第5章 基本分类学习

CHAPTER 5

本章讨论分类问题，介绍几种基本分类算法。为了概念的清楚，本章将只有两种类型的二分类问题和有多于两种类型的多类问题分开讨论。分类问题的表示方法比回归问题更丰富，本章将通过几个比较简单的方法理解分类中遇到的多种表示方式和目标函数，这些内容在后续章节中得以继续应用和扩展。例如，本章介绍的逻辑回归所使用的目标函数和优化算法，加以推广则可用于神经网络的学习。

5.1 基本分类问题

ML08-分类学习-1

设有满足独立同分布条件(I.I.D)的训练数据集为

$$\boldsymbol{D}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_N,y_N)\}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N} \tag{5.1.1}$$

对于分类问题，训练集的标注 y 仅取有限的离散值，即 y 表示其所代表的类型编号。可将标注值表示为 $y\in\{1,2,\cdots,C\}$，其中 C 表示一个学习任务中待分类类型的数目。当 $C=2$ 时表示二分类任务，这是一种基本的分类任务。由于二分类可以清楚地说明分类的学习方法且表示简单、概念清晰，故本章重点讨论二分类问题。当 $C>2$ 时称为多分类任务，可在二分类的概念和方法基础上进一步扩展至多分类。

首先讨论在分类中如何表示类型，包括标注和分类器的输出，以标注为例进行讨论。在二分类中，由于标注 y 仅取两个值，常用二值单变量表示类型，即 $y\in\{0,1\}$ 或 $y\in\{1,-1\}$，本章多采用前者。用 $y=1$ 表示类型1，也可用符号 C_1 表示类型1；$y=0$ 表示类型2，也可用符号 C_2 表示类型2。即 $y=1$ 与 C_1 代表相同含义，类似地，$y=0$ 与 C_2 相同含义。

在表示多分类任务中，标注 y 可以选择两种不同的表示方式，一种是 y 直接取一个标量值，即 y 取集合 $\{1,2,\cdots,C\}$ 中的值。机器学习中较多采用另一种 C 维向量编码方式表示 y，即用以下形式的向量表示多类标注。

$$\boldsymbol{y}=[0,\cdots,0,1,0,\cdots,0]^{\mathrm{T}} \tag{5.1.2}$$

$\boldsymbol{y}$ 中各分量 y_i 只取0或1，且 $\sum_i y_i=1$，即只有一个分量 $y_k=1$ 表示 $\boldsymbol{y}$ 标注的是第 k 类，其他 $y_i=0, i\neq k$。后面将会看到，这种编码向量表示多类型有其方便性。

与回归问题一样，利用训练数据集训练一个分类模型，模型的一般表示为

$$\hat{\boldsymbol{y}}=f(\boldsymbol{x}) \tag{5.1.3}$$

其中，$\boldsymbol{x}$ 表示一个新的特征输入；$\hat{\boldsymbol{y}}$ 表示分类输出。式(5.1.3)是分类模型的一般性表示，在实际中，分类输出分为确定性输出和概率输出两大类，然后再区分两种不同的概率表示方

式，可以把式(5.1.3)表示的分类模型分成3种情况，下面分别介绍。

1. 判别函数模型

与第4章介绍的基本回归模型类似，分类输出可表示为确定性的函数，如线性模型

$$\hat{y}(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + w_0 \tag{5.1.4}$$

通过训练确定参数 $\boldsymbol{w}$ 和 w_0，对于新的 $\boldsymbol{x}$ 计算 $\hat{y}(\boldsymbol{x})$ 并与门限比较确定分为哪一类。也可以通过一个非线性函数(称为激活函数)得到以下广义线性模型。

$$\hat{y}(\boldsymbol{x}) = f(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + w_0) \tag{5.1.5}$$

其中，$f(\cdot)$为激活函数，在二分类问题中，最简单的是 $f(\cdot)$的输出只取$\{0,1\}$或$\{1,-1\}$，如取 $f(\cdot)$为符号函数 $\mathrm{sgn}(\cdot)$。模型的输出可以直接分类。

有几种分类器属于判别函数的分类模型，如经典的感知机、Fisher判别函数和支持向量机(SVM)。

2. 判别概率模型

判别概率模型是常用的一种分类模型，也简称判别模型。式(5.1.3)不再是一个确定函数，而是一个后验概率。在二分类问题中，式(5.1.3)的函数 $f(\boldsymbol{x})$实际是通过特征输入 $\boldsymbol{x}$ 计算分类为第1类的后验概率，即

$$\hat{y} = f(\boldsymbol{x}) = p(C_1 \mid \boldsymbol{x}) \tag{5.1.6}$$

则输入 $\boldsymbol{x}$ 分类为第2类的后验概率为 $1-\hat{y}=1-p(C_1|\boldsymbol{x})$，然后由第3章介绍的决策原理得到最终分类输出。

若推广到多类情况，则式(5.1.3)的 $\hat{\boldsymbol{y}}$ 是一个向量，其分量表示分类为 C_k 的后验概率，即

$$\hat{y}_k = p(C_k \mid \boldsymbol{x}) \tag{5.1.7}$$

同样可根据决策原理得到分类输出。

应用后验概率进行分类具有灵活性，如第3章讨论的，若对每类分类错误的代价是相同的，则直接将输入特征向量分类为后验概率最大的一个，即分类为

$$C_k = \arg\max_{C_i}\{p(C_i \mid \boldsymbol{x})\} \tag{5.1.8}$$

但当不同分类的分类错误代价不同时，如第3章所讨论的，可以通过对后验概率的加权得到分类决策，只要得到了后验概率，调整加权不需要重新计算后验概率。因此，求得后验概率后，可根据不同任务设置加权矩阵，获得不同要求下的分类输出。后验概率还有许多其他可用来扩展问题的特性，此处不再赘述。

3. 生成概率模型

一个构成更完整的概率描述的方法是生成概率模型，简称生成模型。通过训练样本集学习得到特征向量 $\boldsymbol{x}$ 和类型输出 $\boldsymbol{y}$ 的联合概率 $p(\boldsymbol{x},\boldsymbol{y})$。一般情况下得到联合概率比得到后验概率更困难，因此，生成模型一直是机器学习中更有挑战性的工作。

从分类任务来讲，表示类型的 $\boldsymbol{y}$ 的取值是有限的，若按式(5.1.2)的编码模式表示 $\boldsymbol{y}$，则 $\boldsymbol{y}$ 只有 C 种取值，即 $y_k=1, y_{i\neq k}=0, k=1,2,\cdots,C$，故联合概率可以表示成由 C 个固定 $\boldsymbol{y}$ 取值的概率集合，即$\{p(\boldsymbol{x},y_k=1), k=1,2,\cdots,C\}$。为了突出类型，可用 $p(\boldsymbol{x},C_k)$代表 $p(\boldsymbol{x},y_k=1)$，即用 C_k 代表 $y_k=1, y_{i\neq k}=0$ 的情况。在许多实际问题中，可能更容易得到

类型作为条件下的特征向量 $\boldsymbol{x}$ 的概率函数 $p(\boldsymbol{x}\mid y_k=1)=p(\boldsymbol{x}\mid C_k)$ 和类型的先验概率 $p(y_k=1)=p(C_k)$，这里的 $p(\boldsymbol{x}\mid C_k)$ 称为类条件概率。由如式(5.1.9)所示的作为分类问题的生成模型。由概率公式

$$p(\boldsymbol{x},C_k)=p(\boldsymbol{x}\mid C_k)p(C_k) \tag{5.1.9}$$

对于所有 k，可得到 $p(\boldsymbol{x},C_k)$ 或 $p(\boldsymbol{x}\mid C_k)$ 与 $p(C_k)$，它们是等同的。由贝叶斯公式，有了生成模型，则后验概率为

$$p(C_k\mid\boldsymbol{x})=\frac{p(\boldsymbol{x},C_k)}{p(\boldsymbol{x})}=\frac{p(\boldsymbol{x}\mid C_k)p(C_k)}{p(\boldsymbol{x})}=\frac{p(\boldsymbol{x}\mid C_k)p(C_k)}{\sum\limits_k p(\boldsymbol{x}\mid C_k)p(C_k)} \tag{5.1.10}$$

得到生成模型后，可直接得到类后验概率，利用决策原理进行分类。生成模型有更多的信息可用，按照样本是由对联合概率 $p(\boldsymbol{x},\boldsymbol{y})$ 采样获得的假设，既然已得到了较准确的联合概率，则可以通过一些采样技术获得新的增广样本，改善学习质量。

注意到，对于判别概率模型和生成概率模型，由于最终做分类决策的都是用类后验概率，容易混淆其区别。对于判别概率模型，通过样本直接训练得到类后验概率表达式，并通过后验概率进行分类决策；对于生成概率模型，直接学习得到的是联合概率，然后利用联合概率获得类后验概率进行分类决策，但是联合概率可以获得其他应用。判别概率模型学习过程中没有联合概率的出现，生成概率模型重要的是获得联合概率(或其等价物)，用联合概率计算类后验概率用于分类只是其部分功能。

5.3 节将介绍的逻辑回归是一种基本的判别概率模型分类算法，尽管原理简单，但其概念很容易扩展到神经网络的分类。目前神经网络包括深度神经网络的分类任务多数属于判别概率模型。4.4 节介绍了一种简单的生成模型分类器——朴素贝叶斯算法。第 11 章将介绍深度学习中的一类生成模型——对抗生成网络(Generative Adversarial Networks，GAN)(GAN 是一种无监督学习)。

5.2　线性判别函数模型

对于二分类问题，线性判别函数是指学习一个形式(5.1.4)的模型，为方便，将线性判别函数重写为

$$\hat{y}(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0 \tag{5.2.1}$$

对于一个给定的新输入 $\boldsymbol{x}$，若 $\hat{y}(\boldsymbol{x})>0$，判别为第 1 类 C_1；若 $\hat{y}(\boldsymbol{x})<0$，判别为第 2 类 C_2。式(5.2.1)中的 w_0 为偏置参数。$-w_0$ 可看作阈值，$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}>-w_0$ 时可判别为 C_1。$\hat{y}(\boldsymbol{x})=0$ 时可随机选择判别为 C_1 或 C_2，或不做判决。

设 $\boldsymbol{x}$ 为 K 维向量，则 $\hat{y}(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0=0$ 是 K 维空间的一个平面，将 K 维空间划分为两个子空间 R_1 和 R_2，$\boldsymbol{x}$ 落入 R_1 时可判别为 C_1，$\boldsymbol{x}$ 落入 R_2 时判别为 C_2。超平面 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0=0$ 称为判决面。若有 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 均处于判决面上，则有

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_1+w_0=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_2+w_0$$

故

$$\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_1-\boldsymbol{x}_2)=0$$

即 $\boldsymbol{w}$ 与判决面上的任意向量正交，$\boldsymbol{w}$ 表示判决面的法线方向，由于 $\boldsymbol{x}\in R_1$ 时 $\hat{y}(\boldsymbol{x})>0$，故 $\boldsymbol{w}$ 指向的方向为 R_1。图 5.2.1 给出了三维空间判决面的示意图。

对于位于判决面之外的任意点 $\boldsymbol{x}$，它在判决面上的投影为 $\boldsymbol{x}_{\mathrm{p}}$，则 $\boldsymbol{x}$ 可表示为

$$\boldsymbol{x}=\boldsymbol{x}_{\mathrm{p}}\pm r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{5.2.2}$$

其中，$\pm$ 分别表示 $\boldsymbol{x}$ 在 R_1 和 R_2 内；$r>0$ 表示 $\boldsymbol{x}$ 到判决面的距离。将式(5.2.2)表示的 $\boldsymbol{x}$ 代入式(5.2.1)得

$$\hat{y}(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0=\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x}_{\mathrm{p}}\pm r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right)+w_0$$

$$=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_{\mathrm{p}}+w_0\pm r\|\boldsymbol{w}\|=\pm r\|\boldsymbol{w}\|$$

即

$$r=\pm\frac{\hat{y}(\boldsymbol{x})}{\|\boldsymbol{w}\|}=\frac{|\hat{y}(\boldsymbol{x})|}{\|\boldsymbol{w}\|} \tag{5.2.3}$$

式(5.2.3)表示空间任意点到判决面的距离。

对于给定的一组数据集，如果可以找到一个判决面将数据集的样本完全正确分类，则称数据集是线性可分的。设一个数据集的 $\boldsymbol{x}_n$ 是二维的，如图 5.2.2 所示，则该数据集是线性可分的。对于二维问题，判决面退化成简单的直线，图 5.2.2 中画出了 3 条判决线，均可以将所有样本正确分类。

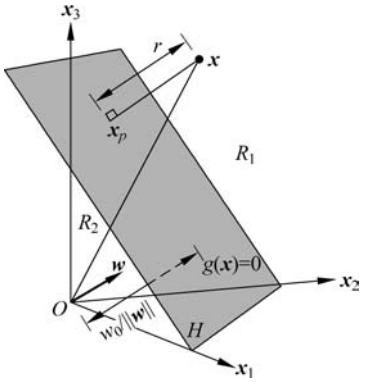

图 5.2.1 三维空间判决面的示意图

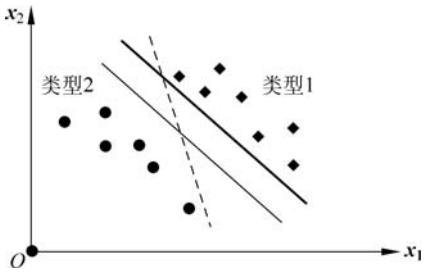

图 5.2.2 线性可分数据集的例子

对于二分类问题，可以通过最小二乘得到权系数 $\boldsymbol{w}$ 的解。分类问题 LS 解的过程与线性回归类似，不同之处在于样本集合 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N}$ 中的标注值 y_n 只取两个不同值。通过 $\hat{y}(\boldsymbol{x}_n)$ 与 y_n 之间误差平方和最小得到 $\boldsymbol{w}$ 的解，解的方程见第 4 章。由于二分类问题的 LS 解存在性能上的诸多缺陷，实际中很少使用，因此本节也不再赘述。

实际中应用最多的一种判别函数方法是支持向量机(SVM)，其详细分析需要较大篇幅，将在第 6 章专门讨论。本节后续将对历史上曾有重要影响的两类方法做概要叙述，分别是 Fisher 线性判别分析和感知机。

5.2.1 Fisher 线性判别分析

Fisher 线性判别分析(Linear Discriminant Analysis，LDA)不是一种标准的线性判别

函数模型，它实际上是一种通过降维对数据类型进行最大分离的方法，但当降维的结果结合阈值进行分类时，与分类的线性判别函数模型是一致的。

首先讨论二分类问题，然后推广到多类问题。

1. 二分类 Fisher LDA

设有一组数据样本 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N}$，这里讨论二分类的情况，故 $y_n=1$ 表示类型 C_1，$y_n=0$ 表示类型 C_2。将属于 C_1 的子样本集记为 $\boldsymbol{D}_1$，共有 N_1 个样本；属于 C_2 的子样本集记为 $\boldsymbol{D}_2$，共有 N_2 个样本。目标是给出一个向量 $\boldsymbol{w}$，将每个样本投影到 $\boldsymbol{w}$ 上得到一维投影值 $\hat{y}$，即

$$\hat{y}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \tag{5.2.4}$$

设 $\boldsymbol{x}$ 为 K 维向量，$\boldsymbol{w}$ 代表 K 维空间的一条直线，若 $\|\boldsymbol{w}\|=1$，则 $\hat{y}$ 为 $\boldsymbol{x}$ 在 $\boldsymbol{w}$ 上的投影值，实际上 $\boldsymbol{w}$ 的方向是重要的，其范数大小无关紧要。

