化其负对数形式，因此以上表达式中的负号可以转化为正号。我们令 $\lambda = \frac{2}{\tau\beta}$，即可得到公式3.44所示的优化目标。

接下来我们讨论如何求解带有 ℓ_1 范数正则项的线性回归问题。与引入 ℓ_2 范数的正则不同，公式3.44在 $w_j = 0$ 处不可导，因此无法直接求得闭式解。一种常见的解决方法是将公式3.44所表示的最优化问题转化为一个二次规划问题。具体而言，我们可以为每个参数 w_j 引入两个非负的辅助变量 w_j^+ 和 w_j^- 来表示 w_j：

$$w_j = w_j^+ - w_j^- \quad \text{且} \quad w_j^+ \geq 0, w_j^- \geq 0 \tag{3.48}$$

通过上述变量替换，公式3.44可以改写为如下形式：

$$\hat{\boldsymbol{w}}^{\text{lasso}} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2} \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{M-1} \phi_j(\boldsymbol{x}_i)(w_j^+ - w_j^-) \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{M-1} (w_j^+ + w_j^-) \right)$$
$$\text{s.t.} \quad w_j^+ \geq 0, w_j^- \geq 0 \tag{3.49}$$

值得注意的是，由于我们希望最小化目标函数，其中的项 $\sum_{j=1}^{M-1}(w_j^+ + w_j^-)$ 会促使每个参数 w_j 仅可能取正值（$w_j^- = 0$）、负值（$w_j^+ = 0$）或零（$w_j^+ - w_j^- = 0$）这三种情况之一。公式3.49可以非常容易地转换为标准形式的二次规划问题，然后可以通过多种经典优化算法进行求解，例如内点法（Interior-point）、有效集法（Active Set）、椭球法（Ellipsoid）、增广拉格朗日法（Augmented Lagrangian）、梯度投影法（Gradient Projection）等 [13]。图3.4展示了在二维情况下，岭回归与套索回归参数估计的示意图。套索回归由于引入了 ℓ_1 正则项，具备产生稀疏解（Sparse Solution）的能力，即部分参数的值可能为零，从而实现对特征的自动选择。

另一种常用且高效的求解套索回归最优化问题的方法是坐标下降法（Coordinate Descent），这是一种迭代优化算法。其核心思想是：在每次迭代中，将所有其他参数保持固定，仅沿某一个坐标方向（即对某一个参数）进行优化更新。具体而言，在每一步迭代中，固定除参数 w_k 外的所有参数的估计值，此时对于第 i 个样本，其残差可以表示为：

$$r_i^{(k)} = y_i - w_0 - \sum_{j \neq k} \phi_j(\boldsymbol{x}_i) w_j \tag{3.50}$$

其中 $r_i^{(k)}$ 表示在当前迭代中，第 i 个样本相对于参数 w_k 的部分残差。在仅优化单个参数 w_k 的情况下，其对应的优化问题可以表示为：

$$\hat{w}_k^{\text{lasso}} = \arg\min_{w_k} \left(\frac{1}{2} \sum_{i=1}^{N} \left(r_i^{(k)} - \phi_k(\boldsymbol{x}_i) w_k \right)^2 + \frac{\lambda}{2} |w_k| \right) \tag{3.51}$$

由于仅对单个参数 w_k 进行优化，我们可以根据 w_k 可能的不同取值情形，分别进行分析和求解。具体来说，我们需要考虑三种情况：$w_k > 0$、$w_k < 0$ 和 $w_k = 0$。

首先，假设 $w_k > 0$ 的情况下，我们对公式3.51关于 w_k 求偏导，并令导数为零，可得：

$$\hat{w}_k = \frac{\sum_{i=1}^{N} r_i^{(k)} \phi_k(\boldsymbol{x}_i) - \frac{\lambda}{2}}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)} = \underbrace{\frac{\sum_{i=1}^{N} r_i^{(k)} \phi_k(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)}}_{\text{记为 } \hat{w}_k^{\text{ols}}} - \underbrace{\frac{\frac{\lambda}{2}}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)}}_{\text{记为 } \rho} \tag{3.52}$$

由于我们假设 $w_k > 0$，因此若上式结果大于零（即 $\hat{w}_k^{\text{ols}} > \rho$），则可根据该公式更新 w_k 的取值；若结果不大于零，则应令 $w_k = 0$。

其次，假设 $w_k < 0$ 的情况下，我们将公式3.51对 w_k 求偏导并令其为零，我们可以得到：

$$\hat{w}_k = \frac{\sum_{i=1}^{N} r_i^{(k)} \phi_k(\boldsymbol{x}_i) + \frac{\lambda}{2}}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)} = \underbrace{\frac{\sum_{i=1}^{N} r_i^{(k)} \phi_k(\boldsymbol{x}_i)}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)}}_{\text{记为 } \hat{w}_k^{\text{ols}}} + \underbrace{\frac{\frac{\lambda}{2}}{\sum_{i=1}^{N} \phi_k^2(\boldsymbol{x}_i)}}_{\text{记为 } \rho} \tag{3.53}$$

同理，考虑到我们当前的假设为 $w_k < 0$，因此若上式结果小于零（即 $\hat{w}_k^{\text{ols}} < -\rho$），则可据此公

算法 3.1: 基于坐标下降的套索回归求解算法

输入: 训练样本集、正则化系数 λ、最大迭代次数 T 和最小变化阈值 ϵ。

初始化: 所有参数 $\{\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_{M-1}\}$ 都设为零，并设迭代次数 $t = 1$;
对拟合目标和特征值（或基函数的值）进行中心化处理;
根据公式3.10计算参数 $\hat{w}_0$;
while $t \leq T$ **do**
  for $k \leftarrow 1$ **to** $M - 1$ **do**
    根据公式3.50计算所有样本在当前迭代中相对于参数 w_k 的部分残差;
    根据公式3.52和公式3.53计算无正则化的最小二乘解 $\hat{w}_k^{\text{ols}}$ 和阈值 ρ;
    使用公式3.54更新参数 $\hat{w}_k$;
  收敛条件: 当所有参数 $\hat{w}_k$ 的更新幅度均小于最小变化阈值 ϵ 时，算法终止;
  $t \leftarrow t + 1$;

输出: 在给定正则化系数 λ 时，套索回归参数的估计 $\hat{\boldsymbol{w}}^{\text{lasso}}$。

式更新 w_k 的取值；若结果大于零，则应令 $w_k = 0$。

最后，假设 $w_k = 0$ 的情况下，则参数 w_k 的取值自然应为零。从公式3.52和公式3.53可以看出，它们最后两项 $\hat{w}_k^{\text{ols}}$ 和 ρ 是完全相同的，只是其间的正负号不同。这里，$\hat{w}_k^{\text{ols}}$ 表示在不引入正则化项时，使用普通最小二乘法（Ordinary Least Squares）得到的参数估计值。结合参数 w_k 取值的三种情形，我们可以得到如下的更新公式：

$$\hat{w}_k = \begin{cases} \hat{w}_k^{\text{ols}} - \rho & \text{若 } \hat{w}_k^{\text{ols}} > \rho \\ 0 & \text{若 } |\hat{w}_k^{\text{ols}}| \leq \rho \\ \hat{w}_k^{\text{ols}} + \rho & \text{若 } \hat{w}_k^{\text{ols}} < -\rho \end{cases} \tag{3.54}$$

以上公式采用了软阈值（Soft Thresholding）方法，其中 ρ 表示阈值。软阈值方法不仅将绝对值低于阈值的参数设置为零，还会对超过阈值的参数进行收缩（即减去或加上阈值 ρ）。基于上述推导，算法3.1给出了使用坐标下降法求解套索回归问题的完整过程。

岭回归与套索回归可以推广至更一般的形式，其对应的优化问题如下所示：

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left(\frac{1}{2} \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{M-1} \phi_j(\boldsymbol{x}_i) w_j \right)^2 + \frac{\lambda}{2} \sum_{j=1}^{M-1} |w_j|^q \right) \tag{3.55}$$

在公式3.55中，当 $q = 1$ 时，即为套索回归；而当 $q = 2$ 时，则为岭回归。图3.5展示了超参数 q 取不同典型值时，正则化项所施加约束的轮廓与形状。

我们还可以将不同取值 q 的正则项进行组合，从而构建出新的正则化形式。例如，文献 [80] 中提出的弹性网（Elastic-net）正则化的形式如下：

$$\lambda \sum_{j=1}^{M-1} \left(\alpha w_j^2 + (1 - \alpha) |w_j| \right) \tag{3.56}$$

弹性网正则是岭回归和套索回归之间的一种折中方法。它既像套索回归那样进行自动特征选择，又能像岭回归那样，按样本方差大小对参数进行压缩。当公式3.56中的超参 α 取 0.2 时，其对应

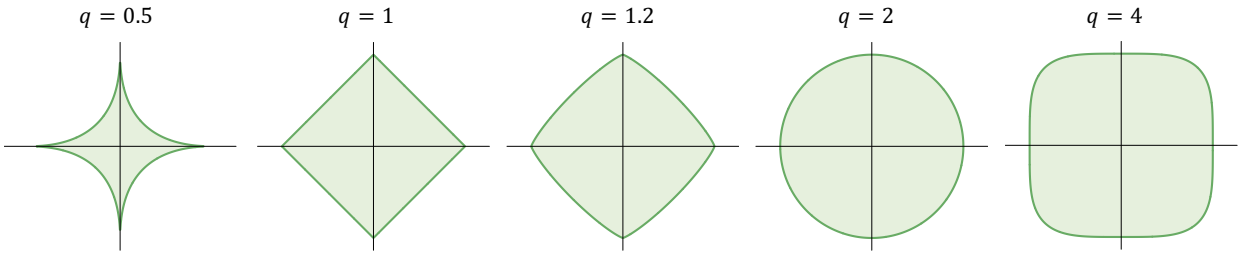


图 3.5: 引入正则的线性回归问题中，超参 q（见公式3.55）取不同值时对应的约束轮廓示意图。

的约束轮廓与图3.5中超参 $q = 1.2$ 时的轮廓形状非常相似。因而，取 $q \in (1, 2)$ 的正则项也可以被视为岭回归与套索回归之间的另一种折中形式。不过需要注意的是，$|w_j|^q$ 在 $q > 1$ 时在零点处是可微的，因此理论上不具备套索回归（即 $q = 1$）那种将某些系数精确地压缩为零的能力。

3.4 线性分类

分类是将 D 维输入 $\boldsymbol{x} = (x_1, \ldots, x_D)^\top$ 划分到 K 个离散的类别之一。通常情况下，每个输入 $\boldsymbol{x}$ 只能被分到唯一的一个类别。因此，输入空间被划分为 K 个互不重叠的区域，每个区域对应一个类别 $\mathcal{C}_k$，其中 $k \in \{1, 2, \ldots, K\}$。这些区域之间的分界被称为**决策边界**（Decision Boundary）。对于线性分类模型，决策边界由输入向量 $\boldsymbol{x}$ 的线性函数所定义。这意味着，决策边界是 D 维输入空间中 $D-1$ 维的仿射集合（Affine Set）或**超平面**（Hyperplane）[1]。图3.6展示了输入向量为二维和三维时的线性分类示意图。在二维空间中，线性决策边界是一条直线；而在三维空间中，线性决策边界则是一个平面。在图3.6的两个例子中，所有数据点被相应的线性决策边界完美地划分成蓝色和红色两类，没有出现误分类的情况。这种情形称为**线性可分**（Linearly Separable）。然而，在实际应用中，数据往往不是完全线性可分的。因此，线性分类的目标通常是寻找一个使分类错误率尽可能低的最优决策边界。

在构建线性分类模型时，通常可以采用三种不同的方法。第一种是线性判别模型。该方法为每个类别 $\mathcal{C}_k$ 构建一个形如 $f_k(\boldsymbol{x}) = w_{k0} + \boldsymbol{x}^\top \boldsymbol{w}_k$ 的判别函数（Discriminant Function），其中 w_{k0} 和 $\boldsymbol{w}_k$ 分别是类别 $\mathcal{C}_k$ 所对应的偏置项和权重参数，而判别函数 $f_k(\boldsymbol{x})$ 的值可以看作是输入 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_k$ 的得分。对于任意两个类别 $\mathcal{C}_i$ 和 $\mathcal{C}_j$，它们之间的决策边界由满足 $f_i(\boldsymbol{x}) = f_j(\boldsymbol{x})$ 的点集构成，即线性方程 $(w_{i0} - w_{j0}) + \boldsymbol{x}^\top(\boldsymbol{w}_i - \boldsymbol{w}_j) = 0$ 的解集合。第二种是概率判别模型。该方法直接对条件概率 $p(\mathcal{C}_k|\boldsymbol{x})$ 进行建模，然后利用训练样本估计出该条件概率的参数。条件概率的形式通常是将输入 $\boldsymbol{x}$ 的线性函数再通过某种映射转换为 $[0, 1]$ 区间内的概率值。第三种方法是概率生成模型。此方法建模样本 $\boldsymbol{x}$ 和类别 $\mathcal{C}_k$ 的联合分布 $p(\boldsymbol{x}, \mathcal{C}_k)$，该联合分布可以分解为 $p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)$，其中 $p(\boldsymbol{x}|\mathcal{C}_k)$ 为给定类别 $\mathcal{C}_k$ 的情况下生成 $\boldsymbol{x}$ 的概率，而 $p(\mathcal{C}_k)$ 为类别 $\mathcal{C}_k$ 的先验概率。当利用训练样本估计出联合分布的参数后，就可以使用以下贝叶斯公式来计算 $\boldsymbol{x}$ 属于

[1]严格来说，经过原点的超平面是线性子空间，而不经过原点的超平面则是仿射集合。在机器学习中，通常将这两种情况统称为超平面，本文也沿用这一习惯。

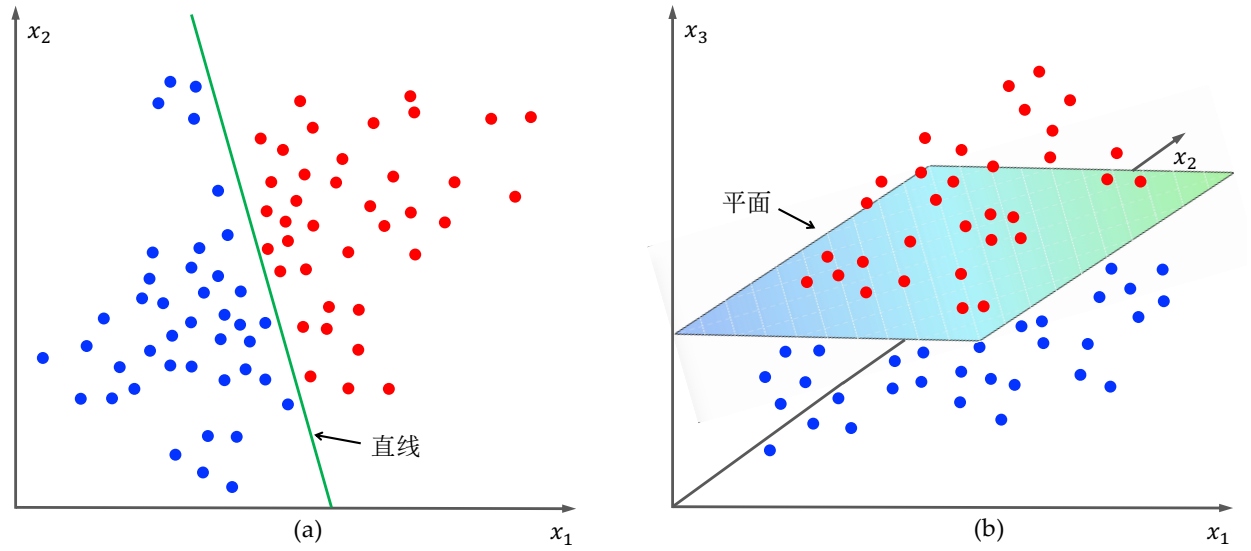


图 3.6: 线性分类示意图。蓝色与红色的点分别表示属于两个不同类别的数据。(a) 输入空间为二维时，决策边界是一条直线。(b) 输入空间为三维时，决策边界是一个平面。

