Acknowledgements
Some slides in this lecture are adapted from Manning (Stanford), and Web Mining tutorial by Baeza Yates (SIGIR 2008).
Web Graph for Search

• Useful to look beyond the *content* of documents
  – Consider the hyperlinks between them

• Big questions:
  – Do the links provide *authority* to some pages? Is this *useful* for ranking, crawling, indexing?
  – Do links give more information about *content* of pages?

• Big application areas
  – The Web
  – Email
  – Social networks
Applications to Search

• Enhancing search
  – Scoring and ranking
  – Link-based clustering – topical structure from links
  – Links as features in classification – documents that link to one another are likely to be on the same subject

• Crawling
  – Based on the links seen, where do we crawl next?
The Web as a Directed Graph (Revisited)

Assumption 1: A hyperlink between pages denotes author perceived relevance (quality signal)

Assumption 2: The anchor of the hyperlink describes the target page (textual context)
Assumption 1: reputed pages

SIGIR Test of Time Awards

The SIGIR Test of Time Award recognizes research that has had long-lasting influence, including impact on a subarea of information retrieval research, across subareas of information retrieval research, and outside of the information retrieval research community (e.g. non-information retrieval research or industry). The winning paper is selected from the set of full papers presented at the main SIGIR conference 10-12 years before.

SIGIR 2015

Stuff I’ve seen: a system for personal information retrieval and re-use
Susan Dumais, Edward Cutrell, JJ Cudiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins
SIGIR 2003

Honorable Mentions

- Document clustering based on non-negative matrix factorization
  Wei Xu, Xin Liu, and Yihong Gong
  SIGIR 2003
- Automatic image annotation and retrieval using cross-media relevance models
  J. Jeon, V. Lavrenko, and R. Manmatha
  SIGIR 2003
- Modeling annotated data
  David M. Blei and Michael I. Jordan
  SIGIR 2003

http://sigir.org/awards/test-of-time-awards/
Assumption 2: annotation of target
Anchor Text

WWW Worm - McBryan [Mcbr94]

- For *ibm* how to distinguish between:
  - IBM’s home page (mostly graphical)
  - IBM’s copyright page (high term freq. for ‘ibm’)
  - Rival’s spam page (arbitrarily high term freq.)

A million pieces of anchor text with “ibm” send a strong signal

A million pieces of anchor text with “ibm” send a strong signal
Indexing anchor text

• When indexing a document $D$, include (with some weight) anchor text from links pointing to $D$. 

Armonk, NY-based computer giant IBM announced today 

Joe’s computer hardware links 
Sun HP IBM 

Big Blue today announced record profits for the quarter
Indexing anchor text

- Can sometimes have unexpected effects, e.g., spam, miserable failure
- Can score anchor text with weight depending on the authority of the anchor page’s website
  - E.g., if we were to assume that content from cnn.com or yahoo.com is authoritative, then trust (more) the anchor text from them
  - Increase the weight of off-site anchors (non-nepotistic scoring)
Example: Using Anchor Text

- https://www.google.com/advanced_search
- Browser
- Search engine
The Web graph

• Edges can be directed or undirected
• Graph is highly dynamic
  – Nodes and edges are added/deleted often
  – Content of existing nodes is also subject to change
  – Pages and hyperlinks created on the fly
• Apart from primary connected component there are also smaller disconnected components
Web Graph

http://www.touchgraph.com/TGGoogleBrowser.html
Web Graph: Statistics of Interest

• Size and connectivity of the graph
• Number of connected components
• Distribution of pages per site
• Distribution of incoming and outgoing connections per site
• Average and maximal length of the shortest path between any two vertices (diameter)
Degree distribution
Power Law Connectivity

- Distribution of number of connections per node follows a power law distribution.
- Study at Notre Dame University reported:
  - $\gamma = 2.45$ for outdegree distribution
  - $\gamma = 2.1$ for indegree distribution
- Random graphs have Poisson distribution if $p$ is large.
  - Decays exponentially fast to 0 as $k$ increases towards its maximum value $n-1$
Power Laws: A Curious Statistic About the Web

- Degree distributions of the web graph are distributed by the power law.
- Component size distributions are distributed by the power law.

