SDS 385: Exercises 7 - Alternating Direction Method of Multipliers (ADMM) for Lasso

October 29, 2016

Professor Scott

Spencer Woody

Problem 1

ADMM

We use ADMM to form Lasso estimates. Let A be our feature matrix, and x is our coefficient vector, and b is the response vector.

$$\min \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|z\|_1 \tag{1}$$

$$\text{s.t. } x - z = 0 \tag{2}$$

The Lagrangian is

$$L_p(x, z, u) = \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|z\|_1 + u^T(x - z) + \frac{\rho}{2} \|x - z\|_2^2. \tag{3}$$

$$\begin{aligned} x^{k+1} &= \arg\min_x L_p(x, z^k, u^k) && (4) \\ &= \arg\min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \lambda \left\|z^k\right\|_1 + (u^k)^T(x - z) + \frac{\rho}{2} \left\|x - z^k\right\|_2^2 \right\} && (5) \end{aligned}$$

We take the gradient of the objective function with respect to x and set it equal to 0,

$$\begin{aligned} \nabla_x L_p(x, z, u) &= A^T(Ax^{k+1} - b) + u^k + \rho(x^{k+1} - z^k) && (6) \\ &= (A^TA + \rho I)x^{k+1} - (A^Tb + \rho z^k - u^k) = 0 && (7) \\ x^{k+1} &= (A^TA + \rho I)^{-1}(A^Tb + \rho z^k - u^k), \text{ let } v^k = u^k/\rho && (8) \\ x^{k+1} &= (A^TA + \rho I)^{-1}(A^Tb + \rho(z^k - v^k)) && (9) \end{aligned}$$

$$\begin{aligned} z^{k+1} &= \arg\min_z L_p(x^{k+1}, z, u^k) && (10) \\ &= \arg\min_z \left\{ \frac{1}{2} \left\|Ax^{k+1} - b\right\|_2^2 + \lambda \|z\|_1 + (u^k)^T(x^{k+1} - z) + \frac{\rho}{2} \left\|x^{k+1} - z\right\|_2^2 \right\} && (11) \\ &= \arg\min_z \left\{ \frac{\lambda}{\rho} \|z\|_1 + (v^k)^T(x^{k+1} - z) + \frac{1}{2} \left\|x^{k+1} - z\right\|_2^2 \right\} && (12) \\ &= \arg\min_z \left\{ \frac{\lambda}{\rho} \|z\|_1 - (z - x^{k+1})^T v^k + \frac{1}{2} \left\|z - x^{k+1}\right\|_2^2 \right\} && (13) \\ &= \operatorname*{prox}_{\gamma=1} \frac{\lambda}{\rho} \left\|w^k\right\| , w^k = x^{k+1} + v^k && (14) \end{aligned}$$

So z^{k+1} is subjected to the soft threshholding from the previous two exercises. Finally we update u^k, and, identically, v^k.

$$\begin{aligned} u^{k+1} &= u^k + \rho(x^{k+1} - z^{k+1}) && (15) \\ \rho v^{k+1} &= \rho v^k + \rho(x^{k+1} - z^{k+1}) && (16) \\ v^{k+1} &= v^k + x^{k+1} - z^{k+1} && (17) \end{aligned}$$

With $\lambda = 0.002420476$, the regular proximal gradient method takes 5887 iterations, the accelerated proximal gradient method takes 626 iterations, and the ADMM takes 146 iterations. See Figure 1.

