PageRank Algorithm

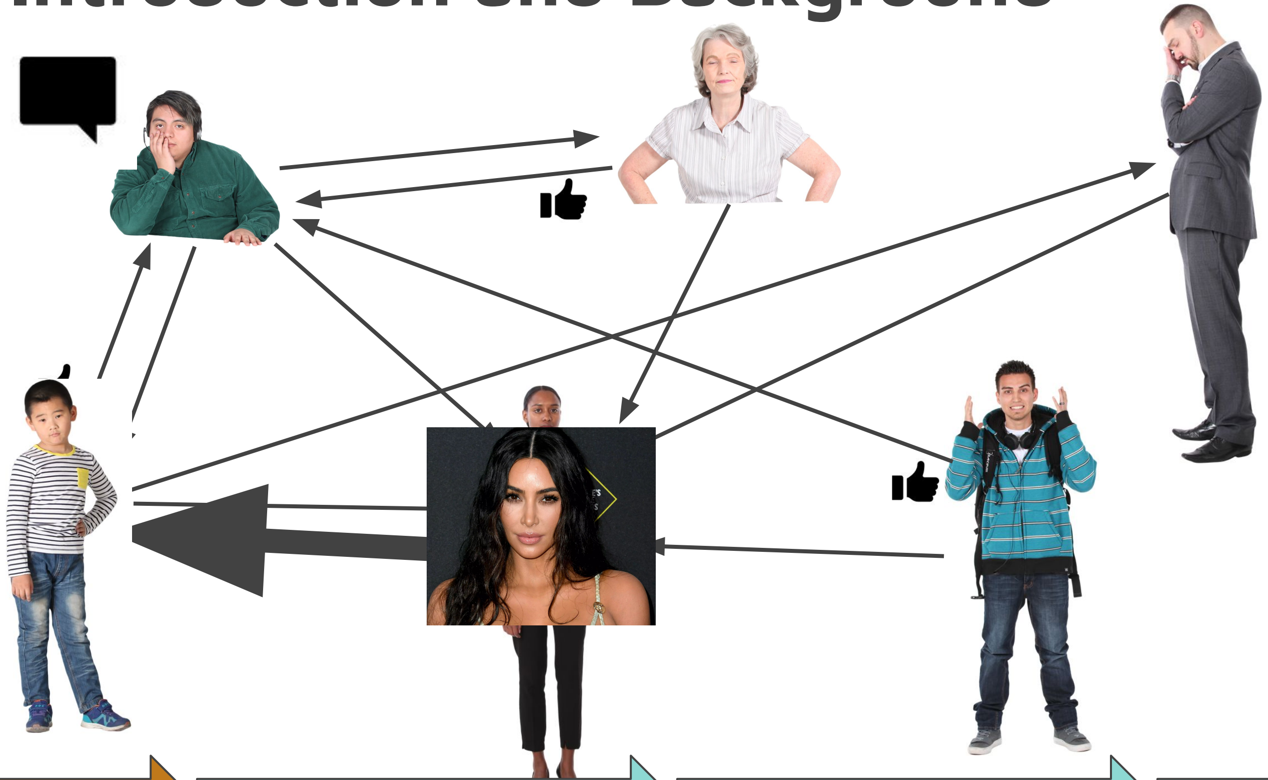
MIE 1624 - Introduction to Data Science and Analytics

Group 2:
Arshdeep Bamrah, Yuan Chen, Luke Fregonese, Qisheng He, Naiyu Hu, Monica Leng, Sugumar Prabhakaran

Agenda

1. Introduction and Background (Monica)
2. Mathematics (Arshdeep and Charles)
3. Implementation (Sugumar and Luke)
4. Variants (Yuan and Qisheng)

