Welcome to Ma221! Lec 11, Fall24

Recall Sparse Cholesky

Choose ordering (RCM, MD, ND...)
only depends on nonzero locations,
not values, to minimize work
Build data structures for A, L
Perform factorization:

Contrast with GE for general A:
ordering needs to depend on entries
usual partial pivoting could could make
lots of fill in

Lots of algorithms, examine a few:

1. Threshold pivoting, among pivot choices at each step pick one within a factor of 2 or 3 of largest, with least fill in: tradeoff stability and speed

② Static Pivoting (SuperLU)

① reorder and scale A to make diagonal as large as possible

Thm: for any nonsingular A
$\exists$ perm P and 2 diagonal D_1, D_2
s.t. $B = D_1 \cdot A \cdot P \cdot D_2$
(*) $|B(i,i)| = 1$ and $|B(i,j)| \le 1$

② reorder rows and columns of B using same techniques as Cholesky, maintains (*)

③ During factorization, If a prospective pivot is too small, make it bigger (rare)

Difference between B and factorized matrix is low rank $\Rightarrow$ use Sherman-Morrison-Woodbury for fast solver, or GMRES (from Chap 6)

Lots of algorithms + software (see class webpage)

Structured Matrices
("data sparse")

could be dense, depend on $O(n)$ data

Many structures (depends on physics, ...)

Common case with common structure

Vandermonde: $V(i,j) = x_i^{j-1}$

Cauchy $C(i,j) = \dfrac{1}{x_i + y_j}$

Toeplitz $T(i,j) = x_{i-j}$

constant along diagonals

Hankel $H(i,j) = x_{i+j}$

Eg: $Vz = b$ mean $\sum_{j=1}^{n} x_i^{j-1} \cdot z_j = b_i$

$\Rightarrow$ polynomial interpolation

$O(n^2)$ using Newton Interp.

similar trick for $V^T z = b$

Eg Multiplying $Tz \equiv$ convolution

$\Rightarrow$ use FFT

Eg: Solving $Cx = b$ arises in

rational interpolation

Common Structure of all these X:

$AX + XB =$ low rank

for some simple A and B

Ex: Vandermonde V

$D = \mathrm{diag}(x_1, \ldots, x_n)$

$D \cdot V = V$ "shifted left"

$V \cdot P = V \cdot \begin{bmatrix} 0 & 0 & & 1 \\ 1 & 0 & & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & 0 & & \\ 0 & 0 & & 0 \end{bmatrix} = V$ shifted left

$DV - VP =$ zero except last column

$=$ rank 1

Ex: Toeplitz T

$P \cdot T - T \cdot P = T$ shifted down

$- T$ shifted left

$=$ mostly zero, nonzero in first row, last column

$=$ rank 2

Ex: Cauchy C

$\mathrm{diag}(x_1, \ldots, x_n) \cdot C + C \cdot \mathrm{diag}(y_1, \ldots, y)$

$=$ all ones $=$ rank 1

Def: this low rank is called "displacement rank"

Thm (Kailath et al) There is an $O(n^2)$ solver if displacement rank is $O(1)$

(stability not guaranteed)

Chap 3: Least Squares

Ex: polynomial fitting

given sample point (y_i, b_i) $i=1,\dots,n$

find "best" polynomial of fixed degree

to minimize $\sum_{i=1}^{m} (p(y_i) - b_i)^2$

minimize $\|Ax - b\|_2$

$A(i,j) = y_i^{j-1}$, x are coeffs of p

$p(y) = x_1 + x_2 \cdot y + x_3 \cdot y^2 + \cdots + x_j y^{j-1}$

Matlab demo: $\sin(\pi y/5) + y/5$

polyfit31.m

Standard Notation:

$\underset{x}{\operatorname{argmin}} \|Ax - b\|_2$ $\quad A^{m\times n}$ $\quad m>n$

