岭回归和LASSO

学业辅导中心

1　岭回归

1.1　为什么研究岭回归?

从数学的角度考虑

由于在最小二乘估计中, 矩阵$X^\top X$求逆会变得不稳定,

- 当设计矩阵X的相关性非常高, 那么$X^\top X$几乎是奇异的.

$$X^\top X = P\begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{pmatrix}P^\top \quad \Rightarrow \quad (X^\top X)^{-1} = P\begin{pmatrix} \frac{1}{\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{\lambda_p} \end{pmatrix}P^\top$$

- 如果协变量的数目大于样本数量时, $X^\top X$是不可逆的.

Hoerl和Kennard(1970)提出来如下的岭估计作为OLS的改进:

$$\hat{\beta}^{\text{ridge}}(\lambda) = (X^\mathrm{T}X + \lambda I_p)^{-1}X^\mathrm{T}Y, \quad (\lambda > 0)$$

- λ是一个正的调节参数.
- 目标: 使$\min(\lambda_i)$远离0.

从统计的角度考虑

最小二乘估计量最小化的是残差平方和(residual sum of square):

$$\text{RSS}(b_0, b_1, \ldots, b_p) = \sum_{i=1}^{n}(y_i - b_0 - b_1 x_{i1} - \cdots - b_p x_{ip})^2.$$

最小二乘估计量的方差当在回归中加入额外的协变量时方差会增大, 这可能导致了一些很大的点估计的值.

为了避免过大的点估计的值, 我们可以乘法最小化残差平方和的准则, 化为

$$\hat{\beta}^{\text{ridge}}(\lambda) = \arg\min_{b_0, b_1, \ldots, b_p}\left\{\text{Rss}(b_0, b_1, \ldots, b_p) + \lambda\sum_{j=1}^{p} b_j^2\right\}.$$

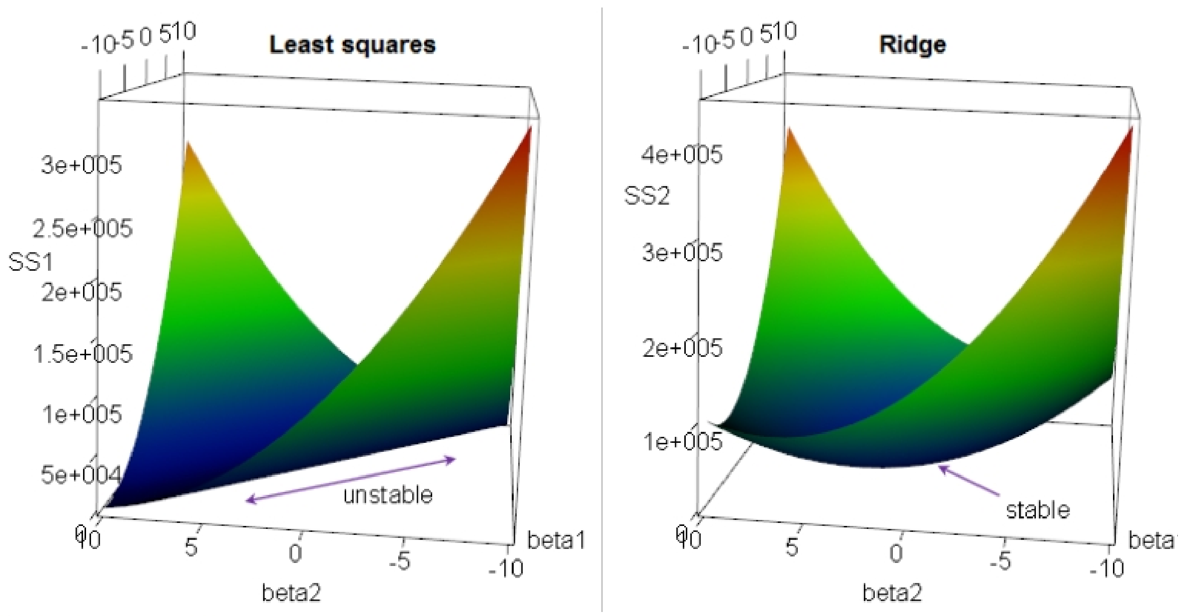
- 当$\lambda = 0$时, 上述估计就是最小二乘估计; 当$\lambda = \infty$时, 除了截距项, 其它系数的估计一定是0.
- 当$0 < \lambda < \infty$时, 岭估计一般比最小二乘估计要小, 因此当惩罚项收缩到0时, 岭估计变化为最小二乘估计量.

为什么叫岭回归?

I'll give an intuitive sense of why we're talking about ridges first (which also suggests why it's needed), then tackle a little history. The first is adapted from my answer here:

If there's multicollinearity, you get a "ridge" in the likelihood function (likelihood is a function of the β's). This in turn yields a long "valley" in the RSS (since RSS=$-2\log\mathcal{L}$).

Ridge regression "fixes" the ridge - it adds a penalty that turns the ridge into a nice peak in likelihood space, equivalently a nice depression in the criterion we're minimizing:



[Clearer image]

The actual story behind the name is a little more complicated. In 1959 A.E. Hoerl [1] introduced *ridge analysis* for response surface methodology, and it very soon [2] became adapted to dealing with multicollinearity in regression ('ridge regression'). See for example,

1.2 两种写法的等价性

$$
\begin{aligned}
f(b) &= \|y - Xb\|^2 + \lambda\|b\|^2 \\
&= b^\top(X^\top X + \lambda I_p)b - 2b^\top X^\top y + y^\top y.
\end{aligned}
$$

由于$X^\top X$是半正定的, $\lambda > 0$, 因此$X^\top X + \lambda I_n$是正定的. 根据数学分析的知识, 多元函数$f(b)$是严格凸的, 因此极小值点就是最小值点. 为求极小值点

$$\nabla f(b) = 2(X^\top X + \lambda I_p)b - 2X^\top y \overset{set}{=} 0,$$

于是$(X^\mathsf{T} X + \lambda I_p)^{-1} X^\mathsf{T} Y$是最小化问题的一个解.

反过来, 由于严格凸函数的极小值点是唯一的, 因此该解是唯一的, 因此两种写法是等价的.

