

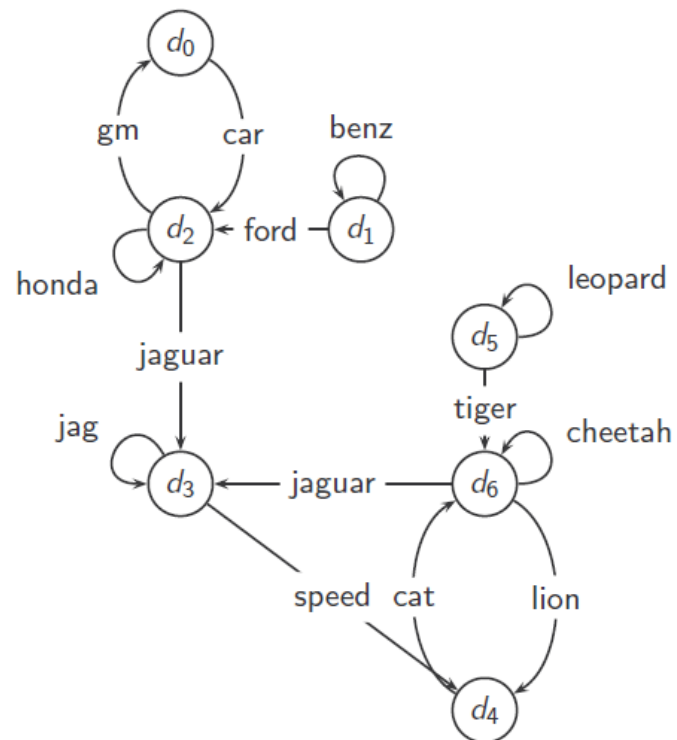
Name: Gerosh George

Reg No: 19BCE1403

29/9/2021

HITS algorithm – Lab assignment

Example web graph



PROGRAM CODE:

```
# hyperlink-Induced Topic Search

# Rows denote Hub links [Outlinks]
# Columns denote Authority links [Inlinks]

import numpy as np

# adjacency matrix for Q1
ag1 = np.matrix((
    [0,0,1,0,0,0,0],
    [0,0,1,0,0,0,0],
    [1,0,0,1,0,0,0],
    [0,0,0,0,1,0,0],
    [0,0,0,0,0,0,1],
    [0,0,0,0,0,0,1],
    [0,0,0,1,1,0,0]
))

# adjacency matrix for Q2
ag2 = np.matrix((
    [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]
))

# adjacency matrix for Q3
ag3= np.matrix((
    [0,0,1,0,0,0,0],
    [0,1,1,0,0,0,0],
    [1,0,1,2,0,0,0],
    [0,0,0,1,1,0,0],
    [0,0,0,0,0,0,1],
    [0,0,0,0,0,1,1],
    [0,0,0,2,1,0,1]
))

# logic from here

adjac_grp = ag1 # equate to the required adjacency matrix
```

```

hub_scores = np.transpose([[1]* len(adjac_grp)])
auth_scores = np.transpose([[1]* len(adjac_grp)])

#hub_scores = np.round(hub_scores / np.sum(hub_scores),2)
#auth_scores = np.round(auth_scores / np.sum(auth_scores),2)

# for norm: np.linalg.norm(hub_scores)
hub_scores = np.round(hub_scores / np.linalg.norm(hub_scores) ,3)
auth_scores = np.round(auth_scores / np.linalg.norm(auth_scores) , 3)

adjac_grp_T = np.transpose(adjac_grp)

K=25

for i in range(K):
    new_hub_scores = np.dot(adjac_grp,auth_scores) # Outlinks * authority rank or score
    new_auth_scores = np.dot(adjac_grp_T,hub_scores) # Inlinks * hub rank or score

    hub_scores = new_hub_scores
    auth_scores = new_auth_scores

    # hub_scores = np.round(hub_scores / np.sum(hub_scores),2)
    # auth_scores = np.round(auth_scores / np.sum(auth_scores),2)

    hub_scores = np.round(hub_scores / np.linalg.norm(hub_scores) ,3)
    auth_scores = np.round(auth_scores / np.linalg.norm(auth_scores) , 3)

print(f"Authority Scores: {auth_scores.T[0]}")
print(f"Hub Scores: {hub_scores.T[0]}")

```

1. Form the adjacency graph of this **IGNORING SELF LOOPS** with the following principles
 - a. **dx-dy entry is 1** if there is **alink**
 - b. **dx-dy entry is 0** if there is **no link**
 - c. Calculate the Hub score and Authority score for this graph **IGNORING SELF LOOPS** by writing a program in python. Perform 25 iterations and print out the final values of Hub score and authority score for all nodes.

OUTPUT:

```
(venv) PS C:\Users\Gerosh\Desktop\VIT\Third Year\Web Mining\Programs> python hits.py
Adjacency Graph:
[[0 0 1 0 0 0 0]]
[[0 0 1 0 0 0 0]]
[[1 0 0 1 0 0 0]]
[[0 0 0 0 1 0 0]]
[[0 0 0 0 0 0 1]]
[[0 0 0 0 0 0 1]]
[[0 0 0 1 1 0 0]]

Scores after 25 iterations
Authority Scores: [0.328, 0.0, 0.002, 0.737, 0.591, 0.0, 0.002]
Hub Scores: [0.001, 0.001, 0.591, 0.328, 0.001, 0.001, 0.737]
```

2. Form the adjacency graph of this **INCLUDING SELF LOOPS** with the following principles.
 - a. **dx-dy entry is 1** if there is **alink**
 - b. **dx-dy entry is 0** if there is **no link**
 - c. Calculate the Hub score and Authority score for this graph **INCLUDING SELF LOOPS** by writing a program in python. Perform 25 iterations and print out the final values of Hub score and authority score for all nodes.

OUTPUT:

```
(venv) PS C:\Users\Gerosh\Desktop\VIT\Third Year\Web Mining\Programs> python hits.py
Adjacency Graph:
[[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]]

Scores after 25 iterations
Authority Scores: [0.207, 0.069, 0.332, 0.665, 0.458, 0.088, 0.427]
Hub Scores: [0.137, 0.166, 0.499, 0.465, 0.177, 0.213, 0.642]
```

3. Assume the below matrix is the adjacency graph

Raw matrix A for HITS

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	2	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	2	1	0	1

NOTE: You have entries having value more than 1 –This indicates there are multiple links from d_x to d_y and all links from d_x to d_y must be considered for computing Hub score and Authority score.

Calculate the Hub score and Authority score for this graph **INCLUDING SELF LOOPS** by writing a program in python. Perform 25 iterations and print out the final values of Hub score and authority score for all nodes.

The program as well as the output should be uploaded.

OUTPUT:

```
(venv) PS C:\Users\Gerosh\Desktop\VIT\Third Year\Web Mining\Programs> python hits.py
Adjacency Graph:
[[0 0 1 0 0 0 0]]
[[0 1 1 0 0 0 0]]
[[1 0 1 2 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 2 1 0 1]]

Scores after 25 iterations
Authority Scores: [0.187, 0.022, 0.229, 0.873, 0.3, 0.023, 0.243]
Hub Scores: [0.067, 0.074, 0.637, 0.345, 0.072, 0.078, 0.674]
```