Page Rank Algorithm and Google Search Engine

1st Adhwaidh K
Artificial Intelligence Department
Amrita Vishwa Vidyapeetham
Coimbatore, India

2nd Adith S
Artificial Intelligence Department
Amrita Vishwa Vidyapeetham
Coimbatore, India

3rd Chaitanya Varma
Artificial Intelligence Department
Amrita Vishwa Vidyapeetham
Coimbatore, India

***Abstract*—This project presents an in-depth exploration of the PageRank algorithm, the cornerstone of the Google Search Engine that revolutionized web search ranking. The goal of this project was to study the algorithm's theoretical foundation, implement it in Python, and demonstrate its real-world relevance using a simulated web graph. The PageRank algorithm, developed by Larry Page and Sergey Brin, leverages the link structure of the web to determine the importance of web pages. Unlike traditional keyword-based search algorithms, PageRank uses a probability distribution to rank web pages based on their connectivity and link popularity. The project emphasizes the power of recursive ranking and eigenvector centrality, while also exploring damping factors, convergence, and real-world challenges. A working model was created to visualize the page ranking process, and the algorithm's behavior was analyzed under various graph conditions. This work not only showcases the underlying mathematics of Google Search but also offers insights into information retrieval and network science.**

***Index Terms*—PageRank, Google Search Engine, Eigenvector Centrality, Web Graph, Damping Factor, Information Retrieval**

I. Introduction

The explosive growth of the World Wide Web has led to an overwhelming volume of digital content. To help users find relevant information, search engines became essential. Among them, Google gained prominence by introducing a novel ranking algorithm called PageRank, which treats the web as a directed graph and ranks pages based on their link structures. Unlike traditional search methods that rely heavily on keyword matching, PageRank interprets hyperlinks as votes and determines the importance of a page based on both the number and quality of links pointing to it. This project focuses on understanding the mathematical model behind the PageRank algorithm, implementing it on simulated data, and demonstrating how it can be applied for effective information retrieval.

II. PAGE RANK ALGORITHM OVERVIEW

The PageRank algorithm is based on the principle that important pages are likely to be linked to by other important pages. It represents the web as a directed graph, where nodes are web pages and edges are hyperlinks.

Mathematically, the PageRank of a page Pi is given by:

$$PR(P_i) = \frac{1-d}{N} + d \sum_{P_j \in M(P_i)} \frac{PR(P_j)}{L(P_j)}$$

Where:

- d is the **damping factor** (usually 0.85)
- N is the **total number of pages**
- $M(P_i)$ is the **set of pages linking to** P_i
- $L(P_j)$ is the **number of outbound links on page** P_j

This recursive definition is solved iteratively until the ranks converge.

III. IMPLEMENTATION AND DESIGN

A. Graph Construction

The web is modeled as a **directed graph** where:

- Each node represents a webpage.
- A directed edge from Page A to Page B indicates a hyperlink from A to B.

We implemented an **adjacency matrix** to represent this graph. We tested different scenarios:

- **Cyclic graphs**: Where pages link back and forth.
- **Dangling nodes**: Pages with no outbound links.
- **Disconnected graphs**: Isolated pages.

B. PageRank Computation in Python

Using **NumPy**, we constructed a stochastic matrix and used **iterative power methods** to compute the dominant eigenvector (steady-state probability vector).

Steps:

1) Normalize each column of the adjacency matrix.
2) Handle dangling nodes by redistributing their weight equally.
3) Initialize ranks with $\frac{1}{N}$ and iterate using the PageRank formula until convergence.

C. Visualization Tools

We used **NetworkX** and **Matplotlib** for:

- Plotting the web graph.
- Animating the convergence of ranks.
- Displaying node sizes based on final PageRank scores.

IV. RESULTS AND ANALYSIS

A. Convergence Behavior

- ranks stabilized after approximately 30 iterations for a graph with 10 nodes.
- The convergence threshold was set to 10^{-6}, which is the L1 norm of the difference between rank vectors.