1.3 对偶问题(dual problem)

给一个最小化的优化问题:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ \text{subject to } &h_i(x) \le 0, i = 1, \dots m \\ &\ell_j(x) = 0, j = 1, \dots r \end{aligned}$$

我们定义如下的拉格朗日型

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

以及拉格朗日对偶函数:

$$g(u, v) = \inf_{x\in\mathbb{R}^n} L(x, u, v) = \inf \left\{ f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x) \right\}.$$

此时, 上述极小化问题的对偶问题是:

$$\begin{gathered} \max_{u\in\mathbb{R}^m, v\in\mathbb{R}^r} g(u, v) \\ \text{subject to } u \ge 0 \end{gathered}$$

在岭回归中, 对偶问题是:

$$\begin{gathered} \hat{\beta}^{\text{ridge}}(t) = \arg \min_{b_0, b_1, \dots, b_p} \text{Rss}(b_0, b_1, \dots, b_p) \\ \text{s.t. } \sum_{j=1}^{p} b_j^2 \le t. \end{gathered}$$

岭估计没有线性不变性

- 在X的不同尺度下估计的量是不等价的.
- 依赖于X的单位(尺度)
- 惩罚项对于每一个系数的惩罚是相同的.

一个常见的约定是对X进行中心标准化. 岭估计中的投影矩阵是: $H(\lambda) = X(X^\top X + \lambda I_p)^{-1} X^\top$.

注记. 在R语言中, `lm.ridge()`函数先计算基于标准化的协变量和结果变量得到系数, 在将这些系数变换为原来的尺度. 也就是说先用标准化数据得到

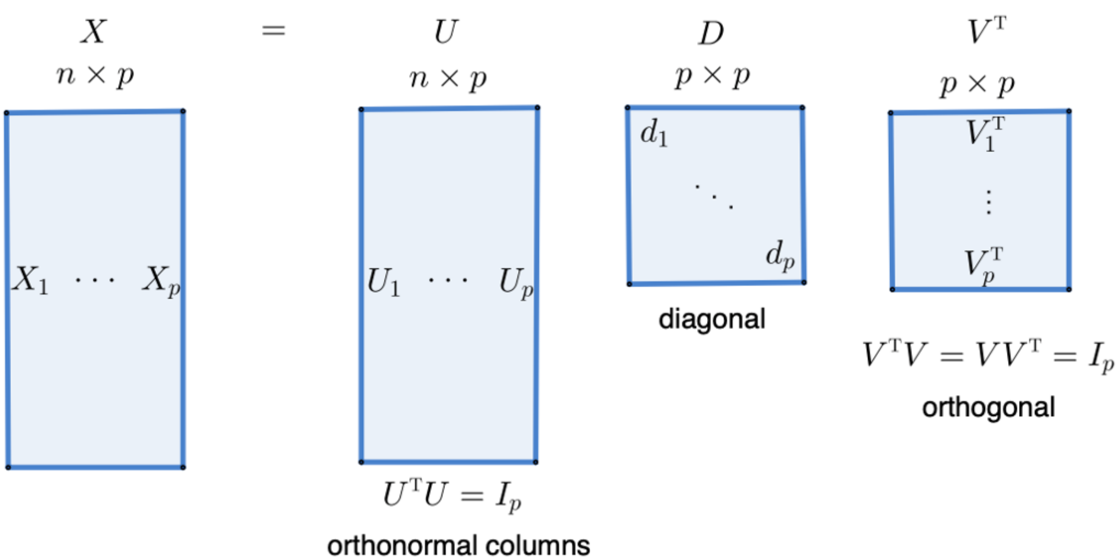
$$\{\hat{\beta}_1^{\text{ridge}}(\lambda), \ldots, \hat{\beta}_p^{\text{ridge}}(\lambda)\}.$$

再进行变换,

$$\hat{y}_i(\lambda) - \bar{y} = \hat{\beta}_1^{ridge}(\lambda)(x_{i1} - \bar{x}_1)/sd_1 + \cdots + \hat{\beta}_p^{ridge}(\lambda)(x_{ip} - \bar{x}_p)/sd_p$$
$$\hat{y}_i(\lambda) = \hat{\alpha}^{\text{ridge}}(\lambda) + \hat{\beta}_1^{\text{ridge}}(\lambda)/\text{sd}_1 \times x_{i1} + \cdots + \hat{\beta}_p^{\text{ridge}}(\lambda)/\text{sd}_p \times x_{ip}$$
$$\hat{\alpha}^{\text{ridge}}(\lambda) = \bar{y} - \hat{\beta}_1^{\text{ridge}}(\lambda)\bar{x}_1/\text{sd}_1 - \cdots - \hat{\beta}_p^{\text{ridge}}(\lambda)\bar{x}_p/\text{sd}_p$$

1.4 计算方法

一个矩阵的SVD分解(奇异值分解)可以写成$X = UDV^\top$.



我们可以得到以下的结果:

性质 (岭估计)**.**

$$\hat{\beta}^{\text{ridge}}(\lambda) = V\text{diag}\left(\frac{d_j}{d_j^2 + \lambda}\right)U^{\text{T}}Y$$

证明.

$$\begin{aligned}
\hat{\beta}^{\text{ridge}}(\lambda) &= (X^{\text{T}}X + \lambda I_p)^{-1}X^{\text{T}}Y \\
&= (VDU^{\text{T}}UDV^{\text{T}} + \lambda I_p)^{-1}VDU^{\text{T}}Y \\
&= V(D^2 + \lambda I_p)^{-1}V^{\text{T}}VDU^{\text{T}}Y \\
&= V(D^2 + \lambda I_p)^{-1}DU^{\text{T}}Y \\
&= V\text{diag}\left(\frac{d_j}{d_j^2 + \lambda}\right)U^{\text{T}}Y.
\end{aligned}$$

□

岭估计和最小二乘相比怎样? 我们看一下岭估计的统计性质:

- 点估计:

$$\begin{aligned} E\{\hat\beta^{\text{ridge}}(\lambda)\} &= V\text{diag}\left(\frac{d_j}{d_j^2+\lambda}\right)U^{\mathrm T}X\beta \quad \text{根据}\mathbb{E}(Y)=X\beta \\ &= V\text{diag}\left(\frac{d_j}{d_j^2+\lambda}\right)U^{\mathrm T}UDV^{\mathrm T}\beta \\ &= V\text{diag}\left(\frac{d_j^2}{d_j^2+\lambda}\right)V^{\mathrm T}\beta \end{aligned}$$

- 估计的方差:

$$\begin{aligned} \text{cov}\{\hat\beta^{\text{ridge}}(\lambda)\} &= \sigma^2 V\text{diag}\left(\frac{d_j}{d_j^2+\lambda}\right)U^{\mathrm T}U\text{diag}\left(\frac{d_j}{d_j^2+\lambda}\right)V^{\mathrm T} \\ &= \sigma^2 V\text{diag}\left(\frac{d_j^2}{(d_j^2+\lambda)^2}\right)V^{\mathrm T}. \end{aligned}$$