![Graphs showing degree and component distributions.](image-url)
Small World Networks

• It is a ‘small world’
  – Millions of people. Yet, separated by “six degrees” of acquaintance relationships
  – Popularized by Milgram’s famous experiment

• Mathematically
  – Diameter of graph is small (log N) as compared to overall size
    • 3. Property seems interesting given ‘sparse’ nature of graph but ...
    • This property is ‘natural’ in ‘pure’ random graphs
The small world of WWW

- Empirical study of Web-graph reveals small-world property
  - Average distance (d) in simulated web:
    \[ d = 0.35 + 2.06 \log(n) \]
    e.g. \( n = 10^9 \), \( d \approx 19 \)
  - Graph generated using power-law model
  - Diameter properties inferred from sampling
    - Calculation of max. diameter computationally demanding for large values of \( n \)
Implications for Web

- Logarithmic scaling of diameter makes future growth of web manageable
  - 10-fold increase of web pages results in only 2 more additional ‘clicks’, but ...
  - Users may not take shortest path, may use bookmarks or just get distracted on the way
  - Therefore search engines play a crucial role
A large scale study (Altavista crawls) reveals interesting properties of web

- Study of 200 million nodes & 1.5 billion links
- Small-world property not applicable to entire web
  - Some parts unreachable
  - Others have long paths
- Power-law connectivity holds though
  - Page indegree ($\gamma = 2.1$), outdegree ($\gamma = 2.72$)
Empirical Numbers for Bow-tie

• Maximal minimal (?) diameter
  – 28 for SCC, 500 for entire graph

• Probability of a path between any 2 nodes
  – ~1 quarter (0.24)

• Average length
  – 16 (directed path exists), 7 (undirected)

• Shortest directed path between 2 nodes in SCC: 16-20 links on average
Web Graph Evolution

Baeza-Yates & Poblete, 2006
Web HTML Links! = citations

• Web links are different from citations:
  – Many links are navigational.
  – Many pages with high in-degree are portals not content providers.
  – Not all links are endorsements.
  – Company websites don’t point to their competitors.
  – Citations to relevant literature is enforced by peer-review.
WebSearch: Query-independent ordering

- First idea: using link counts as simple measures of popularity.
- Two basic suggestions:
  - Undirected popularity:
    - Each page gets a score = the number of in-links plus the number of out-links (3+2=5).
  - Directed popularity:
    - Score of a page = number of its in-links (3).
Query processing

• First retrieve all pages meeting the text query (say *venture capital*).
• Order these by their link popularity (either variant on the previous page).
Spamming simple popularity

• *Exercise*: How do you spam each of the following heuristics so your page gets a high score?
• Each page gets a score = the number of in-links plus the number of out-links.
• Score of a page = number of its in-links.
Many standard documents include *bibliographies* (or *references*), explicit *citations* to other previously published documents.

Using citations as links, standard corpora can be viewed as a graph.

The structure of this graph, independent of content, can provide interesting information about the similarity of documents and the structure of information.

*CiteSeer, DBLP: available.*
Impact Factor

• Developed by Garfield in 1972 to measure the importance (quality, influence) of scientific journals.
• Measure of how often papers in the journal are cited by other scientists.
• Computed and published annually by the Institute for Scientific Information (ISI).
• The *impact factor* of a journal \( J \) in year \( Y \) is the average number of citations (from indexed documents published in year \( Y \)) to a paper published in \( J \) in year \( Y-1 \) or \( Y-2 \).
• Does not account for the quality of the citing article.
Random (Markov) Surfer Model

• Imagine a web user doing a random walk on web pages:
  – Start at a random page
  – At each step, go out of the current page along one of the links on that page, with equal probability

• “In the steady state” each page has a long-term visit rate - use this as the page’s score.
  – Only works for “ergodic” chains (no dead-ends)
Not quite enough

• The web is full of dead-ends.
  – Random walk can get stuck in dead-ends.
  – Makes no sense to talk about long-term visit rates.
Add Teleportation

• Introduce a “random teleportation” probability, $E(p)$, that sends a surfer to any page with small (random) probability.

$R(p) = c \left( \sum_{q:q \to p} \frac{R(q)}{N_q} + E(p) \right)$
• For all $i$, \[ \sum_{j=1}^{n} P_{ij} = 1. \]

• Represent the teleporting random walk as a Markov chain, for this case:
Probability vectors

- A probability (row) vector $\mathbf{x} = (x_1, \ldots, x_n)$ tells us where the random walk is at any point.
- E.g., $(000\ldots1\ldots000)$ means we’re in state $i$.

