第6章

线性模型选择与正则化

Linear Model Selection and Regularization

线性模型

标准线性回归模型：

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

通常用于描述响应变量 Y和一系列预测变量$X^1, X^2, \cdots, X^P$之间的线性关系。

- 参数估计方法：最小二乘法。
- 线性模型虽然简单，但在可解释性和良好的预测性能方面具有明显的优势。
- 改进简单线性模型的方法，用一些拟合程序来取代普通的最小二乘拟合。

最小二乘法

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

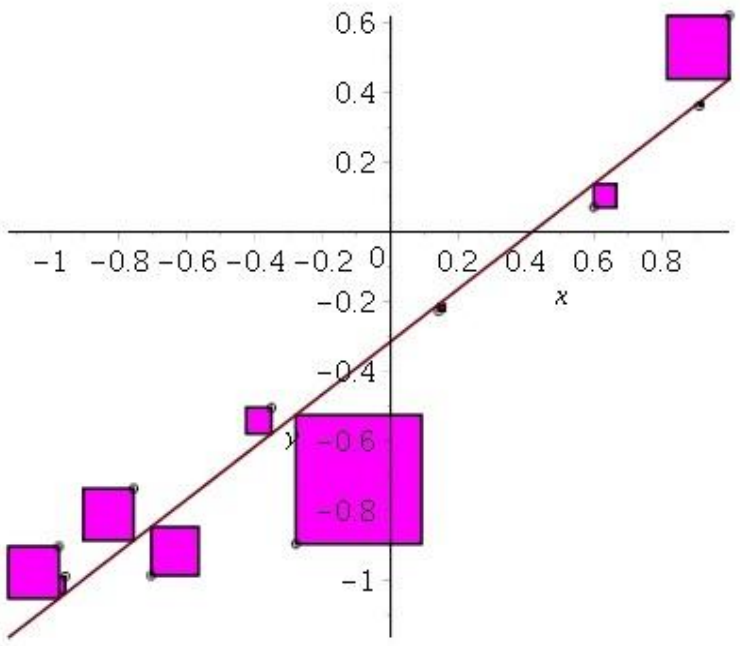
$$e_i = y_i - \hat{y}_i$$

表示第i个残差

残差平方和：residual sum of squares (RSS)

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = e_1^2 + e_2^2 + \cdots + e_n^2$$

$$\mathrm{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \ldots + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2$$



最小二乘法选择β_0、β_1使RSS最小

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

$$\bar{y} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \bar{x} \equiv \frac{1}{n}\sum_{i=1}^{n} x_i$$

准确率与解释力

预测准确率（prediction accuracy）

- 若响应变量和预测变量真实关系近似线性，则最小二乘的偏差较低；
- 若观测个数n远大于p，则最小二乘的方差也较低；
- 若不满足n远大于p，则使用最小二乘可能导致过拟合；
- 若p>n，则最小二乘得到的系数估计结果不唯一：此时方差无穷大，无法使用最小二乘法。

改进措施：通过**限制**或**缩减**待估计系数，在牺牲偏差显著减小估计量方差

准确率与解释力

模型解释力（model interpretability）

- 在多元回归模型中，常常存在一个或多个预测变量与响应变量不存在线性关系的情况，包括一些增加了模型的复杂性、却与模型无关的变量。
- 通过去除不相关的特征，将无关变量的系数设置为0，并移除这些无关变量（去不相关）可得到一个更容易解释的模型， 但最小二乘法很难将系数置为0。

改进措施：通过自动进行**特征选择**或**变量选择**，在多元回归模型中实现对无关变量的筛选。

改进方法

子集选择

从p个预测变量中挑选出与响应变量相关的变量形成子集，再对缩减的变量集合使用最小二乘方法。

压缩估计（正则化）

基于全部p个预测变量进行模型的拟合，与最小二乘方法相比，该方法可以将估计系数往零的方向进行压缩。通过系数缩减(又称正则化) 减少方差，还可以用于变量选择。

降维法

将p维预测变量投影至 M维子空间中 M<p。这通常通过计算这p个变量的M种不同的线性组合或称投影来实现。将这个不同的投影作为预测变量，再使用最小二乘法拟合线性回归模裂。

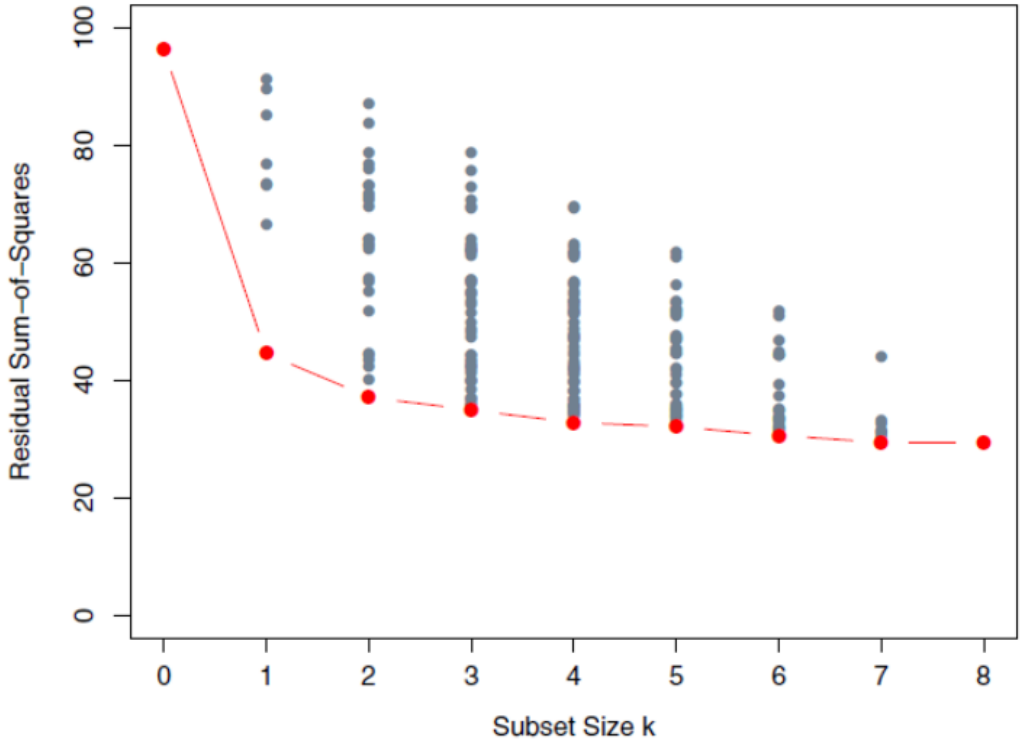
本章内容概况

- 模型选择与正则化
  - 子集选择
    - 最优子集选择：遍历所有可能的特征子集，选择一个最好的
    - 逐步选择
      - 向前逐步选择：即从空集开始，每次增加一个对模型最有帮助的特征。
      - 向后逐步选择：从全部特征集开始，每次删除一个对模型没有帮助的特征。
    - 选择最优模型：AIC、BIC与调整R2；验证集与交叉验证
  - 压缩估计（正则化）
    - 岭回归：在优化目标中加入L2正则化约束
    - lasso：在优化目标中加入L1正则化约束
    - 选择调节参数：参数λ
  - 降维法
    - 主成分回归
      - PCA：主成分分析：寻找一组k维正交基，将原始数据线性变换到这组基中，使方差尽可能大。
      - PCR：PCA+LR
    - 偏最小二乘：PLSR:和PCA相似，区别是这个有监督的

最优子集选择

最优子集选择（best subset selection）：即对p个预测变量的所有可能组合分别使用最小二乘进行拟合。

- 对含有一个预测变量的模型，拟合p个模性；
- 对含有两个预测变量的模型，拟合 $\binom{p}{2} = p(p-1)/2$ 个模型依次类推。最后选取一个最优模型。



红色点为当前k个变量能达到的最小二乘，灰色点为其他非最优的情况

最优子集选择

算法1 最优子集选择过程

1.记不含预测变量的零模型为 $\mathcal{M}_0$ ，只用于估计各观测的样本均值。

2.对于 $k=1,2,\cdots,p$:

（a）拟合 $\binom{p}{k}$ 个包含 k 个预测变量的模型；

（b）在 $\binom{p}{k}$ 个模型中选择 RSS 最小或 R^2 最大的作为最优模型，记为 $\mathcal{M}_k$.

3.根据交叉验证模型误差、C_p（AIC）、BIC 或者调整 R^2 从 $\mathcal{M}_0,\ldots,\mathcal{M}_p$ 个模型中选择出一个最优模型。

最优子集选择

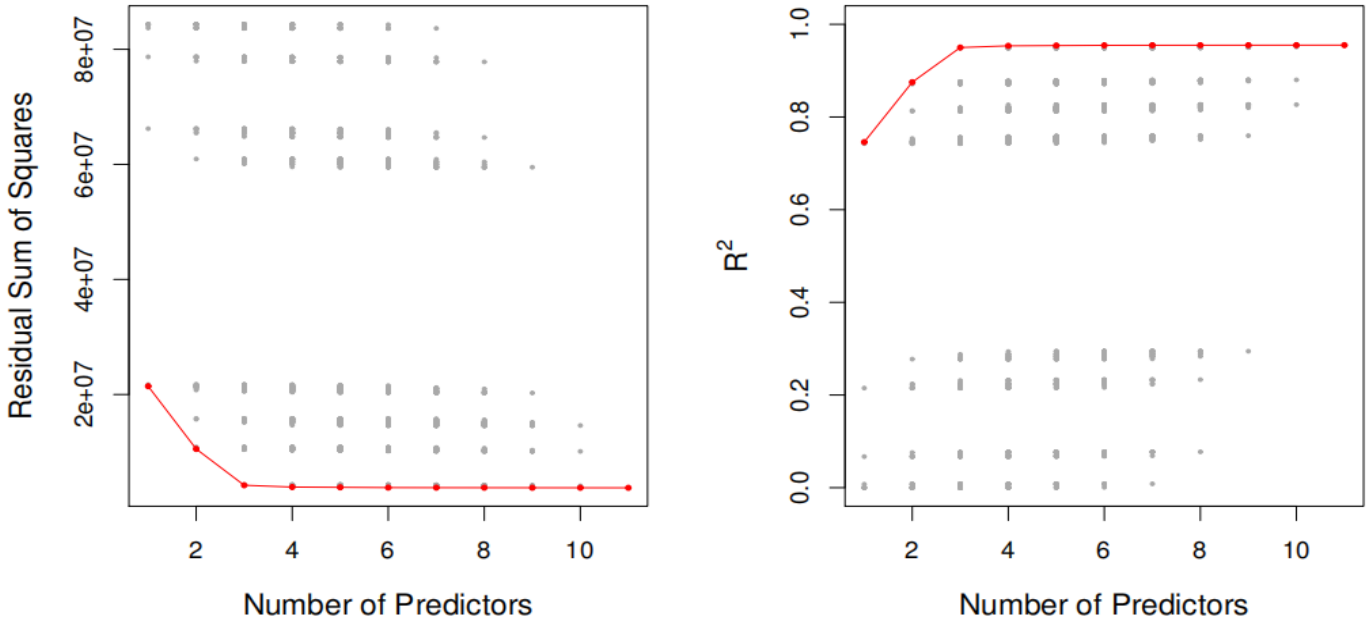


图1.1 展示Credit数据集10个预测变量所有可能子集构成的模型的RSS 及R^2

- 折线为最优模型的RSS和R^2值轨迹。
- 散点图为其他非最优模型的情况

低 RSS 及高 R^2表明模型的训练(traning)误差低，而目标是选择一个测试(test) 误差低的模型，因此需使用如下方法：

- 使用交叉验证预测误差、C_p、BIC。
- 调整R^2从M0, . . . ,Mp模型中进行选择。

最优子集选择

这里使用Hitters（棒球）数据集实践最优子集选择方法。使用若干个与棒球运动员上一年比赛成绩相关的变量来预测该棒球运动员的Salary（薪水）。

```
> library(ISLR)
> fix(Hitters)
> names(Hitters)
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"       "Walks"
 [7] "Years"     "CAtBat"    "CHits"     "CHmRun"    "CRuns"     "CRBI"
[13] "CWalks"    "League"    "Division"  "PutOuts"   "Assists"   "Errors"
[19] "Salary"    "NewLeague"
> dim(Hitters)
[1] 322  20
```

	row.names	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	League	Division	PutOuts	Assists	Errors	Salary	NewLeague
1	-Andy Allanson	293	66	1	30	29	14	1	293	66	1	30	29	14	A	E	446	33	20	NA	A
2	-Alan Ashby	315	81	7	24	38	39	14	3449	835	69	321	414	375	N	W	632	43	10	475	N
3	-Alvin Davis	479	130	18	66	72	76	3	1624	457	63	224	266	263	A	W	880	82	14	480	A
4	-Andre Dawson	496	141	20	65	78	37	11	5628	1575	225	828	838	354	N	E	200	11	3	500	N
5	-Andres Galarraga	321	87	10	39	42	30	2	396	101	12	48	46	33	N	E	805	40	4	91.5	N
6	-Alfredo Griffin	594	169	4	74	51	35	11	4408	1133	19	501	336	194	A	W	282	421	25	750	A

最优子集选择

这里使用Hitters（棒球）数据集实践最优子集选择方法。使用若干个与棒球运动员上一年比赛成绩相关的变量来预测该棒球运动员的Salary（薪水）。

```
> library(ISLR)
> fix(Hitters)
> names(Hitters)
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"       "Walks"
 [7] "Years"     "CAtBat"    "CHits"     "CHmRun"    "CRuns"     "CRBI"
[13] "CWalks"    "League"    "Division"  "PutOuts"   "Assists"   "Errors"
[19] "Salary"    "NewLeague"
> dim(Hitters)
[1] 322  20
> sum(is.na(Hitters$Salary))
[1] 59
> Hitters=na.omit(Hitters)
> dim(Hitters)
[1] 263  20
> sum(is.na(Hitters))
[1] 0
>
```

is.na()函数可用于识别有缺失值的观测。

sum()函数可用于计算所有缺失值的个数。

最优子集选择

```
> library(leaps)
> regfit.full=regsubsets(Salary~.,Hitters)
> summary(regfit.full)
Subset selection object
Call: regsubsets.formula(Salary ~ ., Hitters)
19 Variables  (and intercept)
           Forced in Forced out
AtBat          FALSE      FALSE
Hits           FALSE      FALSE
HmRun          FALSE      FALSE
Runs           FALSE      FALSE
RBI            FALSE      FALSE
Walks          FALSE      FALSE
Years          FALSE      FALSE
CAtBat         FALSE      FALSE
CHits          FALSE      FALSE
CHmRun         FALSE      FALSE
CRuns          FALSE      FALSE
CRBI           FALSE      FALSE
CWalks         FALSE      FALSE
LeagueN        FALSE      FALSE
DivisionW      FALSE      FALSE
PutOuts        FALSE      FALSE
Assists        FALSE      FALSE
Errors         FALSE      FALSE
NewLeagueN     FALSE      FALSE
```

- summary( ) 命令可以输出模型大小不同的情况下最优的预测变量子集
- regsubest( ) 函数可通过建立一系列包含给定数目预测变量的最优模型，实现最优预测变量子集的筛选。

```
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI
1  ( 1 ) " "   " "  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"
2  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"
3  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"
4  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"
5  ( 1 ) "*"   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"
6  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    " "   "*"
7  ( 1 ) " "   "*"  " "   " "  " " "*"   " "   "*"    "*"   "*"    " "   " "
8  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "   "*"    "*"   " "
         CWalks LeagueN DivisionW PutOuts Assists Errors NewLeagueN
1  ( 1 ) " "    " "     " "       " "     " "     " "    " "
2  ( 1 ) " "    " "     " "       " "     " "     " "    " "
3  ( 1 ) " "    " "     " "       "*"     " "     " "    " "
4  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
5  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
6  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
7  ( 1 ) " "    " "     "*"       "*"     " "     " "    " "
8  ( 1 ) "*"    " "     "*"       "*"     " "     " "    " "
> |
```

星号表示列对应的变量包含于行对应的模型当中。

最优子集选择

扩展

- 这里给出了最小二乘回归的最佳子集选择，但也适用于其他类型的模型，如逻辑回归。
- 例如逻辑斯谛回归模型，算法步骤2中应当使用偏差（deviance） 替代原先的 RSS 对模型进行选择，偏差与 RSS 作用相间，但适用范围宽广。偏差定义为 -2 与最大似然函数值的乘积，偏差越小，拟合优度越高。

缺点

- 最优子集选择方法简单直观，但计算效率不高，当p>40，该方法不具计算可行性。
- 随着搜索空间增大，通过此方法找到的模型对新数据不具备很好的预测能力。
- 从一个巨大搜索空间得到的模型通常会有过拟合和系数估计方差高的问题。

逐步选择

向前逐步选择

从一个不包含任何变量的零模型开始，依次在模型中添加变量，直到所有预测变量都在模型中。特别地，每次只将能够对最大限度地提升模型效果变量加入模型中。

算法2 向前逐步选择

1.记不含预测变量的零模型为 $\mathcal{M}_0$

2.对于 $k=0,1,2,\cdots,p-1$：

（a）从 p-k 个模型中进行选择，每个模型都在模型 $\mathcal{M}_k$ 的基础上增加一个变量；

（b）在 p-k 个模型中选择 RSS 最小或 R^2 最高的模型作为最优模型，记为 $\mathcal{M}_{k+1}$.

