Link Analysis
Web Ranking

• Documents on the web are first ranked according to their relevance vrs the query
• Additional ranking methods are needed to cope with huge amount of information
• Additional ranking methods:
  – Classification (manual, automatic)
  – Link Analysis (today’s lesson)
Why link analysis?

• The web is not just a collection of documents – its hyperlinks are important!

• A link from page $A$ to page $B$ may indicate:
  – $A$ is related to $B$, or
  – $A$ is recommending, citing, voting for or endorsing $B$

• Links are either
  – referential – click here and get back home, or
  – Informational – click here to get more detail

• Links affect the ranking of web pages and thus have commercial value.

• The idea of using links is somehow “borrowed” by citation analysis
Citation Analysis

- The **impact factor** of a journal = $A/B$
  - $A$ is the number of *current year citations* to articles appearing in the journal during previous two years.
  - $B$ is the *number of articles* published in the journal during previous two years.

<table>
<thead>
<tr>
<th>Journal Title (AI)</th>
<th>Impact Factor (2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Mach. Learn. Res.</td>
<td>5.952</td>
</tr>
<tr>
<td>IEEE T. Pattern Anal.</td>
<td>4.352</td>
</tr>
<tr>
<td>IEEE T. Evolut. Comp.</td>
<td>3.688</td>
</tr>
<tr>
<td>Artif. Intell.</td>
<td>3.570</td>
</tr>
<tr>
<td>Mach. Learn.</td>
<td>3.258</td>
</tr>
</tbody>
</table>
Co-Citation

- A and B are co-cited by C, implying that they are related or associated.
- The strength of co-citation between A and B is the number of times they are co-cited.
Clusters from Co-Citation Graph

(Larson 96)
Citations vs. Links

- Web links are a bit different than citations:
  - Many links are navigational.
  - Many pages with high in-degree are portals, not content providers.
  - Not all links are endorsements (e.g. pointers to “fake” conferences).
  - Company websites don’t point to their competitors.

However, the general idea that “many citations = authority” has been borrowed in link analysis
Link Analysis

• HITS (Hyperlink Induced Topic Search) Jon Kleinberg
• Page Rank Larry Page, Sergei Brin
Hyperlink Induced Topic Search (HITS)
Main concept of the algorithm

• HITS stands for Hypertext Induced Topic Search.
• HITS is search query dependent.
• When the user issues a search query,
  – HITS first expands the list of relevant pages returned by a search engine and
  – then produces two rankings of the expanded set of pages, authority ranking and hub ranking.
Main concept of the algorithm-cont.

**Authority**: A authority is a page with many in-links.
- The idea is that the page may have good or authoritative content on some topic and thus many people trust it and link to it.

**Hub**: A hub is a page with many out-links.
- The page serves as an organizer of the information on a particular topic and points to many good authority pages on the topic (e.g. a portal).
HITS – Hubs and Authorities –

• A on the left is an authority
• A on the right is a hub
Description of HITS

• A good hub points to many good authorities, and
• A good authority is pointed to by many good hubs.

• Authorities and hubs have a **mutual reinforcement relationship**. The figure shows some densely linked authorities and hubs (a **bipartite sub-graph**).
The HITS algorithm: phase 1

- Given a broad search query, $q$, HITS collects a set of pages as follows:
  - It sends the query $q$ to a search engine.
  - It then collects $t$ ($t = 200$ is used in the HITS paper) highest ranked pages. This set is called the **root** set $W$.
  - It then grows $W$ by including any page pointed to by a page in $W$ and any page that points to a page in $W$. This gives a larger set $S$, **base set**.
Expanding the Root Set
The link graph $G$

- HITS works on the pages in $S$, and assigns every page in $S$ an authority score and a hub score.
- Let the number of pages in $S$ be $n$.
- We use $G = (V, E)$ to denote the hyperlink graph of $S$.
- We use $L$ to denote the adjacency matrix of the graph.

\[
L_{ij} = \begin{cases} 
1 & \text{if } (i, j) \in E \\
0 & \text{otherwise}
\end{cases}
\]
Adjacency Matrix examples

\[
\begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 0 & 1
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}
\]
The HITS algorithm (cont’d)

• Let the authority score of the page $i$ be $a(i)$, and the hub score of page $i$ be $h(i)$.

