Web Mining Lab Assignment 6

## Name: Om Ashish mishra

## Registration Number: 16BCE0789

## Slot: F2

**Page Rank Algorithm:**

The Code:

import numpy as np

from scipy.sparse import csc\_matrix

def pageRank(G, s = .85, maxerr = .001):

n = G.shape[0]

# transform G into markov matrix M

M = csc\_matrix(G,dtype=np.float)

rsums = np.array(M.sum(1))[:,0]

ri, ci = M.nonzero()

M.data /= rsums[ri]

# bool array of sink states

sink = rsums==0

# Compute pagerank r until we converge

ro, r = np.zeros(n), np.ones(n)

while np.sum(np.abs(r-ro)) > maxerr:

ro = r.copy()

# calculate each pagerank at a time

for i in range(0,n):

# inlinks of state i

Ii = np.array(M[:,i].todense())[:,0]

# account for sink states

Si = sink / float(n)

# account for teleportation to state i

Ti = np.ones(n) / float(n)

r[i] = ro.dot( Ii\*s + Si\*s + Ti\*(1-s) )

# return normalized pagerank

return r/sum(r)

if \_\_name\_\_=='\_\_main\_\_':

# Example extracted from 'Introduction to Information Retrieval'

G = np.array([[0,0,1,0,0,0,0],

[0,1,1,0,0,0,0],

[1,0,1,1,0,0,0],

[0,0,0,1,1,0,0],

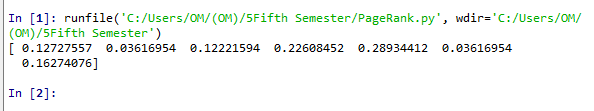
[0,0,0,0,0,0,1],

[0,0,0,0,0,1,1],

[0,0,0,1,1,0,1]])

print (pageRank(G,s=.86))

**The Output:**



Therefore Fifth one is the highest ranked; Second and Third are the smallest.

# The Method taught in lab:

**The Code:**

import numpy as np

print("Enter the number of nodes : ");

x = input()

r, c = 4, 4;

A = [[0 for x in range(r)] for y in range(c)]

i = 0

j = 0

print("Enter the connection for each node to other nodes : \n");

for i in range(0,r):

for j in range(0,c):

print("Enter the value from ",(i+1)," to ",(j+1))

A[i][j]=input()

print("The Main Matrix A is : \n")

for i in range(0,r):

for j in range(0,c):

print(A[i][j],end=" ")

print(" ")

B = [[0 for x in range(c)] for y in range(r)]

print("The Transpose of A is : \n")

for i in range(0,4):

for j in range(0,4):

B[i][j]=A[j][i]

for i in range(0,4):

for j in range(0,4):

print(B[i][j],end=" ")

print(" ")

print("Enter the Damping Factor : ")

d = float(input())

dm = [[0 for x in range(1)] for y in range(r)]

for j in range(0,r):

dm[i][0]=d

print("The Damping Factor Matrix")

for j in range(0,r):

print(dm[i][0])

print("Enter the number of Iterations : ")

itr = input()

sum1 = [[0 for x in range(1)] for y in range(r)]

for k in range(0,int(itr)):

for i in range(0,r):

for j in range(0,c):

sum1[i][0]= float(sum1[i][0] + B[i][j]\*dm[j][0])

dm[i][0] = sum1[i][0]

dm.sort()

dm = dm[::-1]

print("Page ranking")

print(dm)

**The Output:**

runfile('C:/Users/OM/(OM)/5Fifth Semester/Project/PageRank.py', wdir='C:/Users/OM/(OM)/5Fifth Semester/Project')

Enter the number of nodes :

4

Enter the connection for each node to other nodes :

Enter the value from 1 to 1

0

Enter the value from 1 to 2

0.5

Enter the value from 1 to 3

0.5

Enter the value from 1 to 4

0

Enter the value from 2 to 1

0

Enter the value from 2 to 2

0

Enter the value from 2 to 3

0

Enter the value from 2 to 4

1

Enter the value from 3 to 1

0.5

Enter the value from 3 to 2

0

Enter the value from 3 to 3

0

Enter the value from 3 to 4

0.5

Enter the value from 4 to 1

0

Enter the value from 4 to 2

0

Enter the value from 4 to 3

0

Enter the value from 4 to 4

0

The Main Matrix A is :

0 0.5 0.5 0

0 0 0 1

0.5 0 0 0.5

0 0 0 0

The Transpose of A is :

0 0 0.5 0

0.5 0 0 0

0.5 0 0 0

0 1 0.5 0

Enter the Damping Factor :

0.85

The Damping Factor Matrix

0.85

0.85

0.85

0.85

Enter the number of Iterations :

5

Page Ranking: