Discrete Mathematics Project: PageRank Mathematical Formulation and Power Method

Your Name

May 2025

1 Introduction

This document presents the mathematical formulation of the PageRank algorithm as an eigenproblem and the design of the Power Method for computing the PageRank vector. The work is part of a team project in Discrete Mathematics, focusing on formalizing the algorithm, deriving key equations, and analyzing its computational aspects.

2 Mathematical Formulation

Let $G = (V, E)$ be a directed graph with $n = |V|$ nodes. The PageRank vector $r \in \mathbb{R}^n$ assigns each node a score reflecting its importance in the graph.

2.1 Link Matrix Definition

The link matrix $P \in \mathbb{R}^{n \times n}$ represents the transition probabilities based on outgoing links:

$$P_{ij} = \begin{cases} \frac{1}{\text{outdeg}(j)} & \text{if } (j, i) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

For dangling nodes (nodes with zero out-degree), the corresponding column in P sums to zero, which is addressed through teleportation.

2.2 Incorporating Teleportation

To ensure the transition matrix is stochastic, we introduce a damping factor $0 < \alpha < 1$ and a teleportation vector $v \in \mathbb{R}^n$ with $\sum_i v_i = 1$, typically $v_i = \frac{1}{n}$ for uniform teleportation. The Google matrix M is defined as:

$$M = \alpha P^\top + (1 - \alpha)\mathbf{1}v^\top,$$

where $\mathbf{1}$ is the all-ones vector. Since $v_i > 0$ for all i, M is column-stochastic, irreducible, and aperiodic, ensuring a unique stationary distribution.

2.3 Stationary Distribution Eigenproblem

The PageRank vector r is the stationary distribution of M, satisfying:

$$r = Mr,$$

or equivalently:

$$r = \alpha P^\top r + (1 - \alpha)v,$$

with the normalization constraint $\|r\|_1 = 1$. By the Perron–Frobenius theorem, M (being positive and irreducible) has a unique positive eigenvector r corresponding to the eigenvalue 1.

Proof Sketch: The Perron–Frobenius theorem states that a positive, irreducible matrix has a simple eigenvalue 1 with a positive eigenvector. Since M is column-stochastic, the equation $r = Mr$ defines the stationary distribution, which is unique up to scaling. The constraint $\|r\|_1 = 1$ ensures a unique solution.

3 Power Method Algorithm

The Power Method is used to iteratively compute the PageRank vector r. The algorithm initializes a uniform vector and iteratively applies M until convergence.

Algorithm 1 Power Method for PageRank

Require: Link matrix P, teleportation vector v, damping factor α, tolerance ε, maximum iterations max_iter
Ensure: PageRank vector r, number of iterations k

teleport $\leftarrow (1-\alpha)v$
$n \leftarrow$ length(v)
$r_{\text{old}} \leftarrow \frac{1}{n}\mathbf{1}$
for $k = 1$ to max_iter **do**
    $r_{\text{new}} \leftarrow \alpha P^\top r_{\text{old}} +$ teleport
    $r_{\text{new}} \leftarrow \frac{r_{\text{new}}}{\|r_{\text{new}}\|_1}$ ▷ Normalize to ensure $\|r_{\text{new}}\|_1 = 1$
    residual $\leftarrow \|r_{\text{new}} - r_{\text{old}}\|_1$
    if residual $< \varepsilon$ **then**
        return r_{new}, k
    end if
    $r_{\text{old}} \leftarrow r_{\text{new}}$
end for
warn "Power Method did not converge within max_iter iterations"
return r_{new}, max_iter

3.1 Complexity Analysis

- **Time Complexity:** Each iteration computes a matrix-vector product. For a dense matrix P, this costs $O(n^2)$. For a sparse matrix with m nonzeros, it costs $O(m)$. Additional vector operations (addition, normalization, residual computation) cost $O(n)$.
- **Memory Complexity:** Storing a dense P requires $O(n^2)$ memory, while a sparse P requires $O(m)$. The algorithm also uses two n-vectors (r_{new}, r_{old}) and the teleport vector, totaling $O(n)$ additional memory.

4 Parameter Discussion

The Power Method's performance depends on several parameters, summarized below:

- **Damping Factor** α: The default $\alpha = 0.85$ balances the influence of the graph's link structure and teleportation. Smaller α values increase teleportation, making r closer to v, while larger α values emphasize links, potentially slowing convergence.
- **Tolerance** ε: A smaller ε yields a more precise r but requires more iterations. Typical values range from 10^{-8} for high precision to 10^{-4} for faster computation.

Parameter	Default	Range	Description
Damping factor α	0.85	[0.5, 0.95]	Trade-off between link-following and teleportation
Tolerance ε	10^{-6}	10^{-8} to 10^{-4}	Threshold for convergence based on $\|r^{(k)} - r^{(k-1)}\|_1$
Initialization strategy	Uniform	Uniform, Random	Initial vector; affects convergence speed, not final r
Max iterations	100	100–10000	Maximum iterations before termination

Table 1: Power Method parameters and recommended settings

- **Initialization Strategy**: A uniform vector ($r_i^{(0)} = \frac{1}{n}$) is standard and ensures stable convergence. Random initialization may slightly alter the number of iterations but does not affect the final r.
- **Max Iterations**: A cap of 100 iterations is often sufficient for small graphs with $\alpha = 0.85$. Larger graphs or higher α may require up to 10,000 iterations.

5 Conclusion

This deliverable formalizes the PageRank algorithm as an eigenproblem and designs the Power Method for its computation. The mathematical derivation clarifies the role of the Google matrix and teleportation, while the pseudocode and complexity analysis provide a practical implementation guide. The parameter discussion offers insights for tuning the algorithm. Future work includes implementing and testing the algorithm using Python with libraries like NumPy and NetworkX, as well as visualizing convergence behavior.