• The **mutual reinforcing relationship** of the two scores is represented as follows:

\[
\begin{align*}
    a(i) &= \sum_{(j,i) \in E} h(j) \\
    h(i) &= \sum_{(i,j) \in E} a(j)
\end{align*}
\]
HITS in matrix form

• We use $a$ to denote the column vector with all the authority scores,
  $$a = (a(1), a(2), \ldots, a(n))^T,$$
  and

• use $h$ to denote the column vector with all the "hub scores",
  $$h = (h(1), h(2), \ldots, h(n))^T,$$

• Then, we can express previous formulas as:
  $$a = L^T h$$
  $$h = La$$

• It is an equivalent formulation since the sum in previous formula is for $(i,j)$ in $E$, and $L$ has 1 where there is a link between $i$ and $j$. 
Computation of HITS

• The computation of authority scores and hub scores uses power iteration.

• If we use $a_k$ and $h_k$ to denote authority and hub vectors at the $k$th iteration, the iterations for generating the final solutions are

$$a_k = L^T L a_{k-1}$$

$$h_k = L L^T h_{k-1}$$

starting with

$$a_0 = h_0 = (1, 1, ..., 1)$$
The HITS algorithm

- \( h^{(0)} := (1, 1, \ldots) \)
- \( k := 1 \)
- Until convergence, do:
  - \( a^{(k)} := L^T h^{(k-1)} \) (update \( a \))
  - \( h^{(k)} := L a^{(k)} \) (update \( h \))
  - \( a^{(k)} := a^{(k)}/\|a^{(k)}\| \) and \( h^{(k)} := h^{(k)}/\|h^{(k)}\| \) (normalize)

• Assignments can be re-written as:
  - \( a^{(k)} := L^T h^{(k-1)} = L^T L a^{(k-1)} \)
  - \( h^{(k)} := L a^{(k)} = L L^T h^{(k-1)} \)
Meaning of the $L \ L^T$ and $L^T \ L$ matrixes

**L is the adjacency matrix** of the graph

$L^T \ L$ is the authority matrix:

\[
A = L^T \ L = \begin{pmatrix} \ldots & A_{ij} & \ldots \\ \ldots & L_{ik} & \ldots \\ \ldots & L_{kj} & \ldots \end{pmatrix} = \begin{pmatrix} \ldots & \ldots & \ldots \\ \ldots & L_{ik} & \ldots \\ \ldots & \ldots & L_{kj} \end{pmatrix}
\]

$L_{kj}$ means that $j$ is pointed by all non-zero $k$, $L_{ik}^T$ means that $i$ is pointed by all non-zero $k$.

\[
A_{ij} = \sum_{k=1}^{n} L_{ik}^T L_{kj} = \sum_{k=1}^{n} L_{ki} L_{kj}
\]

..is this something you have already seen??????

$A_{ij}$ is the number of co-citations, the number of nodes pointing to both $i$ and $j$. 

..is this something you have already seen???????
Convergence of HITS (power method)

- \( \lambda_1, \lambda_2, \ldots, \lambda_k \) are then eigenvalues of a matrix \( A (=LL^T) \) and
  \( |\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_k| \)
- \( x_1, \ldots, x_k \) are the eigenvectors and they are linearly independent (e.g.):
  \( \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_k x_k = 0 \) iff \( \alpha_1 = \ldots = \alpha_k = 0 \)
- A generic vector \( \nu_0^{(h(0))} \) or \( \alpha^{(0)} \) can be re-written as:
  \[ \nu_0 = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_k x_k \]
  hence:
  \[ -\nu = A \nu = A^m \nu_0 = \alpha_1 A^m x_1 + \alpha_2 A^m x_2 + \ldots + \alpha_k A^m x_k = \alpha_1 \lambda_1^m x_1 + \alpha_2 \lambda_2^m x_2 + \ldots + \alpha_k \lambda_k^m x_k \]

- And in general:
  \( \forall i: x_i, \lambda_i \) eigenvalue,eigenvector of \( A, A = \lambda_i x_i \)
  \[ \lim_{m \to \infty} \frac{1}{\lambda_1^m} \nu_m = \lim_{m \to \infty} \frac{1}{\lambda_1^m} A^m \nu_0 = \alpha_1 x_1 \]
HITS: Example (1)