对于数据集 $\boldsymbol{D}$，根据标注分成两个子集 $\boldsymbol{D}_1$ 和 $\boldsymbol{D}_2$，两个子集的每个样本对应 K 维空间的一个点，将其投影到 $\boldsymbol{w}$ 表示的直线，即投影直线上的一点。对于每个 $\boldsymbol{x}_n$，得到相应的投影 $\hat{y}_n$，可得到数据集中所有样本在 $\boldsymbol{w}$ 直线表示的一维空间中的投影集合 $\{\hat{y}_n\}_{n=1}^{N}$，其中子集 $\boldsymbol{D}_1$ 和 $\boldsymbol{D}_2$ 的投影子集分别记为 $Y_1=\{\hat{y}_n\}_{\boldsymbol{x}_n\in\boldsymbol{D}_1}$ 和 $Y_2=\{\hat{y}_n\}_{\boldsymbol{x}_n\in\boldsymbol{D}_2}$。若希望从投影中将类型区分开，则希望 Y_1 和 Y_2 是可分的，Y_1 和 Y_2 的可分性与 $\boldsymbol{w}$ 的方向是相关的。如图 5.2.3 所示，在二维空间中，有两种类型的样本，若投影到图 5.2.3(a)所示的 $\boldsymbol{w}$ 直线，则 Y_1 和 Y_2 是不可分的；若投影到图 5.2.3(b)所示的 $\boldsymbol{w}$ 直线，则 Y_1 和 Y_2 是可分的。

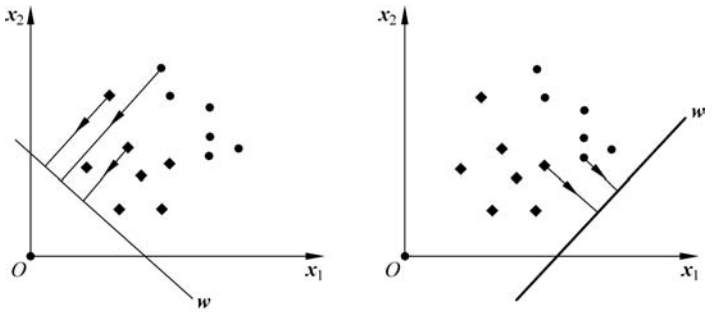

图 5.2.3　样本投影到不同直线的可分离情况

通过这个直观的分析，可将问题叙述为：求最优 $\boldsymbol{w}$ 使集合 Y_1 和 Y_2 具有最大的分离性。为此，用数学形式表示两个集合的分离度，原理上可用两个集合的均值差表示它们的分离程度。对于每个样本 $\boldsymbol{x}_n$，其投影值为 $\hat{y}_n=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n$，故每类的投影均值为

$$\hat{m}_i=\frac{1}{N_i}\sum_{\hat{y}_n\in Y_i}\hat{y}_n=\frac{1}{N_i}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n$$

$$=\boldsymbol{w}^{\mathrm{T}}\frac{1}{N_i}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}\boldsymbol{x}_n=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{m}_i,\quad i=1,2 \tag{5.2.5}$$

注意，这里 $i=1,2$ 分别表示 C_1，C_2 两类样本。式(5.2.5)用到了投影前样本的均值，即

$$\boldsymbol{m}_i=\frac{1}{N_i}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}\boldsymbol{x}_n,\quad i=1,2 \tag{5.2.6}$$

由此得到类投影均值之差为

$$|\hat{m}_1-\hat{m}_2|=|\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}_1-\boldsymbol{m}_2)| \tag{5.2.7}$$

式(5.2.7)的均值差会随着 $\boldsymbol{w}$ 范数的增加而增加。定义类内散布(Scatter)量为

$$\hat{s}_i^2=\sum_{\hat{y}_n\in Y_i}(\hat{y}_n-\hat{m}_i)^2,\quad i=1,2 \tag{5.2.8}$$

用 $\hat{s}_1^2+\hat{s}_2^2$ 表示投影样本的总类内散布,有了这个准备,可将式(5.2.7)的类投影均值之差的平方除以总类内散布定义为 Fisher 准则函数,即

$$J(\boldsymbol{w})=\frac{|\hat{m}_1-\hat{m}_2|^2}{\hat{s}_1^2+\hat{s}_2^2} \tag{5.2.9}$$

一个最优的 $\boldsymbol{w}$,既可使式(5.2.9)的分子尽可能大,即两类投影集尽可能分离,又可使分母尽可能小,即每类在类内聚集得好。因此,使式(5.2.9)最大的 $\boldsymbol{w}$ 是本问题的最优解。为了得到式(5.2.9)与 $\boldsymbol{w}$ 的显式关系,进一步有

$$\begin{aligned}\hat{s}_1^2+\hat{s}_2^2&=\sum_{\hat{y}_n\in Y_1}(\hat{y}_n-\hat{m}_1)^2+\sum_{\hat{y}_n\in Y_2}(\hat{y}_n-\hat{m}_2)^2\\&=\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_1}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{m}_1)^2+\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_n-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{m}_2)^2\\&=\boldsymbol{w}^{\mathrm{T}}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_1}(\boldsymbol{x}_n-\boldsymbol{m}_1)(\boldsymbol{x}_n-\boldsymbol{m}_1)^{\mathrm{T}}\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_2}(\boldsymbol{x}_n-\boldsymbol{m}_2)(\boldsymbol{x}_n-\boldsymbol{m}_2)^{\mathrm{T}}\boldsymbol{w}\\&=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_1\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_2\boldsymbol{w}=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}\end{aligned} \tag{5.2.10}$$

其中,$\boldsymbol{S}_1$ 和 $\boldsymbol{S}_2$ 为类内散布矩阵,即

$$\boldsymbol{S}_i=\sum_{\boldsymbol{x}_n\in D_i}(\boldsymbol{x}_n-\boldsymbol{m}_i)(\boldsymbol{x}_n-\boldsymbol{m}_i)^{\mathrm{T}},\quad i=1,2 \tag{5.2.11}$$

$\boldsymbol{S}_{\mathrm{W}}$ 为总散布矩阵。

$$\boldsymbol{S}_{\mathrm{W}}=\boldsymbol{S}_1+\boldsymbol{S}_2 \tag{5.2.12}$$

注意散布矩阵与自协方差矩阵的估计值之间相差 N 倍的系数。散布矩阵反映了样本的分散度。

类似地,有

$$\begin{aligned}|\hat{m}_1-\hat{m}_2|^2&=(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{m}_1-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{m}_2)^2\\&=\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)^{\mathrm{T}}\boldsymbol{w}\\&=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{w}\end{aligned} \tag{5.2.13}$$

其中,$\boldsymbol{S}_{\mathrm{B}}$ 为类间散布矩阵,即

$$\boldsymbol{S}_{\mathrm{B}}=(\boldsymbol{m}_1-\boldsymbol{m}_2)(\boldsymbol{m}_1-\boldsymbol{m}_2)^{\mathrm{T}} \tag{5.2.14}$$

其秩为 1。将式(5.2.13)和式(5.2.10)代入式(5.2.9)得

$$J(\boldsymbol{w})=\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}} \tag{5.2.15}$$

令式(5.2.15)对 $\boldsymbol{w}$ 的导数等于 0,整理得

$$\frac{2\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}}\left\{(\boldsymbol{m}_1-\boldsymbol{m}_2)-\left[\frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}}\right]\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}\right\}=0 \tag{5.2.16}$$

由于$\frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{m}_1-\boldsymbol{m}_2)}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}}=c$是一个标量，则式(5.2.16)的解为

$$\boldsymbol{w}\propto\boldsymbol{S}_{\mathrm{W}}^{-1}(\boldsymbol{m}_1-\boldsymbol{m}_2)$$

由于$\boldsymbol{w}$的范数大小无关紧要，故取解为

$$\boldsymbol{w}_{\mathrm{o}}=\boldsymbol{S}_{\mathrm{W}}^{-1}(\boldsymbol{m}_1-\boldsymbol{m}_2) \tag{5.2.17}$$

由于$\boldsymbol{S}_{\mathrm{W}}$和$\boldsymbol{m}_1,\boldsymbol{m}_2$都是直接由样本集计算得到，因此，从样本集学习得到最优方向向量$\boldsymbol{w}_{\mathrm{o}}$，对于新的输入向量$\boldsymbol{x}$，可将其投影到$\boldsymbol{w}_{\mathrm{o}}$表示的直线，即投影为$\hat{y}=\boldsymbol{w}_{\mathrm{o}}^{\mathrm{T}}\boldsymbol{x}$，可以通过给出一个门限$b$，利用门限，若$\boldsymbol{x}$满足

$$\boldsymbol{w}_{\mathrm{o}}^{\mathrm{T}}\boldsymbol{x}+b\geqslant 0 \tag{5.2.18}$$

则分类为C_1，否则分类为C_2。

若假设数据的类条件概率$p(\boldsymbol{x}\mid C_i),i=1,2$为高斯分布，且已知类先验概率$p(C_i)$，则可以证明(参考3.4节的讨论)式(5.2.18)的判别式可进一步表示为

$$\boldsymbol{w}_{\mathrm{o}}^{\mathrm{T}}[\boldsymbol{x}-(\boldsymbol{m}_1+\boldsymbol{m}_2)/2]>\ln\frac{p(C_2)}{p(C_1)} \tag{5.2.19}$$

这里，若通过样本集用最大似然估计$p(C_i)$，则有$p(C_1)/p(C_2)=N_1/N_2$。

2. 多类 Fisher LDA

多分类情况下，设共有C类。给出数据样本$\boldsymbol{D}=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1}^{N}$，这里样本标注$\boldsymbol{y}_n$的编码方式不重要，只要由标注将样本分为$C$个子集$\boldsymbol{D}_i,i=1,2,\cdots,C$，每个子集对应第$C_i$类。设$\boldsymbol{x}_n$是$K$维向量，通过$(C-1)$个投影运算，将每个样本投影到$(C-1)$维空间，设$K>C$，投影的第$i$个分量为

$$\hat{y}_i=\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x} \tag{5.2.20}$$

用$\boldsymbol{W}$表示由$\boldsymbol{w}_i$作为第i列的$K\times(C-1)$维矩阵，则$(C-1)$维投影向量$\hat{\boldsymbol{y}}$表示为

$$\hat{\boldsymbol{y}}=[\hat{y}_1,\hat{y}_2,\cdots,\hat{y}_{C-1}]^{\mathrm{T}}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} \tag{5.2.21}$$

对于每个样本$\boldsymbol{x}_n$，由式(5.2.21)产生投影向量$\hat{\boldsymbol{y}}_n$，样本子集$\boldsymbol{D}_i$产生的投影子集$\hat{\boldsymbol{Y}}_i$，$i=1,2,\cdots,C$。与二类问题类似，若定义投影子集的距离和散布矩阵，则其最终都表示为样本散布矩阵和待求$\boldsymbol{W}$之间的表达式，故首先推广二分类的样本散布矩阵到多分类情况。

首先定义样本的类内总散布矩阵为各类内散布矩阵之和，即

$$\boldsymbol{S}_{\mathrm{W}}=\sum_{i=1}^{C}\boldsymbol{S}_i \tag{5.2.22}$$

其中，各类的散布矩阵和类均值为

$$\boldsymbol{S}_i=\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}(\boldsymbol{x}_n-\boldsymbol{m}_i)(\boldsymbol{x}_n-\boldsymbol{m}_i)^{\mathrm{T}},\quad i=1,2,\cdots,C \tag{5.2.23}$$

和

$$\boldsymbol{m}_i=\frac{1}{N_i}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}\boldsymbol{x}_n,\quad i=1,2,\cdots,C \tag{5.2.24}$$

为了导出类间散布矩阵，首先表示全样本的均值$\boldsymbol{m}$和散布矩阵$\boldsymbol{S}_{\mathrm{T}}$为

$$\boldsymbol{m}=\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n=\frac{1}{N}\sum_{i=1}^{C}N_i\boldsymbol{m}_i \tag{5.2.25}$$

和

$$\boldsymbol{S}_{\mathrm{T}}=\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{m})(\boldsymbol{x}_n-\boldsymbol{m})^{\mathrm{T}} \tag{5.2.26}$$

可对 $\boldsymbol{S}_{\mathrm{T}}$ 做分解，一方面，可以将 $\boldsymbol{S}_{\mathrm{T}}$ 分解为类内总散布矩阵与类间散布矩阵之和，即

$$\boldsymbol{S}_{\mathrm{T}}=\boldsymbol{S}_{\mathrm{W}}+\boldsymbol{S}_{\mathrm{B}} \tag{5.2.27}$$

另一方面，得到式(5.2.26)的一种分解形式

$$\begin{aligned}\boldsymbol{S}_{\mathrm{T}}&=\sum_{i=1}^{C}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}(\boldsymbol{x}_n-\boldsymbol{m}_i+\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{x}_n-\boldsymbol{m}_i+\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\\&=\sum_{i=1}^{C}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}(\boldsymbol{x}_n-\boldsymbol{m}_i)(\boldsymbol{x}_n-\boldsymbol{m}_i)^{\mathrm{T}}+\sum_{i=1}^{C}\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_i}(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\\&=\boldsymbol{S}_{\mathrm{W}}+\sum_{i=1}^{C}N_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}}\end{aligned} \tag{5.2.28}$$

因此，得到类间散布矩阵为

$$\boldsymbol{S}_{\mathrm{B}}=\sum_{i=1}^{C}N_i(\boldsymbol{m}_i-\boldsymbol{m})(\boldsymbol{m}_i-\boldsymbol{m})^{\mathrm{T}} \tag{5.2.29}$$

对于数据集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1}^{N}$ 的每个样本的 $\boldsymbol{x}_n$，由式(5.2.21)得到一个投影向量

$$\hat{\boldsymbol{y}}_n=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_n \tag{5.2.30}$$

