坐标下降算法

王璐

对很多带有 L_1 惩罚项的优化问题，坐标下降法 (Coordinate Descent) 是一种简单有效地求解算法，此时目标函数的梯度不是处处存在，因此梯度下降法不适用。

1 Coordinate Descent (CD)

对于如下的一个多元优化问题

$$\min\ f(x_1, x_2, \ldots, x_d)$$

CD 的做法是

1. 选取一个初始值 $\boldsymbol{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_d^{(0)})$.
2. 在第 t 步迭代，依次更新每个变量 x_i, $i = 1, 2, \ldots, d$. 且在更新 x_i 时，将其它变量固定在当前值。即求解如下 d 个优化问题：

$$\begin{aligned}
x_1^{(t)} &\in \underset{x_1}{\operatorname{argmin}}\ f(x_1, x_2^{(t-1)}, \ldots, x_d^{(t-1)}) \\
x_2^{(t)} &\in \underset{x_2}{\operatorname{argmin}}\ f(x_1^{(t)}, x_2, x_3^{(t-1)} \ldots, x_d^{(t-1)}) \\
&\vdots \\
x_d^{(t)} &\in \underset{x_d}{\operatorname{argmin}}\ f(x_1^{(t)}, x_2^{(t)}, \ldots, x_{d-1}^{(t)}, x_d)
\end{aligned}$$

3. 重复上述过程直到收敛, e.g. $\left\| f(\boldsymbol{x}^{(t+1)}) - f(\boldsymbol{x}^{(t)}) \right\| < \varepsilon$.

CD 算法体现了求解优化问题的一个普遍思路：将一个复杂的优化问题转化为求解一系列简单的优化问题，每个子优化问题都是低维甚至一维的。这比直接求解高维优化问题容易很多，特别当目标函数的梯度 ∇f 不存在时。

Remarks

1. 由于上述算法一直在轮流更新每个变量，也被称为 alternating method.

2. 上述每步迭代 t 可以看作轮流地沿每个坐标轴方向 $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ 搜索使 f 减小最多的点

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \sum_{i=1}^{d} \tau_{it} \boldsymbol{e}_i$$
$$\tau_{it} = \underset{\tau}{\operatorname{argmin}}\ f(x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_i^{(t)} + \tau, x_{i+1}^{(t)}, \ldots, x_d^{(t)}),\ i = 1, \ldots, d.$$

因此被称为坐标下降法。

3. 更新坐标的顺序可以任意选取 $\{1, 2, \ldots, d\}$ 的一组排列。

4. 如果将上述算法中的单个变量用一组变量来替换，即 x_i 可以是向量，则称这样轮流更新每一区块 (block) 变量的方法为 **block coordinate descent**.

1.1 收敛性分析

Tseng (2001) 证明了如下定理。

Theorem 1. *如果 f 是连续函数，集合 $\{\boldsymbol{x} : f(\boldsymbol{x}) \le f(\boldsymbol{x}^{(0)})\}$ 对任意 $\boldsymbol{x}^{(0)}$ 都是紧集，且 f 关于每个分量 x_i 只有唯一的最小值点，则由 CD 产生的序列 $\{\boldsymbol{x}^{(t)}\}$ 的每个聚点 $\boldsymbol{x}^\star$ 都是 f 的* ***coordinatewise minimum point****, i.e.*

$$f(\boldsymbol{x}^\star + \tau \boldsymbol{e}_i) \ge f(\boldsymbol{x}^\star),\ \forall \tau \in \mathbb{R}, i = 1, \ldots, d. \tag{1}$$

Proof. 由 CD 算法产生的点列 $\{\boldsymbol{x}^{(t)}\}$ 在一个有界闭集（紧集）上，根据实分析中的 Bolzano-Weierstrass 定理，有界点列必有收敛子列。假设 $\boldsymbol{x}^\star$ 是一个聚点，即存在 $\boldsymbol{x}^{(t_j)} \to \boldsymbol{x}^\star,\ j \to \infty$.

下面证明

$$f(\boldsymbol{x}^\star) = \min_{\tau}\ f(\boldsymbol{x}^\star + \tau \boldsymbol{e}_i),\ i = 1, \ldots, d. \tag{2}$$

假设 $\boldsymbol{x}^\star$ 不满足(2), 则从 $\boldsymbol{x}^\star$ 开始，再做一步 CD 迭代 (轮流沿 $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ 搜索一遍) 可以找到一个点 $\boldsymbol{y}^\star$ 且 $f(\boldsymbol{y}^\star) < f(\boldsymbol{x}^\star)$.

由于 $\boldsymbol{x}^{(t_j)} \to \boldsymbol{x}^\star,\ j \to \infty$，则存在一个很大的 L 使得 $\boldsymbol{x}^{(t_L)} \approx \boldsymbol{x}^\star$. 那么从 $\boldsymbol{x}^{(t_L)}$ 开始再做一步 CD 迭代得到 $\boldsymbol{x}^{(t_L+1)} \approx \boldsymbol{y}^\star$. 但是由于 CD 产生的序列 $\{f(\boldsymbol{x}^{(t)})\}$ 是单调递减的，则有

$$f(\boldsymbol{y}^\star) \approx f(\boldsymbol{x}^{(t_L+1)}) \ge f(\boldsymbol{x}^{(t_{L+1})}) \ge f(\boldsymbol{x}^\star) > f(\boldsymbol{y}^\star)$$

矛盾，因此(2)成立。 □

Remarks

1. 如果在定理1中加入函数 f 连续可导 ($f \in C^1$)，那么 CD 算法产生的聚点 $\boldsymbol{x}^\star$ 是局部极小值点吗?

