Homework #3 – Link Analysis

Yoke Kai Wen, 2020280598

March 21, 2021

1 Introduction

In this assignment, I implemented a site-level PageRank calculation algorithm on a given real website sample collection, comprising 500k sites and 72.6 million links.

2 Program implementation

2.1 Methodology

I implemented an iterative calculation of the PageRank scores with sparse matrices, with the help from `scipy.sparse` Python libraries. Given the graph adjacency matrix $\boldsymbol{A}$, and with convergence limit set to 0.000001, and the damping factor d set to 0.85, the PageRank scores at iteration t, $\boldsymbol{PR}(t)$ is evaluated in the below equation in each iteration:

$$\boldsymbol{PR}(t+1) = d\boldsymbol{M} * \boldsymbol{PR}(t) + \frac{1-d}{N}\mathbf{1}$$

where $\boldsymbol{M} = (\boldsymbol{K}^{-1}\boldsymbol{A})^T$, with $\boldsymbol{K}$ being the diagonal matrix containing the outdegree of each node in its diagonal. Note that for $\boldsymbol{K}$ to be invertible, it cannot contain 0 elements on its diagonal, hence I arbitrarily set the elements which are zero (representing nodes with zero outgoing links) to some non-zero value. Note that it does not affect the calculation of $\boldsymbol{M}$ since the corresponding elements of $\boldsymbol{A}$ during the matrix multiplication operation are zero anyway.

When the L2 norm of $\boldsymbol{PR}(t+1) - \boldsymbol{PR}(t)$ decreases below the convergence limit, the algorithm terminates and outputs the final PageRank scores.

In order to account for nodes with no outgoing links say $\{deadendnodes\}$, the above equation is modified, by further adding the sum of the previous PageRank scores of nodes belonging to $\{deadendnodes\}$, scaled by the total number of nodes in the graph N, to every node in the graph, i.e. $\boldsymbol{PR}(t+1)$.

$$\boldsymbol{PR}(t+1) += \frac{1}{N} \sum_{n \in \{deadendnodes\}} \boldsymbol{PR}(t)[n]$$

This ensures that the PageRank scores of each deadend node are redistributed with equal probability to all other nodes, hence maintaining a total score sum of 1, with the score of each node representing the probability a web surfer will click on it.

2.2 Efficiency of program

I was able to speed up considerably the PageRank computation with the use of `scipy.sparse.csr` and `scipy.sparse.coo` sparse matrix formats. Loading the given edgelist in the `.txt` file into a `coo` sparse matrix is one of the significant bottlenecks, taking about 11.85s. Creating the out-degree diagonal matrix and the intermediate matrix $\boldsymbol{M}$ also takes significant time, about 10s. Finally, each loop takes roughly 0.28s. After running 46 iterations before convergence is reached, a total time of 35s is spent, from loading the edgelist to convergence.

In summary:

Time for one loop: 0.28s

Total running time: 35s

Note that the above timing results are based on the below **computer hardware configuration**:

1. RAM: 16GB
2. CPU: 4 cores, Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz

3 PageRank scores results

3.1 Distribution of PageRank scores

Figure 1 shows the distribution of PageRank scores (log applied) across all 500k web pages.

