Lecture 12
Stability of LU, Cholesky Factorization

MIT 18.335J / 6.337J

Introduction to Numerical Methods

Per-Olof Persson

October 19, 2006

Stability of LU without Pivoting

- For $A = LU$ computed without pivoting:

$$\tilde{L}\tilde{U} = A + \delta A, \qquad \frac{\|\delta A\|}{\|L\|\|U\|} = O(\epsilon_{\text{machine}})$$

- Measures the error in $\tilde{L}\tilde{U}$, not in $\tilde{L}$ or $\tilde{U}$
- Note: $\|L\|\|U\|$ in denominator, not $\|A\|$
- $\|L\|$ and $\|U\|$ can be arbitrarily large, consider e.g.

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

- Therefore, the algorithm is *unstable*

Stability of LU with Pivoting

- When pivoting, all entries of L are ≤ 1 in magnitude, so $\|L\| = O(1)$
- To measure the growth in U, introduce the *growth factor*

$$\rho = \frac{\max_{i,j} |u_{ij}|}{\max_{i,j} |a_{ij}|}$$

which implies $\|U\| = O(\rho\|A\|)$

- We then have for $PA = LU$ computed with pivoting:

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\rho\epsilon_{\text{machine}})$$

- If $\rho = O(1)$, then the algorithm is backward stable

The Growth Factor

- Consider the matrix

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ -1 & -1 & -1 & 1 & \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & 1 \\ & 1 & & & 2 \\ & & 1 & & 4 \\ & & & 1 & 8 \\ & & & & 16 \end{bmatrix}$$

- No pivoting occurs, so this is the $PA = LU$ factorization
- Growth factor $\rho = 16 = 2^{m-1}$ (can be shown to be the worst case)
- Therefore, $\rho \leq 2^{m-1} = O(1)$ uniformly for all matrices of dimension m
- Backward stable according to definitions, but results might be useless
- However, for some reason growth factors are always small in practice

SPD Matrices

- Reminder:
  - $A \in \mathbb{R}^{m\times m}$ is *symmetric* if $a_{ij} = a_{ji}$, or $A = A^T$
  - $A \in \mathbb{C}^{m\times m}$ is *hermitian* if $a_{ij} = \overline{a_{ji}}$, or $A = A^*$
- A hermitian matrix A is *hermitian positive definite* if $x^*Ax > 0$ for $x \neq 0$
  - x^*Ax is always real since $x^*Ay = \overline{y^*Ax}$
  - *Symmetric positive definite*, or *SPD*, for real matrices
- If A is $m \times m$ PD and X has full column rank, then X^*AX is PD
  - Since $(X^*AX)^* = X^*AX$, and if $x \neq 0$ then $Xx \neq 0$ and $x^*(X^*AX)x = (Xx)^*A(Xx) > 0$
  - Any principal submatrix of A is PD, and every diagonal entry $a_{ii} > 0$
- PD matrices have positive real eigenvalues and orthogonal eigenvectors

Cholesky Factorization

- Eliminate below pivot and to the right of pivot:

$$A = \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & K - ww^*/a_{11} \end{bmatrix}$$

$$= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1$$

where $\alpha = \sqrt{a_{11}}$

- $K - ww^*/a_{11}$ is principal submatrix of PD matrix $R_1^{-*}AR_1^{-1}$, therefore its upper-left entry is positive

Cholesky Factorization

- Apply recursively to obtain

$$A = (R_1^* R_2^* \cdots R_m^*)(R_m \cdots R_2 R_1) = R^* R, \qquad r_{jj} > 0$$

- Existence and uniqueness: Every PD matrix has a unique Choleskey factorization
  - Recursive algorithm from previous slide never breaks down
  - Also shows uniqueness, since $\alpha = \sqrt{a_{11}}$ is given at each step, and then the entire row w^*/α is given

The Cholesky Factorization Algorithm

- Factorize hermitian positive definite $A \in \mathbb{C}^{m\times m}$ into $A = R^* R$:

Algorithm: Cholesky Factorization

$R = A$
for $k = 1$ **to** m
  for $j = k+1$ **to** m
    $R_{j,j:m} = R_{j,j:m} - R_{k,j:m} \overline{R}_{kj} / R_{kk}$
  $R_{k,k:m} = R_{k,k:m} / \sqrt{R_{kk}}$

- Operation count

$$\sum_{k=1}^{m} \sum_{j=k+1}^{m} 2(m-j) \sim 2 \sum_{k=1}^{m} \sum_{j=1}^{k} j \sim \sum_{k=1}^{m} k^2 \sim \frac{m^3}{3}$$

Stability

- The computed Cholesky factor $\tilde{R}$ satisfies

$$\tilde{R}^* \tilde{R} = A + \delta A, \qquad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

that is, the algorithm is backward stable

- But the forward errors in $\tilde{R}$ might be large (like for QR Householder), $\|\tilde{R} - R\|/\|R\| = O(\kappa(A)\epsilon_{\text{machine}})$
- Solve $Ax = b$ for positite definite A by Cholesky and 2 back substitutions
  - Operation count $\sim$ Cholesky $\sim m^3/3$
  - Backward stable algorithm:

$$(A + \Delta A)\tilde{x} = b, \qquad \frac{\|\Delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})$$

Backslash in MATLAB

- `x=A\b` for dense `A` performs these steps (stopping when successful):

1. If `A` is upper or lower triangular, solve by back/forward substitution
2. If `A` is permutation of triangular matrix, solve by permuted back substitution (useful for `[L,U]=lu(A)` since `L` is permuted)
3. If `A` is symmetric/hermitian
   - Check if all diagonal elements are positive
   - Try Cholesky, if successful solve by back substitutions
4. If `A` is Hessenberg (upper triangular plus one subdiagonal), reduce to upper triangular then solve by back substitution
5. If `A` is square, factorize $PA = LU$ and solve by back substitutions
6. If `A` is not square, run Householder QR, solve least squares problem