

Project II

Link

Name: 黃柏喻

Student ID: N16064674

Department: Mechanical Engineering

Chapter 1. Implementation detail

First of all, loading file is necessary. Then, I create a class named “Link”. There are three attributes: item name, item children (nodes which item points to) and item parent (nodes which point to item). Next, create a subgraph to store all those information within the big dictionary.

1.1 HITS

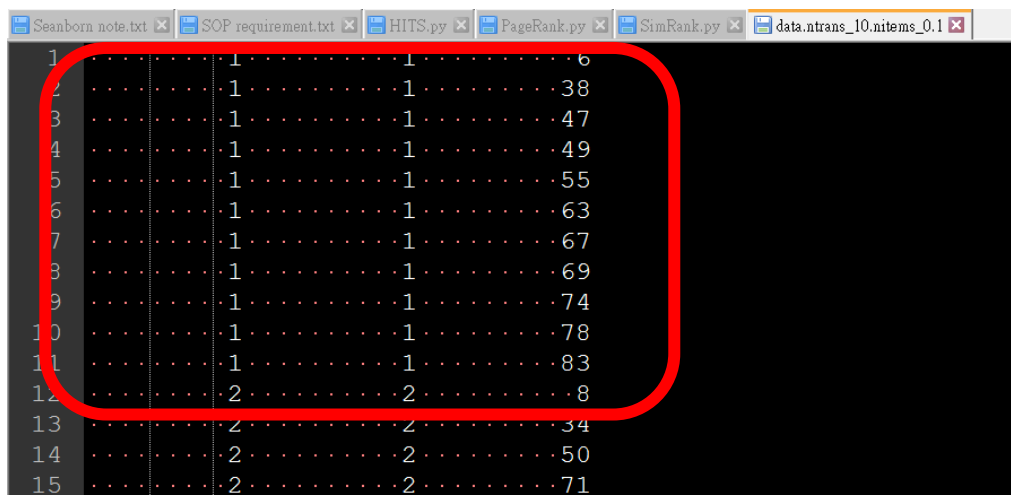
In the beginning, set the authority and hub value by 1. If parent and children of each item exist, new value will be calculated. Otherwise, it will be set to zero. After each iteration, normalization will be implemented. The stopping value is determined by adding absolute value of both hub and authority.

1.2 PageRank

In the beginning, set the PageRank value to 1 divided by number of total node. The value will be re-calculated using equation (1), which I set d value to 0.15 and n equals to the number of total node. The stopping value is determined by adding absolute value of current and previous PageRank value.

$$PR(P_i) = \frac{d}{n} + (1-d) \times \sum_{l_{j,i} \in E} \frac{PR(P_j)}{Out\ deg\ ree(P_j)} \quad \dots\dots Eq\ (1)$$

Figure 1.1 shows the IBM data I chose in HITS and PageRank algorithm. Node 6 is the center of the graph, which means it points to every other node. However, there is no connection between other nodes.



1	1	1	6
2	1	1	38
3	1	1	47
4	1	1	49
5	1	1	55
6	1	1	63
7	1	1	67
8	1	1	69
9	1	1	74
10	1	1	78
11	1	1	83
12	2	2	8
13	2	2	34
14	2	2	50
15	2	2	71

Figure 1.1 IBM data selection

1.3 SimRank

In the beginning, set original matrix to be identity matrix. Then renew the value through iteration. In equation 2, if $a = b$, $S(a,b)$ equals 1. Also, if in-neighbors of a or b are None, $S(a,b)$ equals 0.

$$S(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} S(I_i(a), I_j(b)) \quad \dots\dots \text{Eq}(2)$$

Chapter 2. Result analysis and discussion

2-1. Grpah_1

In this simple graph, the result show in Table 2.1. Since the graph is basically a straight line but not circular, I found out that hub and authority value are exactly the same.

Table 2.1 HITS and PageRank result of graph 1

	HITS	PageRank
Hub	5: 0.348	X
Authority	6: 0.348	X
PageRank	X	6: 0.103 5: 0.0927

The result of SimRank remain identity matrix.

2-2. Grpah_2

This graph is basically a circle and it might cause problems in certain method. The HITS method didn't converge while PageRank immediately converge after two iteration. The results of PageRank are all 0.2.

Table 2.2 HITS and PageRank result of graph 2

	HITS	PageRank
Hub	4: 0.334	X
Authority	1: 0.269	X
PageRank	X	ALL: 0.2

The result of SimRank remain identity matrix.

2-3. Grpah_3

This result is shown in Table 2.3. This graph is basically a symmetrical pattern which node 1 and node 2 are on the right side, and node 3 and node 4 are on the left side.

