CSCI 5304 HW 3

Jingxiang Li

October 19, 2014

Problem 1

Consider the matrices

$$A = \begin{pmatrix} 1 & -1 \\ 1 & -1.00001 \end{pmatrix} \quad B = \begin{pmatrix} 1 & -1 \\ -1 & 1.00001 \end{pmatrix}$$

What is ratio of the largest to smallest eigenvalues (in modulus) for A and for B? Show that $k_2(A) = k_2(B)$. What can you conclude about the ratio of the largest to smallest eigenvalues as a way of estimating sensitivity of a linear system? Would you consider A to be well-conditioned or ill-conditioned?

Solution

First, let's see the ratio of the largest to smallest eigenvalues (in modulus) for A and for B.

```
A = [1, -1; 1, -1.00001];
B = [1, -1; -1, 1.00001];
eig_A = eig(A);
eig_B = eig(B);
max(abs(eig_A)) / min(abs(eig_A))
% > 1.0032
max(abs(eig_B)) / min(abs(eig_B))
% > 4.0000e+05
```

Note that the ratio for A is 1.0032, which is much smaller than B's ratio $4.0000e + 05$.

Then, we show that $k_2(A) = k_2(B)$ by using SVD decomposition and `Matlab` function `cond`.

```
max(svd(A)) / min(svd(A))
% > 4.0000e+05
max(svd(B)) / min(svd(B))
% > 4.0000e+05

cond(A, 2)
% > 4.0000e+05
cond(B, 2)
% > 4.0000e+05
```

Note that both SVD and function `cond` show that $k_2(A) = k_2(B)$.

We can conclude that the ratio of the largest to smallest eigenvalues is not a robust way of estimating sensitivity of a linear system.

Further, I won't consider A to be well-conditioned for the following two reasons:

1. SVD and function `cond` show that the condition number of A is very large.
2. -1.00001 is so close to -1, and if we see it as -1 matrix A become singular, so that A is obviously ill-conditioned.

Problem 2

Problem a

Find the LU factorization of the matrix:

$$A = \begin{pmatrix} 2 & 0 & 5 & 8 \\ 0 & 2 & -1 & -3 \\ -2 & 6 & 2 & -3 \\ 4 & -4 & 0 & 2 \end{pmatrix}$$

Solution

Here we define a function `LU_nopvt` for LU decomposition without partial pivoting for square matrix.

```
%%  LU_nopvt: LU decomposition without partial pivoting using Gaussian
    Transformation
%%  Input:      A, square matrix recommended
%%  Output:     L U, L is lower-triangular, U is upper-triangular.

function [L U] = LU_nopvt(A)

    n = size(A);
    n = n(1);

    M = eye(size(A));
    L = eye(size(A));

    for k = 1 : (n - 1)
        gamma = zeros(n, 1);
        for i = (k + 1) : n
            gamma(i) = A(i, k) ./ A(k, k);
        end
        tmp = zeros(n, 1);
        tmp(k) = 1;
        M = eye(n) - gamma * tmp';
        A = M * A;
        L = L * (eye(n) + gamma * tmp');
    end

    U = A;
end
```

Then we apply function LU_nopvt to matrix A.

```
A = [2, 0, 5, 8; 0, 2, -1, -3; -2, 6, 2, -3; 4, -4, 0, 2];
[L U] = LU_nopvt(A);

L
% >  1.0000         0         0         0
% >  0          1.0000         0         0
% > -1.0000    3.0000    1.0000         0
% >  2.0000   -2.0000   -1.2000    1.0000

U
% >  2.0000         0    5.0000    8.0000
% >       0    2.0000   -1.0000   -3.0000
% >       0         0   10.0000   14.0000
% >       0         0         0   -3.2000
```

Problem b

Find the $PA = LU$ factorization of A using partial pivoting.

