高维数据的回归–LASSO

王 成

上海交通大学数学科学学院

Section 1

高维线性回归模型-LASSO

多元回归模型和最小二乘法

给定样本

$$(\mathbf{x}_1, y_1), \ \cdots, (\mathbf{x}_n, y_n),$$

其中$\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$为解释变量， $y_1, \cdots, y_n \in \mathbb{R}$ 为响应变量.

多元回归模型和最小二乘法

给定样本

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n),$$

其中$\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$为解释变量， $y_1, \cdots, y_n \in \mathbb{R}$ 为响应变量.

最小二乘法寻找最优的线性投影

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2.$$

回归模型的矩阵表示

记

$$\mathbb{X} = \begin{pmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix}_{n\times p} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \mathbb{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},$$

用矩阵形式最小二乘可以表示成

$$\arg\min_{\beta\in\mathbb{R}^p} \ (\mathbb{Y} - \mathbb{X}\beta)^\top(\mathbb{Y} - \mathbb{X}\beta).$$

最小二乘法的推导

对参数β计算导数和Hessian矩阵，

$$\frac{\partial(\mathbb{Y}-\mathbb{X}\beta)^\top(\mathbb{Y}-\mathbb{X}\beta)}{\partial\beta} = 2\mathbb{X}^\top(\mathbb{X}\beta-\mathbb{Y}),$$
$$\frac{\partial^2(\mathbb{Y}-\mathbb{X}\beta)^\top(\mathbb{Y}-\mathbb{X}\beta)}{\partial\beta\partial\beta^\top} = 2\mathbb{X}^\top\mathbb{X} \geq 0.$$

- 当矩阵$\mathbb{X}^\top\mathbb{X}$ 严格正定的时候，我们有 最小二乘估计

$$\hat{\beta} = (\mathbb{X}^\top\mathbb{X})^{-1}\mathbb{X}^\top\mathbb{Y}.$$

最小二乘法的推导

对参数β计算导数和Hessian矩阵，

$$\frac{\partial(\mathbb{Y}-\mathbb{X}\beta)^\top(\mathbb{Y}-\mathbb{X}\beta)}{\partial\beta} = 2\mathbb{X}^\top(\mathbb{X}\beta-\mathbb{Y}),$$
$$\frac{\partial^2(\mathbb{Y}-\mathbb{X}\beta)^\top(\mathbb{Y}-\mathbb{X}\beta)}{\partial\beta\partial\beta^\top} = 2\mathbb{X}^\top\mathbb{X} \geq 0.$$

- 当矩阵$\mathbb{X}^\top\mathbb{X}$ 严格正定的时候，我们有最小二乘估计

$$\hat{\beta} = (\mathbb{X}^\top\mathbb{X})^{-1}\mathbb{X}^\top\mathbb{Y}.$$

- 当$\mathbb{X}^\top\mathbb{X}$不严格正定时候，会有无穷多组解.
(对应解为一个线性空间，维度为$p-\mathrm{rank}(\mathbb{X}^\top\mathbb{X})$).

高维线性回归–稀疏回归

现代大数据分析中，需要处理高维数据$p \gg n$(例如$n = 100, p = 10000$)，这时候:

$$rank(\mathbb{X}^\top \mathbb{X}) \leq n < p.$$

高维线性回归–稀疏回归

现代大数据分析中，需要处理高维数据$p \gg n$(例如$n = 100, p = 10000$), 这时候:

$$rank(\mathbb{X}^\top \mathbb{X}) \le n < p.$$

另外，从高维数据本身需求以及模型的简洁可解释性角度，希望去除特征中的绝大多数噪音，也就是不要使用所有特征的线性组合:

$$\|\beta\|_0 = \sum_{i=1}^{p} I(\beta_i \neq 0) = \sum_{i=1}^{p} |\beta_i|^0 \le k,$$

即希望使用不超过k个特征的线性组合来建模。

变量选择

如果p很小或者k很小，可以通过排列组合的方式来实现算法:

- 如果希望使用1个特征来建模，即在p个特征中选取一个最重要的;
- 如果希望使用2个特征来建模，即在p个特征中挑选2个 C_p^2;
- 希望使用不超过k个特征的线性组合来建模:

$$C_p^0 + C_p^1 + \cdots + C_p^k.$$

变量选择

如果p很小或者k很小，可以通过排列组合的方式来实现算法:

- 如果希望使用1个特征来建模，即在p个特征中选取一个最重要的;
- 如果希望使用2个特征来建模，即在p个特征中挑选2个 C_p^2;
- 希望使用不超过k个特征的线性组合来建模:

$$C_p^0 + C_p^1 + \cdots + C_p^k.$$

经典统计中(p一般不大)，可以通过向前法或者向后法来实现变量选择:

- 向前法: 每次在余下特征中选取一个加入到当前模型;
- 向后法: 每次从当前模型中删除一个特征。

Section 2

高维ℓ_1惩罚方法

ℓ_0优化

为了实现稀疏回归，可以考虑带约束的最小二乘法:

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2, \text{ subject to } \|\beta\|_0 \le k.$$

ℓ_0优化

为了实现稀疏回归，可以考虑带约束的最小二乘法:

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \beta)^2, \text{ subject to } \|\beta\|_0 \leq k.$$

- 从模型的角度，k的选取非常重要；(可以通过交叉验证来解决)
- 从优化的角度，基于ℓ_0的优化通常没有高效算法。(没有解决方案！)

ℓ_1优化

一个自然的改进是引入ℓ_1优化，即

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \beta)^2, \text{ subject to } |\beta|_1 = \sum_{i=1}^{p}|\beta_i| \le t,$$

其中$t \ge 0$是一个调节参数(tuning parameter).

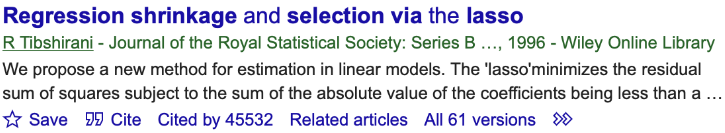
ℓ_1优化

一个自然的改进是引入ℓ_1优化，即

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\beta)^2, \text{ subject to } |\beta|_1 = \sum_{i=1}^{p}|\beta_i| \le t,$$

其中$t \ge 0$是一个调节参数(tuning parameter).

这正是Tibshirani在1996年提出的LASSO方法(Least Absolute Shrinkage and Selection Operator)


Regression shrinkage and selection via the lasso
R Tibshirani - Journal of the Royal Statistical Society: Series B …, 1996 - Wiley Online Library
We propose a new method for estimation in linear models. The 'lasso'minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a …
☆ Save 99 Cite Cited by 45532 Related articles All 61 versions

LASSO方法

对于LASSO方法:

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \beta)^2, \ \text{s.t.}\ |\beta|_1 = \sum_{i=1}^{p}|\beta_i| \leq t,$$

约束条件$|\beta|_1 \leq t$限制了可行解β的选取范围.

LASSO方法

对于LASSO方法:

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\beta)^2, \text{ s.t. } |\beta|_1 = \sum_{i=1}^{p}|\beta_i| \le t,$$

约束条件$|\beta|_1 \le t$限制了可行解β的选取范围. 假设模型的全局最小二乘解存在, 记为$\hat{\beta}_{OLS}$,

- 如果$t \ge |\hat{\beta}_{OLS}|_1$, 全局最优解$\hat{\beta}_{OLS}$满足约束条件, 所以$\hat{\beta} = \hat{\beta}_{OLS}$;
- 当$t < |\hat{\beta}_{OLS}|_1$时候, 我们需要在一个更小的范围内寻找优化解。因为ℓ_1惩罚的性质, 更加容易找到一个稀疏的解.

LASSO等价形式

用Lagrange Multiplier Method，优化问题可以写为：

$$\min \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \beta)^2 + \lambda(|\beta|_1 - t).$$

LASSO等价形式

用Lagrange Multiplier Method，优化问题可以写为：

$$\min \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \beta)^2 + \lambda(|\beta|_1 - t).$$

忽略与β无关项， 问题变为

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top \beta)^2 + \lambda|\beta|_1.$$