将每个子集 $\boldsymbol{D}_i$，$i=1,2,\cdots,C$ 投影到相应子集 $\hat{\boldsymbol{Y}}_i$，$i=1,2,\cdots,C$，对比原数据集的各种散布矩阵，可以得到投影子集 $\hat{\boldsymbol{Y}}_i$ 的均值和散布矩阵如下。

$$\hat{\boldsymbol{m}}_i=\frac{1}{N_i}\sum_{\hat{\boldsymbol{y}}_n\in\hat{\boldsymbol{Y}}_i}\hat{\boldsymbol{y}}_n,\quad i=1,2,\cdots,C \tag{5.2.31}$$

$$\hat{\boldsymbol{m}}=\frac{1}{N}\sum_{i=1}^{C}N_i\hat{\boldsymbol{m}}_i \tag{5.2.32}$$

$$\hat{\boldsymbol{S}}_{\mathrm{W}}=\sum_{i=1}^{C}\sum_{\hat{\boldsymbol{y}}_n\in\hat{\boldsymbol{Y}}_i}(\hat{\boldsymbol{y}}_n-\hat{\boldsymbol{m}}_i)(\hat{\boldsymbol{y}}_n-\hat{\boldsymbol{m}}_i)^{\mathrm{T}} \tag{5.2.33}$$

$$\hat{\boldsymbol{S}}_{\mathrm{B}}=\sum_{i=1}^{C}N_i(\hat{\boldsymbol{m}}_i-\hat{\boldsymbol{m}})(\hat{\boldsymbol{m}}_i-\hat{\boldsymbol{m}})^{\mathrm{T}} \tag{5.2.34}$$

将式(5.2.30)代入式(5.2.33)和式(5.2.34)，可以证明

$$\hat{\boldsymbol{S}}_{\mathrm{W}}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{W} \tag{5.2.35}$$

$$\hat{\boldsymbol{S}}_{\mathrm{B}}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{W} \tag{5.2.36}$$

对于多分类情况，由于投影是向量 $\hat{\boldsymbol{y}}$，其类间散布矩阵和类内散布矩阵无法直接评价其分离性，但是这些矩阵的迹是标量并可描述其分离性，故多分类情况下的 Fisher 准则函数为

$$J(\boldsymbol{W})=\frac{\mathrm{tr}(\hat{\boldsymbol{S}}_{\mathrm{B}})}{\mathrm{tr}(\hat{\boldsymbol{S}}_{\mathrm{W}})}=\frac{\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{W})}{\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{W})} \tag{5.2.37}$$

在 4.1.4 节已看到对迹目标函数求最优的问题，类似地，可以证明（留作习题）式(5.2.37)最大化所得到的 $\boldsymbol{W}$ 最优解满足

$$\boldsymbol{S}_{\mathrm{B}}\boldsymbol{w}_i=\lambda_i\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}_i,\quad i=1,2,\cdots,C-1 \tag{5.2.38}$$

其中，$\boldsymbol{w}_i$ 为 $\boldsymbol{W}$ 的列向量，是对应于 $\boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{S}_{\mathrm{B}}$ 的前$(C-1)$个最大特征值的特征向量。注意到式(5.2.29)中 $\boldsymbol{S}_{\mathrm{B}}$ 的定义，$\boldsymbol{S}_{\mathrm{B}}$ 是由 C 项组成，每项的秩最大为 1，但这 C 项只有$(C-1)$项是独立的，故 $\boldsymbol{S}_{\mathrm{B}}$ 的秩至多为 $C-1$，因此，不为 0 的特征值至多为 $C-1$，所求的解向量 $\boldsymbol{w}_i$ 即为这些非 0 特征值所对应的特征向量。

类似于二分类问题，对于多分类问题，若类条件概率是高斯的，则可以证明：对于新的 $\boldsymbol{x}$，可计算

$$g_i=\ln p(C_i)-\frac{1}{2}\boldsymbol{m}_i^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{m}_i+\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{m}_i,\quad i=1,2,\cdots,C \tag{5.2.39}$$

若分类为 C_k，k 满足

$$k=\arg\max_i\{g_i\} \tag{5.2.40}$$

注意到，尽管可以利用 Fisher 判别分析直接进行分类，但本质上 Fisher 判别分析是一种预处理，将数据投影到具有最大分离性的低维空间。在二分类问题中，低维空间是一维的；在多分类问题中，低维空间是$(C-1)$维的。对于多分类问题，通过 Fisher 判别分析后，可以将数据集从 $\boldsymbol{D}=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1}^N$ 投影到 $\hat{\boldsymbol{D}}=\{(\hat{\boldsymbol{y}}_n,\boldsymbol{y}_n)\}_{n=1}^N$，若数据集是非高斯分布的，则可以用 $\hat{\boldsymbol{D}}$ 作为预处理后的数据集通过后续章节介绍的更高级的多分类器进行分类。

*5.2.2　感知机

感知机(Perceptron)是一种广义线性判别函数，用于二分类问题。感知机的判别函数输出为

$$\hat{y}(\boldsymbol{x};\boldsymbol{w})=\mathrm{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0) \tag{5.2.41}$$

其中，sgn(·)是符号函数，即

$$\mathrm{sgn}(x)=\begin{cases}1, & x\geqslant 0\\ -1, & x<0\end{cases} \tag{5.2.42}$$

感知机的输出只有±1，+1 表示类 C_1，−1 表示类 C_2。在感知机学习中，相应的样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^N$ 的标注 y_n 也取±1，而非 0 和 1。图 5.2.4 所示为感知机的结构框图，图中空心圆表示两方面的计算：元素求和与符号函数 sgn(·)。

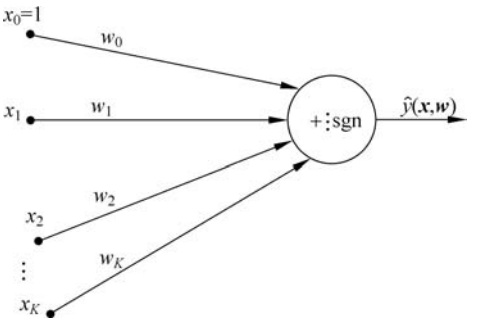

图 5.2.4　感知机的结构框图

感知机与线性判别函数一样，其判决面是 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0=0$ 的超平面。为了训练感知机，需要由样本集 $\boldsymbol{D}$ 学习参数 $\boldsymbol{w}$，为此，需要确定感知机的目标函数。

为了描述算法方便，类似于线性回归的表示，可将 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+w_0$ 表示为 $\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}$，这里 $\bar{\boldsymbol{w}}=[w_0,\boldsymbol{w}^{\mathrm{T}}]^{\mathrm{T}}$，$\bar{\boldsymbol{x}}$ 包含哑元 $x_0=1$ 作为第 1 个元素。如果样本集 $\boldsymbol{D}$ 是可分的，总可以找到一个

$\bar{\boldsymbol{w}}$，使所有样本都能被正确分类。即若 $\boldsymbol{x}_n$ 属于 C_1，则 $\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}_n>0$，且 $y_n=1$；若 $\boldsymbol{x}_n$ 属于 C_2，则 $\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}_n<0$，且 $y_n=-1$。可见，不管 $\boldsymbol{x}_n$ 属于哪一类，正确分类情况下都有 $\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}_n y_n>0$。在感知机的训练初始阶段，给出的 $\bar{\boldsymbol{w}}$ 将一部分样本正确分类，而将另一部分样本错误分类，将错误分类的样本记为集合 $\boldsymbol{D}_{\mathrm{E}}$，则对于一个样本 $\boldsymbol{x}_n\in\boldsymbol{D}_{\mathrm{E}}$，有 $\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}_n y_n<0$，因此，将感知机目标函数定义为

$$J(\bar{\boldsymbol{w}})=-\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_{\mathrm{E}}}\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}_n y_n \tag{5.2.43}$$

原理上，确定 $\bar{\boldsymbol{w}}$ 使感知机目标函数最小。仍然使用梯度算法，可得梯度为

$$\nabla_{\bar{\boldsymbol{w}}}J(\bar{\boldsymbol{w}})=-\sum_{\boldsymbol{x}_n\in\boldsymbol{D}_{\mathrm{E}}}\bar{\boldsymbol{x}}_n y_n \tag{5.2.44}$$

为了迭代实现方便，实际中采用 SGD 算法，即对于一个被错误分类的样本 $(\boldsymbol{x}_n,y_n)$，更新权系数为

$$\bar{\boldsymbol{w}}^{(k+1)}=\bar{\boldsymbol{w}}^{(k)}-\eta\nabla_{\bar{\boldsymbol{w}}}J_n(\bar{\boldsymbol{w}})=\bar{\boldsymbol{w}}^{(k)}+\eta\bar{\boldsymbol{x}}_n y_n \tag{5.2.45}$$

其中，上标 (k) 表示权系数更新的迭代序号；学习率 $0<\eta\leqslant 1$。

为了进行感知机的训练，首先给出权向量的初始值 $\bar{\boldsymbol{w}}^{(0)}$，从样本集中按照一定顺序取出一个样本 $(\boldsymbol{x}_n,y_n)$，判断其是否是被错误分类的样本，即是否满足 $\bar{\boldsymbol{w}}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_n y_n<0$，若不满足则跳过该样本，否则进行式(5.2.45)的权更新，直到所有样本都被正确分类，感知机收敛。

可以证明，在样本集满足线性可分的条件下，感知机是收敛的；在样本集不满足线性可分的条件下，感知机不收敛。本节介绍感知机的主要目的是对历史的回忆，感知机算法由 Frank Rosenblatt 于 1962 年提出，是最早的有影响力的机器学习算法之一，代表了神经网络的早期工作。目前人们很少再用感知机设计一个实际的分类器，本节也不再对其进行更详细的讨论，最后只是简要介绍一下感知机的“异或”问题。

异或是一种逻辑运算，输入特征向量 $\boldsymbol{x}$ 是二维的，仅有两个分量 x_1 和 x_2，每个分量只取 0 或 1，当 $x_1=x_2$ 时输出 $y=-1$(标准逻辑运算时 $y=0$，这里为了与感知机的输出一致，采用了 -1)，当 $x_1\neq x_2$ 时输出 $y=1$。因此，异或问题只有 4 个样本，即样本集为

$$\boldsymbol{D}=\{((0,0)^{\mathrm{T}},-1),((0,1)^{\mathrm{T}},1),((1,0)^{\mathrm{T}},1),((1,1)^{\mathrm{T}},-1)\} \tag{5.2.46}$$

样本如图 5.2.5(a)所示，可以看到，找不到一条直线可将两类样本分开，故这是线性不可分的，感知机无法将两类样本正确分类。

式(5.2.41)和图 5.2.4 所示的感知机是早期神经网络的一个代表，实际上这只是一个神经元的结构，尚未构成“网络”。这种简单线性单元无法解决类似“异或”这样简单的线性不可分问题，限制了其应用。解决这类问题有两种直接办法，一是多个神经元并联和级联组成多层感知网络，二是引入非线性变换。第一种办法中的多层感知机或神经网络将在第 9 章详细讨论。引入非线性变换的方法在第 4 章的回归模型中已经采用过，一种简单的方法是由 $\boldsymbol{x}$ 映射到一组基函数向量 $\boldsymbol{\varphi}(\boldsymbol{x})=[\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x}),\cdots,\varphi_{M-1}(\boldsymbol{x})]^{\mathrm{T}}$，将式(5.2.41)扩充为

$$\hat{y}(\boldsymbol{x};\boldsymbol{w})=\mathrm{sgn}[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+w_0] \tag{5.2.47}$$

将 $\boldsymbol{x}$ 映射到 $\boldsymbol{\varphi}(\boldsymbol{x})$ 空间，可将线性不可分样本集变换成 $\boldsymbol{\varphi}(\boldsymbol{x})$ 所表示空间中的线性可分集，从而用基函数感知机正确分类。

对于异或问题，可定义一个多项式函数 $\boldsymbol{\varphi}(\boldsymbol{x})$，其中

$$\begin{cases}\varphi_1(\boldsymbol{x})=2(x_1-0.5)\\ \varphi_2(\boldsymbol{x})=4(x_1-0.5)(x_2-0.5)\end{cases} \tag{5.2.48}$$

把样本 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}$ 映射成 $\boldsymbol{D}_\varphi=\{(\boldsymbol{\varphi}(\boldsymbol{x}_n),y_n)\}$，则式(5.2.46)的样本集映射为

$$\boldsymbol{D}_\varphi=\{((-1,1)^{\mathrm{T}},-1),((-1,-1)^{\mathrm{T}},1),((1,-1)^{\mathrm{T}},1),((1,1)^{\mathrm{T}},-1)\}$$

注意，映射后的样本示如图 5.2.5(b)所示，坐标轴分别记为 φ_1 和 φ_2，在$\boldsymbol{\varphi}(\boldsymbol{x})$各分量为坐标轴的空间内数据集是线性可分的，一条判决线是 $\varphi_2=0$，或$-\varphi_2>0$ 可判决为正样。对应到 $\boldsymbol{x}$ 空间，对应的判决线是 $\varphi_2(\boldsymbol{x})=4(x_1-0.5)(x_2-0.5)=0$，即对应了图 5.2.5(a)中的十字交叉虚线，$(x_1-0.5)(x_2-0.5)<0$ 为正样的判决区间，是由十字交叉虚线分割的 4 个区域的左上和右下区域，可见，这是正确的分类。

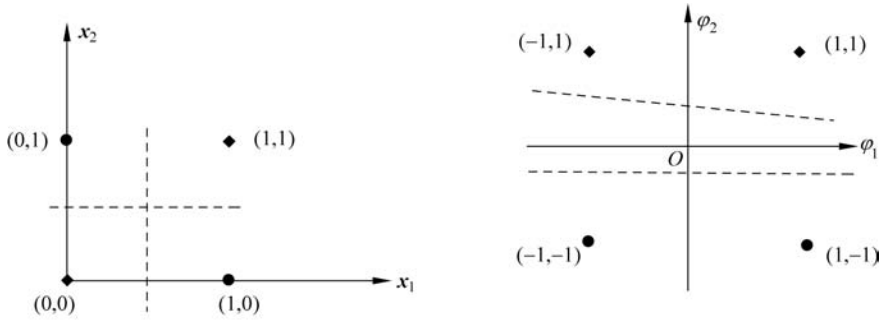

图 5.2.5　异或的示意图

通过映射$\boldsymbol{\varphi}(\boldsymbol{x})$，将样本映射到新空间，这些样本在新空间可能是线性可分的，由 $\varphi_i(\boldsymbol{x})$ 的线性组合构成分类器的判别函数，判决面在新空间是超平面，判决面映射到 $\boldsymbol{x}$ 空间则是非线性曲面。选择合理的映射$\boldsymbol{\varphi}(\boldsymbol{x})$，可容易地解决异或问题。

对于该问题，$\varphi_2=0$ 是最优的判决线，其他判决线(如图 5.2.5(b)的几条虚线)也可以做出正确分类，但分类性能都不如 $\varphi_2=0$ 判决线性能稳健(即泛化性能好)，即输入存在误差时可靠分类的能力。在将样本映射为 $\boldsymbol{D}_\varphi$ 后，训练式(5.2.4)的感知机，不能保证得到的判决线是 $\varphi_2=0$，由初始值和样本使用顺序，可能得到如图 5.2.5(b)虚线所示的判决线，这样的判决线映射到 $\boldsymbol{x}$ 空间，也不再是图 5.2.5(a)的十字虚线，而是有相应的变化，对标准输入样本仍可正确分类，但输入存在误差时则更容易出错。

