BIBDR 北京大数据研究院 | 博雅大数据学院 BOYA BIG DATA COLLEGE | 数据酷客 COOKDATA

北京市大数据人才培训示范基地

第二讲：回归

欧高炎，北京大学博士、博士后

数据酷客创始人，博雅大数据学院院长

“机器学习十讲”直播安排

时间	主题	介绍
5/14	机器学习介绍	机器会学习吗？
5/21	回归	回归初心，方得始终
5/28	分类	分门别类，各得其所
6/4	模型提升	三个臭皮匠，顶个诸葛亮
6/11	聚类	物以类聚，人以群分
6/18	降维	取其精华，去其糟粕
6/23	最优化	摸着石头过河，蒙着眼睛爬山
7/2	维度灾难	来自维数的诅咒
7/9	深度学习	深层次学习的艺术
7/16	强化学习	让机器像人类一样学习

矩阵的逆

- 对于 $n \times n$ 方阵 $\mathbf{A}$ ，如果存在矩阵 $\mathbf{B}$ 使得 $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$，则称 $\mathbf{B}$ 为 $\mathbf{A}$ 的逆矩阵，记为 $\mathbf{A}^{-1}$
- 若 $\mathbf{A}$ 为可逆矩阵，则其逆矩阵是唯一的
- 如何判断矩阵是否可逆？
  - 行列式不等于0
  - 满秩
  - 行（或列）向量组线性无关
  - …

NumPy 矩阵求逆函数

- numpy.linalg 模块包含线性代数的函数，可计算逆矩阵、求特征值、解线性方程组以及求解行列式等。

- 行列式：np.linalg.det(A)

- 计算逆矩阵：np.linalg.inv(A)

```
import numpy as np
#格式化numpy输出
np.set_printoptions(formatter={'float': '{: 0.2f}'.format})
A = np.array([[4,5,1],[0,8,3],[9,4,7]])
#求行列式
print("行列式: " + str(np.linalg.det(A)))
B = np.linalg.inv(A) #求逆
print("A的逆矩阵为B: ")
print(A_inv)
print("BA:") #结果验证
print(B.dot(A))
print("AB:")
print(A.dot(B))
```

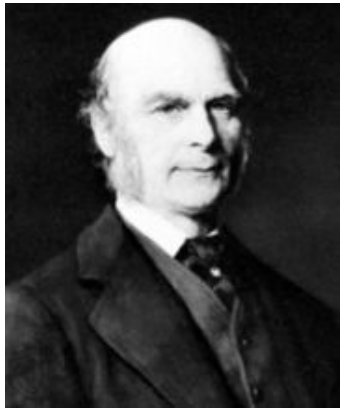
```
行列式: 239.00000000000009
A的逆矩阵为B:
[[ 0.18 -0.13  0.03]
 [ 0.11  0.08 -0.05]
 [-0.30  0.12  0.13]]
BA:
[[ 1.00  0.00  0.00]
 [ 0.00  1.00  0.00]
 [-0.00  0.00  1.00]]
AB:
[[ 1.00 -0.00  0.00]
 [ 0.00  1.00  0.00]
 [ 0.00  0.00  1.00]]
```

什么是回归？

- 回归（Regression）这一概念最早由英国生物统计学家高尔顿和他的学生皮尔逊在研究父母亲和子女的身高遗传特性时提出
- “子女的身高趋向于高于父母的身高的平均值，但一般不会超过父母的身高。”-- 《遗传的身高向平均数方向的回归》

