第 9 章 惩罚回归

9.1 高维回归的挑战

大数据(big data)的一种表现形式为“高维数据”(high dimensional data)，即特征向量$\mathbf{x}_i$的维度p大于样本容量n，也称为“数据丰富的环境”(data-rich environment)。

比如，某研究收集了 100 位病人的信息，其中每位病人均有 2 万条基因(即 2 万个变量)的数据，需要研究哪些基因导致了某种疾病。假设受成本限制，样本容量$n=100$难以再扩大，而变量个数p远大于样本容量。

考虑传统的线性回归模型：

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \quad (i = 1, \cdots, n) \tag{9.1}$$

更简洁地，写为矩阵的形式：

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{9.2}$$

其中，响应向量$\mathbf{y} \equiv (y_1 \ y_2 \cdots y_n)'$，残差向量$\boldsymbol{\varepsilon} \equiv (\varepsilon_1 \ \varepsilon_2 \cdots \varepsilon_n)'$，而$n \times p$数据矩阵(data matrix)$\mathbf{X}$包含所有特征向量$\mathbf{x}_i$的信息：

$$\mathbf{X} \equiv \begin{pmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \vdots \\ \mathbf{x}_n' \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \ddots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{9.3}$$

对于高维数据，由于$n < p$，故矩阵$\mathbf{X}$不满列秩(存在严格多重共线性)，因为

$$\mathbf{X}\text{的列秩} = \mathbf{X}\text{的行秩} \leq n < p \tag{9.4}$$

故逆矩阵$(\mathbf{X}'\mathbf{X})^{-1}$不存在，故 OLS 不存在唯一解，无法进行 OLS 回归。

直观上，对于$n < p$的高维数据，可用来解释y_i的特征变量$(x_{i1}\ x_{i2} \cdots x_{ip})$太多。

如果使用传统的 OLS 回归，虽可得到完美的样本内拟合(in-sample fit)，但外推预测的效果则可能很差。

例 假设 $n = p = 100$。进一步，假定这 100 个特征变量$\mathbf{x}$与响应变量y毫无关系(比如，相互独立)，但将y对$\mathbf{x}$作 OLS 回归，也能得到拟合优度$R^2 = 1$的完美拟合。

在特征向量$\mathbf{x}$所存在的 100 维空间中，最多只可能有 100 个线性无关的向量，而加入$\mathbf{y} \equiv (y_1\ y_2 \cdots y_n)'$之后，必然导致线性相关，即$\mathbf{y}$可由这 100 个特征变量所线性表出，故残差为 0，而$R^2 = 1$。

根据此样本数据估计的回归函数，将毫无外推预测的价值；因为y与$\mathbf{x}$事实上相互独立。

这种拟合显然过度了，故名“过拟合”(overfit)，因为它不仅拟合了数据中的信号(signal)，而且拟合了数据中的很多噪音(noise)。

在此极端例子中，由于数据中全是噪音而毫无信号，故 OLS 完美地拟合了数据中的噪音，自然毫无意义。

对于传统的低维数据，样本容量大于变量个数($n > p$)，一般很少出现“严格多重共线性”(strict multicollinearity)。即使偶然出现，只要将多余的变量去掉就行。

对于$n < p$的高维数据，则严格多重共线性成为常态。

比如，在任意$(n+1)$个变量之间，一般就存在严格多重共线性。

此时，简单地丢掉导致多重共线性的变量将无济于事，因为需要扔掉很多变量。

比如，假设样本为 100 个病人，但有 2 万个基因的变量，如果通过去掉变量消除严格多重共线性，则难免将婴儿与洗澡水一起倒掉。

9.2 岭回归

作为高维回归的方法之一，**岭回归**(Ridge Regression)最早由 Hoerl and Kennard (1970)提出，其出发点正是为了解决多重共线性。

当时还几乎没有高维数据。在传统的低维回归(low-dimensional regression)，虽然严格多重共线性很少见，但近似(不完全)的多重共线性却不时出现，即特征变量$(x_1 \cdots x_p)$之间高度相关，比如相关系数大于 0.9。

矩阵$\mathbf{X}'\mathbf{X}$变得“几乎”不可逆(类似于 1 除以非常小的数)，导致$(\mathbf{X}'\mathbf{X})^{-1}$变得很大，使得 OLS 估计量$\hat{\boldsymbol{\beta}}_{OLS} \equiv (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$的方差也变得很大。

Hoerl and Kennard (1970)的解决方案为，在矩阵$\mathbf{X}'\mathbf{X}$的主对角线上都加上某常数$\lambda > 0$，以缓解多重共线性，使所得矩阵$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})$变得“正常”。由此可得岭回归估计量：

$$\hat{\boldsymbol{\beta}}_{ridge} \equiv (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} \qquad (9.5)$$

岭回归只是在 OLS 表达式中加入“山岭”$\lambda\mathbf{I}$，故名“岭回归”(ridge regression)。

其中，参数λ称为**调节参数**(tuning parameter)。

由于 OLS 估计量是无偏的(unbiased)，即$\mathrm{E}(\hat{\boldsymbol{\beta}}_{OLS})=\boldsymbol{\beta}$，故凭空加上此“山岭”$\lambda\mathbf{I}$后，岭回归估计量其实是有偏的(biased)，其偏差为

$$\mathrm{Bias}(\hat{\boldsymbol{\beta}}_{ridge})\equiv\mathrm{E}(\hat{\boldsymbol{\beta}}_{ridge})-\boldsymbol{\beta}\neq\mathbf{0} \qquad (9.6)$$

我们的目标是最小化均方误差(Mean Squared Errors，简记 MSE)，而并非强求无偏估计。

对于任意估计量$\hat{\boldsymbol{\beta}}$，其均方误差可分解为方差与偏差平方之和(参见附录 A9.1)：

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}) \equiv \mathrm{E}\left[(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})'\right] = \mathrm{Var}(\hat{\boldsymbol{\beta}}) + \left[\mathrm{Bias}(\hat{\boldsymbol{\beta}})\right]\left[\mathrm{Bias}(\hat{\boldsymbol{\beta}})\right]' \tag{9.7}$$

在一维情况下，此分解公式可简化为

$$\mathrm{MSE}(\hat{\beta}) = \mathrm{Var}(\hat{\beta}) + \left[\mathrm{Bias}(\hat{\beta})\right]^2 \tag{9.8}$$

使均方误差最小化，可视为在方差与偏差之间进行权衡(trade-off)。

比如，一个无偏估计量(偏差为 0)，如果方差很大，则可能不如一个虽然有偏差但方差却很小的估计量。

在(严格)多重共线性的情况下，虽然 OLS 估计量无偏，但其方差太大(无穷大)，而岭回归虽有少量偏差，但可大幅减少方差，使得岭回归估计量的均方误差(MSE)可能比 OLS 更小，参见图 9.1。

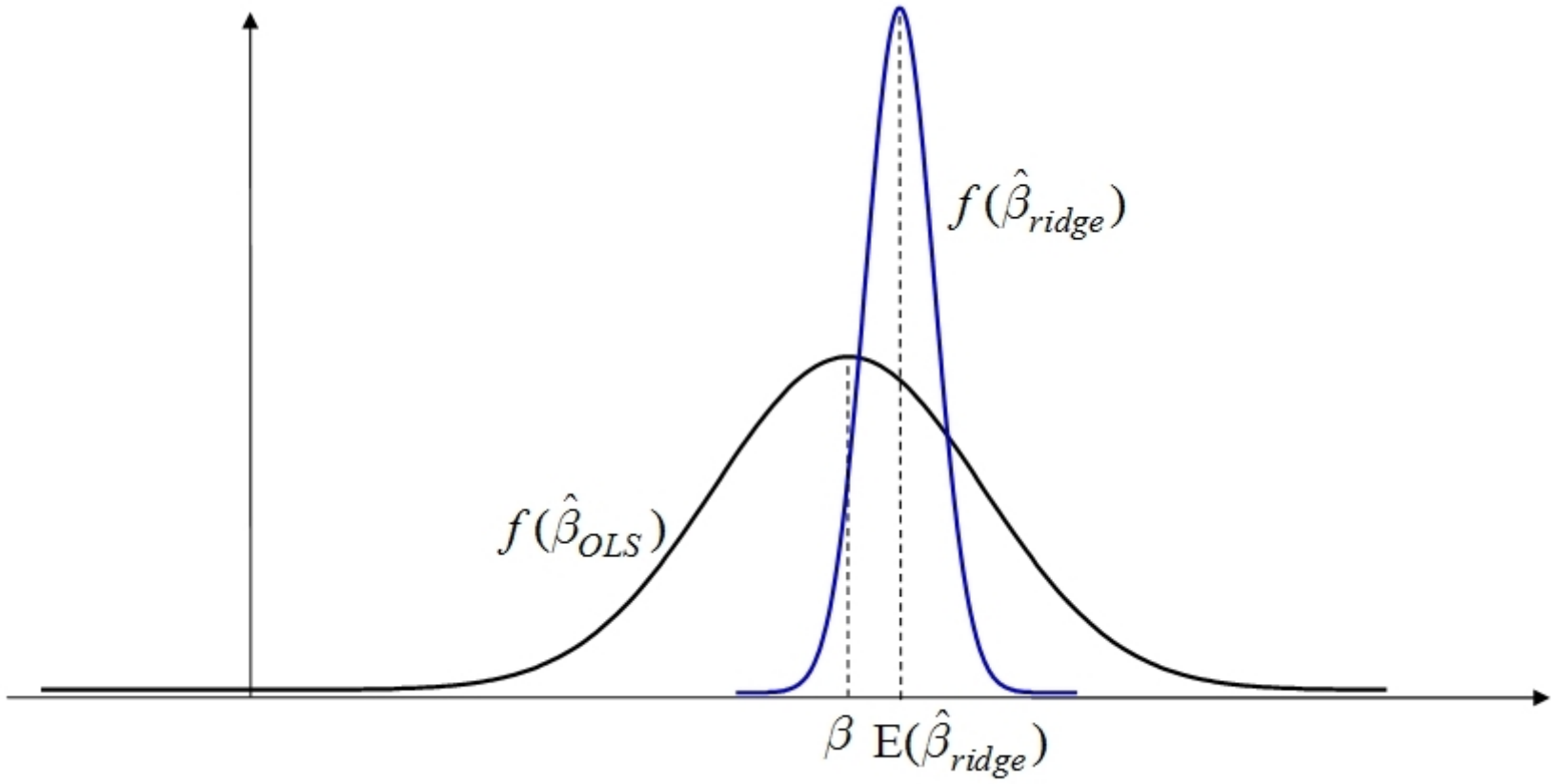

图 9.1 无偏估计量$\hat{\beta}_{OLS}$与有偏估计量$\hat{\beta}_{ridge}$之间的权衡

岭回归的理论基础是什么？

在严格多重共线性的情况下，不存在唯一的 OLS 估计量$\hat{\boldsymbol{\beta}}_{OLS}$。

这意味着有许多$\hat{\boldsymbol{\beta}}_{OLS}$，都能使残差平方和等于 0，而拟合优度$R^2=1$。

为得到参数向量$\boldsymbol{\beta}$的唯一解，须对其取值范围有所限制，进行所谓**正则化**(regularization)。

为此，考虑在损失函数(loss function)中加入“惩罚项”(penalty term)，进行**惩罚回归**(penalized regression)：

$$\min_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta}) = \underbrace{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}_{SSR} + \underbrace{\lambda\|\boldsymbol{\beta}\|_2^2}_{penalty} \qquad (9.9)$$

