第 5 章

线 性 模 型

数据挖掘的本质是从已知的经验数据对中，挖掘出一个最优的表示模型，该模型可以最大限度地描述经验数据中输入输出的关系。例如，医生通过病人是否发烧、咳嗽、呼吸困难、浑身乏力，胸部 X 光片有无磨玻璃影表现诊断是否为新冠肺炎。而数据挖掘根据医生提供的这些经验数据，挖掘出一个决策模型，模型输入为患者的上述病症的临床表现，据此综合判定是否为新冠阳性。在模型挖掘之前，如何描述映射模型的形式，是数据挖掘任务所要解决的第一个问题。线性模型作为最简单的模型形式，是数据挖掘中最基础的表示模型。本章内容会分别对线性模型的两种应用场景——回归与分类任务进行介绍，并给出对应模型的挖掘方法。

5.1 基 本 形 式

假设具有 d 个属性的模式 $\boldsymbol{x}=[x_1,x_2,\cdots,x_d]^{\mathrm{T}}$，其中，$x_i$ 为模式 $\boldsymbol{x}$ 的第 i 个属性。例如，以每一个位置的像素值描述一幅图像，x_i 表示不同位置的像素值。用线性模型表示函数关系，可以定义为如下方式。

$$f(\boldsymbol{x})=w_1x_1+w_2x_2+\cdots+w_dx_d+b \tag{5.1}$$

其中，$w_1,w_2,\cdots,w_d$，b 为线性函数参数。通过选择不同函数参数，可以实现不同函数关系的映射。式（5.1）可以简化为如下的向量形式。

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{5.2}$$

其中，$\boldsymbol{w}=[w_1,w_2,\cdots,w_d]^{\mathrm{T}}$。

线性模型形式简单、易于建模，却蕴含着数据挖掘中一些重要的基本思想，许多功能更为强大的非线性模型可以在线性模型的基础上，通过引入层级结构或高维映射而得到。此外，由于参数 w 直观地表达了各属性在预测中的重要性，因此线性模型具有良好的可解释性，便于理解和分析数据挖掘过程中各特征对结果的影响。例如，分别以身高 x_h、体重 x_w、肤色 x_c 三种属性预测某人的颜值，线性函数 $f(\boldsymbol{x})=0.4x_h+0.3x_w+0.3x_c+1$ 表示身高属性对颜值的影响最大，身高越高，颜值越高。

线性函数模型简单，相对而言，其函数表达能力也会受到一定限制。如图 5.1 所示，散点所示的函数关系，无法通过线性模型进行拟合。为了提高模型的表示能力，通常的做法是将线性模型输出经过一个非线性映射，具体如式（5.3）所示，以此提高模型的非线性表示能力，如图 5.2 所示。

$$f(\boldsymbol{x})=g(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b) \tag{5.3}$$

其中，$g(x)$ 为非线性映射函数，常用的形式有 Sigmoid 函数 $g(x)=\dfrac{1}{1+\mathrm{e}^{-x}}$、tanh 函数 $g(x)=\dfrac{\mathrm{e}^{x}-\mathrm{e}^{-x}}{\mathrm{e}^{x}+\mathrm{e}^{-x}}$。

