Last time: $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$, $b \in \mathbb{C}^n$, $\operatorname{rank}(A) = n$

Least Squares: find $\hat{x} = \underset{x \in \mathbb{C}^n}{\operatorname{argmin}} \|b - Ax\|_2^2$

Solution: $\hat{x} = (A^*A)^{-1}A^* b$ (normal equation)

$= V\Sigma^{+}U^* b$ (SVD, $A = U\Sigma V^*$)

method	normal eqn	SVD
conditioning	$\approx \kappa_2^2(A)$	$\approx \kappa_2(A)$ ($\|r\|$ small)
operation count	$\approx 2mn^2$ FLOPs	$\approx 4mn^2$ FLOPs

- Solving least-squares via QR factorization

  Computing SVD is expensive. can we work with orthogonal transform but lower cost?

- (reduced) QR factorization

$$A = Q\,R$$

$R \in \mathbb{C}^{n\times n}$ upper triangular matrix

$\boxed{\text{Not } QQ^* = I_m}$

$Q \in \mathbb{C}^{m\times n}$

$Q^*Q = I_n$

orthonormal columns $(m \geq n)$

when $\operatorname{rank}(A) = n$, $\Rightarrow \operatorname{rank}(R) = n$

hence $R(A) = R(Q)$

Let $Q = [\,q_1 \cdots q_n\,]$, $q_i \in \mathbb{C}^m$

projection onto $R(A)$

Then $\quad P_A x = P_Q x = \sum_{i=1}^{n} q_i (q_i^* x) = Q Q^* x$

Hence $\quad \hat{x} \in \arg\min_x \|Ax - b\|_2^2$

$$\iff \quad r = b - A\hat{x} \perp R(A)$$

$$\iff \quad P_Q r = 0$$

$$\iff \quad QR\hat{x} = QQ^* b$$

$$\iff \quad R\hat{x} = Q^* b$$

$$\iff \quad \hat{x} = R^{-1} Q^* b$$

Actually, from normal equations,

$$\hat{x} = (A^*A)^{-1} A^* b = (R^*R)^{-1} R^* Q^* b = R^{-1} Q^* b$$

- How to compute QR factorization?

Idea 1: convert basis to orthonormal one

Gram–Schmidt: given $a_1, \cdots, a_n \in \mathbb{C}^m$, produce orthonormal $q_1, \cdots, q_n$ with $\operatorname{span}\{q_1, \cdots, q_n\} = \operatorname{span}\{a_1, \cdots, a_n\}$

$$q_1 = \frac{a_1}{\|a_1\|_2}$$

$$q_2' = a_2 - (q_1^* a_2) q_1, \qquad q_2 = \frac{q_2'}{\|q_2'\|_2}$$

$$\cdots$$

$$q_j' = a_j - \sum_{i=1}^{j-1} (q_i^* a_i) q_i, \qquad q_j = \frac{q_j'}{\|q_j'\|_2} \qquad j = 3, \cdots, n$$

$\iff$

$$a_1 = \|a_1\|_2 q_1$$

$$a_2 = (q_1^* a_2) q_1 + q_2 \|a_2 - (q_1^* a_2) q_1\|_2$$

$$\cdots$$

$$a_j = \sum_{i=1}^{j-1} (q_i^* a_i) q_i + q_j \|a_j - \sum_{i=1}^{j-1} (q_i^* a_i) q_i\|_2$$

$\Leftrightarrow$ $A = QR$

$A = [\, a_1 \cdots a_n \,]$, $\quad R_{ij} = \begin{cases} q_i^* a_j, & i<j \\ \| a_j - \sum_{k=1}^{j-1} (q_k^* a_k) q_k \|_2, & i=j \\ 0, & \text{otherwise} \end{cases}$

Implementation :

