第3章 基本回归算法

CHAPTER 3

本章讨论监督学习的一种类型——回归，首先介绍线性回归，然后推广到基函数回归。通过正则化可以有效地控制模型复杂度与泛化性的关系，本章将详细讨论回归中的正则化技术，介绍求解线性回归的批处理算法和常用的递推算法——随机梯度下降(stochastic gradient descent，SGD)算法。

视频讲解

3.1 线性回归

作为监督学习的一种，回归的数据集是带标注的，即形式为$\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$，这里$\boldsymbol{x}_n$是样本的特征向量，$y_n$是标注。为了处理方便，先假设$y_n$是标量，但可以很直接地推广到向量情况。在回归问题中，标注y_n是连续值，通过学习过程得到一个模型$\hat{y}=f(\boldsymbol{x};\boldsymbol{w})$，该模型的输出是连续量。这里$f(\cdot)$表示一个数学函数，是预先选定的一类函数；$\boldsymbol{w}$表示函数的参数向量，需通过学习来确定；$\boldsymbol{x}$是输入的特征向量，分号“;”表示只有$\boldsymbol{x}$是函数的变量，$\boldsymbol{w}$只是一个参数向量，这样的模型是一个参数化模型。用带标注的训练数据集通过学习过程确定参数向量$\boldsymbol{w}$，则得到回归模型。模型一旦确定，对于一个新输入$\boldsymbol{x}$，带入函数中可计算出回归输出$\hat{y}$。确定模型参数的过程称为学习过程或训练过程，带入新输入计算回归输出的过程称为预测或推断。

本章讨论最基本的一类回归模型，模型表达式$f(\boldsymbol{x};\boldsymbol{w})$是参数$\boldsymbol{w}$的线性函数，称为线性回归模型。在线性回归模型中，$f(\boldsymbol{x};\boldsymbol{w})$与$\boldsymbol{x}$的关系可以是线性的，也可以是非线性的。当$f(\boldsymbol{x};\boldsymbol{w})$与$\boldsymbol{x}$是线性关系时，这是一种最简单的情况，即基本线性回归模型。若通过一种变换函数$\boldsymbol{x}\mapsto\boldsymbol{\phi}(\boldsymbol{x})$将特征向量$\boldsymbol{x}$变换为基函数向量$\boldsymbol{\phi}(\boldsymbol{x})$，并用$\boldsymbol{\phi}(\boldsymbol{x})$替代基本线性回归模型中的$\boldsymbol{x}$，则回归输出为$\boldsymbol{w}$的线性函数、$\boldsymbol{x}$的非线性函数，称为线性基函数回归模型。本节讨论基本线性回归模型。

3.1.1 基本线性回归

设有满足独立同分布条件(I.I.D)的训练数据集

$$\boldsymbol{D}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_N,y_N)\}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^N \tag{3.1.1}$$

用通用符号$\boldsymbol{x}=[x_1,x_2,\cdots,x_K]^{\mathrm{T}}$表示$K$维特征向量(输入向量)，若取数据集中一个指定样本的特征向量，则表示为$\boldsymbol{x}_n=[x_{n1},x_{n2},\cdots,x_{nK}]^{\mathrm{T}}$。为分析方便，假设标注值$y$是标量，后续可以推广到标注为向量的情况。

回归学习的目标是利用这个数据集，训练一个线性回归函数。定义线性回归函数为

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=w_0+\sum_{k=1}^{K}w_k x_k=\sum_{k=0}^{K}w_k x_k=\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.1.2}$$

其中

$$\boldsymbol{w}=[w_0,w_1,w_2,\cdots,w_K]^{\mathrm{T}} \tag{3.1.3}$$

为模型的权系数向量，而

$$\bar{\boldsymbol{x}}=[1,x_1,x_2,\cdots,x_K]^{\mathrm{T}} \tag{3.1.4}$$

是扩充特征向量，即在 $\boldsymbol{x}$ 向量的第一个元素之前，增加了哑元 $x_0=1$，对应系数 w_0 表示线性回归函数的偏置值。图 3.1.1 是线性回归学习的原理性示意图，这是一个最简单情况，即 $\boldsymbol{x}$ 只是一维的标量，图 3.1.1 中每一个点表示数据集中的一个样本，斜线是通过学习得到的回归模型，即相当于已确定了参数的式(3.1.2)。图 3.1.2 是线性回归的计算结构，图中的空心圆仅表示多元素的加法运算。

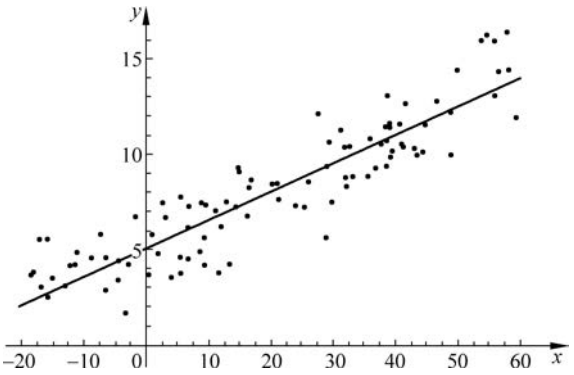


图 3.1.1　线性回归学习的原理性示意图

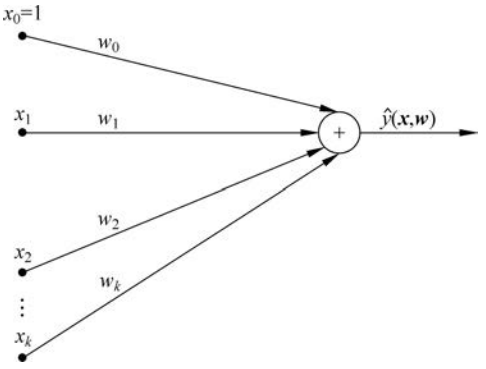


图 3.1.2　线性回归的计算结构

为了从数据集学习模型参数 $\boldsymbol{w}$，用式(3.1.2)逼近训练数据集。对于每个样本$(\boldsymbol{x}_i,y_i)$，将特征向量带入回归函数计算得到的输出 $\hat{y}(\boldsymbol{x}_i,\boldsymbol{w})$是对标注 y_i 的逼近，假设存在逼近误差 ε_i，则有

$$y_i=\hat{y}(\boldsymbol{x}_i,\boldsymbol{w})+\varepsilon_i=\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i+\varepsilon_i \tag{3.1.5}$$

为了得到问题的有效解，通常对误差 ε_i 给出一种概率假设。这里假设 ε_i 服从高斯分布，且均值为 0，方差为 σ_ε^2，则 y_i 的概率密度函数表示为

$$p_y(y_i\mid\boldsymbol{w})=\frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{1}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2\right] \tag{3.1.6}$$

注意，这里把 y_i 看作随机变量，$\boldsymbol{x}_i$ 看作已知量。

如果将所有样本的标注值表示为向量

$$\boldsymbol{y}=[y_1,y_2,y_3,\cdots,y_N]^{\mathrm{T}} \tag{3.1.7}$$

则由样本集的 I.I.D 性，得到 $\boldsymbol{y}$ 的联合概率密度函数为

$$p_y(\boldsymbol{y}\mid\boldsymbol{w})=\prod_{i=1}^{N}p_y(y_i\mid\boldsymbol{w})=\prod_{i=1}^{N}\frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{1}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2\right]$$

$$=\frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{N}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2\right] \tag{3.1.8}$$

