**线性回归**

02

回归问题是一种常见的监督机器学习任务，其在很多领域均有广泛应用。典型应用包括销量预测、库存预测、股票价格预测、天气预测等。

本章我们将讨论线性回归，包括线性回归模型的目标函数（损失函数和正则函数）、目标函数的优化求解、回归模型的性能评价指标、Scikit-Learn中的线性回归模型的应用程序接口（Application Program Interface，API）以及线性回归模型的应用案例。

## **标题2** 2.1 线性回归简介

回归是一种监督学习任务，给定带标签的训练数据 ，其中为样本数目，X为第个样本的输入特征，Y为对应的输出、响应或标签， Y。回归的目标是学习一个从输入X到输出Y的映射，并根据该模型，对新的测试数据，预测其对应的响应：。

若映射是一个线性函数，即

|  |  |
| --- | --- |
|  | （2-1） |

我们称之为线性回归模型，其中为线性回归模型中各特征的回归系数或权重，为截距项。通常我们对输入向量进行增广，即增加1维常数1，得到增广后的输入特征向量，其中为输入特征的维度，这样截距与各特征的回归系数可以统一处理。下面如无特别说明，均假设向量包含常量1，这样模型简写成：

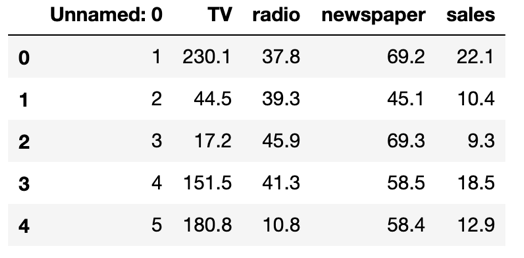
|  |  |
| --- | --- |
|  | （2-2） |

本章我们主要讨论线性回归模型，后续会学习更复杂的非线性回归模型。

例2-1：基于广告费用的产品销量预测

我们分析广告费用对商品销量的影响。数据集包含200个样本，每个样本包括3个输入特征（电视广告费用、广播媒体广告费用和报纸媒体上的广告费用），和响应变量（商品销量）。商品销量为连续值，因此这是一个回归问题。数据集的前5条记录如表2-1所示。

表2-1 广告数据集前5条记录



我们采用线性回归模型来做商品销量预测，即我们假设商品在电视、广播媒体和报纸媒体上的广告费用与商品销量之间的关系为：

，

其中为截距项，分别为3维特征对应的回归系数或权重。

从数据集中随机选取其中80%作为训练数据，训练最小二乘线性回归模型。代码如下：

#导入pandas工具包

import pandas as pd

#读取数据

dpath = "../data/"

df = pd.read\_csv(dpath + "Advertising.csv")

# 从原始数据中分离输入特征x和输出y

y = df['sales']

X = df.drop(['sales', 'Unnamed: 0'], axis = 1)

#将数据分割训练数据与测试数据，随机采样20%的数据构建测试样本，其余作为训练样本

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, random\_state=33, test\_size=0.2)

# 最小二乘线性回归

from sklearn.linear\_model import LinearRegression

# 1使用默认配置初始化学习器实例

lr = LinearRegression()

# 2用训练数据训练模型参数

lr.fit(X\_train, y\_train)

得到的最小二乘线性回归模型为：

。

若采用L1正则的线性回归模型，代码如下：

# L1正则的线性回归Lasso

from sklearn.linear\_model import LassoCV

#1设置超参数搜索范围（默认超参数搜索范围）

#2生成LassoCV实例（用交叉验证确定最佳超参数）

lasso = LassoCV()

#3训练（内含CV）

lasso.fit(X\_train, y\_train)

得到的模型为：

。

我们看到两个模型的总体趋势大致相同，但又不完全相同。这些模型是如何得到？哪个模型更好？本章后续章节将回答这些问题。

## **标题2**  2.2 线性回归模型的目标函数

确定模型并给定训练数据后，根据训练数据来训练模型，得到最佳模型参数。将1.3.1节中监督学习模型的目标函数式（1-1）中用线性回归模型代入，得到线性回归模型的目标函数：

|  |  |
| --- | --- |
|  | （2-3） |

下面我们分别讨论在回归模型中常用的损失函数和线性回归中常用的正则函数。

### **2.2.1** 回归模型的损失函数

#### 1．L2损失

令预测残差 表示模型预测值和真值之间的差异。回归任务常用的损失函数是L2损失：

|  |  |
| --- | --- |
|  | （2-4） |

即残差的平方。此时训练样本上的损失之和为残差平方和 （Residual Sum of Squares，RSS）：

|  |  |
| --- | --- |
|  | （2-5） |

L2损失处处连续，优化求解方便。

从概率角度，最小经验风险等价于高斯白噪声假设下的极大似然估计。令，其中噪声为高斯白噪声，则

我们回顾一下极大似然估计（Maximize Likelihood Estimator，MLE）的基本概念。假设数据是由某个未知模型产生，模型的参数用表示，则定义在该模型下数据的似然为数据出现的概率（各样本为独立同分布的样本）：

|  |  |
| --- | --- |
| 。 | （2-6） |

为了计算方便，我们通常对似然函数取对数运算，得到log似然：

|  |  |
| --- | --- |
| 。 | （2-7） |

极大似然估计为使得似然值最大的模型：

|  |  |
| --- | --- |
| 。 | （2-8） |

在回归中，将代入式（2-7），得到：

|  |  |
| --- | --- |
|  | （2-9） |

式（2-9）的第一项与模型参数无关，只和噪声水平有关；第二项为，因此极大似然等价于最小化（相差常数倍不影响函数取极值的位置）。

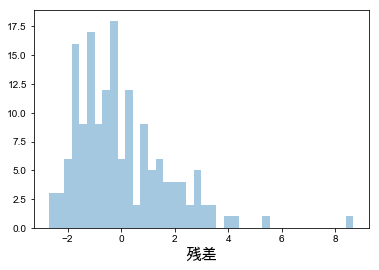
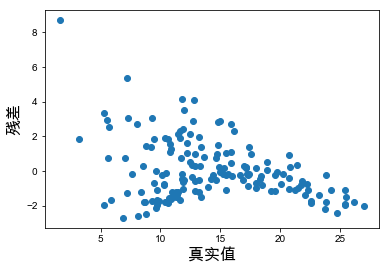
更一般地，损失函数可取负log似然（Negative Log Likelihood，NLL），即，其中为概率分布的参数。此时最小训练集上的损失函数之和等价于极大似然。在回归分析中，若假设噪声为高斯白噪声，去掉和参数无关的常数项和常数倍，得到

|  |  |
| --- | --- |
|  |  |

即此时负log似然等于L2损失。

所以我们可以通过检查残差的直方图是否符合0均值的高斯分布来检查预测模型的性能。例2-1中的最小二乘线性回归在训练样本上的残差分布如图2-1（a）所示。从图中可以看出，残差的分布并不符合0均值的正态分布。

