Matrix Theory of Google's PageRank

Edward E. Daisey

27th of April, 2025

1. Introduction

The Internet has grown tremendously in popularity over the past few decades. According to the United Nations, 37% of the human population have not used the Internet.[1] This is roughly 3 billion individuals. As the technology sector expands into emerging markets, Internet adoption will continue to rise, bringing more users to search engines like Google. This paper provides a summary of PageRank[2–5] – Google's algorithm for ranking pages – as well as an example to help the reader understand the associated matrix theory. Matrix theory concepts – such as norms, eigenvectors, and eigenvalues – may be found in this paper. The paper will end with a theorem and a proof.

2. PageRank Summary & Example

Ranking a big and constantly changing web – a collection of web pages (or, more simply, pages) that one can visit through the Internet – is a very tough challenge for researchers to solve. Researchers needed a solution that is mathematically sound. They also needed a solution that works at a massive scale. PageRank achieved both of these. This was done by representing the underlying process with probability and matrix theory. Different pages vote for one another through the usage of hyperlinks (or, more simply, links). The flow of these votes eventually will settle into a stable ranking. The ranking of pages is contingent upon how many pages have a link to a given page. The ranking also depends on how important the linking pages are. PageRank effectively and efficiently captures this by turning the web into a series of connected probabilistic events. Matrices are great at representing these type of probability driven events. In this section, one will develop an understanding of how to build the PageRank algorithm step by step – defining each matrix – and illustrating each concept with a tiny three-page web.

2.1. The Hyperlink Matrix H.

A key concept associated with PageRank is that a link from page P_j to page P_i may be thought of as a vote. Links are represented with arrows in the diagram on the following page. That is by creating a link from my page to your page, I am providing you a vote of support. Under this model if page P_j has d_j links to other pages, then each link carries a weight of $1/d_j$ – which effectively spreads P_j's importance in equal proportion to the pages that it links to. A page with many links connected to other pages distributes its importance among many other pages. This can be represented with an $n \times n$ hyperlink matrix H, where the entries in H_{ij} represent the proportion of P_j's importance that gets passed to P_i:

$$H_{ij} = \begin{cases} \frac{1}{d_j}, & \text{if page } P_j \text{ links to page } P_i, \\ 0, & \text{otherwise.} \end{cases}$$

Here it is assumed that the page must have at least one link to another page (this assumption will be explored later on in the paper), and the columns will sum to one. In terms of probability, each column j in H is indicative of a probability of transitioning to the next page when starting at page P_j. Each column of H may be thought of as the probability of moving from one web page to another web page. Together they form a set of web pages that are connected to each other.

Example. Suppose P, Q, and R represent three different pages comprising a web on the Internet. They are linked to each other as follows:

$$\mathrm{P} \to \{\mathrm{Q}, \mathrm{R}\}, \mathrm{Q} \to \{\mathrm{P}, \mathrm{R}\}, \mathrm{R} \to \{\mathrm{R}, \mathrm{P}, \mathrm{Q}\}.$$

Furthermore, one can calculate the values of the weights by setting $d_P = d_Q = 2$, $d_R = 3$, and construct a hyperlink matrix:

$$H = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \end{pmatrix},$$

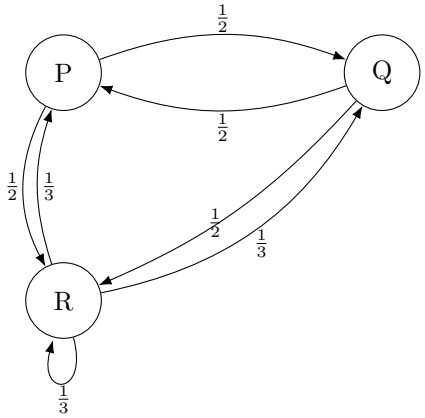

Figure 1. Graphical representation of the hyperlink matrix H and the three-page web that was created.

