Welcome to MA 221! (Feb 22)

GE + Cholesky on sparse matrices
  How to pick order of rows/cols
to minimize memory/#flats
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

\[ V = V_1 \cup V_2 \cup V_3 \] (disjoint)

\[ \uparrow \text{separator} \]

1. \(|V_1| \approx |V_2|\)
2. \(|V_3| \text{ much smaller}\)
3. \(\text{no edges from } V_1 \text{ to } V_2\)
Order vertices in $V_1$ first, then $V_2$

$A = \begin{bmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \Rightarrow L = \begin{bmatrix} L_{11} & 0 & 0 \\ 0 & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$

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

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

Apply to 7x7 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, Schwarz, 2009)
#words moved between main memory and cache for LU or Cholesky is
\[ \Omega \left( \frac{\text{#flops}}{VM} \right) \quad M = \text{cache size} \]
For Cholesky on nxn mesh \[ \Omega \left( \frac{n^3}{VM} \right) \]

Thm. (David, D., Grigor, Peyronnet, 2010)
Attainable by ND, done "carefully" bottleneck is last (largest) separator
Contrast with bandsolver:
\[ \text{#flops} = O(bw \cdot \text{dimension}) = n^4 \text{ for 2D mesh} \]
More algorithms for ND in CS267

What about 3D meshes? nxnxn
ND still good idea - mesh
\[ \text{dimension} = n^3 \]
dense Cholesky costs \( O(n^4) \)
band Cholesky costs \( O(n^3) \)
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 data structure
at every step

1. Threshold 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)
   1. 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.
   $B = D_1 \cdot AP \cdot D_2$
1B(i,j)| = 1 and 1B(i,j)| ≤ 1 i ≠ j

2. Reorder rows and columns of $B$ using same techniques as for Cholesky, keeps $|B(i,i)| = 1$
   $|B(i,j)| ≤ 1$ i ≠ j

Build data structures for $B$, $L$

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

Changes $A$ to $A+$ rank 1 matrix

Can solve $A$ using factorization for $A+$ low rank using

Sherman-Morrison, GMRES

From Chap 6 (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
Henkel: $H(i,j) = x^{i+j}$

Eq: $Vz = b$ means $\sum_i x_i^{j-1} z_i = b$
$\Rightarrow$ polynomial interpolation
$O(n^2)$ using Newton Interpolation
similar trick works for $V^T z = b$

Eq: Multiplying $Tz$ same as convolution, use FFT

Eq: Solving $Cx = b$, arises in rational interpolation
Common Structure of all these $X$

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

Det: this rank called “displacement rank”

**Ex:** Vandermonde $V$

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

$D \cdot V = V \text{ “shifted left”}$

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

$D \cdot V - V \cdot P = \text{zero except last column}$

$= \text{rank 1}$

**Ex:** Toeplitz $T$

$P \cdot T - T \cdot P = T \text{ shifted down}$

$= T \text{ shifted left}$

$= \text{nonzero only in first row, last column}$

$= \text{rank 2}$

**Ex:** Cauchy $C$

$\text{diag}(x_1, \ldots, x_n) \cdot C + C \cdot \text{diag}(y_1, \ldots, y_n)$

$= \text{all ones} = \text{rank 1}$
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 \( \| A\mathbf{x} - \mathbf{b} \|_2 \)

\[
A(i,j) = y_i x_j^{-1}, \ \mathbf{x} \text{ are coeffs of } p
\]

\[
p(y) = x_1 + x_2 y + x_3 y^2 + \cdots + x_d y^{d-1}
\]

(\text{polyfit31})

Standard notation!

\[
\arg \min_{\mathbf{x}} \| A\mathbf{x} - \mathbf{b} \|_2 \quad A^{m \times n} \quad m > n
\]

\( m > n \) means overdetermined

Don't expect \( A\mathbf{x} = \mathbf{b} \) exactly.
Other variants (all in \texttt{LAPACK})

Constrained LS: $\quad \arg\min_{x: Bx=y} \|Ax-b\|_2$

where $\#\text{rows}(B) \leq \#\text{cols}(A) \leq \#\text{rows}(A) + \#\text{rows}(B)$

answer unique if $A, B$ full rank

Weighted LS $\quad \arg\min_{x, y} \|y\|_2$ s.t. $b = Ax + By$

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

if $B$ square, nonsingular $\quad \arg\min_{x} \|B(Ax-b)\|_2$

Underdetermined: $\#\text{rows}(A) < \#\text{cols}(A)$ or if $A$ not full column rank $\Rightarrow$

space of solutions (add any $\xi$ s.t. $A\xi = 0$ to a solution, get another $\xi$)

To make solution unique:

$\arg\min_{x} \|x\|_2$ s.t. $Ax = b$

Ridge Regression (Lecture 1):

$\arg\min_{x} \|Ax-b\|_2^2 + \lambda \|x\|_2^2$

$\lambda > 0$ tuning parameter

always unique solution if $\lambda > 0$
Total Least Squares:
\[ \arg \min_x \| [E, r] x \|_2 \]
\[ x : (A+E)x = b + r \]

Algorithms for overdetermined LS
(all used in different cases)

Solve Normal Equations (NE)
\[ A^T A x = A^T b \] (real case)
\[ A^T A \text{ spd. } \Rightarrow \text{Cholesky} \]
fastest in dense case (fewest flops, least comm.)
not stable if \( A \) ill-conditioned

Use QR decomposition \( A^{m \times n} = Q R \)

\( Q^{m \times n} \) 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 $\text{NE}$

SVD: most "complete" solution, requires 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 $A^T Ax = A^T b$ (NE) minimizes $\|Ax - b\|^2$

pf: Assume $x$ satisfies NE

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

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

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

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

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

$+ 2e^T A^T (Ax - b)$

$= 0$ by NE

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

$\geq \|Ax - b\|^2$ minimized at $e = 0$
all vectors $A \cdot y$

$A x - b \perp A y \quad \forall y$

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

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

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

$= 0$