另外我们也可以通过残差与预测值的散点图来检查残差是否与预测值无关。因为我们假设每个样本残差与无关（因此也与模型预测值无关），如果残差是否与预测值有关说明模型不准确。例2-1中的最小二乘线性回归在训练样本上残差与预测值的散点图如图2-1（b）所示。从散点图来看，当真实值较小时，残差的绝对值较大，预测不准确；当真实值较大时，残差大多小于0，预测值偏小。可见该例中最小二乘回归模型并不是特别准确。

（a）残差直方图 （b真实值与残差的散点图

图2-1 回归模型的预测残差的假设检验

#### 2 L1损失

L2损失在回归分析中很常用，但L2损失对离群点（Outliers）敏感。离群点通常远离大部分数据，如果根据大部分数据（去除离群点）得到理想模型，则理想模型的残差的绝对值比较大，L2损失为，其值更大。为了使L2损失小，加入离群点后的数据集训练得到的模型会偏向于离群点而远离理想模型，这是我们不希望看到的。

当数据中存在离群点时，可采用L1损失，即残差的绝对值，

|  |  |
| --- | --- |
| 。 | （2-10） |

L1损失虽然对离群点不敏感，但绝对值函数在原点（）不连续，优化求解相对麻烦。

从概率角度，L1损失可解释为噪声为拉普拉斯分布假设下的极大似然估计。

令，其中噪声，即，。在拉普拉斯噪声分布假设下， log似然为：

|  |  |
| --- | --- |
|  | （2-11） |

式（2-11）的第一项与模型参数无关，第二项等于训练集上L1损失函数之和的倍。因此最小所有训练样本的L1损失之和等价于噪声为拉普拉斯分布假设下的极大似然估计。

**说明**

高斯和拉普拉斯均对回归分析的误差分析做出了突出贡献。高斯和拉普拉斯对正态分布/高斯分布、拉普拉斯分布以及误差分析的贡献可参考统计之都网站上的小故事：正态分布的前世今生。

#### 3 胡伯（Huber）损失

Huber损失综合L2损失和L1损失的优点，定义为：

|  |  |
| --- | --- |
|  | （2-12） |

其中

|  |  |
| --- | --- |
| 。 | （2-13） |

Scikit-Learn建议，参数通过训练得到，目的是当的取值范围发生变化时，的值不变。

从Huber损失的定义可以看出，当残差的绝对值较小时，Huber损失函数为L2范数（0处连续）；当残差的绝对值较大时，取L1损失（对离群点不敏感）。因此Huber损失既处处连续，优化方便，又对离群点不敏感。

### **2.2.2** 线性回归模型的正则函数

#### 1．无正则：最小二乘线性回归（Ordinary Least Square，OLS）

由于线性回归模型比较简单，实际应用中有时正则项为空。若损失函数采用L2损失，得到最小二乘线性回归（此时目标函数中只有残差平方和，“平方”在古时被称为“二乘”）：

|  |  |
| --- | --- |
|  | （2-14） |

由于最小二乘回归模型的目标函数中只有残差平方和，因此从概率的角度，最小二乘回归等价于高斯白噪声假设下的极大似然估计。

#### 2．L2正则：岭回归（Ridge Regression）

一种常用的正则函数是L2正则，即参数的L2范数的平方。带L2正则的线性回归模型被称为岭回归模型。岭回归模型的目标函数为：

|  |  |
| --- | --- |
|  | （2-15） |

其中表示特征的维数，为正则参数，控制正则惩罚的强度。注意正则项中不包含截距项。

从概率角度，岭回归等价于参数先验分布为正态分布的贝叶斯估计。

我们简单回忆一下贝叶斯估计。假设在给定模型（模型参数为）下数据产生过程为，则在该模型下数据产生的似然为

|  |  |
| --- | --- |
| 。 |  |

假设模型参数的先验分布为，根据贝叶斯公式，模型参数的后验估计为

|  |  |
| --- | --- |
|  | （2-16） |

表示看到数据后模型的分布。

在实际应用中，我们通常取概率最大的模型，得到最大后验估计（Maximum a posteriori estimation, MAP）为

|  |  |
| --- | --- |
|  | （2-17） |

同极大似然估计一样，为计算方便，我们对后验分布取对数运算，得到：

|  |  |
| --- | --- |
| 。 | （2-18） |

在回归任务中，假设残差的分布为 ，如式（2-9）所示，线性回归的似然函数为：

若假设参数中每维之间相互独立（的联合分布等于各维特征边缘分布的乘积），且每一维的先验分布为正态分布，即 ，则

|  |  |
| --- | --- |
|  | （2-19） |

的先验分布为均值为0的正态分布，表示我们偏向于较小的系数值，从而得到的模型比较简单，其中 控制先验的强度（越大，越小，先验分布的方差越小，表示每个在0附近的概率更大，要求在0附近的意愿越强烈）。

根据贝叶斯公式，省略与参数无关的项，参数的后验分布为：

|  |  |
| --- | --- |
|  | （2-20） |

参数的最大后验估计等价于最小如下函数（去掉负号，最大变成最小，同时去掉前面的常数倍）：

|  |  |
| --- | --- |
|  | （2-21） |

对比式（2-16）中岭回归的目标函数

|  |  |
| --- | --- |
|  |  |

我们发现，岭回归模型等价于最大后验估计，其中为正则参数，表示先验相对于数据的强度。越大，正则惩罚项的比重越大（越小，先验越强），得到的模型更简单；反之，越小，正则惩罚项的比重越小（越大，先验越弱，数据更重要），得到的模型更复杂。

