CS3081 Computational Mathematics

Séamus Woods
15317173

14/03/2019

0.1 Question 4.23

Question: Write a user-defined MATLAB function that decomposes an n x n matrix [A] into a lower triangular matrix [L] and an upper triangular matrix [U] (such that [A] = [L][U]) using the Gauss elimination method (without pivoting). For the function name and arguments, use [L, U] = LUdecomp-Gauss (A), where the input argument A is the matrix to be decomposed and the output arguments L and U are the corresponding upper and lower triangular matrices. Use LUdecompGaus s to determine the LU decomposition of the following matrix:

$$\begin{bmatrix} 4 & -1 & 3 & 2 \\ -8 & 0 & -3 & -3.5 \\ 2 & -3.5 & 10 & 3.75 \\ -8 & -4 & 1 & -0.5 \end{bmatrix}$$

Answer: My MATLAB code for calculating this can be seen below.

```
% Creating function LUdecompgauss()
function [L,U] = LUdecompgauss(A)
% Getting dimensions of matrix
[m, n] = size(A);
% Making sure matrix is square..
if (m ~= n)
     disp("Square matrices only");
end
% Creating template for lower triagnular matrix..
L = zeros(m);
for i = 1:m
   L(i,i) = 1;
end
% Creating copy of matrix..
U = A;

depth = 1;
for i = 1:m
     for j = 1:n
          if i == j
```

```
            temp = depth;
            while temp < m
                temp = temp + 1;
                constant = U(temp,j)/U(i,j);
                for x = 1:m
                    U(temp,x) = U(temp,x) - (constant*U(i,x));
                end
                L(temp,j) = constant;
            end
        end
    end
    depth = depth + 1;
end

disp(L);
disp(U);
```

This code outputs:

$$\begin{array}{cc} [L] & [U] \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 0.5 & 1.5 & 1 & 0 \\ -2 & 3 & -0.5 & 1 \end{bmatrix} & \begin{bmatrix} 4 & -1 & 3 & 2 \\ 0 & -2 & 3 & 0.5 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix} \end{array}$$

0.2 Question 5.17

A football conference has six teams. The outcome of the games is recorded in a binary fashion. For example, ifteam 1 defeats teams 5 and 6, then the equation x1 = x5 + x6 is written to indicate these results. At the end of the season, the wins and loses are tabulated in this fashion to produce the following ranking matrix:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

(a) Find the eigenvalues and the corresponding eigenvectors of [A], using MATLAB's built-in function *eig*.

(b) Find the eigenvector from part (a) whose entries are all real and of the same sign (it does not matter if they are all negative or all positive), and rank the teams from best (i.e., with most win) to worst (i.e., with fewest wins) based on the indices of the teams corresponding to the largest to the smallest entries in that eigenvector.

Answer:
(a) Eigenvalues:
2.6180 + 0i
0.3820 + 0i
0 + 0i
-1 + 0i
-1 - 0i
-1 + 0i

(b) Eigenvector whose entries are all real and of the same sign:
0.1761 + 0i
0.5155 + 0i
0.3938 + 0i
0.4611 + 0i
0.5155 - 0i

```
0.2642 + 0i
```

So the best teams in order from best to worst:
1. team2
2. team5
3. team4
4. team3
5. team6
6. team1