对于异或的例子，感知机不能保证训练过程得到 $\varphi_2=0$ 的判决线，对于该问题，第 6 章讨论的 SVM 则可保证得到 $\varphi_2=0$ 作为判决线。一般在线性可分情况下，SVM 可得到泛化性能更好的分类器；在不可分情况下，SVM 也可以良好地工作，由于 SVM 在机器学习中的重要性，专门在第 6 章做详细讨论。

5.3　逻辑回归

逻辑回归(Logistic Regression)是一种典型的判别概率模型，本节讨论逻辑回归的目标函数和学习算法。对于中文名词“逻辑回归”稍做解释，尽管名称中有“回归”一词，逻辑回归却是一种基本的分类模型；另外，单词 Logistic 并不是“逻辑”的意思，为此，也有作者使用

音译名词“逻辑斯蒂回归”，考虑到“逻辑回归”一词在国内文献中使用已经很广泛，且用词简练，本书沿用“逻辑回归”的说法。作为判别概率模型，逻辑回归直接从样本中训练类后验概率表示。首先讨论二分类这一基本问题，然后推广到多分类问题。本节讨论的后验概率表示、模型的目标函数等内容，在第 9 章将直接推广到神经网络。

5.3.1 二分类问题的逻辑回归

首先考虑只有两类的情况，分别表示为 C_1 和 C_2，在数据集的样本 $(\boldsymbol{x}_n, y_n)$ 中，标注 $y_n=1$ 代表第 1 类，即 C_1，$y_n=0$ 代表第 2 类，即 C_2。通过一个数据集学习一个模型，当给出一个新的特征向量输入 $\boldsymbol{x}$ 时，由模型计算分类为 C_1 的后验概率 $p(C_1|\boldsymbol{x})$，则分类为 C_2 的后验概率为 $p(C_2|\boldsymbol{x})=1-p(C_1|\boldsymbol{x})$。由于 $0\leqslant p(C_1|\boldsymbol{x})\leqslant 1$，需要给出一种函数表示这种后验概率，这里定义一种 Logistic Sigmoid 函数（简称为 Sigmoid 函数）表示后验概率。Sigmoid 函数定义为

$$\sigma(a)=\frac{1}{1+\mathrm{e}^{-a}} \tag{5.3.1}$$

这里，用符号 $\sigma(\cdot)$ 表示 Sigmoid 函数，自变量 a 称为该函数的激活值，后面讨论实际分类模型时，将 a 表示为特征向量 $\boldsymbol{x}$ 的函数。

之所以用 $\sigma(\cdot)$ 函数表示后验概率，一个重要原因是其满足概率的性质，即 $0\leqslant\sigma(a)\leqslant 1$，$\sigma(a)$ 是 a 的单调函数，且 $\sigma(-\infty)=0$，$\sigma(\infty)=1$，$\sigma(0)=0.5$，$\sigma(x)$ 的图形如图 5.3.1 所示。

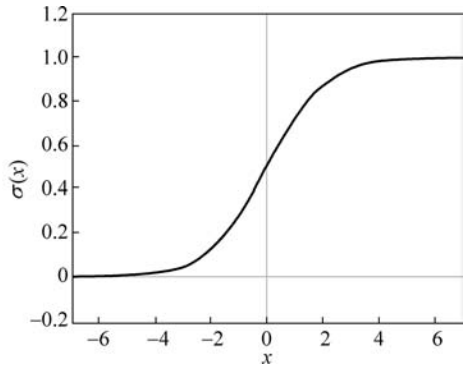

图 5.3.1 Sigmoid 函数的图形

Sigmoid 函数 $\sigma(a)$ 的另一个性质是处处光滑、处处可导，这便于数学处理。本节后续用到 $\sigma(a)$ 的两个基本性质，仅在下面列出，证明留作习题。

$$\sigma(-a)=1-\sigma(a) \tag{5.3.2}$$

$$\frac{\mathrm{d}\sigma(a)}{\mathrm{d}a}=\sigma(a)[1-\sigma(a)] \tag{5.3.3}$$

如前所述，可通过 $\sigma(a)$ 表示类后验概率，由于类后验概率是 $\boldsymbol{x}$ 的函数，可以定义激活值 a 是 $\boldsymbol{x}$ 的函数。在逻辑回归中，一般可选择 a 是 $\boldsymbol{x}$ 的线性函数，如线性回归关系，即

$$a(\boldsymbol{x})=\sum_{k=0}^{K}w_k x_k=\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{5.3.4}$$

其中，$x_0=1$ 为哑元，$\bar{\boldsymbol{x}}$ 为包含了哑元的特征向量；$\boldsymbol{w}$ 为待学习的参数向量。更一般地，可

以与第 4 章一样，定义基函数向量

$$\boldsymbol{\varphi}(\boldsymbol{x})=[1,\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x}),\cdots,\varphi_{M-1}(\boldsymbol{x})]^{\mathrm{T}} \tag{5.3.5}$$

则 a 表示为如下基函数线性回归。

$$a(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}) \tag{5.3.6}$$

由于线性回归相当于是$\boldsymbol{\varphi}(\boldsymbol{x})=\bar{\boldsymbol{x}}$ 的一个特例，本节后续讨论中，a 采用更一般的式(5.3.6)表示的基函数形式。

将式(5.3.6)代入 $\sigma(a)$的定义，用于表示 C_1 的后验概率 $p(C_1|\boldsymbol{x})$，即逻辑回归的输出为

$$\begin{aligned}\hat{y}(\boldsymbol{x},\boldsymbol{w})&=p(C_1\mid\boldsymbol{x})=\sigma(a)=\sigma[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})]\\&=\frac{1}{1+\exp[-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})]}\end{aligned} \tag{5.3.7}$$

由于式(5.3.7)也可以表示成 $f[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})]$的形式，这里 f 是非线性函数，故式(5.3.7)表示的逻辑回归输出也是一个广义线性模型。图 5.3.2 给出了逻辑回归的结构图，这里画出的是$\boldsymbol{\varphi}(\boldsymbol{x})=\bar{\boldsymbol{x}}$ 的情况，带有符号＋和 σ 的圆圈表示其先通过求和得到激活值 a，再经过激活函数 $\sigma(\cdot)$。这个圆表示了一种复合运算，后续可以看到，在一般神经网络的构成中，图 5.3.2 的结构表示神经网络中的一个神经元。

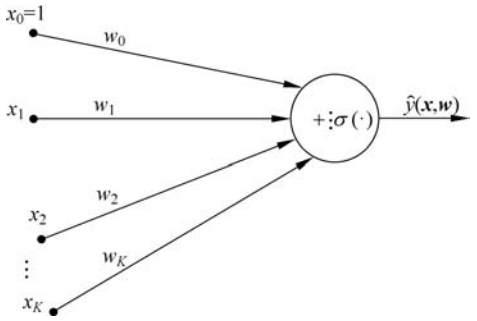

图 5.3.2　逻辑回归的结构图

显然，C_2 的后验概率 $p(C_2|\boldsymbol{x})$可表示为

$$p(C_2\mid\boldsymbol{x})=\frac{\exp[-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})]}{1+\exp[-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})]} \tag{5.3.8}$$

设已选定了$\boldsymbol{\varphi}(\boldsymbol{x})$，接下来通过给出的训练样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N}$ 学习参数向量 $\boldsymbol{w}$。由于已选定了基函数向量，故首先将样本集转换成以基函数向量表示的数据集，即通过转换得到数据集$\{[\boldsymbol{\varphi}(\boldsymbol{x}_n),y_n]\}_{n=1}^{N}=\{(\boldsymbol{\varphi}_n,y_n)\}_{n=1}^{N}$，这里用了简化符号$\boldsymbol{\varphi}_n=\boldsymbol{\varphi}(\boldsymbol{x}_n)$。为了叙述方便，避免多重括号和下标，也简化其他几个符号如下。

$$a_n=a(\boldsymbol{x}_n)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n \tag{5.3.9}$$

$$\hat{y}_n=\hat{y}(\boldsymbol{x}_n,\boldsymbol{w})=p(C_1|\boldsymbol{x}_n)=\sigma(a_n) \tag{5.3.10}$$

对于一个给定的样本$(\boldsymbol{\varphi}_n,y_n)$，把 y_n 作为一个二元随机变量，即伯努利分布，$y_n=1$ 的概率即属于 C_1 类别的概率为 $\hat{y}_n=\sigma(a_n)$，因此写出 y_n 的似然函数为

$$p(y_n\mid\boldsymbol{w})=(\hat{y}_n)^{y_n}(1-\hat{y}_n)^{1-y_n}$$

$$=[\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n)]^{y_n}[1-\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n)]^{1-y_n} \tag{5.3.11}$$

由于样本集的独立同分布假设，则全体标注 $\boldsymbol{y}=[y_1,y_2,y_3,\cdots,y_N]^{\mathrm{T}}$ 的似然函数为

$$p(\boldsymbol{y}\mid\boldsymbol{w})=\prod_{n=1}^{N}p(y_n\mid\boldsymbol{w})=\prod_{n=1}^{N}(\hat{y}_n)^{y_n}(1-\hat{y}_n)^{1-y_n} \tag{5.3.12}$$

以负对数似然函数作为损失函数，即

$$J(\boldsymbol{w})=-\ln p(\boldsymbol{y}\mid\boldsymbol{w})=-\sum_{n=1}^{N}y_n\ln\hat{y}_n+(1-y_n)\ln(1-\hat{y}_n) \tag{5.3.13}$$

式(5.3.13)称为交叉熵准则，尽管该准则是针对逻辑回归问题由最大似然原理导出的，后续可以看到对其他二分类的判别概率模型，交叉熵准则是一个通用准则，是建立在最大似然基础上的二分类问题的一般性准则。交叉熵的取名来自式(5.3.13)的形式，熵是用同一组概率进行计算的，而式(5.3.13)可理解为用了两组概率$\{y_n,1-y_n\}$和其近似值$\{\hat{y}_n,1-\hat{y}_n\}$，故称为交叉熵。最大似然原理对应着交叉熵的最小化，以此为目标得到权系数向量 $\boldsymbol{w}$ 的解。

式(5.3.13)是通过 $\hat{y}_n=\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n)$与 $\boldsymbol{w}$ 联系的，$J(\boldsymbol{w})$是 $\boldsymbol{w}$ 的非线性函数，没有闭式解。这里给出两种迭代求解方法：梯度法和牛顿法。对于规模不太大的逻辑回归问题，两种解法都是有效的，牛顿法一般收敛更快。

1. 随机梯度算法(SGD)

首先讨论梯度算法。直接在式(5.3.13)两侧对 $\boldsymbol{w}$ 求导，得到目标函数的梯度，计算如下

$$\begin{aligned}\nabla_w J(\boldsymbol{w})&=\frac{\partial J(\boldsymbol{w})}{\partial\boldsymbol{w}}\\&=-\sum_{n=1}^{N}y_n\frac{\partial\ln\hat{y}_n}{\partial\boldsymbol{w}}+(1-y_n)\frac{\partial\ln(1-\hat{y}_n)}{\partial\boldsymbol{w}}\\&=-\sum_{n=1}^{N}\frac{y_n}{\hat{y}_n}\frac{\partial\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n)}{\partial\boldsymbol{w}}-\frac{(1-y_n)}{1-\hat{y}_n}\frac{\partial\sigma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}_n)}{\partial\boldsymbol{w}}\\&=-\sum_{n=1}^{N}\frac{y_n}{\hat{y}_n}\hat{y}_n(1-\hat{y}_n)\boldsymbol{\varphi}_n-\frac{(1-y_n)}{1-\hat{y}_n}\hat{y}_n(1-\hat{y}_n)\boldsymbol{\varphi}_n\\&=\sum_{n=1}^{N}(\hat{y}_n-y_n)\boldsymbol{\varphi}_n\end{aligned} \tag{5.3.14}$$

注意，式(5.3.14)中从第 3 行到第 4 行用到了 $\sigma(\cdot)$函数的导数性质，即式(5.3.3)。式(5.3.14)是所有样本对梯度的贡献，若采样 SGD 算法，则单一样本对梯度的贡献为

$$\nabla_w J_n(\boldsymbol{w})=(\hat{y}_n-y_n)\boldsymbol{\varphi}_n=\{\sigma[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_n)]-y_n\}\boldsymbol{\varphi}(\boldsymbol{x}_n) \tag{5.3.15}$$

注意到随机梯度由一个误差项乘以基函数向量构成(不采用基函数时为特征向量)，而误差项为逻辑回归的输出与标注之间的误差，对于线性和广义线性模型，这是梯度表达式的一般形式。

有了梯度公式，给出权系数的初始值 $\boldsymbol{w}^{(0)}$，每次从样本集抽取一个样本$(\boldsymbol{\varphi}_n,y_n)$，每次样本抽取可以是随机的，故样本的标号 n 和权系数的迭代序号 k 使用不同的符号。SGD 算法的权系数向量更新为

$$\begin{aligned}\boldsymbol{w}^{(k+1)} &= \boldsymbol{w}^{(k)} - \eta \nabla_w J_n(\boldsymbol{w}^{(k)}) = \boldsymbol{w}^{(k)} - \eta(\hat{y}_n - y_n)\boldsymbol{\varphi}_n \\ &= \boldsymbol{w}^{(k)} - \eta(\sigma(\boldsymbol{w}^{(k)\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_n)) - y_n)\boldsymbol{\varphi}(\boldsymbol{x}_n) \end{aligned} \tag{5.3.16}$$

其中，η 为学习率。可以一次抽取 m 个小批量样本，将 SGD 推广到小批量算法，这种推广是直接的，具体公式可参考 4.1 节，此处不再重复。

2. 重加权最小二乘算法

SGD 是一阶迭代算法，为了更快的收敛速率，也可采用二阶迭代算法，二阶迭代算法的代表是牛顿法，牛顿法用了式(5.3.13)目标函数对 $\boldsymbol{w}$ 的二阶导数，即汉森(Hessian)矩阵，汉森矩阵的定义为

$$\boldsymbol{H} = \nabla_w^2 J(\boldsymbol{w}) = \left[\frac{\partial^2 J(\boldsymbol{w})}{\partial w_i \partial w_j}\right]_{M\times M} \tag{5.3.17}$$

为了导出 $\boldsymbol{H}$ 的表达式，将式(5.3.14)的结果写为如下矩阵与向量积的紧凑形式。

$$\nabla_w J(\boldsymbol{w}) = \sum_{n=1}^{N}(\hat{y}_n - y_n)\boldsymbol{\varphi}_n = \boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y}) \tag{5.3.18}$$

其中，$\hat{\boldsymbol{y}}$ 为由 $\hat{y}_n$ 构成的向量；$\boldsymbol{\Phi}$ 为基函数数据矩阵，其定义见式(4.3.6)，其第 n 行为$\boldsymbol{\varphi}_n^{\mathrm{T}}$。式(5.3.18)再次对 $\boldsymbol{w}$ 求导，得到如下矩阵(证明留作习题)。

$$\begin{aligned}\boldsymbol{H} = \nabla_w^2 J(\boldsymbol{w}) &= \nabla_w \sum_{n=1}^{N}(\hat{y}_n - y_n)\boldsymbol{\varphi}_n \\ &= \sum_{n=1}^{N}\hat{y}_n(1-\hat{y}_n)\boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^{\mathrm{T}} = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi}\end{aligned} \tag{5.3.19}$$

