Project Report

On

STUDY OF PAGERANK ALGORITHM

Submitted

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

MATHEMATICS

by

ANOOJA T A

(Register No. SM21MAT003)

(2022-2023)

Under the Supervision of

NISHA OOMMEN

DEPARTMENT OF MATHEMATICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI - 682011

APRIL 2022

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

CERTIFICATE

This is to certify that the dissertation entitled, **STUDY OF PAGERANK ALGORITHM** is a bonafide record of the work done by Ms. **ANOOJA T A** under my guidance as partial fulfillment of the award of the degree of **Master of Science in Mathematics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date: 18·04·2023
Place: Ernakulam

NISHA OOMMEN
Assistant Professor,
Department of Mathematics,
St. Teresa's College(Autonomous),
Ernakulam.

ST TERESA'S COLLEGE ... TICS

Dr.Ursala Paul
Assistant Professor and HOD,
Department of Mathematics,
St. Teresa's College(Autonomous),
Ernakulam.

External Examiners

1: Lakshmi Girish

2: PARAMESWARAN R

DECLARATION

I hereby declare that the work presented in this project is based on the original work done by me under the guidance of **NISHA OOMMEN**, Assistant Professor, Department of Mathematics, St. Teresa's College(Autonomous), Ernakulam and has not been included in any other project submitted previously for the award of any degree.

Ernakulam.

Date: 18/4/2023

ANOOJA T A

SM21MAT003

ACKNOWLEDGEMENTS

I must mention several individuals who encouraged me to carry this work. Their continuous invaluable knowledgably guidance throughout the course of this study helped me to complete the work up to this stage

I am very grateful to my project guide NISHA OOMMEN for the immense help during the period of work

In addition, very energetic and competitive atmosphere of the Department had much to do with this work. I acknowledge with thanks to faculty, teaching and non-teaching staff of the department and Colleagues.

I also very thankful to HoD for their valuable suggestions, critical examination of work during the progress.

Ernakulam.
Date: 18/4/2023

ANOOJA T A
SM21MAT003

ST.TERESA'S COLLEGE (AUTONOMOUS) ERNAKULAM

Certificate of Plagiarism Check for Thesis

Author Name	Anooja T A
Course of Study	MSc Mathematics
Name of Guide	Ms. Nisha Oommmen
Department	Post Graduate Mathematics
Acceptable Maximum Limit	20%
Submitted By	library@teresas.ac.in
Paper Title	Study of Page Rank Algorithm
Similarity	2%
Paper ID	720684
Submission Date	2023-04-10 17:07:40

Signature of Student

Signature of Guide

Checked By
College Librarian

* This report has been generated by DrillBit Anti-Plagiarism Software

Contents

Chapter 1

BASICS OF PAGERANK ALGORIHM

Page Rank algorithm was developed by Sergey Brin and Larry Page while they were students at Stanford University. The algorithm was initially used as part of a research project to improve the accuracy of search results,and it eventually became the basis for the Google search engine when the company was founded in 1998. [10] The basic idea behind PageRank is to assign a score to each web page based on the number and quality of other pages that link to it .Pages that are linked to by many high -quality pages are considered more important and are assigned a higher score.

The PageRank algorithm uses a recursive algorithm to determine the score of each page.It starts with an initial set of scores for each page,then iteratively updates the scores based on the scores of the pages that link to it. The process continues until the scores converge to a stable value. When a user performs a search query on google ,the search engine uses PageRank scores to determine which pages should be included in the search results and in what order. Pages with higher PageRank scores are considered more relevant to the search query and are more likely to appear at the top of the search results .

It is important to note that the PageRank algorithm is just one of many factors that google uses to determine search rankings . Google's search algorithm is constantly evolving ,and the company uses hundreds of

different factors to rank pages and determine which pages to show in response to a user's search query.
In this project in chapter 1 we discuss the basic concepts of the PageRank algorithm ,chapter 2 is about the iteration method ,chapter 3 is about the Power iteration method and in chapter 4 we discuss its applications.
The goal of this project is to give an overall view of Page Rank Algorithm and its applications .As my work I mainly focus on its application.

1.1 PAGE RANK

Page Rank(PR) is an algorithm used by Google Search to rank websites in their search engine results. It was named after Larry Page ,one of the founders of Google. Page Rank is used to measure the importance of web pages.

