Math 426/CISC 410 08F, All Sections

R.J. Braun

Homework 7 Solutions, Hints and Answers

Problem 3.2.1. Here we use the normal equation approach.

$$A^TA = \begin{bmatrix} 8 & -6 \\ -6 & 6 \end{bmatrix}, \quad A^Tb = \begin{bmatrix} -10 \\ 6 \end{bmatrix}.$$

Solving $A^TAx = A^Tb$ gives $x = [-2 \ -1]^T$.

3.2.2. We can compute all the quantities we need to get the pseudoinverse, even if there is only one column in A. Here $A^TA = 1^2 + (-2)^2 + 3^2 = 14$. Then

$$A^+ = (A^TA)^{-1}A^T = \frac{1}{14}\,[1 \ -2 \ 3]\,.$$

3.2.9. Let P and Q be orthogonal matrices; then for either of them, the inverse is the same as the transpose. If the product PQ is orthogonal then $(PQ)^T(PQ) = (PQ)(PQ)^T = I$. Let's try it.

$$(PQ)^T(PQ) = Q^TP^TPQ = Q^TIQ = Q^TQ = I;$$

we can do this if both P and Q are orthogonal. Also

$$(PQ)(PQ)^T = PQQ^TP^T = PIP^T = PP^T = I.$$

Because the order of the products doesn't matter and we get the identity matrix as a result, the product PQ of orthogonal matrices is still an orthogonal matrix.

3.2.10. (a) For an orthogonal matrix, $Q^{-1} = Q^T$; that is, $QQ^T = Q^TQ = I$. Let $P = Q^T$, and now try the same approach as in the last problem.

$$PP^T = Q^T(Q^T)^T = Q^TQ = I, \text{ and } P^TP = (Q^T)^TQ^T = QQ^T = I;$$

thus, Q^T is also an orthogonal matrix.

(b) Now $\kappa_2(Q) = ||Q^{-1}||_2||Q||_2 = ||Q^T||_2||Q||_2$. Recall that $||Q|| = \max_{||x||_2=1}||Qx||_2$ is the definition of the 2-norm of the matrix. Then for an orthogonal matrix,

$$||Qx||_2^2 = (Qx)^T(Qx) = x^TQ^TQx = x^TIx = x^Tx = ||x||_2^2.$$

Because in the definition of the norm we must have $||x||_2 = 1$, then $||Q||_2 = 1$. Now $||Q^T||_2 = 1$ because Q^T is also an orthogonal matrix. Then $\kappa_2(Q) = (1)(1) = 1$.

3.3.4. The QR factorization can be written as follows.

$$A = QR = \begin{bmatrix}\hat{Q} \ Q_0\end{bmatrix}\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}.$$

Note that A is m-by-n, $\hat{Q}$ is m-by-n and $\hat{R}_0$ is n-by-n; the others fill out the dimensions with Q_0 being m-by-$(m-n)$ and the zero matrix is $(m-n)$-by-n. Because of that zero matrix, we can reconstruct A with the compressed QR facorization: $A = \hat{Q}\hat{R}$. Now use this in the formula for the pseudoinverse.

$$\begin{aligned} A^+ &= (A^T A)^{-1} A^T \\ &= [(\hat{Q}\hat{R})^T \hat{Q}\hat{R}]^{-1} [\hat{Q}\hat{R}]^T, \\ &= [\hat{R}^T \hat{Q}^T \hat{Q}\hat{R}]^{-1} \hat{R}^T \hat{Q}^T, \\ &= [\hat{R}^T \hat{R}]^{-1} \hat{R}^T \hat{Q}^T, \\ &= \hat{R}^{-1} [\hat{R}^T]^{-1} \hat{R}^T \hat{Q}^T, \\ &= \hat{R}^{-1} I \hat{Q}^T = \hat{R}^{-1} \hat{Q}^T. \end{aligned}$$

3.3.6. (b) Use the definition of the norm.

$$\begin{aligned} ||A||_2 &= \max_{||x||_2=1} ||Ax||_2 \\ &= \max_{||x||_2=1} ||QRx||_2 \\ &= \max_{||x||_2=1} ||Q(Rx)||_2 \\ &= \max_{||x||_2=1} ||Rx||_2 \\ &= ||R||_2. \end{aligned}$$

(a) Count the number of flops in the `qrfact` function; use $m = n$. Note that there are no flops in lines 9 to 13, but there is important information there: the sizes of everything are established. For example, `v` and `z` are $n - k + 1$ long. Let $s = n - k + 1$ for convenience.

- **Line 14:** There are 3 flops before the semicolon and there are $s - 1$ flops (multiplication by -1 after it, for a total of $s - 1 + 3 = s + 2$.
- **Line 15:** This line is non-trivial to count! Start with the parentheses. The last term is `v'*v`, which is a dot product; the result is a scalar that takes $2s-1$ flops. The other term in parentheses is `v*v'`, which is an outer product giving an s-by-s matrix as output; this matrix takes s^2 flops. (You have to write out the matrix to see this.) Now the division sign takes s^2 flops, as does the multiplication by 2, and the substraction from the s-by-s identity matrix. Thus the total flops count is $4s^2 + 2s - 1$ for this line.
- **Line 16:** This is multiplication of two matrices of size s-by-s; there are s^2 locations and each one takes a dot product requiring $2s - 1$ flops, so there are $s^2(2s - 1)$ flops required here.
- **Line 17:** Same as line 16, so $s^2(2s - 1)$ flops here too.

That's it for the flops. Now we need to total them up, for each k we have

$$(s + 2) + (4s^2 + 2s - 1) + [s^2(2s - 1)] + [s^2(2s - 1)] = 4s^3 - 2s^2 + 3s + 1, \;\; s = n - k + 1.$$

Now summing over the loop gives

$$\begin{aligned}\sum_{k=1}^{n} 4s^3 - 2s^2 + 3s - 3 &= \sum_{k=1}^{n} 4(n-k+1)^3 - 2(n-k+1)^2 + 3(n-k+1) + 1, \\ &= \sum_{k=1}^{n} 4n^3 - 4k^3 + O(n^2) + O(k^2), \\ &\sim 4\left(\sum_{k=1}^{n} n^3 - \sum_{k=1}^{n} k^3\right), \\ &\sim 4(n^4 - n^4/4), \\ &= 3n^4.\end{aligned}$$

Thus, for `qrfact.m`, the asymptotic flop count is $3m^4$. This is an inefficient implementation; the theoretical flop count is $(4/3)n^3$, which is twice the flop count of LU factorization. In `qrfact.m`, the matrix multiplication in lines 16 and 17 is what really drives up the flop count. We would avoid this in an efficient implementation by not explicitly forming `P`; we would only code formulas that have the same effect as making `P` and doing the multiplication.

3.4.2. Consider one step needed in using the normal equations approach.

$$A^T A = \begin{bmatrix} 1 + 10^{-18} & 1 \\ 1 & 1 + 10^{-18} \end{bmatrix}.$$

In the Matlab, the unit round is `eps`$\approx 2 \times 10^{-16}$, and so in the computer the first and last elements in $A^T A$ will become $1 + 10^{-18} \to 1$. Thus,

$$A^T A \approx \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \widetilde{A^T A},$$

where the tilde indicates the computer result. The computed matrix is singular, and the problem can't be solved this way in Matlab. More generally, the normal equations are prone to becoming ill-conditioned; this is an extreme example.