Data Mining Techniques

CS 6220 - Section 3 - Fall 2016

Lecture 17: Link Analysis

Jan-Willem van de Meent
(credit: Yijun Zhao, Yi Wang, Tan et al., Leskovec et al.)
Graph Data: Media Networks

Connections between political blogs
Polarization of the network [Adamic-Glance, 2005]

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>26 Oct Midterm exam</td>
</tr>
<tr>
<td></td>
<td>28 Oct Project Proposal presentations</td>
</tr>
<tr>
<td>9</td>
<td>04 Nov <strong>Frequent Pattern Mining 1: Apriori</strong></td>
</tr>
<tr>
<td></td>
<td>07 Nov <strong>Frequent Pattern Mining 2: PCY, FP-Growth</strong></td>
</tr>
<tr>
<td>10</td>
<td>09 Nov Link Analysis: Page-rank, Trust-rank</td>
</tr>
<tr>
<td></td>
<td>11 Nov (Veteran's Day)</td>
</tr>
<tr>
<td>11</td>
<td>16 Nov Time Series: Hidden Markov Models</td>
</tr>
<tr>
<td></td>
<td>18 Nov Community Detection: Betweenness, Spectral Clustering</td>
</tr>
<tr>
<td>12</td>
<td>23 Nov (Thanksgiving Holiday)</td>
</tr>
<tr>
<td></td>
<td>25 Nov (Thanksgiving Holiday)</td>
</tr>
<tr>
<td>13</td>
<td>30 Nov Bonus Topic: Deep Learning</td>
</tr>
<tr>
<td></td>
<td>02 Dec Project Presentations</td>
</tr>
<tr>
<td>14</td>
<td>07 Dec (Review)</td>
</tr>
<tr>
<td></td>
<td>09 Dec (Review)</td>
</tr>
<tr>
<td>15</td>
<td>14 Dec Final Exam</td>
</tr>
<tr>
<td>16</td>
<td>19 Dec (Final grades posted)</td>
</tr>
</tbody>
</table>

**Proposals due:**
- HKP: 6; HTF: 14; Aggarwal: 4.5; TSK: 6
- HKP: 6; HTF: 14; Aggarwal: 4.5; TSK: 6
- LRU: 5; Aggarwal: 18.4

**Reports due:**
- Bishop: 13.1-2; HKP: 13.1.1
- LRU: 10
Web search before PageRank

- Human-curated (e.g. Yahoo, Looksmart)
- Hand-written descriptions
- Wait time for inclusion
- Text-search (e.g. WebCrawler, Lycos)
- Prone to term-spam

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Web as a Directed Graph

(adapted from:: Mining of Massive Datasets, http://www.mmds.org)
PageRank: Links as Votes

Not all pages are equally important

- Pages with **more inbound links** are more **important**
- Inbound **links from important pages** carry more weight

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Example: PageRank Scores

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Simple Recursive Formulation

- A link’s vote is proportional to the importance of its source page
- If page $j$ with importance $r_j$ has $n$ out-links, each link gets $r_j/n$ votes
- Page $j$’s own importance is the sum of the votes on its in-links

$r_j = r_i/3 + r_k/4$

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Equivalent Formulation: Random Surfer

- At time $t$ a surfer is on some page $i$
- At time $t+1$ the surfer follows a link to a new page at random
- Define rank $r_i$ as fraction of time spent on page $i$

$r_j = r_i/3 + r_k/4$

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
PageRank: The “Flow” Model

- 3 equations, 3 unknowns
- Impose constraint: \( r_y + r_a + r_m = 1 \)
- Solution: \( r_y = 2/5, \ r_a = 2/5, \ r_m = 1/5 \)

“Flow” equations:

\[
\begin{align*}
  r_j &= \sum_{i \rightarrow j} \frac{r_i}{d_i} \\
  r_y &= \frac{r_y}{2} + \frac{r_a}{2} \\
  r_a &= \frac{r_y}{2} + r_m \\
  r_m &= \frac{r_a}{2}
\end{align*}
\]

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
PageRank: The “Flow” Model

Matrix $M$ is stochastic (i.e. columns sum to one)
PageRank: Eigenvector Problem

• PageRank: Solve for eigenvector \( r = M r \)
  with eigenvalue \( \lambda = 1 \)

• Eigenvector with \( \lambda = 1 \) is guaranteed to exist since \( M \) is a stochastic matrix (i.e. if \( a = M b \) then \( \Sigma a_i = \Sigma b_i \))

• Problem: There are billions of pages on the internet. How do we solve for eigenvector with order \( 10^{10} \) elements?
PageRank: Power Iteration

Model for random Surfer:

