TP5 - Page Rank and Graph Analysis

Atul Sinha
e-mail: atul.sinha@unige.ch

May 10, 2021

1 Goal

In this exercise, we develop a web-page ranking algorithm based on the hyperlink structure of the web, where web pages and hyperlinks can be represented as vertices and directed edges, respectively.

Further, we will experimentally compare the PageRank algorithm to a random walk.

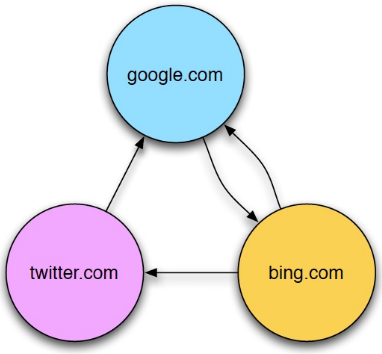
2 Introduction

The motivation for ranking web-pages is due to the so-called abundance problem:"the number of pages that could reasonably be returned as relevant is far too large for a human user to digest" (Kleinberg,1999). Thus, the goal of a graph-based ranking method is to identify the most important documents using the hyperlinked structure of the web. In the following we consider a simplified version of the Google's PageRank (Page et al., 1999), which will be a basis of this exercise.
The basic premise of graph-based algorithms, such as PageRank, comes from the academic citation analysis. It is based on the observation that the importance of an academic contribution is directly proportional to the number of times it is cited by other authors. In the context of the world wide web, this roughly corresponds to the number of times a given web page is referenced by other pages. PageRank takes this idea even further and computes the importance of a web page not only from the number of incoming links, but also from the importance of web pages where those links originate. This notion is captured by the following recursive formula that

computes the rank $R(u)$ of a web page u:

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{L(v)}, \tag{1}$$

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



For instance, using a miniature model of the WWW with 3 web pages twitter.com, google.com, bing.com as in the above Figure, we obtain

Web Page	B_u	$L(v)$
google.com	twitter.com, bing.com	1
twitter.com	bing.com	1
bing.com	google.com	2

Start with $R(google.com) = R(twitter.com) = R(bing.com) = 1/3$, we compute the PageRank formula iteratively until convergence, when $R(google.com) = 0.4$, $R(twitter.com) = 0.2$, $R(bing.com) = 0.4$.

The above iterative process can be interpreted formally with matrix notations. Given a graph of N web-pages, R denotes an N dimensional column vector, where $R(u)$ is the PageRank value of page u. Initially $R(u) = 1/N$ for all web-pages. Let A be the $N \times N$ graph adjacency matrix, where $A(u, v)$ denotes the probability for a random surfer to jump from page(u) to page(v). There are two cases: if there is a hyperlink from page(u) to page(v), then $A(u, v) = 1/L(u)$; if otherwise, $A(u, v) = 0$. Note that A is not necessarily symmetric. In the case of the 3-page example, we have matrix A represented as

$A(u, v)$	google.com	twitter.com	bing.com
google.com	0	0	1
twitter.com	1	0	0
bing.com	0.5	0.5	0

Thus, the PageRank is updated as

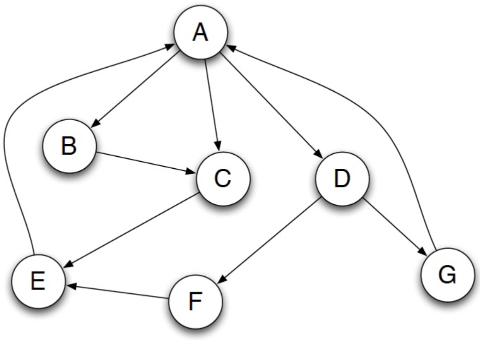
$$R_k = A^T R_{k-1}, \tag{2}$$

with $k = 1, 2, 3, ...$, until convergence. The convergence can be detected by checking the L_1-norm of $R_k - R_{k-1}$. When this norm is sufficiently close to zero, the iteration stops. This iterative procedure coincides with the "Power Method" to find the principal eigenvector R of matrix A^T, that is, the eigenvector corresponding to the largest eigenvalue.

3 Tasks

3.1 Page Rank

1. Consider a set of web pages A, B, C, D, E, F, G interconnected by hyperlinks as shown below. Give the above defined adjacency matrix for this graph.



2. Implement the PageRank iterative algorithm as a function that takes an adjacency matrix as input and outputs a vector of web-page ranks. Find the webpage from the given set A, B, C, D, E, F, G that has the highest rank. Compare the computed PageRank values with the number of incoming links for each web-page and explain your findings.

3. Compute the principal eigenvector of A^T, where A is the adjacency matrix that you constructed in Task 1. Compare this eigenvector with the PageRank vector you obtained in Task 2, and explain your result.

4. (Optional / Extra credit) Say, the web-page F in Task 1 is someone's homepage. He (she) wants F to be ranked as high as possible. Therefore he (she) creates

M spam sites (for example, $M = 100$) and make all of them link to F. Assume no page links to the M spam sites, and there is no links within the spam sites. Re-compute the PageRank values of all $M + 7$ web-pages and explain your result.

5. (Optional / Extra credit) Rank $A - G$ with the $HITS$ algorithm instead of PageRank. Discuss the relation between these two algorithms.

3.2 Graph Analysis

For this part you can use any graph package of your choice. The use of `https://networkx.github.io/` is recommended.

1. Generate a random Graph using the Barabsi-Albert model.
2. Compute PageRank eigenvalue.
3. Compute the ranking defined as the number of traversal from a random initial node.
4. Experiments with different sizes of graphs.

4 Assessment

The assessment is based on your report. It should include all experimental results, your answers to all questions, and your analysis and comments of the experimental results. It should be around 5 pages for this assignment. Please try to detail the report by giving examples and conclusions. Please archive your report and codes in "PrenomNomTP4.zip", and upload to `http://moodle.unige.ch` under TP5 before May 22, 2021, 23:59PM. Later submission will not be accepted.