第 3 章

线性回归及分类算法

学习目标

- 熟练掌握线性回归算法；
- 理解逻辑回归算法的基础知识；
- 熟练掌握分类算法相关知识及运用。

机器学习中回归和分类是两种常见问题。回归用于预测连续变量，通过线性或非线性模型研究变量间关系，实现建模与预测。分类是根据种类、等级或性质将数据归类。线性回归通过线性关系描述变量间的回归问题，分类通过类似方法处理分类问题。本章首先介绍线性回归算法，然后介绍逻辑回归算法，最后介绍分类算法及其使用方法。

3.1 概述

线性回归和分类算法是机器学习中的重要组成部分。线性回归是一种通过研究一个或多个自变量与因变量之间的关系来预测或解释因变量变化的统计工具。线性回归广泛用于经济预测、工程建模、医疗诊断等领域，例如，通过线性回归模型可以预测股票价格、房产价值以及病患的康复情况。

分类算法通过将数据分为不同类别，实现数据的分类与识别。常见的分类算法包括 K 近邻算法、朴素贝叶斯算法、支持向量机等，这些算法通过度量样本之间的相似性或利用概率论知识对未知样本进行分类。分类算法在图像识别、文本分类、语音识别等领域具有广泛应用，例如，通过分类算法可以实现垃圾邮件过滤、手写数字识别和图像分类。

回归和分类在机器学习中具有不同的应用场景和解决问题的方式，回归用于处理连续变量的预测问题，分类用于将数据分为不同的离散类别，两者通过不同的模型和算法来处理各自的问题，但在实际应用中往往相辅相成，共同解决复杂的数据分析任务。

理解与掌握线性回归和分类算法的基本概念及方法，对于解决实际问题和进行数据分析至关重要。这些算法在各领域中的成功应用展示了强大的实用性和广泛的适应性。通过深入学习和实践，能够有效地应用这些算法进行数据建模和预测，提升在机器学习领域的实

践能力。

3.2　线性回归算法

3.2.1　回归分析

回归分析是一种重要的统计工具，通过研究一个或多个自变量与因变量之间的关系来预测或解释因变量的变化。在初等数学中常利用函数关系表示自变量与因变量之间的关系，然而，在实际的统计关系中观测值并不严格局限于函数图像上，如图 3.1 所示。

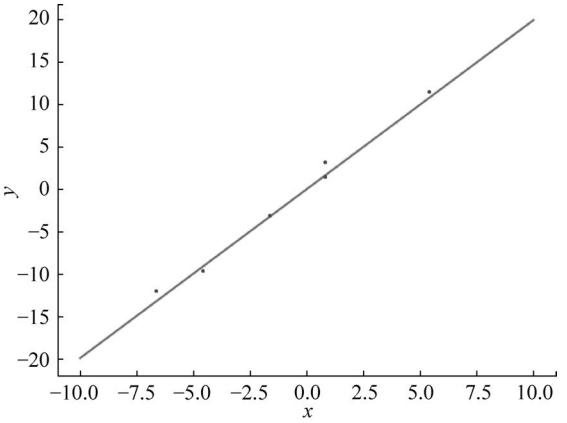

图 3.1　实际统计关系的例子(一元一次函数)

基于这种朴素的函数思想，可以发现统计学中两个基本问题：一是用尽可能系统的形式表现因变量 Y 与自变量 X 的关系；二是表现实际观测值与之前得到的统计关系之间的散布情况。基于以上两个问题，必须假设对于每个 X 存在 Y 的一个概率分布，且该概率分布的均值以一些系统的方式随 X 均匀变化。与上述函数关系不同的是，因为 Y 依赖 X 的概率分布，即使知道了 X，也不一定能确定 Y 的具体值。

回归模型可以有多个自变量，甚至无限个，为了方便，实际工程中尽可能选择有限个自变量或者观察变量。一般情况下不知道回归函数的具体形式，或者即使知道回归函数而函数比较复杂，因此一般采用线性回归函数或者二次回归函数表现近似值。本章只考虑线性模型的情况。

