HW 6

Evan Ott
UT EID: eao466

October 16, 2016

1 Proximal operators

(A)

$$\begin{aligned}
f(x) \approx \hat{f}(x; x_0) &= f(x_0) + (x - x_0)^\top \nabla f(x_0) \\
\text{prox}_\gamma \hat{f}(x) &= \arg\min_z \left[\hat{f}(z) + \frac{1}{2\gamma}\|z - x\|_2^2\right] \\
&= \arg\min_z \left[f(x_0) + (z - x_0)^\top \nabla f(x_0) + \frac{1}{2\gamma}\|z - x\|_2^2\right] \\
0 &= \frac{\partial}{\partial z}\left[f(x_0) + (z - x_0)^\top \nabla f(x_0) + \frac{1}{2\gamma}\|z - x\|_2^2\right] \\
&= 0 + \nabla f(x_0) + \frac{1}{2\gamma}(2z^* - 2x) = \nabla f(x_0) + \frac{1}{\gamma}(z^* - x) \\
\text{prox}_\gamma \hat{f}(x) = z^* &= x - \gamma \nabla f(x_0)
\end{aligned}$$

which is indeed the gradient-descent step for $f(x)$ of size γ starting at x_0.

(B)

$$\begin{aligned}
l(x) &= \frac{1}{2}x^\top P x - q^\top x + r \\
\text{prox}_{1/\gamma} l(x) &= \arg\min_z \left[\hat{l}(z) + \frac{\gamma}{2}\|z - x\|_2^2\right] \\
&= \arg\min_z \left[\frac{1}{2}z^\top P z - q^\top z + r + \frac{\gamma}{2}\|z - x\|_2^2\right] \\
0 &= \frac{\partial}{\partial z}\left[\frac{1}{2}z^\top P z - q^\top z + r + \frac{\gamma}{2}\|z - x\|_2^2\right] \\
&= P z^* - q + \gamma(z^* - x) \\
\gamma x + q &= (P + \gamma I)\, z^* \\
\text{prox}_{1/\gamma} l(x) = z^* &= (P + \gamma I)^{-1} (\gamma x + q)
\end{aligned}$$

assuming $(P + \gamma I)^{-1}$ exists.

If we have $y|x \sim N(Ax, \Omega^{-1})$ with y having n rows, then

$$\begin{aligned}
L(y|x) &= \frac{1}{\sqrt{2\pi}^n}|\Omega|^{-1/2} \exp\left[-\frac{1}{2}(y - Ax)^\top \Omega^{-1}(y - Ax)\right] \\
n(y|x) = -\log L(y|x) &= \frac{1}{2}\log|\Omega| + \frac{n}{2}\log(2\pi) + \frac{1}{2}(y - Ax)^\top \Omega^{-1}(y - Ax) \\
&= \frac{1}{2}y^\top \Omega^{-1} y - (Ax)^\top \Omega^{-1} y + \frac{1}{2}(Ax)^\top \Omega^{-1} Ax + \frac{1}{2}\log|\Omega| + \frac{n}{2}\log(2\pi)
\end{aligned}$$

So $P = \Omega^{-1}$, $q = \Omega^{-1}Ax$ (because $\Omega = \Omega^\top$ since it is a covariance matrix), and $r = \frac{1}{2}(Ax)^\top \Omega^{-1} Ax + \frac{1}{2}\log|\Omega| + \frac{n}{2}\log(2\pi)$.

(C)

$$
\begin{aligned}
\phi(x) &= \tau\|x\|_1 \\
\operatorname{prox}_\gamma \phi(x) &= \arg\min_z \left[\phi(z) + \frac{1}{2\gamma}\|z-x\|_2^2\right] \\
&= \arg\min_z \left[\tau\|z\|_1 + \frac{1}{2\gamma}\|z-x\|_2^2\right] \\
&= \arg\min_z \left[\tau\sum_{i=1}^n (|z_i|) + \frac{1}{2\gamma}\sum_{i=1}^n \left((z_i-x_i)^2\right)\right] \\
&= \arg\min_z \left[\sum_{i=1}^n \frac{1}{2\gamma}(z_i-x_i)^2 + \tau|z_i|\right] \\
&= \arg\min_z \left[\sum_{i=1}^n \frac{1}{2}(z_i-x_i)^2 + \tau\gamma|z_i|\right] \qquad \text{(multiplying by positive scalar yields same optimization)}
\end{aligned}
$$