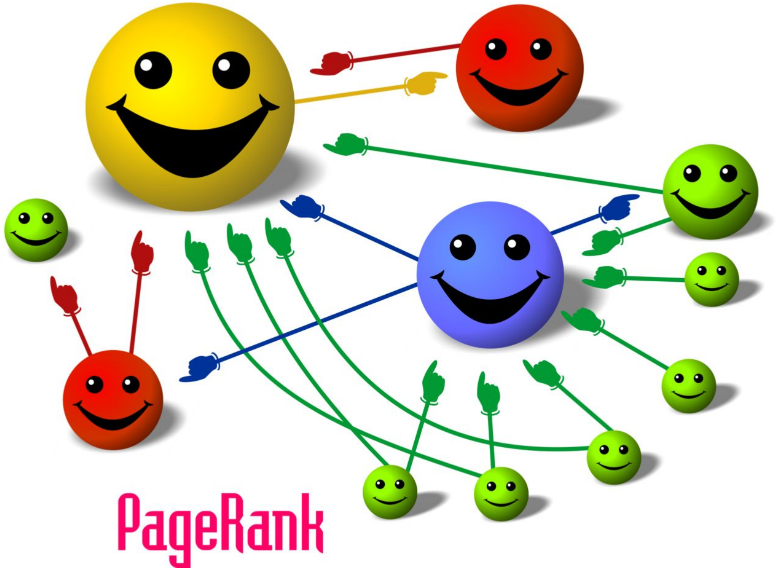
Introduction and Background



Introduction

https://www.businessinsider.in/entertainment/news/people-are-dragging-kim-kardashian-for-describing-life-as-paradise-on-a-day-the-us-recorded-a-record-number-of-covid-19-deaths/articleshow/80376504.cms

Introduction and Background

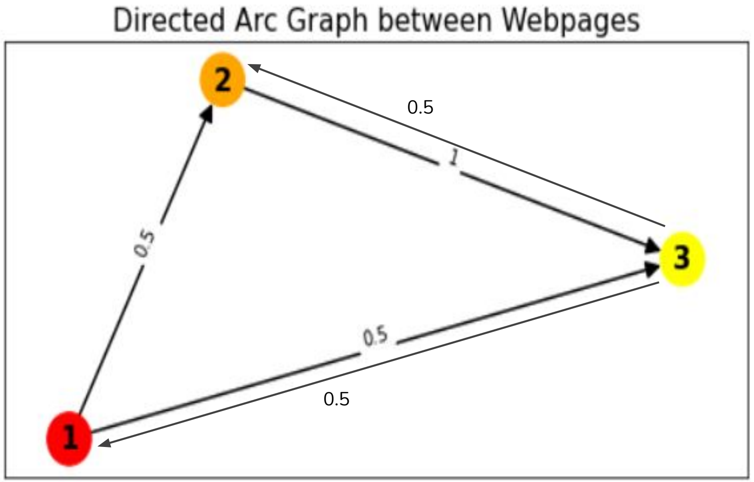


Introduction

https://www.theverge.com/2019/12/4/20994361/google-alphabet-larry-page-sergey-brin-sundar-pichai-co-founders-ceo-timeline
https://en.wikipedia.org/wiki/PageRank

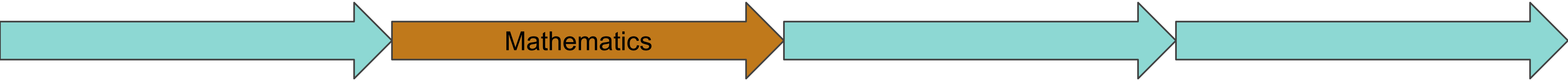
Simplified Algorithm

Divide the score of a page by the # of outgoing links, and equally assign to its destinations

