Analysing the Algorithm: How Linear Algebra Shaped the Success of Google's PageRank Algorithm

Shailender Goyal
Department of Computer Science Engineering
IIIT Hyderabad Team-74
shailender.goyal@research.iiit.ac.in

Zainab Raza
Department of Electronics and Communication Engineering
IIIT Hyderabad Team-74
zainab.raza@students.iiit.ac.in

Soham Vaishnav
Department of Electronics and Communication Engineering
IIIT Hyderabad Team-74
soham.vaishnav@research.iiit.ac.in

***Abstract*—This paper aims to explore the mathematical aspects of the PageRank algorithm, its analysis and its applications in today's times. Towards the end, the study of one of the platforms where the algorithm is extensively put into effect is presented along with some key mathematical (Linear Algebra) concepts that form the foundation of PageRank.**

***Index Terms*—PageRank, World Wide Web (WWW), Internet, hyper-linking, Markov chains, hyperlink matrices, adjacency matrix, Stochastic matrices, Eigen-values, Eigen-vectors, graphs**

I. Introduction

With the onset of the Digital Era, if there is one thing to have revolutionised the way communication and sharing of information is perceived, it has to be the invention of the Internet and the World Wide Web, which binds (we will return to this soon) the network.

The very idea that something as dynamic, and as virtually realistic as Internet is possible itself stands out as a creation unparalleled. And the invention of World Wide Web, in addition, only added value to Internet. With a magnanimous amount of data available in its entirety almost at the blink of an eye, organisation was the key!

The WWW, although served as a platform for creators to host their content, it didn't do too well to satisfy the demands of the users because of the fact that the way the data was channelised to a user based on one's requirement was not too efficient. After all, at the core of it, the WWW was simply supposed to present the data regardless of any order, and it did serve that purpose well - unorganised nonetheless.

However, a research paper in the year 1998, titled "The Anatomy of a Large-Scale Hyper-textual Web Search Engine" by Sergei Brin and Larry Page proposed a new search engine based on an algorithm that they defined as the **PageRank**. The definition of the algorithm is quite intuitive given the name it has been given - to rank the pages/websites being hosted on the WWW. A simple, yet an effective idea that would help organise the bulk of data available and make surfing a worthy experience (in terms of time spent and information gained).

The world wide web, as stated earlier, **binds** the internet, so as to say. The web hosts the websites (the pages available on the web and hence the name) by using a concept of **hyper-linking**.

This concept will be touched upon in detail in one of the sections in this paper, but here we must present a brief insight into what it is.

Hyper-linking roots from the word **hyper-link**. Hyper-links are the channels via which two pages are connected, that is, one page may contain links that would redirect the user to another site and that new page may contain some suggestions about what sites can the user visit next, and so on. This network is what builds the web of information. However, the networking is not as simple as it seems, as it brings along a certain set of rules that these links must follow in order for the algorithm to work well. The discussion of these rules and properties is not in the scope of this section and will be explored across sections in the rest of the paper.

Next comes the probabilistic model that will form the basis of the algorithm, and the necessity of its occurrence at such an instant is quite intuitive. The model for ranking of websites and hence organising the web, is fundamentally of no particular use if it does not provide insights into the future of what shape will the web take. To exemplify, a website highly preferred currently by users may not be the one best suited to be ranked first because of reasons, one of which can be that the only way to reach a useful/resourceful site is via this popular site. So the users are not left with a choice but to visit the useless site. But, since this site is not too useful, the sites linking to this site would be quite less. So, if the algorithm just takes into consideration the current scenario without looking at the possible future, it would render a worthless experience to the user. Therefore, alongside the current scenario, taking a broader outlook into the **next** possible states would help largely.

Therefore, a probabilistic model is unavoidable. The properties of this model have been discussed in sections across the paper.

With foundational elements discussed thus far and explored further, the **PageRank** algorithm serves as a very powerful tool in building the backbone of Google's search engine (discussed in the paper).

II. Problem Statement

Before 1998, the web, or more precisely the World Wide Web was just a highly entangled space of information, with negligible or zero classification of data that would render a worthy experience to the users in terms of time spent and content gained. Earlier, the web-pages were suggested based on the traditional content score which can be explained as the amount of content retrieved or gained from visiting a particular site. To elaborate on it, the sites that contained the maximum amount of information related to the query put forth by the user were suggested first. But, people found a loophole in the structure and therefore exploited the very basis of the existing ranking system.

For a particularly popular query, rather than updating and actually increasing the valuable information content on their website, people tended to just increase the usage or occurrence of that particular query or of a few words of it in their content. This raised a false impression on the system about the website which tricked the system into thinking that the site really contained an ample amount of useful content helpful to the user, and thereby the entire facade led to that website being ranked above other ones in the competition. And, by 1998, the web had already fallen prey to such false-full and malicious attacks by hackers.

Overall, it was clear that by relying just on the traditional content scores, however useful they are, is not the only metric or measure that can be used to define the usefulness of a page, and the over-reliance of the current system was exploited at the expense of user's time.

Therefore, there arose a need to define a new model, that combined the existing structure and its foundations, as well as brought into picture the popularity score. To define what a popularity score is, a work-place analogy would do well :
Intuitively, for a work to get done effectively and efficiently, most of the people in a work-place would recommend reaching out to a few set of people who would have better knowledge or hold of the matter. This implies that along with quality, popularity plays a role.
Thus, a model had to be adopted which would appropriate place weights to popularity as well as quality and deliver a notable service to the user.

Another aspect of the service was retrieval of information in as less time as possible, and this hinted at the possibility of a model that would process all or most of the concerned websites, along with some fluctuating parameters like change in content and availability of that site, and derive the results to present them to the users.

What follows in the paper is a comprehensive visualisation of the model with factors discussed herein and exploration of the mathematical and theoretical aspects of it.

III. Applications of Linear Algebra

We will now provide a comprehensive exploration of the applications of linear algebra in the PageRank algorithm. By examining the mathematical foundations and computational techniques involved, we can appreciate the significance of linear algebra in revolutionizing web search and understand how PageRank has become a cornerstone in ranking web pages based on their importance and relevance.

A. Markov Chain

Markov chains is a mathematician model describing possible events where the probability of each event depends only on the state attained in the previous event. At each time t the system moves from state v to u with probability p_{uv} that does not depend on t. p_{uv} is called as transition probability which is an important feature of Markov chain.

Markov chain uses only a matrix and a vector to model and predict it. Markov chains play a crucial role in the PageRank algorithm. The algorithm utilizes the concept of Markov chains to model and analyze the link structure of the web.

B. Transition Matrix

A transition matrix is a square matrix where each element represents the probability of transitioning from one state to another state. The elements of the matrix must satisfy certain conditions:

1) Each element must be a non-negative value between 0 and 1.
2) The sum of elements in each row must be equal to 1. (Stochastic Matrix)

C. Eigen Vectors

Given a square matrix A, an eigenvector is a non-zero vector v such that when A is multiplied by v, the result is a scalar multiple of v. Mathematically, this can be expressed as:

$$Av = \lambda v$$

Here, λ is the eigenvalue corresponding to the eigenvector v. Eigenvalues are the scalar factors by which the eigenvectors are scaled when multiplied by the matrix A.
A transition matrix always has 1 as an eigenvalue and there exists an eigenvector with eigenvalue 1 such that, a distribution v over the states is called stationary distribution of the Markov chain with transition matrix A if

$$v = vA$$

D. Graph Theory

A **Graph** is a mathematical object that comprises a non-empty set of vertices and a separate set of edges.

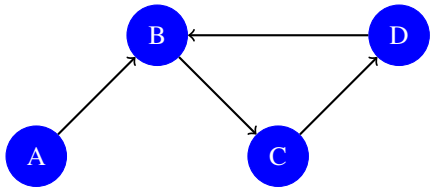

Fig. 1. A Directed Graph

A **directed** graph, also known as a digraph, consists of a set of nodes and a collection of directed edges, each connecting an ordered pair of vertices. In a directed graph, any two vertices, denoted as i and j, are considered **adjacent** if there exists an

edge from i to j or from j to i. This adjacency relationship can be represented effectively using an **adjacency matrix** in linear algebra.

A graph is said to be **connected** if, for any two distinct nodes i and j, there exists a directed path either from i to j or from j to i.

An **adjacency matrix** is a square matrix used to represent the connectivity of a graph in the form of matrix that can computed upon. For a directed graph with n vertices, the adjacency matrix is an $n \times n$ matrix where each element a_{ij} represents the presence or absence of a directed edge from vertex i to vertex j.

If there is a directed edge from vertex i to vertex j, then a_{ij} is assigned a non-zero value (e.g., 1) to indicate the connection. On the other hand, if there is no edge between vertex i and vertex j, then a_{ij} is assigned a zero.

The corresponding adjacency matrix for the graph in Figure 1 can be represented as follows:

$$\text{Adjacency Matrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The **indegree** of a node in graph theory refers to the count of nodes that have a directed edge pointing towards that particular node.

The **outdegree** of a node in graph theory, on the other hand, refers to the count of nodes that have a directed edge originating from that specific node.

IV. Usage of Linear Algebra in PageRank Algorithm

A. Markov Chains

A Markov chain can be used to model the behavior of a random web surfer navigating through web pages. Imagine a random surfer surfing the Web, going from one page to another page by randomly choosing an outgoing link from one page to go to the next one. The PageRank algorithm models web surfing as a random walk on a Markov chain. Each web page corresponds to a state in the chain, and each hyperlink corresponds to a transition between states with some probability.

