PageRank

Ben Burns, Dan Magazu, Lucas Chagas,
Thomas Webster, Trung Do

Fall 2021

Table of Contents

Motivation

- Problem: the internet has a *lot* of web pages
- A lot of the information out there either isn't relevant to us, or is inaccurate
- Motivation: we want a program that, when provided a phrase, returns webpages with information relevant to the input
- Intuitive solution: return back websites that either contain that phrase, or contain similar phrases
- This helps us find more *relevant* pages, but we can't know if we're getting the best information (much less *accurate* information) without manually going through each result
- We desire a stronger solution

PageRank

- Invented by Sergey Brin and Larry Page (1998)[1]
  - Publication marks them becoming co-founders of Google
- Idea: we want some way to numerically score each webpage based on how "important" it is
- Algorithm numerically scores each page p based on
  - How many other pages link to p (or "cite" it)
  - The "importance" of each of p's citations
- We then numerically order pages to rank them
- PageRank: the procedure for scoring each website
- Google: the database that indexes the PageRank of each website for search

[1] Many use the year of the original manuscript, 1996

Underlying Assumption

- Running our basic search engine gives us a collection of pages with information relevant to our query
- Assumption: More "important" and useful websites will be the ones with proportionally more inbound links
- Pages with very reliable, primary information are likely to be cited by lots of website authors, and therefore will have lots of "flow" into them

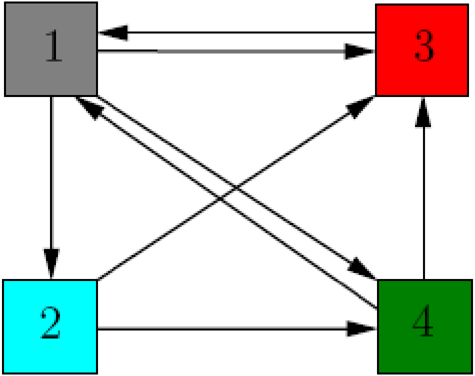
Finding Less Important Information

- A worry you may have is that we'll only find pages with lots of inbound links
- You can still find niche information by making your query more specific so that it won't match more general pages
- Searching "PageRank" will likely get you Wikipedia, but "Anatomy of PageRank architecture" gives you the original research literature.
- Your returned urls are still proportionally important results, your query just filtered out the numerically more "important", yet less relevant pages.

Formalizing the PageRank problem

- We're going to construct a directed graph $G = (V, E)$
- For each website we consider, we construct a node $v_i \in V$
- For two distinct nodes $v_i,\ v_j \in V$, the *directed* edge $v_i v_j \in E$ iff there is a link on website i that goes to website j.
- If v_i and v_j are not distinct (a website is linking to itself), we ignore the link and do not construct a loop edge.
  - G is not a psuedograph
- Multiple hyperlinks on page i to page j are all represented by the single, directed edge
  - G is not a multigraph.

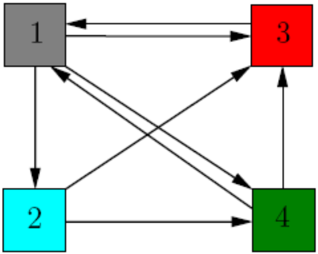
Visual Representation



- We have a set of 4 websites
- Each edge represents a hyperlink from the origin node to the destination node

Applying PageRank Values

- At first, we assign each vertex $v \in V$ with a weight of $\frac{1}{|V|}$.
- All vertices have equal weight, and our weights sum up to 1. The weight of a particular vertex v_i is denoted $PR(v_i)$.
- Consider the set V_i of vertices that v_i has an edge to:
  - $V_i = \{v_j | v_i v_j \in E\}$
- A user on page 1 can choose to click a link to traverse to either 2, 3, or 4. In other words, $V_1 = \{2, 3, 4\}$.



Traversing from Page to Page

- Assumption: A user on page i user is equally likely to choose to visit each vertex in V_i (our set of vertices that v_i cites)
- From v_i, $P(v_j) = \frac{1}{|V_i|}$ if $v_j \in V_i$, and 0 else.
- In our example from before, the probabilty that our user on page 1 visits page 2 is $P(v_2) = \frac{1}{|V_1|} = \frac{1}{3}$
- So a third of page 1's visitors will "flow" to page 2, a third to page 3, and a third to page 4

Iteration Model

- All at once, all users will click one of the links on their curreny page
- At click 0, the PageRank value of all vertices $PR(v) = \frac{1}{|V|}$
- On click 1, each page will equally split its PageRank value among the pages it cites
- Page 1 starts with $PR(v_1) = \frac{1}{4}$ at click 0, and contributes $\frac{1}{12}$ to each of 2, 3, and 4 on click 1
- Conversely, page 1 will receive nothing from page 2, all of $PR(v_3)$, and half of $PR(v_4)$. So after click 1, $PR(v_1) = 0 \cdot PR(v_2) + PR(v_3) + \frac{1}{2}PR(v_4) = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$
- After click n, $PR(v_1) = \frac{1}{|V_2|}PR(v_2) + \frac{1}{|V_3|}PR(v_3) + \frac{1}{|V_4|}PR(v_4)$