3.2.2　线性模型

设已知观察变量 $\boldsymbol{x}=(x_1;\ x_2;\ x_3;\ \cdots;\ x_d)$，其中 x_i 为 $\boldsymbol{x}$ 的第 i 个属性。线性回归函数可以表示为

$$f(x)=w_1x_1+w_2x_2+\cdots+w_dx_d+b \tag{3.1}$$

用向量表示为

$$f(x)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \tag{3.2}$$

式中：$\boldsymbol{w}=(w_1;\ w_2;\ \cdots;\ w_d)$，可近似看作属性值的权重；$b$ 可以看成随机误差值。$\boldsymbol{w}$ 和 b 都是我们需要“学习”得到的量。

将线性回归函数代入线性回归问题中，假设只考虑一个属性，即 $d=1$。数据集表示为

$D=\{(x_i,y_i)\}_{i=1}^{m}$，这里的 $x_i\in\mathbb{R}$。在学习得到 $f(x)$后，将 $f(x)$与 y 进行比对。回归任务中常用的性能度量均方误差(MSE)定义如下：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(x_i)-y_i)^2 \tag{3.3}$$

代入上述问题，即

$$(\boldsymbol{w}^*,b^*)=\arg\min_{w,b}\sum_{i=1}^{m}(y_i-\boldsymbol{w}x_i-b)^2 \tag{3.4}$$

显然，这里可以看作一个凸优化中的最小二乘法问题。凸函数即二阶导数恒大于 0 的函数，或者表示为

$$f_i(\alpha x+\beta y)\leqslant\alpha f_i(x)+\beta f_i(y) \tag{3.5}$$

凸优化问题的目标函数和约束函数都是凸函数。如果对于没有约束条件($m=0$)的凸优化问题，目标函数又是若干项的平方和，且每一项具有 $\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i$，即可以表示为

$$\min f_0(x)=\|A\boldsymbol{x}-b\|_2^2=\sum_{i=1}^{k}(\boldsymbol{a}_i^{\mathrm{T}}x-b_i)^2 \tag{3.6}$$

回到本问题中的线性回归问题，由于式子比较简单且没有约束条件，目标函数分别对 $\boldsymbol{w}$ 和 b 求偏导，可得

$$\frac{\partial E(\boldsymbol{w},b)}{\partial\boldsymbol{w}}=2\left(\boldsymbol{w}\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)x_i\right) \tag{3.7}$$

$$\frac{\partial E(\boldsymbol{w},b)}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-\boldsymbol{w}x_i)\right) \tag{3.8}$$

令式(3.7)和式(3.8)为 0，可得到 $\boldsymbol{w}$ 和 b 的解析解，即

$$\boldsymbol{w}=\frac{\sum_{i=1}^{m}y_ix_i-\left(\sum_{i=1}^{m}x_i\right)^2}{m\sum_{i=1}^{m}x_i^2-\left(\sum_{i=1}^{m}x_i\right)^2},x=\frac{1}{m}\sum_{i=1}^{m}x_i \tag{3.9}$$

$$b=\frac{1}{m}\sum_{i=1}^{m}(y_i-\boldsymbol{w}x_i) \tag{3.10}$$

以上是多元线性回归的情况。

通过这样的优化和求解过程，可以更好地理解线性回归模型的基础原理，并应用于实际的机器学习任务中。

3.3 逻辑回归算法

在线性回归问题中得到的预测结果是一条连续的直线。然而，有时希望得到一组离散的结果，这些结果可以视为不同的类别。这种问题称为分类问题，是机器学习领域中非常重要的一类问题。

在分类问题中，如果只有两个类别，一般采用 Logistic 函数来解决，称其为二元分类问题。二元分类问题可以抽象成“是”或“否”的问题。在机器学习中，表示“是”的类别称为“正类”，对应的样本称为“正样本”；表示“否”的类别称为“负类”，对应的样本称为“负样本”。