回顾偏倚-方差分解

$$\begin{aligned} MSE\left\{\hat f(x)\right\} &= E_f\left[\left(\hat f(x)-f(x)\right)^2\right] \\ &= E\left[\left(\hat f(x)-E\hat f(x)+E\hat f(x)-f(x)\right)^2\right] \\ &= E\left[\left(\hat f(x)-E\hat f(x)\right)^2\right]+E\left[\left(E\hat f(x)-f(x)\right)^2\right]+2E\left[\underbrace{\left(\hat f(x)-E\hat f(x)\right)}_{\text{外面期望求进来为零}}\underbrace{\left(E\hat f(x)-f(x)\right)}_{\text{与}\hat f\text{无关, 相当于常数}}\right] \\ &= E\left[\left(\hat f(x)-E\hat f(x)\right)^2\right]+\left(E\hat f(x)-f(x)\right)^2 \\ &= Var(\hat f(x))+Bias(\hat f(x))^2 \end{aligned}$$

于是对于岭估计的MSE,

$$\begin{aligned} \text{MSE}(\lambda) &\triangleq E\left[\left\{\hat\beta^{\text{ridge}}(\lambda)-\beta\right\}^{\mathrm T}\left\{\hat\beta^{\text{ridge}}(\lambda)-\beta\right\}\right] \\ &= \underbrace{[E\{\hat\beta^{\text{ridge}}(\lambda)\}-\beta]^{\mathrm T}[E\{\hat\beta^{\text{ridge}}(\lambda)\}-\beta]}_{C_1\text{:偏差的平方}}+\underbrace{\text{trace}[\text{cov}\{\hat\beta^{\text{ridge}}(\lambda)\}]}_{C_2\text{:方差}}. \end{aligned}$$

由于$C_1=\beta^{\top}V\text{diag}\left(\frac{\lambda}{d_j^2+\lambda}\right)^2V^{\top}\beta$, 令$\gamma=V^{\top}\beta$就有

$$C_1=\gamma^{\top}\text{diag}\left(\frac{\lambda^2}{(d_j^2+\lambda)^2}\right)\gamma=\sum_j\frac{\lambda^2\gamma_j^2}{(d_j^2+\lambda)^2}=\lambda^2\sum_{j=1}^p\frac{\gamma_j^2}{(d_j^2+\lambda)^2},$$

$$C_2=\text{trace}[\text{cov}\{\hat\beta^{\text{ridge}}(\lambda)\}]=\text{trace}\left(\sigma^2V\text{diag}\left(\frac{d_j^2}{(d_j^2+\lambda)^2}\right)V^{\mathrm T}\right)=\sigma^2\sum_{j=1}^p\frac{d_j^2}{(d_j^2+\lambda)^2}$$

于是

$$\mathrm{MSE}(\lambda)=C_1+C_2=\lambda^2\sum_{j=1}^{p}\frac{\gamma_j^2}{(d_j^2+\lambda)^2}+\sigma^2\sum_{j=1}^{p}\frac{d_j^2}{(d_j^2+\lambda)^2}.$$

因此岭估计中的偏倚-方差权衡是: 当λ增大时, 偏差增大, 方差减小; 当λ减小时, 偏差减小, 方差增大.

注记. *岭回归不能用作统计推断, 因为虽然方差可以计算, 但是偏差是未知的, 因此仅仅知道一个有偏估计量的方差对推断几乎没有帮助.*

1.5 调节参数λ的选择

求导

我们可以从参数估计都角度考虑, 我们可以选取λ使得MSE最小, 于是对参数求导, 我们有

$$\begin{aligned}\frac{\partial \mathrm{MsE}(\lambda)}{\partial\lambda}&=2\sum_{j=1}^{p}\gamma_j^2\frac{\lambda}{d_j^2+\lambda}\frac{d_j^2+\lambda-\lambda}{(d_j^2+\lambda)^2}-2\sigma^2\sum_{j=1}^{p}\frac{d_j^2}{(d_j^2+\lambda)^3}=0\\ &\Longleftrightarrow \lambda\sum_{j=1}^{p}\frac{\gamma_j^2d_j^2}{(d_j^2+\lambda)^3}=\sigma^2\sum_{j=1}^{p}\frac{d_j^2}{(d_j^2+\lambda)^3}.\end{aligned}$$

但是这里我们不知道$\gamma=V^\top\beta$和σ^2的值. 一个直观的想法是先估计再带入,

- Dempster et al.(1977)用最小二乘估计$\widehat{\sigma}^2,\widehat{\gamma}=V^\top\widehat{\beta}$, 求解$\lambda\sum_{j=1}^{p}\frac{\hat{\gamma}_j^2d_j^2}{(d_j^2+\lambda)^3}=\hat{\sigma}^2\sum_{j=1}^{p}\frac{d_j^2}{(d_j^2+\lambda)^3}$,
- Heorl et al.(1975)假设$X^\top X=I_p$, 解出$\lambda\sum_{j=1}^{p}\frac{\hat{\beta}_j^2}{(1+\lambda)^3}=\hat{\sigma}^2\sum_{j=1}^{p}\frac{1}{(1+\lambda)^3}, \lambda_{\mathrm{HKB}}=p\hat{\sigma}^2/\|\hat{\beta}\|^2$.

PRESS

与之前的PRESS类似, 这里我们有

- 岭估计: $\hat{\beta}(\lambda)=(X^\mathrm{T}X+\lambda I_p)^{-1}X^\mathrm{T}Y$
- 残差向量: $\hat{\varepsilon}(\lambda)=Y-X\hat{\beta}(\lambda)$
- 杠杆值:$h_{ii}(\lambda)=x_i^\mathrm{T}(X^\mathrm{T}X+\lambda I_p)^{-1}x_i$
- 预测残差: $\hat{\varepsilon}_{[-i]}(\lambda)=y_i-x_i^\mathrm{T}\hat{\beta}_{[-i]}(\lambda)$

可以证明: $\widehat{\boldsymbol{\beta}}-\widehat{\boldsymbol{\beta}}^{(-i)}=\frac{(\mathbf{X}^\top\mathbf{X})^{-1}\boldsymbol{x}_i\widehat{e}_i}{1-h_{ii}}$, 于是:

$$\begin{aligned}&\hat{\beta}_{[-i]}(\lambda)=\hat{\beta}(\lambda)-\{1-h_{ii}(\lambda)\}^{-1}(X^\mathrm{T}X+\lambda I_p)^{-1}x_i\hat{\varepsilon}_i(\lambda)\\ &\hat{\varepsilon}_{[-i]}(\lambda)=\hat{\varepsilon}_i(\lambda)/\{1-h_{ii}(\lambda)\}\end{aligned}$$