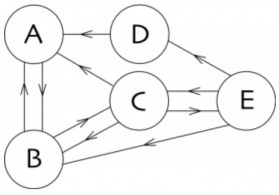

Fig. 2. A Markov Chain depicting the behavior of a random surfer

To represent the behavior of the random surfer using a Markov chain, we construct a transition probability matrix. This matrix captures the probabilities of transitioning from one web page to another. The rows of the transition matrix represent the probabilities of transitioning from a specific web page to all other web pages, while the columns represent the probabilities of transitioning from all web pages to a specific web page. The elements within each row or column capture the likelihoods of these transitions, allowing us to model and analyze the behavior of the random surfer within the web graph.

$$P = \begin{array}{c} \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} & 1 & 0 \\ 1 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & 0 & 0 \end{pmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix}$$

Fig. 3. A stochastic transition matrix depicting the behaviour of a random surfer

The PageRank algorithm aims to assign a numerical weight, known as the PageRank value, to each web page in a way that reflects its importance or relevance within the web graph. The PageRank value is calculated using the equation:

$$v = (1 - d)e + dMv$$

where:

- v is a vector representing the PageRank values of all pages.
- e is a vector of all ones, representing equal probabilities initially.
- d is the damping factor, typically set to 0.85
- M is the transition probability matrix of the Markov chain

To find the PageRank values, we need to solve this equation for v. Mathematically, this is equivalent to finding the eigenvector of M corresponding to the eigenvalue 1. The eigenvector corresponding to the eigenvalue 1 provides the steady-state probabilities of the Markov chain, which are the PageRank values.

The eigenvector of a transition matrix M can be found using the **Power Method**. The algorithm starts with an initial vector and repeatedly applies the matrix M to the vector until it converges to the dominant eigenvector. The resulting vector represents the PageRank values of all pages.

In summary, the PageRank algorithm uses Markov chains to model the behavior of a random web surfer navigating through web pages. The transition probabilities between pages are captured in a transition probability matrix. The PageRank values are then determined by finding the dominant eigenvector of this matrix using the Power Method. This process assigns numerical weights to web pages, reflecting their importance in search engine rankings.

B. Graph Theory

The Internet is formed by the WWW which stands for the "World Wide Web" which literally means a web spanning the world as the WWW contains billions of webpages each linked to several other that it forms a spider-like Web. Hence to accurate depict and work with the internet on a mathematical

level, it is best represented using Graphs and Adjacency Matrices.

The web graph serves as the foundation for the PageRank algorithm. It represents the interconnectedness of web pages, with each page being a node in the graph and hyperlinks serving as directed edges. This directed nature of the edges signifies the one-way nature of web links, capturing the notion that a page can link to another without a reciprocal link.

The Graph of the Internet is based on the simple conventions such as

- **webpage** is represented as a node in the graph.
- **directed edge** is show a hyper-link from source node to the destination node

The web graph is typically represented as a directed graph because web page links have a specific direction. A directed edge from node A to node B indicates that there is a hyperlink from page A to page B.

The PageRank algorithm uses the concepts of graph theory to rank pages by their popularity.

To efficiently analyze the web graph, the PageRank algorithm employs an adjacency matrix. This matrix provides a compact representation of the graph, where each row and column correspond to a web page. The elements of the matrix indicate the presence or absence of links between pages, enabling a systematic examination of the connectivity of the web graph.

To emulate user behavior on the web, the PageRank algorithm adopts the random surfer model. This model envisions a user randomly clicking on hyperlinks on a particular webpage which takes it to a adjacent node on the graph and so on navigating through the web graph. By following the directed edges, the random surfer moves from one page to another based on the available links, simulating the browsing patterns of users.

V. Analysis of the PageRank Algorithm

At the crux of the Google's PageRank Algorithm lie the concepts discussed earlier in the paper. The web of data or the internet is represented efficiently when concepts of graph are applied and to operate on them mathematically, the concept of matrices, eigen-values and eigen-vectors come in handy.

More often than not, the PageRank algorithm makes use of the Stochastic matrices, and the confluence of the Stochastic Matrix and Markov processes is what primarily helps in gaining an insight into the shape that the web will take after a long time, and thereby helps in effectively ranking the pages.

The analysis of the PageRank algorithm takes these concepts and furthers them with by adding more qualitative analysis.

The driving mathematical equation behind the algorithm is as follows:

$$PR(u) = \frac{1-d}{N} + \sum_{v \in M_u} \frac{PR(v)}{L(v)}$$

Where M_u represents the set of all nodes **v** that redirect the user to the page **u**, and (1-d) is the **damping factor**. We will return to a simplified version of this equation later in the section but for now let us assume that something like this exists and instead try to delve deep into how this came up to take the form it has by analysing the algorithm and its functioning.

