第 1 章　绪　论

第 2 章　概率分布

第 3 章　线性回归模型

第 4 章　线性分类模型

在前面的章节中，我们研究了一类分析和计算属性比较简单的回归模型。我们现在将会讨论一类与之类似的模型，不过这次的模型是用于分类问题的。分类问题中，我们要给输入向量 $\mathbf{x}$ 分配 K 个类别 $\mathcal{C}_k, k = 1, ..., K$ 中的一个。在大多数相关的书中，这些类别是相互独立的，所以每个输入有且仅有一个分类。于是输入空间会被划分成决策域，其边界为决策边界或决策面。在本章节中，我们研究的是线性分类模型，线性分类模型的决策边界为输入向量 $\mathbf{x}$ 的线性函数，即位于 D 维输入空间中的 $(D-1)$ 维的超平面。可以由线性决策边界划分的数据集被称为线性可分 (linearly separable)。

对于回归问题，目标变量 $\mathbf{t}$ 为实数组成的向量，其各个分量就是我们希望预测的。在分类问题中，表示分类标签的目标值就有很多种方法了。对于二分类问题的概率模型，最简单的方法就是二值表示，即设定目标变量 $t \in \{0, 1\}$，$t = 1$ 表示类别 $\mathcal{C}_1$，$t = 0$ 表示属于类别 $\mathcal{C}_2$。我们可以将 t 的值视为分类为 $\mathcal{C}_1$ 的概率，不过这个概率仅有 $0, 1$ 两个取值。对于 $K > 2$ 分类问题，可以使用"1-of-K" 编码的形式来表示分类，设 $\mathbf{t}$ 为 K 维向量，假如分类为 $\mathcal{C}_j$，那么就将 t_j 以外的全部 $\mathbf{t}$ 的元素 t_k 都设为 0，唯独将 t_j 设置为 1。举个例子来说，假如是一个 $K = 5$ 的分类问题，那么对于某个分类为 2 的输入，其对应的目标向量为

$$\mathbf{t} = (0, 1, 0, 0, 0)^{\mathrm{T}} \tag{4.1}$$

和前面一样，我们可以将 t_k 的值看作是分类为 $\mathcal{C}_k$ 的概率。对于非概率模型，可能会有其他更加方便的目标变量表示方法。

在第 1 章中，我们对 3 种不同的分类方法进行了区分，其中最简单的是构建直接将输入向量分配到某一类别中的判别函数。不过，在推断步骤中构建条件概率分布 $p(\mathcal{C}_k|\mathbf{x})$ 然后利用它进行最优决策的方法要更好一些。从第 1.5.4 节中可以看出，通过将推断与决策拆分开来，可以获得很大的收益。而确定条件概率 $p(\mathcal{C}_k|\mathbf{x})$ 又有两种不同的方法，其一是直接对条件概率建模，比如将它们表示为参数模型，然后利用训练集去优化参数；此外，也可以采用生成方法，对分类条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 建模，

同时确定分类的先验概率 $p(\mathcal{C}_k)$，然后根据贝叶斯定理计算后验概率

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \tag{4.2}$$

我们会在本章节中研究以上 3 种方法的应用案例。

在第 3 章中讨论的线性回归模型中，模型的预测 $y(\mathbf{x}, \mathbf{w})$ 是参数 $\mathbf{w}$ 的线性函数。在最简单的情况下，模型同样也是输入量 $\mathbf{x}$ 的线性函数，于是其形式为 $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$，于是 y 为实数。不过，对于分类问题而言，我们希望得到的是离散的类别标签，或者说得更一般一些，希望得到的是处于 $(0, 1)$ 内的后验概率。为了做到这一点，我们需要研究这种模型的扩展形式，也就是将 $\mathbf{w}$ 的线性函数进行非线性变换 $f(\cdot)$，于是

$$y(\mathbf{x}) = f(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \tag{4.3}$$

在机器学习中，$f(\cdot)$ 被称为激活函数 (activation function)，其反函数在统计学中被称为联系函数 (link function)。决策边界上 $y(\mathbf{x})$ 为常数，于是 $\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$ 为常数，所以不管 $f(\cdot)$ 是不是线性函数，决策边界都是 $\mathbf{x}$ 的线性函数。根据这个性质，(4.3) 中的分类模型被称为广义线性模型 (generalized linear models, McCullagh and Nelder, 1989)。不过需要注意的是，与回归模型不同，由于 $f(\cdot)$ 的出现，分类模型不再是参数的线性函数。这样一来，分析性质和计算性质不可避免地要比线性回归模型复杂一些了。尽管如此，广义线性模型仍然比后续章节要学习的其他非线性模型简单很多。

本章节中所讨论的算法同样适用于像第 3 章中那样，利用基底函数 $\boldsymbol{\phi}(\mathbf{x})$ 对输入变量进行非线性变换的情况。我们首先研究直接对 $\mathbf{x}$ 的输入空间进行分类的方法，在第 4.3 节中我们会发现利用基底函数的表示方法对后续的章节有很大的帮助。

4.1 判别函数

判别函数是直接为输入向量 $\mathbf{x}$ 赋予其分类 $\mathcal{C}_k, k = 1, ..., K$ 的函数。在本章中，我们将主要研究线性判别函数，其决策边界为超平面。为了简化讨论，我们首先研究二分类问题，然后再扩展到 $K > 2$ 的情况。

4.1.1 二分类问题

最简单的线性判别函数为输入变量的线性函数，于是

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0 \tag{4.4}$$

其中 $\mathbf{w}$ 为权重向量 (weight vector)，w_0 为偏差——注意不要与统计学中的偏差混淆。偏差的相反数有时被称为阈值 (threshold)。如果 $y(\mathbf{x}) \geqslant 0$，那么输入向量 $\mathbf{x}$ 的类别为

$\mathcal{C}_1$，否则为 $\mathcal{C}_2$。而对应的决策边界则是 $y(\mathbf{x}) = 0$，事实上它是一个 D 维输入空间中的 $(D-1)$ 维超平面。对于决策平面上的两个点 $\mathbf{x}_\mathrm{A}$ 和 $\mathbf{x}_\mathrm{B}$，由于 $y(\mathbf{x}_\mathrm{A}) = y(\mathbf{x}_\mathrm{B}) = 0$，所以 $\mathbf{w}^\mathrm{T}(\mathbf{x}_\mathrm{A} - \mathbf{x}_\mathrm{B}) = 0$，于是向量 $\mathbf{w}$ 与决策平面内的任意向量都是正交的，也就是说 $\mathbf{w}$ 确定了决策平面的方向。类似地，如果 $\mathbf{x}$ 是决策平面上的点，那么 $y(\mathbf{x}) = 0$，于是从原点到决策平面的距离为

$$\frac{\mathbf{w}^\mathrm{T}\mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{w}\|} \tag{4.5}$$

于是可以看出偏差参数 w_0 确定了决策平面的位置。如图 4.1 所示为 $D = 2$ 的情况。

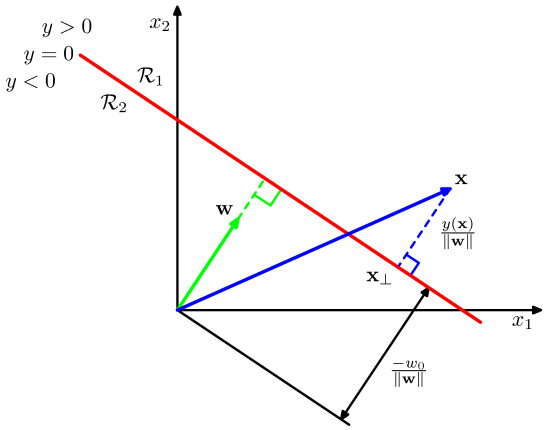

图 4.1: 二维空间中线性判别函数的几何表示。红线为决策平面 (这里也可以说成是决策边界)，可以看出它是与 $\mathbf{w}$ 垂直的；决策平面相对原点的偏移量由偏差参数 w_0 控制。此外，对于一个一般的点 $\mathbf{x}$，它到决策平面的正交距离为 $y(\mathbf{x})/\|\mathbf{w}\|$。

另外还需要注意到，$y(\mathbf{x})$ 的值给出了点 $\mathbf{x}$ 到决策平面的垂直距离 r，而且是带有符号的。为了验证这一点，设对于一个一般的点 $\mathbf{x}$，令 $\mathbf{x}_\perp$ 为 $\mathbf{x}$ 在决策平面上的正交投影，于是

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{4.6}$$

在等式两侧同时乘以 $\mathbf{w}^\mathrm{T}$ 并加上 w_0，根据 $y(\mathbf{x}) = \mathbf{w}^\mathrm{T}\mathbf{x} + w_0$ 和 $y(\mathbf{x}_\perp) = \mathbf{w}^\mathrm{T}\mathbf{x}_\perp + w_0 = 0$，可以得到

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|} \tag{4.7}$$

这一结果如图 4.1 所示。

和第 3 章中研究线性回归模型时一样，有时利用另一种比较紧凑的表示方法要更方便一些，也就是加上额外的输入 $x_0 = 1$ 从而定义 $\widetilde{\mathbf{w}} = (w_0, \mathbf{w})$ 和 $\widetilde{\mathbf{x}} = (x_0, \mathbf{x})$，于是

$$y(\mathbf{x}) = \widetilde{\mathbf{w}}^\mathrm{T}\widetilde{\mathbf{x}} \tag{4.8}$$

在这种情况下，决策平面就变成了 $D+1$ 维的增广输入空间中的 D 维超平面，而且是必经过原点的。

4.1.2 多分类问题

现在我们开始研究 $K > 2$ 情况下的线性判别函数。比较直观的想法是，通过将一系列二分类的判别函数结合起来形成一个 K 分类的判别函数，不过我们马上会发现这样的做法会导致一系列的麻烦 (Duda and Hart, 1973)。

假设现在有 $K-1$ 个分类器，每个分类器都是针对二分类问题设计的，也就是说，每个分类器都只能判定某个点是否属于类别 $\mathcal{C}_k$，即” 一对其余”(one-versus-rest, OvR) 分类器。图 4.2 中的左图展示的是 3 个类别的情况，可以看出多个 OvR 分类器结合的方法会使输入空间的划分存在无法判定的模糊区域。

另外有一种替代的方法，即使用 $K(K-1)/2$ 个二分类判别函数，每个判别函数有且仅有两个可能的类别输出，即” 一对一”(one-versus-one, OvO)【译者注: 翻译到这里我已经昆 0v0 了】分类器。然后根据判别函数的多数投票对每个点进行分类。然而，这样的做法也会遇到模糊区域的问题，如图 4.2 中的右图所示。

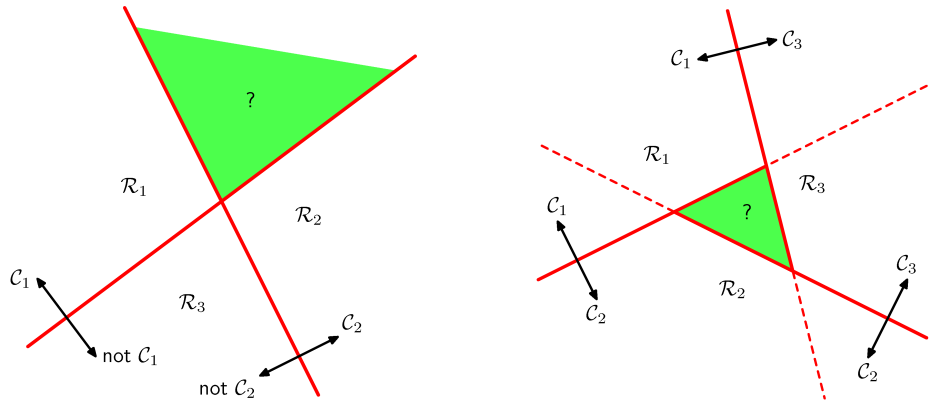

图 4.2: 通过一系列二分类判别函数建立 K 分类模型的做法会导致模糊区域的问题，如图中绿色区域所示。左图为使用 OvR 分类器时的情况，右图为 OvO 分类器时的情况。