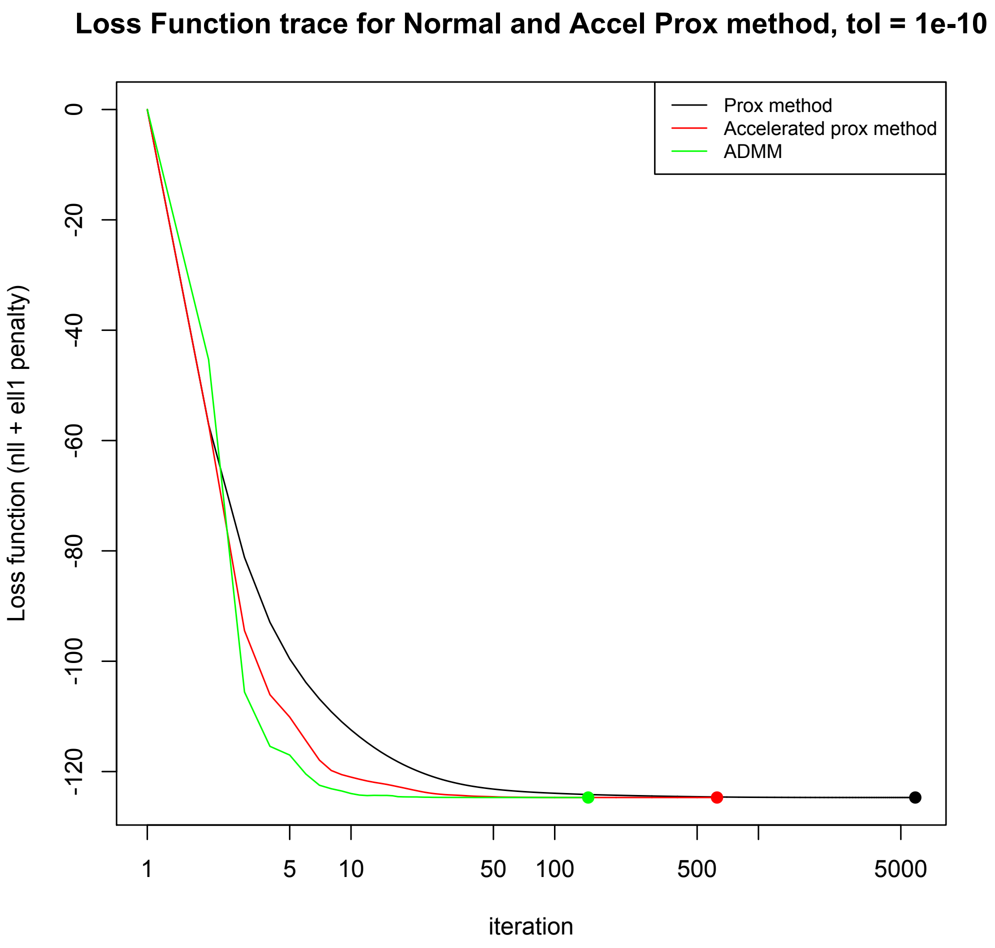

Figure 1: Comparison of convergence for different methods

Problem 2

Proximal Operators

(A) First, we state the definition of the Moreau envelope and the proximal gradient:

$$E_\gamma f(x) = \min_z \left\{ f(z) + \frac{1}{2\gamma} \|z - x\|_2^2 \right\} \leq f(x) \tag{18}$$

$$\operatorname*{prox}_\gamma f(x) = \arg\min_z \left\{ f(z) + \frac{1}{2\gamma} \|z - x\|_2^2 \right\} \tag{19}$$

For the local linear approximation of $f(x)$ about the point x_0,

$$f(x) \approx \hat{f}(x; x_0) = f(x_0) + (x - x_0)^T \nabla f(x_0) \tag{20}$$

the proximal operator is derived as follows:

$$\operatorname*{prox}_\gamma \hat{f}(x; x_0) = \arg\min_x \left\{ f(x_0) + (x - x_0)^T \nabla f(x_0) + \frac{1}{2\gamma} \|x - x_0\|_2^2 \right\} \tag{21}$$

$$= \arg\min_x \left\{ x^T \nabla f(x_0) + \frac{1}{2\gamma} \|x - x_0\|_2^2 \right\} \tag{22}$$

We take the gradient of the objective function with respect to the feature vector x and set it to zero to find the arg min.

$$\nabla f(x_0) - \frac{1}{\gamma}(x - x_0) = 0 \tag{23}$$

$$x = x_0 + \gamma \nabla f(x_0) \tag{24}$$

This is equivalent to gradient-step for $f(x)$ with size γ.

(B) Suppose we have a negative log-likelihood in the form

$$l(x) = \frac{1}{2} x^T P x - q^T x + r. \tag{25}$$

The proximal operator with parameter $1/\gamma$ of $l(x)$ is

$$\operatorname*{prox}_{1/\gamma} l(x) = \arg\min_z \left\{ l(z) + \frac{\gamma}{2} \|z - x\|_2^2 \right\} \tag{26}$$

$$= \arg\min_z \left\{ \frac{1}{2} z^T P z - q^T z + r + \frac{\gamma}{2} \|z - x\|_2^2 \right\} \tag{27}$$

Once again, taking the gradient of the objective function with respect to z and setting it to zero yields

$$Pz - q + \gamma(z - x) = 0 \tag{28}$$

$$(P + \gamma I)z = (\gamma x + q) \tag{29}$$

$$z = (P + \gamma T)^{-1}(\gamma x + q) \tag{30}$$

where I is the identity matrix, assuming that $(P + \gamma I)$ is invertible. Now suppose that y are generated

Problem 2 continued on next page. . .