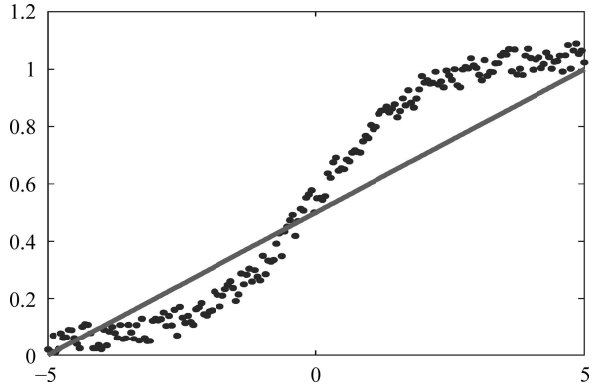

图 5.1　数据的线性表示

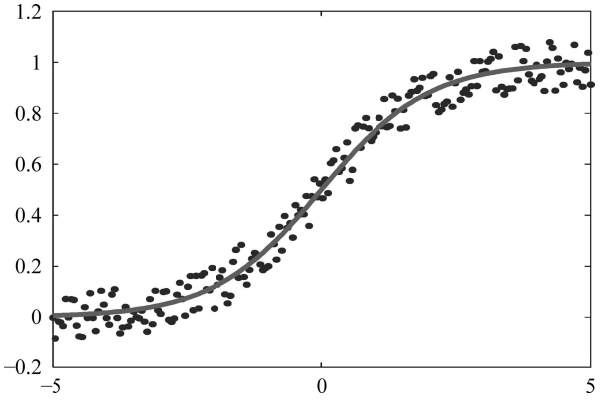

图 5.2　数据的非线性化表示

5.2　线 性 回 归

线性模型给定了学习函数的表达形式，确定函数的形式后，如何根据已知数据的特性确定函数的具体参数是数据挖掘中所需解决的核心问题。假设给定一组如图 5.3 所示的蓝色散点集合 $D=\{(\boldsymbol{x}_i,y_i),i=1,2,\cdots,N\}$，线性回归任务就是寻找如图中所示红色直线，使得当前直线尽可能准确地描述当前数据的输入输出关系。

假设图中直线的表示形式为

$$f(x)=wx+b \tag{5.4}$$

确定函数形式后，下一步任务是需要根据当前数据确定函数参数 w、b 的值。

最优的函数参数应使学习得到的函数尽可能地符合已知数据的映射关系，也就是说，使得函数预测值 $f(x_i)$ 与当前位置的真实值 y_i 尽可能相等。因此，可采用均方误差衡量预测函数与真实数据的差异性，均方误差最小时对应的参数为最优函数参数。

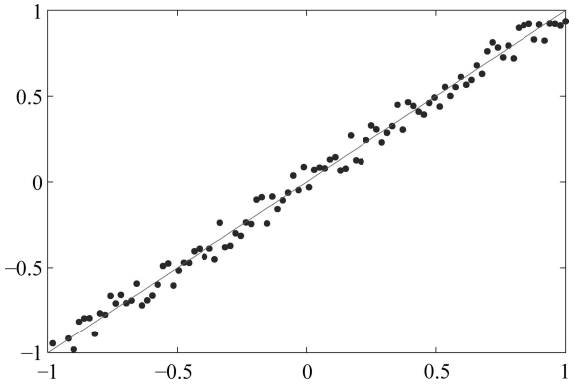

图 5.3　线性回归

$$E=\frac{1}{N}\sum_{i=1}^{N}(f(x_i)-y_i)^2=\frac{1}{N}\sum_{i=1}^{N}(wx_i+b-y_i)^2 \tag{5.5}$$

均方误差 E 是关于参数 w、b 的二次函数，函数在导数为 0 处具有全局最小值。因此，w、b 的最优值为 $\frac{\partial E}{\partial w}=0$、$\frac{\partial E}{\partial b}=0$ 对应的值，即

$$\frac{\partial E}{\partial w}=g_1(w,b)=\frac{2}{N}\sum_{i=1}^{N}(wx_i+b-y_i)x_i=0 \tag{5.6}$$

$$\frac{\partial E}{\partial w}=g_2(w,b)=\frac{2}{N}\sum_{i=1}^{N}(wx_i+b-y_i)=0 \tag{5.7}$$

联合方程 $g_1(w,b)=0$、$g_2(w,b)=0$，可解得：

$$w=\frac{\sum_{i=1}^{N}y_i(x_i-\overline{x})}{\sum_{i=1}^{N}x_i^2-\frac{1}{N}(\sum_{i=1}^{N}x_i)^2} \tag{5.8}$$

$$b=\frac{1}{N}\sum_{i=1}^{N}(y_i-wx_i) \tag{5.9}$$

其中，$\overline{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$ 为 x 的均值。

上述线性函数的输入属性仅有一维，而实际中描述一个模式通常需要较高的维度，例如，图像的描述需要用所有位置的像素值。因此，更一般的线性回归模型，也被称为多元线性回归，可表示为如下形式。

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \quad \text{使得 } f(\boldsymbol{x}_i)\cong y_i,\ i=1,2,\cdots,N \tag{5.10}$$

采用矩阵的形式简化表述上述问题：

$$f(\boldsymbol{X})=\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}\cong\boldsymbol{y} \tag{5.11}$$

其中，$\tilde{\boldsymbol{w}}=[\boldsymbol{w};b]\in R^{d+1}$，$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N\\ 1 & 1 & \cdots & 1\end{bmatrix}\in R^{(d+1)\times N}$，$\boldsymbol{y}=[y_1,y_2,\cdots,y_N]\in R^N$。

