SI 211: Numerical Analysis Homework 6 solution

Prof. Boris Houska

Due on Nov , 2020, 23:59

1. Matrix-Norm Inequality
   step1: proof matrix $I-A$ is invertible(5 points) Assume $I-A$ is singular, then

   $$(I-A)x = 0$$

   has non-zero solution x, and x=Ax, then $0 \leq \|x\| = \|Ax\| \leq \|A\|\|x\|$, which means $\|A\| \geq 1$

   $$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \geq 1$$

   ,which is a contradiction.
   step2(5 points) notice $\|I\| = 1$,

   $$I = (I-A)(I-A)^{-1} = (I-A)^{-1} - A(I-A)^{-1}$$

   $$1 = \left\|(I-A)^{-1} - A(I-A)^{-1}\right\| \geq \left\|(I-A)^{-1}\right\|(1 - \|A\|)$$

   then

   $$\|(I-A)^{-1}\| \leq \frac{1}{1-\|A\|}$$

2. Condition number
   solution $x1 = 1, x2 = 1, \bar{x1} = 1.5, \bar{x2} = 0.5$

   $$\operatorname{cond}_\infty(A) = \|A\|_\infty \cdot \left\|A^{-1}\right\|_\infty = 4 \cdot = 10000$$
   $$\frac{\|\delta x\|_\infty}{\|x\|_\infty} = \frac{\frac{1}{2}}{1} = \frac{1}{2}$$
   $$\frac{\|\delta b\|_\infty}{\|b\|_\infty} = \frac{2 \cdot 10^{-4}}{4} = 5 \cdot 10^{-5}$$

   the solution shows that when the right hand side vector is changed very small(1e-4), the change of solution is relative big(0.5), and the perturbation of solution is bounded by the condition number of coefficient matrix, that is,

   $$\frac{\|\delta x\|_\infty}{\|x\|_\infty} = \operatorname{cond}_\infty(A)\left(\frac{\|\delta b\|_\infty}{\|b\|_\infty}\right) = \frac{1}{2}$$

   this case can show the equality can be reached.

3. Gaussian elimination
   solution :

   $$x = [3 \quad 2 \quad 1]^\top$$

   Julia code :

```
using LinearAlgebra
function LU(A)
    m,n = size(A)
    B=A
    #store L, U in the matrix A
    P=Matrix{Float64}(I, m, m)
    #L=Matrix{Float64}(I, m, m)
    for i in 1:n-1
        #Column principal LU decompose
        index=argmax(A[i:n,i])
        A[i,:],A[index+i-1,:]=A[index+i-1,:],A[i,:]
        P[i,:],P[index+i-1,:]=P[index+i-1,:],P[i,:] #print(P)
        if A[i,i]==0
            print("submatrix should be full rank")
        else
            A[i+1:n,i]=1/A[i,i]*A[i+1:n,i]#update L
            #update U
            A[i+1:n,i+1:n]=A[i+1:n,i+1:n]-A[i+1:n,i]*reshape(A[i,i+1:n],(1,n-i))
    #print(A)
        end
    end
    return A,P
end
A=[1.0 -2.0 1.0;2.0 1.0 -3.0;4.0 -7.0 1.0]
b=[0.0 5.0 -1.0]
A,P=LU(A)
```

```
function L_slover(L,b)
    m,n=size(L)
    if m==n
        for j in 1:m-1
                b[j]=b[j]/L[j,j]
                b[j+1:m]=b[j+1:m]-b[j]*L[j+1:n,j]
        end
            b[m]=b[m]/L[m,m]
        else
        print("L should be squared")
    end
        return b
end

function U_slover(U,y)
    m,n=size(U)
    if m==n
        for j in m:-1:2
                y[j]=y[j]/U[j,j]
                y[1:j-1]=y[1:j-1]-y[j]*U[1:j-1,j]
        end
            y[1]=y[1]/U[1,1]
        else
        print("L should be squared")
    end
        return y
end
```