As discussed earlier, the reason PageRank stands out as one of the most effective and influential ranking algorithms is due to the fact that it takes into consideration two kinds of scores for each page:

- Traditional content scores, and
- Popularity scores

What these terms stand for is quite intuitive and the intuition behind combining them to rank a page was discussed earlier in the problem statement.

However, to begin the analysis of the PageRank algorithm, a deeper understanding of what **popularity** means in reference to web-pages is needed.

To put in simplest words, popularity of a page represents its reputation and authority in the Web, i.e., it demonstrates how many other pages lead the user to this page more often than the other pages.

A. Example

For instance, let there be a 4 node (web-pages) network as shown in Fig. 4:

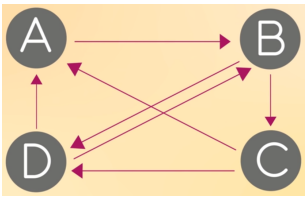

Fig. 4. Example Network with 4 nodes (websites)

From the assumed network as depicted in Fig. 4, the following points demonstrate the most basic analysis which indeed is quite intuitive: (Note that further in the section we may refer to the node just by its name)

- A is referred to (or endorsed) by C and D. Which implies that given the a user lands on A it is equally likely that it has been redirected from either D or C. Further the user is absolutely likely to go to B next, from A.
- B is endorsed by D and A. Similar to A, a user is equally likely to come from A or D given he/she is on B. And from B, there is again an 50% chance that user next goes to either C or D.
- Next, if the user is on C, it is determined that he/she has been redirected from B, or in other words, the only possibility for user to be on C currently is when it was previously on B. And from C, there's an equal chance of user for visiting either A or D.
- Lastly, D is being referred to by sites B and C and out-links from D lead to A and B with equal probability.

Mathematically, the data can be represented using the following matrix P:

$$\begin{array}{cccccc} . & A & B & C & D \\ A & 0 & 0 & 1/2 & 1/2 \\ B & 1 & 0 & 0 & 1/2 \\ C & 0 & 1/2 & 0 & 0 \\ D & 0 & 1/2 & 1/2 & 0 \end{array}$$

The above matrix is known as the Transition Matrix which has been elaborately explored in one of the previous sections of the paper.

Analysing the matrix further reveals a little more about the nature of our assumed network:

- The rows containing non-zero values in the column corresponding to a particular node represent the nodes **to which** the column node contains a hyperlink. Therefore, the sum of the all the elements of a column will be 1. (Note that in PageRank Algorithm, we consider the probability distribution of choosing a particular out-link from the current website to be uniform).
- The columns containing non-zero values in the row corresponding to a particular node represent the nodes **from which** the row node contains in-links i.e,. those nodes which endorse that particular row node.

Generalising the above statement for any $n \times n$ transition matrix, where n is the number of nodes in the network, we have the following definition associated with it:

$\forall i \in \{1, 2, ..., n\}$
if $P_{ij} > 0$, implies that there is a link from j^{th} node to i^{th} node
and similarly,
$\forall j \in \{1, 2, ..., n\}$
if $P_{ij} > 0$, implies that there is a link to i^{th} node from j^{th} node

Further analysis into our assumed system reveals something worth taking note of, and this newfound insight is what forms one of the pillars of the PageRank Algorithm - the Popularity measure of each page.

Note that the page A has only one out-link and that redirects the user to page B. Which implies that B is certainly highly preferred over any other site. This also points at the possibility of B containing more useful information content that adds to the **authority** or more correctly, the **reputation** of B. In addition to this it also pointed to by another page D. Overall, B has two hyperlinks that redirect the user to it, but so is the case with D and A. But what makes B **more reputed** is the fact that it is **the only** outgoing link to one of the nodes that point to it.

Next, let us analyse C. In case of C, we have just one link pointing to it - from B. Now this intuitively gives an idea that C is the least preferred site. To support this, we have two reasons:

- It is pointed to by only one node - B
- it redirects the user to two pages with an equally likely probability

This emphasizes on the fact the C has less useful information content and hence has a **lesser reputation** as compared to others.

Now, analysing D and A simultaneously, we find that D is pointed to by B and C, and it further points to B and A, and A is being pointed to by D and C and it points to just B. Here, D and A both have to in-links and moreover, they share a common previous node - C. But what makes the difference is the fact other than C, B points to D and D points to A. Since, B has **highest reputation**, it is more likely that its out-links are more reliable. Therefore, a user has more chances to end up being on D instead of being on A, and D has **higher reputation** that A.

