Simulation and Analysis of Google PageRank: Convergence Analysis & Graph Topology

Ganesh Somashekhar (gs33799), Arhaan Nisar (an34299)

12/08/2025

1 Introduction

Quantifying Page Authority in the Web

During the 1990s, the rapid growth of the World Wide Web presented a particular challenge in which engines struggled to effectively find, rank, and retrieve information from billions of interconnected documents. Because early search engines relied heavily on keyword matching, it often led to easily manipulated results and poor relevance. Furthermore, other methods such as a simple link count was seen as insufficient since the importance of a link coming from a minor blog and a link from a major news organization This issue soon transitioned into a core problem: How can one objectively quantify the relevant authority or importance of a web page? A reliable method was needed in order to ensure that the most relevant and important sources appeared at the top in search results.

Theoretical Background: Markov Chains

This project addresses the problem of ranking the most relevant and important sources using a theoretical model known as PageRank developed by Larry Page and Sergey Brin. This algorithm incorporates the behavior of a "random surfer" that navigates the web by randomly clicking on links throughout each page. The overall mathematical foundation that structures this algorithm lies within the Markov Chain theory which describes the searching algorithm as transitions between a set of states based on specific probabilities.

$$\pi^{(k+1)} = \pi^{(k)}W \tag{1}$$

1. π is a stationary distribution vector in which each component of π_i is the PageRank of i. Additionally, the sum of all components within that row vector is 1.
2. W is the Transition Matrix of the web graph. This matrix consists of a dimension of $N*N$ where the element of W_{ij} represents the the probability of transitioning or moving from page i to page j.

This equation represents the iterative approach which starts with an arbitrary distribution vector of $\pi^{(0)}$. The subsequent distributions are then calculated by a series of repeated multiplication as shown in $\pi^{(k)}W$. The stationary vector π is an eigenvector corresponding to the eigenvalue of 1 for the transition matrix of the web.[1] The significance of this theory and mathematical equation is that the distribution represents the long-term frequency that the random surfer will be found on any given page after an infinite number of clicks.

Approach: Simulation of the Web

The project establishes a working model of the web using two C++ classes (Page and Web). The Web class incorporates an abstract data type that manages a collection of interconnected Page objects. These connections are randomized, which were structured to simulate a real-world graph structure. The process of clicking links is also simulated through a random walk, which allows one to gather statistics on the frequency of page visitation. Furthermore, the process also includes Breadth-First Search as the main implementation of the graph traversal algorithm to calculate structural properties such as the graph's diameter and the reachability of pages. The ranking is computed through a series of iterations using the Power Method, which is the numerical technique derived from the Markov Chains equation to find the principal eigenvector. The structure of the web is converted into a transition matrix of W where the matrix element of W_{ij} represents the probability of transitioning from page i to j. The probability distribution vector is then multiplied by the transition matrix until the distribution converges to the final stationary distribution vector.

Findings

The simulations presented through this project replicate the core PageRank algorithm while utilizing a graph-based random walk with a matrix-based iterative solution derived from the foundation of the Markov Chains equation. There were three imperative findings that confirmed the theoretical basis of this project:

1. The iterative approach converges to a stable probability distribution which indicates the existence of the final stationary distribution vector or the PageRank vector.

2. The algorithm is susceptible to particular manipulative strategies that could impact the overall result of the algorithm. For instance, if one artificially linked a large number of pages to a new, low-quality page, its overall PageRank score could be inflated which can affect the overall outcome of the simulation.

3. Certain edge cases such as pages with no outgoing links are crucial for convergence. It is necessary for a mechanism to be implemented to randomly revitalize the random surfer to maintain the probability distribution.

2 Conclusion

PageRank Convergence: The Power Method

The final stationary distribution vector was determined by iteratively applying the Transition Matrix to the current probability distribution until the overall distribution stabilized.

Matrix-Vector Multiplication

The iterative step of $\pi^{k+1} = \pi^k W$ is handled through the implementation of the `matrixVectorMultiplication` function which performs the calculation implied from the Markov Chains equation. This function showcases the iterative approach which also emulates the main purpose of the Power Method.

```
vector<double>
   matrixVectorMultiplication(vector<vector<double>>
   matrix, vector<double> vect)
```

```
    {
        int n = matrix.size();
        vector<double> result(n, 0.0);
        for (int i = 0; i < n; i++)
        {

            for (int j = 0; j < n; j++)
            {
                result[i] += (matrix[i][j] * vect[j]);
            }
        }
        return result;
    }
```

This function calculates the dot product between each column of the Transition Matrix and the previous PageRank vector to determine the next rank of the page *i*, which models the movement of the random surfer in a single step.