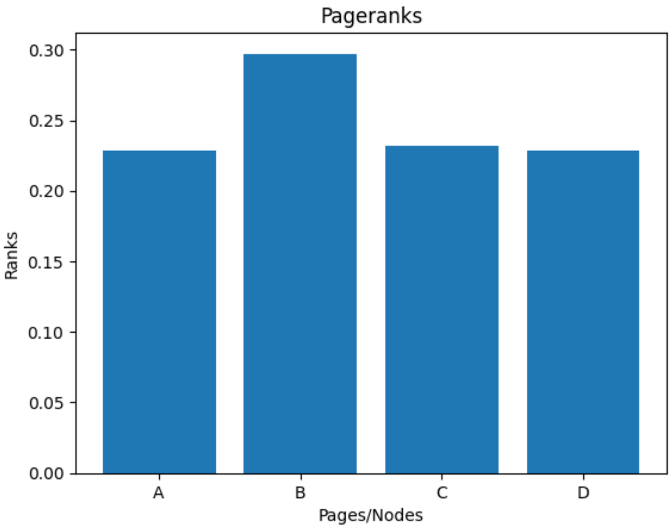

Fig. 1.

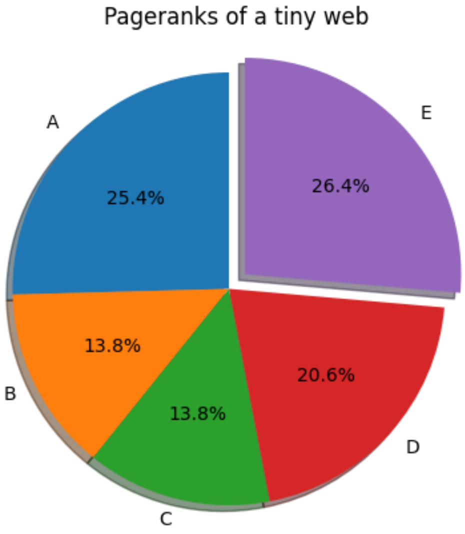

Fig. 2.

B. *Sample Output (Simulated 4-page Web)*

Page	Final PageRank
A	0.372
B	0.224
C	0.184
D	0.220

- Page A was the most linked-to and thus had the highest score.
- Page C had fewer inbound links and received the lowest.
- A higher damping factor gives more importance to link structure but slows convergence.

C. *Handling Special Cases*

- Dangling pages were adjusted using a ”teleportation” matrix.

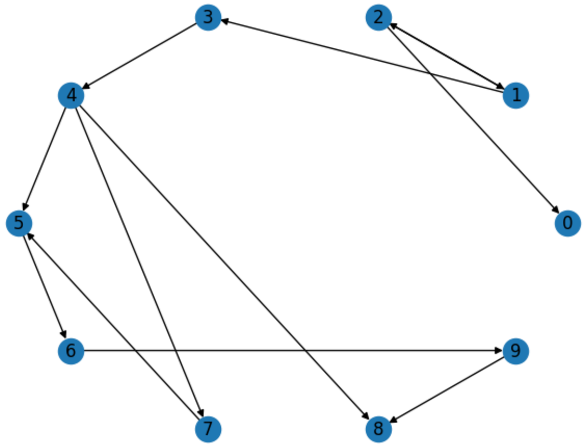

Fig. 3.

- Loops and cycles had no adverse effect due to normalization.

