Welcome to Ma 221! (Feb 22)

GE + Cholesky on sparse matrices

How to pick order of rows/cols to minimize memory / #flops

Cholesky easier: stable for all orders

GE: trickier

Cholesky: RCM, MD last time

today: nested dissection (ND)

use Graph language:

vertices = rows and cols

edges = nonzero locations

weights = nonzero values

ND: Bisect vertices into 3 sets

$$V = V_1 \cup V_2 \cup \underset{\substack{\uparrow \\ \text{separator}}}{V_3} \quad \text{(disjoint)}$$

1. $|V_1| \approx |V_2|$
2. $|V_3|$ much smaller
3. no edges from V_1 to V_2

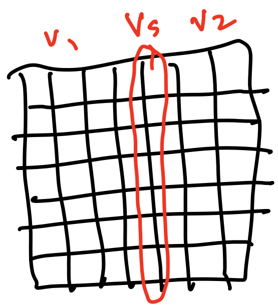
Order vertices in V_1 first, then V_2

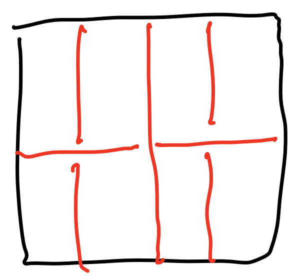
then V_s

$$A = \begin{array}{cc} & \begin{matrix} V_1 & \;\; V_2 & \;\; V_s \end{matrix} \\ \begin{matrix} V_1 \\ V_2 \\ V_s \end{matrix} & \begin{bmatrix} A_{11} & 0 & A_{1s} \\ 0 & A_{22} & A_{2s} \\ A_{s1} & A_{s2} & A_{ss} \end{bmatrix} \end{array} \Rightarrow L = \begin{bmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ L_{s1} & L_{s2} & L_{ss} \end{bmatrix}$$

$$A_{11} = L_{11} L_{11}^T \;,\; A_{22} = L_{22} L_{22}^T$$

apply bisection recursively to A_{11}, A_{22}





Apply to 7×7 mesh

Thm (George, Hoffman/Martin/Rose, Gilbert/Tarjan, 70s-80s)

Any ordering for Cholesky on $n \times n$ mesh does $\Omega(n^3)$ flops

Attained by nested dissection

Applies to planar graphs (drawable on paper without edges crossing)

Thm (Ballard, D., Holtz, Schwartz, 2009)

\# words moved between main memory and cache for LU or Cholesky is

$\Omega\left(\frac{\#\text{flops}}{\sqrt{M}}\right)$ M = cache size

For Cholesky on $n \times n$ mesh, $\Omega\left(\frac{n^3}{\sqrt{M}}\right)$

Thm (David, D., Grigori, Peyronnet, 2010)

Attainable by ND, done "carefully"

bottleneck is last (largest) separator

Contrast with band solver:

$\#\text{flops} = O(\text{bw}^2 \cdot \text{dimension}) = n^4$ for 2D mesh

(more algorithms for ND in CS267)

What about 3D meshes? $n \times n \times n$ mesh

ND still good idea

dimension $= n^3$

dense Cholesky costs $O(n^9)$

band Cholesky costs $O(n^7)$

ND costs $O(n^6)$

Steps of Sparse Cholesky

Choose ordering (RCM, MD, ND, ...)

Build ~~data~~ structure for A and L

Perform factorization

Contrast to GE with pivoting:

partial pivoting could cause large fill in, or rebuilding datastructure at every step

(1) Theshold Pivoting: among pivot choices at each step, pick one within a factor of 2 or 3 (user's choice) of largest, with least fill in

(2) Static Pivoting (SuperLU_Dist)

(i) reorder and scale A to make diagonal as large as possible

Thm: For any nonsingular A there is a perm P and 2 diagonal D_1, D_2 s.t.

$$B = D_1 \cdot AP \cdot D_2$$

$|B(i,i)| = 1$ and $|B(i,j)| \le 1 \quad i \ne j$

(2) Reorder rows and columns of B using same techniques as for Cholesky, keeps $|B(i,i)| = 1$
$|B(i,j)| \le 1 \quad i \ne j$

Build data structures for B, L

(3) During factorization, if a prospective pivot is tiny make it bigger (rare).