其中，损失函数的第 1 项依然为残差平方和(SSR)；

第 2 项为惩罚项，也称为“正则项”(regularization)。

$\lambda \geq 0$为“调节参数”(tuning parameter)，控制惩罚的力度。

$\|\boldsymbol{\beta}\|_2$为参数向量$\boldsymbol{\beta}$的长度，即$\boldsymbol{\beta}$到原点的欧氏距离，也称为 2-范数(L_2 norm)：

$$\|\boldsymbol{\beta}\|_2 \equiv \sqrt{\beta_1^2 + \cdots + \beta_p^2} \qquad (9.10)$$

将惩罚项写为$\lambda\|\boldsymbol{\beta}\|_2^2 = \lambda(\beta_1^2 + \cdots + \beta_p^2) = \lambda\boldsymbol{\beta}'\boldsymbol{\beta}$，则损失函数可写为

$$\min_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\boldsymbol{\beta} \qquad (9.11)$$

由于惩罚项$\lambda\boldsymbol{\beta}'\boldsymbol{\beta}$只是简单的二次型，使用向量微分的规则，得到相应的一阶条件：

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0} \qquad (9.12)$$

经移项整理可得：

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}'\mathbf{y} \qquad (9.13)$$

所得最优解正是岭回归估计量：

$$\hat{\boldsymbol{\beta}}_{ridge}(\lambda) \equiv (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} \qquad (9.14)$$

9.3 岭回归的计算

对于 OLS 回归，变量单位对于回归系数没有实质性影响。比如，当变量x_j的单位从“万元”变为“亿元”时，x_j的取值将缩小一万倍，而其回归系数$\hat{\beta}_j$将相应增大一万倍。

由于岭回归通过惩罚项$\lambda\|\boldsymbol{\beta}\|_2^2$，惩罚过大的回归系数，故变量单位对于岭回归有实质性影响。

另外，我们一般不希望惩罚常数项，因为常数项仅代表响应变量y的平均值。

在进行岭回归(或其他形式的惩罚回归)时，一般先将每个变量$x_j\ (j=1,\cdots,p)$标准化，即减去其均值$\overline{x}_j$，再除以标准差$sd(x_j)$，然后使用标准化的变量$(x_j-\overline{x}_j)/sd(x_j)$进行岭回归。

在标准化之后，特征变量的样本均值为 0。

对响应变量y进行中心化，即变换为$(y-\overline{y})$，故响应变量的样本均值也为 0。

可以证明，如果不惩罚常数项，则对于常数项的岭回归估计就是响应变量y的样本均值(已中心化为 0)，故常数项不必放入岭回归方程中。

不失一般性，以一元回归为例：

$$y_i = \alpha + \beta x_i + \varepsilon_i \qquad (9.15)$$

岭回归的目标函数为

$$\min_{\alpha,\,\beta} \; L(\alpha,\beta) = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 + \lambda \beta^2 \qquad (9.16)$$

其中，$\lambda \geq 0$，且不惩罚常数项α。对α求偏导数可得一阶条件：

$$\frac{\partial L}{\partial \alpha}=\frac{\partial \sum_{i=1}^{n}(y_i-\alpha-\beta x_i)^2}{\partial \alpha}+\underbrace{\frac{\partial(\lambda\beta^2)}{\partial \alpha}}_{=0} \tag{9.17}$$

$$=-2\sum_{i=1}^{n}(y_i-\alpha-\beta x_i)=0$$

在上式两边同除$(-2n)$，并移项可得：

$$0=\bar{y}=\hat{\alpha}+\hat{\beta}\underbrace{\bar{x}}_{=0}=\hat{\alpha} \tag{9.18}$$

在上式中，由于将变量标准化或中心化，$\overline{x}=0$，$\overline{y}=0$，故在不惩罚常数项的情况下，对常数项的岭回归估计量为$\hat{\alpha}=0$。

这意味着，无须在岭回归中放入常数项。

9.4 岭回归的几何解释

由于岭回归的目标函数包含对过大参数的惩罚项，故岭回归为**收缩估计量**(shrinkage estimator)。

调节参数λ也称**收缩参数**(shrinkage parameter)。这是因为，与 OLS 估计量相比，岭回归估计量更为向原点收缩。

岭回归的目标函数可等价地写为如下有约束的极值问题：

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \\ &s.t. \ \|\boldsymbol{\beta}\|_2^2 \le t \end{aligned} \tag{9.19}$$

其中，$t \ge 0$为某常数。

对于此约束极值问题，可引入拉格朗日乘子函数，并以λ作为其乘子：

$$\min_{\boldsymbol{\beta},\,\lambda} \ \tilde{L}(\boldsymbol{\beta}) = (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda(\|\boldsymbol{\beta}\|_2^2 - t) \tag{9.20}$$

对$\boldsymbol{\beta}$求偏导数，并注意到$\|\boldsymbol{\beta}\|_2^2 = \boldsymbol{\beta}'\boldsymbol{\beta}$，可得一阶条件：

$$\frac{\partial \tilde{L}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = \mathbf{0} \qquad (9.21)$$

上式与岭回归的一阶条件(9.12)完全相同。因此，约束极值问题(9.19)与岭回归的最小化问题(9.9)等价。

给定最小化问题(9.9)的λ值，则存在约束极值问题(9.19)唯一的t值，使得二者的最优解$\hat{\boldsymbol{\beta}}$相同；反之亦然。

下面从几何角度考察约束极值问题(9.19)。其目标函数为残差平方和，与 OLS 相同。

在$\boldsymbol{\beta}$的参数空间(parameter space)，约束条件$\|\boldsymbol{\beta}\|_2^2 \leq t$对应于以$\sqrt{t}$为半径的圆球之内部。

以二元回归为例，则$\boldsymbol{\beta}=(\beta_1\ \beta_2)'$，约束条件$\|\boldsymbol{\beta}\|_2^2 \leq t$可写为$\beta_1^2+\beta_2^2 \leq t$，参见图 9.2。

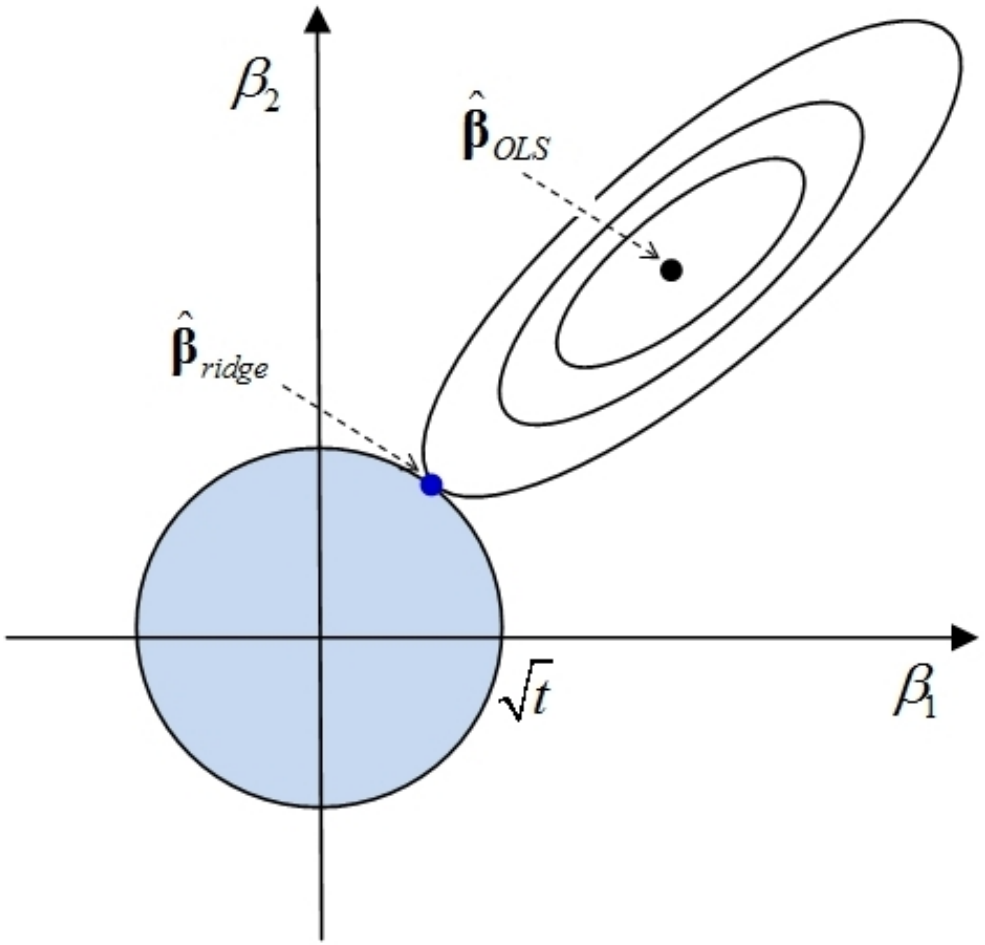

图 9.2　岭回归的约束条件示意图

在图 9.2 中，满足约束条件$\beta_1^2+\beta_2^2\leq t$的可行解(feasible solutions)，在以$\sqrt{t}$为半径的圆球之内部。

图中黑点为 OLS 估计量$\hat{\boldsymbol{\beta}}_{OLS}$(如 OLS 估计量不唯一，则构成一个集合)。

围绕$\hat{\boldsymbol{\beta}}_{OLS}$的一圈圈椭圆为残差平方和的“等值线”(contour)或“水平集”(level set)。

直观上，可将$\hat{\boldsymbol{\beta}}_{OLS}$想象为“山谷”的最低点，而越靠近$\hat{\boldsymbol{\beta}}_{OLS}$的等值线，其残差平方和越低。

在可行集中最小化残差平方和，其最优解$\hat{\boldsymbol{\beta}}_{ridge}$必然发生于等值线与圆周“$\beta_1^2+\beta_2^2=t$”相切的位置，参见图 9.2 中的蓝点。