More generally, the vector $\mathbf{x} = (x_1, \ldots, x_n)$ means the walk is in state $i$ with probability $x_i$.

$$\sum_{i=1}^{n} x_i = 1.$$
Change in probability vector

• If the probability vector is $\mathbf{x} = (x_1, \ldots, x_n)$ at this step, what is $\mathbf{x}'$ at the next step?

• Recall that row $i$ of the transition probability Matrix $\mathbf{P}$ tells us where we go next from state $i$.

• So from $\mathbf{x}$, our next state is computed as $\mathbf{x}\mathbf{P}$. 
Steady state example

- The steady state looks like a vector of probabilities $\mathbf{a} = (a_1, \ldots, a_n)$:
  - $a_i$ is the probability that we are in state $i$.

For this example, $a_1 = 1/4$ and $a_2 = 3/4$.
Iterative (Forward) Way of computing $a$

• Recall, regardless of where we start, we eventually reach the steady state $a$.

• Start with any distribution (say $x=(10...0)$).

• After one step, we’re at $xP$;

• after two steps at $xP^2$, then $xP^3$ and so on.

• “Eventually” means for “large” $k$, $xP^k = a$.

• Algorithm: multiply $x$ by increasing powers of $P$ until the product looks stable.
PageRank Algorithm

Let $S$ be the total set of pages.

Let $\forall p \in S: E(p) = \alpha / |S|$ (for some $0 < \alpha < 1$, e.g. 0.15)

Initialize $\forall p \in S: R(p) = 1 / |S|

Until ranks do not change (much) (convergence)

For each $p \in S$:

$$R'(p) = \left[ (1 - \alpha) \sum_{q: q \to p} \frac{R(q)}{N_q} \right] + E(p)$$

$$c = 1 / \sum_{p \in S} R'(p)$$

For each $p \in S$: $R(p) = cR'(p)$ (normalize)
Speed of Convergence

- Early experiments on Google used 322 million links.
- PageRank algorithm converged (within small tolerance) in about 52 iterations.
- Number of iterations required for convergence is empirically $O(\log n)$ (where $n$ is the number of links).
- Therefore calculation is quite efficient.
Simple Title Search with PageRank

• Use simple Boolean search to search web-page titles and rank the retrieved pages by their PageRank.

• Sample search for “university”:
  – Altavista returned a random set of pages with “university” in the title (seemed to prefer short URLs).
  – Primitive Google returned the home pages of top universities.
Which page has highest PageRank?

• c.f. 1997: Netscape!
• c.f. 2005: Wikipedia!
  – Maybe not (hard to measure externally)
• Some other sites with high PageRank
  – Google
  – MS Internet Explorer, Firefox homepages
Random Surfer Model

• Imagine a browser doing a random walk on web pages:
  – Start at a random page
  – At each step, go out of the current page along one of the links on that page, equiprobably

• “In the steady state” each page has a long-term visit rate - use this as the page’s score.
Not quite enough

• The web is full of dead-ends.
  – Random walk can get stuck in dead-ends.
  – Makes no sense to talk about long-term visit rates.
Another problem with Initial Idea

- A group of pages that only point to themselves but are pointed to by other pages act as a “rank sink” and absorb all the rank in the system.

Rank flows into cycle and can’t get out
• Introduce a “rank source” $E$ that continually replenishes the rank of each page, $p$, by a fixed amount $E(p)$.

\[
R(p) = c \left( \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} + E(p) \right)
\]
Result of teleporting

• Now cannot get stuck locally.
• There is a long-term rate at which any page is visited
• Reduces effect of sinks
• How do we compute this visit rate?
Markov chains

- A Markov chain consists of $n$ states, plus an $n \times n$ transition probability matrix $P$.
- At each step, we are in exactly one of the states.
- For $1 \leq i,j \leq n$, the matrix entry $P_{ij}$ tells us the probability of $j$ being the next state, given we are currently in state $i$. 

![Diagram of Markov chain](image)
Markov chains

- For all $i$, $\sum_{j=1}^{n} P_{ij} = 1$.
- Markov chains are abstractions of random walks.
- **Exercise**: represent the teleporting random walk as a Markov chain, for this case:

\[
\sum_{j=1}^{n} P_{ij} = 1
\]
Ergodic Markov chains

- A Markov chain is **ergodic** if
  - you have a path from any state to any other
  - you can be in any state at every time step, with non-zero probability.