$$Y = 0.8567 + 0.516 * X \quad \text{(单位为米)}$$

- 回归如今指的用一个或多个自变量来预测因变量的数学方法
- 在机器学习中，回归指的是一类预测变量为连续值的有监督学习方法

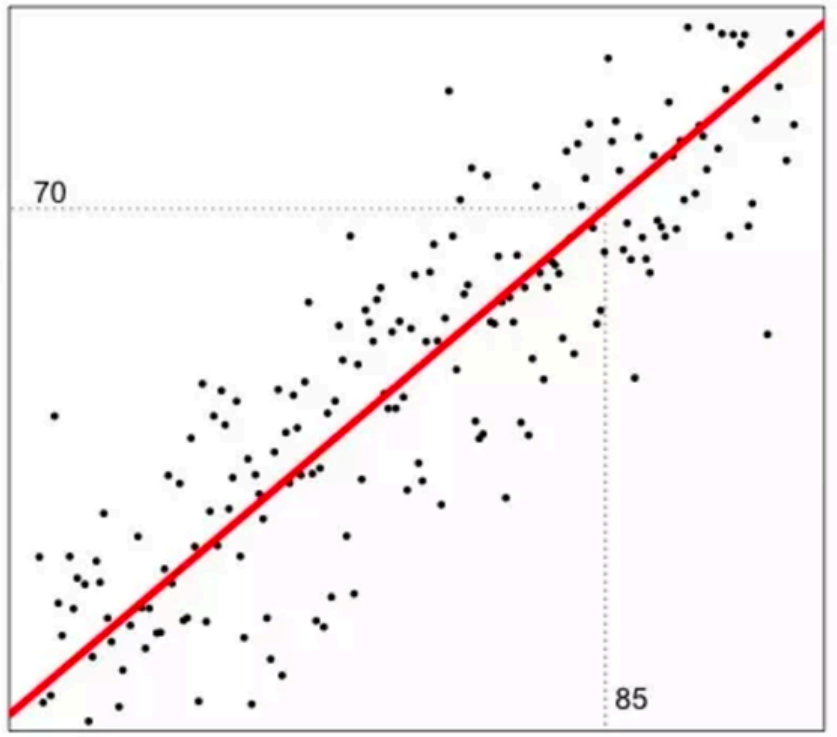


高尔顿

皮尔逊

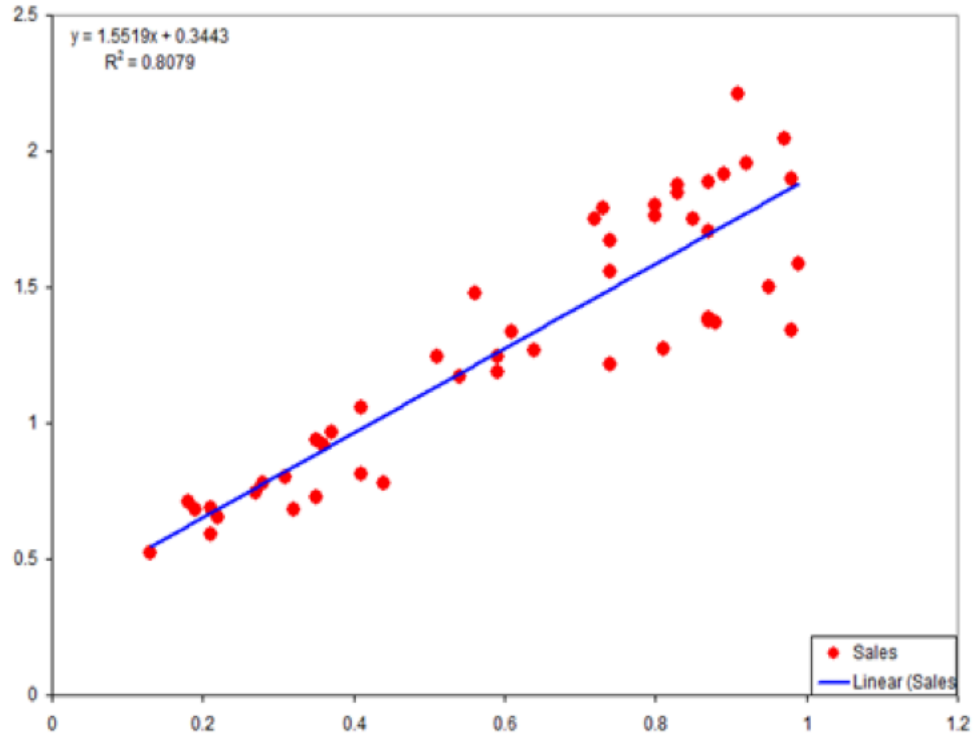
在回归模型中，需要预测的变量叫做因变量，用来解释因变量变化的变量叫做自变量。

成为领导者可能性



IQ

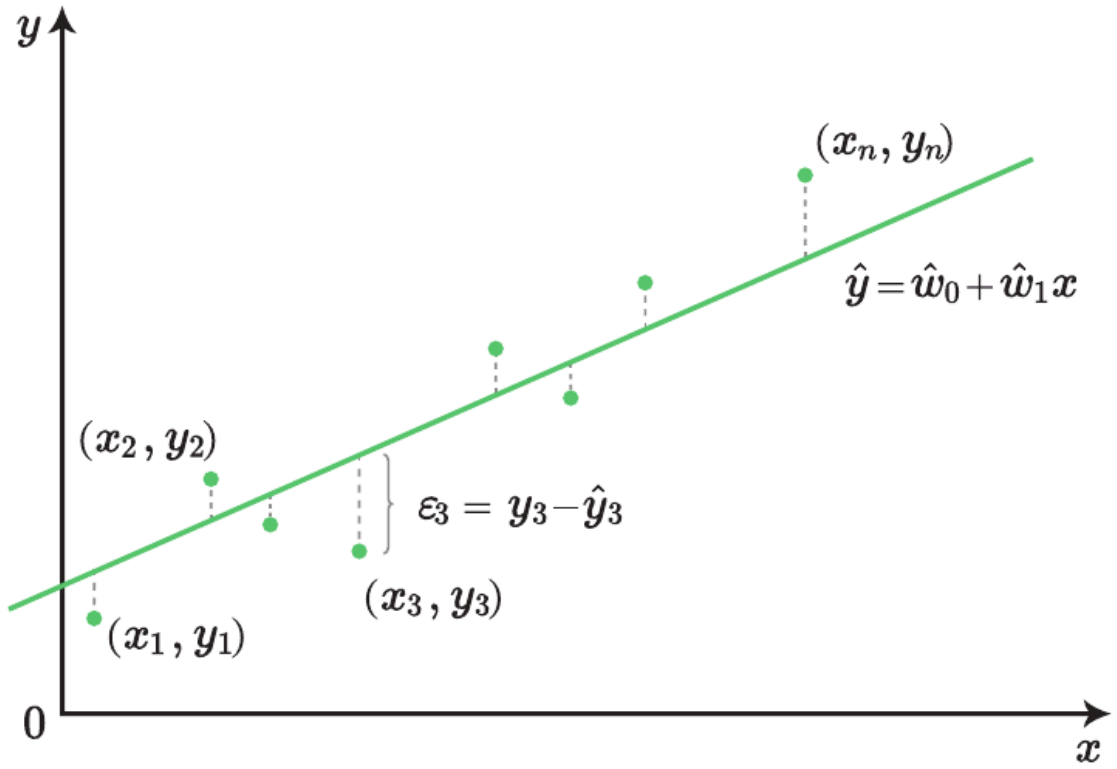
销量



广告投入

- 模型为 $y = w_1 x + w_0$ ，其中 w_0, w_1 为回归系数
- 给定训练集 $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ ，我们的目标是找到一条直线 $y = w_1 x + w_0$ 使得所有样本尽可能落在它的附近。
- 优化目标为：

$$\min_{(w_0, w_1)} \sum_{i=1}^{n} (y_i - w_1 x_i - w_0)^2$$



- 记优化目标为 $\mathrm{L}(\mathrm{w}_1, \mathrm{w}_0) = \sum_{i=1}^{n}(y_i - \mathrm{w}_1 x_i - \mathrm{w}_0)^2$
  - $\mathrm{L}(\mathrm{w}_1, \mathrm{w}_0)$ 为二次凸函数，分别对$\mathrm{w}_1, \mathrm{w}_0$求导并令导数为零：

$$\begin{cases} \dfrac{\partial \mathrm{L}(\mathrm{w}_1, \mathrm{w}_0)}{\partial \mathrm{w}_1} = 2\sum_{i=1}^{n}(y_i - \mathrm{w}_1 x_i - \mathrm{w}_0)(-x_i) = 0 \\ \dfrac{\partial \mathrm{L}(\mathrm{w}_1, \mathrm{w}_0)}{\partial \mathrm{w}_0} = 2\sum_{i=1}^{n}(y_i - \mathrm{w}_1 x_i - \mathrm{w}_0)(-1) = 0 \end{cases}$$

