CS3245

Information Retrieval

12

Lecture 12: Crawling and Link Analysis

Last Time

Chapter 11

1. Probabilistic Approach to Retrieval / Basic Probability Theory
2. Probability Ranking Principle
3. OKAPI BM25

Chapter 12

1. Language Models for IR

Today

Chapter 20

- Crawling

Chapter 21

- Anchor Text
- PageRank

CRAWLING

Copyright violations

Chilling Effects

Home | Weather Reports | Report Receiving Sending a C&D Notice | Search the Database | Topics

Topic Home | FAQs

Monitoring the legal climate for Internet activity

Chilling Effects Clearinghouse > Piracy or Copyright Infringement > Frequently Asked Questions

Frequently Asked Questions (and Answers) about Piracy or Copyright Infringement

http://chillingeffects.org/piracy/faq.cgi

- Q: What is the purpose of copyright law?
- Q: If I am accused of "piracy," what does this mean?
- Q: Is all copying piracy?
- Q: Why is "piracy" such a big issue now?
- Q: My website contains a disclaimer that clearly states that I do not support or promote copyright infringment. Will this protect me?
- Q: Why are copyright holders concerned about piracy?
- Q: What are the penalties for copyright infringement, such as making infringing copies of software?
- Q: I run a website but I never actually upload or download copyrighted materials. Could I be liable for what visitors to my site do?
- Q: What is vicarious liability?
- Q: What is contributory infringement?
- Q: So am I better off not monitoring my website if I want to avoid contributory infringement liability?
- Q: Am I protected by Digital Millennium Copyright Act's Safe Harbor?
- Q: Can I copy or distribute software that is out of print and has been abandoned for years?
- Q: Aren
- Q: Isn't sending my friend a music file from a CD I already own just like loaning her the physical CD?
- Q: Aren't I allowed to make a backup copy of my software?

What any crawler should do

- Be capable of distributed operation
- Be scalable: need to be able to increase crawl rate by adding more machines
- Fetch pages of higher quality first
- Continuous operation: get fresh version of already crawled pages

How hard can crawling be?

- Web search engines must crawl their documents.
- Getting the content of the documents is easier for many other IR systems.
  - E.g., indexing all files on your hard disk: just do a recursive descent on your file system
- Ok: for web IR, getting the content of the documents takes longer . . .

  . . . because of latency.
- But is that really a design/systems challenge?

Basic crawler operation

- Initialize queue with URLs of known seed pages
- Repeat
  - Take URL from queue
  - Fetch and parse page
  - Extract URLs from page
  - Add URLs to queue
- Fundamental assumption: The web is well linked.

What's wrong with this crawler?

```
urlqueue := (some carefully selected set of seed urls)
while urlqueue is not empty:
myurl := urlqueue.getlastanddelete()
mypage := myurl.fetch()
fetchedurls.add(myurl)
newurls := mypage.extracturls()
for myurl in newurls:
if myurl not in fetchedurls and not in urlqueue:
urlqueue.add(myurl)
addtoinvertedindex(mypage)
```

What's wrong with the simple crawler

- Scale: we need to distribute.
- We can't index everything: we need to subselect. How?
- Duplicates: need to integrate duplicate detection
- Spam and spider traps: need to integrate spam detection
- Politeness: we need to be "nice" and space out all requests for a site over a longer period (hours, days)
- Freshness: we need to recrawl periodically.
  - Because of the size of the web, we can do frequent recrawls only for a small subset.
  - Again, subselection problem or prioritization

Magnitude of the crawling problem

- To fetch 20,000,000,000 pages in one month . . .
  . . . we need to fetch almost 8000 pages per second!
- Actually: many more since many of the pages we attempt to crawl will be duplicates, unfetchable, spam etc.