由于约束集为圆(球)，而等值线是椭圆(球)，故二者相切的位置一般不会碰巧在坐标轴上。

岭回归通常只是将所有的回归系数都收缩，而不会让某些回归系数严格等于 0。

在高维回归中，由于变量太多，如果所有变量的系数都非零，将使得模型的解释变得很困难。比如，如何同时考察 2 万个回归系数？

9.5 套索估计量

在进行高维回归时，有时希望从大量的特征变量中，筛选出真正对y有影响的少数变量。比如，从 2 万个基因中，找到真正影响疾病的少数基因。

此时，一般期待真实模型(true model)，或数据生成过程(data generating process)，为**稀疏模型**(sparse model)。

需要一个估计量，能挑选出那些真正有影响的(基因)变量，而使其他无影响或影响微弱的(基因)变量的回归系数变为 0。

为此，Tibshirani(1996)提出**套索估计量**(Least Absolute Shrinkage and Selection Operator，简记 LASSO)，将岭回归惩罚项中的“2-范数”改为“1-范数”：

$$\min_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta}) = \underbrace{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}_{SSR} + \underbrace{\lambda\|\boldsymbol{\beta}\|_1}_{penalty} \qquad (9.22)$$

其中，$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p}|\beta_j|$为参数向量$\boldsymbol{\beta}$的 **1−范数**($L_1$norm)，即$\boldsymbol{\beta}$各分量的绝对值之和。

由于损失函数包括惩罚项$\lambda\left\|\boldsymbol{\beta}\right\|_1$，故 Lasso 也是“收缩估计量”(shrinkage estimator)，其最优解$\hat{\boldsymbol{\beta}}_{Lasso}$比 OLS 估计量$\hat{\boldsymbol{\beta}}_{OLS}$更为向原点收缩。

进一步，由于对$\boldsymbol{\beta}$各分量的绝对值之和进行惩罚，故称为“绝对值收缩”(absolute shrinkage)。

类似于岭回归，Lasso 最小化问题也可等价地写为如下约束极值问题：

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) \qquad (9.23)$$
$$s.t. \ \left\|\boldsymbol{\beta}\right\|_1 \le t$$

其中，$t \ge 0$为某常数。

此约束极值问题的约束集不再是圆球，而是菱形(钻石形)或高维的菱状体。

仍以二元回归为例，则$\boldsymbol{\beta}=(\beta_1,\beta_2)'$。

约束条件$\|\boldsymbol{\beta}\|_1 \le t$可写为$|\beta_1|+|\beta_2| \le t$，参见图 9.3。

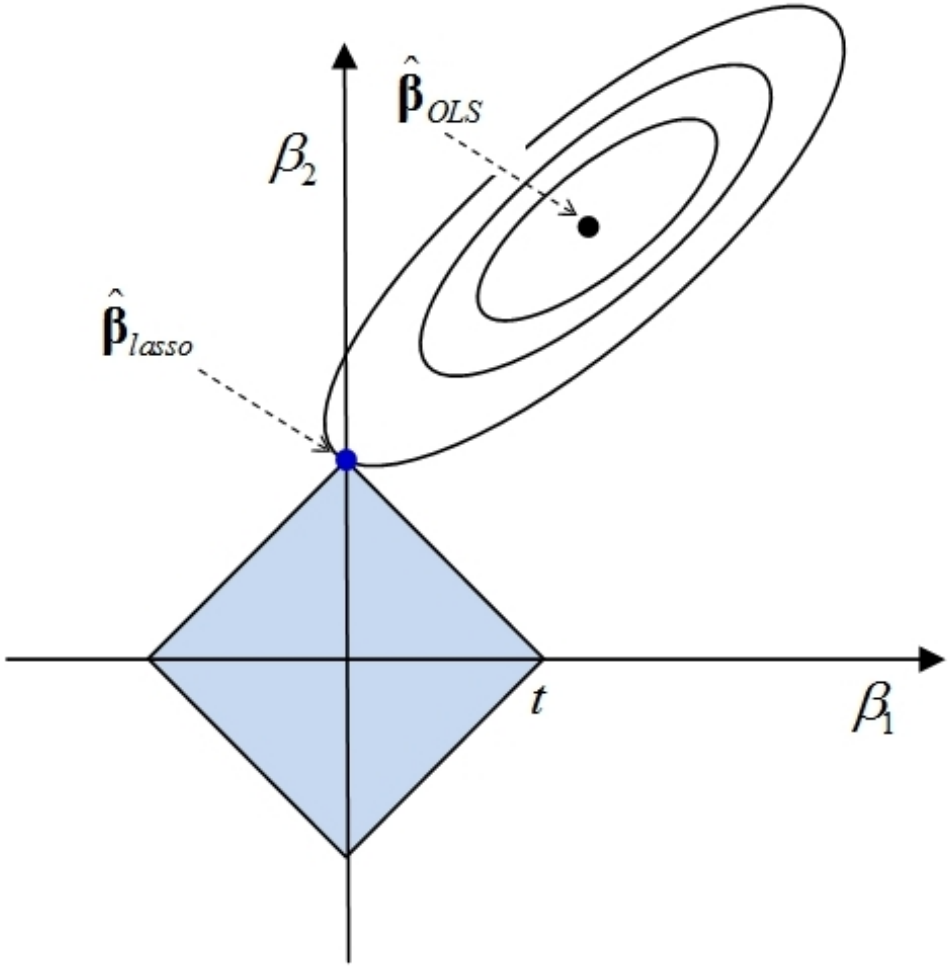

图 9.3　Lasso 的约束条件示意图

在图 9.3 中，满足约束条件$|\beta_1|+|\beta_2|\leq t$的可行解，在图中菱形的内部。

图中黑点依然为 OLS 估计量$\hat{\boldsymbol{\beta}}_{OLS}$，而围绕$\hat{\boldsymbol{\beta}}_{OLS}$的一圈圈椭圆仍为残差平方和的等值线。

为了在可行集中最小化残差平方和，最优解$\hat{\boldsymbol{\beta}}_{lasso}$必然发生于等值线与菱形“$|\beta_1|+|\beta_2|\leq t$”相切的位置，即图 9.3 中的红点。

由于 Lasso 的约束集为带尖角的菱形(而菱形的顶点恰好在坐标轴上)，故等值线较易与约束集相切于坐标轴的位置，导致 Lasso 估计量的某些回归系数严格等于 0，从而得到“稀疏解”(sparse solution)。

Lasso 的这种独特性质，使得它具备“筛选变量”(variable selection)的功能，故也称为“筛选算子”(selection operator)。

由于 Lasso 为“绝对值收缩”(absolute shrinkage)，故合称为“最小绝对值收缩与筛选算子”(least absolute shrinkage and selection operator)，简记 LASSO。

在英文中，Lasso 一词的原意为“套索”，而套索本来就有收缩的功能，故中文译为“套索估计量”。

Lasso 与岭回归孰优孰劣？

从预测的角度，如果真实模型(或数据生成过程)确实是稀疏的，则 Lasso 一般更优。

但如果真实模型并不稀疏，则岭回归的预测效果可能优于 Lasso。

在实践中，一般并不知道模型是否稀疏，可用“交叉验证”(cross-validation)进行选择，参见下文的弹性网估计量。

从模型易于解释(interpretability)的角度，则 Lasso 显然是赢家，因为岭回归一般只是收缩回归系数，并不具备变量筛选的功能。

9.6 套索估计量的计算

由于 Lasso 使用带绝对值的 1-范数，而绝对值函数不光滑(在原点处有尖点，参见图 9.4)，使得 Lasso 的目标函数不可微，故一般情况下不存在解析解。

幸运的是，Lasso 的目标函数依然为凸函数(convex function)，因为绝对值函数为凸函数；故存在很有效率的数值迭代算法。

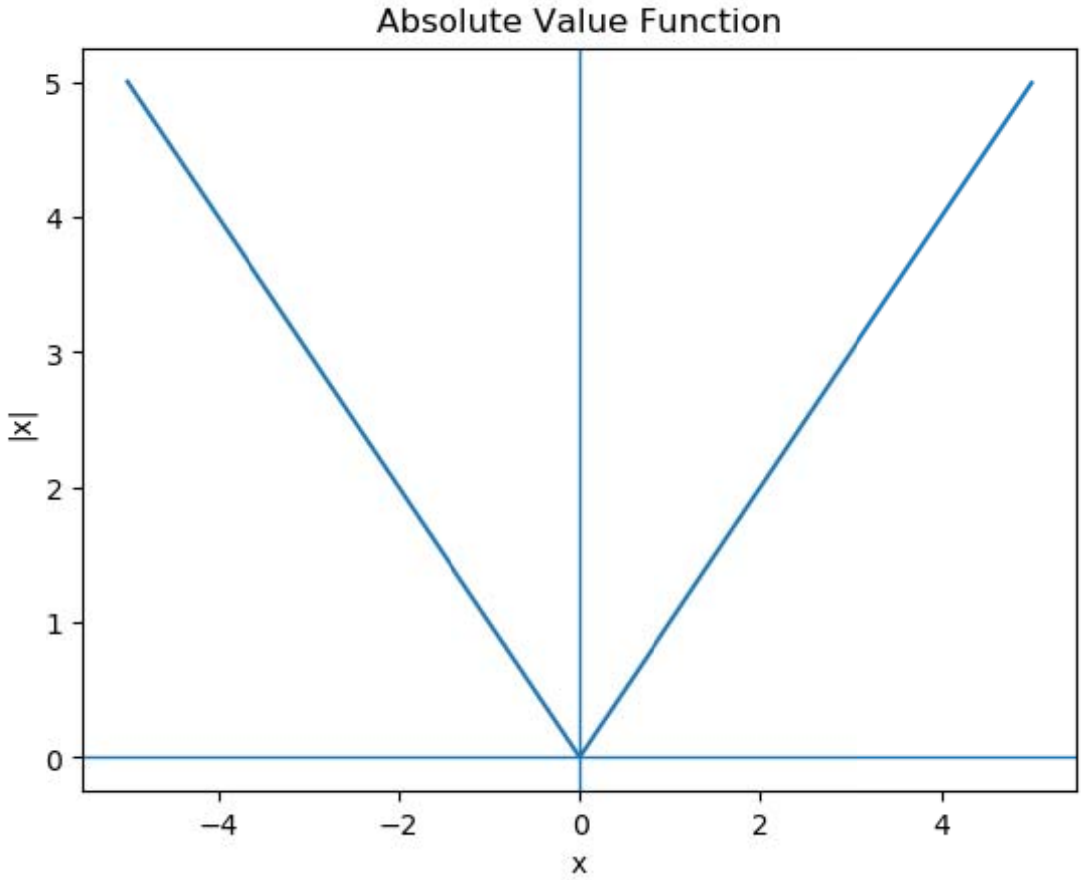

图 9.4　绝对值函数

在 2008 年之前，最好的 Lasso 算法为“最小角回归”(least angle regression)，由 Efron et al. (2004)所提出。