Ex: 3 and 4 are “co-cited” by 5

$LL^T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 2 & 1 & 1 \\ 4 & 0 & 0 & 1 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 3 \\ 6 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
\[
-a^{(1)} := L^T h^{(0)} \\
-h^{(1)} := L a^{(1)}
\]

HITS: Example (2)

**Authorities**

**Hubs**

\[
\begin{align*}
\mathbf{a}_1 &= \begin{pmatrix} 0.258 \\
0 \\
0.516 \\
0.258 \\
0.775 \\
0 \end{pmatrix} \\
\mathbf{h}_0 &= \begin{pmatrix} 1 \\
0 \\
1 \\
1 \\
1 \\
1 \end{pmatrix} \\
\mathbf{T} &= \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{align*}
\]

\[
\begin{pmatrix} 0.258 \\
0 \\
0.516 \\
0.258 \\
0.775 \\
0 \end{pmatrix} \begin{pmatrix} 1 \\
0 \\
1 \\
0 \\
0 \\
0 \end{pmatrix} = 0.258 \\
\begin{pmatrix} 0.687 \\
0.137 \\
0.412 \\
0.412 \end{pmatrix}
\]

(normalization step is not shown)
HITS: Example (3)

\[-a^{(2)} := L^T h^{(1)} \]
\[-h^{(2)} := L a^{(2)} \]

**Authorities**

\[ \begin{pmatrix}
  1 & 0 & 0 & 1 & 0 & 1 \\
  2 & 1 & 0 & 0 & 0 & 0 \\
  3 & 0 & 0 & 0 & 0 & 1 \\
  4 & 0 & 0 & 0 & 0 & 0 \\
  5 & 0 & 0 & 1 & 1 & 0 \\
  6 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix} h_1 \end{pmatrix}^T = \begin{pmatrix}
  0.687 \\
  0.137 \\
  0.412 \\
  0.412 \\
  0.412 \\
  0.412
\end{pmatrix}
\]

\[ a_2 \begin{pmatrix}
  0.072 \\
  0.573 \\
  0.215 \\
  0.788 \\
  0 \\
  0
\end{pmatrix} = \begin{pmatrix}
  1 & 0 & 0 & 1 & 0 & 1 \\
  2 & 1 & 0 & 0 & 0 & 0 \\
  3 & 0 & 0 & 0 & 0 & 1 \\
  4 & 0 & 0 & 0 & 0 & 0 \\
  5 & 0 & 0 & 1 & 1 & 0 \\
  6 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

\[-a^{(2)} := L^T h^{(1)} \]
\[-h^{(2)} := L a^{(2)} \]

**Hubs**

\[ \begin{pmatrix}
  1 & 0 & 0 & 1 & 0 & 1 \\
  2 & 1 & 0 & 0 & 0 & 0 \\
  3 & 0 & 0 & 0 & 0 & 1 \\
  4 & 0 & 0 & 0 & 0 & 0 \\
  5 & 0 & 0 & 1 & 1 & 0 \\
  6 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \begin{pmatrix} h_2 \end{pmatrix}^T = \begin{pmatrix}
  0.706 \\
  0.037 \\
  0.409 \\
  0.409
\end{pmatrix}
\]

\[ a_2 \begin{pmatrix}
  0.072 \\
  0.573 \\
  0.215 \\
  0.788 \\
  0 \\
  0
\end{pmatrix} = \begin{pmatrix}
  1 & 0 & 0 & 1 & 0 & 1 \\
  2 & 1 & 0 & 0 & 0 & 0 \\
  3 & 0 & 0 & 0 & 0 & 1 \\
  4 & 0 & 0 & 0 & 0 & 0 \\
  5 & 0 & 0 & 1 & 1 & 0 \\
  6 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]
HITS: Example (4)

\[-a^{(3)} := L^T \mathbf{h}^{(2)}\]

\[-h^{(3)} := L a^{(3)}\]