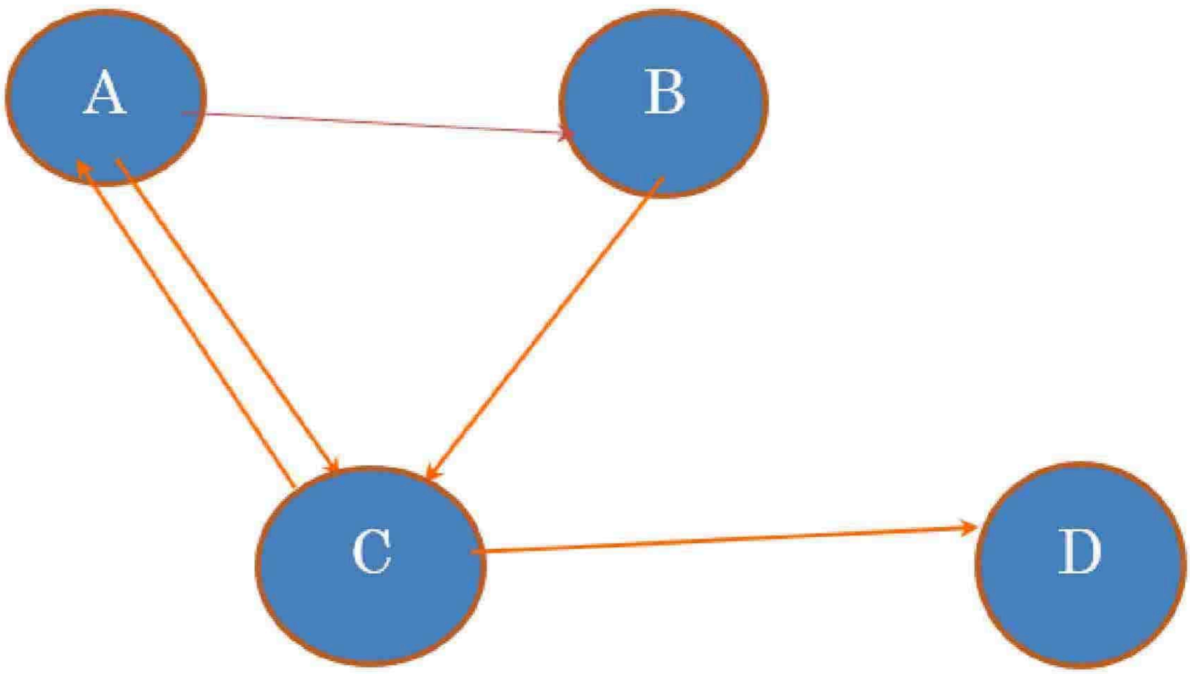

Figure 1.1: DIAGRAMMATIC REPRESENTATION OF PAGE RANK

1.2 Definition

1.2.1 Inbound links or Inlinks

Inbound links are links into the site from the outside. Inbound links are one way to increase a site's total page rank. Example; consider the node 'C', the inlink is link from A (A to C).

1.2.2 Outbound links or Outlinks

Outbound links or out links are links from a page to another page in a site or other site .Example ; Consider node 'B' ,the link from B to C is out link.

1.2.3 Dangling link

Dangling links are simply links that point to any page with no outgoing links .Example; link from C to D is dangling links of D.

1.2.4 Internal linking

A website's maximum Page Rank is distributed between its pages by internal links. The maximum amount of Page Rank in a site increases as the number of links to the site increases.

1.2.5 Damping factor

Damping factor is a mathematical constant usually set to 0.85 and is denoted by 'd' . It usually represents the probability that when a user gets bored of a current page and stops clicking the current page and moves to a random page. It prevents rank-sink problems, that is when a user gets trapped in a part of the web that is not well-connected to the rest of the web . If the damping factor is not used the user could be stuck in this part of the web , and the PageRank scores of the pages

in this area could become artificially inflated.

1.2.6 Dangling nodes and Disconnected components

Dangling nodes and disconnected components actually are two problems common on the internet , considering a large web. In order to deal with these two problems, a positive constant 'd' between 0 and 1 is introduced as the damping factor . PR = (1-d) + d.

1.2.7 Algorithm

The original page rank algorithm was described by Larry Page and Sergey Brin is given by:
PR(A) =(1-d)/N +d[PR(T1)/C(T1) +. . . . . . +PR(Tn)/C(Tn)]
Where,
PR(A) is the Pagerank value of node A
d is the damping factor,which can be set to 0.85
N is the total number of nodes in the graph
T1. . . . . . Tn are the nodes that link to node A
C(T1).....C(Tn) are the number of outgoing links for each of those nodes.

Example of Page Rank

Let's consider a simple web graph with 5 pages A,B,C,D and E,where each page has one or more outgoing links:
A- B,C,D
B- C,D
C- A
D- A,C,E
E- B,D,E

To apply the PageRank algorithm we first initialize each page with an equal PageRank a score of 1/5=0.2.We then iterate through the graph

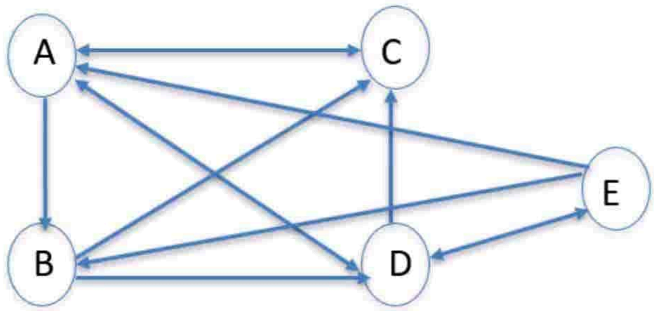

Figure 1.2: example

and update the PageRank score for each page based on the scores of the pages that link to it.

Here are the possible iterations ,

Initial score of each page is:-

A=0.2 B=0.2 C=0.2 D=0.2 E=0.2

PageRank of A

A has incoming links from C,D,E.

N=5

PR(A) = (1-d)/N + d[PR(C)/C(C)+PR(D)/C(D)+PR(E)/C(E)]

= (1-0.85)/5+0.85[0.2/1+0.2/3+0.2/3]

= 0.313

PageRank of B

B has incoming links from A,E

N=5

PR(B) = (1-d)/N + d[PR(A)/C(A)+PR(E)/C(E)]

= (0.15)/5+0.85[0.2/3+0.2/3]

=0.143

PageRank of C

C has incoming links from A,B,D

N=5

PR(C) =(1-d)/N + d[PR(A)/C(A)+PR(B)/C(B)+PR(D)/C(D)]

= (0.15)/5+0.85[0.2/3+0.2/2+0.2/3]

=0.228

PageRank of D

D has incoming links from A,B,E

N=5

PR(C) =(1-d)/N + d[PR(A)/C(A)+PR(B)/C(B)+PR(E)/C(E)]

= (0.15)/5+0.85[0.2/3+0.2/2+0.2/3]

=0.228

PageRank of E

E has incoming links from D

N=5

PR(E) = (1-d)/N + d[PR(D)/C(D)]

= (0.15)/5+0.85[0.2/3]

=0.086

The PageRank scores of the pages are:

A=0.313

B=0.143

C=0.228

D=0.228

E=0.086

In this example we can see that page A has the highest PageRank score,followed by C,D,B and E.From this we can say that page A is the most important page in the graph,since it has most incoming links from other important pages.