Solution

```
[L,U,P] = lu(A);

L
% >  1.0000         0         0         0
% > -0.5000    1.0000         0         0
% >  0.5000    0.5000    1.0000         0
% >  0         0.5000   -0.5000    1.0000

U
% > 4     -4      0      2
% > 0      4      2     -2
% > 0      0      4      8
% > 0      0      0      2

P
% > 0      0      0      1
% > 0      0      1      0
% > 1      0      0      0
% > 0      1      0      0
```

Problem c

What is the determinant of A?

Solution

Given $PA = LU$, we have

$$det(A) = \frac{det(L)det(U)}{det(P)}$$

it's easy to see that $det(L) = 1$, $det(U) = \prod_{i=1}^{4} U_{ii} = 128$, $det(P) = -1$, so that

$$det(A) = 128/-1 = -128$$

Problem d

Using the LU factors obtained in (a) find the second column of the inverse of A, without computing the whole inverse.

Solution

Note that

$$A^{-1} \cdot (0,1,0,0)^T = A_2^{-1}$$

then let $x = A_2^{-1}$, $b = (0,1,0,0)^T$, We have $Ax = b$, i.e. $LUx = b$. To obtain x, we first solve $Ly = b$, then $Ux = y$.

We first define two functions `Forw_sub` and `Back_sub` for forward substitution and backward substitution.

```
%% Back_sub: Backward substitution to solve Ax = b where A is upper-triangular
%% Input: A, b
%% Output: x
function [x] = Back_sub(A, b)
    n = length(b);
    foo = 0;

    for i = n : -1 : 2
        for j = (i - 1) : -1 : 1
            foo = A(j, i) ./ A(i, i);
            A(j, :) = A(j, :) - foo * A(i, :);
            b(j) = b(j) - foo * b(i);
        end
    end

    x = b ./ diag(A);
end
```

```
%%  Forw_sub: Forward substitution to solve Ax = b where A is lower-triangular
%%  Input: A, b
%%  Output: x
function [x] = Forw_sub(A, b)
    n = length(b);
    foo = 0;

    for i = 1 : (n - 1)
        for j = (i + 1) : n
            foo = A(j, i) ./ A(i, i);
            A(j, :) = A(j, :) - foo * A(i, :);
            b(j) = b(j) - foo * b(i);
        end
    end

    x = b ./ diag(A);
end
```

Then we apply `Forw_sub` and `Back_sub` to the LU decomposition result, to obtain the second column of the inverse of A.

```
[L U] = LU_nopvt(A);
b = [0, 1, 0 ,0]';
y = Forw_sub(L, b);
x = Back_sub(U, y);

x
% >  0.5000
% >  0.7500
% > -1.0000
% >  0.5000
```

Problem 3

Problem a

Show that if A is Symmetric Positive Definite (SPD) then $Trace(AX) > 0$ for all SPD matrices X.

Solution

We calculate Cholesky decomposition for A and X, say $A = L_A L_A^T$, $X = L_X L_X^T$, then

$$Trace(AX) = Trace(L_A L_A^T L_X L_X^T) = Trace(L_A^T L_X L_X^T L_A)$$

let $M = L_A^T L_X$, then

$$Trace(AX) = Trace(MM^T)$$

Let $M = (M_1, M_2, M_3, M_4)^T$, then

$$Trace(MM^T) = \sum_{i=1}^{4}(||M_1||_2^2) > 0$$

Q.E.D.

Problem b

Show that if $Trace(AX) \geq 0$ for all Symmetric Positive Semi-Definite (PSD) matrices X then A is PSD.

Solution

We first decompose $X = LL^T$, then

$$Trace(AX) = Trace(ALL^T) = Trace(L^T AL)$$

L is a lower-triangular matrix, Let $L = (L_1, \ldots, L_n)$. Then

$$Trace(AX) = L_1^T AL_1 + L_2^T AL_2 + \cdots + L_3^T AL_3 \geq 0$$

We will then prove it by contradiction. Suppose A is not PSD, then $\exists x \;\; s.t. \;\; x^T Ax < 0$. then we can construct $L^* = (x, 0, 0, \ldots)$ and let $X^* = L^* L^{*T}$, then $Trace(AX^*) = x^T Ax < 0$, contradiction. So that A is PSD.

Q.E.D.