类别 $\mathcal{C}_k$ 的后验概率：

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)} \tag{3.57}$$

线性函数 $f(\boldsymbol{x}) = w_0 + \boldsymbol{x}^\top \boldsymbol{w}$ 会将输入 $\boldsymbol{x}$ 映射到实数域。对于回归问题，这正是我们所期望的，因为输出目标通常取连续的实数。然而，在分类任务中，我们希望将输入 $\boldsymbol{x}$ 映射到离散的类别。特别是在采用概率判别或生成模型的情况下，我们需要将线性函数的值进一步映射到 $[0, 1]$ 的区间内，以确保其可以解释为有效的概率值。为此，我们需要在线性函数 $f(\boldsymbol{x})$ 的基础上引入一个非线性变换函数 $\sigma(\cdot)$，从而得到如下形式的输出：

$$y(\boldsymbol{x}) = \sigma(w_0 + \boldsymbol{x}^\top \boldsymbol{w}) \tag{3.58}$$

该函数被称为**激活函数**（Activation Function）。在广义线性模型中，它也称为逆连接函数（Inverse Link Function），而其反函数则称为**连接函数**（Link Function）。这类非线性函数 $\sigma(\cdot)$ 通常要求是某种单调变换，即能够保持原有输入之间的顺序关系。这种单调性保证了线性函数的输出值与类别得分或概率之间具有一致的对应关系，从而保留了原线性函数对于类别倾向的建模与解释能力。我们将在第3.6节中进一步讨论这类非线性函数的具体形式与特性。上述形式也是**广义线性模型**（Generalized Linear Model）的核心结构。广义线性模型是传统线性回归模型的推广，它允许输出变量服从指数族中的不同概率分布，并通过连接函数，将输出变量的条件期望与输入特征的线性组合联系起来。因此，广义线性模型不仅包括传统的线性回归，也包括逻辑回归和泊松回归等模型。需要注意的是，引入非线性变换后，模型的最终输出不再是模型参数的线性函数，这使其在分析和计算上可能比普通线性回归模型更为复杂。不过，由于模型仍然以输入特征的线性组合为基础，相较于后续章节将要介绍的更为通用的非线性模型，广义线性模型依然具有结构简洁、易于分析的优点。

3.5 线性判别模型

我们首先来讨论线性判别模型（Linear Discriminant Model），即其决策边界由判别函数直接确定的情形。为便于理解，我们先从最基本的二分类问题（$K=2$）入手，然后再推广到类别数 $K>2$ 的多分类问题。在线性判别模型中，判别函数的最简单形式是输入向量的线性组合：

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{x}^\top \boldsymbol{w} \tag{3.59}$$

其中 $\boldsymbol{w}$ 为权重向量，而 w_0 为偏置项。由于偏置项 w_0 在分类问题中有特殊意义，我们将其单独表示，并用 $\boldsymbol{w}$ 表示除 w_0 之外的其他权重参数。在需要表示所有权重参数（包括偏置项）时，则使用 $\tilde{\boldsymbol{w}}$。对于二分类问题，我们只需要定义一个线性判别函数 $f(\boldsymbol{x})$。因为我们可以设定如下判别规则：当 $f(\boldsymbol{x}) \geq 0$ 时，判定输入 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_1$；而 $f(\boldsymbol{x}) < 0$ 时，则判定 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_2$。由此可见，若有 $\boldsymbol{x}^\top \boldsymbol{w} \geq -w_0$ 时，则 $f(\boldsymbol{x}) \geq 0$。因此，偏置项的负值 $-w_0$ 又称为分类的**阈值**（Threshold）。特别地，当输入 $\boldsymbol{x}$ 恰好位于决策边界（即两个类别的分界线）上时，则有 $f(\boldsymbol{x}) = 0$。

假设两个不同的数据点 $\boldsymbol{x}_a$ 和 $\boldsymbol{x}_b$ 均位于决策边界上，则它们对应判别函数的值均为零，即 $f(\boldsymbol{x}_a) = f(\boldsymbol{x}_b) = 0$。将它们代入线性判别函数，可得 $(\boldsymbol{x}_a - \boldsymbol{x}_b)^\top \boldsymbol{w} = 0$。这表明，向量 $\boldsymbol{x}_a - \boldsymbol{x}_b$ 与权重向量 $\boldsymbol{w}$ 正交。由于 $\boldsymbol{x}_a$ 和 $\boldsymbol{x}_b$ 可取决策边界上任意不同的两个点，权重向量 $\boldsymbol{w}$ 必然垂直于决策边界上的所有向量，即 $\boldsymbol{w}$ 为该决策边界的法向量。同理，若某一个点 $\boldsymbol{x}$ 位于决策边界上，即 $f(\boldsymbol{x}) = 0$，则该决策边界到原点的法向距离（Normal Distance）为：

$$\frac{\boldsymbol{x}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|} = -\frac{w_0}{\|\boldsymbol{w}\|} \tag{3.60}$$

上式左侧是数据点 $\boldsymbol{x}$ 在 $\boldsymbol{w}$ 方向上的投影长度。设向量 $\boldsymbol{x}$ 与 $\boldsymbol{w}$ 之间的夹角为 θ，根据点积公式 $\boldsymbol{x}^\top \boldsymbol{w} = \|\boldsymbol{x}\|\|\boldsymbol{w}\|\cos(\theta)$，可得向量 $\boldsymbol{x}$ 在 $\boldsymbol{w}$ 方向上的投影长度为：$\|\boldsymbol{x}\|\cos(\theta) = \frac{\boldsymbol{x}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|}$。由此可见，权重向量 $\boldsymbol{w}$ 决定了决策边界的方向，而偏置项 w_0 则决定了该决策边界相对于坐标原点的距离。图3.7展示当输入空间二维时，线性判别函数、决策边界及法向量（即权重向量 $\boldsymbol{w}$）之间的关系。

从图3.7可以看出，任意向量 $\boldsymbol{x}$ 可以表示为其在决策边界上的正交投影 $\boldsymbol{x}_\perp$ 与沿权重向量 $\boldsymbol{w}$ 方向的分量之和：

$$\boldsymbol{x} = \boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{3.61}$$

其中 r 表示 $\boldsymbol{x}$ 沿 $\boldsymbol{w}$ 方向到决策边界的带符号距离。当 r 取正号时，说明 $\boldsymbol{x}$ 位于 $\boldsymbol{w}$ 指向的一侧，可判为类别 $\mathcal{C}_1$；而当 r 取负号时，则位于 $\boldsymbol{w}$ 背向的一侧，对应类别 $\mathcal{C}_2$。将公式3.61的两边同时乘以 $\boldsymbol{w}$，并加上偏置项 w_0，我们可得：

$$\begin{aligned}
\boldsymbol{x}^\top \boldsymbol{w} + w_0 &= \boldsymbol{x}_\perp^\top \boldsymbol{w} + r\frac{\boldsymbol{w}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|} + w_0 && \Rightarrow \\
f(\boldsymbol{x}) &= r\frac{\boldsymbol{w}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|} \quad (f(\boldsymbol{x}) = w_0 + \boldsymbol{x}^\top \boldsymbol{w},\ \text{且 } f(\boldsymbol{x}_\perp) = w_0 + \boldsymbol{x}_\perp^\top \boldsymbol{w} = 0) && \Rightarrow \\
r &= \frac{f(\boldsymbol{x})}{\|\boldsymbol{w}\|} \quad (\frac{\boldsymbol{w}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|} = \frac{\|\boldsymbol{w}\|^2}{\|\boldsymbol{w}\|} = \|\boldsymbol{w}\|)
\end{aligned}$$

从上式可以看出，$f(\boldsymbol{x})$ 的绝对值正比于向量 $\boldsymbol{x}$ 到其在决策边界上的正交投影 $\boldsymbol{x}_\perp$ 之间的距离 r，

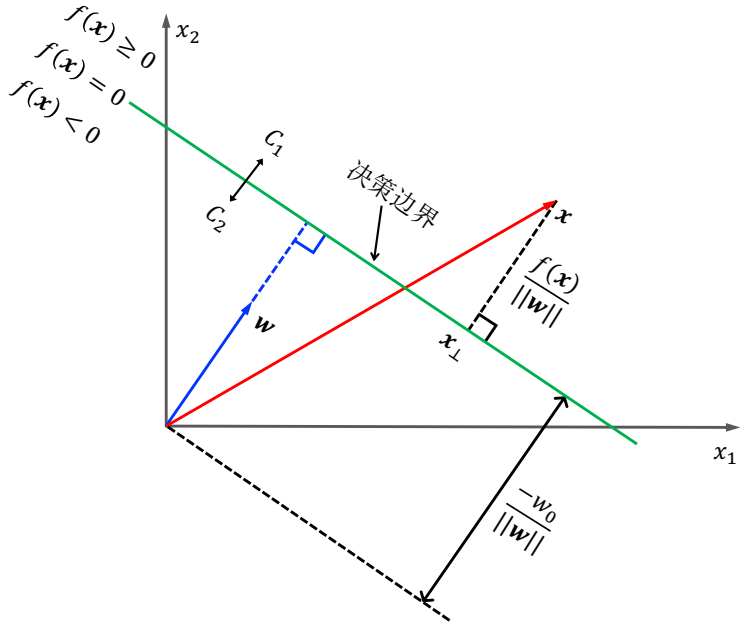


图 3.7: 线性判别函数在二维输入空间中的几何示意图。绿色线表示决策边界，它垂直于蓝色的权重向量 $\boldsymbol{w}$，其相对于原点的位置由偏置项 w_0 决定。此外，任意一点 $\boldsymbol{x}$ 到决策边界的带符号垂直距离为 $\frac{f(\boldsymbol{x})}{\|\boldsymbol{w}\|}$。

而 $f(\boldsymbol{x})$ 的符号则反映了输入 $\boldsymbol{x}$ 的类别归属。

我们接下来讨论如何将线性判别方法从二分类推广到多分类情形（即类别数 $K > 2$）。基本思路是通过构建多个二分类的线性判别函数，并将它们组合起来以实现多分类。主要有“一对其余”、“一对一”和“最大值判决”三种构建策略。其中前两种策略可能在某些区域出现类别无法判定的情况（如图3.8所示），而第三种“最大值判决”策略则能够有效避免这一问题。

- “一对其余”（One-versus-the-rest）策略：为每个类别单独构建一个用于区分该类别与其他所有类别的线性判别函数。对于 K 个类别，该策略通常需要构建 K 个判别函数。由于这些判别函数是独立训练的，在某些区域可能会出现多个判别函数均为正值，或者所有判别函数均为负值的情况。如果仅根据输出的正负号确定类别，就可能出现分类冲突或无法确定类别的问题。实际应用中，通常可以通过选择判别函数值最大的类别来解决这一问题。
- “一对一”（One-versus-one）策略：为每一对类别构建一个二分类的线性判别函数。对于输入 $\boldsymbol{x}$，所有判别函数分别作出判断，并通过投票方式决定其最终类别。该策略需要构建 $K(K-1)/2$ 个二分类判别函数，每个判别函数 $f_{ij}(\boldsymbol{x})$ 用于区分类别 $\mathcal{C}_i$ 与类别 $\mathcal{C}_j$。该策略仍然会出现因多个类别票数相同或判别结果冲突，从而导致无法明确分类。
- “最大值判决”策略：为每个类别各自构建一个线性判别函数，最终将输入 $\boldsymbol{x}$ 划归入对应判别函数值最大的类别。该策略不仅实现方式简洁，还能有效避免前两种策略中因冲突或区域未定义而导致的类别不确定问题。

在采用“最大值判决”策略进行多分类时，我们为每一个类别 $\mathcal{C}_k$ 构建相应的线性判别函数：

$$f_k(\boldsymbol{x}) = w_{k0} + \boldsymbol{x}^\top \boldsymbol{w}_k \tag{3.62}$$

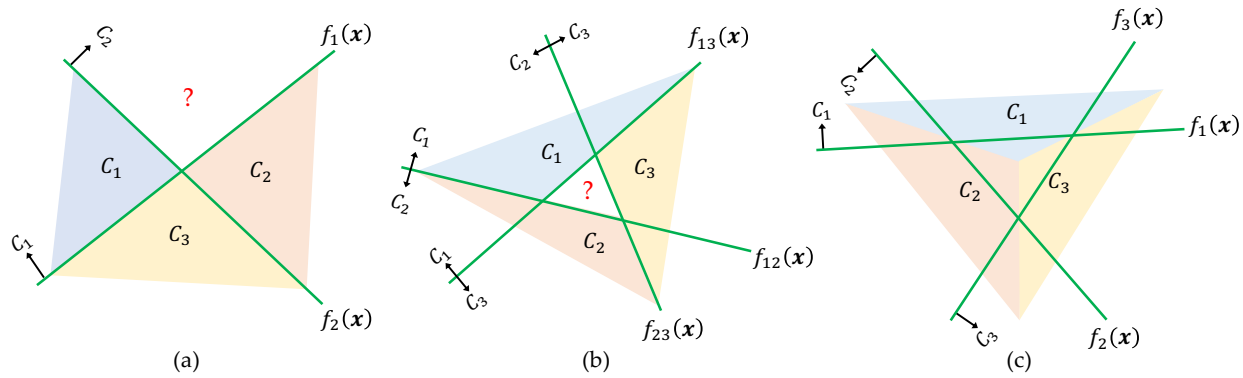


图 3.8: 基于多个二分类线性判别函数构建多分类模型的三种策略。(a)“一对其余” 策略。为每个类别单独构建一个区分该类别与其他类别的二分类判别函数。图中以问号标记的区域同时被判定为属于类别 $\mathcal{C}_1$ 和 $\mathcal{C}_2$，因此根据各判别函数的二分类结果无法确定其所属类别。(b)“一对一” 策略。为每对类别构建一个二分类判别函数，根据所有判别函数投票决定最终分类。此策略同样可能产生无法确定其类别的区域。(c)“最大值判决” 策略。为每个类别构建一个线性判别函数，最终将样本划归对应判别函数值最大的类别。该方法可以有效避免前两种策略可能出现的某些区域难以明确分类的问题。

对于给定输入样本 $\boldsymbol{x}$，若该样本对于所有的 $j \neq k$，均满足 $f_k(\boldsymbol{x}) > f_j(\boldsymbol{x})$，则将其判定为属于类别 $\mathcal{C}_k$。因此，类别 $\mathcal{C}_k$ 与 $\mathcal{C}_j$ 之间的决策边界由判别函数值相等的条件 $f_k(\boldsymbol{x}) = f_j(\boldsymbol{x})$ 确定，其对应的超平面方程为：

$$(w_{k0} - w_{j0}) + \boldsymbol{x}^\top(\boldsymbol{w}_k - \boldsymbol{w}_j) = 0 \tag{3.63}$$

这一形式与二分类情形下的决策边界基本一致，类似的几何性质与解释也同样适用。对于二分类问题，既可以按本节开头介绍的方式，仅构造一个线性判别函数，也可以采用 “最大值判决” 策略，分别为两个类别构建相应的判别函数 $f_1(\boldsymbol{x})$ 和 $f_2(\boldsymbol{x})$。对于多分类问题，除非另有说明，之后默认使用 “最大值判决” 作为分类策略。

接下来我们介绍三种具有代表性的线性判别函数参数学习方法：最小二乘法（Least Squares）、费雪线性判别（Fisher's Linear Discriminant）和感知机算法（Perceptron Algorithm）。

采用最小二乘法对线性判别函数的参数进行求解，本质上是将分类任务视为一个拟合问题。此时的拟合目标是以 1-of-K 编码或独热编码（One-hot Encoding）形式表示的类别向量。设共有 K 个类别，若某个样本 $\boldsymbol{x}$ 属于第 j 类，则其对应的 1-of-K 编码为一个 K 维向量，其中第 j 个元素为 1，其余元素为 0。例如，在一个五分类问题中，若样本属于第二类，则其 1-of-K 编码可以表示为：

$$\boldsymbol{t} = (0, 1, 0, 0, 0)^\top$$

如前所述，我们会为每一个类别 $\mathcal{C}_k$ 构建一个线性判别函数 $f_k(\boldsymbol{x}) = w_{k0} + \boldsymbol{x}^\top \boldsymbol{w}_k$，其中 $k \in \{1, 2, \ldots, K\}$。对于某个样本 $\boldsymbol{x}$，可以将这 K 个判别函数的输出值用如下形式的向量表示：

