UPC - FIB
Information Retrieval, MIRI
Lab 3: Implementing Pagerank
Fall semester 2016/2017

Authors: Krishna Kalyan, Roman Kopšo

Date: 5th November 2016

Report:

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

Our data set consists of two files. Airports.txt and routes.txt that contain 7663 and 68820 observations respectively. For this assignment we will use networkx a pythonic language data structures for graphs, digraphs, and multigraphs.

The following steps were taken to prepare the data for pagerank algorithm.
- Remove empty IATA airport values from airports.txt
- Retain routes that appear in airports dataset
- Encode airport codes to integers with map

We used the directed graph data structure to implement this algorithm. Every node represents an airport and edges represent directed routes between the airports. These edges have also weights which represent the number of links from source to destination. We used the damping factor 0.85 and implemented the pagerank algorithm as was written in pseudo code in the document.

We have succeeded in running the code and we mapped the pagerank score to specific airports. Results are saved in csv file pageranks.csv. To compare our result, we have used networkx's implementation of pagerank. The results are almost the same with some minor differences in the order of airports scores.

As far as the comments about the features of the pagerank are concerned, we have observed that the damping factor , when set to 0, gives very bad results because it sets pagerank score of nearly all of the airports to the same value and takes only one iteration. On the other hand, when set to 1, it never stops iterating because it never converges. We can imagine a person who is randomly clicking on the links. Pagerank says that the person should eventually stop clicking and the reason why we use damping factor is to prevent the sinks which are basically nodes with no outgoing edges. The probability that the person will continue clicking is set by this damping factor. Basically 0 for damping factor are clicks that all represent random restart and 1 means that the person will click forever. Therefore the best value is around 0.85 because it is a sort of weighted average between these two extremes.

Speed of convergence depends on the damping factor. The higher the damping factor, the more iterations it takes for the convergence.

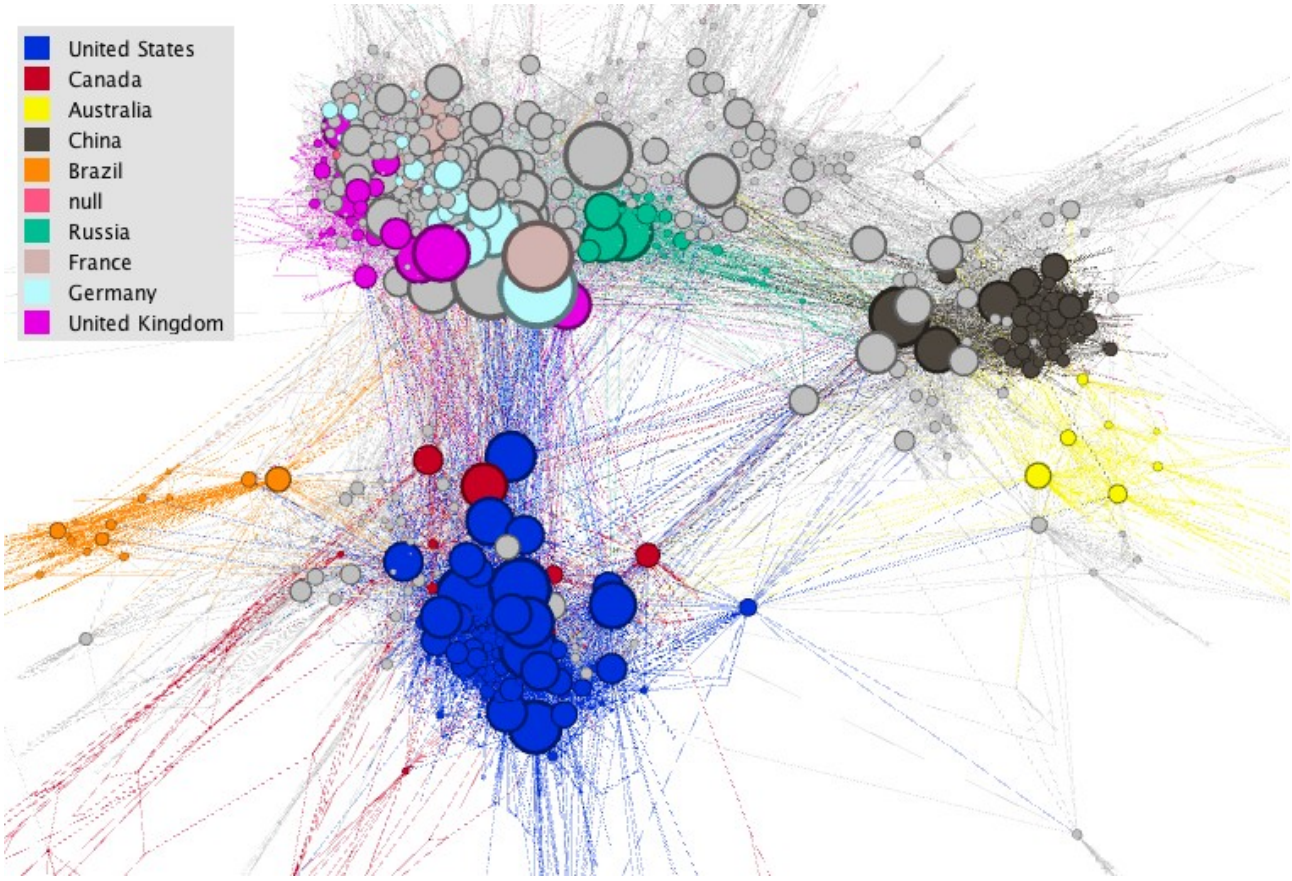
Stopping condition for the pagerank loop should be the above mentioned convergence or fixed number of iterations. Generally, when converging the rule of thumb is that pagerank has converged

when the difference between two pagerank score arrays (one of them is one iteration ahead) is less or equal to T times the number of nodes in the graph, where T is either 10^{-4}, 10^{-8}, 10^{-16} etc. Depending on our precision needs, if we want to be more precise, we select 10^{-16} , if we want to be less precise, we select 10^{-4}. The reason, why multiply the T with the number of nodes is because it reduces the iteration count by 50% and does not have a high effect on affecting the precision src[http://www4.ncsu.edu/~ipsen/ps/slides_imacs.pdf#21].

However, we have to admit one flaw in our implementation of pagerank. The sum of pagerank scores does not equal precisely to 1 but some number very close to one (for example 1.003724112). We were trying to fix this problem but we were not successful. We have suspicion that the reason behind this is the floating point of the numbers when doing the divisions and that probably spoils the clear 1.

All in all, we are really sorry for this above mentioned imperfection but in this assignment we believe that we have successfully understood the principles, know-how and the purpose of the pagerank algorithm.

We also did some extra work and plotted the graph of the airports using software *Gephi*.



Pic no. 1 – Graph representation of the airports and their routes

On the picture, larger the node, larger the number of edges that node has. Some of the top countries with airports are colored while the rest is in grey.

PS: We apologize for breaking the one page limit but we believe that because of this graph, it was worth it. Just in case, we have sent our code in 3 formats (.py, .pdf, and ipython notebook).