上述模型的均方误差可表示为

$$E=\frac{1}{N}(\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{y})(\tilde{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{y})^{\mathrm{T}} \tag{5.12}$$

上述均方误差 E 同样是关于参数 $\tilde{\boldsymbol{w}}$ 的二次函数。令 $\frac{\partial E}{\partial \tilde{\boldsymbol{w}}}=0$，可得：

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\tilde{\boldsymbol{w}}=\boldsymbol{X}\boldsymbol{y}^{\mathrm{T}} \tag{5.13}$$

当 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 为满秩矩阵或正定矩阵，可得：

$$\tilde{\boldsymbol{w}}=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}\boldsymbol{y}^{\mathrm{T}} \tag{5.14}$$

其中，$(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}$ 为逆矩阵。

学习得到上述参数向量 $\tilde{\boldsymbol{w}}$ 后，线性函数模型则可表示为

$$f(\boldsymbol{x})=\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}=\boldsymbol{y}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\tilde{\boldsymbol{x}} \tag{5.15}$$

其中，$\tilde{\boldsymbol{x}}=[\boldsymbol{x};1]$。

多元线性回归算法总结如图 5.4 所示

输入：原始数据 $\{(\boldsymbol{x}_i,y_i),i=1,2,\cdots,N\}$，$\boldsymbol{x}_i\in R^d$

算法流程：

1. 构造输入增广矩阵：$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N\\ 1 & 1 & \cdots & 1\end{bmatrix}\in R^{(d+1)\times N}$

2. 构造输出向量：$\boldsymbol{y}=[y_1,y_2,\cdots,y_N]\in R^N$

3. 计算线性回归参数：$\tilde{\boldsymbol{w}}=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}\boldsymbol{y}^{\mathrm{T}}$

4. 给定输入 $\boldsymbol{x}$，调整为增广形式：$\tilde{\boldsymbol{x}}=[\boldsymbol{x};1]$

5. 利用回归模型预测输出：$f(\boldsymbol{x})=\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}=\boldsymbol{y}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\tilde{\boldsymbol{x}}$

图 5.4 多元回归算法流程

【例 5.1】 气温预测任务：已知过往 5 天的气温数据为(1, 1)、(2, 2)、(3, 3)、(4, 4)、(5, 5)，其中，第一个值表示第几天，第二个值表示当天温度。采用线性回归模型预测第 6 天的温度。

【解析】 首先，通过线性模型表示气温与天数的关系：

$$\boldsymbol{y}=\boldsymbol{w}\boldsymbol{x}+b=[\boldsymbol{w},b][\boldsymbol{x};1]=\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}$$

（1）根据线性回归算法构造输入增广矩阵。

$$\boldsymbol{X}=\begin{bmatrix}1 & 2 & 3 & 4 & 5\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}\in R^{2\times 5}$$

（2）构造输出向量。

$$\boldsymbol{y}=[1,2,3,4,5]\in R^5$$

（3）计算线性回归参数。

$$\tilde{\boldsymbol{w}}=[w,b]^{\mathrm{T}}=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}\boldsymbol{y}^{\mathrm{T}}=\begin{bmatrix}55 & 15\\ 15 & 5\end{bmatrix}^{-1}\begin{bmatrix}1 & 2 & 3 & 4 & 5\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}1 & 2 & 3 & 4 & 5\end{bmatrix}^{\mathrm{T}}=\begin{bmatrix}1\\ 0\end{bmatrix}$$

（4）输入第 6 天，将输入调整为增广形式。

$$\tilde{\boldsymbol{x}}^{*}=\begin{bmatrix}6 & 1\end{bmatrix}^{\mathrm{T}}$$

（5）利用线性模型预测输出。

$$f(\boldsymbol{x})=\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}=\begin{bmatrix}1\\0\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}6\\1\end{bmatrix}=6$$

