第 3 章 线性模型

如无必要，勿增实体。

—— 奥卡姆剃刀

线性模型是机器学习中应用最广泛的模型，在很多情况下，即使线性模型不能直接解决问题，至少也能得到一个不错的逼近。**当你面对一个棘手的问题时，应该首先试试线性方程组。**

> 线性模型如此普遍的深层原因是存在于许多或大部分物理现象中的**平滑性**（“自然不允许跳跃”）。每一个光滑（可微）函数都可以在一个点 x_c 附近得到它的**泰勒展开式逼近**。这个展开式序列的第二项就是线性的，由梯度 $\nabla f(x_c)$ 和位移量的标量积给定，余弦以二阶的速度收敛到零：
>
> $$f(x) = f(x_c) + \nabla f(x_c) \cdot (x - x_c) + O(\|x - x_c\|^2)$$
>
> 因此，在平滑系统中，如果考虑与一个特定点 x_c 相距很近的点，那么线性逼近是一个合理的起点。

3.1 线性回归

输入与输出的线性相关是一个广泛采用的模型，模型中每一项的**权重系数**都为这一项对应的特征的重要性提供了一个直观的解释：某一项权重系数的绝对值越大，对应的属性的影响就越大。

输出与输入参数成线性关系的这一假设可以表示为：

$$y_i = \boldsymbol{w}^\top \boldsymbol{x}_i + \epsilon_i \tag{3-1}$$

其中 $\boldsymbol{w} = (w_1, \dots, w_d)$ 为待确定的**权重向量**，ϵ_i 是误差项。在大多数情况下，假设误差项 ϵ_i 服从高斯分布。即使一个线性模型能正确地解释这些现象，误差仍然会在测量时产生。

现在我们寻找一个权重向量 $\boldsymbol{w}$，使得线性函数

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} \tag{3-2}$$

尽可能逼近实验数据。这一目标可以通过寻找使平方误差和最小的 $\boldsymbol{w}^*$ 来达到（**最小二乘逼近**）：

$$\mathrm{Error}(\boldsymbol{w}) = \sum_{i=1}^{l} (\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 \tag{3-3}$$

在一个零测量误差和完美线性模型的不现实场景里，人们只需要解一系列**线性方程** $\boldsymbol{w}^\top \boldsymbol{x}_i = y_i$，每一个这样的方程对应于一次测量。如果这个系统是良好定义的（d 个非冗余方程对应 d 个未知数），我们就可以通过对系数矩阵求逆来解这些方程。

在实际操作中，让模型误差 Error 达到零是不可能的，另外数据点 $(\boldsymbol{x}_i, y_i)$ 的个数会远远大于参数个数 d。因此，我们需要**通过允许误差的存在来将线性方程组的解进行一般化**，从而一般化矩阵的逆。

使误差函数 Error 最小化很简单：计算梯度，然后要求其为零。我们首先将误差函数 Error 写成矩阵形式，矩阵 $\boldsymbol{X}$ 是以向量 $\boldsymbol{x}_i$ 为行的矩阵，$\boldsymbol{y} = (y_1, \dots, y_l)$，为了求导方便在前面乘以 $\frac{1}{2}$ 记为 $J(\boldsymbol{w})$：

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{l}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 = \frac{1}{2}\Big[(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^\top (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})\Big] \tag{3-4}$$

计算梯度：

$$\nabla_{\boldsymbol{w}} J(\boldsymbol{w}) = \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{y} \tag{3-5}$$

我们令梯度为零可以得到 $\boldsymbol{w}$ 的最优值：

$$\boldsymbol{w}^* = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \tag{3-6}$$

矩阵 $(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$ 称为**广义逆矩阵**，它是对那些非方阵矩阵的矩阵求逆的一种很自然的延伸。如果矩阵是可逆的，并且这个问题可以零误差地求解，那么广义逆矩阵就等于逆矩阵。但是一般情况下，例如训练实例数大于权重系数的个数时，寻求一个最小二乘解能避免无法找到精确解的尴尬，并且能提供一个统计上有意义的“折中”解。

解等式 (3-6) 的方法是“一招制胜”：从实验数据中算出广义逆矩阵，然后相乘得到最优权重系数。在某些情况下，如果训练实例数非常大，**基于迭代方法的梯度下降**就会更受欢迎：从一个初始权重系数开始，然后沿着负梯度的方向小步移动，直到梯度变为零，到达一个稳定点。

