9. QR factorization

- solving the normal equations
- QR factorization
- modified Gram-Schmidt algorithm
- Cholesky factorization versus QR factorization

Least-squares methods

minimize
$$||Ax - b||^2$$

(A is $m \times n$ and left-invertible)

normal equations

$$A^T A x = A^T b$$

- method 1: solve the normal equations using the Cholesky factorization
- method 2: use the QR factorization

method 2 has better numerical properties; method 1 is faster

Method 1: Cholesky factorization

- 1. calculate $C=A^TA$ (C is symmetric: $\frac{1}{2}n(n+1)(2m-1)\approx mn^2$ flops)
- 2. Cholesky factorization $C = LL^T$ ($(1/3)n^3$ flops)
- 3. calculate $d = A^T b$ (2mn flops)
- 4. solve Lz = d by forward substitution (n^2 flops)
- 5. solve $L^T x = z$ by back substitution (n^2 flops)

cost for large m, n: $mn^2 + (1/3)n^3$ flops

Example

$$A = \begin{bmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ 7 \\ 2 \end{bmatrix}$$

1.
$$A^T A = \begin{bmatrix} 25 & -50 \\ -50 & 101 \end{bmatrix}$$
 and $A^T b = \begin{bmatrix} 25 \\ -48 \end{bmatrix}$

2. Cholesky factorization:
$$A^TA = \begin{bmatrix} 5 & 0 \\ -10 & 1 \end{bmatrix} \begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix}$$

3. solve
$$\begin{bmatrix} 5 & 0 \\ -10 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 25 \\ -48 \end{bmatrix}$$
: solution is $z_1 = 5$, $z_2 = 2$

4. solve
$$\begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$
: solution is $x_1 = 5$, $x_2 = 2$

QR factorization

if A is $m \times n$ and left-invertible then it can be factored as

$$A = QR$$

- R is $n \times n$ and upper triangular with $r_{ii} > 0$
- Q is $m \times n$ and orthogonal $(Q^TQ = I)$

can be computed in $2mn^2$ flops (more later)

Least-squares method 2: QR factorization

rewrite normal equations $A^TAx = A^Tb$ using QR factorization A = QR:

$$A^{T}Ax = A^{T}b$$

$$R^{T}Q^{T}QRx = R^{T}Q^{T}b$$

$$R^{T}Rx = R^{T}Q^{T}b (Q^{T}Q = I)$$

$$Rx = Q^{T}b (R nonsingular)$$

algorithm

- 1. QR factorization of A: $A = QR (2mn^2 \text{ flops})$
- 2. form $d = Q^T b$ (2mn flops)
- 3. solve Rx = d by back substitution (n^2 flops)

cost for large m, n: $2mn^2$ flops

Example

$$A = \begin{bmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} -1 \\ 7 \\ 2 \end{bmatrix}$$

1. QR factorization: A = QR with

$$Q = \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix}$$

2. calculate $d = Q^T b = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$

3. solve $\begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$: solution is $x_1 = 5$, $x_2 = 2$

Computing the QR factorization

partition A = QR as

$$\begin{bmatrix} a_1 & A_2 \end{bmatrix} = \begin{bmatrix} q_1 & Q_2 \end{bmatrix} \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

- ullet a_1 and q_1 are m-vectors; A_2 and Q_2 are m imes (n-1)
- r_{11} is a scalar, R_{12} is $1 \times (n-1)$, R_{22} is upper triangular of order n-1
- q_1 and Q_2 must satisfy

$$\begin{bmatrix} q_1^T \\ Q_2^T \end{bmatrix} \begin{bmatrix} q_1 & Q_2 \end{bmatrix} = \begin{bmatrix} q_1^T q_1 & q_1^T Q_2 \\ Q_2^T q_1 & Q_2^T Q_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix},$$

i.e.,

$$q_1^T q_1 = 1,$$
 $Q_2^T Q_2 = I,$ $q_1^T Q_2 = 0$

recursive algorithm ('modified Gram-Schmidt algorithm')

$$\begin{bmatrix} a_1 & A_2 \end{bmatrix} = \begin{bmatrix} q_1 & Q_2 \end{bmatrix} \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} = \begin{bmatrix} q_1 r_{11} & q_1 R_{12} + Q_2 R_{22} \end{bmatrix}$$