   Ans: 不一定。如果 f 连续可导，则由(2)可得 $\dfrac{\partial f(\boldsymbol{x}^\star)}{\partial x_i} = 0, i = 1, \ldots, d.$ 因此 $\boldsymbol{x}^\star$ 是一个驻点, 不一定是局部极小值点。在这种情况下, 可能存在某个方向向量 $\boldsymbol{d}$ 和常数 τ 使得 $f(\boldsymbol{x}^\star + \tau \boldsymbol{d}) < f(\boldsymbol{x}^\star)$, 如图1所示。

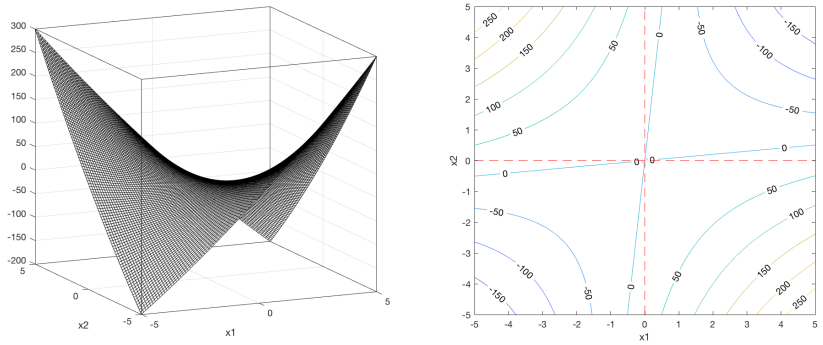

Figure 1: 点 (0,0) 使函数在每个坐标方向取到最小值，但不是局部极小值点。

注意为满足定理1紧集的要求，总可以在远处将图1下降的曲面向上弯曲且保持曲面光滑。

2. 如果在定理1中加入函数 f 是凸函数，那么 $\boldsymbol{x}^\star$ 是全局最小值点吗?

   Ans: 不一定，如图2所示，注意此时 f 不是处处可导。

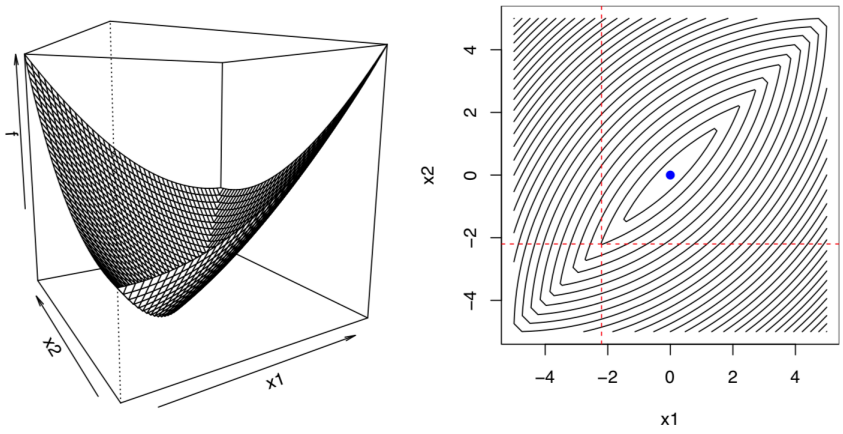

Figure 2: 红虚线交叉点使函数在每个坐标方向取到最小值，但不是全局最小值点。Picture source: Pradeep Ravikumar

3. 如果在定理1中加入函数 f 是凸函数且 $f \in C^2$, 那么 $\boldsymbol{x}^\star$ 是全局最小值点吗?

   Ans: 是。由第 1 点可知此时 $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$. 由于 f 是凸函数, 二阶导数 Hessian matrix 处处是半正定矩阵，因此 $\forall \boldsymbol{x}$,

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) = \underbrace{\nabla f(\boldsymbol{x}^\star)}_{=\mathbf{0}}^\top (\boldsymbol{x} - \boldsymbol{x}^\star) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \underbrace{\nabla^2 f(\tilde{\boldsymbol{x}})}_{p.s.d} (\boldsymbol{x} - \boldsymbol{x}^\star) \geq 0.$$

   可见 f 是否连续可导对 CD 产生的聚点性质有很大影响。但是当 $f \in C^2$ 且是凸函数时，总可以使用收敛速度更快的算法, 如 Newton-Raphson.

4. 如果 f 可以写成 $f(\boldsymbol{x}) = g(\boldsymbol{x}) + \sum_{i=1}^d h_i(x_i)$, 其中 $g : \mathbb{R}^d \to \mathbb{R}$ 是凸函数且 $g \in C^2$，每个 $h_i : \mathbb{R} \to \mathbb{R}$ 都是凸函数, 那么 CD 算法产生的聚点 $\boldsymbol{x}^\star$ 是全局最小值点吗?

