



第4章 回归

无平不陂，无往不复。

——《周易·泰》

北京交通大学《机器学习》课程组

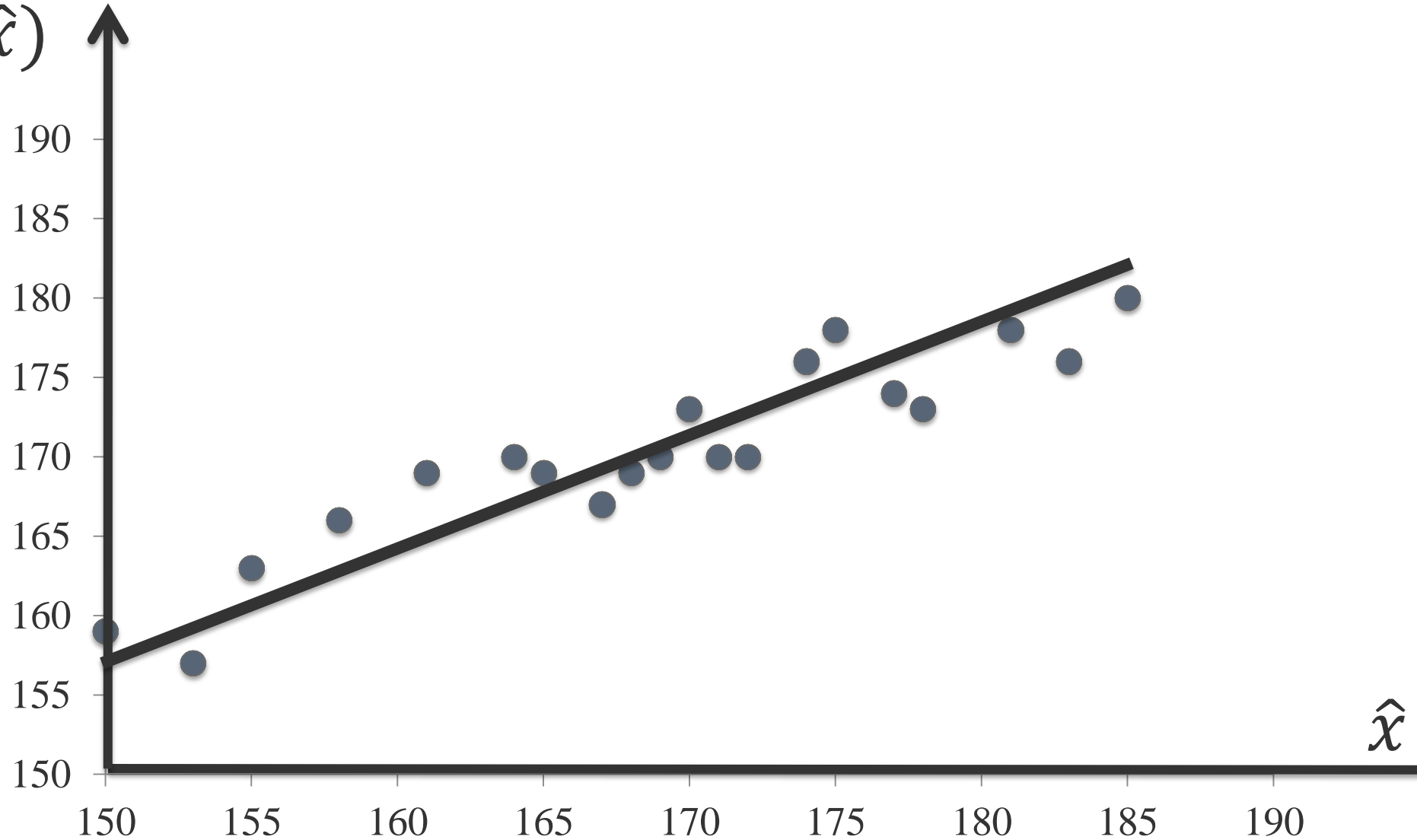




父亲身高 vs 孩子身高



$f(\hat{x})$



\hat{x}





什么是回归?

■ **回归**: 已知 $x = (\hat{x}, f(\hat{x}))$ 的 n 个观测值 $(\hat{x}_1, f(\hat{x}_1)), (\hat{x}_2, f(\hat{x}_2)), \dots, (\hat{x}_n, f(\hat{x}_n))$, 但不知 $(\hat{x}, f(\hat{x}))$, 这里 f 称为期望回归函数, 试求 $(\hat{x}, f(\hat{x}))$

注:

- 1) $(\hat{x}_k, f(\hat{x}_k))$ 表征一个样本, 其中 \hat{x}_k 即我们理解的特征, $f(\hat{x}_k)$ 即我们理解的标签;
- 2) \hat{x}_k 是观测值而不是估计值
- 3) $(\hat{x}, f(\hat{x}))$ 表征我们想学到的知识, 也就是 \hat{x} 与 $f(\hat{x})$ 的关系



什么是回归?

■ **回归**: 已知 $x = (\hat{x}, f(\hat{x}))$ 的 n 个观测值 $(\hat{x}_1, f(\hat{x}_1)), (\hat{x}_2, f(\hat{x}_2)), \dots, (\hat{x}_n, f(\hat{x}_n))$, 但不知 $(\hat{x}, f(\hat{x}))$, 这里 f 称为期望回归函数, 试求 $(\hat{x}, f(\hat{x}))$

$$\text{令 } X = \begin{bmatrix} \hat{x}_1 & f(\hat{x}_1) \\ \hat{x}_2 & f(\hat{x}_2) \\ \dots & \dots \\ \hat{x}_n & f(\hat{x}_n) \end{bmatrix}, \quad Y = \begin{bmatrix} \hat{x}_1 & F(\hat{x}_1) \\ \hat{x}_2 & F(\hat{x}_2) \\ \dots & \dots \\ \hat{x}_n & F(\hat{x}_n) \end{bmatrix}, \quad \underline{X} = (\hat{x}, f(\hat{x})), \underline{Y} = (\hat{x}, F(\hat{x}))$$

$$U = [1, 1, L, 1]_{1 \times N}^T, V = [1, 1, L, 1]_{1 \times N}^T$$

易知, 回归的输入可以表示为 $(X, U, \underline{X}, D_{S_X})$, 其输出可以表示为 $(Y, V, \underline{Y}, D_{S_Y})$ 。
因此, 回归也可以看作单类归类问题。



什么是回归?

类一致性准则:

好的结果应该是使得 (\vec{X}, \underline{X}) 和 (\vec{Y}, \underline{Y}) 之间具有最小误差的归类结果



贝叶斯估计

假设对 θ 得到估计 $\hat{\theta}$, 根据以上分析:

$$\underline{X} = \theta, \underline{Y} = \hat{\theta}$$

$$Sim_Y(x, \hat{\theta}) = p(x|\hat{\theta}), Sim(\hat{\theta}, \theta_0) = p(\hat{\theta}|\theta_0)$$

■ 根据 **类紧致准则** (希望最大类内相似度) :

$$\max_{\hat{\theta}} \prod_{k=1}^N Sim_Y(x_k, \hat{\theta}) = \max_{\hat{\theta}} \prod_{k=1}^N p(x_k|\hat{\theta}) \quad (3.1)$$

■ 根据 **类一致性准则**:

$$\max_{\hat{\theta}} Sim(\hat{\theta}, \theta_0) = \max_{\hat{\theta}} p(\hat{\theta}|\theta_0)$$

综合两准则, 应最大化目标函数:

$$Sim(\hat{\theta}, \theta_0) \prod_{k=1}^N Sim_Y(x_k, \hat{\theta}) = p(\hat{\theta}|\theta_0) \prod_{k=1}^N p(x_k|\hat{\theta}) \quad (3.11)$$



什么是回归?

类一致性准则:

好的结果应该是使得 (\vec{X}, \underline{X}) 和 (\vec{Y}, \underline{Y}) 之间具有最小误差的归类结果

根据类一致性准则, 一个好的类表示 \underline{Y} 应该最小化目标函数:

$$\min L = D(\underline{X}, \underline{Y}) = D(f(\hat{x}), F(\hat{x})) \quad (4.1)$$

注: $D(f(\hat{x}), F(\hat{x}))$ 的不同近似估计将导出不同的回归模型。

通常定义 $D(f(\hat{x}), F(\hat{x})) = \sum_{k=1}^N \mathbf{P} f(\hat{x}_k) - F(\hat{x}_k) \mathbf{P}^2$



目录

- 4.1 线性回归 (Linear Regression)**
- 4.2 岭回归 (Ridge Regression)**
- 4.3 Lasso回归 (Lasso Regression)**



4.1 线性回归

■ 当回归函数 F 采用线性模型表示时，此类模型为线性回归（Linear Regression）。

■ 一元线性方程（最简单的情况）：

$$F(\hat{x}) = \omega \hat{x} + b \quad (4.2)$$

根据类一致性准则，为了最小化 $D(f(X), F(X))$ ，常使用最小二乘的形式，所以，一元线性回归的损失函数为：

$$D(f(X), F(X)) = L(\omega, b) = \frac{1}{N} \sum_{k=1}^N (\omega \hat{x}_k + b - f(\hat{x}_k))^2 \quad (4.3)$$