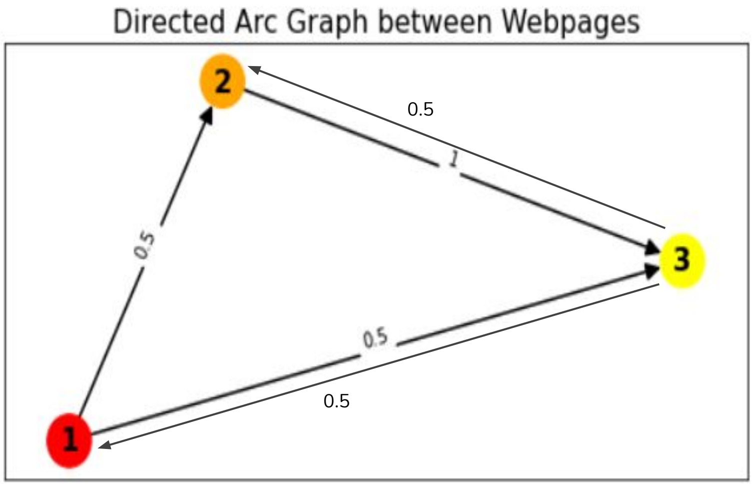


$$PR(p_i) = \sum_{p_j \in W(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- Pi - individual webpages
- PR(Pi) - PageRank of Pi
- W(Pi) - set of pages that link to Pi
- L(Pi) - number of outbound links of Pi



Markov Chain/Stochastic Matrix



i\j	1	2	3
1	0	0	0.5
2	0.5	0	0.5
3	0.5	1	0

Probability Matrix: $$M_{ij} = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 1 & 0 \end{bmatrix}$$

Mij is a pattern of Markov Chain/Stochastic Matrix

$$M\vec{x} = \lambda\vec{x}$$

M = Matrix
x = Eigenvector
λ = Eigenvalue

Mathematics

Eigenvalue and Eigenvector

Perron-Frobenius Theorem: If M is a positive, column stochastic matrix, then:

1. Eigenvalue equals to 1.
2. For the eigenvalue 1 there exists a unique eigenvector with the sum of its entries equal to 1.

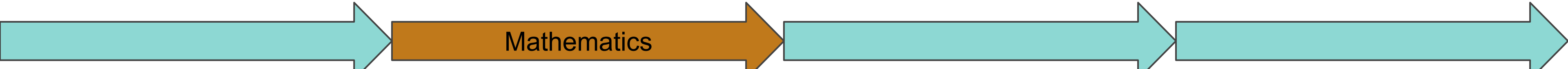
Simplified Algorithm:

$$R = M \times R$$

$$M_{ij} = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 1 & 0 \end{bmatrix}$$

$$R_0 = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ PR(p_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

The simplified algorithm could be solved with M as a Markov Chain / Stochastic matrix. R is the eigenvector, where the eigenvalue is 1.



http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html

Power Method

Power Method Convergence Theorem: Let M be a positive, column stochastic n × n matrix and R be a probabilistic eigenvector corresponding to the eigenvalue 1 and with all entries equal to 1/n. Then the sequence R, MR, ..., M^kR converges to the vector R^*.

Starting Eigenvector Ro :

$$R_0 = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ PR(p_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

Adjacency Matrix Mij :

$$M_{ij} = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 1 & 0 \end{bmatrix}$$

$$R_1 = M_{ij} \times R_0 = \begin{bmatrix} 0.167 \\ 0.333 \\ 0.5 \end{bmatrix}$$

$$R_2 = {M_{ij}}^2 \times R_0 = \begin{bmatrix} 0.25 \\ 0.333 \\ 0.417 \end{bmatrix}$$

$$R_3 = {M_{ij}}^3 \times R_0 = \begin{bmatrix} 0.208 \\ 0.333 \\ 0.458 \end{bmatrix}$$

