How does Google crawl the Web?

**Inverted Indexing**

- **Term List**
  - Fruitvale Station
  - American
  - and
  - directed
  - drama

- **Posting Lists**
  - a
  - (1, [21, 96, 103, ...], [2, 1])
  - american
  - (1, [28, 123], [5, ...])
  - and
  - (1, [70, 157, ...], [2, ...])
  - directed
  - (1, [61, 212, ...], [4, ...])
  - drama
  - (1, [38, 87, ...], [16, ...])

**Inverted Index**:

1. Fruitvale Station
2. [List of postings]

**INFORMATION RETRIEVAL: RANKING**

How Does Google Get Such Good Results?

- Two Sides to Ranking: Relevance
  - Relevant results are displayed first.
  - Non-relevant results are displayed later.

- Google uses a combination of factors to rank pages, including relevance, authority, and freshness.
Two Sides to Ranking: Importance

Example Query: move freedom wallace

Matches in a Document:
- Freedom: 7 occurrences
- Movie: 16 occurrences
- Wallace: 88 occurrences
Usefulness of Words

- movie: occurs very frequently
- freedom: occurs frequently
- wallace: occurs occasionally

Estimating Relevance

- Rare words more important than common words
  - wallace (49M) more important than freedom (198M)
    more important than movie (835M)

- Words occurring more frequently in a document indicate higher relevance
  - wallace (88) more matches than movie (16) more
    matches than freedom (7)

Relevance Measure: **TF–IDF**

- **TF**: Term Frequency
  - Measures occurrences of a term in a document
  - $tf(t, d) = \text{count}(t, d)$
    - Raw count of occurrences
    - Logarithmically scaled
    - Normalised by document length
    - $tf(t, d) = \frac{\text{count}(t, d)}{\text{count}(\text{all terms in document})}$
    - A combination / something else 😐

- **IDF**: Inverse Document Frequency
  - Measures how rare/common a term is across all documents
  - $idf(t, D) = \log\left(\frac{|D|}{\text{count}(t, D)}\right)$
    - Logarithmically scaled document occurrences

Relevance Measure: **TF–IDF**

- **TF–IDF**: Combine Term Frequency and Inverse Document Frequency:
  - $tf-idf(t, d) = tf(t, d) \times idf(t, D)$

- Score for a query
  - Let query: $q = (t_1, \ldots, t_n)$
  - Score for a query: $score(q, d) = \sum_{t \in q} tf-idf(t, d)$
Relevance Measure: TF-IDF

Term Frequency

\[ tf(t, d) = \frac{1}{|D|} \sum_{d \in D} t \cdot d \]

Inverse Document Frequency

\[ idf(d) = \log \left( \frac{|D|}{|\{d \in D : t \in d\}|} \right) \]

<table>
<thead>
<tr>
<th>t</th>
<th>tf(t, d)</th>
<th>idf(d)</th>
<th>( tf(t, d) \times idf(d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>movie</td>
<td>16</td>
<td>835,000,000</td>
<td>33.66</td>
</tr>
<tr>
<td>freedom</td>
<td>7</td>
<td>189,000,000</td>
<td>57.62</td>
</tr>
<tr>
<td>wallace</td>
<td>43</td>
<td>40,200,000</td>
<td>231.91</td>
</tr>
</tbody>
</table>

Google: the

| D | 11, 410, 000, 000 |

score(q, d) = \sum_{t \in q} tf(t, d) \times idf(d)

score((movie, freedom, wallace), http://en.wikipedia.org/Braveheart) \approx 430.17
Vector Space Model (a mention)

\[ l = \sqrt{\sum_{c \in Q} tf(t, d)^2} \]

Dividing by \( l \) normalises length of vector to 1

Two Sides to Ranking: Relevance

Field-Based Boosting

- Not all text is equal: titles, headers, etc.

- Cosine Similarity
  \[ \text{sim}(d, d') = \frac{\langle d, d' \rangle}{\|d\| \cdot \|d'\|} \]

- Note:
  \[ a \cdot b = |a| \cdot |b| \cdot \cos(\angle(a, b)) \]
  \[ |\langle d \rangle| = \|d\| = 1 \]
  Hence the similarity is the cosine of the angle between the vectors
Anchor Text

- See how the Web views/tags a page

Information Retrieval & Relevance

Apache to the rescue again!

**Lucene: An Inverted Index Engine**

- Open Source Java Project
- Will play with it in the labs

**RANKING: IMPORTANCE**

Two Sides to Ranking: Importance

Link Analysis

Which will have more links: Barack Obama's Wikipedia Page or Mount Obama's Wikipedia Page?
Link Analysis

- Consider links as votes of confidence in a page
- A hyperlink is the open Web's version of...

(... even if the page is linked in a negative way.)

So if we just count the number of inlinks a web-page receives we know its importance, right?

Link Spamming

Which is more "important": a link from Barack Obama's Wikipedia page or a link from buyv1agra.com?