3.根据交叉验证预测误差、C_p（AIC）、BIC 或者调整 R^2 从 $\mathcal{M}_0,\ldots,\mathcal{M}_p$ 个模型中选择出一个最优模型。

逐步选择

向前逐步选择在运算效率上优于最佳子集选择，但该方法产生了一系列的嵌套模型，因此无法保证找到的模型是所有2^p个模型中最优的。

表1 对Credit数据使用最优子集选择和向前逐步选择的前四个模型结果

变量个数	最优子集选择	向前逐步选择
1	rating	rating
2	rating, income	rating, income
3	rating, income, student	rating, income, student
4	cards, income, student,limit	rating, income, student,limit

逐步选择

向后逐步选择

它以包含全部 p个变量的全模型为起点，逐次迭代，每次移除一个对模型拟合结果最不利的变量。

算法3 向后逐步选择

1.记包含 p 预测模型的全模型为 $\mathcal{M}_p$。

2.对于 $k = p, p-1, \cdots, 1$：

（a）从 k 个模型中进行选择，在模型 $\mathcal{M}_k$ 的基础上减少一个变量，则模型只含 k-1 个变量；

（b）在 k 个模型中选择 RSS 最小或 R^2 最高的模型作为最优模型，记为 $\mathcal{M}_{k-1}$.

3.根据交叉验证预测误差、C_p（AIC）、BIC 或者调整 R^2 从 $\mathcal{M}_0, \ldots, \mathcal{M}_p$ 个模型中选择出一个最优模型。

逐步选择

向后逐步选择特性：

- 只通过1 + p(p + 1)/2模型进行搜索，可应用于p太大而无法应用最佳子集选择的情况。
- 不能保证产生包含p个预测因子子集的最佳模型。
- 需满足样本量n大于变量个数p (保证全模型可以被拟合)的条件。而向前逐步选择即使在n <p 的情况下也可以使用，因此当n非常大的时候，向前逐步选择是唯一可行的方法。

逐步选择

R中向前逐步选择和向后逐步选择可以分别通过设定regsubsets()函数中的参数method="forward"和method="backward实现"

```
> regfit.fwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="forward")
> summary(regfit.fwd)

> regfit.bwd=regsubsets(Salary~.,data=Hitters,nvmax=19,method="backward")
> summary(regfit.bwd)
```

使用coef ( ) 函数可提取以上模型的参数估计值

```
> coef(regfit.full,7)
 (Intercept)         Hits        Walks       CAtBat        CHits       CHmRun    DivisionW      PutOuts
  79.4509472    1.2833513    3.2274264   -0.3752350    1.4957073    1.4420538 -129.9866432    0.2366813
> coef(regfit.fwd,7)
 (Intercept)        AtBat         Hits        Walks         CRBI       CWalks    DivisionW      PutOuts
109.7873062   -1.9588851    7.4498772    4.9131401    0.8537622   -0.3053070 -127.1223928    0.2533404
> coef(regfit.bwd,7)
 (Intercept)        AtBat         Hits        Walks        CRuns       CWalks    DivisionW      PutOuts
105.6487488   -1.9762838    6.7574914    6.0558691    1.1293095   -0.7163346 -116.1692169    0.3028847
```

选择最优模型

包含所有预测变量的模型总是具有最小的RSS和最大的R^2，这些统计量与训练误差有关。但需要找到具有最小测试误差的模型，并不适用于对包含不同个数预测变量的模型进行模型选择。

为了达到基于测试误差选择最优模型的目的，需要估计测试误差，通常有两种方法:

1. 根据过拟合导致的偏差对训练误差进行调整，间接地估计测试误差。

Cp（AIC）、BIC或者调整R^2

2. 采用验证集方法或交叉验证方法，直接估计测试误差。

验证与交叉验证

选择最优模型

C_p

采用最小二乘法拟合一个包含d个预测变量的模型，C_p值计算如下：

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

- 其中$\hat{\sigma}^2$是公式中各个响应变量观测误差的方差ε的估计值。
- C_p是测试均方误差的无偏估计。
- 测试误差较低的模型C_p统计量取值也较低，可以通过选择具有最低C_p的模型作为最优模型。

选择最优模型

AIC

赤池信息量准则（Akaike information criterion, AIC）

AIC准则适应于许多使用极大似然法进行拟合的模型，式中L为估计模型似然函数的最大值。

$$AIC = -2\log L + 2 \cdot d$$

AIC准则适用于许多使用极大似然法进行拟合的模型，若模型误差项服从高斯分布，极大似然估计和最小二乘估计是等价的，AIC可由下式给出：

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2)$$

此时C_p和AIC批次成比例。

选择最优模型

BIC

贝叶斯信息准则（Bayesian information criterion,BIC）

对于包含d个预测变量的最小二乘模型，BIC通常由下式给出：

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2)$$

- 测试误差较低的模型BIC统计量取值也较低，因此通常选取具有最低BIC的模型作为最优模型。
- BIC将C_p使用的 $2d\hat{\sigma}^2$ 替换为 $\log(n)d\hat{\sigma}^2$ 项，其中n为观测数。

选择最优模型

R^2统计量

是衡量拟合度的另一个标准。R^2统计量采取比例形式，所以它的值总在0和1之间，与Y的量级无关。R^2用下式表示：

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2$ 响应变量的总平方和

$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ 残差平方和

- R^2统计量接近0说明回归没有解释太多响应变量的变异；R^2统计量接近1说明回归可以解释响应变量的大部分变异；
- R^2随着模型包含的变量个数的增加而增加，在样本容量一定的情况下，增加解释变量可以减少残差，也可使其自由度减少。

选择最优模型

调整R^2

调整R^2统计量是另一种常用的对一系列具有不同变量个数的模型进行选择的方法。

对于包含d个变量的最小二乘模型，调整R^2统计量为下式：

$$调整R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

- 调整R^2的值越大，模型测试误差越低。
- 若加入冗余变量，d值增加，便会导致调整R2的值降低，因此，理论上拥有最大调整R^2的模型只包含正确变量，无冗余变量。
- 调整R^2统计量对纳入不必要变量的模型引入了惩罚。

选择最优模型

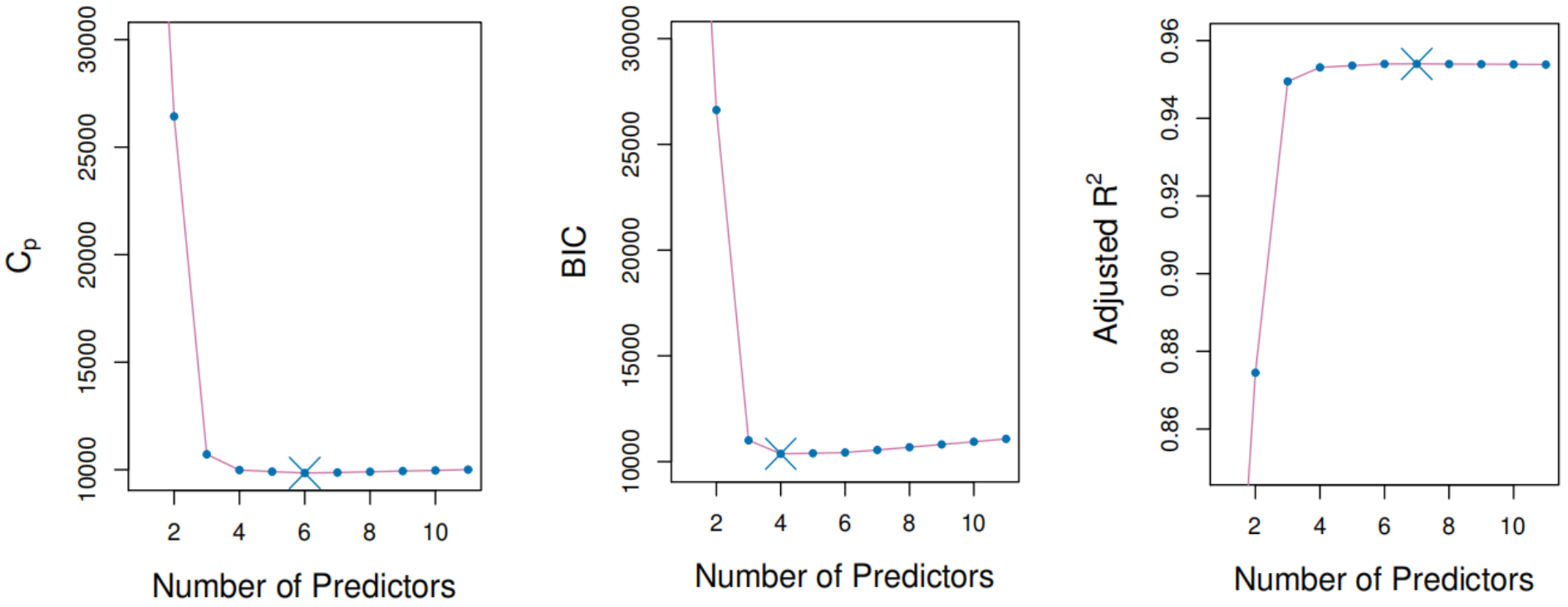


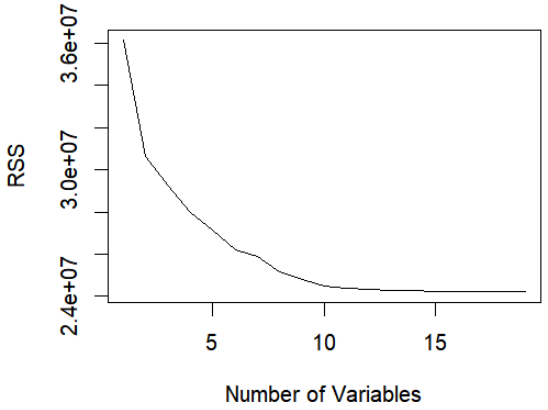
图1.2 中显示对Credict数据集中运用Cp、 BIC和调整R2对不同规模的模型进行最优子集选择的结果

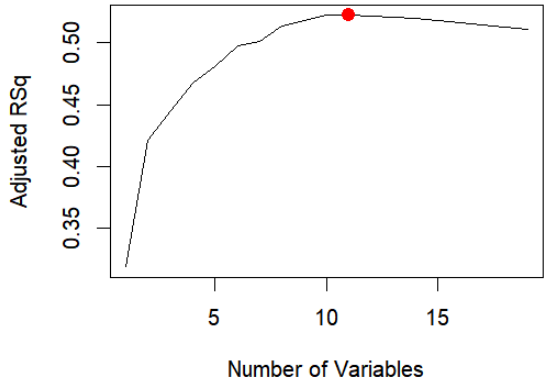
选择最优模型

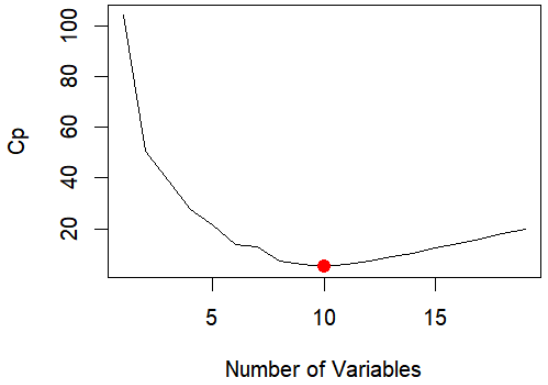
summary ( ) 函数可返回相应模型的R^2、RSS、调整R^2、C_p及BIC。

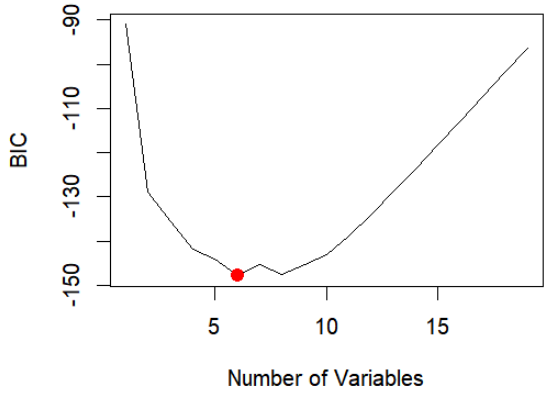
```
> reg.summary=summary(regfit.full)
> names(reg.summary)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
> reg.summary$rsq
 [1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227 0.5285569 0.5346124 0.5404950
[11] 0.5426153 0.5436302 0.5444570 0.5452164 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159


> reg.summary$rss
 [1] 36179679 30646560 29249297 27970852 27149899 26194904 25906548 25136930 24814051 24500402 24387345
[12] 24333232 24289148 24248660 24235177 24219377 24209447 24201837 24200700
> reg.summary$bic
 [1]  -90.84637 -128.92622 -135.62693 -141.80892 -144.07143 -147.91690 -145.25594 -147.61525 -145.44316
[10] -143.21651 -138.86077 -133.87283 -128.77759 -123.64420 -118.21832 -112.81768 -107.35339 -101.86391
[19]  -96.30412

> reg.summary$adjr2
 [1] 0.3188503 0.4208024 0.4450753 0.4672734 0.4808971 0.4972001 0.5007849 0.5137083 0.5180572 0.5222606
[11] 0.5225706 0.5217245 0.5206736 0.5195431 0.5178661 0.5162219 0.5144464 0.5126097 0.5106270
> reg.summary$cp
 [1] 104.281319  50.723090  38.693127  27.856220  21.613011  14.023870  13.128474   7.400719   6.158685
[10]   5.009317   5.874113   7.330766   8.888112  10.481576  12.346193  14.187546  16.087831  18.011425
[19]  20.000000
```

选择最优模型







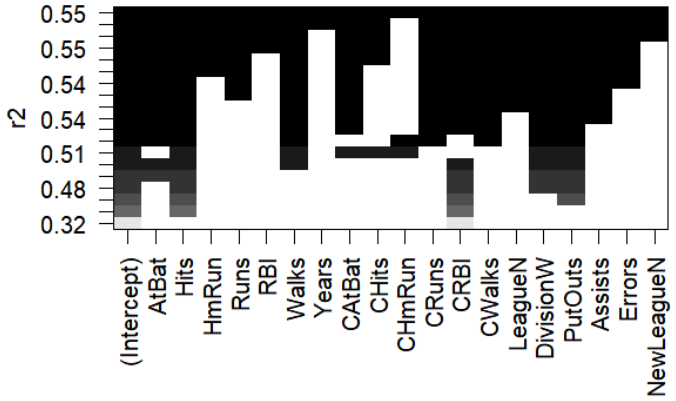


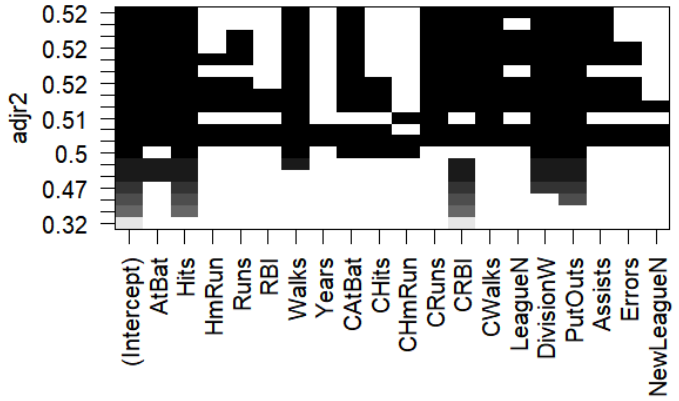

```
> plot(reg.summary$rss,xlab="Number of Variables",ylab="RSS",type="l")
> plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted RS
q",type="l")
> which.max(reg.summary$adjr2)
[1] 11
> points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)
> plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type='l')
> which.min(reg.summary$cp)
[1] 10
> points(10,reg.summary$cp[10],col="red",cex=2,pch=20)
> which.min(reg.summary$bic)
[1] 6
> plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type='l')
> points(6,reg.summary$bic[6],col="red",cex=2,pch=20)
```

which.max以及which.min挑选相关最优结果。

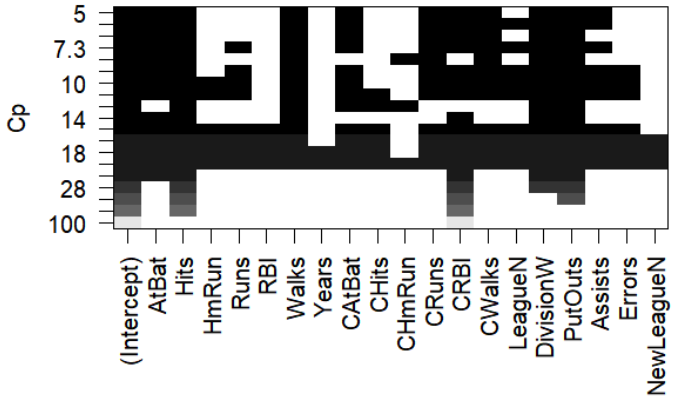
选择最优模型

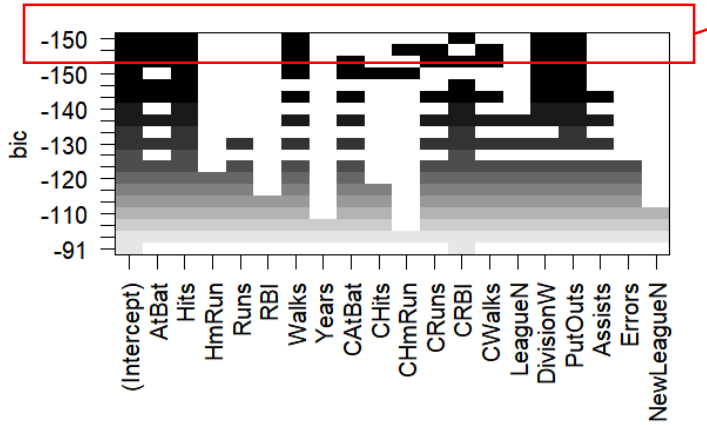
按照BIC、Cp、调整R^2或AIC排序后，输出包含给定个数预测变量的最优模型所含变量的情况。






```
> plot(regfit.full,scale="r2")
> plot(regfit.full,scale="r2")
> plot(regfit.full,scale="adjr2")
> plot(regfit.full,scale="Cp")
> plot(regfit.full,scale="bic")
```