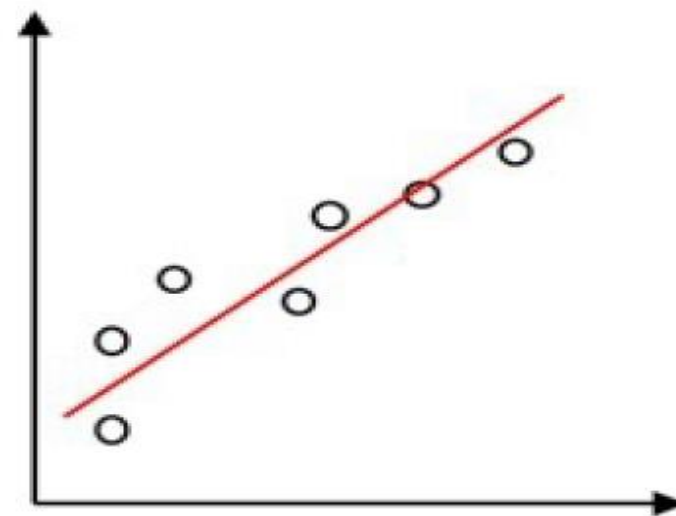


图4.1 一元线性回归示意图



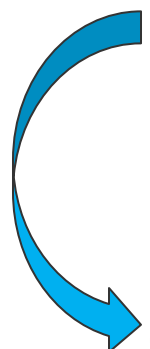
4.1 线性回归

因此，求解一元线性回归函数问题转化为一个优化问题：

$$\arg \min_{w,b} L(\omega, b) = \arg \min_{w,b} \frac{1}{2N} \sum_{k=1}^N (\omega \hat{x}_k + b - f(\hat{x}_k))^2 \quad (4.4)$$

为了最优化上述目标函数，对 b 和 ω 求偏导，令导数为零，即：

$$\frac{\partial L(\omega, b)}{\partial b} = 0, \frac{\partial L(\omega, b)}{\partial \omega} = 0 \quad (4.5)$$


$$\left\{ \begin{aligned} \omega &= \frac{\sum_{k=1}^N \hat{x}_k f(\hat{x}_k) - N\bar{x} \bar{f}}{\sum_{k=1}^N \hat{x}_k^2 - N\bar{x}^2} \\ b &= \bar{f} - \omega \bar{x} \end{aligned} \right. \quad (4.6)$$

$$\text{其中, } \bar{x} = \sum_{k=1}^N \frac{\hat{x}_k}{N}, \bar{f} = \sum_{k=1}^N \frac{f(\hat{x}_k)}{N}$$



4.1 线性回归

例4.1 假设我们试图对某一社区中个人的受教育程序（用 \hat{x} 表示）对年平均收入（用 $f(\hat{x})$ 表示）的影响进行研究。

教育年限	平均收入	教育年限	平均收入
\hat{x}	$f(\hat{x})$	\hat{x}	$f(\hat{x})$
6	5	16	13
10	7	5	5
9	6	10	10
9	6	12	12
16	9	8	10
12	18		



4.1 线性回归

教育年限	平均收入	教育年限	平均收入
\hat{x}	$f(\hat{x})$	\hat{x}	$f(\hat{x})$
6	5	16	13
10	7	5	5
9	6	10	10
9	6	12	12
16	9	8	10
12	18		

$$\left\{ \begin{aligned} \omega &= \frac{\sum_{k=1}^N \hat{x}_k f(\hat{x}_k) - N\bar{x} \bar{y}}{\sum_{k=1}^N \hat{x}_k^2 - N\bar{x}^2} \\ b &= \bar{f} - \omega \bar{x} \end{aligned} \right. \quad (4.6)$$

解：已知数据只有一个输入特征，所以设回归函数为 $y = \omega x_i + b$ 利用式 (4.6) , 计算各分量：

$$\bar{x} = (6 + 10 + 9 + 9 + 16 + 12 + 16 + 5 + 10 + 12 + 8) / 11 = 10.27$$

$$\bar{f} = (5 + 7 + 6 + 6 + 9 + 8 + 13 + 5 + 10 + 12 + 10) / 11 = 8.27$$

$$\sum_{k=1}^{11} \hat{x}_k f(\hat{x}_k) = 6 \times 5 + 10 \times 7 + ? \cdot + 8 \times 10 = 1005$$

$$\sum_{k=1}^{11} \hat{x}_k^2 = 6^2 + 10^2 + \dots + 8^2 = 1287$$



4.1 线性回归

$$\begin{aligned} \bar{x} &= 10.27 & \bar{f} &= 8.27 \\ \sum_{k=1}^{11} \hat{x}_k f(\hat{x}_k) &= 1005 & \sum_{k=1}^{11} \hat{x}_k^2 &= 1287 \end{aligned} \quad \left\{ \begin{aligned} \omega &= \frac{\sum_{k=1}^N \hat{x}_k f(\hat{x}_k) - N\bar{x}\bar{f}}{\sum_{k=1}^N \hat{x}_k^2 - N\bar{x}^2} \\ b &= \bar{f} - \omega\bar{x} \end{aligned} \right. \quad (4.6)$$

$$\omega = \frac{\sum_{k=1}^{11} \hat{x}_k f(\hat{x}_k) - 11\bar{x}\bar{f}}{\sum_{k=1}^{11} \hat{x}_k^2 - 11\bar{x}^2} = \frac{1005 - 11 \times 10.27 \times 8.27}{1287 - 11 \times 10.27^2} = \frac{70.74}{126.80} = 0.56$$

$$b = \bar{f} - \omega\bar{x} = 8.27 - 0.56 \times 10.27 = 2.52$$

因此，所求的线性回归方程为：

$$F(\hat{x}_k) = 0.56\hat{x}_k + 2.52$$



4.1 线性回归

■ 多元线性回归（更一般的情况）：

当输入数据有 p 个特征时，给定如下方程进行数据拟合

$$F(\hat{x}) = \omega^T \hat{x} + b \quad (4.10)$$

其中 \hat{x} 为输入的 p 维列向量， $\omega \in \mathbb{R}^p$ 为方程系数， b 为截距。

为了最小化 $D(f(X), F(X))$ ，最常用最小二乘的形式。对于 N 个样本，则给定误差平方为：

$$D(f(X), F(X)) = \sum_{k=1}^N (f(\hat{x}_k) - F(\hat{x}_k))^2 = \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k - b)^2 \quad (4.11)$$



4.1 线性回归

$$D(f(X), F(X)) = \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k - b)^2$$

为表示方便, 令

- A 为 $(p+1) \times N$ 的矩阵且第一行为全1的向量, 第二行至第 $p+1$ 行数据对应于训练数据的输入: $A = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$, 其中 \hat{x}_k 是 $p+1$ 维向量
- $B \in \mathbb{R}^N$ 为 N 个训练数据的输出: $B = [f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_N)]$
- $\omega_* = (b, \omega^T)^T \in \mathbb{R}^{p+1}$, 则式 (4.11) 可写成:
 $L(\omega_*) = (\omega_*^T A - B^T)(\omega_*^T A - B^T)^T = \omega_*^T A A^T \omega_* - 2B^T A^T \omega_* + B^T B$ (4.12)

$$\alpha^T \beta = \beta^T \alpha$$



4.1 线性回归

$$L(\omega_*) = (\omega_*^T A - B^T)(\omega_*^T A - B^T)^T = \omega_*^T A A^T \omega_* - 2B^T A^T \omega_* + B^T B$$

最小化上式求解 ω_* 就是对 ω 求偏导，有

$$\frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B} + \mathbf{B}^T) \mathbf{x}$$

$$\frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}$$

$$\frac{\partial L(\omega_*)}{\partial \omega_*} = 2A A^T \omega_* - 2A B = 0$$

若A为行满秩矩阵，有：

$$\omega_* = (A A^T)^{-1} A B$$



4.1 线性回归

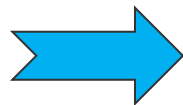
- 当输出为d个一元变量，其线性模型为：

$$B = W^T A \quad (4.15)$$

其中 $B \in \mathbb{R}^{d \times N}$ 为输出矩阵， $A \in \mathbb{R}^{(p+1) \times N}$ 为输入矩阵，并且其第一行为全1， $W \in \mathbb{R}^{(p+1) \times d}$ 为系数矩阵。

为了最小化 $D(f(X), F(X))$ ，类似有：

$$\begin{aligned} & D(f(X), F(X)) \\ &= \sum_{k=1}^N \mathbf{P} f(\hat{x}_k) - F(\hat{x}_k) \mathbf{P}^2 \\ &= \sum_{k=1}^N \mathbf{P} f(\hat{x}_k) - \mathbf{W}^T (1, \hat{x}_k^T)^T \mathbf{P}^2 \\ &= \text{trace}[(B - W^T A)^T (B - W^T A)] \end{aligned}$$



$$W = (AA^T)^{-1} AB^T$$



4.1 线性回归

$$\begin{aligned} D(f(X), F(X)) \\ &= \text{trace}[(B - W^T A)^T (B - W^T A)] \\ &= \text{trace}(B^T B) - \text{trace}(B^T W^T A) - \text{trace}(A^T W B) + \text{trace}(A^T W W^T A) \end{aligned}$$