2.2. Fixing Dangling Nodes: $\mathbf{S = H + A}$.

As mentioned in the previous section, it was assumed that the pages must have at least one link to another page. As one can easily imagine a web page with no links to other pages, it is not always the case that this assumption is true. These web pages are referred to as dangling nodes in the search engine literature. In H, these pages are represented by a column of zeroes. In the context of PageRank, this would result in an Internet user getting stuck on a page that is a dangling node, which would impede the algorithm. To account for when dangling nodes pop up, H is modified by introducing a correction matrix A that it can be summed with. A is a matrix of zeroes everywhere – with the exception of the columns that correspond to dangling node pages. These dangling node columns are filled with $1/n$, where n is the total number of pages in a web. This represents the users heading to another web page with equal probability. The final matrix S results when matrices H and A are summed together:

$$S = H + A.$$

The introduction of A allows PageRank to continue functioning properly even in the presence of a dangling node.

2.3. Teleportation and the Google Matrix G.

Even with the adaptation to account for dangling node pages, there still exists the risk that the Internet user may become trapped in a certain subgraph (which may be thought of as a set of additional nodes not connected to the aforementioned example). This type of scenario would cause the PageRank algorithm fail at producing a meaningful ranking. To account for this instance, PageRank introduces a teleportation step. Let $\alpha \in (0, 1)$ be representative of following a link leaving the page at any given moment of time. 1 - α is associated with the probability of ignoring a link to another page and successfully teleporting to a random page, where each page has the equal probability $(1/n)$ of being teleported to. This teleportation event ensures that the Internet user never gets stuck in a particular subgraph and will eventually explore the entire web over time. This leads to what is known as the Google matrix or G matrix:

$$G \;=\; \alpha\, S \;+\; (1-\alpha)\,\frac{1}{n}\,\mathbf{1}\,\mathbf{1}^{T},$$

where $\mathbf{1}\mathbf{1}^{\mathrm{T}}$ is an $n \times n$ matrix composed of all ones. The G matrix leverages this structure and teleportation in particular in order to ensure that every page in a web has a probability that is non-zero.

Example Continued. For our three-page web there are no dangling node pages, so $A = 0$ and $S = H$. With $n = 3$ pages and teleportation factor $\alpha = 0.85$ (based on early web data research)[2] one now has $(1-\alpha)/n = 0.15/3 = 0.05$. Hence the Google matrix is:

$$\mathbf{G} = \alpha\,\mathbf{S} + (1-\alpha)\,\frac{1}{n}\,\mathbf{1}\mathbf{1}^{\mathsf{T}} = 0.85\,\mathbf{H} + 0.05\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix} = \begin{pmatrix}0.050 & 0.475 & 0.333\\ 0.475 & 0.050 & 0.333\\ 0.475 & 0.475 & 0.333\end{pmatrix}.$$

2.4. The Power Method.

The vector I is a probability that remains constant after each time a user clicks, and it satisfies:

$$G\,I = I.$$

I is the eigenvector of G with an eigenvalue of one. Starting with a uniform guess $I^{(0)} = (1/n, \ldots, 1/n)^T$, one iterates as such:

$$I^{(k+1)} = G\,I^{(k)},$$

track the user's clicks over time and stop iterating when:

$$\left\|I^{(k+1)} - I^{(k)}\right\|_1 < 10^{-6},$$

or when the total change in probability from one step to the next falls below one in a million. 10^{-6} is a common value noted in technical documentation of important languages and web services.[6,7] Iteration is used because each application of G models another click by the user. This captures the evolution of user's visit probabilities over time until convergence to the final PageRank vector I is achieved.

Example Continued. We start with the uniform guess:

$$I^{(0)} = \begin{pmatrix}\frac{1}{3}\\ \frac{1}{3}\\ \frac{1}{3}\end{pmatrix}.$$

Then each iteration multiplies by G:

$$I^{(1)} = G\,I^{(0)} \approx \begin{pmatrix}0.286\\ 0.286\\ 0.427\end{pmatrix}, \qquad \left\|I^{(1)} - I^{(0)}\right\|_1 \approx 0.189,$$

$$I^{(2)} = G\,I^{(1)} \approx \begin{pmatrix}0.292\\ 0.292\\ 0.414\end{pmatrix}, \qquad \left\|I^{(2)} - I^{(1)}\right\|_1 \approx 0.027,$$

$$I^{(3)} = G\,I^{(2)} \approx \begin{pmatrix}0.291\\ 0.291\\ 0.416\end{pmatrix}, \qquad \left\|I^{(3)} - I^{(2)}\right\|_1 \approx 0.004.$$

This continues until $\|I^{(k+1)} - I^{(k)}\|_1 < 10^{-6}$ produces the final PageRank scores. Suppose the values end up as:

$$I = \begin{pmatrix}0.291\\ 0.291\\ 0.416\end{pmatrix},$$

where $R > P = Q$. This final matrix captures the importance of each page in our three page web.