目前最有效率的 Lasso 算法则为**坐标下降法**(coordinate descent algorithm)，由 Friedman et al. (2007)与 Wu and Lange(2008)提出。

所谓坐标下降法，就是依次沿着一个坐标轴的方向进行最优化，使得损失函数下降，直至最低点。

假设损失函数为

$$L(\boldsymbol{\beta}) = f(\beta_1, \beta_2, \cdots, \beta_p) \qquad (9.24)$$

假定在迭代过程中，$\boldsymbol{\beta}$的当前取值为$(\beta_1^*, \beta_2^*, \cdots, \beta_p^*)$。

首先，给定$(\beta_2^*, \cdots, \beta_p^*)$，将函数$f(\beta_1, \beta_2^*, \cdots, \beta_p^*)$针对$\beta_1$进行一元最小化，得到最优解$\beta_1^{**}$。

其次，给定$(\beta_1^{**}, \beta_3^*, \cdots, \beta_p^*)$，将函数$f(\beta_1^{**}, \beta_2, \beta_3^*, \cdots, \beta_p^*)$针对$\beta_2$进行一元最小化，得到最优解$\beta_2^{**}$；

以此类推，直至收敛，参见图 9.5。

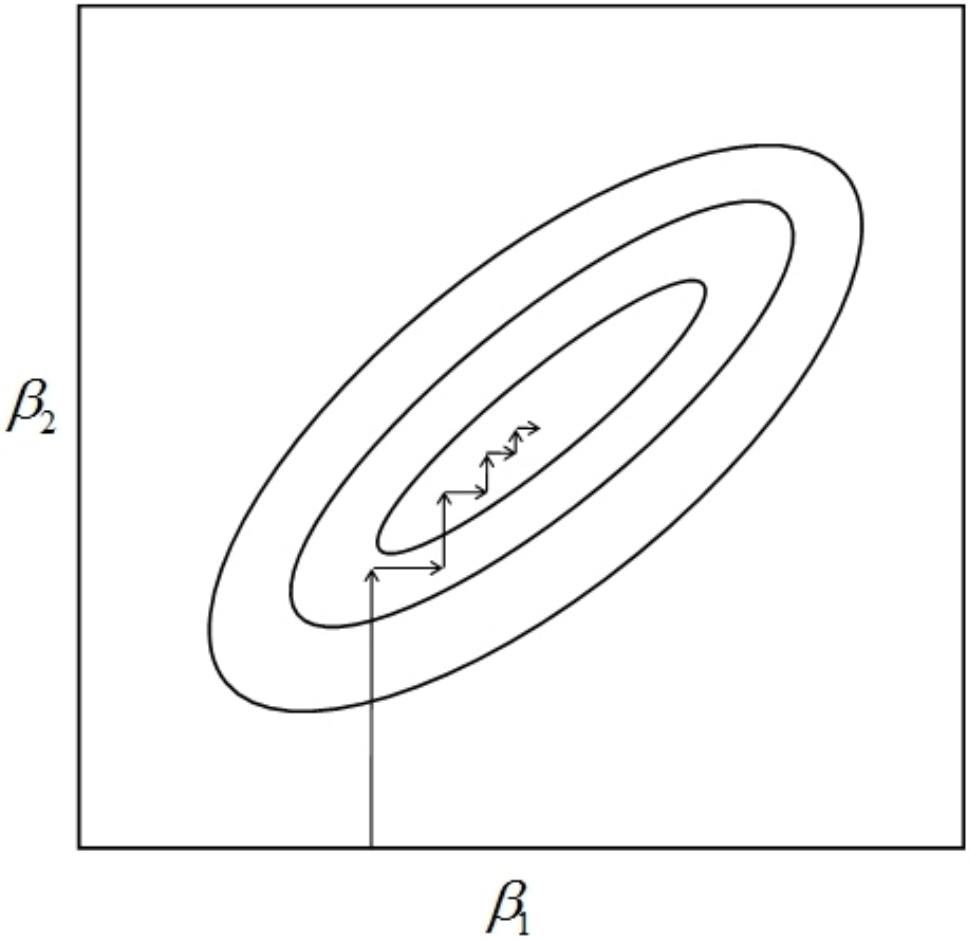

图 9.5　坐标下降法的示意图

之所以套索估计量适用于坐标下降法，因为一维的 Lasso 问题有解析解。

在进行坐标下降法的每步迭代时，可直接代入解析表达式，不必进行梯度下降，故更有效率。

更一般地，只要设计矩阵$\mathbf{X}$的每一列均为单位向量，且相互正交，即所谓**标准正交设计**(orthonormal design)，则 Lasso 都有解析解。

对于一维的 Lasso 问题，其设计矩阵$\mathbf{X}$只有一列，故只要将此列向量的长度标准化为 1，即为标准正交设计，因此有解析解。

考虑以下一维的 Lasso 问题：

$$\min_{\beta} \ L(\beta)=\sum_{i=1}^{n}(y_i-\beta x_i)^2+\lambda\left|\beta\right| \qquad (9.25)$$

虽然无法对上式中的绝对值$\left|\beta\right|$求导数，但可求其“次导数”(subderivative)，进而得到“次微分”(subdifferential)，参见附录 A9.2。

由此可证明，一元 Lasso 问题的最优解可写为 OLS 估计量$\hat{\beta}_{OLS}$的函数：

$$\hat{\beta}_{lasso}=sign(\hat{\beta}_{OLS})\cdot\left(\left|\hat{\beta}_{OLS}\right|-\lambda/2\right)_{+} \qquad (9.26)$$

其中，$sign(\cdot)$为“符号函数”(sign function)，即

$$sign(x)=\begin{cases}1 & if\ x>0\\ 0 & if\ x=0\\ -1 & if\ x<0\end{cases} \tag{9.27}$$

而“$(\cdot)_+$”为取“正部”(positive part)的运算，即

$$(x)_+=\begin{cases}x & if\ x\geq 0\\ 0 & if\ x<0\end{cases} \tag{9.28}$$

在机器学习中，函数“$(\cdot)_+$”也称为**修正线性单元**(Rectified Linear Unit，简记 ReLU)或**线性整流函数**，广泛用于人工神经网络，参见图 9.6。

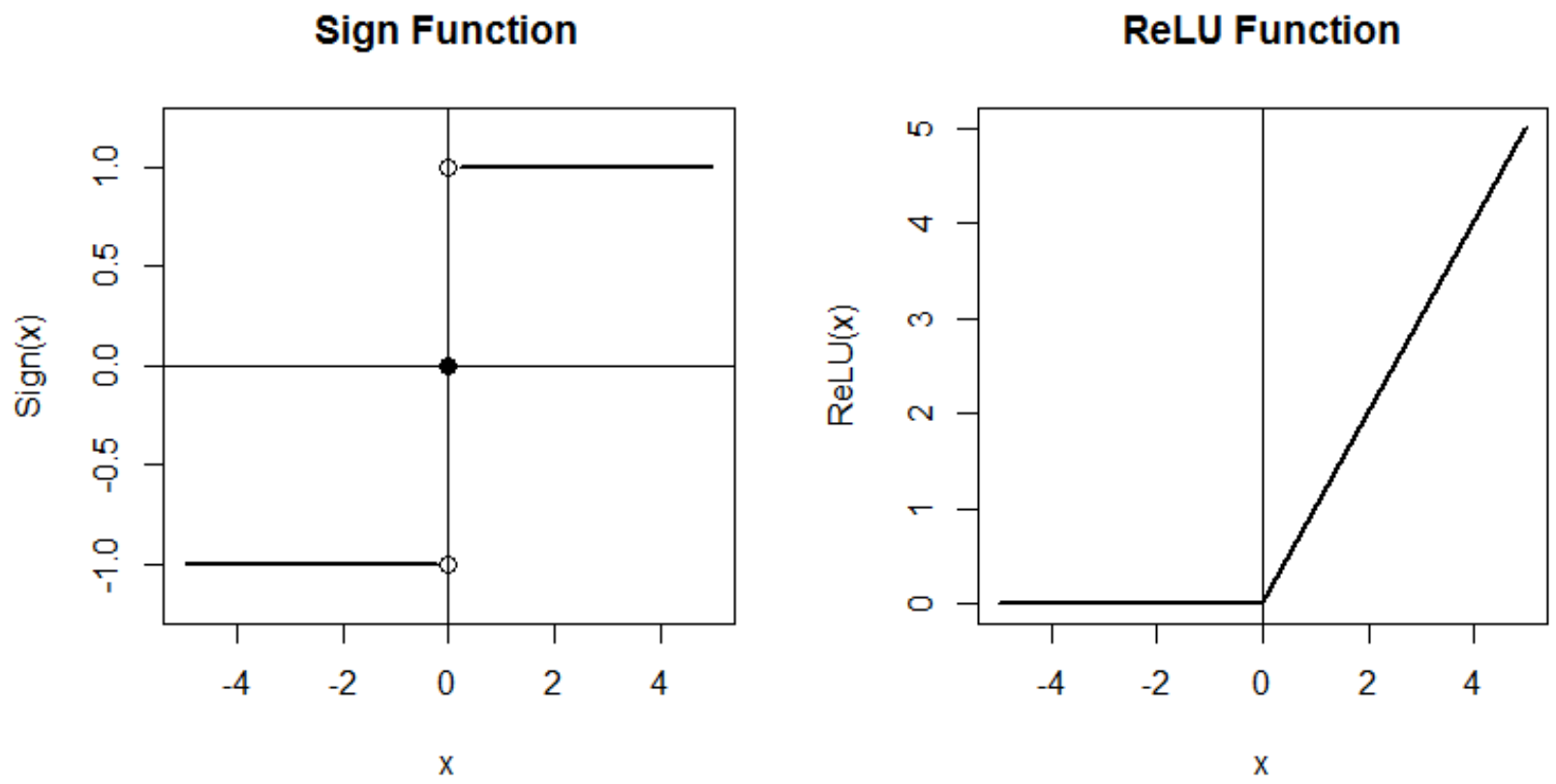

图 9.6　符号函数与线性整流函数

将表达式(9.26)写为分段函数，可将一元 Lasso 的最优解$\hat{\beta}_{Lasso}$写为

$$\hat{\beta}_{Lasso}=\begin{cases}\hat{\beta}_{OLS}-\lambda/2 & if\ \hat{\beta}_{OLS}>\lambda/2\\ 0 & if\ \left|\hat{\beta}_{OLS}\right|\le\lambda/2\\ \hat{\beta}_{OLS}+\lambda/2 & if\ \hat{\beta}_{OLS}<-\lambda/2\end{cases} \tag{9.29}$$