What a crawler must do

Be polite

- Don't hit a site too often
- Only crawl pages you are allowed to crawl: robots.txt

Be robust

- Be immune to spider traps, duplicates, very large pages, very large websites, dynamic pages etc

Robots.txt

- Protocol for giving crawlers ("robots") limited access to a website, originally from 1994
- Example:

  User-agent: *

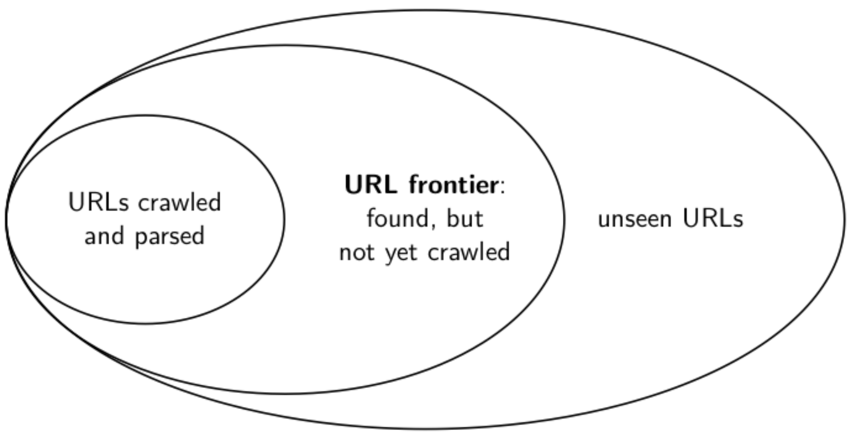
  Disallow: /yoursite/temp/

  User-agent: searchengine

  Disallow: /

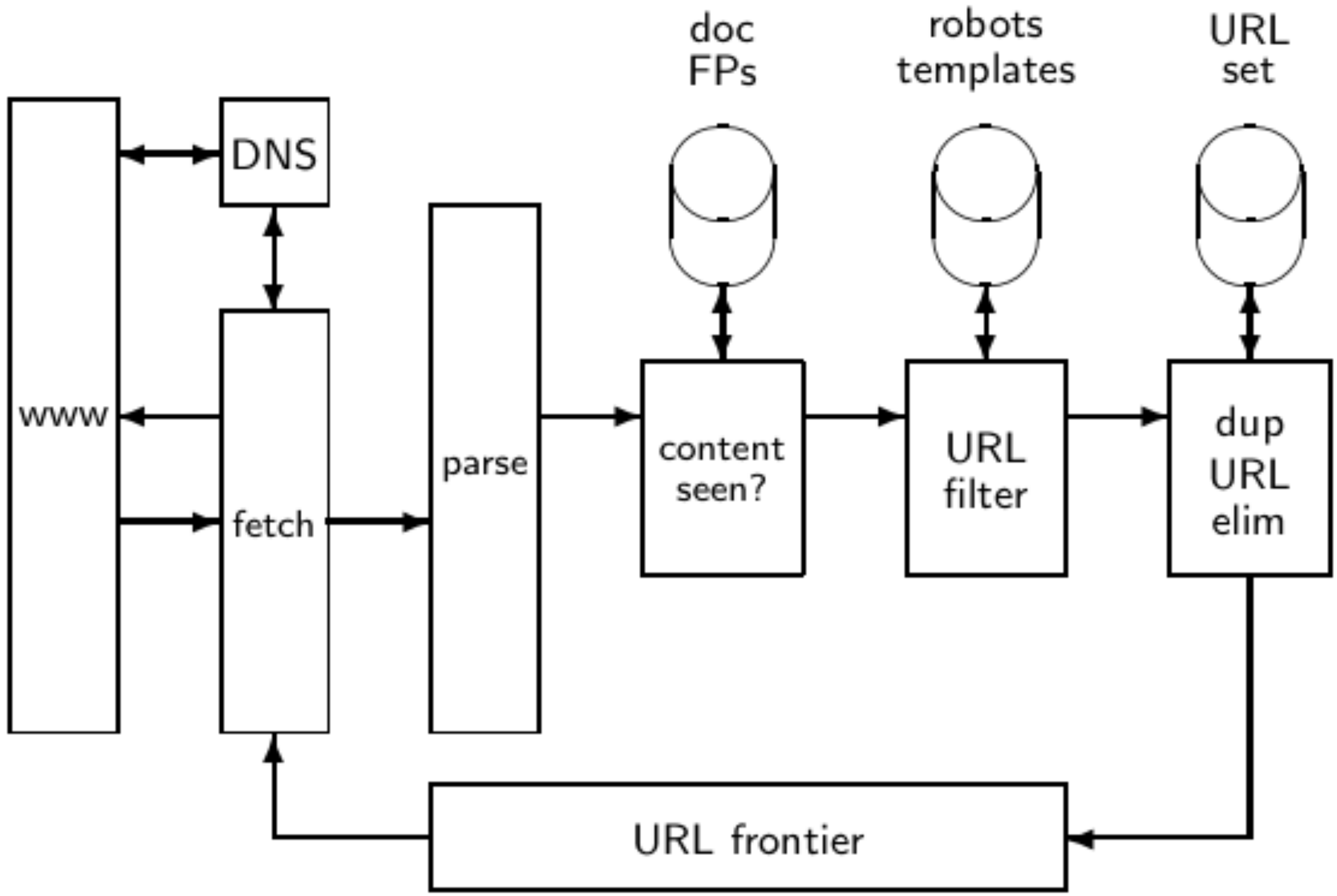
- **Important**: cache the robots.txt file of each site we are crawling