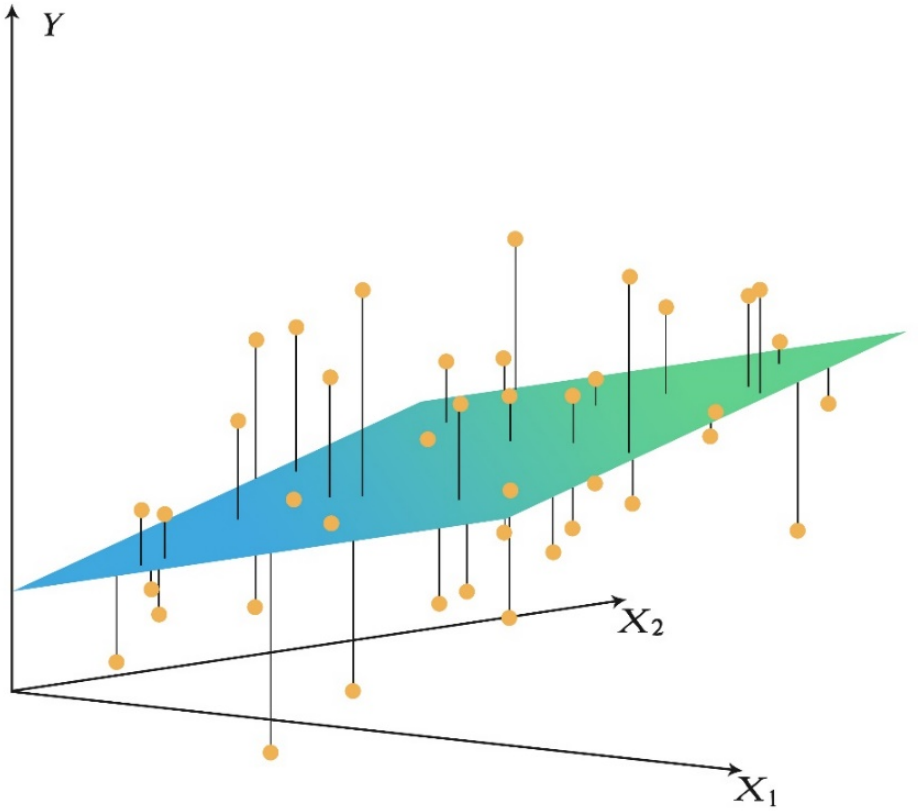
- 解上述线性方程组得

$$\begin{cases} \mathrm{w}_1 = \dfrac{n\sum_{i=1}^{n} y_i x_i - (\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \\ \mathrm{w}_0 = \dfrac{\sum_{i=1}^{n} y_i}{n} - \dfrac{\sum_{i=1}^{n} x_i}{n}\mathrm{w}_1 \end{cases}$$

- y 是多个特征的线性组合

$$y = \mathrm{w}_1 x_1 + \mathrm{w}_2 x_2 + \cdots + \mathrm{w}_d x_d + \mathrm{w}_0$$

- 训练集 $\mathrm{D} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_\mathrm{n}, y_\mathrm{n})\}$
- $\boldsymbol{x}_\mathrm{i}$为 d 维特征向量 $\boldsymbol{x}_\mathrm{i} = (x_{11}, x_{12}, \cdots, x_{1d})^\mathrm{T}$
- 假设模型对 $\boldsymbol{x}_\mathrm{i}$ 的预测值为 $\hat{y}_i = \mathrm{w}_1 x_{i1} + \cdots + \mathrm{w}_d x_{id} + \mathrm{w}_0$
- 优化目标为 $\mathrm{L}(\mathbf{w}) = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$



寻找一个超平面，使得训练集中样本到超平面的误差平方和最小

多元线性回归的矩阵表示

- 假设训练集的特征部分记为 $n\times(d+1)$ 矩阵 $\mathbf{X}$，其中最后一列取值全为 1
- 标签部分记为 $\boldsymbol{y}=(y_1,y_2,\ldots,y_n)^{\mathrm{T}}$，参数记为 $\mathbf{w}=(w_1,w_2,\ldots,w_d,w_0)^{\mathrm{T}}$
- 多元线性回归模型为：

$$\hat{\boldsymbol{y}}=\mathbf{X}\mathbf{w}$$

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{21} & \cdots & x_{21} & 1 \\ \cdots & \cdots & \cdots & \cdots & 1 \\ x_{i1} & x_{i2} & \cdots & x_{id} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} & 1 \end{bmatrix} \quad \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_d \\ w_0 \end{bmatrix} \quad \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \cdots \\ \hat{y}_i \\ \cdots \\ \hat{y}_n \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_i \\ \cdots \\ y_n \end{bmatrix}$$

$$\mathbf{X} \qquad \mathbf{w} \qquad \hat{\boldsymbol{y}} \qquad \boldsymbol{y}$$

优化目标的矩阵表示

- 最小化均方误差函数为：$\mathrm{L}(\mathbf{w}) = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$

- $\mathrm{L}(\mathbf{w}) = (\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$

  $= (\mathbf{X}\mathbf{w} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \boldsymbol{y})$

$$\hat{\boldsymbol{y}} - \boldsymbol{y}$$

$$\begin{bmatrix} \hat{y}_1 - y_1 \\ \hat{y}_2 - y_2 \\ \cdots \\ \hat{y}_i - y_i \\ \cdots \\ \hat{y}_n - y_n \end{bmatrix}$$

$$(\hat{\boldsymbol{y}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y}) = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

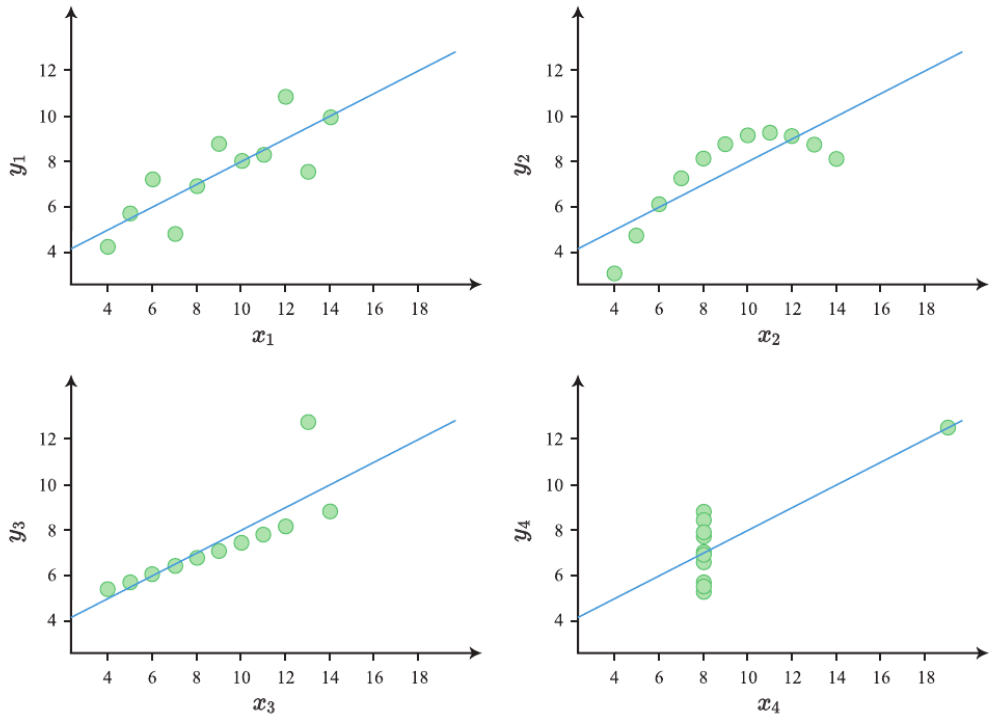
- 优化目标函数为：$\mathrm{L}(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \boldsymbol{y})$
- 当矩阵 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 满秩时：