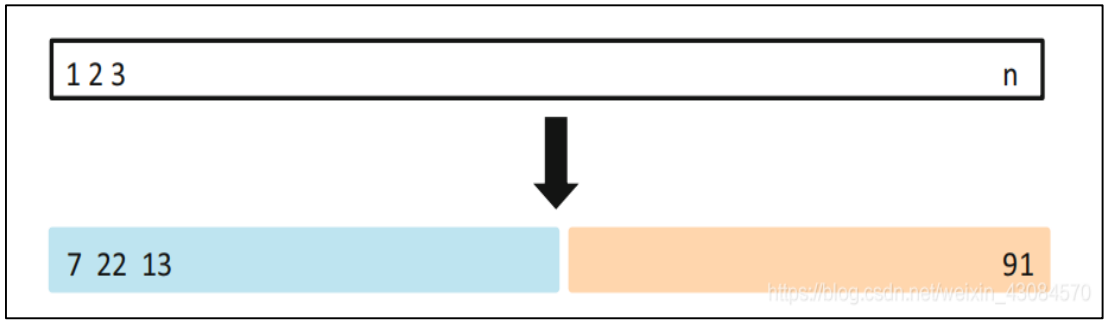


BIC指标最小的模型所包含的变量。

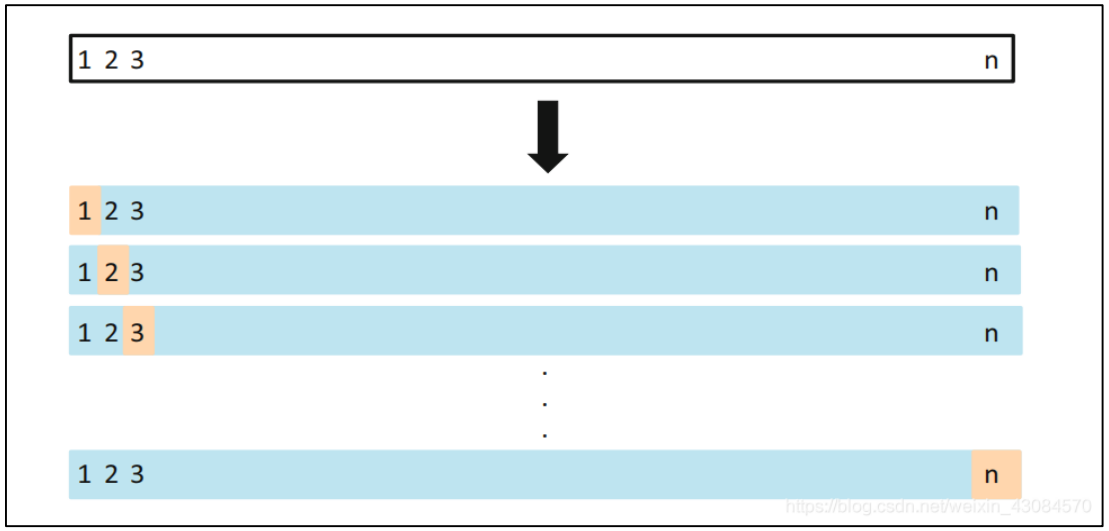
每个图像的第一行黑色方块表示根据相应统计指标选择的最优模型所含的变量。

选择最优模型

验证集方法



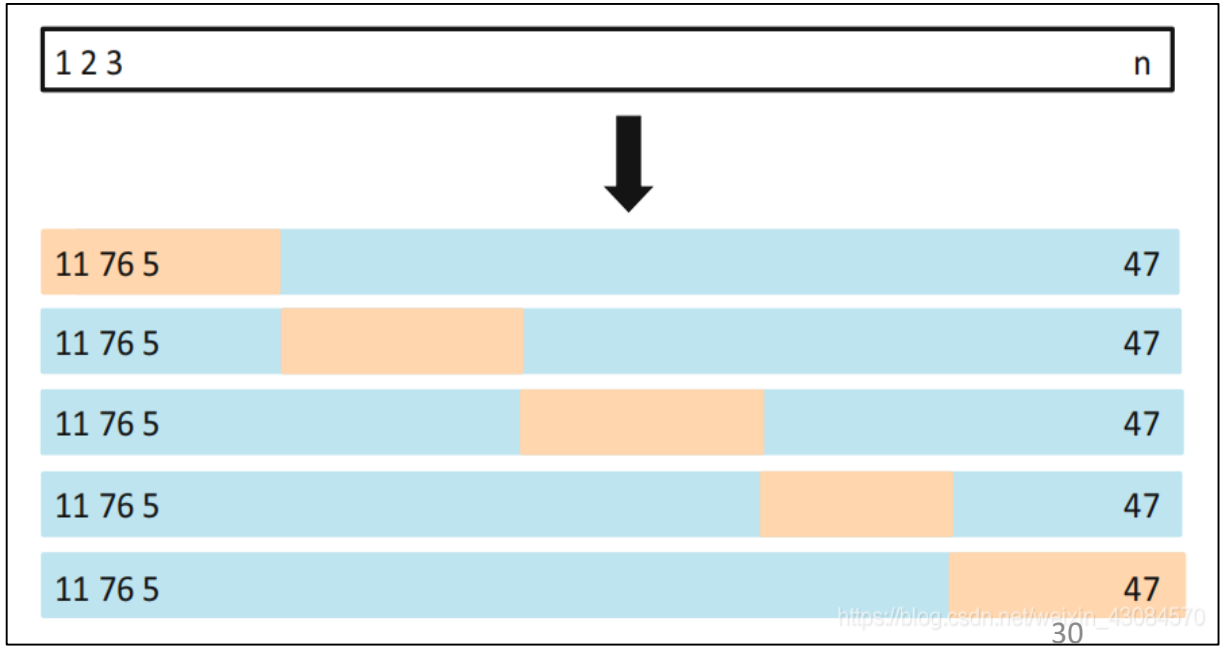
留一交叉验证



优势：

- 提供了测试误差的直接估计，并且对真实的潜在模型有较少的假设。
- 适用范围广，即使是在难以精确确定模型自由度(如模型中的预测数)或难以估计误差方差σ^2的情况下。

k折交叉验证（此图k=5）



选择最优模型

使用Credict数据集， 验证误差通过随机选择四分之三的观测作为训练集，其余观测作为验证集计算得到。交叉验证误差按k=10折进行计算。

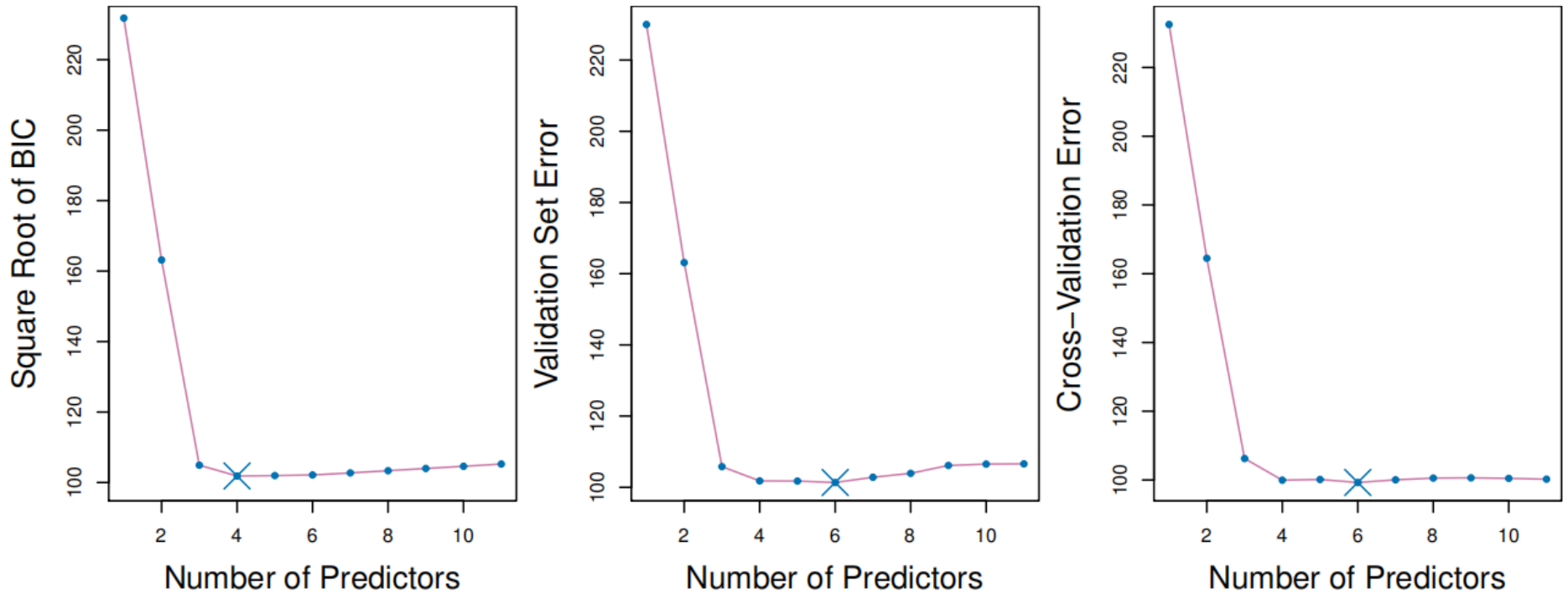


图1.3 显示Credit 数据集包含d个变量的最优模型三种统计量的值，d范围在1-11之间。

选择最优模型

验证集方法先将数据分为训练集和测试集。生成随机种子保证获得同样的训练集和测试集

```
> set.seed(1)
> train=sample(c(TRUE,FALSE), nrow(Hitters),rep=TRUE)
> test=(!train)
```

在训练集上完成模型最优子集选择，直接调用数据训练集

```
> regfit.best=regsubsets(Salary~.,data=Hitters[train,],nvmax=19)
```

计算在不同模型大小的情况下，最优模型的验证集误差。使用测试数据生成回归矩阵

```
> test.mat=model.matrix(Salary~.,data=Hitters[test,])
```

选择最优模型

```
> test.mat=model.matrix(Salary~.,data=Hitters[test,])
```

查看test.mat

```
> head(test.mat)
                  (Intercept) AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI CWalks LeagueN
-Alvin Davis                1   479  130    18   66  72    76     3   1624   457     63   224  266    263       0
-Alfredo Griffin            1   594  169     4   74  51    35    11   4408  1133     19   501  336    194       0
-Andre Thornton             1   401   92    17   49  66    65    13   5206  1332    253   784  890    866       0
-Alan Trammell              1   574  159    21  107  75    59    10   4631  1300     90   702  504    488       0
-Buddy Biancalana           1   190   46     2   24   8    15     5    479   102      5    65   23     39       0
-Bruce Bochy                1   127   32     8   16  22    14     8    727   180     24    67   82     56       1
                  DivisionW PutOuts Assists Errors NewLeagueN
-Alvin Davis              1     880      82     14          0
-Alfredo Griffin          1     282     421     25          0
-Andre Thornton           0       0       0      0          0
-Alan Trammell            0     238     445     22          0
-Buddy Biancalana         1     102     177     16          0
-Bruce Bochy              1     202      22      2          1
```

```
> fix(test.mat)
```

	row.names	(Intercept)	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	LeagueN	DivisionW
1	-Alvin Davis	1	479	130	18	66	72	76	3	1624	457	63	224	266	263	0	1
2	-Alfredo Griffin	1	594	169	4	74	51	35	11	4408	1133	19	501	336	194	0	1
3	-Andre Thornton	1	401	92	17	49	66	65	13	5206	1332	253	784	890	866	0	0
4	-Alan Trammell	1	574	159	21	107	75	59	10	4631	1300	90	702	504	488	0	0
5	-Buddy Biancalana	1	190	46	2	24	8	15	5	479	102	5	65	23	39	0	1
6	-Bruce Bochy	1	127	32	8	16	22	14	8	727	180	24	67	82	56	1	1
7	-Barry Bonds	1	413	92	16	72	48	65	1	413	92	16	72	48	65	1	0
8	-Bobby Bonilla	1	426	109	3	55	43	62	1	426	109	3	55	43	62	0	1

选择最优模型

使用循环语句进行参数估计与预测

```
> val.errors=rep(NA,19)
> for(i in 1:19){
+     coefi=coef(regfit.best,id=i)
+     pred=test.mat[,names(coefi)]%*%coefi
+     val.errors[i]=mean((Hitters$Salary[test]-pred)^2)
+ }
```

第i次循环提取模型大小为i时最优模型的参数估计结果

参数估计向量乘以测试集生成的回归设计矩阵

计算出预测值和测试集的MSE。

通过验证集方法可得到最优模型含有7个预测变量[注]

```
> val.errors
 [1] 164377.3 144405.5 152175.7 145198.4 137902.1 139175.7 126849.0 136191.4 132889.6
[10] 135434.9 136963.3 140694.9 140690.9 141951.2 141508.2 142164.4 141767.4 142339.6
[19] 142238.2
> which.min(val.errors)
[1] 7
> coef(regfit.best,7)
 (Intercept)        AtBat         Hits        Walks        CRuns       CWalks
  67.1085369   -2.1462987    7.0149547    8.0716640    1.2425113   -0.8337844
   DivisionW      PutOuts
-118.4364998    0.2526925
```

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

选择最优模型

可根据以上步骤编写预测函数

```
> predict.regsubsets=function(object,newdata,id,...){
+   form=as.formula(object$call[[2]])
+   mat=model.matrix(form,newdata)
+   coefi=coef(object,id=id)
+   xvars=names(coefi)
+   mat[,xvars]%*%coefi
+   }
```

对整个数据集使用最优子集选择，选出最优的7变量模型。

```
> regfit.best=regsubsets(Salary~.,data=Hitters,nvmax=19)
> coef(regfit.best,7)
 (Intercept)         Hits        Walks       CAtBat        CHits
  79.4509472    1.2833513    3.2274264   -0.3752350    1.4957073
      CHmRun    DivisionW      PutOuts
   1.4420538 -129.9866432    0.2366813
```

基于整个数据集建立的最优十变量模型包含的变量不同于基于训练集建立的最优7变量模型。

压缩估计

压缩估计：对系数进行约束或加罚的技巧对包含p个预测变量的模型进行拟合，即将系数估计值往零的方向压缩。这种方法通过压缩系数估计值，显著减少了估计量方差。

岭回归（ridge regression）

Ridge

目标：使 $J(\theta) = MSE(y, \hat{y}; \theta) + \alpha \frac{1}{2} \sum_{i=1}^{n} \theta_i^2$

lasso

LASSO

目标：使 $J(\theta) = MSE(y, \hat{y}; \theta) + \alpha \sum_{i=1}^{n} |\theta_i|$

岭回归

通过最小化如下函数对 $\beta_0, \beta_1, \ldots, \beta_p$ 进行估计来拟合最小二乘回归：

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

岭回归系数估计值通过最小化下式得到：

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2$$

压缩惩罚

与最小二乘一样，岭回归通过使RSS变小来寻找适合数据的系数估计。

$\lambda \geq 0$是一个调节参数，选择合适的 λ十分重要。

岭回归

λ=0，岭回归系数估计结果与最小二乘估计结果相同。　λ=∞, 岭回归系数估计值是一个零向量。

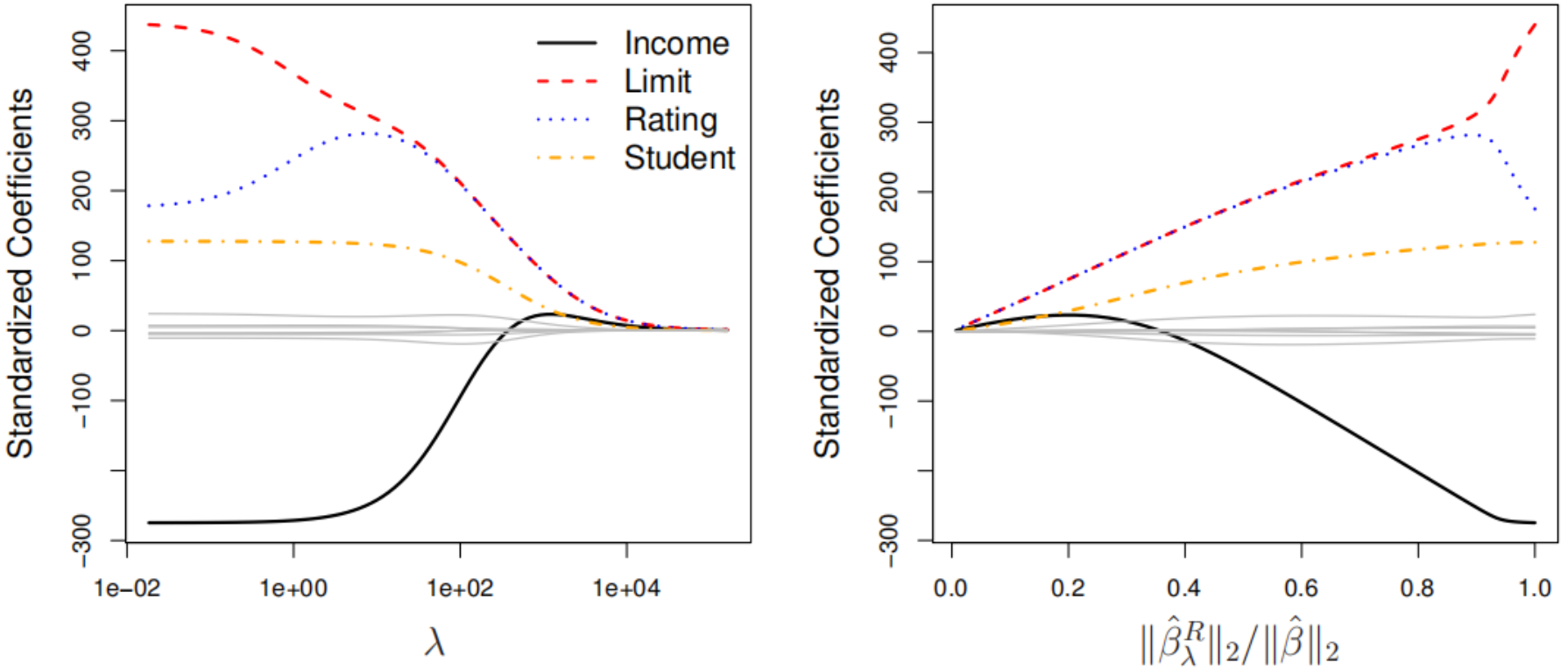


图2.1　Credict 数据集标准化后岭回归系数随着λ和 $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ 的变化情况。

岭回归

由于岭回归公式中系数平方和项的存在，$X_j\hat{\beta}^R_{j,\lambda}$ 的值不只取决于λ,也取决于第 j 个预测变量的尺度。

对预测变量进行标准化，使所有变量都具有同一尺度：

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

式中，分母为第j个估计变量的标准偏差估计值，故标准化后的变量的标准差为1.