如果超过两个类别，一般采用 Softmax 函数来实现，称为多分类问题。

3.3.1　Logistic 函数

二元分类问题，即将结果拟合成离散的两个结果，由此可联想到信号与系统中的阶跃函数，即

$$u(t)=\begin{cases}1, & t\geqslant 0\\ 0, & t<0\end{cases} \tag{3.11}$$

阶跃函数是不可导函数，不可导函数无法使用优化算法进行优化，因此，需要找到另一个连续且可导的函数来替代阶跃函数。在这种情况下，Logistic 函数应运而生。

Logistic 函数图像如图 3.2 所示。Logistic 图像显示，自变量越小于 0，函数值越接近 0；自变量越大于 0，函数值越接近 1。将之前线性模型得到的结果作为该 Logistic 函数的输入，可以将线性回归问题映射为分类问题。

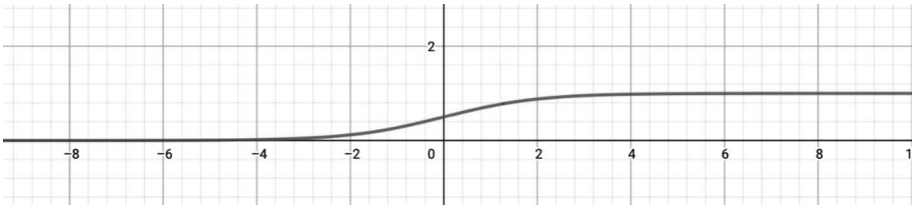

图 3.2　Logistic 函数图像

Logistic 函数的表达式：

$$\text{Logistic}(z)=\frac{1}{1+\mathrm{e}^{-z}} \tag{3.12}$$

将线性模型的公式代入求式(3.2)得到

$$H(x)=\frac{1}{1+\mathrm{e}^{-(\boldsymbol{\omega}^{\mathrm{T}}x_i+b)}} \tag{3.13}$$

这样，原本一条直线经过 Logistic 函数的映射就可以得到离散的输出。

3.3.2　Logistic 回归的损失函数

Logistic 回归的损失函数为

$$L(x)=-[y\log H(x)+(1-y)\log(1-H(x))] \tag{3.14}$$

$H(x)$介于 0～1 之间，满足非负性和归一性。可以把 $H(x)$看作一个概率。事实上，的确有这样一个概率分布——Logistic 分布，其分布和密度函数分别为

$$F(x)=P(X\leqslant x)=\frac{1}{1+\mathrm{e}^{-\frac{x-\mu}{\gamma}}} \tag{3.15}$$

$$f(x)=F'(X\leqslant x)=\frac{\mathrm{e}^{-\frac{x-\mu}{\gamma}}}{\gamma(1+\mathrm{e}^{-\frac{x-\mu}{\gamma}})^2} \tag{3.16}$$

式中：μ 为位置参数；γ 为形状参数，$\gamma>0$。

之前采用的 Logistic 函数实际上是其 $\mu=0,\gamma=1$ 的特殊形式。既然 $H(x)$可以看作概

率，那么可以简单地得到一个损失函数，即

$$L(x)=-[H(x)^{y_i}(1-H(x))^{1-y_i}] \tag{3.17}$$

由于 y_i 只能取 0 或 1，乘积中必有一项为 1。式(3.17)并不是凸函数，无法进行后续各种优化算法。对式(3.17)两边取对数，便得到了最终的损失函数。

通过以上介绍，了解了逻辑回归算法的基础知识，包括 Logistic 函数及其损失函数的推导。在实际应用中，Logistic 回归算法广泛用于解决二元分类问题，如邮件垃圾分类、信用风险评估等。掌握这一算法对于理解和应用其他复杂的机器学习算法具有重要的意义。

