2019 Introduction to Massive Data Analysis

Assignment 2

Deadline: 2019/10/23 23:59

Questions: PageRank

Given a big matrix M. Specifically the column-normalized adjacency matrix where each column represents a webpage (vertex) and where it links to the non-zero entries. Write a program that calculates Google Matrix A:

$$A = \beta M + (1 - \beta)\left[\frac{1}{N}\right]_{N\times N}$$

With PageRank equation [Brin-Page, '98]

$$r_j = \sum_{i\to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

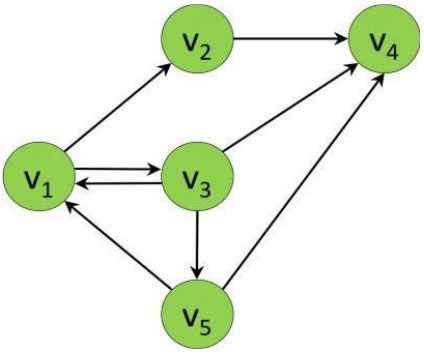
forming recursive problem: r=A· r

If M contains dead-ends, we have to renormalize r^{new} :

$$\forall j: r_j^{new} = r'^{new}_j + \frac{1-S}{N} \quad \text{where: } S = \sum_j r'^{new}_j$$

NOTE: Please set β =0.8, and initial PageRank value = 1/N in this homework.

A simple example:



V={1,2,3,4,5} and E={(1,2),(1,3),(2,4),(3,1),(3,4),(3,5),(5,1),(5,4)}

If we set β =0.8, initial PageRank value = 1/5, and run a single round of PageRank, we get the following values:

i	1	2	3	4	5
r_i^1	0.205	0.152	0.152	0.365	0.125

If we run 10 rounds of PageRank, we get the following values:

i	1	2	3	4	5
r_i^{10}	0.193	0.170	0.170	0.329	0.138

Structure:

[Mapper] A node passes its PageRank "contributions" to the nodes it is connected to.

[Reducer]Each node sums up all PageRank contributions that have been passed to it and updates its PageRank score.

Data format Input:

A file that contains one line for each link, and each line contains a pair of numbers that represent the vertices that are connected by the link.

```
1	2
1	3
2	4
3	1
3	4
3	5
5	1
5	4
```

Download here
https://snap.stanford.edu/data/p2p-Gnutella04.html

Output:

There should be one line for each vertex, and each line should contain the vertex identifier and the PageRank values.

```
4	0.329
1	0.193
2	0.170
3	0.170
5	0.138
```

Report Requirements:

a. Final output. (We require **20** iterations result)

PS. In addition, you could run processes until convergence (value at nodes no longer change) and present the result in your report.

NOTE: Please show the top ten vertices sorted by rank.

b. Explain how you design your mapper and reducer.

c. Please make sure that your .java file has the same name as your class name, which must be PageRank. If you implement the algorithm with Python, please name your .ipynb file as PageRank, too.

d. Please pack the above files into a zip file. Name it as **"MDA_HW2_studentID.zip".**