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
0.706 \\
0.037 \\
0.409 \\
0 \\
0.409 \\
0.409
\end{bmatrix}
=
\begin{bmatrix}
0.019 \\
0 \\
0.577 \\
0.212 \\
0.789 \\
0
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
0 & 0 & 1 & 0 & 1 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 \\
6 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0.019 \\
0 \\
0.577 \\
0.212 \\
0.789 \\
0 \\
0.001
\end{bmatrix}
=
\begin{bmatrix}
0.707 \\
0.001 \\
0.408 \\
0.408 \\
0.408 \\
0.408
\end{bmatrix}
\]

- authorities
- hubs
HITS: Esempio (5)

\[-a^{(4)} := L^T h^{(3)}
\]

\[-h^{(4)} := L a^{(4)}\]

\[
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix}^T
\begin{pmatrix}
0.707 \\
0.001 \\
0.408 \\
0 \\
0.408 \\
0 \\
\end{pmatrix} =
\begin{pmatrix}
0 \\
0.577 \\
0.211 \\
0.789 \\
0 \\
0 \end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 \\
4 & 0 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 1 & 0 \\
6 & 0 & 0 & 0 & 0 & 1 \\
\end{pmatrix} a^{(4)}
\begin{pmatrix}
0 \\
0 \\
0.577 \\
0.211 \\
0.789 \\
0 \\
\end{pmatrix} =
\begin{pmatrix}
0.408 \\
0 \\
0.408 \\
0.408 \end{pmatrix}
\]

authorities

hubs
Strengths and weaknesses of HITS

• **Strength**: its ability to rank pages according to the query topic, which may be able to provide more relevant authority and hub pages.

• **Weaknesses**:
  - **It is easily spammed**: It is in fact quite easy to influence HITS since adding out-links in one’s own page is so easy.
  - **Topic drift**: Many pages in the expanded set may not be on topic.
  - **Inefficiency at query time**: The query time evaluation is slow. Collecting the root set, expanding it and performing eigenvector computation are all expensive operations.
Applications of HITS

• Search engine querying (speed an issue)
• Finding web communities.
• Finding related pages.
• Populating categories in web directories.
• Citation analysis.
Link Analysis

• HITS (Hyperlink Induced Topic Serach) Jon Kleinberg

• Page Rank  Larry Page, Sergei Brin
Page Rank

• Ranks pages by authority.

• Applied to the entire web rather than a local neighborhood of pages surrounding the results of a query.

• Not query-dependent

• It is the Google algorithm for ranking pages
PageRank----Idea

Every page has some number of out-links and in-links
PageRank----Idea

Two cases PageRank is interesting:

1. Web pages vary greatly in terms of the number of backlinks (in-links) they have. For example, the Netscape home page has 62,804 backlinks compared to most pages which have just a few backlinks. Generally, highly linked pages are more “important” than pages with few links.
PageRank----Idea

2. Backlinks coming from important pages convey more importance to a page. For example, if a web page has a link off the Yahoo home page, it may be just one link but it is a very important one.

A page has high rank if the sum of the ranks of its incoming links is high. This covers both the case when a page has many in-links and when a page has a few highly ranked in-links.
PageRank---Definition

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

The equation is recursive, but it may be computed by starting with any set of ranks and iterating the computation until it converges.

\[ u: \text{a web page} \]
\[ F_u: \text{set of pages } u \text{ points to} \]
\[ B_u: \text{set of pages that point to } u \]
\[ N_u = |F_u|: \text{the number of links from } u \]
\[ c: \text{a factor used for normalization} \]
Parto da pesi casuali
After several iterations...

Why stops here?
A probabilistic interpretation of PageRank

- The definition corresponds to the probability distribution of a **random walk** on the web graphs.
What is a Random Walk?

• Given a graph and a starting point (node), we select a neighbor of it at random, and move to this neighbor;

• Then we select a neighbor of this node and move to it, and so on;

• The (random) sequence of nodes selected this way is a random walk on the graph
An example

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Adjacency matrix A

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
</tr>
</tbody>
</table>

Transition matrix P

---

*Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview*
An example

$t=0, \ A$

Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview
An example

$t=0$, $A$

$A$  

$B$  

$C$

$t=1$, $AB$

$A$  

$B$  

$C$

Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview
An example

$t=0, A$

$t=1, AB$

$t=2, ABC$

Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview
An example

$t=0$, A

$t=1$, AB

$t=2$, ABC

$t=3$, ABCA ABCB

Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview
Probabilistic interpretation

- S total number of web pages
- k outgoing links from page j
- P Transition matrix with elements:
  \[ P_{ij} = \begin{cases} 
  \frac{1}{k} & \text{if } i \to j \\
  0 & \text{otherwise} 
  \end{cases} \]
  \[ P_{ii} > 0 \quad \forall i \]

- The PageRank formulation can be written as:
  \[ \vec{r} = \vec{r} \cdot P \]
How to compute the vector $r$ of page ranks?