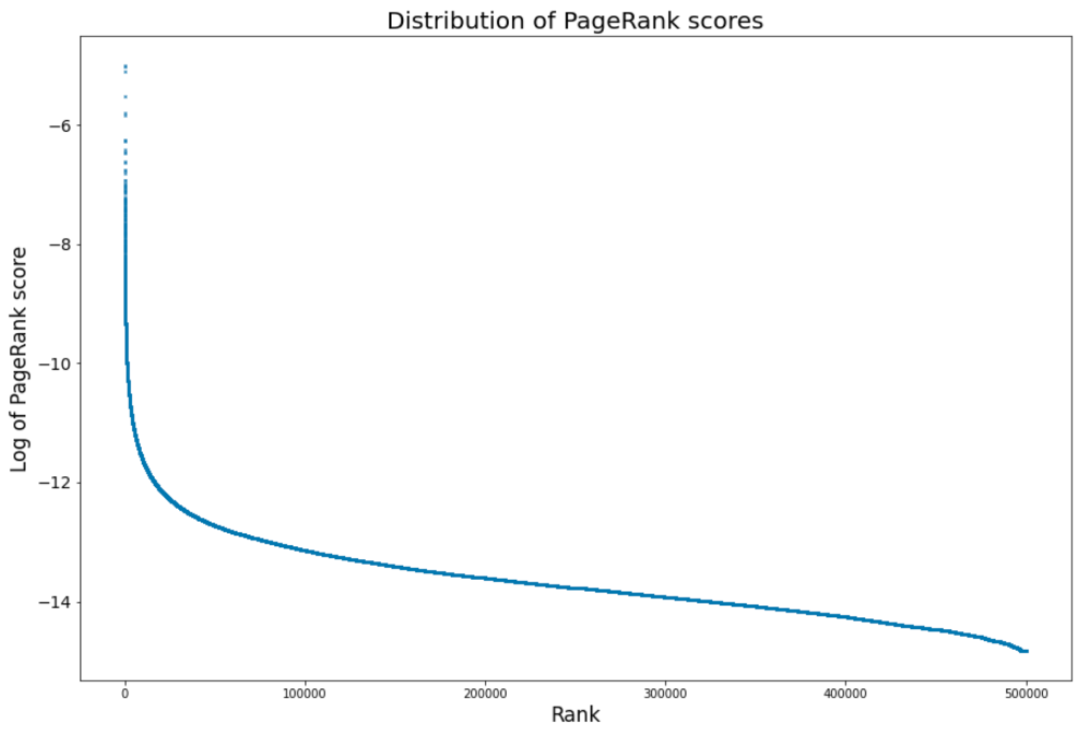
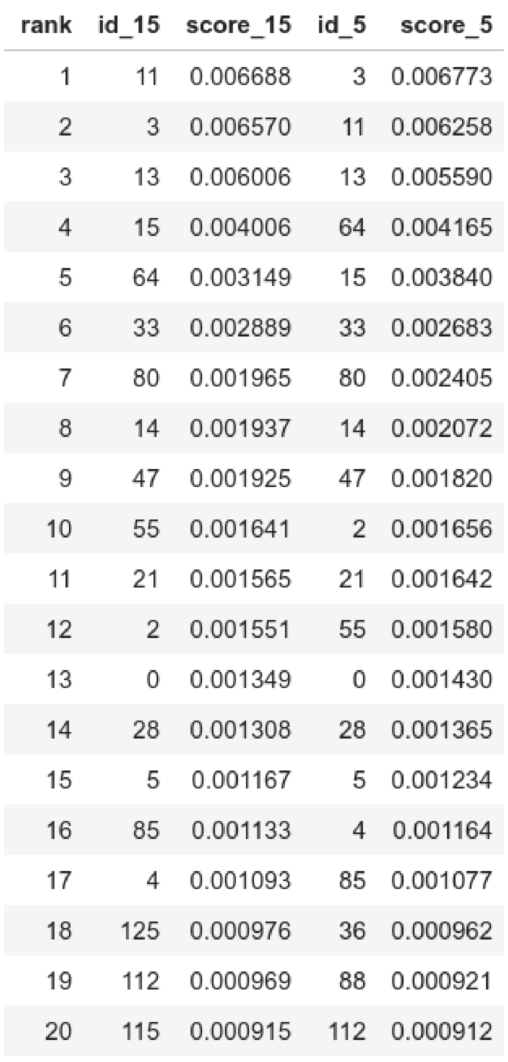


Figure 1: Distribution of log of PageRank scores for pages ranked 1st to 500000th

3.2 Comparison on changes of top 20 PageRank sites and scores when loop=5 and loop=15

Figure 2 shows the top 20 PageRank sites and scores when 5 and 15 iterations are run. Clearly, there are some small differences in scores and therefore rankings, but the pages appearing in the top 20 for both 5 and 15 iterations are mostly the same,with a few exceptions: pages 36, 88 appear in 'score_5' but not 'score_20'; pages 125, 112 appear in 'score_20' but not 'score_5'. Such differences are to be expected given that convergence has not been reached yet at the point of 5 iterations.

