



Software Project Lab - 1

Google's PageRank Algorithm

Presented by

Shah Alam Abir

Roll: 1439

Session: 2021-2022

Supervised by

Dr. Ahmedul Kabir

Associate Professor

IIT, University of Dhaka

Table of
contents

CONTENTS

1 Introduction

2 The web as a Graph

3 Why we should RANK webpages

4 Random Surfer Model

5 PageRank Formula



1

Introduction

● Introduction

- Google's PageRank Algorithm is an algorithm used by Google Search Engine to rank web pages in search engine results.
- It was developed by **Larry Page** and **Sergey Brin**, the co-founders of Google, while they were graduate students at Stanford University.

● Introduction

- PageRank is designed to assess the importance and relevance of web pages in the context of search engine results.
- The underlying idea is that valuable and authoritative web pages are more likely to be linked to by other web pages.
- Therefore, a web page with many high-quality incoming links is considered more valuable and is likely to rank higher in search results.

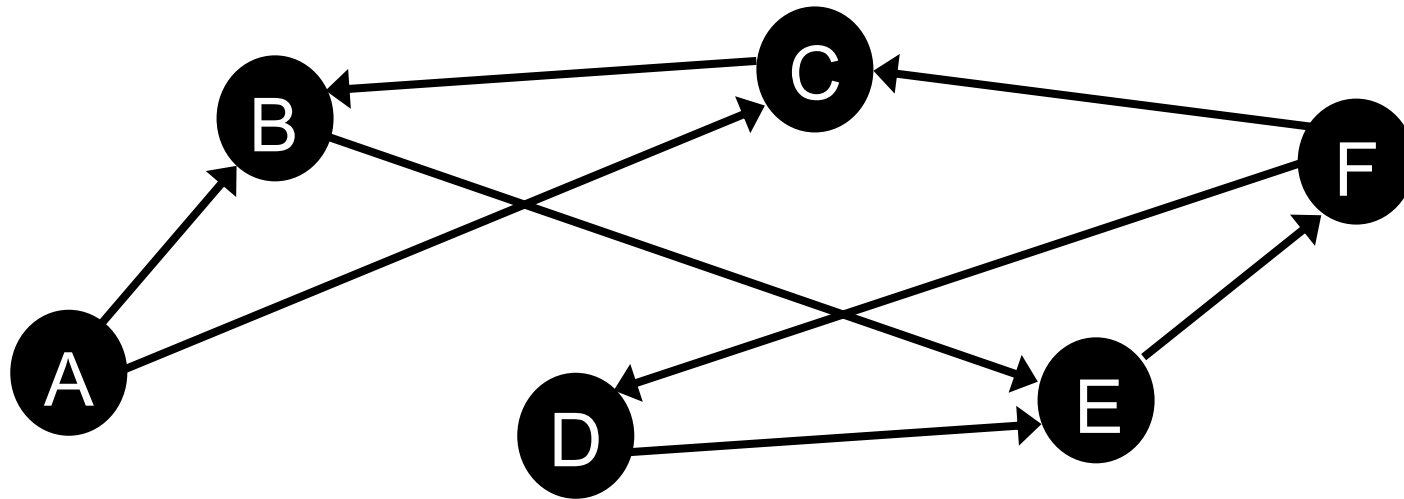


2

The web as a Graph

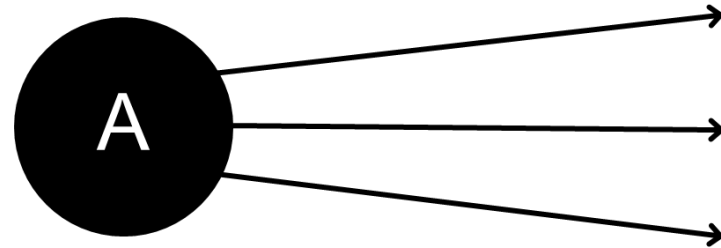
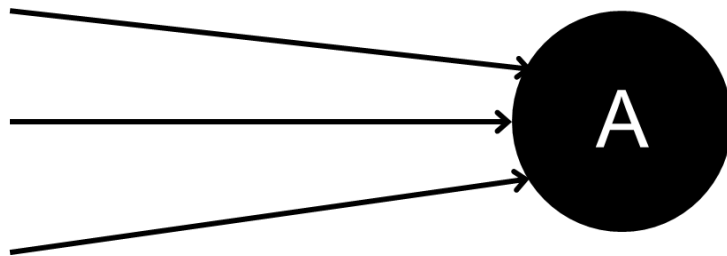
● The web as a Graph

- We can represent WWW's structure as a huge directed graph, where node's of that graph are the webpages and edges are the links between webpages.



