Lecture 23
Alternating Direction Method of Multipliers

09 December 2015

Taylor B. Arnold
Yale Statistics
STAT 312/612

Yale

Class Notes

- Problem Set 7 - Available now, hand in at 24 Hillhouse by 4pm on December 16th

Midterm II

Easy solution to question 1:

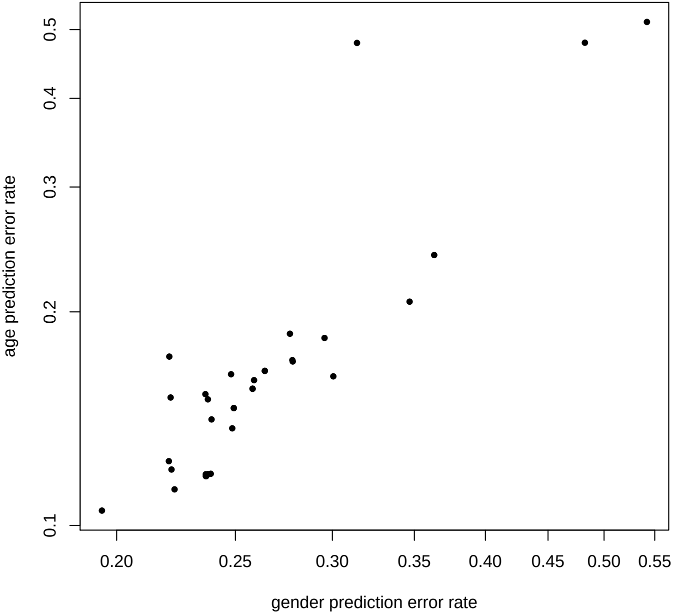
```
> X <- matrix(rnorm(12),nrow=4)
> s <- svd(X)
> svals <- c(1,1,1e-10)
> X <- s$u %*% diag(svals) %*% t(s$v)
> X
           [,1]        [,2]      [,3]
[1,] -0.3425962  0.50634449 0.1048543
[2,]  0.2080486 -0.53566801 0.3398572
[3,] -0.3472912  0.07954308 0.8733534
[4,]  0.2120066 -0.45074100 0.1781087
```

Now notice that a direct solve does not work well:

```
> beta <- c(1,1,1)
> y <- X %*% beta
> solve(t(X) %*% X, t(X) %*% y, tol=0)
            [,1]
[1,] -0.62017237
[2,]  0.01868727
[3,]  0.44511010
```

But the pseudo inverse does:

```
> pseudo <- s$v %*% diag(1/svals) %*% t(s$u)
> pseudo %*% y
         [,1]
[1,] 1.000001
[2,] 1.000001
[3,] 1.000000
```


0.5
0.4
0.3
0.2
0.1
age prediction error rate
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
gender prediction error rate

ADMM

Consider the following minimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

Consider the following minimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

The Lagrangian function is defined as:

$$L(x, y) = f(x) + y^t(Ax - b)$$

As described last time, the dual function is defined as:

$$g(y) = \inf_x L(x, y)$$

And the corresponding dual problem is:

$$y^* = \arg\max_y g(y)$$

Which yields the solution x^*:

$$x^* = \arg\min_x L(x, y^*)$$

With most of the work occurring in solving the dual problem.

If we have an analytic form of the function g and its gradient, gradient ascent can be used to repeatedly update y until convergence by:

$$y^{k+1} = y^k + \alpha \cdot \nabla g(y^k)$$

If we have an analytic form of the function g and its gradient, gradient ascent can be used to repeatedly update y until convergence by:

$$y^{k+1} = y^k + \alpha \cdot \nabla g(y^k)$$

Likewise, dual ascent is given by:

$$\begin{aligned} x^{k+1} &= \arg\min_x L(x, y^k) \\ y^{k+1} &= y^k + \alpha \cdot (Ax^{k+1} - b) \end{aligned}$$

Which converges to the correct solution under strong assumptions.

