Web and PageRank
Lecture 4

CSCI 4974/6971

12 Sep 2016
Today’s Biz

1. Review MPI
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
Today’s Biz

1. **Review MPI**
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
MPI Review

- Basic functions
  - `MPI_Init(&argc, &argv)`
  - `MPI_Comm_rank(MPI_COMM_WORLD, &rank)`
  - `MPI_Comm_size(MPI_COMM_WORLD, &size)`
  - `MPI_Finalize()`
  - `MPI_Barrier(MPI_COMM_WORLD)`

- Point to point communication
  - `MPI_Send(sbuf, count, MPI_TYPE, to, tag, MPI_COMM_WORLD)`
  - `MPI_Recv(rbuf, count, MPI_TYPE, from, tag, MPI_COMM_WORLD)`

- Reductions
  - `MPI_Reduce(sbuf, rbuf, count, MPI_TYPE, MPI_OP, MPI_COMM_WORLD)`
  - `MPI_Allreduce(sbuf, rbuf, count, MPI_TYPE, MPI_OP, root, MPI_COMM_WORLD)`
Today’s Biz

1. Review MPI
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
Reminders

▶ Assignment 1: Monday 19 Sept 16:00
▶ Project Proposal: Thursday 22 Sept 16:00
▶ Office hours: Tuesday & Wednesday 14:00-16:00 Lally 317
  ▶ Or email me for other availability
▶ Class schedule (for next month):
  ▶ Web analysis methods
  ▶ Social net analysis methods
  ▶ Bio net analysis methods
  ▶ Random networks and usage
Today’s Biz

1. Review MPI
2. Reminders
3. **Structure of the web**
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
Structure of the Web

Slides from Jure Leskovec and Anand Rajaraman, Stanford University
Webgraph structure and PageRank
Two More Datasets Available

- TheFind.com
  - Large set of products (~6GB compressed)
  - For each product
    - Attributes
    - Related products
- Craigslist
  - About 3 weeks of data (~7.5GB compressed)
    - Text of posts, plus category metadata
    - e.g., match buyers and sellers
How big is the Web?

- Technically, infinite
- Much duplication (30-40%)
- Best estimate of “unique” static HTML pages comes from search engine claims
  - Google = 8 billion(?), Yahoo = 20 billion

What is the structure of the Web? How is it organized?
Web as a Graph

I teach a class on Networks.

Networks Course:
We have a class blog

Networks Class Blog:
This blog post is about Microsoft

Microsoft Home Page
Web as a Graph

- In early days of the Web links were navigational
- Today many links are transactional
Directed graphs

- Two types of directed graphs:
  - DAG – directed acyclic graph:
    - Has no cycles: if u can reach v, then v can not reach u
  - Strongly connected:
    - Any node can reach any node via a directed path

- Any directed graph can be expressed in terms of these two types
Strongly connected component (SCC) is a set of nodes $S$ so that:

- Every pair of nodes in $S$ can reach each other
- There is no larger set containing $S$ with this property
Graph structure of the Web

- Take a large snapshot of the web and try to understand how it’s SCCs “fit” as a DAG.

- **Computational issues:**
  - Say want to find SCC containing specific node $v$?
  - Observation:
    - $\text{Out}(v)$ ... nodes that can be reachable from $v$ (BFS out)
    - SCC containing $v$:
      - $= \text{Out}(v, G) \cap \text{In}(v, G)$
      - $= \text{Out}(v, G) \cap \text{Out}(v, \overline{G})$
    - where $\overline{G}$ is $G$ with directions of all edge flipped
Graph structure of the Web

- There is a giant SCC
- Broder et al., 2000:
  - Giant weakly connected component: 90% of the nodes
Bow-tie structure of the Web

- 250 million webpages, 1.5 billion links [Altavista]

[Broder et al., ‘00]
Diameter of the Web

- Diameter (average directed shortest path length) is 19 (in 1999)

[Albert et al., '99]
Diameter of the Web

- Average distance:
  75% of time there is no directed path from start to finish page
  - Follow in-links (directed): 16.12
  - Follow out-links (directed): 16.18
  - Undirected: 6.83

- Diameter of SCC (directed):
  - At least 28

[Broder et al., '00]
Degree distribution on the Web

[Broder et al., ’00]
Degrees in real networks

- Take real network plot a histogram of $p_k$ vs. $k$
Degrees in real networks (2)

- Plot the same data on log-log axis:

\[ p_k = \beta k^{-\alpha} \]

\[ \log p_k = \log \beta - \alpha \log k \]
Exponential tail vs. Power-law tail

Power law:

\[ Y \sim X^{-2} \]