Clearly and intuitively, according to the analysis presented, the order must be as follows:

$$B > D > A > C$$

,
and interestingly enough, this is how the rankings turn out to be even mathematically, when we take the initial state vector X to be a column matrix with the initial probabilities of starting on any of the four nodes is equally likely. The following calculation shows it:

$$X_0 = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix},$$

Let X be the final steady state vector, then

$$P \times X_0 = X,$$

$$\begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 \end{bmatrix} \times \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} = X$$

Thus, iteratively calculating X, we find that

$$\begin{bmatrix} 0.217 \\ 0.348 \\ 0.174 \\ 0.261 \end{bmatrix}$$

Now that we have the final steady state matrix X, our intuition has turned to be indeed true, that is, B has the highest chance of being visited, followed by D, then A and finally C.

Above stated example demonstrates a very basic logic behind the working of the algorithm.

B. The structure of the network

One of the most crucial aspects of the algorithm is the scenario in which it is put into use. The method discussed above does not always work because of the following conditions on the hyper-link network:

- It needs to be a-periodic
- It must converge
- It must not have any dangling nodes

Let's take the above points one by one.

First looking at the condition of a-periodicity, it can be derived that the structure of the web must not allow a user to toggle between just a few of the many sites which may or may not be useful. This can occur when a user is periodically redirected to a particular node, which leads to another predefined set of nodes, and finally renders the process deterministic, which eventually ends again at that node.

This problem can be better demonstrated by using a two node system where both nodes point at each other (Fig. 5).

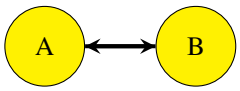

Fig. 5. Two node system where both point at each other

In Fig. 5, we observe that suppose the user starts at node A, he is 100% likely to return back to A, and then again to B, and then A, and so on. This forms a cycle which is periodic in nature.

The problem with such networks is that they render an oscillatory motion of the user across the web thereby not converging at a steady state. And since achievability of a steady state (in other words - convergence) is what characterises a Markov chain, the system of nodes in discussion defies this notion by inducing oscillation. The heavy dependence of the algorithm on the Markov chains makes this a loop-hole to exploit.

Another problem that arises is of the case of the dangling node. A dangling node is defined as the node which has no out-links. In other words, it is a node that if encountered does not redirect a user to any other page. Consider Fig. 6 as pictorial representation of a four-node network with a dangling node:

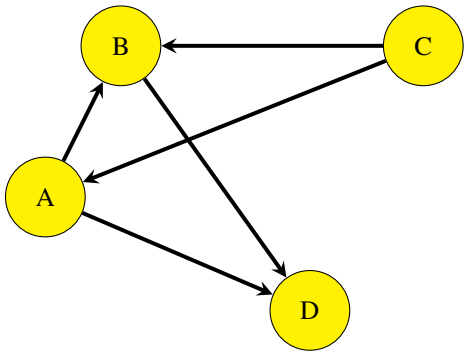

Fig. 6. The four-node network where D is the dangling node

Let's observe what the transition matrix of such a system would look like:

$$P = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \end{bmatrix}$$

We see a fascinating situation here. The column corresponding to D is a zero-column which implies that regardless of whatever initial state vector X_0 we start with, we will end up with a steady state vector that looks as follows:

$$X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

This means that all of the user-population will eventually end up at node D, and this basically allows the node D to gain absolute authority or dominate over the entire web. Such a configuration leads the algorithm to converge at 1 and destroys the fundamental essence of PageRank.

C. Fixing the loopholes

To make an algorithm work at such a large scale, there definitely has to be a workaround for these issues. Since the structure of the web is not in the hands of the algorithm, it is the algorithm that has to go through alterations so as to overcome the difficulties.

1) The Random Surfer Model and Damping Factor: Google, then came up with a model of PageRank less susceptible to such tweaks in the structure by adopting a random surfer model. What the model essentially means is that the algorithm now takes into account the fact that after some point the user is psychologically bound to visit some other random site.

- Suppose there exists a cyclic nature in the system, then after a point when the user realises that a periodic or oscillatory motion is occurring, he or she is bound to visit some other random site which is not preferred by any of the nodes that create the cycle.
- Also, suppose there is a dangling node in the system, then as we saw in the previous section, the surfer will end up at the dangling web-page thereby rendering a worthless experience. Here again the user is bound to explore some other random website in order to search for and gain something meaningful out of the surfing experience.

Evidently, the random surfer model lends a better alternative to the previously encountered problems.

Now, to explore and implement it mathematically, we introduce something called the **damping factor**. Let's denote is as **d/N**, where N is the number of nodes.

This damping factor d/N accounts for the randomness in the surfing activity of the user. But this creates a new problem, that the sum of all the probabilities of the out-links at a node do not add up to one.

This issue is resolved by multiplying the transition matrix by a factor of **1-d**.

Therefore the new transition matrix P_{new} can be shown as:

$$P_{new} = (1-d)P + \tfrac{d}{N} I_{N \times N}$$

These changes now assure that a-periodicity is maintained while taking care of the dangling nodes which initially had a zero column associated to them.

