CSE 494/598
Lecture-7: Link Analysis

LYDIA MANIKONDA
HTTP://WWW.PUBLIC.ASU.EDU/~LMANIKON/

**Content adapted from last year’s slides**
Announcements

• Project part-2 due on **March 18th 2016**
  • Hardcopy
  • **Submit code through email to** cse494s16@gmail.com

• Extended deadline till **March 21st 2016**
  • Hardcopy: 4 pm
  • Submit your code by 11:59 pm
Today

- PageRank algorithm
- Introduction to Social Networks
Two (very similar) ideas for assessing page importance

AUTHORITIES/HUBS (HITS)

View hyper-linked pages as authorities and hubs.
- Authorities are pointed to by Hubs (and derive their importance from who are pointing to them)
- Hubs point to authorities (and derive their importance from who they point to)

Return good Hub and Authority pages...

PAGERANK

View hyper-linked pages as a markov chain
- A page is important if the probability of a random surfer landing on that page is high

Return pages with “high probability of landing”
PageRank (Importance as Stationary Visit Probability on a Markov Chain)

Basic Idea:

- Think of web as a big graph.
- A random surfer keeps randomly clicking on the links.
- The importance of a page is the probability that the surfer finds herself on that page.
- Talk of transition matrix instead of adjacency matrix.
PageRank (Importance as Stationary Visit Probability on a Markov Chain)

Transition matrix $M$ derived from adjacency matrix $A$

- If there are $F(u)$ forward links from a page $u$, then the probability that the surfer clicks on any of those is $1/F(u)$
  - Columns sum to 1. Stochastic matrix

Even a dumb user may once in a while do something other than follow URLs on the current page

- Put a small probability that the user goes off to a page not pointed to by the current page
- Question – When you are bored, *where* do you go?
- Reset distribution – can be different for different people
Example: Suppose the Web Graph is:

\[
A = \begin{pmatrix}
A & B & C & D \\
A & 0 & 0 & 1 & 0 \\
B & 0 & 0 & 1 & 0 \\
C & 0 & 0 & 0 & 1 \\
D & 1 & 1 & 0 & 0
\end{pmatrix}
\]

\[
M = \begin{pmatrix}
A & B & C & D \\
A & 0 & 0 & 0 & ½ \\
B & 0 & 0 & 0 & ½ \\
C & 1 & 1 & 0 & 0 \\
D & 0 & 0 & 1 & 0
\end{pmatrix}
\]
Let $R$ be the vector of occupation probabilities of the pages at the steady state.

By definition of steady state, $R = M \times R$.

Suppose we start with the initial vector $R_0$ and “power iterate” $R_{i+1} \leftarrow M \times R_i$.

If this procedure converges, then we get $R$.

(So $R$ is the eigenvector of matrix $M$ with eigenvalue being 1).

Principal eigen value for a stochastic matrix is 1.

But are we sure this will always happen?

Do all Markov chains have a unique steady state occupation probability distribution?
Markov Chains

- Markov Chains & Stationary distribution
  - Necessary conditions for existence of unique steady state distribution
    - Aperiodicity
    - Irreducibility
  - Aperiodicity: It is not a big cycle
  - Irreducibility: Each node can be reached from every other node with non-zero probability
    - Must not have sink nodes – which have no out links
    - Must not have disconnected components

Why no sink nodes?
- Because we can have several different steady state distributions based on which sink we get stuck in
- If there are sink nodes, change them so that you can transition from them to every other node with low probability

Why no disconnected components?
- Because we can have several different steady state distributions depending on which disconnected component we get stuck in
- Sufficient to put a low probability link from every node to every other node (in addition to the normal weight links corresponding to actual hyperlinks)
- This can be used as the “reset” distribution – the probability that the surfer gives up navigation and jumps to a new page

\[ M^* = c(M+Z) + (1-c)K \]
Random Surfer Model

Main parameters:

• $c$ – the probability that surfer follows a link on the page
  • The larger it is, the more the surfer sticks to this page

• $M$ – the way link matrix is converted to markov chain
  • Can make the links have differing transition probability
    • E.g. Query specific links have higher probability; Links in bold have higher probability; etc.