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_K(\boldsymbol{x})) = \tilde{\boldsymbol{x}}^\top \tilde{\mathbf{W}}$$

其中 $\boldsymbol{f}$ 是一个 K 维行向量，$\tilde{\boldsymbol{x}}$ 是在原始输入向量 $\boldsymbol{x}$ 的前面添加一个值为 1 的元素（对应于偏置

项）之后的增广向量，即 $\tilde{\boldsymbol{x}} = (1, \boldsymbol{x}^\top)^\top$，$\tilde{\mathbf{W}}$ 是所有 K 个线性判别函数的权重与偏置项合并后的矩阵，其第 k 列为 $D+1$ 维向量 $(w_{k0}, \boldsymbol{w}_k^\top)^\top$。我们将所有训练样本的增广向量合并为矩阵 $\tilde{\mathbf{X}}$，该矩阵的第 i 行对应第 i 个样本的增广向量 $\tilde{\boldsymbol{x}}_i^\top$。同样地，我们也将所有样本所属类别的 1-of-$K$ 编码组合成一个目标矩阵 $\mathbf{T}$，该矩阵的第 i 行为第 i 个样本的 1-of-K 编码 $\boldsymbol{t}_i^\top$。基于以上的符号定义，最小二乘法的平方误差损失函数可表示为：

$$\mathrm{RSS}(\tilde{\mathbf{W}}) \triangleq \frac{1}{2}\mathrm{tr}\left(\left(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T}\right)^\top \left(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T}\right)\right) \tag{3.64}$$

我们知道矩阵 $\mathbf{A}$ 的转置与自身相乘后的迹 $\mathrm{tr}(\mathbf{A}^\top\mathbf{A})$ 正好等于矩阵 $\mathbf{A}$ 所有元素的平方和（矩阵的迹为其主对角线元素的总和），也就是矩阵的 Frobenius 范数（记为 $\|\mathbf{A}\|_F$）的平方。在公式3.64中，矩阵 $(\tilde{\mathbf{X}}\tilde{\mathbf{W}} - \mathbf{T})$ 的每个元素表示模型对每个样本在各类别上的预测误差，而取迹的运算实际上是将所有误差的平方求和，从而得到整体的平方误差损失。将公式3.64对 $\tilde{\mathbf{W}}$ 求偏导，并令导数为零，可以得到与公式3.21相同形式的闭式解：

$$\tilde{\mathbf{W}} = \left(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^\top\mathbf{T} \tag{3.65}$$

虽然最小二乘法为线性判别函数的权重参数提供了简洁的闭式解。然而，最小二乘法对异常值缺乏必要的鲁棒性。我们已知，线性模型中最大化似然函数等价于最小化平方误差损失（见公式1.3）。从统计学角度看，这相当于在误差服从高斯分布的假设下，采用最大似然估计来求解参数。然而，高斯分布本身不具备重尾（Heavy Tail）特性，对少数离群点较为敏感（参见图2.14）。当数据中存在异常点时，平方误差损失便不再适用。如图3.9右图所示，仅仅在左下角添加少量异常样本，最小二乘法所得到的决策边界就发生了明显偏移，而这些新增异常点在左图原有的决策边界下依然可以被正确分类。其原因在于，平方误差函数不仅对预测误差进行惩罚，还会对“过于正确”的预测（如当某个样本在其所属类别对应的判别函数值已经远大于 1 时）进行惩罚。即使一个样本已经被正确分类，若其离决策边界过远，也会增加损失函数的值，从而导致模型在少数异常值的干扰下出现整体偏移。

在类别数 $K \geq 3$ 的多分类任务中，最小二乘法还会引发更严重的“类别掩蔽”问题。如图3.10左图所示，尽管蓝、橙、红三类样本数量相同（每类各有 100 个），橙色类别占据的判别区域（淡黄色背景）却明显偏小，被蓝色和红色类别“掩盖”。这一现象的原因在于，最小二乘法试图为每个类别拟合一个线性判别函数（在二维空间中为一条直线），使属于该类别样本的函数值接近于 1，而其他类别的函数值接近于 0。然而，对于位于中间区域的橙色类别，无论如何调整其线性判别函数的参数，都难以满足以下要求：对于左上方的蓝色样本，其判别函数值接近于 0；对于中间的橙色样本，其判别函数值接近于 1；而对于右下方的红色样本，其判别函数值又接近于 0。为了在左右两侧压缩函数值以逼近 0，橙色类别的判别函数不得不牺牲中间区域的响应，结果造成该判别函数往往过于平坦（对应直线趋于水平），其函数值总是低于其它两类的判别函数（左侧低于蓝色类别的判别函数，右侧则低于红色类别的判别函数）。即使该类样本的数量上并不处于劣势，其判别函数仍会被两侧的类别所压制，从而出现了“类别掩蔽”现象。

使用最小二乘法来求解线性判别函数的权重参数不仅对异常数据缺乏鲁棒性，在多分类问

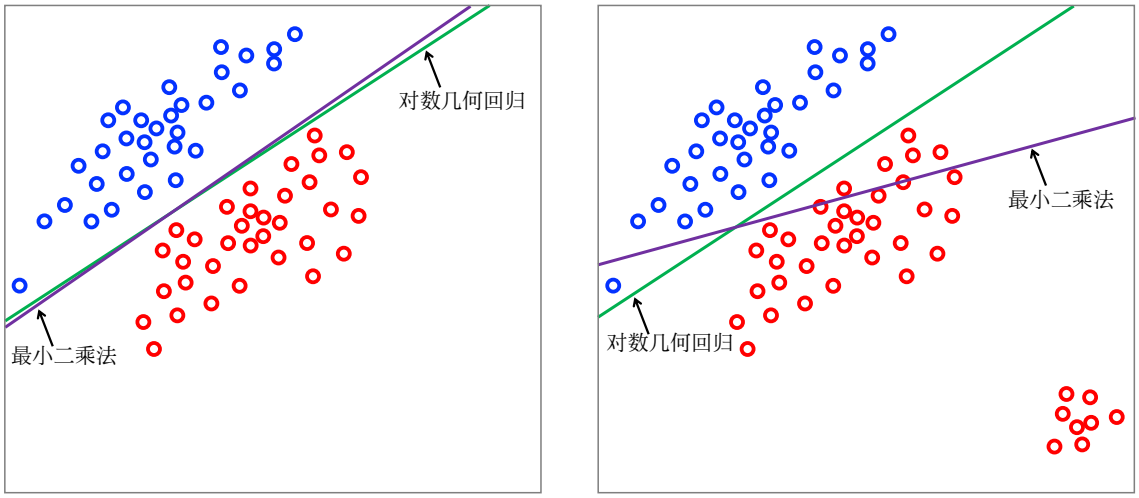


图 3.9: 左图展示了对于蓝色和红色圆圈表示的两类数据，分别采用最小二乘法（紫色直线）和对数几率回归（绿色直线）所得到的决策边界。我们将在第3.6节详细介绍对数几率回归（即 Logistic 回归）。右图显示了在左图数据的基础上，在左下角添加少量异常数据点对两种方法所得到决策边界的影响。从图中可以看出，与对数几率回归相比，最小二乘法对异常数据点非常敏感。

题中还会导致“类别掩蔽”现象。其原因在于，直接使用不受约束的线性函数拟合独热编码时，平方误差损失并不太适用于分类任务。对于分类任务，我们更关注样本是否被正确划分，而非预测值对目标值的精确逼近。然而，平方误差损失函数目标是最小化预测值与目标值之间的差异，即使某个样本已被正确分类，只要其判别函数的输出值与目标值之间不完全一致，损失函数仍会给出非零误差，并且这种误差会被平方进一步放大，从而导致决策边界偏离其最优位置。在后续内容中，我们将介绍更适合分类任务的损失函数，如交叉熵损失和感知机损失等。

线性分类问题还可以从降维的角度来理解。以二分类为例，先将原始 D 维的输入 $\boldsymbol{x}$ 投影到由权重向量 $\boldsymbol{w}$ 定义的一维子空间上，即计算投影值 $\boldsymbol{x}^\top \boldsymbol{w}$。然后通过设置阈值 w_0 来进行分类：若 $\boldsymbol{x}^\top \boldsymbol{w} \geq -w_0$，将 $\boldsymbol{x}$ 归为类别 $\mathcal{C}_1$；否则，归为类别 $\mathcal{C}_2$。一种直观的想法是：选择一个投影方向，使得两个类别在该方向上的投影均值尽可能远离，即最大化类别均值在投影空间的距离。假设存在包含 N_1 个样本点的类别 $\mathcal{C}_1$ 和 N_2 个样本点的类别 $\mathcal{C}_2$，它们的均值向量分别是：

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{\boldsymbol{x}_i \in \mathcal{C}_1} \boldsymbol{x}_i, \quad \boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{\boldsymbol{x}_i \in \mathcal{C}_2} \boldsymbol{x}_i \tag{3.66}$$

我们的目标是最大化两个类别均值在投影空间中的间隔：

$$\mu_1 - \mu_2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{w} \tag{3.67}$$

其中：

$$\mu_1 = \boldsymbol{\mu}_1^\top \boldsymbol{w}, \quad \mu_2 = \boldsymbol{\mu}_2^\top \boldsymbol{w} \tag{3.68}$$

即为两个类别的均值向量在权重向量 $\boldsymbol{w}$ 所定义方向上的投影。通过最大化公式3.67来求解 $\boldsymbol{w}$ 会面临以下两个问题。其一，只要不断增加 $\boldsymbol{w}$ 的模长，就可以使目标函数（即公式3.67）的值无限增大，从而导致该优化问题在当前形式下没有上界。为克服这一问题，可以引入约束条件，如

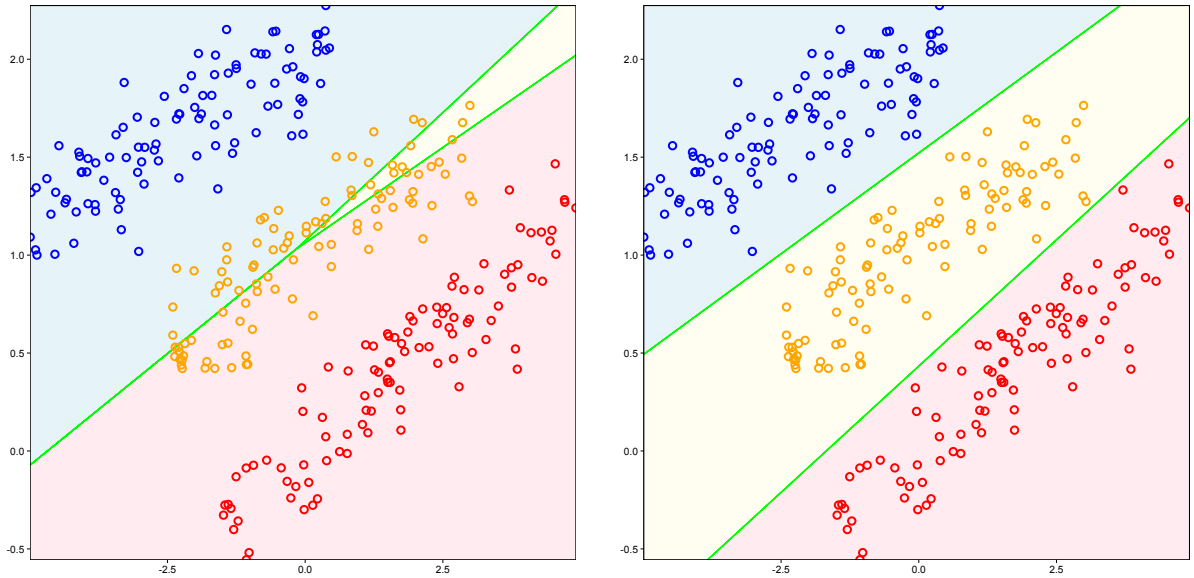


图 3.10: 左图显示了由蓝色、橙色和红色圆圈所表示的三类数据。图中绿色直线表示通过最小二乘法所得到的决策边界，背景颜色表示各类别在输入空间中的划分区域。可以看出，划分给橙色类别的区域（淡黄色背景）过小，导致该类别中的大部分样本被错误分类。右图则展示了对数几率回归方法所得到的决策边界和划分区域，该方法实现了对这些样本数据的准确分类。

规定 $\boldsymbol{w}$ 的模长为 1 加以解决。其二，即使两个类别在原始空间中是线性可分的，它们在投影到一维后仍可能产生重叠区域（如图3.11的左子图所示）。为了应对这一问题，费雪（Fisher）提出的思路是：在增大两类样本投影均值之间的距离的同时，尽可能减小各类别样本在该方向上的类内散度（散度通常表示样本偏离均值的平方和，与方差的主要区别在于不用除以样本数量），使同一类别的样本更集中，而不同类别的样本尽可能分离（如图3.11的右子图所示）。

两个类别在投影方向上的类内散度可以通过以下公式计算，即分别计算各类别中所有样本的投影值与该类别投影均值之差的平方和：

$$s_1^2 = \sum_{\boldsymbol{x}_i \in \mathcal{C}_1} (\boldsymbol{x}_i^\top \boldsymbol{w} - \mu_1)^2, \quad s_2^2 = \sum_{\boldsymbol{x}_i \in \mathcal{C}_2} (\boldsymbol{x}_i^\top \boldsymbol{w} - \mu_2)^2 \tag{3.69}$$

其中 μ_1 和 μ_2 分别表示两个类别的样本在投影方向上的均值。我们将两个类别的类内散度之和 $s_1^2 + s_2^2$ 定义为总类内散度。费雪准则（Fisher Criterion）的基本思想是，使不同类别的样本投影均值尽可能远离，同时使同一类别的样本投影值尽可能集中。为此，费希尔准则通过最大化两类投影均值之差的平方与总类内散度的比值，来确定最优的投影方向：

$$\frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \tag{3.70}$$

根据公式3.68和3.69的定义，上式可以改写为：

$$\frac{\boldsymbol{w}^\top \mathbf{B} \boldsymbol{w}}{\boldsymbol{w}^\top \mathbf{S} \boldsymbol{w}} \tag{3.71}$$

其中 $\mathbf{B}$ 称为类间散度矩阵，其定义如下：

$$\mathbf{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \tag{3.72}$$

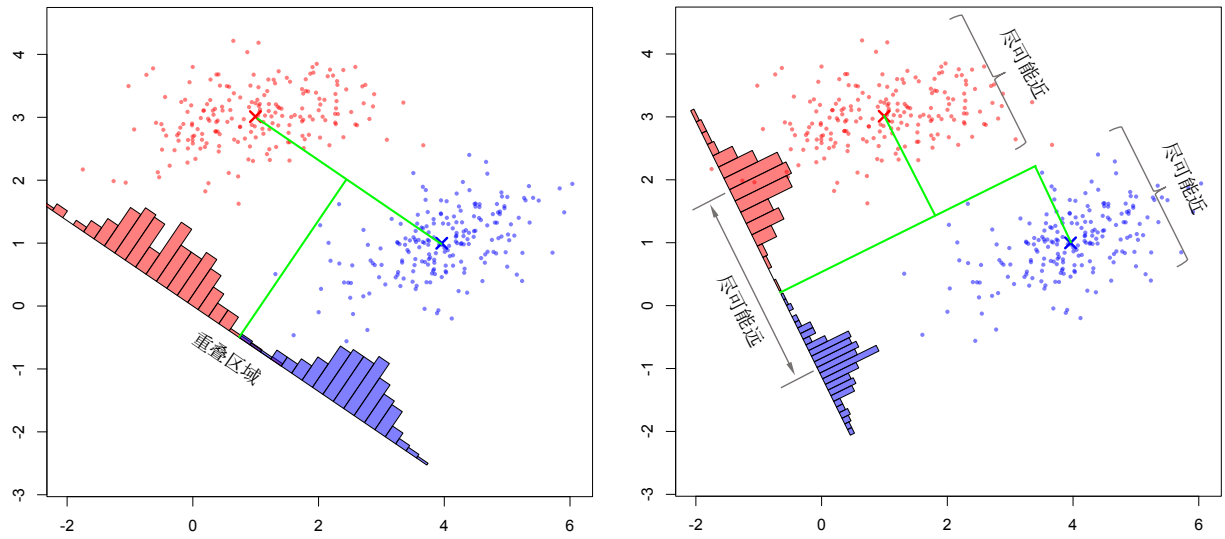


图 3.11: 费雪线性判别思想的示意图。左图展示了将红色和蓝色两类样本投影到连接两类均值（以叉号表示）的直线上的情形。尽管这些样本点在二维空间中是线性可分的，但从投影后分布的柱状图可以看出，两个类别在投影轴上仍有重叠区域。右图则显示了使用费雪线性判别所得到的投影方向，它不仅最大化了两类均值之间的距离，同时尽可能减小了类内散度，从而显著提升了投影后的类别可分性。

