소셜 미디어 분석 개론

Assignment1 : Pagerank algorithm

Project Due : Monday, May 16 at 23:59

The goal of this project is to use linear algebra concepts to describe Google's Page Rank algorithm. The goal of PageRank is to determine how "important" a certain webpage is. The PageRank computation models a theoretical web surfer. Given that the surfer is on a particular webpage, the algorithm assumes that they will follow any of the outgoing links with equal probability. Thus, the PageRank value of the webpage is divided equally among each of the linked webpages, raising each of their values. Using this reasoning, we end up with the following formula for computing a webpage j's PageRank:

At iteration 0, all ranks are uniformly initialized where N is the number of pages.

$$r_j = \frac{1}{N}$$,where N is the number of nodes.

and PageRank equation is

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$, where β is random teleport parameter.

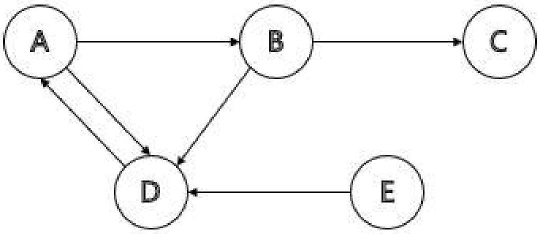
If the sum of pagerank does not equal 1, re-insert the leaked PageRank.

Use numpy and sparse matrix formulation to implement the above algorithms.

Input graph

web-Google_10k.txt : This dataset contains 10,000 web pages and 78323 links. DO NOT assume that page ids are from 0 to 10,000. Each row consists of 2 values represents the link from the web page in the 1st column to the web page in the 2nd column. For example, if the row is 0 1134, this means there is a directed link from the page id 0 to the page id 1134. The output file contains pagerank for all nodes. Each row consists of 2 values represents the ID and pagerank of the node. Arrange pagerank in descending order.

Example : Simple graph



Sparse Matrix Encoding

source node	degree	destination
A	2	B, D
B	2	C, D
C	0	
D	1	A
E	1	D

Initialize step

Let's look an example of PageRank in action. Suppose we have five web pages arranged like below. Each circle is a webpage, and the arrows represent outgoing links. After the initialize step runs, we give each page an initial rank of 0.2:

Update step : Iteration 1

We then run the update step once, to obtain the following new ranks. We update the scores of each web pages by the following. (Note: we've picked as $\beta = 0.85$).

- Page A new rank:

$$\underbrace{\beta \cdot 0.2/1}_{\text{from page D}} + \underbrace{(1-\beta)/5}_{\text{new surfers}} + \frac{1-S}{N} = 0.234$$

- Page B new rank:

$$\underbrace{\beta \cdot 0.2/2}_{\text{from page A}} + \underbrace{(1-\beta)/5}_{\text{new surfers}} + \frac{1-S}{N} = 0.149$$

■ Page C new rank:

$$\underbrace{\beta \cdot 0.2/2}_{\text{from page B}} + \underbrace{(1-\beta)/5}_{\text{new surfers}} + \frac{1-S}{N} = 0.149$$

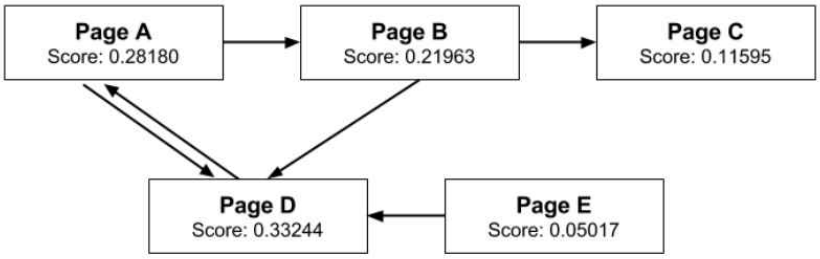
■ Page D new rank:

$$\underbrace{\beta \cdot 0.2/2}_{\text{from page A}} + \underbrace{\beta \cdot 0.2/2}_{\text{from page B}} + \underbrace{\beta \cdot 0.2/1}_{\text{from page }\mathbf{E}} + \underbrace{(1-\beta)/5}_{\text{new surfers}} + \frac{1-S}{N} = 0.404$$

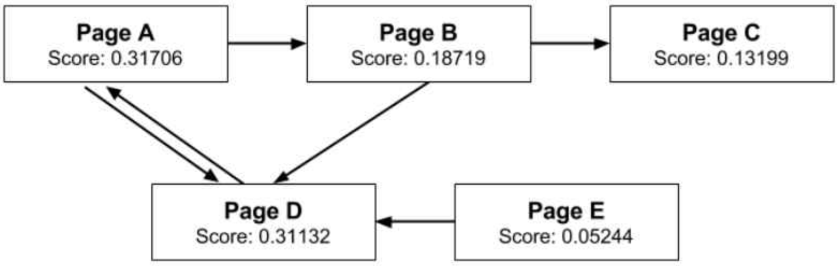
■ Page E new rank:

$$\underbrace{(1-\beta)/5}_{\text{new surfers}} + \frac{1-S}{N} = 0.064$$

Update step : Iteration 3



Update step : Iteration 100


Page A
Score: 0.31706
Page B
Score: 0.18719
Page C
Score: 0.13199
Page D
Score: 0.31132
Page E
Score: 0.05244