• The random surfer (or random walks) model can be represented using Markov Chains
Markov Chains (1)

• A Markov Chain consists in n states (let $S$ the set of possible states), and a matrix of transition probabilities $n \times n$, $P$.

• At each step, the system is precisely in one state.

• For $1 \leq i, j \leq n$, $P(s_i \rightarrow s_j) = P_{ij}$ is the probability of jumping to $s_j$, given we are in $s_i$.

• Furthermore, if $X_k$ is the random variable indicating the state $s$ reached at time $t_k$ ($X$ gets values in $S$), then:

$$P(X_k /X_1, X_2, \ldots, X_{k-1}) = P(X_k /X_{k-1})$$

• The value of $X$ at time $k$ depends only from the value of the random variable at time $k-1$! (This is the basic property of Markov Chains)
Markov chains (2)

• Clearly \( \sum_{j=1}^{n} P_{ij} = 1. \)

• Markov Chains are a model of random walks.
**Probability Vectors**

- Let \( x^{(t)} = (x_1, \ldots x_n) \) be an \( S \)-dimensional vector indicating the state reached at time \( t \).
- Ex: \((000\ldots1\ldots000)\) means we are in \( s_i \).

But **since we are modeling a stochastic process**, we must rather consider a **vector of probabilities** \( x^{(t)} = (P(s_1), \ldots P(s_n)) = (x_1, \ldots x_n) \), indicating that at step \( t \) the walk will bring to state \( s_i \) with probability \( x_i \), and

\[
\sum_{i=1}^{n} x_i = 1.
\]
Ergodic Markov Chains

• A Markov Chain is **ergodic** if:
  
  – There is a path between any pair of states
  
  – Starting from any state, after a transition time \( T_0 \), the probability to reach any other state in a finite time \( T > T_0 \) is always different from zero.

  – **Note**: not true for the web graph!
Ergodic Chains

• If a Markov Chain is ergodic, every state has a **stationary probability** of being visited, regardless of the initial state of the random walker.

  – The vector $\mathbf{x}(t)$ of state probabilities converges to a **stationary vector** $\mathbf{r}$ as $t \to \infty$
Computing State Probability Vector

• If \( \mathbf{x}^{(k)} = (x_1, \ldots x_n) \) is the vector \( \mathbf{x}^{(t)} \) in step \( t=k \), how would it change after the next jump?

• The adjacency matrix \( \mathbf{P} \) tells us where we are likely to jump from any state (since it has all transition probabilities from \( s_i \) to the other linked states):

• Therefore, from \( \mathbf{x}^{(k)} \), the probability of next state \( \mathbf{x}^{(k+1)} \) is computed according to: \( \mathbf{x}^{(k+1)} = \mathbf{P}\mathbf{x}^{(k)} \)

• If the process is ergodic, \( \mathbf{x} \) will converge to a vector \( \mathbf{r} \) such that \( \mathbf{r} = \mathbf{P}\mathbf{r} \)

• Since \( \mathbf{P} \) is a matrix and \( \mathbf{r} \) is a vector, which vector is \( \mathbf{r} \)??
Again: the Power method

- \( x^{(k+1)} = P x^{(k)} \)
- The sequence of vectors \( x^k \) converge to the stationary vector \( r \)
- To compute \( r \) we use the same method as for HITS
  \( x^{(k+1)} = x P^k = x^{(k)} P = x^{(k)} \)
- The method converges provided there is a dominant (principal) eigenvector
- Since the stationary condition is: \( r = rP \), \( r \) is the principal eigenvector of \( P \)
- Remember definition of eigenvectors!
Example
The normalized adjacency matrix $P$

$$
P = \frac{1}{N(u_i)} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\
1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0
\end{bmatrix}
$$
\[ x^{(k+1)} = P x^{(k)} \]

