Algorithm for lasso

C.Q.Deng[1]

May,2019

1 Introduction

Suppose we are given N sapmles $\{(x_i, y_i)\}_{i=1}^N$, the lasso is

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|Y - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1$$

We can solve it anlyticly, since we need the differential be zero,

$$-2\mathbf{X}^\mathsf{T}(Y - \mathbf{X}\beta) + \lambda\eta = 0,\ \eta \in \partial\|\beta\|_1$$

The subdifferenial of $\|\beta\|_1$ is $\partial\|\beta\|_1 = \{\eta \,|\, \eta_j = sign(\beta_j),\ \beta_j \neq 0\,;\ \eta_j \in [-1, 1],\ \beta_j = 0\}$. So, we need

$$\mathbf{X}^\mathsf{T}\mathbf{X}\hat{\beta} = \mathbf{X}^\mathsf{T}Y - \frac{\lambda}{2}\hat{\eta},\ \hat{\eta}_j = \begin{cases} sign(\hat{\beta}_j) & \hat{\beta}_j \neq 0 \\ [-1, 1] & \hat{\beta}_j = 0 \end{cases}$$

Under the orthonormal setting, the explicit form of the lasso can be written as.

$$\hat{\beta}_j = \begin{cases} 0 & |\mathbf{X}_j^\mathsf{T}Y| \leq \frac{\lambda}{2} \\ \mathbf{X}_j^\mathsf{T}Y - \frac{\lambda}{2}\, sign(\mathbf{X}_j^\mathsf{T}Y) & |\mathbf{X}_j^\mathsf{T}Y| > \frac{\lambda}{2} \end{cases}$$

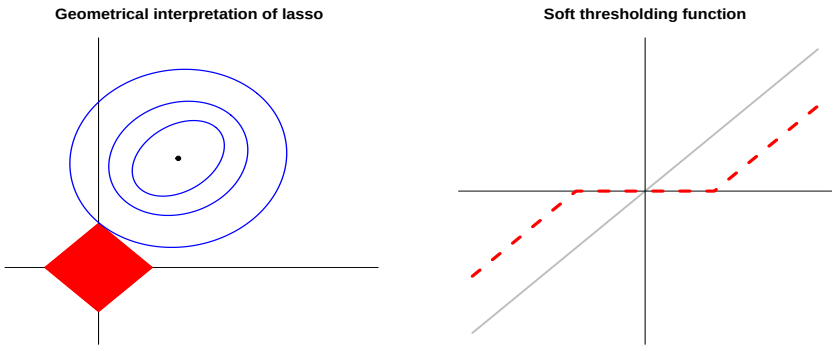

Figure 1: Explain of lasso

It is known as the soft-thresholding function, as is shown in Figure 1.

[1]Contact: upcdcq@outlook.com

2 Coordinate descent algorithm

Coordinate descent is an iterative algorithm that updates from β^t to β^{t+1} by choosing a single coordinate to update, and then performing a univariate minimization over this coordinate. More precisely, if coordinate k is chosen at iteration t, then the update is given by

$$\beta_k^{t+1} = \arg\min_{\beta_k} f(\beta_1^t, \cdots, \beta_{k-1}^t, \beta_k, \beta_{k+1}^t, \cdots, \beta_p^t)$$

and $\beta_j^{t+1} = \beta_j^t$ for $j \neq k$. A typical choice would be to cycle through the coordinates in some fixed order.

The lasso problem has many similar loss functions, and hence different algorithms to update the parameters. Since it all follows the same method, we then start from the loss function as

$$\begin{aligned} L(\beta) =& \|Y - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \\ =& \sum_{i=1}^{N}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{n} |\beta_i| \end{aligned}$$

Algorithm 1 Coordinate descent for solving lasso without bias

Input: Unnormalized data $\{\mathbf{X}_i, Y_i\}_{i=1}^n$

1: Initialize the starting point $\beta = \beta^0$
2: Compute $z_j = \sum_{i=1}^N x_{ij}^2$,
3: **repeat**
4: **for** $j = 1, \cdots, p$ **do**
5: Update $r_j = \sum_{i=1}^n (y_i - \sum_{k\neq j} x_{ik}\beta_k)x_{ij}$
6: Compute $\beta_j = \dfrac{1}{z_j} S_\lambda(2r_j)$
7: **end for**
8: **until** convergence

Output: The sequence $\{\beta_j\}_{j=1}^p$

Our current focus is on β_j, treating the others are given. Let

$$z_j = \sum_{i=1}^{N} x_{ij}^2, \ r_j = \sum_{i=1}^{n}(y_i - \sum_{k\neq j} x_{ik}\beta_k)x_{ij},$$