1.3 Merits and Demerits of Page Rank

1.3.1 Merits:

- **PageRank is highly effective in identifying the most relevant and important websites for a particular search query .**
- **It is a simple and efficient algorithm that can be easily imple-**

mented.

- **PageRank is able to account for the popularity and authority of a website ,which helps to ensure that the most reputable and trustworthy sources are ranked higher in search results .**

1.3.2 Demerits:

- **It could be computationally expensive for very large web graphs.**
- **It may not perform well in certain situations like ranking images or videos.**
- **Spider traps : A group of pages with no out-links which is able to step by step acquire the best rank. [7]**

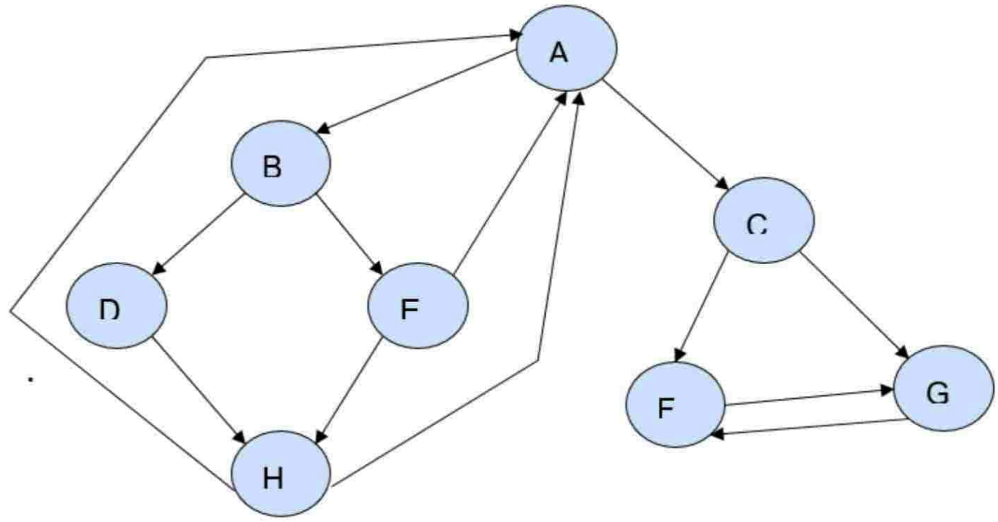

Figure 1.3: spider trap

- **Dangling Link : The link during a page that points to a page with no out-links.**
- **Dead ends : The page that don't have any out edges , PageRank doesn't handle these edges as a result they decrease the PageRank.**

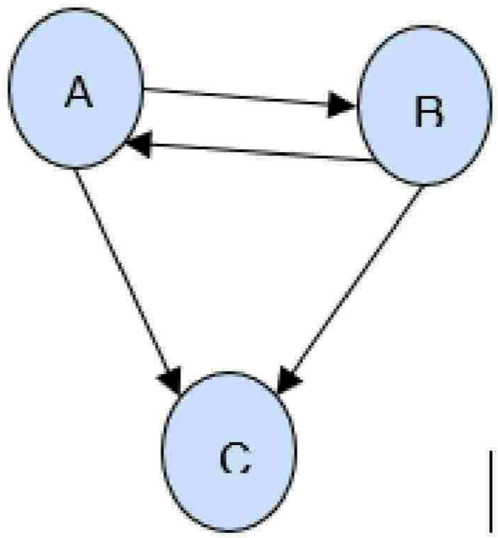

Figure 1.4: Dead End

Chapter 2

ITERATIVE METHOD OF PAGE RANK

The Page Rank algorithm uses an iterative method to calculate the importance of each page in a network of pages. The page is considered important if it is linked to many other important pages.In this process we first consider the initial values of the pages and compute it iteratively until it converges to a final score where it no longer changes for further iteration and this final score is called PageRank.

The equation defines the rank PR(P) for each page 'P'

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{(PR_t)(P_j)}{C(P_j)}$$

$$PageRankofPage = \sum_{i=1}^{n} \frac{PRofinboundlinkofpage}{Numberofoutboundlinksoftheinboundlinkpage}$$

2.1 To find Page Rank by Iteration

In the given figure A,B,C and D are nodes or webpages. The arrowed line from one page to another are links. In this figure we have outbound links and inbound links .