• $Z$ – sink node elimination matrix
  • If $M$ has all zero columns, put an all $1/n$ column in $Z$

• $K$ – reset distribution of the surfer
  • It is quite feasible to have $m$ different reset distributions corresponding to $m$ different populations of users (or $m$ possible topic-oriented searches)
  • It is also possible to make the reset distribution depend on other things such as:
    • Trust of the page (TrustRank)
    • Recency of the page (Recency-Sensitive rank)
Computing PageRank

\[ M^* = c (M + Z) + (1 - c) \times K \]

\( M^* \) is irreducible.

\( M^* \) is stochastic, the sum of all entries of each column is 1 and there are no negative entries.

Therefore, if \( M \) is replaced by \( M^* \) as in

\[ R_i = M^* \times R_{i-1} \]

then the convergence is guaranteed.
Reset Distribution Matrix

- An nXn matrix, where the $i^{th}$ column gives the probability that the user will go off to a random page when he wants to “get-out”
  - All we need thus is – all columns add up to 1
  - No requirement that the columns define a uniform distribution
    - They can capture the user’s special interests (e.g., more probability mass concentrated on CS pages, and less on news sites)
  - No requirement that the columns must all be the same distribution
    - They can capture the fact that the users might want to visit pages that are very different
      - E.g., A user who wants to get out of a CS page may decide to go to a non-CS (e.g., news) page with higher probability; while the same user who has done enough news surfing for the day might want to get out with higher preference to CS pages
Computing PageRank – Example

Suppose the Web graph is:

A → B → C
A → D
D → C

Matrix:

\[
\begin{bmatrix}
A & B & C & D \\
A & 0 & 0 & 0 & \frac{1}{2} \\
B & 0 & 0 & 0 & \frac{1}{2} \\
C & 1 & 1 & 0 & 0 \\
D & 0 & 0 & 1 & 0
\end{bmatrix}
\]
Suppose $c = 0.8$. All entries in $Z$ are 0 and all entries in $K$ are $\frac{1}{4}$.

\[ M^* = 0.8 (M+Z) + 0.2 K = \begin{pmatrix} 0.05 & 0.05 & 0.05 & 0.45 \\ 0.05 & 0.05 & 0.05 & 0.45 \\ 0.85 & 0.85 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.85 & 0.05 \end{pmatrix} \]

Compute rank by iterating

\[ R := M^* R \]

MATLAB says:

- $R(A) = 0.338$ (176)
- $R(B) = 0.338$ (176)
- $R(C) = 0.6367$ (332)
- $R(D) = 0.6052$ (315)

Eigen decomposition gives the *unit* vector. To get the “probabilites” just normalize by dividing every number by the sum of the entries.
Comparing PR & A/H on the same graph

Eigen vectors =

\[
\begin{array}{cccc}
0.3380 & -0.1581 + 0.2739i & -0.1581 - 0.2739i & 0.7071 \\
-0.3380 & -0.1581 + 0.2739i & -0.1581 - 0.2739i & -0.7071 \\
-0.6366 & 0.6325 & 0.6325 & 0.0000 \\
-0.6052 & -0.3162 - 0.5477i & -0.3162 + 0.5477i & -0.0000 \\
\end{array}
\]

Eigen values =

\[
\begin{array}{cccc}
1.0000 & 0 & 0 & 0 \\
0 & -0.4000 + 0.6928i & 0 & 0 \\
0 & 0 & -0.4000 - 0.6928i & 0 \\
0 & 0 & 0 & 0.0000 \\
\end{array}
\]

Actually, this one has eigen gap zero
Which means both the right most and
The one next to it can be seen as primary
Eigen vectors—both of them provide
Stable A/H scores..

Eigenvalues=

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 2 & 0 \\
0 & 0 & 0 & 2 \\
\end{array}
\]

auth =

\[
\begin{array}{cccc}
-0.7071 & 0 & 0 & 0.7071 \\
0.7071 & 0 & 0 & 0.7071 \\
0 & 0 & 1.0000 & 0 \\
0 & 1.0000 & 0 & 0 \\
\end{array}
\]

Hub=

\[
\begin{array}{cccc}
-0.7071 & 0 & 0 & 0.7071 \\
0.7071 & 0 & 0 & 0.7071 \\
0 & 1.0000 & 0 & 0 \\
0 & 0 & 0 & 1.0000 \\
\end{array}
\]
auth =
0 0 0 1.0000 0 q1
1.0000 0 0 0 0 q2
0 1.0000 0 0 0 q3
0 0 0.6154 0 -0.7882 p1
0 0 -0.7882 0 -0.6154 p2

v =
0 0 0 0 0
0 0 0 0 0
0 0 0.4384 0 0
0 0 0 1.0000 0
0 0 0 0 4.5616

hub =
0.7071 0 0.2610 0 0.6572
0.0000 0 -0.9294 0 0.3690
-0.7071 0 0.2610 0 0.6572
0 0 0 1.0000 0
0 1.0000 0 0 0

Big Authorities ➔ Big Page Rank
Pure Hub ➔ low page rank
When to do Importance Computation?