其中，$\boldsymbol{R}$ 为对角矩阵，$\boldsymbol{R} = \mathrm{diag}\{\hat{y}_1(1-\hat{y}_1), \cdots, \hat{y}_N(1-\hat{y}_N)\}$。有了汉森矩阵，则牛顿迭代算法为

$$\begin{aligned}\boldsymbol{w}^{(k+1)} &= \boldsymbol{w}^{(k)} - \boldsymbol{H}^{-1}\nabla_w J_n(\boldsymbol{w}^{(k)}) \\ &= \boldsymbol{w}^{(k)} - (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y}) \\ &= (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi})^{-1}[\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi}\boldsymbol{w}^{(k)} - \boldsymbol{\Phi}^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})] \\ &= (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{z}\end{aligned} \tag{5.3.20}$$

其中，向量 $\boldsymbol{z}$ 为

$$\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{w}^{(k)} - \boldsymbol{R}^{-1}(\hat{\boldsymbol{y}} - \boldsymbol{y}) \tag{5.3.21}$$

牛顿法的权更新公式如式(5.3.20)所示，这实际是一个加权最小二乘的计算公式，在每次迭代时，通过旧权系数 $\boldsymbol{w}^{(k)}$ 计算 $\boldsymbol{R}$ 和 $\boldsymbol{z}$，数据矩阵$\boldsymbol{\Phi}$ 不变。对角矩阵 $\boldsymbol{R}$ 相当于最小二乘的加权矩阵，每次迭代需要重新计算，故式(5.3.20)的迭代算法称为重加权最小二乘算法(Iterative Reweighted Least Squares，IRLS)。由于$\boldsymbol{\Phi}$ 不变，$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi}$ 中只有对角权矩阵$\boldsymbol{R}$ 每次迭代时变化，可以导出高效的求逆算法。

3. 正则化逻辑回归

正则化逻辑回归(Regularized Logistic Regression)可用于解决过拟合等问题，通过正则化控制模型复杂性。一种基本的权衰减正则化目标函数为

$$\begin{aligned}J(\boldsymbol{w}) &= -\ln p(\boldsymbol{y} \mid \boldsymbol{w}) + \frac{1}{2}\lambda \| \boldsymbol{w} \|_2^2 \\ &= -\sum_{n=1}^{N} y_n \ln\hat{y}_n + (1-y_n)\ln(1-\hat{y}_n) + \frac{1}{2}\lambda\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}\end{aligned} \tag{5.3.22}$$

在正则化目标函数下，可得到梯度向量为

$$\nabla_w J(\boldsymbol{w}) = \sum_{n=1}^{N} (\hat{y}_n - y_n)\,\boldsymbol{\varphi}_n + \lambda \boldsymbol{w} \tag{5.3.23}$$

如果使用随机梯度算法，则梯度向量为

$$\nabla_w J_n(\boldsymbol{w}) = (\hat{y}_n - y_n)\,\boldsymbol{\varphi}_n + \lambda \boldsymbol{w} \tag{5.3.24}$$

注意，式(5.3.24)和式(5.3.23)中的 λ 取值不同，因为都是超参数，一般需要通过交叉验证获得，故用了同一个符号。使用随机梯度式(5.3.24)得到正则化情况下的 SGD 权更新公式为

$$\boldsymbol{w}^{(k+1)} = (1 - \eta\lambda)\boldsymbol{w}^{(k)} - \eta(\hat{y}_n - y_n)\boldsymbol{\varphi}_n$$

5.3.2 多分类问题的逻辑回归

在表示多分类任务中，设共有 C 类，如式(5.1.2)所示用 C 维向量编码表示类型标注 $\boldsymbol{y}$，为方便重写如下。

$$\boldsymbol{y} = [0, \cdots, 0, 1, 0, \cdots, 0]^{\mathrm{T}} \tag{5.3.25}$$

$\boldsymbol{y}$ 中各分量 y_i 只取 0 或 1，且 $\sum_i y_i = 1$，即只有一个分量 $y_k = 1$ 表示第 k 类，其他 $y_i = 0, i \neq k$，用符号 C_k 表示第 k 类。

在多分类问题的逻辑回归方法中，对于给出的特征向量 $\boldsymbol{x}$，输出它被分类为每类的后验概率，用 C 维向量 $\hat{\boldsymbol{y}}$ 表示输出，其分量表示分类为 C_k 的后验概率，即

$$\hat{y}_k(\boldsymbol{x}) = p(C_k \mid \boldsymbol{x}) \tag{5.3.26}$$

为了用一种数学形式表示这种后验概率，即

$$0 \leqslant \hat{y}_k(\boldsymbol{x}) \leqslant 1, \quad \sum_{k=1}^{C} \hat{y}_k(\boldsymbol{x}) = 1 \tag{5.3.27}$$

推广式(5.3.1)的 Sigmoid 函数到 Softmax 函数。首先，针对每个 $\hat{y}_k$，定义一个相应的激活值，即

$$a_k(\boldsymbol{x}) = \boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{\varphi}(\boldsymbol{x}), \quad k = 1, 2, \cdots, C \tag{5.3.28}$$

其中，$\boldsymbol{w}_k$ 为一组待学习的参数向量，则 Softmax 函数定义为

$$\hat{y}_k(\boldsymbol{x}) = p(C_k \mid \boldsymbol{x}) = \frac{\exp[a_k(\boldsymbol{x})]}{\sum_{j=1}^{C} \exp[a_j(\boldsymbol{x})]}, \quad k = 1, 2, \cdots, C \tag{5.3.29}$$

可见，式(5.3.29)定义的 Softmax 函数满足式(5.3.27)的要求。先不考虑 $\boldsymbol{x}$，只考虑 $\hat{y}_k$ 作为 a_j 的函数，可以证明一个基本性质如下(留作习题)。

$$\frac{\partial \hat{y}_k}{\partial a_j} = \hat{y}_k(I_{kj} - \hat{y}_j) \tag{5.3.30}$$

其中

$$I_{kj} = \begin{cases} 1, & k = j \\ 0, & k \neq j \end{cases} \tag{5.3.31}$$

给出训练集 $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$，通过最大似然准则导出多分类逻辑回归的学习算法。首先，通过确定的基函数向量，将数据集变换为 $\{\boldsymbol{\varphi}(\boldsymbol{x}_n), \boldsymbol{y}_n\}_{n=1}^N = \{\boldsymbol{\varphi}_n, \boldsymbol{y}_n\}_{n=1}^N$。为了表示清楚，

给出一些符号的表示或缩写。标注 $\boldsymbol{y}_n$ 可以进一步写为 $\boldsymbol{y}_n=[y_{n1},y_{n2},\cdots,y_{nC}]^{\mathrm{T}}$，将 $\boldsymbol{y}_n$ 转置作为第 n 行构成标注值矩阵 $\boldsymbol{Y}=[y_{nk}]_{N\times C}$，对于一个样本 $\boldsymbol{x}_n$，对应的基函数向量简写为 $\boldsymbol{\varphi}_n$，逻辑回归的 Softmax 输出 $\hat{\boldsymbol{y}}_n$ 的一个分量记为 $\hat{y}_{nk}=\hat{y}_k(\boldsymbol{x}_n)=p(C_k\mid\boldsymbol{x}_n)$。对于一个样本$(\boldsymbol{\varphi}_n,\boldsymbol{y}_n)$，将 $\boldsymbol{y}_n$ 作为 C 维二元向量，其概率表示为

$$p(\boldsymbol{y}_n\mid\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C)=\prod_{k=1}^{C}(\hat{y}_{nk})^{y_{nk}} \tag{5.3.32}$$

由所有样本的 I.I.D 性质，所有样本的联合概率为

$$p(\boldsymbol{Y}\mid\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C)=\prod_{n=1}^{N}\prod_{k=1}^{C}(\hat{y}_{nk})^{y_{nk}} \tag{5.3.33}$$

式(5.3.33)中对 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C$ 的依赖是通过 $\hat{y}_{nk}$ 表现的。由于 $\boldsymbol{Y}$ 是已知的，式(5.3.33)是关于 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C$ 的似然函数。类似地，定义负对数似然函数作为损失函数，即

$$J(\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C)=-\ln p(\boldsymbol{Y}\mid\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_C)=-\sum_{n=1}^{N}\sum_{k=1}^{C}y_{nk}\ln\hat{y}_{nk} \tag{5.3.34}$$

以上函数存在约束条件

$$\sum_{k=1}^{C}y_{nk}=1 \tag{5.3.35}$$

式(5.3.34)是多分类情况下的交叉熵准则。为了采用梯度法求解各权系数向量 $\boldsymbol{w}_j$，需要求得式(5.3.34)对 $\boldsymbol{w}_j$ 的梯度，可以证明梯度为

$$\frac{\partial J}{\partial\boldsymbol{w}_j}=\sum_{n=1}^{N}(\hat{y}_{nj}-y_{nj})\boldsymbol{\varphi}_n,\quad j=1,\cdots,C \tag{5.3.36}$$

式(5.3.36)的证明如下。

$$\begin{aligned}\frac{\partial J}{\partial\boldsymbol{w}_j}&=-\frac{\partial}{\partial\boldsymbol{w}_j}\sum_{n=1}^{N}\sum_{k=1}^{C}y_{nk}\ln\hat{y}_{nk}=-\sum_{n=1}^{N}\sum_{k=1}^{C}y_{nk}\frac{1}{\hat{y}_{nk}}\frac{\partial\hat{y}_{nk}}{\partial a_{nj}}\frac{\partial a_{nj}}{\partial\boldsymbol{w}_j}\\&=-\sum_{n=1}^{N}\sum_{k=1}^{C}y_{nk}\frac{1}{\hat{y}_{nk}}\hat{y}_{nk}(I_{kj}-\hat{y}_{nj})\boldsymbol{\varphi}_n\\&=-\sum_{n=1}^{N}\Big(y_{nj}\boldsymbol{\varphi}_n-\sum_{k=1}^{C}y_{nk}\hat{y}_{nj}\boldsymbol{\varphi}_n\Big)=\sum_{n=1}^{N}(\hat{y}_{nj}-y_{nj})\boldsymbol{\varphi}_n\end{aligned}$$

以上证明从第 1 行到第 2 行用了式(5.3.30)，最后一个等号用了约束条件式(5.3.35)。

注意到，对于每个权向量 $\boldsymbol{w}_j$，梯度公式与二分类逻辑回归的梯度公式相同，如果采用 SGD 算法，每次随机抽取一个训练集样本，每个权向量用式(5.3.37)进行更新迭代。

$$\boldsymbol{w}_j^{(k+1)}=\boldsymbol{w}_j^{(k)}-\eta_j(\hat{y}_{nj}-y_{nj})\boldsymbol{\varphi}_n,\quad j=1,2,\cdots,C \tag{5.3.37}$$

对于多类逻辑回归算法，每个权向量的更新公式与只有一个向量的二分类逻辑回归是相同的。因此，对每个权向量 $\boldsymbol{w}_j$ 可导出相同的 IRLS 算法和正则化算法，扩展是直接的，不再详述。

5.4　朴素贝叶斯方法

ML09-分类学习-2

朴素贝叶斯方法是生成模型的一种，由于做了条件独立性假设，使其解得以简化。作为生成模型需要通过学习得到联合概率 $p(\boldsymbol{x},\boldsymbol{y})$，再通过贝叶斯公式得到分类的后验概率

$p(\boldsymbol{y}|\boldsymbol{x})$，因此，朴素贝叶斯方法属于贝叶斯学习方法中的一类，由于其简单性，放在本章中作为分类的生成模型的例子。

为了叙述简单，本节只讨论二分类的例子，故 y 只取 0 和 1。朴素贝叶斯的出发点是假设 y 作为条件下，$\boldsymbol{x}$ 的各分量统计独立。这里设 $\boldsymbol{x}$ 为 D 维向量，即

$$\boldsymbol{x}=[x_1,x_2,\cdots,x_D]^{\mathrm{T}} \tag{5.4.1}$$

则朴素贝叶斯的假设为

$$p(\boldsymbol{x}\mid y)=p(x_1,x_2,\cdots,x_D\mid y)=\prod_{i=1}^{D}p(x_i\mid y) \tag{5.4.2}$$

即在类型确定的条件下，特征向量的各分量是统计独立的。

在 3.6.1 节有向图模型的介绍时，曾以朴素贝叶斯假设作为有向图的一个实例说明了式(5.4.2)的假设尽管简单，但仍是一种合理的假设，作为复习，图 5.4.1 重画出朴素贝叶斯假设的有向图模型。因此，本节的内容也可以看作图模型从建模到参数学习再到推断和决策的一个简单但完整的实例。

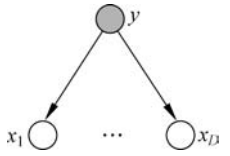

图 5.4.1 朴素贝叶斯假设的贝叶斯网络模型

在朴素贝叶斯方法中，常见的一种情况是假设 $\boldsymbol{x}$ 的每个分量取值是离散的，即 x_i 只可能取 M 个有限值。在以下介绍中，首先假设一种简单情况，x_i 只取两个值，即 $x_i\in\{0,1\}$，这种情况下，给出朴素贝叶斯算法的详细推导，这个推导过程对于理解概率原理在机器学习中的应用是一个很好的例子，最后再把结果推广到 x_i 可取 $M>2$ 个不同值的一般情况。

在 $x_i\in\{0,1\}$ 的简单情况下，设 $D=10$，$\boldsymbol{x}$ 的一个例子为 $\boldsymbol{x}=[0,1,1,0,1,0,0,1,0,0]^{\mathrm{T}}$。由于各分量的独立性，对于 $\boldsymbol{x}$ 的每个分量，分别定义两个参数为

$$\mu_{i|1}=\mu_{i|y=1}=p(x_i=1|y=1)=p(x_i=1|C_1) \tag{5.4.3}$$

$$\mu_{i|0}=\mu_{i|y=0}=p(x_i=1|y=0)=p(x_i=1|C_2) \tag{5.4.4}$$

由于 x_i 是一个伯努利随机变量，在条件 $y=k$，$k\in\{0,1\}$ 下 $\boldsymbol{x}$ 的各分量独立，故可得到类条件概率为

$$p(\boldsymbol{x}|y=k)=\prod_{i=1}^{D}\mu_{i|k}^{x_i}(1-\mu_{i|k})^{1-x_i},\quad k\in\{0,1\} \tag{5.4.5}$$

为了得到联合概率，需要 $y=k$，$k\in\{0,1\}$ 的先验概率，定义如下符号表示该先验概率。

$$\begin{cases}p(y=1)=\pi\\ p(y=0)=1-\pi\end{cases} \tag{5.4.6}$$

有了这些准备，可以得到 y 取不同值时，联合概率的表示分别为

$$\begin{aligned}p(\boldsymbol{x},y=1)&=p(y=1)\,p(\boldsymbol{x}|y=1)\\&=\pi\prod_{i=1}^{D}\mu_{i|1}^{x_i}(1-\mu_{i|1})^{1-x_i}\end{aligned} \tag{5.4.7}$$