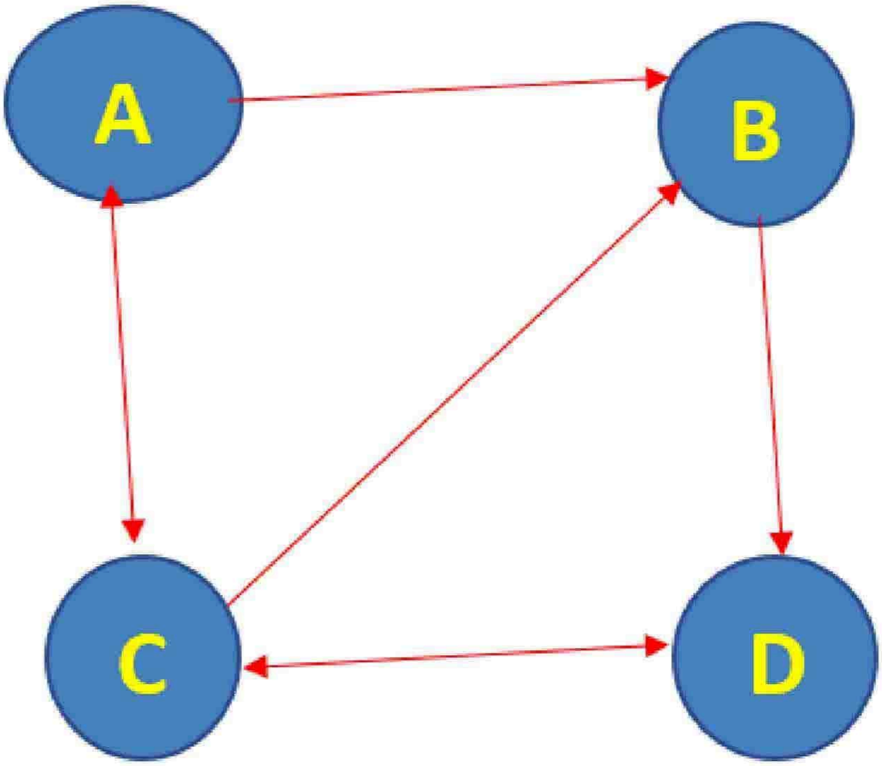

Figure 2.1: Iteration

2.1.1 Method:

The total number of pages in the figure = 4. So in iteration 0 we initialize the ranks of the web page to be, $PR(P_i) = \frac{1}{Numberoftotalpages}$

So ,

PR(A) = 1/4=0.25

PR(B)=1/4= 0.25

PR(C) = 1/4=0.25

PR(D) =1/4= 0.25

In Iteration 1,

$PRP_i = \sum_{P_i} \frac{PR(P_j)}{C(P_j)}$

To find page rank of page A , we first consider the page rank of page

A in iteration 0 and that is 1/4 .Also the number of outbound links are 3 . Thus ,

$PR(A) = \frac{PRofCofpreviousiteration}{Numberofoutgoinglinks}$

PR(A) = 1/(4/3) =1/12 =0.08

Next to find the page rank of page B ,

$PR(B) = \frac{PR(A)}{NumberofoutgoinglinksofA} + \frac{PR(C)}{NumberofoutgoinglinksofC}$

PR(B) = 1/(4/2) +1/(4/3) = 2.5/12=0.20

Similar to page B ,

$PR(C) = \frac{PR(A)}{NumberofoutgoinglinksofA} + \frac{PR(D)}{NumberofoutgoinglinksofD}$

PR(C) = 1/(4/2) + 1/(4/1) = 4.5/12 =0.37

PR(D) = 1/4 + 1/(4/3) =4/12=0.33

For iteration 2,

The calculation is similar to iteration 1 but the only difference is that here we initialize the values of iteration 1 for each page .

PR(A)=(4.5/12)/3 = 1.5/12=0.12

PR(B)=(1/12)/2 + (4.5/12)/3 = 2/12=0.16

PR(C)=(1/12)/2 +(4.5/12)/3 = 4.5/12=0.37

PR(D)=(2.5/12)/1 + (4.5/12)/3 =2.5/12 + 4.5/(12*3) =4.5/12=0.33

For iteration 3, we get

PR(A)=0.12

PR(B)=0.18

PR(C)=0.39

PR(D)=0.29

For iteration 4,we get

PR(A)=0.13

PR(B)=0.19

PR(C)=0.35

PR(D)=0.31

For iteration 5, we get

PR(A)=0.11

PR(B)=0.18

PR(C)=0.38

PR(D)=0.31

For iteration 6 ,we get

PR(A)=0.12

PR(B)=0.18

PR(C)=0.36

PR(D)=0.31

For iteration 7,we get

PR(A)=0.12

PR(B)=0.18

PR(C)=0.37

PR(D)=0.30

For iteration 8,we get

PR(A)=0.12

PR(B)=0.18

PR(C)=0.36

PR(D)=0.31

For iteration 9,we get

PR(A)=0.12

PR(B)=0.18

PR(C)=0.37

PR(D)=0.30

For iteration 10, we get
PR(A)=0.12
PR(B)=0.18
PR(C)=0.37
PR(D)=0.30
After 10 iterations the values converge and the scores of each page is given as 0.12 for page A,0.18 for page B,0.37 for page C and 0.30 for page D.

From the score we can say a random surfer is viewing page A 12% of the time ,page B 18% of the time ,page C 37% of the time and page D 30% of the time. This final probability is called PageRank.Here page C has the highest page rank. Also page C has more inbound links ,the page with more inbound links will have the highest PageRank.

Iteration	Page A	Page B	Page C	Page D
0	0.25	0.25	0.25	0.25
1	0.08	0.20	0.37	0.33
2	0.12	0.16	0.37	0.33
3	0.12	0.18	0.39	0.29
4	0.13	0.19	0.35	0.31
5	0.11	0.18	0.38	0.31
6	0.12	0.18	0.36	0.31
7	0.12	0.18	0.37	0.30
8	0.12	0.18	0.36	0.31
9	0.12	0.18	0.37	0.30
10	0.12	0.18	0.37	0.30

Figure 2.2: Table

2.2 Merits and Demerits of Iteration

2.2.1 Merits:

- **It is simple to implement and understand .**
- **It is it is computationally efficient and can handle large– scale web graphs with millions of pages .**
- **It can handle web pages with no inbound links known as "sink nodes".**

2.2.2 Demerits:

- **It is slow to converge to a steady state, especially for large web graphs.**
- **High computational cost : The algorithm requires matrix operations which are computationally expensive and it can be difficult to scale the algorithms to large web graphs .**
- **Iteration method sometimes fails to handle disconnected pages or nodes of "dead ends" or "spider traps" which can lead to a problem of ranking them.**
- **Difficulty in handling the damping factor: The algorithm requires the damping factor to set manually and selecting an appropriate value for the damping factor can be challenging.**

Chapter 3

POWER ITERATION METHOD

The iterative calculation of PageRank is equivalent to calculating the eigenvector corresponding to the eigenvalue 1 . To understand more about this first we need to understand a theorem known as Perron-Frobenius Theorem.