   Ans: 是。如果函数 $h : \mathbb{R}^d \to \mathbb{R}$ 可以写为 $h(\boldsymbol{x}) = \sum_{i=1}^d h_i(x_i)$, 称函数 h 是可分的 (separable).

   Proof. 对 $\forall \boldsymbol{x}$,

$$\begin{aligned}
f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) &= g(\boldsymbol{x}) - g(\boldsymbol{x}^\star) + \sum_{i=1}^d [h_i(x_i) - h_i(x_i^\star)] \\
&= \nabla g(\boldsymbol{x}^\star)^\top (\boldsymbol{x} - \boldsymbol{x}^\star) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^\star)^\top \underbrace{\nabla^2 g(\tilde{\boldsymbol{x}})}_{p.s.d} (\boldsymbol{x} - \boldsymbol{x}^\star) + \sum_{i=1}^d [h_i(x_i) - h_i(x_i^\star)] \\
&\geq \nabla g(\boldsymbol{x}^\star)^\top (\boldsymbol{x} - \boldsymbol{x}^\star) + \sum_{i=1}^d [h_i(x_i) - h_i(x_i^\star)] \\
&= \sum_{i=1}^d \left[\frac{\partial g(\boldsymbol{x}^\star)}{\partial x_i}(x_i - x_i^\star) + h_i(x_i) - h_i(x_i^\star) \right]
\end{aligned}$$

   当 $\boldsymbol{x} \to \boldsymbol{x}^\star$ 时，$x_i \to x_i^\star$, $i = 1, \ldots, d$. 则

$$\begin{aligned}
f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) &\geq \sum_{i=1}^d \left[\frac{\partial g(\boldsymbol{x}^\star)}{\partial x_i}(x_i - x_i^\star) + h_i(x_i) - h_i(x_i^\star) \right] \\
&\approx \sum_{i=1}^d \left(g\left(\boldsymbol{x}^\star + (x_i - x_i^\star)\boldsymbol{e}_i\right) - g(\boldsymbol{x}^\star) + h_i(x_i) - h_i(x_i^\star) \right) \qquad (3) \\
&= \sum_{i=1}^d \underbrace{f\left(\boldsymbol{x}^\star + (x_i - x_i^\star)\boldsymbol{e}_i\right) - f(\boldsymbol{x}^\star)}_{\geq 0} \geq 0 \qquad (4)
\end{aligned}$$

   其中(3)根据偏导数定义

$$\lim_{x_i \to x_i^\star} \frac{g\left(\boldsymbol{x}^\star + (x_i - x_i^\star)\boldsymbol{e}_i\right) - g(\boldsymbol{x}^\star)}{x_i - x_i^\star} = \frac{\partial g(\boldsymbol{x}^\star)}{\partial x_i}.$$

   (4)根据定理1.

   因此 $\boldsymbol{x}^\star$ 是局部极小值点，由于 f 是凸函数, 因此局部极小值点也是全局最小值点。 □

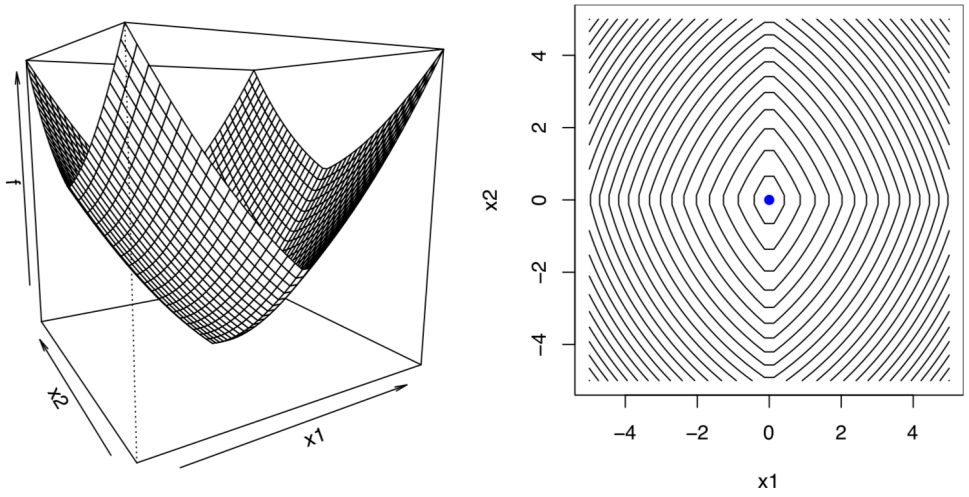

Figure 3: 函数 $f(\boldsymbol{x}) = g(\boldsymbol{x}) + \sum_{i=1}^{d} h_i(x_i)$ 的一个例子。Picture source: Pradeep Ravikumar

图3给出了这种情况下 f 的一个实例。当凸函数 f 不可导的部分是可分的 (separable)，不可导的点只分布在平行于坐标轴的线上。由于 CD 算法相邻两次搜索的方向正交，此时在任一点总有一个坐标方向落在过该点的等高线围成的凸型区域内。

2 CD 应用：LASSO

很多变量选择的问题需要在模型中加入不可导的惩罚项, 比如 LASSO.