在正则函数中，不同特征对应的权重（不同对应的）的地位相同，因此在实际应用中，特征的量纲应该相同。如果不同，可以通过数据标准化或最小最大缩放等方式去量纲，在Scikit-Learn中可分别采用类和 类实现。

**说明**

向量范数表征向量空间中向量的大小。维向量的Lp范数范数定义为：，其中为向量的维数。常用的有L0范数（）、L1范数（）和L2范数（）。

L0范数:  向量中非0元素的数目：。注意非0元素的零次方为1，0的0次方为0。

L1范数:  向量各个元素绝对值之和：。

L2范数:   向量各个元素平方和的1/2次方：。L2范数又称欧氏（Euclidean）范数或者斐波那契（Frobenius）范数。

#### 3．L1正则：最小绝对值收缩和选择算子Lasso（Least Absolute Shrinkage and Selection Operator，Lasso）

另一个常用的正则函数为L1正则，即参数的L1范数。带L1正则的线性回归模型被称为Lasso，其目标函数为：

|  |  |
| --- | --- |
|  | （2-22） |

其中为正则参数，控制正则惩罚的强度。当取合适值时，Lasso的结果是稀疏的（的某些元素为0），起到特征选择作用，因此被称为选择算子。

同L2正则类似，从概率角度，Lasso回归模型等价于参数先验分布为拉普拉斯分布的贝叶斯估计。

假设参数的每维相互独立且每一维的先验分布为拉普拉斯分布，即：

|  |  |
| --- | --- |
|  | （2-23） |

的先验分布为均值为0的拉普拉斯分布，表示我们偏向于较小的系数值，从而得到的模型比较简单，其中 控制先验的强度（越大，越小，先验分布的方差越小，表示每个在0附近的概率更大，要求在0附近的意愿越强烈）。

类似岭回归中的推导，我们可以得到Lasso中参数的后验分布为：

|  |  |
| --- | --- |
|  | （2-24） |

对比式（2-22）中Lasso回归模型的目标函数

|  |  |
| --- | --- |
|  |  |

Lasso回归模型等价于贝叶斯最大后验估计，其中为正则参数，表示先验相对于数据的强度。越大，正则惩罚项的比重越大，得到的模型更简单；反之，越小，正则惩罚项的比重越小，得到的模型更复杂。

#### 4．L2正则+L1正则：弹性网络

正则函数亦可同时包含L2正则和L1正则。带L2正则和L1正则L1正则的线性回归模型被称为弹性网络（ElasticNet），其目标函数为：

|  |  |
| --- | --- |
|  | （2-25） |

其中和分别为L2正则参数和L1正则参数。

## **标题2** 2.3 线性回归模型的优化求解

模型的目标函数确定后，我们采用合适的优化方法寻找最佳的模型参数。在线性回归模型中，模型参数包括线性回归系数和正则参数，其中正则参数控制模型的复杂度，我们称之为超参数。本节我们先讨论在给定超参数的情况下，回归系数的优化求解。超参数的选择我们在2.5节讨论。

最佳模型参数为使得目标函数取极小值的参数，即

|  |  |
| --- | --- |
| 。 | （2-26） |

根据优化理论，函数极值点只能在边界点、不可导点或导数为0的点，其中导数为0的点称为函数的临界点。对多元函数，临界点满足所有自变量的偏导均为0，即梯度为向量：

如果海森（Hessian）矩阵

是正定矩阵，临界点的函数值为函数的极小值。

根据模型的特点和问题的复杂程度，可选择不同的优化算法。下面我们以L2损失为例介绍线性回归模型的优化求解过程，其他损失函数读者可自行推导（通常不单独使用L1损失，Huber损失推导类似L2损失）。

### **2.3.1**解析求解法

当训练数据集不大时，最小二乘线性回归和岭回归均可采用解析求解法求解。而Lasso因为有L1正则，没有封闭形式的解析解。

#### 1．最小二乘线性回归解析求解

最小二乘线性回归的目标函数为式（2-14）：。

根据向量求导公式：

|  |  |
| --- | --- |
|  | （2-27） |
|  | （2-28） |

对式（2-14）的求导，得到

|  |  |
| --- | --- |
|  | （2-29） |

令式（2-29）中的，整理后得到：

|  |  |
| --- | --- |
| **。** | （2-30） |

式（2-30）被称为正规方程组（Normal Equations）。

当矩阵满秩时，可逆，式（2-30）两边同乘以，得到

|  |  |
| --- | --- |
| **。** | （2-31） |

**算法**2-1**：**最小二乘线性回归的正规方程组求解****

输入：训练数据，以为行向量组成输入矩阵，个样本构成输出向量

输出：特征的权重向量

过程：

1. 计算的转置和；

2. 计算的逆矩阵；

3. 计算。

根据正规方程组采用式（2-31）求解需要计算矩阵的逆矩阵。但在数值计算上，通过对矩阵进行奇异值分解（singular value decomposition，SVD）求解更稳定。

对的矩阵进行奇异值分解，得到：

|  |  |
| --- | --- |
| ， | （2-32） |

其中是具有正交列的的矩阵，是的准对角矩阵（对角线以外的元素为0），是的正交矩阵。正交矩阵意味着，。同时正交变换还具有保范性质，即对任何向量，。

最小二乘线性回归求使得最小的向量。由于正交变换的保范性质，。记，问题变成最小化。由于为准三角矩阵，上述优化问题变得很简单。令第行对角线的元素值为，得到最佳的，进而得到。需要注意的是，由于奇异值为除数，当矩阵接近不满秩时，某些奇异值的值很小，此时的绝对值会很大，模型不稳定。矩阵接近不满秩意味着特征之间存在共线性，即特征之间有冗余。