Similarly, r_j is the marginal regression coefficient of $\mathbf{X}_j$ repecting to errors which remove the other variables. Then on this specific dimension,

$$\frac{\partial L(\beta)}{\partial \beta_j} = -2r_j + 2z_j\beta_j + \lambda\frac{\partial|\beta_j|}{\beta_j}$$

Since $L(\beta)$ is convex, the minimization of β_j is the solution in β_j of $\dfrac{\partial L(\beta)}{\partial \beta_j} = 0$ when such a solution exists and is $\beta_j = 0$ otherwise. So the function $L(\beta_j|\beta_{\setminus j})$ is minimized at

$$\beta_j = \frac{1}{z_j}\, sign(2r_j)\,(2r_j - \lambda)_+ = \frac{1}{z_j} S_\lambda(2r_j)$$

Where $S_\lambda(x) = sign(x)\,(x-\lambda)_+$. So it moves on to the next β component until convergence as is shown in Algorithm 1.

Finally, if the model include bias term β_0, then the cost function is

$$\begin{aligned} L(\beta) &= \|Y - \beta_0 \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \\ &= \sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{i=1}^{n}|\beta_i| \end{aligned}$$

Taking derivatives for β_0, we have

$$\frac{\partial L(\beta)}{\partial \beta_0} = -2\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_{ik}\beta_k) = 2N\beta_0 - 2\sum_{i=1}^{n}(y_i - \sum_{k=1}^{p} x_{ik}\beta_k)$$

Hence, set $\dfrac{\partial L(\beta)}{\partial \beta_0} = 0$, we get

$$\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{n}(y_i - \sum_{k=1}^{p} x_{ik}\beta_k)$$

The algorithm for solving lasso with bias is shown in Algorithm 2. Similarly, orther loss function can be designed.

Algorithm 2 Coordinate descent for solving lasso with bias

Input: Unnormalized data $\{\mathbf{X}_i, Y_i\}_{i=1}^n$

1: Initialize the starting point $\beta = \beta^0$
2: Compute $z_j = \sum_{i=1}^N x_{ij}^2$,
3: **repeat**
4: **for** $j = 1, \cdots, p$ **do**
5: Update $r_j = \sum_{i=1}^n (y_i - \beta_0 - \sum_{k\neq j} x_{ik}\beta_k)x_{ij}$
6: Compute $\beta_j = \dfrac{1}{z_j} S_\lambda(2r_j)$ and $\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^n (y_i - \sum_{k=1}^p x_{ik}\beta_k)$
7: **end for**
8: **until** convergence

Output: The sequence β_0, $\{\beta_j\}_{j=1}^p$

3 Simulation studies

In this section, we give some simulation studies and empirical experiments where we use the softwart R to do research. In the first simulation we have $X_1 \sim \mathcal{N}(-4, 1.5^2)$, $X_2 \sim \text{Exp}(4)$, $X_3 \sim \Gamma(5)$, $X_4 \sim t(2)$. My profile is initialized randomly which is different from the traditional way, but the results shows that still works.

Data generation	dimension	design matrix X and ε	parameter β
$Y = X\beta + \varepsilon$	$N = 500, p = 4$	$\varepsilon \sim \mathcal{N}(0, 0.5^2)$	$(1, 2, 3, 0.001)^\mathsf{T}$
$Y = X\beta$	$N = 200, p = 10$	$X_i \sim \mathcal{N}(0, 1)$	$(\underbrace{1, \cdots, 1}_{5}, 0, \cdots, 0)^\mathsf{T}$
$Y = X\beta$	$N = 100, p = 200$	$X_i \sim \mathcal{N}(0, 1)$	$(\underbrace{1, \cdots, 1}_{5}, 0, \cdots, 0)^\mathsf{T}$
dateset *diabetes*	$N = 442, p = 10$	NA	NA

Table 1: Simulation experiments

We fit the model by coordinate descent algorithm with some modifications and plot the 10-fold CV for different λ. All the figures shown blow is compared with the R package *glmnet*.