URL Frontier



- The URL frontier is the data structure that holds and manages URLs we've seen, but that have not been crawled yet.
- Can include multiple pages from the same host
- Must avoid trying to fetch them all at the same time
- Must keep all crawling threads busy

Basic Crawling Architecture



URL normalization

- Some URLs extracted from a document are relative URLs.
- E.g., at http://mit.edu, we may have aboutsite.html
  - This is the same as: http://mit.edu/aboutsite.html
- During parsing, we must normalize (expand) all relative URLs.

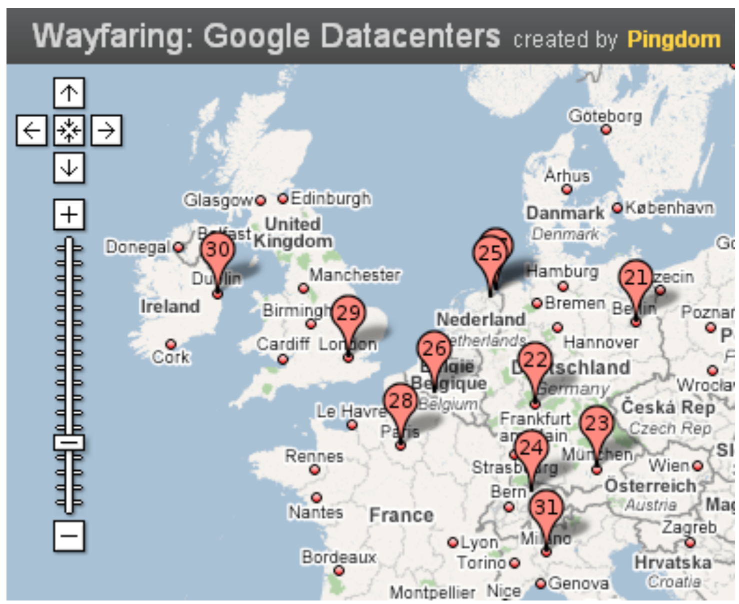
Content seen

- For each page fetched: check if the content is already in the index
- Check this using document fingerprints or shingles
- Skip documents whose content has already been indexed

