Experiment 2

Google's Mechanism for ranking WebPages (Random Surfer)

Aim:

To Understand the Random Surfer Algorithm which was used initially in Google's Search Engine and was developed by Lawrence Page (and Sergey Brin) using MATLAB.

Mathematical Background:

According to Google:

"PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites." (Term is Citation)

For a web of pages A, B, C, D,… the PageRank of A is given by:

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

PR(A) denotes PageRank of A and L() denotes the total number of outgoing links. In our experiment d = 1. (Generally dampning factor, d = 0.85). Links from a page to itself (known as self-citation) are ignored and multiple outgoing links to a single page are counted as 1 only.

Solving for the PR() equations(linear system) involves the use of Matrices and further the use of Eigen Values and Eigen Vectors. The system of these linear equations conclude to AX = X. Hence, we can say that X is a eigen vector corresponding to eigen value 1. This X eigen vector will give us a non-trivial Solutions.

Code:

```
%Pagerank Algorithm - 16BCE0783
A = input('Input the (Transition)matrix: ');
k = size(A);
if (k(1)~=k(2))
    fprintf('\nNot a square matrix\n\n')
    return;
end
[X,Y] = eig(A);
cor_vec = 0;
for i=1:k
    if round(Y(i,i),10) == 1
        cor_vec = i;    %cor_vec is diagonal number of eigen vector corresponding to eigen
value 1
```

```
        break;
    end
end
if cor_vec==0
    fprintf('\nNone of the eigenvalues is 1\n\n')
    return;
end
rank = X(:,cor_vec)/sum(X(:,cor_vec));  %normalization - Dividing by sum of the elements
of eigen vector corresponding to eigen value 1s
for i=1:k
    [val,pos] = max(rank);
    fprintf('\nRank %d is page %c with a Probability of %s\n',i,64+pos,num2str(val));
    rank(pos) = [-1];
end
%can use sort also [A,B] = sort()
```

Question 1: Checking for a stochastic Matrix

Code:

```
%Checking Stochastic Matrix - 16BCE0783
A = input('Enter the Matrix: ');
k = size(A);
if k(1) ~= k(2)
    disp('Not a Square Matrix!')
    return;
end
for j = 1:k(1)^2
    if A(j)<0
        disp('Given Matrix is not a stochastic Matrix since it contains a non-positive
number')
        return;
    end
end
option = input('Enter 1 to check for column stochastic or 2 for row stochastic: ');
flag = 1;
if option == 1
    for i=1:k(1)
        if sum(A(:,i))~=1
            flag = 0;
            break;
        end
    end
else
    for i=1:k(1)
        if sum(A(i,:))~=1
            flag = 0;
            break;
        end
    end
end
if flag == 1
    disp('Given Matrix is a stochastic Matrix')
else
    disp('Given Matrix is not a stochastic Matrix')
end
```

Input and Output:

```
12 -        end
```

Command Window

```
>> exp2q1
Enter the Matrix: [1/3 1/2 0;1/3 0 1;1/3 1/2 0]
Enter 1 to check for column stochastic or 2 for row stochastic: 1
Given Matrix is a stochastic Matrix
>> exp2q1
Enter the Matrix: [1/3 1/2 -1;1/3 0 1;1/3 1/2 1]
Given Matrix is not a stochastic Matrix since it contains a non-positive number
>>
```

Question 2: Getting Pageranks for the given web of pages

Code: As written in the main Code Section above

Input and Outputs:

Command Window

```
>> pagerank_algo
Input the (Transition)matrix: [0 1/3 1/3 0 1/3 0;0 0 0 1/3 1/3 1;1/3 0 0 1/3 1/3 0;1/3 1/3 1/3 0 0 0;1/3 1/3 1/3 0 0 0;0 0 0 1/3 0 0]

Rank 1 is page D with a Probability of 0.1875

Rank 2 is page E with a Probability of 0.1875

Rank 3 is page C with a Probability of 0.1875

Rank 4 is page B with a Probability of 0.1875

Rank 5 is page A with a Probability of 0.1875

Rank 6 is page F with a Probability of 0.0625
>> pagerank_algo
Input the (Transition)matrix: [0 1/3 1 1/3 0;1/2 0 0 0 0;0 1/3 0 1/3 0;1/2 0 0 0 0;0 1/3 0 1/3 0]

None of the eigenvalues is 1

>> pagerank_algo
Input the (Transition)matrix: [0 1/3 1 1/3 0;1/2 0 0 0 0;0 1/3 0 1/3 1;1/2 0 0 0 0;0 1/3 0 1/3 0]

Rank 1 is page A with a Probability of 0.33333

Rank 2 is page C with a Probability of 0.22222

Rank 3 is page D with a Probability of 0.16667

Rank 4 is page B with a Probability of 0.16667

Rank 5 is page E with a Probability of 0.11111
>>
```

-----> For Qu. 2 (b)

Error

-----> For Qu. 2 (a)