conditionally on x by $(y|x) \sim N(Ax, \Omega^{-1})$. The log-likelihood of x may be expressed by

$$
\begin{aligned}
l(x) &\propto \frac{1}{2}(y - Ax)^T\Omega(y - Ax) && (31)\\
&= \frac{1}{2}(y^T - x^TA^T)\Omega(y - Ax) && (32)\\
&= \frac{1}{2}(y^T\Omega - x^TA^T\Omega)(y - Ax) && (33)\\
&= \frac{1}{2}(y^T\Omega y - y^T\Omega Ax - x^TA^T\Omega y + x^TA^T\Omega Ax) && (34)\\
&= \frac{1}{2}(y^T\Omega y - 2y^T\Omega Ax + x^TA^T\Omega Ax) && (35)\\
&= \frac{1}{2}x^TA^T\Omega Ax - y^T\Omega Ax + y^T\Omega y && (36)
\end{aligned}
$$

Therefore the negative log-likelihood takes the form expressed in (25) with $P = A^T\Omega A$, $q = A^T\Omega^T y$, and $r = y^T\Omega y$.

(C) Let $\phi(x) = \tau \|x\|_1$.

$$
\begin{aligned}
\operatorname*{prox}_{\gamma} \phi(x) &= \arg\min_z \left\{ \phi(x) + \frac{1}{2\gamma}\|z - x\|_2^2 \right\} && (37)\\
&= \arg\min_z \left\{ \tau\|z\|_1 + \frac{1}{2\gamma}\|z - x\|_2^2 \right\} && (38)\\
&= \arg\min_z \left\{ \tau\sum_{i=1}^{p}|z_i| + \frac{1}{2\gamma}\|z - x\|_2^2 \right\} && (39)
\end{aligned}
$$

We find the element-wise solution to find each z_i.

$$
\arg\min_{z_i} \left\{ \tau|z_i| + \frac{1}{2\gamma}(z_i - x_i)^2 \right\} = \arg\min_{z_i} \left\{ \frac{1}{2}(x_i - z_i)^2 + \gamma\tau|z_i| \right\} \quad (40)
$$

In the previous set of exercises, we showed that this is equal to the soft-thresholding function

$$
\operatorname{sign}(x_i)(|x_i| - \gamma\tau)_+. \quad (41)
$$

Problem 3

The proximal gradient method

(A) We want to minimize an objective function $f(x) = l(x) + \phi(x)$ where $l(x)$ is differentiable but $\psi(x)$ is not. We define the linear approximator of $l(x)$ as

$$l(x) \approx \tilde{l}(x; x_0) = l(x_0) + (x - x_0)^T \nabla l(x_0) + \frac{1}{2\gamma} \|x - x_0\|_2^2 . \tag{42}$$

Now our original objective function can be approximated with

$$f(x) \approx \tilde{f}(x; x_0) = \tilde{l}(x; x_0) + \phi(x). \tag{43}$$

We look to minimize this approximation of the objective function.

$$\hat{x} = \arg\min_x \{\tilde{f}(x; x_0)\} \tag{44}$$

$$= \arg\min_x \{\tilde{l}(x; x_0) + \phi(x)\} \tag{45}$$

$$= \arg\min_x \left\{ l(x_0) + (x - x_0)^T \nabla l(x_0) + \frac{1}{2\gamma} \|x - x_0\|_2^2 + \phi(x) \right\} \tag{46}$$

We can disregard the $l(x_0)$ term in (46) because it does not relate to x. Furthermore we can introduce another term $\frac{\gamma}{2} \nabla l(x_0)^T \nabla l(x_0)$ which also does not relate to x in order to complete the square. Now the ojective function becomes

$$\arg\min_x \left\{ \phi(x) + (x - x_0)^T \nabla l(x_0) + \frac{1}{2\gamma} \|x - x_0\|_2^2 + \frac{\gamma}{2} \nabla l(x_0)^T \nabla l(x_0) \right\} \tag{47}$$

$$= \arg\min_x \left\{ \phi(x) + (x - x_0)^T \nabla l(x_0) + \frac{1}{2\gamma} (x - x_0)^T (x - x_0) + \frac{\gamma}{2} \nabla l(x_0)^T \nabla l(x_0) \right\} \tag{48}$$

$$= \arg\min_x \left\{ \phi(x) + \frac{1}{2\gamma} \left(2\gamma (x - x_0)^T \nabla l(x_0) + (x - x_0)^T (x - x_0) + \gamma^2 \nabla l(x_0)^T \nabla l(x_0) \right) \right\} \tag{49}$$

$$= \arg\min_x \left\{ \phi(x) + \frac{1}{2\gamma} \left((x - x_0)^T (x - x_0) + 2\gamma (x - x_0)^T \nabla l(x_0) + \gamma^2 \nabla l(x_0)^T \nabla l(x_0) \right) \right\} \tag{50}$$

$$= \arg\min_x \left\{ \phi(x) + \frac{1}{2\gamma} (x - x_0 + \gamma \nabla l(x_0))^T (x - x_0 + \gamma \nabla l(x_0)) \right\} \tag{51}$$

$$= \arg\min_x \left\{ \phi(x) + \frac{1}{2\gamma} \|x - x_0 + \gamma \nabla l(x_0)\|_2^2 \right\} \tag{52}$$

$$= \arg\min_x \left\{ \phi(x) + \frac{1}{2\gamma} \|x - (x_0 - \gamma \nabla l(x_0))\|_2^2 \right\} \tag{53}$$