**GLOBAL**

Do A/H (or PageRank) computation once for the whole corpus

**Advantage:** Importance computation done before the query time

**Disadvantage:** Importance is not sensitive to the individual queries

**Compromise:**
Do Importance computation w.r.t. topics
At query time, map query to topics and use the appropriate importance values

**QUERY-SPECIFIC**

Do A/H (or PageRank) computation w.r.t the query results (and their backward/forward neighbors)

**Advantage:** Importance computation sensitive to queries

**Disadvantage:** Importance computation is done at query time! (Slows down querying)
How to combine Importance and Relevance (Similarity) metrics?

• If you do query-specific importance computation, then you first do similarity and then importance..

• If you do global importance computation, then you need to combine apples and oranges..
Query-dependent ranking: the neighborhood graph

1. Subgraph associated to each query

- Back Set
  - \( b_1 \)
  - \( b_2 \)
  - \( \ldots \)
  - \( b_m \)

- Query Results
  - \( \text{Result}_1 \)
  - \( \text{Result}_2 \)
  - \( \ldots \)
  - \( \text{Result}_n \)

- Forward Set
  - \( f_1 \)
  - \( f_2 \)
  - \( \ldots \)
  - \( f_s \)

An edge for each hyperlink, but no edges within the same host
Combining PageRank & Content Similarity

Incorporate the ranks of pages into the ranking function of a search engine

• The ranking score of a web page can be a weighted sum of its regular similarity with a query and its importance

• \[ \text{ranking\_score}(q, d) \]

  \[
  = w \times \text{sim}(q, d) + (1-w) \times \text{R}(d), \text{ if } \text{sim}(q, d) > 0
  = 0, \text{ otherwise}
  \]

  where \( 0 < w < 1 \)

• Both \( \text{sim}(q, d) \) and \( \text{R}(d) \) need to be normalized to between \([0, 1]\)

Who sets \( w \)?
We can pick and choose

TWO ALTERNATE WAYS OF COMPUTING PAGE IMPORTANCE

I1: As authorities/hubs
I2: As stationary distribution over the underlying markov chain

TWO ALTERNATE WAYS OF COMBINING IMPORTANCE WITH SIMILARITY

C1: Compute importance over a set derived from top-100 similar pages
C2: Combine apples and oranges
   ◦ a*importance + b*similarity

We can pick any pair of alternatives
(even though I1 was originally proposed with C1 and I2 with C2)
Making Link Analysis even more query specific..

Should all links be equally treated?

Two considerations:

1. Some links may be more meaningful or important than other links
2. Website creators may trick the system to make their pages more authoritative by adding dummy pages pointing to their cover pages (spamming)
Handling spam links

**Transverse links**: Links between pages with different domain names

**Intrinsic links**: Links between pages with the same domain name

-- **Domain name**: the first level of the URL of a page.

Transverse links are more important than intrinsic links.

Two ways to incorporate this:

1. Use only transverse links and discard intrinsic links
2. Give lower weights to intrinsic links
Handling spam links

How to give lower weights to intrinsic links?