Rapid Convergence to Stationary Distribution

The results derived from these iterations confirm the fundamental property of the Power Method in which the PageRank score rapidly converges to a stable stationary distribution. For instance, Figure 1 demonstrates a simulation of 10 iterations that determines convergence within a specific time.

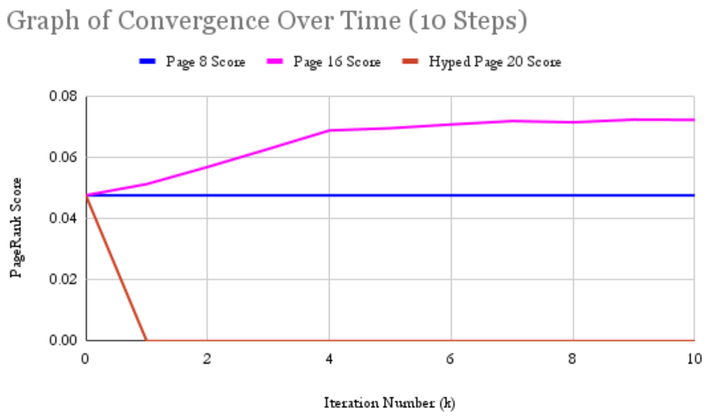

Figure 1: Convergence over 10 steps

The lines for the Page 8 Score and the Page 16 score rise significantly in the first few iterations which then begins to flatten out during the final few iterations. This transition indicates the stabilization of the distribution towards its final stages of their iterations. The line for the Hyped Page 29 score highlight the dead-end flaw which illustrates the loss of probability mass due to the unhandled edge case of dead-end pages within the model. Similarly, the simulation was run with 1000 steps to see how the model behaves under a large number of iterations as seen in Figure 2.

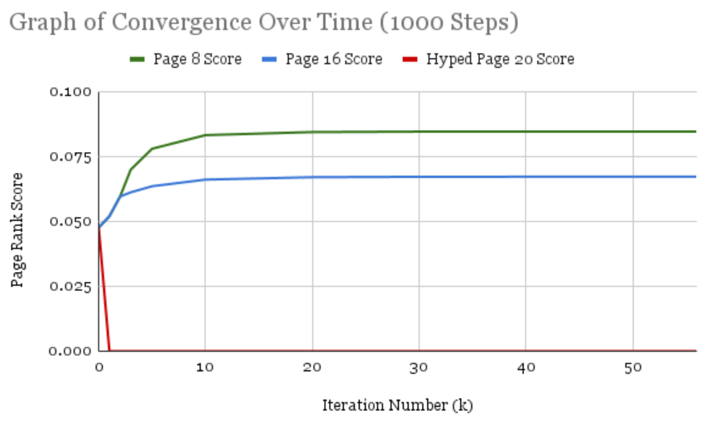

Figure 2: Convergence over 1000 steps

The lines for the Page 8 Score and the Page 16 behave relatively the same as Figure 1; however, the key difference between both iterations is that each model experience stabilization at different times. For instance, Figure 1 highlights the dramatic change of stabilization within the first 5 to 7 iterations, demonstrating the rapid initial approach to its stationary distribution. However, the 1000 step simulation proves that true convergence is not achieved until k = 56 steps, highlighting a later stage of stabilization. [3]

Failure to Handle Dead Ends (Limitation)

Throughout the final stages of the project, the simulation resulted in a flawed attempt of optimizing the search engine by including the Hyped Page 20 to emulate incoming links but zero outgoing links.

```
// exercise 50.11
    // create a transition matrix
    vector<vector<double>> transitionMatrix(int n)
    {
        for (int i = 0; i < n; i++)
        {

            int numLinks = pages[i]->pageLinks.size();
            double prob;

            if (numLinks == 0){
                prob = 1.0;
            }
            else{
                prob = 1.0 / numLinks;
            }
        }
    }
```

The Hyped Page 20 score or the ”dead end” page causes the loss of probability mass from the system due to its lack of stochastic properties. As shown in Figure 1 and Figure 2, the Hyped Page 20 score instantly drops from the initial value of k = 0 which supports the idea that the current model is not stochastic. Although this is a result that is perceived as a failed attempt, it can be seen as a form of limitation and an opportunity to highlight the need for a a damping factor and a proper dead-end fix to prevent rank deflation.

