2021 Introduction to Massive Data Analysis

HW2 - PageRank

Deadline: 2021.10.27 (Wed.) 23:59

Please write a MapReduce program in Hadoop(Java) or spark(python) to solve the following question.

Question: 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]:

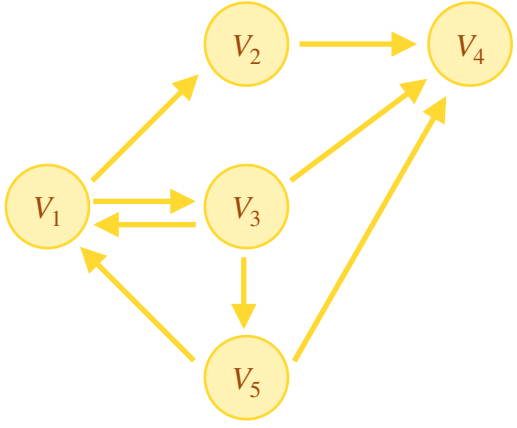
$$r_j = \sum_{i\to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

forming recursive problem: $r = A \cdot r$

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

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

Example:



$$V = \{1,2,3,4,5\}$$

$$E = \{(1,2),(1,3),(2,4),(3,1),(3,4),(3,5),(5,1),(5,4)\}$$

If we set $\beta = 0.8$, **initial PageRank value** $= \frac{1}{5}$, 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	0.125	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

Data format:

Input: "%d\t%d\n"
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
```

Output: "%d\t%f\n"
There should be one line for each vertex, and each line should be:
vertex ID, a tab, the PageRank value (round to the **third non-zero** digit after decimal point).

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

Assignment Requirements:

Please set **β = 0.8**, and **initial PageRank value = 1/N**.

Show the **top 10 vertices sorted by rank** (if ranks are equal, sorted by ID in ascending order), after **20 iterations.**

Part1 Code(80%)

Please make sure that your file has the same name as **PageRank**.
(PageRank.java or PageRank.ipynb)

Part2 Report(20%)

Java :

1. **Report.pdf** (Explain how do you design your mapper and reducer)
2. **Outputfile.txt** (write your result of **input.txt** to this file)

Python:

1. **Report.pdf** or **markdown** in .ipynb file (Explain how do you design your mapper and reducer.)
2. **Outputfile.txt** (write your result of **input.txt** to this file)

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

Should notice:

- **How to get N ?**
- (-5) Wrong output format
- (-5) Round-off error
- (-10) Cannot output the result of input-test.txt