和

$$\begin{aligned}p(\boldsymbol{x},y=0)&=p(y=0)\,p(\boldsymbol{x}|y=0)\\&=(1-\pi)\prod_{i=1}^{D}\mu_{i|0}^{x_i}(1-\mu_{i|0})^{1-x_i}\end{aligned} \tag{5.4.8}$$

由于 y 只有两个取值，把两个取值考虑进来（相当于 y 是一个伯努利变量），得到联合概率为

$$\begin{aligned}&p(\boldsymbol{x},y\mid\pi,\mu_{i\mid1},\mu_{i\mid0})\\&=\left[\pi\prod_{i=1}^{D}\mu_{i\mid1}^{x_i}(1-\mu_{i\mid1})^{1-x_i}\right]^{y}\times\left[(1-\pi)\prod_{i=1}^{D}\mu_{i\mid0}^{x_i}(1-\mu_{i\mid0})^{1-x_i}\right]^{1-y}\end{aligned}\tag{5.4.9}$$

在以上联合概率中，$\pi,\mu_{i\mid1},\mu_{i\mid0},i=1,2,\cdots,D$ 是联合概率的参数，共有 $2D+1$ 个参数待定。若给出训练样本 $\{\boldsymbol{x}_n,y_n\}_{n=1}^{N}$，通过训练样本学习得到所有参数 $\pi,\mu_{i\mid1},\mu_{i\mid0},i=1,2,\cdots,D$，则学习过程得到了联合概率。设训练样本是 I.I.D 的，且样本已知，故似然函数为

$$\begin{aligned}&p(\boldsymbol{y},\boldsymbol{X}\mid\pi,\mu_{i\mid1},\mu_{i\mid0})\\&=\prod_{n=1}^{N}\left[\pi\prod_{i=1}^{D}\mu_{i\mid1}^{x_{ni}}(1-\mu_{i\mid1})^{1-x_{ni}}\right]^{y_n}\left[(1-\pi)\prod_{i=1}^{D}\mu_{i\mid0}^{x_{ni}}(1-\mu_{i\mid0})^{1-x_{ni}}\right]^{1-y_n}\end{aligned}\tag{5.4.10}$$

将目标函数定义为负的对数似然，则

$$\begin{aligned}J(\pi,\mu_{i\mid1},\mu_{i\mid0})&=-\ln p(\boldsymbol{y},\boldsymbol{X}\mid\pi,\mu_{i\mid1},\mu_{i\mid0})\\&=-\sum_{n=1}^{N}y_n\ln\pi+(1-y_n)\ln(1-\pi)-\\&\sum_{n=1}^{N}y_n\sum_{i=1}^{D}x_{ni}\ln\mu_{i\mid1}+(1-x_{ni})\ln(1-\mu_{i\mid1})-\\&\sum_{n=1}^{N}(1-y_n)\sum_{i=1}^{D}x_{ni}\ln\mu_{i\mid0}+(1-x_{ni})\ln(1-\mu_{i\mid0})\end{aligned}\tag{5.4.11}$$

将式(5.4.11)分别对各参数求导并令导数为 0，得到各参数的解。令

$$\frac{\partial\ln p(\boldsymbol{y},\boldsymbol{X}\mid\pi,\mu_{i\mid1},\mu_{i\mid0})}{\partial\pi}=0$$

解得

$$\pi=\frac{1}{N}\sum_{n=1}^{N}y_n\tag{5.4.12}$$

令

$$\frac{\partial\ln p(y,\boldsymbol{X}\mid\pi,\mu_{i\mid1},\mu_{i\mid0})}{\partial\mu_{i\mid1}}=0$$

解得

$$\mu_{i\mid1}=\frac{\sum_{n=1}^{N}y_nx_{ni}}{\sum_{n=1}^{N}y_n},\quad i=1,2,\cdots,D\tag{5.4.13}$$

令

$$\frac{\partial\ln p(\boldsymbol{y},\boldsymbol{X}\mid\pi,\mu_{i\mid1},\mu_{i\mid0})}{\partial\mu_{i\mid0}}=0$$

解得

$$\mu_{i|0}=\frac{\sum_{n=1}^{N}(1-y_n)x_{ni}}{\sum_{n=1}^{N}(1-y_n)},\quad i=1,2,\cdots,D \tag{5.4.14}$$

当得到了参数 $\pi,\mu_{i|1},\mu_{i|0},i=1,2,\cdots,D$ 后，则式(5.4.5)～式(5.4.9)的各种概率都已得到，包括类的先验概率、类条件概率和联合概率。除此之外，由这组参数还可以得到在给定一个新的特征输入 $\boldsymbol{x}$ 时分类的后验概率。利用贝叶斯公式，得到 $\boldsymbol{x}$ 被分为 C_1 类的后验概率为

$$\begin{aligned}p(C_1\mid\boldsymbol{x})=p(y=1\mid\boldsymbol{x})&=\frac{p(\boldsymbol{x}\mid y=1)\,p(y=1)}{p(\boldsymbol{x})}\\&=\frac{\left[\prod_{i=1}^{D}p(x_i\mid y=1)\right]p(y=1)}{\left[\prod_{i=1}^{D}p(x_i\mid y=1)\right]p(y=1)+\left[\prod_{i=1}^{D}p(x_i\mid y=0)\right]p(y=0)}\\&=\frac{\pi\prod_{i=1}^{D}\mu_{i|1}^{x_i}(1-\mu_{i|1})^{1-x_i}}{\pi\prod_{i=1}^{D}\mu_{i|1}^{x_i}(1-\mu_{i|1})^{1-x_i}+(1-\pi)\prod_{i=1}^{D}\mu_{i|0}^{x_i}(1-\mu_{i|0})^{1-x_i}}\end{aligned} \tag{5.4.15}$$

显然，$\boldsymbol{x}$ 被分为 C_2 类的后验概率为

$$p(C_2\mid\boldsymbol{x})=p(y=0\mid\boldsymbol{x})=1-p(y=1\mid\boldsymbol{x}) \tag{5.4.16}$$

本节针对相对简单的情况：二分类且 $\boldsymbol{x}$ 的每个分量只有两个取值，详细推导了朴素贝叶斯算法。下面针对两方面问题，再做一些讨论。

1. 拉普拉斯平滑

在以上讨论中，已经得到当有一个新的输入时，分类为 C_1 的后验概率，看上去问题得以圆满解决。但在实际应用中，式(5.4.15)可能会遇到 0/0 困境。注意到，若在估计的参数中，有一个 $\mu_{i|1}=0$ 和一个 $\mu_{j|0}=0$ 存在，这里 i 和 j 可以表示任意两个分量，则代入式(5.4.15)有

$$p(y=1\mid\boldsymbol{x})=\frac{0}{0} \tag{5.4.17}$$

这是一个无解的情况，无法做出判决。在 $\boldsymbol{x}$ 的维度 D 很大，训练样本集规模有限时，这是很可能发生的情况，这是由最大似然估计的局限性所致。

第 2 章曾提到，在离散事件的情况下，最大似然估计的概率就是样本集中一类事件出现的比例，最大似然的概率估计是符合对概率的“频率”解释的。在本节中也是如此，可以看到式(5.4.12)～式(5.4.14)都给出了所估计概率的按“比例”分配的解释。例如，式(5.4.12)中，分子统计了训练样本集中标注为 1 的样本数目 N_1，$\pi=p(C_1)=N_1/N$ 就是样本集中标注为 C_1 的样本所占比例；式(5.4.13)中 $\mu_{i|1}$ 的公式复杂一些，但也代表了同样的含义，它的分母为 N_1，即样本集中所有标注为 C_1 的样本总数，而分子的求和项 $y_n x_{ni}$ 的含义是标注为 C_1 的同时 $x_{ni}=1$ 的样本贡献一个计数 1，因此分子的含义是在所有标注为 C_1 的样本中，$\boldsymbol{x}$ 的第 i 分量为 1 的样本数之和，$\mu_{i|1}$ 表示了在所有标注为 C_1 的样本中 $\boldsymbol{x}$ 的第 i 分

量为1的样本所占的比例。无须重复，$\mu_{i|0}$的意义是相同的。在本问题中π的估计没有问题，样本数足够估计π的，但是一些$\mu_{i|1}$或$\mu_{i|0}$可能估计为0。例如，若$D=1000$，样本总数为10 000，则很有可能在样本集中$\boldsymbol{x}$的某个分量从来没有出现过$x_{ni}=1$的情况，此时$\mu_{i|1}$估计为0。

解决这一问题的系统化的方法是采用贝叶斯估计替代最大似然估计，这样做需要给出每个参数一个合理的先验概率，这并不总是容易做到的。一种改善最大似然零概率估计的简单方法是拉普拉斯平滑。拉普拉斯平滑用于改善离散随机变量的概率估计。设一个离散随机变量$\boldsymbol{x}$，其仅取k个可能的值，即$x\in\{1,2,\cdots,k\}$，给出一组样本$\{x_1,x_2,\cdots,x_m\}$，需要估计概率

$$\mu_i=p(x=i),\quad i=1,2,\cdots,k$$

则标准最大似然估计为

$$\mu_i=\frac{\sum_{n=1}^{m}I(x_n=i)}{m} \tag{5.4.18}$$

其中，$I(z)$为示性函数，z为真时$I(z)=1$，z非真时$I(z)=0$。最大似然估计μ_i统计的就是$x_n=i$在样本集中的比例，而拉普拉斯平滑做了如下修改。

$$\mu_i=\frac{1+\sum_{n=1}^{m}I(x_n=i)}{m+k} \tag{5.4.19}$$

拉普拉斯平滑作为一种简单的修改，保证$\mu_i>0$的同时满足概率的基本约束$\sum_{i=1}^{k}\mu_i=1$。

我们可以这样理解拉普拉斯平滑：对待估计的随机变量取值施加等概率的先验分布，但是不显式地使用贝叶斯框架，而是虚拟地产生k个附加样本。由于假设等概率先验，假想可将集合$\{1,2,\cdots,k\}$中每个值各取一次作为一个虚拟样本，将这k个虚拟样本加到样本集中，构成一个增广样本集，则拉普拉斯变换相当于用这一增广样本集做的最大似然估计。可见，拉普拉斯平滑相当于一种“弱”贝叶斯方法，或相当于一种正则化的最大似然方法。在机器学习中，直接对样本集进行增广是一种常用的正则化技术。

将拉普拉斯平滑应用于$\mu_{i|1}$和$\mu_{i|0}$估计式(5.4.13)和式(5.4.14)，注意在这两个公式中，相当于$k=2$。$\mu_{i|1}$和$\mu_{i|0}$的估计公式修改为

$$\mu_{i|1}=\frac{1+\sum_{n=1}^{N}y_nx_{ni}}{2+\sum_{n=1}^{N}y_n},\quad i=1,2,\cdots,D \tag{5.4.20}$$

$$\mu_{i|0}=\frac{1+\sum_{n=1}^{N}(1-y_n)x_{ni}}{2+\sum_{n=1}^{N}(1-y_n)},\quad i=1,2,\cdots,D \tag{5.4.21}$$

这样就解决了式(5.4.17)所表示的问题。

2. 朴素贝叶斯模型的更一般情况

为了把前述的简单情况下的朴素贝叶斯算法推广到更一般情况，通过示性函数，给出式(5.4.12)～式(5.4.14)的一种等价表示，用示性函数表示的参数估计公式为

$$\pi=\frac{1}{N}\sum_{n=1}^{N}I(y_n=1) \tag{5.4.22}$$

$$\mu_{i|1}=\frac{\sum_{n=1}^{N}I(y_n=1\cap x_{ni}=1)}{\sum_{n=1}^{N}I(y_n=1)} \tag{5.4.23}$$

$$\mu_{i|0}=\frac{\sum_{n=1}^{N}I(y_n=0\cap x_{ni}=1)}{\sum_{n=1}^{N}I(y_n=0)} \tag{5.4.24}$$

其中，符号∩表示两个条件的交，即两个条件均满足才为真。

讨论更一般的情况，首先是多分类问题，分类任务有 C 个类型，可用 C_k，$k=1,2,\cdots,C$ 表示每类，与前面讨论一致，用 C 维向量编码表示类型 $\boldsymbol{y}$，由于 $\boldsymbol{y}$ 中只有一个分量为 1，其余为 0，故 $y_k=1$ 等价于 C_k。类似于二分类情况，可以定义每类的先验概率为

$$\pi_k=p(C_k)=p(y_k=1) \tag{5.4.25}$$

对特征向量 $\boldsymbol{x}$ 也推广到更一般化，$\boldsymbol{x}$ 的每个分量仍是离散的，但不再局限于只取两个可能值，设 $\boldsymbol{x}$ 的第 i 个分量可取 M_i 个值，不失一般性，设 $x_i\in\{0,1,\cdots,M_i-1\}$。给出一个概率参数的更一般的符号为

$$\mu_{i|k}^{(l)}=p(x_i=l\,|\,y_k=1)=p(x_i=l\,|\,C_k),$$
$$l=0,1,\cdots,M_i-1,\quad k=1,\cdots,C,\quad i=1,2,\cdots,D \tag{5.4.26}$$

给出训练样本 $\{\boldsymbol{x}_n,\boldsymbol{y}_n\}_{n=1}^{N}$，把式(5.4.22)～式(5.4.24)推广到这种更一般的情况，相应参数估计公式为

$$\pi_k=\frac{1}{N}\sum_{n=1}^{N}I(y_{nk}=1),\quad k=1,2,\cdots,C \tag{5.4.27}$$

$$\mu_{i|k}^{(l)}=\frac{\sum_{n=1}^{N}I(y_{nk}=1\cap x_{ni}=l)}{\sum_{n=1}^{N}I(y_{nk}=1)},$$
$$l=0,1,\cdots,M_i-1,\quad k=1,2,\cdots,C,\quad i=1,2,\cdots,D \tag{5.4.28}$$

学习了这些参数后，可以得到联合概率。如果主要用于分类，则可以得到类后验概率，由贝叶斯公式和朴素贝叶斯的假设，给出一个新的特征向量 $\boldsymbol{x}$，得到类后验概率为

$$p(C_k\,|\,\boldsymbol{x})=\frac{p(\boldsymbol{x}\,|\,C_k)p(C_k)}{\sum_{j=1}^{C}p(\boldsymbol{x}\,|\,C_j)p(C_j)}$$

$$=\frac{\pi_k \prod_{i=1}^{D} p(x_i \mid C_k)}{\sum_{j=1}^{C} \pi_j \prod_{i=1}^{D} p(x_i \mid C_j)}, \quad k=1,2,\cdots,C \tag{5.4.29}$$

