PageRank

A Discrete Mathematics Perspective

---

1. Problem Statement

1.1 What is PageRank?

PageRank is a mathematical algorithm designed to rank the importance of nodes (webpages) in a directed graph (the web). Developed by Larry Page and Sergey Brin at Stanford University in 1996-1998, PageRank forms the foundational algorithm that launched Google as a search engine.

At its core, PageRank assigns a numerical weight to each element of a linked set of objects (such as hyperlinked documents on the World Wide Web) with the purpose of measuring its relative importance within the set. The intuition behind PageRank is that more important webpages likely receive more links from other webpages.

1.2 Why Rank Nodes in a Directed Graph?

Ranking nodes in a directed graph serves to quantify the structural importance or centrality of each node within the network. This problem appears across numerous domains:

- In web search, determining which pages are most relevant to user queries
- In social networks, identifying influential individuals
- In citation networks, finding seminal research papers
- In biological networks, discovering critical proteins or genes
- In recommendation systems, identifying key products or content

The key insight of PageRank is that importance flows through connections—a node is important if important nodes point to it.

1.3 Motivation and Applications

Web Search Ranking

PageRank revolutionized web search by providing a way to rank search results based on link structure rather than just keyword matching. Before PageRank, search engines struggled with determining which of thousands of pages containing the same keywords were most relevant. Statistics demonstrate its impact:

- Google's market share grew from 0% to over 85% in two decades largely due to superior ranking
- Studies show that over 90% of users rarely go beyond the first page of results, making accurate ranking critical

Influence Measures in Network Analysis

Beyond web search, PageRank serves as a fundamental measure of centrality in network analysis:

- In social media platforms like Twitter, modified PageRank algorithms help identify influential users
- Academic repositories use similar methods to identify seminal papers
- Financial institutions apply these algorithms to understand systemic risk in banking networks

Recommendation Systems

Modern recommendation engines incorporate PageRank-like algorithms to:

- Suggest products in e-commerce platforms
- Recommend content in streaming services
- Identify potential connections in professional networks

2. Literature Review

2.1 Seminal Papers

Brin & Page (1998): "The Anatomy of a Large-Scale Hypertextual Web Search Engine"

This foundational paper introduced PageRank and the initial Google architecture. Brin and Page framed the web as a vast directed graph and proposed using its link structure to determine the importance of pages. They defined the "random surfer" model that became the intuitive explanation for PageRank. The paper established that a page has high rank if the sum of the ranks of its backlinks is high—a recursive definition that led to an elegant mathematical solution using Markov chains.

Langville & Meyer (2004): "Deeper Inside PageRank"

This paper explored the mathematical properties of PageRank in depth, particularly focusing on the sensitivity and stability of the algorithm. Langville and Meyer provided a comprehensive analysis of the convergence properties of the power method for computing PageRank, and examined how small changes in the web graph affect the final rankings. They established important bounds on the rate of convergence and the effect of various parameters like the damping factor.

Haveliwala (2002): "Topic-Sensitive PageRank"

Haveliwala extended the original PageRank concept by introducing context into the ranking process. Rather than computing a single global ranking, this paper proposed computing multiple PageRank vectors biased toward different topics. When a query is issued, the ranking system combines these pre-computed vectors based on the topic of the query. This innovation significantly improved the relevance of search results by considering not just link structure but also content themes.

This paper demonstrated how PageRank could be made context-sensitive, addressing one of the main limitations of the original algorithm—its inability to distinguish between different user interests and query contexts.

Jeh & Widom (2003): "Scaling Personalized Web Search"

This paper introduced the concept of Personalized PageRank, which tailors the ranking to individual user preferences. The authors developed efficient methods for computing these personalized rankings on-the-fly, making personalization computationally feasible at scale. The technique involves creating and storing a set of basis vectors that can be combined to quickly approximate personalized rankings.

2.2 Extensions of PageRank

Personalized PageRank

Personalized PageRank modifies the original algorithm by adjusting the random jump probabilities to reflect user preferences:

- Instead of jumping to any page with equal probability, the surfer jumps to pages from a user's preferred set
- This creates rankings biased toward a user's interests
- Applications include personalized recommendations and content filtering

Mathematically, this changes the teleportation matrix in the PageRank formula to direct random jumps toward specific pages rather than distributing them uniformly.

Topic-Sensitive PageRank

Topic-Sensitive PageRank pre-computes multiple PageRank vectors, each biased toward a particular topic:

- When a query is issued, the appropriate topic-biased ranking is used
- Significantly improves search relevance compared to a single global ranking
- Computationally efficient since topic vectors can be pre-computed

3. Course Connection:

The Discrete Mathematics of PageRank

3.1 Graph Theory Foundations

PageRank is fundamentally built on discrete mathematics concepts, particularly graph theory. Before delving into the algorithm, let's establish the mathematical framework:

Directed Graphs

A directed graph (or digraph) G = (V, E) consists of:

- A set V of vertices (or nodes)
- A set E of ordered pairs of vertices called edges (or arcs)

In the context of the web:

- Vertices represent webpages
- Edges represent hyperlinks from one page to another

Adjacency Matrix

For a graph with n nodes, the adjacency matrix A is an n×n matrix where:

- A[i,j] = 1 if there is a directed edge from node i to node j
- A[i,j] = 0 otherwise

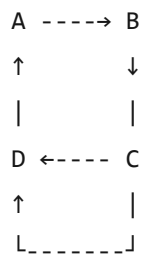
For PageRank, we typically work with the column-stochastic transition matrix derived from the adjacency matrix:

$$P[i,j] = A[j,i] / \sum_k A[j,k]$$

- The denominator $\sum_k A[j,k]$ is the total number of outgoing links from page j.
- The result P[i,j] is the probability of moving from page j to page i in a random walk.