不过这个问题是可以解决的。建立一个包含 K 个线性函数的 K 分类判别函数:

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \tag{4.9}$$

这样一来，如果对于一切 $j \neq k$ 都有 $y_k(\mathbf{x}) > y_j(\mathbf{x})$，那么 $\mathbf{x}$ 的类别就会被划分为 $\mathcal{C}_k$，而且类别 $\mathcal{C}_k$ 和 $\mathcal{C}_j$ 之间的决策边界就是 $y_k(\mathbf{x}) = y_j(\mathbf{x})$，即 $(D-1)$ 维的超平面:

$$(\mathbf{w}_k - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{k0} - w_{j0}) = 0 \tag{4.10}$$

这与第 4.1.1 节中二分类问题的决策边界形式相同，而且具有相同的几何性质。

这样的判别函数决策域通常是单连接的，而且通常是凸的。为了验证这一点，如图 4.3 所示，假设决策域 $\mathcal{R}_k$ 中存在两个点 $\mathbf{x}_\mathrm{A}$ 和 $\mathbf{x}_\mathrm{B}$，对于一切位于 $\mathbf{x}_\mathrm{A}$ 和 $\mathbf{x}_\mathrm{B}$ 连线上的点 $\hat{\mathbf{x}}$ 都可以表示为

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_\mathrm{A} + (1-\lambda)\mathbf{x}_\mathrm{B} \tag{4.11}$$

其中 $0 \leqslant \lambda \leqslant 1$。根据判别函数，

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_\mathrm{A}) + (1-\lambda) y_k(\mathbf{x}_\mathrm{B}) \tag{4.12}$$

由于 $\mathbf{x}_\mathrm{A}$ 和 $\mathbf{x}_\mathrm{B}$ 都位于决策域 $\mathcal{R}_k$ 内，所以对于一切 $j \neq k$，一定有 $y_k(\mathbf{x}_\mathrm{A}) > y_j(\mathbf{x}_\mathrm{A})$，以及 $y_k(\mathbf{x}_\mathrm{B}) > y_j(\mathbf{x}_\mathrm{B})$，所以 $y_k(\hat{\mathbf{x}}) > y_j(\hat{\mathbf{x}})$，所以 $\hat{\mathbf{x}}$ 同样位于决策域 $\mathcal{R}_k$ 中。于是决策域 $\mathcal{R}_k$ 一定是单连接且凸的。

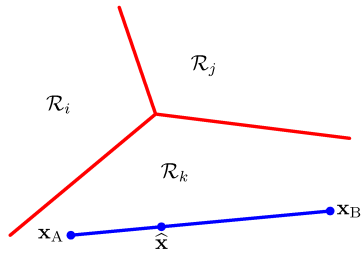

图 4.3: 多分类问题中的线性判别决策域示意图，其中红色表示决策边界。如果两个点 $\mathbf{x}_\mathrm{A}$ 和 $\mathbf{x}_\mathrm{B}$ 都位于决策域 $\mathcal{R}_k$ 内，那么其连线上的任意点 $\hat{\mathbf{x}}$ 都一定位于 $\mathcal{R}_k$ 内，所以决策域一定是单连接且凸的。

对于二分类问题，我们同样可以使用这样的方法，即针对两个判别函数 $y_1(\mathbf{x})$ 和 $y_2(\mathbf{x})$ 或者仅使用一个第 4.1.1 节中更加简单但等效的判别函数 $y(\mathbf{x})$ 进行类似的讨论。

接下来我们会研究 3 种可以确定线性判别函数中参数的方法，分别是最小二乘法、Fisher 线性判别分析和感知机算法。

4.1.3 分类问题的最小二乘法

在第 3 章中，我们研究了参数的线性函数形式的模型，并发现通过对平方和误差函数求取最小化可以得到参数值的闭式解。所以我们自然而然地想到，能否利用类似的方法来处理分类问题。假设对于一个一般的 K 分类问题，利用 1-of-K 的形式表示目标向量 $\mathbf{t}$。在这个条件下使用最小二乘法的理由之一是，它近似等于给定输入向量条件下的目标变量的条件期望 $\mathbb{E}[\mathbf{t}|\mathbf{x}]$。在二进制编码下，该条件期望由向量的后验分类概率给出。不过不幸的是，这样的概率近似值通常非常差劲，这是因为线性

模型灵活程度有限所导致的，有时候甚至会有近似值超过范围 (0,1) 的情况。

任意的分类 $\mathcal{C}_k$ 都有各自的线性模型，

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}} + w_{k0} \tag{4.13}$$

其中 $k = 1, ..., K$。于是我们可以将其整合在一起，形成一个大的向量，于是

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} \tag{4.14}$$

其中，矩阵 $\widetilde{\mathbf{W}}$ 的第 k 列为 $(D+1)$ 维向量 $\widetilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^{\mathrm{T}})^{\mathrm{T}}$，$\widetilde{\mathbf{x}}$ 则表示输入向量 $(1, \mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$，即包含了 $x_0 = 1$ 的向量。这一表示方法已经在第 3.1 节中详细介绍过。对于任意的输入向量 $\mathbf{x}$，将它划分到哪一类的时候输出 $y_k = \widetilde{\mathbf{w}}_k^{\mathrm{T}}\widetilde{\mathbf{x}}$ 取得最大值，就将它划分到哪一类。

和第 3 章一样，现在我们要通过平方和误差函数的最小化来确定参数矩阵 $\widetilde{\mathbf{W}}$。假设训练集为 $\{\mathbf{x}_n, \mathbf{t}_n\}, n = 1, ..., N$，并定义矩阵 $\mathbf{T}$，其第 n 行为向量 $\mathbf{t}_n^{\mathrm{T}}$，以及定义矩阵 $\widetilde{\mathbf{X}}$，其第 n 行为 $\widetilde{\mathbf{x}}_n^{\mathrm{T}}$。于是平方和误差函数为

$$E_D(\widetilde{\mathbf{W}}) = \frac{1}{2}\mathrm{Tr}\left\{(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})^{\mathrm{T}}(\widetilde{\mathbf{X}}\widetilde{\mathbf{W}} - \mathbf{T})\right\} \tag{4.15}$$

对 $\widetilde{\mathbf{W}}$ 求导并令导数为 0，并稍作整理，可以得到 $\widetilde{\mathbf{W}}$ 的解

$$\widetilde{\mathbf{W}} = (\widetilde{\mathbf{X}}^{\mathrm{T}}\widetilde{\mathbf{X}})^{-1}\widetilde{\mathbf{X}}^{\mathrm{T}}\mathbf{T} = \widetilde{\mathbf{X}}^{\dagger}\mathbf{T} \tag{4.16}$$

和第 3.1.1 节中一样，其中 $\widetilde{\mathbf{X}}^{\dagger}$ 表示矩阵 $\widetilde{\mathbf{X}}$ 的伪逆。于是接下来就可以确定完整的判别函数

$$\mathbf{y}(\mathbf{x}) = \widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{x}} = \mathbf{T}^{\mathrm{T}}\left(\widetilde{\mathbf{X}}^{\dagger}\right)^{\mathrm{T}}\widetilde{\mathbf{x}} \tag{4.17}$$

多目标变量的最小二乘解有一个有趣的性质，如果训练集中的每个目标向量都满足某参数为 $\mathbf{a}$ 和 b 的线性约束

$$\mathbf{a}^{\mathrm{T}}\mathbf{t}_n + b = 0 \tag{4.18}$$

那么模型对任意 $\mathbf{x}$ 产生的预测也将同样满足这一约束，即

$$\mathbf{a}^{\mathrm{T}}\mathbf{y}(\mathbf{x}) + b = 0 \tag{4.19}$$

所以如果使用 1-of-K 编码的形式表示分类，那么模型所做出的预测一定满足一条性质——$\mathbf{y}(\mathbf{x})$ 的所有元素之和为 1。不过，这一约束还不足以使得模型输出可以视为概率，因为并没有将元素约束在区间 $(0, 1)$ 内。

最小二乘法确实可以求出判别函数参数的闭式解。不过，这样得到的判别函数是直接进行预测，并未考虑任何概率层面上的解释，在应用时会存在很多问题。从图

4.4 可以看出，最小二乘解缺乏针对异常值的鲁棒性，这一问题同样会出现在分类问题中。其中，右侧图中所示的附加数据点对决策边界产生了严重的影响，更可恨的是，要是像左图中那样进行划分，这些点还是可以被正确分类的。平方和误差函数会对远离决策边界，但处于正确区域的”过于正确”的预测进行惩罚。在第 7.1.2 节中，我们会研究若干种分类问题中其他类型的误差函数，而且会发现它们就不会遇到这样的麻烦。

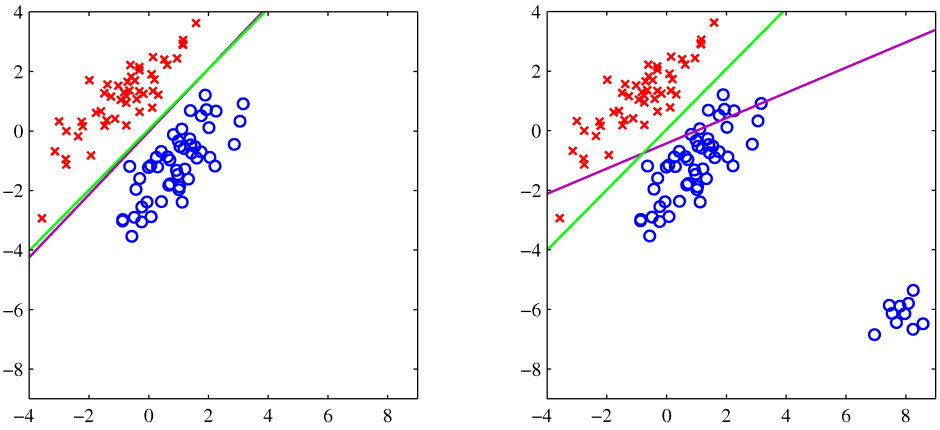

图 4.4: 左图中展示的是两类数据以及通过最小二乘法 (紫红色) 和 logistic 回归模型 (绿色，即将在第 4.3.2 节中讨论) 得到的决策边界，两类数据分别表示为个红色叉号和蓝色圆圈。右图中展示的是添加了一些附加数据后决策边界的变化情况，可以看出最小二乘法得到的决策边界对异常值过于敏感，而 logistic 回归方法则不会这样。

