Linear Systems of Eq.'s

Part 6: Stability & Conditioning of Least-Squares

Recap: $A = QR$

1. Triangular Orthogonalization (Gram-Schmidt)

$$\underbrace{\begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix}}_{A} \underbrace{R_1 R_2 \cdots R_n}_{\substack{= R^{-1} \\ \text{(upper tri.)}}} = \underbrace{\begin{bmatrix} | & & | \\ q_1 & \cdots & q_n \\ | & & | \end{bmatrix}}_{Q}$$

$$q_k = \frac{1}{r_{kk}} \left[a_j - (q_1^* a_j) q_1 - \cdots - (q_{k-1}^* a_j) q_{k-1} \right]$$

2. Orthogonal triangularization (Householder)

$$Q_n \cdots Q_2 Q_1 \underbrace{\begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \diagdown & \\ & \end{bmatrix}}_{R}$$

$$Q_k = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 \end{matrix}}_{\substack{(k-1)\times(k-1) \\ \text{identity}}} & \\ & F_k \end{bmatrix} \quad \text{where} \quad F_k = I - 2\frac{x_k x_k^*}{x_k^* x_k}$$

Householder reflection

"reflectors" = vectors that generate F_k

Householder QR factorization (pseudocode)

for $k = 1$ to n

$\quad v = A_{k:m,k}$

$\quad x_k = \text{sign}(v_1)\|v\|_2 e_1 + v$

$\quad x_k = x_k / \|x_k\|$

$\quad A_{k:m,k:n} = A_{k:m,k:n} - 2x_k(x_k^* A_{k:m,k:n})$

end

<u>Cost:</u> $A_{k:m,j} - 2x_k(\underbrace{x_k^* A_{k:m,j}}_{\approx 2\ell})$ for $j=k$ to n, $k=1$ to n

flops $\quad \ell \quad \ell$

where $\ell = m-k+1$ is length of vectors.

for each j,k, # flops is $\approx 4(m-k+1)$

$$\# \text{flops} = \sum_{k=1}^{n} \underbrace{\sum_{j=k}^{n} 4(m-k+1)}_{=4(m-k+1)(n-k+1)}$$

$$= 4m\underbrace{\sum_{k=1}^{n} n-k+1}_{\sim \frac{1}{2}mn^2} - 4\underbrace{\sum_{k=1}^{n}(1-k)(n-k+1)}_{\sim \frac{1}{6}n^3}$$

Householder: # flops $\sim 2mn^2 - \frac{2}{3}n^3$

Compare to Gram-Schmidt: # flops $\sim 2mn^2$

For a geometric perspective on flop counting see Lectures 8 and 10, Trefethen.

Stability

A key issue with CGS & MGS stability is loss of orthogonality (see Lecture 9 notes).

MGS: $\|\tilde{Q}^*\tilde{Q} - I\| \leq C_m \left[\frac{\sigma_1}{\sigma_n}\right] \varepsilon_{mach}$

singular vals of A

CGS: $\|\tilde{Q}^*\tilde{Q} - I\| \approx \tilde{C}_m \left[\frac{\sigma_1}{\sigma_n}\right]^2 \varepsilon_{mach}$

computed $\tilde{Q}$ need not even be close to an orthogonal matrix, so $\tilde{Q}\tilde{R}$ is not a QR factorization of any matrix A.

Householder does not suffer loss of orthogonality b/c $\tilde{Q}$ is computed implicitly.

$\tilde{Q}^* = \tilde{Q}_n \cdots \tilde{Q}_1$, where

$$\tilde{Q}_k = \begin{bmatrix} \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \cdots & & 1 \end{matrix} & \\ & \tilde{F}_k \end{bmatrix} \quad \text{where} \quad \tilde{F}_k = I - 2\frac{\tilde{x}_k \tilde{x}_k^*}{\tilde{x}_k^* \tilde{x}_k}$$

(k-1)x(k-1) identity

Householder reflection w.r.t. computed $\tilde{x}_k$

Entries of Q_k are not computed or stored explicitly - only the vectors x_k that generate the Householder reflections are. Even in floating point with x_k perturbed to $\tilde{x}_k$, $\tilde{Q}_k$ is still an orthogonal transformation!

To apply $\tilde{Q}$, note that $\tilde{Q}_k^* = \tilde{Q}_k$ $(k=1,\dots,n)$ so that

$$\tilde{Q}^* b = \tilde{Q}_n \cdots \tilde{Q}_2 \tilde{Q}_1 b \iff \tilde{Q} b = \tilde{Q}_1 \tilde{Q}_2 \cdots \tilde{Q}_n b$$

With this implicit computation of $\tilde{Q}$, which preserves orthogonality, we have backward stability.

<u>Thm</u> I Let $A = QR$ be computed on by Householder triangularization in floating point arithmetic. The computed factors satisfy

$$\tilde{Q}\tilde{R} = A + \delta A, \quad \|\delta A\| = \|A\| \, \mathcal{O}(\varepsilon_{mach})$$

i.e. the algorithm is backward stable.

Sketch of Pf Key pt is that applying a sequence of orthogonal matrices to A is backward stable: homework problem 16.1 (Trefethen).

$$\Rightarrow \tilde{Q}_n^* \cdots \tilde{Q}_1^* (A + \delta A) = \tilde{R} \iff A + \delta A = \tilde{Q}\tilde{R}$$

Least-Squares via Householder

A backward stable algorithm for computing $A = QR$ is a powerful building block for other backward stable algorithms.