Exponential

\[ Y \sim e^{-X} \]
Power law degree exponents

- Power law degree exponent is typically $2 < \alpha < 3$
  - Web graph [Broder et al. 00]:
    - $\alpha_{in} = 2.1$, $\alpha_{out} = 2.4$
  - Autonomous systems [Faloutsos et al. 99]:
    - $\alpha = 2.4$
  - Actor collaborations [Barabasi-Albert 00]:
    - $\alpha = 2.3$
  - Citations to papers [Redner 98]:
    - $\alpha \approx 3$
  - Online social networks [Leskovec et al. 07]:
    - $\alpha \approx 2$
**Power-law network**

Random network
(Erdos-Renyi random graph)

Degree distribution is Binomial

Scale-free (power-law) network

Degree distribution is Power-law

Function is scale free if:
\[ f(ax) = c f(x) \]
Structure of the Web – Revisited

Slides from Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, Christian Bizer, Universität Mannheim
Graph Structure in the Web
Revisited

Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, Christian Bizer
Textbook Knowledge about the Web Graph

- used two AltaVista crawls (200 million pages, 1.5 billion links)
- Results

Power Laws

![In-degree (total, remote-only) distr.](image)

- Total in-degree
- Power law, exponent 2.09
- Remote-only in-degree
- Power law, exponent 2.1

Bow-Tie

![Diagram of Bow-Tie structure](image)
This talk will:

1. Show that the textbook knowledge might be wrong or dependent on crawling process.

2. Provide you with a large recent Web graph to do further research.
Outline

1. Public Web Crawls
2. The Web Data Commons Hyperlink Graph
3. Analysis of the Graph
   1. In-degree & Out-degree Distributions
   2. Node Centrality
   3. Strong Components
   4. Bow Tie
   5. Reachability and Average Shortest Path
4. Conclusion
Public Web Crawls

1. AltaVista Crawl distributed by Yahoo! WebScope 2002
   • Size: 1.4 billion pages
   • Problem: Largest strongly connected component 4%

2. ClueWeb 2009
   • Size: 1 billion pages
   • Problem: Largest strongly connected component 3%

3. ClueWeb 2012
   • Size: 733 million pages
   • Largest strongly connected component 76%
   • Problem: Only English pages
Common Crawl is a non-profit foundation dedicated to building and maintaining an open crawl of the web, thereby enabling a new wave of innovation, education and research.
The Common Crawl Foundation

- Regularly publishes Web crawls on Amazon S3.
- Five crawls available so far:

<table>
<thead>
<tr>
<th>Date</th>
<th># Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>2.5 billion</td>
</tr>
<tr>
<td>Spring 2012</td>
<td>3.5 billion</td>
</tr>
<tr>
<td>Spring 2013</td>
<td>2.0 billion</td>
</tr>
<tr>
<td>Winter 2013</td>
<td>2.0 billion</td>
</tr>
<tr>
<td>Spring 2014</td>
<td>2.5 billion</td>
</tr>
</tbody>
</table>

- Crawling Strategy
  - breadth-first visiting strategy
  - at least 71 million seeds from previous crawls and from Wikipedia
Web Data Commons – Hyperlink Graph

- extracted from the Spring 2012 version of the Common Crawl
- size

3.5 billion nodes

128 billion arcs

- pages originate from 43 million pay-level domains (PLDs)
  - 240 million PLDs were registered in 2012 * (18%)
- world-wide coverage

Downloading the WDC Hyperlink Graph

- http://webdatacommons.org/hyperlinkgraph/

- 4 aggregation levels:

<table>
<thead>
<tr>
<th>Graph</th>
<th>#Nodes</th>
<th>#Arcs</th>
<th>Size (zipped)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page graph</td>
<td>3.56 billion</td>
<td>128.73 billion</td>
<td>376 GB</td>
</tr>
<tr>
<td>Subdomain graph</td>
<td>101 million</td>
<td>2,043 million</td>
<td>10 GB</td>
</tr>
<tr>
<td>1st level subdomain graph</td>
<td>95 million</td>
<td>1,937 million</td>
<td>9.5 GB</td>
</tr>
<tr>
<td>PLD graph</td>
<td>43 million</td>
<td>623 million</td>
<td>3.1 GB</td>
</tr>
</tbody>
</table>

- Extraction code is published under Apache License
  - Extraction costs per run: ~ 200 US$ in Amazon EC2 fees
Analysis of the Graph
In-Degree Distribution

Broder et al. (2000)
Power law with exponent 2.1

WDC Hyperlink Graph (2012)
Best power law exponent 2.24
In-Degree Distribution

- Power law fitted using \texttt{plfit-tool}.
- Maximum likelihood fitting.
- Starting degree: 1129
- Best power law exponent: 2.24
Goodness of Fit Test