具体而言，梯度下降每一次迭代时建议新的点为：

$$\boldsymbol{w}' = \boldsymbol{w} - \epsilon \nabla_{\boldsymbol{w}} J(\boldsymbol{w})$$

其中，ϵ 为学习率，确定步长的大小。实际操作中，我们在梯度 $\nabla_{\boldsymbol{w}} J(\boldsymbol{w})$ 接近零时停止迭代。算法首先将步长 ϵ 和容差 δ 设为小的正数，然后

while $\|\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{y}\|_2 > \delta$ do
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon\Big(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^\top \boldsymbol{y}\Big)$
end while

当训练实例数很大的时候，式 (3-6) 是超定情况的线性系统的解（线性方程多于变量），特别是矩阵 $\boldsymbol{X}^\top \boldsymbol{X}$ 必须是非奇异的。 在很多情况下，即使 $\boldsymbol{X}^\top \boldsymbol{X}$ 是可逆的，也有可能因为训练点集的分布不那么合适而导致不稳定。

> **稳定性**是指样本点中的微扰只会造成结果中的微小改变。

如果没有办法来改变训练样本点的选择，而样本点又没有如愿地分布时，用以保证数值稳定性的数学工具是**岭回归**，它在需要最小化的（最小二乘） 误差函数中加入了一个正则化项：

$$\text{error}(\boldsymbol{w}; \lambda) = \sum_{i=1}^{l}(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i)^2 + \lambda \boldsymbol{w}^\top \boldsymbol{w} \tag{3-7}$$

对 $\boldsymbol{w}$ 进行最小化，得到：

$$\boldsymbol{w}^* = (\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \tag{3-8}$$

在对角线上插入这些小的项使得求逆变得更加具有健壮性。事实上，我们通常也会将权重向量的规模列入考虑范围，以避免出现陡峭的差值平面。可以想象，较大的 λ 值会导致总权重的收缩。

> 这一方法的理论基础是 Tichonov 正则化，它是处理众多不适定问题的最通用的方法。通过同时最小化实验误差和惩罚项，使得模型不仅能很好地拟合，并且还足够简单，避免在估计复杂模型时出现大的变化。

3.2 处理非线性函数关系的技巧

在很多情况下，函数 $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ 是不实用的，因为它存在很多限制。例如它假设 $f(0) = 0$ ，这可以通过加入一个常数项 w_0 从线性模型变成仿射模型来解决：$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^\top \boldsymbol{x}$。 这一常数项可以并入到内积中，只需要重新定义 $\boldsymbol{x} = (1, x_1, \ldots, x_d)$，这样式 (3-2) 对仿射模型依然成立。

然而其他部分还属于最小二乘逼近的简单情况。这可以用一个技巧来解决：仍然用线性模型，只不过将原输入数据 $\boldsymbol{x}$ 进行非线性转换得到**非线性属性**，并在其上应用线性模型。

可以定义这样一个函数集：

$$\phi_1, \ldots, \phi_n : \mathbb{R}^d \to \mathbb{R}^n$$

它从输入空间映射到某个更为复杂的空间，使得我们可以用向量 $\boldsymbol{\varphi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x}))$ 进行线性回归，而不是原始数据 $\boldsymbol{x}$。

> 例如，如果 $d = 2$ 并且输入向量 $\boldsymbol{x} = (x_1, x_2)$，输出的二次相关可以通过如下的**基函数**得到：
>
> $$\begin{gathered}\phi_1(\boldsymbol{x}) = 1, \quad \phi_2(\boldsymbol{x}) = x_1, \quad \phi_3(\boldsymbol{x}) = x_2, \\ \phi_4(\boldsymbol{x}) = x_1 x_2, \quad \phi_5(\boldsymbol{x}) = x_1^2, \quad \phi_6(\boldsymbol{x}) = x_2^2\end{gathered}$$
>
> 这里定义 $\phi_1(\boldsymbol{x})$ 是为了使函数中允许常数项的存在。线性回归方法可以用在经过这些基函数变换后的六维向量上，而不是原来的二维参数向量。

更精确地说，我们寻找如下的一个函数，它是权重向量 $\boldsymbol{w}$ 和属性向量 $\boldsymbol{\varphi}(\boldsymbol{x})$ 的标量积：

$$\hat{f}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{\varphi}(\boldsymbol{x}) \tag{3-9}$$