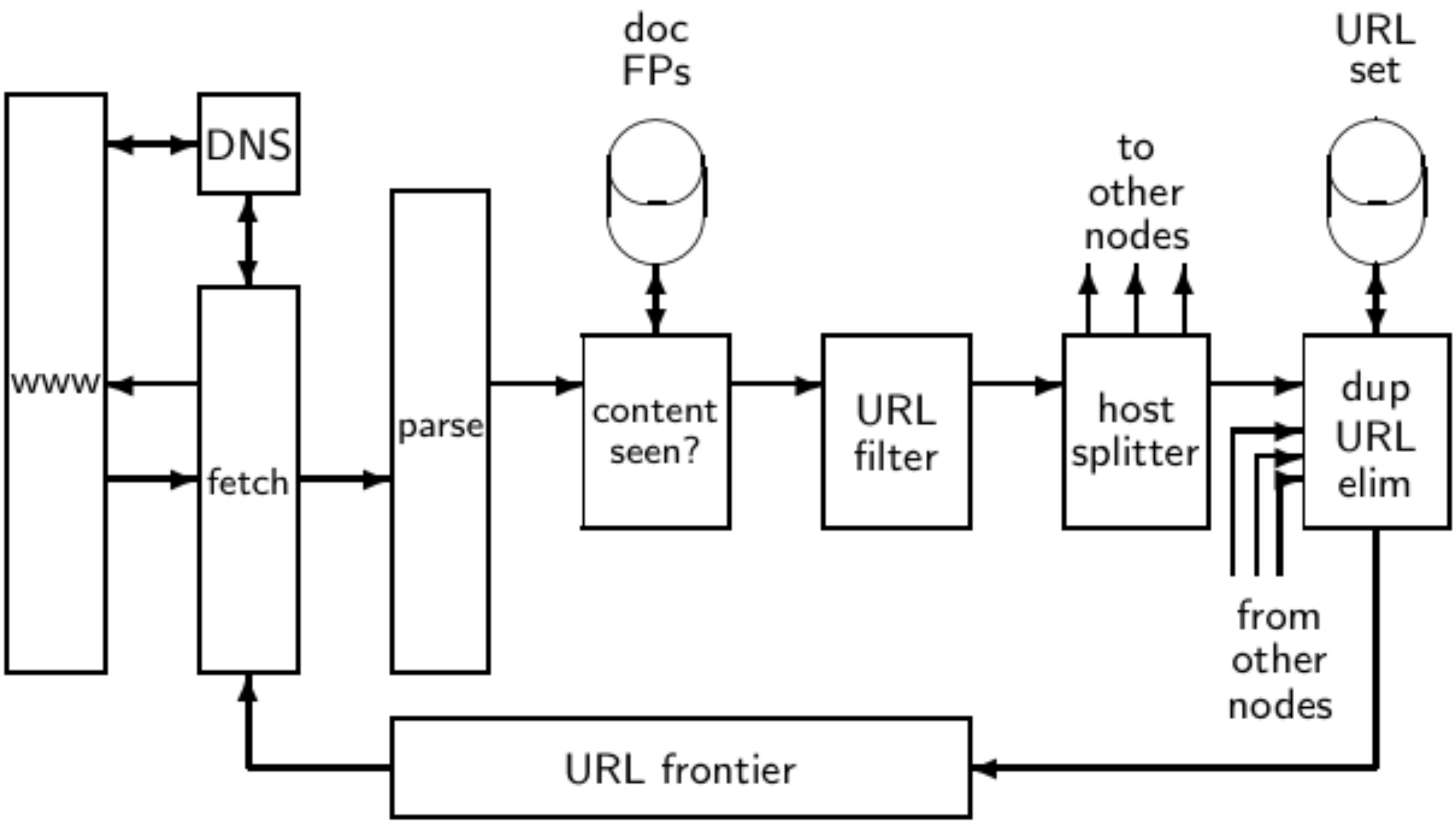
- Still need to consider **Freshness**: Crawl some pages (e.g., news sites) more often than others

Distributing the crawler

- Run multiple crawl threads, potentially at different nodes
  - Usually geographically distributed nodes
- Partition hosts being crawled into nodes