**算法**2-2**：**最小二乘线性回归的奇异值分解求解****

输入：训练数据，以为行向量组成输入矩阵，个样本构成输出向量

输出：特征的权重向量

过程：

1. 计算的SVD分解：；
2. 计算；
3. 计算，其中为的第个对角线元素；
4. 计算。

将式（2-31）代入线性回归模型（2-2），得到

|  |  |
| --- | --- |
| ， | （2-33） |

其中矩阵称为帽矩阵或投影矩阵，因为通过乘以矩阵，使得原始响应向量变成预测值。帽矩阵的迹（trace）称为模型的自由度（Degrees of Freedom）。

#### 2．岭回归解析求解

岭回归的目标函数如式（2-15）所示，与最小二乘线性回归只相差一个L2正则项（的二次函数）。对目标函数求导，得到：

|  |  |
| --- | --- |
|  | （2-34） |

令，得到参数估计为：

|  |  |
| --- | --- |
|  | （2-35） |

其中为的单位矩阵。

即使不逆，也是可逆的。所以即使特征之间存在共线性关系，岭回归也能得到稳定的模型。

类似最小二乘线性回归的解析求解，岭回归计算时也可以采用对进行SVD得到：令*，*，则

|  |  |
| --- | --- |
|  | （2-36） |

其中。将（2-36）代入（2-35），得到

|  |  |
| --- | --- |
|  | （2-37） |

式（2-35）中岭回归的解与式（2-31）中最小二乘线性回归的解之间的关系为：

|  |  |
| --- | --- |
|  | *（2-38）* |

*当*时，

所以岭回归的L2正则起到系数模长减少的效果，称为权重衰减。

将式（2-35）代入线性回归模型（2-2），得到

|  |  |
| --- | --- |
|  | （2-39） |

其中矩阵为岭回归的帽矩阵。

令为矩阵的特征值，则帽矩阵的迹：

|  |  |
| --- | --- |
|  | （2-40） |

为岭回归模型的自由度。

采用SVD法求解岭回归系统的一个额外好处是，只需对矩阵进行一次SVD，即可根据式（2-37）得到不同对应的回归系数。

### **2.3.2**梯度下降法

梯度下降（Gradient Descent，GD）法是求解无约束优化问题最常采用的方法之一，亦被称为最速下降法。最小二乘回归和岭回归均可采用梯度下降法求解，Lasso由于目标函数中有L1正则函数不可导，不能采用梯度下降法求解。

假设函数在处可导，对任意小的，函数的一阶泰勒（Taylor）展开近似为：

|  |  |
| --- | --- |
|  | （2-40） |

其中为函数对每个变量的偏导数组成的向量，即梯度，有时也记为。

令式（2-40）中，其中为较小的正数，，则，式子第二项。令，则

|  |  |
| --- | --- |
|  | （2-41） |

即函数值减小。如果进行多次迭代，凸函数将收敛到一个全局最小值，非凸函数收敛到一个局部最小值。注意要足够小，以保证足够小，这样 在 的邻域内，从而可以忽略泰勒展开的高次项。

**算法**2-3**：梯度下降**

1. 初始化：（上标括号中的数字表示迭代次数）；
2. 计算函数在当前位置处的梯度。若，返回为最佳参数；
3. 根据当前学习率，更新位置：

|  |  |
| --- | --- |
|  | （2-42） |

1. ，转第2步。

梯度的计算与目标函数的具体表达式有关。对最小二乘线性回归，目标函数式为式（2-14）：，梯度为式（2-29）：。将上述梯度代入算法2-3，得到最小二乘线性回归的梯度下降更新公式：

|  |  |
| --- | --- |
|  | （2-43） |

类似地，可得到岭回归的梯度下降更新公式：

|  |  |
| --- | --- |
|  | （2-44） |

学习率是梯度下降算法一个很重要的参数。图2-2给出了对例2-1中的数据集，采用不同学习率求解最小二乘回归模型参数的过程。图中给出了目标函数的值随迭代次数的变化。请注意图2-2（a）与图2-2（b）虽然图形形状相似，但达到收敛所用的迭次次数不同。当时，迭代30次达到收敛。当时，只需迭代15次即达到收敛。学习率大，收敛速度快，所需的迭代次数少。如果学习率设置得太小，需要花费较长的时间才能收敛。但学习率过大，会导致参数更新过大，可能会跨越最佳值，使得目标函数值反而增大，发生过冲现象（Overshooting）。如图2-2（c）所示，当时，迭代3次即可收敛，但精度不够，若继续迭代，目标函数的值反而增大。所以如果观测到目标函数随迭代次数增加反而增大的情况，说明学习率设置得过大，此时应调小学习率。通常可以在优化开始阶段采用较大的学习率加快学习速度，但后续应该慢慢衰减。第9章我们将讨论自适应的学习率设置。

在线性回归中，如果特征的取值范围不同，理论上不同的特征需要设置不同的学习率，因为在泰勒展开中，近似只在足够小的情况下才成立。以最小二乘回归为例，第维特征的更新量，其中为矩阵的第列。假设某维特征的绝对值相对其他特征特别大，如果所有特征的学习率相同，为了使绝对值值大的特征对应的参数更新量足够小，这个全局学习率就须设得足够小，从而影响总体的收敛效果。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| （a） | （b） | （c） |

图2-2 梯度下降中学习率的影响

我们在例2-1中的数据集上验证了这一点。为了可视化，我们对数据进行中心化使得截距项为0，同时去除对响应影响不大的特征，这样模型中只包含2维特征：和，分别对应的参数的为和。如果对输入特征进行标准化，变换后每列特征的均值为0，标准差为1，初始值为，学习率为0.1时，最小二乘回归的参数更新轨迹如图2-3（a）所示。我们看到，此时目标函数的等高线基本呈圆形，任意点的负梯度均指向圆心（最小值对应点），轨迹近似直线，收敛速度非常快（30次达到收敛）。

如果对输入特征不做处理，特征的取值范围为，特征的取值范围为。从而目标函数在竖直方向比在水平方向的斜率的绝对值更大，等高线呈椭圆形。因此，固定学习率，梯度下降法中参数在竖直方向比在水平方向移动幅度更大。所以我们需要一个较小的学习率，以避免参数在竖直方向上越过目标函数最优解。然而，较小的学习率会造成参数在水平方向上朝最优解移动变慢。在本例中，若设学习率为，由于学习率过大，算法不能收敛。当学习率设为时，经过89次迭代，算法达到收敛，参数更新轨迹如图2-3（b）所示。由于目标函数的等高线呈椭圆形，除非初始点刚好靠近椭圆的轴，否则负梯度不是指向椭圆圆心并且会趋向于与短轴平行，从而造成在长轴上呈“之”字形反复跳跃，缓慢向最小值逼近。

|  |  |
| --- | --- |
|  |  |
| （a）特征通过标准化进行缩放 | （b）特征不缩放 |

图2-3 梯度下降中特征缩放的影响

上述梯度下降算法被称为“批处理梯度下降”（Batch Gradient Descent，BGD），因为在梯度计算中用到了成批的所有样本。如对最小二乘线性回归，梯度为式（2-29）：。

