Problem formulation (1998)

- Suppose we are given a collection of documents on some broad topic:
  - e.g., stanford, evolution, iraq
  - perhaps obtained through a text search
- Can we organize these documents in some manner?
  - Page rank offers one solution
  - HITS (Hypertext-Induced Topic Selection) is another
    - proposed at approx the same time

HITS Model

- Interesting documents fall into two classes
  1. **Authorities** are pages containing useful information
     - course home pages
     - home pages of auto manufacturers
  2. **Hubs** are pages that link to authorities
     - course bulletin
     - list of US auto manufacturers
### Idealized view

<table>
<thead>
<tr>
<th>Hubs</th>
<th>Authorities</th>
</tr>
</thead>
</table>

### Mutually recursive definition

- A good hub links to many good authorities
- A good authority is linked from many good hubs
- Model using two scores for each node
  - Hub score and Authority score
  - Represented as vectors $h$ and $a$

### Transition Matrix $A$

- HITS uses a matrix $A[i, j] = 1$ if page $i$ links to page $j$, 0 if not
- $A^T$, the transpose of $A$, is similar to the PageRank matrix $M$, but $A^T$ has 1’s where $M$ has fractions
Hub and Authority Equations

- The hub score of page P is proportional to the sum of the authority scores of the pages it links to
  - \( h = \lambda Aa \)
  - Constant \( \lambda \) is a scale factor
- The authority score of page P is proportional to the sum of the hub scores of the pages it is linked from
  - \( a = \mu A^T h \)
  - Constant \( \mu \) is scale factor

Iterative algorithm

- Initialize \( h, a \) to all 1’s
- \( h = Aa \)
- Scale \( h \) so that its max entry is 1.0
- \( a = A^T h \)
- Scale \( a \) so that its max entry is 1.0
- Continue until \( h, a \) converge
Example

\[
A = \begin{bmatrix}
1 & 1 & 1 \\
1 & 0 & 1 \\
0 & 1 & 0 \\
\end{bmatrix}, \quad A^T = \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{bmatrix}
\]

\[
a(yahoo) = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}
\]

\[
a(amazon) = \begin{bmatrix} 1 & 1 & 4/5 & 0.75 & \cdots & 0.732 \end{bmatrix}
\]

\[
a(m'soft) = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}
\]

\[
h(yahoo) = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}
\]

\[
h(amazon) = \begin{bmatrix} 1 & 2/3 & 0.71 & 0.73 & \cdots & 0.732 \end{bmatrix}
\]

\[
h(m'soft) = \begin{bmatrix} 1 & 1/3 & 0.29 & 0.27 & \cdots & 0.268 \end{bmatrix}
\]

Existence and Uniqueness

\[ h = \lambda A a \]
\[ a = \mu A^T h \]
\[ h = \lambda \mu A A^T h \]
\[ a = \lambda \mu A^T A a \]

Under reasonable assumptions about \( A \), the dual iterative algorithm converges to vectors \( h^* \) and \( a^* \) such that:

- \( h^* \) is the principal eigenvector of the matrix \( AA^T \)
- \( a^* \) is the principal eigenvector of the matrix \( A^T A \)

Bipartite cores
Secondary cores

- A single topic can have many bipartite cores
  - corresponding to different meanings, or points of view
  - abortion: pro-choice, pro-life
  - evolution: darwinian, intelligent design
  - jaguar: auto, Mac, NFL team, *panthera onca*
- How to find such secondary cores?

Non-primary eigenvectors

- $A A^T$ and $A^T A$ have the same set of eigenvalues
  - An eigenpair is the pair of eigenvectors with the same eigenvalue
  - The primary eigenpair (largest eigenvalue) is what we get from the iterative algorithm
- Non-primary eigenpairs correspond to other bipartite cores
  - The eigenvalue is a measure of the density of links in the core

Finding secondary cores

- Once we find the primary core, we can remove its links from the graph
- Repeat HITS algorithm on residual graph to find the next bipartite core
- Technically, not exactly equivalent to non-primary eigenpair model
Creating the graph for HITS

- We need a well-connected graph of pages for HITS to work well

Page Rank and HITS

- Page Rank and HITS are two solutions to the same problem
  - What is the value of an inlink from S to D?
  - In the page rank model, the value of the link depends on the links into S
  - In the HITS model, it depends on the value of the other links out of S
- The destinies of Page Rank and HITS post-1998 were very different
  - Why?

Web Spam

- Search has become the default gateway to the web
- Very high premium to appear on the first page of search results
  - e.g., e-commerce sites
  - advertising-driven sites
What is web spam?

