CS475 / CS675
Lecture 12: June 9, 2016

Householder reflections and

Givens rotations

Reading: [TB] Chapter 10

Householder QR Factorization algorithm (full version)

For $k = 1,2,\dots,n$

$x = A(k{:}m,k)$

$v_k = x + sign(x_1)\left|\left|x\right|\right|_2 e_1$

$v_k = v_k/\left|\left|v_k\right|\right|$

(construct Q_k from v_k)

for $j = k, k+1,\dots,n$

$A(k{:}m,j) = A(k{:}m,j) - 2v_k(v_k^T A(k{:}m,j))$

end

(apply Q_k to cols a_j)

End

(Notation: $A(k{:}m,j) = j^{th}$ col of A from row k to row m)

Householder QR factorization

- Notes:
  1. At the end of algorithm, A is reduced to R
  2. Q is not constructed. In fact only v_k's are kept

     Note $Q^T = Q_n \dots Q_2 Q_1$

     $Q = Q_1 Q_2 \dots Q_n$ $\qquad (Q_k = Q_k^T)$

- To compute $Q^T b$:

  for $k = 1,2, \dots, n$

  $$b(k{:}m) = b(k{:}m) - 2v_k \left(v_k^T b(k{:}m) \right)$$

  end

Householder QR factorization

- To compute Qx:

  for $k = n, n-1, \ldots, 1$

  $$x(k:m) = x(k:m) - 2v_k\left(v_k^T x(k:m)\right)$$

  end

- To compute the reduced QR, i.e., $A = \hat{Q}\hat{R}$
  - Then $\hat{Q} = [Qe_1\ Qe_2\ \ldots Qe_n]$ where $e_j = j^{th}$ col of I

Householder Factorization Complexity

- The work is dominated by the inner-most loop:

$$A(k{:}m, j) = A(k{:}m, j) - 2v_k(v_k^T A(k{:}m, j))$$

$$flops\left(v_k^T A(k{:}m, j)\right) \sim 2(m - k + 1)$$

$$flops\left(2v_k\left(v_k^T A(k{:}m, j)\right)\right) \sim m - k + 1$$

$$flops(subtraction) \sim m - k + 1$$

$$\therefore\ subtotal \sim 4(m - k + 1)$$

Householder Factorization Complexity

- The inner loop is done $n - k + 1$ times (j-loop)
  - i.e., $flops = 4(m - k + 1)(n - k + 1)$

- Total flops $= \sum_{k=1}^{n} 4(m - k + 1)(n - k + 1)$

$$\sim 2mn^2 - \frac{2}{3}n^3$$

- When $m = n$, $flops(QR) \sim \frac{4}{3}n^3 = 2 \times flops(LU)$
- It does not include the computation of Q

Example

Givens Rotations

- Turn non-zero entries into zeros more selectively
- Givens rotations have the form:

$$G(I, k, \theta)^T =$$

Givens Rotations

- Easy to check that $G(i,k,\theta)$ is orthogonal
- Consider $y = G(i,k,\theta)^T x$. Then

$$y_j = \begin{cases} cx_i - sx_k & j = i \\ sx_i + cx_k & j = k \\ x_j & j \neq i,k \end{cases}$$

To make $y_k = 0$, let $c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}$, $s = -\frac{x_k}{\sqrt{x_i^2 + x_k^2}}$

- Notes
  1. 5 flops to compute c & s
  2. θ is not needed
  3. When computing $G^T(i,k,\theta)A$, only rows i, k affected

Example

Givens QR factorization

- Big picture

Givens QR method

- Let $G_j = j^{th}$ Givens rotation. Then

$$G_k^T \dots G_2^T G_1^T A = R$$

$$A = QR \qquad Q = G_1 G_2 \dots G_k$$

- $flops(Givens\ QR) = 3mn^2 - 3n^3$

$$= 1.5 \times flops(Householder\ QR)$$

Hessenberg QR via Givens

- A Hessenberg matrix has the form:

- i.e., $G_{n-1}^T G_{n-2}^T \dots G_1^T A = R$

  $A = QR$ where $Q = G_1 \dots G_{n-1}$

- $flops(QR) \sim 3n^2$