对一般的线性回归模型，被解释变量 $Y_i \in \mathbb{R}$ 和解释变量 $X_i \in \mathbb{R}^p$ 之间的关系是 $E(Y_i \mid X_i = \boldsymbol{x}_i) = \alpha + \boldsymbol{x}_i^\top \boldsymbol{\beta}$, $i = 1, \ldots, n$. 当变量个数 $p > n$ 时, 为了保证参数可识别，可以在目标函数中加入对 $\boldsymbol{\beta}$ 范数的惩罚，比如 ridge regression 惩罚的是 $\|\boldsymbol{\beta}\|_2^2$:

$$\min_{\alpha, \boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{5}$$

其中 $\lambda > 0$ 是一个给定的惩罚系数。

构造一个 $n \times p$ 的 design matrix $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$. 如果对 X 的每一列做 demean 处理，使得 $\sum_{i=1}^{n} x_{ij}/n = 0$，则 α 的最优解总是 $\hat{\alpha} = \bar{y}$. 因为(5)关于 α 是一个凸函数，令其一阶偏导数等

于 0 得:

$$
\begin{aligned}
-2\sum_{i=1}^{n}(y_i-\alpha-\sum_{j=1}^{p}\beta_j x_{ij})&=0\\
\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}\sum_{j=1}^{p}\beta_j x_{ij}&=n\alpha\\
\sum_{i=1}^{n}y_i-\sum_{j=1}^{p}\beta_j\underbrace{\sum_{i=1}^{n}x_{ij}}_{=0}&=n\alpha
\end{aligned}
$$

因此 $\hat{\alpha}=\dfrac{1}{n}\sum_{i=1}^{n}y_i=\bar{y}$.

当然总可以对 $\boldsymbol{y}=(y_1,\ldots,y_n)^\top$ 做 demean 处理使得 $\bar{y}=0$. 假设 X 和 $\boldsymbol{y}$ 已经做过 demean, 此时可以只考虑省略截距项 α 的模型:

$$
\min_{\boldsymbol{\beta}}\ \sum_{i=1}^{n}\left(y_i-\boldsymbol{x}_i^\top\boldsymbol{\beta}\right)^2+\lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} \tag{6}
$$

目标函数(6)是 $\boldsymbol{\beta}$ 的二次函数，由一阶导数条件

$$
-2X^\top(\boldsymbol{y}-X\boldsymbol{\beta})+2\lambda\boldsymbol{\beta}=\mathbf{0}
$$

可得最优解

$$
\hat{\boldsymbol{\beta}}=(X^\top X+\lambda I)^{-1}X^\top\boldsymbol{y}. \tag{7}
$$

Remarks

1. 与 OLS 相比, ridge 估计量(7)更稳定：即使 $X^\top X$ 不可逆，$(X^\top X+\lambda I)$ 总是可逆的 $(\lambda>0)$.
2. Ridge regression 很难将某个变量的系数 β_j 彻底估计为 0，而且它倾向于将相关变量的系数估计得很相近。极端情况下，如果在回归中放入 k 个完全相同的解释变量, 则每个变量系数的 ridge 估计值都一样，且是只放一个该变量时系数的 $\dfrac{1}{k}$ (Friedman et al., 2010).

LASSO 惩罚的是 $\|\boldsymbol{\beta}\|_1$:

$$
\min_{\boldsymbol{\beta}}\ \sum_{i=1}^{n}(y_i-\boldsymbol{x}_i^\top\boldsymbol{\beta})^2+\lambda\sum_{j=1}^{p}|\beta_j| \tag{8}
$$

LASSO 估计的 $\hat{\boldsymbol{\beta}}$ 一般更稀疏, 因为 LASSO 可以将某些变量的系数恰好估计为 0.

优化问题(6)和(8)分别等价于

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 \\ &\text{s.t. } \boldsymbol{\beta}^\top \boldsymbol{\beta} \leq t \end{aligned} \tag{9}$$

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2 \\ &\text{s.t.} \sum_{j=1}^{p} |\beta_j| \leq t \end{aligned} \tag{10}$$

由于 $\sum_{i=1}^{n} \left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right)^2$ 是 $\boldsymbol{\beta}$ 的二次函数，它的等高线是一系列椭圆，最小值点对应的是 OLS 估计量。如图4所示，当 $p = 2$ 时，ridge (9)的限制区域是一个圆，LASSO (10)的限制区域是矩形，它们的最优解是椭圆等高线第一次触碰到限制区域的点。图4说明了为什么 LASSO 可以将某些变量的系数恰好估计为 0. 当 $q < 1$ 时，$\|\boldsymbol{\beta}\|_q \leq t$ 围成的区域不是凸集，对应的目标函数也不是凸函数，如图5所示。

