Announcements

- Final Project implementation plan due 24 hours earlier than “normal”
- The implementation plan is meant to help you! Do not stress about format/expected length/etc
- (Short) homework for next Monday posted soon
- Preregistration is happening…let us know if you have question about CS

Last Time

- Tom told you about server programming

Today’s Plan

- Indexing the web: revisit web crawling
- Ranking the web: take a closer look at Google’s PageRank algorithm
Indexing the Web

- Recall our web crawling lab
  - You (recursively) followed href links on web pages to find other pages
  - Determine the number of incoming links and outgoing links for a page
- Google does this constantly!
- Create an inverted index that maps words to the webpages that contain those words

Determining Relevance

- Google's incredible success largely came from its ability to return highly relevant results to searches
- How do you determine the relevance of a web page?

Hyperlink Trick

- The hyperlink trick determines how "authoritative" a page is based on the number of incoming links
- Bert's page is more relevant than Ernie's
- What about negative reviews? (They don't have a big impact in practice.)

Authority Trick

- Are all webpages created equal?
- No! Experts have more "authority" and their opinions are more relevant than non-experts' opinions. We should take that into account when ranking search results, too...

Image source: "The Ingenious Ideas that Drive Today's Computers"
Determining Authority

How do you know if someone is an expert?

Calculate the number of incoming links to their webpage

Solution: Random Surfer

Imagine a person randomly surfing the web

Pick a start page at random from entire WWW

Examine all (outgoing) links on page

Pick one at random and click on it

Continue process

There is also a “restart probability” on each page, which is the prob that the surfers gets bored and starts over (we’ll use 15%)

Handling Cycles

Hard to calculate the authority score on pages whose links form a cycle!
Example

Start on page A. Follow links (shaded pages). Restarts are shown by dashed lines.

Example

Numbers of times each page was visited after 1000 simulated steps

Example

Percentage of visits to each page after 1,000,000 simulated steps

- Incorporates both hyperlink (page D) and authority tricks (pages A vs C)!
- Quality and quantity of incoming links matter
- Cycles are no longer a problem

Google’s PageRank

- Compute a measure $R(v)$ for every web page $v$
- $R(v)$ should reflect importance of pages that link to $v$
Out Degree

\[ C(1) = 3 \]
\[ C(2) = 2 \]
\[ C(3) = 1 \]
\[ C(4) = 2 \]
In Degree

\[ \text{In}(1) = 2 \]
\[ \text{In}(2) = 1 \]
\[ \text{In}(3) = 3 \]
\[ \text{In}(4) = 2 \]

PageRank: Parameters

- Parameters of web graph \( G \)
  - \( N \) & \( L \): Number of nodes/vertices & links/edges in \( G \)
  - \( C(v) \): out-degree of \( v \) (number of links from \( v \))
  - \( R(v) \): the rank of \( v \) (to be computed)

Big idea
- Google Juice = liquid rank

PageRank: Google Juice

- Ranking as (fluid) flow in a network
- Each page shares its importance with pages it links to
  - Page \( u \) gives each neighbor \( R(u)/C(u) \) of its importance
  - So each page gets importance from pages that link to it
  - If \( u_1, ..., u_{\text{In}(v)} \) are pages linking to page \( v \)
    - then \( R(v) = R(u_1)/C(u_1) + ... + R(u_{\text{In}(v)})/C(u_{\text{In}(v)}) \)
PageRank: Iterated Rankings

Goal: Find a ranking satisfying
\[ R(v) = \frac{R(u_1)}{C(u_1)} + ... + \frac{R(u_{In(v)})}{C(u_{In(v)})} \]

The Algorithm:
- Find an initial ranking: For example, \( R_0(v) = \frac{In(v)}{L} \)
- Let “Google Juice” flow to give new ranking
  \[ R_1(v) = \frac{R_0(u_1)}{C(u_1)} + ... + \frac{R_0(u_{In(v)})}{C(u_{In(v)})} \]
- Repeat many times to get rankings \( R_2, R_3, R_4, ... \)
- Stop when \( R_n \) is not much different from \( R_{n-1} \)