D 取得极值, 需 $\frac{\partial D}{\partial W} = 0$

$$\frac{\partial \text{trace}(B^T W^T A)}{\partial W} = AB^T$$

$$\frac{\partial \text{trace}(A^T W B)}{\partial W} = AB^T$$

$$\frac{\partial \text{trace}(A^T W W^T A)}{\partial W} = 2AA^T W$$

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A} \mathbf{X}^T \mathbf{B}) = \mathbf{B} \mathbf{A}$$

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A} \mathbf{X} \mathbf{B}) = \mathbf{A}^T \mathbf{B}^T$$

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{A} \mathbf{X} \mathbf{B} \mathbf{X}^T \mathbf{C}) = \mathbf{A}^T \mathbf{C}^T \mathbf{X} \mathbf{B}^T + \mathbf{C} \mathbf{A} \mathbf{X} \mathbf{B}$$

$$\frac{\partial D}{\partial W} = -2AB^T + 2AA^T W \xrightarrow{P < N, AA^T \text{ 满秩}} W = (AA^T)^{-1} AB^T$$



4.1 线性回归

$$W = (AA^T)^{-1} AB^T$$

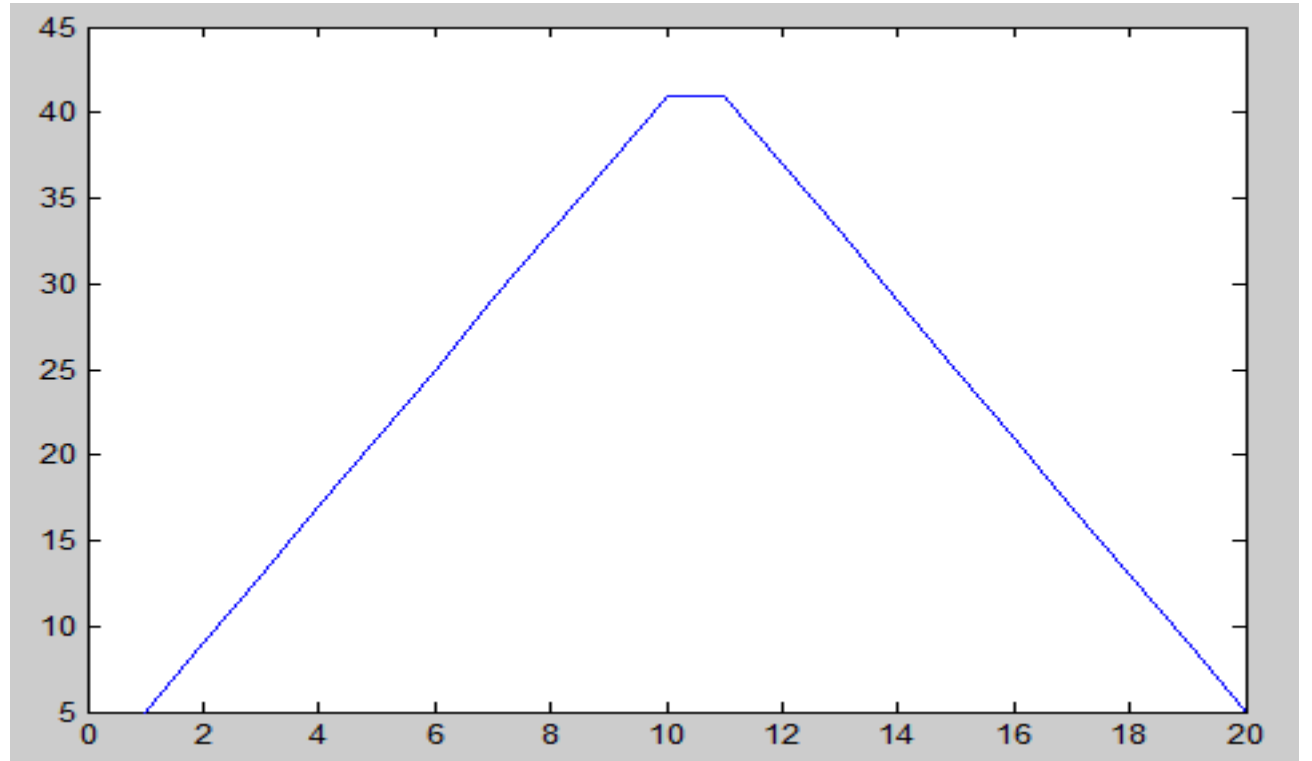


4.1 线性回归

```
X1 = [1:10, 10:-1:1]';  
X2 = ones(20, 1);  
X3 = rand(20, 1);  
X = [X1, X2, X3];  
Y = X * [4, 1, 0]';  
W1 = inv(X' * X) * X' * Y;  
W1'
```

ans =

4.0000 1.0000 0.0000





4.1 线性回归

```
X4 = rand(20, 100);
X = [X1, X2, X4];
Y = X * [4, 1, zeros(1, 100)]';
W2 = inv(X' * X) * X' * Y;
W2'
```

```
X4(20, 100) = 0.9;
X = [X1, X2, X4];
Y = X * [4, 1, zeros(1, 100)]';
W3 = inv(X' * X) * X' * Y;
W3'
```

```
ans =

1.0e+17 *

Columns 1 through 20

-0.0000    0.4885   -2.1196   -0.5741    0.3817   -0.0148    1.1135    1.8039    0.8979

Columns 21 through 40

-2.8830    2.2279   -2.3952    2.2295    0.8181    0.6429    1.8535    1.4251    4.9337

Columns 41 through 60

2.4869   -1.7615    1.4645    0.6019    3.8057    0.3477   -0.5004   -1.4324   -5.4474

Columns 61 through 80

0.0937    1.7467   -6.2112   -2.1706   -0.3795   -2.3239    0.6883    1.7815   -0.6692

Columns 81 through 100

2.8888   -2.9582    0.0338   -0.4906   -0.0877   -0.7078    0.8262    2.5318   -5.2409

Columns 101 through 102

1.5540    0.0000
```

```
ans =

1.0e+18 *

Columns 1 through 20

-0.0000   -0.0648   -0.2047   -0.4701   -0.0916   -0.3206    0.1929    0.4460

Columns 21 through 40

-0.3202    0.2809   -0.4607    0.3171   -0.1355   -0.0505    0.3851   -0.1895

Columns 41 through 60

0.1079   -0.2841   -0.2224    0.2829    0.9074    0.2326   -0.3556    0.6246

Columns 61 through 80

0.3214   -0.3406    0.0653   -0.3658    0.0857   -0.0374   -0.0029    0.0553

Columns 81 through 100

0.1699   -0.2826    0.2488    0.0500   -0.2393    0.1447   -0.0972   -0.3660

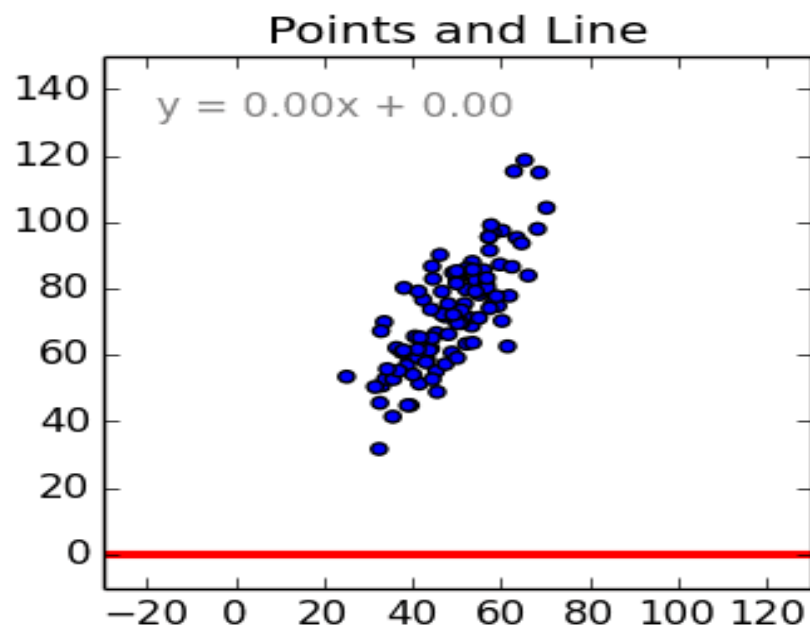
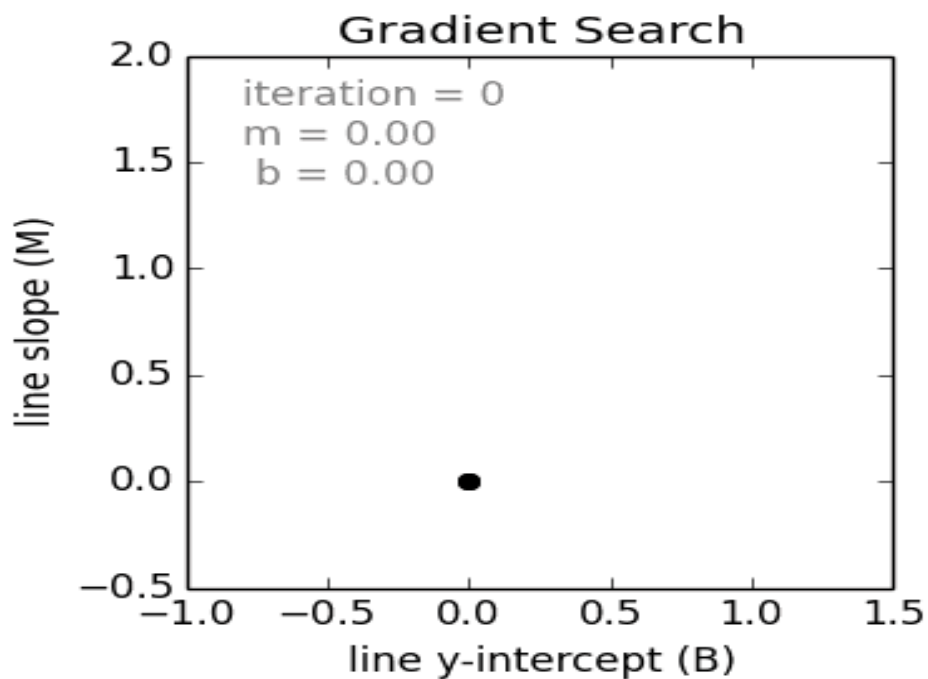
Columns 101 through 102

-0.1143    0.0000
```