- **Spamming** = any deliberate action solely in order to boost a web page’s position in search engine results, incommensurate with page’s real value.
- **Spam** = web pages that are the result of spamming.
- This is a very broad definition.
  - SEO industry might disagree!
  - SEO = search engine optimization.
- Approximately 10-15% of web pages are spam.

---

Web Spam Taxonomy

- We follow the treatment by Gyongyi and Garcia-Molina [2004].
- **Boosting techniques**
  - Techniques for achieving high relevance/importance for a web page.
- **Hiding techniques**
  - Techniques to hide the use of boosting:
    - From humans and web crawlers.

---

Boosting techniques

- **Term spamming**
  - Manipulating the text of web pages in order to appear relevant to queries.
- **Link spamming**
  - Creating link structures that boost page rank or hubs and authorities scores.
Term Spamming

- **Repetition**
  - of one or a few specific terms e.g., free, cheap, sale, promotion, ...
  - Goal is to subvert if-idf ranking schemes

- **Dumping**
  - of a large number of unrelated terms
  - e.g., copy entire dictionaries

- **Weaving**
  - Copy legitimate pages and insert spam terms at random positions

- **Phrase Stitching**
  - Glue together sentences and phrases from different sources

**Term spam targets**

- Body of web page
- Title
- URL
- HTML meta tags
- Anchor text

The **tf–idf** weight (term frequency–inverse document frequency) is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus (a large and structured set of texts). The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf–idf weighting scheme are often used by search engines to score and rank a document’s relevance given a user query.
## Link spam

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

### Link Farms

- **Spammer’s goal**
  - Maximize the page rank of target page \( t \)
- **Technique**
  - Get as many links from accessible pages as possible to target page \( t \)
  - Construct “link farm” to get page rank multiplier effect

### Link Farms diagram

![Link Farms Diagram](attachment:link_farms_diagram.png)

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

Suppose rank contributed by accessible pages = x
Let page rank of target page = y
Rank of each “farm” page = \( \frac{\beta y}{M} + \frac{(1-\beta)}{N} \)
\[ y = x + \beta M \left( \frac{\beta y}{M} + \frac{(1-\beta)}{N} \right) + \frac{(1-\beta)}{N} \]
Very small; ignore
\[ y = x/(1-\beta^2) + cM/N \] where \( c = \frac{\beta}{(1+\beta)} \)

Analysis

\[ y = x/(1-\beta^2) + cM/N \] where \( c = \frac{\beta}{(1+\beta)} \)
For \( \beta = 0.85, 1/(1-\beta^2) = 3.6 \)
- Multiplier effect for “acquired” page rank
- By making M large, we can make y as large as we want

Hiding techniques

- Content hiding
  - Use same color for text and page background
- Cloaking
  - Return different page to crawlers and browsers
- Redirection
  - Alternative to cloaking
  - Redirects are followed by browsers but not crawlers
Detecting Spam

Term spamming
- Analyze text using statistical methods e.g., Naive Bayes classifiers
- Similar to email spam filtering
- Also useful: detecting approximate duplicate pages

Link spamming
- Open research area
- One approach: TrustRank

TrustRank idea

Basic principle: approximate isolation
- It is rare for a "good" page to point to a "bad" (spam) page

Sample a set of "seed pages" from the web
- Have an oracle (human) identify the good pages and the spam pages in the seed set
- Expensive task, so must make seed set as small as possible

Trust Propagation

Call the subset of seed pages that are identified as "good" the "trusted pages"
- Set trust of each trusted page to 1
- Propagate trust through links
  - Each page gets a trust value between 0 and 1
  - Use a threshold value and mark all pages below the trust threshold as spam
Rules for trust propagation

- Trust attenuation
  - The degree of trust conferred by a trusted page decreases with distance

- Trust splitting
  - The larger the number of outlinks from a page, the less scrutiny the page author gives each outlink
  - Trust is "split" across outlinks

Simple model

- Suppose trust of page p is t(p)
  - Set of outlinks O(p)
- For each q in O(p), p confers the trust
  - $\beta t(p)/|O(p)|$ for $0<\beta<1$
- Trust is additive
  - Trust of p is the sum of the trust conferred on p by all its inlinked pages
- Note similarity to Topic-Specific Page Rank
  - Within a scaling factor, trust rank = biased page rank with trusted pages as teleport set
Picking the seed set

- Two conflicting considerations
  - Human has to inspect each seed page, so seed set must be as small as possible
  - Must ensure every "good page" gets adequate trust rank, so need make all good pages reachable from seed set by short paths

Approaches to picking seed set

- Suppose we want to pick a seed set of k pages
- PageRank
  - Pick the top k pages by page rank
  - Assume high page rank pages are close to other highly ranked pages
  - We care more about high page rank "good" pages