Visual Example

Consider this simple directed graph representing a tiny web of 4 pages:



The adjacency matrix for this graph would be:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

3.2 Markov Chains

PageRank's mathematical elegance comes from its connection to Markov chains—a core concept in discrete probability.

Definition and Properties

A Markov chain is a stochastic model describing a sequence of possible events where the probability of each event depends only on the state in the previous event.

Key properties relevant to PageRank:

- **State space**: The set of all possible states (webpages in our context)
- **Transition probabilities**: The probability of moving from one state to another
- **Stationarity**: Transition probabilities don't change over time
- **Irreducibility**: It's possible to get from any state to any other state
- **Aperiodicity**: The chain doesn't get stuck in cycles

The Random Surfer Model

PageRank cleverly models web browsing as a Markov process:

- A surfer randomly follows links from page to page
- At each step, the surfer has two options:
    1. With probability d (typically 0.85), follow a random link from the current page.
    2. With probability (1-d), jump to any random page.

This model ensures that the Markov chain is both irreducible and aperiodic, guaranteeing that a unique stationary distribution exists.

The random surfer model translates the abstract mathematical concept of Markov chains into an intuitive model of user behavior, while the damping factor d addresses practical issues like trapped states in the graph.

3.3 Eigenvalues and Eigenvectors

The mathematical heart of PageRank lies in finding the principal eigenvector of a modified web adjacency matrix.

Stochastic Matrix Formulation

We first convert the web graph into a stochastic matrix M:

1. Start with the adjacency matrix A
2. For each column in A (representing outlinks from a page):
    - If the column sum is not zero, divide each entry by the sum
    - If the column sum is zero (dangling nodes), replace with uniform probabilities 1/n
3. Apply the damping factor adjustment:

$$M = d \times S + (1\text{-}d) \times (1/n)E$$

where:

- M is the Google matrix (the modified transition matrix used in PageRank)
- S is the stochastic matrix from steps 1-2
- E is a matrix of all 1's
- d is the damping factor (typically 0.85)
- n is the number of pages

PageRank as an Eigenvector Problem

The PageRank vector p is the solution to:

$$p = M \times p$$

In other words, p is the eigenvector of M corresponding to eigenvalue 1. This eigenvector represents the stationary distribution of the Markov chain (Google matrix) —the long-term probability of being at each page when randomly surfing.

Power Method Solution

Since direct computation of eigenvectors for massive matrices is impractical, PageRank typically uses the power method:

1. Start with an initial vector:

$$p_0 = [1/n, 1/n, ..., 1/n]$$

   Where:

   - p^0 is the initial PageRank vector
   - n is the number of nodes (webpages)
   - The vector assigns equal importance to all pages as a starting point

2. Iteratively compute:

$$p_{k+1} = M \times p_k$$

   Where:

   - $p^{(k)}$is the PageRank vector at iteration t
   - $p^{(k+1)}$ is the updated PageRank vector
   - M is the Google matrix

This step repeatedly applies the transition matrix to the current vector.

3. Component-wise Calculation:

$$p\_j^{(k+1)} = \sum(i \rightarrow j)\ p\_i^{(k)}/d\_i$$

Where:

- $p\_j^{(k+1)}$ is the updated PageRank value for page j
- The sum is over all pages i that link to page j
- $p\_i^{(k)}$ is the current PageRank value of page i
- d_i is the out-degree of page i (number of outgoing links)

This formula shows how the PageRank flows through the network: each page distributes its current rank equally among all the pages it links to.

4. Repeat until convergence:

$$||p_{k+1} - p_k|| < \varepsilon$$

Where:

- $|\cdot|_1$ denotes the L_1 norm (sum of absolute differences)
- ε is a small positive number representing the convergence tolerance

The discrete nature of this iterative process makes it particularly well-suited for computational implementation, often converging in 50-100 iterations for the web graph.

References

1. Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems, 30(1-7), 107-117.
2. Langville, A. N., & Meyer, C. D. (2004). Deeper inside PageRank. Internet Mathematics, 1(3), 335-380.
3. Haveliwala, T. H. (2002). Topic-sensitive PageRank. Proceedings of the 11th International Conference on World Wide Web, 517-526.
4. Jeh, G., & Widom, J. (2003). Scaling personalized web search. Proceedings of the 12th International Conference on World Wide Web, 271-279.
5. Berkhin, P. (2005). A survey on PageRank computing. Internet Mathematics, 2(1), 73-120.
6. Boldi, P., Santini, M., & Vigna, S. (2005). PageRank as a function of the damping factor. Proceedings of the 14th International Conference on World Wide Web, 557-566.