岭回归

- 与最小二乘相比，岭回归的优势在于它综合权衡了误差与方差。
- 随着λ的增加，岭回归拟合结果的光滑度降低，虽然方差降低，但偏差增加。

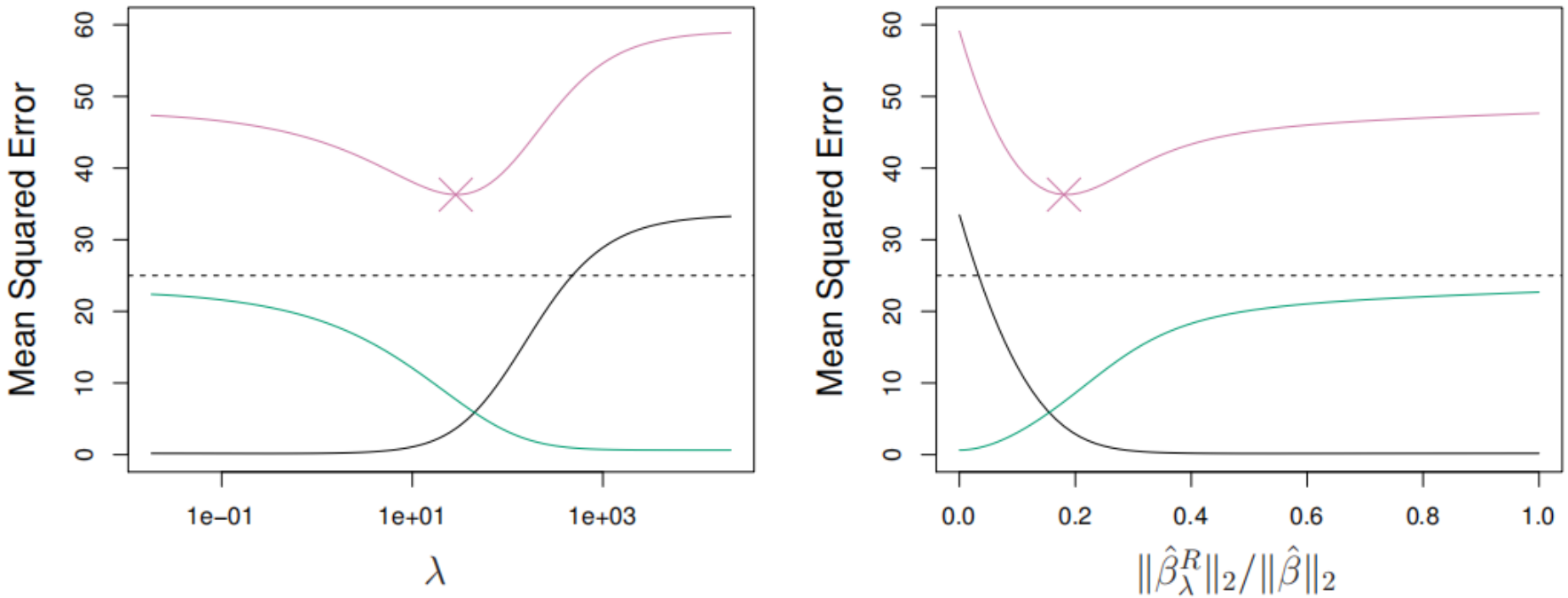


图2.2 岭回归在模拟测试集上预测结果的偏差的平方（黑线），方差（绿线）和测试均方误差（紫线）的随着λ和 $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ 的变化情况

lasso

- 岭回归劣势：最终模型包含全部的p个变量，当变量个数p非常大时，不便于模型解释。
- lasso是一种相对较新的替代岭回归的方法，其系数 $\hat{\beta}_\lambda^L$ 通过下式求解最小值得到：

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

$$\|x\|_p=\left(\sum_{i=1}^{n}|x_i|^p\right)^{\frac{1}{p}}$$

Lp范数

Ridge $\sum_{i=1}^{n}\beta_i^2$ L2正则项

lasso $\sum_{i=1}^{n}|\beta_i|$ L1正则项

lasso

- 岭回归劣势：最终模型包含全部的p个变量，当变量个数p非常大时，不便于模型解释。
- lasso是一种相对较新的替代岭回归的方法，其系数 $\hat{\beta}_\lambda^L$ 通过下式求解最小值得到：

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

- lasso也将系数估计值往0的方向进行缩减，选择一个合适的λ值十分重要。
- lasso可完成变量选择，更易于解释建立的模型。
- lasso得到了稀疏模型，只包含所有变量的一个子集的模型。

lasso

- λ=0，lasso与最小二乘等价。当λ足够大时，lasso估计得到一个零模型，所有系数估计值均为0。
- 根据不同的λ取值，lasso可得到包含不同变量个数的模型，而岭回归得到的模型始终包含所有变量，虽然系数估计值会随着λ变化。

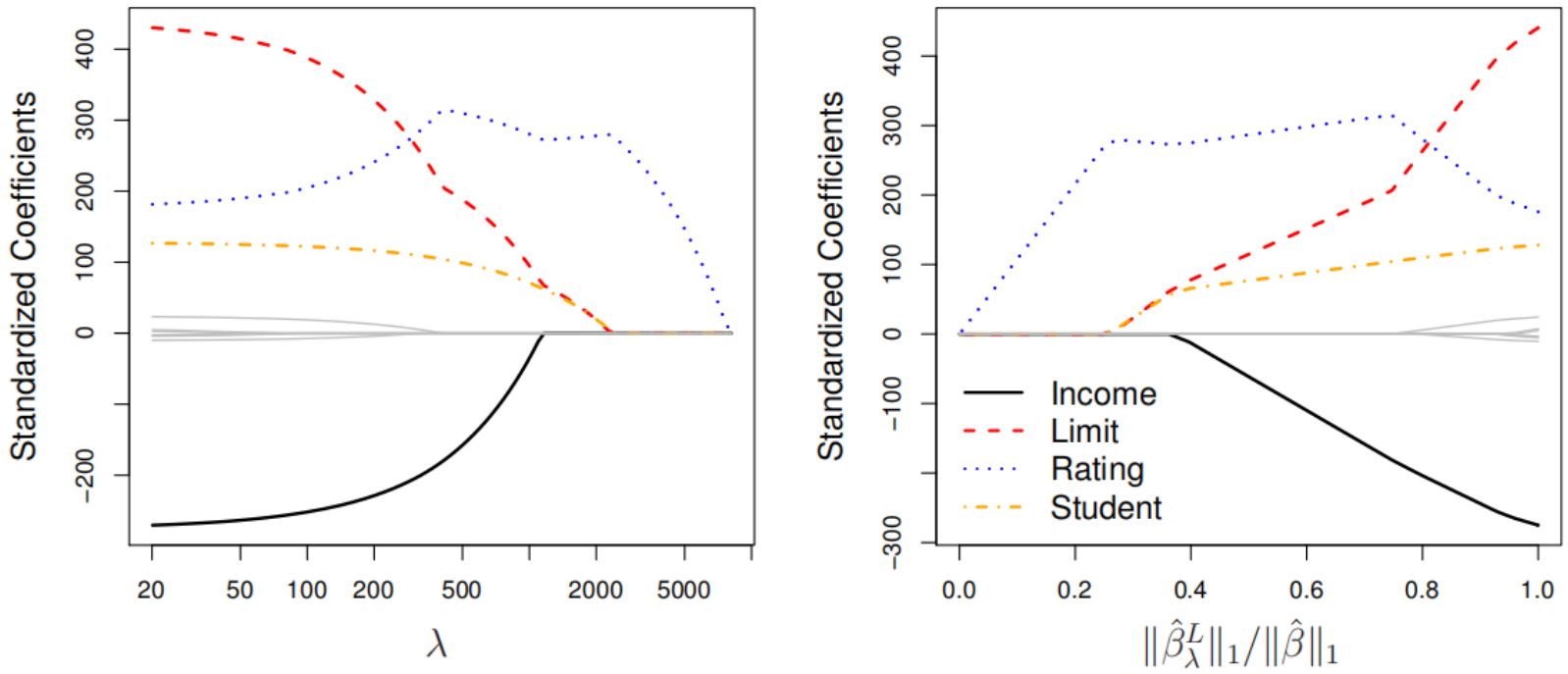


图2.3 Credict数据集标准化lasso系数随着λ和 $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ 的变化情况

lasso

岭回归和lasso的其他形式

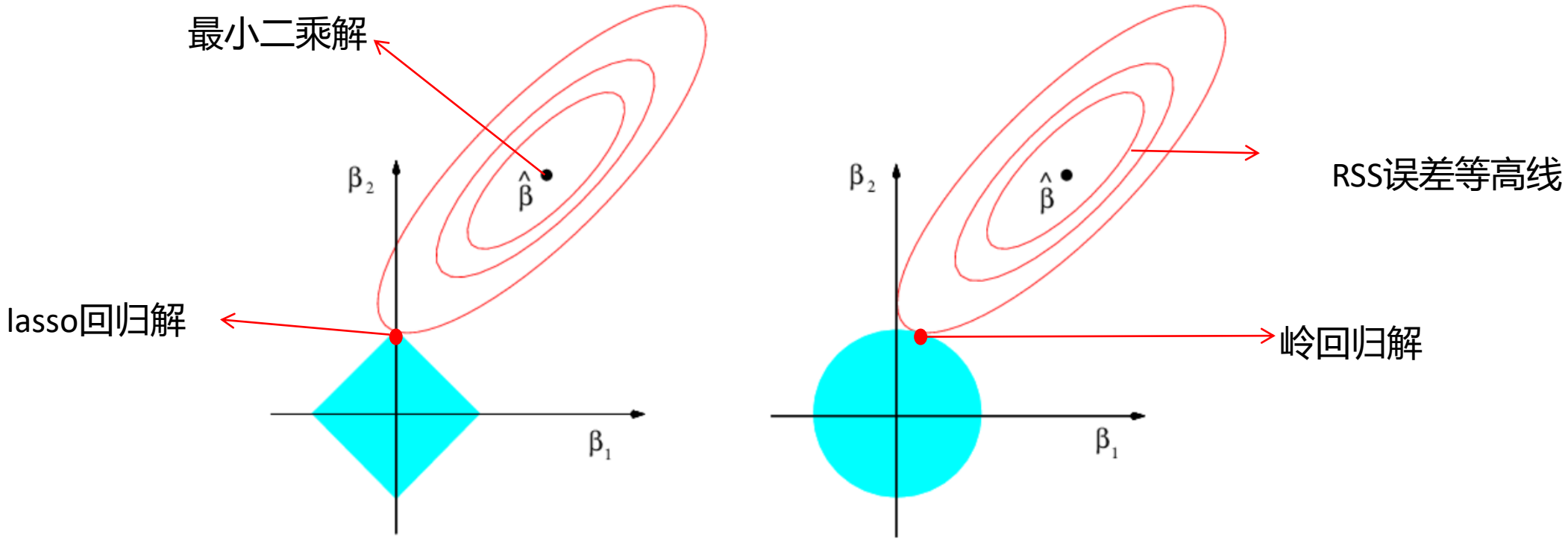
最小二乘法 $\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$ 无限制

Lasso方法 $\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$ $\sum_{j=1}^{p} |\beta_j| \leq s$

领回归方法 $\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$ $\sum_{j=1}^{p} \beta_j^2 \leq s$

lasso

最小二乘估计系数记为$\hat{\beta}$ 菱形与圆形分别代表lasso和岭回归的上述式中的限制条件区域，图中为p=2的简单情况。



- 当p=3，岭回归的限制条件区域将会变成一个球体，而lasso的限制区域将变成多面体。
- 当p>3,岭回归的限制条件区域将会变成超球面，而lasso的将变成多面体。

lasso

对比lasso和岭回归

（1）右图显示岭回归的方差稍小于lasso的方差，导致其最小均方误差同样稍小于lasso。lasso潜在假设一些系数真值为零，这时从误差上看岭回归表现比lasso好。

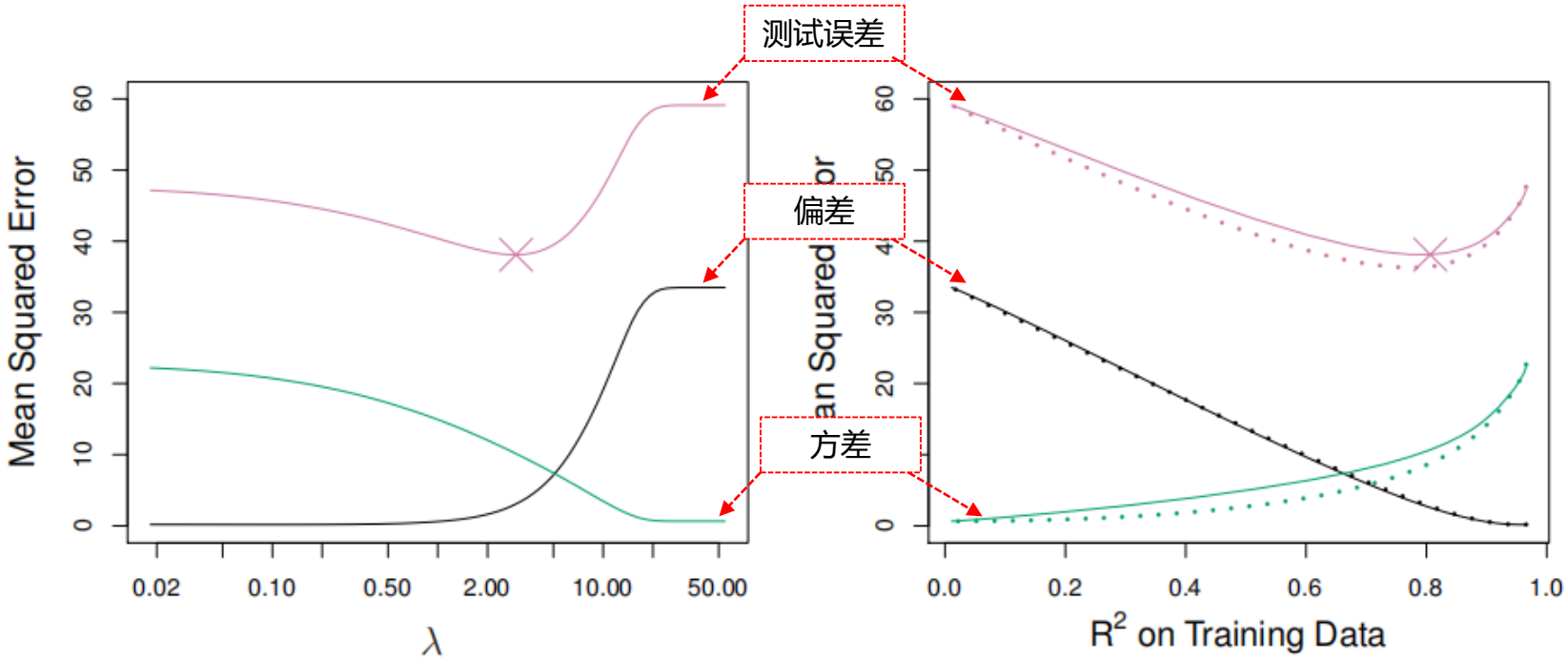


图2.4 左：**模拟数据集**上lasso得到的偏差平方和（黑）、方差（绿）以及测试均方误差（紫）。

右：对比lasso（实线）和岭回归（虚线）的偏差平方和、方差和测试均方误差。

两图均根据训练数据集的R^2作图，图中交点代表了均方误差最小的lasso模型

lasso

对比lasso和岭回归

（2）下图数据中响应变量只是45个预测变量中2个变量的函数。在这种**模拟数据集**情况下，lasso在偏差、方差和均方误差等方面表现要好于岭回归。

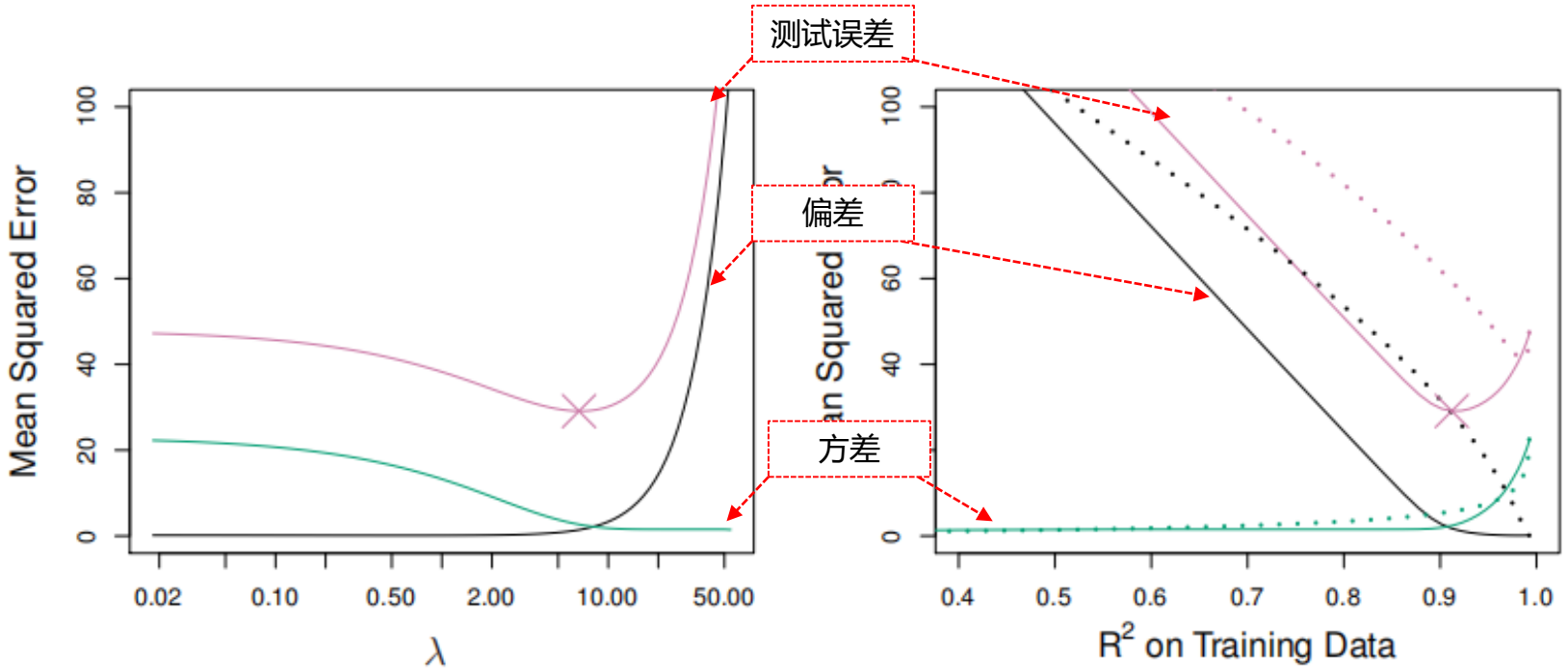


图2.5 左：lasso的偏差平方和（黑）、方差（绿）以及测试均方误差（紫）

右：对比lasso（实线）和岭回归（虚线）的偏差平方和、方差和测试均方误差。

两图均根据训练数据集的R^2作图，图中交点代表了均方误差最小的lasso模型

lasso

对比lasso和岭回归

（3）从贝叶斯视角理解：对于回归，贝叶斯理论假设回归系数向量β具有先验分布 $p(\beta)=(\beta_0,\beta_1,\cdots,\beta_p)^T$

