Information Retrieval and Web Search

Cornelia Caragea

Computer Science
University of Illinois at Chicago

The PageRank Algorithm for Web Ranking

Required Reading

- “Information Retrieval” textbook
  - Chapter 21: Link Analysis

The PageRank Algorithm

PageRank

Query-independent ranking algorithm:

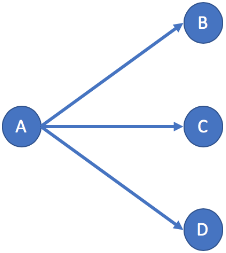
- Alternative link-analysis method used by Google (Brin & Page, 1998)
- Does not attempt to capture the distinction between hubs and authorities
- Ranks pages just by authority
- Applied to the entire web rather than a local neighborhood of pages surrounding the results of a query
- The endorsement that forms the basis for the PageRank measure of importance is that a page is important if it is pointed to by other important pages

Idea Behind PageRank

- Each node in the Web graph receives a PageRank score, which depends on the link structure of the Web
- Just measuring in-degree does not account for the authority of the source of a link
- PageRank starts with the simple "voting" based on in-links
- Nodes repeatedly pass endorsements across their out-going links, with the weight of a node's endorsement based on the current estimate of its PageRank
  - More important nodes make stronger endorsements

Initial PageRank Idea

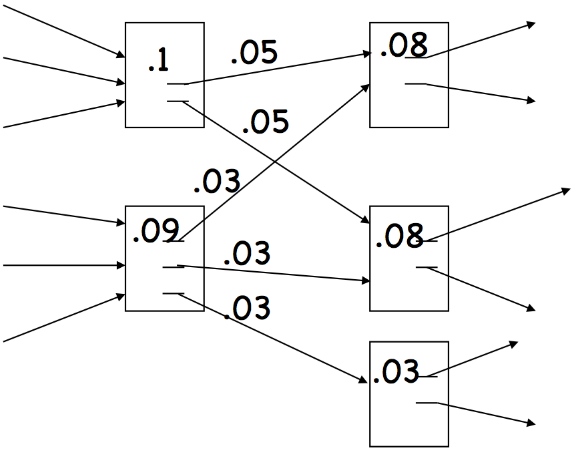
- A surfer begins at a Web page
- At each time step, the surfer follows a link from the current page chosen at random.



- During the random walk, the surfer visits some nodes more often than others.
- Pages visited more often are more important.

Initial PageRank Idea

- Example:

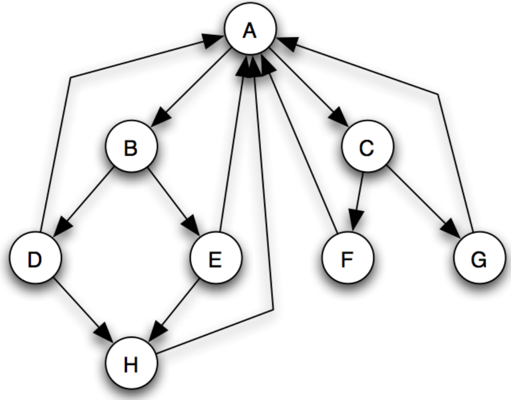


Initial PageRank Algorithm

- Iterate rank-flowing process until convergence:
- Let V be the set of Web pages
- Initialize $S(A) = \frac{1}{|V|} = \frac{1}{n}$ for all $A \in V$
- Until ranks do not change (convergence)
    - For each $A \in V$:

$$S(A) = \sum_{B \to A} \frac{S(B)}{out(B)}$$

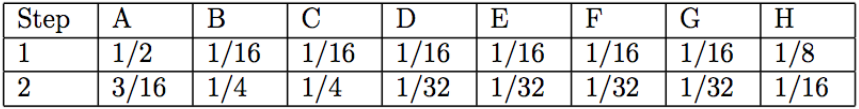
PageRank Algorithm - Example



What are the PageRank values after the first two updates?

PageRank Algorithm - Example

Result



Step	A	B	C	D	E	F	G	H
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	3/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

PageRank - Matrix Notation

- In matrix notation, if M is the adjacency matrix of the Web graph G = (V, E), and $\widetilde{M}$ is the normalized form of matrix M with $\widetilde{m_{ij}} \in \widetilde{M}$ defined as:

$$\widetilde{m_{ij}} = \begin{cases} m_{ij} / \sum_{j=1}^{|V|} m_{ij} & \text{if } \sum_{j=1}^{|V|} m_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

  - Then the vector of PageRank scores at time $t+1$ is iteratively calculated using:

$$S(t+1) = S(t) \cdot \widetilde{M}$$

where

$$S(0) = \left[\frac{1}{|V|}, \cdots, \frac{1}{|V|} \right]$$

Problem with Initial Idea

- The web is full of dead-ends
    - What if the current location of the surfer has no out-links?
    - Consider the graph $A \to B$; $A \to C$; $B \to C$.
- Introducing the idea of *teleporting*.

Teleporting

- The surfer jumps from a node to any other node in the Web graph uniformly at random.
- At a dead end, jump to a random web page
- At any non-dead end, with probability 15%, jump to a random web page
- With remaining probability (85%), go out on a random link
  - 15% - the ϵ parameter

$$S(A) = \frac{\epsilon}{n} + (1-\epsilon) \sum_{(B,A)\in G} \frac{S(B)}{out(B)}$$

- Result of teleporting: it cannot get stuck locally

The PageRank Algorithm

- Let V be the total set of pages and $n = |V|$
- Choose ϵ s.t. $0 < \epsilon < 1$, e.g., 0.15
- Initialize $S(A) = \frac{1}{n}$ for all $A \in V$
- Until ranks do not change (convergence)
  - For each $A \in V$:

$$S(A) = \left[(1-\epsilon) \sum_{B \to A} \frac{S(B)}{out(B)} \right] + \frac{\epsilon}{n}$$

- In matrix notation,

$$S(t+1) = (1-\epsilon) S(t) \cdot \widetilde{M} + \epsilon \cdot \tilde{p}$$

where

$$\tilde{p} = \left[\frac{1}{n}, \cdots, \frac{1}{n} \right]$$

The Random Surfer Model

- PageRank can be seen as modeling a "random surfer" that starts on a random page and then at each point:
  - With probability ϵ randomly jumps to page A
  - Otherwise, randomly follows a link on the current page
- S(A) models the probability that this random surfer will be on page A at any given time
- "Jumps" are needed to prevent the random surfer from getting "trapped" in web sinks with no outgoing links

Speed of Convergence

- Early experiments on Google used 322 million links
- PageRank algorithm converged (within small tolerance) in about 52 iterations
- Number of iterations required for convergence is empirically $O(logn)$ (where n is the number of links)

PageRank Retrieval

- Preprocessing:
  - Given graph of links, compute the rank of each page A
- Query processing:
  - Retrieve pages meeting query
  - Rank them by their PageRank
  - Order is query-independent
- The reality
  - PageRank is used in Google, but so are many other clever heuristics

PageRank vs. HITS

- Computation
    - Once for all documents and queries (offline)
- Query-independent
    - Requires combination with query-dependent criteria

- Computation
    - Requires computation for each query
- Query-dependent
- Quality depends on quality of start set
- Gives hubs as well as authorities

Link Analysis Conclusions

- Link analysis uses information about the structure of the web graph to aid search
- It is one of the major innovations in web search
- It is the primary reason for Google's success