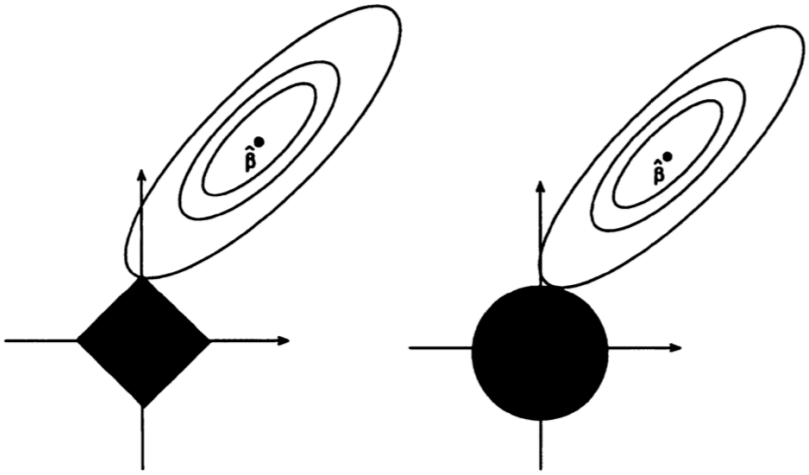

Figure 4: $p = 2$ 时 lasso 的解 (左) 和 ridge regression 的解 (右)。Picture source: Tibshirani (1996)

求解 LASSO 优化问题 (10)的一个简单想法是将绝对值不等式写成一系列线性不等式的形式，

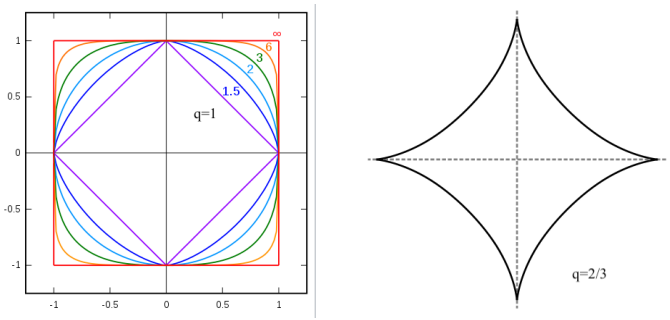

Figure 5: 不同 q 下，$\|\boldsymbol{\beta}\|_q \leq 1$ 在平面上对应的区域，其中 $\|\boldsymbol{\beta}\|_q = \left(\sum_{j=1}^{p} |\beta_j|^q\right)^{1/q}$. Picture source: Wikipedia

考虑每一种系数符号的组合。以 2 个变量的情况为例，(10)中的限制条件可以写为:

$$\begin{cases} \beta_1 + \beta_2 \leq t \\ -\beta_1 + \beta_2 \leq t \\ \beta_1 - \beta_2 \leq t \\ -\beta_1 - \beta_2 \leq t \end{cases}$$

遍历上述每个限制条件寻找最小值点的计算复杂度随 p 指数增加，因为 p 个变量对应 2^p 个线性不等式。此外人们还提出了 interior point method, shooting, iterated ridge regression 等算法，最终胜出的是 CD 算法。

2.1 CD 算法估计 LASSO

注意到(8)中的残差平方和 (residual sum of squares, RSS) 随 n 增加，为了避免对 $\|\boldsymbol{\beta}\|_1$ 的惩罚比重随 n 变化，考虑用 MSE 替代 RSS. 因此将 LASSO 的损失函数写为:

$$f(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{11}$$

对于 β_j, 如果已知其它系数 $\beta_k (k \neq j)$ 的估计量，注意到 $\dfrac{\partial f}{\partial \beta_j}$ 只在 $\beta_j = 0$ 不存在:

$$\frac{\partial f}{\partial \beta_j} = \begin{cases} -\dfrac{1}{n} \sum_{i=1}^{n} x_{ij} (y_i - \tilde{y}_i^{(j)}) + \dfrac{1}{n} \left(\sum_{i=1}^{n} x_{ij}^2\right) \beta_j + \lambda, & \text{if } \beta_j > 0 \\ -\dfrac{1}{n} \sum_{i=1}^{n} x_{ij} (y_i - \tilde{y}_i^{(j)}) + \dfrac{1}{n} \left(\sum_{i=1}^{n} x_{ij}^2\right) \beta_j - \lambda, & \text{if } \beta_j < 0 \end{cases}$$

其中 $\tilde{y}_i^{(j)} = \sum_{k\neq j} x_{ik}\beta_k$ 是去掉 x_{ij} 后对 y_i 的预测值。令

$$A_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2$$
$$C_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}(y_i - \tilde{y}_i^{(j)})$$

注意到 $A_j \geq 0$. 若 $A_j = 0$, 则 $x_{ij} = 0, i = 1, \ldots, n$, 说明第 j 个变量是退化的，应该剔除。注意此时 X 的每一列都做过 demean，若 $A_j = 0$，说明第 j 个变量的观察值都是常数。

经过简单讨论 C_j 的正负及 $|C_j|$ 与 λ 的大小关系，可得 β_j 的最优估计值为：

$$\hat{\beta}_j = \frac{\text{sign}(C_j)\,(|C_j| - \lambda)_+}{A_j} \tag{12}$$

(12)的分子使用了一个二元算子 soft-thresholding operator $S(z, r)$,

$$S(z,r) = \text{sign}(z)\,(|z| - r)_+ = \begin{cases} z - r & \text{if } z > 0 \text{ and } r < |z| \\ z + r & \text{if } z < 0 \text{ and } r < |z| \\ 0 & \text{if } r \geq |z| \end{cases}$$