- At time $t = 0$ pick a page at random
- At each subsequent time $t$ follow an outgoing link at random

Probabilistic interpretation:

\[
p(z_0 = i) = \frac{1}{N}
\]

\[
p(z_t = i \mid z_{t-1} = j) = M_{ij}
\]

\[
p(z_t = i) = \sum_j p(z_t = i, z_{t-1} = j)
\]

\[
= \sum_j M_{ij} p(z_{t-1} = j)
\]
PageRank: Power Iteration

\[ p^t = M p^{t-1} = M^t p^0 \]

\[
p^0 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} \quad M = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}
\]

\[
p^t = \begin{bmatrix} \frac{1}{3} \\ 2/6 \\ \frac{1}{3} \end{bmatrix} \begin{bmatrix} 2/6 \\ \frac{1}{3} \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 4/12 \\ 3/12 \end{bmatrix} \begin{bmatrix} 9/24 \\ 11/24 \\ 4/24 \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{bmatrix} \approx \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}
\]

\[ p^t \text{ converges to } r. \text{ Iterate until } |p^t - p^{t-1}| < \varepsilon \]

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Aside: Ergodicity

- PageRank is assumes a *random walk* model for individual surfers
- *Equivalent assumption*: flow model in which equal fractions of surfers follow each link at every time
- *Ergodicity*: The equilibrium of the flow model is the same as the asymptotic distribution for an individual random walk

\[
\begin{align*}
  r &= Mr \\
p^t &= Mp^{t-1} \\
  \lim_{t \to \infty} p^t &= r
\end{align*}
\]
Aside: Ergodicity

• PageRank is assumes a *random walk* model for individual surfers

• *Equivalent assumption*: flow model in which equal fractions of surfers follow each link at every time

• *Ergodicity*: The equilibrium of the flow model is the same as the asymptotic distribution for an individual random walk

\[
p(z_t = i) = \sum_j M_{ij} p(z_{t-1} = j)
\]

\[
\lim_{T \to \infty} \mathbb{E} \left[ \frac{1}{T} \sum_{t=1}^{T} I[z_t = i] \right] = r_i
\]
Aside: Ergodicity

- PageRank is assumes a random walk model for individual surfers
- *Equivalent assumption*: flow model in which equal fractions of surfers follow each link at every time
- *Ergodicity*: The equilibrium of the flow model is the same as the asymptotic distribution for an individual random walk

Averaging over individuals is equivalent to averaging single individual over time
PageRank: Problems

1. **Dead Ends**
   - Nodes with no outgoing links.
   - Where do surfers go next?

2. **Spider Traps**
   - Subgraph with no outgoing links to wider graph
   - Surfers are “trapped” with no way out.

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Dead Ends

\[ p^t = M p^{t-1} = M^t p^0 \]

\[
p^0 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix}
\]

\[
p^t = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 2/6 \\ 1/6 \\ 1/6 \end{bmatrix} \begin{bmatrix} 3/12 \\ 1/12 \\ 1/12 \end{bmatrix} \ldots \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
\]

Probability not conserved

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Dead Ends

\[ p^t = M p^{t-1} = M^t p^0 \]

\[
p^0 = \begin{bmatrix}
1/3 \\
1/3 \\
1/3
\end{bmatrix}
\]

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

(teleport at dead ends)

\[
p^t = \begin{bmatrix}
1/3 & 8/18 & 49/108 \\
1/3 & 5/18 & 34/108 \\
1/3 & 5/18 & 35/108
\end{bmatrix}
\]

Fixes “probability sink” issue

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Spider Traps

\[ p^t = M p^{t-1} = M^t p^0 \]

\[
p^0 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}
\]

\[
p^t = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 2/6 \\ 1/6 \\ 3/6 \end{bmatrix} \begin{bmatrix} 3/12 \\ 2/12 \\ 7/12 \end{bmatrix} \begin{bmatrix} 5/24 \\ 3/24 \\ 16/24 \end{bmatrix} \ldots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]

Probability accumulates in traps (surfers get stuck)

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Solution: Random Teleports

Model for teleporting random surfer:

- At time $t = 0$ pick a page at random
- At each subsequent time $t$
  - With probability $\beta$ follow an outgoing link at random
  - With probability $1-\beta$ teleport to a new initial location at random

PageRank Equation [Page & Brin 1998]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1 - \beta) \frac{1}{N}$$

(adapted from:: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Teleports

\[ p^t = \beta M p^{t-1} + (1 - \beta)p^0 = \tilde{M} p^{t-1} \]

\[ \tilde{M} = \beta M + (1 - \beta) \begin{bmatrix} - & p_1^0 & - \\ - & \cdots & - \\ - & p_N^0 & - \end{bmatrix} \]