当样本数目很大时，上述梯度计算很费时。此时我们可采用效率更高的随机梯度下降法（Stochastic Gradient Descent，SGD）。在随机梯度下降中，每次计算梯度只用一个样本：

|  |  |
| --- | --- |
| 。 | （2-45） |

所有样本都用过一次称为一轮（epoch）迭代。

实际应用中，我们通常采用介于随机梯度下降和批处理梯度下降之间的策略：小批量梯度下降（Mini-Batch Gradient Descent）。在小批量梯度下降中，每次不只看一个样本，而是一小批样本。每批次中样本的数目称为批容量（Batch Size）。批大小的选择是在内存效率和内存容量之间寻找最佳平衡。如果批容量过小，梯度震荡大，算法难收敛。增大批批容量，相对处理速度加快，所需内存容量增加，同时需要相应增加迭代的轮数以达到最好结果。相比批处理梯度下降，小批量梯度下降通常收敛更快。

Scikit-Learn建议样本数目超过10000采用随机梯度下降或小批量梯度下降。另外在小批量梯度下降中，每轮须对训练样本重新洗牌，以增加随机性。

### **2.3.3**坐标轴下降法

由于目标函数中有L1正则项，目标函数在原点处不可导，其优化求解推荐采用坐标轴下降法。Lasso还有一种求解方式为最小角度回归（Least Angle Regression, LAR），感兴趣的读者可参考[4]。

#### 1. 坐标轴下降法

顾名思义，坐标轴下降法是沿着坐标轴的方向移动，使得函数值下降。为了找到一个函数的局部极小值，坐标轴下降法在每次迭代中在当前位置沿一个坐标轴方向进行一维搜索，在整个过程中循环使用不同的坐标轴方向。一个周期的一维搜索迭代过程相当于一个梯度迭代。坐标轴下降法在稀疏矩阵上的计算速度非常快，也是Lasso回归优化求解最快的解法。

坐标轴下降法的数学依据主要是如下结论（此处不做证明）：一个可微的凸函数，其中是维向量。如果在某一点，使得在每一个坐标轴上都是最小值，那么就是一个全局的最小值。因此我们的优化目标就是在的个坐标轴上对函数做迭代下降，当所有的坐标轴上的都达到收敛时，函数值最小，此时的即为我们要求的结果。

**算法**2-4**：坐标轴下降法**

初始化为一随机初值，记为；

对第轮迭代，我们依次计算：

|  |  |
| --- | --- |
| ， | （2-46） |

此时 中只有是变量，这个只有一个未知量的优化问题很容易计算。

检查向量 和向量在各个维度上的变化情况，如果在所有维度上变化都足够小，那么即为最终结果，否则转入第2步，继续第轮的迭代。

#### 2．Lasso优化求解之坐标轴下降法

Lasso的目标函数为式（2-22）：，用坐标轴下降法求解时，需要计算目标函数在每一维上的梯度。但正则项中 在 处不可导。为了处理不可导函数，我们扩展导数的表示，定义一个（凸）函数在点处的次梯度（subgradient）或次导数（subderivative）为一个标量，使得

|  |  |
| --- | --- |
| I， | （2-47） |

其中I为包含的某个区间。

定义区间的次梯度（Subgradient）集合为

|  |  |
| --- | --- |
|  | （2-48） |

所有次梯度的区间称为函数在处的次微分（Subdifferential），用表示。

例：绝对值函数 ，其次梯度为

|  |  |
| --- | --- |
|  | （2-49） |

如果函数处处可微，。

同标准的微积分类似，可以证明当且仅当 时，为的局部极值点。

Lasso问题的目标函数为：

|  |  |
| --- | --- |
|  |  |

我们采用坐标轴下降算法求模型参数，分别对第维坐标参数进行分析。其中可微项的梯度：

|  |  |
| --- | --- |
|  | （2-50） |

其中表示向量中除了第维的其他个元素，表示向量中除了第维的其他个元素，表示向量中第维元素。令

|  |  |
| --- | --- |
|  | （2-51） |
|  | （2-52） |

不可微的正则项的次梯度：

|  |  |
| --- | --- |
|  | （2-53） |

综合式（2-50）和（2-53），得到Lasso目标函数的次梯度为

|  |  |
| --- | --- |
|  | （2-54） |

当0属于目标函数的次梯度时，目标函数取极小值，即

|  |  |
| --- | --- |
|  | （2-55） |

**算法2-5：Lasso的坐标轴下降求解：**

1. 预计算；
2. 初始化参数（全0或随机） ；
3. 循环直到收敛：选择变化幅度最大的维度或者轮流更新:

计算；

从上面算法可以看出，当，即当第维特征与去掉该维特征的模型的预测残差弱相关（相关系数的绝对值小于）时，，该维特征可以从模型中去掉，所以L1正则可以起到特征选择作用。当时，所有回归系数，

|  |  |
| --- | --- |
|  | （2-56） |

其中表示所有样本第维的值，表示所有样本的标签的值。

## **标题2** 2.4 模型评估

本节我们讨论回归任务中的常用评价指标。

• 均方误差（Mean Squared Error，MSE）：

|  |  |
| --- | --- |
|  | （2-57） |

MSE计算的是预测残差的平方和，数值越小越好。由于均方误差对大误差的样本有更多的惩罚，因此对离群点敏感。

• （Rooted Mean Squared Error，RMSE）：

|  |  |
| --- | --- |
|  | （2-58） |

RMSE在MSE的基础上开方，这样量纲和相同。

• 平均绝对误差（Mean Absolute Error，MAE）：

|  |  |
| --- | --- |
|  | （2-59） |

MAE为预测残差的绝对值之和，值越小越好。

• 绝对误差的中值（Median Absolute Error，MedianAE）：

|  |  |
| --- | --- |
|  | （2-60） |

其中为计算中值。所以MedianAE为预测残差的中值（中位数），对离群点不敏感。

• 均方对数误差（Mean Squared Logarithmic Error，MSLE）

|  |  |
| --- | --- |
|  | （2-61） |

当的分布范围比较广时（如房屋价格可以从0到非常大的数），如果使用MAE、MSE、RMSE 等误差，这将使得模型更关注于那些较大的样本。而MSLE 关注的是预测误差的比例，使得值较小的样本也同等重要。当数据中存在标签较大的异常值时，MSLE 能够降低这些异常值的影响。