![Diagram of ergodic and non-ergodic Markov chains]

Not ergodic (even/odd).
Ergodic Markov chains

• For any ergodic Markov chain, there is a unique long-term visit rate for each state.
  – *Steady-state distribution.*

• Over a long time-period, we visit each state in proportion to this rate.

• It doesn’t matter where we start.
A probability (row) vector $\mathbf{x} = (x_1, \ldots x_n)$ tells us where the walk is at any point.

E.g., (000...1...000) means we’re in state $i$.

More generally, the vector $\mathbf{x} = (x_1, \ldots x_n)$ means the walk is in state $i$ with probability $x_i$.

$$\sum_{i=1}^{n} x_i = 1.$$
Change in probability vector

• If the probability vector is $\mathbf{x} = (x_1, \ldots, x_n)$ at this step, what is it at the next step?

• Recall that row $i$ of the transition probability matrix $\mathbf{P}$ tells us where we go next from state $i$.

• So from $\mathbf{x}$, our next state is distributed as $\mathbf{xP}$. 
Steady state example

- The steady state looks like a vector of probabilities $a = (a_1, \ldots, a_n)$:
  - $a_i$ is the probability that we are in state $i$.

For this example, $a_1 = 1/4$ and $a_2 = 3/4$. 
How do we compute this vector?

• Let \( \mathbf{a} = (a_1, \ldots, a_n) \) denote the row vector of steady-state probabilities.

• If we our current position is described by \( \mathbf{a} \), then the next step is distributed as \( \mathbf{aP} \).

• But \( \mathbf{a} \) is the steady state, so \( \mathbf{a} = \mathbf{aP} \).

• Solving this matrix equation gives us \( \mathbf{a} \).
  – So \( \mathbf{a} \) is the (left) eigenvector for \( \mathbf{P} \).
  – (Corresponds to the “principal” eigenvector of \( \mathbf{P} \) with the largest eigenvalue.)
  – Transition probability matrices always have large eigenvalue 1.
One way of computing \( a \)

- Recall, regardless of where we start, we eventually reach the steady state \( a \).
- Start with any distribution (say \( x=(10\ldots0) \)).
- After one step, we’re at \( xP \);
- after two steps at \( xP^2 \), then \( xP^3 \) and so on.
- “Eventually” means for “large” \( k \), \( xP^k = a \).
- Algorithm: multiply \( x \) by increasing powers of \( P \) until the product looks stable.
Let $S$ be the total set of pages.

Let $\forall p \in S: E(p) = \alpha/|S|$ (for some $0<\alpha<1$, e.g. 0.15)

Initialize $\forall p \in S: R(p) = 1/|S|$

Until ranks do not change (much) (**convergence**)

For each $p \in S$:

$$R'(p) = \left((1-\alpha) \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}\right) + E(p)$$

$$c = 1/\sum_{p \in S} R'(p)$$

For each $p \in S$: $R(p) = cR'(p)$ (**normalize**)
Speed of Convergence

• Early experiments on Google used 322 million links.
• PageRank algorithm converged (within small tolerance) in about 52 iterations.
• Number of iterations required for convergence is empirically $O(\log n)$ (where $n$ is the number of links).
• Therefore calculation is quite efficient.
Simple Title Search with PageRank

• Use simple Boolean search to search web-page titles and rank the retrieved pages by their PageRank.

• Sample search for “university”:
  – Altavista returned a random set of pages with “university” in the title (seemed to prefer short URLs).
  – Primitive Google returned the home pages of top universities.
Which page has highest PageRank?

- c.f. 1997: Netscape!
  - Maybe not (hard to measure externally)
- Some other sites with high PageRank
  - Google
  - MS Internet Explorer, Firefox homepages
Pagerank: Issues and Variants

• How realistic is the random surfer model?
  – What if we modeled the back button? [Fagi00]
  – Surfer behavior sharply skewed towards short paths [Hube98]
  – Search engines, bookmarks & directories make jumps non-random.