由于标注集 $\boldsymbol{y}$ 是已知的,式(3.1.8)随 $\boldsymbol{w}$ 的变化是似然函数,令似然函数最大可求得 $\boldsymbol{w}$ 的解,这就是 $\boldsymbol{w}$ 的最大似然解,为了求解更方便,取对数似然函数为

$$\log p_y(\boldsymbol{y}\mid\boldsymbol{w})=-\frac{N}{2}\log(2\pi\sigma_\varepsilon^2)-\frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2 \tag{3.1.9}$$

若求 $\boldsymbol{w}$ 使得式(3.1.9)的对数似然函数最大,则等价于求如下和式最小:

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{i=1}^{N}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2 \tag{3.1.10}$$

即最大似然等价于 $J(\boldsymbol{w})$最小,这里 $J(\boldsymbol{w})$是训练集上回归函数 $\hat{y}(\boldsymbol{x}_i,\boldsymbol{w})$与标注 y_i 的误差平方之和,式(3.1.10)求和号前的系数 1/2 只是为了后续计算方便。

对于求解回归模型的参数 $\boldsymbol{w}$ 来讲,误差平方和准则(等价于样本的均方误差准则)和高斯假设下的最大似然准则是一致的,故在后续讨论回归问题时,根据方便可使用其中之一。由式(3.1.5),重写误差平方和式(3.1.10)如下:

$$\begin{aligned}J(\boldsymbol{w})&=\frac{1}{2}\sum_{n=1}^{N}\varepsilon_i^2=\frac{1}{2}\sum_{i=1}^{N}[y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w})]^2\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i)^2\\&=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})\\&=\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2\end{aligned} \tag{3.1.11}$$

这里 $\boldsymbol{y}$ 如式(3.1.7)所示,为所有样本的标注向量,$\boldsymbol{X}$ 为数据矩阵,表示为

$$\boldsymbol{X}=\begin{bmatrix}\bar{\boldsymbol{x}}_1^{\mathrm{T}}\\\bar{\boldsymbol{x}}_2^{\mathrm{T}}\\\vdots\\\bar{\boldsymbol{x}}^{\mathrm{T}}{}_N\end{bmatrix}=\begin{bmatrix}1&x_{11}&\cdots&x_{1K}\\1&x_{21}&\cdots&x_{2K}\\\vdots&\vdots&\cdots&\vdots\\1&x_{N1}&\cdots&x_{NK}\end{bmatrix} \tag{3.1.12}$$

为求使式(3.1.11)最小的 $\boldsymbol{w}$,求 $J(\boldsymbol{w})$对 $\boldsymbol{w}$ 的导数,即梯度(标量函数对向量的梯度公式见附录 B),该导数是 $K+1$ 维向量,即

$$\frac{\partial J(\boldsymbol{w})}{\partial\boldsymbol{w}}=\frac{\partial\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})}{\partial\boldsymbol{w}}=-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}+\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}$$

令参数向量的解为 $\boldsymbol{w}_{\mathrm{ML}}$,即当取 $\boldsymbol{w}=\boldsymbol{w}_{\mathrm{ML}}$ 时上式为 $\boldsymbol{0}$,回归系数 $\boldsymbol{w}$ 满足方程

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}_{\mathrm{ML}}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3.1.13}$$

若 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆,有

$$\boldsymbol{w}_{\mathrm{ML}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3.1.14}$$

如果 $\boldsymbol{X}$ 满秩,即 $\boldsymbol{X}$ 的各列线性无关,$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 存在,称

$$\boldsymbol{X}^{+}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}} \tag{3.1.15}$$

为 $\boldsymbol{X}$ 的伪逆矩阵。得到权系数向量后，线性回归函数确定为

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=\boldsymbol{w}_{\mathrm{ML}}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.1.16}$$

将 $\boldsymbol{w}_{\mathrm{ML}}$ 的解(3.1.14)代入式(3.1.10)并除以样本数 N(同时省略系数 1/2)，得到数据集上的均方误差为

$$\begin{aligned}J_{\min}&=\frac{1}{N}J(\boldsymbol{w}_{\mathrm{ML}})=\frac{1}{N}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}\\&=\frac{1}{N}[\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}]^{\mathrm{T}}[\boldsymbol{y}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}]\\&=\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}[\boldsymbol{I}-\boldsymbol{X}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}]^{\mathrm{T}}\boldsymbol{y}\\&=\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}_{\mathrm{ML}})=\frac{1}{N}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{y}-\hat{\boldsymbol{y}})\end{aligned} \tag{3.1.17}$$

式(3.1.17)中，$\hat{\boldsymbol{y}}$ 向量表示由训练集各样本特征向量带入式(3.1.16)得到的对标注集向量 $\boldsymbol{y}$ 的逼近，即

$$\begin{aligned}\hat{\boldsymbol{y}}&=[\hat{y}_1,\hat{y}_2,\hat{y}_3,\cdots,\hat{y}_N]^{\mathrm{T}}\\&=[\boldsymbol{w}_{\mathrm{ML}}^{\mathrm{T}}\bar{\boldsymbol{x}}_1,\quad \boldsymbol{w}_{\mathrm{ML}}^{\mathrm{T}}\bar{\boldsymbol{x}}_2,\quad \cdots,\quad \boldsymbol{w}_{\mathrm{ML}}^{\mathrm{T}}\bar{\boldsymbol{x}}_N]^{\mathrm{T}}=\boldsymbol{X}\boldsymbol{w}_{\mathrm{ML}}\end{aligned} \tag{3.1.18}$$

用式(3.1.14)表示的线性回归权系数向量的解，称为最小二乘(LS)解。若存在一个独立的测试集，也可计算测试集上的均方误差，若测试集误差也满足预定要求，则可确定式(3.1.16)为通过训练过程求得的线性回归函数。当给出一个新的特征向量 $\boldsymbol{x}$，将其代入式(3.1.16)可计算出相应的预测值 $\hat{y}(\boldsymbol{x},\boldsymbol{w})$。

在训练集上可对线性回归的解给出一个几何解释，重写式(3.1.18)如下

$$\hat{\boldsymbol{y}}=\boldsymbol{X}\boldsymbol{w}_{\mathrm{ML}}=[\tilde{\boldsymbol{x}}_0\quad \tilde{\boldsymbol{x}}_1\quad \cdots\quad \tilde{\boldsymbol{x}}_K]\boldsymbol{w}_{\mathrm{ML}}=\sum_{i=0}^{K}\boldsymbol{w}_{\mathrm{ML},i}\tilde{\boldsymbol{x}}_i \tag{3.1.19}$$

这里 $\tilde{\boldsymbol{x}}_i$ 表示 $\boldsymbol{X}$ 的第 i 列(序号以 0 起始)，若由 $\boldsymbol{X}$ 的各列向量为基张成一个向量子空间(称为数据子空间)，则可将 $\hat{\boldsymbol{y}}$ 看作在数据子空间上的投影，投影系数由 $\boldsymbol{w}_{\mathrm{ML}}$ 的各系数确定。

在训练集上，线性回归函数对每个标注值的逼近误差写成误差向量，即

$$\boldsymbol{\varepsilon}=\boldsymbol{y}-\hat{\boldsymbol{y}}=\boldsymbol{y}-\boldsymbol{P}\boldsymbol{y}=\boldsymbol{P}^{\perp}\boldsymbol{y} \tag{3.1.20}$$

这里 $\boldsymbol{P}$ 表示投影矩阵，$\boldsymbol{P}^{\perp}=\boldsymbol{I}-\boldsymbol{P}$ 表示误差投影矩阵。可以证明

$$\boldsymbol{\varepsilon}^{\mathrm{T}}\hat{\boldsymbol{y}}=0 \tag{3.1.21}$$

即两者正交。可见 $\hat{\boldsymbol{y}}$ 是 $\boldsymbol{y}$ 在数据子空间的正交投影，误差向量 $\boldsymbol{\varepsilon}$ 与投影 $\hat{\boldsymbol{y}}$ 正交，因此 $\boldsymbol{\varepsilon}$ 的平方范数最小。图 3.1.3 给出了正交投影的示意图。