将先验分布与似然函数相乘可以得到后验分布。形式如下：

$$p(\beta \mid X,Y) \propto f(Y \mid X,\beta)\,p(\beta \mid X) = f(Y \mid X,\beta)\,p(\beta)$$

假设普通线性模型为：

$$Y = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p + \varepsilon$$

进一步假设：

$$p(\beta) = \prod_{j=1}^{p} g(\beta_i)$$

g 是密度函数

lasso

对比lasso和岭回归

- 最小二乘法中估计参数不受任何约束，会比较奔放，容易过拟合。而lasso和岭回归引入先验分布，做了约束有减少过拟合的功效。

- 岭回归是假设密度函数g(x)服从高斯分布，比较平滑，没有稀疏性的作用，但是性能会好一点。

- Lasso是假设密度函数g(x)服从Laplace分布，而根据分布Laplace倾向舍弃影响较小的参数（分布中间取为0比较陡峭）。

$$p(\beta)=\prod_{j=1}^{p} g(\beta_i)$$

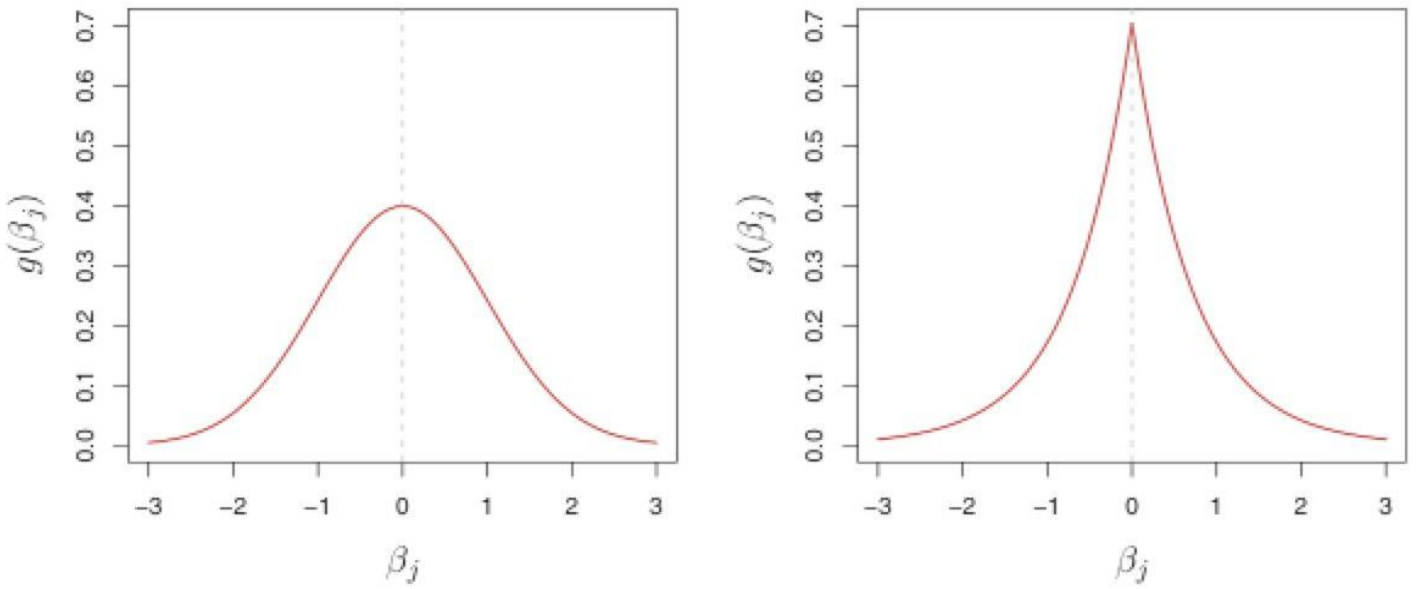


图2.6 在高斯分布假设下，岭回归系数的后验形式（左）；在双指数先验分布假设下lasso系数的后验形式（右）

lasso

对比lasso和岭回归

- 在性质上相似：随着λ增大，方差减小而偏差平方和增大。
- 当最小二乘估计出现较大方差时，lasso可得到更精确的预测结果。
- 当一小部分预测变量是真实有效的而其他预测变量系数非常小或者等于零时，lasso更好。
- 当响应变量是很多预测变量的函数并且这些变量系数大致相等，岭回归较好。
- 对于真实的数据集，与响应变量相关的变量个数是无法先验知道的，可用交叉验证等技术来确定哪种方法对特定数据集更好。

选择调节参数

通过交叉验证方法，调节参数 λ 或者是限制条件 s，确定最优模型。下图为用岭回归对**Credit数据集**进行留一交叉验证的结果。

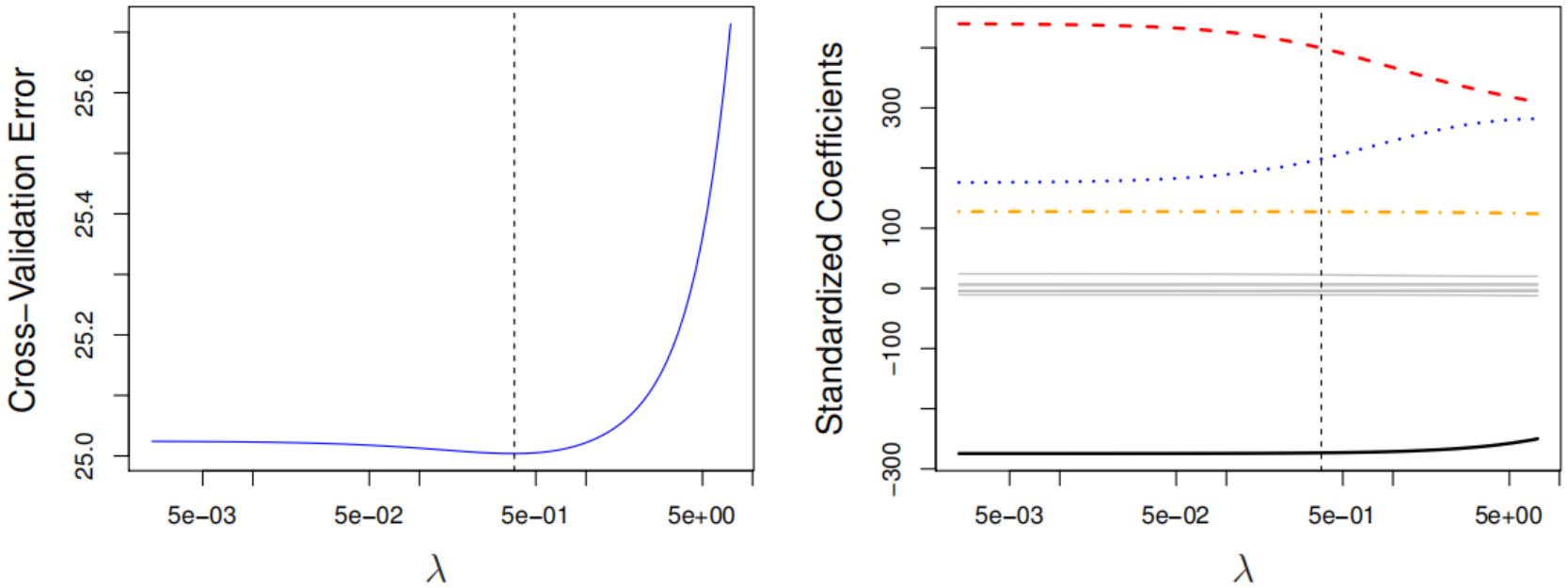


图2.7　左：对**Credict数据集**用岭回归的交叉验证误差，λ取各种不同值。
右：系数估计作为λ的函数。垂直虚线表示根据交叉验证选择的参数。

选择调节参数

下图为用lasso对稀疏模拟数据集进行10折交叉验证的结果。

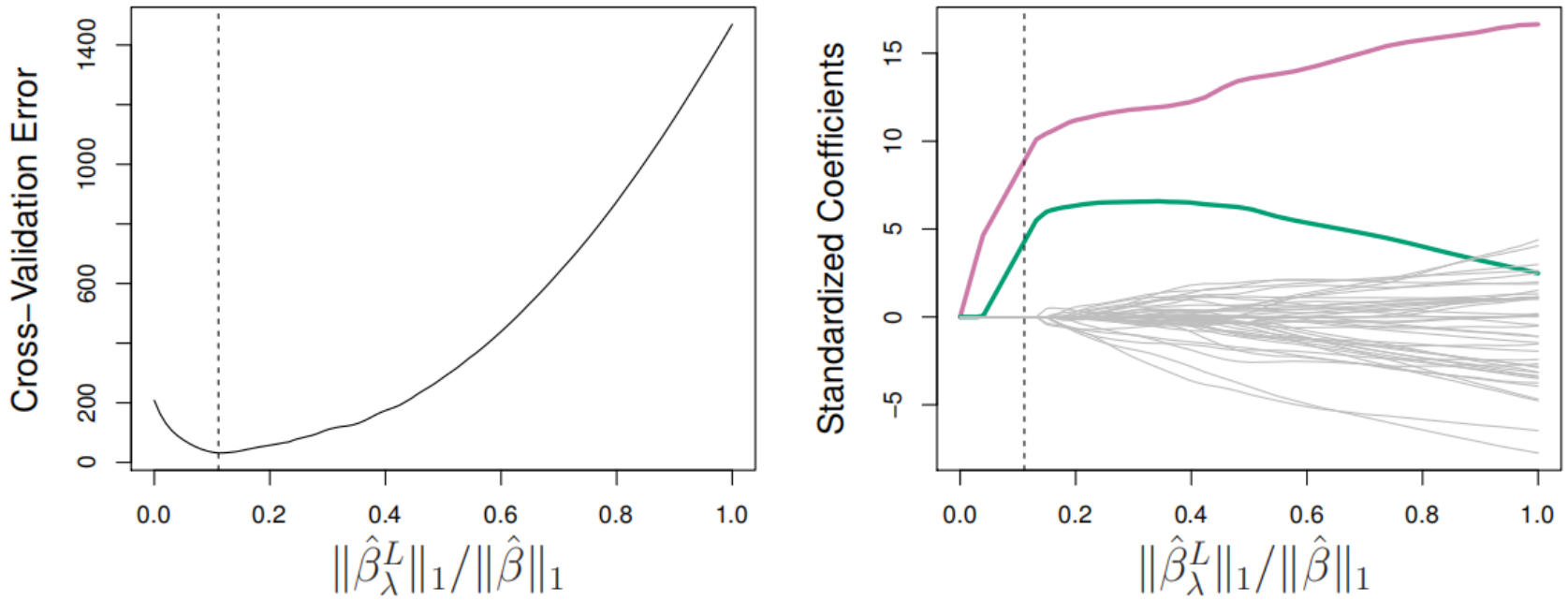


图 2.8 左：用lasso对稀疏模拟数据进行10折交叉验证的均方误差。
右：相应的lasso系数。垂直虚线表示使得交叉验证误差最小的系数。

岭回归-R实验

glmnet( )函数可拟合岭回归和lasso等模型。基于Hitters数据集使用岭回归和lasso模型预测Salary变量。

```
> x=model.matrix(Salary~.,Hitters)[,-1]
> y=Hitters$Salary
> library(glmnet)
> grid=10^seq(10,-2,length=100)
> ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
```

alpha=0 ： 拟合岭回归

alpha=1 ： 拟合lasso回归

- 若采用默认设置：函数自动选择参数λ值的范围进行岭回归。也可设置其他范围。

参数λ的每个取值对应一个岭回归的系数向量，在R语言中可用coef ( ) 函数来提取系数向量矩阵。

```
> dim(coef(ridge.mod))
[1]  20 100
```

岭回归-R实验

使用l_2范数，λ=11498时系数估计结果：

```
> ridge.mod$lambda[50]
[1] 11497.57
> coef(ridge.mod)[,50]
  (Intercept)         AtBat          Hits         HmRun          Runs           RBI
407.356050200   0.036957182   0.138180344   0.524629976   0.230701523   0.239841459
        Walks         Years        CAtBat         CHits        CHmRun         CRuns
  0.289618741   1.107702929   0.003131815   0.011653637   0.087545670   0.023379882
         CRBI        CWalks       LeagueN     DivisionW       PutOuts       Assists
  0.024138320   0.025015421   0.085028114  -6.215440973   0.016482577   0.002612988
       Errors    NewLeagueN
 -0.020502690   0.301433531
> sqrt(sum(coef(ridge.mod)[-1,50]^2))
[1] 6.360612
```

使用l_2范数，λ=705时系数估计结果：

```
> ridge.mod$lambda[60]
[1] 705.4802
> coef(ridge.mod)[,60]
 (Intercept)         AtBat          Hits         HmRun          Runs           RBI
 54.32519950   0.11211115   0.65622409   1.17980910   0.93769713   0.84718546
       Walks         Years        CAtBat         CHits        CHmRun         CRuns
  1.31987948   2.59640425   0.01083413   0.04674557   0.33777318   0.09355528
        CRBI        CWalks       LeagueN     DivisionW       PutOuts       Assists
  0.09780402   0.07189612  13.68370191 -54.65877750   0.11852289   0.01606037
      Errors    NewLeagueN
 -0.70358655   8.61181213
> sqrt(sum(coef(ridge.mod)[-1,60]^2))
[1] 57.11001
```

岭回归-R实验

使用predict ( ) 函数可以完成多种任务。可获得新λ值对应的岭回归系数，如λ=50.

```
> predict(ridge.mod,s=50,type="coefficients")[1:20,]
  (Intercept)          AtBat           Hits          HmRun           Runs
 4.876610e+01 -3.580999e-01  1.969359e+00 -1.278248e+00  1.145892e+00
         Walks          Years         CAtBat          CHits         CHmRun
 2.716186e+00 -6.218319e+00  5.447837e-03  1.064895e-01  6.244860e-01
         CRBI         CWalks        LeagueN      DivisionW       PutOuts
 2.186914e-01 -1.500245e-01  4.592589e+01 -1.182011e+02  2.502322e-01
       Errors     NewLeagueN
-3.278600e+00 -9.496680e+00
```

岭回归-R实验

分割为训练集与测试集，首先生成一个随机种子，保证实验可重复性。

```
> set.seed(1)
> train=sample(1:nrow(x), nrow(x)/2)
> test=(-train)
> y.test=y[test]
```

基于训练集建立岭回归模型，并计算λ=4时测试集的MSE。

```
> ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh=1e-12)
> ridge.pred=predict(ridge.mod,s=4,newx=x[test,])#newx=x[test,]获得测试集的预测值
> mean((ridge.pred-y.test)^2)
[1] 142199.2
```

```
> mean((mean(y[train])-y.test)^2)
[1] 224669.9
```

拟合只含有截距项模型，模型对测试集中的每个观测给出的预测值为训练集数据的均值。结果表明远大于λ=4时拟合的岭回归模型测试集的MSE。

```
> ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])
> mean((ridge.pred-y.test)^2)
[1] 224669.8
```