令 $\frac{\partial \mathrm{L}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^{\mathrm{T}}(\boldsymbol{y} - \mathbf{X}\mathbf{w}) = \mathbf{0}$，可得：$\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$

线性回归的问题

- 实际数据可能不是线性的？
- 多项式回归：使用原始特征的二次项、三次项
  - 线性回归解决非线性问题
  - 问题：维度灾难、过度拟合

安斯库姆四重奏



线性回归的问题

- 多重共线性

最小二乘的参数估计为 $\hat{\mathbf{w}} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$ ，如果变量之间存在较强的共线性，则$\mathbf{X}^{\mathrm{T}}\mathbf{X}$近似奇异，对参数的估计变得不准确，造成过度拟合现象。

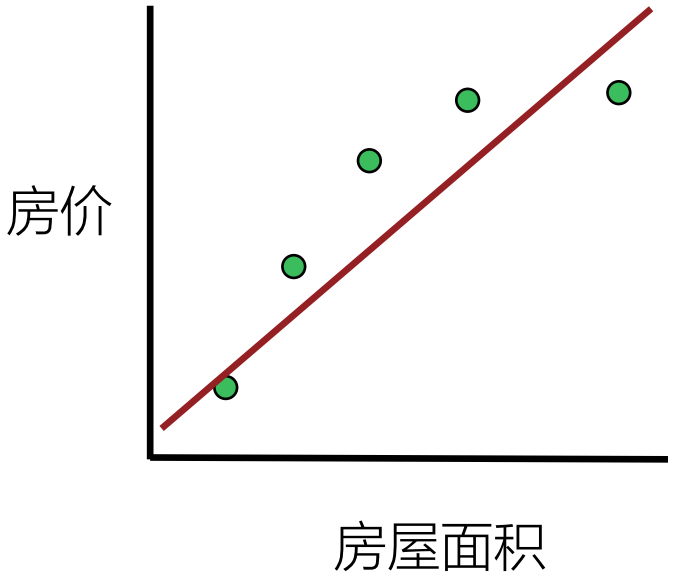
- 解决方法：正则化、主成分回归、偏最小二乘回归

线性回归的问题

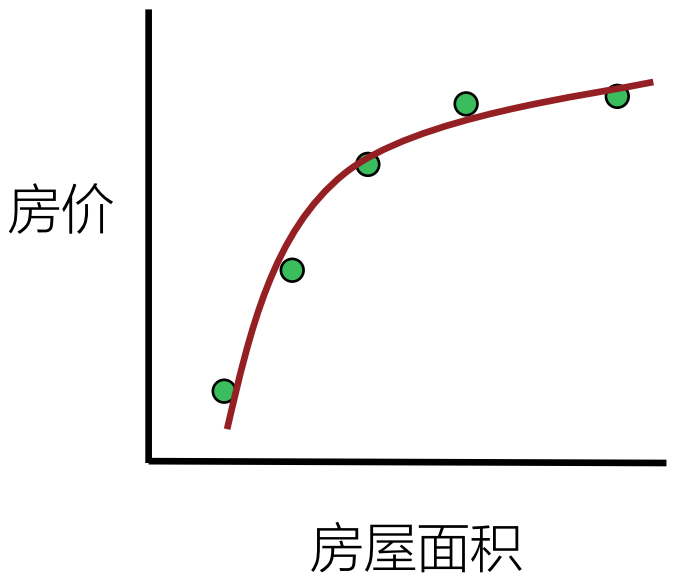
- 过度拟合问题：当模型的变量过多时，线性回归可能会出现过度拟合问题

例如在房价预测问题中，假设x表示房屋面积，如果将x^2, x^3等作为变量引入可能出现如下情况：

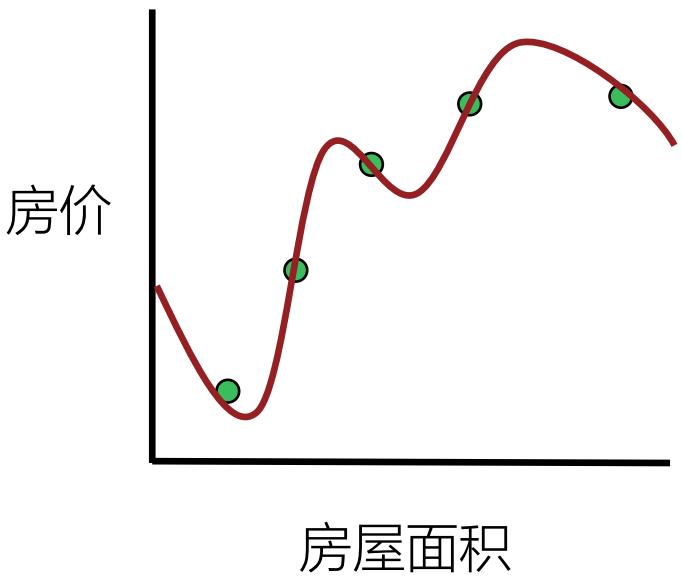
$w_0 + w_1 x$：
训练误差大



$w_0 + w_1 x + w_2 x^2$：
拟合很好



$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$：
