BLG222E
Numerical Methods in Computer Engineering
Project-1 Report

Ahmet Akın
150200027

May 9, 2024

1 Introduction

This project aims to implement a pageRank algorithm to the cora.cites dataset which includes some article ID's which cites another article or is being cited in another article. This report will cover main objective, challenges and implementation details regarding the project.

2 Problem Definition

PageRank algorithm is an algorithm developed by google to optimize their search engine for giving better and more related results to their users. What this algorithm does is simply giving ranks to each page derived from number of the links from the page and number of the links to the page, by putting all of the pages in a matrix as each column is set to show links from each page to links to another pages with regarding indices in the matrix. For example if page A has 2 outer links to pages B and C, in the matrix column that represents page A will have 1/2 value in rows represent B and C and if there are no other links other elements in the column will be set to 0. Values in matrix[B,C][A] is set to 1/2 because in pageRank algorithm the matrix is expected to be column stochastic and all columns are expected to sum up to 1 on the other hand there are no such definiton for rows (Ascher and Greif, 2011).
After understanding the purpose of pageRank algorithm, what is expected in the project is stated as implementing pageRank algorithm to cora.cites dataset which includes 2 data in its each line and each of them is written as ¡paper1¿¡paper2¿ which means paper2 cites paper1 thus there is a link from paper2 to paper1. The main objective is to get numerical results from the calculations and decide which papers are the ones with the most links into them in other words papers with the highest rank.

3 Implementation Details

3.1 Environment and Language

This project is expected to be written upon conda environment with jupyter notebook using python programming language as primary language to implement the algorithms. numpy, matplotlib and networkx libraries are used for implementations as well, numpy for general numerical values such as creating matrices and manipulating them and doing operations on them, matplotlib and networkx are mainly just used for drawing graph for the papers of highest 10 rank.

3.2 What is included in the code?

Python code to make the necessary implementations to get expected results include firstly a class definition of "Paper" to hold vital information regarding each paper such as their id, papers they link to and they are linked by and also their rank with useful methods to get and set these values with an easier manner.

Passing the paper links and weight of their link to a matrix was an important part of this project and this function is called on a paper set which includes all of the papers as objects for later advantage ("papers" array is used in lots of parts in main code). "PaperMatrix" function creates a 2-D paper array regarding to their outer and inner links. While doing this PaperMatrix function also supplies solution to dangling node and dead-end problems. For dangling node problem it changes all elements in the column with 1/paper number for the papers with 0 outerlinks. To solve dead-end problem a more probabilistic approach is implemented as suggested in the book from Ascher and Greif, a damping factor is added to the matrix as a probability of a surfer will follow the links being 0.85 and randomly jump to pages with a probability of 0.15 .
Implementing the pageRank algorithm on the matrix generated by PaperMatrix function was not trivial as it's expected to be. It simply is an implementation of the power method to obtain the most dominant eigenvector of a given matrix.
After all the functions are created what is left is to use them on given inputs which is cora.cites dataset. Firstly, the file is opened to read from and then all the lines are parsed to get necessary information to where it belongs. While doing this what the code does is, it gets a line, seperates it to 2 parts first being the cited paper id and second being the citing paper id, after this it creates object of a paper of given id if the specific id is not owned by another paper which means this paper was appeared in the dataset before so it doesn't create a duplicate but just passes the citing paper to it's reference part.
After this part, since an array of papers exists what is left to do is generating a matrix from this array and then call pageRank function on them. The rest is for getting outputs easier for the 20 papers with highest and lowest ranks and for the graph.