使用非常大的λ值拟合岭回归模型以获得同样MSE。最小二乘回归是当λ=0时的岭回归模型。

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

岭回归-R实验

使用交叉验证选择调节参数λ。在R中内置的交叉验证函数为cv.glmnet()。默认为十折交叉验证。

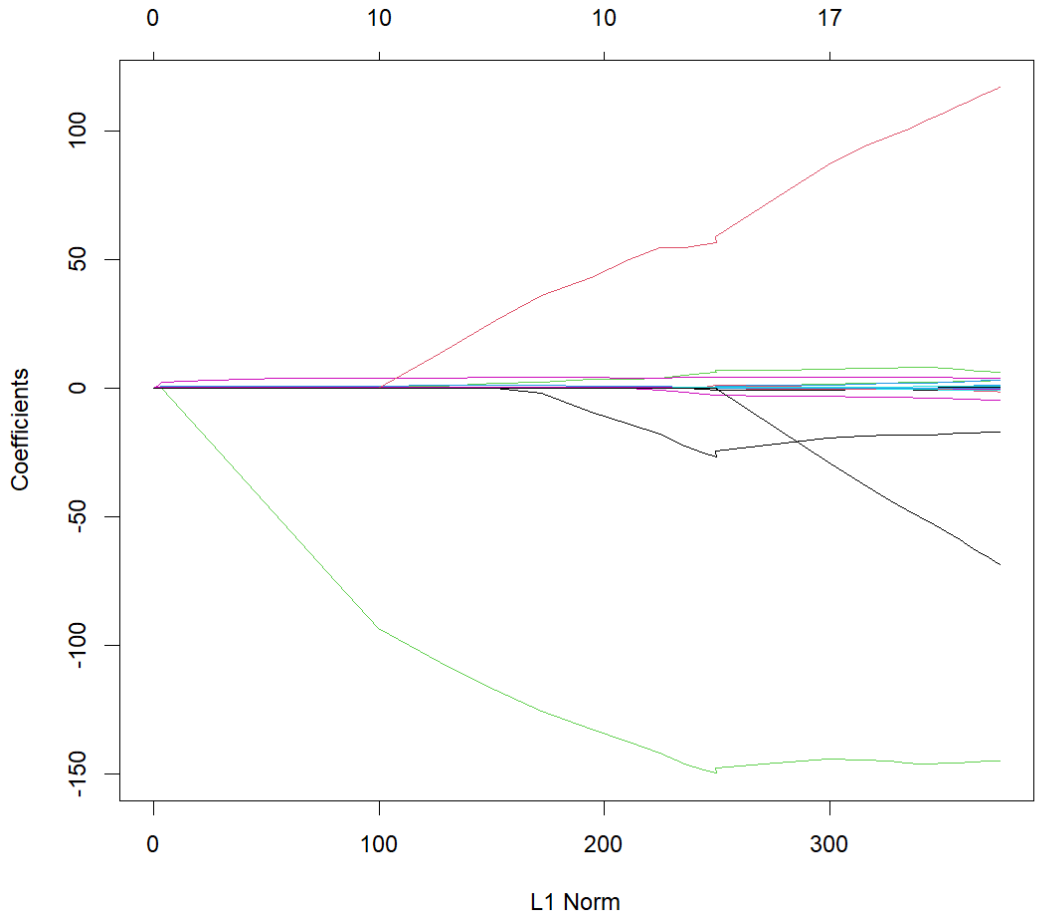
```
> set.seed(1)
> cv.out=cv.glmnet(x[train,],y[train],alpha=0)
> plot(cv.out)
> bestlam=cv.out$lambda.min
> bestlam
[1] 326.0828
> ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])
> mean((ridge.pred-y.test)^2)
[1] 139856.6
> out=glmnet(x,y,alpha=0)
> predict(out,type="coefficients",s=bestlam)[1:20,]
 (Intercept)        AtBat         Hits        HmRun         Runs
 15.44383120   0.07715547   0.85911582   0.60103106   1.06369007
         RBI        Walks        Years       CAtBat        CHits
  0.87936105   1.62444617   1.35254778   0.01134999   0.05746654
      CHmRun        CRuns         CRBI        CWalks       LeagueN
  0.40680157   0.11456224   0.12116504   0.05299202  22.09143197
   DivisionW      PutOuts      Assists       Errors   NewLeagueN
-79.04032656   0.16619903   0.02941950  -1.36092945   9.12487765
```

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

lasso-R实验

使用glmnet( ) 函数中拟合lasso模型，除alpha=1外其他语句与上述拟合岭回归模型语句一样。

```
> lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)
> plot(lasso.mod)
```

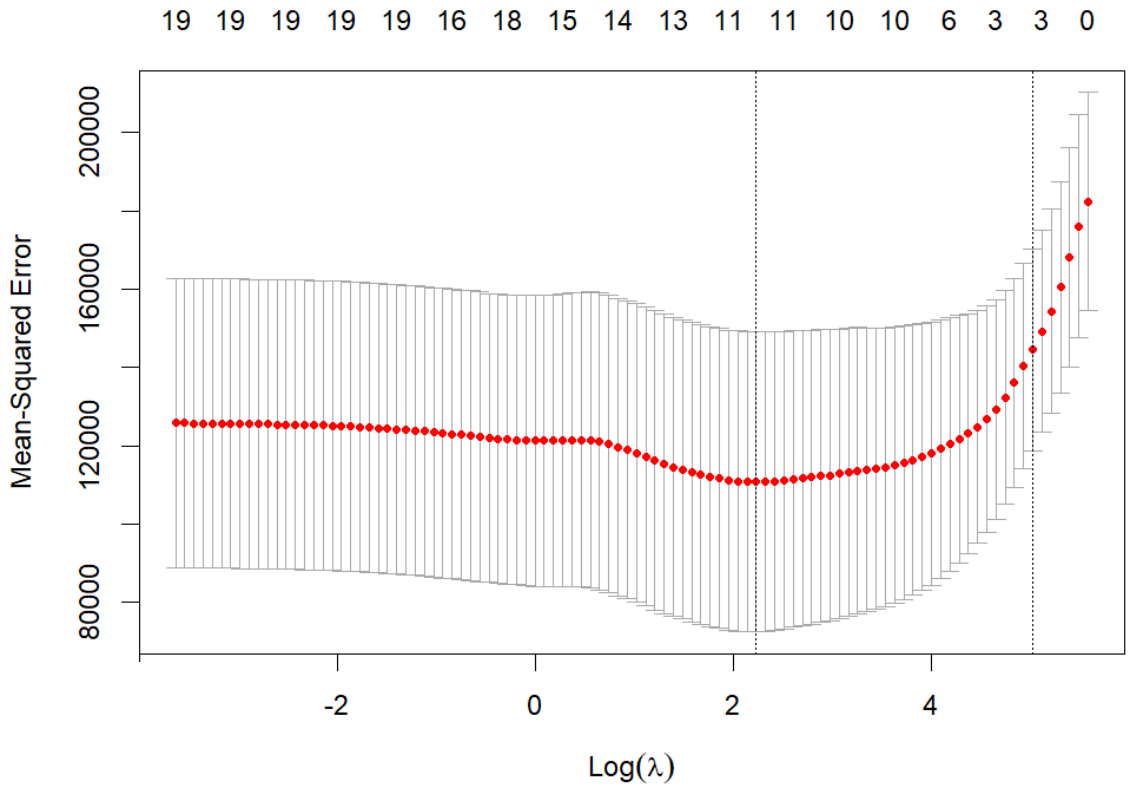


随着调节参数的选择不同，某些预测变量的系数会变为0。 MSE结果明显小于空模型与最小二乘模型。

lasso-R实验

使用交叉验证并计算相应的测试误差。

```
> set.seed(1)
> cv.out=cv.glmnet(x[train,],y[train],alpha=1)
> plot(cv.out)
```



输出结果的测试 MSE 现显小于空模型和最小二乘模型。

```
> bestlam=cv.out$lambda.min
> bestlam
[1] 9.286955
> lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
> mean((lasso.pred-y.test)^2)
[1] 143673.6
```

计算得到交叉验证误差最小的λ值，那么可求出与该值相对应的测试 MSE

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

lasso-R实验

19个预测变量中8个的系数为0，即使用交叉验证选择λ值建立的lasso模型仅包含11个预测变量（不包括beta0)。

```
> out=glmnet(x,y,alpha=1,lambda=grid)
> lasso.coef=predict(out,type="coefficients",s=bestlam)[1:20,]
> lasso.coef
  (Intercept)         AtBat          Hits         HmRun
   1.27479059   -0.05497143    2.18034583    0.00000000
         Runs           RBI         Walks         Years
   0.00000000    0.00000000    2.29192406   -0.33806109
       CAtBat         CHits        CHmRun         CRuns
   0.00000000    0.00000000    0.02825013    0.21628385
         CRBI        CWalks       LeagueN     DivisionW
   0.41712537    0.00000000   20.28615023 -116.16755870
      PutOuts       Assists        Errors    NewLeagueN
   0.23752385    0.00000000   -0.85629148    0.00000000
> lasso.coef[lasso.coef!=0]
  (Intercept)         AtBat          Hits         Walks
   1.27479059   -0.05497143    2.18034583    2.29192406
        Years        CHmRun         CRuns          CRBI
  -0.33806109    0.02825013    0.21628385    0.41712537
     LeagueN     DivisionW       PutOuts        Errors
  20.28615023 -116.16755870    0.23752385   -0.85629148
```

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

降维方法

降维方法：将预测变量进行转换，然后用转换之后的变量拟合最小二乘模型。

令 $Z_1, Z_2, \ldots, Z_M$ 表示M个原始预测变量的线性组合（M<p,共有p个原始变量）即：

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j$$

其中 $\phi_{m1}, \ldots, \phi_{mp}$ 是常数，m=1，2，…，M.可用最小二乘拟合线性回归模型。

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n,$$

即降维可使估计p+1个系数 $\beta_0, \beta_1, \cdots, \beta_p$ 的问题转化为估计M+1个系数 $\theta_0, \theta_1, \ldots, \theta_M$ 的问题，这里M<p,即维度从p+1降到M+1

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{mj} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{mj} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij} \qquad \beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}$$

降维方法

降维过程一般为两个过程：（1）将原始变量转换为 $Z_1, Z_2, \cdots, Z_M$

（2）用M个变量建立模型。

选择 $Z_1, Z_2, \cdots, Z_M$ ，也就是选择 ϕ_{jm} 可以通过多种不同方法，如以下两种方法

主成分回归（principal conmponents regression）

偏最小二乘 (partial least squares)

