PageRank

CS16: Introduction to Data Structures & Algorithms
Spring 2018
Outline

- Background
  - The Internet
  - World Wide Web
  - Search Engines
- The PageRank Algorithm
  - Basic PageRank
  - Full PageRank
- Spectral Analysis
- Applications
The Internet + the World Wide Web
The Internet

- Global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices

- TCP/IP
  - TCP is a protocol that allows two parties to establish a connection and send each other data
  - IP is a protocol that defines how the data should be “packaged”

- Internet is based on a system of routers
  - Routers receive and send IP packets
  - Distributed graph that decides how to route packets

- Initially developed for the Department of Defense
The World Wide Web

- Information space where pages are identified by URLs that can be accessed by the Internet
- Uniform Resource Locator (URL)
  - Specifies a web resource at its location in the Internet
- We’ll think about the Web as a collection of HTML pages composed of text and hyperlinks
- Hyperlink
  - Connects two web pages together
Timeline

1965
First computer networks created

1969
ARPANET established

1983
TCP/IP formalized

1989
First commercial ISPs formed

1990
First search engine created (Archie)

1991
World Wide Web invented

1994
First full text search engine created (WebCrawler)

1998
Google founded

1999
First mobile Internet service launched (i-mode)

2001
Dot com bubble bursts
Search Engines

- How can you best search for a webpage?
- Idea: look for all pages containing a given key word
  - Binary search tree
  - Hash table
- How do you rank the resulting pages?
  - What about ranking by frequency?
Search Engines

Rank by frequency
Larry Page and Sergey Brin

- Stanford PhD students
- Part of a research group looking to improve search engines
- In 1998, wrote a paper describing their algorithm PageRank
- Founded Google Inc.
- Legal battle over patent rights with Stanford
What is PageRank?

- Used to rank web pages by importance
  - Unlike text files, web pages have links to other web pages
  - If a web page is linked to by multiple other web pages, it is viewed as more important
- Underlying assumption is that “more important websites are likely to receive more links from other websites.”
The PageRank Algorithm

- The Web is a graph \( G = (V, E) \)
  - Vertices are web pages
  - Edges are links
- The pagerank of a page \( P \) is a function of the number of other pages that point to it and the rank of those pages
- If the pages that point to \( P \) have a high pagerank, \( P \)'s pagerank will be very high
- We don’t really know if this is what Google does
Basic PageRank Pseudocode

```python
BasicPageRank(G, k):
    for v in V:
        v.rank = 1/|V|;
    for i from 1 to k
        for v in V:
            for u in v.incoming:
                v.rank = v.rank + u.rank/|u.outgoing|;
```

Runtime: $O(|V| + |E|)$
Basic PageRank Example
Basic PageRank Example

**Activity #1**

\[ k = 3 \]

```python
BasicPageRank(G, k):
    for v in V:
        v.rank = 1/|V|;
    for i from 1 to k
        for v in V:
            for u in v.incoming:
                v.rank = v.rank + u.rank/|u.outgoing|;
```

4 min
Basic PageRank Example

Initial state

A

B

C

D

E

rank = 1/5

rank = 1/5

rank = 1/5

rank = 1/5

rank = 1/5
Basic PageRank Example

B

rank = 7/10

D

k = 1

rank = 1/2

A

rank = 1/5

C

rank = 1/5

E

rank = 1/5
Basic PageRank Example

k = 2

A

rank = 3/2

B

rank = 4/5

C

rank = 1/5

D

rank = 4/5

E

rank = 1/5
Basic PageRank Example

rank = 13/5

rank = 1/5

rank = 1/5

k = 3

rank = 11/10

rank = 1/5

rank = 1/5
Basic PageRank

- What is an issue with this algorithm?
  - Pagerank can be concentrated between a subset of nodes in the graph
Full PageRank Algorithm

- To prevent concentrated pagerank, we’re going to slightly modify how we calculate rank each time

\[
\text{rank}(v) = \sum_{u \in \text{incoming}(v)} \left( \delta \cdot \frac{\text{rank}(u)}{|\text{outgoing}(u)|} + \frac{(1 - \delta)}{|V|} \cdot \text{rank}(u) \right)
\]

- Let’s break this equation down
Full PageRank Algorithm

- There are two main differences between these pagerank algorithms
  - The delta ($\delta$) multiplied with $\frac{\text{rank}(u)}{|\text{outgoing}(u)|}$
  - The $\frac{(1-\delta)}{|V|}$ multiplied with $\text{rank}(u)$
- This helps us prevent concentration and artificial inflation
- $\delta$ is determined experimentally, and is commonly $\sim 0.8$
Full PageRank Algorithm

- In the real PageRank algorithm, $u$ does not give all its pagerank to its neighbors
  - Instead, it gives its neighbors only a $\delta$ fraction of its pagerank
  - It then distributes the remaining $(1 - \delta) \cdot \text{rank}(u)$ of its pagerank evenly among all vertices
- This guarantees that the algorithm can handle any type of directed graph - even like the one we saw!
- Runtime: $O(|V| + |E|)$
How do we know when to stop?

