PageRank

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

MATH 455

Fall 2021

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

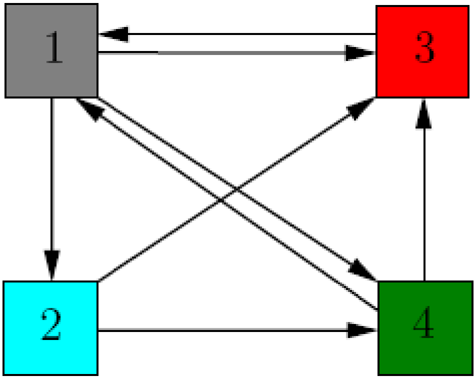
Underlying Assumption

- 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
- If what you're looking for is very niche and therefore won't have many inbound links, the best results will still have proprotionally more citations than other relevant sources
- 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

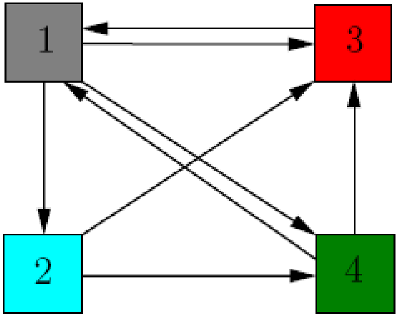
- We're going to construct a directed graph $G = (V, E)$
- For each website we consider, we construct a node $v_i \in V$
- For two distinct nodes $v_i,\ v_j \in V$, the *directed* edge $v_i v_j \in E$ iff there is a link on website i that goes to website j.
- If v_i and v_j are not distinct (a website 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 destinationx node

Adjacency Matrix



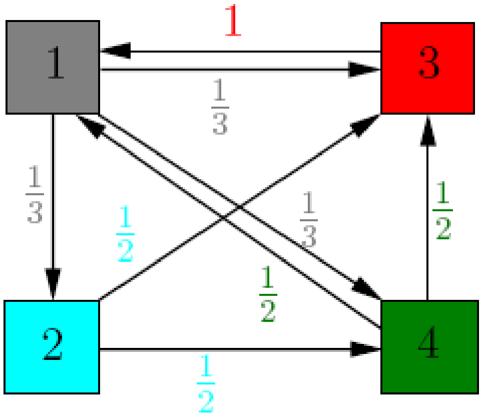
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

- No self loops means the main diagonal is all zeros

Applying PageRank Values and Edge Weights

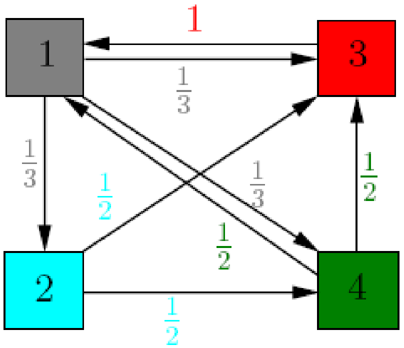
- At first, we assign each vertex with a weight of $\frac{1}{|V|}$
  - All vertices have equal weight, and our weights sum up to 1
- Consider the set of vertices that v_i has an edge to
  - $V_i = \{v_j | v_i v_j \in E\}$
- For each vertex in the set, the weight of the edge to that vertex will be the PageRank value of the source node divided by the number of outbound edges it has
  - $\forall v_j \in V_i : w(v_i v_j) = \frac{PR(v_i)}{|V_i|}$
- In other words, all the outbound edges from a particular vertex will have the same weight

Visual Representation



- In this case, all nodes have a PageRank value of 1.

Transition Matrix



$$T = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}$$

- All entries of the transition matrix are non-negative
- If v_i has at least one outgoing edge, the sum of the entries in row i is 1
  - Else, the sum is 0.
- This is a *column stochastic matrix*