过度拟合

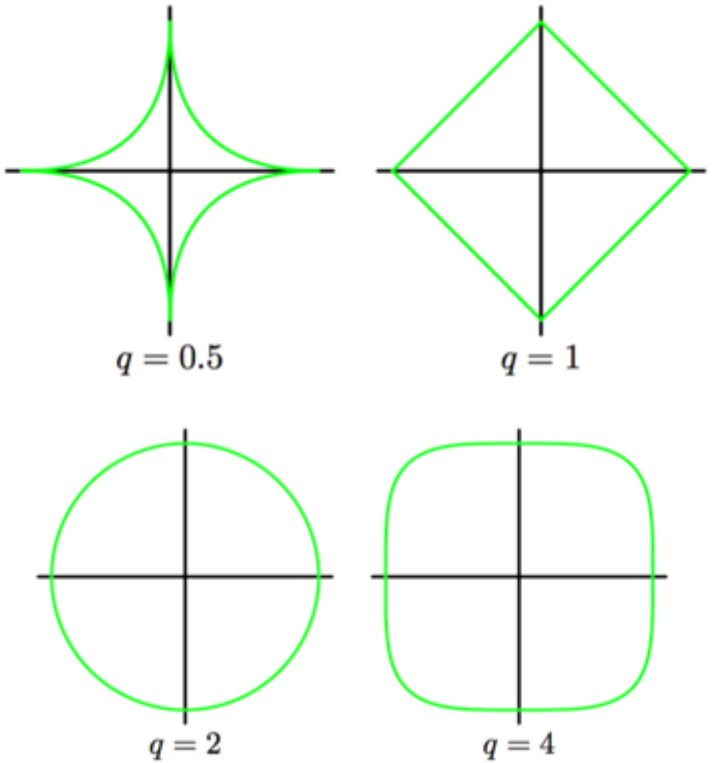


- 正则化可以减小线性回归的过度拟合和多重共线性等问题

$$(\mathbf{Xw}-\boldsymbol{y})^{\mathrm{T}}(\mathbf{Xw}-\boldsymbol{y})+\lambda\|\mathbf{w}\|_{\boldsymbol{q}}^{\boldsymbol{q}}$$

正则项或惩罚函数

- $q=2$：岭回归（Ridge）
- $q=1$：LASSO



$\|w\|_q$ 示意图

- 岭回归思路：在最小二乘法的目标函数上加上一个对$\mathbf{w}$的惩罚函数

$$\lambda\|\mathbf{w}\|_2^2$$

- 线性回归的目标函数：$(\mathbf{Xw}-\boldsymbol{y})^{\mathrm{T}}(\mathbf{Xw}-\boldsymbol{y})$
- 岭回归的目标函数变为：$(\mathbf{Xw}-\boldsymbol{y})^{\mathrm{T}}(\mathbf{Xw}-\boldsymbol{y})+\lambda\|\mathbf{w}\|_2^2$
- 对 $\mathbf{w}$ 求导并令导数等于零易得：$\hat{\mathbf{w}}^{\mathrm{ridge}}=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$

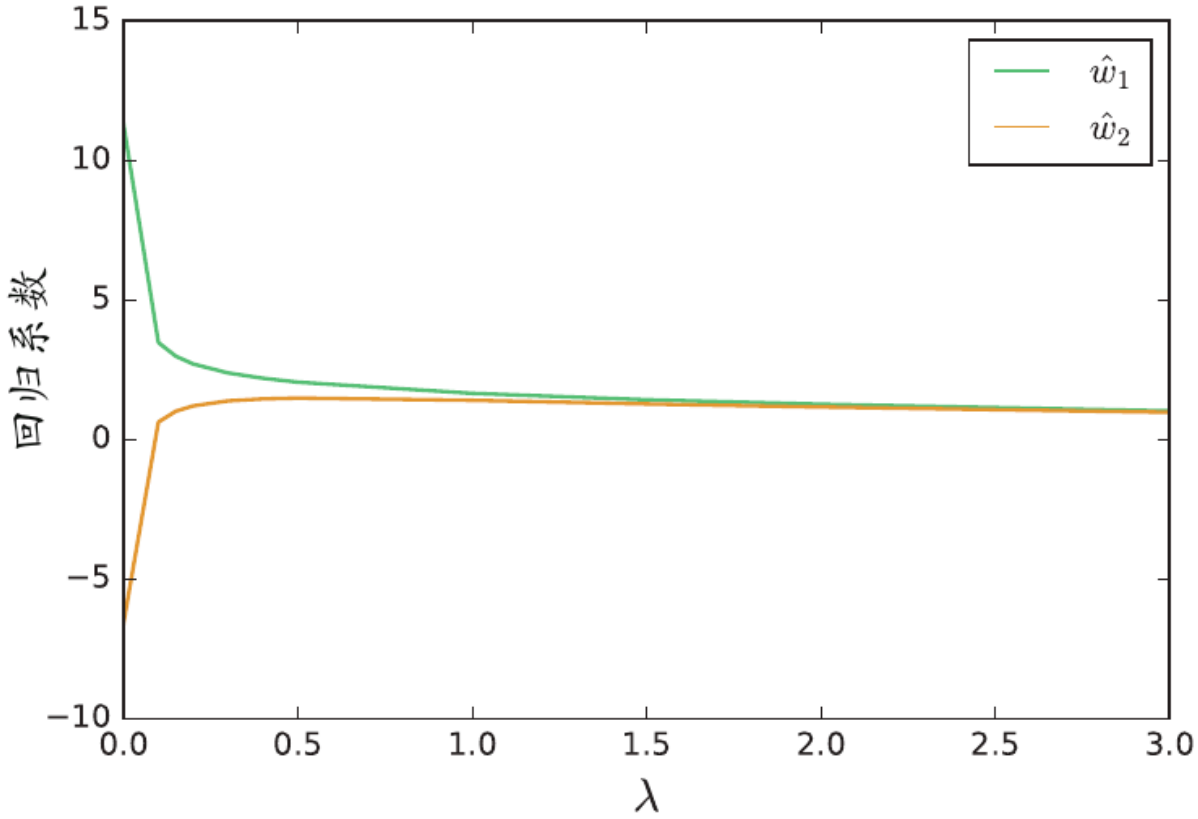
- 当不断增大正则化参数λ，估计参数$\hat{\mathbf{w}}^{\text{ridge}}(\lambda)$（也称岭回归系数）在坐标系上的变化曲线称为岭迹。岭迹波动很大，说明该变量有共线性。