5.3 线 性 分 类

线性回归模型解决了输入输出之间映射关系的学习方法，但线性回归模型却并不适用于分类问题。分类问题的本质是对可能发生的各个事件的概率进行预测，预测类别为概率最大对应的事件。具体以二分类为例，其输出标记为{0，1}，0 表示一类，1 表示另一类。二分类的输出为离散值，而线性模型预测结果为连续值，无法直接通过线性模型进行分类问题。为了使线性模型适应于分类问题，可将线性模型的输出作为阶跃函数的输入。

$$y=\begin{cases}0, & z<0\\ 0.5, & z=0\\ 1, & z>0\end{cases} \tag{5.16}$$

其中，z 为线性模型的输出。若 z 为正值，则判定输入为正类别；z 为负值，则判定为负类别；z 为 0 时，可判定为任意类别。

阶跃函数在 0 值附近具有不连续特性，无法通过梯度下降等方式求解。因此，常用如下所示的 Sigmoid 函数代替阶跃函数：

$$g(z)=\frac{1}{1+\mathrm{e}^{-z}} \tag{5.17}$$

其函数形状如图 5.5 所示。

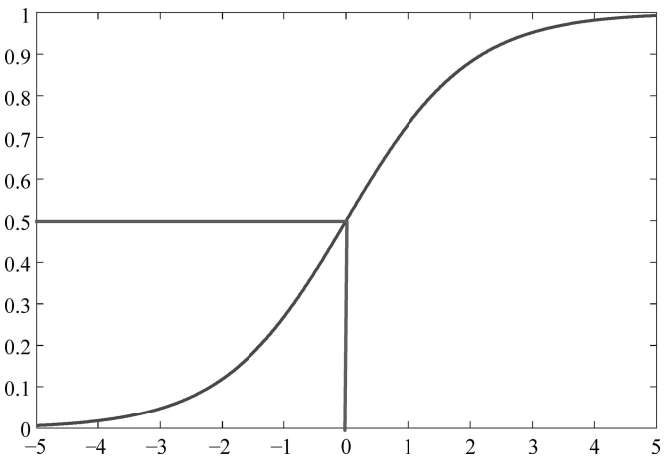

图 5.5 Sigmoid 函数

Sigmoid 函数的输出范围为 0～1，可以将其解释为样本被预测为某一类的概率。假设线性模型经过 Sigmoid 函数映射后为样本被判别为类别 1 的概率，其数学模型为

$$P(y=1\mid\boldsymbol{x})=g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}) \tag{5.18}$$

由于为二分类问题，所以样本被判别为类别 0 的概率为

$$P(y=0\mid\tilde{\boldsymbol{x}})=1-P(y=1\mid\tilde{\boldsymbol{x}})=1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}) \tag{5.19}$$

分类问题可以转换为上述概率值的比较，如果 $P(y=1\mid\tilde{\boldsymbol{x}})>P(y=0\mid\tilde{\boldsymbol{x}})=1-P(y=1\mid\tilde{\boldsymbol{x}})$，即 $P(y=1\mid\tilde{\boldsymbol{x}})>0.5$，则样本被判定为类别 1，否则被判定为类别 0。

针对上述概率预测模型的学习问题，可以采用极大自然估计方法实现对于参数的学习，其基本思想是已发生的事件在最优参数下对应的概率最大。对具有 N 个样本的集合 $D=\{(\boldsymbol{x}_i,y_i),i=1,2,\cdots,N\}$，每一组单独的样本为相互独立的事件。由上述描述可知，单一事件发生的概率可以统一描述为

$$P(y_i\mid\tilde{\boldsymbol{x}}_i)=g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)^{y_i}(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))^{1-y_i} \tag{5.20}$$

则 N 组独立事件同时发生的概率，也被称为似然概率，可以表示为

$$L(\tilde{\boldsymbol{w}})=\prod_{i=1}^{N}g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)^{y_i}(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))^{1-y_i} \tag{5.21}$$

为了方便计算，将上述函数两端同时取对数：

$$l(\tilde{\boldsymbol{w}})=\sum_{i=1}^{N}(y_i\log g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)+(1-y_i)\log(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))) \tag{5.22}$$

最优函数参数 $\tilde{\boldsymbol{w}}$ 就是使上述对数自然函数值最大。而优化方法通常是针对最小值求解，因此对上述对数自然函数取负数。此外，为了去除样本数量的影响，取上述函数均值，可得线性模型分类函数优化的代价函数：

$$J(\tilde{\boldsymbol{w}})=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)+(1-y_i)\log(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))) \tag{5.23}$$