3.4 基于距离的分类算法

回归问题和分类问题都属于监督学习问题。监督学习是指从标注数据中学习预测模型的机器学习问题回归问题本身就有一些自变量，也已知一些自变量对应的因变量，这些因变量就是它的标签。同样，在分类问题中也已知一些变量对应的类别，类别也是标签。监督学习中输入与输出所有可能的集合称为输入/输出空间。输入通常用特征向量表示，所有特征向量存在的空间称为特征空间。因此监督学习往往有训练数据和测试数据之分。首先利用训练数据得到一个学习模型，在监督学习领域中假设输入变量 X 与输出变量 Y 满足联合概率分布 $P(X,Y)$，最后得到的模型往往用条件概率分布 $P(Y|X)$ 或者决策函数 $Y=f(x)$ 表示。条件概率分布和决策函数都可以看作输入到输出的一个映射。映射可以有多个，所有可映射组成的集合称为假设空间。

监督学习步骤可以简单概括成学习与预测，分别由学习系统与预测系统完成。学习系统即是利用已有标签的训练数据集得到一个模型。预测系统则对测试样本集由模型 $y_{N+1}=\hat{f}(x_{N+1})$ 或 $y_{N+1}=\arg\ \max_y \hat{P}(y|x_{N+1})$ 给出相应的输出 y_{N+1}。

3.4.1 距离度量

特征空间中两个点的距离是一种度量相似性的重要方法。一般采用 L_p 距离或者闵可夫斯基(Minkowski)距离来定义。

设特征空间 χ 是 n 维向量空间 $\mathbb{R}^n$，$\boldsymbol{x}_i,\boldsymbol{x}_j\in\chi$，其中

$$\boldsymbol{x}_i=\begin{pmatrix} x_i^{(1)} \\ x_i^{(2)} \\ \vdots \\ x_i^{(n)} \end{pmatrix},\quad \boldsymbol{x}_j=\begin{pmatrix} x_j^{(1)} \\ x_j^{(2)} \\ \vdots \\ x_j^{(n)} \end{pmatrix}$$

$\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的 L_p 距离定义为

$$L_p(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{l=1}^{n}\left|\boldsymbol{x}_i^{(l)}-\boldsymbol{x}_j^{(l)}\right|^p\right)^{1/p},\quad p\geqslant 1 \tag{3.18}$$

如果 $p=1$，该距离就是曼哈顿(Manhattan)距离。如果 $p=2$，该距离就是欧几里得(Euclidean)距离。如果 $p=\infty$，该距离就是各个坐标距离的最大值。

3.4.2　分类算法的理解

在分类问题中经常用到的一个数据集是鸢尾花数据集。鸢尾花数据集是一类多重变量分析的数据集，数据集包含 150 个数据集，分为 3 类，每类 50 个数据，每个数据包含 4 个属性。可通过花萼长度、花萼宽度、花瓣长度、花瓣宽度 4 个属性预测鸢尾花卉属于 Setosa、Versicolour、Virginica 三个种类中的哪一类。以花瓣长度为横轴、花瓣宽度为纵轴得到的散点图如图 3.3 所示。

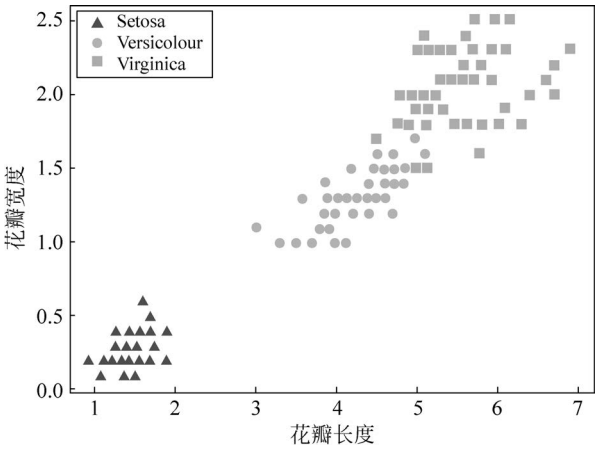

图 3.3　花瓣散点图

如图 3.3 所示，Setosa 鸢尾花基本分布在左下角，而 Versicolour 鸢尾花大约分布于中间，Virginica 鸢尾花大约分布于右上侧。这样看起来，似乎存在某种“天然边界”将各个类别分离，形成某种“同类相吸”的现象。由此想到另一种分类方法，就是找到这个“边界”，直接得到测试数据的类别。