VI. Visualization of PageRank

The visualization of PageRank is an effective way to understand the importance and ranking of nodes in a graph. The

provided code utilizes the NetworkX library and matplotlib to calculate and visualize the PageRank scores of nodes in a directed graph.

The PageRank algorithm assigns a score to each node based on its importance within the graph. The algorithm iteratively calculates the scores, considering both the inbound links and the damping factor, which determines the likelihood of a random jump from one node to another. The algorithm continues until the scores converge or the maximum number of iterations is reached.

Let's consider the following test case matrix and graph:

.	A	B	C	D
A	0	1	1	1
B	0	0	1	1
C	1	0	0	0
D	1	0	1	0

The matrix represents the connections between nodes, where a value of 1 indicates a connection from the row node to the column node.

The graph visualization of this test case is shown in Figure 7.

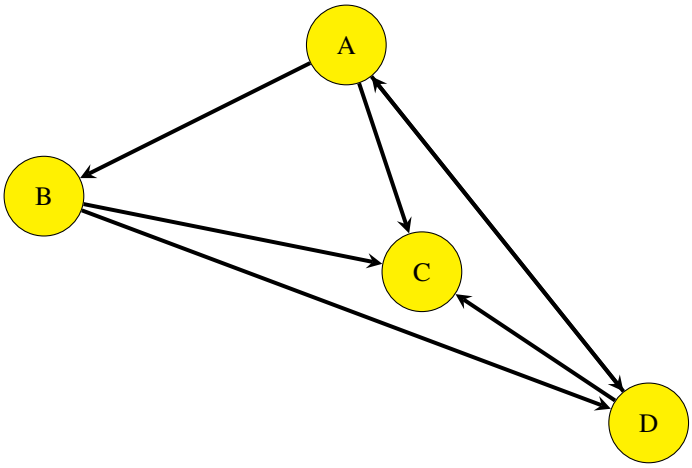

Fig. 7. Graph representation of the sample case

The PageRank algorithm is then applied to this graph, resulting in the following rankings and scores:

- Rank 1: A (PageRank: 0.3681506538366117)
- Rank 2: C (PageRank: 0.2879616007311895)
- Rank 3: D (PageRank: 0.2020783388856028)
- Rank 4: B (PageRank: 0.1418094065465959)

The graph is visualized again, this time with node sizes proportional to the PageRank scores and labels indicating the ranks and scores. The updated visualization is shown in Figure 8.

Additionally, the changes in Page-Rank scores with each iteration are shown in Figure 9. The x-axis represents the iterations, and the y-axis represents the PageRank scores. Each line corresponds to a node, with labels indicating the node names.

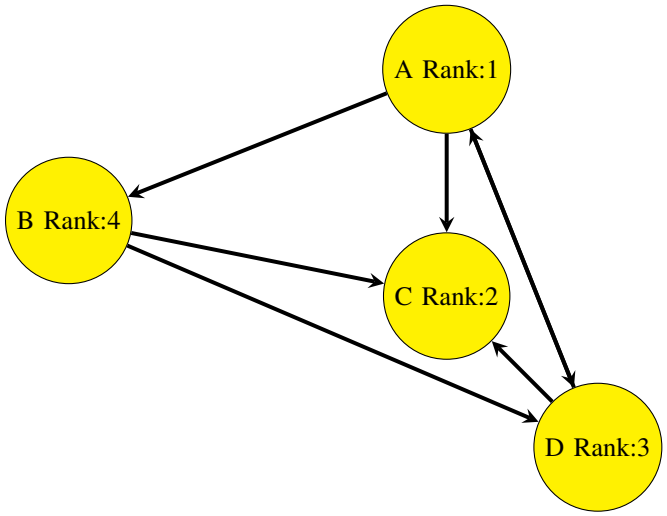

Fig. 8. Graph visualization with Page-Rank rankings

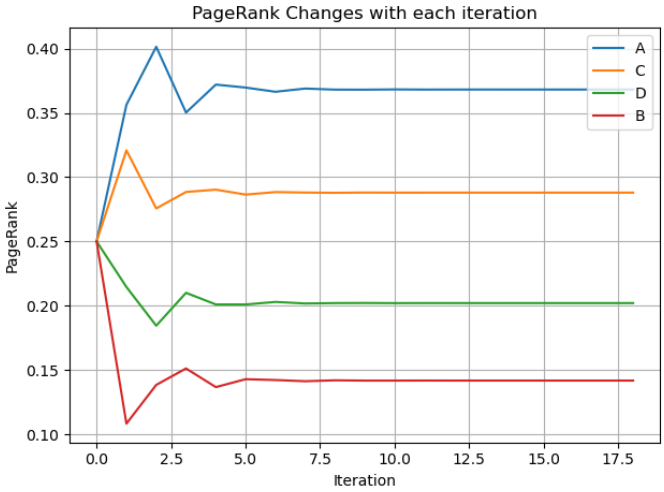

Fig. 9. Changes in PageRank scores with each iteration

https://www.overleaf.com/project/648433838b827cbb2ce9fade

This visualization provides insights into the ranking and importance of nodes in the given graph using the PageRank algorithm.