Now, consider if f can be separated as follows:

$$f(x) = \sum_i f_i(x_i)$$

The Lagrangian can the be written as:

$$\begin{aligned} L(x, y) &= \sum_i L_i(x_i, y) \\ &= \sum_i f_i(x) + y^t A_i x_i - y^t b \end{aligned}$$

And therefore the x-step in dual ascent can be parallelized over each x_i.

Specifically:

$$x_i^{k+1} = \arg\min_{x_i} L_i(x_i, y^k)$$
$$y^{k+1} = y^k + \alpha \cdot (Ax^{k+1} - b)$$

Which again, converges to the correct solution under strong assumptions.

Now, consider the augmented Lagrangian:

$$L_\rho(x, y) = f(x) + y^t(Ax - b) + (\rho/2)\,||Ax - b||_2^2$$

For some value of $\rho > 0$. This makes dual ascent far more robust, and yields the following updates:

$$x^{k+1} = \arg\min_x L_\rho(x, y^k)$$
$$y^{k+1} = y^k + \rho(Ax^{k+1} - b)$$

Notice that the α in the y-step has been replaced by the ρ in the augmented Lagrangian. This is called the **method of multipliers**.

The method of multipliers will converge under much more relaxed conditions, but we have a squared norm of the penalty this will no longer allow for splitting the optimization problem across $f_i(x_i)$'s.

The method **alternating direction method of multipliers**, or ADMM, combines the splitting capability of dual ascent with the robustness of the method of multipliers. It solves the optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

With the augmented Lagrangian:

$$L_\rho(x, z, y) = f(x) + g(z) + y^t(Ax + Bz - c) + (\rho/2)\,||Ax + Bz - c||_2^2$$

To solve this, we add three types of updates:

$$
\begin{aligned}
x^{k+1} &= \arg\min_x L_\rho(x, z^k, y^k) \\
z^{k+1} &= \arg\min_z L_\rho(x^{k+1}, z, y^k) \\
y^{k+1} &= y^k + \rho \cdot (Ax^{k+1} + Bz^{k+1} - c)
\end{aligned}
$$

Where solving the first two steps simultaneously would yield the same solution as the method of multipliers. The *alternating* in the name refers to alternating between x-updates and z-updates.

If we replace $u_k = (1/\rho)y^k$, this allows combining the linear and quadratic terms in the augmented Lagrangian. The update now can be written explicitly:

$$
\begin{aligned}
x^{k+1} &= \arg\min_x \left\{ f(x) + (\rho/2)||Ax + Bz^k - c + u^k||_2^2 \right\} \\
z^{k+1} &= \arg\min_z \left\{ f(x) + (\rho/2)||Ax^{k+1} + Bz - c + u^k||_2^2 \right\} \\
u^{k+1} &= u^k + (Ax^{k+1} + Bz^{k+1} - c)
\end{aligned}
$$

Assume f and g are convex, closed, and proper and L_0 has a saddle point. Then ADMM converges in both feasibility (does the constraint hold) and optimality (is the function to be optimized near its optimal value).

Consider now the lasso problem. We'll write it here in 'numerical analysis' notation (so A is the data matrix, x in the unknown parameter, and b is the response):

$$\arg\min_{x} (1/2)||Ax - b||_2^2 + \lambda||x||_1$$

As we did last class, we'll write this as a a constrained problem:

$$\begin{array}{ll} \text{minimize} & (1/2)||Ax - b||_2^2 + \lambda||z||_1 \\ \text{subject to} & x - z = 0 \end{array}$$

Where we have separated the ℓ_2-loss as a function of x and the ℓ_1-penalty as a function of z.