Distributed crawling architecture

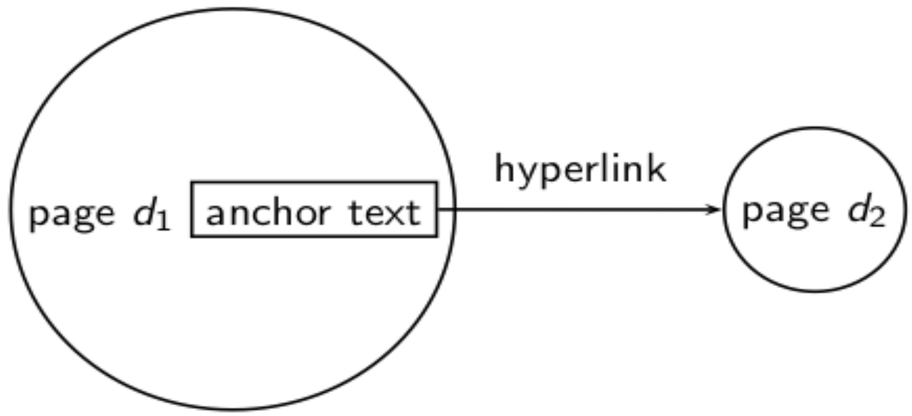


A Crawler Issue: Spider traps

- Malicious server that generates an infinite sequence of linked pages
- Sophisticated spider traps generate pages that are not easily identified as dynamic.