而 $\mathbf{S}$ 称为类内散度矩阵，其定义如下：

$$\mathbf{S} = \sum_{\boldsymbol{x}_i \in \mathcal{C}_1} (\boldsymbol{x}_i - \boldsymbol{\mu}_1)(\boldsymbol{x}_i - \boldsymbol{\mu}_1)^\top + \sum_{\boldsymbol{x}_i \in \mathcal{C}_2} (\boldsymbol{x}_i - \boldsymbol{\mu}_2)(\boldsymbol{x}_i - \boldsymbol{\mu}_2)^\top \tag{3.73}$$

其中 $\boldsymbol{\mu}_1$ 和 $\boldsymbol{\mu}_2$ 分别表示两个类别在原始特征空间中的均值向量。因此，根据费希尔准则构建的目标函数，也可以理解为投影方向上类间散度与类内散度的比值。换句话说，我们希望找到一个合适的投影方向，使 D 维输入数据在投影后，同一类别的样本尽可能集中，而不同类别的样本尽可能分离，从而提高不同类别之间的可区分性。

从公式3.71可以看出，我们关注的是该比值的大小。因此，在求解权重向量 $\boldsymbol{w}$ 时，我们可以通过固定分母 $\boldsymbol{w}^\top \mathbf{S} \boldsymbol{w} = 1$ 来简化优化过程。在添加这种约束下，优化问题可以表示为：

$$\begin{aligned} &\max_{\boldsymbol{w}} \boldsymbol{w}^\top \mathbf{B} \boldsymbol{w} \\ &\text{s.t.} \quad \boldsymbol{w}^\top \mathbf{S} \boldsymbol{w} = 1 \end{aligned} \tag{3.74}$$

为了求解以上约束优化问题，我们引入拉格朗日乘子 λ，从而将其转化为以下等价的拉格朗日优化形式：

$$\boldsymbol{w}^\top \mathbf{B} \boldsymbol{w} - \lambda(\boldsymbol{w}^\top \mathbf{S} \boldsymbol{w} - 1) \tag{3.75}$$

将上式对 $\boldsymbol{w}$ 求导，并令导数为零，可得：

$$2\mathbf{B}\boldsymbol{w} - 2\lambda \mathbf{S}\boldsymbol{w} = 0 \tag{3.76}$$

由此可以推导出以下被称为广义特征值问题（Generalized Eigenvalue Problem）的方程：

$$\mathbf{B}\boldsymbol{w} = \lambda\mathbf{S}\boldsymbol{w} \tag{3.77}$$

以下推导假设 $\mathbf{S}$ 是对称正定矩阵（若不满足正定条件，可采用降维或正则化，例如使用 $\mathbf{S}+\lambda\mathbf{I}$ 形式），因此可以采用科列斯基（Cholesky）分解将其表示为：

$$\mathbf{S} = \mathbf{L}\mathbf{L}^{\top} \tag{3.78}$$

其中 $\mathbf{L}$ 是一个下三角矩阵，且其主对角线上的元素均为正值。令 $\boldsymbol{v} = \mathbf{L}^{\top}\boldsymbol{w}$，则有 $\boldsymbol{w} = (\mathbf{L}^{\top})^{-1}\boldsymbol{v}$，将其代入广义特征值问题公式3.77，可得：

$$\mathbf{B}(\mathbf{L}^{\top})^{-1}\boldsymbol{v} = \lambda\mathbf{S}(\mathbf{L}^{\top})^{-1}\boldsymbol{v} \tag{3.79}$$

上式两边同时左乘 $\mathbf{L}^{-1}$，我们有：

$$\mathbf{L}^{-1}\mathbf{B}(\mathbf{L}^{\top})^{-1}\boldsymbol{v} = \lambda\mathbf{L}^{-1}\mathbf{S}(\mathbf{L}^{\top})^{-1}\boldsymbol{v} \tag{3.80}$$

由于 $\mathbf{S} = \mathbf{L}\mathbf{L}^{\top}$，我们可以得到：

$$\mathbf{L}^{-1}\mathbf{B}(\mathbf{L}^{\top})^{-1}\boldsymbol{v} = \lambda\mathbf{I}\boldsymbol{v} \tag{3.81}$$

令 $\mathbf{C} = \mathbf{L}^{-1}\mathbf{B}(\mathbf{L}^{\top})^{-1}$，则以上方程可以转化为以下标准的特征值问题：

$$\mathbf{C}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{3.82}$$

其中 λ 是特征值，$\boldsymbol{v}$ 为对应的特征向量。求得特征向量 $\boldsymbol{v}$ 之后，可以用等式 $\boldsymbol{w} = (\mathbf{L}^{\top})^{-1}\boldsymbol{v}$ 求解出原问题的权重向量 $\boldsymbol{w}$。从公式3.72可知，矩阵 $\mathbf{B}$ 由两个向量的外积构成，因此其秩为 1。由于左右乘可逆矩阵不改变矩阵的秩，矩阵 $\mathbf{C}$ 的秩仍为 1，因此公式3.82最多只有一个非零特征值。

根据公式3.74所定义的优化问题所求得的权重向量 $\boldsymbol{w}$，仅用于确定输入样本 $\boldsymbol{x}$ 的投影方向及其在该方向上的投影值，并不能直接完成分类任务。在实际进行分类时，通常假设各类别在该投影方向上的投影值服从高斯分布，其均值和方差分别可通过公式3.68和3.69计算得到。在此高斯分布假设下，设某一样本 $\boldsymbol{x}$ 的投影值为 $y = \boldsymbol{x}^{\top}\boldsymbol{w}$，则其属于类别 $\mathcal{C}_i$ 的概率可以表示为：

$$p(\mathcal{C}_i|y) = \frac{p(y|\mathcal{C}_i)p(\mathcal{C}_i)}{p(y)}, \quad i \in \{1,2\} \tag{3.83}$$

若 $p(\mathcal{C}_1|y) > p(\mathcal{C}_2|y)$，则我们将样本 $\boldsymbol{x}$ 归为类别 $\mathcal{C}_1$；否则，归为类别 $\mathcal{C}_2$。在上式中，分母 $p(y)$ 对所有类别是相同的常数项，因此在比较概率大小时可以忽略。类别 $\mathcal{C}_i$ 的先验概率 $p(\mathcal{C}_i)$ 通常可以采用以下公式进行估计：

$$p(\mathcal{C}_1) = \frac{N_1}{N_1+N_2}, \quad p(\mathcal{C}_2) = \frac{N_2}{N_1+N_2} \tag{3.84}$$

需要注意的是，如果训练集中各类样本的比例不能准确反映真实应用中的先验分布，或者存在明显的类别样本数量不平衡问题，我们也可以假设所有类别具有相等的先验概率。至于公式3.83中的条件概率 $p(y|\mathcal{C}_i)$，可以通过下式进行计算：

$$p(y|\mathcal{C}_i) = \frac{1}{(2\pi s_i^2)^{1/2}}\exp\left(-\frac{1}{2s_i^2}(y-\mu_i)^2\right), \quad i \in \{1,2\} \tag{3.85}$$

如果需要确定线性判别函数中的阈值（或偏置）w_0，可以令两类的后验概率相等来计算：

$$
\begin{aligned}
p(\mathcal{C}_1|y_0) &= p(\mathcal{C}_2|y_0) && \Rightarrow \\
p(y_0|\mathcal{C}_1)p(\mathcal{C}_1) &= p(y_0|\mathcal{C}_2)p(\mathcal{C}_2) \text{（忽略相同的 } p(y) \text{ 项）} && \Rightarrow \\
\frac{1}{(2\pi s_1^2)^{1/2}} \exp\left(-\frac{1}{2s_1^2}(y_0-\mu_1)^2\right) p(\mathcal{C}_1) &= \frac{1}{(2\pi s_2^2)^{1/2}} \exp\left(-\frac{1}{2s_2^2}(y_0-\mu_2)^2\right) p(\mathcal{C}_2) && \Rightarrow \\
\frac{(y_0-\mu_2)^2}{2s_2^2} - \frac{(y_0-\mu_1)^2}{2s_1^2} &= \ln\left(\frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}\right) + \frac{1}{2}\ln\left(\frac{s_1^2}{s_2^2}\right) \text{（两边取对数并整理）}
\end{aligned}
$$

如果 $s_1 \neq s_2$，则需要求解以上二次方程来确定 y_0 的值。但若假设 $s_1 = s_2 = s$，该方程将大为简化，此时可解得：

$$
y_0 = \frac{\mu_1 + \mu_2}{2} + \frac{s^2}{\mu_1 - \mu_2} \ln\left(\frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)}\right) \tag{3.86}
$$

若进一步假设两类的先验概率相等，则上式右侧的最后一项为零，此时：

$$
y_0 = \frac{\mu_1 + \mu_2}{2} \tag{3.87}
$$

也就是将 y_0 设为两类投影均值的中点。为了与常见的线性判别函数的形式一致，我们令偏置项 $w_0 = -y_0$，则在原始 D 维输入空间中，得到如下形式的线性判别函数：

$$
f(\boldsymbol{x}) = w_0 + \boldsymbol{x}^\top \boldsymbol{w} \tag{3.88}
$$

相应的分类规则为：当 $f(\boldsymbol{x}) \geq 0$ 时，将输入样本 $\boldsymbol{x}$ 判定为类别 $\mathcal{C}_1$；而当 $f(\boldsymbol{x}) < 0$ 时，则判定其属于类别 $\mathcal{C}_2$。

在将费雪线性判别推广至 $K > 2$ 的多分类问题时，通常要求输入空间的维度 D 大于类别数 K。在此情形下，我们通过构造 M 组权重向量 $\boldsymbol{w}_j$，将原始输入 $\boldsymbol{x}$ 映射（或降维）到 M 维子空间，其中每一维的投影由 $y_j = \boldsymbol{x}^\top \boldsymbol{w}_j$ 给出。子空间的维度 M 需满足 $m \leq \min(D, K-1)$，其具体原因将在后续的讨论中解释。我们将所有 M 组权重向量组成一个矩阵 $\mathbf{W}$，其中第 j 列为向量 $\boldsymbol{w}_j$。利用该矩阵 $\mathbf{W}$，可以将输入向量 $\boldsymbol{x}$ 映射到 M 维子空间，其投影表示为：

$$
\boldsymbol{y}^\top = \boldsymbol{x}^\top \mathbf{W} \tag{3.89}
$$

类内散度矩阵的计算方式与二分类情况下的公式3.73类似，其形式为：

$$
\mathbf{S} = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \in \mathcal{C}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top \tag{3.90}
$$

其中 $\boldsymbol{\mu}_k$ 为类别 $\mathcal{C}_k$ 的均值，其计算公式为：

$$
\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_i \in \mathcal{C}_k} \boldsymbol{x}_i \tag{3.91}
$$

上式中的 N_k 表示属于类别 $\mathcal{C}_k$ 的样本数。总体散度矩阵 $\mathbf{T}$ 可以分解为类内散度矩阵 $\mathbf{S}$ 与类间散度矩阵 $\mathbf{B}$ 之和，即：

$$
\mathbf{T} = \mathbf{S} + \mathbf{B} \tag{3.92}
$$

总体散度矩阵 $\mathbf{T}$ 的计算公式如下所示：

$$\mathbf{T} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \tag{3.93}$$

其中 N 表示所有样本的总数，而 $\boldsymbol{\mu}$ 为所有样本的均值，其计算公式如下：

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k \tag{3.94}$$

根据公式3.92，类间散度矩阵等于总体散度矩阵减去类内散度矩阵。结合公式3.90和3.93，类间散度矩阵的计算表达式为：

$$\mathbf{B} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\top \tag{3.95}$$

类似于二分类情况下的公式3.71，费雪准则旨在投影空间中尽可能增大类间散度，同时尽量减小类内散度。基于该准则，其目标函数可表示为：

$$\text{tr}\left(\mathbf{W}^\top \mathbf{B}\mathbf{W} (\mathbf{W}^\top \mathbf{S}\mathbf{W})^{-1}\right) \tag{3.96}$$

通过最大化该目标函数，可以求得权重矩阵 $\mathbf{W}$。对于输入样本 $\boldsymbol{x}$，其在子空间中的投影为 $\boldsymbol{y}^\top = \boldsymbol{x}^\top \mathbf{W}$。我们通常假设各类别在子空间中的投影服从多维高斯分布，并且据此估计各类别样本在投影空间中的均值向量和协方差矩阵。在此基础上，对于给定样本 $\boldsymbol{x}$，可计算其对应投影 $\boldsymbol{y}$ 在各类别下的联合概率 $p(\boldsymbol{y}|\mathcal{C}_k)p(\mathcal{C}_k)$，并将联合概率最大的类别判定为样本 $\boldsymbol{x}$ 的所属类别。

在讨论费雪线性判别的最后部分，我们来解释为什么投影子空间的维度 M 最多为 $K-1$。从公式3.95可知，类间散度矩阵 $\mathbf{B}$ 是 K 个矩阵的加和，其中每个矩阵的秩都为 1，因为它们都是两个向量的外积。此外，根据公式3.94，所有样本的均值 $\boldsymbol{\mu}$ 是各类别均值 $\boldsymbol{\mu}_k$ 的加权平均。这一约束导致 K 个类均值之间并非完全线性独立，从而使得中心化后的类均值向量至多张成 $K-1$ 维空间。因此，类间散度矩阵 $\mathbf{B}$ 的秩最多为 $K-1$，这意味着它至多只有 $K-1$ 个非零特征值。换句话说，尽管原始输入空间是 D 维的，但是仅有最多 $K-1$ 个方向能够反映类别中心之间的差异，超出这 $K-1$ 维的投影分量不再提供更多关于类均值差异的信息。

接下来，我们将介绍线性判别函数参数学习的最后一种方法，即感知机算法。该算法不仅是人工神经网络发展早期的一个里程碑，也在整个模式识别算法的发展历程中占有重要的地位。感知机算法首先利用一组固定的基函数对输入向量 $\boldsymbol{x}$ 进行变换，提取出特征向量 $\boldsymbol{\phi}(\boldsymbol{x})$，然后再基于该特征向量构建一个带阶跃激活函数的线性分类模型，其形式为：

$$y(\boldsymbol{x}) = \sigma(\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w}) \tag{3.97}$$

其中特征向量 $\boldsymbol{\phi}(\boldsymbol{x})$ 通常会包含一个偏置项 $\phi_0(\boldsymbol{x}) = 1$。非线性激活函数 $\sigma(\cdot)$ 是一个阶跃函数（Step Function），其具体形式如下：

$$\sigma(z) = \begin{cases} +1 & \text{若 } z \geq 0 \\ -1 & \text{若 } z < 0 \end{cases} \tag{3.98}$$