注意到 $\dfrac{C_j}{A_j}$ 是 partial residual $(y_i - \tilde{y}_i^{(j)})$ 对 x_{ij} 做回归的最小二乘估计量, 所以 LASSO 估计量相当于对 OLS 估计量做了一个 soft thresholding，如图6的左图所示。而 ridge 估计量相当于给 OLS 估计量乘了一个缩小因子 (shrinkage factor): 当解释变量相互正交时，即 $X^\top X = I$, $\hat{\beta}_{j,ridge} = \dfrac{1}{1+\lambda}\hat{\beta}_{j,ols}$，如图6的右图所示。

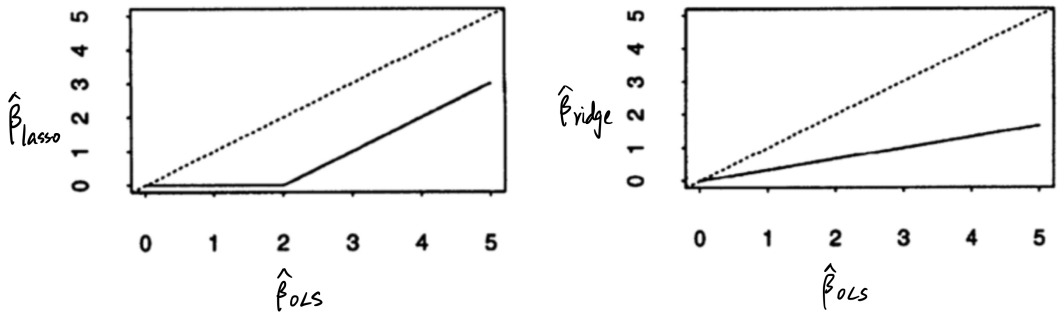

Figure 6: LASSO，ridge 估计量与 OLS 估计量的关系。Picture source: Tibshirani (1996)

Remarks

1. CD 算法可以很快计算出任意 λ 下 $\boldsymbol{\beta}$ 的最优解, 如图7所示。

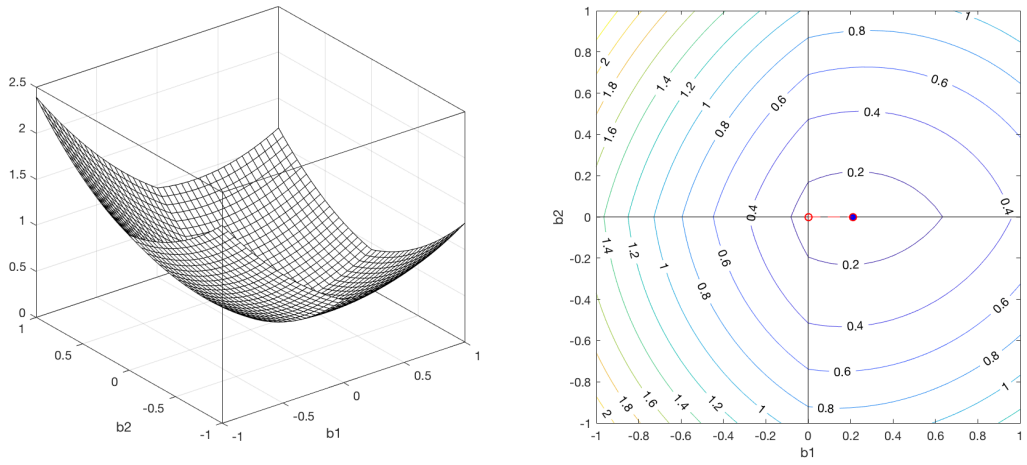

Figure 7: 左图：$\lambda = 0.3$ 对应的损失函数 (11)，其中真实的数据生成机制为 $y_i = 0.5x_{i1} + 0.01x_{i2}$，$x_{i1}, x_{i2} \overset{iid}{\sim} N(0,1)$，$i = 1, \ldots, 100$; 右图：contour plot of (11)，在该例中从初始值 (0,0) 出发只需一步迭代即到达全局最小值点 (0.21, 0).

2. 如何选取惩罚参数 λ? 先计算 λ 取不同值下 $\boldsymbol{\beta}$ 的估计量，然后用测试数据或交叉验证 (cross validation) 选出最优的 λ 取值，如图8所示。

3. 选取 λ 序列的具体做法：首先寻找一个最小的 $\lambda_{\max}$ 使得整个估计量 $\hat{\boldsymbol{\beta}} = \mathbf{0}$，再选取 $\lambda_{\min} = \epsilon\lambda_{\max}$，然后在 $\log(\lambda_{\max})$ 和 $\log(\lambda_{\min})$ 之间等间距地取 K 个 $\log(\lambda)$ 的值，得到从 $\lambda_{\max}$ 到 $\lambda_{\min}$ 的一列递减的 λ 取值，一般令 $\epsilon = 0.001$, $K = 100$.