在式(5.4.29)中，若 $\boldsymbol{x}$ 给定了，则 x_i 的取值就给定了，若 $x_i=l$，则 $p(x_i \mid C_k)=\mu_{i \mid k}^{(l)}$ 就确定了，用式(5.4.29)可以得到所有类的后验概率，并通过决策过程完成分类。由于式(5.4.29)的分母对所有 C_k 是相等的，若所有类的分类错误代价是相等的，则只需要分子部分的比较即可决定分类结果，即可分类为

$$C_{k_o}=\arg\max_{C_k}\left[\pi_k \prod_{i=1}^{D} p(x_i \mid C_k)\right] \tag{5.4.30}$$

其中，C_{k_o} 为分类结果。

最后通过实例说明朴素贝叶斯方法的应用。朴素贝叶斯方法用于很多实际分类问题中，一个典型的应用是用于垃圾邮件检测。收到一封电子邮件后，通过一个检测器判断其是否是垃圾邮件。要完成垃圾邮件检测，需要收集大量实际邮件，并由人工标注其是否为垃圾邮件，设垃圾邮件为 C_1 类，正常邮件为 C_2 类。为了用一个向量 $\boldsymbol{x}$ 表示一封邮件，预先构造一个词汇表，按照顺序，$\boldsymbol{x}$ 的一个分量表示一个词汇。若 x_i 为 0/1 变量，则 $x_i=1$ 表示对应的词汇存在于该邮件中，在这种应用中，$\boldsymbol{x}$ 的维数 D 与词汇表中的词汇数相等，可能是一个很大的数，如 $D=5000$。若只选择使用关键词汇表，则 D 可取更小的值，如 $D=200$。

朴素贝叶斯假设下，描述 $\boldsymbol{x}$ 的概率参数为 $2D$，若不使用朴素贝叶斯的假设，则参数为 2^{2D}，这是难以存储的参数量。对收集的邮件完成标注和特征向量抽取后，得到样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N}$，通过样本集得到朴素贝叶斯分类器的所有参数，当一个新邮件收到后，通过同样的方式检查邮件中的词汇，得到特征向量，则由式(5.4.15)计算该邮件是垃圾邮件的后验概率并做出判决。

当 x_i 可取多值时，可用 x_i 表示一封邮件中某词汇出现的次数(若超过 M_i 则限定为可表示的最大值)，这样可得到更丰富的信息。利用朴素贝叶斯方法可有效判别垃圾邮件，对于大多数邮箱服务器可满足要求。但这类算法仅通过词汇级知识，若通过深度学习进行文法分析和关联知识分析，则可以更准确地检测出垃圾邮件。

*5.5　机器学习理论简介

在第 4 章介绍了回归算法后，以回归问题为例，讨论了偏和方差的折中问题，本节将以分类问题为例，讨论机器学习的另一个理论问题——泛化界。偏和方差的折中与泛化界是机器学习理论中关注的两个基本问题，都是关于泛化误差的，两者之间也有密切的联系。

在机器学习模型的训练过程中，一般只有一组训练集，通过训练误差的最小化学习到一个模型，但真正关心的是泛化误差，即对不存在于训练集中的新样本，模型的预测性能如何？所以我们要关心一个基本问题：训练误差和泛化误差之间有多大的差距？机器学习的概率近似正确(Probably Approximately Correct，PAC)理论对这个问题进行了研究。下面对该理论给出一个极为简要的介绍。

本节以二分类问题为例，讨论 PAC 理论的一些最基本概念和结论。假设所面对的样本

可表示为$(\boldsymbol{x},y)$,$\boldsymbol{x}$ 为输入特征向量$\boldsymbol{x}\in\mathcal{X}$,$\mathcal{X}$表示输入空间,$y\in\{0,1\}$表示类型,$(\boldsymbol{x},y)$满足概率分布 $p_{\mathcal{D}}(\boldsymbol{x},y)$,简写为 $p_{\mathcal{D}}$,故可用$(\boldsymbol{x},y)\sim p_{\mathcal{D}}$ 表示样本服从的分布。从 $p_{\mathcal{D}}$ 中采样得到满足独立同分布(I.I.D)的训练样本集为

$$\boldsymbol{D}=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{N} \tag{5.5.1}$$

其中,每个样本$(\boldsymbol{x}_i,y_i)\sim p_{\mathcal{D}}$。

用机器学习理论常用的术语,将前文所述的一个机器学习模型称为一个假设 h,$h(\boldsymbol{x})$输出$\{0,1\}$表示类型,即完成映射 $h:\mathcal{X}\to\{0,1\}$。在一个机器学习过程中,所有可能选择的假设构成一个假设空间$\mathcal{H}$。对于任意假设 $h\in\mathcal{H}$,其在训练样本集上的分类错误率定义为训练误差或经验风险,经验风险表示为

$$\hat{R}(h)=\frac{1}{N}\sum_{i=1}^{N}I[h(\boldsymbol{x}_i)\neq y_i] \tag{5.5.2}$$

经验风险表示假设 h 在训练集上的分类错误率。式(5.5.2)中 $I(\cdot)$为示性函数。可定义一般的泛化误差为

$$R(h)=P_{(\boldsymbol{x},y)\sim p_{\mathcal{D}}}[h(\boldsymbol{x})\neq y] \tag{5.5.3}$$

对于任意样本$(\boldsymbol{x},y)\sim p_{\mathcal{D}}$,不管其是否存在于训练集中,泛化误差均表示其总体统计意义上的误分类率。

若不考虑可实现性,理论上,希望学习到的假设是从$\mathcal{H}$空间找到使泛化误差最小的假设,即

$$h^{*}=\arg\min_{h\in\mathcal{H}}R(h) \tag{5.5.4}$$

实际上,由于无法准确获得 $p_{\mathcal{D}}(\boldsymbol{x},y)$,故无法通过泛化误差优化获得最优假设,实际上总是通过经验风险最小化(Empirical Risk Minimization,ERM)得到一个假设,即

$$\hat{h}=\arg\min_{h\in\mathcal{H}}\hat{R}(h) \tag{5.5.5}$$

我们关心的一个理论问题:通过 ERM 得到的假设 $\hat{h}$ 与真正的泛化误差最小的 h^{*} 之间的泛化误差差距有多大?即 $R(h^{*})$与 $R(\hat{h})$差距有多大?

在继续讨论之前,首先通过一个例子进一步理解以上概念。

例 5.5.1 一个假设空间的例子。在 5.2 节的感知机的例子中,为了与本节分类输出用$\{0,1\}$表示相符,将分类假设修改为

$$h(\boldsymbol{x})=I(\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}\geqslant 0) \tag{5.5.6}$$

这里,$\bar{\boldsymbol{x}}$ 包含了哑元,$\bar{\boldsymbol{w}}$ 包含了偏置系数,是$(K+1)$维参数向量。式(5.5.6)是一个假设,则假设空间为

$$\mathcal{H}=\{h_{\bar{\boldsymbol{w}}}\,|\,h_{\bar{\boldsymbol{w}}}(\boldsymbol{x})=I(\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}}\geqslant 0),\quad \bar{\boldsymbol{w}}\in\mathbf{R}^{K+1}\} \tag{5.5.7}$$

$\mathcal{H}$表示 K 维向量空间中的所有线性分类器集合,其中 $\mathbf{R}^{K+1}$ 为 $K+1$ 维实数集合。由于不同 $\bar{\boldsymbol{w}}$ 构成$\mathcal{H}$的不同成员,故式(5.5.5)可具体化为

$$\hat{h}=\arg\min_{\bar{\boldsymbol{w}}\in\mathbf{R}^{K+1}}\hat{R}(h_{\bar{\boldsymbol{w}}}) \tag{5.5.8}$$

$\hat{h}$ 为 ERM 意义下的假设,一般不能通过学习得到 $h_{\bar{\boldsymbol{w}}}^{*}$。

实际上,感知机的目标函数式(5.2.43)是式(5.5.2)的经验风险的一种逼近,故训练得

到的感知机是对 $\hat{h}$ 的一种逼近。

对逻辑回归也可做类似理解，同样，其目标函数交叉熵也是式(5.5.2)经验风险函数的一种近似。

为了研究 $R(h^*)$ 与 $R(\hat{h})$ 的关系，给出以下引理。

引理 5.5.1　设 $Z_1, Z_2, \cdots, Z_N$ 为 N 个独立同分布的随机变量，均服从伯努利分布，且 $P(Z_i=1)=\mu$，定义样本均值为 $\hat{\mu}=\frac{1}{N}\sum_{i=1}^{N}Z_i$，令 $\varepsilon>0$ 为一个固定值，则

$$P(|\mu-\hat{\mu}|>\varepsilon)\leqslant 2\exp(-2\varepsilon^2 N) \tag{5.5.9}$$

该引理说明对于独立同分布样本，当 N 充分大时，对于给定的 $\varepsilon>0$，均值估计和实际概率值之差大于 ε 的概率是很小的。

接下来，对于$\mathcal{H}$有限的情况，利用引理 5.5.1 导出训练误差和泛化误差的误差界，然后将结论推广到$\mathcal{H}$无限的情况。

5.5.1　假设空间有限时的泛化误差界

首先假设空间$\mathcal{H}$是有限的，即$\mathcal{H}=\{h_1, h_2, \cdots, h_K\}$，假设空间成员数目 $K=|\mathcal{H}|$ 可能很大，但是有限的。例如 5.4 节介绍的朴素贝叶斯方法，当特征分量取离散值时，其假设空间是有限的。若每个特征变量取有限值，则第 7 章介绍的决策树假设空间也是有限的。对于例 5.5.1，若 $\bar{w}$ 取值为实数，则假设空间是无限的，但若 $\bar{w}$ 的每个分量是由有限位二进制表示的数值，则其假设空间是有限的(详细讨论见例 5.5.2)。首先讨论$\mathcal{H}$有限的情况。

可从$\mathcal{H}$空间选择一个固定的假设 h_k，利用引理 5.5.1 容易得到对于 h_k，其训练误差和泛化误差的关系。为此定义一个随机变量，对于$(\boldsymbol{x}, y)\sim p_{\mathcal{D}}$，定义

$$Z=I[h_k(\boldsymbol{x})\neq y] \tag{5.5.10}$$

即当 h_k 对样本$(\boldsymbol{x}, y)$不能正确分类时 $Z=1$。对式(5.5.1)所示的每个样本有 $Z_i=I[h_k(\boldsymbol{x}_i)\neq y_i]$，显然，$h_k$ 的训练误差为

$$\hat{R}(h_k)=\frac{1}{N}\sum_{i=1}^{N}Z_j \tag{5.5.11}$$

对比引理 5.5.1，Z_i 是 I.I.D 的伯努利随机变量，$\hat{R}(h_k)$是对 $R(h_k)$ 的样本均值估计，则由式(5.5.9)直接得到

$$P[|R(h_k)-\hat{R}(h_k)|>\varepsilon]\leqslant 2\exp(-2\varepsilon^2 N) \tag{5.5.12}$$

以上是对于一个固定的 h_k，泛化误差和训练误差之差(绝对值)大于 ε 的概率。对于给定 ε，若样本数 N 足够大，则泛化误差与训练误差相差大于 ε 的概率很小。利用这个关系，可导出一个更一般的结果。因此，定义事件 A_k 为 $|R(h_k)-\hat{R}(h_k)|>\varepsilon$，则有 $P(A_k)\leqslant 2\exp(-2\varepsilon^2 N)$，利用概率性质，至少存在一个 h(表示为 $\exists h$)，其 $|R(h)-\hat{R}(h)|>\varepsilon$ 的概率为

$$\begin{aligned}P[\exists h\in\mathcal{H}, |R(h)-\hat{R}(h)|>\varepsilon]&=P(A_1\cup A_2\cup\cdots\cup A_K)\\&\leqslant\sum_{k=1}^{|\mathcal{H}|}P(A_k)\leqslant\sum_{k=1}^{|\mathcal{H}|}2\exp(-2\varepsilon^2 N)\end{aligned}$$

$$=2|\mathcal{H}|\exp(-2\varepsilon^2 N) \tag{5.5.13}$$

由于是概率值，由互补性，式(5.5.13)的等价表示为对于所有 h，有

$$P[|R(h)-\hat{R}(h)|\leqslant\varepsilon,\quad \forall h\in\mathcal{H}]\geqslant 1-2|\mathcal{H}|\exp(-2\varepsilon^2 N) \tag{5.5.14}$$

在式(5.5.14)中，令

$$\delta=2|\mathcal{H}|\exp(-2\varepsilon^2 N) \tag{5.5.15}$$

式(5.5.14)有丰富的内涵，其中有 3 个量：δ,ε,N，以下从几个方面讨论式(5.5.14)的含义，并讨论这 3 个量的关系。

(1) 式(5.5.14)可重写为

$$P[|R(h)-\hat{R}(h)|\leqslant\varepsilon]\geqslant 1-\delta,\quad \forall h\in\mathcal{H} \tag{5.5.16}$$

对于给定的 ε，所有假设 $\forall h\in\mathcal{H}$ 都以不小于 $1-\delta$ 的概率满足 $|R(h)-\hat{R}(h)|\leqslant\varepsilon$，即泛化误差和训练误差之差不大于界 ε。这里 $1-\delta$ 是一个置信概率，当 N 很大时，δ 很小，以很高的概率满足 $|R(h)-\hat{R}(h)|\leqslant\varepsilon$。

(2) 假设空间的元素数目 $|\mathcal{H}|$ 是确定的，若给出 ε 和 δ，则可得到满足以 $1-\delta$ 为概率达到 $|R(h)-\hat{R}(h)|\leqslant\varepsilon$ 所需的样本数目。固定 δ,ε，从式(5.5.15)反解 N 为

$$N=\frac{1}{2\varepsilon^2}\ln\frac{2|\mathcal{H}|}{\delta} \tag{5.5.17}$$

由式(5.5.15)，N 增大，δ 减小，故可将式(5.5.17)看作满足 δ,ε 约束的最小样本数。故对于给定 δ,ε，样本数可取为

$$N\geqslant\frac{1}{2\varepsilon^2}\ln\frac{2|\mathcal{H}|}{\delta} \tag{5.5.18}$$

(3) 在式(5.5.15)中，固定 δ,N，解得

$$\varepsilon=\sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{H}|}{\delta}} \tag{5.5.19}$$

这给出了式(5.5.16)另一种解释，对于给定的 δ,N，误差界满足

$$|R(h)-\hat{R}(h)|\leqslant\sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{H}|}{\delta}} \tag{5.5.20}$$

将以上解释总结为定理 5.5.1。

定理 5.5.1 对于假设空间 $\mathcal{H}$，固定 δ,N，则以概率不小于 $1-\delta$，泛化误差与训练误差满足

$$|R(h)-\hat{R}(h)|\leqslant\sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{H}|}{\delta}}$$

或固定 δ,ε，若样本数目取

$$N\geqslant\frac{1}{2\varepsilon^2}\ln\frac{2|\mathcal{H}|}{\delta}$$

则以概率不小于 $1-\delta$ 满足 $|R(h)-\hat{R}(h)|\leqslant\varepsilon$。

以上结论对于假设空间中的任意假设 $\forall h\in\mathcal{H}$ 均成立。我们更感兴趣的一个问题是，对于以经验风险最小化学习得到的假设 $\hat{h}$ 和理论上泛化误差最小对应的 h^* 之间的泛化误差