线性回归特点

- 建模快速简单，特别适用于要建模的关系不是非常复杂且数据量不大的情况。
- 有直观的理解和解释。
- 线性回归对异常值非常敏感。





4.2 岭回归

对数据 X 均值归零后，类表示为 $(\hat{x}, F(\hat{x}) = (\hat{x}, \omega^T \hat{x})$ 。如果类表示的复杂度定义为 $\|\omega\|^2$ ，则奥卡姆剃刀准则要求选取具有最小范数的可行解。

■ 岭回归(Ridge Regression)

$$\left\{ \begin{array}{ll} \min_{\omega} = \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k)^2 & \text{类一致性准则} \\ \min \|\omega\|^2 & \text{奥卡姆剃刀准则} \end{array} \right.$$



4.2 岭回归

$$\left\{ \begin{array}{l} \min_{\omega} = \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k)^2 \\ \min \mathbf{P} \omega \mathbf{P}^2 \end{array} \right. \quad (4.18)$$

综合考虑 (4.18) , 目标函数可转化为:

$$\min_{\omega} \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k)^2 + \lambda \mathbf{P} \omega \mathbf{P}_2^2 \quad (4.19)$$

其中, 正则化参数 $\lambda \geq 0$, 用来控制收缩程度; λ 越大, 收缩程度越大;
 $\lambda=0$ 时岭回归退化为原始线性回归。



4.2 岭回归

$$\begin{aligned} \min_{\omega} \sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k)^2 + \lambda \mathbf{P} \omega \mathbf{P}_2^2 \\ = \min \left(\text{trace}[(B - W^T X)^T (B - W^T X)] + \lambda W^T W \right) \end{aligned}$$

记 $\hat{X} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N)$, 针对 ω 求偏导置0, 得到:

$$\frac{\partial D}{\partial W} = -2XB^T + 2XX^TW$$

$$\frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{B} + \mathbf{B}^T) \mathbf{x}$$

$$-2XB^T + 2XX^TW + 2\lambda W = 0$$

$$\omega^{ridge} = (\hat{X}\hat{X}^T + \lambda I)^{-1} \hat{X}B \quad (4.20)$$

- 其中, $I \in \mathbb{R}^{p \times p}$ 是单位阵。
- 即使 $\hat{X}\hat{X}^T$ 不是满秩的, 加上 λI 也可组成非奇异矩阵。



4.2 岭回归

λ 是超参数, 人为设定

$$\omega^{ridge} = (\hat{X}\hat{X}^T + \lambda I)^{-1} \hat{X}B$$

$$W = (AA^T)^{-1} AB^T$$



4.2 岭回归

```
>> W_Ridge'
```

```
ans =
```

```
Columns 1 through 20
```

```
3.8520    1.0991    0.5902    0.0108    0.0083    0.0063   -0.0013    0.0076    0.0096
```

```
Columns 21 through 40
```

```
0.0085    0.0112   -0.0064    0.0160   -0.0109   -0.0009   -0.0066    0.0051   -0.0103
```

```
Columns 41 through 60
```

```
0.0081   -0.0128    0.0059    0.0020    0.0063   -0.0061    0.0088   -0.0079    0.0186
```

```
Columns 61 through 80
```

```
-0.0045   -0.0059   -0.0055   -0.0053    0.0057   -0.0025   -0.0053    0.0134    0.0097
```



4.2 岭回归优缺点

- 优点：Ridge回归在不抛弃任何一个变量的情况下，缩小了回归系数，使得模型相对而言比较的稳定。
- 缺点：但这会使得模型的变量特别多，模型解释性差。容易导致过拟合。



4.3 Lasso回归

■ Lasso回归的提出

- 岭回归使用 $\|\omega\|_2^2$ 计算类表示的复杂度，对系数进行整体收缩；
- 当变量个数很多时，关心哪些变量或特征与回归目标最相关（解释性）；
- 此时，使用系数中非零值的个数来计算类表示的复杂度更合理；
- 为减少计算量，通常使用系数 $\sum_{j=1}^p |\omega_j|$ 代替系数非零值的个数。

类表示复杂度定义为： $\|\omega\|_1 = \sum_{j=1}^p |\omega_j|$

■ Lasso回归目标函数：

类一致性准则

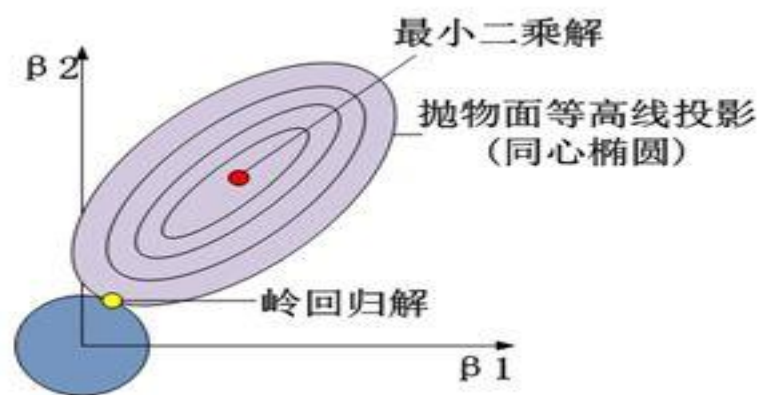
$$\min_{\omega} \left[\sum_{k=1}^N (f(\hat{x}_k) - \omega^T \hat{x}_k)^2 \right] + \lambda \sum_{j=1}^p |\omega_j| \quad (4.21)$$

奥卡姆剃刀准则

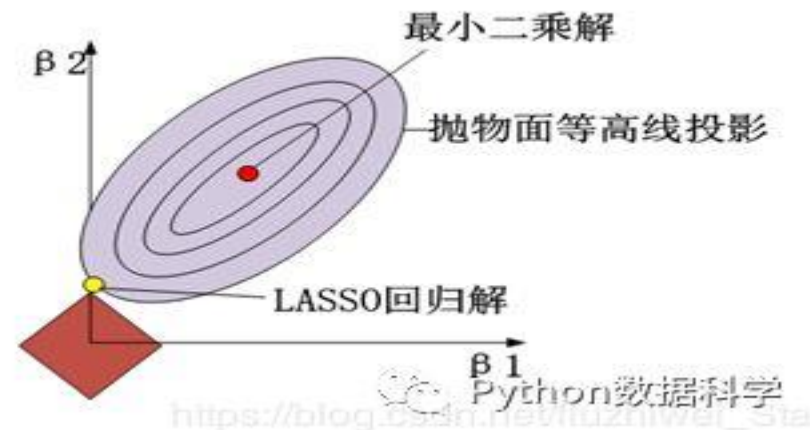


岭回归和Lasso回归收缩的差异

- 求解的交点不再是最小二乘的最小值（红点），而变成了与正则项的交点（黄点）。
- 岭回归中两个图形（没有棱角）的交点永远不会落在两个轴上
- LASSO回归中，正则化的几何图形是有棱角的，可以很好的让交点落在某一个轴上。这种稀疏化的不同也导致了LASSO回归可以用于特征选择（让特征权重变为0从而筛选掉特征），而岭回归却不行。



岭回归可行域 $\omega_1^2 + \omega_2^2 < t$ （圆形）



Lasso回归的可行域 $|\omega_1| + |\omega_2| < t$ （菱形）

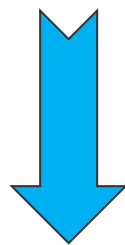


4.3 Lasso回归

式 (4.21) 的矩阵形式:

$$\min_{\omega} \mathbf{P} \omega^T \hat{\mathbf{X}} - \mathbf{B} \mathbf{P}^2 + \lambda \mathbf{P} \omega \mathbf{P}_1 \quad (4.22)$$

注: 求解问题 (4.22) 的难点在于 l_1 -范数在0点不可导。



解决方案

1. 坐标轴下降法
2. 快速迭代收缩阈值 (FIST) 算法



4.3 Lasso回归

用坐标下降法求解lasso回归

- 坐标轴下降法是沿着坐标轴的方向去下降，这和梯度下降不同。梯度下降是沿着梯度的负方向下降。不过梯度下降和坐标轴下降的共性就都是迭代法。
- 坐标轴下降法的数学依据：一个可微的凸函数 $J(\theta)$ ，其中 θ 是 $n \times 1$ 的向量，即有 n 个维度。如果在某一点 θ ，使得 $J(\theta)$ 在每一个坐标轴 $\theta_i (i = 1, 2, \dots, n)$ 上都是最小值，那么 $J(\theta_i)$ 就是一个全局的最小值。



