MTH 9821 - L8

2017/10/26

Why LU decomposition is faster than Cholesky decomposition?

1) Cholesky main diagonal $\Rightarrow$ need square computation.

   LU main diagonal $\Rightarrow$ do not need square computation. use "1".

2). LU decomposition main diagonal are "1", save computation of division.

3). But Cholesky has a gain: when update a tridiagonal matrix. only need to update the diagonal one.

$$\left[\begin{array}{c|cccc} x & x & & & \\ \hline x & x & x & & \\ & x & x & x & \\ & & x & x & x \\ & & & x & x \end{array}\right] : \begin{bmatrix} x & x & & \\ x & x & x & \\ & x & x & x \\ & & x & x \end{bmatrix} - \begin{pmatrix} x \\ 0 \\ 0 \\ 0 \end{pmatrix} (x\ 0\ 0\ 0) = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix} - \begin{bmatrix} x & 0 & 0 & 0 \\ 0 & 0 & & \\ 0 & & \ddots & \\ 0 & & & \end{bmatrix}$$

Solve $Ax=b$: $Ly=b$, $Ux=y$.

LU decomposition $= 8n + O(1)$.

$[L, U] = $ lu_tridiag_spd (A) $(3n)$

$y = $ forward (L, b) $(2n)$

$x = $ backward (U, y). $(3n)$

Solve $Ax=b$.

$U = $ cholesky_tridiag (A) $(4n)$

$y = $ forward_subst (U^t, b) $(3n)$

$x = $ backward_subst (U, y) $(3n)$

Cholesky decomposition $= 10n + O(1)$.

Psuedo Code for Cholesky

$U = $ Cholesky_tridiag (A)

for $k = 1 : (n-1)$

$u(k,k) = \sqrt{A(k,k)}$; $u(k,k+1) = \dfrac{A(k,k+1)}{u(k,k)}$

$A(k+1,k+1) = A(k+1,k+1) - (u(k,k+1))^2$.

end.

$u(n,n) = \sqrt{A(n,n)}$

y= forward_subst (u^t, b)

$y(1) = b(1)/u^t(1,1)$.

for $i = 2:m$

$$y(i) = \frac{b(i) - u^t(i,i-1)\,y(i-1)}{u^t(i,i)}$$

end.

Backward Euler (for European)

for $m = 1:M$

$Au^m = b^m$

end.

$$b^m = u^{m-1} + \alpha \underbrace{\begin{pmatrix} u_0^m \\ 0 \\ \vdots \\ 0 \\ u_N^m \end{pmatrix}}_{N-3}$$

$$A = \begin{pmatrix} 1+2\alpha & -\alpha & \cdots & 0 \\ -\alpha & 1+2\alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1+2\alpha \end{pmatrix}$$

For American:

$$\begin{cases} u^m \geq \text{early\_exercise\_premium}. \\ A\tilde{u}^M \geq b^M \end{cases}$$

$\because$ Early Exercise

$\therefore$ $\tilde{u}^m$ is always greater than u^m.

Crank - Nicolson.

for $m = 1:M$.

$Au^m = \tilde{b}^m$

$$A = \begin{pmatrix} 1+2\alpha & -\frac{\alpha}{2} & & \\ -\frac{\alpha}{2} & 1+2\alpha & \ddots & \\ & \ddots & \ddots & -\frac{\alpha}{2} \\ & & -\frac{\alpha}{2} & 1+2\alpha \end{pmatrix}$$

We have to introduce an iterator solver for linear equation

Find X such that $Ax = b$. solve the recursion

Given x_0 : $x_{n+1} = R x_n + C$. for $n \geq 0$. until convergence.

Issue: (1) Whether the iteration is convergent ?

(2) If convergent, when will it convergent to x^* that $Ax^* = b$?

· Theorem:

The recurssion $x_{n+1} = Rx_n + C$ is convergent iff $\rho(R) < 1$.

$\rho(R)$ is called spectral radius of R.

$\rho(R) = \max|\lambda|$. λ: evalue of R.

<If R is bigger $\Rightarrow$ The convergence is faster>.

General splitting technique to generate recursion for solving $Ax=b$.

Let $A = M+N$. $(M+N)x = b$.

$Mx = -Nx + b$

$x = -M^{-1}Nx + M^{-1}b$.

Recurssion: $x_{n+1} = -M^{-1}Nx_n + M^{-1}b$. $\forall n \geq 0$

If $x_n \to \bar{x}$, then $A\bar{x} = b$.

1) M must be nonsingular

2) $\rho(M^{-1}N) < 1$ and smaller

3) Easy to solve $Mv = w$

$\Rightarrow$ M ideally should be diagonal matrix.

Jacobi Recurssion

Choose $M = D = \text{diag}(A(i,i))$ $i = 1:n$. $\Rightarrow A = L_A + D + U_A$

$N = L_A + U_A$

Jacobian Recursion: $x_{n+1} = -D^{-1}(L_A + U_A)x_n + D^{-1}b$. $\forall n \geq 0$.