<u>Classical GS</u> Input: $A \in \mathbb{C}^{m\times n}$ of rank n

Output: $Q \in \mathbb{C}^{m\times n}$, $R \in \mathbb{C}^{n\times n}$

for $j = 1, \cdots, n$

for $i = 1, \cdots, j-1$

$R_{ij} = q_i^* a_j \longrightarrow (j-1)(2m-1)$

end

$q_j' = a_j - \sum_{i=1}^{j-1} R_{ij} q_i \longrightarrow 2m(j-1)$

$R_{jj} = \| q_j' \|_2$, $\quad q_j = q_j' / R_{jj}$

$\hookrightarrow 2m \qquad \hookrightarrow m$

end

* operation count:

$\sum_{j=1}^{n} (j-1)(2m-1) + 2(j-1)m + 2m + m$

$\approx \sum_{j=1}^{n} 4jm$

$\approx 2mn^2$ FLOPs

• weakness: there is nothing to force orthonormality of Q in classical Gram-Schmidt

ex. $A = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & & \\ & \varepsilon & \\ & & \varepsilon \end{bmatrix}$

assume ε so small such that $fl(1+\varepsilon^2) = 1$

then $Q_{GS} = \begin{bmatrix} 1 & 0 & 0 \\ \varepsilon & -1/\sqrt{2} & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 1/\sqrt{2} \end{bmatrix}$

$q_1 \quad q_2 \quad q_3$

$$q_2^* q_3 = \frac{1}{2}$$

- Remedy: Instead of orthogonalizing a_j at step j only, can orthogonalize a_j to q_i as soon as q_i is computed

modified GS

Let $a_k^{(1)} = a_k, \quad k = 1, \cdots, n$

operation counts $\approx 2mn^2$

for $k = 1, \cdots, n$

$\quad R_{kk} = \|a_k^{(k)}\|_2, \quad q_k = a_k^{(k)} / R_{kk}$

$\quad$ for $j = k+1, \cdots, n$

$\quad\quad R_{kj} = q_k^* a_j^{(k)}, \quad a_j^{(k+1)} = a_j^{(k)} - R_{kj} q_k$

$\quad$ end

end

Classical GS is equivalent to modified GS $\leftarrow$ exercise

Thm Suppose Modified GS is applied to $A \in \mathbb{R}^{m \times n}$ of rank n yielding $\hat{Q} \in \mathbb{R}^{m \times n}$, $\hat{R} \in \mathbb{R}^{n \times n}$,

$\exists\ c_i = c_i(m, n)$, s.t.

1) $A + \Delta A_1 = \hat{Q}\hat{R}, \quad \|\Delta A_1\|_2 \le c_1 \varepsilon_{mach} \|A\|_2$

$\|\hat{Q}^T\hat{Q} - I\|_2 \le c_2 \varepsilon_{mach} \kappa_2(A) / (1 - c_2' \varepsilon_{mach} \kappa_2(A))$

2) $\exists\ Q \in \mathbb{R}^{m \times n}$ with orthonormal columns s.t.

$A + \Delta A_2 = Q\hat{R}, \quad \|\Delta A_2\|_2 \le c_3 \varepsilon_{mach} \|A\|_2$

$\|Q - \hat{Q}\|_2 \le c_4 \varepsilon_{mach} \kappa_2(A) / (1 - c_4' \varepsilon_{mach} \kappa_2(A))$

Pf: Higham Thm 19.13 $\blacksquare$

Remark: The theorem states that the departure from orthonormality of $\hat{Q}$ is bounded by $O(\kappa_2(A)\,\varepsilon_{mach})$

Remark: Part 2) of the theorem states that $\hat{R}$ is the exact triangular QR factor of a matrix near to A, i.e. it is a good R-factor.