输出是这些变换后的属性的加权和。其求解方法与线性回归非常类似，也可以通过最小二乘法来计算。令 $\boldsymbol{x}'_i = \boldsymbol{\varphi}(\boldsymbol{x}_i), i = 1, \ldots, l$ 表示训练输入元组 $\boldsymbol{x}_i$ 的变形。如果 $\boldsymbol{X}'$ 是以 $\boldsymbol{x}_i$ 为行向量的矩阵，那么关于最小二乘逼近的最优权重系数可以这样求得：

$$\boldsymbol{w}^* = (\boldsymbol{X}'^\top \boldsymbol{X}')^{-1} \boldsymbol{X}'^\top \boldsymbol{y} \tag{3-10}$$

3.3 最小二乘法的统计学解释

卡方检验函数被广泛应用于估计拟合优度，用以表示真实值与模型估计值之间的偏离程度：

$$\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - f(x_i)}{\sigma_i} \right)^2 \tag{3-11}$$

如果参数 σ_i 都等于 1，那么 χ^2 测量真实值 y_i 和模型估计值 $f(x_i)$ 之间的平方误差和，也就是 3.1 节中描述的 Error。但是在某些情况下，不同数据点的测量过程可能是不同的，人们对于某个测量的误差 σ_i 有一个估计，假设为标准差。卡方的定义就是当计算 χ^2 时，误差必须除以标准差 σ_i 进行归一化，这样得到的结果是与实际误差规模无关的一个数，并且它的含义是经过了标准化的。

回顾最小二乘拟合的过程：

1. 假设大自然和实验程序会产生独立的实验样本 (x_i, y_i)。假设 y_i 的测量值受到了误差的影响，这个误差服从正态（高斯）分布。
2. 如果模型参数 $\boldsymbol{c}$ 是已知的，那么就可以估计我们测量数据的概率，这在统计学中称为数据的**似然率**。
3. 最小二乘法拟合所找到的就是使得我们数据的似然率最大化的参数。最小二乘是一种**最大似然估计**，使得选择的模型和观察到的数据之间的“契合度”最大化。

高斯分布中，对于单个数据点，它的位置与测量值 y_i 的距离在区间 $\mathrm{d}y$ 中的概率正比于：

$$\exp\left(-\frac{1}{2}\left(\frac{y_i - f(x_i, \boldsymbol{c})}{\sigma_i}\right)^2\right)\mathrm{d}y \tag{3-12}$$

由于数据点是独立生成的，整个实验序列（似然性）的概率是单个概率的乘积：

$$\mathrm{d}P \propto \prod_{i=1}^{N}\exp\left(-\frac{1}{2}\left(\frac{y_i - f(x_i, \boldsymbol{c})}{\sigma_i}\right)^2\right)\mathrm{d}y \tag{3-13}$$

由于我们是求关于 $\boldsymbol{c}$ 的最大值，常数因子（像 $(\mathrm{d}y)^N$）是可以略去的。最大化这个似然率等价于最大化它的对数，当常数项被略去的时候，式 (3-13) 的对数就正好是式 (3-11) 中的卡方的定义。因此，最小二乘拟合事实上就是最大似然估计。

3.4 线性模型分类器

假设输出变量是二值的（如 ± 1），此时线性模型可以用作判别器，基本思路是让一个垂直于向量 $\boldsymbol{w}$ 的超平面将这两类隔离开。

> 平面是直线的一般化；同样，当维度大于 3 时，超平面就是平面的一般化。

训练过程的目标是找到最佳的超平面，使得属于不同类别的实例分别位于这个超平面的两边。用数学语言描述就是，要找到最佳的系数向量 $\boldsymbol{w}$ 使得决策程序

$$y = \begin{cases} +1 & \text{如果 } \boldsymbol{w}^\top \boldsymbol{x} \geq 0 \\ -1 & \text{其他情况} \end{cases} \tag{3-14}$$

表现得最好。决定**最优线性分离函数**（几何上的一个超平面）的方法取决于分类标准和误差度量的选择。

如果要进行回归，可以要求第一类的点映射到 $+1$，第二类的点映射到 -1。这是一个比可分离更强的要求，但是让我们能够使用回归方法，像梯度下降法和广义逆矩阵法。此外，最小二乘法不仅可以实现两个类别样本的分类（如果这两类样本是线性可分的），还可以让分类是健壮的，分割超平面离样本都很远。

> 通过强制模型的输出为 $+1$ 或 -1，加上平方误差惩罚项，可以避免得到过多的分割超平面，从而提升模型的稳定性。

如果训练实例线性不可分，要么忍受一些训练误差，要么尝试使用 3.2 节中的技巧，从原始数据中计算出一些非线性的属性，使得变化后的输入能够被分离。