• 均方根对数误差（Root Mean Squared Logarithmic Error，RMSLE）

|  |  |
| --- | --- |
|  | （2-62） |

类似RMSE和MSE的关系，RMSLE是在MSLE的基础上开方。

• 可解释方差分数（Explained Variance Score）：

|  |  |
| --- | --- |
|  | （2-63） |

其中表示方差。可解释方差分数值最大为1，越接近1越好。

• R方分数（R2 score）：

R方分数既考虑预测值与真值之间的差异，也考虑问题本身真值之间的差异：

|  |  |
| --- | --- |
|  | （2-64） |

R方分数的最佳值为1，也可能为负值。如果结果是0，说明模型跟随机猜测差不多。如果结果是1，说明模型无错误。如果结果是0-1之间的数，表示模型的好坏程度。如果结果是负数，说明模型还不如随机猜测。

Scikit-Learn中的模块给出了支持的回归评价指标包括，R方分数也是Scikit-Learn 中回归模型默认的评价指标。

## **标题2** 2.5 线性回归模型的超参数调优

2.3节我们讨论了在给定超参数（正则参数）情况下，线性回归模型参数的优化求解。我们讨论线性模型超参数的调优，即根据任务指定的性能评价指标，确定最优的超参数。

在线性回归模型中，最小二乘线性回归模型没有需要调整的超参数，岭回归模型和Lasso模型的超参数为正则系数。若采用MSE/RMSE为性能评价指标，可采用信息准则直接估计不同超参数对应的模型的性能。常用的信息准则包括赤池信息准则（Akaike information criterion，AIC）和贝叶斯信息准则（Bayes Information criterion，BIC）。若采用其他性能评价指标，只能在验证集上估计模型的性能。如果训练数据集较小，可采用交叉验证得到验证集。

AIC定义为：

|  |  |
| --- | --- |
|  | （2-65） |

其中为模型的在训练集上的预测残差平方和，为超参数对应模型的自由度。岭回归模型的自由度如式（2-40）所示，Lasso模型的自由度为非零系数的数目。

BIC与AIC类似，其定义为：

|  |  |
| --- | --- |
| 。 | （2-66） |

基于信息准则的模型选择非常快，无需验证集，但其依赖于模型自由度的估计，只有在大样本（渐近）、且假设模型是正确的情况下，才能取得好的效果。

对岭回归模型，我们通常采用高效的广义交叉验证（Generalized Cross Validation，GCV）来近似留一交叉验证（Leave One Out Cross Validation, LOOCV），无需循环次即可实现留一交叉验证。

假设模型预测为，为帽矩阵，LOOCV误差估计为：

|  |  |
| --- | --- |
|  | （2-67） |

其中表示去掉第个样本得到的模型。GCV对LOOCV误差的近似为：

|  |  |
| --- | --- |
|  | （2-68） |

其中计算矩阵的迹。

## **标题2** 2.6 案例分析1—广告费用与销量预测

本节我们以例2-1的广告数据集为例，分析线性回归模型的应用。

#### 1．数据分析

给定任务后，我们首先分析数据集的特点，为后续的特征工程、机器学习模型、以及模型优化算法选择提供依据。只有当数据特点与模型假设吻合时才能取得好的效果。

数据分析通常需检查数据集的规模、各特征的分布、特征与特征之间的关系、以及特征与响应之间的关系。另外还要注意数据是有离群点和缺失值。单个特征的分布信息可以直方图或统计量来查看，两个特征之间的关系可以通过散点图查看。对两个连续特征或响应，还可以计算它们之间的相关系数查看这两个特征或者特征与响应之间的线性相关程度。Pandas工具包提供了大量数据分析的方法，可以使用类似SQL的方式非常方便地加载、处理、分析表格形式的数据。搭配Matplotlib和Seaborn可对数据分析结果进行可视化。

图2-4给出了例2-1的广告数据集中各特征与响应的散点图和各特征和响应的相关系数。从图中可以看出，电视广告费用特征（）和广播广告费用特征（）与销量（）线性相关性较强，而报纸广告费用特征（）与销量（）相关性弱。同时3个广告费用特征之间的相关性也较弱。

|  |  |
| --- | --- |
|  |  |
| （a）特征TV与响应sales的散点图 | （b）特征radio与响应sales的散点图 |
|  |  |
| （c）特征newspaper与响应sales的散点图 | （d）特征及响应之间的相关系数 |

图2-4 广告数据集数据分析。

#### 2．特征工程

特征工程是对原始输入数据进行适当的处理，使得数据符合模型的假设及格式要求。在线性回归模型中，特征值直接参与算术运算，因此特征值必须是数值。如果输入特征有离散型特征（如天气类别），需要将离散型特征转换成数值。另外线性回归模型中，我们假设输入特征与响应之间为线性关系，如果输入数据不满足该假设，需要对输入特征或响应做处理，如特征区间化、特征的多项式组合、运算等。

数值型特征常用的特征工程或预处理介绍如下。

（1）去量纲：去量纲的目的是使得特征的量纲或取值范围相同，这样各个特征在模型中的地位不受量纲的影响，可采用Scikit-Learn中的或实现。模型中的正则项和梯度下降优化算法均要求特征的量纲相同或取值范围相同。

（2）分区间量化：线性回归模型假设输入特征与响应之间存在线性关系。如果输入特征与响应不满足线性关系时（如年龄与商品销量之间的关系），我们可以将特征量化成多个区间，变成离散型特征，然后再通过独热编码变成多个新的数值型特征。