上述函数是关于模型参数的连续凸函数，根据凸优化理论，可采用梯度下降法对模型参数进行优化：

$$\tilde{\boldsymbol{w}}=\tilde{\boldsymbol{w}}-\eta\frac{\partial J(\tilde{\boldsymbol{w}})}{\partial\tilde{\boldsymbol{w}}} \tag{5.24}$$

$$\begin{aligned}\frac{\partial J(\tilde{\boldsymbol{w}})}{\partial\tilde{\boldsymbol{w}}}&=-\frac{1}{N}\sum_{i=1}^{N}y_i\frac{\partial\log(g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))}{\partial\tilde{\boldsymbol{w}}}+(1-y_i)\frac{\partial\log(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))}{\partial\tilde{\boldsymbol{w}}}\\&=-\frac{1}{N}\sum_{i=1}^{N}y_i\frac{1}{g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)}g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))\tilde{\boldsymbol{x}}_i\\&\quad+(1-y_i)\frac{1}{1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)}(-1)g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)(1-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))\tilde{\boldsymbol{x}}_i\\&=-\frac{1}{N}\sum_{i=1}^{N}(y_i-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))\tilde{\boldsymbol{x}}_i=\frac{1}{N}\sum_{i=1}^{N}(y_i-P(y_i\mid\tilde{\boldsymbol{x}}_i))\tilde{\boldsymbol{x}}_i\end{aligned} \tag{5.25}$$

其中，η 为学习率，$P(y_i\mid\tilde{\boldsymbol{x}}_i)=g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i)$ 为线性分类模型的预测输出。

用矩阵的形式简化上述公式：

$$\frac{\partial J(\tilde{\boldsymbol{w}})}{\partial\tilde{\boldsymbol{w}}}=-\frac{1}{N}\sum_{i=1}^{N}(y_i-g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}_i))\tilde{\boldsymbol{x}}_i=-\frac{1}{N}\tilde{\boldsymbol{X}}(\boldsymbol{y}-\boldsymbol{P})^{\mathrm{T}} \tag{5.26}$$

其中，$\boldsymbol{X}=\begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N\\ 1 & 1 & \cdots & 1\end{bmatrix}\in R^{(d+1)\times N}$ 为增广输入矩阵，$\boldsymbol{y}=\left[y_1,y_2,\cdots,y_N\right]\in R^N$ 为实际输出向量，$\boldsymbol{P}=\left[P(y_1\mid\boldsymbol{x}_1),P(y_2\mid\boldsymbol{x}_2),\cdots,P(y_N\mid\boldsymbol{x}_N)\right]$ 为预测概率向量。

通过上述梯度下降方法，不断迭代直至代价函数收敛，则停止迭代，停止时网络参数 $\tilde{\boldsymbol{w}}$ 则为最优网络参数。给定某一未知样本 $\tilde{\boldsymbol{x}}*$，其类别可通过如下方式预测。

$$P(y \mid \boldsymbol{x}*) = g(\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}*) \tag{5.27}$$

当 $P(y \mid \boldsymbol{x}*) > 0.5$ 时，则样本为 1 类；否则为 0 类。

利用 Sigmoid 函数输出作为类别判定的依据，当输出 y>0.5 时，样本被判定为正类；当 y<0.5 时，样本被判定为负类；当 y=0.5 时，样本的类别可以任意判定。

线性分类算法总结如图 5.6 所示。

输入：原始数据 $\{(\boldsymbol{x}_i, y_i), i = 1,2,\cdots,N\}$，$\boldsymbol{x}_i \in R^d, y_i \in \{0,1\}$

算法流程：

（1）构造增广输入矩阵 $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in R^{(d+1)\times N}$ 以及真实输出向量 $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}$。

（2）随机初始化线性分类模型参数 $\tilde{\boldsymbol{w}} \in \mathrm{R}^{d+1}$。

（3）计算线性分类模型预测输出向量 $\boldsymbol{P} = \left[P(y_1 \mid \boldsymbol{x}_1), P(y_2 \mid \boldsymbol{x}_2), \cdots, P(y_N \mid \boldsymbol{x}_N)\right]$。

（4）通过梯度下降法优化线性分类模型参数 $\tilde{\boldsymbol{w}} = \tilde{\boldsymbol{w}} - \eta \dfrac{\partial J(\tilde{\boldsymbol{w}})}{\partial \tilde{\boldsymbol{w}}}$。

（5）重复（3）（4）步，直至代价函数收敛。

（6）通过 $P(y \mid \boldsymbol{x}*) = g(\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}*)$ 预测样本 $\tilde{\boldsymbol{x}}*$ 的类别。