The term being minimized for each component z_i is exactly $S_{\tau\gamma}(x_i)$ from the notation last week, and there are no interaction terms between the z_i and z_j for $i \neq j$, so

$$
\left(\operatorname{prox}_\gamma \phi(x)\right)_i = S_{\tau\gamma}(x_i)
$$

2 The proximal gradient method

(A)

$$
\begin{aligned}
\hat{x} &= \arg\min_x \left\{\tilde{l}(x;x_0) + \phi(x)\right\} \\
&= \arg\min_x \left\{l(x_0) + (x-x_0)^\top \nabla l(x_0) + \frac{1}{2\gamma}\|x-x_0\|_2^2 + \phi(x)\right\} \\
&= \arg\min_z \left\{l(x_0) + (z-x_0)^\top \nabla l(x_0) + \frac{1}{2\gamma}\|z-x_0\|_2^2 + \phi(z)\right\} \\
&= \arg\min_z \left\{\phi(z) + l(x_0) + (z-x_0)^\top \nabla l(x_0) + \frac{1}{2\gamma}\left(z^\top z - 2x_0^\top z + x_0^\top x_0\right)\right\} \\
&= \arg\min_z \left\{\phi(z) + (z-x_0)^\top \nabla l(x_0) + \frac{1}{2\gamma}\left(z^\top z - 2x_0^\top z + x_0^\top x_0\right)\right\} \qquad \text{(add/subtract a constant for same optimization)} \\
&= \arg\min_z \left\{\phi(z) + \frac{\gamma}{2}\left[\nabla l(x_0)\right]^\top \nabla l(x_0) + 2\frac{1}{2\gamma}(z-x_0)^\top \gamma\nabla l(x_0) + \frac{1}{2\gamma}\left(z^\top z - 2x_0^\top z + x_0^\top x_0\right)\right\} \\
&= \arg\min_z \left\{\phi(z) + \frac{1}{2\gamma}\left[\gamma\nabla l(x_0)\right]^\top \gamma\nabla l(x_0) + 2\frac{1}{2\gamma}(z-x_0)^\top \gamma\nabla l(x_0) + \frac{1}{2\gamma}\left(z^\top z - 2x_0^\top z + x_0^\top x_0\right)\right\} \\
&= \arg\min_z \left\{\phi(z) + \frac{1}{2\gamma}\left(\left[\gamma\nabla l(x_0)\right]^\top \gamma\nabla l(x_0) + 2(z-x_0)^\top \gamma\nabla l(x_0) + z^\top z - 2x_0^\top z + x_0^\top x_0\right)\right\} \\
&= \arg\min_z \left[\phi(z) + \frac{1}{2\gamma}\|z - x_0 + \gamma\nabla l(x_0)\|_2^2\right] \\
&= \arg\min_z \left[\phi(z) + \frac{1}{2\gamma}\|z - (x_0 - \gamma\nabla l(x_0))\|_2^2\right] \\
u &= x_0 - \gamma\nabla l(x_0) \\
\hat{x} &= \operatorname{prox}_\gamma \phi(u)
\end{aligned}
$$

(B)

Now, we want to play around with our results to cast the lasso regression into a proximal gradient problem.

$$
\begin{aligned}
\hat{\beta} &= \arg\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \\
l(\beta|X,y) &= \|y - X\beta\|_2^2 = y^\top y - 2y^\top X\beta + \beta^\top X^\top X\beta \\
\hat{\beta} &= \arg\min_{\beta} \left\{ l(\beta|X,y) + \lambda \|\beta\|_1 \right\} \\
l(\beta|X,y) \approx \hat{l}(\beta|X,y;\beta_0) &= l(\beta_0|X,y) + (\beta - \beta_0)^\top \nabla l(\beta_0|X,y) \\
\nabla l(\beta|X,y) &= 0 - 2X^\top y + 2X^\top X\beta \\
\hat{l}(\beta|X,y;\beta_0) &= \|y - X\beta_0\|_2^2 + (\beta - \beta_0)^\top \left(-2X^\top y + 2X^\top X\beta_0 \right)
\end{aligned}
$$

Now, in the linear approximation to $l(\beta|X,y)$, we add in the regularization:

$$
\tilde{l}(\beta|X,y;\beta_0) = \|y - X\beta_0\|_2^2 + (\beta - \beta_0)^\top \left(-2X^\top y + 2X^\top X\beta_0 \right) + \frac{1}{2\gamma} \|\beta - \beta_0\|_2^2
$$

Now, we let $l(\beta|X,y) \approx \tilde{l}(\beta|X,y;\beta_0)$ when β is near β_0. This is now exactly the form of surrogate optimization referenced above so

$$
\begin{aligned}
\phi(\beta) &= \lambda \|\beta\|_1 \\
u^{(t)} &= \beta^{(t)} - \gamma^{(t)} \nabla l(\beta^{(t)}|X,y) = \beta^{(t)} - \gamma^{(t)} \left(2X^\top X\beta^{(t)} - 2X^\top y \right) \\
\beta^{(t+1)} &= \mathrm{prox}_{\gamma^{(t)}} \phi(u^{(t)}) \\
\beta_i^{(t+1)} &= S_{\lambda\gamma^{(t)}} \left(u_i^{(t)} \right) = \mathrm{sign}\left(u_i^{(t)} \right) \left(\left| u_i^{(t)} \right| - \lambda\gamma^{(t)} \right)_+
\end{aligned}
$$

So, to go from step t to step $t+1$, we just compute $u^{(t)}$ then use its components to compute $\beta_i^{(t+1)}$.

There's a relatively high one-time cost to compute $X^\top X$ and $X^\top y$, and (depending on how big p, the number of elements of β, is) this cost carries over each iteration to compute $X^\top X\beta^{(t)}$. That's a $O(p^2)$ calculation (at least in the dense case). Beyond that, the rest of the operations are $O(p)$.