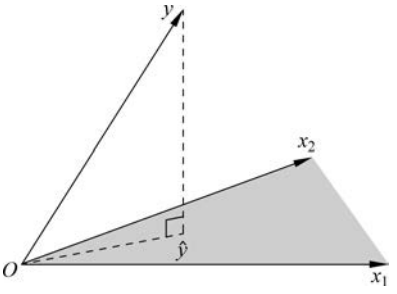


图 3.1.3　正交投影示意

3.1.2　线性回归的递推学习

线性回归函数的学习是现代机器学习中最简单的算法之一。若给出式(3.1.1)表示的数据集，按式(3.1.12)的方式构成数据矩阵 $\boldsymbol{X}$，按式(3.1.7)构成标注向量 $\boldsymbol{y}$，则可通过解析表达式(3.1.14)计算得到线性回归模型的权系数向量 $\boldsymbol{w}_{\mathrm{ML}}$，本质上该权向量是最大似然

解。这种将数据集中所有数据写到数据矩阵,然后通过一次计算得到权系数向量的方法称为批处理。批处理需要集中进行运算,当问题的规模较大时,批处理需要集中处理大量运算,实际中可以考虑更经济的增量计算方法。

为方便,将数据集重写如下:

$$\boldsymbol{D}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_N,y_N)\}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N} \tag{3.1.22}$$

当特征向量 $\boldsymbol{x}_n$ 的维数 K 较大(例如 $K>100$)且数据集的规模较大(例如 $N>10^4$)时,数据矩阵 $\boldsymbol{X}$ 相当大,直接计算式(3.1.14)需要集中处理大批量运算。一种替换方式是一次取出一个样本,构成递推计算,这种递推算法可在线实现。

将式(3.1.11)的误差和重新写为

$$\begin{aligned}J(\boldsymbol{w})&=\frac{1}{2}\sum_{n=0}^{N-1}\varepsilon_i^2=\frac{1}{2}\sum_{i=1}^{N}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2\\&=\frac{1}{2}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i)^2=\sum_{i=1}^{N}J_i(\boldsymbol{w})\end{aligned} \tag{3.1.23}$$

这里 $J_i(\boldsymbol{w})=(y_i-\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i)^2$ 表示单个样本 i 的误差函数,即可以将总体误差函数分解为各样本误差函数之和。

为了导出一种递推算法,使用梯度下降算法,即假设从 $\boldsymbol{w}$ 的一个初始猜测值 $\boldsymbol{w}^{(0)}$ 开始,按照目标函数式(3.1.23)的负梯度方向不断递推,最终收敛到 $\boldsymbol{w}$ 的最优解。设已得到第 k 次递推的权系数向量为 $\boldsymbol{w}^{(k)}$,用该向量计算式(3.1.23)对 $\boldsymbol{w}$ 的梯度,即

$$\begin{aligned}\frac{1}{N}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}&=\frac{1}{N}\sum_{i=1}^{N}\frac{J_i(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}\\&=-\frac{1}{N}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i)\bar{\boldsymbol{x}}_i\end{aligned} \tag{3.1.24}$$

注意,为了避免当样本数 N 太大时,以上和式的梯度太大,式(3.1.24)除以 N 以得到各样本对梯度贡献的均值。根据梯度下降算法,系数向量更新为

$$\begin{aligned}\boldsymbol{w}^{(k+1)}&=\boldsymbol{w}^{(k)}-\eta\frac{1}{N}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}\\&=\boldsymbol{w}^{(k)}+\frac{\eta}{N}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i)\bar{\boldsymbol{x}}_i\end{aligned} \tag{3.1.25}$$

式(3.1.25)是权系数向量的递推算法,称为梯度算法。由于式(3.1.25)使用了所有样本的平均梯度进行运算,并不是逐个样本更新的在线算法。实际上,由式(3.1.24)可知,总的梯度是所有样本点的梯度平均,在每次更新时,若只选择一个样本对梯度的贡献,即只取

$$\frac{J_i(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}=-(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i)\bar{\boldsymbol{x}}_i \tag{3.1.26}$$

作为梯度进行权系数向量的更新,则有

$$\begin{aligned}\boldsymbol{w}^{(k+1)}&=\boldsymbol{w}^{(k)}-\eta\frac{\partial J_i(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}\\&=\boldsymbol{w}^{(k)}+\eta(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i)\bar{\boldsymbol{x}}_i\end{aligned} \tag{3.1.27}$$

由于样本值 y_i、$\boldsymbol{x}_i$ 取自随机分布的采样且具有随机性,因此式(3.1.26)表示的梯度也具有

随机性，称为随机梯度。当样本量充分大时，式(3.1.24)中 N 项求平均的梯度逼近随机梯度的期望值，趋向一个确定性的梯度。因此，式(3.1.25)为梯度算法，式(3.1.27)的递推公式称为随机梯度下降(stochastic gradient descent，SGD)算法。针对线性回归问题的这种SGD算法也称为LMS(least-mean-squares)算法，这是最早使用随机梯度解决机器学习中优化问题的算法。在相当长的时间内，LMS算法在信号处理领域作为自适应滤波的经典算法，应用非常广泛。本质上回归学习和自适应滤波是等价的。

式(3.1.25)和式(3.1.27)中的参数 $\eta>0$ 是控制迭代步长的，称为学习率，用于控制学习过程中的收敛速度。η 过大，递推算法不收敛，η 过小，收敛速度太慢，选择合适的 η 很关键。对于式(3.1.25)的梯度算法和式(3.1.27)的SGD算法，可以证明 $\eta<1/\lambda_{\max}$ 可以保证收敛。这里 $\lambda_{\max}$ 是矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/N$ 的最大特征值，但由于计算 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/N$ 的特征值并不容易(若容易计算 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/N$ 的特征值，就可以直接用式(3.1.14)的批处理，不必用在线算法)，实际上参数 η 的确定大多通过经验或实验来确定，或通过一些对特征值的近似估算确定一个参考值，再通过实验调整。实际学习率随迭代次数变化可记为 η_k，关于随机梯度中学习率 η_k 满足的收敛条件等更一般性的讨论，将在第10章给出。

现代机器学习领域经常使用小批量SGD算法，这种算法是式(3.1.25)和式(3.1.27)的一个折中，即从式(3.1.22)的数据集中随机抽取一小批量样本，重新记为

$$\boldsymbol{D}_{k+1}=\{(\boldsymbol{x}_m,y_m)\}_{m=1}^{N_1} \tag{3.1.28}$$

这里，小批量样本 $\boldsymbol{D}_{k+1}$ 中的下标表示将用于计算 $\boldsymbol{w}^{(k+1)}$，小批量样本的元素 y_m 的下标是在该集合中重新标号的，它随机抽取于大数据集。$N_1\ll N$ 是小样本集的样本数。小批量SGD算法如下

$$\boldsymbol{w}^{(k+1)}=\boldsymbol{w}^{(k)}+\eta\frac{1}{N_1}\sum_{m=1}^{N_1}(y_m-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_m)\bar{\boldsymbol{x}}_m \tag{3.1.29}$$

这里为了使小批量SGD算法与式(3.1.27)的单样本SGD算法的学习率 η 保持同量级，对小批量各样本的梯度进行了平均，即除以 N_1。

注意，在式(3.1.27)的算法中，迭代序号 k 和所用的样本序号 i 并不一致。实际中，在第 k 次迭代时，可随机地从样本集抽取一个样本，即样本一般不是顺序使用的，一些样本可能被重用，小批量梯度算法也是如此。