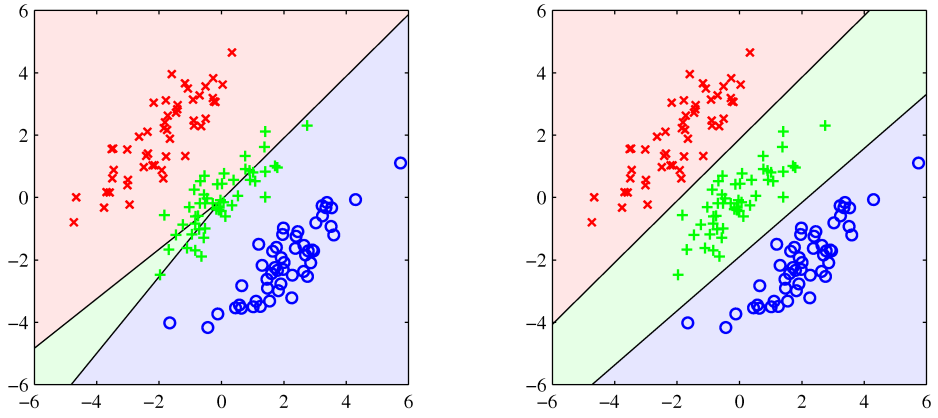

图 4.5: 对于一个包含 3 个类别的线性可分数据集，其中训练数据的类别分别表示为红色叉号、绿色十字和蓝色圆圈。图中的直线表示决策边界，背景颜色表示各自的决策域。左图是采用最小二乘法进行判别的结果。可以看出分给绿色类别的决策域非常小，绝大多数点都被误分类了。右图是采用第 4.3.2 节中即将介绍的 logistic 回归进行判别的结果，很明显分类的效果要好很多。

然而，最小二乘法存在的问题还不止于此。如图 4.5 所示，在二维输入空间 (x_1, x_2) 中存在一个包含 3 个类别的数据集，而且该数据集是线性可分的。利用本章后续即将介绍的 logistic 回归方法，可以得到右图那样很好的结果，但最小二乘法得到的结果却很差，输入空间只有很小的一部分划分给了绿色的类别。

最小二乘法的不尽如人意其实并不令人以外，因为最小二乘法事实上对应的是高斯条件分布假设下的最大似然，但二元的目标向量的分布很明显与高斯分布差了十万八千里。采用更加合适的概率模型，可以得到比最小二乘法效果更好的分类方法。不过现在我们先继续研究确定线性分类模型参数的非概率方法。

4.1.4 Fisher 线性判别分析

从降维的视角看待线性分类模型的一种不错的方法。首先研究二分类问题，假设输入向量 $\mathbf{x}$ 的维度为 D，经过如下的投影：

$$y = \mathbf{w}^{\mathrm{T}}\mathbf{x} \tag{4.20}$$

可以将其变为一维，假如存在某个关于 y 的阈值，如果 $y \geqslant -w_0$ 则分类为 $\mathcal{C}_1$，否则分类为 $\mathcal{C}_2$，这样一来就得到了前面所讨论的线性分类器。一般而言，将向量投影到一维空间中不可避免地会造成信息的损失，而且在原始的 D 维空间中所划分的类别会在一维空间中存在严重的重叠。不过，通过调整权重向量 $\mathbf{w}$，我们可以选择一个使得类别分离度最大化的投影方式。首先，假设在二分类问题中有 N_1 个点属于类别 $\mathcal{C}_1$，N_2 个点属于类别 $\mathcal{C}_2$，那么两个类别各自的均值向量分别为

$$\mathbf{m}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1}\mathbf{x}_n, \mathbf{m}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2}\mathbf{x}_n \tag{4.21}$$

当投影到 $\mathbf{w}$ 上时类别均值的投影是类别分离度最简单的衡量标准。也就是说我们应当选择使得

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.22}$$

取得最大值的 $\mathbf{w}$，其中

$$m_k = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k \tag{4.23}$$

为类别 $\mathcal{C}_k$ 的数据投影后的均值。不过这里有个毛病，随便增加 $\mathbf{w}$ 的大小，就可以使 m_k 变成任意大小。为了解决这个问题，我们可以给 $\mathbf{w}$ 加上单位化的限定，即 $\sum_i w_i^2 = 1$。利用拉格朗日乘数法构建约束优化问题，可以得出一个结论——$\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$。

不过这个方法还是有个问题，如图 4.6 所示。假设在二维输入空间 (x_1, x_2) 中存在线性可分的两类数据，但在投影到两类数据均值的连线上之后出现了严重的重叠现象。当类别的概率分布的协方差矩阵与对角矩阵相差较大时就会出现这样的问题。

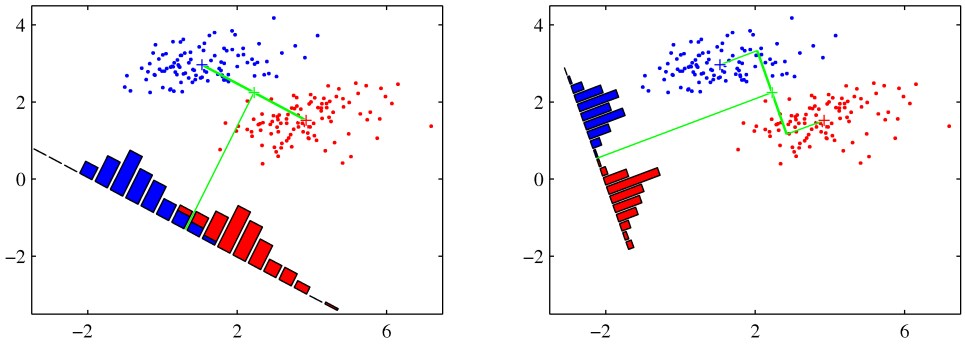

图 4.6: 左图中展示的是两类数据 (红色和蓝色) 投影到两类数据各自均值的连线上之后的直方图。需要注意的是，其中出现了类别重叠的问题。右图中展示的是基于 Fisher 线性判别分析的投影结果，很明显具有更好的效果。

Fisher 提出的方法是对某个函数进行最大化，这个函数可以使得投影后的类别均值分离度较大，同时使每个类别内的方差较小，从而使类别重叠最小化。

投影公式 (4.20) 可以将一系列的 $\mathbf{x}$ 投影到一维空间 y 中。每个类别 $\mathcal{C}_k$ 内的数据经过变换后，其方差为

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \tag{4.24}$$

其中 $y_n = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$。在整个数据集上，所有类别内方差的总和可以定义为 $s_1^2 + s_2^2$。Fisher 的方法就是定义了类别间方差与类别内方差的比值，即

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{4.25}$$

利用 (4.20)，(4.23) 和 (4.24)，将以上公式改写为

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}} \tag{4.26}$$

其中 $\mathbf{S}_{\mathrm{B}}$ 为类别间的协方差矩阵，

$$\mathbf{S}_{\mathrm{B}} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^{\mathrm{T}} \tag{4.27}$$

$\mathbf{S}_{\mathrm{W}}$ 为类别内协方差矩阵的总和，

$$\mathbf{S}_{\mathrm{W}} = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^{\mathrm{T}} + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^{\mathrm{T}} \tag{4.28}$$

对 (4.26) 关于 $\mathbf{w}$ 求导，可以得出当

$$(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w})\mathbf{S}_{\mathrm{W}}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w})\mathbf{S}_{\mathrm{B}}\mathbf{w} \tag{4.29}$$

时，$J(\mathbf{w})$ 取得最大值。根据 (4.27)，可以看出 $\mathbf{S}_{\mathrm{B}}\mathbf{w}$ 始终处于 $(\mathbf{m}_2 - \mathbf{m}_1)$ 方向上。另外，我们不太关心 $\mathbf{w}$ 的大小，而是只关心它的方向，所以尺度因子 $\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{B}}\mathbf{w}$ 和 $\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{W}}\mathbf{w}$ 可以直接扔掉不管。在 (4.29) 等式两侧同时乘以 $\mathbf{S}_{\mathrm{W}}^{-1}$，

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.30}$$

需要注意的是，如果类别内协方差是各项同性的，那么 $\mathbf{S}_{\mathrm{W}}$ 将是单位矩阵的倍数，而 $\mathbf{w}$ 与类别之间的差距成正比。

公式 (4.30) 就是 Fisher 线性判别公式。不过严格来说，它更像是数据投影到一维空间时的方向选择，而非判别函数。然而，经过投影的数据接下来就可以构建判别函数了，只需设置一个阈值 y_0，当 $y(\mathbf{x}) \geqslant y_0$ 时分类为 $\mathcal{C}_1$，否则分类为 $\mathcal{C}_2$。举例而言，我们可以利用高斯分布对类别条件概率密度 $p(y|\mathcal{C}_k)$ 进行建模，然后利用第 1.2.4 节中的最大似然方法求取高斯分布的参数。得到投影类别的近似高斯分布后，可以利用第 1.5.1 节中的方法求出最优的阈值。由于 $y = \mathbf{w}^{\mathrm{T}}\mathbf{x}$ 是一系列随机变量的和，所以可以利用中心极限定理对高斯分布进行一些假设。

4.1.5 与最小二乘法的关系

用于确定线性判别函数的最小二乘法建立在一个原则之上，那就是让目标变量的预测尽可能地接近真实值。相比之下，Fisher 方法则是要尽可能在输出空间使类别的分离度达到最大。这两种方法之间的关系也比较有意思。特别地，对于二分类问题，Fisher 方法事实上是最小二乘法的一种特殊情况。

到目前为止我们一直在用 1-of-K 编码来表示目标变量。不过，如果我们换一种目标变量的表示方式，那么权重向量的最小二乘解将等价于 Fisher 方法的解 (Duda and Hart, 1973)。特别地，令属于类别 $\mathcal{C}_1$ 的目标值为 N/N_1，其中 N_1 为类别 $\mathcal{C}_1$ 中的模式数量，N 为模式总数。这个目标值近似等于类别 $\mathcal{C}_1$ 的先验概率的倒数。对于类别 $\mathcal{C}_2$，可以令目标值为 $-N/N_2$，其中 N_2 为类别 $\mathcal{C}_2$ 中的模式数量。

平方和误差函数可以写成

$$E = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n)^2 \tag{4.31}$$

分别令 E 关于 w_0 和 $\mathbf{w}$ 的导数等于 0，可以得到

$$\sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n) = 0 \tag{4.32}$$

$$\sum_{n=1}^{N}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + w_0 - t_n)\mathbf{x}_n = 0 \tag{4.33}$$

根据 (4.32) 和目标变量 t_n 的表示方法，可以确定偏差的表达式

$$w_0 = -\mathbf{w}^{\mathrm{T}}\mathbf{m} \tag{4.34}$$

其中用到了一个结果，

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0 \tag{4.35}$$

以及 $\mathbf{m}$ 是整个数据集上的均值，其表达式为

$$\mathbf{m} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N}(N_1\mathbf{m}_1 + N_2\mathbf{m}_2) \tag{4.36}$$

经过一系列计算之后，(4.33) 会演变成

$$\left(\mathbf{S}_{\mathrm{W}} + \frac{N_1 N_2}{N}\mathbf{S}_{\mathrm{B}}\right)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2) \tag{4.37}$$

其中 $\mathbf{S}_{\mathrm{W}}$ 定义于 (4.28)，$\mathbf{S}_{\mathrm{B}}$ 定义于 (4.27)，同时代入了偏差 (4.34)。根据 (4.27)，$\mathbf{S}_{\mathrm{B}}\mathbf{w}$ 始终处于 $(\mathbf{m}_2 - \mathbf{m}_1)$ 的方向上。所以

$$\mathbf{w} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \tag{4.38}$$

其中忽略了无关的尺度因子。由此可见，权重向量恰好与 Fisher 方法所得到的结果相同。此外 (4.34) 给出了偏差 w_0 的值。这个结论告诉我们，对于一个新的向量 $\mathbf{x}$，如果 $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}(\mathbf{x} - \mathbf{m}) > 0$，那么它应该分类到类别 $\mathcal{C}_1$ 中，否则应当分类为 $\mathcal{C}_2$。