The python code for calculating the pagerank and generating the above visualisations can be found here.

VII. Related Works - State of the Art Literature

1) **Anatomy of Hypertextual Web Search Engine** by Lawrence Page and Sergey Brin (1998) - The first paper is a technical paper that describes the anatomy of a large-scale hypertextual web search engine, namely Google. The paper covers various aspects of web search engines, including the challenges involved in creating a large-scale search engine, the use of hypertext structure to improve search results, and the PageRank algorithm.The paper explains that Google makes heavy use of hyper-textual information consisting of link structure and link

text to evaluate the quality of web pages and return relevant search results. It also introduces the concept of PageRank, which is a measure of a page's citation importance that corresponds well with people's subjective idea of importance. The paper describes the calculation of PageRank, which takes into account the quality and quantity of links, and introduces the parameter "d," which is a damping factor that controls the influence of links.The paper also discusses the advancements in technology and web proliferation that have affected the creation of web search engines in recent years. It highlights the challenges of accessing usage data due to its commercial value and the need to build systems that can support novel research activities on large-scale web data.

2) **PageRank: Standing on the Shoulder of Giants** by Massimo Franceschet (June 2011) - This paper explored the PageRank in a more deeper way and explored its theoretical and mathematical aspects. This paper also appreciated the use of this algorithm in Bibliometrics, Sociometry and Econometrics
3) **Mathematics behind Google's PageRank Algorithm** by Brian Moor (August 2018) - This paper uncovers the mathematics and the history behind the working of the algorithm and states reasons as to why the algorithm works in the first place. The paper also covers how page-ranking was different from the keyword-searching search engines of the time.These limitations included the ease of keyword abuse, where website owners could manipulate their rankings by stuffing their pages with popular keywords, even if the content was not relevant. Additionally, keyword-based search engines often struggled to accurately determine the relevance and importance of web pages based solely on keyword matching.PageRank overcame these limitations by introducing a new approach to ranking web pages. Instead of relying solely on keywords, PageRank considered the number and quality of links pointing to a page as an indicator of its importance and relevance. This approach made it much more difficult for website owners to manipulate their rankings by keyword stuffing. By analyzing the link structure of the web, PageRank provided more accurate and reliable search results, ensuring that the most important and useful pages appeared higher in the search results.
4) **Linear Algebra Application: Google PageRank Algorithm** by Jonathan Machado - This paper focuses more on the use of Linear Algebra in PageRank Algorithm and presents an analysis of the Algorithm using those mathematical foundations. In the context of Google's PageRank algorithm, linear algebra and graph theory play crucial roles in calculating web page rankings. Graph theory is used to represent the network of hyperlinks between web pages as directed graphs. Each web page is represented as a node, and the hyperlinks between pages are represented as directed edges. This graph structure allows us to analyze the connectivity and relationships between web pages. Linear algebra comes into play when calculating the PageRank scores for each web page. The PageRank algorithm treats the web page network as a Markov chain, where the probability of transitioning from one web page to another depends on the current page's outgoing links. By representing the web page network as a matrix, known as the transition matrix, linear algebra techniques such as eigenvector calculations can be used to determine the steady-state probabilities or PageRank scores for each web page.
In summary, graph theory helps us model the web page network, while linear algebra provides the mathematical tools to compute the PageRank scores, which ultimately determine the ranking of web pages in search results
5) **Beyond PageRank: Machine Learning for Static Ranking** by Mathew Richardson, Amit Prakash and Eric Bill - This paper aims to appreciate the usefulness of Google's PageRank algorithm and proposes machine learning concepts that can be applied to the alogorithm to increase the efficiency of the algorithm in terms of processing of data and also renders an even better experience to the users
6) **HITS** - One of the most notable papers published alongside PageRank that shares many similarities with the algorithm. This algorithm is used by the Teoma search engine. The HITS technique classifies the web into **hubs** and **authority pages**, wherein the hubs denote the nodes (or pages) that point to many authority pages (the quality and popular ones), and the authority pages are the ones pointed to by the hubs
7) **Google's PageRank and Beyond: The Science of Search Engine Rankings** by Amy N. Langville and Carl D. Meyer - This is a book that aims to explore and thereby appreciate the one of the most revered applications of Linear Algebra in modern world - Google's PageRank Algorithm. It also presents an extremely comprehensive study of PageRank and the HITS algorithm by diving into its history and proposing possible future of these techniques
8) **The BadRank Thesis** - This is a concept that works in the opposite way as the PageRank. In that, the proposed technique aims to rank the bad websites (the ones that don't provide useful quality content or the ones that are least popular). It states that one of the criteria for a site to be bad is if it redirects the user to other bad sites. This implies that the BadRank technique propagates via *outlinks*, whereas PageRank works via *in-link* propagation.