因此, PRESS统计量就是:

$$\mathrm{PRESS}(\lambda)=\sum_{i=1}^{n}\left\{\hat{\varepsilon}_{[-i]}(\lambda)\right\}^2=\sum_{i=1}^{n}\frac{\{\hat{\varepsilon}_i(\lambda)\}^2}{\{1-h_{ii}(\lambda)\}^2}.$$

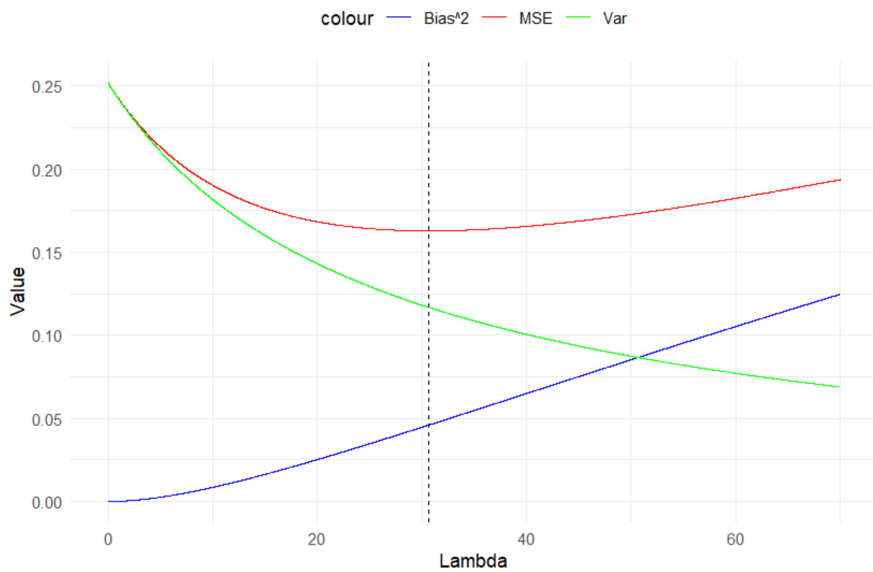
选择λ使得PRESS较小.

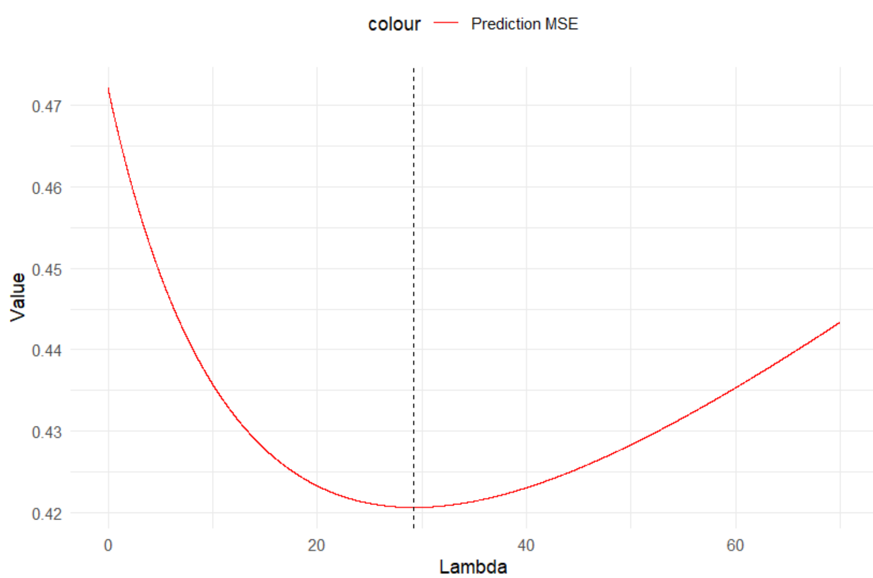
1.6 模拟试验

- $n = 200, p = 100$
- λ从0到70, 步长为0.01.
- 我们将计算:

  1. 理论的偏差, 方差, MSE
  2. 生成两组数据, 一组是测试集, 一组是验证集, 观察MSE.

```
rm(list = ls())
library(MASS)
n <- 200
p <- 100
beta <- rep(1/sqrt(p),p)  #rep 表示复制次p1/ () sqrtp
sig <- 1/2
### uncorrelated covariates
X <- matrix(rnorm(n*p),n,p)
X <- scale(X)          #标准化X


### scale use n-1 but ridge uses n
X <- X*sqrt(n/(n-1))                         #标准化时除的是n 但是中默认是除rn-1
Y <- as.vector(X%*%beta+rnorm(n,0,sig))      #将矩阵转换为行向量 () 按列进行
#MSE
eigenxx <- eigen(t(X)%*%X)          #求‘的特征值和特征向量XX 为了获得中的奇异值SVD
xis <- eigenxx$values
gammas <- t(eigenxx$vectors)%*%beta

lambda.seq <- seq(0,70,0.01)                 #回归中的参数ridgelambda
bias2.seq <- lambda.seq
var.seq <- lambda.seq
mse.seq <- lambda.seq
for (i in 1:length(lambda.seq)){
  ll <- lambda.seq[i]
  bias2.seq[i] <- ll^2*sum(gammas^2/(xis+ll)^2)
  var.seq[i] <- sig^2*sum(xis/(xis+ll)^2)
  mse.seq[i] <- bias2.seq[i]+var.seq[i]
}

data <- data.frame(lambda = lambda.seq, bias2 = bias2.seq, mse = mse.seq, var = var.seq)

ggplot(data, aes(x = lambda)) +
  geom_line(aes(y = bias2, color = "Bias^2")) +
  geom_line(aes(y = mse, color = "MSE")) +
  geom_line(aes(y = var.seq, color = "Var")) +
  geom_vline(xintercept = data$lambda[which.min(mse.seq)], linetype = "dashed", color = "black") +
  labs(x = "Lambda", y = "Value") +
  scale_color_manual(values = c("Bias^2" = "blue", "MSE" = "red", "Var" = "green")) +
  theme_minimal() +
  theme(legend.position = "top")
```


colour
Bias^2
MSE
Var
Value
Lambda
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60


colour
Prediction MSE
Value
Lambda
0.47
0.46
0.45
0.44
0.43
0.42
0
20
40
60

2 LASSO

2.1 为什么研究LASSO?