上式将 Lasso 估计量$\hat{\beta}_{Lasso}$表示为 OLS 估计量$\hat{\beta}_{OLS}$的分段函数，参见图 9.7。

如果$\left|\hat{\beta}_{OLS}\right| \leq \lambda/2$，则 Lasso 直接将 OLS 估计量$\hat{\beta}_{OLS}$收缩至 0。

如果$\hat{\beta}_{OLS} > \lambda/2$，则 Lasso 使 OLS 估计量向原点收缩$\lambda/2$，即$\hat{\beta}_{lasso} = \hat{\beta}_{OLS} - \lambda/2$。

如果$\hat{\beta}_{OLS} < -\lambda/2$，则 Lasso 使 OLS 估计量向原点收缩$\lambda/2$，即$\hat{\beta}_{lasso} = \hat{\beta}_{OLS} + \lambda/2$。

故 Lasso 也称为**软门限算子**(soft thresholding operator)。

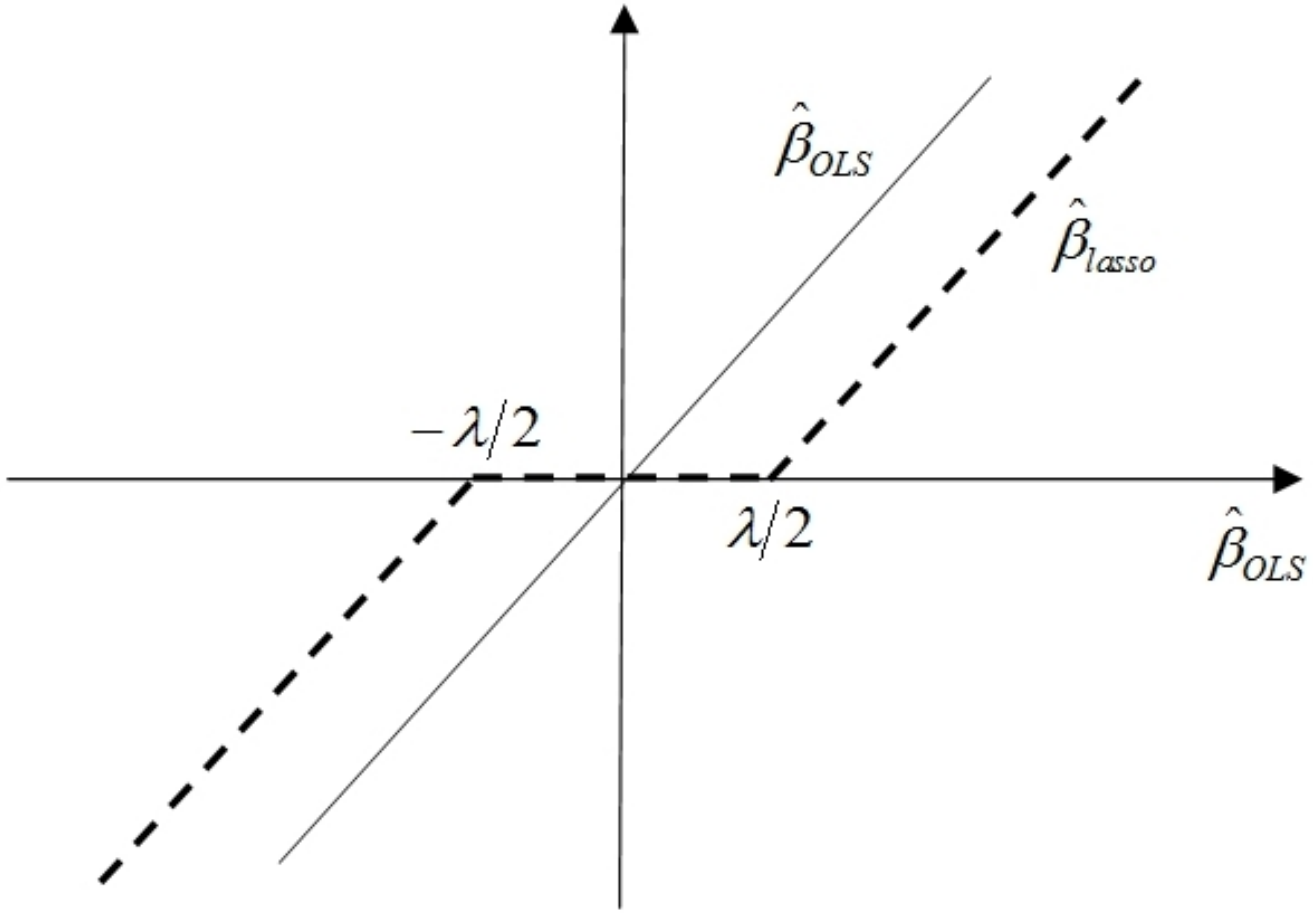

图 9.7 作为软门限算子的 Lasso 估计量

“软门限算子”的称呼与**硬门限算子**(hard thresholding operator)相对。当$\left|\hat{\beta}_{OLS}\right| \leq \lambda/2$时，二者作用相同。

而当$\left|\hat{\beta}_{OLS}\right| > \lambda/2$时，硬门限算子保持$\hat{\beta}_{OLS}$不变。故在$\hat{\beta}_{OLS} = \pm\lambda/2$处，硬门限算子存在断点(不连续)，其函数值突然变为 0，故称为“硬门限算子”，参见图 9.8。

在统计学中，传统上以**最优子集**(best subset)的方法筛选变量，这就是一种硬门限算子。如果某变量被选入模型，则其回归系数为 OLS 系数；而一旦该变量选不上，则其回归系数立即变为 0。

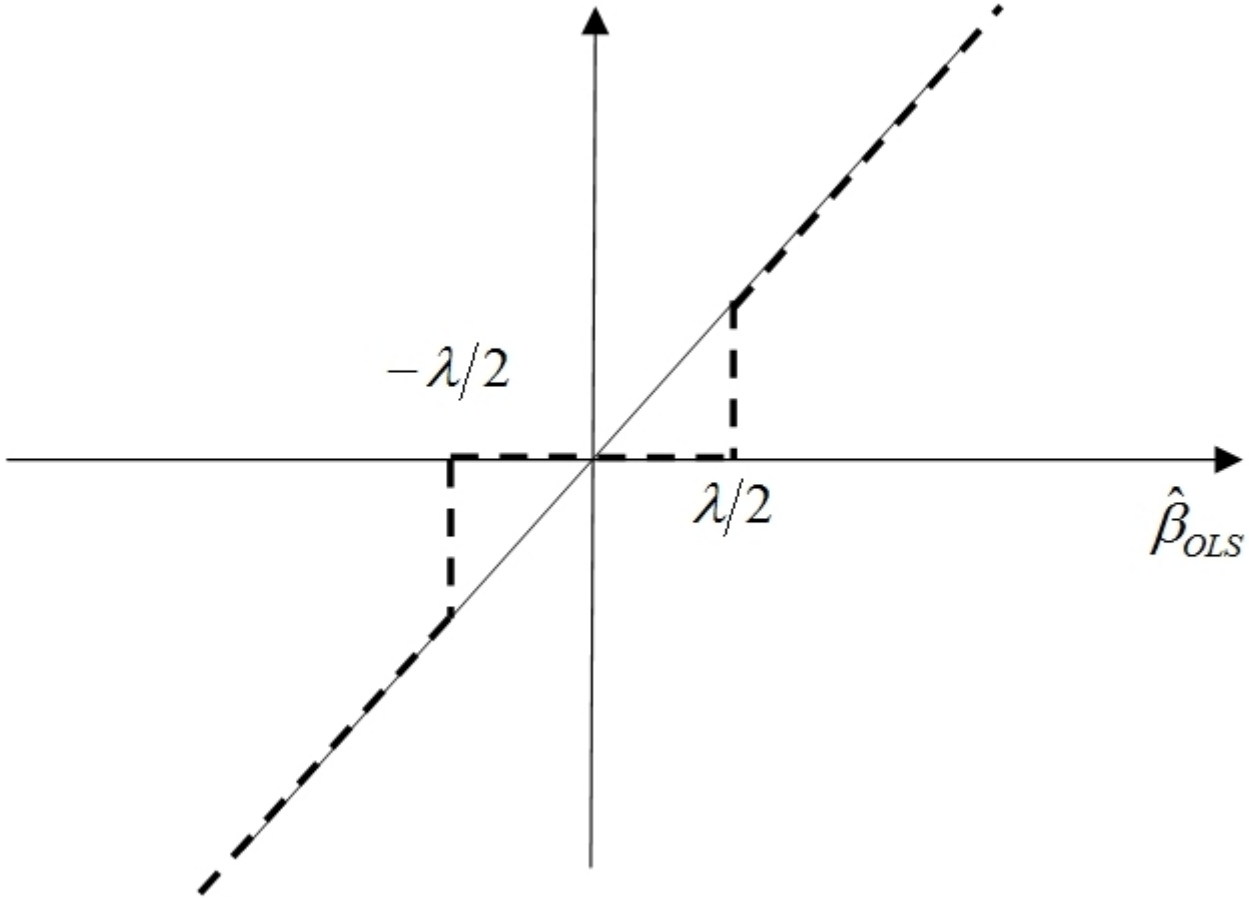

图 9.8 硬门限算子

9.7 调节变量的选择

无论岭回归，还是 Lasso 估计量，其最优解都是调节参数λ的函数，可写为$\hat{\boldsymbol{\beta}}_{ridge}(\lambda)$或$\hat{\boldsymbol{\beta}}_{lasso}(\lambda)$。

变动调节参数λ(调节惩罚力度)，即可得到整条**解的路径**(solution path)或**系数路径**(coefficient path)。

也可将惩罚回归的最优解写为L_2或L_1范数t的函数，即$\hat{\boldsymbol{\beta}}_{ridge}(t)$或$\hat{\boldsymbol{\beta}}_{Lasso}(t)$，参见约束极值问题(9.19)与(9.23)。

我们只想要一个最优解。应如何确定调节参数λ呢？能否把对损失函数对λ求偏导来求得最优的λ？答案是否定的。以岭回归为例，其损失函数为

$$\min_{\boldsymbol{\beta}} \ L(\boldsymbol{\beta})=(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\boldsymbol{\beta}'\boldsymbol{\beta} \qquad (9.30)$$

将损失函数对λ求偏导可得：

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \lambda}=\boldsymbol{\beta}'\boldsymbol{\beta}\geq 0 \qquad (9.31)$$

对于此最小化问题，λ的“最优解”为 0，因为只要λ为正数，且$\boldsymbol{\beta}'\boldsymbol{\beta}>0$，则损失函数$L(\boldsymbol{\beta})$就会上升。但如果$\lambda=0$，则退化为 OLS，并不适用于高维回归，且容易导致过拟合。