对于样本 $\boldsymbol{x}$，当输出 $y(\boldsymbol{x}) = +1$ 时，则将其判定为类别 $\mathcal{C}_1$；当输出为 -1 时，则判定为类别 $\mathcal{C}_2$。

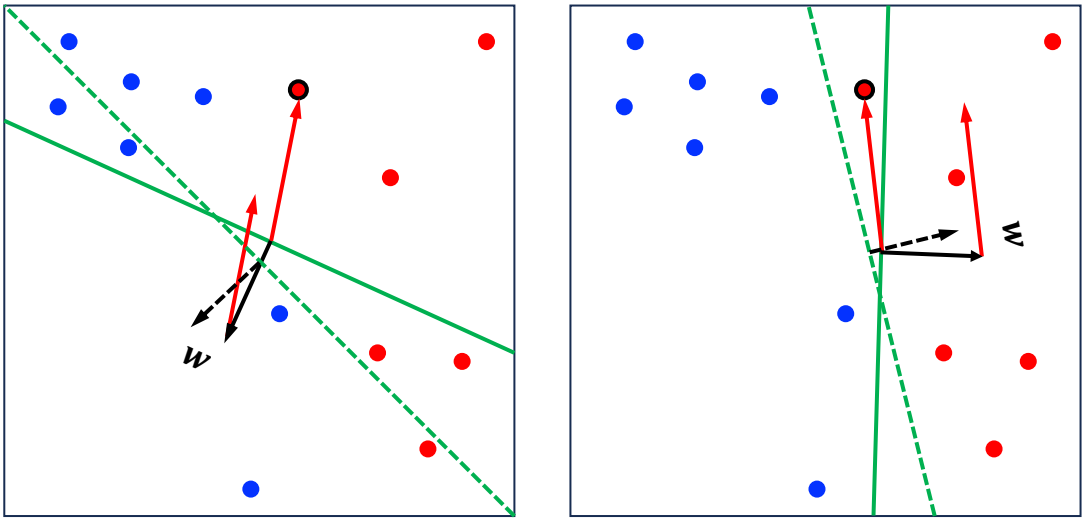


图 3.12: 感知机学习算法权重参数更新的示意图。在二维特征空间中绘制了两个类别的样本点，分别以红色和蓝色表示。左图中初始的权重向量 $\boldsymbol{w}$ 用黑色实线箭头表示，其对应的决策边界以绿色实线表示。箭头的方向指向被分类为红色类别的区域。图中被黑色圆圈标出的红点是一个被错误分类的样本。感知机算法将该样本的特征向量加到当前权重向量上，产生新的权重向量（以黑色虚线箭头表示）及对应的更新后的决策边界（以绿色虚线表示）。右图同样以黑色圆圈标出了仍被错误分类的一个样本。其特征向量再被加到当前权重向量上，得到新的权重向量和相应的决策边界（绿色虚线）。此时，所有样本均被正确分类，表明算法已收敛。

值得注意的是，经典感知机与广义线性模型在数学形式上非常相似，但严格来说，感知机并不属于广义线性模型。这是因为，感知机采用符号函数或阶跃函数直接输出类别标签，并不显式建模类别概率，也不要求输出变量的条件分布属于指数族。而广义线性模型则通过连接函数将输出变量的条件期望与输入特征的线性组合联系起来，并要求其条件分布属于指数族。

由于感知机采用阶跃函数作为非线性激活函数，因此无法直接使用类似公式3.18所示的梯度下降法来更新权重参数 $\boldsymbol{w}$。这是因为当 $z \neq 0$ 时，阶跃函数 $\sigma(z)$ 的输出为常数，其导数为零，无法提供梯度下降所需要的 “梯度信号” 来引导参数的优化过程。为了解决上述问题，弗兰克 · 罗森布拉特（Frank Rosenblatt）提出了感知机准则（Perceptron Criterion），并据此推导出另一种形式的误差函数。根据上述线性模型的定义及其判别规则，我们期望权重参数 $\boldsymbol{w}$ 满足以下条件：当输入样本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_1$ 时，有 $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} > 0$；而当 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_2$ 时，则 $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} < 0$。为了进一步简化损失函数的表达形式，我们将类别 $\mathcal{C}_1$ 对应的目标输出设为 $t = +1$，并将类别 $\mathcal{C}_2$ 对应的目标输出设为 $t = -1$。如此，我们希望所有样本均满足 $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} \cdot t > 0$。感知机准则规定：对于正确分类的样本，其损失为零；而对于错误分类的样本 $\boldsymbol{x}$，其损失为 $-\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} \cdot t$（即最大化 $\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w} \cdot t$）。因此，感知机准则对应的损失函数可以表示为：

$$\mathcal{L}(\boldsymbol{w}) = -\sum_{\boldsymbol{x}_i \in \mathcal{E}} \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{w} \cdot t_i \tag{3.99}$$

其中 $\mathcal{E}$ 表示在当前权重参数下被错误分类的样本集合。由于以上损失函数 $\mathcal{L}(\boldsymbol{w})$ 对于参数 $\boldsymbol{w}$ 是

算法 3.2: 感知机学习算法

输入: 训练样本集（共包含 N 个样本）、学习率 η 和最大训练轮数 H。

初始化: 将权重向量 $\boldsymbol{w}$ 随机初始化或设为零，并设训练轮数 $h=1$ 和迭代步数 $j=1$;
while $h \leq H$ **do**
  for $i \leftarrow 1$ **to** N **do**
    根据公式3.99计算当前样本 $\boldsymbol{x}_i$ 的损失为 $-\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{w}^{(j)} \cdot t_i$;
    if $(\boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{w}^{(j)} \cdot t_i \leq 0)$ **then**
      使用公式3.100更新权重向量 $\boldsymbol{w}^{(j+1)} = \boldsymbol{w}^{(j)} + \eta \boldsymbol{\phi}(\boldsymbol{x}_i) t_i$;
      $j \leftarrow j+1$;
  收敛条件: 当所有样本均分类正确，即本轮中权重向量未发生更新时，算法终止;
  $h \leftarrow h+1$;

输出: 经训练得到的权重向量 $\boldsymbol{w}$。

可导的，我们可以采用次梯度下降法对权重进行迭代更新。相应的更新公式为:

$$\boldsymbol{w}^{(j+1)} = \boldsymbol{w}^{(j)} - \eta \nabla \mathcal{L}_i(\boldsymbol{w}^{(j)}) = \boldsymbol{w}^{(j)} + \eta \boldsymbol{\phi}(\boldsymbol{x}_i) t_i \tag{3.100}$$

其中 $\nabla \mathcal{L}_i(\cdot)$ 表示第 i 个样本损失的次梯度，j 表示迭代步数，而 η 是学习率（或称学习步长）。

从公式3.100可知，当学习率 $\eta=1$ 时，感知机算法具有直观而简洁的解释。如果样本被正确分类，则无需对权重参数进行调整；如果样本被错误分类，则分以下两种情况进行更新：若误分类的样本 $\boldsymbol{x}_i$ 实际属于类别 $\mathcal{C}_1$ 时（$t_i=+1$），则将其对应的特征向量 $\boldsymbol{\phi}(\boldsymbol{x}_i)$ 加到当前权重向量 $\boldsymbol{w}$ 上；若误分类的样本 $\boldsymbol{x}_i$ 实际属于类别 $\mathcal{C}_2$ 时（$t_i=-1$），则将该样本的特征向量 $\boldsymbol{\phi}(\boldsymbol{x}_i)$ 从当前权重向量 $\boldsymbol{w}$ 中减去。示意图3.12展示了感知机算法中权重参数 $\boldsymbol{w}$ 的更新过程。在训练数据线性可分的前提下，只要学习率足够小，每次更新都会使模型的决策边界朝着更优的方向调整。尽管这一过程并不一定会使整体损失函数下降，但至少可以保证当前被误分类样本的损失得到一定程度的减小:

$$-\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w}^{(j+1)} \cdot t = -\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w}^{(j)} \cdot t - \underbrace{\eta \|\boldsymbol{\phi}(\boldsymbol{x}) t\|^2}_{>0} < -\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{w}^{(j)} \cdot t \tag{3.101}$$

当存在能够将两类样本完全分开的决策边界时，使用公式 3.100 的更新规则，在有限次迭代后一定能够构造出一个正确划分所有训练样本的决策边界。此外，由于所有样本的特征向量的模长是有界的，经过有限步的加上或减去有界模长后的权重向量的模也一定是有界的。对于有限且严格线性可分的训练集，感知机算法在固定正学习率下只会发生有限次更新，即算法收敛。

算法3.2给出了感知机学习算法的完整流程。该算法在训练数据线性可分的前提下可以保证收敛。但若训练数据线性不可分，算法则永远无法收敛。在实际应用中，训练一般需要非常多的迭代步数。当算法长时间未能收敛时，我们往往难以判断是因为训练数据集本身线性不可分，还是仅仅由于收敛速度较慢。此外，即使在线性可分条件下收敛，其最终收敛到的权重向量并不唯一，结果会受到权重向量随机初始化和样本遍历顺序的影响。除了学习过程本身存在一定的困难外，感知机学习算法还存在两个主要局限：一是该模型的输出不具备概率解释；二是它在推广到类别数 $K>2$ 的多分类问题时需要额外的处理。

3.6 概率生成模型

构建用于分类的概率生成模型的基本思路是：首先为每一个类别 $\mathcal{C}_k$ 建立类条件概率分布（Class-conditional Distribution）$p(\boldsymbol{x}|\mathcal{C}_k)$，然后结合先验概率 $p(\mathcal{C}_k)$ 运用以下的贝叶斯公式来计算后验概率分布 $p(\mathcal{C}_k|\boldsymbol{x})$，从而进行分类：

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{k=1}^{K} p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)} \tag{3.102}$$

这类方法之所以称为概率生成模型，是因为通过对类条件概率分布 $p(\boldsymbol{x}|\mathcal{C}_k)$ 的建模，在理论上可以从该分布中采样，从而生成类别 $\mathcal{C}_k$ 的合成数据（Synthetic Data）。

我们依然从二分类问题入手展开讨论。样本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_1$ 的后验概率可表示为：

$$\begin{aligned} p(\mathcal{C}_1|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{\frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}}{\frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)} + \frac{p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)}{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}} \\ &= \frac{1}{1 + \exp\left(-\ln \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)}\right)} \\ &= \frac{1}{1 + \exp\left(-z\right)} \qquad \left(z = \ln \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)}\right) \\ &= \sigma(z) \end{aligned} \tag{3.103}$$

其中 $\sigma(z)$ 称为 Sigmoid 函数，其定义如下：

$$\sigma(z) = \frac{1}{1 + \exp\left(-z\right)} \tag{3.104}$$

函数 $\sigma(z)$ 被称为 Sigmoid 函数是因为它的图像呈典型的 S 形曲线（如图3.13所示）。这类函数也被称为“压缩函数”（Squashing Function），因为它能够将整个实数轴的输入挤压到一个有限的区间内。具体而言，Sigmoid 函数将所有实数映射到 $(0, 1)$ 区间，并且当输入小于 -5 或大于 5 时，函数值就已经非常趋近于 0 或 1。

容易验证，Sigmoid 函数满足如下的对称性质：

$$\begin{aligned} \sigma(-z) &= 1 - \sigma(z) \\ &= \frac{\exp\left(-z\right)}{1 + \exp\left(-z\right)} \\ &= \frac{1}{1 + \exp\left(z\right)} \end{aligned} \tag{3.105}$$

此外，对函数 $\sigma(z)$ 关于变量 z 求导，可以得到如下简洁的形式：

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \tag{3.106}$$

这使得在需要使用其导数的情形下，带来了计算上的便利。上式的推导过程如下：

$$\begin{aligned}\sigma'(z) &= \frac{\exp(-z)}{(1+\exp(-z))^2} \\ &= \frac{1}{1+\exp(-z)} \cdot \frac{\exp(-z)}{1+\exp(-z)} \\ &= \sigma(z)\frac{1+\exp(-z)-1}{1+\exp(-z)} \\ &= \sigma(z)\left(1-\frac{1}{1+\exp(-z)}\right) \\ &= \sigma(z)(1-\sigma(z))\end{aligned}$$

Sigmoid 函数的反函数的形式如下：

$$z = \ln\left(\frac{\sigma}{1-\sigma}\right) \tag{3.107}$$

该函数被称为对数几率函数（Logit Function），也常被称为对数几率（Log Odds）。它表示样本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_1$ 与 $\mathcal{C}_2$ 的后验概率之比的对数形式，即 $\ln\left(\frac{p(\mathcal{C}_1|\boldsymbol{x})}{p(\mathcal{C}_2|\boldsymbol{x})}\right) = \ln\left(\frac{p(\mathcal{C}_1|\boldsymbol{x})}{1-p(\mathcal{C}_1|\boldsymbol{x})}\right)$。我们很快将看到，$z(\boldsymbol{x})$ 可以取 $\boldsymbol{x}$ 的线性形式，从而使 $\sigma(z)$ 构成一个形如公式3.58的广义线性模型。值得注意的是，概率生成模型通常不属于广义线性模型，因为它主要通过建立数据的类条件概率分布和类别先验分布来描述数据的生成过程，而广义线性模型则通过连接函数，将输出变量的条件期望与输入特征的线性组合联系起来。不过，在特定条件下，概率生成模型推导出的后验条件概率可以具有广义线性模型的形式。例如，在下面的推导中，当二分类高斯生成模型中的两个类别共享相同的协方差矩阵时，其后验概率可以表示为 Sigmoid 函数（即逆连接函数）作用于输入特征的线性组合，并且与 Logistic 回归具有相同的函数形式。

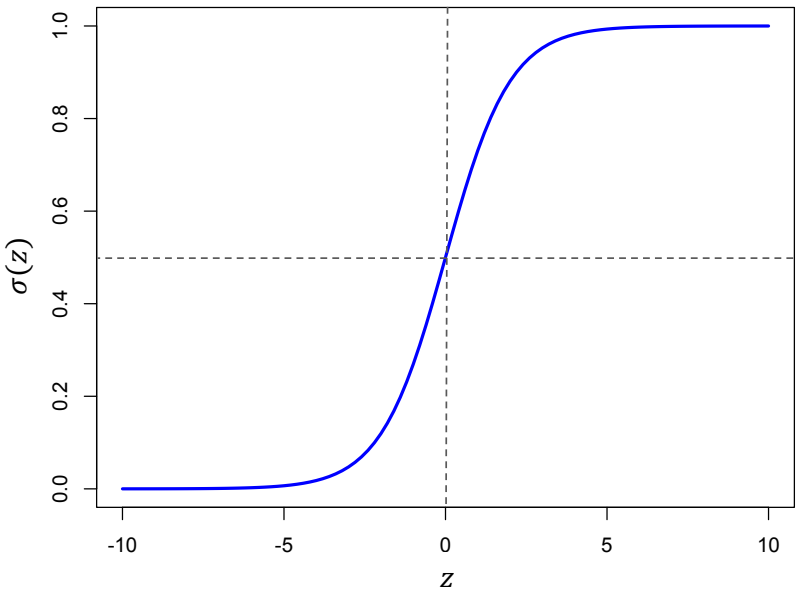


图 3.13: Sigmoid 函数的曲线图。

从公式3.102可见，构建概率生成模型的关键在于合理选择类条件概率分布的形式。常见的做法是假设类条件概率服从高斯分布，并且所有类别均共享同一个协方差矩阵。在这种情况下，类条件概率密度函数可以表示为：

$$p(\boldsymbol{x}|\mathcal{C}_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right) \tag{3.108}$$

其中 D 表示输入样本 $\boldsymbol{x}$ 的维度，$\boldsymbol{\mu}_k$ 是类别 $\mathcal{C}_k$ 对应的均值向量，$\boldsymbol{\Sigma}$ 则是不同类别共享的 $D \times D$ 的协方差矩阵。$|\boldsymbol{\Sigma}|$ 表示矩阵 $\boldsymbol{\Sigma}$ 的行列式，是一个标量。为求得公式3.103中的后验概率，我们先计算该公式推导过程中引入的中间变量 z，其形式如下：

$$z = \ln \frac{p(\boldsymbol{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\boldsymbol{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{3.109}$$