3.1.3　多输出线性回归

前面介绍回归算法时为了表述简单和理解上的直观性，只给出了输出是标量的情况，即所关注的问题只有一个输出值，实际中很多回归问题可能有多个输出。例如，利用同一组经济数据预测几个同行业的股票指数，前面讨论的标量回归问题可很方便地推广到具有多个输出的情况。

由于具有多个输出，样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}_{n=1}^{N}$ 中的标注 $\boldsymbol{y}_n$ 是一个 L 维向量，这里 L 是回归的输出数目，即 $\boldsymbol{y}_n=[y_{n1},y_{n2},\cdots,y_{nL}]^{\mathrm{T}}$，简单地，可将每个输出写为

$$\hat{y}_i(\boldsymbol{x},\boldsymbol{w}_i)=\sum_{k=0}^{K}w_{ik}x_k=\boldsymbol{w}_i^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.1.30}$$

将各输出的权向量作为权矩阵的一列，即

$$\boldsymbol{W}=[\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_L] \tag{3.1.31}$$

则输出向量记为

$$\hat{\boldsymbol{y}}(\boldsymbol{x},\boldsymbol{W})=[\hat{y}_1(\boldsymbol{x},\boldsymbol{w}_1),\cdots,\hat{y}_L(\boldsymbol{x},\boldsymbol{w}_L)]^{\mathrm{T}}=\boldsymbol{W}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.1.32}$$

为了通过样本集训练得到权系数矩阵 $\boldsymbol{W}$,只需要推广式(3.1.11)针对标量输出的目标函数到向量输出情况,这里不再给出详细的推导过程,只给出相应结果。

对于向量输出情况,数据矩阵 $\boldsymbol{X}$ 仍由式(3.1.12)定义,相应的标注值由向量变为矩阵,故标注矩阵表示为

$$\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_N]^{\mathrm{T}} \tag{3.1.33}$$

假设 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆,得到权系数矩阵的解为

$$\boldsymbol{W}_{\mathrm{ML}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y} \tag{3.1.34}$$

这个解的形式与式(3.1.14)基本一致,只是用标注矩阵替代标注向量。若用 $\tilde{\boldsymbol{y}}_k$ 表示 $\boldsymbol{Y}$ 的第 k 列,则输出的第 k 个分量的权系数向量为

$$\boldsymbol{w}_{k,\mathrm{ML}}=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\tilde{\boldsymbol{y}}_k \tag{3.1.35}$$

在多分量回归中,每个分量的权系数矩阵与标准单分量回归一致,仅由 $\boldsymbol{Y}$ 的一列可求得,这种互相无耦合的解是因为假设了各分量的误差满足独立高斯假设的相应结果。

由于已经得到了最优的权系数矩阵,若给出一个新的特征向量 $\boldsymbol{x}$,则多分量回归的输出为

$$\hat{\boldsymbol{y}}(\boldsymbol{x},\boldsymbol{W}_{\mathrm{ML}})=\boldsymbol{W}_{\mathrm{ML}}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.1.36}$$

3.2 正则化线性回归

在线性回归系数向量的解中,要求 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆,实际上当 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的条件数很大时,解的数值稳定性不好。一个矩阵的条件数为其最大特征值与最小特征值之比,设 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的所有特征值记为 $\{\lambda_0,\lambda_1,\lambda_2,\cdots,\lambda_K\}$,若特征值是按从大到小排列的,则其条件数为 λ_0/λ_K。矩阵 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的行列式为 $|\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}|=\prod_{i=0}^{K}\lambda_i$,由于 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 是对称矩阵,其特征值 $\lambda_i\geqslant 0$。若最小特征值 $\lambda_K=0$ 则矩阵不可逆,若有一个到几个特征值很小,相应条件数很大,矩阵行列式值可能很小,根据矩阵求逆的格莱姆法则,则 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 的逆矩阵中有很多大的值,相应解向量可能范数很大且数值不稳定。

当 $\boldsymbol{X}$ 中的一些不同列互成比例时,$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 不满秩,这时 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 不可逆。当 $\boldsymbol{X}$ 的一些列相互近似成比例时,对应大的条件数,尽管此时严格讲 $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 可逆,但当计算精度受限时数值稳定性不好。从以上的分析可见,$\boldsymbol{X}$ 的不同列分别对应权系数向量的一个分量,当 $\boldsymbol{X}$ 的一些列成比例时,相当于对应的权系数有冗余,可以减少权系数数目,即减少模型的复杂性。当 $\boldsymbol{X}$ 的条件数很大时,相当于模型参数数目超过了必需的数目,而过多的参数其实更多地被用于拟合训练数据集中的噪声,使得泛化性能变差。因此,$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ 条件数很大,对应的是模型的过拟合。解决过拟合的基本方法,一是增加数据集规模,二是删除一些冗余变量及相应权系数,三是采用正则化。这里介绍正则化方法。

如第 1 章所引出的结论,所谓正则化是在用误差平方和表示的目标函数中增加一项约束参数向量自身的量,一种常用的约束量选择为参数向量的范数平方,即 $\|\boldsymbol{w}\|_2^2=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$,因此加了正则化约束的目标函数为

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{n=1}^{N}\varepsilon_i^2+\frac{\lambda}{2}\sum_{i=1}^{K}w_i^2=\frac{1}{2}\sum_{i=1}^{N}(y_i-\hat{y}(\boldsymbol{x}_i,\boldsymbol{w}))^2+\frac{\lambda}{2}\sum_{i=1}^{K}w_i^2$$

$$=\frac{1}{2}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i)^2+\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

$$=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\frac{\lambda}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} \tag{3.2.1}$$

这里λ是一个可选择的参数，用于控制误差项与参数向量范数约束项的作用。为求使式(3.2.1)最小的$\boldsymbol{w}$值，计算

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}=\frac{1}{2}\frac{\partial(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})}{\partial \boldsymbol{w}}+\frac{\lambda}{2}\frac{\partial \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{\partial \boldsymbol{w}}$$

$$=-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}+2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}+\lambda\boldsymbol{w}$$

令$\boldsymbol{w}=\boldsymbol{w}_R$时上式为0，得

$$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})\boldsymbol{w}_R=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3.2.2}$$

求得参数向量的正则化解为

$$\boldsymbol{w}_R=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3.2.3}$$

这里，解$\boldsymbol{w}_R$中用下标R表示正则化解。线性回归的正则化是一般性正则理论的一个特例。Tikhonov正则化理论的泛函由两部分组成：一项是经验代价函数，如式(3.1.11)中的误差平方和是一种经验代价函数，另一项是正则化项，它是约束系统结构的，在参数优化中用于约束参数向量的范数。每一种不同的正则化项代表设计的一种“偏爱”，例如权系数范数平方作为正则化项是一种对小范数的权系数的偏爱，这种正则化称为“权衰减”(weigh decay)。

例3.2.1　若给定一个数据集，$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$的最大特征值为$\lambda_{\max}=1.0$，最小特征值为$\lambda_{\min}=0.01$，条件数$T=\lambda_{\max}/\lambda_{\min}=100$。若正则化参数取$\lambda=0.1$，则$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I}$的最大特征值和最小特征值分别为$\lambda_{\max}+\lambda=1.1$和$\lambda_{\min}+\lambda=0.11$，因此条件数变为$T_R=1.1/0.11=10$，对于线性回归来讲，正则化相当于改善了数据矩阵的条件数。