岭回归在模型预测上表现的很好: 见上面的试验. 但是岭回归难以解释那些很小又非0的系数.

Tibshirani(1996)提出来LASSO, 是least absolute shrinkage and selection operator(最小绝对值收敛和选择算子)的缩写. 它的优点在于:

- 同时估计参数, 并且将不必要的变量系数置为0
- 仅仅通过改变岭回归中的惩罚项, 就能自动把一些系数的估计变为0, 因此实现了变量选择.

Tibshirani(1996)提出的lasso的形式是:

$$\hat{\beta}^{\text{lasso}}(t) = \arg\min_{b_0,b_1,\ldots,b_p} \text{Rss}(b_0,b_1,\ldots,b_p)$$
$$\text{s.t.} \sum_{j=1}^{p}|b_j| \le t.$$

Osborne et al.(2000)研究了它的对偶形式:

$$\hat{\beta}^{\text{lasso}}(\lambda) = \arg\min_{b_0,b_1,\ldots,b_p}\left\{\text{Rss}(b_0,b_1,\ldots,b_p) + \lambda\sum_{j=1}^{p}|b_j|\right\}.$$

- 类似岭回归, 要对X标准化来去掉常数项.
- 在: 给定λ存在t两个为题有相同的解的意义下, 两个问题是等价的.
- 当$p > n$时, 解可能不唯一, 但预测的值一定是唯一的.

2.2 几何解释

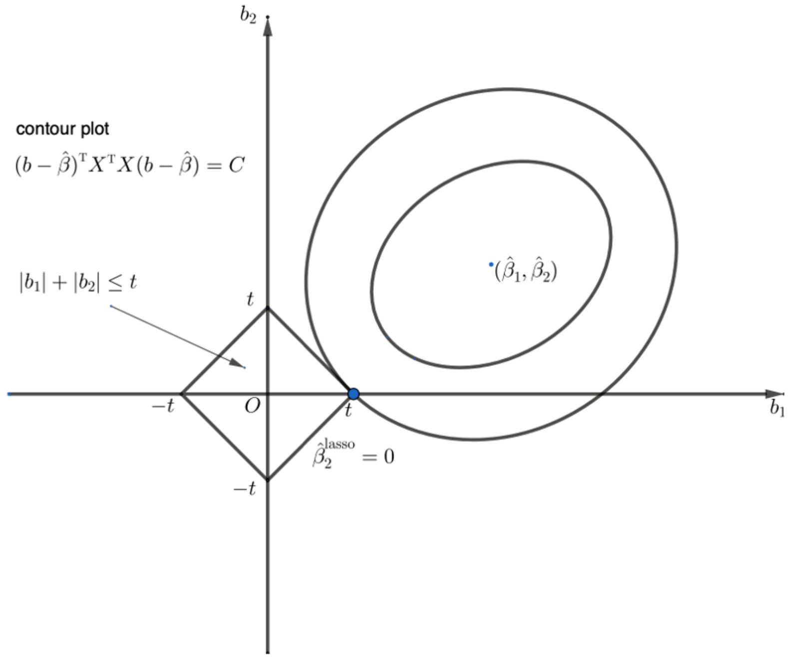
同样的,我们考察b是自变量, 因此$RSS(b) = (Y - Xb)^\top(Y - Xb)$, 可见$b = \hat{\beta}$时, RSS最小.

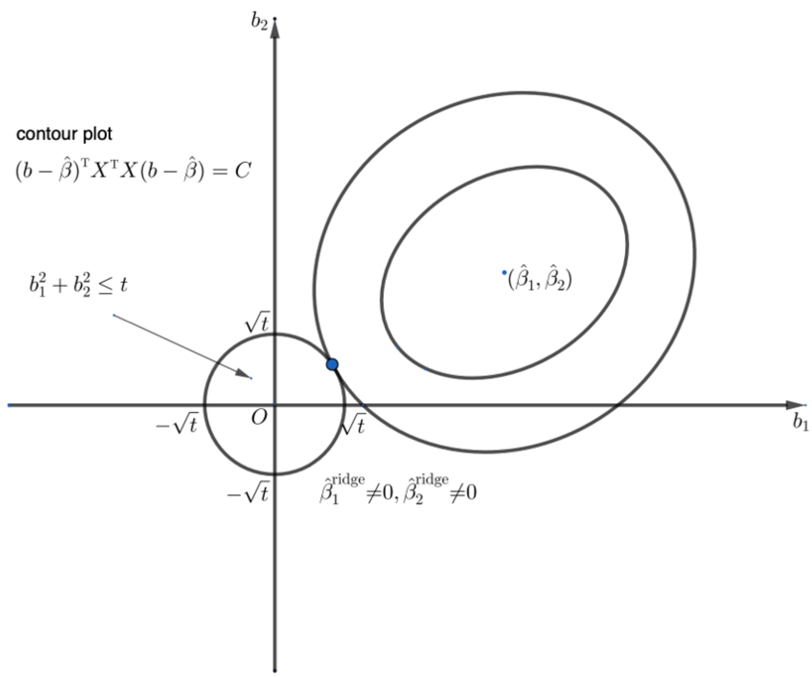
$$\begin{aligned}(Y - Xb)^{\mathrm{T}}(Y - Xb) &= (Y - X\hat{\beta} + X\hat{\beta} - Xb)^{\mathrm{T}}(Y - X\hat{\beta} + X\hat{\beta} - Xb)\\ &= (Y - X\hat{\beta})^{\mathrm{T}}(Y - X\hat{\beta}) + (b - \hat{\beta})^{\mathrm{T}}X^{\mathrm{T}}X(b - \hat{\beta}) - (b - \hat{\beta})^\top\underbrace{X^\top(Y - X\hat{\beta})}_{=0} - \underbrace{(Y - X\hat{\beta})X}_{=0}(b - \hat{\beta})\\ &= (Y - X\hat{\beta})^{\mathrm{T}}(Y - X\hat{\beta}) + (b - \hat{\beta})^{\mathrm{T}}X^{\mathrm{T}}X(b - \hat{\beta})\end{aligned}$$

它的等值线是椭圆, 再考虑惩罚项, 由于满足RSS最小的b未必在限制区域内部. 由于惩罚项的不光滑性, 给出来稀疏的解.

值得注意的是: 当$C = 0$时, 中心点就是最小二乘估计量. 取到坐标轴上的情况就是系数被压缩的情况.