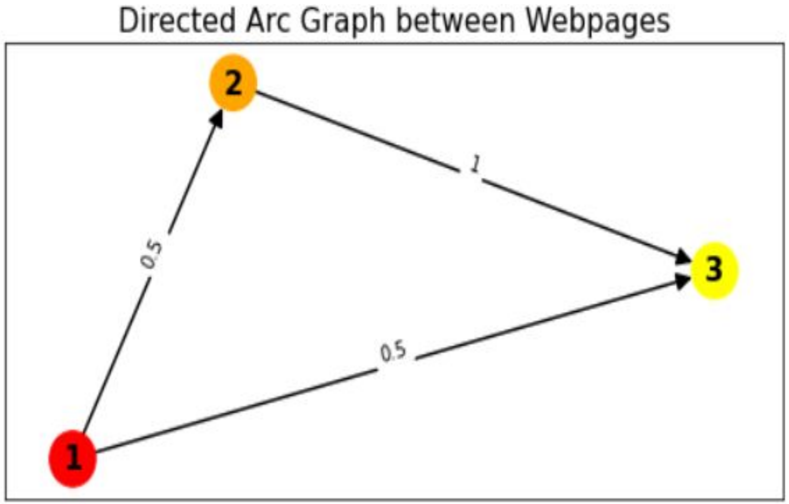
...

$$R_{100} = {M_{ij}}^{100} \times R_0 = \begin{bmatrix} 0.222 \\ 0.333 \\ 0.444 \end{bmatrix}$$

Mathematics

http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html

Issue with the Simplified Algorithm



Problem:

- Sink node (webpage 3) with no outgoing flow resulting in lack of balance of the network
- In this scenario, the PageRank of all webpages converges to 0

$$R_0 = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ PR(p_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} \qquad M_{ij}' = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 1 & 0 \end{bmatrix}$$

$$R_{100} = (M_{ij}')^{100} \times R_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Oops!!

Mathematics

Enhancing the Simplified Algorithm

Simplified Algorithm:

- Divide the score of a page by the # of outgoing links, and equally assign to its destinations

$$PR(p_i) = \sum_{p_j \in W(p_i)} \frac{PR(p_j)}{L(p_j)}$$

Enhancement - Damping Factor:

- Divide the score of the sink node in the graph by total number of node, and equally assign to each node
- Adopt the same approach for all webpages in the network, but against only a portion of its total score (1 - d)
- Various researches recommend a desired damping factor of 0.85

General Equation:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in W(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- Pi - individual webpages
- PR(Pi) - PageRank of Pi
- d - damping factor, typically 0.85
- N - total number of pages
- W(Pi) - set of pages that link to Pi
- L(Pi) - number of outbound links of Pi

Mathematics

Damping in Power Method

Solution of the Google co-founders - Page and Brin

General Equation:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in W(p_i)} \frac{PR(p_j)}{L(p_j)}$$

General Equation in Matrix Form:

$$R_{i+1} = \left(\frac{1-d}{N} \times \mathrm{E} + \mathrm{d}M\right) \times R_i$$

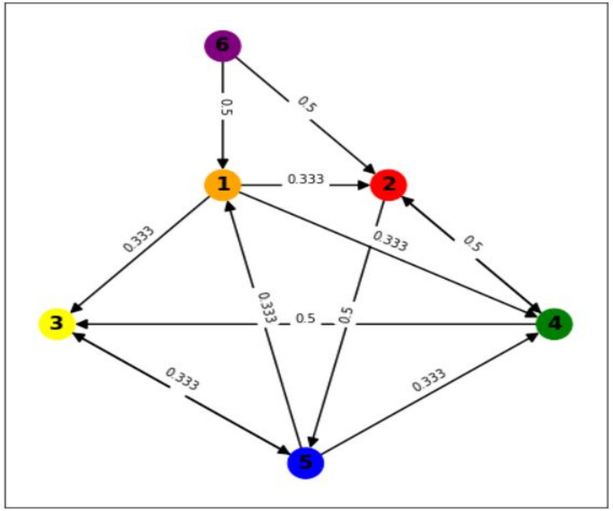
$$M_{damped} = \frac{1-d}{N} \times \mathrm{E} + \mathrm{d}M$$

