SDS385 HW 1

Evan Ott
UT EID: eao466

August 30, 2016

Linear Regression

(A)

WLS objective function:

$$\begin{aligned}
\sum_{i=1}^{N} \frac{w_i}{2}(y_i - x_i^\top \beta)^2 &= \frac{1}{2}\sum_{i=1}^{N} y_i w_i y_i - \sum_{i=1}^{N} y_i w_i x_i^\top \beta + \frac{1}{2}\sum_{i=1}^{N} x_i^\top \beta w_i x_i^\top \beta \\
&= \frac{1}{2} y^\top W y - y^\top W X \beta + \frac{1}{2}(X\beta)^\top W X \beta \\
&= \frac{1}{2}(y - X\beta)^\top W (y - X\beta).
\end{aligned}$$

Minimizing this function means setting the gradient (with respect to β) to zero:

$$\nabla_\beta \left[\frac{1}{2}(y - X\beta)^\top W (y - X\beta)\right] = 0$$

That is

$$\begin{aligned}
\nabla_\beta \left[\frac{1}{2}(y - X\beta)^\top W (y - X\beta)\right] &= 0 - (y^\top W X)^\top + \frac{2}{2} X^\top W X \hat{\beta} = 0 \\
&\Rightarrow (X^\top W X)\hat{\beta} = X^\top W y \qquad \blacksquare
\end{aligned}$$

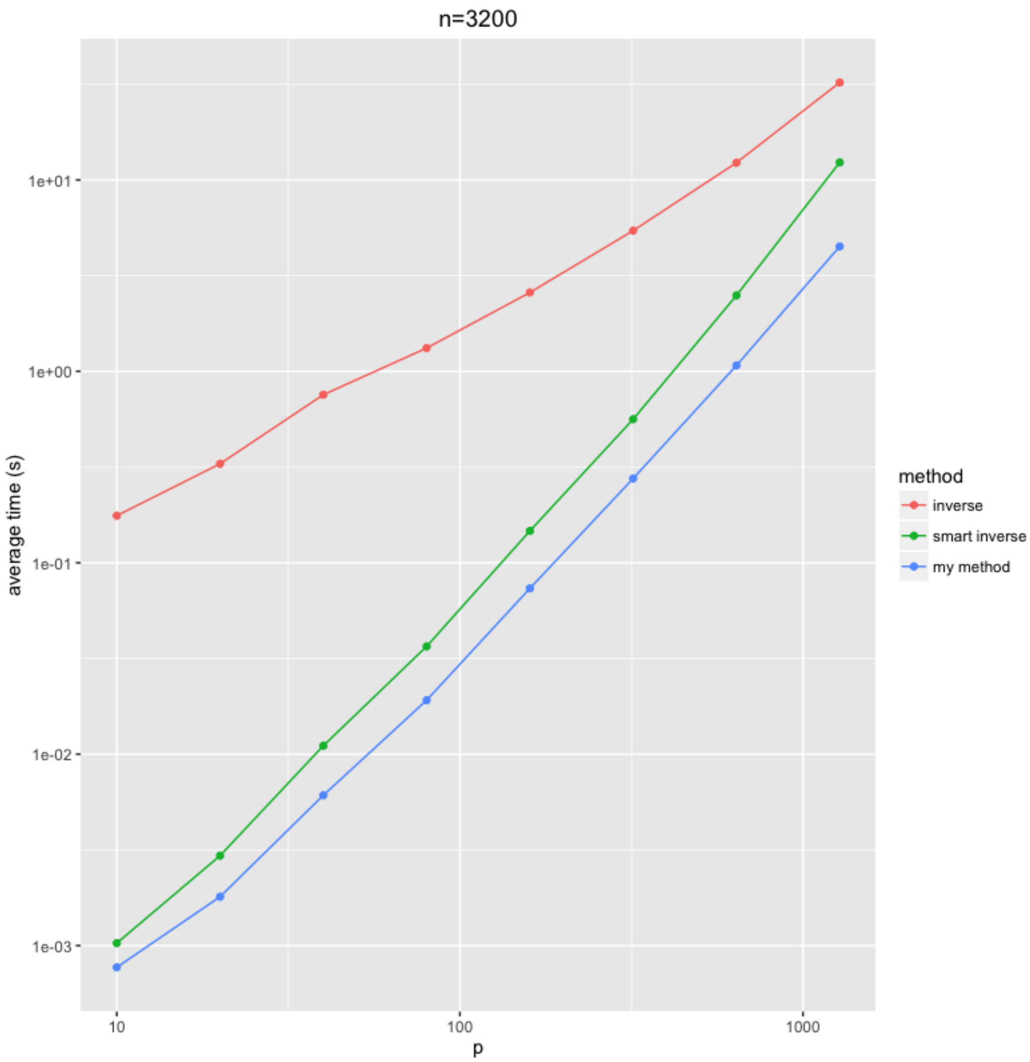
(B)