1. determine q_1 and r_{11} :

$$r_{11} = ||a_1||, \qquad q_1 = (1/r_{11})a_1$$

2. R_{12} follows from $q_1^T A_2 = q_1^T (q_1 R_{12} + Q_2 R_{22}) = R_{12}$:

$$R_{12} = q_1^T A_2$$

3. Q_2 and R_{22} follow from

$$A_2 - q_1 R_{12} = Q_2 R_{22},$$

 $\it i.e.$, the QR factorization of an $m \times (n-1)$ -matrix

cost: $2mn^2$ flops (no proof)

 ${\bf proof}$ that the algorithm works if A is left-invertible

• step 1: $a_1 \neq 0$ because A has a zero nullspace

• step 3: $A_2 - q_1 R_{12}$ is left-invertible

$$A_2 - q_1 R_{12} = A_2 - (1/r_{11})a_1 R_{12}$$

hence if $(A_2 - q_1 R_{12})x = 0$, then

$$\begin{bmatrix} a_1 & A_2 \end{bmatrix} \begin{bmatrix} -R_{12}x/r_{11} \\ x \end{bmatrix} = 0$$

but this implies x=0 because A has a zero nullspace

- ullet therefore the algorithm works for a left-invertible m imes n-matrix if it works for a left-invertible m imes (n-1)-matrix
- obviously it works for a matrix with one nonzero column; hence by induction it works for all left-invertible matrices

Example

$$A = \begin{bmatrix} 9 & 0 & 26 \\ 12 & 0 & -7 \\ 0 & 4 & 4 \\ 0 & -3 & -3 \end{bmatrix}$$

we want to factor A as

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}$$
$$= \begin{bmatrix} q_1r_{11} & q_1r_{12} + q_2r_{22} & q_1r_{13} + q_2r_{23} + q_3r_{33} \end{bmatrix}$$

with

$$q_1^T q_1 = 1,$$
 $q_2^T q_2 = 1,$ $q_3^T q_3 = 1$
 $q_1^T q_2 = 0,$ $q_1^T q_3 = 0,$ $q_2^T q_3 = 0$

and $r_{11} > 0$, $r_{22} > 0$, $r_{33} > 0$

ullet determine first column of Q, first row of R

$$-a_1 = q_1 r_{11}$$
 with $||q_1|| = 1$

$$r_{11} = ||a_1|| = 15,$$
 $q_1 = (1/r_{11})a_1 = \begin{vmatrix} 3/5 \\ 4/5 \\ 0 \\ 0 \end{vmatrix}$

- inner product of q_1 with a_2 and a_3 :

$$q_1^T a_2 = q_1^T (q_1 r_{12} + q_2 r_{22}) = r_{12}$$

 $q_1^T a_3 = q_1^T (q_1 r_{13} + q_2 r_{23} + q_3 r_{33}) = r_{13}$

therefore, $r_{12} = q_1^T a_2 = 0$, $r_{13} = q_1^T a_3 = 10$

$$A = \begin{bmatrix} 9 & 0 & 26 \\ 12 & 0 & -7 \\ 0 & 4 & 4 \\ 0 & -3 & -3 \end{bmatrix} = \begin{bmatrix} 3/5 & q_{12} & q_{13} \\ 4/5 & q_{22} & q_{23} \\ 0 & q_{32} & q_{33} \\ 0 & q_{42} & q_{43} \end{bmatrix} \begin{bmatrix} 15 & 0 & 10 \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}$$

• determine 2nd column of Q, 2nd row or R

$$\begin{bmatrix} 0 & 26 \\ 0 & -7 \\ 4 & 4 \\ -3 & -3 \end{bmatrix} - \begin{bmatrix} 3/5 \\ 4/5 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 10 \end{bmatrix} = \begin{bmatrix} q_2 & q_3 \end{bmatrix} \begin{bmatrix} r_{22} & r_{23} \\ 0 & r_{33} \end{bmatrix}$$

$$i.e.$$
, the QR factorization of $\left[egin{array}{ccc} 0&20\\0&-15\\4&4\\-3&-3 \end{array}
ight]=\left[egin{array}{ccc} q_2r_{22}&q_2r_{23}+q_3r_{33} \end{array}
ight]$