Changes A to A + rank1 matrix

Can solve A using factorization for A + low rank using Sherman-Morrison, GMRES from Chap6 (iterative refinement)

Lots of algorithms, software,
Survey of available software, link on web page.

Structured Matrices
("data sparse")

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

Many Structures (depends on physics, ...)

Look at one common case:

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

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

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

constant along diagonals

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

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

$\Rightarrow$ polynomial interpolation

$O(n^2)$ using Newton Interpolation

similar trick works for $V^T z = b$

Eg Multiplying Tz same as convolution, use FFT

Eg: Solving $Cx = b$, arises in rational interpolation

Common Structure of all these X

$$AX + XB = \text{low rank, for some simple } A, B$$

Def: this rank called "displacement rank"

Ex: Vandermonde V

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

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

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

$D \cdot V - VP$ = zero except last column

= rank 1

Ex: Toeplitz T

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

= nonzero only in first row, last column

= rank 2

Ex: Cauchy C

$\mathrm{diag}(x_1 \dots x_n) \cdot C + C \cdot \mathrm{diag}(y_1, \dots, y_n)$
= all ones = rank 1

Thm (Kailath et al) There are $O(n^2)$ solvers if displacement rank is $O(1)$
(stability not guaranteed)

Chap 3: Least Squares

Ex: polynomial fitting.

given sample points (y_i, b_i), $i=1:m$
find "best" polynomial $p(y)$ of fixed degree to minimize

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

minimize $\|Ax-b\|$

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

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

(polyfit31)

Standard notation:

$\arg\min_x \|Ax-b\|_2$ $\quad A^{m \times n} \quad m>n$

$m>n$ means overdetermined
don't expect $Ax=b$ exactly

Other variants (all in LAPACK)

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

where #rows(B) ≤ #cols(A) ≤ #rows(A) + #rows(B)

answer unique if A, B full rank

Weighted LS $\underset{x}{\operatorname{argmin}} \|y\|_2$ s.t. $b = Ax + By$

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

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

Underdetermined: #rows(A) < #cols(A)

or if A not full column rank ⇒ space of solutions (add any z s.t. $Az=0$ to a solution, get another)

To make solution unique:

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

Ridge Regression (Lecture 1):

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

$\lambda > 0$ tuning parameter

always unique solution if $\lambda > 0$

Total Least Squares:

$$\underset{x:\ (A+E)x=b+r}{\operatorname{argmin}} \|[E,r]\|_2$$

Algorithms for overdetermined LS
(all used in different cases)

Solve Normal Equations (NE)

$A^T A x = A^T b$ (real case)

$A^T A$ spd. $\Rightarrow$ Cholesky

fastest in dense case (fewest flops,
least comm.)

not stable if A ill-conditioned

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

$Q^{m \times n}$ orthonormal columns

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

Gram-Schmidt - unstable if A ill-cond

Householder - stable ($x = A \backslash b$ in Matlab)

blocked Householder - reduces comm.

(possible to get QR via NE,
called Cholesky QR, fast, but can be

as unstable as NE)

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

Convert to a square linear system, matrix contains A, A^T, allows exploitation of sparsity (also possible to do sparse QR) (see Q 3.3)

Normal Equations:

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

pf: Assume x satisfies NE

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

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

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

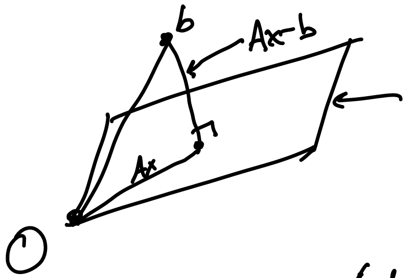
$= (\underline{Ax-b} + \underline{Ae})^T(\underline{Ax-b} + \underline{Ae})$

$= (\underline{Ax-b})^T(\underline{Ax-b}) + (\underline{Ae})^T(\underline{Ae})$

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

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

$\geq \|Ax-b\|_2^2$ minimized at $e=0$



all vectors $A\cdot y$

$$Ax-b \perp Ay \quad \forall y$$

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

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

$$y^T\underbrace{(A^TAx-A^Tb)}_{=0}=0 \quad \forall y$$