注记. *若改为$\sum|x|^{1/p} = t, p > 1$, 则不是凸优化问题, 求解比较困难.*


b_2
contour plot
$(b-\hat{\beta})^{\mathrm{T}}X^{\mathrm{T}}X(b-\hat{\beta})=C$
$(\hat{\beta}_1,\hat{\beta}_2)$
$|b_1|+|b_2|\le t$
t
$-t$
O
t
b_1
$\hat{\beta}_2^{\text{lasso}}=0$
$-t$


b_2
contour plot
$(b-\hat{\beta})^{\mathrm{T}}X^{\mathrm{T}}X(b-\hat{\beta})=C$
$(\hat{\beta}_1,\hat{\beta}_2)$
$b_1^2+b_2^2\le t$
$\sqrt{t}$
$-\sqrt{t}$
O
$\sqrt{t}$
b_1
$-\sqrt{t}$
$\hat{\beta}_1^{\text{ridge}}\neq 0, \hat{\beta}_2^{\text{ridge}}\neq 0$

2.3 用坐标下降法计算LASSO

我们先考虑一个简单的情况: 给定b_0和$\lambda \geq 0$, 我们有

$$\begin{aligned}\arg\min_{b\in\mathbb{R}} \frac{1}{2}(b-b_0)^2 + \lambda|b| &= \mathrm{sign}(b_0)\,(|b_0|-\lambda)_+ \\ &= \begin{cases} b_0-\lambda, & if\ \ b_0 \geq \lambda, \\ 0 & if\ \ -\lambda \leq b_0 \leq \lambda, \\ b_0+\lambda & if\ \ b_0 \leq -\lambda. \end{cases}\end{aligned}$$

证明.

$$\begin{aligned}\frac{1}{2}(b-b_0)^2 + \lambda|b| &= \frac{1}{2}b^2 + (\lambda|b| - b_0 b) + \frac{1}{2}b_0^2 \\ &= \frac{1}{2}b^2 + (\lambda\mathrm{sign}(b) - b_0)b + \frac{1}{2}b_0^2\end{aligned}$$

在$b > 0$部分, 上式$= \frac{1}{2}b^2 + (\lambda - b_0)b + \frac{1}{2}b_0^2$,

1. $b_0 \geq \lambda \Rightarrow \arg\min = b_0 - \lambda$;

2. $b_0 \leq \lambda \Rightarrow \arg\min = 0$

在$b < 0$部分, 上式$= \frac{1}{2}b^2 + (-\lambda - b_0)b + \frac{1}{2}b_0^2$,

1. $b_0 \geq -\lambda \Rightarrow \arg\min = 0$;

2. $b_0 \leq -\lambda \Rightarrow \arg\min = b_0 + \lambda$

□

我们记$S(b_0, \lambda) = \mathrm{sign}(b_0)\,(|b_0| - \lambda)_+$.

- 首先我们标准化我们的数据, 因此我们不需要解截距项.

- 改变前面的系数不会改变优化问题, 因此我们可以把问题写成:

$$\min_{b_1,\ldots,b_p} \frac{1}{2n}\sum_{i=1}^{n}(y_i - b_1 x_{i1} - \cdots - b_p x_{ip})^2 + \lambda\sum_{j=1}^{p}|b_j|.$$

- 给定一个迭代的初始值$\hat{\beta}$.

$$\begin{aligned}
x_1^{(k)} &\in \underset{x_1}{\mathrm{argmin}}\ f\left(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)}\right) \\
x_2^{(k)} &\in \underset{x_2}{\mathrm{argmin}}\ f\left(x_1^{(k)}, x_2, x_3^{(k-1)}, \ldots x_n^{(k-1)}\right) \\
x_3^{(k)} &\in \underset{x_3}{\mathrm{argmin}} f\left(x_1^{(k)}, x_2^{(k)}, x_3, \ldots x_n^{(k-1)}\right) \\
&\ldots \\
x_n^{(k)} &\in \underset{x_n}{\mathrm{argmin}}\ f\left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n\right)
\end{aligned}$$

我们采用如下的方法进行迭代(有点类似于Gibbs sampling): 我们记部分残差为

$$r_{ij} = y_i - \sum_{k\neq j} \hat{\beta}_k x_{ik}.$$

由于$\hat{\beta}$是固定的, 于是最小化原来的(P)问题的等价写法是最小化

$$\frac{1}{2n}\sum_{i=1}^{n}(r_{ij} - b_j x_{ij})^2 + \lambda|b_j|.$$

定义OLS的回归结果是:

$$\hat{\beta}_{j,0} = \frac{\sum_{i=1}^{n} x_{ij} r_{ij}}{\sum_{i=1}^{n} x_{ij}^2 (= n)} = n^{-1}\sum_{i=1}^{n} x_{ij} r_{ij}$$

用加一项减一项,

$$\begin{aligned}\frac{1}{2n}\sum_{i=1}^{n}(r_{ij} - b_j x_{ij})^2 &= \frac{1}{2n}\sum_{i=1}^{n}(r_{ij} - \hat{\beta}_{j,0} x_{ij})^2 + \frac{1}{2n}\sum_{i=1}^{n} x_{ij}^2 (b_j - \hat{\beta}_{j,0})^2 \\ &= \text{constant} + \frac{1}{2}(b_j - \hat{\beta}_{j,0})^2.\end{aligned}$$

根据前面的引理, 迭代式就是

$$\hat{\beta}_j = S(\hat{\beta}_{j,0}, \lambda).$$

- 它的收敛性可以保证.(Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization)
- 初始值可以设置$\beta = 0$, λ比较大.
- λ从大到小, 依次使用CD(coordinate descent)法.
- 关于λ的选择可以用K-折交叉验证.

