SDS 385: Exercises 1 - Preliminaries

August 23, 2016

Professor James Scott

Spencer Woody

Problem 1

(A)

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{N} \frac{w_i}{2} \left(y_i - x_i^T \beta \right)^2 \tag{1}$$
$$= \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} (Y - X\beta)^T W (Y - X\beta) \tag{2}$$

$$\frac{1}{2}(Y - X\beta)^T W (Y - X\beta) = \frac{1}{2}(Y^T - \beta^T X^T) W (Y - X\beta) \tag{3}$$
$$= \frac{1}{2}(Y^T W - \beta^T X^T W)(Y - X\beta) \tag{4}$$
$$= \frac{1}{2}(Y^T W Y - \beta^T X^T W Y - Y^T W X\beta + \beta^T X^T W X \beta) \tag{5}$$
$$= \frac{1}{2}(Y^T W Y - 2(X\beta)^T W Y + \beta^T X^T W X \beta) \tag{6}$$
$$= \frac{1}{2} Y^T W Y - (X\beta)^T W Y + \frac{1}{2} \beta^T X^T W X \beta, \tag{7}$$

because

$$\beta^T X^T W Y = (X\beta)^T W Y, \tag{8}$$

and

$$Y^T W X \beta = (Y^T W X \beta)^T \because Y^T W X \beta \in \mathbb{R}^1 \tag{9}$$
$$(Y^T W X \beta)^T = (W X \beta)^T Y = (X\beta)^T W^T Y = (X\beta)^T W Y. \tag{10}$$

We want to minimize the objective function from Eqn. (7), so we take the gradient with respect to β and set it equal to zero. For each of the three terms, their are respective gradients with respect to β are

(i)

$$\frac{\partial}{\partial \beta} \frac{1}{2} Y^T W Y = 0 \tag{11}$$

(ii)

$$\frac{\partial}{\partial \beta} - (X\beta)^T W Y = -X^T W Y \tag{12}$$

(iii)

$$\frac{\partial}{\partial \beta} \frac{1}{2} \beta^T X^T W X \beta = \frac{1}{2} \beta^T (X^T W X + (X^T W X)^T) \tag{13}$$
$$= X^T W X \beta. \tag{14}$$

Summing these terms and equaling them to zero yields

$$X^T W X \beta - X^T W Y = 0 \therefore \tag{15}$$
$$(X^T W X)\hat{\beta} = X^T W Y \tag{16}$$

(B) The brute force method of solving Eqn. (16) is the *inversion method*, i.e.

$$\hat{\beta} = (X^T W X)^{-1} X^T W y. \tag{17}$$

However, this method is computationally expensive. Therefore I propose two alternative methods to solving this matrix equation using matrix decomposition, the LU decomposition and the Cholesky decomposition.

(i) **LU Decomposition**
Let

$$D = X^T W X, \;\; C = X^T W y \tag{18}$$

so

$$C\hat{\beta} = D. \tag{19}$$

We decompose matrix C into a product of a lower-triangular matrix and an upper-triangular matrix, so that

$$C = LU \therefore \tag{20}$$

$$LU\hat{\beta} = D. \tag{21}$$

Furthermore we define matrix $A = U\hat{\beta}$. Thus we are left with two matrix equations to solve.

$$LA = D \tag{22}$$

$$U\hat{\beta} = A \tag{23}$$

This method will be much less computationally intensive than the inversion method because of the fact that the two left-matrices L and U are triangular.

(ii) **Cholesky Decomposition**
We use a similar method as the LU decomposition, but this time we decompose matrix C into the form the matrix product LL^T. So our method is exactly the same as above, except now $U = L^T$. However, we gain a computational advantage from symmetric exploitation.

(C)

(D)

Problem 2

(A)

$$
\begin{aligned}
\ell(\beta) &= -\log\left\{\prod_{i=1}^{N} p(y_i|\beta)\right\} && (24)\\
&= -\log\left\{\prod_{i=1}^{N} (w_i)^{y_i}(1-w_i)^{1-y_i}\right\} && (25)\\
&= -\log\left\{\prod_{i=1}^{N} \left(\frac{1}{1+\exp(-x_i^T\beta)}\right)^{y_i}\left(1-\frac{1}{1+\exp(-x_i^T\beta)}\right)^{1-y_i}\right\} && (26)\\
&= -\log\left\{\prod_{i=1}^{N} \left(\frac{1}{1+\exp(-x_i^T\beta)}\right)^{y_i}\left(\frac{\exp(-x_i^T\beta)}{1+\exp(-x_i^T\beta)}\right)^{1-y_i}\right\} && (27)\\
&= -\left\{\sum_{i=1}^{N} \left[(y_i)(-\log(1+\exp(-x_i^T\beta))) + (1-y_i)(-x_i^T\beta - \log(1+\exp(-x_i^T\beta)))\right]\right\} && (28)\\
&= -\left\{\sum_{i=1}^{N} [-x_i^T\beta - \log(1+\exp(-x_i^T)) + y_i x_i^T\beta]\right\} && (29)\\
&= \sum_{i=1}^{N} (x_i^T\beta + \log(1+\exp(-x_i^T\beta)) - y_i x_i^T\beta) && (30)
\end{aligned}
$$

(B)

(C)

(D)

(E)

```
###########################################################
######### Created by Spencer Woody on 23 Aug 2016 #########
###########################################################

library(Matrix)
library(microbenchmark)

### No. 1 pt B

# Set N, P, X, W, and y

N <- 2000
P <-  500

X <- matrix(rnorm(N * P), nrow = N)
W <- diag(rep(1, N))
y <- matrix(rnorm(N), nrow = N)

# Inversion method

bhat.Inv <- solve(t(X) %*% W %*% X) %*% t(X) %*% W %*% y


# LU decomp method

D <- t(X) %*% W %*% y
C <- t(X) %*% W %*% X

LUdecomp <- lu(C)
L <- expand(LUdecomp)$L
U <- expand(LUdecomp)$U

A <- solve(L, D)
bhat.LU <- solve(U, A)

# Cholesky decomp method

Cho <- chol(C)

L.Cho <- t(Cho)
U.Cho <- Cho

A.Cho <- solve(L.Cho, D)
bhat.Cho <- solve(U.Cho, A.Cho)



### No. 1 pt C

### No. 1 pt D

# N <- 2000
# P <-  500
```

```
#
# X <- matrix(rnorm(N * P), nrow = N)
# mask <- matrix(rbinom(N * P, 1, 0.05), nrow = N)
# X <- mask * X

# END
```