In adjacency matrix A, entry \((p, q)\) should be assigned as follows:

- If \(p\) has a transverse link to \(q\), the entry is 1
- If \(p\) has an intrinsic link to \(q\), the entry is \(c\), where \(0 < c < 1\)
- If \(p\) has no link to \(q\), the entry is 0
Considering link “context”

For a given link \((p, q)\), let \(V(p, q)\) be the vicinity (e.g., ± 50 characters) of the link

- If \(V(p, q)\) contains terms in the user query (topic), then the link should be more useful for identifying authoritative pages

- To incorporate this: In adjacency matrix \(A\), make the weight associated with link \((p, q)\) to be \(1+n(p, q)\)
  - Where \(n(p, q)\) is the number of terms in \(V(p, q)\) that appear in the query
  - Alternatively, consider the “vector similarity” between \(V(p, q)\) and the query \(Q\)
User study [BH’98]

Valuable pages within 10 top answers
(averaged over 28 topics)

- Original
- Edge Weighting
- EW + Content Analysis

Bar chart showing the comparison of valuable pages in the top 10 answers for authorities and hubs, with different weighting schemes.
Topic Specific PageRank

For each page compute \( k \) different page ranks

- \( k= \) number of top level hierarchies in the Open Directory Project
- When computing PageRank w.r.t. to a topic, say that with \( \varepsilon \) probability we transition to one of the pages of the topic \( k \)
- Could also consider link relevance to the topic

When a query \( q \) is issued,

- Compute similarity between \( q \) (+ its context) to each of the topics
- Take the weighted combination of the topic specific page ranks of \( q \), weighted by the similarity to different topics
PageRank Variants that play with Reset distribution..

• **Topic-specific page rank**
  • User goes to topic-relevant pages with higher probability
  • Think of this as a middle-ground between one-size-fits-all page rank and query-specific page rank

• **Trust rank**
  • User goes to more trustworthy pages with higher probability
  • Think of this as a middle-ground between one-size-fits-all page rank and user-specific page rank

• **Recency rank**
  • User goes to more recently created pages with higher probability
  • Allow recently generated (but probably high-quality) pages to break-through..

• **User-specific page rank**
  • User goes to pages in his social circle with higher probability
Stability w.r.t distributions and attacks

• Is the importance measure robust w.r.t small random changes?
• Is the importance measure robust w.r.t directed changes (“attacks”)?
  • Specifically, how easy is it to game the ranking?
Tyranny of Majority

Which do **you** think are
Authoritative pages?
Which are good hubs?

- *Intuitively, we would say that 4, 8, 5 will be authoritative pages and 1, 2, 3, 6, 7 will be hub pages.*

**BUT** The power iteration will show that

Only 4 and 5 have non-zero authorities
[.923 .382]
And only 1, 2 and 3 have non-zero hubs
[.5 .7 .5]

The authority and hub mass
Will concentrate completely
Among the first component, as
The iterations increase. (See next slide)
Tyranny of Majority

Suppose h0 and a0 are all initialized to 1

\[ a_1(p) = m \]
\[ a_1(q) = n \]

Normalized

\[ a_1(p) = \frac{m}{\sqrt{m^2 + n^2}} \]
\[ a_1(q) = \frac{n}{\sqrt{m^2 + n^2}} \]

\[ h_1(p_i) = \frac{m}{\sqrt{m^2 + n^2}} \]
\[ h_1(q_i) = \frac{n}{\sqrt{m^2 + n^2}} \]

\[ a_2(p) = \frac{m^2}{\sqrt{m^4 + n^4}} \]
\[ a_2(q) = \frac{n^2}{\sqrt{m^4 + n^4}} \]

\[ \frac{a_2(q)}{a_2(p)} = \left( \frac{n}{m} \right)^2 \]

\[ \frac{a_k(q)}{a_k(p)} = \left( \frac{n}{m} \right)^k \rightarrow 0 \]
Impact of Bridges..

When the graph is disconnected, only 4 and 5 have non-zero authorities 
[.923 .382]
And only 1, 2 and 3 have non-zero hubs 
[.5 .7 .5]CV

When the components are bridged by adding one page (9) the authorities change
only 4, 5 and 8 have non-zero authorities 
[.853 .224 .47]
And 01, 2, 3, 6, 7 and 9 will have non-zero hubs 
[.39 .49 .39 .21 .21 .6]

Bad news from stability point of view
→Can be fixed by putting a weak link between any two pages.. (saying in essence that you expect every page to be reached from every other page) (analogy to “vaccination”)
Stability of Rank Calculations (after random Perturbation)