4.3 Lasso回归

Coordinate descent

This suggests that for $f(x) = g(x) + \sum_{i=1}^n h_i(x_i)$ (with g convex, differentiable and each h_i convex) we can use **coordinate descent** to find a minimizer: start with some initial guess $x^{(0)}$, and repeat for $k = 1, 2, 3, \dots$

$$x_1^{(k)} \in \operatorname{argmin}_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \dots, x_n^{(k-1)})$$

$$x_2^{(k)} \in \operatorname{argmin}_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, \dots, x_n^{(k-1)})$$

$$x_3^{(k)} \in \operatorname{argmin}_{x_3} f(x_1^{(k)}, x_2^{(k)}, x_3, \dots, x_n^{(k-1)})$$

...

$$x_n^{(k)} \in \operatorname{argmin}_{x_n} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_n)$$

Note: after we solve for $x_i^{(k)}$, we use its new value from then on



4.3 Lasso回归

Linear regression

Let $f(x) = \frac{1}{2} \|y - Ax\|^2$, where $y \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times p}$ with columns A_1, \dots, A_p

Consider minimizing over x_i , with all x_j , $j \neq i$ fixed:

$$0 = \nabla_i f(x) = A_i^T (Ax - y) = A_i^T (A_i x_i + A_{-i} x_{-i} - y)$$

i.e., we take

$$x_i = \frac{A_i^T (y - A_{-i} x_{-i})}{A_i^T A_i}$$

$$W = (X^T X)^{-1} X^T Y$$

Coordinate descent repeats this update for $i = 1, 2, \dots, p, 1, 2, \dots$



4.3 Lasso回归

Lasso regression

Consider the lasso problem

$$f(x) = \frac{1}{2} \|y - Ax\|^2 + \lambda \|x\|_1$$

Note that the non-smooth part is separable: $\|x\|_1 = \sum_{i=1}^p |x_i|$

Minimizing over x_i , with $x_j, j \neq i$ fixed:

$$0 = A_i^T A_i x_i + A_i^T (A_{-i} x_{-i} - y) + \lambda s_i$$

where $s_i \in \partial|x_i|$. Solution is given by soft-thresholding

$$x_i = S_{\lambda/\|A_i\|^2} \left(\frac{A_i^T (y - A_{-i} x_{-i})}{A_i^T A_i} \right)$$

Repeat this for $i = 1, 2, \dots, p, 1, 2, \dots$

$$\text{soft}(x, T) = \begin{cases} x + T & x \leq -T, \\ 0 & |x| \leq T, \\ x - T & x \geq T. \end{cases}$$



4.3 Lasso回归

伪代码

input: $X(m \times n)$; $Y(m \times 1)$; $J(\theta) = \frac{1}{2n} (X\theta - Y)^T (X\theta - Y) + \alpha \|\theta\|_1$; K ; ε .

output : θ

随机产生 θ 初值

For k form 1 to K (最大迭代次数)

For i form 1 to n

$$\theta_i^{(k)} = \underset{\theta_i}{\operatorname{argmin}} J(\theta_1^{(k)}, \theta_2^{(k)} \dots, \theta_i, \theta_{i+1}^{(k-1)} \dots \theta_n^{(k-1)})$$

end

if $J(\theta^k) - J(\theta^{(k-1)}) < \varepsilon$

return θ^k

else

return 第2步

end

end



4.3 Lasso回归

其中的 $\theta_i^{(k)} = \underbrace{\arg \min}_{\theta_i} J(\theta_1^{(k)}, \theta_2^{(k)} \dots, \theta_i, \theta_{i+1}^{(k-1)} \dots \theta_n^{(k-1)})$ 怎么做?

上一轮中的 $\theta_i^{(k-1)}$ 如果:

- 为0, 则梯度不存在, 本轮为零

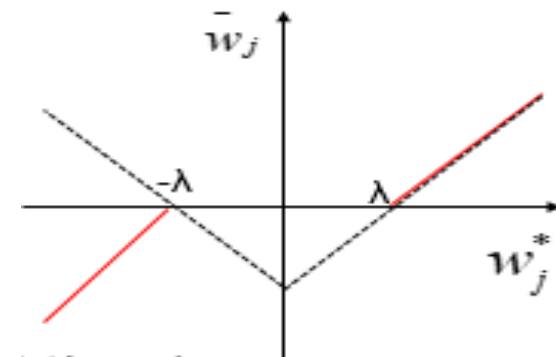
- 不为0, 则:

1. 基于 $[Y, X, \text{其它 } n-1 \text{ 个 } \theta]$ 估算最优 θ_j^* (即前面 w_j^*)

2. 利用 $\bar{w}_j = \text{sgn}(w_j^*) (|w_j^*| - \lambda)_+$ 计算本轮的 $\theta_j^{(k)}$ (即等号左边项), 这样的计算结果有两种可能:

>> $|w_j^*| > \lambda$, 此时得到的 $\theta_j^{(k)}$ **非零**, 下一轮计算 $\theta_j^{(k+1)}$ 时走步骤2

>> $|w_j^*| \leq \lambda$, 此时得到的 $\theta_j^{(k)}$ **为零**, 下一轮计算 $\theta_j^{(k+1)}$ 时走步骤1





4.3 Lasso回归

```
function W = I4_3_Descent_Coordinate(X, Y, Lambda)
    SampleNum = size(X, 1);
    FeatNum = size(X, 2);
    W = ones (FeatNum, 1);

    Cost_Old = 1e20;
    Cost_New = sum((Y - X*W) .^ 2) + Lambda * sum(abs(W));

    K = 5000; k = 1;
    while (k <= K && abs(Cost_Old - Cost_New) > 0.01)
        k
        Cost_Old = Cost_New;

        No = find(W);
        for j = 1 : length(No) ...

            Cost_New = sum((Y - X*W) .^ 2) + Lambda * sum(abs(W))
            k = k+1;
            [W(1:10)^', length(find(W))]'
    end
```



4.3 Lasso回归

```
No = find(W);  
for j = 1 : length>No)  
    OtherNo = find>No(j) ~= [1:FeatNum]);  
    Wstar = X(:, No(j))' * (Y - X(:, OtherNo) * W(OtherNo)) / (X(:, No(j))' * X(:, No(j)));  
  
    LambdaP = Lambda / (X(:, No(j))' * X(:, No(j)));  
    if (Wstar > LambdaP)  
        W>No(j)) = Wstar - LambdaP;  
    elseif (Wstar < -LambdaP)  
        W>No(j)) = Wstar + LambdaP;  
    else  
        W>No(j)) = 0;  
    end  
end
```

$$x_i = S_{\lambda/\|A_i\|^2} \left(\frac{A_i^T (y - A_{-i} x_{-i})}{A_i^T A_i} \right)$$

$$\text{soft}(x, T) = \begin{cases} x + T & x \leq -T, \\ 0 & |x| \leq T, \\ x - T & x \geq T. \end{cases}$$



4.3 Lasso回归

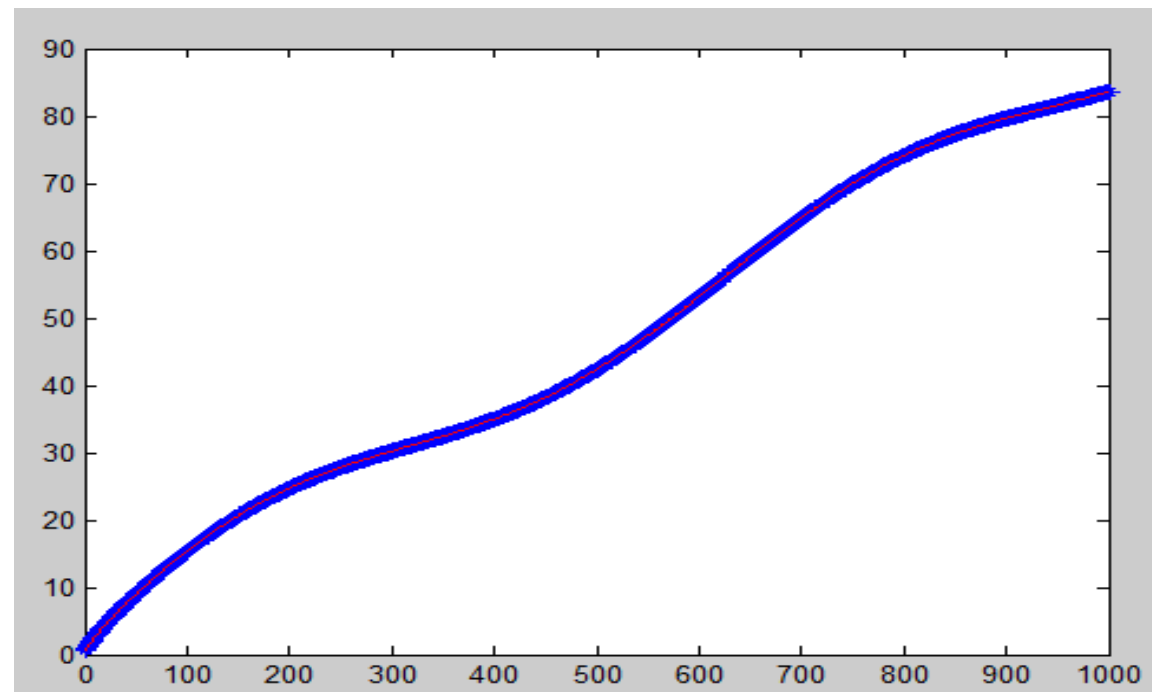
```
clear;clc;
```

```
Xp = rand(1000, 300);  
Xp(:, 1) = sin([0.01:0.01:10]');  
Xp(:, 2) = [0.01:0.01:10]';  
Xp(:, 3) = sqrt([0.01:0.01:10]');  
Yp = Xp * [4, 7, 5, zeros(1, 297)]';
```

```
Lambda = 10;  
W = I4_3_Descent_Coordinate(Xp, Yp, Lambda);  
[W(find(W))';find(W)']
```

```
Lambda = 5;  
W = I4_3_Descent_Coordinate(Xp, Yp, Lambda);  
[W(find(W))';find(W)']
```

```
Lambda = 1;  
W = I4_3_Descent_Coordinate(Xp, Yp, Lambda);  
[W(find(W))';find(W)']
```