这也是LASSO的一种常见等价形式。 这里的等价是说

LASSO等价形式

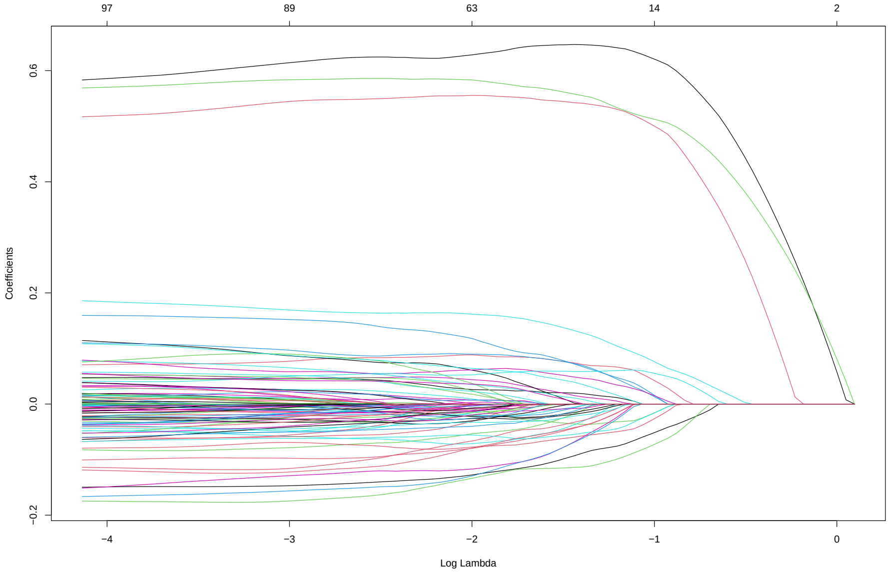
每个$\lambda \geq 0$对应的解$\hat{\beta}(\lambda)$都有一个对应的$t \geq 0$，使得两者是一样的。

LASSO的solution path

```
rm(list=ls())
set.seed(123)
library(glmnet,quietly =TRUE)

## Loaded glmnet 4.1-4

n=100
p=10000
beta=c(1,1,1,rep(0,p-3))
X<-matrix(rnorm(n*p),nrow=p)
epsilon<-rnorm(n)
Y<-t(X)%*%beta+epsilon
aa<-glmnet(t(X),Y,standardize =FALSE,intercept =FALSE)
```


97
89
63
14
2
Coefficients
0.6
0.4
0.2
0.0
−0.2
−4
−3
−2
−1
0
Log Lambda

Section 3

LASSO的变形形式

Compressed Sensing

奈奎斯特，H.

1889-1976

Nyquist

Emmanuel Candes

David Donoho

陶哲轩

Basis Pursuit (Chen et al., 2001)

无噪音情形下, 对于一个Basis $A = A_{n\times p}$ 和投影后的n维向量a:

$$\arg\min_{\beta\in\mathbb{R}^p} \|\beta\|_1, \text{such that} \quad A\beta = a.$$

即给定一个p维向量的n个线性组合,恢复原始p维向量。
(注意: $p \gg n$!).

Basis Pursuit (Chen et al., 2001)

无噪音情形下, 对于一个Basis $A = A_{n\times p}$ 和投影后的n维向量a:

$$\arg\min_{\beta\in\mathbb{R}^p} \|\beta\|_1, \text{such that} \quad A\beta = a.$$

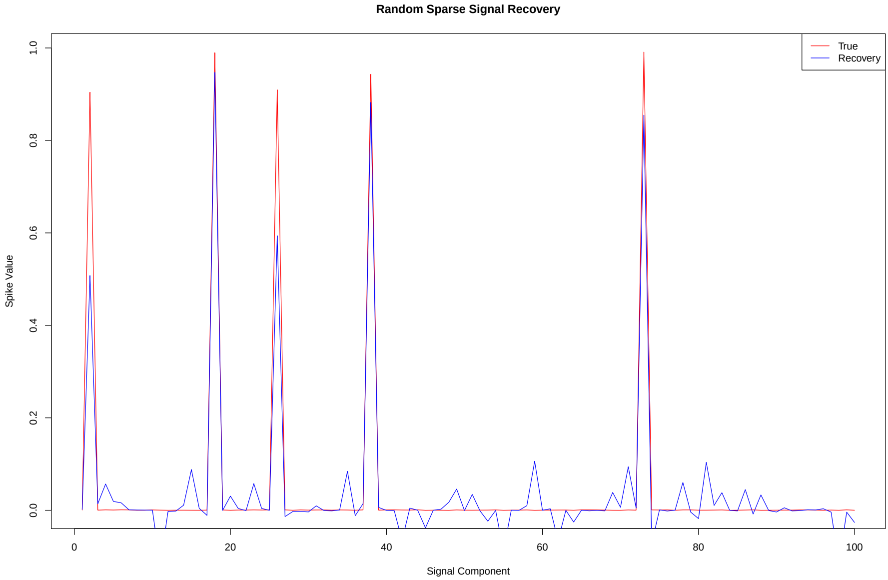
即给定一个p维向量的n个线性组合,恢复原始p维向量。
(注意: $p \gg n$!).