3.1 Perron-Frobenius Theorem

Perron-Frobenius theorem states that for a non-negative square matrix ,then there exists a unique positive eigenvector called the Perron vector with the largest eigenvalue called Perron value.
The PageRank algorithm uses the power iteration method to find this Perron vector, which represents the steady-state probability distribution of the random surfer on the web pages.The elements of this vector represent the PageRank of the individual webpages .Thus in short Perron-Frobenius theorem guarantees the existence of a unique positive eigenvector and the power iteration method is used to find this eigenvector. [13]

3.2 Power Iteration

Power method is an iteration method to compute λ , the dominant eigenvalue and v , the eigenvector of a matrix.It is a simple algorithm which does not compute matrix decomposition and hence it can be used in cases of large space matrices. Power method gives the largest eigenvalue and it converges slowly.

3.2.1 Theoritical background of Power Method

In linear algebra,the eigenvalue of a square matrix is defined as a vector which points in an invariant direction under the associated linear transformation.If λ be the eigenvalue of a square matrix A,it can be mathematically depicted as;
AV = λ V

Where V is an eigenvector which is nonzero. This vector 'V' is said to be normalized if its coordinate of largest magnitude is unity.

Power method,

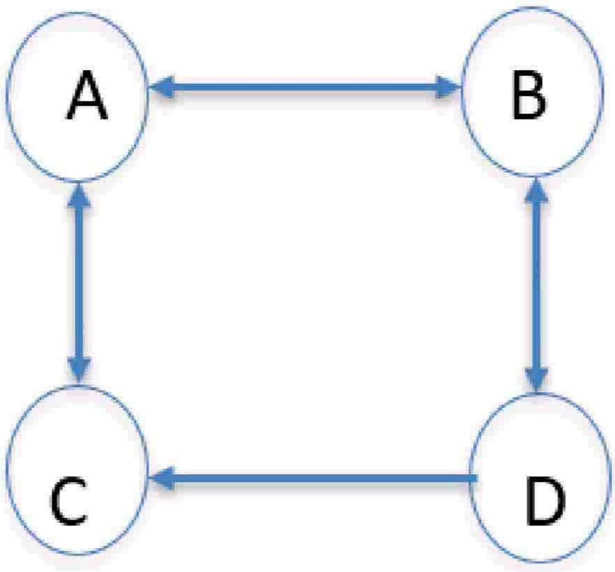

Figure 3.1:

Let's consider a simple web graph of four web pages ,where each page has one or more out links as shown in the figure.

The matrix representation for the following example is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Its transpose matrix is :

$$\mathbf{A}^T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

To compute the Page Rank vector we initialize the following matrix by total number of outbound links on each page and it is done for each column so that the sum total of each column is 1. Then we multiply new matrix with column matrix P_0.

P_0 is taken with the value of damping factor for each page in order to avoid dangling link in any iteration.

$$P_1 = \mathbf{A}^T\ P_0 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.85 \\ 0.85 \\ 0.85 \\ 0.85 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 1.275 \\ 0.85 \\ 0.85 \\ 0.425 \end{bmatrix}$$

$$\mathbf{P}_2 = \mathbf{A}^T\ P_1 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1.2755 \\ 0.85 \\ 0.85 \\ 0.425 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 1.2755 \\ 0.85 \\ 0.85 \\ 0.425 \end{bmatrix}$$

From P1 and P2 the Ranking of each page in decending order is page 1,2,3 and 4 and the Page Rank of each page is :

Page 1=1.9152

Page 2=0.85

Page 3=0.85

Page 4=0.2125

Even if we continue the iteration the ranking will be same so we can stop here.

3.3 Merits and Demerits of this Method:-

3.3.1 Merits

- **It is easy to implement and computationally efficient ,especially for large matrices**
- **It can converge to the dominant eigenvector in a relatively small number of iterations , even for matrices with millions of rows and columns.**

3.3.2 Demerits

- **It can be affected by the presence of spider traps, which are web pages that have many incoming links but no outgoing links , making the algorithm to get stuck in those pages .**
- **It can be effected by the presence of sink nodes ,which are web pages that have many outgoing links but no incoming links, making the algorithm not to give importance to those pages.**

3.4 Python Program in Power Method

```
import numpy as np

d = 0.85
num_iterations = 1


M = np.array([[0, 0.5, 1, 0],
              [0.5, 0, 0, 0.5],
              [0.5, 0, 0, 0.5],
              [0, 0.5, 0, 0]])

V = np.array([[0.85],[0.85], [0.85],[0.85]])

v = (1-d)/4+d*np.dot(M,V)

for i, page_rank in enumerate(v):
    print(f"Page {i+1}: {page_rank[0]:.4f}")
```

```
Page 1: 1.1213
Page 2: 0.7600
Page 3: 0.7600
Page 4: 0.3987
```

Figure 3.2: output

In iteration 2 output is:

```
Page 1: 1.0065
Page 2: 0.6835
Page 3: 0.6835
Page 4: 0.3605
```

Figure 3.3:

Chapter 4

Applications of Page Rank Algorithm

4.1 Ranking Tweets in Twitter

To use PageRank for ranking tweets in Twitter , we can construct a synthetic graph as follows . Represent each user and each tweet by a node .Draw a directed link from user A to B if A follows B. Also, draw a directed edge from user A to a tweet t if A tweets or retweets t .Now we can apply PageRank algorithm on this graph to obtain a ranking for tweets.If we ignore computational details,this algorithm provides a reasonable approach for ranking tweets in twitter. [8] The above method provides a global rank for each tweet.We can use a similar technique to obtain personalized PageRank: We initiate a random walk from the user of interest and teleport back to her with some fixed probability at each step.