The matrix factorization idea basically amounts to trying to prevent the full inverse operation. Overall, you will still probably need something $O(n^3)$, but the constants matter when actually doing computation as opposed to asymptotics. We don't actually want the inverse anyway, we just want to solve $(X^\top W X)\hat{\beta} = X^\top W y$ for β. Using a matrix decomposition can help with that a lot.

I based my solution on the Cholesky decomposition (see `http://www.seas.ucla.edu/~vandenbe/103/lectures/chol.pdf`). This creates matrices $X = LL^\dagger$, where L is lower-triangular. So, then solve $Lz = X^\top W y$ for z and $R\beta = z$ for β. The decomposition is still $O(n^3)$, but faster than inverse. The two final steps are each $O(n^2)$ which are faster.

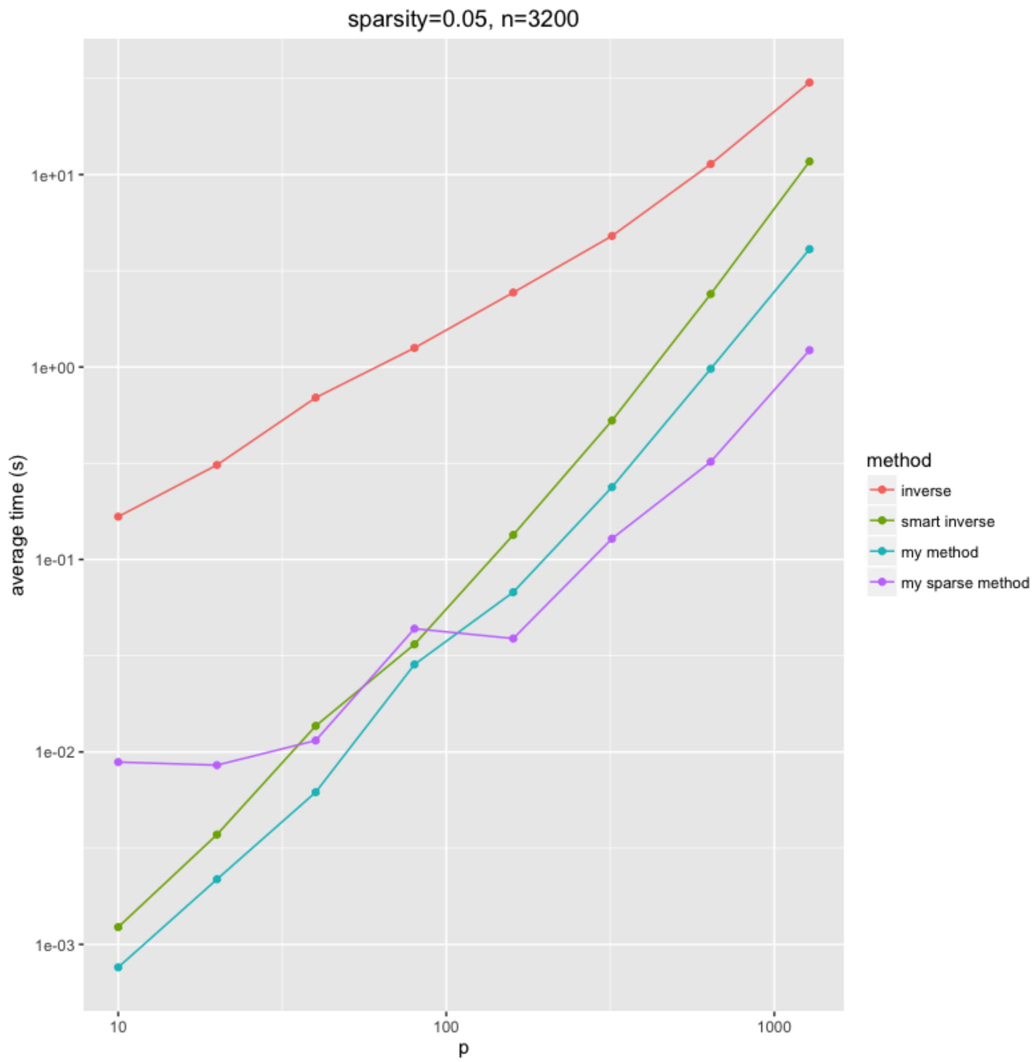
(C)

See code on GitHub (`hw1.R`).