4.3 Lasso回归

Cost_New =

160.1646

Lambda = 10;

ans =

3.9805	7.0180	4.9538
1.0000	2.0000	3.0000

Cost_New =

80.1397

Lambda = 5;

ans =

3.9905	7.0134	4.9658
1.0000	2.0000	3.0000

Cost_New =

16.0256

Lambda = 1;

ans =

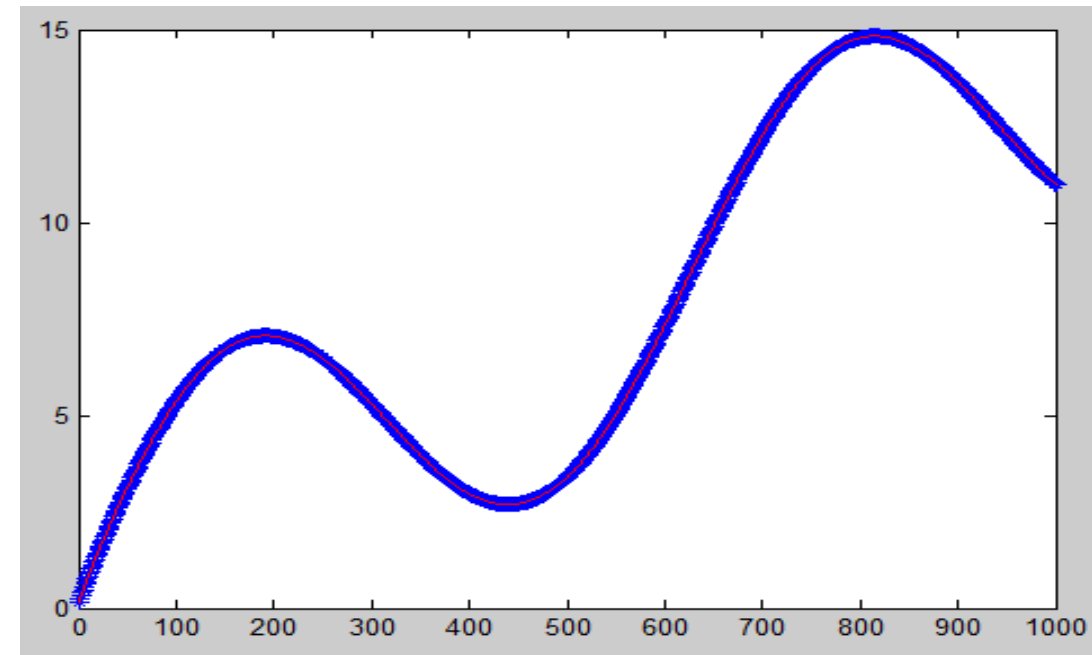
3.9975	7.0037	4.9861	0.0038	0.0089	0.0044
1.0000	2.0000	3.0000	52.0000	77.0000	280.0000

对W的约束作用
随 λ 的减小
而减弱



4.3 Lasso回归

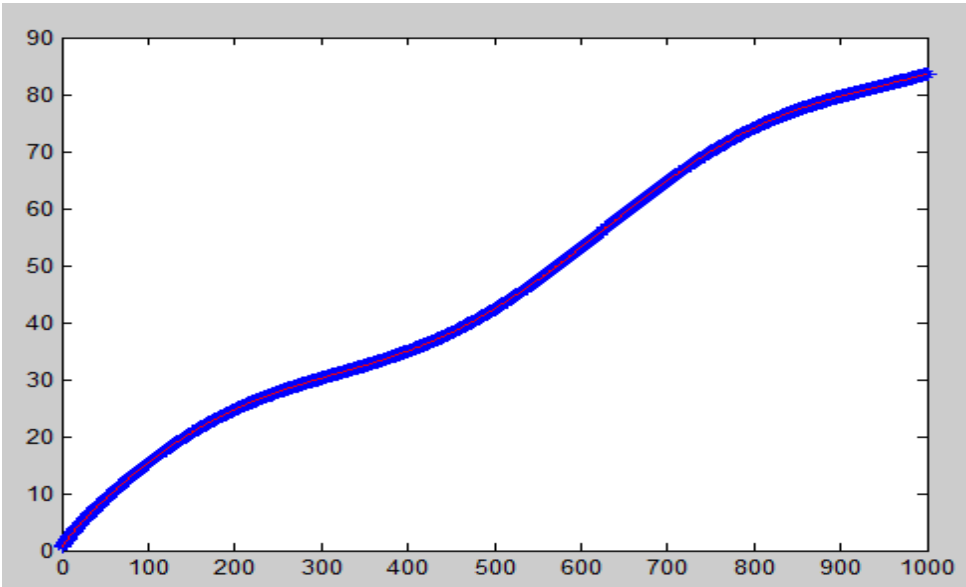
```
Xp = rand(1000, 300) ;  
Xp(:, 1) = sin([0.01:0.01:10]^2) ;  
Xp(:, 2) = [0.01:0.01:10]^2 ;  
Xp(:, 3) = sqrt([0.01:0.01:10]^2) ;  
Yp = Xp * [4, 1, 1, zeros(1, 297)]^2 ;
```





4.3 Lasso回归

超参数 λ 的选择
与数据有关



```
Cost_New =
```

```
30.7717
```

```
ans =
```

```
3.9912
```

```
1.0280
```

```
0.9182
```

```
0.0108
```

```
0.0276
```

```
1.0000
```

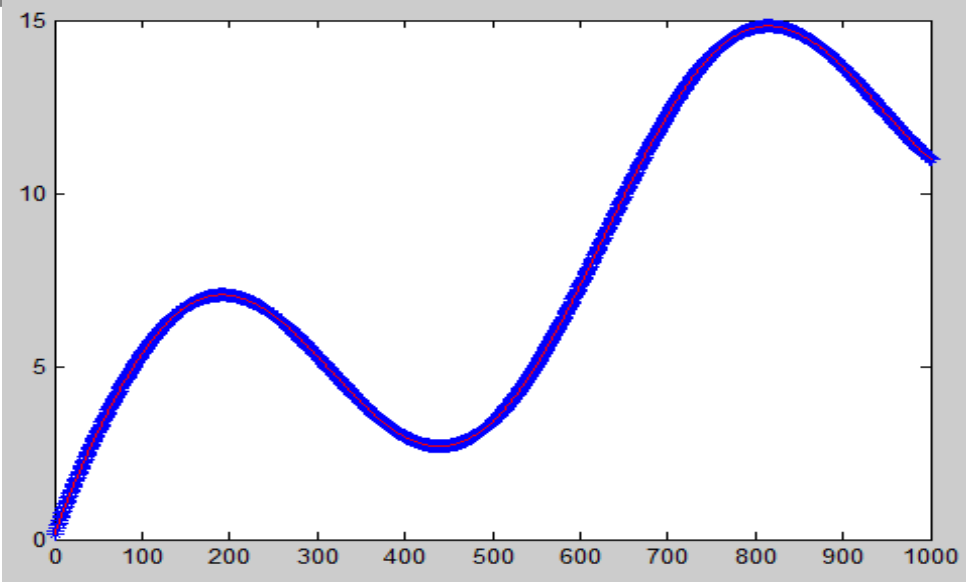
```
2.0000
```

```
3.0000
```

```
168.0000
```

```
208.0000
```

Lambda = 5;



```
Cost_New =
```

```
80.1397
```

Lambda = 5;

```
ans =
```

```
3.9905
```

```
7.0134
```

```
4.9658
```

```
1.0000
```

```
2.0000
```

```
3.0000
```



4.3 Lasso回归

快速迭代收缩阈值 (FIST) 算法

预备知识

该算法用于求解型如下式的目标函数：

$$\min_x F(x) = \min_x f(x) + g(x) \quad (4.23)$$

- ◆ $g(x)$ 为连续凸函数，可不光滑；
- ◆ $f(x)$ 为光滑凸函数，其导数应Lipschitz连续，即存在常数 $L(f) > 0$ ，满足：

$$\| \nabla f(x) - \nabla f(z) \| \leq L(f) \| x - z \| \quad (\forall x, z) \quad (4.24)$$

