Exercise 7 ADMM for Lasso **Cenying(Tracy) Yang**

SDS 385 Due Date: 10/26/2016

Prof. James Scott

Objective Function

$$\underset{x}{\text{minimize}} \quad \frac{1}{2}\|y - X\beta\|_2^2 + \gamma\|\beta\|_1$$

where $\gamma > 0$ is the ℓ_1 norm regularization penalty parameter.

ADMM Form of Problem

$$\begin{aligned} &\underset{x}{\text{minimize}} && \frac{1}{2}\|y - X\beta\|_2^2 + \gamma\|z\|_1 + \frac{\rho}{2}\|\beta - z\|_2^2 \\ &\text{subject to} && \beta - z = 0 \end{aligned}$$

The augmented Lagrangian of the above problem is

$$L(\beta, z, \lambda) = \frac{1}{2}\|y - X\beta\|_2^2 + \gamma\|z\|_1 + \frac{\rho}{2}\|\beta - z\|_2^2 + \lambda(\beta - z) \tag{1}$$

ADMM Algorithm

$$\begin{aligned} \beta^{k+1} &= \arg\min_{\beta} L(\beta, z^k, \lambda^k) \\ z^{k+1} &= \arg\min_{z} L(\beta^{k+1}, z, \lambda^{k+1}) \\ \lambda^{k+1} &= \lambda^k + \rho(\beta^{k+1} - z^{k+1}) \end{aligned}$$

The key takeaway of the algorithm is that it breaks original problem into a sequence of two sub optimization problem. In the first step where we optimize $L(\beta, z, \lambda)$ with respect to only β, the ℓ_1 penalty term $\gamma\|z\|_1$ disappears and the optimization is reduced to simple and efficient least squares regression. In the second step where we optimize $L(\beta, z, \lambda)$ with respect to only z, the term $\frac{1}{2}\|y - X\beta\|_2^2$ disappears, allowing z to be solved independently across each element through soft-thresholding method. Finally, ADMM algorithm updates Lagrangian multiplier λ based on current estimates β and z. Note that the penalty parameter ρ that is introduced into ADMM form of problem plays a special role here, as it allows us to employ an imperfect estimate of λ when solving for both β and z.

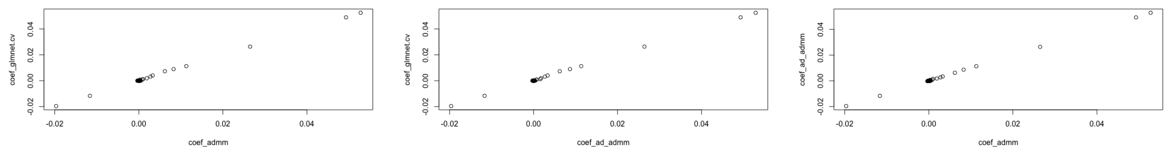
Now, taking partial derivative of equation (1) with respect to β and setting it to 0, we get

$$\begin{aligned} &\frac{\partial L}{\partial \beta} = -(y - X\beta)^T X + \rho(\beta - z) + \lambda = 0 \\ &\Rightarrow \beta = (X^T X + \rho I)^{-1}(X^T y + \rho z - \lambda) \end{aligned}$$

Then the **pseudo-code** for ***ADMM algorithm*** can be summarized as follows

1. $\beta^{k+1} = (X^T X + \rho I)^{-1}(X^T y + \rho z^k - \lambda^k)$
2. $z^{k+1} = S_{\lambda/\rho}(\beta^{k+1} + \frac{\lambda^k}{\rho})$
3. $\lambda^{k+1} = \lambda^k + \rho(\beta^{k+1} - z^{k+1})$

In order to compare the results with those of glm, the objective function is set at $\frac{1}{2n}\|y-X\beta\|_2^2+\gamma\|\beta\|_1$ instead of $\frac{1}{2}\|y-X\beta\|_2^2+\gamma\|\beta\|_1$, where n is the number of observation. I implement two versions of ADMM, one with constant ρ, the other with changing ρ. The stopping criterion can be found in Boyd et al. (2010) Section 3.3. I use the minimum λ obtained from cv.glmnet function as the γ under this context.
Figure 1 shows the estimated coefficients of three algorithms. The approximate linear relationship indicates that ADMM returns similar estimates to glmnet method. The detailed results can be found in csv file.



(a) ADMM with constant rho v.s. glmnet
(b) ADMM with changing rho v.s. glmnet
(c) ADMM with constant rho v.s. ADMM with changing rho

Figure 1: Estimated coefficients for glmnet and two versions of ADMM

Figure 2 presents the comparison of convergence speed of two versions of ADMM. As we can see, ADMM with changing ρ converges faster than ADMM with constant ρ.

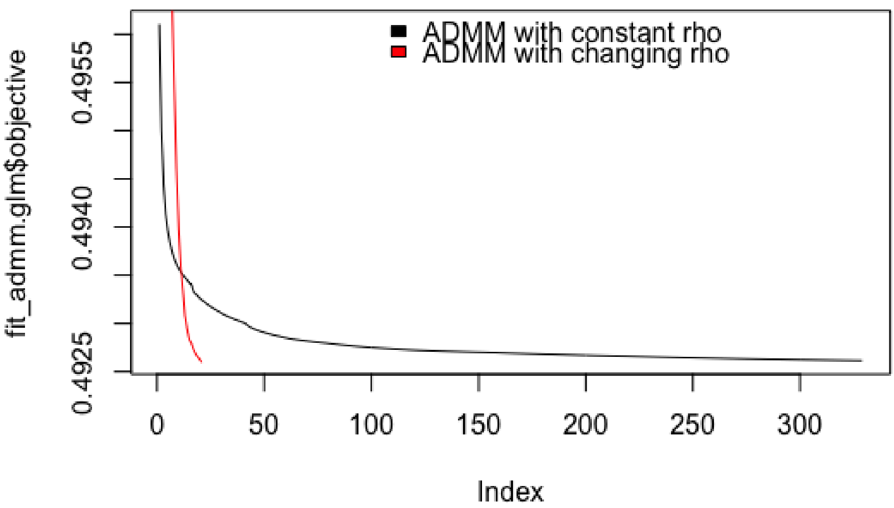


Figure 2: convergence speed for two versions of ADMM