Exercise 6 The proximal gradient method **Cenying(Tracy) Yang**
SDS 385 Due Date: 10/24/2016
Prof. James Scott

Proximal operators

(A) **Solution**

$$\text{prox}_\gamma \hat{f(x;} x_0) = \arg\min_z f(x_0) + (z - x_0)^T \nabla f(x_0) + \frac{1}{x\gamma}\|z - x_0\|_2^2 \tag{1}$$

Take the gradient of the objective function and set it to zero, we have

$$\nabla f(x_0) + \frac{1}{\gamma}(z - x_0) = 0 \Rightarrow z = x_0 - \gamma \nabla f(x_0) \tag{2}$$

Therefore, $\text{prox}_\gamma \hat{f(x;} x_0) = x_0 - \gamma \nabla f(x_0)$, which is indeed identical to a gradient-descent algorithm with step direction $\nabla f(x_0)$ and step size γ starting from x_0.

(B) **Solution**

$$\text{prox}_{1/\gamma}\, l(x) = \arg\min_z \frac{1}{2} z^T P z - q^T z + r + \frac{\gamma}{2}\|z - x\|_2^2 \tag{3}$$

Take the gradient of the objective function and set it to zero, we have

$$Pz - q + \gamma(z - x) = 0 \Rightarrow z = (P + \gamma I)^{-1}(q + \gamma x) \tag{4}$$

Consider that y is generated conditionally on x by $(y|x) \sim N(Ax, \Omega^{-1})$. The log-likelihood of y is

$$\log l(y) = \frac{1}{2}\log(|\Omega^{-1}|) + \frac{1}{2}(y - Ax)^T \Omega (y - Ax) + \frac{n}{2}\log(2\pi) \tag{5}$$

where n is the number of sample data. Drop the terms which do not depend on x, we have

$$\begin{aligned} \log l(y) &\propto \frac{1}{2}(y - Ax)^T \Omega (y - Ax) \\ &= \frac{1}{2}(y^T \Omega y - 2y^T \Omega A x + x^T A^T \Omega A x) \\ &= \frac{1}{2} x^T A^T \Omega A x - y^T \Omega A x + \frac{1}{2} y^T \Omega y \end{aligned} \tag{6}$$

Therefore, $P = A^T \Omega A$, $q = y^T \Omega A$, and $r = \frac{1}{2} y^T \Omega y$.

(C) **Solution**

$$\text{prox}_\gamma\, \Phi(x) = \arg\min_z \tau\|z\|_1 + \frac{1}{2\gamma}\|z - x\|_2^2 \tag{7}$$

The optimality condition for the problem is

$$0 \in \nabla(\frac{1}{2\gamma}\|z - x\|_2^2) + \partial(\tau\|z\|_1) \Leftrightarrow 0 \in \frac{1}{\gamma}(z - x) + \tau \partial\|z\|_1 \tag{8}$$

Consider each of $\|z\|_1$ component separately. Let's examine first the case where $z_i \neq 0$. Then, $\partial\|z_i\|_1 = \text{sgn}(z_i)$ and the optimum z_i^* is obtained as

$$0 = \frac{1}{\gamma}(z_i - x_i) + \tau\,\text{sgn}(z_i) \Leftrightarrow z_i^* = x_i - \tau\gamma\,\text{sgn}(z_i^*) \tag{9}$$

Note that if $z_i^* < 0$, then $x_i < -\tau\gamma$ and equivalently if $z_i^* > 0$, then $x_i > \tau\gamma$. Thus $|x_i| > \tau\gamma$ and $\text{sgn}(z_i^* = \text{sgn}(x_i))$. Substituting in equation (9), we have $z_i^* = x_i - \tau\gamma\,\text{sgn}(x_i)$.
In the case where $z_i = 0$, the subdifferential of the ℓ_1 norm is the interval $[-1, 1]$ and the optimality condition is

$$0 \in -\frac{1}{\gamma}x_i + \tau[-1, 1] \Leftrightarrow |x_i| \leq \tau\gamma \tag{10}$$

Putting all together, we have

$$[\text{prox}_\gamma\, \Phi(x)]_i = \text{sgn}(x_i)\max(|x_i| - \tau\gamma, 0) \tag{11}$$

which is exactly the soft thresholding method with $\lambda = \tau\gamma$.

The proximal gradient method

(A) **Solution**

$$\begin{aligned}\text{prox}_\gamma\, \Phi(u) &= \arg\min_x \Phi(x) + \frac{1}{2\gamma}\|x - x_0 + \gamma\nabla l(x_0)\|_2^2 \\ &= \arg\min_x \Phi(x) + l(x_0) + (x - x_0)^T\nabla l(x_0) + \frac{1}{2\gamma}\|x - x_0\|_2^2 \\ &= \arg\min_x \tilde{l}(x; x_0) + \Phi(x)\end{aligned} \tag{12}$$

(B) **Solution**
The proximal gradient method is an iterative algorithm which can be stated as

$$x^{t+1} = \text{prox}_\gamma\, \phi(u^{t+1}), \quad u^{t+1} = x^t - \gamma\nabla l(x^t) \tag{13}$$

In the case of lasso regression, we find

$$\hat{\beta} = \arg\min_\beta = \|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{14}$$

consider the splitting

$$l(\beta) = \|y - X\beta\|_2^2, \quad \Phi(\beta) = \lambda\|\beta\|_1 \tag{15}$$