Link Importance

PageRank

- Not just a count of inlinks
  - A link from a more important page is more important
  - A link from a page with fewer links is more important
  - A page with lots of inlinks from important pages (which have few outlinks) is more important
PageRank is Recursive

PageRank Model

- The Web: a directed graph

\[ G = (V, E) \]

Vertices (pages)

Edges (links)

out(v) := \{v' \in V \mid (v, v') \in E\}

\[ \text{in}(v) := \{v' \in V \mid (v', v) \in E\} \]

\[ \text{rank}_0(v) = \frac{1}{|V|} \]

\[ \text{rank}_i(v) = \frac{1}{|\text{out}(v)|} \sum_{v' \in \text{in}(v)} \text{rank}_{i-1}(v') \]

Which is the most “important” vertex?

PageRank Model

- The Web: a directed graph

\[ G = (V, E) \]

Vertices (pages)

Edges (links)

\[ \text{rank}_0(a) = \frac{1}{3} \]

\[ \text{rank}_0(b) = \frac{1}{3} \]

\[ \text{rank}_0(c) = \frac{1}{3} \]

\[ \text{rank}_0(d) = \frac{1}{3} \]

\[ \text{rank}_0(e) = \frac{1}{3} \]

\[ \text{rank}_0(f) = \frac{1}{3} \]

PageRank: Random Surfer Model

- What is the probability of being at page x after n hops?
PageRank: Random Surfer Model

- someone surfing the web, clicking links randomly

• What is the probability of being at page \( x \) after \( n \) hops?
• Initial state: surfer equally likely to start at any node

PageRank applied iteratively for each hop: score indicates probability of being at that page after many hops

If the surfer reaches a page without links, the surfer randomly jumps to another page

What would happen with \( g \) over time?

PageRank: Random Surfer Model

- someone surfing the web, clicking links randomly

• What is the probability of being at page \( x \) after \( n \) hops?
• Initial state: surfer equally likely to start at any node

PageRank applied iteratively for each hop: score indicates probability of being at that page after many hops

If the surfer reaches a page without links, the surfer randomly jumps to another page

What would happen with \( g \) and \( i \) over time?

PageRank: Random Surfer Model

- someone surfing the web, clicking links randomly

• What is the probability of being at page \( x \) after \( n \) hops?
• Initial state: surfer equally likely to start at any node

PageRank applied iteratively for each hop: score indicates probability of being at that page after many hops

If the surfer reaches a page without links, the surfer randomly jumps to another page

• The surfer will jump to a random page at any time with a probability \( 1 - d \); this avoids trap and ensures convergence!
**PageRank Model: Final Version**

- The Web: a directed graph

\[ G = (V, E) \]

- Vertices: Pages
- Edges: Links

\[
\begin{align*}
\text{out}(v) &= \{ e' \in V \mid (v, e') \in E \} \\
\text{in}(v) &= \{ e' \in V \mid (e', v) \in E \} \\
\text{rank}_0(v) &= \frac{1}{|V|} \\
V^0 &= \{ v \in V : \text{out}(v) = 0 \} \\
V^m &= \{ v \in V : |\text{out}(v)| \neq 0 \}
\end{align*}
\]

\[ \text{rank}_i(v) = \frac{1}{d} \sum_{e \in \text{out}(v)} \text{rank}_{i-1}(v_e) + \frac{(1-d)}{|V|} \sum_{e \in \text{out}(v)} \text{rank}_0(v_e) \]

**PageRank: Benefits**

- More robust than a simple link count
- Scalable to approximate (for sparse graphs)
- Convergence guaranteed

**Two Sides to Ranking: Importance**

**INFORMATION RETRIEVAL: RECAP**

**How Does Google Get Such Good Results?**

**Ranking in Information Retrieval**

- **Relevance**: Is the document relevant for the query?
  - Term Frequency * Inverse Document Frequency
  - Touched on Cosine similarity

- **Importance**: Is the document an important/prominent one?
  - Links analysis
  - PageRank
CLASS PROJECTS

Class Project
- Done in pairs (typically)
- Goal: Use what you’ve learned to do something cool (basically)
- Expected difficulty: A bit more than a lab’s worth
  - But without guidance (can extend lab code)
- Marked on: Difficulty, appropriateness, scale, good use of techniques, presentation, coolness
  - Ambition is appreciated, even if you don’t succeed: feel free to bite off more than you can chew!
- Process:
  - Pair up (default random) by Wednesday, the end of the lab
  - Start thinking up topics
  - If you need data or get stuck, I will (try to) help out
- Deliverables: 5 minute presentation & 3-page report

Datasets to play with
- Wikipedia information
- IMDb (including ratings, directors, etc.)
- ArnetMiner (CS research papers w/ citations)
- Wikidata (like Wikipedia for data!)
- Twitter
- World Bank
- Find others, e.g., at http://datahub.io/

Course Marking
- 45% for Weekly Labs (~3% a lab!)
- 35% for Final Exam
- 20% for Small Class Project

Datasets to play with
- Wikipedia information
- IMDb (including ratings, directors, etc.)
- ArnetMiner (CS research papers w/ citations)
- Wikidata (like Wikipedia for data!)
- Twitter
- World Bank
- Find others, e.g., at http://datahub.io/
Next Week (May 4th, 6th)

- No official classes or labs next week
- but ...
- Good opportunity to meet with your lab partner to explore project ideas!
- Deadline for finding a topic: May 13th