(From Ng et. al.)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Book Title</th>
<th>Author(s)</th>
<th>Original Rank</th>
<th>Perturbed Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Genetic algorithms in search, optimization...</td>
<td>Goldberg</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Adaptation in natural and artificial systems</td>
<td>Holland</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>Genetic programming: On the programming of...</td>
<td>Koza</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Analysis of the behavior of a class of genetic...</td>
<td>De Jong</td>
<td>4</td>
<td>52</td>
</tr>
<tr>
<td>5</td>
<td>Uniform crossover in genetic algorithms</td>
<td>Syswerda</td>
<td>5</td>
<td>171</td>
</tr>
<tr>
<td>6</td>
<td>Artificial intelligence through simulated...</td>
<td>Fogel</td>
<td>6</td>
<td>135</td>
</tr>
<tr>
<td>7</td>
<td>A survey of evolution strategies</td>
<td>Back+al</td>
<td>7</td>
<td>179</td>
</tr>
<tr>
<td>8</td>
<td>Optimization of control parameters for genetic...</td>
<td>Grefenstette</td>
<td>8</td>
<td>316</td>
</tr>
<tr>
<td>9</td>
<td>The GENITOR algorithm and selection pressure</td>
<td>Whitley</td>
<td>9</td>
<td>257</td>
</tr>
<tr>
<td>10</td>
<td>Genetic algorithms + Data Structures</td>
<td>Michalewicz</td>
<td>13</td>
<td>170</td>
</tr>
<tr>
<td>11</td>
<td>Genetic programming II: Automatic discovery...</td>
<td>Koza</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Learning internal representations by error...</td>
<td>Rumelhart+al</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

Although it might be thought that this variability is intrinsic to the problem, this is not the case, as shown by the results from the PageRank algorithm, which were much more stable:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Book Title</th>
<th>Author(s)</th>
<th>Original Rank</th>
<th>Perturbed Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Genetic Algorithms in Search, Optimization and...</td>
<td>Goldberg</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Learning internal representations by error...</td>
<td>Rumelhart+al</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Adaptation in Natural and Artificial Systems</td>
<td>Holland</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Classification and Regression Trees</td>
<td>Breiman+al</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Probabilistic Reasoning in Intelligent Systems</td>
<td>Pearl</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Genetic Programming: On the Programming of...</td>
<td>Koza</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Learning to Predict by the Methods of Temporal...</td>
<td>Sutton</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Pattern classification and scene analysis</td>
<td>Duda+Hart</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Maximum likelihood from incomplete data via...</td>
<td>Dempster+al</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>UCI repository of machine learning databases</td>
<td>Murphy+Aha</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>Parallel Distributed Processing</td>
<td>Rumelhart+McClelland</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Introduction to the Theory of Neural Computation</td>
<td>Hertz+al</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The results are discussed in more detail in Section C.
To improve stability, focus on the plane defined by the primary and secondary eigen vectors (e.g. take the cross product of the two...)

Two ways you can make A/H just as stable
1. Put weak links for the adjacency matrix too
2. Consider the cross-product of primary & secondary eigen vectors

Figure 1: Jittered scatterplot of hyperlink graph.

Figure 2: Contours of two matrices with different eigengaps.
Finding communities using link analysis

How to retrieve pages from smaller communities?

A method for finding pages in nth largest community:

1. Identify the next largest community using the existing algorithm
2. Destroy this community by removing links associated with pages having large authorities
3. Reset all authority and hub values back to 1 and calculate all authority and hub values again
4. Repeat the above n-1 times and the next largest community will be the nth largest community
### Multiple Clusters on “House”

**Query:** House (first community)

<table>
<thead>
<tr>
<th>No</th>
<th>hub</th>
<th>authority</th>
<th>URL</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.06335</td>
<td>0.711035</td>
<td><a href="http://thomas.loc.gov/">http://thomas.loc.gov/</a></td>
<td>Thomas – u.s. congress on the internet</td>
</tr>
<tr>
<td>2</td>
<td>0.12457</td>
<td>0.602032</td>
<td><a href="http://www.house.gov/">http://www.house.gov/</a></td>
<td>United states house of representatives – 106 congress</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0.143384</td>
<td><a href="http://www.loc.gov/">http://www.loc.gov/</a></td>
<td>Library of congress homepage</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.142734</td>
<td><a href="http://law.house.gov/uscm.htm">http://law.house.gov/uscm.htm</a></td>
<td>u.s. house of representatives – internet law library</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.122806</td>
<td><a href="http://thomas.loc.gov/home/hcomso.html">http://thomas.loc.gov/home/hcomso.html</a></td>
<td>House committee hearing schedules and oversight plans</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.109974</td>
<td><a href="http://www.whitehouse.gov/WH/Welcome.html">http://www.whitehouse.gov/WH/Welcome.html</a></td>
<td>Welcome to white house</td>
</tr>
</tbody>
</table>
### Authority and Hub pages