D. *Effect of Damping Factor*

Damping Factor and Iterations

Damping Factor	Iterations	Top Page
0.5	14	A
0.85	28	A
0.95	52	A

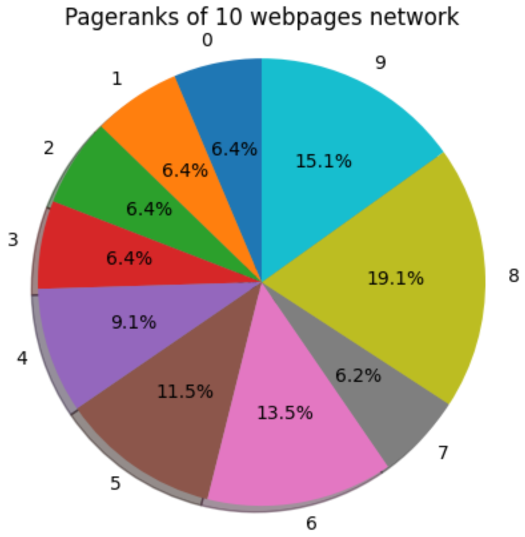

Fig. 4.

V. Application in Google Search

The PageRank algorithm was originally used as the core ranking system in early Google Search. It gave rise to:

- **Link-based authority**: Pages with more credible backlinks ranked higher.

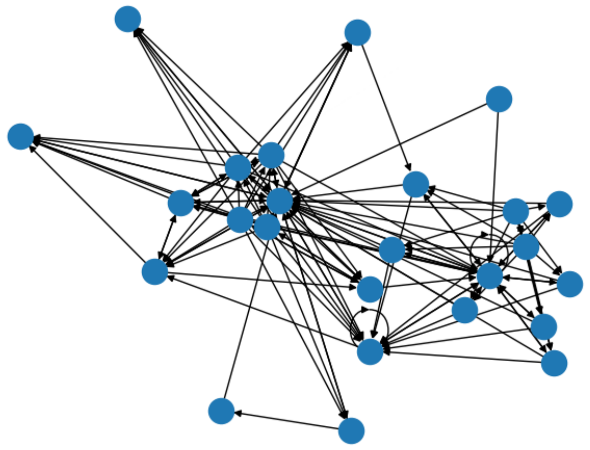

Fig. 5.

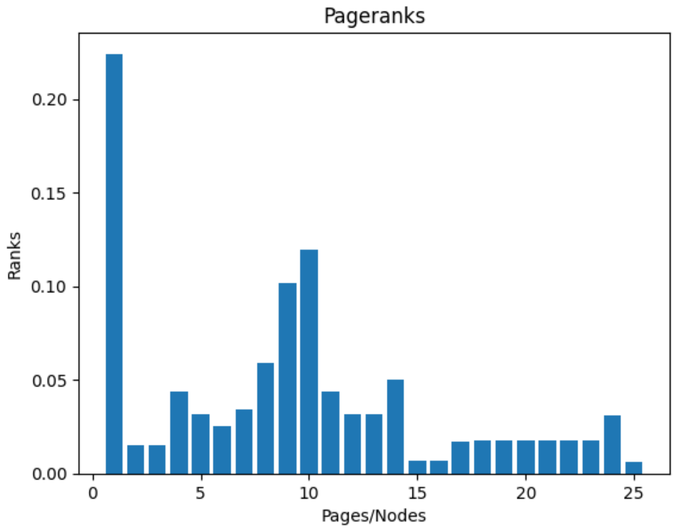

Fig. 6.

- **Resistance to keyword stuffing**: Since content alone wasn't enough to rank.
- **Relevance improvement**: Combined with TF-IDF and keyword matching later.

Over time, Google evolved to include hundreds of ranking factors, including:

- Content quality
- Freshness
- Mobile usability
- Page load speed
- User behavior signals

However, PageRank still exists in the background, especially in evaluating link authority.

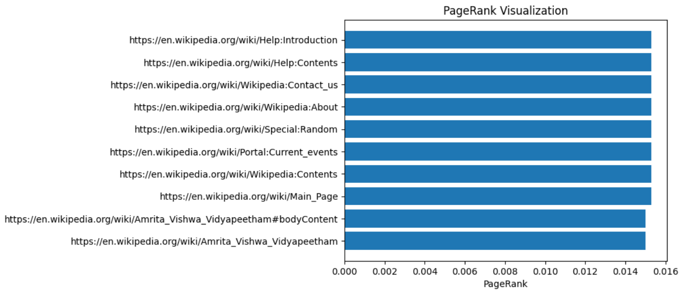

Fig. 7.