可证明：

$$f(x) \leq f(z) + \langle \nabla f(z), x - z \rangle + \frac{L(f)}{2} \| x - z \|^2 \quad (4.25)$$

➤ 不是求 $f(x)$ 的最小值，而是逼近其上限

➤ 将 $f(x)$ 在 $w(t)$ 处展开



4.3 Lasso回归

算法4.1 通过FIST求解问题

输入：数据矩阵A

1. 初始化 $a^{(1)} = \omega^{(0)} = (0, 0, \dots, 0)^T, m^{(1)} = 1, \epsilon = 10^{-3}, T = 100$ (T代表最大迭代次数)

2. while (t < T) do

$$\omega^{(t)} = S_{\frac{\lambda}{L}}(a^{(t)} - \frac{1}{L} \nabla f(a^{(t)}))$$

$$m^{(t+1)} = \frac{1 + \sqrt{1 + 4(m^{(t)})^2}}{2}$$

$$a^{(t+1)} = \omega^{(t)} + \frac{m^{(t)} - 1}{m^{(t+1)}} (\omega^t - \omega^{(t-1)})$$

3. 如果 $\|\omega^{(t)} - \omega^{(t-1)}\|_{\infty} > \epsilon$, $t \leftarrow t + 1$ 。否则, $\omega \leftarrow \omega^{(t)}$, 结束程序

4. end while



4.3 Lasso回归优缺点

- 优点：Lasso相比于岭回归的优势在于压缩变量表现更出色。
- Lasso能选出那些很有价值的预测变量，选出更精确的模型。
- 缺点：Lasso算法仅保留其中之一部分而损失其余同等重要的特征信息。
- 在Lasso回归求解路径中，对于 $N \times p$ 的设计矩阵来说，最多只能选出 $\min(N, p)$ 个变量。
 - 当 $p > N$ 的时候，最多只能选出 N 个预测变量。
 - 对于 $p \sim N$ 的情况，Lasso方法不能够很好地选出真实的模型。如果预测变量具有群组效应，则用Lasso回归时，只能选出其中的一个预测变量。
 - 对于通常的 $N > p$ 的情形，如果预测变量中存在很强的共线性，Lasso的预测表现受控于岭回归。



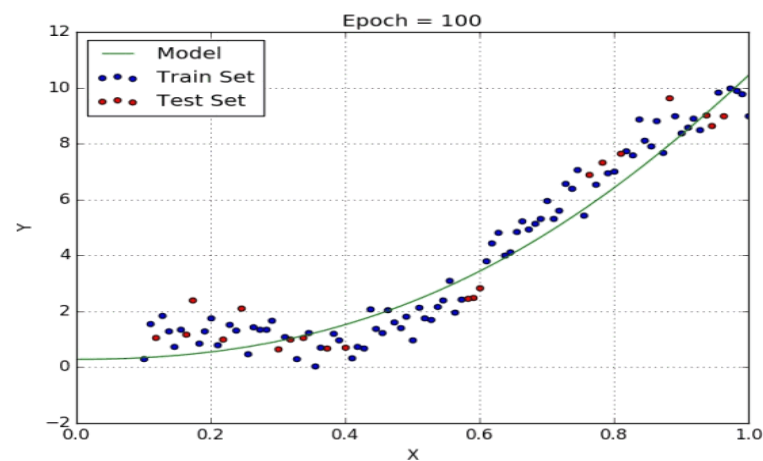
用最小角回归法求解Lasso回归

- 最小角回归法 (Least Angle Regression, LARS)
- 起源于逐步回归 (stepwise regression)
 - 提出者认为如果把回归分析的自变量理解成向量从空间中从原点出发逐步到达应变量的过程, 那么逐步回归本质上都是沿最小二乘的方向逐步前进的过程, 认为逐步回归的步子迈的太大, 有可能漏掉一些关键变量, 于是提出了最小角回归。
 - LARS算法加速了计算过程, 只需 m 步 (m 是自变量的个数) 得到参数的估计值。
 - Bradley Efron et al., Least angle regression. The Annals of Statistics, **2004**, 32(2):407-451.



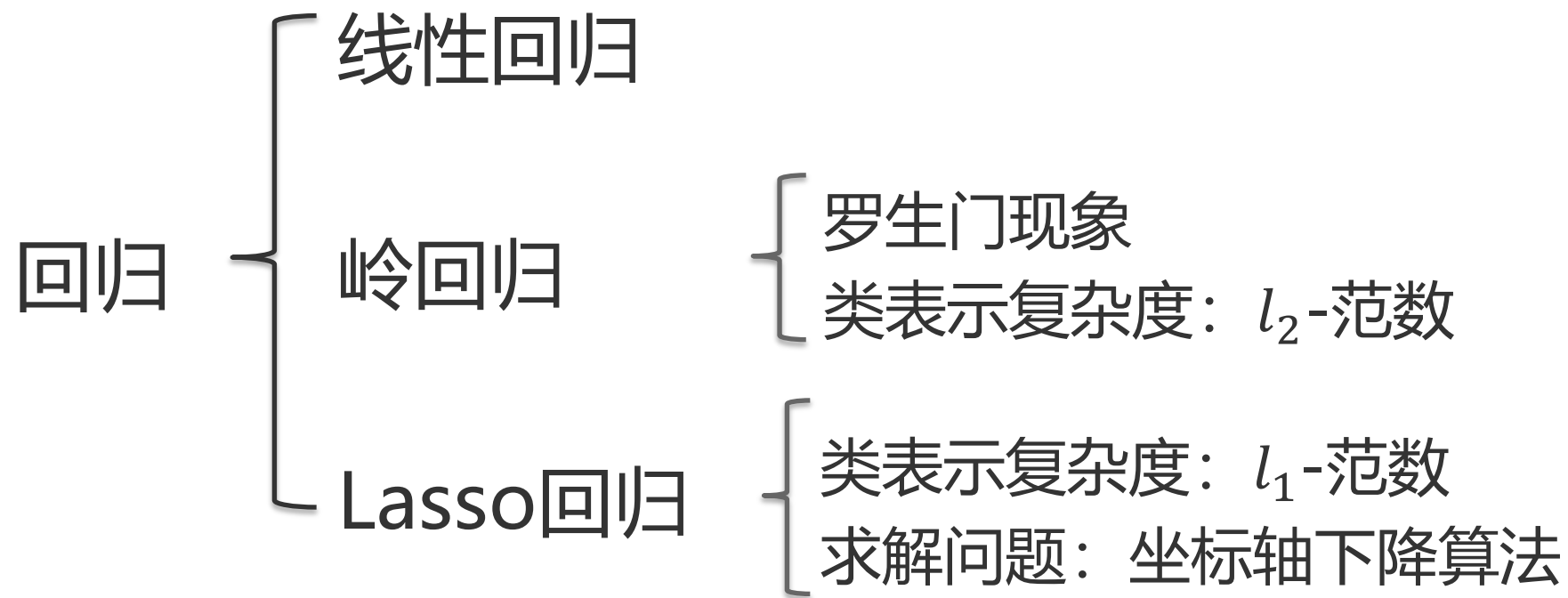
多项式回归特点

- 能够模拟非线性可分的数据;线性回归不能做到这一点。它总体上更灵活, 可以模拟一些相当复杂的关系。
- 完全控制要素变量的建模 (要设置变量的指数)。
- 需要仔细的设计。需要一些数据的先验知识才能选择最佳指数。
- 如果指数选择不当, 容易过拟合。





本章小结





弹性网(Elastic net)回归优势

- Elastic Net基于Lasso进行了有效改进：结合L1、L2范数对模型系数进行约束。
- 与Lasso算法相比，Elastic Net能够将同等重要的特征信息尽可能保留，避免了Lasso算法仅保留其中之一部分而损失其余同等重要的特征信息。



弹性网(Elastic net)回归

➤ Elastic Net损失函数模型

$$\arg \min_w \sum_{n=1}^N (\mathbf{y}^n - \sum_{p=1}^m w_p \mathbf{x}_p^n)^2 + \lambda \sum_{p=1}^m (\alpha |w_p| + \frac{1}{2} (1 - \alpha) (w_p)^2)$$

\mathbf{x}_p^n 代表第 n 个样本的第 p 个特征， \mathbf{y}^n 表示第 n 个样本的类标签，

N 表示样本数目， w_p 为第 p 个特征的回归系数， λ 是控制模型稀

疏度的正则化参数， 正则化参数 α 平衡 l_1 范数和 l_2 范数惩罚项。



回归扩展

➤ 偏最小二乘回归

- 偏最小二乘法可以同时实现回归建模（多元线性回归、数据结构简化（主成分分析）以及两组变量之间的相关性分析（典型相关分析）。
- 不仅可以克服共线性问题，它在选取特征向量时强调自变量对因变量的解释和预测作用，去除了对回归无益噪声的影响，使模型包含最少的变量数。
- 1983年由S.Wold和C.Albano等人首次提出。