4.1.6 多分类问题中的 Fisher 线性判别分析

现在我们研究 Fisher 线性判别分析推广到 $K > 2$ 的多分类问题中的情况，假设输入空间的维度 D 大于类别的数量 K。然后，我们引入 $D' > 1$ 个线性“特征” $y_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$，其中 $k = 1, ..., D'$。这次特征的值可以整合成一个独立的向量 $\mathbf{y}$。类似地，权重向量 $\{\mathbf{w}_k\}$ 可以看成是矩阵 $\mathbf{W}$ 的列，于是

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \tag{4.39}$$

需要注意的是，在 $\mathbf{y}$ 的定义中没有出现偏差参数。根据 (4.28)，K 分类问题的类别内协方差矩阵为

$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K} \mathbf{S}_k \tag{4.40}$$

其中

$$\mathbf{S}_k = \sum_{n\in\mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \tag{4.41}$$

$$\mathbf{m}_k = \frac{1}{N_k}\sum_{n\in\mathcal{C}_k} \mathbf{x}_n \tag{4.42}$$

N_k 表示类别 $\mathcal{C}_k$ 中的数据数量。为了求出类别间协方差矩阵，根据 Duda and Hart(1973)，首先求出协方差矩阵总和

$$\mathbf{S}_{\mathrm{T}} = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}} \tag{4.43}$$

其中 $\mathbf{m}$ 表示整个数据集的均值

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^{K} N_k \mathbf{m}_k \tag{4.44}$$

$N = \sum_k N_k$ 表示数据总数。协方差矩阵的总和可以拆分为类别内协方差矩阵的总和 (4.40) 与类别间协方差矩阵 $\mathbf{S}_{\mathrm{B}}$(4.41) 的和，也就是说

$$\mathbf{S}_{\mathrm{T}} = \mathbf{S}_{\mathrm{W}} + \mathbf{S}_{\mathrm{B}} \tag{4.45}$$

其中

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^{\mathrm{T}} \tag{4.46}$$

这些协方差矩阵都是定义在原始 $\mathbf{x}$ 空间内的。现在可以定义投影到 D' 维 $\mathbf{y}$ 空间内的类似的矩阵：

$$\mathbf{S}_{\mathrm{W}} = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \tag{4.47}$$

$$\mathbf{S}_{\mathrm{B}} = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^{\mathrm{T}} \tag{4.48}$$

其中

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n, \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^{K} N_k \boldsymbol{\mu}_k \tag{4.49}$$

和之前一样，我们希望建立一个在类别间协方差很大的情况下类别内协方差很小的判定标准。这有很多种选择 (Fukunaga, 1990)，比如

$$J(\mathbf{W}) = \mathrm{Tr}\left\{\mathbf{S}_{\mathrm{W}}^{-1} \mathbf{S}_{\mathrm{B}}\right\} \tag{4.50}$$

这个等式可以写成投影矩阵函数的形式

$$J(\mathbf{W}) = \mathrm{Tr}\left\{(\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{W}} \mathbf{W})^{-1} (\mathbf{W}^{\mathrm{T}} \mathbf{S}_{\mathrm{B}} \mathbf{W})\right\} \tag{4.51}$$

对这个函数进行最大化是很直接的，在 Fukunaga(1990) 中有详细的讨论。权重的值则是矩阵 $\mathbf{S}_{\mathrm{W}}^{\mathrm{T}} \mathbf{S}_{\mathrm{B}}$ 最大特征值所对应的特征向量。

对于所有的评价标准来说，有一件事是统一的。首先通过 (4.46) 发现 $\mathbf{S}_{\mathrm{B}}$ 是由 K 个矩阵求和得到的，而且每个矩阵都是两个向量的外积，所以这个矩阵的秩一定为 1。此外，根据 (4.44) 的约束，只有 $(K-1)$ 个矩阵是相互独立的。所以，$\mathbf{S}_{\mathrm{B}}$ 的秩其实至多为 $(K-1)$，也就是说，至多有 $(K-1)$ 个非零特征值。所以，$\mathbf{S}_{\mathrm{B}}$ 的特征向量在 $(K-1)$ 维子空间中的投影跨度并不影响 $J(\mathbf{W})$ 的值，以及我们不可能找到 $(K-1)$ 以上个线性“特征”(Fukunaga, 1990)。

4.1.7 感知机算法

线性判别模型的另一个重要案例是 Rosenblatt 在 1962 年提出的感知机算法。它解决的是二分类问题，首先要将输入向量 $\mathbf{x}$ 进行非线性变换，从而得到新的向量 $\phi(\mathbf{x})$，然后构建广义线性模型

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})\right) \tag{4.52}$$

其中的非线性激活函数 $f(\cdot)$ 为阶跃函数，

$$f(a) = \begin{cases} +1, a \geqslant 0 \\ -1, a < 0 \end{cases} \tag{4.53}$$

向量 $\phi(\mathbf{x})$ 中包含有偏差分量 $\phi_0(\mathbf{x}) = 1$。在以前的一些关于二分类问题的讨论中，我们一般将目标变量的取值设置为 $t \in 0, 1$，这样的做法是为了适应概率模型的需要。不过在感知机算法中，将目标变量设置为 $t \in +1, -1$ 要更好一些，所以我们选择了这样的激活函数。

通过误差函数最小化，可以很容易地得到感知机的参数 $\mathbf{w}$。对于这个误差函数，人们会很自然而然地想到将其设置为误分类数据的总数。不过，这样一来学习算法就会变得复杂，因为误差函数会成为关于 $\mathbf{w}$ 的分段常值函数，当 $\mathbf{w}$ 发生变化时，一旦决策边界发生变化，直接会导致误差函数的不连续。所以这样的话想要对 $\mathbf{w}$ 求取梯度从而对误差函数进行最小化的思路就行不通了，因为这个梯度几乎处处为零。

所以我们需要换一个误差函数，这个误差函数称为感知机准则函数 (perceptron criterion)。首先注意到，我们想求取的权重向量要满足这样的条件，对于属于类别 $\mathcal{C}_1$ 的数据，要有 $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) > 0$，属于类别 $\mathcal{C}_2$ 的数据则要有 $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) < 0$。在 $t \in -1, +1$ 这样的编码规则下，我们希望让所有的数据都能够满足 $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n > 0$。在所有数据都正确分类的情况下，感知机准则函数为 0，而对于误分类的 $\mathbf{x}_n$，感知机准则函数要使得 $\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)t_n$ 的值达到最小。于是，感知机准则函数为

$$E_{\mathrm{P}}(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^{\mathrm{T}}\phi_n t_n \tag{4.54}$$

其中 $\phi_n = \phi(x_n)$，$\mathcal{M}$ 表示误分类集合。错误分类对该函数的造成的影响是 $\mathbf{w}$ 空间中关于 $\mathbf{w}$ 的线性函数，而正确分类对函数的影响则是 0。所以误差函数是分段的线

性函数。

现在利用随机梯度下降法处理误差函数。权重向量 $\mathbf{w}$ 的变化情况为

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta\nabla E_{\mathrm{P}}(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta\phi_n t_n \tag{4.55}$$

其中 η 为学习率参数，τ 表示算法的迭代次数。对 $\mathbf{w}$ 乘以一个常数并不会对感知机函数 $y(\mathbf{x},\mathbf{w})$ 造成影响，所以我们可以不失一般性地将 η 设置为 1。需要注意的是，随着权重向量在训练中的变化，误分类样本的集合也会随之变化。

感知机学习算法有一种简单的理解方式，如下所述。当我们一次又一次地遍历数据集中的数据，对于每一个 $\mathbf{x}_n$ 都计算其对应的感知机函数值 (4.52)。如果这个数据的分类是正确的，那么就不修改权重向量；但如果这个数据的分类是错误的，那么对于类别 $\mathcal{C}_1$ 我们将向量 $\phi(\mathbf{x}_n)$ 加在目前的权重向量 $\mathbf{w}$ 的估计值中，对于类别 $\mathcal{C}_2$ 则从 $\mathbf{w}$ 中减去 $\phi(\mathbf{x}_n)$。感知机学习算法如图 4.7 所示。

现在我们将目光集中到感知机学习算法过程中的一次更新上，并推导它的影响究竟是什么样的。实际上在针对这一误分类数据的更新中，这个数据分类错误的影响是在降低的，因为根据 (4.55)，如果将学习率设置为 $\eta = 1$ 的话，

$$-\mathbf{w}^{(\tau+1)\mathrm{T}}\phi_n t_n = -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n - (\phi_n t_n)^{\mathrm{T}}\phi_n t_n < -\mathbf{w}^{(\tau)\mathrm{T}}\phi_n t_n \tag{4.56}$$

其中利用了 $\|\phi_n t_n\|^2 > 0$ 这一特性。当然，这可并不意味着其他的错误分类数据造成的误差也会降低。更要命的是，在权重向量发生更新时，此前的某些正确分类数据可能会变成错误的。因此，感知机学习算法并不能保证在每一次迭代的过程中总误差函数一定是减小的。

不过，根据感知机收敛定理 (perceptron convergence theorem)，如果感知机算法存在精确解 (或者说训练数据集是线性可分的)，那么感知机算法一定可以在有限的步骤内找到这个精确解。该定理的证明可以在 Rosenblatt(1962)，Block(1962)，Nilsson(1965)，Minsky and Papert(1969)，Hertz et al.(1991) 和 Bishop(1995a) 等文献中找到。不过需要注意的是，达到收敛所需要的步数是很多的，而且在实际应用中，在实现收敛之前，我们根本没办法判定这个问题究竟是收敛速度太慢还是根本不能收敛。

即使数据集是线性可分的，也可能有许多结果，具体会得到哪个结果取决于参数的初始化和数据点的出现顺序。此外，对于线性不可分的数据集，感知器学习算法永远不会收敛。

除了学习上的困难，感知机算法还有一些缺点，即没有概率输出，也不容易推广到 $K > 2$ 的分类问题。此外，最要命的限制其实是它仍然是基于固定基底函数的线性组合 (是不是有点似曾相识？)。关于感知机局限性的讨论可以在 Minsky and Papert(1969) 和 Bishop(1995a) 中找到。

感知器的模拟硬件实现由 Rosenblatt 构建，基于电机驱动的可变电阻器来实现

自适应参数 w_j。如图 4.8 所示。输入是从基于光传感器阵列的简单相机系统获得的，而基函数 可以以各种方式选择，例如基于来自输入图像的随机选择的像素子集的简单固定功能。典型应用涉及学习区分简单形状或字符。

在感知器正在开发的同时，Widrow 及其同事正在研究一种与之密切相关的称为“adaline”系统，该系统是“自适应线性元素”的缩写。该模型的功能形式与感知器的功能形式相同，但采用了不同的训练方法 (Widrow and Hoff，1960; Widrow and Lehr，1990)。