(D)

See code on GitHub (`hw1.R`).



Notes from class

Three main matrix decomposition techniques:

1. Cholesky $\to$ fast, unstable (susceptible to roundoff error)
2. QR $\to$ middle ground
3. SVD $\to$ slow, but works for close-to-rank-deficient matrices

Using QR, we get $W^{1/2}X = QR$, where R is $P \times P$ and upper-triangular (and thus invertible) and Q is $N \times P$ with orthonormal columns.

$$
\begin{aligned}
X^\top W y &= X^\top W X \beta \\
X^\top W^{1/2} W^{1/2} y &= X^\top W^{1/2} W^{1/2} X \beta \\
(QR)^\top W^{1/2} y &= (QR)^\top QR\beta \\
Q^\top W^{1/2} y &= IR\beta = R\beta
\end{aligned}
$$

A note on R, `crossprod` computes $X^\top X$ but recognizes the symmetry so it takes half the time.

Generalized linear models

(A)

$$
\begin{aligned}
w_i(\beta) &= \frac{1}{1+\exp\{-x_i^\top \beta\}} \\
l(\beta) &= -\log\left\{\prod_{i=1}^{N} p(y_i|\beta)\right\} \\
&= -\log\left\{\prod_{i=1}^{N} \binom{m_i}{y_i} w_i(\beta)^{y_i}(1-w_i(\beta))^{m_i-y_i}\right\} \\
&= -\sum_{i=1}^{N} \log\left\{\binom{m_i}{y_i} w_i(\beta)^{y_i}(1-w_i(\beta))^{m_i-y_i}\right\} \\
&= -\sum_{i=1}^{N} \log\binom{m_i}{y_i} + y_i\log(w_i(\beta)) + (m_i-y_i)\log(1-w_i(\beta)) \\
\nabla l(\beta) &= -\sum_{i=1}^{N} 0 + \frac{y_i}{w_i(\beta)}\nabla w_i(\beta) - \frac{m_i-y_i}{1-w_i(\beta)}\nabla w_i(\beta) \\
\nabla w_i(\beta) &= w_i^2(\beta)e^{-x_i^\top\beta}x_i \\
\nabla l(\beta) &= -\sum_{i=1}^{N} w_i^2(\beta)e^{-x_i^\top\beta}x_i\left(\frac{y_i}{w_i(\beta)} - \frac{m_i-y_i}{1-w_i(\beta)}\right) \\
&= -\sum_{i=1}^{N} w_i^2(\beta)e^{-x_i^\top\beta}x_i\left(\frac{y_i - y_iw_i(\beta) - m_iw_i(\beta) + y_iw_i(\beta)}{w_i(\beta)(1-w_i(\beta))}\right) \\
&= -\sum_{i=1}^{N} w_i(\beta)e^{-x_i^\top\beta}x_i\left(\frac{y_i - m_iw_i(\beta)}{1-w_i(\beta)}\right) \\
&= -\sum_{i=1}^{N} w_i(\beta)\left(\frac{1}{w_i(\beta)}-1\right)x_i\left(\frac{y_i - m_iw_i(\beta)}{1-w_i(\beta)}\right) \\
&= -\sum_{i=1}^{N} w_i(\beta)\frac{1-w_i(\beta)}{w_i(\beta)}x_i\left(\frac{y_i - m_iw_i(\beta)}{1-w_i(\beta)}\right) \\
&= -\sum_{i=1}^{N} \left[y_i - m_iw_i(\beta)\right]x_i
\end{aligned}
$$

(B)

See code on GitHub (`glm.R`).

(C)

(D)

Newton's direction is $-\left(\nabla^2 l(\beta)\right)^{-1}\nabla l(\beta)$.