监督学习的输入变量可以分为训练数据和测试数据。训练数据中已经给数据做好了标签，也就是想得到的类别。如果把各种类别看作一个个盒子，那么要把测试数据放入正确的盒子中。放置时要看这个类别有哪些特点，这些特点是从训练数据中得来的。利用训练数据找到与各个类别的相似点，哪个相似点最多，自然就是哪个类别。一般把这种原则称为多数表决。

3.4.3　KNN 算法

KNN 算法是一种采用距离度量分类的经典算法。KNN 全称为 K-Nearest-Neighbor，即选取 K 个最相近的邻居。这 K 个最近点中，哪个类别占比最多，待分类点就属于哪个类别。

需要考虑 K 的选取问题。如果 K 取小了，那么估计误差会增大。估计误差关注测试集，估计误差小了说明对未知数据的预测能力好，且模型本身最接近最佳模型。估计误差的过大会让结果对邻近的点过于敏感，容易产生过拟合现象。如果 K 取大了，那么近似误差会增大，对未知的测试样本将会出现较大偏差的预测，且模型本身不是最接近最佳模型。在现实中 K 一般采用交叉验证等实验方法得到。

选好 K 之后，要面对如何快速搜索近邻点的距离。首先想到的是线性扫描，即逐个计

算输入示例与每个示例的距离，但当训练集很大时，这不太现实，需用到一个数据结构——kd 树。

3.4.4 kd 树

如图 3.4 所示，如果找到距离五角星最近的两个点，那么可以一个个分别计算距离。如果点的数量足够多，就不能这么做。在现实中，凭直觉先找到五角星附近的几个点，再分别测量与五角星的距离，然而计算机并没有这种直觉。这时可以利用二叉树来表征这种“直觉”，把特征空间不断切割放到左右两个子树，找到蕴藏五角星的叶子节点再向上回溯寻找。

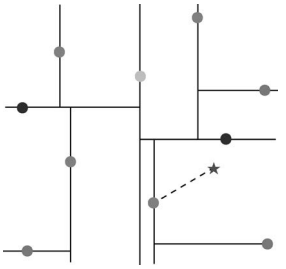

图 3.4 特征空间的分割

对于$\mathbb{R}^2$的情况，需要用一个一维平面来切割成。对于$\mathbb{R}^n$，就需要一个 $n-1$ 维的平面来切割。一般把这个负责切割的平面称为超平面。在数据结构中已经学过二叉排序树利用左小右大的原则依次排序，但面对多维数据显然不能直接这样排序。既然如此，先以一个维度来进行排序。对于二维空间的情况，一般采取奇数层按照 x 轴划分，偶数层按照 y 轴进行划分。其他情况也是按照关键字依次划分。在 kd 树中，一般取所有点该维度坐标的中位数作为切分点(也有利用方差最大的一维作切割点)。这样，左子树保存了对应维度小于切分点的子区域，右子树保存了对应维度大于切分点的子区域。之后，继续这样切割。假设现在到了 j 层，则选择第 l 个维度为坐标轴，这里 $l=(j \mod k)+1$，其中 k 为总维度。于是，这样一直切割，直到没有实例存在时停止。

构造好 kd 树，利用 kd 树解决寻找最近点。与二叉搜索树一样，假设在 kd 树中已发现要找到点，按照维度依次沿左右移动，直至找到一个叶节点，然后标记其已经访问过(可以用一个固定大小为 k 的数据结构存储该节点的信息，该数据结构记为 L)。如果 L 已满，就要将待加入的节点与 L 中距离最长的点相比，如果距离短，就替换掉距离最长的节点。接下来可以回溯到叶子节点的父节点，如果父节点也被访问过，就继续回溯下一个父节点。如果父节点没有访问过，就标记并进行上次同样的操作。然后比较父节点切分线的距离和 L 中最长距离：如果距离长且 L 已满，就无须访问父节点的其他叶子节点；如果距离短或者 L 未满，就访问另一个叶子节点。就这样不断循环，直到访问到根节点为止。