   - 由(12)得，当 $\beta_k = 0$ $(k \neq j)$ 时，如果 $\left|\frac{1}{n}\sum_{i=1}^n x_{ij}y_i\right| < \lambda$, 那么 $\hat{\beta}_j = 0$, 因此可以选取 $\lambda_{\max} = \max_k \left|\frac{1}{n}\sum_{i=1}^n x_{ik}y_i\right|$.
   - 在 $\lambda_{\max}$ 下，$\hat{\boldsymbol{\beta}} = \mathbf{0}$. Friedman et al. (2010) 建议此后在每个 λ 下运行 CD 时，都以前一个（更大的）λ 下估计出的 $\hat{\boldsymbol{\beta}}$ 为初始值 (warm start)，这可以保证结果的稳定性，即 CD 算法的收敛值不会随初始值的选取变化。虽然在理论上使用 CD 可以确保找到(11)的全局最小值点，但有时受数值精度的影响，不同初始值收敛到的解可能会有细微差异。而且有很多例子表明通过 path solution $(\lambda_{\max} > \lambda_1 > \cdots > \lambda_r = \lambda)$ 计算某个较小 λ 下的 $\hat{\boldsymbol{\beta}}$ 比直接使用该 λ 计算用时更短。

4. Friedman et al. (2010) 建议在完整地更新每个系数 β_j 后，只更新 active set $(\beta_k \neq 0)$ 中的系数为算法提速。这种策略在变量很多而有用变量较少的情况下很有优势。

5. R package **glmnet** 使用 CD 算法估计 LASSO.

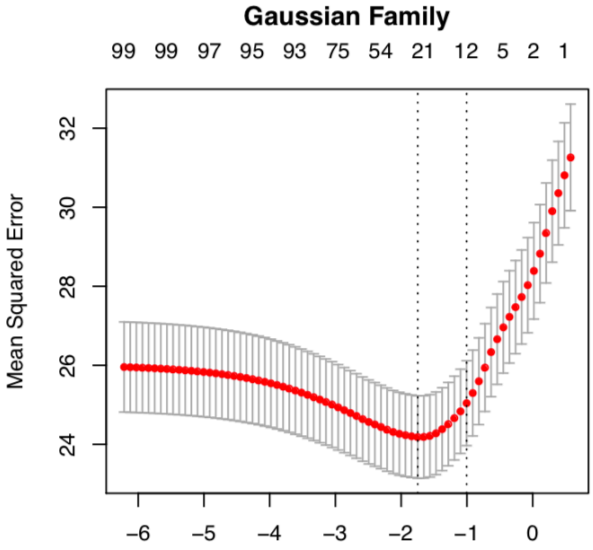

Figure 8: 10-fold cross-validation. 横坐标为 $\log(\lambda)$, 左边的虚线对应最小的 MSE, 右边的虚线对应最小的 MSE 加一个标准差，即 "one-standard-error" rule. Picture source: Friedman et al. (2010)

6. LASSO 的一个不足之处是: 如果 $p > n$, LASSO 最多能选出 n 个变量，因为此时 n 个系数就可以对模型完美拟合 (将残差降为 0), 不需要更多的非零系数。

2.2 The Hadamard product parametrization (HPP)

注意到下面的一元函数

$$f(x) = \frac{a^2}{x} + x, \ (a \neq 0) \tag{13}$$

在 $x > 0$ 上是凸函数。

$$f'(x) = -\frac{a^2}{x^2} + 1$$
$$f''(x) = \frac{2a^2}{x^3} > 0, \ (x > 0)$$

令 $f'(x) = 0$ 得 f 在 $x > 0$ 上的全局最小值是 $x = |a|$.

Hoff (2017) 证明了如下定理

Theorem 2. *对于函数 $f(\boldsymbol{\beta}) = h(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1$ 和 $g(\boldsymbol{u}, \boldsymbol{v}) = h(\boldsymbol{u} \circ \boldsymbol{v}) + \lambda(\boldsymbol{u}^\top \boldsymbol{u} + \boldsymbol{v}^\top \boldsymbol{v})/2$，其中 "$\circ$" 是 Hadamard (element-wise) product，有以下关系成立:*

$$\inf_{\boldsymbol{\beta}} \ f(\boldsymbol{\beta}) = \inf_{\boldsymbol{u}, \boldsymbol{v}} \ g(\boldsymbol{u}, \boldsymbol{v}).$$

Proof. 总能找到 $\boldsymbol{\beta}$ 和 $\boldsymbol{v}$ 使得 $\boldsymbol{u}=\boldsymbol{\beta}/\boldsymbol{v}$，其中 "/" 是 element-wise division.

$$\begin{aligned}\inf_{\boldsymbol{u},\boldsymbol{v}}\ g(\boldsymbol{u},\boldsymbol{v}) &= \inf_{\boldsymbol{\beta},\boldsymbol{v}}\ g(\boldsymbol{\beta}/\boldsymbol{v},\boldsymbol{v}) \\ &= \inf_{\boldsymbol{\beta}}\inf_{\boldsymbol{v}}\left\{h(\boldsymbol{\beta})+\frac{\lambda}{2}\left(\|\boldsymbol{\beta}/\boldsymbol{v}\|_2^2+\|\boldsymbol{v}\|_2^2\right)\right\} \\ &= \inf_{\boldsymbol{\beta}}\inf_{\boldsymbol{v}}\left\{h(\boldsymbol{\beta})+\frac{\lambda}{2}\sum_{j=1}^{p}\left(\frac{\beta_j^2}{v_j^2}+v_j^2\right)\right\} \\ &= \inf_{\boldsymbol{\beta}}\left\{h(\boldsymbol{\beta})+\frac{\lambda}{2}\sum_{j=1}^{p}\inf_{v_j}\left(\frac{\beta_j^2}{v_j^2}+v_j^2\right)\right\} \end{aligned} \tag{14}$$