Problem

$$\text{Solve} \quad x_* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \|Ax - b\|_2$$

Least-Squares via "Thin" and "Full" QR

Gram-Schmidt

$$\begin{array}{cc} & n \\ m & \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \end{array} = \begin{bmatrix} | & & | \\ q_1 & \cdots & q_n \\ | & & | \end{bmatrix} \begin{bmatrix} \diagdown & \\ & \diagdown \end{bmatrix}$$

$A \qquad Q' \qquad R'$

"Thin" QR

Householder

$$\begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} = \left[\begin{array}{ccc:c} | & & | & | \\ q_1 & \cdots & q_n & \cdots q_m \\ | & & | & | \end{array}\right] \left[\begin{array}{cc} \diagdown & \\ & \diagdown \\ \hdashline & \end{array}\right]$$

$A \qquad Q \qquad R$

"Full" QR

Least squares soln solves $\quad Ax_* = P_A b$

$\uparrow$ orthogonal projection onto range(A)

"thin" $\Rightarrow \quad R' x_* = Q'^* b$

"full" $\Rightarrow \quad R_{1:n,1:n}\, x_* = [Q^* b]_{1:n}$

drop last m-n rows corresponding to zero rows of R and columns $q_{n+1}, \dots, q_m$.

Mathematically equivalent, but "full" formulation allows us to apply Q^* to b w/out forming columns $q_1, \dots, q_n$ explicitly - we apply Q^* using Householder reflectors instead.

Algorithm

1) Compute $A = QR$ (Householder)
2) Compute $d = Q^* b$ $(Q_n \cdots Q_2 Q_1 b)$
3) Solve $R_{1:n,1:n} x_* = d_{1:n}$ (back sub.)

Steps 1)-3) are each backward stable:

(*) $(A + \delta A) = \tilde{Q}\tilde{R} \qquad \|\delta A\| \leq \|A\| O(\epsilon_{mach})$ Thm 1

(**) $\tilde{d} = \tilde{Q}^*(b + \delta b) \qquad \|\delta b\| \leq \|b\| O(\epsilon_{mach})$ HW 16.1

$$(***)\quad (\tilde{R} + \delta\tilde{R})_{1:n,1:n}\,\tilde{x}_* = \tilde{d}_{1:n} \qquad \|\delta\tilde{R}\| \leq \|\tilde{R}\|\,O(\varepsilon_{mach})$$

LNT Lecture 17

Collecting backward errors leads to

Thm 2 Let A have full column rank. If the Least-Squares solution is computed by the above Algorithm in floating point arithmetic, then the computed soln $\tilde{x}_*$ satisfies

$$\tilde{x}_* = \operatorname{argmin} \|(A + \delta A)x - b\|_2$$

where $\|\delta A\| \leq \|A\|\,O(\varepsilon_{mach})$.

Sketch of pf

To prove Thm 2, we need to show that $\tilde{x}_*$ solves

$$\tilde{A}\tilde{x}_* = P_{\tilde{A}}\, b \quad \text{where } \|\tilde{A} - A\| = \|A\|\,O(\varepsilon_{mach})$$

i.e. $\tilde{x}_*$ is a least-squares soln to a nearby problem.

From $(***)$, we know that

$$(\tilde{R}+\delta\tilde{R})_{1:n,1:n}\tilde{x}_* = \tilde{d}_{1:n} \quad \text{with} \quad \|\delta\tilde{R}\| = \|\tilde{R}\|O(\varepsilon_{mach}),$$

and from (**), we know that

$$\tilde{d}_{1:n} = \underbrace{\begin{bmatrix} - & \tilde{q}_1^* & - \\ & \vdots & \\ - & \tilde{q}_n^* & - \end{bmatrix}}_{\tilde{Q}'^*}(b+\delta b) \quad \text{with} \quad \|\delta b\| = \|b\|O(\varepsilon_{mach}).$$

Now, combine these & left multiply by $\tilde{Q}'$:

$$\tilde{Q}'(\tilde{R}+\delta\tilde{R})_{1:n,1:n}\tilde{x}_* = \tilde{Q}'\tilde{Q}'^*(b+\delta b).$$

Since $\underbrace{\tilde{Q}'\tilde{R}_{1:n,1:n}}_{\text{thin}} = \underbrace{\tilde{Q}\tilde{R}}_{\text{full}} = A+\delta A$ by (*), we have

$$\underbrace{(A+\delta A+\tilde{Q}'\delta\tilde{R})}_{\tilde{A}}\tilde{x}_* = \underbrace{\tilde{Q}'\tilde{Q}'^*}_{P_{\tilde{A}}}(b+\delta b)$$

The backward error bounds on δA and $\delta\tilde{R}$ imply that $\|\tilde{A}-A\| = \|A\|O(\varepsilon_{mach})$. Moreover, $A = \tilde{Q}'(\tilde{R}+\delta\tilde{R})$ so $P_{\tilde{A}} = \tilde{Q}'\tilde{Q}'^*$.

$\Rightarrow x_*$ solves a least-squares problem w/slightly perturbed data A and b.

⇒ We can complete the proof of Thm 2 by moving $\tilde{Q}'\tilde{Q}'^{*}\delta b$ to left-hand side and looking for matrix δB s.t. $\delta B\,\tilde{x}_{*} = \tilde{Q}'^{*}\delta b$, which is an underdetermined problem for δB. Then

$$\tilde{A} = \tilde{Q}'(\tilde{R} + \delta\tilde{R} + \delta B)$$

with estimates on $\|\tilde{A} - A\|$ following readily.

A good choice for δB is the "minimal norm" soln to the underdetermined problem:

$$\delta B = \left[\tilde{Q}'\delta b\right]\frac{\tilde{x}_{*}^{*}}{\|\tilde{x}_{*}\|^{2}}$$

This last step of "moving" perturbations from one piece of data (b) to another (A) is a common theme in backward error analysis.