• Biased Surfer Models
  – Weight edge traversal probabilities based on match with topic/query (non-uniform edge selection)
  – Bias jumps to pages on topic (e.g., based on personal bookmarks & categories of interest)
Hyperlink-Induced Topic Search (HITS) - Klei98

- In response to a **query**, instead of an ordered list of pages each meeting the query, find **two** sets of inter-related pages:
  - *Hub pages* are good lists of links on a subject.
    - e.g., “Bob’s list of cancer-related links.”
  - *Authority pages* occur recurrently on good hubs for the subject.
- Best suited for “broad topic” queries rather than for page-finding queries.
- Gets at a broader slice of common *opinion*. 
Hubs and Authorities

• Thus, a good hub page for a topic *points* to many authoritative pages for that topic.
• A good authority page for a topic is *pointed* to by many good hubs for that topic.
• Circular definition - will turn this into an iterative computation.
The hope

Long distance telephone companies
High-level scheme

• Extract from the web a base set of pages that could be good hubs or authorities.
• From these, identify a small set of top hub and authority pages;
  → iterative algorithm.
Base set

- Given text query (say *browser*), use a text index to get all pages containing *browser*.
  - Call this the **root set** of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the **base set**.
Visualization
Assembling the base set [Klei98]

• Root set typically 200-1000 nodes.
• Base set may have up to 5000 nodes.
• How do you find the base set nodes?
  – Follow out-links by parsing root set pages.
  – Get in-links (and out-links) from a connectivity server.
  – (Actually, suffices to text-index strings of the form \( \text{href=“URL”} \) to get in-links to \( \text{URL} \).)
Distilling hubs and authorities

• Compute, for each page $x$ in the base set, a **hub score** $h(x)$ and an **authority score** $a(x)$.

• Initialize: for all $x$, $h(x) \leftarrow 1; a(x) \leftarrow 1$;

• Iteratively update all $h(x), a(x)$;

• After iterations
  – output pages with highest $h()$ scores as top hubs
  – highest $a()$ scores as top authorities.
Iterative update

- Repeat the following updates, for all $x$:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$
Scaling

• To prevent the $h()$ and $a()$ values from getting too big, can scale down after each iteration.

• Scaling factor doesn’t really matter:
  – we only care about the relative values of the scores.
How many iterations?

• Claim: relative values of scores will converge after a few iterations:
  – in fact, suitably scaled, $h()$ and $a()$ scores settle into a steady state!

• We only require the relative orders of the $h()$ and $a()$ scores - not their absolute values.

• In practice, ~5 iterations get you close to stability.
Japan Elementary Schools

Hubs

• schools
• LINK Page-13
• “ú—¶šswž
• æ‰„šSwźffyffW
• 100 Schools Home Pages (English)
• K-12 from Japan 10/...rnet and Education )
• http://www...iglobe.ne.jp/~IKESAN
• ,l,fjšSwź,UN,PG•°Œê
• Ðš—‘—šÕš—“Œššwž
• Koulušutu ja oppilaitokset
• TOYODA HOMEAGE
• Education
• Cay's Homepage(Japanese)
• —γ“ššwž,ffyffW
• UNIVERSITY
• %oo—ššwž DRAGON97-TOP
• Æ‰„šSwźT”N,PFfzffyffW
• µ°é%Å© ¥á¥¥á¾¥á¥¥á¾

Authorities

• The American School in Japan
• The Link Page
• %aèš—§ª„šSwźffyffW
• Kids' Space
• Ôåš—§ª“ÀššW•‰šSwź
• {îš³¬ššw•¥ššwž
• KEIMEI GAKUEN Home Page ( Japanese )
• Shiranuma Home Page
• fuzoku-es.fukui-u.ac.jp
• welcome to Miasa E&J school
• ã“µššEš‰oi•š—§’†ššW,šfy
• http://www...p/~m_maru/index.html
• fukui haruyama-es HomePage
• Torisu primary school
• goo
• Yakumo Elementary,Hokkaido,Japan
• FUZOKU Home Page
• Kamishibun Elementary School...
Things to note

• Pulled together good pages regardless of language of page content.
• Use only link analysis after base set assembled
  – iterative scoring is query-independent.
• Iterative computation after text index retrieval - significant overhead.
Issues

• **Topic Drift**
  – Off-topic pages can cause off-topic “authorities” to be returned
    • E.g., the neighborhood graph can be about a “super topic”

• **Mutually Reinforcing Affiliates**
  – Affiliated pages/sites can boost each others’ scores
    • Linkage between affiliated pages is not a useful signal
Resources

