Université de Genève

Information Retrieval
14x011

TP 5 : Page Rank and Graph Analysis

Author: Sajaendra Thevamanoharan

E-mail: Sajaendra.Thevamanoharan@etu.unige.ch

Author: Deniz Sungurtekin

E-mail: Deniz.Sungurtekin@etu.unige.ch

23 May 2021

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES
Département d'informatique

Introduction

In this lab, we will become familiar with the web-page ranking algorithm based on the hyperlink structure of the web. Further, we will expermentally compare the PageRank algorithm to a random walk.

Page Rank

The page ranking algorithm is based on the number of times a given web pages is referenced by other pages. It contains the number of incoming links, but also from the importance of web pages where those links originate. This could be computed with the following formula that compute the rank $R(u)$ of a web page u :

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{L(v)}$$

where B_u is the number of web pages linked to u, $L(v)$ for a given page v is the number of links originating from v.

The PageRank is computed by using A, the graph adjacency matrix:

$$R_k = A^T R_{k-1}$$

with k, the step of the iteration, this update is done until convergence. The convergence stops when the difference between the update and the previous pageRank is below a fixed value. In other words, when the L_1 norm of $R_k - R_{k-1}$ is close to zero.

1. We consider now a set of web pages A,B,C,D,E,F,G interconnected by hyperlinks as shown below.

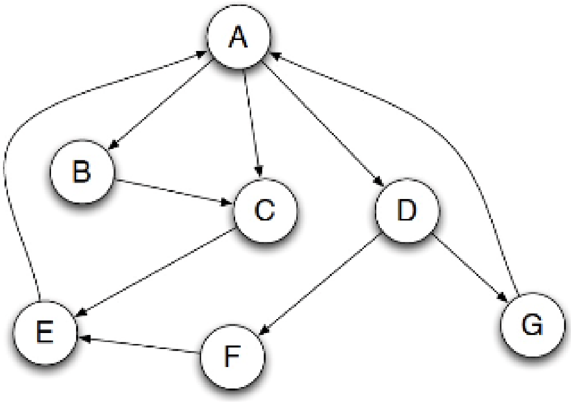
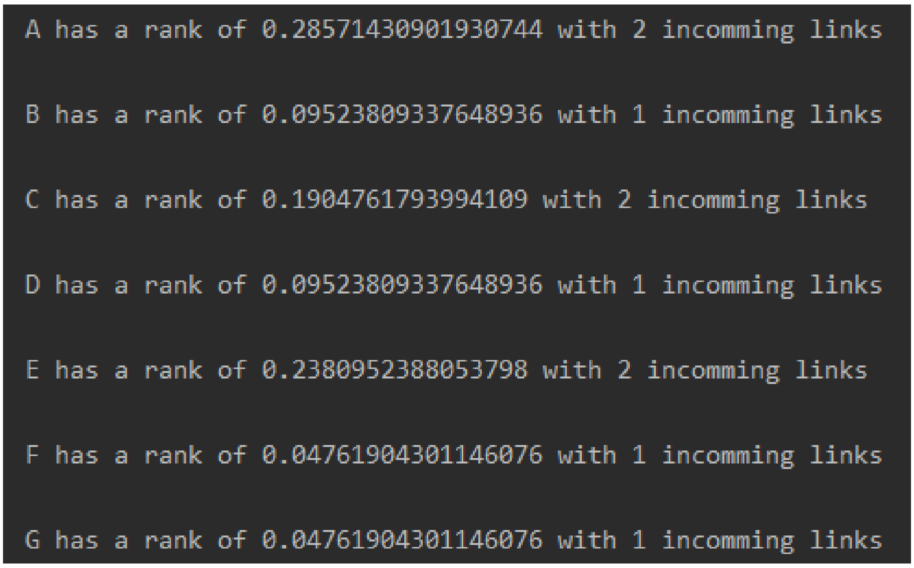


Figure 1: Set of web pages

The adjacency matriy $A(u, v)$ denotes the probability for a random surfer to jump from $page(u)$ to $page(v)$. The adjacency matrix for the previous graph is

$$A = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2. We implement the PageRank iterative algorithm as a function that takes an adjacency matrix as input and outputs a vector of web-page ranks. We have for the previous graph

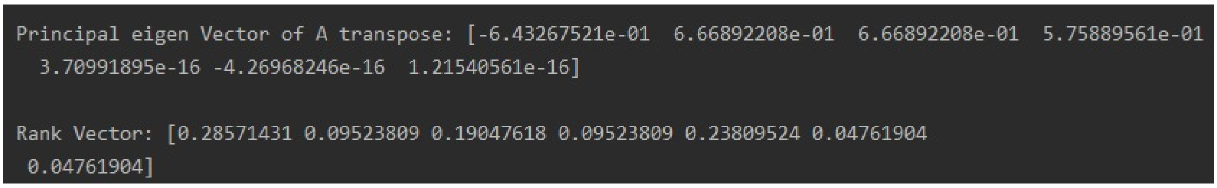


```
A has a rank of 0.28571430901930744 with 2 incomming links

B has a rank of 0.09523809337648936 with 1 incomming links

C has a rank of 0.1904761793994109 with 2 incomming links

D has a rank of 0.09523809337648936 with 1 incomming links

E has a rank of 0.2380952388053798 with 2 incomming links

F has a rank of 0.04761904301146076 with 1 incomming links

G has a rank of 0.04761904301146076 with 1 incomming links
```

Figure 2: Output from the PageRank iterative algorithm

We can see that A has a highest rank which is 0.28, followed by the second highest E with a rank 0.23. If we compared to the amount of incoming links, it matches. The result is obvious because of the definition of the ranking. When the number of incoming links are high, this website will get a higher score/rank. We can also notice that it does not depend only on the incoming links. Website C has two incoming as A but is classified as third. It can be explained by the number of outgoing links. A have two incoming links and three outgoing links, whereas C has only one output link. That's why C has been ranked as third.