将类条件概率密度函数（公式3.108）代入上式，我们可得：

$$\begin{aligned} z &= -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \\ &= \boldsymbol{x}^\top \underbrace{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}_{\boldsymbol{w}} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)}}_{w_0} \\ &= w_0 + \boldsymbol{x}^\top \boldsymbol{w} \end{aligned} \tag{3.110}$$

从上述推导可以看出，由于假设所有类别共享相同的协方差矩阵，高斯分布中指数项关于 $\boldsymbol{x}$ 的二次项相互抵消。因此，样本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_1$ 的后验概率可以表示为如下形式：

$$p(\mathcal{C}_1|\boldsymbol{x}) = \sigma(w_0 + \boldsymbol{x}^\top \boldsymbol{w}) \tag{3.111}$$

其中：

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \tag{3.112}$$

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{3.113}$$

值得注意的是：如果不同类别的协方差矩阵并不相同，则会保留关于 $\boldsymbol{x}$ 的二次项，此时模型将不再是 $\boldsymbol{x}$ 的线性形式。

确定了类条件概率分布的形式之后，我们可以采用最大似然估计来求解模型中的各项参数。给定包含 N 个样本的训练数据集 $\mathcal{D} = \{(\boldsymbol{x}_i, t_i)\}_{i=1}^N$，并且规定：若样本 $\boldsymbol{x}_i$ 属于类别 $\mathcal{C}_1$，则对应的标签 $t_i = 1$；若属于类别 $\mathcal{C}_2$，则 $t_i = 0$。为简化表示，我们将类别 $\mathcal{C}_1$ 的先验概率 $p(\mathcal{C}_1)$ 记作 π，则类别 $\mathcal{C}_2$ 的先验概率 $p(\mathcal{C}_2)$ 就是 $1-\pi$。对于属于类别 $\mathcal{C}_1$ 的样本 $\boldsymbol{x}_i$，其联合概率 $p(\boldsymbol{x}_i, \mathcal{C}_1)$ 可以表示为：

$$p(\boldsymbol{x}_i, \mathcal{C}_1) = p(\mathcal{C}_1)p(\boldsymbol{x}_i|\mathcal{C}_1) = \pi\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \tag{3.114}$$

同理，对于属于类别 $\mathcal{C}_2$ 的样本 $\boldsymbol{x}_i$，其联合概率为：

$$p(\boldsymbol{x}_i, \mathcal{C}_2) = p(\mathcal{C}_2)p(\boldsymbol{x}_i|\mathcal{C}_2) = (1-\pi)\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \tag{3.115}$$

如此，相应的似然函数可表示为：

$$p(\mathbf{X}, \boldsymbol{t}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_{i=1}^{N} \left(\pi \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})\right)^{t_i} \left((1-\pi)\mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\right)^{1-t_i} \tag{3.116}$$

其中 $\mathbf{X}$ 是 $N \times D$ 的矩阵，表示整个训练数据集，而 $\boldsymbol{t} = (t_1, \ldots, t_N)^\top$ 表示所有训练样本的分类标签向量。在回归问题中，似然函数表示在给定参数条件下生成观测数据本身的概率；而在分类问题中，我们需要同时考虑样本特征 $\boldsymbol{x}$ 及其对应的分类标签 t，因此其似然函数应为给定参数条件下生成样本及其对应标签的联合概率。与之前一样，为了简化参数求解过程，我们通常最大化对数似然函数：

$$\sum_{i=1}^{N} \{t_i \ln \pi + t_i \ln \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) + (1-t_i) \ln (1-\pi) + (1-t_i) \ln \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\} \tag{3.117}$$

我们逐一对各个参数进行估计。首先从公式3.117中提取仅与参数 π 相关的部分：

$$\sum_{i=1}^{N} \{t_i \ln \pi + (1-t_i) \ln (1-\pi)\} \tag{3.118}$$

对上式关于参数 π 求导，并令导数为零，我们可得：

$$\begin{aligned} \sum_{i=1}^{N} \left(\frac{t_i}{\pi} - \frac{1-t_i}{1-\pi}\right) = 0 \quad &\Rightarrow \\ \frac{N_1}{\pi} - \frac{N_2}{1-\pi} = 0 \quad &\Rightarrow \\ \pi = \frac{N_1}{N_1 + N_2} \end{aligned}$$

其中 N_1 表示属于类别 $\mathcal{C}_1$ 的样本数量，而 N_2 表示属于类别 $\mathcal{C}_2$ 的样本数量。因此，参数 π 的最大似然估计等于类别 $\mathcal{C}_1$ 在所有样本中所占的比例。

接着，我们来估计参数 $\boldsymbol{\mu}_1$。同样地，从似然函数公式3.117中提取仅与参数 $\boldsymbol{\mu}_1$ 相关的项：

$$\sum_{i=1}^{N} t_i \ln \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \left(-\frac{1}{2} t_i (\boldsymbol{x}_i - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_1) + t_i \ln \left(\frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}|^{1/2}}\right)\right) \tag{3.119}$$

将上式对参数 $\boldsymbol{\mu}_1$ 求导，并令导数为零，可得：

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{i=1}^{N} t_i \boldsymbol{x}_i \tag{3.120}$$

上式表明参数 $\boldsymbol{\mu}_1$ 的最大似然估计为所有属于类别 $\mathcal{C}_1$ 样本的均值。同理，参数 $\boldsymbol{\mu}_2$ 的最大似然估计为：

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{i=1}^{N} (1-t_i) \boldsymbol{x}_i \tag{3.121}$$

即为所有属于类别 $\mathcal{C}_2$ 样本的平均值。

最后，我们使用最大似然估计来求解协方差矩阵 $\boldsymbol{\Sigma}$ 的参数。从对数似然函数（公式3.117）

中提取出与 $\boldsymbol{\Sigma}$ 有关的项，可以得到：

$$\begin{aligned}
&\sum_{i=1}^{N}\{t_i \ln \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_1,\boldsymbol{\Sigma}) + (1-t_i)\ln \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_2,\boldsymbol{\Sigma})\} \\
=&-\frac{1}{2}\sum_{i=1}^{N} t_i \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{N} t_i(\boldsymbol{x}_i-\boldsymbol{\mu}_1)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_1) \\
&-\frac{1}{2}\sum_{i=1}^{N}(1-t_i)\ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{N}(1-t_i)(\boldsymbol{x}_i-\boldsymbol{\mu}_2)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}_2) + \text{const}
\end{aligned}$$

其中 const 是与 $\boldsymbol{\Sigma}$ 无关的常数项。上式可以进一步简化为以下形式：

$$-\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}) \tag{3.122}$$

其中：

$$\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2 \tag{3.123}$$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{\boldsymbol{x}_i\in\mathcal{C}_1}(\boldsymbol{x}_i-\boldsymbol{\mu}_1)(\boldsymbol{x}_i-\boldsymbol{\mu}_1)^\top \tag{3.124}$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{\boldsymbol{x}_i\in\mathcal{C}_2}(\boldsymbol{x}_i-\boldsymbol{\mu}_2)(\boldsymbol{x}_i-\boldsymbol{\mu}_2)^\top \tag{3.125}$$

我们对公式3.122关于协方差矩阵 $\boldsymbol{\Sigma}$ 求导（对矩阵求导是指对矩阵中所有元素分别求偏导，得到一个由各偏导数组成的“梯度矩阵”），并令导数为零，我们可以得到：

$$\begin{aligned}
-\frac{N}{2}\boldsymbol{\Sigma}^{-1} + \frac{N}{2}\boldsymbol{\Sigma}^{-1}\mathbf{S}\boldsymbol{\Sigma}^{-1} = 0 \quad &\Rightarrow \\
\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}^{-1}\mathbf{S}\boldsymbol{\Sigma}^{-1} \quad &\Rightarrow \\
\mathbf{S}\boldsymbol{\Sigma}^{-1} = \mathbf{I} \quad &\Rightarrow \\
\boldsymbol{\Sigma} = \mathbf{S}&
\end{aligned}$$

协方差矩阵 $\boldsymbol{\Sigma}$ 的最大似然估计即为矩阵 $\mathbf{S}$。由公式3.123可知，矩阵 $\mathbf{S}$ 是两个类别样本协方差矩阵的加权形式。

上述方法通过对输入 $\boldsymbol{x}$ 进行线性变换，再将结果传入 Sigmoid 函数，从而得到 $\boldsymbol{x}$ 属于某个类别的后验概率。值得注意的是，虽然该高斯生成模型导出的后验概率具有与 Logistic 回归相同的函数形式，但两者的建模方式和参数估计方法有所不同。前者通过建立类条件概率分布和类别先验分布来计算后验概率，而后者则直接对条件概率进行建模和参数估计，我们将在下一节中详细介绍。接下来，我们将上述二分类概率生成模型扩展到 $K > 2$ 的多分类情形。此时样

本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_k$ 的后验概率可表示为：

$$\begin{aligned} p(\mathcal{C}_k|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{j=1}^{K} p(\boldsymbol{x}|\mathcal{C}_j)p(\mathcal{C}_j)} \\ &= \frac{\exp\left(\ln\left(p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)\right)\right)}{\sum_{j=1}^{K} \exp\left(\ln\left(p(\boldsymbol{x}|\mathcal{C}_j)p(\mathcal{C}_j)\right)\right)} \\ &= \frac{\exp\left(z_k\right)}{\sum_{j=1}^{K} \exp\left(z_j\right)} \end{aligned} \tag{3.126}$$

其中：

$$z_k = \ln\left(p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)\right) \tag{3.127}$$

公式3.126被称为归一化指数函数（Normalized Exponential），它可以看作 Sigmoid 函数在多类别情形下的推广。该归一化指数函数也被称为 **Softmax 函数**，因为它可以视为“最大值”函数的一种平滑近似。具体而言，当某个 z_k 远大于所有其他 z_j $(j \neq k)$ 时，有 $p(\mathcal{C}_k|\boldsymbol{x}) \simeq 1$，而 $p(\mathcal{C}_j|\boldsymbol{x}) \simeq 0$（符号 $\simeq$ 表示近似相等）。对于输入样本 $\boldsymbol{x}$，Softmax 函数会为每个类别分配一个介于 0 和 1 之间的概率值，这些概率的总和为 1。它不会将所有概率集中在一个类别上，所以被称为“软”（Soft）的分类结果，从而有别于“硬”（Hard）的分类结果（仅有一个类别的概率为 1，其余类别为 0）。

类似于二分类情形，我们假设每个类别的类条件概率分布 $p(\boldsymbol{x}|\mathcal{C}_k)$ 均服从高斯分布，且所有类别共享同一个协方差矩阵 $\boldsymbol{\Sigma}$。在这一假设下，将高斯形式的类条件概率代入公式3.126和3.127，可以推导出每个类别对应的 z_k 也呈如下的线性形式：

$$z_k = w_{k0} + \boldsymbol{x}^\top \boldsymbol{w}_k \tag{3.128}$$

其中：

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k) \tag{3.129}$$

$$\boldsymbol{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \tag{3.130}$$

与二分类的情形类似，在假设所有类别共享同一个协方差矩阵的情况下，计算 z_k 时关于 $\boldsymbol{x}$ 的二次项会相互抵消，从而使 z_k 成为输入 $\boldsymbol{x}$ 的线性函数。如果我们放宽这一假设，允许每个类条件概率分布 $p(\boldsymbol{x}|C_k)$ 拥有各自不同的协方差矩阵 Σ_k，那么上述抵消将不再发生，此时得到的 z_k 将是关于 $\boldsymbol{x}$ 的二次函数。由于本章主要讨论线性分类模型，在此不对二次判别函数（Quadratic Discriminant）的具体形式和性质作进一步展开讨论。

采用最大似然估计，多分类概率生成模型的参数可通过以下公式进行估算：

$$p(\mathcal{C}_k) = \frac{N_k}{N} \tag{3.131}$$

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{\boldsymbol{x}_i \in \mathcal{C}_k} \boldsymbol{x}_i \tag{3.132}$$

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{k=1}^{K}\left(\sum_{\boldsymbol{x}_i \in \mathcal{C}_k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\top\right) \tag{3.133}$$

其中 N_k 表示属于类别 $\mathcal{C}_k$ 的样本数，N 则表示所有样本的总数。在本节的最后，有必要指出：

由于高斯分布的概率密度大部分集中在中心区域，而在其两端分配了相对较少的概率，这使得高斯分布在拟合包含异常值的数据时表现出较差的鲁棒性（详见第2.5节的相关讨论）。因此，基于高斯分布的最大似然估计对异常值也较为敏感。

3.7 概率判别模型

在概率生成模型中，我们首先需要为类条件概率分布选择合适的形式，并结合类别的先验分布，通过贝叶斯公式推导出后验分布，从而估计样本属于各类别的概率。以输入特征维度为 D 维，并选用高斯分布作为类条件概率分布为例。如果有 K 个类别，则概率生成模型需要估计以下参数：每个类别的均值，共 KD 个参数；各类别共享的对称协方差矩阵，共需 $D(D+1)/2$ 个参数；各类别的先验概率，共 K 个参数（考虑到概率归一性，通常实际为 $K-1$ 个参数）。总的参数量为 $\frac{1}{2}(D^2+(2K+1)D+2(K-1))$。由此可见，随着特征维度 D 的增加，所需估计的参数数量呈二次增长。相比之下，如果我们能够直接确定如公式3.128所示的广义线性模型中权重参数，则仅需估计 $K(D+1)$ 个参数（包含偏置项 w_{k0}）。当特征维度或类别数较大时，概率判别模型因其参数更少而具有明显的优势。

概率生成模型和概率判别模型主要区别在于：前者先对每个类别的均值、协方差和先验分布进行估计，再通过这些统计量间接计算线性模型的权重参数（公式3.110显示了模型参数之间的关系），而后者则跳过中间过程，直接对线性模型的权重参数进行学习。

接下来，我们仍以二分类问题为例，探讨如何采用概率判别模型的思路，直接对类别条件概率进行建模，并通过训练数据估计模型的权重参数。具体而言，我们采用 Sigmoid 函数将输入特征的线性组合映射为类别概率。这种方法称为 **Logistic 回归**（Logistic Regression）。尽管方法名字中带有“回归”，但它主要用于分类任务，尤其是二分类问题。给定包含 N 个样本的训练数据集 $\mathcal{D}=\{(\tilde{\boldsymbol{x}}_i,t_i)\}_{i=1}^N$，其中 $\tilde{\boldsymbol{x}}$ 是在原输入向量 $\boldsymbol{x}$ 的前面添加一个值为 1 的元素（对应于偏置项）之后的增广向量，即 $\tilde{\boldsymbol{x}}=(1,\boldsymbol{x}^\top)^\top$。我们继续沿用之前的标签编码方式：若样本 $\tilde{\boldsymbol{x}}_i$ 属于类别 $\mathcal{C}_1$，则对应的标签 $t_i=1$；若属于类别 $\mathcal{C}_2$，则 $t_i=0$。在此设定下，模型的似然函数可以表示为：

$$p(\boldsymbol{t}|\tilde{\boldsymbol{w}})=\prod_{i=1}^{N}y_i^{t_i}(1-y_i)^{(1-t_i)} \tag{3.134}$$

其中 $\boldsymbol{t}=(t_1,\dots,t_N)^\top$，$y_i=p(\mathcal{C}_1|\tilde{\boldsymbol{x}}_i)=\sigma(\tilde{\boldsymbol{x}}_i^\top\tilde{\boldsymbol{w}})$，而 $\tilde{\boldsymbol{w}}$ 是包含了偏置项 w_0 后的增广权重向量。对上述似然函数取对数并加上负号后，可将其转化为如下形式的损失函数 (最小化该损失函数等价于最大化原似然函数)：

$$\mathcal{L}(\tilde{\boldsymbol{w}})=-\ln p(\boldsymbol{t}|\tilde{\boldsymbol{w}})=-\sum_{i=1}^{N}(t_i\ln y_i+(1-t_i)\ln(1-y_i)) \tag{3.135}$$