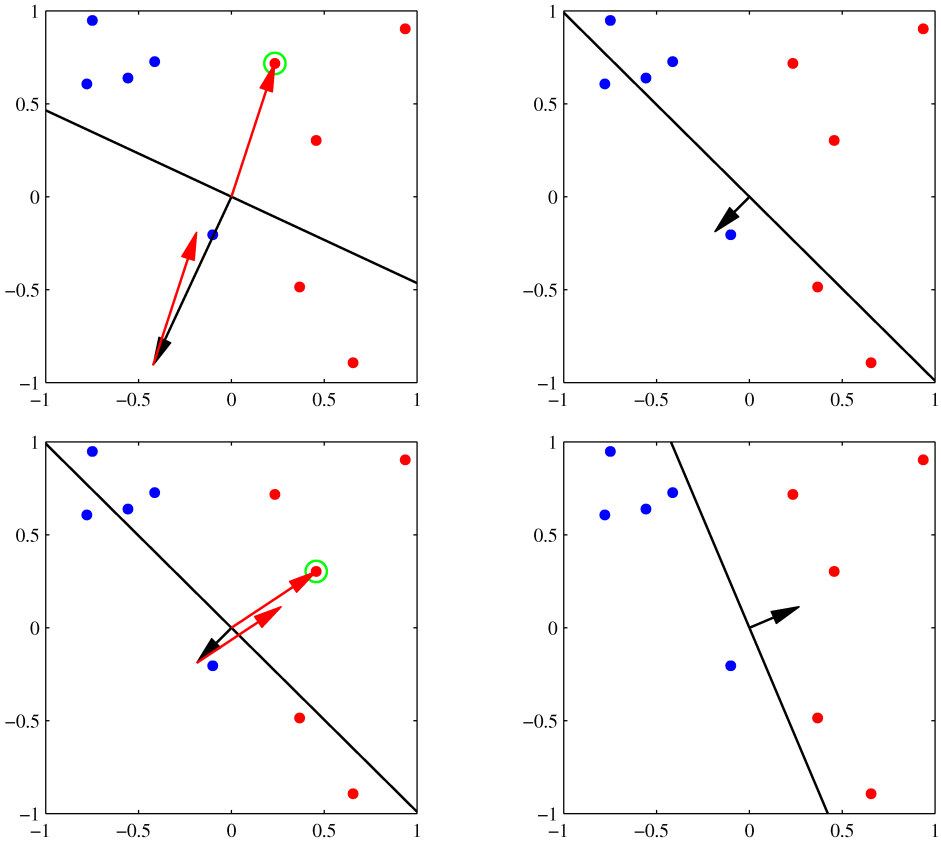

图 4.7: 感知器学习算法的收敛过程，这里展示了在二维特征空间 (ϕ_1, ϕ_2) 中两个类别 (红色和蓝色) 的数据点。左上图中的黑色箭头为初始参数向量 $\mathbf{w}$，黑线为相应的决策边界，箭头指向的区域为红色类别的决策区域。绿色圈出的数据点为错误分类点，因此其特征向量被添加到当前的权重向量中，从而在右上图中得到了新的决策边界。左下图展示的是下一个要考虑的错误分类点，由绿色圆圈表示，并将其特征向量再次添加到权重向量中，从而得到右下图中的决策边界，这次所有数据点的分类都正确了。

图 4.8: 这段是介绍历史，还特别长，就不翻译了哈。。。【逃跑】

4.2 概率生成模型

现在我们换个思路，从分类模型的概率视角入手，并展示如何从数据分布的简单假设得到线性决策边界。在第 1.5.4 节中，我们已经讨论了分类问题中判别模型和生成模型的区别。现在我们会采用生成模型，对类别条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 和 $\mathcal{C}_k$ 进行建模，然后利用它们根据贝叶斯定理计算后验概率 $p(\mathcal{C}_k|\mathbf{x})$。

首先研究二分类问题。类别 $\mathcal{C}_1$ 的后验概率可以表示为

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) \tag{4.57}$$

其中，

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \tag{4.58}$$

另外，$\sigma(a)$ 表示的是 logistic sigmoid 函数，定义为

$$\sigma(a) = \frac{1}{1+\exp(-a)} \tag{4.59}$$

其图像如图 4.9 所示。“sigmoid” 是 “S 型” 的意思。这一类型的函数有时也被称为 “挤压函数” (squashing function)，因为它可以将整个实轴映射到有限的区间内。logistic sigmoid 函数在早些时候已经提到过了，扮演了很多分类问题中的重要角色。

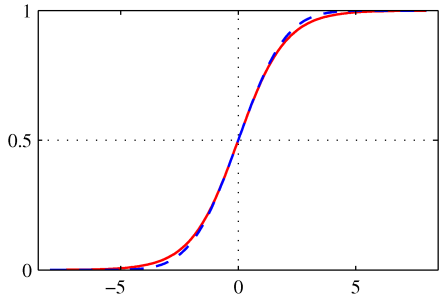

图 4.9: 公式 (4.59) 所定义的 logistic sigmoid 函数 $\sigma(a)$ 的函数图像 (红色曲线)，同时展示了经过放缩的逆概率函数 (inverse probit function)$\Phi(\lambda a)$，其中 $\lambda^2 = \pi/8$(蓝色虚线)，$\Phi(a)$ 的定义详见公式 (4.114)。放缩因子 $\pi/8$ 是为了让两个函数在 $a = 0$ 的时候具有相同的函数值。

【译者注：关于 logistic sigmoid 函数的翻译一直是个老大难问题。《统计学习方法》(李航) 采用了 “逻辑斯谛函数” 的直接音译，《机器学习》(周志华) 将 logistic function 称为 “对数几率函数”。译者私以为原本的英文说法更能让人体会到该函数的特点，故在这里保留了 logistic sigmoid 函数这样的写法。】

logistic sigmoid 函数满足如下的对称性质：

$$\sigma(-a) = 1 - \sigma(a) \tag{4.60}$$

这个很容易就可以证明了。logistic sigmoid 函数的反函数为

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right) \tag{4.61}$$

这个函数称为 logit 函数。它表示的是概率的比值 $\ln[p(\mathcal{C}_1|\mathbf{x})/p(\mathcal{C}_2|\mathbf{x})]$，同时也称为 log odds。

在 (4.57) 中，我们将后验概率写成了比较简单的等价形式，所以 logistic sigmoid 函数的定义可能显得比较空洞。不过，如果 $a(x)$ 的函数形式比较简单，那么这样做是很有意义的。接下来我们就会研究 $a(\mathbf{x})$ 为线性函数的情况，在这种情况下，后验概率是由广义线性模型得到的。

对于 $K > 2$ 的分类问题，

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{4.62}$$

这个公式为归一化指数 (normalized exponential)，可以看成是 logistic sigmoid 在多类别情况下的推广形式。其中，a_k 的值为

$$a_k = \ln(p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)) \tag{4.63}$$

归一化指数也被称为 softmax 函数。对于一切 $j \neq k$，如果 $a_k \gg a_j$，那么 $p(\mathcal{C}_k|\mathbf{x}) \approx 1$，$p(\mathcal{C}_j|\mathbf{x}) \approx 0$。

现在我们研究一个问题，如果将类别条件概率密度选取了特定的形式，会造成怎样的后果。首先研究 $\mathbf{x}$ 为连续输入变量的情况，然后研究离散输入的情况。

4.2.1 输入量为连续变量的情况

假设类别条件概率密度为高斯分布。在开始阶段，我们首先假设所有的类别都具有相同的协方差矩阵。于是，类别 $\mathcal{C}_k$ 的概率密度为

$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\} \tag{4.64}$$

对于二分类问题，根据 (4.57) 和 (4.58)，

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0) \tag{4.65}$$

其中，

$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{4.66}$$

$$w_0 = -\frac{1}{2}\boldsymbol{\mu}_1^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \tag{4.67}$$

可以看出由高斯概率密度函数的指数项而来的关于 $\mathbf{x}$ 的二次项消失了，因为我们做出了关于协方差矩阵的假设，于是 logistic sigmoid 函数的自变量就变成了关于 $\mathbf{x}$ 的线性函数。如图 4.10 所示的是当输入空间 $\mathbf{x}$ 为二维空间时的情况。由此得到的决策边界所对应的是后验概率 $p(\mathcal{C}_k|\mathbf{x})$ 为常数的曲面，由于是关于 $\mathbf{x}$ 的线性函数，所以决策边界也是输入空间中的线性函数。先验概率 $p(\mathcal{C}_k)$ 仅仅会对偏差参数 w_0 产生影响，所以先验的变化会使决策边界发生平移，也就是平移后验概率的常数轮廓线。

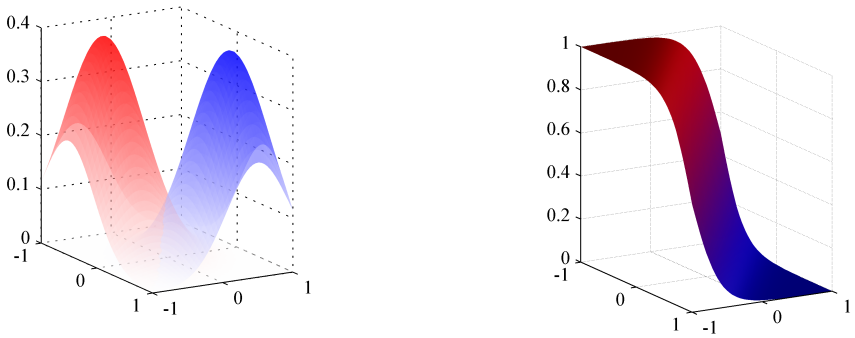

图 4.10: 左图展示的是两个类别的类别条件概率密度，分别表示为红色和蓝色。右图为对应的后验概率 $p(\mathcal{C}_1|\mathbf{x})$，它是一个以 $\mathbf{x}$ 的线性函数为自变量的 logistic sigmoid 函数。右图的曲面采用了渐变色，红色表示 $p(\mathcal{C}_1|\mathbf{x})$，蓝色表示 $p(\mathcal{C}_2|\mathbf{x}) = 1 - p(\mathcal{C}_2|\mathbf{x})$，所以整个曲面呈现了从红到蓝的变化过程。

对于 K 分类问题，根据 (4.62) 和 (4.63)，

$$a_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0} \tag{4.68}$$

其中，

$$\mathbf{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k \tag{4.69}$$

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln p(\mathcal{C}_k) \tag{4.70}$$

和之前的一样，$a_k(\mathbf{x})$ 又成了 $\mathbf{x}$ 的线性函数。于是，当两个最大的后验概率相等时，可以得到使得误分类率达到最小的决策边界，它是一个关于 $\mathbf{x}$ 的线性函数，于是最终得到的是一个广义线性模型。

如果我们将协方差矩阵的假设放宽，允许每个类别的条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 都有其各自的协方差矩阵 $\boldsymbol{\Sigma}_k$，那么上面函数中的二次项就无法消除了，最后得到的是一个关于 $\mathbf{x}$ 的二次函数，麻烦也随之升级，变成了二次判别分析 (quadratic discriminant)。线性决策边界和二次决策边界如图 4.11 所示。