(can use power iteration as normal)

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Teleports

\[
p^t = \beta M p^{t-1} + (1 - \beta) p^0 = \tilde{M} p^{t-1}
\]

\[
\tilde{M} = \beta M + (1 - \beta) \\
\begin{bmatrix}
- & p_1^0 & - \\
- & \cdots & - \\
- & p_N^0 & -
\end{bmatrix}
\]

(can use power iteration as normal)

\[
\tilde{M} = \frac{4}{5} \begin{bmatrix}
\frac{1}{2} & \frac{1}{2} & 0 \\
\frac{1}{2} & 0 & 0 \\
0 & \frac{1}{2} & 1
\end{bmatrix} + \frac{1}{5} \begin{bmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{bmatrix} = \begin{bmatrix}
\frac{7}{15} & \frac{7}{15} & \frac{1}{15} \\
\frac{7}{15} & \frac{7}{15} & \frac{1}{15} \\
\frac{1}{15} & \frac{1}{15} & \frac{1}{15}
\end{bmatrix}
\]

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Power Iteration: Teleports

\[ p^t = \tilde{M} p^{t-1} = \tilde{M}^t p^0 \]

\[ p^0 = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad \tilde{M} = \begin{bmatrix} \frac{7}{15} & \frac{7}{15} & \frac{1}{15} \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} \\ \frac{1}{15} & \frac{1}{15} & \frac{1}{15} \end{bmatrix} \]

(can use power iteration as normal)

\[ p^t = \begin{bmatrix} 1/3 & 0.33 & 0.24 & \ldots & \end{bmatrix} \begin{bmatrix} 0.20 \\ 0.46 \end{bmatrix} = \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix} \]

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Computing PageRank

\[ p^t = \beta M p^t + \frac{1 - \beta}{N} \]

- $M$ is sparse - only store nonzero entries
- Space proportional roughly to number of links
- Say 10N, or $4 \times 10 \times 1$ billion = 40GB
- Still won’t fit in memory, but will fit on disk

<table>
<thead>
<tr>
<th>source node</th>
<th>degree</th>
<th>destination nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1, 5, 7</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>17, 64, 113, 117, 245</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>13, 23</td>
</tr>
</tbody>
</table>

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Block-based Update Algorithm

- Break $r^{\text{new}}$ into $k$ blocks that fit in memory
- Scan $M$ and $r^{\text{old}}$ once for each block

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Break M into stripes: Each stripe contains only destination nodes in the corresponding block of \( r^{\text{new}} \)

(adapted from: Mining of Massive Datasets, [http://www.mmds.org](http://www.mmds.org))
First Spammers: Term Spam

How do you make your page appear to be about movies?

(1) Add the word movie 1,000 times to your page.
Set text color to the background color, so only search engines would see it.

(2) Or, run the query “movie” on your target search engine.
See what page came first in the listings.
Copy it into your page, make it “invisible”.

These and similar techniques are term spam.

(adapted from:: Mining of Massive Datasets, http://www.mmds.org)
Google’s Solution to Term Spam

• Believe what people say about you, rather than what you say about yourself
  • Use words in the anchor text (words that appear underlined to represent the link) and its surrounding text
  • PageRank as a tool to measure the “importance” of Web pages

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Google vs. Spammers: Round 2!

- Once Google became the dominant search engine, spammers began to work out ways to fool Google

- **Spam farms** were developed to concentrate PageRank on a single page

- **Link spam:**
  - Creating link structures that boost PageRank of a particular page

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Link Spamming

• Three kinds of web pages from a spammer’s point of view
  • Inaccessible pages
  • Accessible pages
    • e.g., blog comments pages
    • spammer can post links to his pages
  • Owned pages
    • Completely controlled by spammer
    • May span multiple domain names

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
Link Farms

• **Spammer’s goal:**
  • Maximize the PageRank of target page \( t \)

• **Technique:**
  • Get as many links from accessible pages as possible to target page \( t \)
  • Construct “link farm” to get PageRank multiplier effect

(adapted from:: Mining of Massive Datasets, [http://www.mmds.org](http://www.mmds.org))
Link Farms

One of the most common and effective organizations for a link farm

(adapted from: Mining of Massive Datasets, http://www.mmds.org)
PageRank: Extensions

\[ p^t = \beta M p^{t-1} + (1 - \beta) p^0 = \tilde{M} p^{t-1} \]

- **Topic-specific PageRank:**
  - Restrict teleportation to some set \( S \) of pages related to a specific topic
  - Set \( p^0_i = 1/|S| \) if \( i \in S \), \( p^0_i = 0 \) otherwise

- **Trust Propagation**
  - Use set \( S \) of trusted pages for teleport set