的比较，由以上结果，若对 $\forall h \in \mathcal{H}$ 有 $|R(h)-\hat{R}(h)| \leqslant \varepsilon$，则

$$R(h) \leqslant \hat{R}(h)+\varepsilon \tag{5.5.21}$$

对于 $\hat{h}$，可得到如下不等式。

$$R(\hat{h}) \leqslant \hat{R}(\hat{h})+\varepsilon \leqslant \hat{R}(h^*)+\varepsilon \leqslant R(h^*)+2\varepsilon \tag{5.5.22}$$

式(5.5.22)中第 1 个不等式只是将 $\hat{h}$ 代入式(5.5.21)；第 2 个不等式用了 $\hat{R}(\hat{h}) \leqslant \hat{R}(h^*)$，这是因为 $\hat{h}$ 是经验误差最小的假设；第 3 个不等式对 h^* 再次使用式(5.5.21)。式(5.5.22)的结论是 ERM 学习得到的假设 $\hat{h}$ 与泛化误差最优的 h^*，其泛化误差之差不大于 2ε。将该结论总结为重要的定理 5.5.2。

定理 5.5.2　对于假设空间 $\mathcal{H}$，固定 δ，N，则以概率不小于 $1-\delta$，泛化误差满足

$$R(\hat{h}) \leqslant \min_{h \in \mathcal{H}} R(h)+2\sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{H}|}{\delta}} \tag{5.2.23}$$

或固定 δ，ε，若样本数目取

$$N \geqslant \frac{1}{2\varepsilon^2}\ln\frac{2|\mathcal{H}|}{\delta}$$

则以概率不小于 $1-\delta$ 满足 $R(\hat{h}) \leqslant \min_{h \in \mathcal{H}} R(h)+2\varepsilon$。

由两个定理可见，若固定 δ，N，式(5.5.19)计算了一个界 ε，对于任意 $h \in \mathcal{H}$，训练误差和泛化误差之差以概率 $1-\delta$ 不大于 ε，同时 EMR 最小化得到的假设 $\hat{h}$ 与最小泛化误差之间的差距不大于 2ε。

例 5.5.2　讨论例 5.5.1 的假设空间

$$\mathcal{H}=\{h_{\bar{w}} \mid h_{\bar{w}}(\boldsymbol{x})=I(\bar{\boldsymbol{w}}^{\mathrm{T}}\bar{\boldsymbol{x}} \geqslant 0),\ \bar{\boldsymbol{w}} \in \mathbf{R}^{K+1}\}$$

$\bar{\boldsymbol{w}}$ 有 $K+1$ 个系数，若 $\bar{\boldsymbol{w}}$ 的每个分量用字长 L 的二进制表示(计算机中常取 L 为 16、32 或 64，近期一些研究采用低比特实现神经网络时，甚至取 L 为 8、4)，则表示 $\bar{\boldsymbol{w}}$ 所需的二进制位数为 $L(K+1)$，故假设空间 $\mathcal{H}$ 共有 $|\mathcal{H}|=2^{L(K+1)}$ 个元素，由式(5.5.18)可得对于固定 δ，ε 时，样本数需满足

$$N \geqslant \frac{1}{2\varepsilon^2}\ln\frac{2|\mathcal{H}|}{\delta}=\frac{1}{2\varepsilon^2}\ln\frac{2\times 2^{L(K+1)}}{\delta}=O\left(\frac{KL}{\varepsilon^2}\ln\frac{1}{\delta}\right)=O_{\varepsilon,\delta}(K) \tag{5.2.24}$$

或记为

$$N \sim O_{\varepsilon,\delta}(K) \tag{5.2.25}$$

其中，$O(\cdot)$ 表示量级；$O_{\varepsilon,\delta}(\cdot)$ 表示量级函数，其中 δ，ε 是其参数。式(5.2.25)尽管有比例系数存在，但需要的样本数 N 与模型的参数数目 K 是呈线性关系的。

5.5.2　假设空间无限时的泛化误差界

将定理 5.5.2 的结论推广到假设空间 $\mathcal{H}$ 无限的情况。如前所述，像例 5.5.1 所示的线性模型或后续章节介绍的支持向量机和神经网络等模型，当参数取实数时，假设空间是无限的，即 $|\mathcal{H}|=\infty$，这种情况下式(5.5.23)变得无意义。需要对其进行推广。这里对假设空间无限的情况，只做一个非常扼要的介绍。

当$|\mathcal{H}|=\infty$时，为了表示假设空间的表示能力（或容量），给出两个概念：打散（Shatters）和VC维（Vapnik-Chervonenkis Dimension），这里只给出其简单、直观性的介绍。

首先给出打散的概念，对于一个包含d个点的集合$S=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_d\}$，称$\mathcal{H}$可打散$S$是指：对点集$S$对应加上一个任意标注集$\{y_1,y_2,\cdots,y_d\}$，则必存在$h\in\mathcal{H}$，使$h(\boldsymbol{x}_i)=y_i,i=1,2,\cdots,d$。

对于一个假设空间$\mathcal{H}$，至少存在一个最大元素数为d的点集合S，$\mathcal{H}$可打散S，则$\mathcal{H}$的VC维为d，记为$\mathrm{VC}(\mathcal{H})=d$。这里，$d$是能被$\mathcal{H}$打散的点集的最大元素数，对于有$d+1$个元素的点集合，$\mathcal{H}$均不可能打散它。

例5.5.3 图5.5.1给出了二维平面上的3个点组成的点集和其对应的各种标注，直线表示判决线，假设空间为二维线性分类器，即$\mathcal{H}=\{h(\boldsymbol{x})=\theta_1x_1+\theta_2x_2+\theta_0\,|\,\theta_0,\theta_1,\theta_2\in R\}$，图中每条判决线属于$\mathcal{H}$，对这个3个元素的点集，$\mathcal{H}$可将其打散。如果是4个点，则对于标注为异或运算，$\mathcal{H}$不能正确分类，故$\mathcal{H}$不能打散4个点的点集，因此，平面线性分类器的VC维为3，即$\mathrm{VC}(\mathcal{H})=3$。

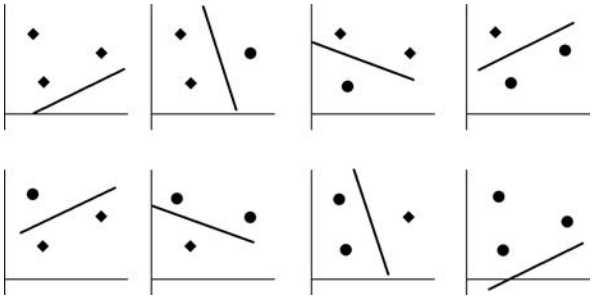

图5.5.1 可由线性分类器打散的点集

若用VC维d表示一个假设空间的容量，则可得到与定理5.5.2类似的结论，这里只给出对应的定理。

定理5.5.3 对于假设空间$\mathcal{H}$，若其VC维为$d=\mathrm{VC}(\mathcal{H})$，则对于所有$h\in\mathcal{H}$，以概率不小于$1-\delta$，满足

$$|R(h)-\hat{R}(h)|\leqslant O\left(\sqrt{\frac{d}{N}\ln\frac{N}{d}+\frac{1}{N}\ln\frac{1}{\delta}}\right) \tag{5.5.26}$$

对于$\hat{h}$满足

$$R(\hat{h})\leqslant\min_{h\in\mathcal{H}}R(h)+O\left(\sqrt{\frac{d}{N}\ln\frac{N}{d}+\frac{1}{N}\ln\frac{1}{\delta}}\right) \tag{5.5.27}$$

对于以概率不小于$1-\delta$满足$R(\hat{h})\leqslant\min_{h\in\mathcal{H}}R(h)+2\varepsilon$，样本数目需要求

$$N\sim O_{\varepsilon,\delta}(d) \tag{5.5.28}$$

注意，这里只用了量级函数$O(\cdot)$而忽视了表达式中的一些具体常数系数，对于了解性能界这就足够了。从式(5.5.28)看，对于无限假设空间，所需样本数与假设空间的VC维呈线性关系。

机器学习理论给出对学习性能的整体性洞察，是非常有意义的，但目前对于一类实际模型的学习算法（如逻辑回归、神经网络、决策树等），其给出的一些要求（如样本数等）与具体

算法的实际需求还有较大距离，在指导一类具体算法的参数选择上的实际意义还有待改善，故实际中仍更常用交叉验证等技术确定机器学习模型的各类参数。

5.6　本章小结

本章介绍了基本的分类算法。对于确定性的判别函数方法，介绍了 Fisher 判别函数和感知机算法；对于概率判别方法，介绍了逻辑回归算法；对于简化的生成模型方法，介绍了朴素贝叶斯算法。这几类算法尽管简单，仍有其应用价值，尤其是逻辑回归和朴素贝叶斯方法，在小样本集和低复杂度的问题上是一种有效且简单的学习算法。由逻辑回归算法所引出的交叉熵目标函数和随机梯度算法，可直接推广到神经网络包括深度神经网络的学习中。

本章最后给出了机器学习理论的一个极其简单的介绍。本书没有设置专门的机器学习理论章节，而是结合回归介绍了偏和方差的折中问题，结合分类介绍了泛化界定理。这样做的目的是，将有关理论分散在不同章节，并结合一类学习算法进行介绍，增加其直观性，使其更容易理解。本书更侧重于机器学习算法的介绍，有关机器学习理论的讨论非常简略，对机器学习理论更感兴趣的读者，可参考 Mohri 等的 *Foundations of Machine Learning* 和 Shai 等的 *Understanding Machine Learning*：*From Theory to Algorithms*，这两本书对机器学习理论进行了较为深入的讨论，均由张文生等译成了中文。对于统计学习理论的一个简明版的读本是 Vapnik 的 *The Nature of Statistical Learning Theory*，已由张学工译成中文。

习题

1. 证明：式(5.2.35)和式(5.2.36)的投影的总类内散布矩阵和类间散布矩阵分别可表示为

$$\hat{\boldsymbol{S}}_{\mathrm{W}}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{W},\quad \hat{\boldsymbol{S}}_{\mathrm{B}}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{W}$$

2. 证明准则

$$J(\boldsymbol{W})=\frac{\mathrm{tr}(\hat{\boldsymbol{S}}_{\mathrm{B}})}{\mathrm{tr}(\hat{\boldsymbol{S}}_{\mathrm{W}})}=\frac{\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{B}}\boldsymbol{W})}{\mathrm{tr}(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{W}}\boldsymbol{W})}$$

最大化所得到 $\boldsymbol{W}$ 的最优解满足 $\boldsymbol{S}_{\mathrm{B}}\boldsymbol{w}_i=\lambda_i\boldsymbol{S}_{\mathrm{W}}\boldsymbol{w}_i$，$i=1,2,\cdots,C-1$，其中 $\boldsymbol{w}_i$ 为 $\boldsymbol{W}$ 的列向量，是对应于 $\boldsymbol{S}_{\mathrm{W}}^{-1}\boldsymbol{S}_{\mathrm{B}}$ 的前$(C-1)$个最大特征值的特征向量。

3. 异或的样本集 $\boldsymbol{D}=\{((0,0)^{\mathrm{T}},-1),((0,1)^{\mathrm{T}},1),((1,0)^{\mathrm{T}},1),((1,1)^{\mathrm{T}},-1)\}$ 是线性不可分的，可定义 $\boldsymbol{x}$ 映射到基函数向量 $\boldsymbol{\varphi}(\boldsymbol{x})=[\varphi_1(\boldsymbol{x}),\varphi_2(\boldsymbol{x}),\cdots,\varphi_{M-1}(\boldsymbol{x})]^{\mathrm{T}}$，在 $\boldsymbol{\varphi}(\boldsymbol{x})$ 表示下设计感知机为 $\hat{y}(\boldsymbol{x};\boldsymbol{w})=\mathrm{sgn}[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+w_0]$，对于异或问题，可定义一个多项式函数 $\boldsymbol{\varphi}(\boldsymbol{x})$，其中

$$\varphi_1(\boldsymbol{x})=2(x_1-0.5)$$

$$\varphi_2(\boldsymbol{x})=4(x_1-0.5)(x_2-0.5)$$

把样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}$ 映射成 $\boldsymbol{D}_{\varphi}=\{(\boldsymbol{\varphi}(\boldsymbol{x}_n),y_n)\}$，样本集映射为

$$\boldsymbol{D}_{\varphi}=\{((-1,1)^{\mathrm{T}},-1),((-1,-1)^{\mathrm{T}},1),((1,-1)^{\mathrm{T}},1),((1,1)^{\mathrm{T}},-1)\}$$

在样本集 $\boldsymbol{D}_{\varphi}$ 上手动训练一个感知机(自行给出初始值和样本使用顺序)。

4. 在第 3 题中，通过映射函数

$$\varphi_1(\boldsymbol{x})=2(x_1-0.5)$$

$$\varphi_2(\boldsymbol{x})=4(x_1-0.5)(x_2-0.5)$$

将异或样本映射成线性可分的，请自行另设计一组映射函数，将异或样本映射为可分情况。

5. 对于 Sigmoid 函数

$$\sigma(a)=\frac{1}{1+\mathrm{e}^{-a}}$$

证明：$\sigma(-a)=1-\sigma(a)$，$\dfrac{\mathrm{d}\sigma(a)}{\mathrm{d}a}=\sigma(a)(1-\sigma(a))$。

6. 逻辑回归的目标函数为 $J(\boldsymbol{w})=-\sum_{n=1}^{N}y_n\ln\hat{y}_n+(1-y_n)\ln(1-\hat{y}_n)$，对 $\boldsymbol{w}$ 求二阶导数得到汉森矩阵，证明：汉森矩阵表示为

$$\boldsymbol{H}=\sum_{n=1}^{N}\hat{y}_n(1-\hat{y}_n)\boldsymbol{\varphi}_n\boldsymbol{\varphi}_n^{\mathrm{T}}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Phi}$$

7. 由 Softmax 的定义

$$\hat{y}_k=\frac{\exp(a_k)}{\sum_{j=1}^{C}\exp(a_j)},\quad k=1,2,\cdots,C$$

证明：$\dfrac{\partial\hat{y}_k}{\partial a_j}=\hat{y}_k(I_{kj}-\hat{y}_j)$。

*8. 在网络上搜索并下载 Iris 数据集，该数据集样本属于鸢尾属下的 3 个亚属，分别为山鸢尾(Setosa)、变色鸢尾(Versicolor)和弗吉尼亚鸢尾(Virginica)。4 个特征被用作样本的定量描述，它们分别是花萼和花瓣的长度和宽度。该数据集包含 150 个数据，每类 50 个数据。

(1) 从数据集中取出变色鸢尾(Versicolor)和维吉尼亚鸢尾(Virginica)的 100 个样本，训练一个二分类的逻辑回归分类器，对其进行分类(建议：每类 40 个样本做训练，10 个样本做测试)。

(2) 设计一个用于三分类问题进行分类的逻辑回归分类器，对 3 种类型进行分类(建议：每类 40 个样本做训练，10 个样本做测试)。