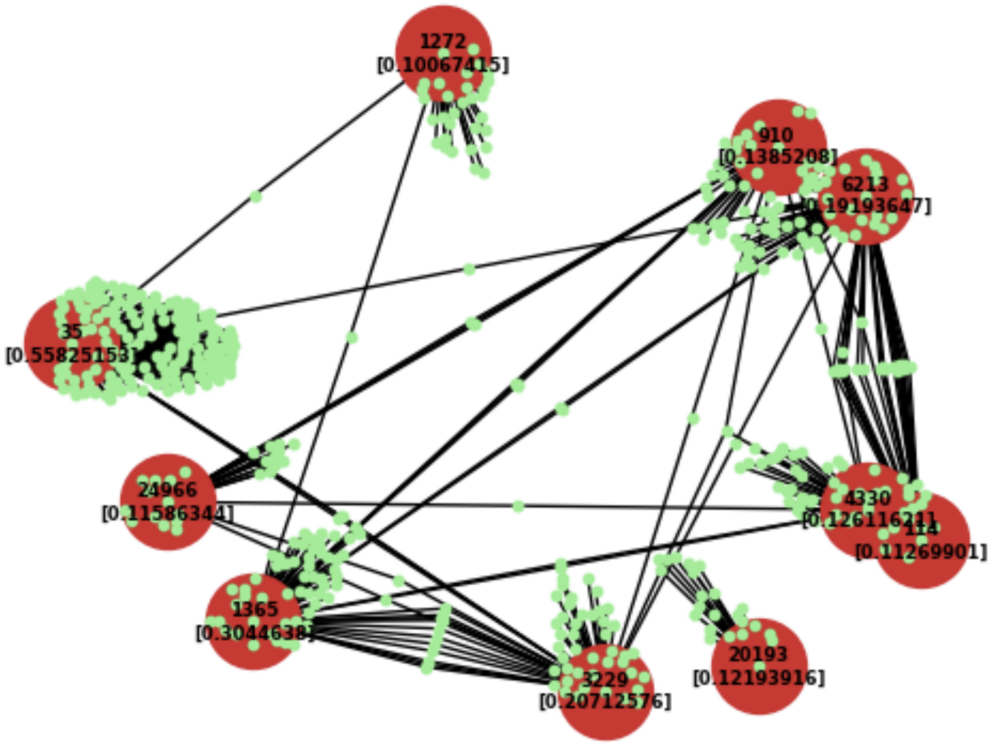
3.3 Challenges

Challenging parts of this project includes firsly generating a matrix from these dataset. Since what supposed to be done is firstly reducing a 2 input line to 1 paper and then again turning them to a 2-D matrix, it was not a piece of cake for sure but after writing everything down on a paper it got easier. PageRank algorithm was easy to implement because it is just mathematical calculations and since python is a viable programming language to do so it did not create much of an issue.
What else challenged me is file operations on python, since it is not a concept I am used to i spent some time on it to figure out exactly how it works and afterwards it was solved, file operations with python is not hard as I thought.
Last but not least, drawing a graph of the 10 papers with the highest rank was also challenging because it was the only library I've never used before. Dividing papers into arrays and using papers they link them was tricky till i get to another solution by browsing the library website for a long time but luckily I overcame that problem as well and right now it works as expected.

4 Dataset and Results

The cora.cites dataset includes over 5000 lines and 2728 different paper ID's. Some of these ID's are referenced by many other papers but some of them are referenced by none, by the definitions given before the papers referred a lot has higher rank than the ones they are not referred.
Results from the running the algorithm shows that the paper with the highest rank is 35. Papers with lowest rank can vary because some papers have share the same rank value.
20 Papers with the highest rank are:
1 - 35 : [0.55825153]
2 - 1365 : [0.3044638]
3 - 3229 : [0.20712576]
4 - 6213 : [0.19193647]
5 - 910 : [0.1385208]
6 - 4330 : [0.12611621]
7 - 20193 : [0.12193916]
8 - 24966 : [0.11586344]
9 - 114 : [0.11269901]
10 - 1272 : [0.10067415]
11 - 2440 : [0.0964971]

12 - 19621 : [0.09168716]
13 - 887 : [0.08396593]
14 - 8224 : [0.08282674]
15 - 2665 : [0.08181412]
16 - 3231 : [0.0808015]
17 - 4584 : [0.07814337]
18 - 15429 : [0.07282712]
19 - 31353 : [0.07219423]
20 - 22563 : [0.07029557]
20 Papers with the lowest ranks (They share same rank) are:
1133008, 1134031, 1134056, 1154230, 1134197, 735311, 1135455, 1140548, 1135955, 1136631, 1136634, 1136040, 1136110, 1136447, 1136449, 1137140, 1138619, 820661, 1139009, 1140231 with the rank value of [0.00270323]



Graph for 10 papers with the highest rank

5 Conclusion

This project helped me to understand how to use eigenvectors and values to obtain another value that can be used in ranking and deciding among a set of data. Also I get a brief understanding on machine learning algorithms in a better way because I knew to implement machine learning algorithms math knowledge is necessary but implementing a numerical algorithm to get some results give me a better understanding.

References

Ascher, Uri M. and Greif, Chen, A First Course in Numerical Methods, 2011