- first column is q_2r_{22} where $||q_2||=1$, hence

$$r_{22} = 5,$$
 $q_2 = \begin{bmatrix} 0 \\ 0 \\ 4/5 \\ -3/5 \end{bmatrix}$

- inner product of q_2 with 2nd column gives r_{23}

$$q_2^T \begin{bmatrix} 20 \\ -15 \\ 4 \\ -3 \end{bmatrix} = q_2^T (q_2 r_{23} + q_3 r_{33}) = r_{23}$$

therefore, $r_{23} = 5$

QR factorization so far:

$$A = \begin{bmatrix} 9 & 0 & 26 \\ 12 & 0 & -7 \\ 0 & 4 & 4 \\ 0 & -3 & -3 \end{bmatrix} = \begin{bmatrix} 3/5 & 0 & q_{13} \\ 4/5 & 0 & q_{23} \\ 0 & 4/5 & q_{33} \\ 0 & -3/5 & q_{43} \end{bmatrix} \begin{bmatrix} 15 & 0 & 10 \\ 0 & 5 & 5 \\ 0 & 0 & r_{33} \end{bmatrix}$$

 \bullet determine 3rd column of Q, 3rd row of R

$$\begin{bmatrix} 26 \\ -7 \\ 4 \\ -3 \end{bmatrix} - \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 4/5 \\ 0 & -3/5 \end{bmatrix} \begin{bmatrix} 10 \\ 5 \end{bmatrix} = q_3 r_{33}$$
$$\begin{bmatrix} 20 \\ -15 \\ 0 \\ 0 \end{bmatrix} = q_3 r_{33}$$

with $||q_3|| = 1$, hence

$$r_{33} = 25, \qquad q_3 = \begin{bmatrix} 4/5 \\ -3/5 \\ 0 \\ 0 \end{bmatrix}$$

in summary,

$$A = \begin{bmatrix} 9 & 0 & 26 \\ 12 & 0 & -7 \\ 0 & 4 & 4 \\ 0 & -3 & -3 \end{bmatrix} = \begin{bmatrix} 3/5 & 0 & 4/5 \\ 4/5 & 0 & -3/5 \\ 0 & 4/5 & 0 \\ 0 & -3/5 & 0 \end{bmatrix} \begin{bmatrix} 15 & 0 & 10 \\ 0 & 5 & 5 \\ 0 & 0 & 25 \end{bmatrix}$$
$$= QR$$

Cholesky factorization versus QR factorization

example: minimize $||Ax - b||^2$ with

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \\ 0 & 0 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ 10^{-5} \\ 1 \end{bmatrix}$$

solution

normal equations $A^TAx = A^Tb$:

$$\begin{bmatrix} 1 & -1 \\ -1 & 1+10^{-10} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 10^{-10} \end{bmatrix}$$

solution: $x_1 = 1$, $x_2 = 1$

let us compare both methods, rounding intermediate results to 8 significant decimal digits

method 1 (Cholesky factorization)

 A^TA and A^Tb rounded to 8 digits:

$$A^T A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \qquad A^T b = \begin{bmatrix} 0 \\ 10^{-10} \end{bmatrix}$$

no solution (singular matrix)

method 2 (QR factorization): factor A = QR and solve $Rx = Q^Tb$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \end{bmatrix}, \qquad Q^T b = \begin{bmatrix} 0 \\ 10^{-5} \end{bmatrix}$$

rounding does not change any values

solution of $Rx = Q^T b$ is $x_1 = 1$, $x_2 = 1$

conclusion: numerical stability of QR factorization method is better

Summary

cost for dense A

- method 1 (Cholesky factorization): $mn^2 + (1/3)n^3$ flops
- method 2 (QR factorization): $2mn^2$ flops
- method 1 is always faster (twice as fast if $m \gg n$)

cost for large sparse A

- method 1: we can form A^TA fast, and use a sparse Cholesky factorization (cost $\ll mn^2 + (1/3)n^3$)
- method 2: exploiting sparsity in QR factorization is more difficult

numerical stability: method 2 is more accurate

in practice: method 2 is preferred; method 1 is often used when A is very large and sparse