Beyond PageRank

[Richardson et al., WWW2006]

• Dynamic Ranking (query dependent)
  – Give the best answers for a query

• Static Ranking (query independent)
  – Order Web pages before you have a query
  – Growth of Web → many bad pages to ignore
Web Page Ranking

• Dynamic Ranking (query dependent)
  – Give the best answers for a query

• Static Ranking (query independent)
  – Order Web pages before you have a query
  – Growth of Web → many bad pages to ignore
A Search Engine

Web → Crawl → Build Index → Answer Queries

- Which pages to crawl
- Efficient index order
- Informs dynamic ranking

Static Rank
In Search of a Good Static Rank

• PageRank
  – Not as effective as expected
    • Homepages, good companies [Upstill et al. 2003]
    • TREC Web/VLC competitions [Hawking & Craswell]
  – Computationally intensive
  – Ignores page content
  – Difficult to incorporate other features
PageRank...
PageRank...
fRank

Web

- Words on page
- # Inlinks
- Contains ‘Viagra’
- PageRank

Machine Learning Model

fRank
Machine Learning for Static Ranking

- Proposal: apply machine learning
  - Reactive to new spam techniques
  - Can use advances in machine learning
    - Adversarial learning
    - Outlier detection
  - Useful in intranet domains

- Note: PageRank can be an input!
Features

- Popularity
- Anchor text and inlinks
- Page
- Domain
- PageRank
Features: Popularity

- Data from MSN Toolbar
- Smoothed

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact URL</td>
<td>cnn.com/2005/tech/wikipedia.html?v=mobile</td>
</tr>
<tr>
<td>No Params</td>
<td>cnn.com/2005/tech/wikipedia.html</td>
</tr>
<tr>
<td>Page</td>
<td>wikipedia.html</td>
</tr>
<tr>
<td>URL-1</td>
<td>cnn.com/2005/tech</td>
</tr>
<tr>
<td>URL-2</td>
<td>cnn.com/2005</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>cnn.com</td>
</tr>
<tr>
<td>Domain+1</td>
<td>cnn.com/2005</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Features: Anchor, Page, Domain

• Anchor text and inlinks
  – Total amount of anchor text, unique anchor text words, number of inlinks, etc.

• Page
  – 8 Features based on page alone: Words in body, frequency of most common term, etc.

• Domain
  – Averages in domain: average #outlinks, etc.
Features: PageRank

• Computed on 5 billion pages
  – Most previous studies use much smaller corpora
• Standard settings
Data

• Human judgments
  1. Randomly choose query from MSN users
  2. Chose top URLs by search engine
  3. Rate quality of URL for that query
• 500k (Query,URL,Rating) tuples
• Judged URLs biased to good pages
  – Results apply to index ordering, relevance
  – Crawl ordering requires unbiased sample
Becoming Query Independent

• \((\text{Query,URL,Rating}) \rightarrow (\text{URL,Rating})\)
• Take maximum rating for each URL
  – Good page if relevant for at least one query
• Queries are common \(\rightarrow\) likely correct index order and relevance order
Measure

• Goal: Find static ranking algorithm that most correctly reproduces judged order

\[
\text{pairwise accuracy} = \frac{|H_p \cap S_p|}{|H_p|}
\]

• Fraction of pairs that, when the humans claim one is better than the other, the static rank algorithm orders them correctly
Results

- fRank significantly outperforms PageRank

<table>
<thead>
<tr>
<th>Technique</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Baseline)</td>
<td>50.00</td>
</tr>
<tr>
<td>PageRank</td>
<td>56.70</td>
</tr>
<tr>
<td>fRank</td>
<td>67.43</td>
</tr>
</tbody>
</table>
Accuracy of Each Feature Set

- Accuracy with only one given feature set

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>56.70</td>
</tr>
<tr>
<td>Popularity</td>
<td>60.82</td>
</tr>
<tr>
<td>Anchor</td>
<td>59.09</td>
</tr>
<tr>
<td>Page</td>
<td>63.93</td>
</tr>
<tr>
<td>Domain</td>
<td>59.03</td>
</tr>
<tr>
<td>All Features</td>
<td>67.43</td>
</tr>
</tbody>
</table>
Accuracy of Each Feature Set

- Accuracy with only the given feature set
- Every feature set outperformed PageRank
- Best feature sets contain no link information