上式也被称为**交叉熵**（Cross-entropy）损失，在分类任务中被广泛采用，它能够度量真实分布与预测分布之间差异。此处，对于输入样本 $\tilde{\boldsymbol{x}}_i$，我们将 $(t_i,1-t_i)^\top$ 看成真实分布，而将 $(y_i,1-y_i)^\top$

视为模型的预测分布。我们通过调整参数向量 $\tilde{\boldsymbol{w}}$ 来最小化真实分布与预测分布之间的差异，从而提升模型的分类性能。下面我们简要说明为何最小化交叉熵可以减少两个分布之间的差异。

在信息论中，两个离散概率分布 p 和 q 之间的交叉熵定义如下：

$$\mathrm{H}[p,q] = -\sum_{x} p(x)\ln q(x) \tag{3.136}$$

通过如下推导，交叉熵 $\mathrm{H}[p,q]$ 可以被表示为分布 p 的熵 $\mathrm{H}[p]$ 与分布 p 和 q 之间 KL 散度之和：

$$\begin{aligned}
\mathrm{H}[p,q] &= -\sum_{x} p(x)\ln q(x) \\
&= -\sum_{x} p(x)\ln\left(\frac{q(x)p(x)}{p(x)}\right) \\
&= -\sum_{x} p(x)\left(\ln p(x) + \ln\frac{q(x)}{p(x)}\right) \\
&= -\sum_{x} p(x)\ln p(x) - \sum_{x} p(x)\ln\frac{q(x)}{p(x)} \\
&= \mathrm{H}[p] + D_{\mathrm{KL}}(p\|q)
\end{aligned}$$

如第2.4节中所述，KL 散度可以用于衡量两个概率分布之间的差异程度。由于分布 p 的熵 $\mathrm{H}[p]$ 在优化过程中是常数，因此在最小化交叉熵时可以忽略不计。因此，最小化交叉熵实际上等价于最小化分布 p 和 q 之间的 KL 散度，本质上就是在最小化真实分布和预测分布之间的差异。

对损失函数（公式3.135）关于参数 $\tilde{\boldsymbol{w}}$ 求导数，我们得到：

$$\begin{aligned}
\nabla\mathcal{L}(\tilde{\boldsymbol{w}}) &= -\sum_{i=1}^{N}(t_i(1-y_i)\tilde{\boldsymbol{x}}_i - y_i(1-t_i)\tilde{\boldsymbol{x}}_i) \\
&= \sum_{i=1}^{N}(t_i y_i\tilde{\boldsymbol{x}}_i - t_i\tilde{\boldsymbol{x}}_i + y_i\tilde{\boldsymbol{x}}_i - t_i y_i\tilde{\boldsymbol{x}}_i) \\
&= \sum_{i=1}^{N}(y_i - t_i)\tilde{\boldsymbol{x}}_i
\end{aligned} \tag{3.137}$$

从公式3.137可以看出，每个样本对损失函数梯度的贡献等于误差（即模型预测值 y_i 与目标值 t_i 之差）与该样本的特征向量的乘积。据此，我们可以设计一个序贯学习算法来逐步更新参数 $\tilde{\boldsymbol{w}}$：

$$\tilde{\boldsymbol{w}}^{(j+1)} = \tilde{\boldsymbol{w}}^{(j)} - \eta\nabla\mathcal{L}_i(\tilde{\boldsymbol{w}}^{(j)}) = \tilde{\boldsymbol{w}}^{(j)} - \eta(y_i - t_i)\tilde{\boldsymbol{x}}_i \tag{3.138}$$

其中 $\mathcal{L}_i(\cdot)$ 表示第 i 个样本的损失，j 表示迭代的步数，而 η 表示学习率（或称学习步长）。

由公式3.138可知，参数 $\tilde{\boldsymbol{w}}$ 的更新依据当前一个样本（或者一批样本）对损失函数下降方向的一阶估计。只要模型的预测值 y 与目标值 t 存在偏差，就会对权重参数 $\tilde{\boldsymbol{w}}$ 进行相应的调整。这种更新策略的趋势是：对于属于类别 $\mathcal{C}_1$ 的样本 $\tilde{\boldsymbol{x}}$，通过让 $\tilde{\boldsymbol{x}}^{\top}\tilde{\boldsymbol{w}}$ 的值持续增大，使其预测值逐渐趋近 1；而对于属于类别 $\mathcal{C}_2$ 的样本，则让 $\tilde{\boldsymbol{x}}^{\top}\tilde{\boldsymbol{w}}$ 不断减小，使其预测值趋近于 0。如果训练集是线性可分的，那么一定存在某个方向 $\tilde{\boldsymbol{w}}$ 可以将两类样本完全分开。我们只要在这个方向上不断放大 $\tilde{\boldsymbol{w}}$ 的模长（即乘上一个正因子），就可以使属于 $\mathcal{C}_1$ 的样本对应的 $\tilde{\boldsymbol{x}}^{\top}\tilde{\boldsymbol{w}}$ 更大，而属于

$\mathcal{C}_2$ 样本的 $\tilde{\boldsymbol{x}}^\top\tilde{\boldsymbol{w}}$ 更小，从而使 Sigmoid 函数的输出更趋近于 1 或 0（但永远不可能等于 1 或 0）。在此过程中，负对数似然损失函数（公式3.135）会持续下降。在没有对参数 $\tilde{\boldsymbol{w}}$ 施加正则化约束的情况下，参数向量 $\tilde{\boldsymbol{w}}$ 的范数（或模长）会趋于无穷大，从而导致严重的过拟合问题。

通过对损失函数（公式3.135）的分析可知，它是一个凸函数。这意味着该函数存在最小值，并且是唯一的 [13]。因此，我们有必要进一步探寻更好的优化算法，以确保训练过程能够稳定地收敛到全局最优解。接下来的内容将先证明该损失函数的凸性，然后再讨论相应的优化算法。

我们在参数 $\tilde{\boldsymbol{w}}$ 处，对公式3.135所定义的损失函数进行二阶泰勒展开，可得：

$$\mathcal{L}(\tilde{\boldsymbol{w}}+\boldsymbol{h}) \approx \mathcal{L}(\tilde{\boldsymbol{w}}) + \nabla\mathcal{L}(\tilde{\boldsymbol{w}})^\top\boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^\top\mathbf{H}\boldsymbol{h} \tag{3.139}$$

其中：

$$\mathbf{H} = \nabla^2\mathcal{L}(\tilde{\boldsymbol{w}}) = \sum_{i=1}^{N} y_i(1-y_i)\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i^\top = \tilde{\mathbf{X}}^\top\mathbf{R}\tilde{\mathbf{X}} \tag{3.140}$$

上式中的 $\mathbf{H}$ 被称为海森矩阵（Hessian Matrix），它是由损失函数关于参数 $\tilde{\boldsymbol{w}}$ 的所有二阶偏导数组成的一个方阵。$\tilde{\mathbf{X}}$ 是一个维度为 $N\times(D+1)$ 的矩阵，其第 i 行为 $\tilde{\boldsymbol{x}}_i^\top$。$\mathbf{R}$ 是一个 $N\times N$ 的对角矩阵，其第 i 个对角线元素为 $\mathbf{R}_{ii} = y_i(1-y_i)$。对于任意非零的 $D+1$ 维向量 $\boldsymbol{v}$，我们有：

$$\boldsymbol{v}^\top\mathbf{H}\boldsymbol{v} = (\tilde{\mathbf{X}}\boldsymbol{v})^\top\mathbf{R}(\tilde{\mathbf{X}}\boldsymbol{v}) = \boldsymbol{u}^\top\mathbf{R}\boldsymbol{u} = \sum_{i=1}^{N} y_i(1-y_i)u_i^2 \tag{3.141}$$

其中 $\boldsymbol{u} = \tilde{\mathbf{X}}\boldsymbol{v}$。由于所有 y_i 均满足 $0<y_i<1$，从上式可知 $\boldsymbol{v}^\top\mathbf{H}\boldsymbol{v}>0$，说明 $\mathbf{H}$ 是一正定矩阵。由此可知，公式3.139右侧的最后一项恒为正，因而可以推出：

$$\mathcal{L}(\tilde{\boldsymbol{w}}+\boldsymbol{h}) \geq \mathcal{L}(\tilde{\boldsymbol{w}}) + \nabla\mathcal{L}(\tilde{\boldsymbol{w}})^\top\boldsymbol{h} \tag{3.142}$$

将上式中的 $\tilde{\boldsymbol{w}}$ 记作 $\tilde{\boldsymbol{w}}_a$，并将 $\tilde{\boldsymbol{w}}+\boldsymbol{h}$ 记作 $\tilde{\boldsymbol{w}}_b$，则上式可以改写为：

$$\mathcal{L}(\tilde{\boldsymbol{w}}_b) \geq \mathcal{L}(\tilde{\boldsymbol{w}}_a) + \nabla\mathcal{L}(\tilde{\boldsymbol{w}}_a)^\top(\tilde{\boldsymbol{w}}_b - \tilde{\boldsymbol{w}}_a) \tag{3.143}$$

以上公式称为凸函数的一阶条件。只要函数满足这一条件，就可以证明它是凸的。为此，我们任取 $0\leq\lambda\leq1$，并令 $\tilde{\boldsymbol{w}}_c = \lambda\tilde{\boldsymbol{w}}_a + (1-\lambda)\tilde{\boldsymbol{w}}_b$。应用公式3.143两次，可以得到：

$$\mathcal{L}(\tilde{\boldsymbol{w}}_a) \geq \mathcal{L}(\tilde{\boldsymbol{w}}_c) + \nabla\mathcal{L}(\tilde{\boldsymbol{w}}_c)^\top(\tilde{\boldsymbol{w}}_a - \tilde{\boldsymbol{w}}_c) \tag{3.144}$$

$$\mathcal{L}(\tilde{\boldsymbol{w}}_b) \geq \mathcal{L}(\tilde{\boldsymbol{w}}_c) + \nabla\mathcal{L}(\tilde{\boldsymbol{w}}_c)^\top(\tilde{\boldsymbol{w}}_b - \tilde{\boldsymbol{w}}_c) \tag{3.145}$$

对公式3.144两边同时乘以 λ，并对公式3.145两边同时乘以 $1-\lambda$，再将两式相加，可得：

$$\lambda\mathcal{L}(\tilde{\boldsymbol{w}}_a) + (1-\lambda)\mathcal{L}(\tilde{\boldsymbol{w}}_b) \geq \mathcal{L}(\tilde{\boldsymbol{w}}_c) \tag{3.146}$$

以上不等式正是凸函数的定义形式，因此也就证明了负对数似然损失函数（公式3.135）是一个凸函数。从公式3.142可知，若损失函数在 $\tilde{\boldsymbol{w}}$ 处的一阶导数为零（即该处为局部最小值），则它必然也是全局最小值。这是因为该不等式 $\mathcal{L}(\tilde{\boldsymbol{w}}+\boldsymbol{h}) \geq \mathcal{L}(\tilde{\boldsymbol{w}})$ 对任意方向的扰动向量 $\boldsymbol{h}$ 都成立，因而在整个定义域内不可能存在更小的函数值。

根据损失函数的二阶泰勒展开形式（公式3.139），为了找到能使损失函数下降最快的参数更新方向 $\boldsymbol{h}$，我们可以将该二阶泰勒展开对 $\boldsymbol{h}$ 求偏导，并令导数为零，从而得到最优更新方向

的解析形式：

$$\nabla\mathcal{L}(\tilde{\boldsymbol{w}}) + \mathbf{H}\boldsymbol{h} = 0 \quad \Rightarrow$$
$$\boldsymbol{h} = -\mathbf{H}^{-1}\nabla\mathcal{L}(\tilde{\boldsymbol{w}}) \tag{3.147}$$

按以上更新方向，则权重参数的更新公式为 [25]：

$$\tilde{\boldsymbol{w}}^{(j+1)} = \tilde{\boldsymbol{w}}^{(j)} - \mathbf{H}^{-1}\nabla\mathcal{L}(\tilde{\boldsymbol{w}}) \tag{3.148}$$

上述更新规则来源于经典的牛顿–拉弗森方法（Newton-Raphson Method），该方法通过引入目标函数的二阶导数信息来加快参数优化过程。由于海森矩阵 $\mathbf{H}$ 的计算依赖于当前的参数向量 $\tilde{\boldsymbol{w}}$。因此，每一次迭代也需要根据当前的 $\tilde{\boldsymbol{w}}$ 重新计算相应的海森矩阵 $\mathbf{H}$。

接下来，我们来讨论多分类的情形。之前讨论概率生成模型的时候已经提到，样本 $\boldsymbol{x}$ 属于类别 $\mathcal{C}_k$ 的后验概率可以表示为：

$$p(\mathcal{C}_k|\boldsymbol{x}) = y_k(\boldsymbol{x}) = \frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)} \tag{3.149}$$

其中：

$$z_k = \tilde{\boldsymbol{x}}^{\top}\tilde{\boldsymbol{w}}_k \tag{3.150}$$

$\tilde{\boldsymbol{x}}$ 是在特征向量 $\boldsymbol{x}$ 中添加一个值为 1 的元素（对应于偏置项）之后的增广向量，即 $\tilde{\boldsymbol{x}} = (1, \boldsymbol{x}^{\top})^{\top}$，而 $\tilde{\boldsymbol{w}}$ 是包含了偏置项 w_0 后的增广权重向量。我们采用 1-of-K 编码来表示的类别向量。假设共有 K 个类别，对属于第 j 类的某个样本 $\boldsymbol{x}_i$，其目标输出向量（类别向量）$\boldsymbol{t}_i$ 为一个 K 维向量，其中第 j 个元素 $t_{ij} = 1$，其余元素均为 0。给定包含 N 个样本的训练数据集 $\mathcal{D} = \{(\tilde{\boldsymbol{x}}_i, \boldsymbol{t}_i)\}_{i=1}^{N}$，其似然函数可以表示为：

$$p(\mathbf{T}|\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_K) = \prod_{i=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\boldsymbol{x})^{t_{ik}} = \prod_{i=1}^{N}\prod_{k=1}^{K} y_{ik}^{t_{ik}} \tag{3.151}$$

其中 $y_{ik} = y_k(\boldsymbol{x}_i)$，而 $\mathbf{T}$ 为一个维度为 $N \times K$ 的矩阵，其第 i 行对应于样本 $\boldsymbol{x}_i$ 的目标输出向量 $\boldsymbol{t}_i^{\top}$。对以上似然函数取对数并在前面加上负号后，我们可以将其转化为如下形式的损失函数：

$$\mathcal{L}(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_K) = -\ln p(\mathbf{T}|\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_K) = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_{ik}\ln y_{ik} \tag{3.152}$$

上式被称为多分类问题的交叉熵损失函数。与概率生成模型不同，我们通过直接最小化交叉熵损失函数来优化权重参数 $(\tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_K)$。在优化过程中，需要应用链式法则来计算损失函数对每组参数向量 $\tilde{\boldsymbol{w}}_j$ 的偏导数，从而获得梯度下降的更新方向。

在梯度计算的过程中，较为复杂的一步是对公式3.149中的 Softmax 函数求导。为简化后续计算，我们首先推导 Softmax 函数输出 y_k 关于输入变量 z_j 的偏导数，具体形式如下：

$$\frac{\partial y_k(z_j)}{\partial z_j} = y_k(\mathbb{I}(k = j) - y_j) \tag{3.153}$$

其中 $\mathbb{I}(k=j)$ 称为指示函数（Indicator function），其定义如下：

$$\mathbb{I}(k=j)=\begin{cases}1 & 若\ \ k=j\\ 0 & 若\ \ k\neq j\end{cases} \tag{3.154}$$

为了推导出公式3.153，我们分 $k=j$ 和 $k\neq j$ 两种情况进行分析。当 $k\neq j$ 时，我们有：

$$\begin{aligned}\frac{\partial y_k(z_j)}{\partial z_j}&=\frac{-\exp(z_k)\exp(z_j)}{\left(\sum_{h=1}^{K}\exp(z_h)\right)^2}\\&=\frac{\exp(z_k)}{\sum_{h=1}^{K}\exp(z_h)}\left(0-\frac{\exp(z_j)}{\sum_{h=1}^{K}\exp(z_h)}\right)\\&=y_k(0-y_j)\end{aligned} \tag{3.155}$$