Residual-Based Convergence Criterion

Stop iteration when $\|b - Ax_n\| \leq tol \|b - Ax_0\|$.

given A, b, x_0, tol. $r_0 = b - Ax_0$.

$r = r_0$. stop_iter_resid = tol · norm(r_0).

while norm(r) > stop_iter_resid.

$x = Rx + C$

$r = b - Ax$

end.

euclidean: all the norm are actually the same.

Consecutive Approximation Criterion.

Stop iteration when $\|X_{n+1} - X_n\| \leq tol$

Given A, b, x_0, tol. $X_{old} = x_0$; $diff = 10 \cdot tol$.

while ($\|diff\| > tol$):

$X_{new} = R \cdot X_{old} + C$; $diff = X_{new} - X_{old}$.

$X_{old} = X_{new}$;

end.

Entry-by-entry recursion

for $j = 1 : P$:

$X_{n+1}(k)$ is a better choice

$$(*) \quad X_{n+1}(j) = \frac{1}{A(j,j)} \left(\sum_{k=1}^{j-1} A(j,k) X_n(k) + \sum_{k=j+1}^{P} A(j,k) \cdot X_n(k) \right) + \frac{b(j)}{A(j,j)}.$$

end.

Jacobian Recursion (for convergence proof)

$X_{n+1} = (L+U) X_n + D^{-1} b.$ $\qquad L = D^{-1} L_A$

$X_{n+1} = R_J X_n + C_J.$ $\qquad U = D^{-1} U_A$

where $R_J = L + U$. $C_J = D^{-1} b$.

(Another choice for M besides diagonal matrix is lower triangular).

Gaussian Siedal Recursion

$X_{n+1} = -(D + L_A)^{-1} U_A X_n + (D + L_A)^{-1} b.$ $\quad (X_{n+1} = -M^{-1} N X_n + M^{-1} b)$.

$(D + L_A) X_{n+1} = -U_A X_n + b$

$D X_{n+1} = -L_A X_{n+1} - U_A X_n + b.$

$X_{n+1} = -D^{-1} (L_A X_{n+1} + U_A X_n) + D^{-1} b.$

Then: $(*)$.

Proof:

$L_A = -DL; \quad U_A = -DU$

$$x_{n+1} = -(D-DL)^{-1}(-DU)x_n + (D-DL)^{-1}b$$

$$= (I-L)^{-1}D^{-1}DUx_n + (I-L)^{-1}D^{-1}b.$$

$$x_{n+1} = (I-L)^{-1}Ux_n + (I-L)^{-1}D^{-1}b.$$

$$R_{GS} = (I-L)^{-1}U.$$

SOR (Successive Over Relaxation). <Faster than Jacobian & Gaussian Siedal),

Entry-by-Entry recursion:

for $j=1:p$:

$$x_{n+1}(j) = (1-w)x_n(j) - \frac{w}{A(j,j)}\left(\sum_{k=1}^{j-1} A(j,k)\,x_{n+1}(k) + \sum_{k=j+1}^{p} A(j,k)\,x_n(k) + \frac{w\,b(j)}{A(j,j)}\right).$$

end.

$$x_{n+1,SOR}(j) = (1-w)\,x_{n,SOR}(j) + w\,x_{n+1,GS}(j).$$

SOR Recursion: $x_{n+1} = (D+wL_A)^{-1}((1-w)D - wU_A)x_n + w(D+wL_A)^{-1}b.$

$$x_{n+1} = R_{SOR}x_n + b_{SOR}.$$

$$R_{SOR} = (I-wL)^{-1}((1-w)I + wU).$$

If $w=1 \Rightarrow$ SOR $\equiv$ Gauss Siedal.

The increase of dimension will increase the complexity, and recursion method will be more efficient

Convergence Properties Jacobi, GS, SOR.

Theorem 1: SOR convergent $\Rightarrow 0<w<2.$

Theorem 2: A spd $\Rightarrow$ SOR convergent for $0<w<2$ (thus, GS convergent)

Theorem 3: A strictly diagonally dominant

$\Rightarrow$ Jacobi and GS convergent

Proof theorem 1:

SOR convergent $\Leftrightarrow$ $\rho(R_{SOR}) < 1$.

Let $\lambda_1, \lambda_2, \cdots, \lambda_p$ are evalues of R_{SOR}. $|\lambda| < 1$. $\forall$ $i = 1:p$.

$R_{SOR} = (I - \omega L)^{-1} ((1-\omega) I + \omega U)$.

$\det(R_{SOR}) = \det((I - \omega L)^{-1}) \cdot \det((1-\omega) I + \omega U)$. $= (1-\omega)^n = \det(R_{SOR}) = \prod_{i=1}^{p} \lambda_i$

$\prod_{i=1}^{p} |\lambda_i| = |1-\omega|^n$

SOR convergent $\Rightarrow$ $|\lambda_i| < 1$ $\forall$ $i = 1 \cdots p$.

$\Rightarrow$ $\prod_{i=1}^{p} |\lambda_i| < 1$

$\Rightarrow$ $|1-\omega|^n < 1$ $\Rightarrow$ $|1-\omega| < 1$ $\Rightarrow$ $0 < \omega < 2$.