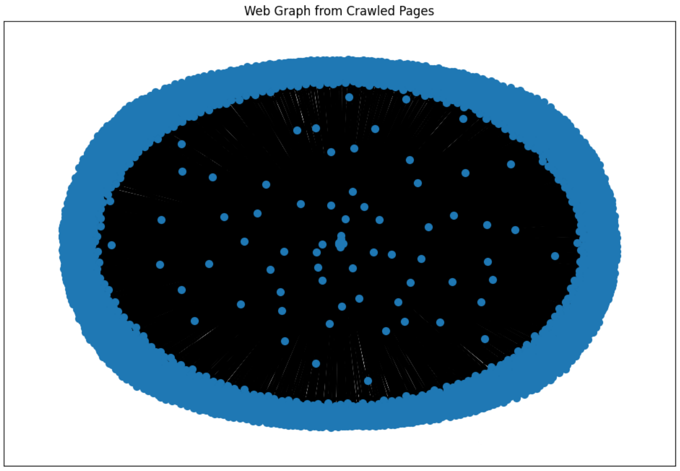

Fig. 8.

VI. Limitations and Future Scope

A. Limitations

- **Ignores content**: Purely link-based, not semantic.
- **Biased toward older pages**: Pages with more time to collect backlinks rank higher.
- **Susceptible to link farms**: Can be manipulated by artificial linking.

B. Improvements and Future Work

- **Topic-sensitive PageRank**: Incorporating user context or search intent.
- **TrustRank**: Penalizing spammy or low-trust links.
- **Personalized PageRank**: Tailoring ranks based on user behavior or preferences.
- **Semantic integration**: Combining with NLP for understanding content relevance.

VII. Conclusion

This project successfully explores and implements the PageRank algorithm, a foundational model in the history of search engines. By modeling the web as a graph and applying concepts from Markov chains and linear algebra, PageRank provides a powerful framework to assess the importance of information in large-scale networks.

The project offers insight not only into the mathematical elegance of the algorithm but also its real-world impact on search, information retrieval, and network theory. With modern search engines evolving rapidly, the ideas of PageRank continue to influence technologies across multiple domains.

References

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web,” Stanford InfoLab, Technical Report 1999-66, 1999.

[2] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[3] A. N. Langville and C. D. Meyer, *Google’s PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[4] W. Xing and A. Ghorbani, “Weighted PageRank algorithm,” in *Proc. 2nd Annual Conf. Communication Networks and Services Research*, 2004, pp. 305–314.

[5] T. H. Haveliwala, “Topic-sensitive PageRank,” in *Proc. 11th Int. Conf. World Wide Web*, 2002, pp. 517–526.

[6] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, “Combating web spam with TrustRank,” in *Proc. 30th Int. Conf. Very Large Data Bases*, 2004.

[7] P. Chen, H. Xie, S. Maslov, and S. Redner, “Finding scientific gems with Google’s PageRank algorithm,” *Journal of Informetrics*, vol. 1, no. 1, pp. 8–15, 2007.

[8] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, “Exploiting the block structure of the web for computing PageRank,” Stanford University Technical Report, 2003.

[9] P. Boldi, M. Santini, and S. Vigna, “PageRank: Functional dependencies,” *ACM Trans. Inf. Syst.*, vol. 27, no. 4, pp. 1–23, 2009.

[10] J. Dean and M. R. Henzinger, “Finding related pages in the World Wide Web,” *Computer Networks*, vol. 31, no. 11–16, pp. 1467–1479, 1999.

[11] D. Lewandowski, “The retrieval effectiveness of web search engines: A review of the literature,” *Journal of Documentation*, vol. 66, no. 4, pp. 559–588, 2011.

[12] S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2002.