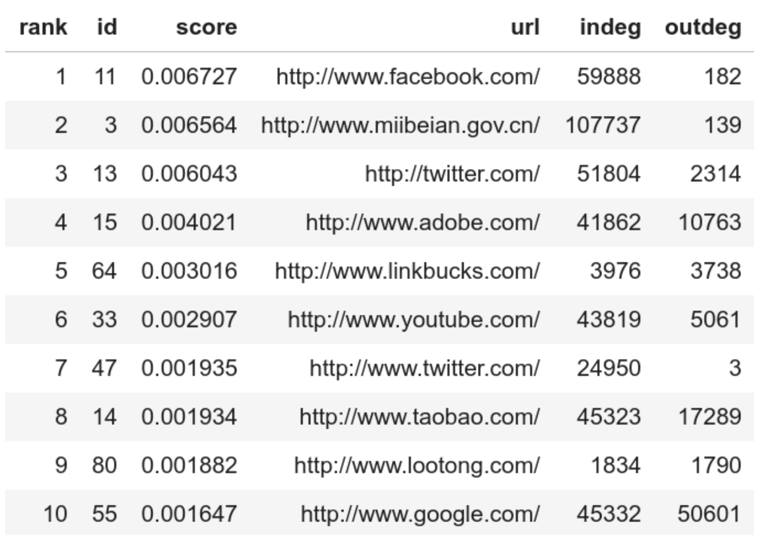


rank	id_15	score_15	id_5	score_5
1	11	0.006688	3	0.006773
2	3	0.006570	11	0.006258
3	13	0.006006	13	0.005590
4	15	0.004006	64	0.004165
5	64	0.003149	15	0.003840
6	33	0.002889	33	0.002683
7	80	0.001965	80	0.002405
8	14	0.001937	14	0.002072
9	47	0.001925	47	0.001820
10	55	0.001641	2	0.001656
11	21	0.001565	21	0.001642
12	2	0.001551	55	0.001580
13	0	0.001349	0	0.001430
14	28	0.001308	28	0.001365
15	5	0.001167	5	0.001234
16	85	0.001133	4	0.001164
17	4	0.001093	85	0.001077
18	125	0.000976	36	0.000962
19	112	0.000969	88	0.000921
20	115	0.000915	112	0.000912

Figure 2: Top 20 PageRank sites and scores when loop=5 and loop=15

3.3 Individual analysis of top 10 sites

After running my PageRank algorithm till convergence (46 iterations), I obtained the top 10 ranked web pages as shown in Figure 3.



rank	id	score	url	indeg	outdeg
1	11	0.006727	http://www.facebook.com/	59888	182
2	3	0.006564	http://www.miibeian.gov.cn/	107737	139
3	13	0.006043	http://twitter.com/	51804	2314
4	15	0.004021	http://www.adobe.com/	41862	10763
5	64	0.003016	http://www.linkbucks.com/	3976	3738
6	33	0.002907	http://www.youtube.com/	43819	5061
7	47	0.001935	http://www.twitter.com/	24950	3
8	14	0.001934	http://www.taobao.com/	45323	17289
9	80	0.001882	http://www.lootong.com/	1834	1790
10	55	0.001647	http://www.google.com/	45332	50601

Figure 3: Top 10 PageRank sites

1. http://www.facebook.com/

   It is not surprising that Facebook, the world's most popular social networking site, has the highest PageRank score. We see from Figure 3 that Facebook has the second largest number of in-links, which also leads to it having a high PR score. For instance, people and companies would post a link to their Facebook profiles on their websites for greater social media presence.