如果 $\beta_j=0$, 那么当 $v_j=0$ 时，$\dfrac{\beta_j^2}{v_j^2}+v_j^2$ 取到最小值 0. 如果 $\beta_j\neq 0$, 根据(13), 当 $v_j^2=|\beta_j|$ 时，$\dfrac{\beta_j^2}{v_j^2}+v_j^2$ 取到最小值 $2|\beta_j|$. 不论哪种情况，(14)都可以化简为

$$\begin{aligned}\inf_{\boldsymbol{u},\boldsymbol{v}}\ g(\boldsymbol{u},\boldsymbol{v}) &= \inf_{\boldsymbol{\beta}}\left\{h(\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}|\beta_j|\right\} \\ &= \inf_{\boldsymbol{\beta}}\ f(\boldsymbol{\beta})\end{aligned}$$

□

Remarks

1. 定理2表明加入 L_1 惩罚的优化问题 $\min\limits_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$ 可以转化为 $\min\limits_{\boldsymbol{u},\boldsymbol{v}} g(\boldsymbol{u},\boldsymbol{v})$, 后者的目标函数 $g(\boldsymbol{u},\boldsymbol{v})$ 关于 $\boldsymbol{u}$, $\boldsymbol{v}$ 都可导，因此总可以使用梯度下降法求解。

2. 对于 LASSO 的目标函数(11), 此时 $h(\boldsymbol{\beta})=\dfrac{1}{2n}(\boldsymbol{y}-X\boldsymbol{\beta})^\top(\boldsymbol{y}-X\boldsymbol{\beta})$, 它是 $\boldsymbol{\beta}$ 的二次函数。对应的 $g(\boldsymbol{u},\boldsymbol{v})=h(\boldsymbol{u}\circ\boldsymbol{v})+\lambda(\boldsymbol{u}^\top\boldsymbol{u}+\boldsymbol{v}^\top\boldsymbol{v})/2$ 关于 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 都是 partial quadratic function.

   - 固定 $\boldsymbol{v}$, $g(\boldsymbol{u},\boldsymbol{v})$ 可以写为

   $$\begin{aligned}g(\boldsymbol{u},\boldsymbol{v}) &= \frac{1}{2n}(\boldsymbol{u}\circ\boldsymbol{v})^\top X^\top X(\boldsymbol{u}\circ\boldsymbol{v})-\frac{1}{n}(\boldsymbol{u}\circ\boldsymbol{v})^\top X^\top\boldsymbol{y}+\frac{1}{2n}\boldsymbol{y}^\top\boldsymbol{y}+\frac{\lambda}{2}(\boldsymbol{u}^\top\boldsymbol{u}+\boldsymbol{v}^\top\boldsymbol{v}) \\ &= \boldsymbol{u}^\top\left[\frac{1}{2n}X^\top X\circ\boldsymbol{v}\boldsymbol{v}^\top+\frac{\lambda}{2}I\right]\boldsymbol{u}-\boldsymbol{u}^\top\frac{1}{n}\left(X^\top\boldsymbol{y}\circ\boldsymbol{v}\right)+\cdots\end{aligned}$$

   因此给定 $\boldsymbol{v}$, $\boldsymbol{u}$ 的最小值点为

   $$\hat{\boldsymbol{u}}=\left[X^\top X\circ\boldsymbol{v}\boldsymbol{v}^\top+n\lambda I\right]^{-1}\left(X^\top\boldsymbol{y}\circ\boldsymbol{v}\right). \tag{15}$$

   同理可得，给定 $\boldsymbol{u}$, $\boldsymbol{v}$ 的最小值点为

   $$\hat{\boldsymbol{v}}=\left[X^\top X\circ\boldsymbol{u}\boldsymbol{u}^\top+n\lambda I\right]^{-1}\left(X^\top\boldsymbol{y}\circ\boldsymbol{u}\right). \tag{16}$$

- 由此可以构造一个估计 LASSO 的简便算法：
  - 选取初始值 $\boldsymbol{u}^{(0)}$, $\boldsymbol{v}^{(0)}$;
  - 按照(15), (16)轮流更新 $\boldsymbol{u}$, $\boldsymbol{v}$ 直到收敛。

  上述算法是一种 block coordinate descent, 且每部分有唯一的 (条件) 最小值点，因此算法会收敛到 g 的一个驻点 (Luenberger and Ye, 2008), 是否为局部极小值点还需借助 g 的二阶导数信息判断。由于 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的条件最小值点(15)和(16)的形式与 ridge regression 估计量(7)的形式相同，上述算法也被称为 iterative ridge regression.

References

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Hoff, P. D. (2017). Lasso, fractional norm and structured sparse estimation using a hadamard product parametrization. *Computational Statistics & Data Analysis*, 115:186–198.

Luenberger, D. G. and Ye, Y. (2008). *Linear and nonlinear programming*. Springer New York.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494.