图 5.6　线性分类算法流程

【例 5.2】 二分类任务：已知（0, 0）点为 0 类，（1, 1）点为 1 类，试预测（0.1, 0）点的类别。

【解析】 构建线性分类模型 $y_i = 1/(1+\exp(-(w_1 x_{i1} + w_2 x_{i2} + b)))$，其中，$x_{i1}$ 表示点的第一个坐标，x_{i2} 表示点的第二个坐标，w_1、w_2、b 为模型待学习参数。用向量形式简化模型为 $y_i = 1/(1+\exp(-\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}_i))$，其中，$\tilde{\boldsymbol{w}} = \begin{bmatrix} w_1 & w_2 & b \end{bmatrix}^{\mathrm{T}}$，$\tilde{\boldsymbol{x}}_i = \begin{bmatrix} x_{11} & x_{12} & 1 \end{bmatrix}^{\mathrm{T}}$。线性分类模型训练流程如下。

（1）构造输入增广矩阵 $\boldsymbol{X} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$ 以及真实输出 $\boldsymbol{y} = \begin{bmatrix} 0 & 1 \end{bmatrix}$。

（2）随机初始化分类模型参数 $\tilde{\boldsymbol{w}} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \end{bmatrix}^{\mathrm{T}}$。

（3）根据 $P(y_i \mid \tilde{\boldsymbol{x}}_i) = g(\tilde{\boldsymbol{w}}^{\mathrm{T}} \tilde{\boldsymbol{x}}_i)$ 预测输出概率。

$$\begin{aligned} \boldsymbol{P} &= \left[P(y_1 \mid \boldsymbol{x}_1), P(y_2 \mid \boldsymbol{x}_2)\right] \\ &= \left[1/(1+\exp(-(0.1\times 0 + 0.1\times 0 + 1))), 1/(1+\exp(-(0.1\times 1 + 0.1\times 1 + 1)))\right] \\ &= [0.7311, 0.7685] \end{aligned}$$

（4）通过梯度下降优化模型参数，已知学习率 $\eta = 0.001$。

$$\tilde{\boldsymbol{w}}=\tilde{\boldsymbol{w}}-\eta\frac{\partial J(\tilde{\boldsymbol{w}})}{\partial\tilde{\boldsymbol{w}}}=\tilde{\boldsymbol{w}}+\frac{1}{N}\eta\tilde{\boldsymbol{X}}(\boldsymbol{y}-\boldsymbol{P})^{\mathrm{T}}$$

$$=\begin{bmatrix}0.1 & 0.1 & 0.1\end{bmatrix}^{\mathrm{T}}+\frac{1}{2}0.001\begin{bmatrix}0 & 1\\ 0 & 1\\ 1 & 1\end{bmatrix}\left(\begin{bmatrix}0 & 1\end{bmatrix}-\begin{bmatrix}0.7311 & 0.7685\end{bmatrix}\right)^{\mathrm{T}}$$

$$=[0.1001 \quad 0.1001 \quad 0.0998]^{\mathrm{T}}$$

（5）重复步骤（3）和（4）直至代价函数收敛，此处收敛后 $\tilde{\boldsymbol{w}}=[1.2723 \quad 1.2723 \quad -0.8045]^{\mathrm{T}}$。

（6）通过 $P(y\mid\boldsymbol{x}^*)=g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*)$ 预测样本 $\tilde{\boldsymbol{x}}^*$ 的类别。

$$P(y\mid\boldsymbol{x}^*)=g(\tilde{\boldsymbol{w}}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*)=1/(1+\exp(-[1.2723 \quad 1.2723 \quad -0.8045][0.1 \quad 0 \quad 1]^{\mathrm{T}}))=0.3369$$

所以点（0.1, 0）为第 0 类。

5.4 多分类策略

数据挖掘中最常见的任务是多分类问题，而线性分类模型可以轻松推广至多分类场景。N 个类别 $C_1,C_2,\cdots,C_N$ 分类任务可以拆分为若干个二分类任务。最经典的拆分策略包括“一对一”（One vs. One，OvO）和“一对其余”（One vs. Rest，OvR）。

1.“一对一”拆分策略

给定数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\},y_i\in\{C_1,C_2,\cdots,C_N\}$，一对一策略是将 N 个类别两两配对，用对应类别的数据训练二分类模型。