Query: House (second community)

<table>
<thead>
<tr>
<th>No</th>
<th>hub</th>
<th>Authority</th>
<th>URL</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.780042</td>
<td><a href="http://www.homepath.com/hpc2.html">http://www.homepath.com/hpc2.html</a></td>
<td>How much house can you afford?</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0.625727</td>
<td><a href="http://www.homepath.com/hpc4.html">http://www.homepath.com/hpc4.html</a></td>
<td>How much house can you afford with a specific monthly payment?</td>
</tr>
</tbody>
</table>
Robustness against adversarial attacks..

- Stability talks about “random” addition of links
  - Stability can be improved by introducing weak links

- Robustness talks about the extent to which the importance measures can be co-opted by the adversaries..

- Robustness is a bigger problem for “global” importance measures (as against query-dependent ones)
  - Search King
  - JC Penney / Overstock in Spring 2011
    - Mails asking you to put ads on your page..
Effect of collusion on PageRank

Assuming $\alpha=0.8$ and $K=[1/3]$

$M^* = \begin{bmatrix} 0.66 & 0.066 & 0.866 \\ 0.866 & 0.066 & 0.066 \\ 0.066 & 0.866 & 0.066 \end{bmatrix}$

$M^* = \begin{bmatrix} 0.66 & 0.066 & 0.466 \\ 0.866 & 0.066 & 0.466 \\ 0.066 & 0.866 & 0.066 \end{bmatrix}$

Rank(A)=Rank(B)=Rank(C)= 0.5774

Rank(A)=0.37

Rank(B)=0.6672

Rank(C)=0.6461

Moral: By referring to each other, a cluster of pages can artificially boost their rank (although the cluster has to be big enough to make an appreciable difference).

Solution: Put a threshold on the number of intra-domain links that will count

Counter: Buy two domains, and generate a cluster among those..

Solution: Google dance → manually change the page rank once in a while...

Counter: Sue Google!
Google wins over SearchKing in PageRank case

(June 2 2003) Last year Searchking sued Google for reducing the PageRank of many of its member sites. Now the court has dismissed the case.

As reported by Pandia last fall Google reduced the PageRank of the SearchKing home page. The SearchKing owned PR Ad Network had started selling text ads on the Searchking network of sites based on Google PageRank. The home page of the PR Ad Network dropped to zero after the move.

Later Google restored SearchKing's ranking in search engine results. However, the PageRank remained low.

SearchKing was not amused, and refused to withdraw its lawsuit against Google. "This suit has never been about the priority we were given in a search or even our PageRank," said Bob Massa, president of SearchKing, Inc. and PR Ad Network.

"The case is about Google's attempt to squelch competition by targeting businesses and arbitrarily reducing their PageRank or search status. So they've restored it ñ for now. Next month, what is to stop them from reducing my ranking again?"

PageRank is a measure of Web page popularity and will to a certain extent influence the rankings of a Web page in Google search results.
Content Farms

eHow, Associated content etc

• Track what people are searching for
• Make up pages with those words, and have freelancers write shoddy articles
• Demand Media – which owned eHow – went public in Spring 2011 and became worth 1.6 billion dollars

http://www.wired.com/magazine/2010/02/ff_google_algorithm/
Why pay when you can induce people to put links freely?

But how?
→ Be a good business (nyah—takes too long)
→ Be a bad Business and people will put links to your page with their complaints
Being bad to your customers is bad for business
12/01/2010 12:06:00 PM
A recent article by the New York Times related a disturbing story. By treating your customers badly, one merchant told the paper, you can generate complaints and negative reviews that translate to more links to your site, which in turn, make it more prominent in search engines. The main premise of the article was that being bad on the web can be good for business.

We were horrified to read about Ms. Rodriguez’s dreadful experience. Even though our initial analysis pointed to this being an edge case and not a widespread problem in our search results, we immediately convened a team that looked carefully at the issue. That team developed an initial algorithmic solution, implemented it, and the solution is already live. I am here to tell you that being bad is, and hopefully will always be, bad for business in Google’s search results.