- 下表是一个二元岭回归的结果：

λ	0	0.1	0.15	0.2	0.3	0.4	0.5	1.0	1.5	2.0	3.0
$\hat{\mathbf{w}}_1^{\text{ridge}}(\lambda)$	11.31	3.48	2.99	2.71	2.39	2.20	2.06	1.66	1.43	1.27	1.03
$\hat{\mathbf{w}}_2^{\text{ridge}}(\lambda)$	-6.59	0.63	1.02	1.21	1.39	1.46	1.49	1.41	1.28	1.17	0.98

岭迹分析

- 根据岭迹做超参数 λ 的选择
- 在 λ ∈(0,0.5)的范围内波动较大，故需要加入正则化项重新进行参数估计，可选 λ=1

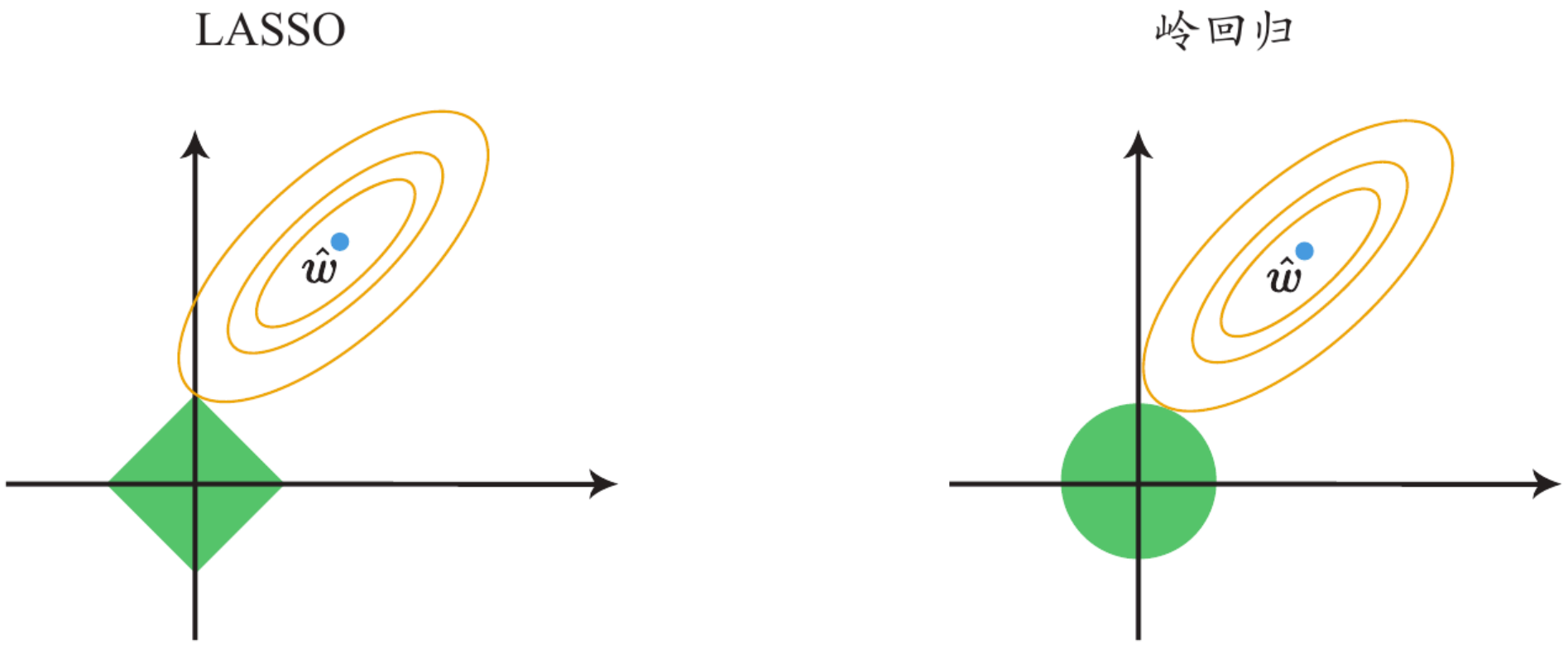


LASSO: Least Absolute Shrinkage and Selection Operator

- LASSO 是一种系数压缩估计方法，它的基本思想是通过追求稀疏性自动选择重要的变量
- LASSO 的目标函数：$(\mathbf{Xw}-\boldsymbol{y})^{\mathrm{T}}(\mathbf{Xw}-\boldsymbol{y})+\lambda\|\mathbf{w}\|_1$
- LASSO 的解 $\widehat{\mathbf{w}}^{\mathbf{LASSO}}$ 没有解析表达式，常用的求解算法包括坐标下降法、LARS算法和ISTA算法等