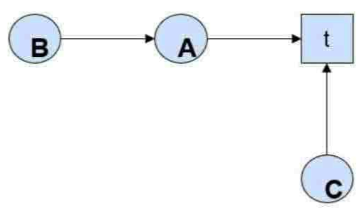

Figure 4.1:

4.2 Application in Bibliometrics

Bibliometrics is the quantitative analysis of publications , often used to measure and evaluate the impact of scientific research.In bibliometrics, the PageRank algorithm can be used to analyze the citation network of scholarly publications.
In the context of bibliometrics , the PageRank algorithm works by treating each publication as a node in a network ,and the citations between publications as edges.The algorithm assigns a score to each publication based on the number and quality of the citation it receives . The more citations a publication receives from other highly cited publications,the higher its PageRank score.
The PageRank algorithm can also be used to identify influential authors,institutions,and journals within a particular field.By analyzing the citation network of publications, the algorithm can identify the most influential researchers in a given field. The information can be used to evaluate the impact of research , to identify emerging trends, and to support strategic decision-making in research management and funding.

4.3 Application in biology

The PageRank algorithm has been applied in various ways in biology to analyze and understand biological systems . One application of PageRank in biology is in the analysis of protein-protein interaction networks. Proteins are known to interact with each other in complex networks, and understanding the importance of each protein in these networks is crucial for understanding their functions. PageRank can be used to identify the most important proteins in these networks based on the number and quality of their interactions with other proteins. This information can then be used to identify potential drug targets or predict the effects of genetic mutations.
Another application of PageRank in biology that control their expression. PageRank can be used to identify the most important genes in

these networks based on their regulatory influence on other genes. This information can then be used to identify key regulatory pathways and potential therapeutic targets. [2] Overall, the application of PageRank in biology is in the analysis of gene regulatory networks. These networks describe the interactions between genes and the proteins has the potential to provide valuable insights into complex biological systems and aid in the development of new treatments for a wide range of diseases.

4.4 Application in Neuroscience

In neuroscience, PageRank has been used to analyze brain networks and identify the most influential brain regions. For example, a study published in the journal "NeuroImage" in 2016 used PageRank to analyze brain connectivity in individuals with Alzheimer's disease. The researchers found that the most influential brain regions in these individuals were those associated with memory, which is consistent with the known cognitive deficits in Alzheimer's disease. [6] Another study published in the journal "Frontiers in Human Neuroscience" in 2015 used PageRank to analyze functional brain networks in individuals with autism spectrum disorder (ASD). The researchers found that the most influential brain regions in individuals with ASD were those associated with social cognition, which is a known deficit in this disorder.

Overall, PageRank has proven to be a useful tool in analyzing brain networks and identifying important brain regions in various neurological and psychiatric disorders.

4.5 Application in Chemistry

PageRank algorithm has also found applications in the field of chemistry, particularly in the analysis of molecular networks and identifying the most important molecules or compounds in a network.

One of the most common applications of PageRank in chemistry is in an-

alyzing protein-protein interaction networks. Proteins are the building blocks of cells and are involved in various biological processes. By analyzing the interactions between different proteins, scientists can gain insight into the underlying biological mechanisms. [3] A study published in the journal "PLoS One" in 2012 used PageRank to analyze a protein-protein interaction network involved in DNA replication. The researchers found that the most important proteins in this network were those involved in regulating the initiation of DNA replication.
Another application of PageRank in chemistry is in analyzing chemical reaction networks. Chemical reactions are fundamental to many natural and industrial processes. By analyzing the relationships between different chemical reactions, scientists can gain insight into the underlying mechanisms and optimize these processes.
A study published in the journal "Physical Chemistry Chemical Physics" in 2013 used PageRank to analyze a chemical reaction network involved in the synthesis of ammonia. The researchers found that the most important reactions in this network were those involved in the conversion of nitrogen to ammonia.
Overall, PageRank has proven to be a useful tool in analyzing molecular networks and identifying the most important molecules or compounds in a network in chemistry.

4.6 My work

We can use PageRank algorithm to rank friendship between the students in a class. First we create a graph of friends, where each friend is represented by a node in a graph .An edge is created between two friends if they are connected or have some kind of relationship .Once the graph is created , the PageRank algorithm can be applied to calculate the importance of each friend based on their connections to other friends.
Here is how to rank friends using PageRank algorithm :
Lets consider a group of 10 friends :
Anooja, Alka, Ann, Bismi, Sanika,Rosmy,Nimisha,Sreelakshmi, Lakshmi and Saranya.

Anooja is friends with Sanika , Alka ,Rosmy and Lakshmi .
Alka is friends with Anooja ,Ann,Sanika and Nimisha .
Ann is friends with Alka, Nimisha ,Bismi and Sanika.
Bismi is friends with Nimisha ,Ann and Rosmy.
Sanika is friends with Anooja , Rosmy , Alka, Ann and Lakshmi.
Rosmy is friends with Lakshmi , Sanika , Anooja , Nimisha and Saranya.
Nimisha is friends with Bismi , Ann, Lakshmi , Alka, Sreelakshmi and Anooja.
Sreelakshmi is friends with Nimisha, Ann, Sanika , Bismi and Lakshmi.
Lakshmi is friends with Rosmy, Anooja, Nimisha , Saranya and Sreelakshmi.
Saranya is Friends with Lakshmi , Sreelakshmi, Rosmy and Sanika.