若训练的实例远大于空间维数，则 kd 树的复杂度是 $O(\log N)$。若训练实例与空间维数接近，则搜索效率会无限接近线性模型的搜索效率。因此，kd 树更适合训练实例远大于空间维数的情况。

kd 树算法流程见算法 3.1。

算法 3.1 kd 树算法流程

输入：数据集 D，当前维度 d，总维度数 k

输出：kd-tree 节点

过程：

1. 若数据集 D 为空，则返回空节点
2. 若数据集 D 包含的点数为 1，则返回创建叶子节点(唯一点)

3. 按当前维度 d 排序数据集 D
4. 计算中位数索引＝数据集 D 的大小//2
5. 选择中位数点＝数据集 D[中位数索引]
6. 创建新节点(中位数点)
7. 切分数据集 D 为左子集和右子集
8. 选择下一维度＝(当前维度＋1)%总维度数 k
9. 递归构造左子树和右子树
   左子树＝kd-tree(左子集,下一维度)
   右子树＝kd-tree(右子集,下一维度)
10. 设置节点左子树和节点右子树
11. 返回节点

3.5　基于概率论的朴素贝叶斯算法

3.5.1　概率论知识

条件概率的定义为

$$P(A \mid B)=\frac{P(A \cap B)}{P(B)} \tag{3.19}$$

式中：$P(A|B)$为 B 发生下 A 的条件概率。如果式(3.19)等号两边同乘 $P(B)$($P(B)>0$),可以得到 $P(AB)=P(B)P(A|B)$。因此,如果 A、B 两事件独立,即 $P(A|B)=P(A)$,则可以得到

$$P(A \cap B)=P(A)P(B) \tag{3.20}$$

这就是乘法公式的一个形式。

假设样本空间 Ω 可以分为 $B_1,B_2,\cdots,B_n$,这些分割互不相容。于是,得到

$$A=A\Omega=A\left(\bigcup_{i=1}^{n}B_i\right)=\bigcup_{i=1}^{n}(AB_i) \tag{3.21}$$

概率论中三大公理：非负性公理,即概率大于或等于 0；正则性公理,即 $P(\Omega)=1$；可列可加性公理,即如果 $B_1,B_2,\cdots,B_n,\cdots$互不相容,则有

$$P\left(\bigcup_{i=1}^{\infty}B_i\right)=\sum_{i=1}^{\infty}P(B_i) \tag{3.22}$$

由可列可加性公理能得到概率的有限可加性,再将式(3.19)代入式(3.22),得到

$$P(A)=\sum_{i=1}^{n}P(B_i)P(A \mid B_i) \tag{3.23}$$

式(3.23)为全概率公式。再利用一次条件概率的公式,得到

$$P(B_i \mid A)=\frac{P(AB_i)}{P(A)} \tag{3.24}$$

分子利用乘法公式,分母利用全概率公式,得到

$$P(B_i \mid A)=\frac{P(B_i)P(A \mid B_i)}{\sum_{j=1}^{n}P(B_j)P(A \mid B_j)} \tag{3.25}$$

式(3.25)为贝叶斯公式,也是这个朴素贝叶斯算法理论的核心。一般把 $P(B_i|A)$称为后验概率,$P(A|B_i)$称为似然度,$P(B_i)$称为先验概率。由于一般情况下先验概率已知,贝叶斯公式分母不变,默认后验概率正比于似然度。

3.5.2 朴素贝叶斯算法

把贝叶斯公式代入机器学习分类的场景中。不妨把 A 看作样本的各种特征,用特征向量的形式表示;B 看作各种类别。

我们同时还注意到朴素贝叶斯算法里的“朴素”两个字。那这个算法究竟“朴素”在哪里呢?之前贝叶斯公式中已经得知各个类别互不相容,现在仍假设各个特征相互独立,也就是条件独立性假设。其目的是简化计算。在实际机器学习的过程中有成千上万个特征,不假设它们独立,很难求出解析解。将特征用$\{a^{(1)},a^{(2)},\cdots,a^{(n)}\}$表示,得到

$$P(A \mid B_i)=P(a_1,a_2,\cdots,a_n \mid B_i)=\prod_{i=1}^{n}P(a_i \mid B_i) \tag{3.26}$$

