Page Ranking Algorithm (SCL Project)

Dhikshitha (21PD26)
Sharmila (21PD33)

Introduction

- PageRank (PR) is an algorithm used by Google Search to rank websites in their search results.
- Page Rank was named after Larry Page, one of the founders of Google.
- PageRank is a way of measuring the importance of website pages.
- According to Google PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is.

Libraries Used

Numpy

Numpy library functions are used to calculate a rough estimate of rank of each pages

Matplotlib

Matplotlib library function barplot is used to visualize the probability of a person landing in a webpage

Random

Used randint to generate adjacency matrix of webgraph.

Implementation of Page Ranking

- The web graph is taken as input with the help of Adjacency Matrix
- Initially it is assumed that user will navigate to another website only with the help of links that are specified in the website . So a matrix L is evaluated using given adjacency matrix
- Then with the help of Power Method Dominant eigenvector of L is identified. This gives us the rank of each page.
- Sometimes a website might contains links only to itself. In this case that website will have higher rank which is not sensible. Almost all the traffic will be taken by this website

- So in order to combat this a small probability that the user doesn’t follow the link on the webpage (Random link is used)
- Let d be the probability that the user follows the link on the webpage and (1-d) be the probability that the user navigates using random link.
- Whatever is discussed above is the case where d=1.
- If d=0 user completely makes use of random links meaning all pages will be equally ranked. A new matrix M is calculated after each user visits the website

The new matrix M is given by

$$M = d*L + ((1-d)/n)*J$$

- Where L is the previous case matrix
- d is probability that user makes use of links in the web page to navigate. The value of d is taken randomly
- J is square matrix containing all 1’s whose order is n (number of nodes in web graph)

- With the help of barplot rank of all the pages is visualized
- Then a generalized program/function is defined for a web graph of n nodes. Taking n as input , the adjacency matrix is generated using randint function.
- Later the same process specified above is performed. Here we made of Linear Algebra’s dominant eigenvector is used.
- Now Page Rank algorithm is implemented without the use of Linear Algebra concepts

In this case the probability/rank of landing in a web page is given by

Pr(A) = (1-d) + d * (Pr(B)/Cout(B) + Pr(C)/Cout(C) + … +Pr(N)/Cout(N))

- A is the page for which probability is calculated
- d is probability that user makes use of links in the web page to navigate. The value of d is taken randomly
- Pr(Page) is the probability of landing in that page
- Cout(Page) is the number of distinct hyperlinks from that page

- This is also implemented in the program. And using barplot the probabilities are visualized.
- Then a comparison for the same webgraph has been made using the two methods.
- The probabilities are almost same and the ranks of the webpages are the same

References

- https://www.geeksforgeeks.org/page-rank-algorithm-implementation/
- https://www.geeksforgeeks.org/page-rank-algorithm-in-data-mining/
- https://towardsdatascience.com/pagerank-algorithm-fully-explained-dc794184b4af
- https://en.wikipedia.org/wiki/PageRank#:~:text=PageRank%20(PR)%20is%20an%20algorithm,the%20importance%20of%20website%20pages.
- https://www.youtube.com/watch?v=a5zPyhQf7xw

Code implementation of page ranking

https://colab.research.google.com/drive/1tMcOcvO1jKtQ6Jvhs1lUwSzgozXTpMBz#scrollTo=SKCrJsI-mAwV