3. We compute now the principal eigenvector of A^T, where A is the adjacency matrix. We compare it with the eigenvector with the PageRank vector we obtained in Question 2.



```
Principal eigen Vector of A transpose: [-6.43267521e-01  6.66892208e-01  6.66892208e-01  5.75889561e-01
  3.70991895e-16 -4.26968246e-16  1.21540561e-16]

Rank Vector: [0.28571431 0.09523809 0.19047618 0.09523809 0.23809524 0.04761904
 0.04761904]
```

Figure 3: Pricipal eigen vector of A^T and the PageRank vector

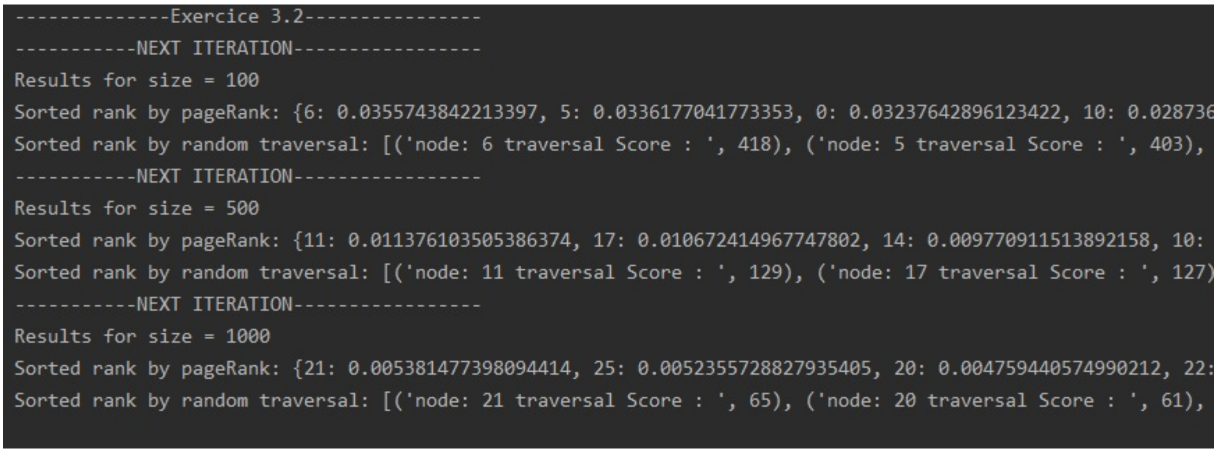
Since the pagerank algorithm is an iterative application of the link matrix, the ultimate pagerank vector will look a lot like an eigenvector associated with the highest eigenvalue of the link matrix.

So back there we repeatedly applied the link matrix to the pagerank vector. This is equivalent to repeatedly multiplying the link matrix by itself, and then after enough self-multiplications, applying that result to the pagerank vector. However, repeated multiplication is exponentiation. We can do that by exponentiating the matrix, which we can do by exponentiating the diagonal matrix with the eigenvalues in it. So, the diagonal matrix comes to be dominated by the largest (original) eigenvalue, and so does the ultimate link-following matrix come to be dominated by the vector associated with the largest eigenvalue, thus the ultimate pagerank vector will be dominated by that vector too. These eigenvectors will give us an idea where we should stop the iteration.

Graph Analysis

For this part, we generate a random Graph. Then, we compute the PageRank eigenvalue. Using the random walk method, we define the ranking as the number of traversal from a random initial node.

We experiment with different size of graphs $\in \{100, 500, 1000\}$. We get



```
--------------Exercice 3.2----------------
-----------NEXT ITERATION-----------------
Results for size = 100
Sorted rank by pageRank: {6: 0.0355743842213397, 5: 0.0336177041773353, 0: 0.03237642896123422, 10: 0.028736
Sorted rank by random traversal: [('node: 6 traversal Score : ', 418), ('node: 5 traversal Score : ', 403),
-----------NEXT ITERATION-----------------
Results for size = 500
Sorted rank by pageRank: {11: 0.011376103505386374, 17: 0.010672414967747802, 14: 0.009770911513892158, 10:
Sorted rank by random traversal: [('node: 11 traversal Score : ', 129), ('node: 17 traversal Score : ', 127)
-----------NEXT ITERATION-----------------
Results for size = 1000
Sorted rank by pageRank: {21: 0.005381477398094414, 25: 0.0052355728827935405, 20: 0.004759440574990212, 22:
Sorted rank by random traversal: [('node: 21 traversal Score : ', 65), ('node: 20 traversal Score : ', 61),
```

Figure 4: Result from graph analysis

We can notice that in both cases, we obtain the same ranking. The random walk is to simulate a normal surfing on internet through multiple websites. We create a Pagerank model to get the rank of the website based on its connexion. The result is corresponding with a random surfing. We can conclude that the PageRank method is efficient to measure the ranking of any website.