UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

CAIM Lab, Session 4:

Pagerank

Fall 2021-22

Marco Madalin Farcas Vinter
Gerard Caravaca Ibañez

Index

1.Setbacks

The main problem from this practice was understanding how the Page Rank algorithm works; the core functionality of it is simple but understanding the nature of each calculation took us long enough as we tried to think of the most optimal way to apply it in our case. A much simpler quadratic algorithm based on the number of routes could have been made, however, it would have been incredibly more inefficient in time.
Once we knew how we would approach things we investigated what information the text files provide as we needed to know how to read the input and process it.
Another problem we faced was figuring out how to make the sum of the page rank equal to 1. In our first version of the code this problem was persistent even though we knew the source of it; yet we did not have a solution so we started testing out some potential ones. In the end we came up with distributing the pagerank of the sink nodes in order to preserve their value so they do not lose it.

2.Desing decisions and experiments

As there were two variables we were meant to play with (damping factor and stopping condition) in order to calculate the pagerank, we had to make some decisions and experiment a bit in order to find the expected results.

2.1.Dampling factor

- Small values (0.2) make the pagerank's values more similar to each other and take very little time (< 1 sec).
- For very large values (0.95), results tend to vary a lot among each other and much more time for computing is needed (around 10-15s).
- For values between the range 0.8 and 0.9 as they are the popular ones values are still distant from each other but not too far from the results obtained with 0.95, however, the time has been 3 times faster (3-5s), as a result this range had to be the sweet spot that balances execution time and pagerank values,we decided to go with 0.8.

We've attached some of the outputs with different conditions in the outputs folder.

2.2.Stopping condition

Our first version of the code was with a fixed number of iteration big enough to get correct results so as to make it simple and start testing as soon as we get a working script. Now for the final stopping condition we decided to go with the similarity between pagerank values, that is we take the difference between consecutive values and check if this value was less than a certain number. Different numbers were tested from 0.1 to 1*10^-20 and in the end we decided to go with 1*10^-15 as values weren't that meaningful further down and from a point onwards execution time started to take a noticeable toll.