例如，用 C_1 类别的数据 $D_1=\{(\boldsymbol{x}_1^1,C_1),(\boldsymbol{x}_2^1,C_1),\cdots,(\boldsymbol{x}_m^1,C_1)\}$ 以及 C_2 类别的数据 $D_2=\{(\boldsymbol{x}_1^2,C_2),(\boldsymbol{x}_2^2,C_2),\cdots,(\boldsymbol{x}_m^2,C_2)\}$ 训练分类器 f_{12}。N 类样本共形成 $N(N-1)/2$ 个分类器。给定一个未知样本 $\boldsymbol{x}$，其预测类别可以通过将该样本输入所有二分类器中，统计所有分类器的结果，被预测次数最多的类别就是最终的分类结果，具体如图 5.7 所示。

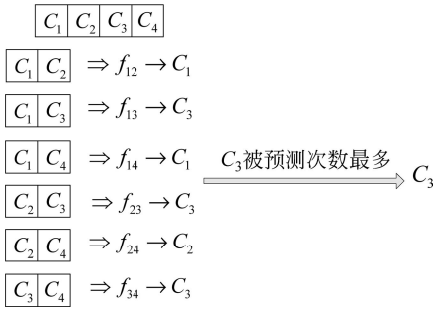

图 5.7 一对一拆分策略

一对一分类策略总结如图 5.8 所示。

【例 5.3】 三分类任务：已知（0, 0）点为 0 类，（1, 1）点为 1 类，（0, 1）点为 2 类。试预测（0.1, 0）点的类别。

【解析】 三分类问题根据一对一拆分策略可以分为 0/1 二分类、0/2 二分类以及 1/2 二

输入：原始数据 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\},y_i\in\{C_1,C_2,\cdots,C_N\}$

算法流程：

（1）从所有类别中选择两组不同的类别的训练数据 $D_i=\{(\boldsymbol{x}_1^i,C_i),(\boldsymbol{x}_2^i,C_i),\cdots,(\boldsymbol{x}_m^i,C_i)\}$ 与 $D_j=\{(\boldsymbol{x}_1^j,C_j),(\boldsymbol{x}_2^j,C_j),\cdots,(\boldsymbol{x}_m^j,C_j)\}$。

（2）根据线性分类模型原理训练分类器 $f_{ij}(\boldsymbol{x})$，具体原理见线性分类器一节，将 C_i 作为正类 1，将 C_j 作为负类 0。

（3）重复步骤（1）（2），直至所有类别组合遍历完毕，得到 $N(N-1)/2$ 组二分类器。

（4）给定未知样本 $\boldsymbol{x}^*$，输入 $N(N-1)/2$ 组二分类器预测其类别，被预测次数最多的类别确定为预测结果。

图 5.8　一对一分类算法流程

分类。

（1）将分类任务分为：（0, 0）与（1, 1）的二分类，（0, 0）与（0, 1）的二分类，（1, 1）与（0, 1）的二分类。

（2）将（0, 0）作为 0 类，（0, 1）作为 1 类，训练线性分类模型 f_{01}，训练结果为

$$f_{01}(\boldsymbol{x}_i)=1/(1+\exp(-(1.2723x_{i1}+1.2723x_{i2}-0.8045)))$$

将（0, 0）作为 0 类，（0, 1）作为 1 类，训练线性分类模型 f_{02}。训练结果为

$$f_{02}(\boldsymbol{x}_i)=1/(1+\exp(-(0.1000x_{i1}+1.6958x_{i2}-0.5454)))$$

将（1, 1）作为 0 类，（0, 1）作为 1 类，训练线性分类模型 f_{12}。训练结果为

$$f_{12}(\boldsymbol{x}_i)=1/(1+\exp(-(-1.7214x_{i1}+0.3481x_{i2}+0.3481)))$$

（3）将（0.1, 0）输入 $f_{01}(\boldsymbol{x}_i)=1/(1+\exp(-(1.2723\times0.1+1.2723\times0-0.8045)))=0.3369$，所以被当前分类器分为 0 类。

将（0.1, 0）输入 $f_{02}(\boldsymbol{x}_i)=1/(1+\exp(-(0.1000\times0.1+1.6958\times0-0.5454)))=0.3693$，所以被当前分类器分为 0 类。

将（0.1, 0）输入 $f_{12}(\boldsymbol{x}_i)=1/(1+\exp(-(-1.7214\times0.1+0.3481\times0+0.3481)))=0.5439$，所以被当前分类器分为 2 类。