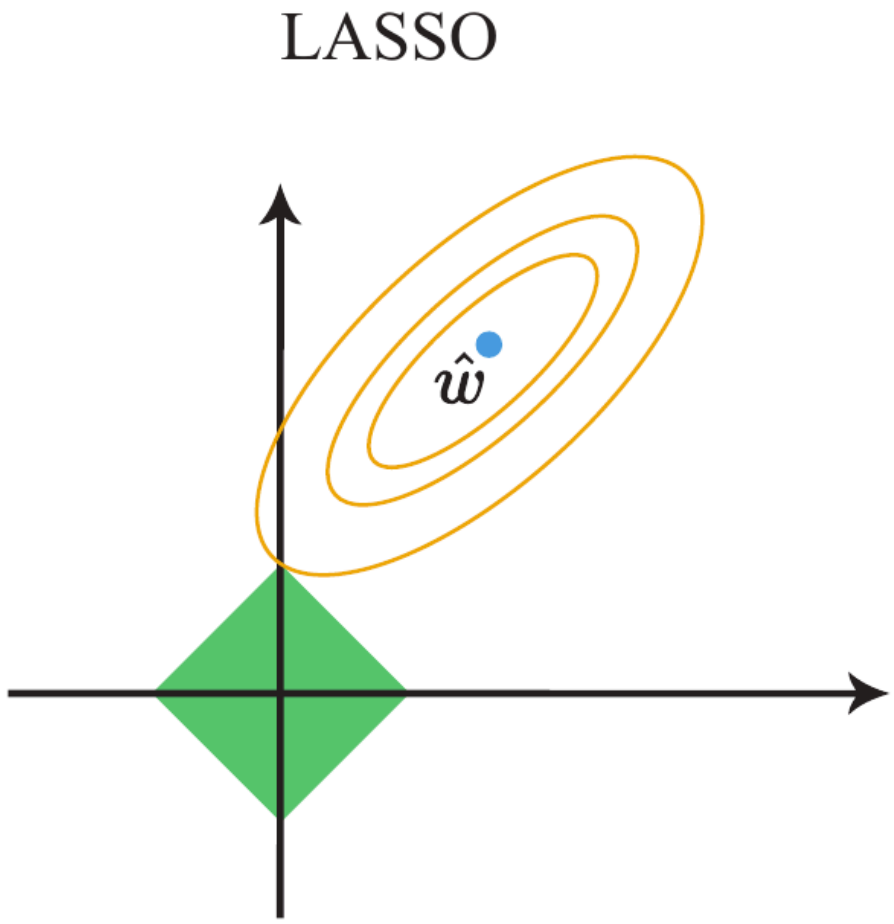
岭回归与 LASSO 如何寻求最优解？

惩罚函数（绿色）与目标函数（橙色）

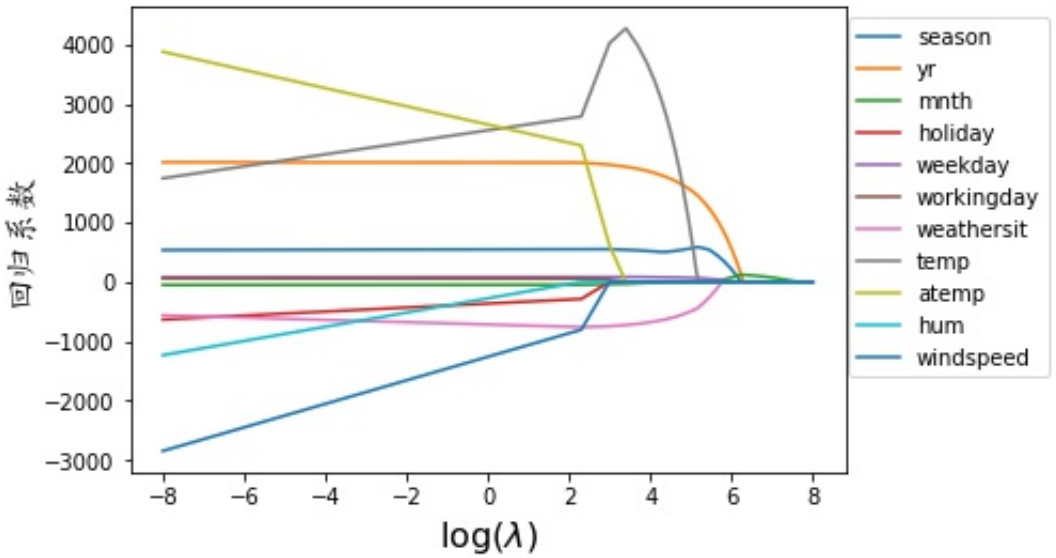


L1惩罚函数形式

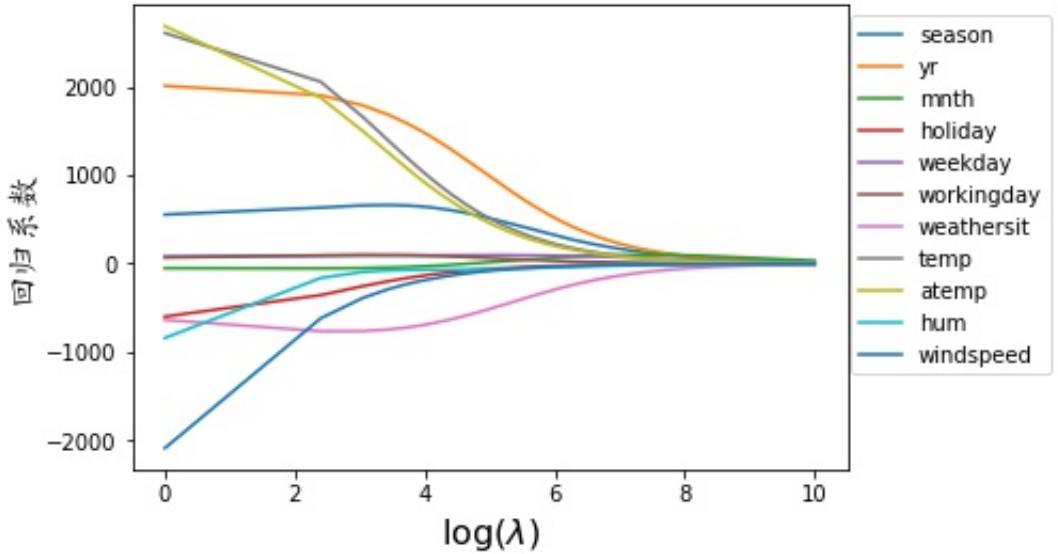
- 如右图所示：图中绿色区域表示约束区域，黄色线为残差平方和函数的等高线
- 通过添加 L1惩罚函数，LASSO 方法可以得到角点解，即稀疏的最优解 $\hat{\mathbf{w}}$，此时 $\hat{\mathbf{w}}_i = \mathbf{0}$，我们可以将对应的变量从模型中删除。