$m > n$ means overdetermined

don't expect $Ax = b$ exactly

Other variants (all in LAPACK)

Constrained LS $\quad \underset{x:\, Bx = y}{\operatorname{argmin}} \|Ax - b\|_2$

where #rows(B) $\le$ # cols(A)... so $Bx = y$ not overdetermined

$\le$ #rows(A) + # rows(B)... so x unique

Weighted LS: $\quad \underset{x}{\operatorname{argmin}} \|y\|_2$ s.t.

$$b = Ax + By$$

if $B = I$, $y = b - Ax \Rightarrow$ standard LS

if B square, nonsingular

$$\underset{x}{\operatorname{argmin}} \|B^{-1}(Ax - b)\|_2$$

Underdetermined $\quad$ #rows(A) < #cols(A)

so $\underset{x}{\operatorname{argmin}} \|Ax - b\|$ not unique

can also arise if A not full rank $\Rightarrow$

space of solutions (add any $z: Az = 0$ to a solution to get another)

to make solution unique, use

$$\underset{x}{\operatorname{argmin}} \|x\|_2 \quad \text{s.t.} \quad Ax = b$$

Ridge Regression

$$\operatorname*{argmin}_x \|Ax-b\|_2^2 + \lambda \|x\|_2^2$$

$\lambda > 0$ tuning parameter

solution unique if $\lambda > 0$

Total Least Squares

$$\operatorname*{argmin}_x \|[E, r]\|_2$$

$$(A+E)x = b+r$$

Algorithms for overdetermined LS

(building blocks for all other cases)

Solve: Normal Equations (NE)

$$A^T A x = A^T b \quad \text{(real case)}$$

$A^T A$ s.p.d. $\Rightarrow$ Cholesky

fastest in dense case

(fewest flops, least comm)

not stable if A ill-conditioned

Use QR decomposition $A^{m\times n} = QR$

$Q^{m\times n}$ orthogonal

$R^{n\times n}$ upper triangular

$$x = R^{-1} Q^T b$$

Gram-Schmidt - unstable if A ill-conditioned
(lots of variants, trading off speed and stability)

Householder - stable ($x = A\backslash b$ in Matlab)
(blocked Householder, to use BLAS3)

(possible to get Q and R via NE, called CholeskyQR, fast but can be as unstable as NE)

SVD: most "complete" solution
gives cond. number, error bounds,
works in rank deficient case,
expensive.

Convert to a square Linear system with A, and A^T in a bigger matrix (also for sparsity) (see Q 3.3)

Normal Equations:

Thm: A full column rank, then solution of $A^TAx = A^Tb$ (NE) minimizes $\|Ax-b\|_2$

proof: Assume x satisfies NE, show it minimizes $\|Ax-b\|_2$: show

$\|A(x+e)-b\|_2^2$ minimized at $e=0$

$$= (A(x+e)-b)^T (A(x+e)-b)$$

$$= (Ax + Ae - b)^T (Ax + Ae - b)$$

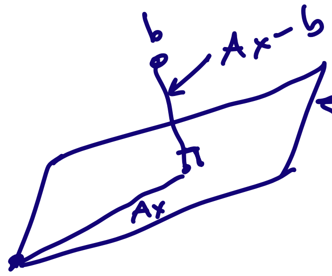
$$= (Ax-b + Ae)^T (Ax-b + Ae)$$

$$= (Ax-b)^T(Ax-b) + (Ae)^T(Ae) + \underbrace{2e^T(A^T(Ax-b))}_{=0 \text{ by NE}}$$