选择最优λ的常见方法为K**折交叉验证**(Cross-Validation，简记 CV)。

比如，将全样本随机分为大致相等的 10 个子样本，即 10 折($K=10$)。然后，以其中的 9 折作为训练集，进行惩罚回归(岭回归或 Lasso)，并以所得模型预测作为验证集的其余 1 折，得到该折的均方误差。

如此重复，可得到每一折的均方误差：

$$\text{MSE}_k(\lambda) \equiv \frac{1}{n_k} \sum_{i \in fold_k} \left(y_i - \hat{y}_i(\lambda) \right)^2, \quad k = 1, \cdots, K \qquad (9.32)$$

其中，n_k为第k折的样本容量。

将这 10 折的均方误差进行平均，即可得到**交叉验证误差**(Cross-validation Error，简记 CV Error)：

$$\mathrm{CV}(\lambda) \equiv \overline{\mathrm{MSE}}(\lambda) \equiv \frac{1}{K}\sum_{k=1}^{K}\mathrm{MSE}_k(\lambda) \qquad (9.33)$$

交叉验证误差$\mathrm{CV}(\lambda)$是λ的函数。可选择最优λ，使$\mathrm{CV}(\lambda)$最小化：

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}}\ \mathrm{CV}(\lambda) \qquad (9.34)$$

在实践中，可对调节变量λ进行网格化处理。

首先，由于$\lambda \geq 0$，故λ的最小值为 0。

其次，如果λ足够大，则对于$\|\boldsymbol{\beta}\|_2^2$或$\|\boldsymbol{\beta}\|_1$的惩罚非常严厉，可使$\hat{\boldsymbol{\beta}}_{ridge}$或$\hat{\boldsymbol{\beta}}_{lasso}$变为 0。记刚好能使$\hat{\boldsymbol{\beta}}_{ridge}$或$\hat{\boldsymbol{\beta}}_{lasso}$变为 0 的调节参数取值为$\lambda_{\max}$。

故只要将区间$[0, \lambda_{\max}]$进行网格等分(比如 100 等分)，然后在每个等分点上计算$\mathrm{CV}(\lambda)$，即可求得最优的$\hat{\lambda}$。

$\mathrm{CV}(\lambda)$一般为凸函数，而λ太小或太大均不利于最小化$\mathrm{CV}(\lambda)$，故最优的$\hat{\lambda}$位于中间区域，参见图 9.9。

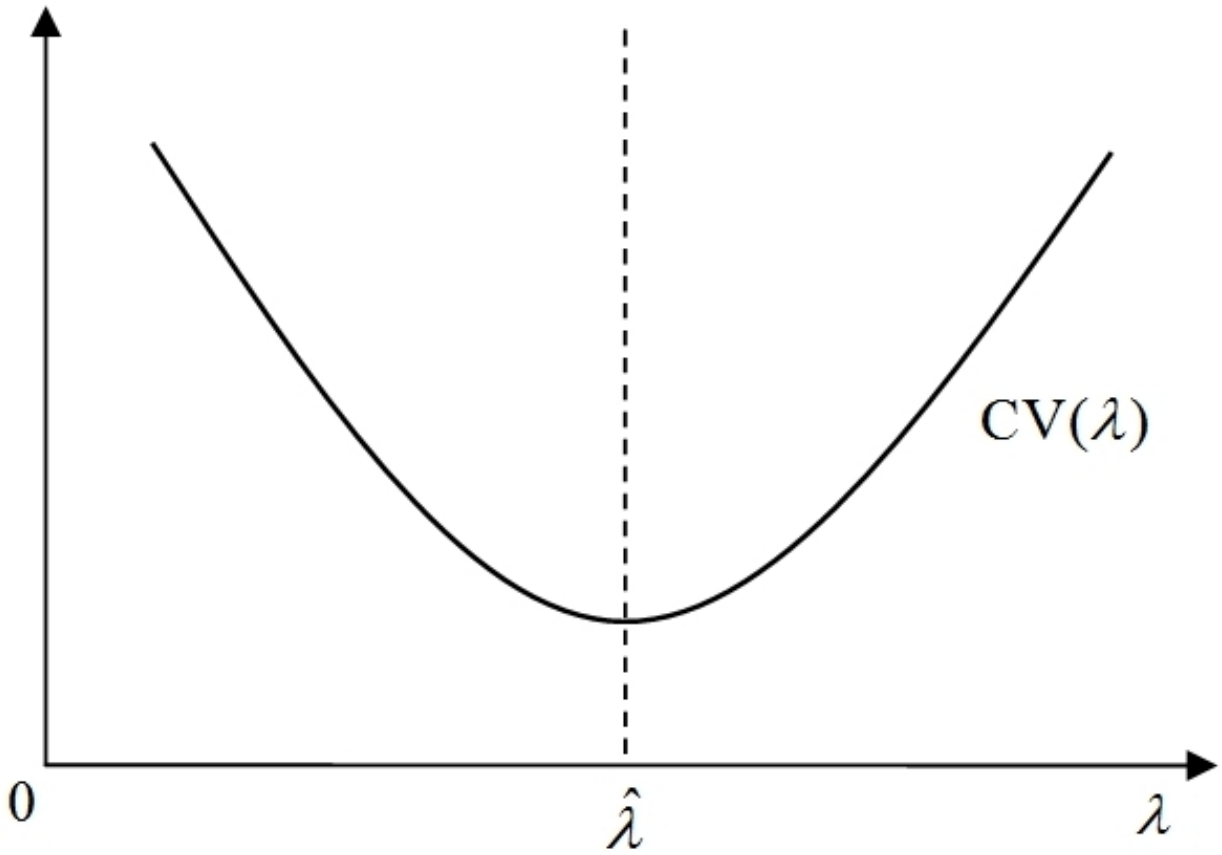

图 9.9 交叉验证误差的示意图

进一步，由于在λ的每个网格点，均计算了$\mathrm{MSE}_1(\lambda),\cdots,\mathrm{MSE}_{10}(\lambda)$，共有 10 个数，故可计算相应的样本标准差，记为$sd_{\mathrm{MSE}}(\lambda)$：

$$sd_{\mathrm{MSE}}(\lambda)\equiv\sqrt{\frac{1}{9}\sum_{k=1}^{10}\left[\mathrm{MSE}_k(\lambda)-\overline{\mathrm{MSE}}(\lambda)\right]^2} \qquad (9.35)$$

有时也将正负标准差$\pm sd_{\mathrm{MSE}}(\lambda)$画在图 9.9 中，以评估$\mathrm{CV}(\lambda)$的不确定性(uncertainty)。

9.8 弹性网估计量

Lasso 虽然具有筛选变量的功能，但此功能并不完美。比如，当几个变量高度相关时，Lasso 可能随意选择其中一个。

为此，Zou and Hastie (2005)将 Lasso 与岭回归相结合，提出**弹性网**(Elastic Net)估计量。在弹性网估计量的损失函数中，同时包含L_1与L_2惩罚项：

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\|\boldsymbol{\beta}\|_2^2 \qquad (9.36)$$

其中，$\lambda_1 \geq 0$与$\lambda_2 \geq 0$都是调节参数。

由于λ_1与λ_2的取值范围均为无穷，不便于使用交叉验证选择其最优值。为此，定义$\lambda \equiv \lambda_1 + \lambda_2$，$\alpha \equiv \lambda_1 / \lambda$，则可将损失函数(9.36)等价写为

$$\min_{\boldsymbol{\beta}} \ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\left[\alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2\right] \qquad (9.37)$$

其中，$\lambda \geq 0$与$0 \leq \alpha \leq 1$为调节参数。

由于调节参数α的取值局限于区间$[0,1]$，故便于通过交叉验证选择其最优值。

如果$\alpha=0$，则弹性网退化为岭回归；而如果$\alpha=1$，则弹性网退化为 Lasso。故岭回归与 Lasso 都是弹性网的特例。

如果$0<\alpha<1$，则弹性网为岭回归与 Lasso 之间的折衷。容易看出，上式可等价地写为以下约束极值问题：

$$\begin{aligned}&\min_{\boldsymbol{\beta}}\ (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\\&s.t.\ \alpha\|\boldsymbol{\beta}\|_1+(1-\alpha)\|\boldsymbol{\beta}\|_2^2\le t\end{aligned}\tag{9.38}$$

其中，$t\ge 0$为调节参数。

约束集“$\alpha\left\|\boldsymbol{\beta}\right\|_1+(1-\alpha)\left\|\boldsymbol{\beta}\right\|_2^2\le t$”不同于岭回归或 Lasso 的约束集，但兼具二者的特点。

仍以二元回归为例，$\boldsymbol{\beta}=(\beta_1\ \beta_2)'$，则弹性网估计量的约束集为$\alpha(\left|\beta_1\right|+\left|\beta_2\right|)+(1-\alpha)(\beta_1^2+\beta_2^2)\le t$。

图 9.10 展示了弹性网($\alpha=0.5$)，Lasso 及岭回归的约束集。

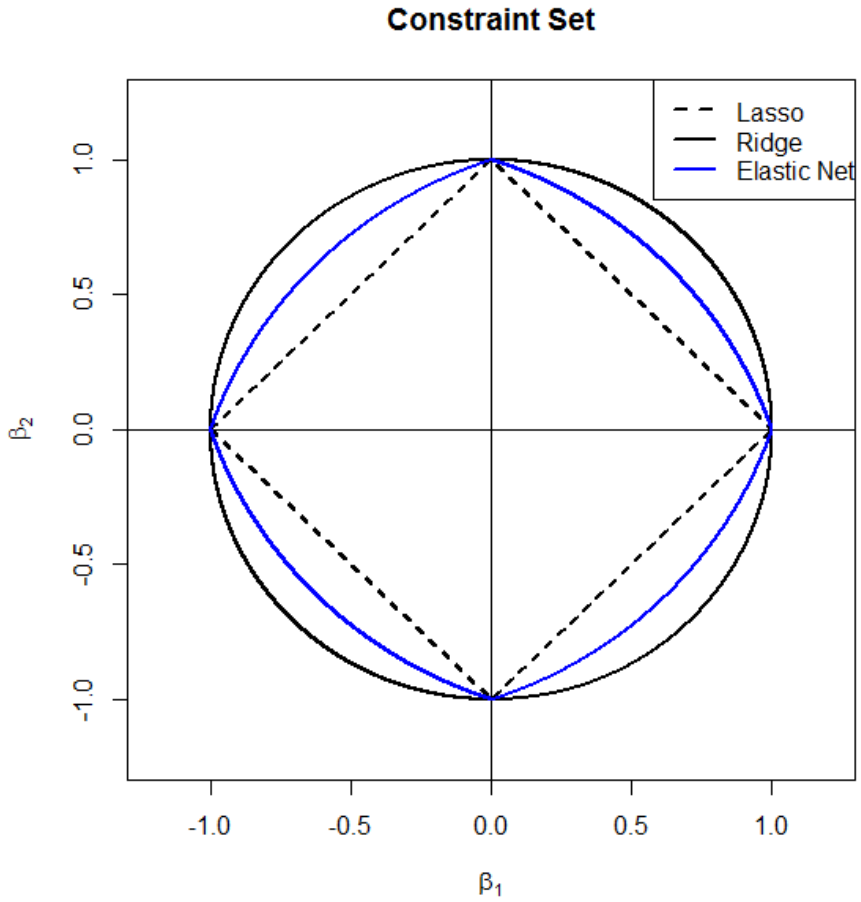

图 9.10 弹性网、Lasso 与岭回归的约束集

从图 9.10 可知，弹性网的约束集介于 Lasso 与岭回归约束集之间。

与 Lasso 类似，弹性网的约束集也在坐标轴上有四个尖角，故弹性网也具有筛选变量的功能。

与岭回归的圆形约束集类似，弹性网的约束集在四个象限也呈弧形，故弹性网具有类似于岭回归的收缩参数之功能。

当若干特征变量之间高度相关时(比如，几个高度相关的基因都对某种疾病有影响)，弹性网倾向于将这些高度相关的变量都选上。

由于岭回归与 Lasso 均为弹性网的特例，而弹性网可通过交叉验证选择最优的调节参数α，故弹性网的预测能力肯定不差于前二者。

自从 Tibshirani(1996)提出 Lasso 之后，惩罚回归成为统计学研究的活跃前沿，相继出现 Lasso 的各种变种，在惩罚项上推陈出新，以适应不同的数据特点。