把式(3.26)代入式(3.25),得到

$$P(B_i \mid A)=\frac{P(B_i)\prod_{j=1}^{n}P(a_j \mid B_i)}{\sum_{k=1}^{n}P(B_k)\prod_{j=1}^{n}P(a_j \mid B_k)} \tag{3.27}$$

式(3.27)为朴素贝叶斯公式的基本公式。于是,得到了朴素贝叶斯分类器。

之前贝叶斯公式中分母对任意 B_i 相同,所以可以把分母约去,得到

$$f(x)=\arg\max_{B_i}P(B_i)\prod_{j=1}^{n}P(a_j \mid B_i) \tag{3.28}$$

式中:arg max 表示下面的参数取什么值时函数值最大,这里就是 $P(B_i)\prod_{j=1}^{n}P(a_j \mid B_i)$ 取最大值时,B_i 为多少。

具体估计 B_i 的过程其实就是参数统计的过程,可以采用最大似然估计或者贝叶斯估计的方法来估计。如果含有隐变量的情况,还需要采用 EM 算法来逐步迭代求解,这时就不是解析解,而只是一个数值解。

3.6 案例分析:糖尿病预测问题

3.6.1 糖尿病预测问题

1. 问题描述

糖尿病是一种常见的慢性疾病,对患者的生活质量和健康有着严重影响。通过早期预测和干预,可以有效地减少并发症的发生,提高患者的生活质量。本案例将使用逻辑回归算法来预测个体是否患有糖尿病。使用的数据集包含 768 名患者的详细信息,有 8 个特征变量(怀孕次数、葡萄糖浓度、血压水平、皮褶厚度、胰岛素水平、体质指数(BMI)、糖尿病家族遗传和年龄)和 1 个目标变量(是否患有糖尿病)。目标变量是一个二元分类值,表示患者是否患有糖尿病(1 表示患有糖尿病,0 表示未患糖尿病)。

2. 解题思路

（1）数据预处理：包括处理缺失值、数据标准化等，以确保数据的一致性和可用性。

（2）模型构建：使用逻辑回归算法构建预测模型。逻辑回归适用于二分类问题，通过学习数据中的特征，建立一个能够输出二元结果（患病或不患病）的模型。

（3）模型训练：将预处理后的数据分为训练集和测试集，使用训练集进行模型训练，使模型能够学习数据中的模式和特征。

（4）模型评估：使用测试集对训练好的模型进行评估，通过准确率、混淆矩阵和分类报告等指标来衡量模型的性能。

通过以上步骤可以得到能够预测糖尿病风险的逻辑回归模型，并评估其在实际应用中的效果。

3.6.2　糖尿病预测问题的 Python 语言示例

在进行数据分析和预测时，逻辑回归是一种常用的分类算法，特别适用于二分类问题。在本例中，将使用逻辑回归模型来预测糖尿病的患病风险。数据集包含多种健康指标，通过这些指标来预测患者是否患有糖尿病。以下代码展示了数据读取、预处理、模型训练以及模型评估的完整流程。通过这段代码，可以了解如何使用逻辑回归模型进行分类预测，并评估模型的性能。

```
#导入必要的库
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

#读取数据集
data = pd.read_csv('diabetes.csv')
#检查数据集信息
print(data.info())

#数据预处理
#将特征和目标变量分开
X = data.drop('Outcome', axis = 1)
y = data['Outcome']
#处理缺失值(此数据集没有缺失值,仅作示例)
X.fillna(X.mean(), inplace = True)
#数据标准化
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
#将数据分为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size = 0.2, random_state =
42)
#创建逻辑回归模型
```

```
model = LogisticRegression()
#训练模型
model.fit(X_train, y_train)

#预测
y_pred = model.predict(X_test)
#模型评估
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(f'准确率: {accuracy}')
print(f'混淆矩阵:\n{conf_matrix}')
print(f'分类报告:\n{class_report}')
```