正则化路径分析



LASSO



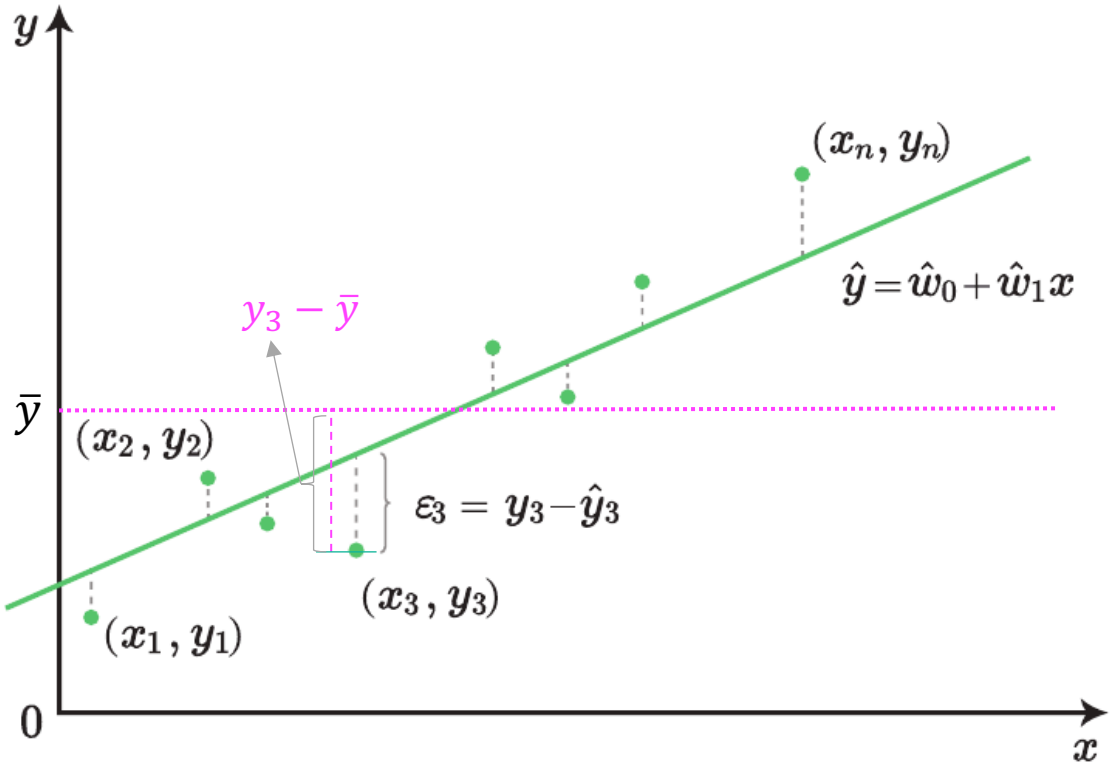
岭回归

随着 λ 增大，LASSO的特征系数逐个减小为0，可以做特征选择；而岭回归变量系数几乎同时趋近于0。

回归模型的评价指标

- 均方误差：$\mathrm{MSE}(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$
- 均方根误差：$\mathrm{RMSE}(\boldsymbol{y},\hat{\boldsymbol{y}})=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$
- 平均绝对误差：$\mathrm{MAE}(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|$
- 决定系数：$\mathrm{R}^2(\boldsymbol{y},\hat{\boldsymbol{y}})=1-\frac{SS_{res}}{SS_{tot}}=1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$

其中y_i为真实值，$\bar{y}$为真实值的平均值，$\hat{y}_i$为模型估计值



案例：使用回归模型预测鲍鱼年龄

基于一份鲍鱼数据集，利用回归模型建立自动鲍鱼年龄预测模型。

1. 使用 Python 实现线性回归和岭回归算法，并与 Sklearn 中的实现进行对比

2. 借助 Sklearn 工具，对线性回归、岭回归和 LASSO 三种模型的预测效果使用 MAE 和决定系数进行效果评估

3. 残差图和正则化路径对模型表现进行分析

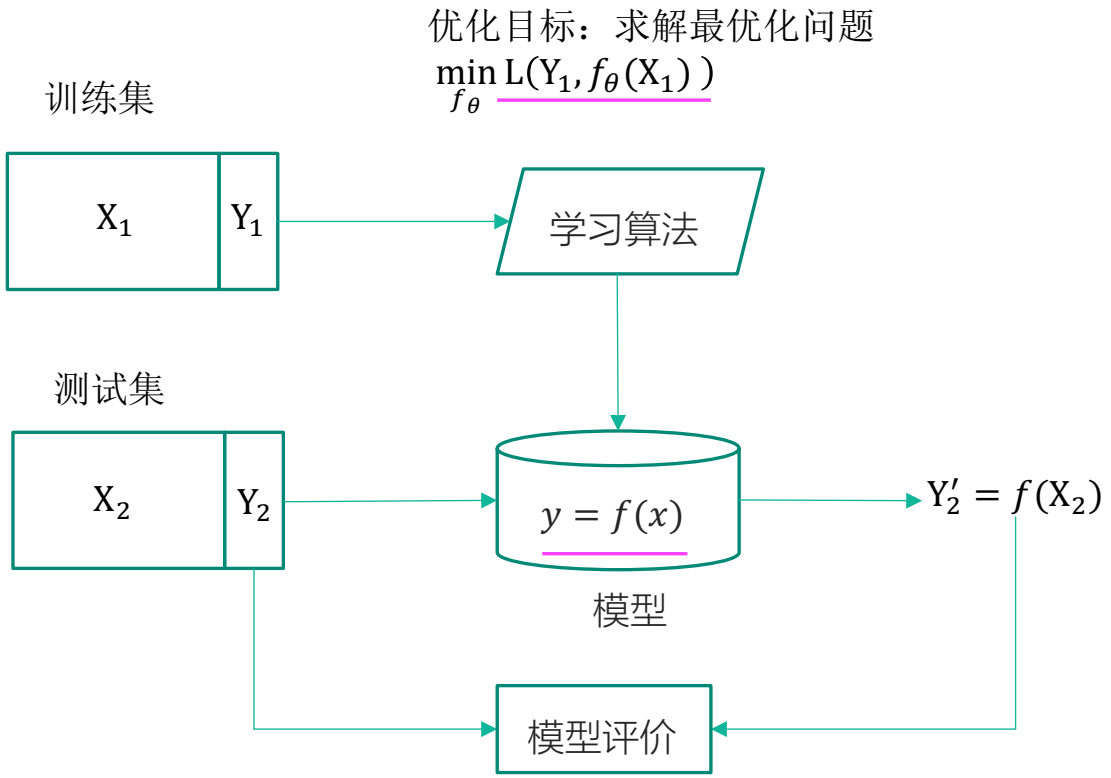
BIBDR 北京大数据研究院 | 博雅大数据学院 BOYA BIG DATA COLLEGE | 数据酷客 COOKDATA

数据酷客

数据科学人工智能

总结

- 回归：典型的有监督学习，y 为连续型
- 线性回归：$y = f(x) = \mathrm{w}_1 x_1 + \mathrm{w}_2 x_2 + \cdots + \mathrm{w}_d x_d + \mathrm{w}_0$
  - 目标函数：$\mathrm{L}(\mathbf{w}) = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = (\mathbf{X}\mathbf{w} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \boldsymbol{y})$
  - 解：$\hat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 岭回归：防止过度拟合
  - 目标函数：$(\mathbf{X}\mathbf{w} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \boldsymbol{y}) + \lambda\|\mathbf{w}\|_2^2$
  - 解：$\hat{\mathbf{w}}^{\mathrm{ridge}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- LASSO：同时进行特征选择，防止过度拟合
  - 目标函数：$(\mathbf{X}\mathbf{w} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \boldsymbol{y}) + \lambda\|\mathbf{w}\|_1$
- 回归模型评价和分析
  - MSE、RMSE、R^2、残差图



数据酷客

数据科学人工智能

加入数据酷客交流群