当 $k=j$ 时，则有：

$$\begin{aligned}\frac{\partial y_k(z_j)}{\partial z_j}&=\frac{\exp(z_k)}{\sum_{h=1}^{K}\exp(z_h)}+\frac{-\exp(z_k)\exp(z_k)}{\left(\sum_{h=1}^{K}\exp(z_h)\right)^2}\\&=\frac{\exp(z_k)}{\sum_{h=1}^{K}\exp(z_h)}\left(1-\frac{\exp(z_k)}{\sum_{h=1}^{K}\exp(z_h)}\right)\\&=y_k(1-y_k)\end{aligned} \tag{3.156}$$

将公式3.155和3.156合并后，即可得到公式3.153。利用该公式，我们可以计算损失函数关于权重参数 $\tilde{\boldsymbol{w}}_j$ 的偏导数。这一步是实现梯度下降等优化算法的关键，因为它为我们提供了损失函数在当前参数时的变化方向。具体地，利用公式3.153所给出的 Softmax 函数导数形式，我们可以得到如下关于 $\tilde{\boldsymbol{w}}_j$ 的梯度表达式：

$$\nabla_{\tilde{\boldsymbol{w}}_j}\mathcal{L}(\tilde{\boldsymbol{w}}_1,\ldots,\tilde{\boldsymbol{w}}_K)=-\sum_{i=1}^{N}\sum_{k=1}^{K}\frac{t_{ik}}{y_{ik}}y_{ik}(\mathbb{I}(k=j)-y_{ij})\tilde{\boldsymbol{x}}_i=\sum_{i=1}^{N}(y_{ij}-t_{ij})\tilde{\boldsymbol{x}}_i \tag{3.157}$$

以上推导过程中，只有 $t_{ij}=1$ 的那一项会被保留，其余项因 $t_{ik}=0$ 而被消去。同样地，我们可以据此设计一个序贯学习算法，该算法每次输入一个样本，并按照类似公式3.138对每组权重向量进行迭代更新。我们还可以进一步计算损失函数对权重参数的二阶偏导数，该导数对应于海森矩阵中的第 (j,k) 个子块，具体形式如下：

$$\nabla_{\tilde{\boldsymbol{w}}_k}\nabla_{\tilde{\boldsymbol{w}}_j}\mathcal{L}(\tilde{\boldsymbol{w}}_1,\ldots,\tilde{\boldsymbol{w}}_K)=\sum_{i=1}^{N}y_{ik}(\mathbb{I}(k=j)-y_{ij})\tilde{\boldsymbol{x}}_i\tilde{\boldsymbol{x}}_i^{\top} \tag{3.158}$$

因此，我们也可以采用类似公式3.148的方式，对权重参数进行迭代更新。需要指出的是，本节介绍的方法同样适用于对输入特征向量 $\boldsymbol{x}$ 通过一组基函数进行变换后所得到的新特征空间。在原始特征空间中线性不可分的问题，在经过适当的基函数映射后，可能会在变换后的特征空间中变得线性可分。

主要符号表

数学符号

$\boldsymbol{x}$	向量
y	标量
D	向量维度
x_j	向量的第 j 个分量（向量分量为标量）
f	输入到输出的某种映射
$\mathbf{X}^\top$	矩阵转置
$\mathbf{X}^{-1}$	逆矩阵
$\|\mathbf{X}\|$	矩阵的行列式
$\mathrm{tr}(\mathbf{X})$	矩阵的迹（矩阵所有特征值之和，也等于主对角线元素的总和）
$\ln$	以自然常数 e 为底的对数
e^x 或 $\exp(x)$	自然常数 e 的指数函数
$f'(x)$	一阶导数
$f''(x)$	二阶导数
$\nabla f(\boldsymbol{x})$	一阶梯度（梯度向量）
$\nabla^2 f(\boldsymbol{x})$	二阶梯度（海森矩阵）
$\frac{\partial f(x)}{\partial x}$	一阶偏导
$\frac{\partial^2 f(x)}{\partial x^2}$	二阶偏导

机器学习

$\mathcal{D}$	数据集
N	样本数量
$\boldsymbol{x}_i$	第 i 个训练样本
y_i	第 i 个训练样本对应的输出
$\kappa(\boldsymbol{x}, \boldsymbol{x}')$	核函数
$\mathcal{L}(\cdot)$	损失函数
$\mathcal{J}(\cdot)$	目标函数或代价函数

概率统计

$\mathbb{E}[x]$	均值或期望
$\text{Var}[x]$	方差
$\text{Cov}[x, z]$	随机变量 x 和 z 之间的协方差
$\Gamma(z)$	伽玛函数
$\mathcal{N}(x\|\mu, \sigma^2)$	高斯分布，其中 μ 为均值，σ 为标准差
$\mathcal{N}(x\|\mu, \lambda^{-1})$	高斯分布，其中 μ 为均值，λ 为精度
$\text{Bern}(x\|\mu)$	伯努利分布
$\text{Bin}(k\|\mu, N)$	二项分布
$\text{Beta}(\mu\|\alpha, \beta)$	贝塔分布
$\text{Poisson}(k\|\lambda)$	泊松分布
$\text{Exp}(\lambda\|\beta)$	指数分布
$\text{Gamma}(\lambda\|\alpha, \beta)$	伽玛分布，其中 α 为形状参数，β 为逆尺度参数
$\text{Mult}(\boldsymbol{m}\|\boldsymbol{\mu}, N)$	多项分布
$\text{Dir}(\boldsymbol{\mu}\|\boldsymbol{\alpha})$	狄利克雷分布
$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}, \boldsymbol{\Sigma})$	多维高斯分布，其中 $\boldsymbol{\mu}$ 为均值向量，$\boldsymbol{\Sigma}$ 为协方差矩阵
$\mathcal{N}(\boldsymbol{x}\|\boldsymbol{\mu}, \boldsymbol{\Lambda})$	多维高斯分布，其中 $\boldsymbol{\mu}$ 为均值向量，$\boldsymbol{\Lambda}$ 为精度矩阵
$\mathcal{W}(\boldsymbol{\Lambda}\|\mathbf{W}, \eta)$	威沙特分布，其中 $\mathbf{W}$ 为尺度矩阵，η 为自由度
$\text{NormalGamma}(\mu, \lambda\|\upsilon, \nu, \alpha, \beta)$	高斯-伽玛分布
$\text{NormalWishart}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\|\boldsymbol{\upsilon}, \nu, \mathbf{W}, \eta)$	高斯-威沙特分布
$\text{St}(x\|\mu, \tau, \eta)$	学生氏分布，其中 μ 为均值，τ 为精度，η 为自由度
$\text{St}(\boldsymbol{x}\|\boldsymbol{\mu}, \boldsymbol{\Lambda}, \eta)$	多维学生氏分布，其中 $\boldsymbol{\mu}$ 为均值向量，$\boldsymbol{\Lambda}$ 为精度矩阵
$\text{Laplace}(x\|\mu, \tau)$	拉普拉斯分布，其中 μ 为位置参数，τ 为尺度参数
$\chi^2(x\|\eta)$	卡方分布，其中 η 为自由度
$\text{U}(x\|a, b)$	连续型均匀分布，其中 a 为下界、b 为上界 $(a < b)$
$\text{H}[x]$	信息熵，其中 x 为随机变量
$\text{H}[y\|x]$	条件熵，其中 x 和 y 为随机变量
$\text{H}[p, q]$	分布 p 和 q 之间的交叉熵
$D_{\text{KL}}(p\|\|q)$	分布 p 和 q 之间的 KL 散度

参考文献

[1] Daniel J. Amit, Hanoch Gutfreund, and Haim Sompolinsky. “Storing infinite numbers of patterns in a spin-glass model of neural networks”. In: *Physical Review Letters* 55.14 (1985), pp. 1530–1533.

[2] Mikel Artetxe et al. “On the role of bidirectionality in language model pre-training”. In: *Findings of the Conference on Empirical Methods in Natural Language Processing* (2022), pp. 3973–3985.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv: 1607.06450* (2016).

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166.

[5] Jame O. Berger. *Statistical decision theory and Bayesian analysis (second edition)*. Springer, 1980.

[6] James Bergstra and Yoshua Bengio. “Random search for hyper-parameter optimization”. In: *Journal of Machine Learning Research* 13.2 (2012).

[7] Jeremy Bernstein and Laker Newhouse. “Old optimizer, new norm: An anthology”. In: *Proceedings of the 16th Annual Workshop on Optimization for Machine Learning* (2024), pp. 1–19.

[8] J Besag. “On spatio-temporal models and Markov fields”. In: *Transactions of the 7th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes* (1974), pp. 47–75.

[9] Julian Besag. “On the statistical analysis of dirty pictures”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 48.3 (1986), pp. 259–279.

[10] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[11] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. 1992, pp. 144–152.

[12] George E.P. Box and Norman Richard Draper. *Empirical model-building and response surfaces*. John Wiley & Sons, 1987.

[13] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[14] Tom Brown et al. “Language models are few-shot learners”. In: *Proceedings of the Conference on Neural Information Processing Systems* 33 (2020), pp. 1877–1901.

[15] Kyunghyun Cho et al. “Learning phrase representations using RNN encoder–decoder for statistical machine translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (2014), pp. 1724–1734.

[16] Peter Clifford. “Markov random fields in statistics”. In: *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley* (1990), pp. 19–32.

[17] Thomas H. Cormen et al. *Introduction to algorithms (third edition)*. The MIT Press, 2022.

[18] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley, 1991.

[19] George Cybenko. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[20] Morris H. DeGroot and Mark J. Schevish. *Probability and statistics (forth edition)*. China Machine Press, 2012.

[21] Jacob Devlin et al. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2019), pp. 4171–4186.

[22] Alexey Dosovitskiy et al. “An image is worth 16×16 words: Transformers for image recognition at scale”. In: *Proceedings of the International Conference on Learning Representations* (2021).

[23] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of Machine Learning Research* 12.7 (2011).

[24] Walter D. Fisher. “On grouping for maximum homogeneity”. In: *Journal of the American Statistical Association* 53.284 (1958), pp. 789–798.

[25] Roger Fletcher. *Practical methods of optimization (second edition)*. Wiley, 1987.

[26] Yoav Freund and Robert E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of Computer and System Sciences* 55.1 (1997), pp. 119–139.

[27] Stuart Geman and Donald Geman. “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984), pp. 721–741.

[28] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber. “Learning precise timing with LSTM recurrent networks”. In: *Journal of Machine Learning Research* 3 (2002), pp. 115–143.

[29] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256.

[30] Ian J Goodfellow et al. “Generative adversarial nets”. In: *Proceedings of the Conference on Neural Information Processing Systems* 27 (2014).

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction (second edition)*. Springer, 2009.

[32] Kaiming He and Jian Sun. “Convolutional neural networks at constrained time cost”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5353–5360.

[33] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778.

[34] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1026–1034.

[35] Nicholas J. Higham. *Functions of matrices: Theory and computation*. SIAM, 2008.

[36] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. “Distilling the knowledge in a neural network”. In: *arXiv: 1503.02531* (2015).

[37] Geoffrey E. Hinton. “Training products of experts by minimizing contrastive divergence”. In: *Neural Computation* 14.8 (2002), pp. 1771–1800.

[38] John J. Hopfield. “Neural networks and physical systems with emergent collective computational abilities”. In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558.

[39] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the International Conference on Machine Learning* (2015), pp. 448–456.

[40] Arieh Iserles. *A first course in the numerical analysis of differential equations*. 44. Cambridge university press, 2009.

[41] Max Jaderberg et al. “Population based training of neural networks”. In: *arXiv: 1711.09846* (2017).

[42] Kevin Jamieson and Ameet Talwalkar. “Non-stochastic best arm identification and hyperparameter optimization”. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (2016), pp. 240–248.

[43] Keller Jordan et al. *Muon: An optimizer for hidden layers in neural networks*. 2024. URL: https://kellerjordan.github.io/posts/muon/.

[44] Diederik P. Kingma and Jimmy L. Ba. “Adam: A method for stochastic optimization”. In: *Proceedings of the International Conference on Learning Representations* (2015).

[45] Josef Kittler and Janos Föglein. “Contextual classification of multispectral pixel data”. In: *Image and vision computing* 2.1 (1984), pp. 13–29.

[46] Solomon Kullback and Richard A. Leibler. “On information and sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[47] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (2002), pp. 2278–2324.

[48] Mike Lewis et al. “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 7871–7880.

[49] Yaron Lipman et al. “Flow matching guide and code”. In: *arXiv: 2412.06264* (2024).

[50] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv: 1711.05101* (2017).

[51] Calvin Luo. “Understanding diffusion models: A unified perspective”. In: *arXiv: 2208.11970* (2022).

[52] Marvin Minsky and Seymour Papert. *Perceptrons: An introduction to computational geometry*. The MIT Press, 1969.

[53] Kevin P. Murphy. *Machine learning: A probabilistic perspective*. The MIT Press, 2012.

[54] Brady Neal et al. “A modern take on the bias-variance tradeoff in neural networks”. In: *arXiv: 1810.08591* (2018).

[55] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *Proceedings of the Conference on Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[56] Judea Pearl. *Probabilistic reasoning in intelligent systems*. Morgan kaufmann, 2014.

[57] Matthew E. Peters et al. “Deep Contextualized Word Representations”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2018), pp. 2227–2237.

[58] John C. Platt. “Fast training of support vector machines using sequential minimal optimization”. In: *Advances in Kernel Methods* (1999), pp. 185–208.

[59] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *Proceedings of the International Conference on Learning Representations* (2016).

[60] Alec Radford et al. “Improving language understanding by generative pre-training”. In: *OpenAI’s Technical Report* (2018).

[61] Colin Raffel et al. “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67.

[62] Scott Reed et al. “Generative adversarial text to image synthesis”. In: *Proceedings of the International Conference on Machine Learning* (2016), pp. 1060–1069.

[63] Edmund T. Rolls. *Cerebral cortex: principles of operation*. Oxford University Press, 2016.

[64] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional networks for biomedical image segmentation”. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (2015), pp. 234–241.

[65] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536.

[66] Leonard J. Savage. *The subjective basis of statistical practice*. University of Michigan, 1961.

[67] Andrew M Saxe, James L McClelland, and Surya Ganguli. “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks”. In: *Proceedings of the International Conference on Learning Representations* (2014).

[68] Robert E. Schapire. “The strength of weak learnability”. In: *Machine learning* 5.2 (1990), pp. 197–227.

[69] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural machine translation of rare words with subword units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (2016), pp. 1715–1725.

[70] Claude Elwood Shannon. “A mathematical theory of communication”. In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423.

[71] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical Bayesian optimization of machine learning algorithms”. In: *Proceedings of the Conference on Neural Information Processing Systems* 25 (2012).

[72] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: *Proceedings of the International Conference on Learning Representations* (2021).

[73] Gilbert Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press, 2016.

[74] Jianlin Su et al. “Roformer: Enhanced transformer with rotary position embedding”. In: *Neurocomputing* 568 (2024), p. 127063.

[75] Tijmen Tieleman and Geoffrey E. Hinton. “Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural Networks for Machine Learning* 4.2 (2012), pp. 26–26.

[76] Antonio Torralba, Phillip Isola, and William Freeman. *Foundations of computer vision*. The MIT Press, 2024.

[77] Ashish Vaswani et al. “Attention is all you need”. In: *Proceedings of the Conference on Neural Information Processing Systems* 30 (2017).

[78] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. “Deep learning for Chinese word segmentation and POS tagging”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), pp. 647–657.

[79] Ji Zhu et al. “Multi-class AdaBoost”. In: *Statistics and its Interface* 2.3 (2009), pp. 349–360.

[80] Hui Zou and Trevor Hastie. “Regularization and variable selection via the elastic net”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320.