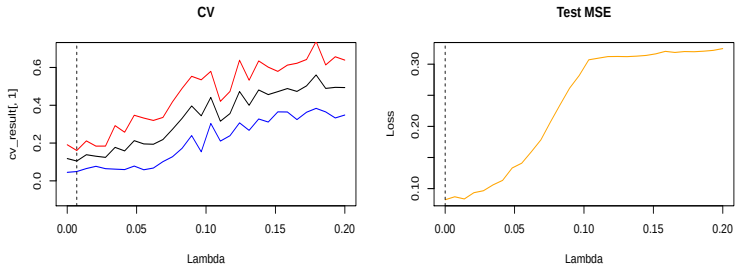

Figure 2: The cross-validation error and test MSE of Experment 1

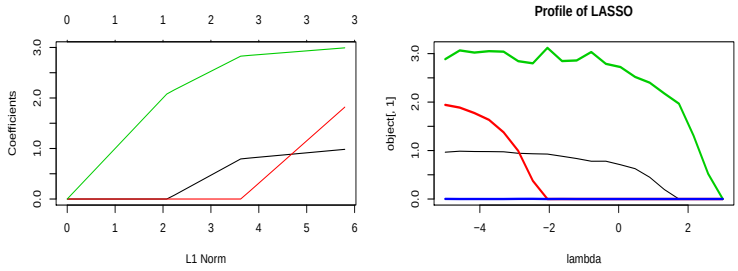

Figure 3: The profile of lasso for Experment 1

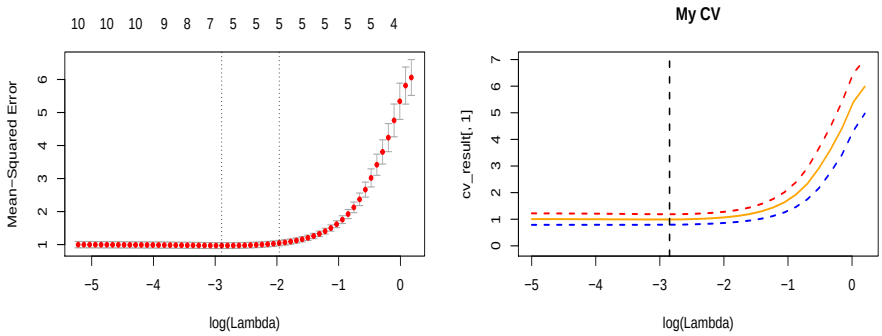

Figure 4: The cross-validation error comparision of Experment 2

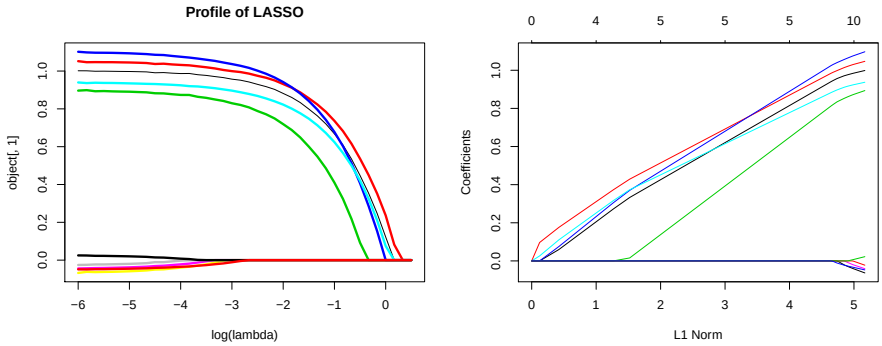

Figure 5: The profile of lasso for Experment 2