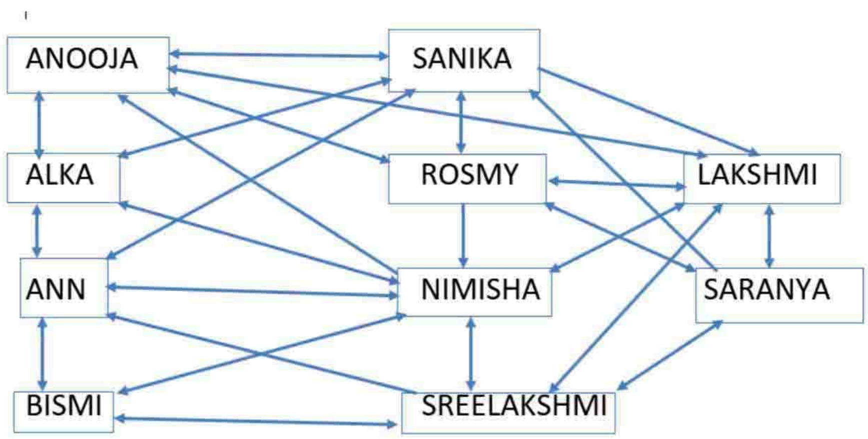

Figure 4.2:

Use the PageRank formula to calculate the updated PageRank value for each friend.The formula is :

PR(A) = (1-d)/N+d*(PR(T1)/C(T1)+. . . . . . +PR(Tn)/C(Tn))

Where:

PR(A) is the PageRank value of node A

d is the damping factor

N is the total number of nodes in the graph

T1. . . ..Tn are the nodes that link to node A

C(T1). . . ..C(Tn) are the number of outgoing links for each of those nodes.

here N = 10 ,d = 0.85, initial PR = 0.1

PageRank value for Anooja:

PR(Anooja)=(1-0.85)/10+d[PR(Sanika)/C(Sanika)+ PR(Alka)/C(Alka)+PR(Rosr

PR(Lakshmi)/C(lakshmi)]+PR(Nimisha)/C(Nimisha)

=(0.15)/10+0.85[0.1/5+0.1/4+0.1/5+0.1/6+0.1/5]

=0.1014

PageRank value for Alka:

PR(Alka) = (1-0.85)/10+d[PR(Anooja)/C(Anooja)+PR(Ann)/C(Ann)+ PR(Nimisha)/C(Nimisha)+PR(Sanika)/C(Sanika)]

=(0.15)/10+0.85(0.1/4+0.1/4+0.1/6+0.1/5]

=0.0886

PageRankvalue for Ann:

PR(Ann) = (1-0.85)/10+d[PR(Alka)/C(Alka)+PR(Bismi)/C(Bismi)+
PR(Nimisha)/C(Nimisha)+PR(Sanika)/C(Sanika)+ PR(Sreelakshmi/C(Sreelakshn
=(0.15)/10+0.85(0.1/4+0.1/3+0.1/6+0.1/5+0.1/5]
=0.11275

PageRank value for Bismi:

PR(Bismi)= (1-0.85)/10+d[PR(Ann)/C(Ann)+PR(Nimisha)/C(Nimisha)+
PR(Sreelakshmi/C(Sreelakshmi)]
=(0.15)/10+0.85(0.1/4+0.1/6+0.1/5]
=0.0674

PageRank value for Sanika:

PR(Sanika)= (1-0.85)/10+d[PR(Anooja)/C(Anooja)+PR(Ann)/C(Ann)+
PR(Alka)/C(Alka)+PR(Rosmy)/C(Rosmy)+ PR(Saranya)/C(Saranya)]
=(0.15)/10+0.85(0.1/4+0.1/4+0.1/4+0.1/5+0.1/4]
=0.117

PageRank value for Rosmy:

PR(Rosmy)= (1-0.85)/10+d[PR(Anooja)/C(Anooja)+ PR(Saranya)/C(Saranya)+
PR(Lakshmi)/C(lakshmi)+PR(Bismi)/C(Bismi)+ PR(Sanika)/C(Sanika)]
=(0.15)/10+0.85(0.1/4+0.1/4+0.1/5+0.1/3+0.1/5]
=0.1198

PageRank value for Nimisha:

PR(Nimisha)= (1-0.85)/10+d[PR(Alka)/C(Alka)+PR(Bismi)/C(Bismi)+
PR(Ann)/C(Ann)+PR(Sreelakshmi/C(Sreelakshmi)+ PR(Lakshmi)/C(lakshmi)+I
=(0.15)/10+0.85(0.1/4+0.1/4+0.1/5+0.1/5+0.1/5]
=0.1085

PageRank value for Sreelakshmi:

PR(Sreelakshmi)= (1-0.85)/10+d[PR(Saranya)/C(Saranya)+ PR(Lakshmi)/C(laks

PR(Bismi)/C(Bismi)]
=(0.15)/10+0.85(0.1/4+0.1/5+0.1/6+0.1/3]
=0.09575

PageRank value for Lakshmi:
PR(Lakshmi)= (1-0.85)/10+d[PR(Sanika)/C(Sanika)+PR(Anooja)/C(Anooja)+ PR(Nimisha)/C(Nimisha)+PR(Sreelakshmi/C(Sreelakshmi)+ PR(Saranya)/C(Sar
=(0.15)/10+0.85(0.1/5+0.1/4+0.1/6+0.1/4+0.1/5]
=0.1056

PageRank value for Saranya:
PR(Saranya)= (1-0.85)/10+d[PR(Sreelakshmi/C(Sreelakshmi)+ PR(Lakshmi)/C(l
=(0.15)/10+0.85(0.1/5+0.1/5+0.1/5]
=0.066

Sort the friends in descending order based on their updated PageRank values.The friend with the highest PageRank value is considered the most influential in the group .In this case , the ranking of friends based on their PageRank values would be:
1.Rosmy -0.1198
2.Sanika-0.117
3.Ann-0.1127
4.Nimisha-0.1085
5.Lakshmi-0.1056
6.Anooja-0.1014
7.Sreelakshmi-0.0957
8.Alka-0.0886
9.Bismi-0.0674
10.Saranya-0.066