（3）log变换：当特征的取值范围很大（右斜分布）时，可以考虑对特征进行变换。因为运算拉伸较小幅度的特征值，压缩大幅度的特征值。

（4）特征交叉或组合：线性回归模型中，假设特征和响应之间为线性关系。如果该假设不满足，我们可以对特征进行多项式扩展，变换后的特征组合的线性组合相当于对原始特征的非线性组合。不过需要注意的是，多项式扩展后的特征数目会随多项式的阶数呈指数增长，所以多项式的阶数通常限制为2。Scikit-Learn的类可以实现该功能。

离散型特征常用的特征工程方法介绍如下。

（1）标签编码（Label Encoding）：对有序特征，一种编码方式是直接变换整数。如表示尺寸的特征，特征取值有：XS、S、M、L、XL，我们可以将其变换为整数：1，2，3，4，5，6。不过考虑到变换为整数后，特征和响应可能不满足线性关系，最好采用独热编码。Scikit-Learn中的工具可实现将离散型特征值转换成整数。亦可手动指定特征值与整数之间的映射关系，可通过Pandas的函数实现。

（2）独热编码（One-Hot Encoding）：独热编码是离散型特征编码用的最多的编码方式。考虑到任意具有种取值的离散型特征，独热编码方案将该属性编码或变换成维二进制特征向量（向量中的每一维的值只能为0或1），且其中只有某一维的值为1（独热））。独热编码可通过联合使用Scikit-Learn中的和实现，亦可用Pandas的函数实现。二者的使用场合稍有不同，请参看案例代码的详细介绍。由于独热编码后的特征维度与特征的可能的取值数目有关，所以通常当较小时才采用。

（3）计数编码（Counting-Based Coding）：当特征的可能取值数目非常多时（如 IP 地址），一种方式是使用基于概率的统计信息，用该特征值取响应值的概率进行编码。例如，基于过去 IP 地址历史数据和 DDOS 攻击中所使用的历史数据，用每个 IP 地址会被 DDOS 攻击的可能性编码该IP地址，描述了如果将来出现该IP 地址，引起 DDOS 攻击的概率值。该编码方案需要详尽的历史数据以得到可靠的概率模型。

（4）哈希编码（Hashing Coding）：哈希编码也用于特征取值很多的情况，将特征值变换为一个低维稠密向量。Scikit-Learn的 类实现了特征哈希方案。

（5）嵌入式编码（Embedding Coding）：嵌入式编码的使用场合和编码结果同哈希编码类似，不同的是编码值不是通过确定的哈希函数，而是通过某种方式学习得到，学习方式可以与响应无关（如Word2Vec），也可以与响应有关（如CTR预估中，用户ID和商品ID等的编码与CTR预估模型一起学习）。

另外对时间型特征和地理位置型特征，可能也需要根据具体任务进行适当编码。本书通过案例对涉及的各种情况在代码中详细讲解。

广告数据集中3个特征均为数值型特征，且量纲相同，我们暂且假设输入特征销量之间为线性关系，因此特征工程部分无需操作。但3个特征的取值范围不同，如果采用梯度下降/随机梯度下降法求解，还是将所有特征的取值范围缩放到相同区间。为保险起见，我们还是对输入特征进行标准化处理。

#### 3．线性回归模型应用

在Scikit-Learn中，最小二乘线性回归、岭回归、Lasso和弹性网络模型分别为：、、和 。

无超参数，当我们确信特征之间无共线性关系时使用。默认采用的是如式（2-68）所述的GCV对超参数进行调优，和 采用折交叉验证对超参数进行调优。

的目标函*数为*，同式（2-15）一致，只是这里正则参数用表示。Scikit-Learn中的岭回归实现支持多种优化算法，可根据数据集的情况选择合适的算法（通过参数设置）。

的目标函数为，同式（2-22）一致，只是这里训练误差用样本数的2倍做了平均，正则参数用表示。在Lasso模型中，当正则参数超过如式（2-56）所示的最大值，所有系数均为0，因此默认的正则参数搜索范围为，其中，并且对之间的值在域上均匀采样个值。

 的目标函数为，同式（2-25）一致，只是这里训练误差用样本数的2倍做了平均，L1正则和L2正则参数用另外两个参数来表示：正则参数，和L1正则的比例参数。其中参数的默认搜索范围同中的相同，。

Scikit-Learn中各种学习器（）的API接口几乎相同，方便我们快速掌握不同学习器的使用。常用的学习器API接口包括：构造函数、模型训练、预测。

我们从中数据集中随机选择20%样本作为测试数据，其他80%样本为训练数据，采用、、和 等4个模型预测广告费用与产品销量之间的关系，得到4个模型的回归系数如表2-1所示。其中 得到的模型和相同，即最佳L1正则的比例为1.0。可以看出，岭回归、Lasso和弹性网络得到的回归系数绝对值均比最小二乘险线性回归小，即起到了权值收缩的效果。另外模型Lasso和弹性网络得到的模型中，特征的系数为0，体现了其稀疏的性质。

表2-1. 广告数据集上不同线性回归模型的系数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 特征 | 最小二乘线性回归系数 | 岭回归系数 | Lasso系数 | 弹性网络系数 |
| TV | 3.983944 | 3.981524 | 3.921642 | 3.921642 |
| radio | 2.860230 | 2.858304 | 2.806374 | 2.806374 |
| newspaper | 0.038194 | 0.038925 | 0.000000 | 0.000000 |
| 截距项 | 13.969091 | 13.969282 | 13.972528 | 13.972528 |

4个模型在训练集和测试集上的性能如表2-2所示，表中我们采用R方分数作为性能度量指标。可以看出，最小二乘线性回归模型在训练集上性能最好，但在测试集上性能最差；Lasso模型在测试集上性能最好。

表2-2. 广告数据集上不同线性回归模型的性能

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | 最小二乘线性回归 | 岭回归 | Lasso | 弹性网络 |
| 训练集上性能 | 0.896285 | 0.896285 | 0.895925 | 0.895925 |
| 测试集上性能 | 0. 893729 | 0. 893865 | 0. 899197 | 0. 899197 |

