PageRank Algorithm

Guxiao Hu, Puyuan Zhang, Xiu Chen, Zipei Zhu

Wednesday 3rd December, 2025

Table of Contents

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

- Relevance

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

- Relevance
  - NLP-focused approaches. May be discussed at the end of the pre (also linear-algebra heavy!).

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

- Relevance
  - NLP-focused approaches. May be discussed at the end of the pre (also linear-algebra heavy!).
- Importance

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

- Relevance
    - NLP-focused approaches. May be discussed at the end of the pre (also linear-algebra heavy!).
- Importance
    - Sorting on accessing counts?

How do search engines work?

You are an engineer in Gugloo designing a search engine in the 1990s.

Consider: how to rank the results for each query?

- Relevance
  - NLP-focused approaches. May be discussed at the end of the pre (also linear-algebra heavy!).
- Importance
  - Sorting on accessing counts?

PageRank algorithm focuses on *how important* each webpage is.

The “Random Surfer” Model

The PageRank algorithm

- iterates on a **graph** where webpages are **nodes**, and links are **directed edges**,
- calculates the probability of a “random surfer” visiting each of the webpages,
- outputs a probability distribution vector.

$$\boldsymbol{p} = \begin{bmatrix} \Pr(1) & \Pr(2) & \cdots & \Pr(n) \end{bmatrix}^{\mathsf{T}}$$

Algorithm

Notations and Conventions

- Let n denote the number of nodes (i.e. total number of webpages).
- Let $\deg^+(u)$ denote the **out degree** of node u (i.e. number of outreaching links on webpage u).

Definition

Google PageRank[a] **iteration**: Initially, $\Pr(i) := 1/n$ for each webpage $1 \le i \le n$. Then iterate by

$$\Pr(v) := \frac{1-d}{n} + d \sum_{\text{edge } u \to v} \frac{1}{\deg^+(u)} \Pr(u)$$

where d is the **damping factor** $(0 < d < 1)$.

[a] Trademark of Google; U.S. patent 6,285,999.

Computing PageRank

- Storing graphs in computers → adjacent matrices;

Computing PageRank

- Storing graphs in computers → adjacent matrices;
- Converting adjacent matrices to “multiplicable” matrices for iterations;

Computing PageRank

- Storing graphs in computers → adjacent matrices;
- Converting adjacent matrices to “multiplicable” matrices for iterations;
- Linear iterations → multiplication of matrices.

Matrix Algebra I

A Simplified Model

Let **stochastic matrix**

$$\mathscr{M}_{ij} := \begin{cases} \dfrac{1}{\deg^+(j)} & \text{edge } j \to i \\ 0 & \text{otherwise} \end{cases}$$

i.e., given **adjacent matrix** $A_{ij} = [\text{edge } i \to j]$ and diagonal matrix K with the outdegrees in the diagonal,

$$\mathscr{M} := (K^{-1}A)^{\mathsf{T}}$$

Matrix Algebra II

Definition

Let **probability distribution vector** of the k-th iteration be

$$\boldsymbol{p}(k) := \begin{bmatrix} \Pr(1) & \Pr(2) & \cdots & \Pr(n) \end{bmatrix}^\mathsf{T}$$

which initially sets to

$$\boldsymbol{p}(0) := \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \end{bmatrix}^\mathsf{T}$$

Matrix Algebra III

Definition

The **Google matrix** $\widehat{\mathscr{M}}$ is defined by

$$\widehat{\mathscr{M}} := d\mathscr{M} + \frac{1-d}{n}E$$

where E is $n \times n$ matrix of all ones (so that $E\boldsymbol{p} = \mathbf{1}$), and $0 < d < 1$ is the damping factor.

The Power Method

Google PageRank can be computed by

$$\boldsymbol{p}(k+1) = \widehat{\mathscr{M}}\boldsymbol{p}(k) \Longrightarrow \boldsymbol{p}(k) = \left(\widehat{\mathscr{M}}\right)^k \boldsymbol{p}(0)$$

Applications

- Evaluating academic papers based on citation counts.

Further Reading

Suggested topics for further reading:

- Markov Chains
- Some Easy Ways to Briefly Analyze “Relevance”:
    - Word frequency (TF / TF-IDF)
    - Proximity scoring (word vectors)
    - Matching phrases (bags-of-words)

Thanks!