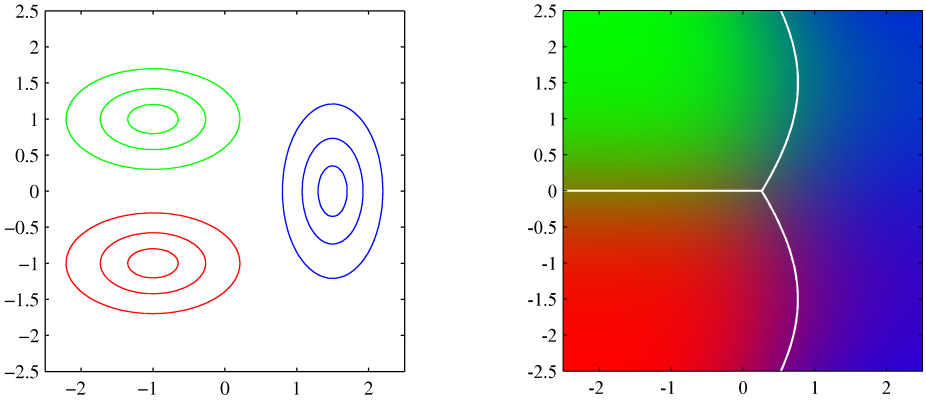

图 4.11: 左图显示了三个类别的类别条件概率密度，每个类别各自对应一个高斯分布，分别表示为红色，绿色和蓝色，其中红色和绿色的类别具有相同的协方差矩阵。右图展示了相应的后验概率，其中 RGB 颜色矢量分别相应三个类别的后验概率。同时也展示了决策边界。需要注意的是，在具有相同协方差矩阵的红色和绿色类别之间，决策边界是线性的，而其他的边界则是二次的。

4.2.2 最大似然方法

一旦我们为类别条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 指定了一个带参数的函数形式，配合类别先验概率 $p(\mathcal{C}_k)$，就可以利用最大似然方法确定函数中的参数值。这自然需要一个训练集，里面包含有 $\mathbf{x}$ 和它们各自对应的类别标签。

首先分析二分类问题，假设两个类别的条件概率密度都是高斯分布，而且具有相同的协方差矩阵，以及训练集为 $\{\mathbf{x}_n, t_n\}, n-1, ..., N$。当 $t_n = 1$ 时表示属于类别 $\mathcal{C}_1$，$t_n = 0$ 时表示属于类别 $\mathcal{C}_2$。设先验概率 $p(\mathcal{C}_1) = \pi$，那么 $p(\mathcal{C}_2)$ 自然为 $1-\pi$。对于某个属于 $\mathcal{C}_1$ 的数据 $\mathbf{x}_n$，有 $t_n = 1$，于是

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n|\mathcal{C}_1) = \pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

类似地，对于类别 $\mathcal{C}_2$，$t_n = 0$，于是

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

于是似然函数为

$$p(\mathbf{t}, \mathbf{x}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left[\pi\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})\right]^{t_n} \left[(1-\pi)\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})\right]^{1-t_n} \tag{4.71}$$

其中 $\mathbf{t} = (t_1, ..., t_N)^{\mathrm{T}}$。和往常一样，将其取对数之后再进行最大化会方便很多。首先关于 π 进行最大化。在对数似然函数中，与 π 有关的内容是

$$\sum_{n=1}^{N} \{t_n \ln\pi + (1-t_n)\ln(1-\pi)\} \tag{4.72}$$

令关于 π 的导数为 0 并整理，可以得到

$$\pi = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \tag{4.73}$$

其中 N_1 表示属于类别 $\mathcal{C}_1$ 的数据总数，N_2 表示属于类别 $calC_2$ 的数据总数。所以和预想的一样，π 的最大似然估计就是属于类别 $\mathcal{C}_1$ 的数据所占的比例。这个结论可以很轻松地推广到多分类问题中，即某一类别的先验概率等于属于该类别的数据在所有数据中所占的比例。

接下来计算关于 $\boldsymbol{\mu}_1$ 的最大似然。仍然是将和 $\boldsymbol{\mu}_1$ 有关的项挑出来，

$$\sum_{n=1}^{N} t_n \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2}\sum_{n=1}^{N} t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{const} \tag{4.74}$$

令关于 $\boldsymbol{\mu}_1$ 的导数为 0 并整理，可以得到

$$\boldsymbol{\mu}_1 = \frac{1}{N_1}\sum_{n=1}^{N} t_n \mathbf{x}_n \tag{4.75}$$

这事实上是所有属于类别 $\mathcal{C}_1$ 的输入向量 $\mathbf{x}_n$ 的均值。通过类似的过程，可以得到 $\boldsymbol{\mu}_2$ 的最大似然解

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - t_n) \mathbf{x}_n \tag{4.76}$$

也就是所有属于类别 $\mathcal{C}_2$ 的输入向量 $\mathbf{x}_n$ 的均值。

最后计算协方差矩阵的最大似然解。将对数似然函数中与 $\boldsymbol{\Sigma}$ 有关的项挑出来，

$$\begin{aligned}
&-\frac{1}{2}\sum_{n=1}^{N} t_n \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) \\
&-\frac{1}{2}\sum_{n=1}^{N} (1 - t_n) \ln|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{n=1}^{N} (1 - t_n) (\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_2) \\
&= -\frac{N}{2}\ln|\boldsymbol{\Sigma}| - \frac{N}{2}\mathrm{Tr}\left\{\boldsymbol{\Sigma}^{-1}\mathbf{S}\right\}
\end{aligned} \tag{4.77}$$

其中，

$$\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2 \tag{4.78}$$

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^{\mathrm{T}} \tag{4.79}$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^{\mathrm{T}} \tag{4.80}$$

利用高斯分布最大似然解的结论，可以看出 $\boldsymbol{\Sigma} = \mathbf{S}$，也就是说协方差矩阵的最大似然解是两个类别各自协方差矩阵的加权平均值。

这个结论也可以很轻易地推广到 K 分类问题中，当然要求各个类别的条件概率密度是具有相同协方差矩阵的高斯分布。需要注意的是，拟合成高斯分布的做法对异常值并不鲁棒，因为高斯分布的最大似然解是不够鲁棒的。

4.2.3 输入变量为离散变量的情况

现在来研究一下输入的特征值 x_i 为离散变量的情况。简单起见，首先分析二元输入变量 $x_i \in \{0, 1\}$ 的情况，然后简单讨论一下推广为一般情况后的结论。假设有 D 个输入，那么完整的分布就是一个包含有 2^D 个变量的表，其中 $2^D - 1$ 个为独立变量。由于随着特征数量的提升，这个数值会呈指数型上升，使得我们需要寻求其他能用的表示方法。这里我们做出朴素贝叶斯假设 (naive Bayes assumption)，即在给定类别 $\mathcal{C}_k$ 的条件下每个特征都是相互独立的。于是可以写出类别条件概率分布

$$p(\mathbf{x}|\mathcal{C}_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1 - \mu_{ki})^{1-x_i} \tag{4.81}$$

每个类别都包含了对应的 D 个独立参数。将其代入 (4.63)，可得

$$a_k(\mathbf{x}) = \sum_{i=1}^{D} \{x_i \ln \mu_{ki} + (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k) \tag{4.82}$$

这又是一个关于输入变量 x_i 的线性函数。对于 $K = 2$ 的情况，我们也可以利用 logistic sigmoid 公式 (4.57)。当离散输入变量的取值方式 $M > 2$ 时，结果也是类似的。

4.2.4 指数族分布

我们已经在从前的讨论中看到，不论是服从高斯分布的连续输入变量，还是离散变量，其分类的后验概率都是由带有 logistic sigmoid 激活函数 ($K = 2$) 或 softmax 激活函数 ($K > 2$) 的生成线性模型给出的。它们事实上都是某种一般结论的特殊情况，即假设类别条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 属于指数分布族的情况下得到的后验概率。

利用 (2.194) 中指数分布族的形式，可以看出 $\mathbf{x}$ 的分布可以写成

$$p(\mathbf{x}|\boldsymbol{\lambda}_k) = h(\mathbf{x})g(\boldsymbol{\lambda}_k) \exp\left\{\boldsymbol{\lambda}_k^{\mathrm{T}} \mathbf{u}(\mathbf{x})\right\} \tag{4.83}$$

现在将注意力集中在 $\mathbf{u}(\mathbf{x}) = \mathbf{x}$ 这样的分布上。利用 (2.236) 引入放缩参数 s，于是可以得到如下的属于指数分布族一个子集的条件概率密度函数

$$p(\mathbf{x}|\boldsymbol{\lambda}_k + s) = \frac{1}{s} h(\frac{1}{s}\mathbf{x}) g(\boldsymbol{\lambda}_k) \exp\left\{\frac{1}{s}\boldsymbol{\lambda}_k^{\mathrm{T}} \mathbf{x}\right\} \tag{4.84}$$

需要注意的是，每个类别都有各自的参数向量 $\boldsymbol{\lambda}_k$，但放缩参数 s 都设置成一样的。

对于二分类问题，我们将这个表达式代入 (4.58)，可以看出分类的后验概率仍然是以线性函数 $a(\mathbf{x})$ 为自变量的 logistic sigmoid 函数，

$$a(\mathbf{x}) = \frac{1}{s}(\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)^{\mathrm{T}}\mathbf{x} + \ln g(\boldsymbol{\lambda}_1) - \ln g(\boldsymbol{\lambda}_2) + \ln p(\mathcal{C}_1) - \ln p(\mathcal{C}_2) \tag{4.85}$$

类似地，对于 K 分类问题，将类别的条件概率密度代入 (4.63)，可以得到

$$a_k(\mathbf{x}) = \frac{1}{s}\boldsymbol{\lambda}_k^{\mathrm{T}}\mathbf{x} + \ln g(\boldsymbol{\lambda}_k) + \ln p(\mathcal{C}_k) \tag{4.86}$$

毫无疑问，又是一个关于 $\mathbf{x}$ 的线性函数。

4.3 概率判别模型

对于二分类问题，类别为 $\mathcal{C}_1$ 的后验概率可以写成以 $\mathbf{x}$ 的线性函数为自变量的 logistic sigmoid 函数的形式，而且类别条件概率密度【译者注：这里的原文是 class-conditional distributions，按理来说应该翻译成“类别条件概率分布”，但这个表达在整本书中仅出现了 3 次，而且这一段的后文马上又改回了“条件概率密度”的表达，所以这里翻译成条件概率密度。】$p(\mathbf{x}|\mathcal{C}_k)$ 有很多种选择。类似地，对于多分类问题，类别 $\mathcal{C}_k$ 的后验概率可以写成以 $\mathbf{x}$ 的线性函数为自变量的 softmax 变换的形式。对于某种指定的类别条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$，可以利用最大似然方法确定概率密度的参数和类别先验 $p(\mathcal{C}_k)$，然后利用贝叶斯定理确定后验概率。

不过，还可以利用另一种方法显式地表达出广义线性模型，然后直接利用最大似然方法确定其参数。这个方法称为迭代重加权最小二乘法 (IRLS, iterative reweighted least squares)，它同样也是一个非常有力的方法。

求取广义线性模型参数的间接方法一般是这样的，拟合类别条件概率密度和类别先验，然后利用贝叶斯定理计算，这是典型的生成模型方法 (generative modelling)，因为我们可以从边缘分布 $p(\mathbf{x})$ 中抽取 $\mathbf{x}$，从而生成数据。在直接方法中，则是要对一个关于条件分布 $p(\mathcal{C}_k|\mathbf{x})$ 的似然函数进行最大化，这个方法是典型的判别模型方法。使用判别模型的优势之一是需要确定的参数比生成模型要少，这个内容我们稍后就会看到。另外，它的预测性能也要更好一些，尤其是在假设的类别条件概率密度形式与真实分布相去甚远的情况下。