在带正则的模型中，不同正则参数对应的模型性能不同。、和中不同正则参数用交叉验证估计得到的MSE变化如图2-5所示。从图中可以看出，随着正则参数的增大，模型变得越来越简单，交叉验证估计的测试误差会先减小，后增大。MSE的最低点对应最佳的超参数，图中用竖直虚线表示。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| （a）岭回归 | （b）Lasso | （c）弹性网络 |

图2-5 广告数据集上不同超参数对应模型的交叉验证的得到测试误差估计（MSE）

## **标题2** 2.7 案例分析2—共享单车骑行量预测

这节我们在共享单车数据集上采用线性回归模型实现骑行量预测。与2.6节的广告数据集相比，共享单车数据集中包含有更多冗余特征，因此正则变得更重要。

#### 1．数据分析

共享单车数据集包含了2年中共731天的共享单车骑行量，以及每天的天气特征。其中特征有22维，包含时间（日期、年、季节、月份、星期、是否节假日）和天气（晴、阴、雨、雪）等离散型特征，以及温度、体感温度、湿度和风速等数据值特征。可以看出，多个日期特征，如季节和月份，有冗余，同时温度和体感温度之间的相关系数高达到99%，严重冗余。因此我们预期不带正则的最小二乘线性回归效果不会太好，需要对模型加正则约束。

#### 2．数据探索和特征工程

对数据集中的离散型特征，由于每个特征的取值数目不太多，可采用独热编码编码。如对季节特征，由于其有4种取值（1，2，3，4），独热编码后变成4维特征。对每个样本，这4维特征种有且仅有1维为1，如表2-3所示。对数据集中的数值型特征，进行标准化处理去量纲。

表2-3. 离散特征编码示例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **原始特征**season | **独热编码后的特征** | | | |
| season\_1 | season\_2 | season\_3 | season\_4 |
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 |

#### 3．线性回归模型应用

我们从中数据集中随机选择20%样本作为测试数据，其他80%样本为训练数据，采用、、等3个模型预测每天的共享单车骑行量。3个模型的回归参数如图2-6所示。由于模型参数较多（33维），我们只挑选了绝对值最大的正10个系数和负10个系数展示。从图中可以看出，最小二乘回归模型的系数的绝对值非常大（），而带正则的岭回归模型和Lasso模型中系数的绝对值小得多（），且Lasso模型中有6个系数为0。

3个模型在训练集和测试集上的性能如表2-4所示，性能度量指标为。可以看出，最小二乘回归模型在训练集上性能最好，但在测试集上性能最差；岭回归模型在测试集上性能最好，Lasso的性能介入二者之间。

|  |  |  |
| --- | --- | --- |
|  |  |  |
| （a）最小二乘线性回归 | （b）岭回归 | （c）Lasso |

图2-6. 共享单车数据集上不同线性回归模型的回归系数

表2-4. 共享单车骑行量预测数据集上不同线性回归模型的性能

|  |  |  |  |
| --- | --- | --- | --- |
| 数据集 | 最小二乘线性回归 | 岭回归 | Lasso |
| 训练集上的性能 | 752.257390 | 754.036662 | 752.643468 |
| 测试集上的性能 | 785.595792 | 776.975361 | 784.878890 |

## **标题2** 2.8 小结

本章从以下5个方面介绍了线性回归模型。

（1）模型的形式：线性回归模型模型假设响应与输入特征之间满足线性关系：。

（2）模型的目标函数：线性回归模型目标函数包含两部分：训练集上的损失之和、正则项。线性回归模型的损失函数可取L2损失或Huber损失，正则项可取L2正则、L1正则、或L2正则+L1正则。

（3）目标函数的优化求解：线性回归模型的目标函数为凸函数，可采用包括解析法、梯度下降法、坐标轴下降法和牛顿法（第3章介绍）等诸多优化算法求解，可根据数据的规模和特点选择合适的优化算法。

（4）模型性能指标：回归任务的性能指标包括MSE、MAE、R方分数，可根据任务要求选择合适的性能评价指标。

（5）超参数调优：岭回归采用GCV进行超参数调优，Lasso和弹性网络采用普通交叉验证调优超参数。

## **标题2** 2.9 习题

1. 关于回归问题中的残差，哪一种说法是正确的？

（A）残差的平均值总是零

（B）残差平均值总是小于零

（C）残差平均值总是大于零

（D）对于残差没有限制

2. 如果Lasso的正则参数很大，会发生什么？

（A）一些系数将变为零

（B）一些系数将接近零，但不是绝对零

3. 下面哪种模型的参数没有解析解？

（A） 岭回归

（B） Lasso

（C） 岭回归和Lasso

（D） 两者都不是

4. 我们可以用正规方程组方法来计算线性回归系数。关于正规方程组，下列哪个不正确的？

（A）不必选择学习率

（B）当特征数很大时，速度变慢

（C）不需要迭代

（D）需要迭代

5. 采用梯度下降法，求函数的最小值，对比学习率分别为0.1, 0.3和0.9时算法的收敛速度。

6. 采用线性回归模型对波士顿房价数据集进行建模。数据集共有506个样本，每个样本包含波士顿某地区的房屋的13个属性和该地区的房价中位数，我们根据该地区房屋的属性来预测该地区的房价中位数。数据集中各字段说明如表2-5。

（1）分析各特征和响应的分布，并对特征进行适当变换。

（2）随机选择其中80%做训练数据，剩下20%为测试数据。

（3）用训练数据训练最小二乘线性回归、岭回归和Lasso模型，注意岭回归和Lasso模型的正则超参数调优，评价指标为RMSE。

（4）比较用上述三种模型得到的各特征的回归系数，以及各模型在测试集上的性能。

表2-4. 波士顿房价数据集的字段说明

|  |  |
| --- | --- |
| 字段名 | 说明 |
| CRIM | 城镇人均犯罪率 |
| ZN | 占地面积超过2.5万平方英尺的住宅用地比例 |
| INDUS | 城镇非零售业务地区的比例 |
| CHAS | 是否靠近查尔斯河（1表示在河边；否则为0） |
| NOX | 一氧化氮浓度（每1000万份） |
| RM | 平均每居民房数 |
| AGE | 在1940年之前建成的所有者占用单位的比例 |
| DIS | 与五个波士顿就业中心的加权距离 |
| RAD | 辐射状公路的可达性指数 |
| TAX | 每10,000美元的全额物业税率 |