LINK ANALYSIS

The web as a directed graph

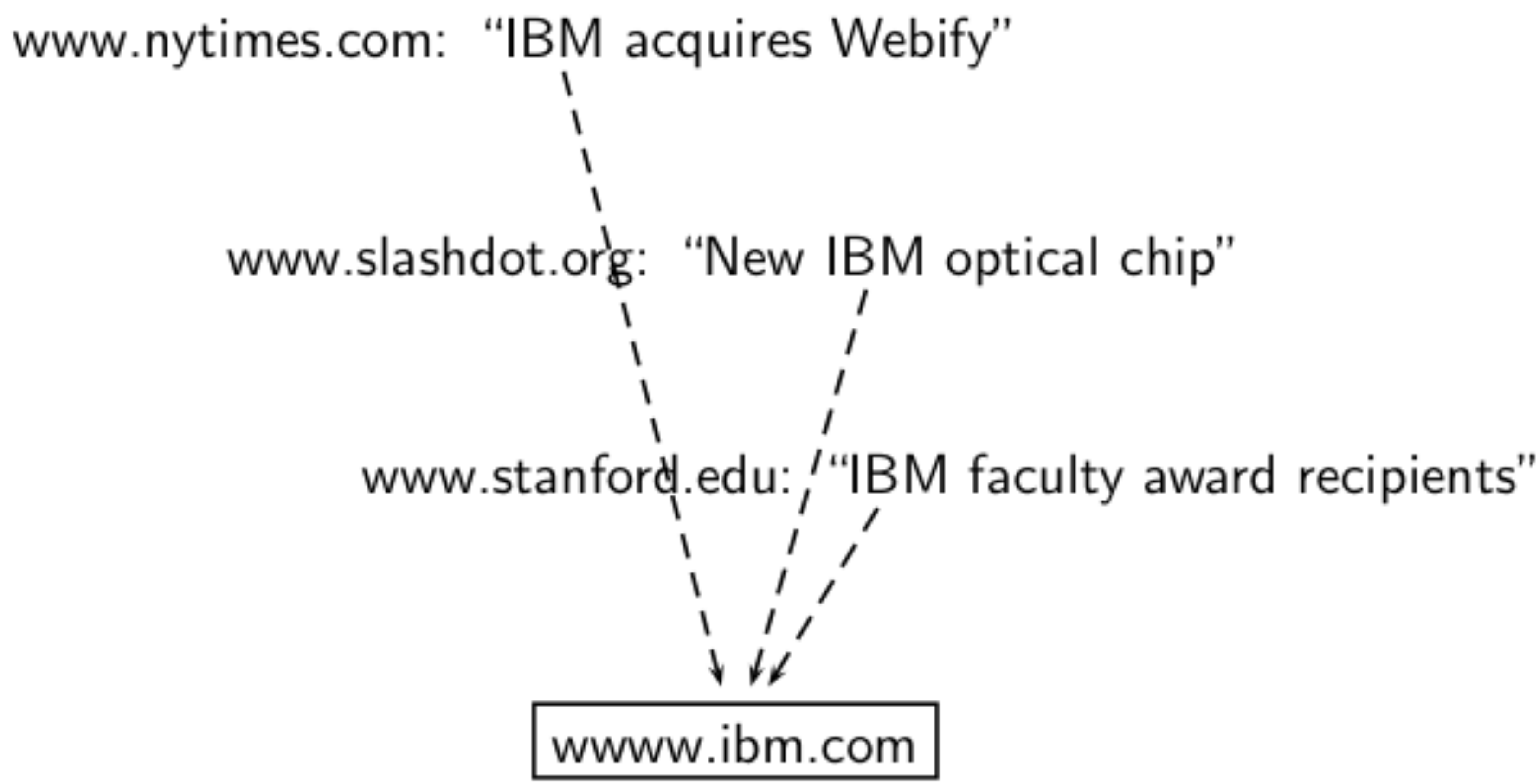


- Assumption 1: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- Assumption 2: The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink .
  - Example: "You can find cheap cars here. "
  - Anchor text: "You can find cheap cars here"

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page!
  - … if IBM home page is mostly graphics
- Searching on [anchor text → d_2] is better for the query *IBM*.
  - In this representation, the page with most occurences of *IBM* is www.ibm.com

Anchor text containing *IBM* pointing to www.ibm.com



Indexing anchor text

- Thus: Anchor text is often a better description of a page's content than the page itself.
- Anchor text can be weighted more highly than document text.

  (based on Assumption 1 & 2)

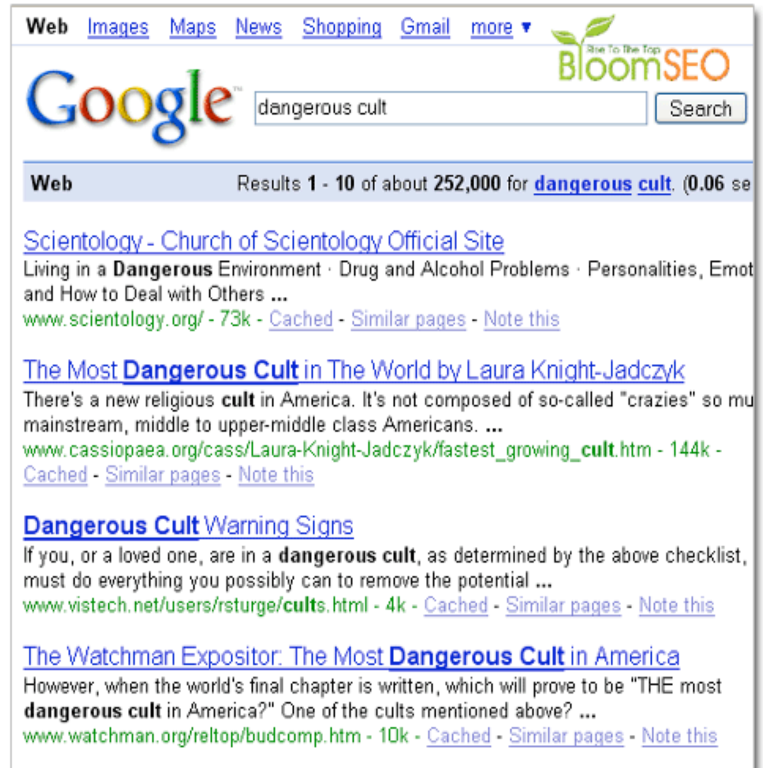
Assumptions underlying PageRank

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.
- Assumption 2: The anchor text describes the content of the linked-to page.

- Is Assumption 1 true in general?
- Is Assumption 2 true in general?

Google bombs

- Is a search with "bad" results due to maliciously manipulated anchor text.
- E.g., [dangerous cult] on Google, Bing, Yahoo
  - Coordinated link creation by those who dislike the Church of Scientology
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.
- Defused Google bombs: [who is a failure?], [evil empire]



PAGERANK

Origins of PageRank: Citation analysis – 1

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.
  - Cocitation similarity on the web: Google's "find pages like this" or "Similar" feature.