3.7 阅读材料

弗朗西斯·高尔顿(图 3.5)是英国著名的生物学家和统计学家,涉猎科学范围广泛,被称为“维多利亚女王时代最博学的人”,他也是线性回归方法的提出者。为了研究父代与子代身高的关系,高尔顿和学生卡尔·皮尔逊搜集了 1078 对父母及其儿子的身高数据。他发现这些数据的散点图大致呈直线状态,也就是说,总的趋势是父母的身材偏高(矮)时,儿子的身材也偏高(矮)。具体来说,以每对父母的平均身高作为自变量,他们的一个成年儿子的身高作为因变量,父母身高和儿子身高的关系可以拟合成一条直线,即儿子的身高 y 与父母平均身高 x 大致可归结为

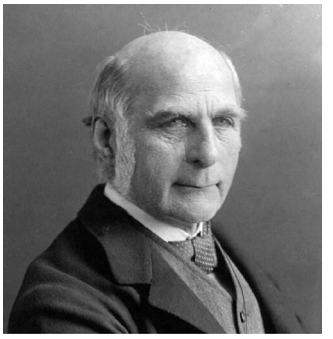

图 3.5 弗朗西斯·高尔顿

$$y=0.8567+0.516x$$

这个拟合关系表明,通过父母身高可以预测子女(成年)的身高。假如父母的平均身高为 1.70m,则预测子女的身高约为 1.73m。(大家可以用自己身边的家庭身高数据作为测试样本验证这个公示的准确度)

通过对这些数据进一步深入分析,高尔顿发现了一个更为有趣的现象:当父母高于平均身高时,他们的儿子身高比父母更高的概率要小于比他们更矮的概率;父母矮于平均身高时,他们的儿子身高比他们更矮的概率要小于比他们更高的概率。结合前文的线性关系可以得出结论:身材较高的父母,他们的孩子也较高,但这些孩子的平均身高并没有他们的父母的平均身高高;身材较矮的父母,他们的孩子也较矮,但这些孩子的平均身高却比他们的父母的平均身高高。这反映了一个规律,即儿子的身高,有向他们父母的平均身高回归的趋势。对于这个结论的一般解释是:大自然具有一种约束力,使人类身高的分布相对稳定而不产生两极分化。

1855 年,高尔顿将上述结果发表在论文《遗传的身高向平均数方向的回归》中,这就是统计学上“回归”定义的第一次出现。虽然“回归”的初始含义与线性关系拟合的一般规则无关(“线性”和“回归”是研究父代与子代身高得出的两个方面的结论),但“线性回归”的术语

因此沿用下来，作为根据一种变量预测另一种变量或多种变量关系的描述方法。

3.8　本章小结

本章介绍了线性回归和分类算法，包括线性回归算法、逻辑回归方法以及基于距离分类的算法和基于概率论的分类方法。回归与分类问题是机器学习的基础和常用工具，理解和掌握回归和分类对后续机器学习内容学习和日后工程需求解决都具有重要的意义。

习题

1. 已知 X、Y 值为(1,1),(2,2),(4,5),(100,99),(200,202)，使用线性回归思想建模，当 $X=15$ 时，预测 Y 是多少？

2. 已知学生每日学习时间与毕业通过情况之间的关系如下：

学生学习时间：[0.50,0.75,1.00,1.25,1.50,1.75,1.75,2.00,2.25,2.50,2.75,3.00,3.25,3.50,4.00,4.25,4.50,4.75,5.00,5.50]，对应顺利毕业(通过答辩毕业为 1，不通过为 0)：[0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1]。

使用逻辑回归预测平均学习 3h 通过答辩顺利毕业的概率。

3. 使用 KNN 算法对手写字体库 MNIST 的数字分类。