Therefore we see the solution for $\hat{x}$ is equivalent to

$$\hat{x} = \operatorname*{prox}_{\gamma} \phi(u), \text{ where } u = x_0 - \gamma \nabla l(x_0) \tag{54}$$

(B) The proximal gradient method is an iterative algorithm which expressed as

$$x^{(t+1)} = \operatorname*{prox}_{\gamma^{(t)}} \phi(u^{(t)}), \quad u^{(t)} = x^{(t)} - \gamma^{(t)} \nabla l(x^{(t)}) \tag{55}$$

With lasso linear regression, we find

$$\hat{\beta} = \arg\min_\beta \left\{ \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} . \tag{56}$$

Problem 3 continued on next page. . .

We may use proximal gradient method to accomplish this. Define

$$l(\beta) = \frac{1}{2}\|y - X\beta\|_2^2 = \frac{1}{2}(y - X\beta)^T(y - X\beta) = \frac{1}{2}y^Ty - \beta^TX^Ty + \frac{1}{2}\beta^TX^TX\beta \tag{57}$$

$$\nabla l(\beta) = X^TX\beta - X^Ty = -X^T(y - X\beta) \tag{58}$$

$$\phi(\beta) = \lambda\|\beta\|_1 . \tag{59}$$

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

$$\beta^{(t+1)} = \operatorname*{prox}_{\gamma^{(t)}} \lambda \left\| u^{(t)} \right\|_1 \tag{60}$$

$$u^{(t)} = \beta^{(t)} - \gamma^{(t)}\nabla l(\beta^{(t)}) \tag{61}$$

$$= \beta^{(t)} + \gamma^{(t)}X^T(y - X\beta^{(t)}) \tag{62}$$

Using the results from the previous exercise, we see that the element-wise solution is

$$\beta_i^{(t+1)} = \text{sign}(u_i^{(t)})(|u_i^{(t)}| - \gamma\lambda)_+ \tag{63}$$

The primary computational cost of this algorithm computing the two matrix products. Finding the proximal gradient is trivial, only involving the `sign` and `max` commands within R. Implementation in R gives similar results to the output of the `glmnet` package. See Figure 2. Note that in my code, I scale λ by N, the number of data points.

Figure 2: Comparison of output from my proximal gradient method (left) and output of `glmnet` (right)

(C) Now we implement the accelerated proximal gradient algorithm. The $z^{(t+1)}$ term is an extropolated version of $\beta^{(t+1)}$ based on the magnitude of the previous step. If we have taken a big step, we are more confident in going further in the direction we have gone.

$$\beta^{(t+1)} = \operatorname*{prox}_{\gamma^{(t)}} \lambda \left\| u^{(t)} \right\|_1 \tag{64}$$

$$u^{(t)} = z^{(t)} - \gamma^{(t)}\nabla l(z^{(t)}) \tag{65}$$

$$s^{(t+1)} = \frac{1 + (1 + 4(s^{(t)}))^{1/2}}{2} \tag{66}$$

$$z^{(t+1)} = \beta^{(t+1)} + \left(\frac{s^{(t)} - 1}{s^{(t+1)}}\right)\left(\beta^{(t+1)} - \beta^{(t)}\right) \tag{67}$$

When implemented in R, the accelerated method converges much faster than the regular method. See Figure 3. My convergence criteria is a relative change in negative log-likelihood of less than 10^{-10}. Holding $\lambda = 0.001$, the accelerated method converges in 616 iterations, while the regular method converges in 19,953 iterations. Comparing the final given values of the coefficients, 63 of 64 covariates have the same sign for their coefficients. The one outstanding covariate, `hdl.tch`, has a coefficient of 0.00267 in the unaccelerated method, and a value of zero in the accelerated method. Figure 4 compares all coefficients for the two methods. Clearly the coefficients are in close agreement when computed under the two different methods.

Figure 3: Comparison of convergence for prox method vs. accelerated prox method, $\lambda = 0.001$

Figure 4: Comparison of coefficients for prox method vs. accelerated prox method, $\lambda = 0.001$

Figure 5: Convergence of $\frac{s^{(t)}-1}{s^{(t+1)}}$

R script for `myfuns.R`

R script for `e6.R`