4.3.1 固定基底函数

在本章节到目前为止的内容中，我们研究的分类模型都是直接对原始的输入向量 $\mathbf{x}$ 进行操作的。不过，即使我们首先利用基底函数向量 $\boldsymbol{\phi}(\mathbf{x})$ 对输入量进行固定的非线性变换，之前所讨论的算法也同样适用。在这种情况下所得到的决策边界将

会是特征空间 ϕ 中的线性函数，当然，在初始的 $\mathbf{x}$ 空间中肯定是非线性的了，如图 4.12 所示。在特征空间 $\phi(\mathbf{x})$ 中线性可分的类别并不需要在 $\mathbf{x}$ 空间中也是线性可分的。需要注意的是，之前我们在讨论线性回归模型时也提到过，基底函数中总有一个会被设置为常数，比如 $\phi_0(\mathbf{x}) = 1$，所以它对应的参数 w_0 就是之前总提到的偏差 bias。在本章的剩余内容中我们会使用固定的基底函数变换 $\phi(\mathbf{x})$，这样一来就可以表现出与第 3 章中的回归模型相通的地方了。

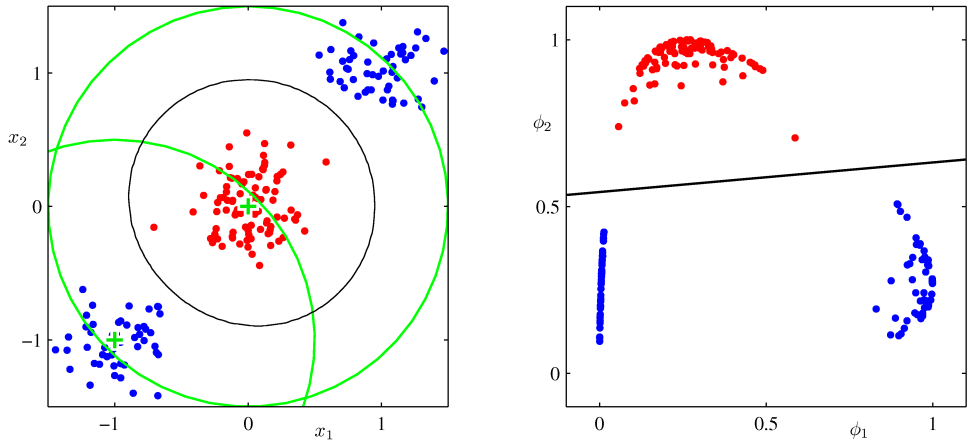

图 4.12: 非线性基底函数在线性分类模型中的应用。左图中展示了原始的输入空间 (x_1, x_2) 和两类数据 (红色和蓝色)。同时在这个空间中定义了两个“高斯”基底函数 $\phi_1(\mathbf{x})$ 和 $\phi_2(\mathbf{x})$，其中心位于绿色的“+”，其轮廓为图中绿色的圆。右图中展示的是对应的特征空间 (ϕ_1, ϕ_2) 和利用 logistic 回归模型 (详见第 4.3.2 节) 确定的线性决策边界。这条决策边界在原始空间中对应的是一个非线性的决策边界，也就是左图中黑色的曲线。

在实际问题中，类别条件概率密度 $p(\mathbf{x}|\mathcal{C}_k)$ 之间会存在明显的重叠，使得后验概率 $p(\mathcal{C}_k|\mathbf{x})$ 对于某些 $\mathbf{x}$ 并不能得到 0 或 1 的结果。在这样的情况下，可以对后验概率进行精确的建模，然后利用第 1 章中决策论的内容得到最优解。需要注意的是，非线性变换 $\phi(\mathbf{x})$ 并不能解决类别重叠的问题，反而有可能使重叠的现象更加严重，甚至会在原始空间中没有出现重叠的地方搞出重叠来。不过，合适的非线性变换是可以简化后验概率建模的过程的。

这样的固定基底函数有明显的局限性，在后面的章节中我们会使基底函数能够适应数据，从而解决这个问题。即使有这样那样的不足，固定基底函数模型仍然在实际应用中扮演了重要的角色，而且可以引出很多更加复杂的模型所需要的重要概念。

4.3.2 logistic 回归

一如既往，仍然从二分类问题入手来研究广义线性模型。通过在第 4.2 节中讨论的生成方法，在一般的假设条件下，类别 $\mathcal{C}_1$ 的后验概率可以写成以特征向量 ϕ 为自变量的 logistic sigmoid 函数的形式，于是

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma(\mathbf{w}^{\mathrm{T}}\phi) \tag{4.87}$$

以及 $p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi)$。其中 $\sigma(\cdot)$ 为 logistic sigmoid 函数，也就是 (4.59)。在统计学中，这个模型称为 logistic 回归模型，尽管这是一个分类模型而非回归模型。

对于 M 维的特征空间 ϕ，这个模型中含有 M 个参数。相比之下，如果利用高斯条件概率密度建模，然后用最大似然方法计算参数，那就需要确定 $2M$ 个关于均值的参数和 $M(M+1)/2$ 个关于协方差矩阵的参数。算上先验 $p(\mathcal{C}_1)$，足足要确定 $M(M+5)/2+1$ 个参数，这个数量级可是 M 的平方，比 logistic 回归中要确定的参数数量多到不知道哪里去了。对于较大的 M，直接利用 logistic 回归模型更是具有极大的优势。

现在利用最大似然方法确定 logistic 回归模型中的参数。首先对 logistic sigmoid 函数求导，将结果表示为 sigmoid 函数自身的形式要更加简单：

$$\frac{\mathrm{d}\sigma}{\mathrm{d}a} = \sigma(1-\sigma) \tag{4.88}$$

对于数据集 $\{\phi_n, t_n\}$，其中 $t_n \in \{0,1\}, \phi_n = \phi(\mathbf{x}_n), n = 1, ..., N$，似然函数可以写成

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n} \tag{4.89}$$

其中 $\mathbf{t} = (t_1, ..., t_N)^{\mathrm{T}}$，$y_n = p(\mathcal{C}_1|\phi_n)$。和往常一样，可以对似然函数求负对数然后将其定义为误差函数，实际上得到的是如下的交叉熵误差函数 (cross-entropy error function)：

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} \tag{4.90}$$

其中 $y_n = \sigma(a_n), a_n = \mathbf{w}^{\mathrm{T}}\phi_n$。对误差函数关于 $\mathbf{w}$ 求梯度，可以得到

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n \tag{4.91}$$

其中利用了 (4.88)。可以看出，关于 logistic sigmoid 导数的项都小时了，使得对数似然函数梯度的形式变得非常简单。特别地，某个数据 n 对梯度造成的影响是“误差” $(y_n - t_n)$(也就是目标值和模型预测值之间的差异) 乘以基底函数向量 ϕ_n。此外，

与 (3.13) 对比一下可以看出，这个梯度的形式与线性回归模型中的平方和误差函数的梯度具有相同的形式。

如果需要的话，可以利用 (4.91) 得到顺序算法，其中的权重向量可以利用 (3.22) 进行更新，其中的 ∇E_n 为 (4.91) 中的第 n 项。

值得注意的是，对于线性可分的数据集，最大似然方法可能会产生严重的过拟合现象。这是因为最大似然解所对应的是 $\sigma = 0.5$ 的超平面，也就是 $\mathbf{w}^{\mathrm{T}}\phi = 0$ 的情况，从而将两个类别分开，而且 $\mathbf{w}$ 的大小趋近于无限大。在这种情况下，logistic sigmoid 函数在特征空间中会变得无比陡峭，形成一个跳变的阶梯函数，所以每个类别 k 中的数据都会得到 $p(\mathcal{C}_k|\mathbf{x}) = 1$ 的后验概率。此外，通常来讲，这些解之间是存在连续性的，因为任何的分割超平面都会对训练数据给出相同的后验概率，如图 10.13 所示。最大似然方法无法区分解的优劣，而且具体得到哪个解，要取决于优化的方法和初始参数的选择。需要注意的是，只要数据集是线性可分的，那么即使数据点的数量比模型中参数的数量大很多，也很容易出现过拟合的问题。通过引入先验并计算 $\mathbf{w}$ 的最大后验概率解，或者等价地在误差函数中添加正则项，可以避免这个问题。

4.3.3 迭代重加权最小二乘法

在第 3 章中讨论线性回归模型时，假设噪声为高斯模型条件下的最大似然解是可以得到闭式解的。这是由于对数似然函数是参数向量 $\mathbf{w}$ 的二次函数。对于 logistic 回归而言，闭式解就不复存在了，因为 logistic sigmoid 函数是非线性函数。不过这个影响不是很大，因为误差函数是凸函数，所以具有唯一的最小值。此外，误差函数可以使用基于 Newton-Raphson 优化的迭代计算方式求取对数似然函数的局部二次近似。对于最小化目标函数 $E(\mathbf{w})$，Newton-Raphson 更新的形式为 (Fletcher, 1987; Bishop and Nabney, 2008)

$$\mathbf{w}^{(\mathrm{new})} = \mathbf{w}^{(\mathrm{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) \tag{4.92}$$

其中 $\mathbf{H}$ 为 Hessian 矩阵，其元素是 $E(\mathbf{w})$ 关于 $\mathbf{w}$ 各个分量的二阶导数。

【译者注：求取 Hessian 矩阵的公式为：】

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

首先对线性回归模型 (3.3) 的平方和误差函数 (3.12) 使用 Newton-Raphson 方法。该误差函数的梯度和 Hessian 矩阵为

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n - t_n)\boldsymbol{\phi}_n = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\mathbf{w} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{4.93}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \boldsymbol{\phi}_n \boldsymbol{\phi}_n^{\mathrm{T}} = \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} \tag{4.94}$$

其中 $\boldsymbol{\Phi}$ 为 $N \times M$ 维的设计矩阵，其第 n 行为 $\boldsymbol{\phi}_n^{\mathrm{T}}$。那么 Newton-Raphson 更新的形式可以写成

$$\begin{aligned} \mathbf{w}^{(\mathrm{new})} &= \mathbf{w}^{(\mathrm{old})} - \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1} \left\{\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\mathbf{w}^{(\mathrm{old})} - \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\right\} \\ &= \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \end{aligned} \tag{4.95}$$

这正是标准的最小二乘解。需要注意的是，此时的误差函数是二次函数，所以 Newton-Raphson 公式可以给出每一步的确切解。

接下来对 logistic 回归模型的交叉熵误差函数 (4.90) 使用 Newton-Raphson 方法。根据 (4.91)，该误差函数的梯度和 Hessian 矩阵为

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\boldsymbol{\phi}_n = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t}) \tag{4.96}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n)\boldsymbol{\phi}_n \boldsymbol{\phi}_n^{\mathrm{T}} = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi} \tag{4.97}$$

其中利用了公式 (4.88)。另外还引入了 $N \times N$ 维的对角矩阵 $\mathbf{R}$，其对角元素为

$$R_{nn} = y_n(1 - y_n) \tag{4.98}$$

可以看出，Hessian 矩阵不再是一个常数了，而是通过权重矩阵 $\mathbf{R}$ 与 $\mathbf{w}$ 扯上了关系，于是乎误差函数也不再是一个二次函数了。根据 logistic sigmoid 函数的形式，有 $0 < y_n < 1$，所以对于任意的向量 $\mathbf{u}$ 都有 $\mathbf{u}^{\mathrm{T}}\mathbf{H}\mathbf{u} > 0$，所以 Hessian 矩阵 $\mathbf{H}$ 是正定的。这表明误差函数是一个关于 $\mathbf{w}$ 的凸函数，所以是有唯一最小值的。