Computing Rank Functions \( R_n() \)

PageRank: Amazing Result

- On any “reasonably structured” graph, this method will converge!
- Reasonably structured:
  - For every pair of vertices \{u,v\} there is a directed path from u to v and one from v to u. [G is strongly connected]
  - Not all cycle-lengths are multiples of a common value \( k > 1 \) [G is aperiodic]
PageRank as Random Walk

- Think of $R_0$ as a probability distribution
- $R_0(v)$: probability of starting at v (or)
- $R_0(v)$: probability of being at v after 0 steps

Random Walks on Graphs

- Similarly, $R_i(j)$ is the probability of being at page j after exactly i clicks (given starting distribution $R_0$)
- Rename $R_i()$ to be $Pr_i()$ to emphasize this fact

Random Surfer (revisited)

$Pr_i(j) = \text{prob. at page j after i clicks}$

Random Surfer

$Pr_{i+1}(2) = Pr_i(1)/3$
Random Surfer

$Pr_{i+1}(1) = ?$

$Pr_{i+1}(1) = Pr_i(3) + Pr_i(4)/2$

Rather than $P_0(v) = \ln(v)/L$, assume we are equally likely to start on any of our 4 pages.

<table>
<thead>
<tr>
<th></th>
<th>$P_0$</th>
<th>$P_1$</th>
<th>$P_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1/4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1/4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[ \begin{align*}
\text{Pr}_{i+1}(1) &= \text{Pr}_i(3) + \text{Pr}_i(4)/2 \\
\text{Pr}_{i+1}(2) &= \text{Pr}_i(1)/3 \\
\text{Pr}_{i+1}(3) &= \text{Pr}_i(1)/3 + \text{Pr}_i(2)/2 + \text{Pr}_i(4)/2 \\
\text{Pr}_{i+1}(4) &= \text{Pr}_i(1)/3 + \text{Pr}_i(2)/2
\end{align*} \]
What is Happening?

- The distributions $Pr_i()$ converge to a probability distribution $Pr_\infty()$
- And it's the same regardless of starting distribution $Pr_0$
- $Pr_\infty()$ depends only on the structure of graph $G$
- How can we think about $Pr_\infty()$?
Understanding $\Pr_\infty()$

- $\Pr_\infty(v)$ is the probability of **eventually** being at vertex $v$ after some **very long** random walk through the web graph, starting from a randomly selected vertex.
- $\Pr_\infty(v) = \sum_u \Pr_\infty(u)/C(u)$ summing over all $u \to v$.
- $\Pr_\infty()$ is called an **equilibrium distribution** for $G$.
- If $G$ is “properly structured”, $\Pr_\infty()$ exists and is unique!

<table>
<thead>
<tr>
<th></th>
<th>$P_0$</th>
<th>$P_1$</th>
<th>$P_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/4</td>
<td>0.38</td>
<td>0.44</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>0.08</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>1/4</td>
<td>0.33</td>
<td>0.27</td>
</tr>
<tr>
<td>4</td>
<td>1/4</td>
<td>0.21</td>
<td>0.17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$P_0$</th>
<th>$P_1$</th>
<th>$P_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/4</td>
<td>0.38</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>1/4</td>
<td>0.33</td>
<td>0.46</td>
</tr>
<tr>
<td>3</td>
<td>1/4</td>
<td>0.21</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>1/4</td>
<td>0.08</td>
<td>0.13</td>
</tr>
</tbody>
</table>

What Could Go Wrong?

- The web graph is not strongly connected.
- There are pages with no links (sink).
- There are groups of pages with no links leaving the group (connected component).
Avoiding Traps

- The web graph is not strongly connected
- There are pages with no links (sink)
- There are groups of pages with no links leaving the group (connected component)
- What can we do?

Adjust probabilities to allow for random page jumping

Let $E(v)$ be a probability distribution

Idea: $E(v) = \text{probability that user randomly jumped to page } v \text{ from some other page (this is just the restart probability!)}$

Food for Thought: Ethics of Google

- Socio-economic implications of rankings?
- Some details of PageRank intentionally left out...why?
- Cheating the system?
- Regulating Google?