- Method
  - $p$-value < 0.1 $\Rightarrow$ power law not a plausible hypothesis

- Goodness of fit result
  - $p$-value = 0

- Conclusions:
  - in-degree does not follow power law
  - in-degree has non-fat heavy-tailed distribution
  - maybe log-normal?
Out-Degree Distribution

Broder et al.: Power law exponent 2.78

WDC: Best power law exponent 2.77

p-value = 0
# Node Centrality

http://wwwranking.webdatacommons.org

<table>
<thead>
<tr>
<th>Harmonic centrality</th>
<th>Indegree centrality</th>
<th>Katz's index</th>
<th>PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. youtube.com</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2. en.wikipedia.org</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>3. twitter.com</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>4. google.com</td>
<td>7</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>5. wordpress.org</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6. flickr.com</td>
<td>8</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>7. facebook.com</td>
<td>19</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>8. apple.com</td>
<td>44</td>
<td>35</td>
<td>31</td>
</tr>
<tr>
<td>9. vimeo.com</td>
<td>17</td>
<td>17</td>
<td>27</td>
</tr>
<tr>
<td>10. creativecommons.org</td>
<td>16</td>
<td>13</td>
<td>20</td>
</tr>
</tbody>
</table>

1 - 10 of 101717775 items 10 Per Page  
Page 1 of 10171778
Average Degree

Broder et al. 2000: \(7.5\)

WDC 2012: \(36.8\)

\(\Rightarrow\) Factor 4.9 larger

Possible explanation: HTML templates of CMS
Strongly Connected Components

Calculated using WebGraph framework on a machine with 1 TB RAM.

Largest SCC
- Broder: 27.7%
- WDC: 51.3 %

\( \Rightarrow \) Factor 1.8 larger
The Bow-Tie Structure of Broder et al. 2000

- Balanced size of IN and OUT: 21%
- Size of LSCC: 27%
The Bow-Tie Structure of WDC Hyperlinkgraph 2012

- IN much larger than OUT: 31% vs. 6%
- LSCC much larger: 51%
The Chinese web looks like a tea-pot.
Reachability and Average Shortest Path

Broder et al. 2000
- Pairs of pages connected by path: 25%
- Average shortest path: 16.12

WDC Webgraph 2012
- Pairs of pages connected by path: 48%
- Average shortest path: 12.84
Conclusions

1. Web has become more dense and more connected
   - Average degree has grown significantly in last 13 years (factor 5)
   - Connectivity between pairs of pages has doubled

2. Macroscopic structure
   - There is large SCC of growing size.
   - The shape of the bow-tie seems to depend on the crawl

3. In- and out-degree distributions do not follow power laws.
Today’s Biz

1. Review MPI
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
PageRank Centrality

Slides from Fei Li, University of Michigan
The PageRank Citation Ranking: Bring Order to the web

- Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd

- Presented by Fei Li
Motivation and Introduction

Why is Page Importance Rating important?
• Huge number of web pages: 150 million by 1998
  1000 billion by 2008
• Diversity of web pages: different topics, different quality, etc.

What is PageRank?
• A method for rating the importance of web pages objectively and mechanically using the link structure of the web.
The History of PageRank

PageRank was developed by Larry Page (hence the name *Page-Rank*) and Sergey Brin.

It is first as part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.

Shortly after, Page and Brin founded Google.

16 billion…
Recent News

There are some news about that PageRank will be canceled by Google.

There are large numbers of Search Engine Optimization (SEO).

SEO use different trick methods to make a web page more important under the rating of PageRank.
Link Structure of the Web

150 million web pages $\rightarrow$ 1.7 billion links

Backlinks and Forward links:
- A and B are C’s backlinks
- C is A and B’s forward link

Intuitively, a webpage is important if it has a lot of backlinks.