于是，logistic 回归模型的 Newton-Raphson 更新公式可以写成

$$\begin{aligned} \mathbf{w}^{(\mathrm{new})} &= \mathbf{w}^{(\mathrm{old})} - (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t}) \\ &= (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1} \left\{\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi}\mathbf{w}^{(\mathrm{old})} - \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t})\right\} \\ &= (\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{z} \end{aligned} \tag{4.99}$$

其中 $\mathbf{z}$ 为 N 维向量，其元素为

$$\mathbf{z} = \boldsymbol{\Phi}\mathbf{w}^{(\mathrm{old})} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t}) \tag{4.100}$$

可以看出，更新公式 (4.99) 其实是一个加权最小二乘问题中的一堆正规方程构成的集合。由于权重矩阵 $\mathbf{R}$ 依赖于参数向量 $\mathbf{w}$ 而非常数，所以必须要迭代使用正规方程，每次迭代都利用新的权重向量 $\mathbf{w}$ 计算新的权重矩阵 $\mathbf{R}$。所以这个算法称为迭代重加权最小二乘法 (iterative reweighted least squares, IRLS, Rubin, 1983)。与加权最小二乘问题一样，对角权重矩阵 $\mathbf{R}$ 的元素可以看成是变量，因为 logistic 回归模型中的 t 的均值和方差为

$$\mathbb{E}[t] = \sigma\mathbf{x} = y \tag{4.101}$$

$$\mathrm{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y) \tag{4.102}$$

其中利用了当 $t \in \{0, 1\}$ 时 $t^2 = t$ 的性质。实际上，我们可以将 IRLS 看成是变量 $a = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}$ 空间中线性问题的解。$\mathbf{z}$ 中的第 n 个元素 z_n 可以看成是在这个空间中，通过对 logistic sigmoid 函数在 $\mathbf{w}^{(\mathrm{old})}$ 附近进行局部线性近似得到的目标值：

$$\begin{aligned} a_n(\mathbf{w}) &\approx a_n(\mathbf{w}^{(\mathrm{old})}) + \left.\frac{\mathrm{d}a_n}{\mathrm{d}y_n}\right|_{\mathbf{w}^{(\mathrm{old})}} (t_n - y_n) \\ &= \boldsymbol{\phi}^{\mathrm{T}}\mathbf{w}^{(\mathrm{old})} - \frac{(y_n - t_n)}{y_n(1-y_n)} = z_n \end{aligned} \tag{4.103}$$

4.3.4 多分类 logistic 回归

在讨论多分类问题的生成模型时。我们已经看到，对于一类分布而言，后验概率是由特征变量线性函数的 softmax 变换得到的，于是

$$p(\mathcal{C}_k|\boldsymbol{\phi}) = y_k(\boldsymbol{\phi}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{4.104}$$

其中的“激励”【我怎么知道会用到自动控制原理的名词。。。】a_k 为

$$a_k = \mathbf{w}_k^{\mathrm{T}}\boldsymbol{\phi} \tag{4.105}$$

在这里我们利用最大似然方法来分别确定类别条件概率密度和类别先验，然后利用贝叶斯定理求取相应的后验概率，从而确定参数 $\{\mathbf{w}_k\}$。不过现在，我们可以直接利用最大似然方法确定模型参数 $\{\mathbf{w}_k\}$。首先要做的第一件事是将 y_k 关于所有的激励 a_j 求偏导，于是

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \tag{4.106}$$

其中 I_{kj} 为单位矩阵的元素。

然后写出似然函数。在这个问题中采用 1-of-K 编码是最方便的，于是似然函数为

$$p(\mathbf{T}|\mathbf{w}_1, ..., \mathbf{w}_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\boldsymbol{\phi}_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}} \tag{4.107}$$

其中 $y_{nk} = y_k(\boldsymbol{\phi}_n)$。$\mathbf{T}$ 是 $N \times K$ 维的矩阵，其元素 t_{nk} 为目标变量。对这个函数取负对数，

$$E(\mathbf{w}_1, ..., \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, ..., \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk} \tag{4.108}$$

这就是多分类问题的交叉熵误差函数。

现在将误差函数关于其中一个参数向量 $\mathbf{w}_j$ 求梯度。利用 (4.106) 中 softmax 函数求导的结果，可以得到

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, ..., \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\boldsymbol{\phi}_n \tag{4.109}$$

其中用到了 $\sum_k t_{nk} = 1$ 的性质。此刻涛声依旧【拖走】，我们所得到的梯度与线性模型的平方和误差函数、logistic 回归模型的交叉熵误差函数具有相同的形式，即误差 $(y_{nj} - t_{nj})$ 与基底函数 $\boldsymbol{\phi}_n$ 的乘积。再次涛声依旧【再拖走】，我们可以利用这个公式来构建顺序算法，仍然利用 (3.22) 来更新权重向量。

很明显，线性回归模型在数据点 n 处关于参数向量 $\mathbf{w}$ 的对数似然函数的形式为“误差” $y_n - t_n$ 乘以特征向量 $\boldsymbol{\phi}_n$。类似地，对于 logistic sigmoid 激活函数和它的交叉熵误差函数 (4.90)，以及 softmax 激活函数和它的多分类交叉熵误差函数 (4.108) 都具有类似的形式。它们都是一个更加一般的结果的特殊情况，我们会在第 4.3.6 节中看到这个内容。

为了求出可以批量处理的算法，我们再次利用 Newton-Raphson 更新来确定多分类问题的 IRLS 算法。这需要求一个 $M \times M$ 维的 Hessian 矩阵，其位于 (j, k) 位置的元素为

$$\nabla_{\mathbf{w}_k}\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, ..., \mathbf{w}_K) = -\sum_{n=1}^{N} y_{nk}(I_{kj} - y_{nj})\boldsymbol{\phi}_n \boldsymbol{\phi}_n^{\mathrm{T}} \tag{4.110}$$

和二分类问题一样，多分类 logistic 回归模型的 Hessian 矩阵是正定的，所以误差函数同样是具有唯一最小值的。多分类问题中 IRLS 实际应用的相关细节可以在参考文献 Bishop and Nabney(2008) 中找到。

4.3.5 probit 回归

我们已经看到，在指数分布族这个广泛的类别条件概率分布范围中，最后得到的后验概率是特征变量线性函数的 logistic 变换 (或者 softmax 变换)。不过，并不是所有的类别条件概率密度都会得到这么简单的后验概率 (比如高斯混合模型)。所以寻找其他类型的概率判别模型或许是更好的选择。在本节中，我们会回到二分类问题的研究中，以及重新回到广义线性模型的基本框架，即

$$p(t = 1|a) = f(a) \tag{4.111}$$

其中 $a = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}$，$f(\cdot)$ 为激活函数。

确定联系函数的一种方式是通过噪声阈值模型。对于任意的输入 $\boldsymbol{\phi}_n$，令 $a_n = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}_n$，并将目标值设置为

$$\begin{cases} t_n = 1 & \text{if } a_n \geqslant \theta \\ t_n = 0 & \text{otherwise} \end{cases} \tag{4.112}$$

如果 θ 的值来自于概率密度 $p(\theta)$，那么相应的激活函数将是一个分布函数

$$f(a) = \int_{-\infty}^{a} p(\theta)\mathrm{d}\theta \tag{4.113}$$

如图 4.13 所示。

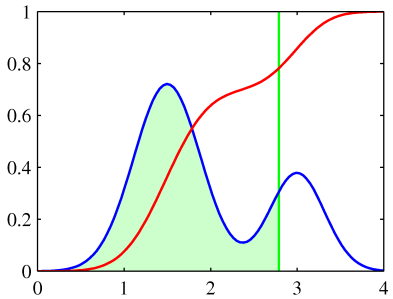

图 4.13: 概率密度函数 $p(\theta)$ 如图中的蓝色曲线所示，该概率密度是一个高斯混合模型，包含有 2 个分量。图中的红色曲线为概率分布函数。其中，蓝色曲线对应的函数值 (例如垂直绿色直线处) 是该点处红色曲线的斜率，而红色曲线对应的函数值则等于蓝色曲线下方绿色阴影部分的面积。在随机阈值模型中，如果 $a = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}$ 的值超过阈值则将类别标签标记为 $t = 1$，否则 $t = 0$。这与分布函数 $f(a)$ 给出的激活函数是等价的。

举一个特殊的例子，假设概率密度 $p(\theta)$ 是均值为 0、方差为 1 的高斯分布。于是相应的概率分布函数为

$$\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1)\mathrm{d}\theta \tag{4.114}$$

这就是逆 probit 函数。它与 logistic sigmoid 函数相似，都是 S 型的，其图像如图 4.9 所示。需要注意的是，即使使用不这么特殊的高斯分布，也不会对这个模型造成影响，因为它对应的是一个经过缩放的线性系数 $\mathbf{w}$。许多用于计算这个函数的数值计算包都与下面的这个函数紧密相关：

$$\mathrm{erf}(a) = \frac{2}{\sqrt{\pi}} \int_{0}^{a} \exp(-\theta^2/2)\mathrm{d}\theta \tag{4.115}$$

这个函数称为 erf 函数或者 error 函数 (与误差函数不是一个东西哈)。它对应的逆 profit 函数为

$$\Phi(a) = \frac{1}{2}\left\{1 + \frac{1}{\sqrt{2}}\mathrm{erf}(a)\right\} \tag{4.116}$$

基于 profit 激活函数的广义线性模型称为逆 probit 回归模型。

我们可以通过最大似然方法来确定这个模型的参数。在实际应用中，利用 probit 回归得到的结果与 logistic 回归类似，不过我们会在第 4.5 节中看到，在研究贝叶斯观点下的 logistic 回归时，probit 模型还有另外的一些用途。

在实际应用中，异常值是一个很大的问题，在输入向量 $\mathbf{x}$ 的检测出现错误，或者目标值 t 的标注出现错误时，都会出现异常值。由于这样的数据会对决策边界产生严重的不利影响，所以对分类器的影响也是相当恶劣的。需要注意的是，logistic 回归模型和 probit 回归模型在这个问题上的表现很是不同，因为当 $x \to \infty$ 时，logistic sigmoid 函数的衰减是 $\exp(-x)$ 量级的，而 probit 激活函数则是 $\exp(-x^2)$，所以 probit 模型对异常值更加敏感。

不过，logistic 模型和 probit 模型都建立在同一个假定之上，即直接利用了数据的标签。这样一来，错误标定会很容易对概率模型造成影响。在这里假设误分类的概率为 ϵ(Opper and Winther, 2000a)，那么数据 $\mathbf{x}$ 相应目标值的分布为

$$\begin{aligned} p(t|\mathbf{x}) &= (1-\epsilon)\sigma(\mathbf{x}) + \sigma(1-\sigma(\mathbf{x})) \\ &= \epsilon + (1-2\epsilon)\sigma(\mathbf{x}) \end{aligned} \tag{4.117}$$

其中 $\sigma(\mathbf{x})$ 为输入向量 $\mathbf{x}$ 的激活函数。其中 ϵ 的值可以提前设定，也可以当成一个超参数从数据中推断得到。

4.3.6 标准联系函数

对于带有高斯噪声的线性回归模型，