回归扩展

➤ 其它回归

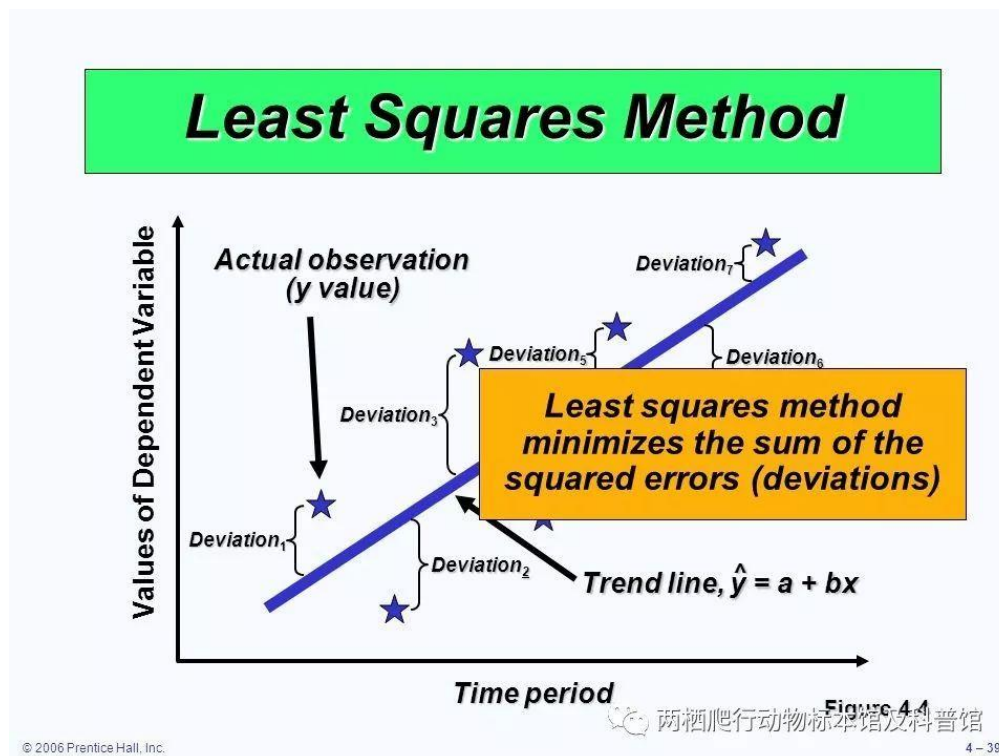
- K近邻回归、支持向量机回归
- 决策树回归
- 贝叶斯回归
- RANSAC随机抽样一致性算法 (1981)
- 随机森林回归、Bagging, Aaboost (adaptive boosting)
- 梯度提升决策树 (GBDT) 是基于Boosting思想的融合算法。



回归历史

➤ 最小二乘法

- 最小二乘法（Least Squares Estimation, 简记为LSE），又称最小平方法，是一种数学优化技术，通过**最小化误差的平方和**寻找数据的**最佳函数匹配**。利用最小二乘法可以简便地**求得未知的数据**，并使得这些求得的数据与实际数据之间误差的平方和为最小。





回归历史

➤ 最小二乘法历史

- 最小二乘法最开始是从天文学和地理测量学领域发展起来的。当时科学家和数学家要为探索时代的地球海洋挑战提供解决方案。
- 1805年，勒让德A. M. Legendre的《计算彗星轨道的新方法》出版，发表了最小二乘法的第一个清晰和简明的阐述。
- 1808年，美国人罗伯特·阿德雷恩（Robert Adrain）也独立制定了最小二乘分析的思想。这也让他开始受到学界重视。
- 1809年，高斯Carl Friedrich Gauss著作《关于绕日行星运动的理论》出版。在此书中声称他自1799年以来就使用最小二乘方法，由此爆发了一场与勒让德的优先权之争。
- 1829年，高斯提供了最小二乘法的优化效果强于其他方法的证明，因此被称为高斯-莫卡夫定理。



回归历史

➤ 最小二乘法历史

- 近代学者经过对原始文献的研究，前两人可能独立发明了这个方法。但首先见于书面形式的，还是**勒让德**最早。
- 可现今教科书和著作中，多把这个发明权归功于高斯。主要原因是此时高斯有更大的名气。另一个主要原因可能是因为高斯的**正态误差理论**对这个方法有非常重要的意义。
 - ✓ 勒让德对最小二乘方法的优点有所阐述，但缺少误差分析，主要影响在天文估算。
 - ✓ 阿德雷恩将它应用于实际测量。
 - ✓ 高斯的作用主要在于他提出的正态误差理论，成功地将最小二乘法与概率原理和正态分布联系起来。
- 近两个多世纪，误差理论和统计学中的研究者发现了许多实现最小二乘的不同方法。



回归历史

➤ 回归概念的提出

- “**回归**”是由英国著名生物学家兼统计学家高尔顿(Francis Galton, 1822 ~ 1911) 在研究人类遗传问题时提出来的。
- **1855年**，高尔顿发表《遗传的身高向平均数方向的回归》一文
- 他和他的学生卡尔·皮尔逊Karl·Pearson通过观察1078对夫妇的身高数据，以每对夫妇的平均身高作为自变量，取他们的一个成年儿子的身高作为因变量，分析儿子身高与父母身高之间的关系，发现父母的身高可以预测子女的身高，两者近乎一条直线。
- $Y = 0.8567 + 0.516 * X$ (单位为米)



回归历史

➤ 回归的发展

- Tikhonov和Phillips提出**岭回归**（也称Tikhonov regularization），它是一种有偏估计，是对最小二乘估计的改进。
- 1981年Fischler和Bolles提出随机抽样一致（Random Sample Consensus, RANSC）算法。
- 1983年由S.Wold和C.Albano等人首次提出偏最小二乘回归。
- 1996年斯坦福大学Robert Tibshirani首次提出将L1范数作为最小二乘的正则项，从而产生**Lasso回归**模型。
 - ✓ Least absolute shrinkage and selection operator, Tibshirani(1996)
- 2005年Zou和Hastie提出了结合L1范数和L2范数的**弹性网回归**模型。
 - ✓ Zou.H.,&Hastie,T.(2005).Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society:Series B (Statistical Methodology),67(2),301-320.



回归分类

- 最小二乘问题分为两类：具体取决于残差在所有未知数中是否均为线性
 - 线性或普通最小二乘 (Ordinary least squares, OLS)
 - 非线性最小二乘 (Nonlinear least squares, NLS)
- 普通最小二乘回归 (Ordinary least squares, OLS)
 - 是一种线性最小二乘法，用于估计线性回归模型中的未知参数。 它有一个闭式解
 - ✓ 多项式回归使用非线性模型拟和数据，但作为统计估计问题，它是线性的，属于多元线性回归的特例
 - ✓ 当观察来自指数家族（某种概率分布）并且满足一定条件时，最小二乘估计和最大似然估计是相同的。
- 非线性最小二乘 (Nonlinear least squares, NLS)
 - 用非线性模型拟合一组观测值，非线性问题通常通过迭代细化来解决。在每次迭代中，系统都由一个线性系统近似。



正则化最小二乘回归RLS

➤ 普通最小二乘回归 (Ordinary least squares, OLS)

- OLS通过最小二乘原理选择一组解释变量的线性函数的参数：通过线性函数最小化给定数据集中观察到的因变量与预测的变量之间的差的平方和。

➤ 正则化最小二乘回归 (Regularized least squares, RLS)

- RLS是使用正则化进一步约束所得的解来解决最小二乘问题的一系列方法。

Name ◆	Regularization function ◆	Corresponding prior ◆	Methods for solving ◆
Tikhonov regularization	$\ w\ _2^2$	Normal	Closed form
Lasso regression	$\ w\ _1$	Laplace	Proximal gradient descent, least angle regression
ℓ_0 penalization	$\ w\ _0$	–	Forward selection, Backward elimination, use of priors such as spike and slab
Elastic nets	$\beta\ w\ _1 + (1 - \beta)\ w\ _2^2$	Normal and Laplace mixture	Proximal gradient descent
Total variation regularization	$\sum_{j=1}^{d-1} w_{j+1} - w_j $	–	Split-Bregman method , among others



正则化最小二乘回归RLS-原因

- 当线性系统中的变量数量超过观测数量时，采用RLS解决。
 - 因为相关联的优化问题具有无限多个解决方案，普通的最小二乘问题（Ordinary least squares, OLS）是不适定的，因此无法拟合。
 - RLS允许引入进一步的约束，可以唯一地确定解。
- 变量的数量不超过观察的数量，但是学习的模型具有较差的泛化性。
 - RLS通过在训练时对模型进行约束来提高模型的泛化性。
 - 此约束或者使解以某种方式是“稀疏”的，或者反映其他有关该问题的先验知识，例如有关特征之间相关性的信息。
 - RLS方法通常等同于最小二乘问题解的先验，可以达到贝叶斯理解。



BJTU “Machine Learning” Group

于 剑: jianyu@bjtu.edu.cn;

景丽萍: lpjing@bjtu.edu.cn;

田丽霞: lxtian@bjtu.edu.cn;

黄惠芳: hfhuang@bjtu.edu.cn;

李晓龙: hlli@bjtu.edu.cn;

吴 丹: wudan@bjtu.edu.cn;

万怀宇: hywan@bjtu.edu.cn;

王 晶: wj@bjtu.edu.cn.