正如式(3.1.10)的误差平方和目标函数与最大似然等价，式(3.2.1)的正则化目标函数与贝叶斯框架下的MAP参数估计是等价的。若采用贝叶斯MAP估计，需要给出参数向量$\boldsymbol{w}$的先验分布，假设$\boldsymbol{w}$的各分量为均值为0、方差为σ_w^2且互相独立的高斯分布，则先验分布表示为

$$p_w(\boldsymbol{w})=\frac{1}{(2\pi\sigma_w^2)^{\frac{K+1}{2}}}\exp\left[-\frac{1}{2\sigma_w^2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}\right] \tag{3.2.4}$$

根据MAP估计，求$\boldsymbol{w}$使得下式最大：

$$p(\boldsymbol{w}\mid\boldsymbol{y})\propto p(\boldsymbol{y}\mid\boldsymbol{w})p_w(\boldsymbol{w})=$$

$$\frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{N}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})\right]\frac{1}{(2\pi\sigma_w^2)^{\frac{K+1}{2}}}\exp\left[-\frac{1}{2\sigma_w^2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}\right]$$

对上式取对数可知，求上式最大等价于求$\boldsymbol{w}$使得下式最小：

$$J(\boldsymbol{w})=\frac{1}{2\sigma_\varepsilon^2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\frac{1}{2\sigma_w^2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$$

$$=\frac{1}{2\sigma_\varepsilon^2}\left[(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\frac{\sigma_\varepsilon^2}{\sigma_w^2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}\right] \tag{3.2.5}$$

令 $\lambda=\dfrac{\sigma_e^2}{\sigma_\theta^2}$，则式(3.2.5)中方括号内的内容与式(3.2.1)相同，其参数向量 $\boldsymbol{w}$ 的解为式(3.2.3)。因此，可将正则化线性回归中权系数向量的先验分布看作高斯分布下的贝叶斯 MAP 估计。

用式(3.2.1)第 2 行对 $\boldsymbol{w}$ 求导，不难得到相应于式(3.2.3)解的梯度递推算法，这里只给出小批量 SGD 算法如下：

$$\boldsymbol{w}^{(k+1)}=(1-\lambda\eta)\boldsymbol{w}^{(k)}+\eta\frac{1}{N_1}\sum_{m=1}^{N_1}(y_m-\boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_m)\bar{\boldsymbol{x}}_m \tag{3.2.6}$$

当取 $N_1=1$ 时小批量退化成单样本 SGD。与式(3.1.29)比，在 $\boldsymbol{w}^{(k)}$ 前多了一个收缩因子 $(1-\lambda\eta)$，并增加了一个超参数 λ。

这里可以得到一个基本的结论，若一类机器学习算法的目标函数是通过最大似然得到的，则任何一种对权系数向量施加先验分布 $p_w(\boldsymbol{w})$，从而建立在 MAP 意义下的贝叶斯扩展，均可以等价为一类正则化方法。

通过不同的正则化，可得到不同的针对特定偏爱的结果。例如，用 ℓ_1 范数取代式(3.2.1)中的平方范数，则得到 $\boldsymbol{w}$ 具有稀疏特性(sparsity)的解。所谓稀疏特性是指 $\boldsymbol{w}$ 中的一些分量更偏向于取 0 值。这里 $\boldsymbol{w}$ 的 ℓ_1 范数定义为其各分量的绝对值之和，即 $\|\boldsymbol{w}\|_1=\sum_{k=1}^{K}|w_k|$，故具有稀疏解的回归目标函数为

$$J_s(\boldsymbol{w})=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\frac{\lambda}{2}\|\boldsymbol{w}\|_1 \tag{3.2.7}$$

在一些特定情况下，稀疏约束可得到更有意义的结果，关于回归的稀疏学习可进一步参考文献[20]。

本节注释 在正则化过程中，对系数向量作为约束(惩罚)条件时，一般不将偏置 w_0 作为约束分量。在训练前，首先对输入向量 $\boldsymbol{x}_n$ 的每个分量进行归一化和零均值化，w_0 由训练集标注的均值估计，即 $\hat{w}_0=\dfrac{1}{N}\sum_{n=1}^{N}y_n$，故 w_0 不参与式(3.2.1)的优化，对应式(3.2.3)的系数向量不包括偏置。本节对应式(3.1.12)中 $\boldsymbol{X}$ 的定义内第 1 列的全“1”列被删除。

3.3 线性基函数回归

到目前为止，所讨论的均是线性回归，输出是特征向量或其各分量的线性函数，即

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=\sum_{k=0}^{K}w_kx_k=\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}} \tag{3.3.1}$$

其中扩充特征向量 $\bar{\boldsymbol{x}}$ 和权向量 $\boldsymbol{w}$ 的定义如式(3.1.4)和式(3.1.3)所示。为了将回归的输出与特征向量之间的关系扩展到更一般的非线性关系，可以通过定义一组非线性映射函数来实现。非线性映射函数的一般表示如下：

$$\phi_i(\boldsymbol{x}),\quad i=0,1,2,\cdots,M \tag{3.3.2}$$

每个非线性映射函数 ϕ_i 将 K 维向量 $\boldsymbol{x}$ 映射为一个标量值，其按次序排列为一个 $M+1$ 维向量

$$\boldsymbol{\phi}(\boldsymbol{x})=[\phi_0(\boldsymbol{x}),\phi_1(\boldsymbol{x}),\cdots,\phi_M(\boldsymbol{x})]^{\mathrm{T}} \tag{3.3.3}$$

一般地，令 $\phi_0(\boldsymbol{x})=1$ 为哑元。这里 $\boldsymbol{x}\mapsto\boldsymbol{\phi}(\boldsymbol{x})$ 将 K 维向量映射为 M 维向量，称 $\boldsymbol{\phi}(\boldsymbol{x})$ 为特征向量 $\boldsymbol{x}$ 的基函数向量。

定义权系数向量为

$$\boldsymbol{w}=[w_0,w_1,\cdots,w_M]^{\mathrm{T}}$$

可以通过基函数向量定义新的回归模型为

$$\hat{y}(\boldsymbol{\phi},\boldsymbol{w})=\sum_{k=0}^{M}w_k\phi_k(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) \tag{3.3.4}$$

在式(3.3.4)的模型中，输出与特征向量 $\boldsymbol{x}$ 的关系一般是非线性的，具体非线性形式由 $\boldsymbol{\phi}(\boldsymbol{x})$ 的定义决定，但输出与权系数 $\boldsymbol{w}$ 的关系仍然是线性的，因此称这种模型为线性基函数回归模型。这里线性指的是回归输出与权系数是线性关系，与特征向量的非线性形式由基函数确定。

对于一个训练样本集 $\boldsymbol{D}=\{(\boldsymbol{x}_n,y_n)\}_{n=1}^{N}$，取任意样本，由特征向量 $\boldsymbol{x}_n$ 产生一个对应基函数向量 $\boldsymbol{\phi}(\boldsymbol{x}_n)$，得到模型输出 $\hat{y}(\boldsymbol{\phi}_n,\boldsymbol{w})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)$，注意，这里用到了简写符号 $\boldsymbol{\phi}_n=\boldsymbol{\phi}(\boldsymbol{x}_n)$。模型输出与标注的误差为

$$\varepsilon_n=y_n-\hat{y}(\boldsymbol{\phi}_n,\boldsymbol{w})=y_n-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) \tag{3.3.5}$$