VIII. Alternate Uses of PageRank

The PageRank algorithm, developed by Larry Page and Sergey Brin, is a key component of Google's search engine. While the algorithm is primarily associated with ranking web pages for search results, it has found applications beyond traditional web search. Here are some of the latest uses of the PageRank algorithm:

1) **Social Network Analysis:** The principles of PageRank have been applied to analyze social networks. By treating individuals as nodes and connections between them as edges, PageRank can identify influential users or important communities within a social network.
2) **Protein Networks:** PageRank has been applied to analyze protein-protein interaction networks. By considering the connectivity and importance of proteins within the network, PageRank can identify key proteins that play crucial roles in biological processes and disease pathways.
3) **Food Webs:** PageRank has been used to determine the relative importance of different species in a food web. By analyzing the interactions and dependencies among species in an ecosystem, PageRank can identify the most influential species that have a significant impact on the overall stability and functioning of the food web.
4) **Recommendation Systems:** PageRank has been adapted to create personalized recommendation systems. By considering users' interactions and preferences as links, the algorithm can identify relevant items or content to recommend based on their popularity and connectivity.
5) **Citation Analysis:** PageRank has been employed to assess the impact and influence of scholarly papers. By treating citations as links, the algorithm can identify important and influential research papers within a specific field. By analyzing the citation network among authors and their publications, PageRank can identify influential authors and their contributions to the scientific community
6) **Fraud Detection:** PageRank has been utilized for fraud detection in various domains, such as e-commerce and social media. By analyzing patterns of connections and interactions, the algorithm can identify suspicious or fraudulent entities within a network.
7) **Collaborative Filtering:** PageRank has been employed in collaborative filtering systems, which recommend items based on the preferences of similar users. By considering users' interactions and item connections, the algorithm can identify relevant and popular items to recommend.
8) **Graph Analysis:** PageRank is a fundamental tool in graph analysis. It can be used to evaluate the importance of nodes in a graph and identify central or influential entities within a network.

These are just a few examples of the latest uses of the PageRank algorithm. As a powerful and versatile algorithm, it continues to find applications in various fields involving network analysis, recommendation systems, and data mining.

IX. Future of the PageRank Algorithm

The *future* of the PageRank algorithm is indeed a *past* when we consider its application in Google. Now with the advent of various other Artificial Intelligence techniques, the old-school method of ranking pages *just* by the virtue of their in-links and out-links is not preferred anymore. Bringing such A.I. techniques into play has been a work across decades of many a great computer scientists and some have succeeded in doing so (further discussed in Related Works). Google now makes use of a technique called the **HummingBird Algorithm** which is out of the scope of this paper.

However, the use of PageRank has not died in other fields as already mentioned in the previous section where in some of those uses, it does seem to render a quite fruitful and promising future.

X. Conclusion

PageRank has been by far one of the best inventions in the history of computing that has truly shaped and organised the previously tedious structure of the WWW, and in doing so it has given birth to a stream of information that is available to the user at the blink of an eye. If not more, the PageRank algorithm has set the stepping stones and foundation of the search algorithm which is widely used in practice today, and with the variety of applications that it has found itself in is itself a testimony to the strength and meticulous nature of the algorithm that was developed in 1998 by Lawrence Page and Sergey Brin.

Acknowledgment

We thank the Dr. Chittaranjan Hens for providing us with an opportunity to explore and discover the math, especially Linear Algebra behind one of the most revered and fundamental algorithms in the computing world.

References

[1] Anatomy of Hypertextual Web Search Engine by Lawrence Page and Sergey Brin (1998)
[2] PageRank: Standing on the Shoulder of Giants by Massimo Franceschet (June 2011)
[3] Mathematics behind Google's PageRank Algorithm by Brian Moor (August 2018)
[4] Linear Algebra Application: Google PageRank Algorithm by Jonathan Machado
[5] Beyond PageRank: Machine Learning for Static Ranking by Mathew Richardson, Amit Prakash and Eric Bill
[6] Google's PageRank and Beyond: The Science of Search Engine Rankings by Amy N. Langville and Carl D. Meyer
[7] https://www.youtube.com/watch?v=JGQe4kiPnrUt=808s (Channel - Reducibles)
[8] https://medium.com/@arpanspeaks/handling-dangling-nodes-pagerank-14c31d5b6b62 (Handling Dangling Nodes : Medium.com)