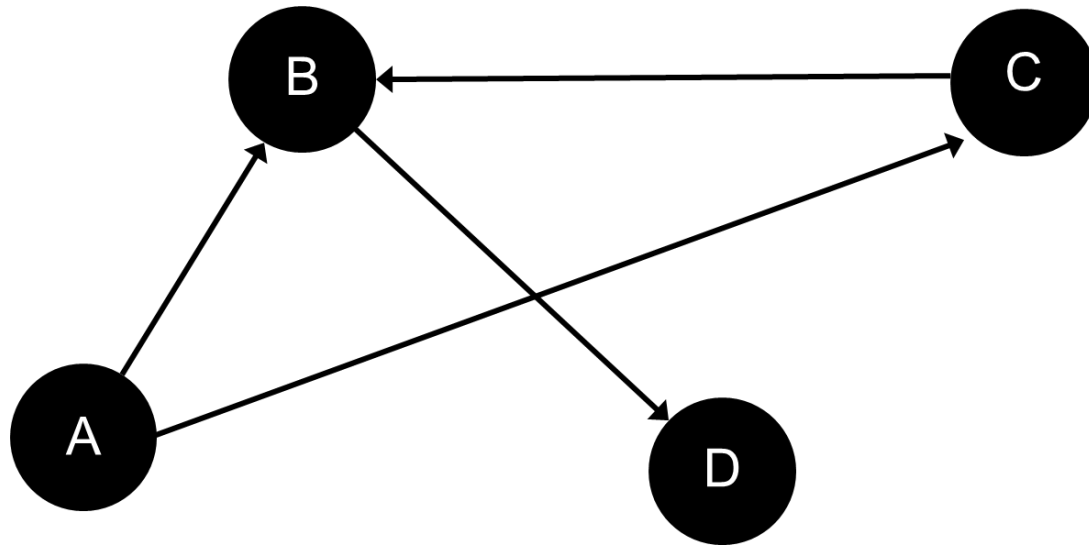
● The web as a Graph

- **Inbound links** are links that point from another website to the target website
- **Outbound links** are links that point to another websites from target website



● The web as a Graph

- **Dangling Nodes** are the nodes with no outbound links
- **Dangling Links** are links that point to the dangling nodes



- Here **D** is a Dangling Node
- **BD** is a dangling Link



3

Why we should RANK webpages

● Why we should RANK webpages

- Relevance
- Efficiency
- Accessibility
- Academic and Research Insights
- Trustworthiness