- Back to least-squares:

Solve least-squares via QR:

$$\hat{x} = \hat{R}^{-1} \underbrace{\hat{Q}^* b}$$

implicitly use $\hat{Q}^*\hat{Q} = I_n$. but suffer from large $\kappa_2(A)$

Remark: If we translation the result to perturbation form, we get that the computed $\hat{x}$ is the exact solution of the LS problem

$$\| b+\Delta b - (A+\Delta A) y \|_2^2$$

where $\|\Delta b\|_2 / \|b\|_2 \lesssim \kappa_2(A)\,\varepsilon_{mach}$, $\|\Delta A\|_2 / \|A\|_2 \lesssim \varepsilon_{mach}$

NOT Backward stable

- To resolve this problem, we can apply Modified GS to $[A \;\; b]$, so that $Q^* b$ is implicitly computed in the last step of GS

$$[A \;\; b] = [Q_1 \;\; q_{n+1}] \begin{bmatrix} R & z \\ 0 & \rho \end{bmatrix}$$

We have

$$Ax - b = [A \;\; b] \begin{bmatrix} x \\ -1 \end{bmatrix}$$

$$= [Q_1 \;\; q_{n+1}] \begin{bmatrix} Rx - z \\ -\rho \end{bmatrix}$$

$$= Q_1(Rx - z) - \rho q_{n+1}$$

Hence $\|Ax-b\|_2^2 = \|Rx - z\|_2^2 + \rho^2$

so $x = R^{-1}z$ is the Least-squares solution

<u>Thm</u> Solving LS via Modified GS for $[A\ b]$ has forward error as good as a backward stable algo.

- It is possible to perform QR factorization faster than $2mn^2$ FLOPs if we don't form Q explicitly

* Householder :

$$\underset{A}{\begin{pmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{pmatrix}} \xrightarrow{Q_1} \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{pmatrix} \xrightarrow{Q_2} \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \end{pmatrix} \xrightarrow{Q_3} \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \end{pmatrix}$$

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & H_2 \end{bmatrix} \qquad Q_3 = \begin{bmatrix} I & \\ & H_3 \end{bmatrix}$$

H_2

$$Q_3 Q_2 Q_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix} \Rightarrow A = (Q_3 Q_2 Q_1)^* \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Q_i unitary

$$= Q \begin{bmatrix} R \\ 0 \end{bmatrix} \leftarrow \text{full QR}$$

$Q \in \mathbb{C}^{m\times n}$ unitary

$\begin{bmatrix} R \\ 0 \end{bmatrix} \in \mathbb{C}^{m\times n}$ upper triangular

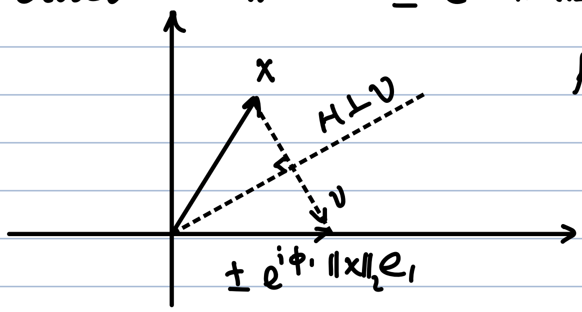
* How to choose Q_k ?

Let $$Q_k = \begin{array}{cc} \begin{array}{cc} k-1 & m-(k-1) \end{array} & \\ \begin{bmatrix} I & 0 \\ 0 & H_k \end{bmatrix} & \begin{array}{c} k-1 \\ m-(k-1) \end{array} \end{array}$$, H_k unitary (thus norm preserving)

keep the first $k-1$ coordinates unchanged

$x \in \mathbb{C}^{m-(k-1)}$ be kth, $\cdots$, mth entries of kth column

Goal: $H_k x = \pm e^{i\phi_1} \|x\|_2 e_1$ $\quad \begin{pmatrix} x \\ x \\ x \end{pmatrix} \xrightarrow{H_k} \begin{pmatrix} x \\ 0 \\ 0 \end{pmatrix}$



Householder transform: reflection across the plane H

$\phi_1 = \arg(x_1)$ is added to make x, $\pm e^{i\phi_1}\|x\|_2 e_1$ symmetric with respect to H, i.e. $v^* x = -v^*(\pm e^{i\phi_1}\|x\|_2 e_1)$.