在所有类别中，0 类被分为 2 次，2 类被分为 1 次，1 类被分为 0 次，所以（0.1, 0）的最终类别为 0 类。

2.“一对其余”拆分策略

给定数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\},y_i\in\{C_1,C_2,\cdots,C_N\}$，一对其余策略是将一个类的样本作为正例，所有其他类的样本作为反例，一共训练 N 个分类器。在测试时，比较 N 组分类器正例的置信度值，最大值对应的类别为预测类别，具体如图 5.9 所示。

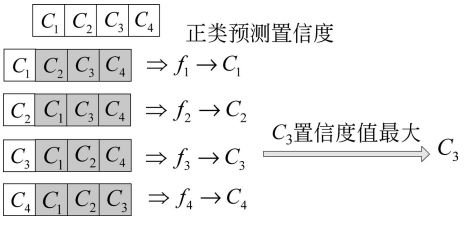

图 5.9　一对其余拆分策略

一对其余分类策略总结如图 5.10 所示。

输入：原始数据 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\},y_i\in\{C_1,C_2,\cdots,C_N\}$

算法流程：

（1）依次选第 i 类作为正例样本 $D_i=\{(\boldsymbol{x}_1^i,C_i),(\boldsymbol{x}_2^i,C_i),\cdots,(\boldsymbol{x}_m^i,C_i)\}$，其余作为负例样本 $D_{j\neq i}=\{(\boldsymbol{x}_1^j,C_j),(\boldsymbol{x}_2^j,C_j),\cdots,(\boldsymbol{x}_m^j,C_j)\},j=1,2,\cdots,N$ 。

（2）根据线性分类模型原理训练分类器 $f_i(\boldsymbol{x})$，具体原理见线性分类器一节，将 C_i 作为正类 1，将其余 $C_{j\neq i}$ 作为负类 0。

（3）重复步骤（1）（2），直至所有类别组合遍历完毕，得到 N 组二分类器。

（4）给定未知样本 $\boldsymbol{x}^*$，输入 N 组二分类器预测其类别，所有正例置信度值最大的对应类别确定为预测结果。

图 5.10 一对其余分类算法流程

【例 5.4】 三分类任务：已知（0, 0）点为 0 类，（1, 1）点为 1 类，（0, 1）点为第 2 类。试预测（0.1, 0）点的类别。

【解析】 三分类问题根据一对其余拆分策略可以分为 0/12 二分类、1/02 二分类以及 2/01 二分类。

（1）将分类任务分为：（0, 0）与{(1, 1), (0, 1)}的二分类，（1, 1）与{(0, 0), (0, 1)}的二分类，（0, 1）与{(0, 0), (1, 1)}的二分类。

（2）将（0, 0）作为 1 类，{(1, 1), (0, 1)}作为 0 类，训练线性分类模型 f_0，训练结果为

$$f_0(\boldsymbol{x}_i)=1/(1+\exp(-(-1.0956x_{i1}-3.0247x_{i2}+1.0301)))$$

将（1, 1）作为 1 类，{(0, 0), (0, 1)}作为 0 类，训练线性分类模型 f_1，训练结果为

$$f_1(\boldsymbol{x}_i)=1/(1+\exp(-(3.1377x_{i1}+0.3768x_{i2}-2.0032)))$$

将（0, 1）作为 1 类，{(1, 1), (0, 1)}作为 0 类，训练线性分类模型 f_2，训练结果为

$$f_2(\boldsymbol{x}_i)=1/(1+\exp(-(-1.8785x_{i1}-0.1182x_{i2}-0.1182)))$$

（3）将（0.1, 0）输入 $f_0(\boldsymbol{x}_i)=1/(1+\exp(-(-1.0956\times0.1-3.0247\times0+1.0301)))=0.7152$，所以为 0 类的置信度值为 0.7152。

将（0.1, 0）输入 $f_1(\boldsymbol{x}_i)=1/(1+\exp(-(3.1377\times0.1+0.3768\times0-2.0032)))=0.1559$，所以为 1 类的置信度值为 0.1559。

将（0.1, 0）输入 $f_2(\boldsymbol{x}_i)=1/(1+\exp(-(-1.8785\times0.1-0.1182\times0-0.1182)))=0.4241$，所以为 2 类的置信度值为 0.4241。

在所有类别中，0 类置信度值为 0.7152，1 类置信度值为 0.1559，2 类置信度值为 0.4241，所以（0.1, 0）的最终类别为 0 类。

实 验

1. 实验目的

掌握线性分类原理，理解线性分类模型学习方法。