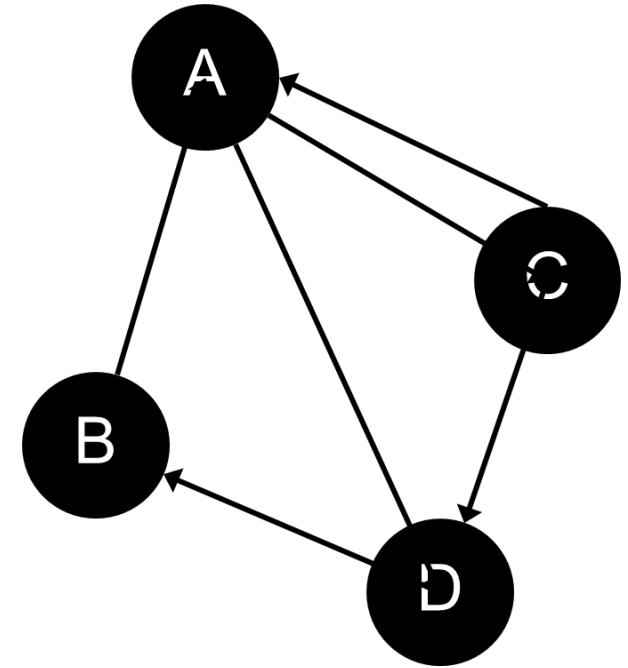


4

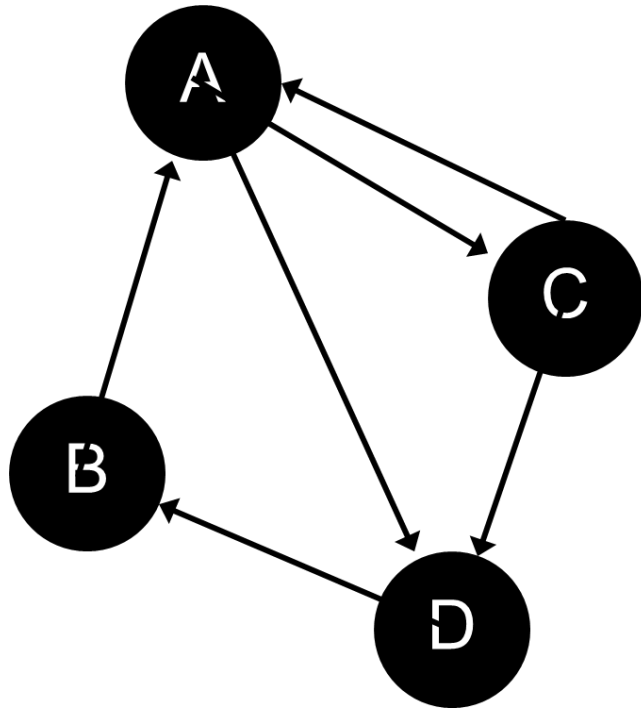
Random Surfer Model

● Random Surfer Model

- The Random Surfer Model is a conceptual framework used in the context of web page ranking algorithms, particularly in understanding Google's PageRank algorithm. It is a simplified representation of how a hypothetical internet user behaves when navigating the web and is used to explain the concept of PageRank.



● Random Surfer Model



Matrix Presentation

$$r_{(k+1)}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$$

The PageRank of a page P_i , denoted $r(P_i)$, is the sum of the Page-Ranks of all pages pointing into P_i .

where B_{P_i} is the set of pages pointing into P_i and $|P_j|$ is the number of out-links from page P_j .

	A	B	C	D
A	0	0	0.5	0.5
B	1	0	0	0
C	0.5	0	0	0.5
D	0	1	0	0

Transition matrix **H**

● Random Surfer Model (Iterative Calculation)

Initializing page rank for every page with $1/n$. Where, n is the total number of webpages of the graph. Thus, we will have another matrix **X** with initialized page-rank value.

From previous slide we got Transition matrix **H**

We will perform matrix multiplication $\mathbf{X} = \mathbf{H}\mathbf{X}$ iteratively until it converge. Then we will get a modified page-rank matrix **X**. Now, if we sort them according to probability, then we will get webpages in descending order.

A	$\frac{1}{4}$
B	$\frac{1}{4}$
C	$\frac{1}{4}$
D	$\frac{1}{4}$

Matrix **X**

● Problems with Random Surfer Model

But, there is two major problem with this Random Surfer Model. These are –

1. If there is any dangling node [webpage], then all the elements of the final matrix will be 0.
2. If some nodes of the graph is highly connected, then probability of these particular group of nodes becomes significantly higher than other nodes. Because of this, a random surfer iterates through a group of nodes again and again.

5

PageRank Formula

● PageRank Formula

To solve first problems-

We have to convert **H** matrix to a *Stochastic* matrix.
This *Stochastic* matrix solves the problem of dangling node by initializing nodes probability with a value. If **S** is a *Stochastic* matrix, then-

$$\mathbf{S} = \mathbf{H} + \mathbf{a}(1/n\mathbf{e}^T)$$

Here,

S, Stochastic Matrix

H, Transition Matrix

a, is a (nx1) matrix. Value of a_i is 1 if a dangling node else 0
 \mathbf{e}^T , is a (1xn) matrix. Every element is 1.

● PageRank Formula

To solve second problems-

We have to make **GOOGLE (G)** matrix from **S** matrix & **H** matrix. This **G** matrix ensures teleportation from the dense part to the sparse part of the graph. If **G** is a **GOOGLE** matrix, then-

$$\mathbf{G} = \alpha \mathbf{S} + (1 - \alpha) \frac{1}{n} \mathbf{e} \mathbf{e}^T = \alpha \mathbf{H} + (\alpha \mathbf{a} + (1 - \alpha) \mathbf{e}) \frac{1}{n} \mathbf{e}^T$$

Here,

G, GOOGLE Matrix

S, Scholastic Matrix

H, Transition Matrix

a, is a (nx1) matrix. Value of a_i is 1 if a dangling node else 0

e, is a (nx1) matrix. Every element is 1

e^T , is a (1xn) matrix. Every element is 1.

α , is the damping factor. It ensures the teleportation.

Damping Factor

The damping factor in the PageRank algorithm is a parameter that represents the probability that a random surfer, navigating the web by clicking on links, will continue to browse and click on links within the current web page rather than jumping to a completely random page.

● PageRank Formula

Now, we got our final formula to make a **GOOGLE matrix** by doing *Stochastic adjustment* & *Primitive adjustment* of the initial *Transition matrix*, **H**.

$$\mathbf{G} = \alpha \mathbf{S} + (1 - \alpha) \frac{1}{n} \mathbf{e} \mathbf{e}^T = \alpha \mathbf{H} + (\alpha \mathbf{a} + (1 - \alpha) \mathbf{e}) \frac{1}{n} \mathbf{e}^T$$

Now, We will perform Iterative matrix multiplication $\mathbf{X} = \mathbf{H}\mathbf{X}$ until it converges. Then we will get a modified page-rank matrix, **X**.

Now, if we sort matrix **X** according to probabilistic value, then we will get webpages in descending order.



Thank you