Citation analysis – 2

- Another application: Citation frequency can be used to measure the impact of an article.
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
  - ... mainly because of link spam.
- Better measure: weighted citation frequency or citation rank
  - An article's vote is weighted according to its citation impact.
  - Circular? No: can be formalized in a well-defined way

Citation analysis – 3

- Better measure: weighted citation frequency or citation rank, invented in the context of citation analysis by Pinsker and Narin in the 1960s.
- This is basically PageRank.

- We can use the same formal representation for
  - citations in the scientific literature
  - hyperlinks on the web
- Appropriately weighted citation frequency is an excellent measure of quality ...
  - ... both for web pages and for scientific publications.

Definition of PageRank

- The importance of a page is given by the importance of the pages that link to it.

$$x_i = \sum_{j \in B_i} \frac{1}{N_j} x_j$$

importance of page *i*

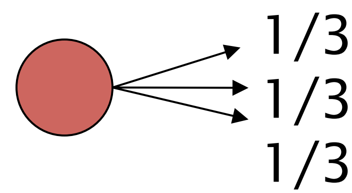
importance of page *j*

pages *j* that link to page *i*

number of outlinks from page *j*

Pagerank scoring

- Imagine a browser doing a random walk on web pages:

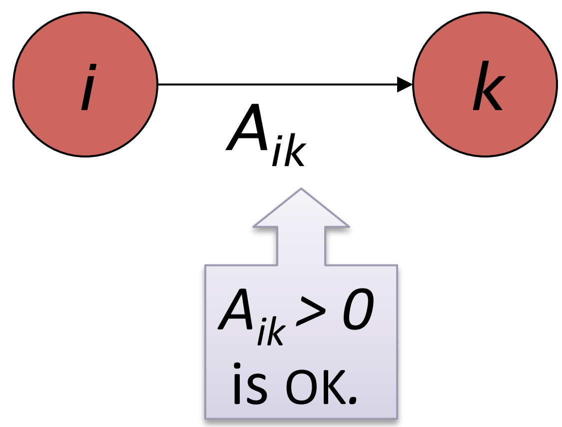


  - Start at a random page
  - At each step, follow one of the *n* links on that page, each with 1/*n* probability
- Do this repeatedly. Use the "long-term visit rate" as the page's score
- This is a global score for the page, based on the topology of the network.
- Think of it as ***g(d)*** from Chapter 7

Markov chains

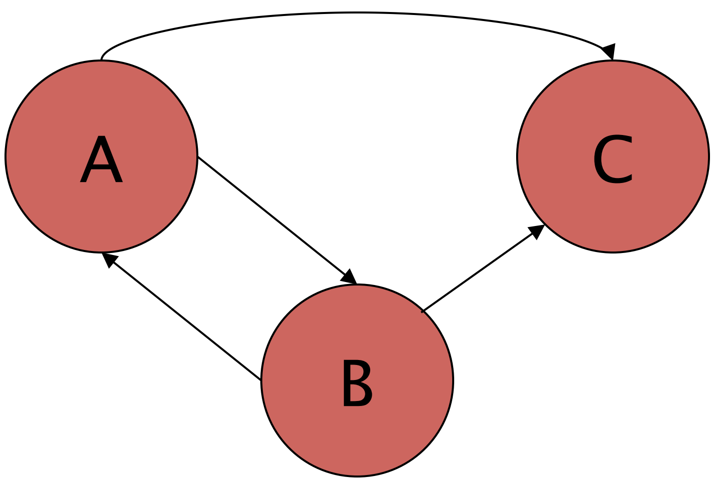
A Markov chain consists of *n* states, plus an *n*×*n* transition probability matrix A.