Table 2.3 HITS and PageRank result of graph 3

	HITS	PageRank
Hub	3: 0.414	X
Authority	3: 0.414	X
PageRank	X	2: 0.324 3: 0.324

The result of SimRank is shown in Table 2.4. Obviously, pair (1, 3) and pair (2, 4) bear the same idea.

Table 2.4 SimRank result of graph 3

	1	2	3	4
1	1.0	0	0.818	0
2	0	1.0	0	0.818
3	0.818	0	1.0	0
4	0	0.818	0	1.0

2-4. Grpah_4

This result is shown in Table 2.5. It is abnormal to see node 5 being the top value in both hub and authority but not the top value in PageRank. I personally think that PageRank method is slightly leaning toward Hub value in HITS method. Because node 2 which has a rather good authority value is pointed by node 1. It will definitely affect the PageRank value of node 1. Also, node 5 is benefited by node 1 too.

Table 2.5 HITS and PageRank result of graph 4

	HITS	PageRank
Hub	5: 0.445	X
Authority	5: 0.31	X
PageRank	X	1: 0.28 5: 0.184

The result of SimRank is shown in Table 2.6. I found out node 4 and node 7 are both pointed by node 1 and pointing to node 5. Moreover, node 4 and node 6 are both pointed by node 5 and pointing to node 5. Maybe it is the reason why they have the highest value.

Table 2.6 SimRank result of graph 4

	1	2	3	4	5	6	7
1	1	0.52	0.51	0.51	0.5	0.566	0.46
2	0.52	1	0.56	0.526	0.568	0.45	0.6
3	0.51	0.56	1	0.59	0.545	0.59	0.596
4	0.51	0.526	0.59	1	0.5	0.67	0.67
5	0.5	0.568	0.545	0.5	1	0.44	0.56
6	0.566	0.45	0.59	0.67	0.44	1	0.44
7	0.46	0.6	0.596	0.67	0.56	0.44	1

2-5. Grpah_5

This result is shown in Table 2.7. Clearly, the result is diversified.

Table 2.7 HITS and PageRank result of graph 5

	HITS	PageRank
Hub	468 : 0.704	X
Authority	461: 0.209	X
PageRank	X	61: 0.00287 122: 0.00283

The result of SimRank is shown in SimRank_Link.html file.

2-6. Grpah_6

This result is shown in Table 2.8. Clearly, the result is diversified.

Table 2.8 HITS and PageRank result of graph 6

	HITS	PageRank
Hub	1183: 0.459	X
Authority	709: 0.195	X
PageRank	X	1052: 0.0007 761: 0.000565

2-7. IBM

This graph demonstrate how a center node can be look like. The result is shown in Table 2.7. Apparently, node 6 is the center and it has the highest hub value. However, authority value is weird since every node other than node 6 should have the same value. Instead, node 83, which is the last link mentioned in the txt file, has the highest authority value. This flaw can be solved in PageRank method, the value of node 6 is 0, but the other node values are all the same.

Table 2.9 HITS and PageRank result of IBM transaction

	HITS	PageRank
Hub	6: 1	X
Authority	83: 0.3898	X
PageRank	X	ALL(no node 6): 0.01479

Chapter 3. Computation performance analysis

3.1 Computation time

In HITS algorithm, I set the stopping value, ϵ to 0.0001.

In PageRank algorithm, I set the stopping value, ϵ to 0.0001

Table 3.1 Time used and iteration count by each method

<i>Method</i>	graph1	graph2	graph3	graph4	graph5	graph6	IBM
<i>HITS(s)</i>	0.001	0.004	0.001	0.001	0.088	0.8	0.001
<i>HITS_iter</i>	9	99	20	18	39	64	2
<i>PageRank(s)</i>	0.0	0.0	0.0	0.001	0.006	0.014	0.001
<i>PR_iter</i>	7	1	10	11	8	6	3
<i>SimRank(s)</i>	0.001	0.001	0.001	0.003	9.64	X	X
<i>SR_iter</i>	10	10	10	10	10	X	X

3.2 Algorithm Complexity

	HITS	PageRank	SimRank
Time complexity	$O(n^2)$	$O(n^2)$	$O(n^3)$
Space complexity	$O(n)$	$O(n)$	$O(n)$

Chapter 4. Discussion

Those three methods are easy to implement. I have learned how to utilize identity matrix to calculate SimRank value easily. I did not use recursive, instead, I used “break” and “continue” inside “if” loop to better understand the method. In terms of first two algorithms, they are relatively simple and I just created a class for me to access the subgraph of each node.

I think that SimRank method is difficult to understand and the material in the handout is too little. Fully understanding of each method is the top priority before actually programming.