主成分回归

主成分分析（PCA）：用投影的方法将高维空间压缩到低维。

简要过程：

输入：样本集 X为 p 行n 列的矩阵（p为原始数据维数）

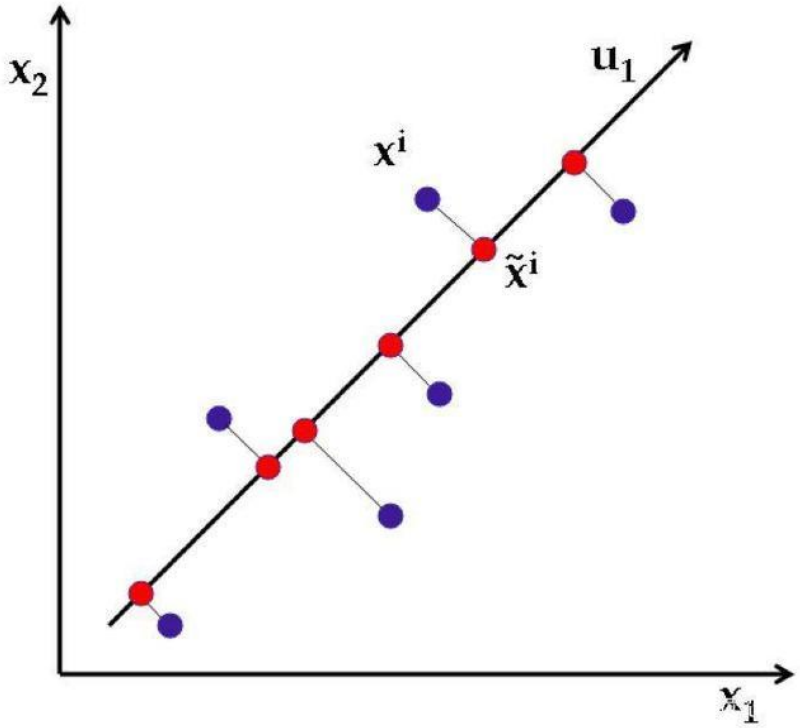
过程：

1：将X的每一行（代表一个属性字段）进行零均值化，即减去这一行的均值；

2：计算样本的协方差矩阵$C=1/n(XX^T)$。

3：求出协方差矩阵的特征值及对应的特征向量；

4：将特征向量按对应特征值大小从上到下按行排列成矩阵，$w_1, w_2, \cdots, w_p$ 取前 M个特征值所对应的特征向量组成矩阵W



输出：投影矩阵 $W = w_1, w_2, \cdots, w_M$，Z=WX 即为降维到 M 维后的数据。

主成分回归

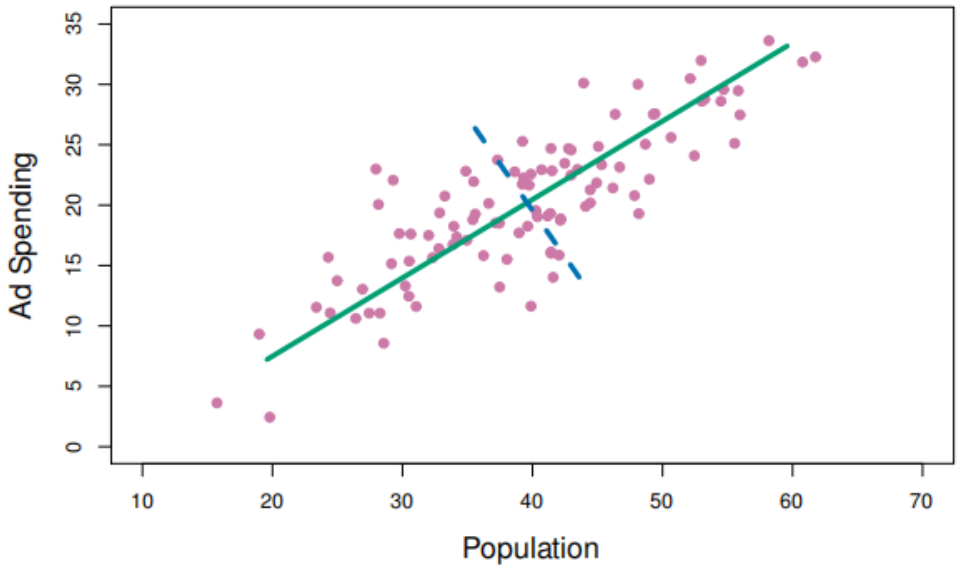


图3.1　圆点表示100个不同城市的人口规模(pop)和广告支出(ad)。绿色实线表示第一个主成分，蓝色虚线表示第二个主成分。

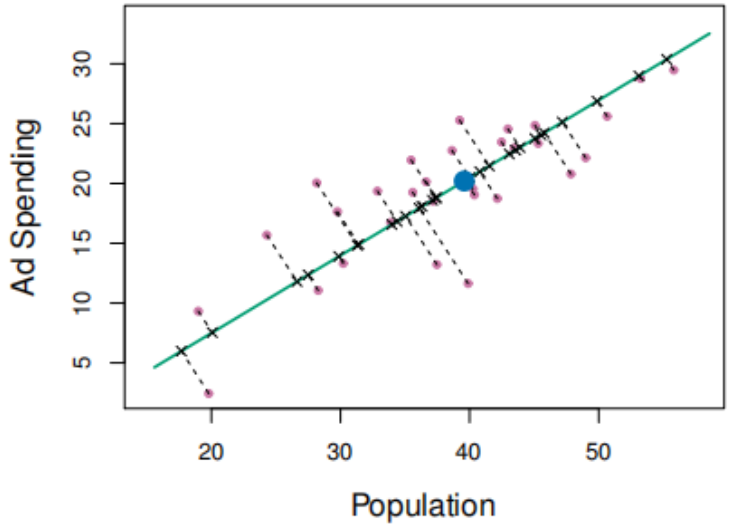


图3.2广告数据子集，第一主成分方向用绿线表示。

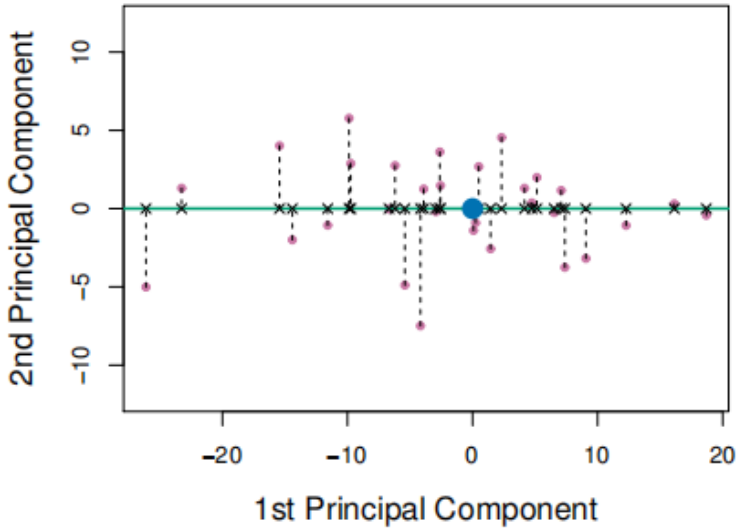


图3.3 经过旋转第一主成分方向同x轴一致。

主成分回归

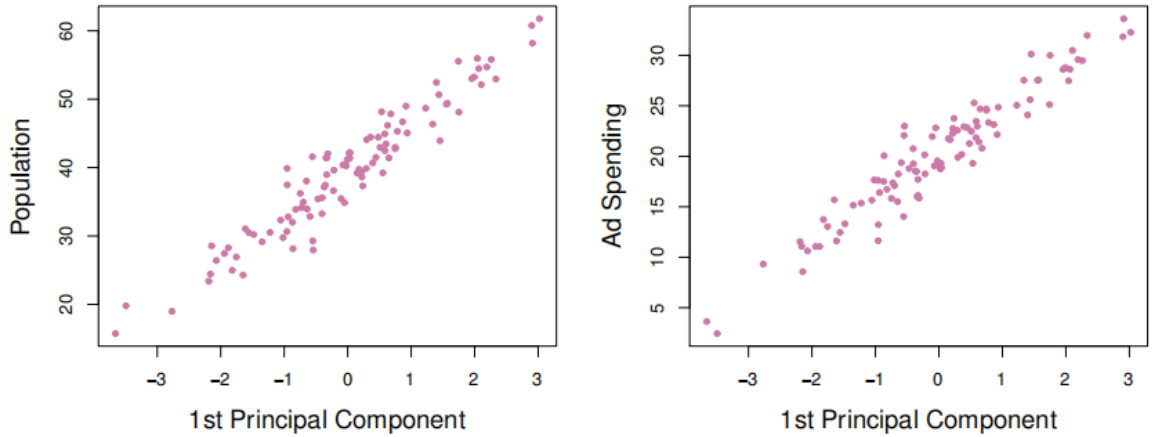


图3.4　第一主成分得分与pop和ad高度相关

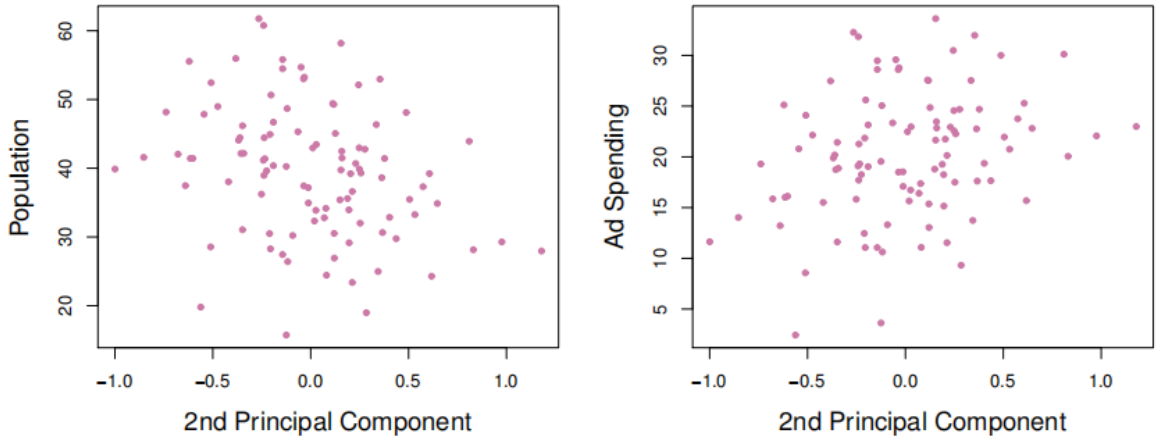
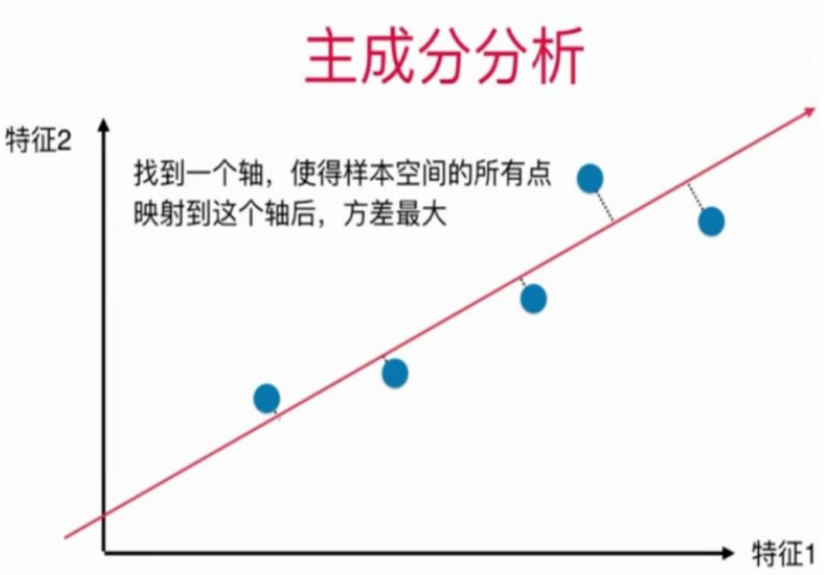


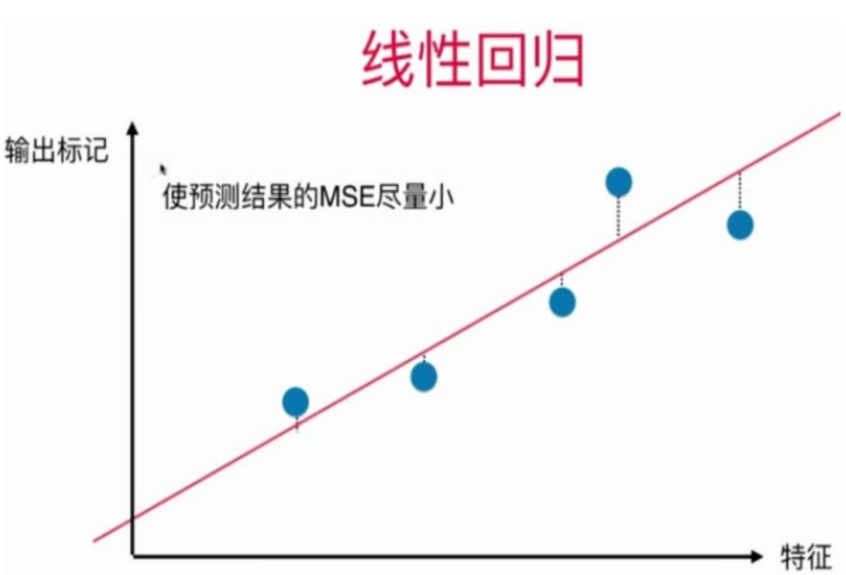
图3.5　第二主成分得分与pop和ad的散点图，相关性较弱

主成分回归

主成分回归（principal conmponents regression，PCR)

即PCA做降维后的特征应用到LR，指构造前M个主成分 $Z_1, Z_2, \ldots, Z_M$ ，然后以这些主成分作为预测变量，用最小二乘拟合线性回归模型。少数的主成分足以解释大部分的数据波动和数据与响应变量之间的关系。





主成分回归

下图显示出与最小二乘相比，三种方法都有显著提升，其中主成分分析与岭回归要比lasso更好一些。

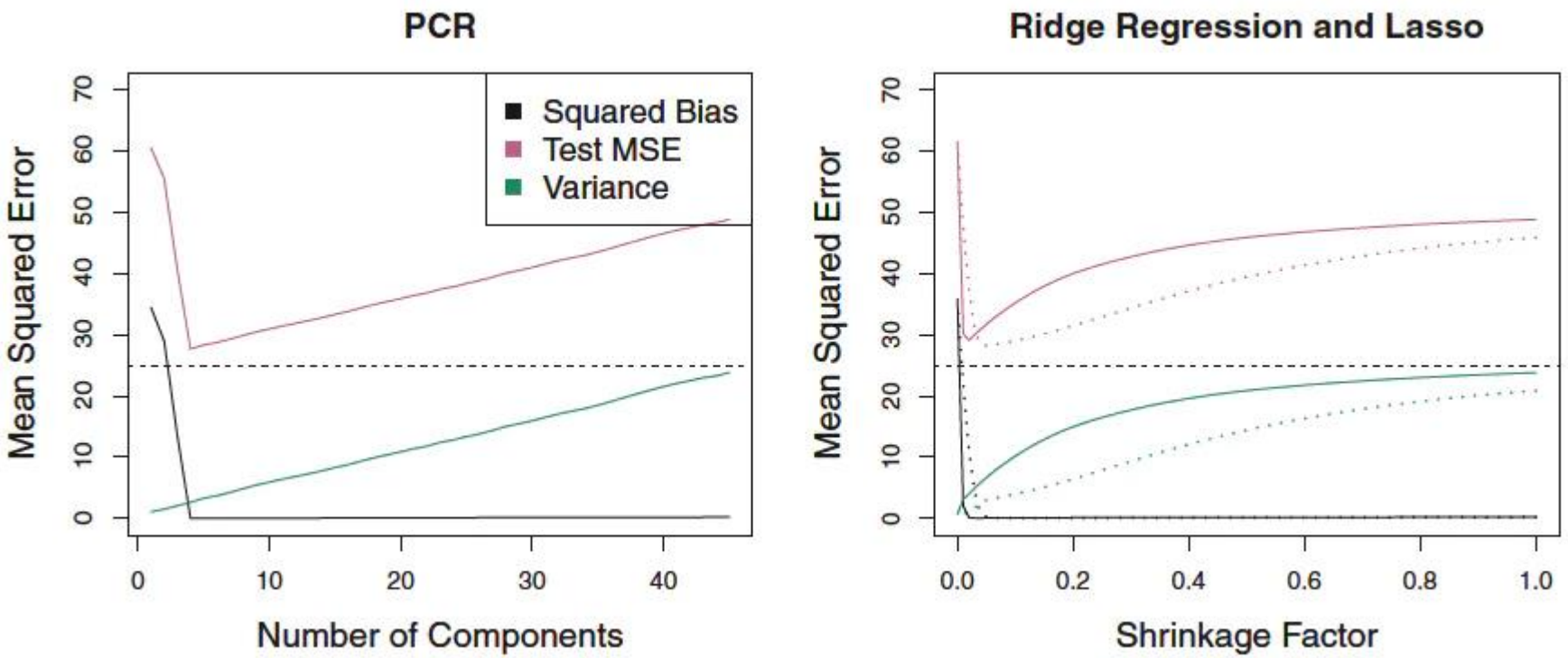


图3.6 主成分回归、岭回归和lasso应用于模拟数据集。

左：主成分回归结果；右lasso（实线）和岭回归（虚线）的结果。

主成分回归

主成分数量M一般通过交叉验证确定，下图中M=10时为最小交叉验证误差，几乎没有实现降维，因为M=11时与简单最小二乘等价。

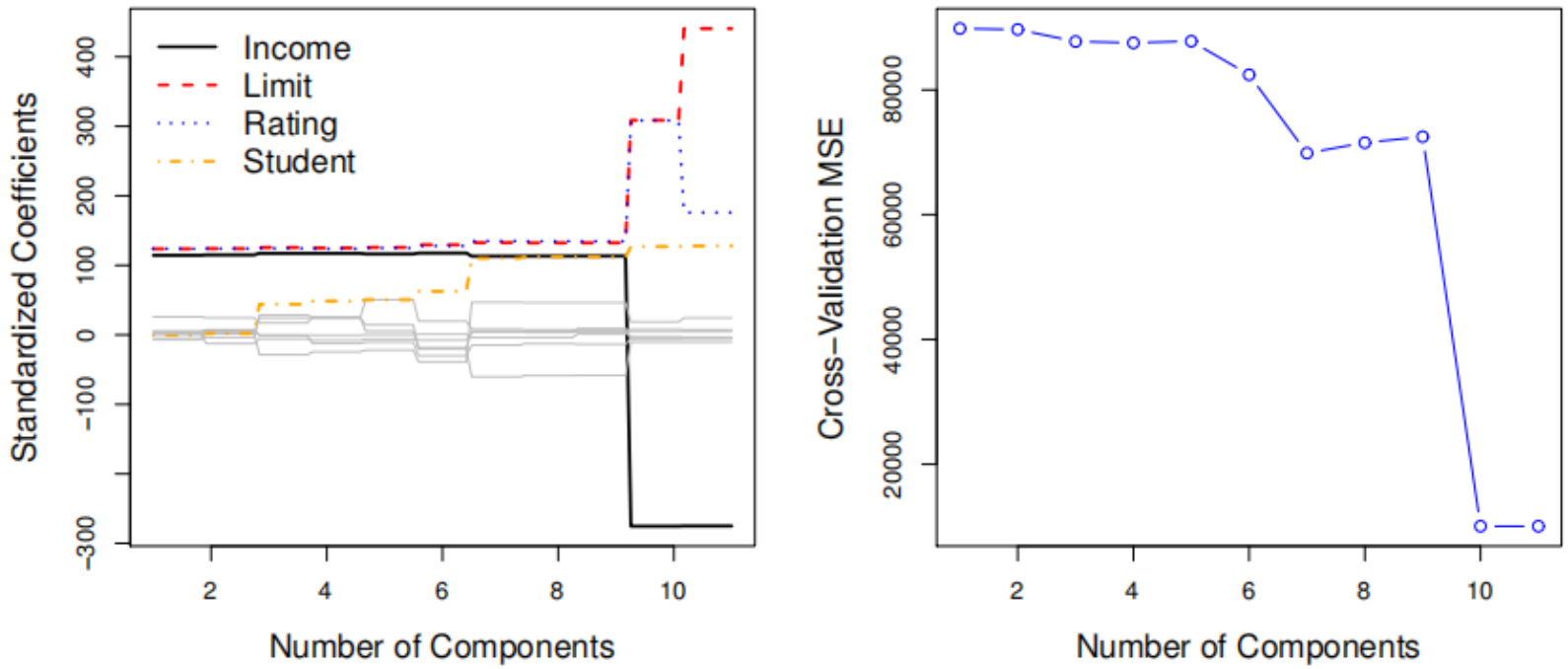


图3.7 左：**Credit数据集**上不同M值下主成分回归标准化系数估计；右：用主成分回归进行10折交叉验证的均方误差随M的变化。

主成分回归

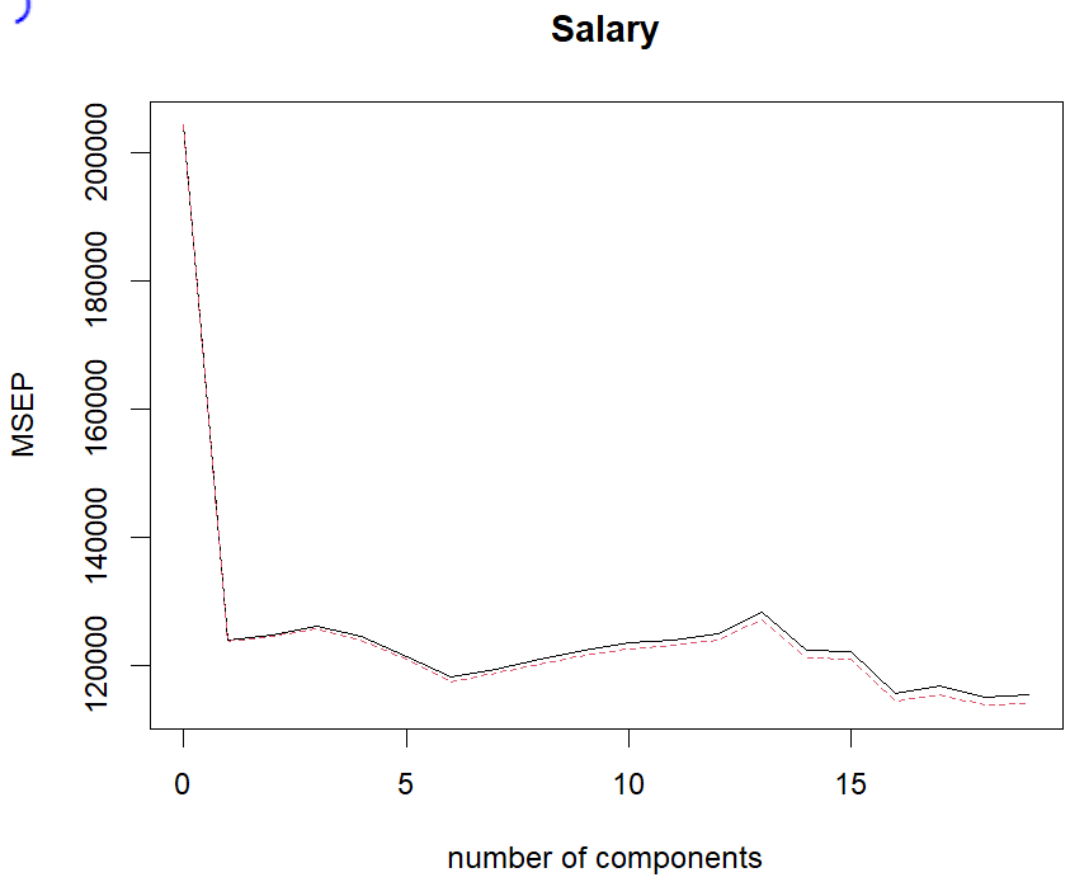
- 使用pls库中的pcr ( ) 函数实现主成分回归（PCR）。

```
> library(pls)
> set.seed(2)
> pcr.fit=pcr(Salary~., data=Hitters,scale=TRUE,validation="CV")
> summary(pcr.fit)
Data:   X dimension: 263 19
        Y dimension: 263 1
Fit method: svdpc
Number of components considered: 19

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
CV             452    351.9    353.2    355.0    352.8    348.4    343.6    345.5    347.7
adjCV          452    351.6    352.7    354.4    352.1    347.6    342.7    344.7    346.7
       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps
CV       349.6     351.4     352.1     353.5     358.2     349.7     349.4     339.9
adjCV    348.5     350.1     350.7     352.0     356.5     348.0     347.7     338.2
       17 comps  18 comps  19 comps
CV        341.6     339.2     339.6
adjCV     339.7     337.2     337.6

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps
X         38.31    60.16    70.84    79.03    84.29    88.63    92.26    94.96    96.28
Salary    40.63    41.58    42.17    43.22    44.90    46.48    46.69    46.75    46.86
        10 comps  11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps
X          97.26     97.98     98.65     99.15     99.47     99.75     99.89     99.97
Salary     47.76     47.82     47.85     48.10     50.40     50.55     53.01     53.85
        18 comps  19 comps
X          99.99    100.00
Salary     54.61     54.61
```

主成分回归

可使用validationplot( ) 函数做出交叉验证得分图像。

```
> validationplot(pcr.fit,val.type="MSEP")
```