2.5. Bringing It All Together.

The PageRank algorithm goes through these steps:

$$H \xrightarrow{\text{Fix Dangling Nodes}} S = H + A \xrightarrow{\text{Add Teleportation}} G = \alpha S + \tfrac{1-\alpha}{n}\mathbf{1}^T \xrightarrow{\text{Power Method}} I.$$

At each stage, key properties – such as the columns summing up to 1 in addition to the matrix values being positive – are preserved. This supports the PageRank algorithm being mathematically sound with respect to its objective. PageRank is remarkable in that it leverages concepts from matrix theory that ensure an elegant, scalable solution to the challenge of ranking many different web pages.

3. Conclusion

PageRank's success to date captures how ideas from matrix theory can be used to rank billions of web pages in a fraction of a second. PageRank produces a ranking that captures the importance of various web pages. This paper explored some foundational material relating to searching the web and serves as a good starting point for more advanced and more nuanced algorithms. By understanding PageRank, one has the foundation to develop more effective and efficient ranking algorithms for large-scale networks. Ultimately, this algorithm highlights the beauty of mathematics in solving real-world problems. A proof associated with one of the exercises in Bryan's paper[3] may be found below.

4. Theorem With A Proof

Statement. *"Prove that in any web the importance score of a page with no backlinks is zero."*[3]

Theorem 1 (No Love For Pages With No Inbound Links)**.** *In any web, the importance score of a page with no backlinks is zero.*

Proof. Let H be the $n \times n$ matrix:

$$H_{ij} = \begin{cases} \frac{1}{d_j}, & \text{if } P_j \to P_i, \\ 0, & \text{otherwise,} \end{cases}$$

and let the importance vector x satisfy:

$$H\,x = x, \quad \sum_{i=1}^{n} x_i = 1.$$

If page k has no inbound links, then:

$$H_{k\,j} = 0 \quad \forall\, j,$$

so

$$x_k = (H\,x)_k = \sum_{j=1}^{n} H_{k\,j}\,x_j = 0.$$

□

Remark. This proof highlights why a web page with no inbound links from another web page would vanish from Google's search engine under the model presented towards the beginning of this paper. This proof demonstrates the need for the teleportation step in the PageRank algorithm. This ensures that every page maintains a positive score.

References

[1] United Nations International Telecommunication Union, *Measuring Digital Development: Facts and Figures 2021*, Geneva, 2021. Available at: ITU Facts and Figures 2021.

[2] Brin, S. and Page, L., *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Computer Networks, 30(1), 107–117, 1998. Available at: The Anatomy of a Large-Scale Hypertextual Web Search Engine.

[3] Bryan, K. and Leise, T., *The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google*, SIAM Review, 48(3), 569–581, 2006. Available at: The $25,000,000,000 Eigenvector.

[4] Haveliwala, T. H. and Kamvar, S. D., *The Second Eigenvalue of the Google Matrix*, Stanford University, 2006. Available at: The Second Eigenvalue of the Google Matrix.

[5] Austin, D., *How Google Finds Your Needle in the Web's Haystack*, Feature Column, American Mathematical Society, December 2006. Available at: How Google Finds Your Needle in the Web's Haystack.

[6] IBM, *rustworkx.pagerank — Rustworkx API Reference*, 2025. Available at: rustworkx.pagerank — Rustworkx API Reference.

[7] Amazon Web Services, *PageRank — Amazon Neptune Analytics User Guide*, 2025. Available at: PageRank — Amazon Neptune Analytics User Guide.