这些变种包括“自适应套索”(adaptive Lasso)、“分组套索”(grouped lasso)、“融合套索”(fused lasso)，“平滑裁剪绝对离差”(Smoothly Clipped Absolute Deviation, 简记 SCAD)惩罚项、“极小极大凹惩罚项”(Minimax Concave Penalty, 简记 MCP)等，参见 Hastie et al. (2009)。

特别地，也可将 Lasso 应用于非线性模型。比如，**逻辑 Lasso**(Logistic Lasso)将 Lasso 惩罚项引入逻辑回归的损失函数：

$$\min_{\boldsymbol{\beta}} \; -\sum_{i=1}^{n} y_i \ln\left[\Lambda(\boldsymbol{x}_i'\boldsymbol{\beta})\right] - \sum_{i=1}^{n} (1-y_i) \ln\left[1-\Lambda(\boldsymbol{x}_i'\boldsymbol{\beta})\right] + \lambda \left\|\boldsymbol{\beta}\right\|_1 \tag{9.39}$$

其中，逻辑回归的对数似然函数之负数即为逻辑回归的损失函数。

更一般地，可使用弹性网的惩罚项，进行**惩罚逻辑回归**(Penalized Logit)：

$$\min_{\boldsymbol{\beta}} \; -\sum_{i=1}^{n} y_i \ln\left[\Lambda(\boldsymbol{x}_i'\boldsymbol{\beta})\right] - \sum_{i=1}^{n}(1-y_i)\ln\left[1-\Lambda(\boldsymbol{x}_i'\boldsymbol{\beta})\right] + \lambda\left[\alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2\right] \tag{9.40}$$

由于各种算法发展很快，到目前为止，相应的统计推断(statistical inference)依然呈滞后状态(Efron and Hastie, 2016)。

例如，对于惩罚回归，迄今还没有公认的标准误(standard errors)。

9.9 惩罚回归的 Python 案例

Tibshirani(1996)使用前列腺癌数据 `prostate` 演示 Lasso 模型。该数据集包含 97 个前列腺癌患者的数据。

响应变量为 `lpsa`(log of prostate specific antigen，前列腺特异抗原对数)。

特征变量包括以下临床指标(clinical measures)：`lcavol` (log of cancer volume，肿瘤体积对数)，`lweight` (log of prostate weight，前列腺重量对数)，`age` (年龄)，`lbph` (log of benign prostatic hyperplasia amount)，`svi` (seminal vesicle invasion)，`lcp` (log of capsular penetration)，`gleason` (Gleason score)以及 `pgg45` (percentage Gleason scores 4 or 5)。

* 详见教材，以及配套 Python 程序（现场演示）。

附录

A9.1 估计量均方误差的分解

命题 9.1（均方误差的分解） 假设$\hat{\beta}$是一维参数β的估计量，其均方误差可分解为方差与偏差平方之和：

$$\mathrm{MSE}(\hat{\beta}) = \mathrm{Var}(\hat{\beta}) + \left[\mathrm{Bias}(\hat{\beta})\right]^2 \qquad (9.41)$$

证明

$$\begin{aligned}\mathrm{MSE}(\hat{\beta}) &\equiv \mathrm{E}(\hat{\beta}-\beta)^2 = \mathrm{E}\left[\hat{\beta}-\mathrm{E}(\hat{\beta})+\mathrm{E}(\hat{\beta})-\beta\right]^2 \quad (\text{加、减}\mathrm{E}(\hat{\beta})) \\ &= \mathrm{E}\left[\hat{\beta}-\mathrm{E}(\hat{\beta})\right]^2+\mathrm{E}\left[\mathrm{E}(\hat{\beta})-\beta\right]^2+2\mathrm{E}\left\{\left[\hat{\beta}-\mathrm{E}(\hat{\beta})\right]\left[\mathrm{E}(\hat{\beta})-\beta\right]\right\} \\ &= \mathrm{Var}(\hat{\beta})+\left[\mathrm{Bias}(\hat{\beta})\right]^2+2\underbrace{\mathrm{E}\left\{\left[\hat{\beta}-\mathrm{E}(\hat{\beta})\right]\left[\mathrm{E}(\hat{\beta})-\beta\right]\right\}}_{=0}\end{aligned}$$

其中，上式的交叉项为 0：

$$\mathrm{E}\left\{\left[\hat{\beta}-\mathrm{E}(\hat{\beta})\right]\left[\mathrm{E}(\hat{\beta})-\beta\right]\right\}=\left[\mathrm{E}(\hat{\beta})-\beta\right]\cdot\mathrm{E}\left[\hat{\beta}-\mathrm{E}(\hat{\beta})\right]=\left[\mathrm{E}(\hat{\beta})-\beta\right]\cdot 0=0$$

同理可证，在多维情况下，也有类似的结论：

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}) \equiv \mathrm{E}\left[(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})'\right] = \mathrm{Var}(\hat{\boldsymbol{\beta}}) + \left[\mathrm{Bias}(\hat{\boldsymbol{\beta}})\right]\left[\mathrm{Bias}(\hat{\boldsymbol{\beta}})\right]' \tag{9.42}$$

A9.2 次梯度向量与次微分

对于光滑函数进行最优化，求导数是一种方便方法。然而，有时目标函数并不光滑，比如 Lasso 的损失函数即包含不光滑的绝对值函数。

对于连续的凸函数，可将导数的概念推广为“次导数”(subderivative)，并将梯度向量的概念推广为“次梯度”(subgradient)，参见图 9.16。

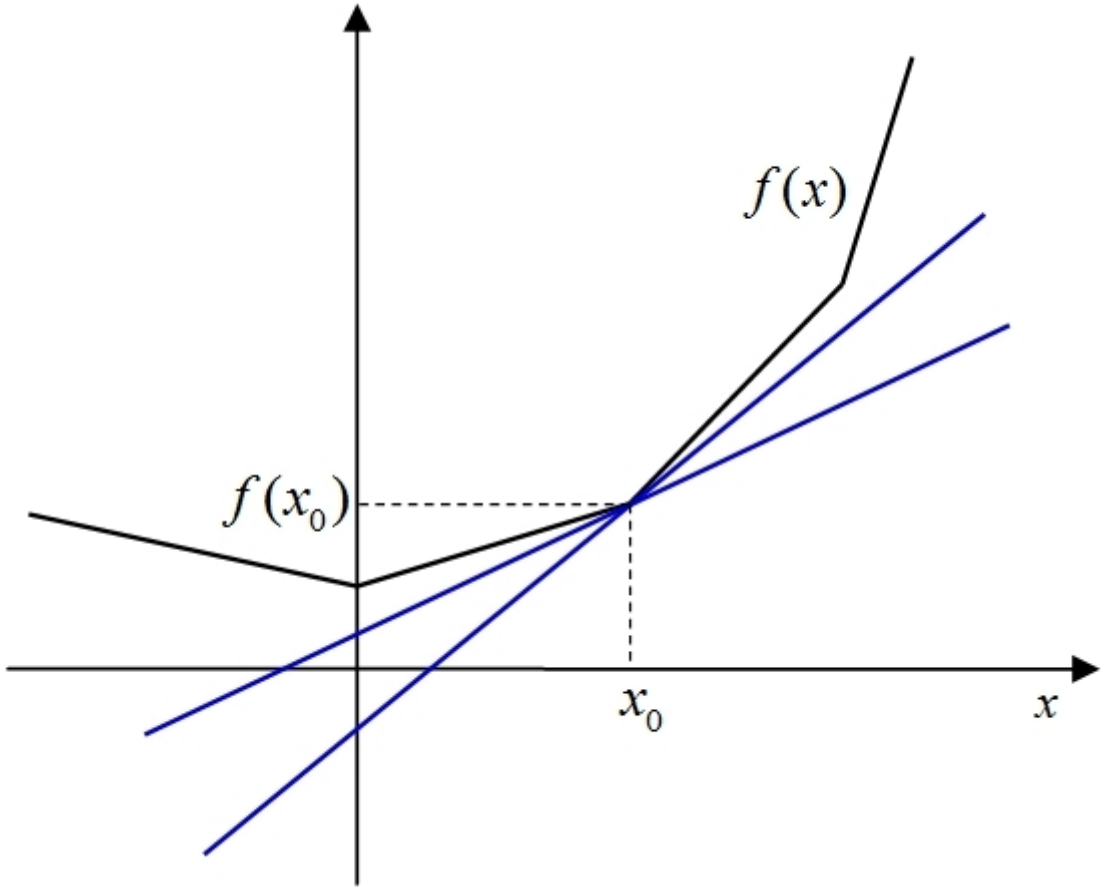

图 9.16 次导数的示意图

在图 9.16 中，$f(x)$为连续的凸函数(convex function)，因为在该函数之上的点之集合为凸集(convex set)。

在x_0处，$f(x)$并不光滑，而存在折线的拐点，故不存在导数。

在$(x_0, f(x_0))$处，可找到位于函数$f(x)$之下，但与$f(x)$仅在x_0处相交(切)的直线；比如图中的两条蓝线。

在某种意义上，这两条蓝线也起着切线的作用：如果$f(x)$在x_0可导，则其切线也位于函数$f(x)$之下，且与$f(x)$也仅在x_0处相交。

在微积分中，将切线的斜率定义为导数。

类似地，可将图中两条蓝线的斜率定义为**次导数**(subderivative)。

次导数未必唯一；比如，图中的两条蓝线各有相应的次导数。

将所有次导数之集合，称为**次微分**(subdifferential)。

类似地，对于多元的连续凸函数，可将次导数的概念推广为“次梯度”(subgradient)。

定义 9.1 假定 $f(\mathbf{x}):\mathcal{X}\subseteq\mathbb{R}^p\to\mathbb{R}$ 为 p 维空间的连续凸函数，其中 $\mathcal{X}$ 为其定义域。称 p 维向量 $\mathbf{g}\in\mathbb{R}^p$ 为函数 $f(\mathbf{x})$ 在 $\mathbf{x}_0$ 处的**次梯度**(subgradient)，如果对于任意 $\mathbf{x}\in\mathcal{X}$，都有 $f(\mathbf{x})\geq f(\mathbf{x}_0)+\mathbf{g}'(\mathbf{x}-\mathbf{x}_0)$。