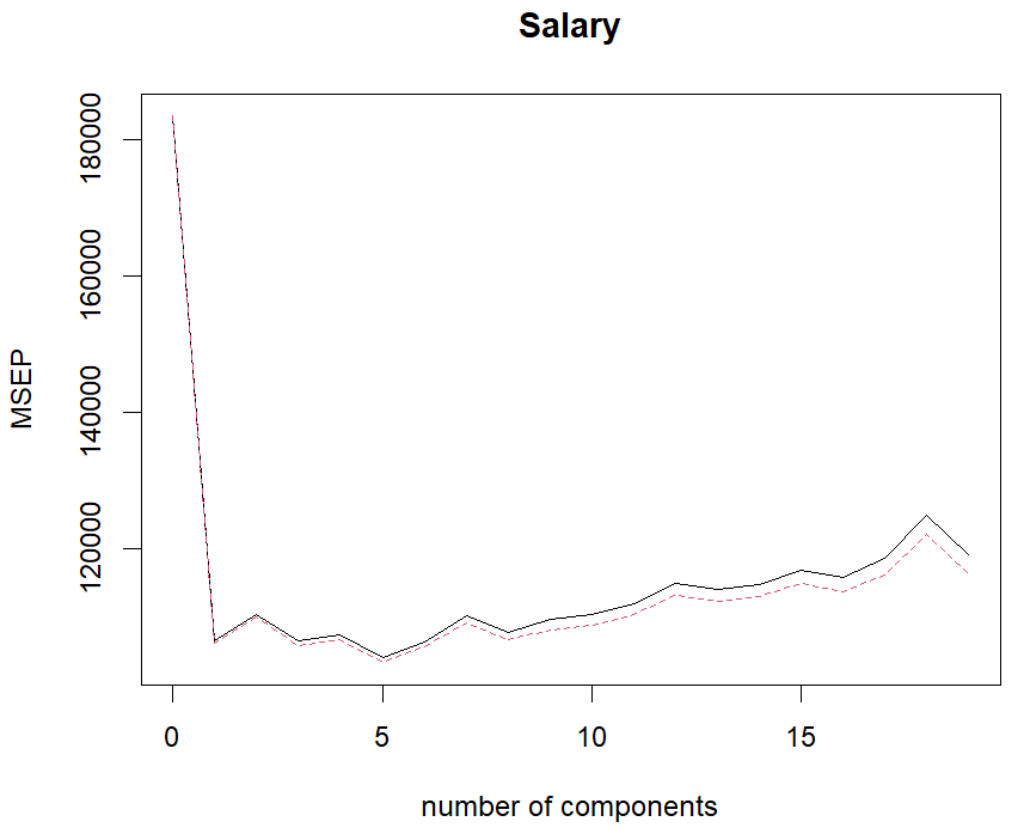
M=18时，交叉验证误差最小。

M=19时，数据维度未降低，PCR模型相当于最小二乘估计。

主成分回归

在训练集上使用PCR，评价该方法在测试集上的使用情况。

```
> set.seed(1)
> pcr.fit=pcr(Salary~., data=Hitters,subset=train,scale=TRUE, validation="CV")
> validationplot(pcr.fit,val.type="MSEP")
> pcr.pred=predict(pcr.fit,x[test,],ncomp=5)
> mean((pcr.pred-y.test)^2)
[1] 142811.8
> pcr.fit=pcr(y~x,scale=TRUE,ncomp=5)
> summary(pcr.fit)
Data:   X dimension: 263 19
        Y dimension: 263 1
Fit method: svdpc
Number of components considered: 5
TRAINING: % variance explained
   1 comps  2 comps  3 comps  4 comps  5 comps
X    38.31    60.16    70.84    79.03    84.29
y    40.63    41.58    42.17    43.22    44.90
```



M=5个成分时，交叉验证误差最小，最后在整个数据集上使用交叉验证选择出的成分个数M=5拟合PCR模型。

注：此处与书上结果不同，原因不明，可能是数据集有所不同。

偏最小二乘

偏最小二乘（partial least square, PLS）

一种有指导的主成分回归替代方法，同样是一种降维手段。它将原始变量的线性组合 $Z_1, Z_2, \ldots, Z_M$ 作为新的变量集，然后用这M个新变量拟合最小二乘模型。

- PCR不能保证最能解释预测因子的方向也是预测反应的最佳方向。而PLS以监督的方式进行新特征提取，试图寻找一个可以同时解释响应变量合预测变量的方向。
- 同PCR一样，个数M也是需要调节的参数，常用交验证选择。
- PLS回归前应对预测变量和响应变量标准化处理。
- 总的来说，与PCR的区别降维过程是有监督的，优化目标权衡了方差最大化和与Y的相关系数最大化。

偏最小二乘

前面提到 $Z_1, Z_2, \ldots, Z_M$表示M个原始预测变量的线性组合（M<p,共有p个原始变量）即：

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j$$

基本过程：

1、对p个预测变量进行标准化；

2、初始设置$Z_1 = \sum_{j=1}^{p} \beta_j \times X_j$，其中$\beta_j$为$Y$对$X_j$的简单回归系数，即$Y = \beta_0 + \beta_j \times X_j + \varepsilon_0$

3、做回归 Y=θ_0+$\theta_1 Z_1 + \varepsilon_1$，可得第一个偏最小二乘方向$\theta_1$，$\theta_0$以及$\varepsilon_1$

4、将残差送给Y，也就是 Y←Y−$\theta_1 Z_1$

5、正交化X变量，即$X_j \leftarrow X_j - \gamma_j * Z_1$，$(j = 1, 2, \ldots, p)$，其中$\gamma_j$系数$X_j$对$Z_1$的回归系数，也就是

$$X_j = \gamma j * Z_1 + \varepsilon_j$$

6、回到第2步，计算Z_2，按照以上步骤进行循环，可得$Z_1, Z_2 \ldots, Z_m$

在计算Z_1时，偏最小二乘将最大权重赋给与响应变量相关性最强的变量。

偏最小二乘

利用plsr( ) 函数可拟合偏最小二乘回归模型，函数在pls库中，句法与pcr( ) 函数句法相似。

```
> set.seed(1)
> pls.fit=plsr(Salary~., data=Hitters,subset=train,scale=TRUE, validation="CV")
> summary(pls.fit)
Data:   X dimension: 131 19
        Y dimension: 131 1
Fit method: kernelpls
Number of components considered: 19

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
CV           428.3    325.5    329.9    328.8    339.0    338.9    340.1
adjCV        428.3    325.0    328.2    327.2    336.6    336.1    336.6
       7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
CV       339.0    347.1    346.4     343.4     341.5     345.4     356.4
adjCV    336.2    343.4    342.8     340.2     338.3     341.8     351.1
       14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
CV        348.4     349.1     350.0     344.2     344.5     345.0
adjCV     344.2     345.0     345.9     340.4     340.6     341.1

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
X         39.13    48.80    60.09    75.07    78.58    81.12    88.21
Salary    46.36    50.72    52.23    53.03    54.07    54.77    55.05
        8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
X         90.71    93.17     96.05     97.08     97.61     97.97     98.70
Salary    55.66    55.95     56.12     56.47     56.68     57.37     57.76
        15 comps  16 comps  17 comps  18 comps  19 comps
X          99.12     99.61     99.70     99.95    100.00
Salary     58.08     58.17     58.49     58.56     58.62
```

偏最小二乘

测试交叉验证误差最小时的M的值，这里M=1，并计算与之相应的MSE。

```
> validationplot(pls.fit,val.type="MSEP")
> pls.pred=predict(pls.fit,x[test,],ncomp=1)
> mean((pls.pred-y.test)^2)
[1] 151995.3
```

使用交叉验证选取M=1个成分在整个数据集上建立的PLS模型

```
> pls.fit=plsr(Salary~., data=Hitters,scale=TRUE,ncomp=1
> summary(pls.fit)
Data:   X dimension: 263 19
        Y dimension: 263 1
Fit method: kernelpls
Number of components considered: 1
TRAINING: % variance explained
        1 comps
X         38.08
Salary    43.05
```

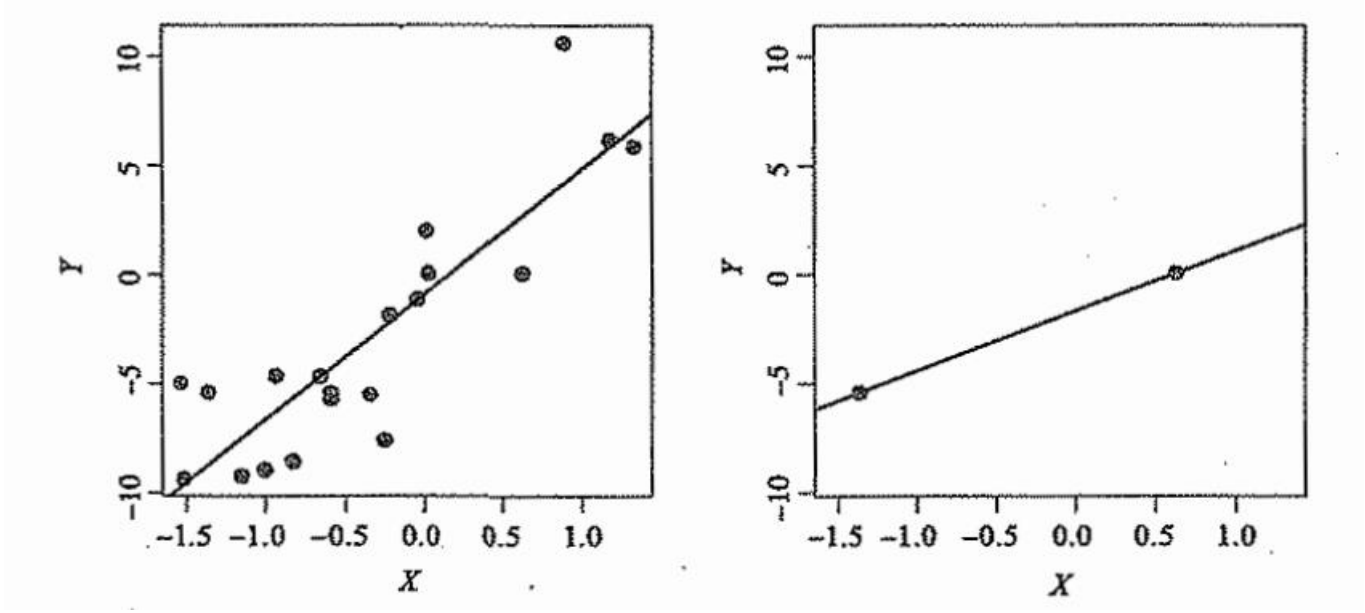
高维数据的问题

低维数据：观测数远大于特征数，$n \gg p$

高维数据：特征数比观测数大，$p > n$

图显示了p=1时，n=20以及n=2时，采用最小二乘拟合的情况。

无法很好拟合
但接近

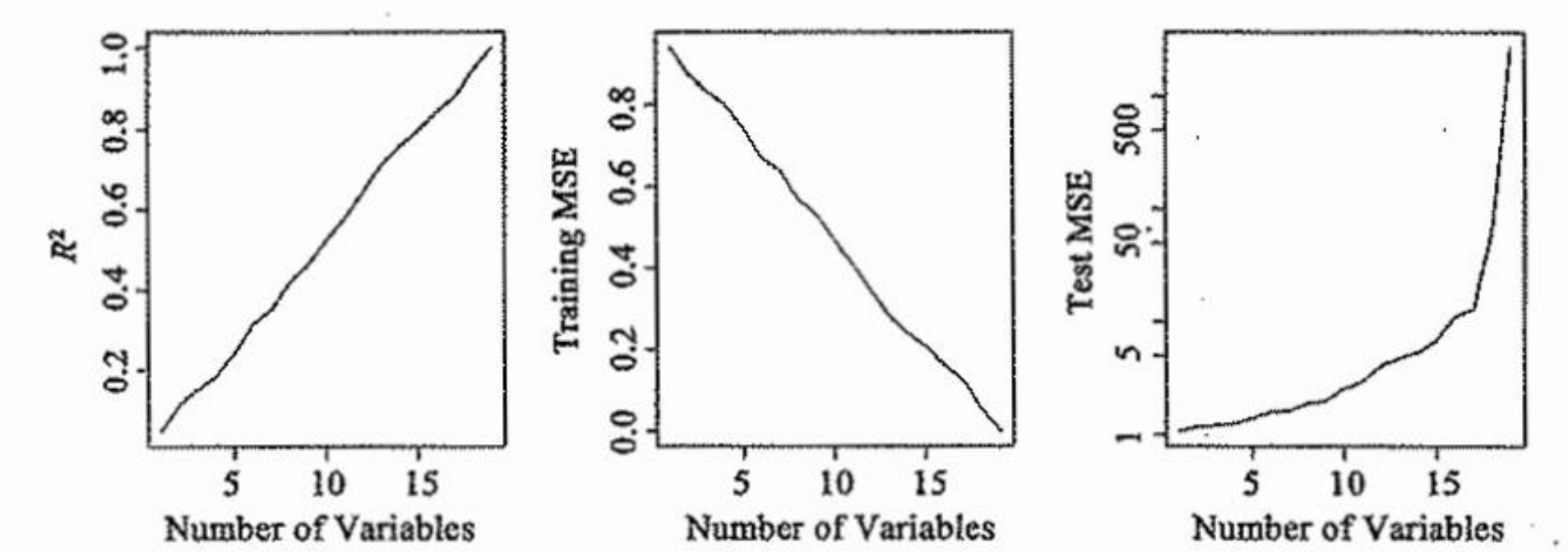


过拟合

左：低维最小二乘回归；右：n=2个观测和两个系数估计（1个截距和1个系数）的最小二乘回归。

高维数据的问题

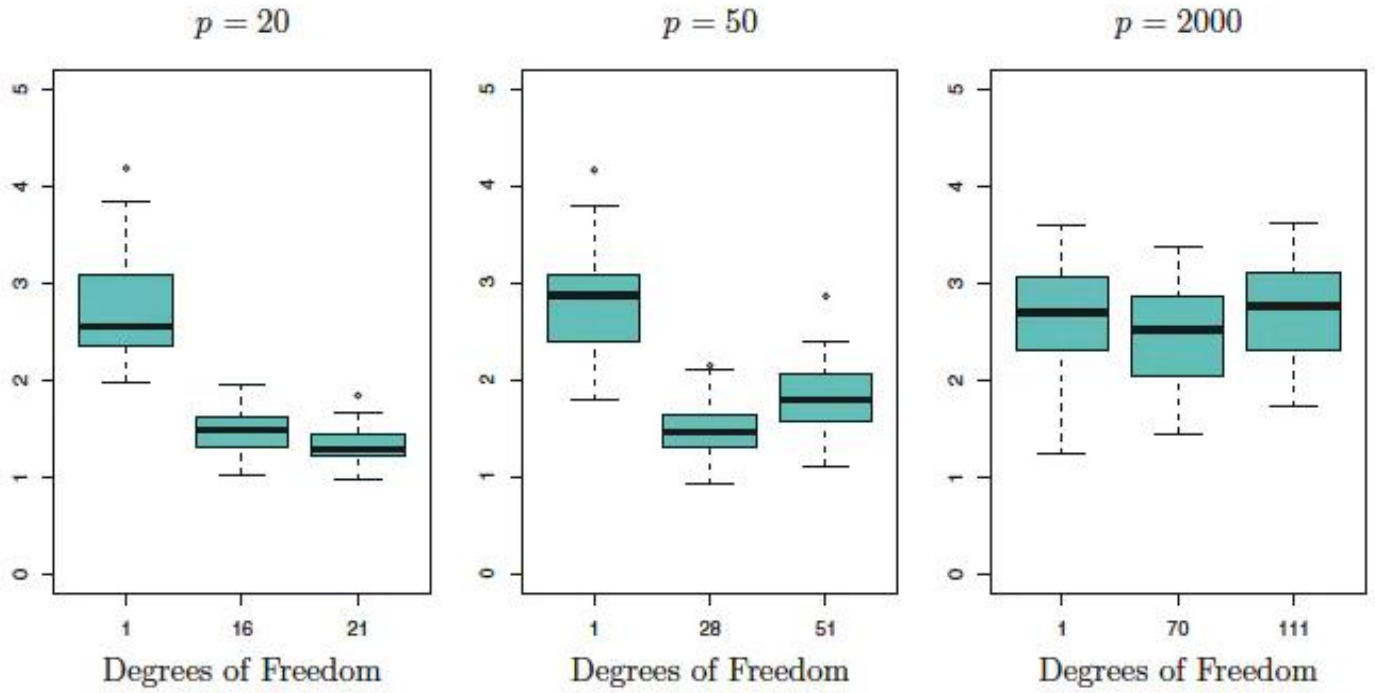
- 一个模拟的例子，n=20个训练观测，向模型中引入与输出变量完全不相关的特征。
- 只考虑R^2和训练数据集均方误差可能得到“包含最多变量的模型是好的”错误的结论。
- 此时应对独立测试集上评估性能给予更多的考虑。



左：随着特征增多，R^2增加到1。中：随着特征增多，训练均方误差下降为0。右：测试均方误差随着特征增多而逐渐增大。

高维数据的回归

高维数据：特征数比观测数大的数据被称为高维数据。传统方法部不在适合解决这种情况。会产生**方差-偏差平衡、过拟合**等问题，



本章所提的用于拟合并不光滑的最小二乘模型的方法，在高维回归中作用很大：

（1）正则或压缩在高维问题中至关重要。

（2）合适的调节参数对于得到好的预测非常关键。

（3）测试误差会随着数据维度的增加而增大，除非新增特征变量与响应变量确实相关。

图3.8 适用lasso对具有n=100个观测数据进行建模，p取三个不同的值。

分析结果的解释

使用lasso、岭回归或其他回归过程拟合高维数据时，必须非常谨慎地解释模型的结果。

共线性问题：模型中的任何一个变量都可以写成其他所有变量的线性组合。无法准确地知道哪个变量可以预测输出变量，也无法确定最优系数。

高维数据拟合模型的误差和拟合效果的解释，不能在训练数据集上用误差平方和、p值、R^2统计量或者其他传统的对模型拟合效果的度量方法来证明高维情况下模型拟合的效果。

本章作业

教材6.8习题1-6、8-10

其中1-2为上周作业，8-10为本章实验