有噪音清晰下，可以松弛后面的等式，考虑

$$\arg\min_{\beta\in\mathbb{R}^p} \|\beta\|_1, \text{such that} \quad \frac{1}{2n}\|A\beta - a\|_2^2 \leq b^2.$$

Chen et al. 2001称之为relaxed basis pursuit.

```
rm(list=ls())
library(R1magic)#  Signal components
set.seed(99)
N <- 100
# Sparse components
K <- 5
#  Up to Measurements  > K LOG (N/K)
M <- 40
# Measurement Matrix (Random Sampling Sampling)
phi <- GaussianMatrix(N,M)
# R1magic generate random signal
xorg <- sparseSignal(N, K, nlev=1e-3)
y <- phi %*% xorg ;# generate measurement
T <- diag(N) ;# Do identity transform
p <- matrix(0, N, 1) ;# initial guess
# R1magic Convex Minimization ! (unoptimized-parameter)
ll <- solveL1(phi, y, T, p)
x1 <- ll$estimate
```



Random Sparse Signal Recovery
True
Recovery
Spike Value
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
Signal Component

Dantzig Selector (Candes and Tao, 2007)

对于LASSO问题，

$$\frac{1}{2n}(\mathbb{Y}-\mathbb{X}\beta)^\top(\mathbb{Y}-\mathbb{X}\beta)+\lambda\|\beta\|_1,$$

对目标函数求导(ℓ_1求次导数), 可以得到

$$\arg\min_{\beta\in\mathbb{R}^p}\|\beta\|_1, \text{ such that } \|\mathbb{X}^\top(\mathbb{X}\beta-\mathbb{Y})\|_\infty\le\lambda,$$

Candes and Tao (2007)称之为Dantzig Selector.

Section 4

ℓ_1惩罚

为什么引入ℓ_1惩罚?

在回归分析中，可以考虑ℓ_0, ℓ_1和ℓ_2(岭回归 Ridge Regression)三种不同惩罚：

$$\ell_0 : \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\beta)^2, \ \text{s.t.} |\beta|_0 = \sum_{i=1}^{p}|\beta_i|^0 \le t;$$

$$\ell_1 : \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\beta)^2, \ \text{s.t.} |\beta|_1 = \sum_{i=1}^{p}|\beta_i| \le t;$$

$$\ell_2 : \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^\top\beta)^2, \ \text{s.t.} |\beta|_2^2 = \sum_{i=1}^{p}|\beta_i|^2 \le t.$$

Proximal projection

给定样本$x_1, \cdots, x_n \in \mathbb{R}$, 考虑最优投影:

$$x_0 = \arg\min_{x \in \mathbb{R}} \frac{1}{2n} \sum_{i=1}^{n} (x_i - x)^2, \quad \text{s.t. } |x|^k \leq t.$$

这里$k = 0, 1, 2$分别对应ℓ_0, ℓ_1和ℓ_2惩罚, $t \geq 0$是调节参数. 也就是从集合

$$\{x : \ |x|^k \leq t\}$$

中找一个点,距离当前样本平均距离最近. (数学上称之为投影)

练习

等价的，考虑优化问题

$$\arg\min_{x\in\mathbb{R}} \frac{1}{2n}\sum_{i=1}^{n}(x_i - x)^2 + \lambda|x|^k,$$

这里$\lambda \geq 0$. 尝试对$k = 0, 1, 2$写出优化问题的显示解.

练习

等价的，考虑优化问题

$$\arg\min_{x\in\mathbb{R}} \frac{1}{2n}\sum_{i=1}^{n}(x_i - x)^2 + \lambda|x|^k,$$

这里$\lambda \geq 0$. 尝试对$k = 0, 1, 2$写出优化问题的显示解.

惩罚函数

一般的对于$\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}^p$, 考虑

$$\arg\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|^k.$$

惩罚函数

- Hard Thresholding 函数:

$$f(x,\lambda) = xI(|x| > \lambda),\ \ \lambda > 0, x \in \mathbb{R};$$

- Soft Thresholding 函数:

$$f(x,\lambda) = (x+\lambda)I(x < -\lambda) + (x-\lambda)I(x > \lambda),\ \ \lambda > 0, x \in \mathbb{R};$$

- Tikhonov Regularization (岭回归)

$$f(x,\lambda) = (1+\lambda)^{-1}x, ,\ \ \lambda > 0, x \in \mathbb{R}.$$

Section 5

如何调参?

监督学习流程

数据科学家

- Step 1: 拿到训练数据(training data)

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n).$$

- Step 2: 经过认真专业的分析，得到函数$f(\cdot)$ ($y_i \approx f(\mathbf{x}_i)$).

- Step 3: 基于$f(\cdot)$输出产品.

客户

Step 4: 拿到产品$f(\cdot)$, 进行测试(代入测试数据, test data) $\mathbf{z}_1, \cdots, \mathbf{z}_m$, 得到预测值：

$$f(\mathbf{z}_1), \cdots, f(\mathbf{z}_m).$$

根据使用效果决定支付数据科学家多少钱.