本质上，此定义要求以次梯度向量 $\mathbf{g}$ 所定义的超平面“$f(\mathbf{x}_0)+\mathbf{g}'(\mathbf{x}-\mathbf{x}_0)$”始终在函数 $f(\mathbf{x})$ 之下。

次梯度未必唯一，故定义次梯度的集合为“次微分”。

定义 9.2 假定 $f(\mathbf{x}):\mathcal{X}\subseteq\mathbb{R}^p\to\mathbb{R}$ 为 p 维空间的连续凸函数，其中 $\mathcal{X}$ 为其定义域。在 $\mathbf{x}_0$ 处，函数 $f(\mathbf{x})$ 的**次微分**(subdifferential)为所有次梯度之集合，即

$$\partial f(\mathbf{x}_0)\equiv\left\{\mathbf{g}:\ f(\mathbf{x})\geq f(\mathbf{x}_0)+\mathbf{g}'(\mathbf{x}-\mathbf{x}_0),\forall\ \mathbf{x}\in\mathcal{X}\right\} \tag{9.43}$$

例 以绝对值函数 $y=|x|$ 为例(参见图 9.4)，其次微分可写为

$$\partial|x|=\begin{cases}1 & if\ x>0\\ -1 & if\ x<0\\ [-1,1] & if\ x=0\end{cases} \tag{9.44}$$

当$x>0$时，绝对值函数$|x|=x$，故其次微分就是唯一的导数 1。

当$x<0$时，绝对值函数$|x|=-x$，故其次微分就是唯一的导数-1。

只有当$x=0$时，绝对值函数$y=|x|$不可导，才真正需要次微分的概念。此时，最大的次导数为 1，而最小的次导数为-1，而二者之间的任意数都是次导数，故在$x=0$处的次微分为闭区间$[-1,1]$。

A9.3 连续凸函数的最小化定理

对于光滑函数，其最优化要求梯度向量为$\mathbf{0}$。

类似地，对于连续凸函数的最小化问题，则要求零向量$\mathbf{0}$为其次梯度。

命题 9.2（连续凸函数的最小化定理） 对于连续凸函数 $f(\mathbf{x})$，其最小化的充分必要条件要求零向量$\mathbf{0}$为其次梯度，即

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\ f(\mathbf{x}) \quad \Leftrightarrow \quad \mathbf{0} \in \partial f(\mathbf{x}^*) \qquad (9.45)$$

证明 首先，证明必要性。

假设$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}}\ f(\mathbf{x})$，即函数$f(\mathbf{x})$在$\mathbf{x}^*$处达到最小值。则根据最小值的定义可知：

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) = f(\mathbf{x}^*) + \underbrace{\mathbf{0}'(\mathbf{x}-\mathbf{x}^*)}_{=0}, \quad \forall \mathbf{x} \in \mathcal{X} \tag{9.46}$$

根据次梯度的定义，上式表明$\mathbf{0} \in \partial f(\mathbf{x}^*)$。

其次，证明充分性。

假设$\mathbf{0} \in \partial f(\mathbf{x}^*)$，则根据次微分的定义：

$$f(\mathbf{x}) \geq f(\mathbf{x}^*) + \underbrace{\mathbf{0}'(\mathbf{x}-\mathbf{x}^*)}_{=0}, \quad \forall \mathbf{x} \in \mathcal{X} \qquad (9.47)$$

由此可知，对于任意$\mathbf{x} \in \mathcal{X}$，都有$f(\mathbf{x}) \geq f(\mathbf{x}^*)$，故$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} f(\mathbf{x})$。 ■

A9.4 标准正交设计下 Lasso 问题的解析解

假定设计矩阵$\mathbf{X}$的每一列均为单位向量，且相互正交，即所谓“标准正交设计”(orthonormal design)。此时，$\mathbf{X}$为“正交矩阵”(orthogonal matrix)，故$\mathbf{X}'\mathbf{X}=\mathbf{I}$。因此，OLS 估计量可写为$\hat{\boldsymbol{\beta}}_{OLS}=\underbrace{(\mathbf{X}'\mathbf{X})^{-1}}_{=\mathbf{I}}\mathbf{X}'\mathbf{y}=\mathbf{X}'\mathbf{y}$。

Lasso 问题的损失函数为

$$\min_{\boldsymbol{\beta}} \; L(\boldsymbol{\beta})=(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\left\|\boldsymbol{\beta}\right\|_1 \qquad (9.48)$$

其中，1-范数的惩罚项$\lambda\left\|\boldsymbol{\beta}\right\|_1$不可微分，但可求其次微分。

根据命题 9.2(连续凸函数的最小化定理)，Lasso 最小化问题的一阶条件为

$$\begin{aligned}\mathbf{0} \in & -2\underbrace{\mathbf{X}'\mathbf{y}}_{=\hat{\boldsymbol{\beta}}_{OLS}} + 2\underbrace{\mathbf{X}'\mathbf{X}}_{=\mathbf{I}}\hat{\boldsymbol{\beta}}_{lasso} + \lambda\partial\left\|\hat{\boldsymbol{\beta}}_{lasso}\right\|_1 \\ & = -2\hat{\boldsymbol{\beta}}_{OLS} + 2\hat{\boldsymbol{\beta}}_{lasso} + \lambda\partial\left\|\hat{\boldsymbol{\beta}}_{lasso}\right\|_1\end{aligned} \tag{9.49}$$

其中，$\partial\left\|\hat{\boldsymbol{\beta}}_{lasso}\right\|_1$为 1-范数$\left\|\hat{\boldsymbol{\beta}}_{lasso}\right\|_1 = \left|\hat{\beta}_{1,lasso}\right| + \cdots + \left|\hat{\beta}_{p,lasso}\right|$的次微分。

在上式两边同除以 2，并将每个分量展开来写：

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \in -\begin{pmatrix} \hat{\beta}_{1,OLS} \\ \vdots \\ \hat{\beta}_{p,OLS} \end{pmatrix} + \begin{pmatrix} \hat{\beta}_{1,lasso} \\ \vdots \\ \hat{\beta}_{p,lasso} \end{pmatrix} + \frac{\lambda}{2} \begin{pmatrix} \partial\left|\hat{\beta}_{1,lasso}\right| \\ \vdots \\ \partial\left|\hat{\beta}_{p,lasso}\right| \end{pmatrix} \qquad (9.50)$$

单独考察上式的第k个分量可得：

$$0 \in -\hat{\beta}_{k,OLS} + \hat{\beta}_{k,lasso} + \frac{\lambda}{2}\partial\left|\hat{\beta}_{k,lasso}\right| \qquad (9.51)$$

其中，$\partial\left|\hat{\beta}_{k,lasso}\right|$为绝对值函数的次微分，故根据附录 A9.2 的例子可知：

$$\partial\left|\hat{\beta}_{k,lasso}\right| = \begin{cases} 1 & if\ \hat{\beta}_{k,lasso} > 0 \\ -1 & if\ \hat{\beta}_{k,lasso} < 0 \\ [-1,1] & if\ \hat{\beta}_{k,lasso} = 0 \end{cases} \qquad (9.52)$$

首先，考虑$\hat{\beta}_{k,lasso} > 0$的情形。此时，$\partial\left|\hat{\beta}_{k,lasso}\right| = 1$，故表达式(9.51)变为等式：

$$0 = -\hat{\beta}_{k,OLS} + \hat{\beta}_{k,lasso} + \frac{\lambda}{2} \qquad (9.53)$$

由此可得：

$$\hat{\beta}_{k,lasso} = \hat{\beta}_{k,OLS} - \frac{\lambda}{2} > 0 \qquad (9.54)$$

显然，这要求$\hat{\beta}_{k,OLS} > \frac{\lambda}{2}$。

其次，考虑$\hat{\beta}_{k,lasso} < 0$的情形。此时，$\partial\left|\hat{\beta}_{k,lasso}\right| = -1$，故表达式(9.51)变为等式：

$$0 = -\hat{\beta}_{k,OLS} + \hat{\beta}_{k,lasso} - \frac{\lambda}{2} \qquad (9.55)$$

由此可得：

$$\hat{\beta}_{k,lasso} = \hat{\beta}_{k,OLS} + \frac{\lambda}{2} < 0 \qquad (9.56)$$

显然，这要求$\hat{\beta}_{k,OLS} < -\frac{\lambda}{2}$。

最后，考虑$\hat{\beta}_{k,lasso} = 0$的情形。此时，$\partial\left|\hat{\beta}_{k,lasso}\right| = [-1,1]$，故表达式(9.51)变为：

$$0 \in -\hat{\beta}_{k,OLS} + \underbrace{\hat{\beta}_{k,lasso}}_{=0} + \frac{\lambda}{2}[-1,1] \qquad (9.57)$$

由此可得：

$$0 \in -\hat{\beta}_{k,OLS} + \left[-\lambda/2, \lambda/2\right] \qquad (9.58)$$

在上式两边同加$\hat{\beta}_{k,OLS}$可得，$\hat{\beta}_{k,OLS} \in \left[-\lambda/2, \lambda/2\right]$，即$\left|\hat{\beta}_{k,OLS}\right| \leq \lambda/2$。综上，$\hat{\beta}_{k,lasso}$的解析解可写为$\hat{\beta}_{k,OLS}$的分段函数：

$$\hat{\beta}_{k,lasso} = \begin{cases} \hat{\beta}_{k,OLS} - \lambda/2 & if\ \hat{\beta}_{k,OLS} > \lambda/2 \\ 0 & if\ \left|\hat{\beta}_{k,OLS}\right| \leq \lambda/2 \\ \hat{\beta}_{k,OLS} + \lambda/2 & if\ \hat{\beta}_{k,OLS} < -\lambda/2 \end{cases} \qquad (9.59)$$

上式适用于任意$k = 1, \cdots, p$。