与基本线性回归相比，只要用 $\boldsymbol{\phi}(\boldsymbol{x}_n)$ 代替 $\boldsymbol{x}_n$，其他是一致的，因此定义新的基函数数据矩阵为

$$\begin{aligned}\boldsymbol{\Phi}&=\begin{bmatrix}\boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{x}_1)\\ \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{x}_2)\\ \vdots\\ \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{x}_N)\end{bmatrix}\\&=\begin{bmatrix}\phi_0(\boldsymbol{x}_1) & \phi_1(\boldsymbol{x}_1) & \cdots & \phi_M(\boldsymbol{x}_1)\\ \phi_0(\boldsymbol{x}_2) & \phi_1(\boldsymbol{x}_2) & \cdots & \phi_M(\boldsymbol{x}_2)\\ \vdots & \vdots & \cdots & \vdots\\ \phi_0(\boldsymbol{x}_N) & \phi_1(\boldsymbol{x}_N) & \cdots & \phi_M(\boldsymbol{x}_N)\end{bmatrix}\end{aligned} \tag{3.3.6}$$

注意到，与基本线性回归问题相比，这里除了数据矩阵 $\boldsymbol{\Phi}$ 由式(3.3.6)通过基函数映射进行计算外，一旦数据矩阵 $\boldsymbol{\Phi}$ 确定了，由于待求参数向量 $\boldsymbol{w}$ 仍保持线性关系，需求解的问题与基本线性回归是一致的，故线性基函数回归系数向量的解为

$$\boldsymbol{w}_{ML}=(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y} \tag{3.3.7}$$

其中，$\boldsymbol{y}$ 是标注值向量。注意到，与线性回归的不同主要表现在数据矩阵 $\boldsymbol{\Phi}$ 中。对于线性回归，若特征向量 $\boldsymbol{x}_n$ 是 K 维的，数据矩阵 $\boldsymbol{X}$ 是 $N\times(K+1)$ 维矩阵，且矩阵的每个元素直接来自训练集中一个特征向量的分量(包括了哑元)；对于线性基函数回归，数据矩阵 $\boldsymbol{\Phi}$ 是 $N\times(M+1)$ 维矩阵，即数据矩阵的列数为 $M+1$，M 由基函数数目确定。一般来讲，

$M \geqslant K$，基函数将特征向量 $\boldsymbol{x}$ 映射到更高维空间，并且数据矩阵$\boldsymbol{\Phi}$ 的每个元素需要通过相应映射函数计算得到，增加了计算量。一旦计算得到数据矩阵$\boldsymbol{\Phi}$，线性基函数回归的求解问题和线性回归是一致的。

基函数的类型有很多，常用的有多项式基函数、高斯函数、正余弦函数集等。下面看几个例子。

例 3.3.1 讨论一个线性基函数回归的问题。设样本集的特征向量是一个三维向量，即

$$\boldsymbol{x}_n = [x_{n,1}, x_{n,2}, x_{n,3}]^{\mathrm{T}}$$

设基函数向量为多项式形式，具体地，本例最高取二阶项，则

$$\begin{aligned}\boldsymbol{\phi}(\boldsymbol{x}_n) &= [\phi_0(\boldsymbol{x}_n), \phi_1(\boldsymbol{x}_n), \cdots, \phi_9(\boldsymbol{x}_n)]^{\mathrm{T}} \\ &= [1, x_{n,1}, x_{n,2}, x_{n,3}, x_{n,1}^2, x_{n,2}^2, x_{n,3}^2, x_{n,1}x_{n,2}, x_{n,2}x_{n,3}, x_{n,1}x_{n,3}]^{\mathrm{T}}\end{aligned}$$

这里 $M=9$，为了与线性回归区别，将线性基函数回归的权系数向量记为

$$\boldsymbol{w}_{\phi} = [\boldsymbol{w}_{\phi,0}, \boldsymbol{w}_{\phi,1}, \boldsymbol{w}_{\phi,2}, \cdots, \boldsymbol{w}_{\phi,9}]^{\mathrm{T}}$$

基函数回归的输出为

$$\begin{aligned}\hat{y}(\boldsymbol{\phi}_n, \boldsymbol{w}_{\phi}) &= \sum_{k=0}^{9} \boldsymbol{w}_{\phi,k}\phi_k(\boldsymbol{x}_n) \\ &= w_{\phi,0} + w_{\phi,1}x_{n,1} + w_{\phi,2}x_{n,2} + w_{\phi,3}x_{n,3} + w_{\phi,4}x_{n,1}^2 + w_{\phi,5}x_{n,2}^2 + \\ &\quad w_{\phi,6}x_{n,3}^2 + w_{\phi,7}x_{n,1}x_{n,2} + w_{\phi,8}x_{n,2}x_{n,3} + w_{\phi,9}x_{n,1}x_{n,3}\end{aligned}$$

假设数据集规模为 $N=50$，则标注向量为

$$\boldsymbol{y} = [y_1, y_2, \cdots, y_{50}]^{\mathrm{T}}$$

数据矩阵$\boldsymbol{\Phi}$ 为

$$\boldsymbol{\Phi} = \begin{bmatrix} 1, x_{1,1}, x_{1,2}, x_{1,3}, x_{1,1}^2, x_{1,2}^2, x_{1,3}^2, x_{1,1}x_{1,2}, x_{1,2}x_{1,3}, x_{1,1}x_{1,3} \\ 1, x_{2,1}, x_{2,2}, x_{2,3}, x_{2,1}^2, x_{2,2}^2, x_{2,3}^2, x_{2,1}x_{2,2}, x_{2,2}x_{2,3}, x_{2,1}x_{2,3} \\ \vdots \\ 1, x_{50,1}, x_{50,2}, x_{50,3}, x_{50,1}^2, x_{50,2}^2, x_{50,3}^2, x_{50,1}x_{50,2}, x_{50,2}x_{50,3}, x_{50,1}x_{50,3} \end{bmatrix}$$

$\boldsymbol{\Phi}$ 是一个 50×10 的数据矩阵，计算$(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}$ 需要求 10×10 方阵的逆矩阵。

注意到，对此问题若采用基本线性回归，则输出写为

$$\hat{y}(\boldsymbol{x}_n, \boldsymbol{w}) = w_0 + w_1x_{n,1} + w_2x_{n,2} + w_3x_{n,3}$$

数据矩阵 $\boldsymbol{X}$ 是 50×4 的矩阵，则$(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 的计算只需求 4×4 方阵的逆矩阵。另外，也需注意，计算$\boldsymbol{\Phi}$ 需要一定的计算量，尤其当$\boldsymbol{\Phi}$ 中存在复杂非线性函数时，附加计算量可能是相当可观的，而写出 $\boldsymbol{X}$ 不需要附加计算量。

例 3.3.2 正余弦类的基函数和高斯基函数的例子。与例 3.3.1 一样，设特征向量是一个三维向量，即

$$\boldsymbol{x}_n = [x_{n,1}, x_{n,2}, x_{n,3}]^{\mathrm{T}}$$

定义正弦基函数向量的一个分量为

$$\phi_k(\boldsymbol{x}_n) = \sin(i_1\pi x_{n,1})\sin(i_2\pi x_{n,2})\sin(i_3\pi x_{n,3})$$

其中，$0 \leqslant i_1, i_2, i_3 \leqslant L$ 取正整数；L 是预先确定的一个整数或作为超参数通过交叉验证确

定，本例中$\boldsymbol{\phi}(\boldsymbol{x}_n)$是$(L+1)^3$维向量。

也可以定义高斯基函数的一个分量为

$$\phi_k(\boldsymbol{x}_n)=\exp\left(-\frac{\|\boldsymbol{x}_n-\boldsymbol{\mu}_k\|^2}{2\sigma_k^2}\right)$$