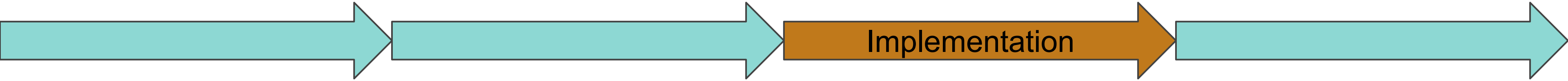
$$\Longrightarrow \quad R_{i+1} = M_{damped} \times R_i$$

, where ERi=1, |R|=1, and all entries of E are 1

Implementation

Code walkthrough of the Python implementation





Weighted PageRank

- Proposed by Wenpu Xing and Ali Ghorbani in 2004.
- Allows distribution of the page rank according to the importance or the popularity of the webpage
- Assigns a weight value to each edge
- If each edge has the same weight, this is identical to the original PageRank algorithm.

Variants

https://www.ijser.org/researchpaper/A-Literature-Survey-on-Page-Rank-Algorithm.pdf
https://pdfs.semanticscholar.org/70f8/d27954ce7ef49e351aca6d4b6b368cfef1c7.pdf

Weighted PageRank

$$PR(u) = (1-d) + d\sum_{v \in B(u)} \frac{PR(v)}{N_v} W^{in}(v,u) W^{out}(v,u)$$

$W^{in}(v,u) = \frac{I_u}{\sum_{p \in R(v)} I_p}$ where:

- I_u and I_p are number of inlinks of pages u and p, respectively
- $R(v)$ is the set of pages pointed by v

and

$W^{out}(v,u) = \frac{O_u}{\sum_{p \in R(v)} O_p}$ where:

- O_u and O_p are number of outlinks of pages u and p, respectively
- $R(v)$ is the set of pages pointed by v

Advantages:

- Takes into account the importance of both the inlinks and outlinks of the pages
- Distributes rank scores based on the popularity of the pages
- Converges very fast

Limitations:

- Does not consider user access pattern

Variants

https://www.ijser.org/researchpaper/A-Literature-Survey-on-Page-Rank-Algorithm.pdf
https://pdfs.semanticscholar.org/70f8/d27954ce7ef49e351aca6d4b6b368cfef1c7.pdf

PageRank based on Visits of links (VOL)

- Proposed by Gyanendra Kumar, Neelam Duhan, A. K. Sharma in 2011 at International Conference on Computer & Communication Technology (ICCCT)-2011.
- Assigns more value to the outgoing links that are most visited by users

https://www.ijser.org/researchpaper/A-Literature-Survey-on-Page-Rank-Algorithm.pdf
https://pdfs.semanticscholar.org/70f8/d27954ce7ef49e351aca6d4b6b368cfef1c7.pdf

PageRank based on Visits of links (VOL)

$$PR(u) = (1-d) + d \sum_{v \in B(u)} \frac{PR(v)L_u}{TL(v)}$$

- $B(u)$ is the set of pages pointing to u
- L_u is the number of visits of links which are pointing from v to u
- $TL(v)$ is the total number of visits of all links from v

Advantages:

- Displays most valuable pages on the top of the result list based on user browsing behaviour

Limitation:

- Converges slower compared to weighted PageRank

https://www.ijser.org/researchpaper/A-Literature-Survey-on-Page-Rank-Algorithm.pdf
https://pdfs.semanticscholar.org/70f8/d27954ce7ef49e351aca6d4b6b368cfef1c7.pdf

Variants

- Weighted PageRank
- PageRank based on visits of links (vol)
- Weighted PageRank based on visits of links
- Personalized PageRank
- Personalized Weighted PageRank
- Topic sensitive PageRank
- Ratio based Weighted PageRank
- etc.

Variants

https://www.ijser.org/researchpaper/A-Literature-Survey-on-Page-Rank-Algorithm.pdf
https://docs.oracle.com/cd/E56133_01/2.4.0/reference/algorithms/pagerank.html

Questions?