- Since a vertex’s pagerank is based on its incoming edges, when do we stop updating it?
- Does it ever stabilize?
- Yes! This is what $k$ represents - the number of iterations at which the rank stabilizes.
- We can use spectral analysis to prove that this stabilizes.
Spectral Analysis

- We use spectral analysis to analyze graphs using tools from linear algebra.
- This is a high level explanation of how the analysis is computed, since linear algebra is not a prerequisite for CS16.
- If you are interested in learning more about this, we have resources you can check out.
Spectral Analysis

- Eigenvectors and eigenvalues
  - An eigenvector of a matrix $M$, is a vector $\mathbf{v}$ such that there exists a scalar $\lambda > 0$ such that $M\mathbf{v} = \lambda \mathbf{v}$
  - Eigenvectors are interesting in our setting because they don’t change direction when they are multiplied by $M$
  - In fact, for any $k \geq 1$, $M^k \mathbf{v} = \lambda^k \mathbf{v}$
- We can represent our graph $G$ as an adjacency matrix
- Instead of storing a 1 representing an edge between vertices, we will store $1/|\text{outgoing}(v_i)|$
Spectral Analysis

- We can now represent our update step mathematically

- Let \( \mathbf{r} \) be an \( n \)-dimensional vector, where its \( i \)th coordinate \( r_i \) corresponds to vertex \( v_i \)'s pagerank

- Let \( N \) be our adjacency matrix

- For Basic PageRank, we can represent our update step for all vertices as:

\[
r'_i := r_1 \cdot N[1, i] + \ldots + r_n \cdot N[n, i]
\]
Spectral Analysis

- For real PageRank, it’s very similar, except we use a matrix \( P \) that stores
  - \( \frac{\delta}{\text{outgoing}(v_i)} + \frac{(1-\delta)}{n} \) if there is an edge
  - \( \frac{(1-\delta)}{n} \) if there is no edge

- Using \( P \), we can write the update step as:
  \[
  r' = P^T \cdot r
  \]

- Where \( P^T \) is the transpose of \( P \), a matrix whose rows are the columns of \( P \)
Spectral Analysis

- Suppose we start the algorithm with a vector of pageranks $r^{(0)}$ and apply the update step $k$ times.

- We end up with a vector of pageranks $r^k = (P^T)^k r^{(0)}$

- Mathematically, stabilizing means that for all $k$ larger than some threshold $t$, the vector $r^k = r^{k+1}$

- Doing some algebra, we get: $r^k = r^{k-1}$

\[
(P^T)^k \cdot r^{(0)} = (P^T)^{k-1} \cdot r^{(0)} \Rightarrow P^T \cdot (P^T)^{k-1} \cdot r^{(0)} = (P^T)^{k-1} \cdot r^{(0)}
\]

- If we let $v = (P^T)^{k-1} \cdot r^{(0)}$, we get $P^T v = v$

- Simply put, pageranks stabilize if $v$ is an eigenvector of $P^T$ with eigenvalue $\lambda = 1$, and this is true!
Downsides of PageRank

Activity #2

What are some possible downsides of PageRank?
Downsides of PageRank

- **Storage**
  - Need to store one copy of the entire web graph
  - Google is estimated to store about 50 billion web pages
  - Average size of a web page is 2 MB
  - …that’s an estimate of 100 petabytes (1 petabyte = $2^{50}$ bytes)!
  - Also hard to compute on such a large data set

- **Cost**
  - Need the funds to store that graph ($$$)
PageRank in Practice

- What happens when we actually perform a search?
- When you search, PageRank has already determined the importance of every webpage
  - Your search keyword narrows down the pages to return, and then they are ranked by their pagerank
- Keyword search could also prioritize titles or a Wikipedia page
PageRank in Practice

- PageRank is not only used for CS - many other fields use it!
  - Chemistry
  - Biology
  - Neuroscience
  - Engineering
  - Mathematics
  - Sports
  - Literature
  - Bibliometrics

<table>
<thead>
<tr>
<th>Div I - FCS</th>
<th>Team 1</th>
<th>beat</th>
<th>Team 2</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brown</td>
<td></td>
<td>Rhode Island</td>
<td>24 - 21</td>
</tr>
<tr>
<td>Div I - FCS</td>
<td>Rhode Island</td>
<td></td>
<td>Albany NY</td>
<td>31 - 14</td>
</tr>
<tr>
<td>Div I - FCS</td>
<td>Albany NY</td>
<td>beat</td>
<td>New Hampshire</td>
<td>15 - 0</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>New Hampshire</td>
<td></td>
<td>Ga Southern</td>
<td>22 - 12</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>Ga Southern</td>
<td></td>
<td>South Alabama</td>
<td>52 - 0</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>South Alabama</td>
<td>beat</td>
<td>Troy</td>
<td>19 - 8</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>Troy</td>
<td></td>
<td>LSU</td>
<td>24 - 21</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>LSU</td>
<td>beat</td>
<td>Auburn</td>
<td>27 - 23</td>
</tr>
<tr>
<td>Div I - FBS</td>
<td>Auburn</td>
<td></td>
<td>Alabama</td>
<td>26 - 14</td>
</tr>
</tbody>
</table>

Therefore, Brown is better than Alabama in 2017.
Thank you!
Sources

- Wikipedia
- http://www.worldwidewebsize.com