$$= \|Ax-b\|_2^2 + \|Ae\|_2^2$$

$$\geq \|Ax-b\|_2^2, \quad = \text{ if } Ae = 0$$

since A full rank, $\Rightarrow e=0$



all vectors $A\cdot y$

$Ax-b \perp Ay$ for all y

$(Ay)^T(Ax-b) = 0 \quad \forall y$

$y^T(A^TAx - A^Tb) = 0 \quad \forall y$

$\Rightarrow A^TAx = A^Tb$

Cost: form A^TA + solve $A^TAx = b$

in flops cost $= mn^2 + \frac{n^3}{3}$

in # words moved

know how to minimize both steps

QR: $A = QR \quad A^{m\times n}, \; Q^{m\times n} \quad R^{n\times n}$

orthonormal columns

solution $\underset{x}{\operatorname{argmin}} \|Ax-b\|_2 = R^{-1}Q^T b$

proof 1: $A = QR = \begin{array}{cc} & \begin{array}{cc} n & m-n \end{array} \\ m & \left[\begin{array}{c|c} Q & \hat{Q} \end{array}\right] \end{array} \begin{array}{cc} \begin{bmatrix} R \\ 0 \end{bmatrix} & \begin{array}{c} n \\ m-1 \end{array} \end{array}$ orthogonal

$$\|Ax-b\|_2^2 = \left\| \begin{bmatrix} Q^T \\ \hat{Q}^T \end{bmatrix} (Ax-b) \right\|_2^2$$

$$= \left\| \begin{bmatrix} Q^T \\ \hat{Q}^T \end{bmatrix} (QRx-b) \right\|_2^2$$

$$= \left\| \begin{bmatrix} Q^TQRx - Q^Tb \\ \hat{Q}^TQRx - \hat{Q}^Tb \end{bmatrix} \right\|_2^2$$

$$= \left\| \begin{bmatrix} Rx - Q^Tb \\ 0 - \hat{Q}^Tb \end{bmatrix} \right\|_2^2$$

$$= \|Rx - Q^Tb\|_2^2 + \|\hat{Q}^Tb\|_2^2$$

$$\geq \|\hat{Q}^Tb\|_2^2, \; = \text{ if } Rx = Q^Tb$$

or $x = R^{-1}Q^Tb$

proof 2: plug into NE

$$x = (A^TA)^{-1}A^Tb$$

$$= ((QR)^T(QR))^{-1}(QR)^Tb$$

$$= (R^T \underbrace{Q^T Q}_{I} R)^{-1} R^T Q^T b$$

$$= (R^T R)^{-1} R^T Q^T b$$

$$= R^{-1} \underbrace{R^{-T} R^T}_{} Q^T b$$

$$= R^{-1} Q^T b$$

Algorithms for $A = QR$

Classical + Modified Gram-Schmidt
CGS and MGS

Equate columns of $A = QR$

$$A(:,i) = \sum_{j=1}^{i} Q(:,j) R(j,i)$$

since columns of Q orthogonal

$$Q(:,j)^T \cdot A(:,i) = R(j,i)$$

for $i = 1$ to n
    $tmp = A(:,i)$
    for $j = 1 : i-1$
cost $= 2m$ ... $R(j,i) = Q(:,j)^T \cdot A(:,i)$ ... CGS
" $R(j,i) = Q(:,j)^T \cdot tmp$ ... MGS
" $tmp = tmp - R(j,i) \cdot Q(:,j)$
    end for
    $R(i,i) = \|tmp\|_2$

$Q(:,i) = tmp / R(i,i)$

end for

$\#\text{flops} = 2mn^2 + O(mn)$

$\sim 2 \cdot \text{cost}(NE)$ if $m \gg n$

Householder - stable

MGS - less stable

CGS - even less stable

2 Metrics for backward stability

want accurate factorization

$A + E = QR \quad \|E\| = O(\varepsilon) \cdot \|A\|$

also want Q close to orthogonal

$\|Q^T Q - I\| = O(\varepsilon)$

Fixes for GS: better but still not guaranteed stable

MGS2: MGS twice $\quad A = QR = (Q_1 \cdot R_1) \cdot R$

$= Q_1 (R_1 \cdot R)$

CGS2 = CGS twice