Graph Structure: Shortest Paths

To construct the topological structure of the randomly generated web, a Breadth-First Search (BFS) algorithm was implemented to help explore the graph structure needed for PageRank computation. [2]

Breath-First Search (BFS) implementation: Shortest Path Analysis

The `shortest` function uses a queue to initialize the BFS algorithm, which is optimal for finding the shortest path within an unweighted graph.

```
void shortest(shared_ptr<Page> start, int numPages)
{
    Q.push(start); // put first page on the queue

    while (!Q.empty())
    {
        auto a = Q.front();
        Q.pop();

        int d = distance[a->index]; // distance it took to
    reach page u

        for (auto b : a->pageLinks)
        {
            if (distance[b->index] == -1)
            {
                distance[b->index] = d + 1;
                Q.push(b);
            }
        }
        }
        }
}
```

The BFS explores the graph l;ayer by layer which guarantees that the first time a page is reached, it is the shortest path from the starting page. The distance vector tracks the number of steps taken, and the queue is the primary data structure that ensures a First-In, First-Out (FIFO) exploration which is essential to the foundation of the BFS to find the shortest path. [4]

Page ID	Distance (Shortest Path)
0	0
19	1
11,17	2
3,6,13,20	3
8,10,12,14	4
4,9,16,18	5
7	6

Table 1: Minimum Number of Clicks (10 Steps)

Table 1 demonstrates the minimum number of clicks required to reach various pages starting from Page 0. The BFS mentioned earlier is imperative in this analysis since the diameter is defined as the maximum shortest path distance between any two nodes in a graph.

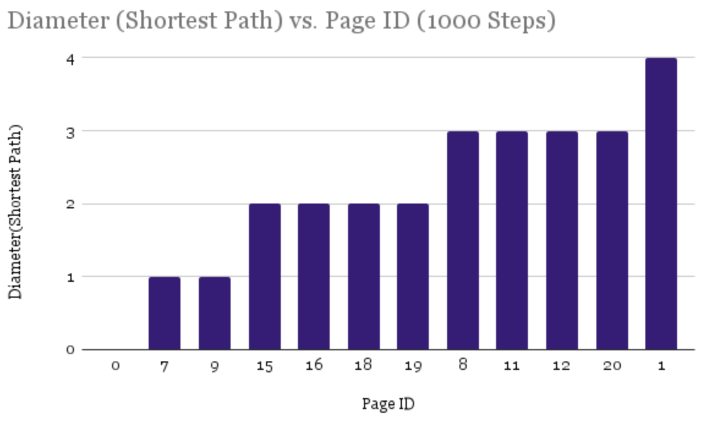

Figure 3: Diameter(Shortest Path) over 1000 steps

Similarly, over the simulation of the 1000 step iteration, it can be seen that the maximum distance within this case is 4 steps signifying the diameter of the connected components within the graph. The left-skewed distribution showcased from the graph characterizes the model's compactness or sparsity between each data point.

Key Differences

The key differences between the caclulated PageRank Score and the Shortest Path is the structural foundation or approach applied to each method. For instance the PageRank Score was determined through the probability of landing on a page after many random clicks, deriving its main functionality from the basis of the Markov Chains equation. However, the main approach for calculating the shortest path between nodes within the graph can be acquire through the implementation of a Breadth-First Search algorithm. This approach resulted in calculating the structural distance or minimum clicks between two pages. While the shortest path distribution defines the boundaries and compactness of the graph, the PageRank score determines which page within those boundaries consists of the highest authority.

Future Implementations

The current simulation demonstrates an analytical approach to the PageRank score method with a Breadth-First Approach to emulate the topology of the web which resulted in particular insights related to convergence and diameter calculation. Some features that could be implemented in order to improve the overall accuracy of the model include:

1. Damping Factor: Introducing a damping factor can help model the probability that a random surfer will ”teleport” to any page instead of following an existing link.
2. Link Weightage: Introducing link weighting in the `transitionMatrix` function can help the simulation become more realistic since this approach takes into account of higher weights to links to pages within the same domain.

3 Bibliography

References

[1] Section 4.5: Markov chains and google's pagerank algorithm. `https://math.libretexts.org/Bookshelves/Linear_Algebra/Understanding_Linear_Algebra_(Austin)/04%3A_Eigenvalues_and_eigenvectors/4.05%3A_Markov_chains_and_Google's_PageRank_algorithm`, n.d. Accessed: 2025-12-08.

[2] Priyanka S. Bandagale and Swati Powar. Web structure mining using breadth first search with pagerank. *International Journal of Creative Research Thoughts (IJCRT)*, 11:e885–e891, February 2023. This paper discusses the combined use of BFS and PageRank in Web Structure Mining.

[3] Jean-Baptiste Lepidi and Dario Malchiodi. *PageRank Convergence Proof.* University of Milan, Course on "Algorithms for massive datasets", 2023.

[4] Remus Radu and Raluca Tanase. Lecture #3: Pagerank algorithm — the mathematics of google search.