Lecture Objectives

- **Learn** about:
  - Working with Massive data
  - Link analysis (PageRank)
  - Anchor text
The Web Document Collection

- Huge / Massive
- Graph / Connected
- No design/co-ordination
- Distributed content publishing
- Content includes truth, lies, obsolete information, contradictions …
- Unstructured (text, html, …), semi-structured (XML, annotated photos), structured (DB) …
- Growth – slowed down from initial “volume doubling every few months” but still expanding
- Content can be dynamically generated

Effect of Massive data

- Web search engines work with huge amount of data
  - 20 PB/day in 2008 → 160 PB/day in 2013 → now??
  - 1 PB = 1,000 TB = 1,000,000 GB
- How this would affect a search engine?
  - Very challenging (storage, processing, networking, …)
  - Very useful still (makes stuff easier), how?
- Assume two good search engines the collects two sub-sets of the web
  - Search engine A collected N docs → precision@10 = 40%
  - Search engine B collected 4N docs → precision@10??
**Effect of Massive data on Precision**

- Assume two good search engines that collect two sub-sets of the web
  - Search engine A collected N docs → precision@10 = 40%
  - Search engine B collected 4N docs → precision@10??
    - Distribution of positive/negative scores stays the same
    - Precision/Recall at a given score stays the same
    - In any decent IR system: more relevant docs exist at the top → $P@n \uparrow \Rightarrow$ precision@10 = 60% (increases)

**Big Data or Clever Algorithm?**

- For Web search, larger index usually would beat a better retrieval algorithm
  - Google Index vs Bing Index
- Similar to other applications
  - Google MT vs IBM MT
    - Statistical methods trained over 10x training data beat deep NLP methods with 1x training data
  - In general ML, the more data, the better the results
    - Tweets classification: using 100x of noisy training data beats 1x of well prepared training data, even with absence of stemming & stopping
  - Question answering task:
    - IBM Watson vs Microsoft experiment
Big Data or Clever Algorithm?

- Question answering task:
  - **Q**: Who created the character of Scrooge?
  - **A**: Scrooge, introduced by Charles Dickens in “A Christmas Carol”
- Requires heavy linguistic analysis, lots of research in TREC

- 2002, Microsoft
  - Identify (subj verb obj), rewrite as queries:
    - Q1: “created the character of Scrooge”
    - Q2: “the character of Scrooge was created by”
  - Search the web for exact phrase, get top 500 results
  - Extract phrase: ■Q1 or Q2■, get most frequent ■
  - Very naive approach, ignores most answers patterns
  - Who cares!! Web is huge, you will find matches anyway

Search “Microsoft”

**Doc1**
- Microsoft.com
- “Microsoft” mentioned 5 times

**Doc2**
- Tutorial.com
  - Tutorial on MS word
- “Microsoft” mentioned 35 times
**The Web as a Directed Graph**

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

Assumption 2: The text in the anchor of the hyperlink describes the target page (textual context)

---

**Links between Pages**

- Google Description of PageRank:
  - Relies on the "uniquely democratic" nature of the web
  - Interprets a link from page A to page B as "a vote"
- A → B: means A thinks B worth something
  - "wisdom of the crowds": many links means B must be good
  - Content-independent measure of quality of B
- Use as ranking feature, combined with content
  - But not all pages that link to B are of equal importance!
    - Importance of a link from CNN >>> link from blog page
- Google PageRank, 1998
  - How many “good” pages link to B?
Search “Microsoft”

Doc1
Microsoft.com
“Microsoft” mentioned 5 times

Doc2
Tutorial.com
Tutorial on MS word
“Microsoft” mentioned 35 times

PageRank: Random Surfer

• Analogy:
  • User starts browsing at a random page
  • Pick a random outgoing link → goes there → repeat forever
  • Example: G → E → F → E → D → B → C
  • With probability 1-\(\lambda\) jump to a random page
    • Otherwise, can get stuck forever A, or B ↔ C

• PageRank of page x
  • Probability of being at page x at a random moment in time
PageRank: Algorithm

- Initialize $PR_0(x) = \frac{100\%}{N}$
  - $N$: total number of pages
  - $PR_0(A) = .. = PR_0(K) = \frac{100\%}{11} = 9.1\%$

- For every page $x$
  $$PR_{t+1}(x) = \frac{1 - \lambda}{N} + \lambda \sum_{y \rightarrow x} PR_t(y) \frac{L_{out}(y)}{L_{out}(x)}$$

- $y \rightarrow x$ contributes part of its PR to $x$
- Spread PR equally among out-links
- Iterate till converge $\rightarrow$ PR scores should sum to 100%

PageRank: Example

- Let $\lambda = 0.82$

- $PR(B) = \frac{0.18}{11} + 0.82 \times \{PR(C) + \frac{1}{2} PR(D) + \frac{1}{3} PR(E) + \frac{1}{2} PR(F) + \frac{1}{2} PR(G) + \frac{1}{2} PR(H) + \frac{1}{2} PR(I) \}$
  $\approx 0.31 = 31\%$

- $PR(C) = \frac{0.18}{11} + 0.82 \times PR(B)$
  $= 0.18 \times 9.1\% + 0.82 \times 9.1\%$
  $= 9.1\%$

- $PR_{t+1}(C) = 0.18 \times 9.1\% + 0.82 \times 31\%$
  $\approx 26\%$
### PageRank: Example result

- Algorithm converges after few iterations

![Diagram showing PageRank example](image)

- Observations
  - Pages with no inlinks: \( PR = (1 - \lambda) / N = 0.18/11 = 1.6\% \)
  - Same (or symmetric) inlinks \( \rightarrow \) same PR (e.g. D and F)
  - One inlink from high PR >> many from low PR (e.g. C vs E)

### Anchor Text

- Anchor Text (text of a link):
  - Description of destination page
  - Short, descriptive like a query
  - Re-formulated in different ways
    - Human “query expansion”
  - Used when indexing page content
    - Add text of all anchor text linking the page
    - Different weights for different anchor text
      - Weighted according to PR of linking page
  - Significantly improves retrieval

![Anchor Text example](image)

**International Business Machines** announced today a deal of $100M ...

We support:
- Sun
- HP
- IBM

**Big Blue** today announced record profits for the quarter

**www.ibm.com**
**Link Spam**

- Trackback links (blogs that link to me)
  - Based on `$_HTTP_REFERER`
  - Artificial feedback loops
    - Similar to “follow back” in Twitter
- Links from comments on sites with high PR
  - Links in comments on CNN
  - One solution: insert `rel=nofollow` into links
    - Link ignored when computing PR
- Link farms
  - Fake densely-connected graph
  - Hundreds of web domains / IPs can be hosted on one machine

**The Reality**

- PageRank is used in Google, but is hardly the full story of ranking
  - A big hit when initially proposed, but just one feature now
  - Many sophisticated features are used
  - Machine-learned ranking heavily used
    - Learning to Rank (L2R)
    - Many features are used, including PR
    - Still counted as a very useful feature
Summary

• Web data is massive
  • Challenging for efficiency, but useful for effectiveness

• PageRank:
  • Probability than random surfer is currently on page x
  • The more powerful pages linking to x, the higher the PR

• Anchor text:
  • Short concise description of target page content
  • Very useful for retrieval

• Link Spam
  • Trackable links, link farms

Resources

• Text book 1: Intro to IR, Section 21.1
• Text Book 2: IR in Practice: 4.5, 10.3
• Page Rank Paper:

• Additional reading:

• YouTube Video: How Search Works
  https://www.youtube.com/watch?v=BNHR6IQJGZs