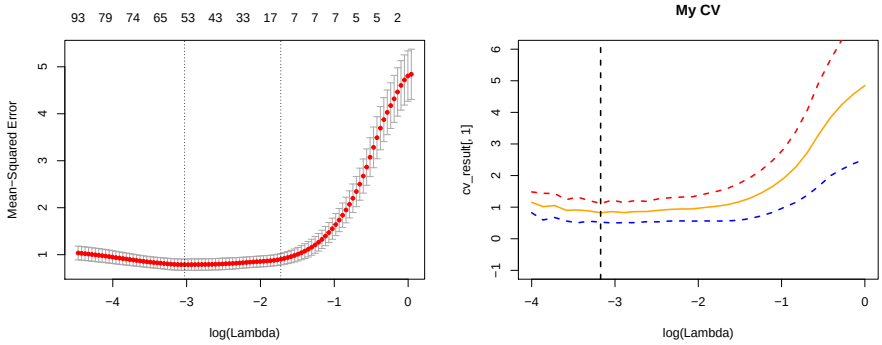

Figure 6: The cross-validation error comparision of Experment 3

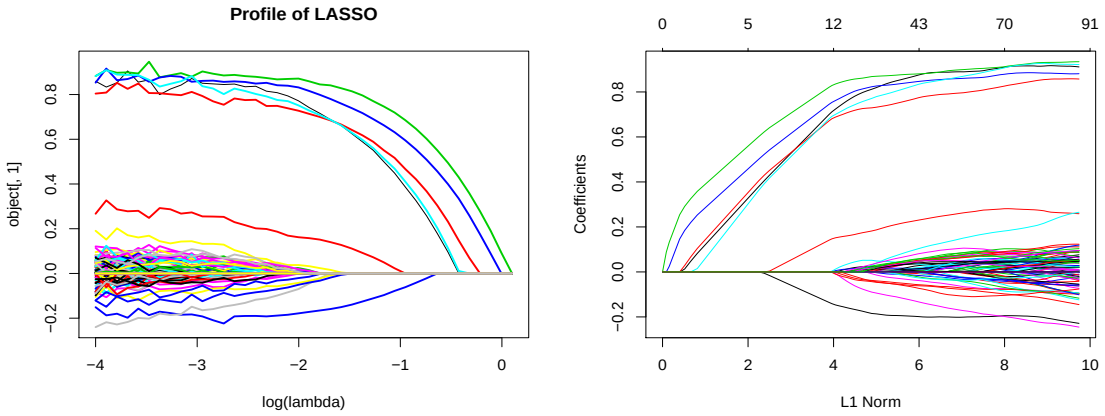

Figure 7: The profile of lasso for Experment 3

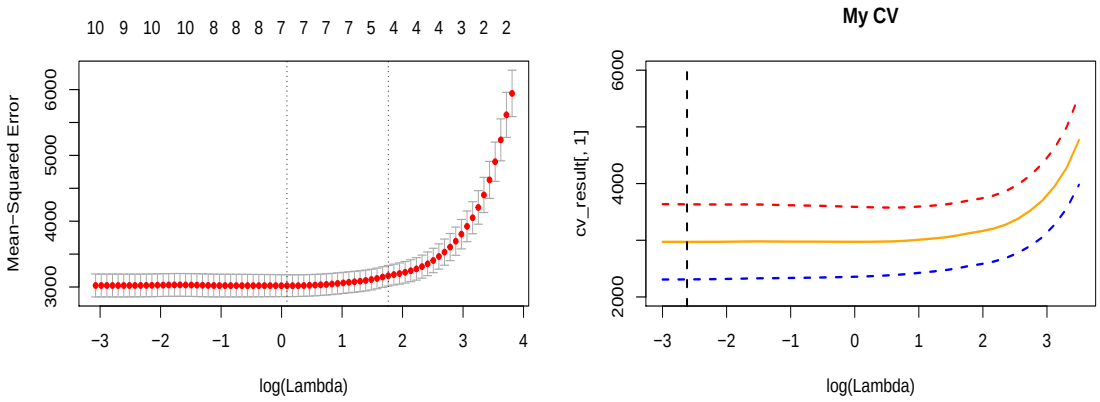

Figure 8: The cross-validation error comparision of Experment 4

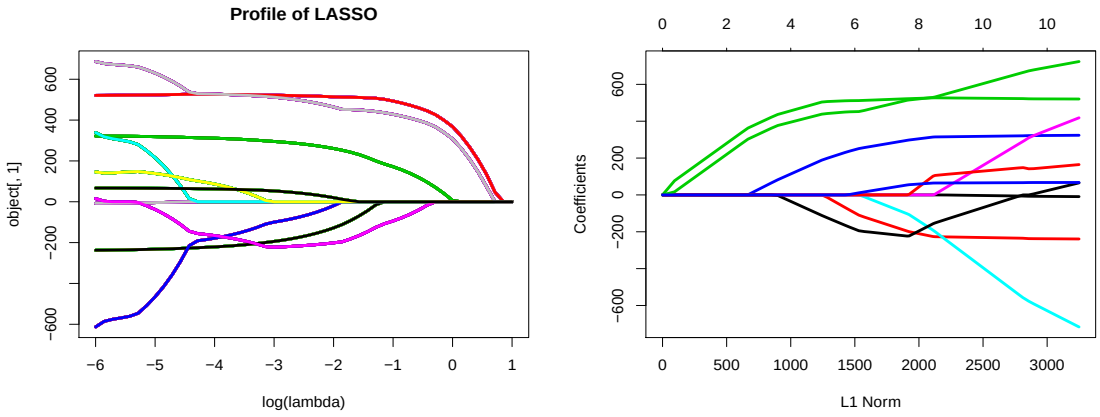

Figure 9: The profile of lasso for Experment 4