Such that $H_k x = \pm e^{i\phi_1}\|x\|_2 e_1$ (otherwise H_k doesn't reflect x to $\pm e^{i\phi_1}\|x\|_2 e_1$!)

$$H_k x = x - 2P_v x = x - 2\left(\frac{v^*}{\|v\|_2} x\right)\frac{v}{\|v\|_2}$$

$$= \underbrace{\left(I - \frac{2vv^*}{\|v\|_2^2}\right)}_{\text{Householder transform}} x$$

(a) For stability, one may choose $v = e^{i\phi_1}\|x\| e_1 + x$

to avoid catastrophic cancellation in computing $\|v\|_2^2$ when $\|x\|_2 \approx |x_1|$ ← why?

(b) Householder matrices H_k is never formed in upper triangularizing A, storage and computations use solely the Householder vector v.

- Implementation:

Householder QR:

For $k = 1, \cdots, n$

$x \leftarrow A(k:m, k)$

$v_k = \text{sign}(x(1))\|x\|_2 e_1 + x$

$v_k \leftarrow v_k / \|v_k\|_2$

substraction $(m-k)(n-k)$ FLOPs; outer product $(m-k)(n-k)$ FLOPs; matrix-vector product $2(m-k)(n-k)$ FLOPs

$A(k:m, k:n) \leftarrow A(k:m,k:n) - 2v_k\left(v_k^* A(k:m, k:n)\right)$

end

- operation count:

$$\approx \sum_{k=1}^{n} 4(m-k)(n-k) \quad \text{FLOPs}$$

$$\approx 4\left(mn^2 - (m+n)\frac{n^2}{2} + \frac{n^3}{3}\right)$$

$$\approx 2mn^2 - \frac{2}{3}n^3$$

- Stability of Householder transform

Since the orthogonal transform is performed by Householder vector, the orthogonality of Q is enforced

<u>Thm</u> Let $\begin{bmatrix} \hat{R} \\ 0 \end{bmatrix} \in \mathbb{R}^{m\times n}$ be the computed upper triangular QR factor of $A \in \mathbb{R}^{m\times n}$, Let $Q = (\hat{Q}_n \cdots \hat{Q}_1)^T$ be the exact orthogonal matrix obtained from Householder vectors computed from the algorithm

Then $\quad A + \Delta A = Q\hat{R} \quad$ with $\|\Delta A\|_2 \le C_{m.n} \|A\|_2$

Let $\hat{Q} = fl((\hat{Q}_n \cdots \hat{Q}_1)^T)$ be the computed orthogonal matrix

Then $\quad \hat{Q} = Q(I_m + \Delta I) \quad$ with $\|\Delta I\|_2 \le C_{m.n}$

$\hat{Q}$ is very close to orthogonal matrix regardless of $K(A)$!

(This a consequence of the backward stability of matrix multiplication)

- Back to solve least-squares

Solve least-squares via QR :

$$\hat{x} = \hat{R}^{-1} \underbrace{\hat{Q}^* b}$$

use $\hat{Q}^*\hat{Q} = I_n$, but NOT suffer from large $K_2(A)$

<u>thm</u> Let $A \in \mathbb{R}^{m\times n}$ have full rank and that the least square problem $\min_x \|Ax - b\|_2$ is solved using Householder QR factorization. The computed solution $\hat{x}$ is the exact solution to

$$\min_{x} \| b+\Delta b - (A+\Delta A)\hat{x} \|_2$$

where $\|\Delta A\|_2 \le C_{m,n} \|A\|_2$, $\|\Delta b\|_2 \le C_{m,n} \|b\|_2$

Backward stable!

Summary:

Operation count:

$O(2mn^2)$ $\longrightarrow$ $O(4mn^2)$

normal equation < Householder QR (without computing Q explicitly) < Modified GS (Augmented) < SVD

Rounding error:

$O(\kappa_2^2(A)\,\varepsilon_{mach})$ $\longrightarrow$ $O(\kappa_2(A)\,\varepsilon_{mach})$

normal equation > Householder QR (without computing Q explicitly) ≈ Modified GS (Augmented) ≈ SVD