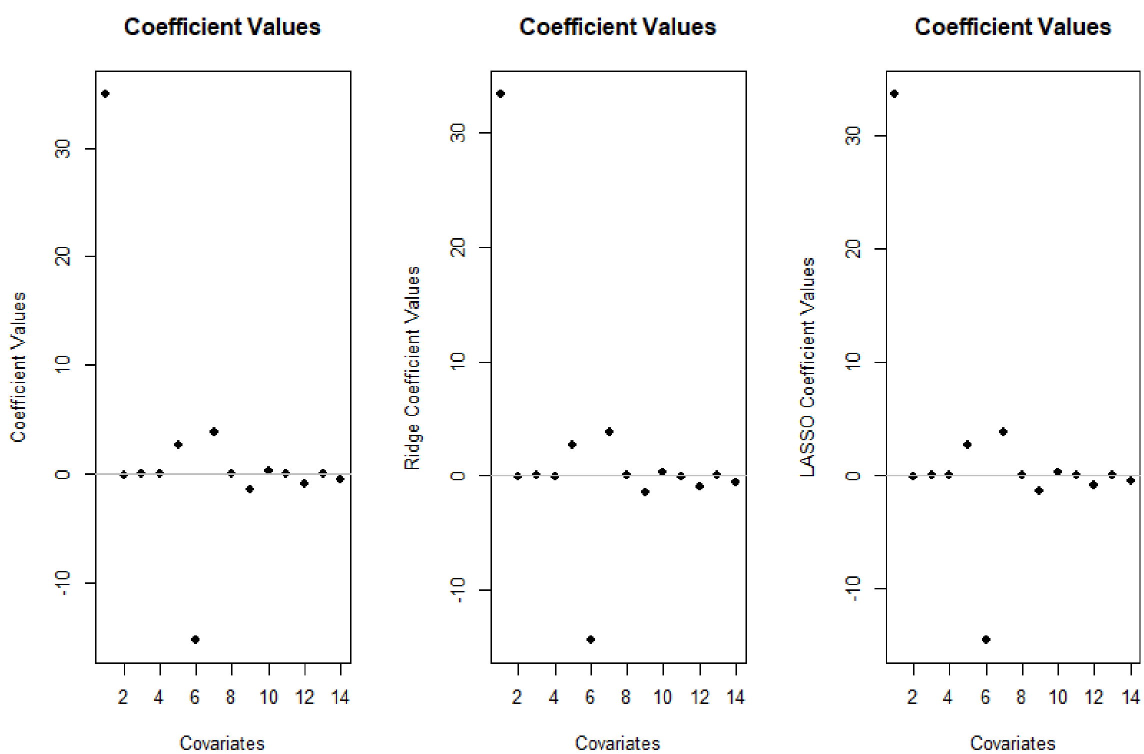
注记. *LASSO的统计推断是困难的(有偏), 而且考虑到λ的随机性是比较困难的.*

2.4 例子: Boston房价

我们用R语言自带的数据, 研究Boston房价和其它协变量之间的关系.

```
### training and testing data
set.seed(1)
#set.seed 表示设置一个种子 后面程序生成的随机数是依赖这个种子生成的
#这样就能保证后面每次（）生成的随机数都是随机但是相同的norm
nsample <- dim(BostonHousing)[1]
trainindex <- sample(1:nsample, floor(nsample*0.9))     #向上取整函数floor
#sample 表示随机抽样函数  这里指的是从1～中随机抽n 0.9个指标n  训练集
#: eg n=10时   从1～10里随机抽9个  8, 5, 4, 3, 2, 7, 6, 1, 10
#目的：将数据分为两组 一组作为测试集，一组作为训练集
xmatrix <- model.matrix(medv~.,data=BostonHousing)[,-1]     #创立设计矩阵
yvector <- BostonHousing$medv
dat <- data.frame(yvector,xmatrix)

## linear regression
```

```r
bostonlm <- lm(yvector~.,data=dat[trainindex,])
predicterror <- dat$yvector[-trainindex]-predict(bostonlm,dat[-trainindex,])#测试集检验
mse.ols <- sum(predicterror^2)/length(predicterror)

## ridge regression
lambdas <- seq(0,5,0.01)
lm0 <- lm.ridge(yvector~.,data=dat[trainindex,],lambda=lambdas)
coefridge <- coef(lm0)[which.min(lm0$GCV),]  #根据选择GCVlambdas
predicterrorridge <- dat$yvector[-trainindex]-cbind(1,xmatrix[-trainindex,])%*%coefridge
mse.ridge <- sum(predicterrorridge^2)/length(predicterrorridge)

## lasso
cvboston <- cv.glmnet(x=xmatrix[trainindex,],y=yvector[trainindex])
#回归lasso  实际上函数默认cv.glmnet(x=xmatrix[trainindex,],y=yvector[trainindex, ]alpha=1)
# 指弹性网络中的alphaalpha 回归时ridge alpha=0
#进行一交叉检验kfold 默认k=10
coeflasso <- coef(cvboston,s="lambda.min")
#lambda.min 指的是k-交叉检验中fold 最小的mselambda
predicterrorlasso <- dat$yvector[-trainindex]-cbind(1,xmatrix[-trainindex,])%*%coeflasso
mse.lasso <- sum(predicterrorlasso^2)/length(predicterrorlasso)
```



可见差距不明显.

```r
> c(mse.ols,mse.ridge,mse.lasso)
[1] 17.57527 17.53140 17.49156
```

他们的预测效果差距也不明显.