What if a webpage has only one link to www.yahoo.com?
A Simple Version of PageRank

\[ R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} \]

- \( u \): a web page
- \( B_u \): the set of \( u \)'s backlinks
- \( N_v \): the number of forward links of page \( v \)
- \( c \): the normalization factor to make \( \|R\|_{L1} = 1 \) (\( \|R\|_{L1} = |R_1 + \ldots + R_n| \))
An example of Simplified PageRank

PageRank Calculation: first iteration

\[ \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \]

PageRank Calculation: rst iteration
An example of Simplified PageRank

PageRank Calculation: second iteration

PageRank Calculation: second iteration
An example of Simplified PageRank

Convergence after some iterations
A Problem with Simplified PageRank

A loop:

During each iteration, the loop accumulates rank but never distributes rank to other pages!
An example of the Problem

$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$
An example of the Problem

\[ M = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix} \]

\[
\begin{bmatrix}
\text{yahoo} \\
\text{Amazon} \\
\text{Microsoft}
\end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}
\]

\[
\begin{bmatrix} \frac{1}{4} \\ \frac{1}{6} \\ \frac{7}{12} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{3} \\ \frac{1}{6} \\ \frac{1}{2} \end{bmatrix}
\]
An example of the Problem

\[ M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \]

\[ \begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \]

\[
\begin{bmatrix}
5/24 & 1/6 & ... & 0 \\
1/8 & 5/48 & ... & 0 \\
2/3 & 35/48 & ... & 1
\end{bmatrix}
\]
Random Walks in Graphs

- The Random Surfer Model
  - The simplified model: the standing probability distribution of a random walk on the graph of the web. simply keeps clicking successive links at random

- The Modified Model
  - The modified model: the “random surfer” simply keeps clicking successive links at random, but periodically “gets bored” and jumps to a random page based on the distribution of E
Modified Version of PageRank

\[ R'(u) = c_1 \sum_{v \in B_u} \frac{R'(v)}{N_v} + c_2 E(u) \]

E(u): a distribution of ranks of web pages that “users” jump to when they “gets bored” after successive links at random.
An example of Modified PageRank

\[
M = \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 \\
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & 1 
\end{bmatrix}
\]

\[
\begin{bmatrix}
yahoo \\
Amazon \\
Microsoft 
\end{bmatrix} = \begin{bmatrix}
\frac{1}{3} \\
\frac{1}{3} \\
\frac{1}{3} 
\end{bmatrix}
\]

\[
C_1 = 0.8 \quad C_2 = 0.2
\]

\[
\begin{bmatrix}
0.333 & 0.333 & 0.280 & 0.259 & 7/33 \\
0.333 & 0.200 & 0.200 & 0.179 & 5/33 \\
0.333 & 0.467 & 0.520 & 0.563 & 21/33 
\end{bmatrix}
\]
Dangling Links

- Links that point to any page with no outgoing links
- Most are pages that have not been downloaded yet
- Affect the model since it is not clear where their weight should be distributed
- Do not affect the ranking of any other page directly
- Can be simply removed before pagerank calculation and added back afterwards
PageRank Implementation

- Convert each URL into a unique integer and store each hyperlink in a database using the integer IDs to identify pages
- Sort the link structure by ID
- Remove all the dangling links from the database
- Make an initial assignment of ranks and start iteration
  - Choosing a good initial assignment can speed up the pagerank
- Adding the dangling links back.
Convergence Property

- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Scaling factor is roughly linear in $\log n$
Convergence Property

The Web is an expander-like graph

- Theory of random walk: a random walk on a graph is said to be rapidly-mixing if it quickly converges to a limiting distribution on the set of nodes in the graph. A random walk is rapidly-mixing on a graph if and only if the graph is an expander graph.

- Expander graph: every subset of nodes S has a neighborhood (set of vertices accessible via outedges emanating from nodes in S) that is larger than some factor $\alpha$ times of $|S|$. A graph has a good expansion factor if and only if the largest eigenvalue is sufficiently larger than the second-largest eigenvalue.
Today’s Biz

1. Review MPI
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. Parallel Pagerank Tutorial
More MPI

Slides from David Cronk, University of Tennessee
MPI_Allgather (sbuf, scount, stype, rbuf, rcount, rtype, comm, ierr)

All arguments are meaningful at every process.

Data from sbuf at all processes in group A is concatenated in rank order and the result is stored at rbuf of every process in group B and vice-versa.

Send arguments in A must be consistent with receive arguments in B, and vice-versa.
MPI_Alltoall (sbuff, scount, stype, rbuf, rcount, rtype, comm, ierr)

Result is as if each process in group A scatters its \textit{sbuff} to each process in group B and each process in group B scatters its \textit{sbuff} to each process in group A.

Data is gathered in \textit{rbuf} in rank order according to the rank in the group providing the data.

Each process in group A sends the same amount of data to group B and vice-versa.
Today’s Biz

1. Review MPI
2. Reminders
3. Structure of the web
4. PageRank Centrality
5. More MPI
6. **Parallel Pagerank Tutorial**
Parallel Pagerank Tutorial

1. Serial
2. OpenMP
3. MPI
4. More advanced (if time)
Parallel PageRank Tutorial
Blank code and data available on website
www.cs.rpi.edu/~slotag/classes/FA16/index.html