作为基函数使用时，高斯函数不需要归一化，每个基函数分量由中心矩$\boldsymbol{\mu}_k$确定，$\boldsymbol{\mu}_k$是预先确定的一组向量，且与特征向量$\boldsymbol{x}$同维度。例如，本例是三维情况，$\boldsymbol{x}$的取值范围限定在三维正方体中，每维平均划分成L份，则三维正方体被划分成L^3个等体积的小正方体，$\boldsymbol{\mu}_k$表示每个小立方体的中心点位置。σ_k^2控制了每个基函数的有效作用范围，一个简单的选择是各个基函数分量的σ_k^2参数共用一个值。

与基本的线性回归算法一样，线性基函数回归也可以通过随机梯度算法实现，同样，只要用$\boldsymbol{\phi}(\boldsymbol{x}_n)$代替$\boldsymbol{x}_n$，可将SGD算法直接用于基函数情况，基本的SGD算法可写为

$$\boldsymbol{w}^{(n+1)}=\boldsymbol{w}^{(n)}+\eta[y_i-\boldsymbol{w}^{(n)\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_i)]\boldsymbol{\phi}(\boldsymbol{x}_i) \tag{3.3.8}$$

其中，i为在权系数的第$n+1$次更新时用到的样本序号。同样，可以将小批量SGD算法直接应用于基函数情况。

可直接将3.2节讨论的正则化技术推广到基函数情况，也可直接将3.1.3节讨论的多输出回归推广到基函数情况，由于这两个推广都是非常直接的，请读者自己完成(正则化公式推广见习题)，此处不再赘述。

例3.3.3　一个数值例子。本例在第1章用于说明概念，本章已经介绍了这个例子所使用的算法，故重新看一下这个例子。假设存在一个输入输出模型，其关系为

$$f(x)=\frac{1}{1+\exp(-5x)}$$

这里x是标量，在区间$x\in[-1,1]$均匀采样产生输入样本集$\{x_n\}_{n=1}^N$，并通过关系式$y_n=f(x_n)+\varepsilon_n$产生标注值y_n，其中$\varepsilon_n\sim N(0,0.15^2)$表示采样噪声。用带噪声的标注数据$\{x_n,y_n\}_{n=1}^N$为$f(x)$建模。作为说明，首先设训练集样本数为$N=10$，$f(x)$和训练样本值如图3.3.1(a)所示。用同样的方法产生100个样本作为测试集。

使用基函数回归，选择多项式基函数向量为

$$\begin{aligned}\boldsymbol{\phi}(x_n)&=[\phi_0(x_n),\phi_1(x_n),\cdots,\phi_M(x_n)]^{\mathrm{T}}\\&=[1,x_n,x_n^2,\cdots,x_n^M]^{\mathrm{T}}\end{aligned}$$

回归模型为

$$\hat{y}(\boldsymbol{\phi}_n,\boldsymbol{w})=\sum_{k=0}^{M}w_k\phi_k(x_n)=\sum_{k=0}^{M}w_kx_n^k$$

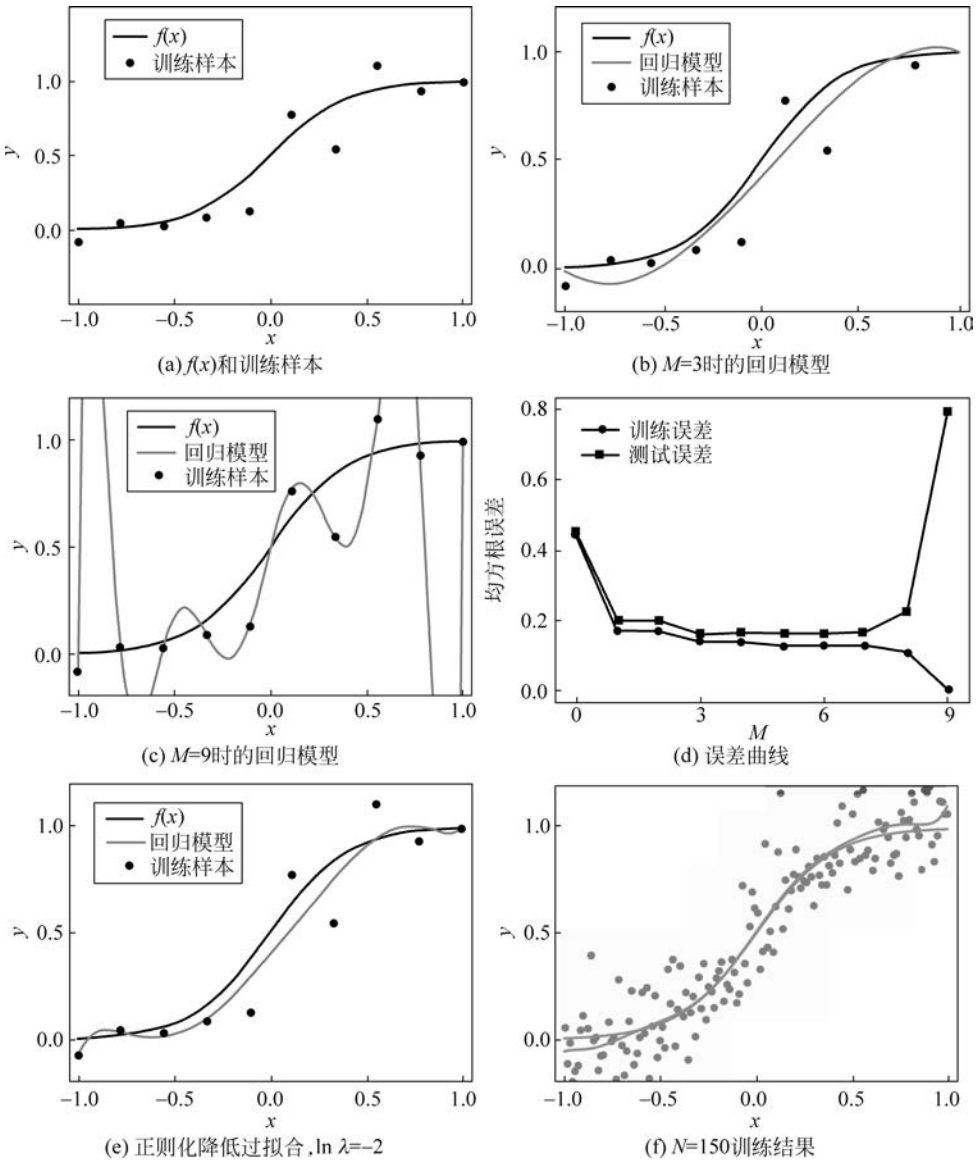
多项式阶数M是一个可选择的值。

首先选择$M=3$，利用式(3.3.7)计算权系数向量，得到的回归模型如图3.3.1(b)所示。注意，为了比较方便，将训练样本和$f(x)$也画于同一图中。然后，选择$M=9$，结果如图3.3.1(c)所示。比较图3.3.1(b)和图3.3.1(c)可见，$M=3$学习到的模型是合适的，尽管存在训练误差，但误差都在较小范围内；$M=9$的模型是过拟合的，尽管其训练误差为0，即学习的模型$\hat{y}(\boldsymbol{\phi},\boldsymbol{w})$通过所有训练样本点，因此在所有样本点处$y_n=\hat{y}(\boldsymbol{\phi}_n,\boldsymbol{w})$，但其泛化性能很差，测试误差很大。原理上讲，$M=9$的模型更复杂，表达能力更强，但在有限训练