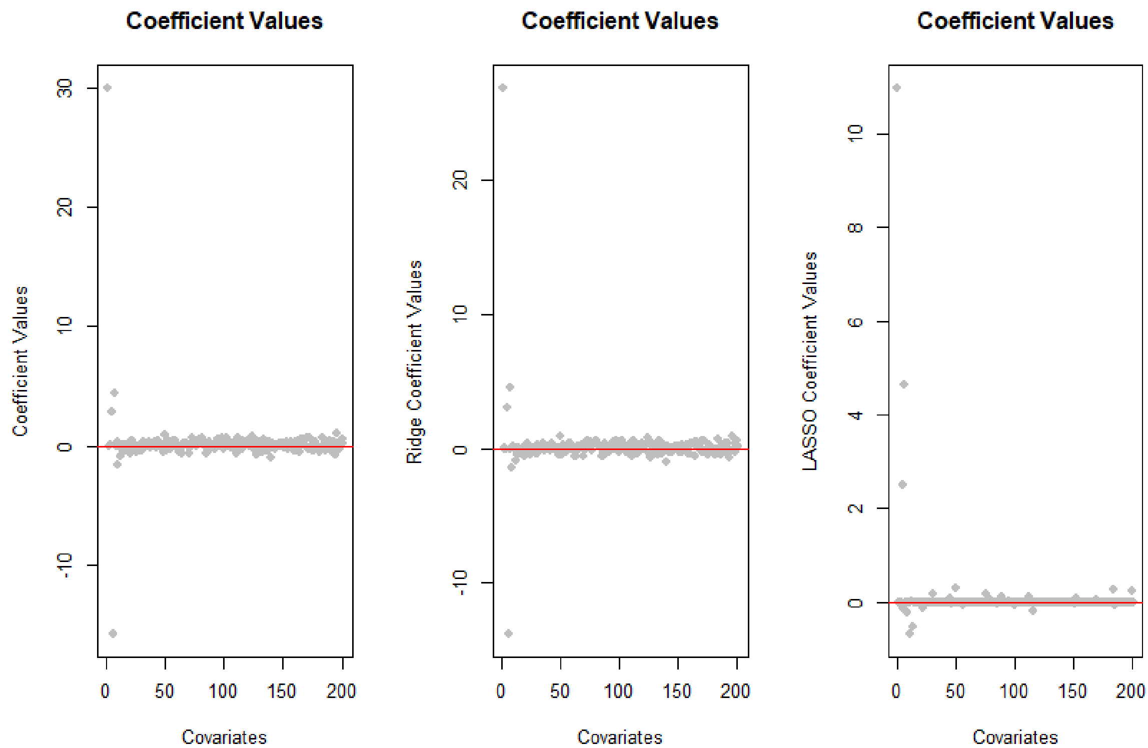
我们添加很多无关的变量到里面(p从13增加到200)

```r
set.seed(1)
nsample <- dim(BostonHousing)[1]
trainindex <- sample(1:nsample, floor(nsample*0.8))
data.noise<-matrix(rnorm(nsample*187),nsample,187)
datas<-cbind(BostonHousing,data.noise)
xmatrix <- model.matrix(medv~.,data=datas)[,-1]      #创立设计矩阵
yvector <- BostonHousing$medv
dat <- data.frame(yvector,xmatrix)
## linear regression
bostonlm <- lm(yvector~.,data=dat[trainindex,])
coefols<-bostonlm$coefficients
predicterror <- dat$yvector[-trainindex]-predict(bostonlm,dat[-trainindex,])#测试集检验
mse.ols <- sum(predicterror^2)/length(predicterror)
## ridge regression
lambdas <- seq(0,5,0.01)
lm0 <- lm.ridge(yvector~.,data=dat[trainindex,],lambda=lambdas)
coefridge <- coef(lm0)[which.min(lm0$GCV),]
predicterrorridge <- dat$yvector[-trainindex]-cbind(1,xmatrix[-trainindex,])%*%coefridge
mse.ridge <- sum(predicterrorridge^2)/length(predicterrorridge)

## lasso
cvboston <- cv.glmnet(x=xmatrix[trainindex,],y=yvector[trainindex])
coeflasso <- coef(cvboston,s="lambda.min")
predicterrorlasso <- dat$yvector[-trainindex]-cbind(1,xmatrix[-trainindex,])%*%coeflasso
mse.lasso <- sum(predicterrorlasso^2)/length(predicterrorlasso)

c(mse.ols,mse.ridge,mse.lasso)
cbind(coefols,coefridge,coeflasso)

par(mfrow = c(1,3))

plot(coefols, type = "n", xlab = "Covariates", ylab = "Coefficient Values", main = "Coefficient
    Values")
points(bostonlm$coefficients, pch = 16, col = "grey")
abline(h = 0, col = "red")


plot(coefridge, type = "n", xlab = "Covariates", ylab = "Ridge Coefficient Values", main = "
    Coefficient Values")
points(coefridge, pch = 16, col = "grey")
abline(h = 0, col = "red")

plot(coeflasso, type = "n", xlab = "Covariates", ylab = "LASSO Coefficient Values", main = "
    Coefficient Values")
points(coeflasso, pch = 16, col = "grey")
abline(h = 0, col = "red")
```

可见LASSO估计了大量的0.

```r
> c(mse.ols,mse.ridge,mse.lasso)
[1] 35.10212 33.49757 19.26522
```

LASSO的预测效果最好, 因为零系数更多(更合理, 随机误差应该无关).



3 其它的压缩估计量(shrinkage estimator)

有人称这种估计量为bridge estimator(Frank and Friedman, 1993):

$$\hat{\beta}(\lambda) = \arg\min_{b_0,b_1,\ldots,b_p} \left\{ \mathrm{Rss}(b_0, b_1, \ldots, b_p) + \lambda \sum_{j=1}^{p} |b_j|^q \right\}$$

对偶形式是:

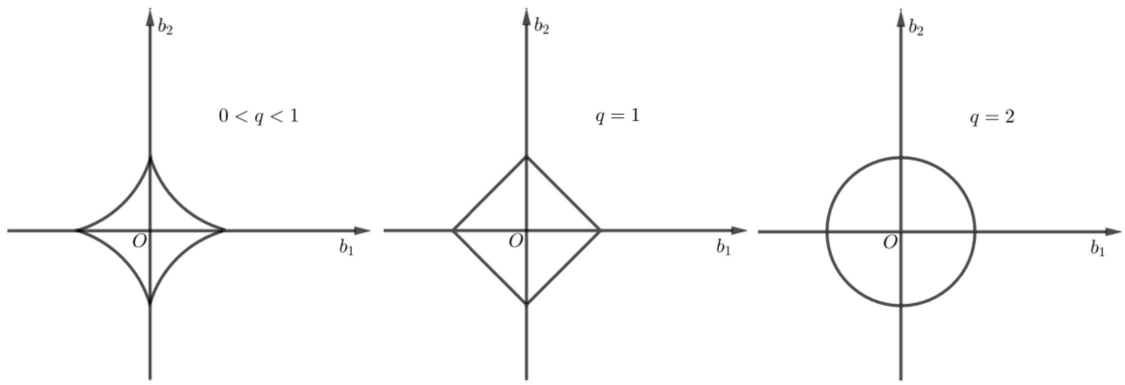
$$\hat{\beta}(t) = \arg\min_{b_0,b_1,\ldots,b_p} \mathrm{Rss}(b_0, b_1, \ldots, b_p)$$

$$\text{s.t. } \sum_{j=1}^{p} |b_j|^q \leq t.$$

Zou and Hastie(2005)提出来弹性网(elastic net), 它组合了岭回归和LASSO.

$$\hat{\beta}^{\mathrm{enet}}(\lambda, \alpha) = \arg\min_{b_0,b_1,\ldots,b_p} \left[\mathrm{Rss}(b_0, b_1, \ldots, b_p) + \lambda \sum_{j=1}^{p} \left\{ \alpha b_j^2 + (1-\alpha)|b_j| \right\} \right]$$

- 由于限制部分是不光滑的, 因此结果都稀疏性类似LASSO一样可以满足.
- 再由于ridge penalty, 它在解决多重共线性问题上也有更好的表现, 相比岭估计和LASSO.
- 在R语言中可以用`glmnet`来实现.



(a) $0 < q < 1$ (b) $q = 1$ (c) $q = 2$