Figure 1: Caption

```
end
function slover(A,b)
    A,P=LU(A)
    m,n=size(A)
    L=LowerTriangular(A)
    for i in 1:m
        L[i,i]=1
    end
    U=UpperTriangular(A)
    y=L_slover(L,P*b)
    x=U_slover(U,y)
    print(x)
end
solve(A,b)
#[3,2,1]
```

Figure 2: Caption

4. Band-Structured Matrices
   if A is a symmetrical positive Band-Structured Matrices with Band width 2n+1 then its Cholesky decomposition factor L is also a Band-Structured Matrices, what is the Bandwidth of L?
   solution: the bandwith is n+1
   use the poporties:
   when $|i-j| > n, a_{ij} = 0$ then change the $L^T L$ from the code, it is easy to see this conclusion. solution1(by $L^T L$):

for $\quad k = 1 : n$

$$A(k,k) = \sqrt{A(k,k)}$$
$$n1 = \min(n, k+m)$$
$$A(k+1:n1,k) = A(k+1:n1,k)/A(k,k)$$

for $\quad j = k+1 : n1$

$$A(j:n1,j) = A(j:n1,j) - A(j:n1,k)A(j,k)$$

end

solution2(provided by most student:) Consider the general case: $A \in R^{n \times n}$, we can partition A as follows

$$\boldsymbol{A} = \begin{bmatrix} \alpha & \boldsymbol{v}^T \\ \boldsymbol{v} & \boldsymbol{B} \end{bmatrix}, \alpha \in R, \boldsymbol{v} \in R^{n-1}, \boldsymbol{B} \in R^{(n-1)\times(n-1)}$$

Now the Cholesky factorization of A can be represented as

$$\boldsymbol{R} = \begin{bmatrix} \beta & \boldsymbol{v}^T/\beta \\ \boldsymbol{v}/\beta & \boldsymbol{R}_1 \end{bmatrix}, \beta = \sqrt{\alpha}$$

Suppose A has semi-bandwidth d, then $v^T = \left[u^T, 0^T\right]$ in (1), where $u \in R^d$, and therefore $C := \boldsymbol{B} - \boldsymbol{v}\boldsymbol{v}^T/\alpha$ differs from $\boldsymbol{B}$ only in the upper left $d \times d$ corner. So $\boldsymbol{C}$ has the same semi-bandwidth as B and A. By induction on n, R in (2) has the same semi-bandwidth as A. Back to our case, A is a symmetric positive matrix with bandwidth $2n+1$, which indicates the semi-bandwidth is n, by the proof above, the Cholesky factor L also has semi-bandwidth n, since L is lower triangular matrix, the bandwidth of L is $n+1$
solution3;(By $L^T DL$)

$$A = LDL^T$$

The general recursion (for $i > j$):

$$D_{j,j} = A_{j,j} - \sum_{k=1}^{j-1} L_{j,k}^2 D_k \quad \text{and} \quad L_{i,j} = \frac{1}{D_j}\left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k} L_{j,k} D_k\right)$$

First, we check the first column of A, we get $A_{i,1} = 0$ for $i \geq n+2$. So

$$L_{i,1} = A_{i,1}/D_1 = 0, \text{ for } n+2 \leq i \leq m$$

Similarly, we check the second column, and get

$$L_{i,2} = \tfrac{1}{D_2}\left(A_{i,2} - L_{i,1} L_{2,1} D_1\right) = 0, \text{ for } n+3 \leq i \leq m$$
$$\vdots$$

Using induction we can get that

$$L_{i,j} = \frac{1}{D_j}\left(A_{i,j} - \sum_{k=1}^{j-1} L_{i,k} L_{j,k} D_k\right) = 0, \text{ for } n+1+j \leq i \leq m$$

Hence, we can get the bandwidth of lower triangular matrix L is $\mathrm{n}+1$