2. http://www.miibeian.gov.cn/

   I was not able to access this url, but a quick search online reveals that this is a link to a Chinese Government Department website that all websites in China have to register and link to. This is substantiated by a scan of the urls linking to miibeian which all end in `.cn`. This explains why there is such a huge number of in-links to miibeian (twice that of Facebook), yet it does not have a higher PR score than Facebook, because many of the sites that link to it are small and unpopular Chinese sites that have low PR scores.

3. http://twitter.com/

   Again, similar reasoning to Facebook, and a lower score probably because Twitter is not as popular as Facebook, which can also been seen from the smaller number of in-links.

4. http://www.adobe.com/

   It comes as a surprise that adobe is ranked fourth, given my impression that it is a website for downloading productivity software. It also has a substantial number of in-links which directly explains its PR score. However after inspecting the in-links, I noticed that many of them were popular social media sites with rich multimedia, such as qq, twitter, various popular email providers etc, which probably use Adobe Flash to support various multimedia functions, hence many of these high scoring sites link to adobe, resulting in a high PR score.

5. http://www.linkbucks.com/

This website has been recently discontinued, but while the service was running, it was an internet advertising company that pays people for linking to other sites in their own sites, and earns money from advertisements displayed on the linked sites. It is still odd to me that linkbucks could achieve such a high PR score, given the small number of in-links. This indicates that sites that link to linkbucks must have very high PR scores. Indeed, we have popular social media sites, like youtube, sina blog, 163 blog, baidu etc that link to linkbucks.

6. http://www.youtube.com/

   Again, not surprising that youtube is ranked high since it is a popular social media site, also evidenced by large number of in-links.

7. http://www.twitter.com/

   It is very odd that IDs 3 and 47 are both pointing to the same twitter page, yet they have different number of in-links and out-links. Perhaps these two IDs should have been combined and could result in the highest PR score, this is probably a case of dirty dataset.

8. http://www.taobao.com/

   As the leading ecommerce site, again it is not surprising that taobao ranks very high, also evidenced by large number of in-links.

9. http://www.lootong.com/

   I am not very sure what lootong is. The website shows a lot of gaudy advertisements and Japanese porn. A quick google search of lootong does not reveal much. It is also odd that lootong is ranked 9 despite the very small number of in-links. I was unable to trace the websites pointing to lootong because for some reason they are unavailable in the `id-url.txt`, but these sites generally have an above average PR score but not too high.

10. http://www.google.com/

    It is quite surprising that such a popular search engine as google is only at the tenth place, it might be because google is banned in China and the dataset was crawled from a Chinese region? But we still see that google has a large number of in-links, perhaps most of them are from low-scoring sites resulting in a lower rank than expected.

4 Other comments

I had a hard time manipulating the large edgelist given to us, and it was only chancing upon sparse matrices that I managed to parse and implement PageRank in a reasonable amount of time without bursting my RAM. With sparse matrix operations, PageRank can be implemented very efficiently and scalably on a very large dataset, leading to my greater appreciation of the genius of this algorithm.

Nevertheless, the individual analysis of the top 10 ranking pages also shows us that link analysis might not always be the best way to find the most popular sites, as it can be gamed. For instance, linkbucks and lootong probably manipulated the linking system somehow because it does not make sense that they have such high PR scores. Also, miibeian should not be appearing so high on PageRank as it is of little interest to any web user, but it was merely a result of a government policy.

5 Instructions on running code

Codes is in `pagerank.py`, which requires the installation of `numpy, scipy, pandas, time` libraries. Note that I also add an additional line `499999 499999` to the edgelist `linkgraph.txt` for more convenient parsing into a sparse adjacency matrix, and then set the element with coordinates (499999, 499999) to zero.