4.6.1 Python program

```
import numpy as np

d = 0.85
num_iterations = 1

M = np.array([[0, 1/4, 0,0,1/5,1/5,1/6,0,1/5, 0],
              [1/4,0,1/4,0,1/5, 0,1/6,0,0 ,0],
              [0, 1/4,0,1/3,1/5,0,1/6,1/5,0,0],
              [0,0,1/4,0,0,0,1/6,1/5,0,0],
              [1/4,1/4,1/4,0,0,1/5,0,1/5,1/5,0],
              [1/4,0,0,1/3,1/5,0,0,0,1/5,1/4],
              [0,1/4,1/4,1/3,0,1/5,0,1/5,1/5,0],
              [0,0,0,0,0,0,1/6,0,1/5,1/4],
              [1/4,0,0,0,1/5,1/5,1/6,1/5,0,1/4],
              [0,0,0,0,0,1/5,0,0,0,1/4]])

v = np.array([[0.85],[0.85],[0.85],[0.85],[0.85],[0.85],[0.8

for i in range(num_iterations):
  v = (1-d)/4+d*np.dot(M,v)

for i, page_rank in enumerate(v):
  print(f"Page {i+1}: {page_rank[0]:.4f}")
```

Output is:

```
Page 1: 0.7720
Page 2: 0.6637
Page 3: 0.8684
Page 4: 0.4830
Page 5: 1.0129
Page 6: 0.9286
Page 7: 1.0731
Page 8: 0.4830
Page 9: 0.9527
Page 10: 0.3626
```

Figure 4.3:

Ranking of friends here is :

Sanika = 1.0129

Nimisha = 1.0731

Lakshmi=0.7562

Anooja=0.7449

Rosmy=0.7006

Ann=0.6762

Alka=0.6514

Bismi=0.3915

Sreelakshmi=3664

Saranya0.2228

From this i conclude that Power iteration method is preferred over iteration method since it converges faster and is more efficient.

CONCLUSION

The Page Rank algorithm is a powerful tool developed by Larry Page and Sergey Brin, the founders of Google, for ranking web pages based on their importance and popularity. In this project, I have studied the fundamental concepts behind the Page Rank algorithm, including the importance of links between web pages, iterative and power iterative methods, and provided examples, merits, and demerits of both methods.

Apart from ranking web pages, the Page Rank algorithm has various applications in different fields such as ranking tweets in twitter, bibliometrics, neuroscience, chemistry, and biology. I mainly focused on the methods and applications of the Page Rank algorithm, and I ranked a group of friends in a class based on their bonding to one another. For this task, I used iterative and power iterative methods and concluded that the power method was more efficient and easy to converge compared to the iterative method, which was time-consuming and required lots of manual steps to converge.

This project was an overall study of the Page Rank algorithm, covering all its aspects and providing an overview of the topic.

LITERATURE REVIEW

There have been many studies and evaluations of the Page Rank algorithm over the years and some of them are:-
In 1998, Rajeev Motwani and Terry Winograd co-authored with Page and Brin published the first paper about the project, describing PageRank and the initial prototype of the Google search engine.

In 1999, Larry published a paper 'The PageRank citation Ranking: Bringing order to the web' in his paper he describes Page Rank , a method for rating webpages objectively ,mechanically and effectively measuring the human interest and attention devoted to them.

In 2010, Dilip Kumar Sharma and Ashok Sharma in an article published a paper on 'A Comparative Analysis of web page ranking algorithm'. The paper deals with analysis and comparison of web page ranking algorithms based on various parameters to find out their advantages and limitations of ranking web pages.

In 2015, MS Mariani published a paper on 'Ranking nodes in growing network: when PageRank fails'. In this paper he studies Page Rank's performance on a network model supported by real data and shows that realistic temporal effects make Page Rank fail in individuating the most valuable nodes for a broad range of model parameters.

In 2014, Madhurdeep kaur and Charanjit Sing published an article: 'A Hybrid Page Rank Algorithm'. In this paper, a Page Rank mechanism called Hybrid Page Rank Algorithm is proposed which is based on both content and link structure of the webpages. This algorithm is used to find more relevant information according to the user's query. This paper also presents the comparison between the Sim Rank algorithm and the Hybrid Page Rank algorithm.

In 2020, Prem Sharma , Divakar Yadav and Pankaj Garg published a

journal 'A Systematic review on Page Ranking algorithms ' in International Journal of information technology . This paper critically surveys various content based on Page Ranking algorithms , Ink structure based Page Ranking algorithm and Hybrid Page Ranking algorithm which have been proposed by many researchers in recent years .

In 2022, Moath Abu Dayeh, Badie Sartawi and Saeed Salah published a journal ' A Bais-free Time-Aware PageRank Algorithm for Paper Ranking in Dynamic Citation Networks '.This paper focuses on solving the problem of bias by proposing a new ranking algorithm based on the PageRank algorithm.

REFERENCES

[1] Rebecca S. Wills, Google's PageRank: The math behind the search engine .In Journal Math. Intelligencer , pages 6-10,2006.

[2] Bioinformatics, Volume 27, Issue 3, February 2011, Pages 405–407, https://doi.org/10.1093/bioinformatics/btq680

[3] PageRank in chemistry, 2012; PageRank in chemical reactions, 2013

[4] Ganit charcha , Mathematics Behind Google's Search Engine,2015

[5] Limitations of Page Rank ,2015 .Retrieved from https://blogs.cornel.edu

[6] PageRank in neuroscience, NeuroImage, 131, 113-123 (2016) .

[7] Krishna S Kumar , M P Vrinda and V R Rajalakshmi ,An Extention To Page Rank Algorithm ,volume 114 No.12 pages 21-28 merits and demerits,2017

[8] Ashish Goel ,MS and E 233 Lecture 8: Application of Page Rank to Recommentation System ,pages 1-4,2018.

[9] Brian Moor , Mathematics Behind Google's Page Rank Algorithm , pages 2-8,2018.

[10] Armani Amine, Page Rank Algorithm and its Implementation ,Towards Data Science ,2020

[11] Ilse Ipsen, The Mathematics Behind Google's PageRank.

[12] Parth Chavan ,Perron Frobenius Theorem and Google Page Rank November 22, 2020

[13] https://en.wikipedia.org/wiki/Perron