As always, we learned a lot from this experience, and we wanted to share some of that with you. Consider the obvious responses we could have tried to fix the problem:

- Block the particular offender. That would be easy and might solve the immediate problem for that specific business, but it wouldn’t solve the larger issue in a general way. Our first reaction in search quality is to look for ways to solve problems algorithmically.

- Use sentiment analysis to identify negative remarks and turn negative comments into negative votes. While this proposal initially sounds promising, it turns out to be based on a misconception. First off, the terrible merchant in the story wasn’t really ranking because of links from customer complaint websites. In fact, many consumer community sites such as Get Satisfaction added a simple attribute called repfdb in their links. This metadata attribute is a general mechanism that allows websites to tell search engines not to give weight to specific links, and it’s perfect for the situation when you want to link to a site without endorsing it. Ironically, some of the most reputable links to Decor My Eyes came from mainstream news websites such as the New York Times and Bloomberg. The Bloomberg article was about someone suing the company behind Decor My Eyes, but the language of the article was neutral, so sentiment analysis wouldn’t have helped here either.

As it turns out, Google has a world-class sentiment analysis system (Large-Scale Sentiment Analysis for News and Blogs). But if we demoted web pages that have negative comments against them, you might not be able to find information about many elected officials, not to mention a list of important but controversial concepts. So far we have not found an effective way to significantly improve search using sentiment analysis. Of course, we will continue trying.

- Yet another option is to expose user reviews and ratings for various merchants alongside their results. Though still on the table, this would not remove poor quality merchants in our results and could still lead users to their websites.

Instead, in the last few days we developed an algorithmic solution which detects the merchant from the Times article along with hundreds of other merchants that, in our opinion, provide an extremely poor user experience. The algorithm we incorporated into our search rankings represents an initial solution to this issue, and Google users are now experiencing a better experience as a result.
And another honest merchant who understood pagerank/link analysis is put behind bars 😞

pressures explained his behavior.

"I was answering personally about 100 e-mails a day and lost control of what I was saying at times," he said, reading a prepared statement before Judge Richard J. Sullivan in Federal District Court in Manhattan. He closed by saying,

"I want to apologize to everyone I hurt in connection with my actions, especially those people I threatened."

Mr. Borker achieved something close to instant notoriety in late November after The New York Times published an article in which he discussed his habit of menacing customers who had complained to him about products bought through DecorMyEyes. Using several aliases, he threatened to kill anyone who would not pay for accessories that he said belonged to the company.
Stability (random change) and Robustness (adversarial change) of Link Importance measures

FOR RANDOM CHANGES

E.g., A randomly added link

we know that stability depends on ensuring that there are no disconnected components in the graph to begin with

- e.g. the “standard” A/H computation is unstable w.r.t. bridges if there are disconnected components—but become more stable if we add low-weight links from every page to every other page

We can always make up a story about these capturing transitions by impatient user 😊

FOR ADVERSARIAL CHANGES

(where someone with an adversarial intent makes changes to link structure of the web, to artificially boost the importance of certain pages),

- It is clear that query specific importance measures (e.g. computed w.r.t. a base set) will be harder to sabotage.
- In contrast query (and user-) independent similarity measures are easier (since they provide a more stationary target).
Project Part 2

LinkAnalysis.java
- depends on two files IntCitations.txt and IntLinks.txt
- To get every document that points to document number 1234, call
  - `link_analysis.getCitations(1234)`
- To get every document that document 1234 points to, call
  - `link_analysis.getLinks(1234)`
- Using this, you can create the adjacency graph
Project 2: Authorities/Hubs

- Use your phase 1 code
- TF-IDF

Lucene Index

Root Set

- Use LinkAnalysis to grow
- Links, citations

Base Set

- Perform Auth/Hub computation
- Rank

Final Results
Project 2: PageRank

Precomputed PageRank + (1-w) Vector Similarity = Final Results

\[ R_i = M^* \times R_{i-1} \]
\[ M^* = c(M + Z) + (1-c) K \]

- \( R_i \): The current PageRank of the pages.
- \( M^* \): The adjusted link matrix
- \( M \): The original adjacency matrix
- \( Z \): Adjustment for sink pages (1/N for sink pages, 0 otherwise)
- \( K \): Reset Matrix (1/N for all entries)
- \( c \): A parameter that you control
Social Networks and their Applications on Web