<table>
<thead>
<tr>
<th></th>
<th>( x^0 )</th>
<th>( x^1 )</th>
<th>( x^2 )</th>
<th>( x^3 )</th>
<th>( x^4 )</th>
<th>( x^{60} )</th>
<th>( x^{611} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0.0278</td>
<td>...</td>
<td>0.06</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.25</td>
<td>0.1667</td>
<td>0.0833</td>
<td>...</td>
<td>0.0675</td>
<td>0.0675</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.25</td>
<td>0.1667</td>
<td>...</td>
<td>0.0675</td>
<td>0.0675</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.1667</td>
<td>0.1111</td>
<td>...</td>
<td>0.0975</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>0.1806</td>
<td>...</td>
<td>0.2025</td>
<td>0.2025</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0833</td>
<td>0.0972</td>
<td>...</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0833</td>
<td>0.3333</td>
<td>...</td>
<td>0.295</td>
<td>0.295</td>
</tr>
</tbody>
</table>
Recap: Pagerank

• Simulate a random surfer by the power iteration method

• Problems
  1. 0 pagerank if there are no incoming links or if there are sinks
  2. Not unique if the graph is disconnected
  3. Computationally intensive?
  4. Stability & Cost of recomputation (web is dynamic)
  5. Does not take into account the specific query
  6. Easy to fool
Problem 1: Rank Sink

If two web pages point to each other but to no other page, during the iteration, this loop will accumulate rank but never distribute any rank.
Rank Sink

• Problem: Pages in a loop accumulate rank but do not distribute it.

• Solution: Teleportation, i.e. with a certain probability the surfer can jump to any other web page to get out of the loop.
Definition modified (with teleporting)

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

- \( E(u) \) is some vector of probabilities over the web pages (for example uniform prob., favorite page etc.) that corresponds to a source of rank. \( c \) is the **dumping factor**.
- \( E(u) \) can be thought as the random surfer gets bored periodically and jumps to a different page and is not kept in a loop forever.
Figure 5: PageRank convergence as a function of the size of the web graph
Figure 6: PageRank convergence as a function of C

Note: “c” is the dumping factor
Teleporting

• This solves:
  – Sink problem (problem 1)
  – Disconnectedness (problem 2)
  – Converges fast if $w$ is chosen appropriately (problem 3??)

• We still have problems:
  1. Still computationally intensive?
  2. Stability & Cost of recomputation (web is dynamic)
  3. Does not take into account the specific query
  4. Easy to fool
The Largest Matrix Computation in the World

• Computing PageRank can be done via matrix multiplication, where the matrix has 3 billion rows and columns.
• The matrix is sparse as average number of outlinks is between 7 and 8.
• Setting $c = 0.15$ or above requires at most 100 iterations to convergence.
• Researchers still trying to speed-up the computation.
Monte Carlo Methods in Computing PageRank

• Rather than following a single long random walk, the random surfer can follow many sampled random walks.

• Each walk starts at a random page and either teleports with probability $c$ or continues choosing a link with uniform probability.

• The PR of a page is the proportion of times a sample random walk ended at that page.

• Rather than starting at a random page, start each walk a fixed number of times from each page.
Personalised PageRank

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

• Change \( cE(v) \) with \( cv \)

• Instead of teleporting uniformly to any page we bias the jump prefer some pages over others.
  – E.g. \( v \) has 1 for your home page and 0 otherwise.
  – E.g. \( v \) prefers the topics you are interested in.
Weblogs influence on PageRank

• A weblog (or blog) is a frequently updated web site on a particular topic, made up of entries in reverse chronological order.

• Blogs are a rich source of links, and therefore their links influence PageRank.

• A “google bomb” is an attempt to influence the ranking of a web page for a given phrase by adding links to the page with the phrase as its anchor text.

• Google bombs date back as far as 1999, when a search for "more evil than Satan himself" resulted in the Microsoft homepage as the top result.
Agoogle bombs
Link Spamming to Improve PageRank

- Spam is the act of trying unfairly to gain a high ranking on a search engine for a web page without improving the user experience.

- Link farms - join the farm by copying a hub page which links to all members.

- Selling links from sites with high PageRank.