客户

Step 4: 拿到产品$f(\cdot)$, 进行测试(代入测试数据, test data) $\mathbf{z}_1, \cdots, \mathbf{z}_m$, 得到预测值：

$$f(\mathbf{z}_1), \cdots, f(\mathbf{z}_m).$$

根据使用效果决定支付数据科学家多少钱.

模型评估与选择(例如使用什么模型、什么样的参数) 是帮助数据科学家自己来评估一下所得产品$f(\cdot)$的效果如何!

训练误差和测试误差

- 训练误差(training error): 数据科学家把$f(\cdot)$代入训练数据的拟合程度.(即$y_i \approx f(\mathbf{x}_i)$的约等于号程度.)
- 测试误差(test error): 用户的使用效果–把$f(\cdot)$代入测试数据得到的预测值与实际值之间的差距.

过拟合和欠拟合

- 过拟合(over-fitting): 数据科学家过于认真，把训练数据独有的很多特征都学到了，放进了最终模型.因为这些特征不具有普适性造成效果不好.(模型过于繁琐).
- 欠拟合(under-fiting)：数据科学家不靠谱，所得模型没有充分挖掘出数据的特征.(模型过于简单)

互动

在这两种情况下，训练误差和测试误差的效果如何?

模型评估

监督学习的目标是具有好的测试误差(基于测试集)，数据科学家拿到的只有训练集，所以需要人为的制造一些“测试集”.

重抽样方法

- Step 1: 把观察数据分成两部分

$$\text{Training data(训练集) 和 Test data(测试集)}$$

- Step 2: 基于训练集合得到模型估计$\hat{f}$;
- Step 3: 把估计模型应用到测试集上(根据问题和目标的不同，计算对应的损失函数)
- Step 4: 重复 Step 1- Step 3

根据重复机制的不同对应不同的重抽样方法。

常用重抽样方法

- Leave-one-out: 每次留下一个数据做训练，用其余的$n-1$个做建模。
- Bootstrapping: 每次随机的从当前n个样本中抽取$m(m < n)$个样本做训练，余下的$n-m$个做测试，整个过程重复N次；
- K-folds 交叉验证： 把数据分成K个folds，每次用$K-1$个folds训练，余下一个fold做测试，重复K次。常用的例如10-folds,5-folds;

重抽样方法的思考

- 每种重抽样方法都改变了原始的训练数据集数量；
- 每次测试或者计算的损失函数都不对应同一个f;
- 实战中不同的重抽样方法可能得到不一致结果;
- $\cdots$

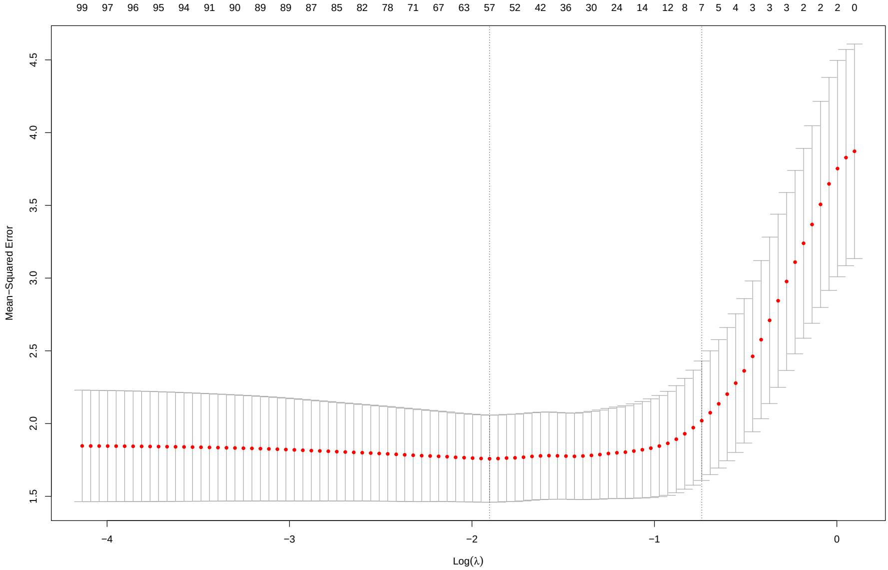
思考

重抽样方法到底在评估什么?

实际应用中采用哪种交叉验证方法取决于：

- 样本容量n;
- 训练模型的计算复杂度；
- 损失函数的光滑性；
- 整个机制的可重复性(稳健性)。

LASSO调参–交叉验证



Thank you !

E. Candes and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35 (6):2313–2351, 2007.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.