The gradient of $l(\beta)$ is

$$\nabla l(\beta) = -2X^T(y - X\beta) = 2X^T(X\beta - y) \tag{16}$$

With this we can frame our proximal gradient method for lasso regression as follows:

$$\beta^{t+1} = \text{prox}_\gamma\, \lambda\|u^{t+1}\|_1 \tag{17}$$

$$\begin{aligned}u^{t+1} &= \beta^t - \gamma\nabla l(\beta^t) \\ &= \beta^t + 2\gamma X^T(y - X\beta^t)\end{aligned} \tag{18}$$

From previous exercise, the element-wise solution is

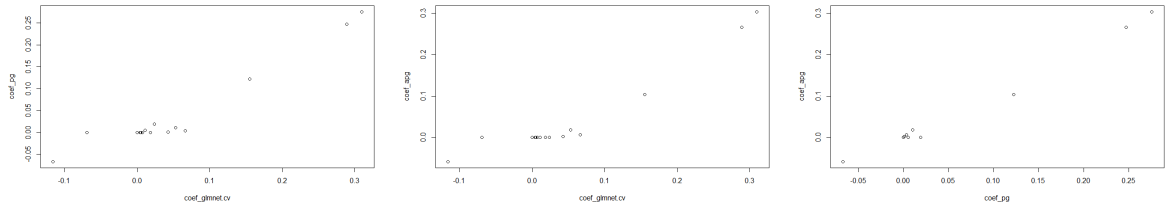
$$\beta_i^{t+1} = \text{sgn}(u_i^t)(|u_i^t| - \gamma\lambda)_+ \tag{19}$$

Evaluating $\nabla l(\beta)$ requires one matrix-vector multiply by X and one by X^T plus a (negligible) vector addition. Evaluating the proximal operator of Φ is negligible. Thus, each iteration of the

proximal gradient method requires one matrix-vector multiply by X, one matrix-vector multiply by X^T, and a few vector operations. And the primary computational costs of this algorithm is the matrix multiply of X and X^T.

(C) **Compare Results**

We compare results of glmnet, proximal gradient (PG), and accelerated proximal gradient (APG) in this part. Figure 1 shows the estimated coefficients of three algorithms. The approximate linear relationship indicates that PG and APG return similar estimates to glmnet method. The detailed results can be found in csv file.



(a) glmnet v.s. PG (b) glmnet v.s. APG (c) PG v.s. APG

Figure 1: Estimated coefficients for glmnet, PG and APG

Figure 2 presents the comparison of convergence speed of PG and APG. As we can see, APG does converge faster than PD.

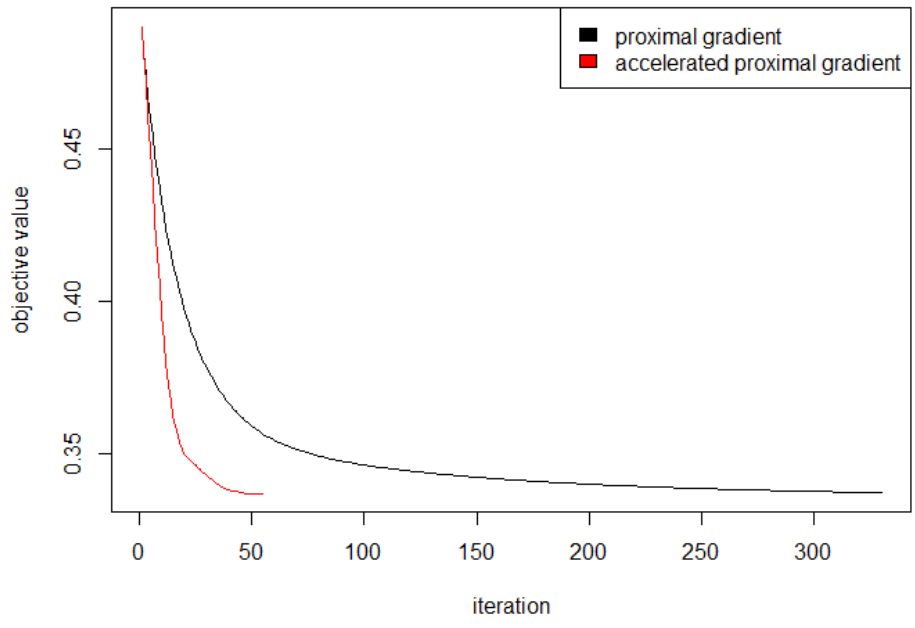


Figure 2: convergence speed for PG and APG

Class Note

In order to show $x^* = \arg\min_x f(x)$, we need to show two things.

- $E_\gamma f(x^*) = f(x^*)$
- $E_\gamma f(\tilde{x}) \geq_\gamma f(x^*)$

First point, $E_\gamma f(x^*) = \min_z f(z) + \frac{1}{2\gamma}\|z - x^*\|^2 = f(x^*)$.
Second point, $E_\gamma f(\tilde{x}) = \min_z f(z) + \frac{1}{2\gamma}\|z - \tilde{x}\|^2 \geq \min_z f(z) + \min_z \frac{1}{2\gamma}\|z - \tilde{x}\|^2 \geq f(x^*)$
The relationship between primal and dual is the same as the relationship between constrained optimization and adding penalized term.
downside of C_p statistics: strong assumption that the underlying model is correct. The method will fail if this is not true.
source of bias of cross validation: we are using part of the data to estimate the coefficient, not the entire data. But ideally, we would like to have the minimum **generalization error** on the entire data. Extra variation: co-variance, the training data might be independent, but the test data might be correlated. Another downside of cross validation is computational cost.