So the x-step in ADMM amounts to finding the minimum over x of the following quantity:

$$f(x) + (\rho/2)||Ax + Bz - c + u||_2^2 = ||Ax - b||_2^2 + (\rho/2)||x - z + u||_2^2$$

So the x-step in ADMM amounts to finding the minimum over x of the following quantity:

$f(x) + (\rho/2)||Ax + Bz - c + u||_2^2 = ||Ax - b||_2^2 + (\rho/2)||x - z + u||_2^2$

If we translate to $w = x - z + u$ this is just ridge regression on w, which we have an analytic formula for. Transforming the variables we get specifically:

$$x^{k+1} = (A^t A + \rho I)^{-1}(A^t b + \rho z^k - u^k)$$

Now, the z-step in ADMM amounts to finding the minimum over x

$$g(z) + (\rho/2)||Ax + Bz - c + u||_2^2 = \lambda||z||_1 + (\rho/2)||x - z + u||_2^2$$

And dividing by ρ gives:

$$\arg\min_z \left\{(1/2)||(x + u) - z||_2^2 + \lambda/\rho||z||_1\right\}$$

Which is just the lasso without an X matrix and on the response $x + u$.

From lecture 17, we have an analytic formula for the simple lasso case where the variables are uncorrelated. Here we have an even more easy formula because X is the identity itself.

From lecture 17, we have an analytic formula for the simple lasso case where the variables are uncorrelated. Here we have an even more easy formula because X is the identity itself.

Specifically, we have the following formula for the lasso regression

$$z_i^{k+1} = \begin{cases} x_i + u_i - \lambda/\rho & (x_i + u_i) \geq \lambda/\rho \\ x_i + u_i + \lambda/\rho & (x_i + u_i) \leq -\lambda/\rho \\ 0 & \text{else} \end{cases}$$

Called soft-thresholding and denoted by (for the penalty λ/ρ):

$$z^{k+1} = S_{\lambda/\rho}(x_i + u_i)$$

So the full ADMM lasso update is given by:

$$
\begin{aligned}
x^{k+1} &= (A^tA + \rho I)^{-1}(A^tb + \rho z^k - y^k) \\
z_i^{k+1} &= S_{\lambda/\rho}(x_i + u_i) \\
u^{k+1} &= u^k + \rho(x^{k+1} - z^{k+1})
\end{aligned}
$$

Or, in other words, we iteratively do ridge regression followed by soft-thresholding.

So the full ADMM lasso update is given by:

$$
\begin{aligned}
x^{k+1} &= (A^t A + \rho I)^{-1}(A^t b + \rho z^k - y^k) \\
z_i^{k+1} &= S_{\lambda/\rho}(x_i + u_i) \\
u^{k+1} &= u^k + \rho(x^{k+1} - z^{k+1})
\end{aligned}
$$

Or, in other words, we iteratively do ridge regression followed by soft-thresholding.

The hard computational part is take the SVD of A, which only needs to be done once, in order to get the first x-update. There are many ways of doing this in parallel, particularly when $n > p$. Otherwise, all of these ADMM steps can be solved locally on the data. This allows for massive parallelization gains.

Closing thoughts / summary

In the first lecture I said:

> *Linear Models is both a capstone to the 241/242 sequence and the breadth to compliment 610's depth. It also serves as a link between the statistical inference courses and the applied data analysis courses.*
>
> *Topics will be oriented around linear models (obviously) but the course is somewhat of a hodgepodge of topics and applications.*

I think this has been generally true; hopefully it has been both a useful and interesting hodgepodge of topics.

Techniques

1. classical linear regression
2. weighted least squares
3. hierarchical linear models
4. logistic regression
5. singular value decomposition
6. ridge regression
7. principal components
8. lasso regression
9. elastic net
10. generalized lasso regression

Applications

1. historical data from 1890's (Galton's experiments)
2. larger datasets (airline)
3. unstructured dataset (image corpus, text corpus)

Numerical algorithms

1. Cholesky w/ backsolve and forwardsolve for LS problem
2. QR of X trick for LS
3. pseudoinverse trick for LS
4. Newton-Raphson, iteratively reweighted least squares (glm)
5. LARs, homotopy path solution for solving lasso
6. coordinate descent (elastic net, problem set 7)
7. Lagrangian dual problem (generalized lasso)
8. alternating direction method of multipliers (today, lasso)

If you enjoyed this, consider taking **Data Mining and Machine Learning (STAT 365/665)** with me in the Spring. It will have an even heavier focus on applications with considerably less theory.