集下，为使训练误差更小，将特别关注匹配标注值，而标注值中的噪声将起到很大的引导作用，尽管训练误差为零，但泛化性很差。

图 3.3.1(d)所示为 M 为 1～9 时，训练误差和测试误差的变化关系。误差度量采用的是均方根误差，可以看到，随着模型复杂度升高，训练误差持续下降，但测试误差先下降再升高，表现为 U 形特性，尽管该图是针对这一具体例子得到的，但这个规律具有一般性。对于一个具体问题，在有限的训练集下，当模型复杂度高到一定程度，将出现过拟合，这时模型对训练集的表现优异，但泛化性能变差。对于本例，M 取值为 3～7 比较合适，两个误差均较小。

如果选择了一个较复杂的模型，可以通过正则化降低过拟合。在本例中，取 $M=9$ 时，通过正则化降低过拟合。取正则化参数为 $\ln\lambda=-2$(实际通过交叉验证确定)得到图 3.3.1(e)的



(a) $f(x)$和训练样本　(b) M=3时的回归模型

(c) M=9时的回归模型　(d) 误差曲线

(e) 正则化降低过拟合，ln λ=−2　(f) N=150训练结果

图 3.3.1　例 3.3.3 的数值实验结果

结果，与图 3.3.1(c)比较，消除了过拟合问题。

在一般的机器学习中，增加数据可改善性能是一个基本原则，即具有大的有效数据集，等价地，可用于训练的数据集增大。本例中若取训练集规模为 $N=150$，$M=9$，不使用正则化，则训练得到的模型如图 3.3.1(f)所示，由于训练数据规模明显增加，尽管选择 $M=9$ 的复杂模型，并且没有使用正则化，学习得到的模型优于 $N=10$ 情况下最好的结果。

对以上例子，选择不同的基函数集，误差性能会不同。例如，可选择傅里叶基函数做以上的实验，这个留作习题，有兴趣的读者可自行编程实验。对于许多实际应用，怎样选择合适的基函数集是一个重要、实际的问题。很多情况下，基函数的选择与所处理的问题密切相关，大多是启发式的选择。基函数方法与所谓核函数方法密切相关，其实任何一个基函数向量$\boldsymbol{\phi}(\boldsymbol{x})$都可以对应一个核函数。一个与基函数向量相对应的核函数定义为

$$\kappa(\boldsymbol{x},\boldsymbol{x}_n)=\boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}_n) \tag{3.3.9}$$

因此，核函数是一个具有两个变元的标量函数，具有许多良好的特性。直接利用核函数构造回归模型并利用误差的高斯假设求解该模型的一类方法称为“高斯过程”(在机器学习中，“高斯过程”有这样的专指，不同于随机过程中一般的高斯过程的概念，高斯过程也用于分类)。利用核函数构造支持向量机(SVM)算法则是核函数最重要的应用之一，在 SVM 框架下，既可以得到回归算法，也可以得到分类算法。有关核函数与 SVM 的详细讨论见第 6 章。

3.4　本章小结

本章介绍了机器学习中一类基本的回归算法——线性回归。尽管线性回归比较简单，但仍可有效解决一些复杂度有限的问题。本章首先详细分析了基本线性回归算法，包括其最小二乘解、正则化方法、随机梯度求解和多输出问题。通过基函数映射，讨论了线性基函数回归，由不同的基函数，使得回归输出与输入特征向量之间得到各种非线性关系。

本章对求解线性回归问题给出了较为详细的介绍，实际上许多机器学习的著作都有对线性回归的较完整介绍，例如 Bishop 或 Murphy 的著作(见本书参考文献[6,33])。本章仅提及了稀疏回归学习的基本概念，限于篇幅未能展开讨论，有关稀疏回归学习可进一步参考 Hastie 等的著作(见本书参考文献[20])。

习题

1. 设 x 是一个标量，共有 3 个(x_i,y_i)样本，即$\{(1,0.8),(1.5,0.9),(2,1.2)\}$，用这些数据训练一个简单的回归模型 $\hat{y}=w_0+w_1x$，请计算模型参数。

2. 设 x 是个标量，共有 3 个(x_i,y_i)样本，即$\{(1,0.8),(1.5,0.9),(2,1.2)\}$，用这些数据训练一个线性回归模型 $\hat{y}=w_0+w_1x$，取 $\lambda=0.05$，使用正则化方法计算模型参数。

3. 设线性回归模型的权系数具有以下广义高斯先验分布，即

$$p(\boldsymbol{w};\alpha,\sigma^2)=\prod_{k=0}^{K}\frac{1}{2\beta\Gamma(1/\alpha)}\exp\left(-\frac{|w_k|^{\alpha}}{\beta^{\alpha}}\right)$$

其中，$\beta=\sigma\sqrt{\dfrac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}}$，$\Gamma(\cdot)$是 Gamma 函数，有 $\Gamma(\alpha)=\int_0^{\infty}t^{\alpha-1}\mathrm{e}^{-t}\mathrm{d}t,\alpha>0$。证明：利用贝叶斯 MAP 方法对参数向量 $\boldsymbol{w}$ 的估计等价于正则化目标函数

$$J(\boldsymbol{w})=\frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda\sum_{i=0}^{K}|w_i|^{\alpha}$$

4. 在基函数回归情况下，正则化约束的目标函数为

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{i=1}^{N}(y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_i))^2+\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

证明：线性基函数回归的正则化解为

$$\boldsymbol{w}_{\mathrm{ML}}=(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\lambda\boldsymbol{I})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}$$

5*. 设 $\boldsymbol{x}=[x_1,x_2]^{\mathrm{T}}$ 是二维向量，定义函数 $g(\boldsymbol{x})=\sin(2\pi x_1)\sin(2\pi x_2)$，产生一组训练样本，在 $\boldsymbol{x}\in[0,1]\times[0,1]$范围均匀采样 225 个点，组成输入集$\{\boldsymbol{x}_n\}_{n=1}^{225}$，对每个 $\boldsymbol{x}_n$，通过 $y_n=g(\boldsymbol{x}_n)+\nu_n$ 产生标注值，这里 $\nu_n\sim N(0,0.05)$是独立高斯噪声，产生的训练样本为$\{\boldsymbol{x}_n,y_n\}_{n=1}^{225}$，再独立但用同样模型产生 100 个样本$\{\boldsymbol{x}_n^*,y_n^*\}_{n=1}^{100}$作为测试集。要求训练一个多项式模型 $\hat{y}(\boldsymbol{\phi},\boldsymbol{w})=\sum_{k=0}^{M}\boldsymbol{w}_k\boldsymbol{\phi}_k(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x})$，用于逼近数据存在的规律 $g(\boldsymbol{x})$，基函数的每项取 $\phi_k(\boldsymbol{x})=x_1^{d_1}x_2^{d_2}$，$0\leqslant d_1+d_2\leqslant M$ 的形式，M 是指定的多项式阶数。

(1) 设 $M=3$，用训练样本学习模型参数，用测试样本计算所训练模型与标注值之间的均方误差。

(2) 取 $M=1$ 和 $M=5$，重复(1)的内容，并比较结果。

(3) (选做)自行实验，取更大的 M，使以上方法出现过拟合，计算过拟合时测试集均方误差。选择适当的 λ，通过正则化克服过拟合问题，并给出正则化情况下的均方测试误差。

6*. 重做例 3.3.3 的数值例子，将基函数向量替换为傅里叶基，即

$$\begin{aligned}\boldsymbol{\phi}(x)&=[\phi_0(x),\phi_1(x),\cdots,\phi_M(x)]^{\mathrm{T}}\\&=[1,\sin(\pi x),\cos(\pi x),\sin(2\pi x),\cos(2\pi x),\cdots,\sin(K\pi x),\cos(K\pi x)]^{\mathrm{T}}\end{aligned}$$

其中，$M=2K+1$，取不同的 M 值，重复例 3.3.3 的各项实验内容。