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>56.70</td>
</tr>
<tr>
<td>Popularity</td>
<td>60.82</td>
</tr>
<tr>
<td>Anchor</td>
<td>59.09</td>
</tr>
<tr>
<td>Page</td>
<td>63.93</td>
</tr>
<tr>
<td>Domain</td>
<td>59.03</td>
</tr>
<tr>
<td>All Features</td>
<td>67.43</td>
</tr>
</tbody>
</table>
Ignoring Link Structure

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageRank</td>
<td>56.70</td>
</tr>
<tr>
<td>fRank</td>
<td>67.43</td>
</tr>
<tr>
<td>fRank without PageRank</td>
<td>67.25</td>
</tr>
<tr>
<td>fRank without PageRank, Anchor, Domain</td>
<td>66.83</td>
</tr>
</tbody>
</table>
Speed

• fRank
  – Linear pass through 5B documents

• PageRank
  – Multiple passes through 370B links

• fRank approximately 100 times faster than PageRank
### Qualitative Evaluation

- **Top ten URLs for PageRank vs. fRank**

<table>
<thead>
<tr>
<th>PageRank</th>
<th>fRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>google.com</td>
</tr>
<tr>
<td>apple.com/quicktime/download</td>
<td>yahoo.com</td>
</tr>
<tr>
<td>amazon.com</td>
<td>americanexpress.com</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>hp.com</td>
</tr>
<tr>
<td>microsoft.com/windows/ie</td>
<td>target.com</td>
</tr>
<tr>
<td>apple.com/quicktime</td>
<td>bestbuy.com</td>
</tr>
<tr>
<td>mapquest.com</td>
<td>dell.com</td>
</tr>
<tr>
<td>ebay.com</td>
<td>autotrader.com</td>
</tr>
<tr>
<td>mozilla.org/products/firefox</td>
<td>dogpile.com</td>
</tr>
<tr>
<td>ftc.gov</td>
<td>bankofamerica.com</td>
</tr>
</tbody>
</table>
Qualitative Evaluation

- Top ten URLs for PageRank vs. fRank

<table>
<thead>
<tr>
<th>PageRank</th>
<th>fRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>google.com</td>
</tr>
<tr>
<td>apple.com/quicktime/download</td>
<td>yahoo.com</td>
</tr>
<tr>
<td>amazon.com</td>
<td>americanexpress.com</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>hp.com</td>
</tr>
<tr>
<td>microsoft.com/windows/ie</td>
<td>target.com</td>
</tr>
<tr>
<td>apple.com/quicktime</td>
<td>bestbuy.com</td>
</tr>
<tr>
<td>mapquest.com</td>
<td>dell.com</td>
</tr>
<tr>
<td>ebay.com</td>
<td>autotrader.com</td>
</tr>
<tr>
<td>mozilla.org/products/firefox</td>
<td>dogpile.com</td>
</tr>
<tr>
<td>ftc.gov</td>
<td>bankofamerica.com</td>
</tr>
</tbody>
</table>

Technology Oriented        Consumer Oriented
Qualitative Evaluation

- Top ten URLs for PageRank vs. fRank

<table>
<thead>
<tr>
<th>PageRank</th>
<th>fRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>google.com</td>
</tr>
<tr>
<td>apple.com/quicktime/download</td>
<td>yahoo.com</td>
</tr>
<tr>
<td>amazon.com</td>
<td>americanexpress.com</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>hp.com</td>
</tr>
<tr>
<td>microsoft.com/windows/ie</td>
<td>target.com</td>
</tr>
<tr>
<td>apple.com/quicktime</td>
<td>bestbuy.com</td>
</tr>
<tr>
<td>mapquest.com</td>
<td>dell.com</td>
</tr>
<tr>
<td>ebay.com</td>
<td>autotrader.com</td>
</tr>
<tr>
<td>mozilla.org/products/firefox</td>
<td>dogpile.com</td>
</tr>
<tr>
<td>ftc.gov</td>
<td>bankofamerica.com</td>
</tr>
</tbody>
</table>

- PageRank bias: **Web authors**
- fRank bias: **Web users**
Summary Re: PageRank

• Static ranking provides key value
  – Crawl priority, index efficiency, result relevance

• PageRank alone has low accuracy

• Machine learning with many features (fRank)
  – Significantly outperforms PageRank
  – May leverage developments in machine learning

• Much more to do for further improvements