- At each step, we are in exactly one of the states.
- For $1 \le i,k \le n$, the matrix entry A_{ik} tells us the probability of k being the next state, given we are currently in state *i*.
- **Memorylessness property**: The next state depends only at the current state (first order Markov Chain)



Markov chains

- Clearly, for all i, $\sum_{k=1}^{n} A_{ik} = 1.$
- Markov chains are abstractions of random walks

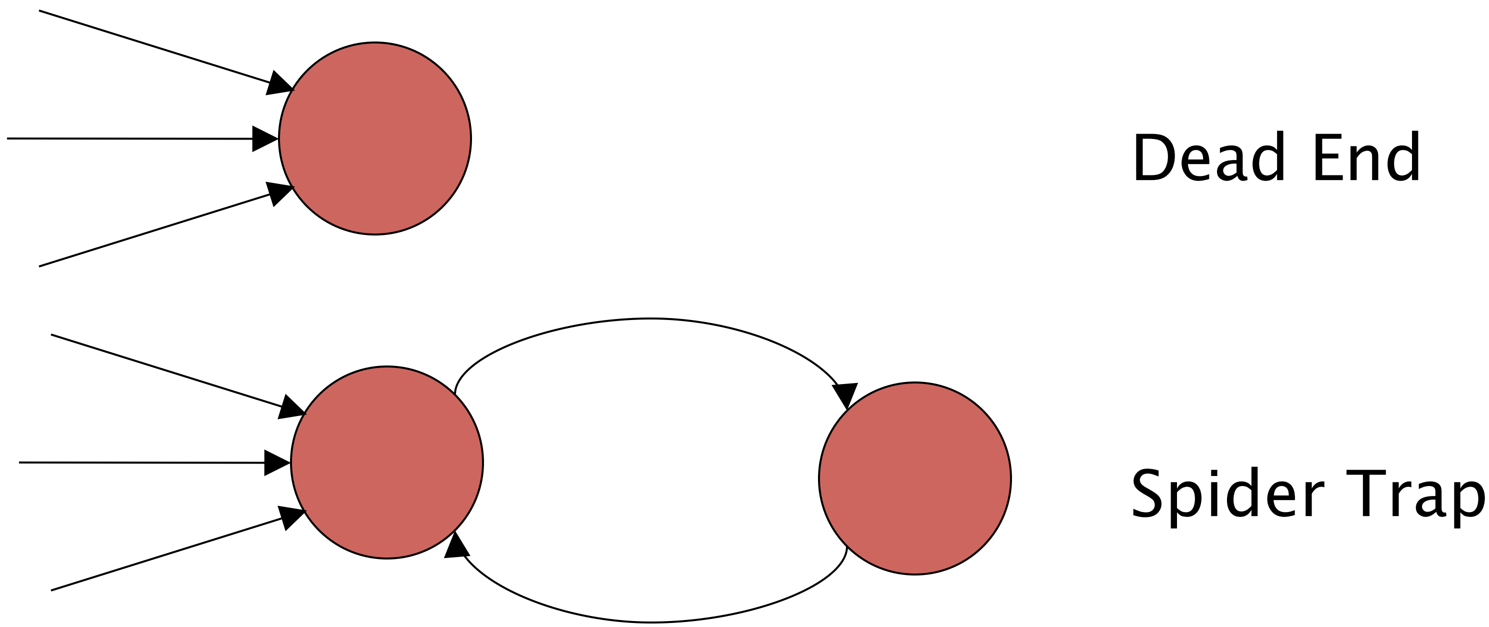


Try this: Calculate the matrix A_{ik} using 1/n possibility

A_{ik}:	A	B	C
A			
B			
C			

Not quite enough

- The web is full of dead ends.
  - What sites have dead ends?
  - Our random walk can get stuck.



Teleporting

- At each step, with probability 10%, teleport to a random web page
- With remaining probability (90%), follow a random link on the page
  - If a dead-end, stay put in this case

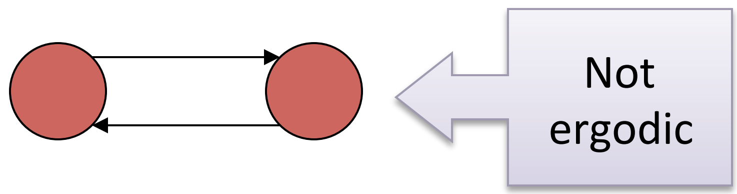
Teleport!

Follow!

$$\vec{rank} = (1-a)A \times \vec{rank} + \alpha \left[\frac{1}{N} \right] N \times 1$$

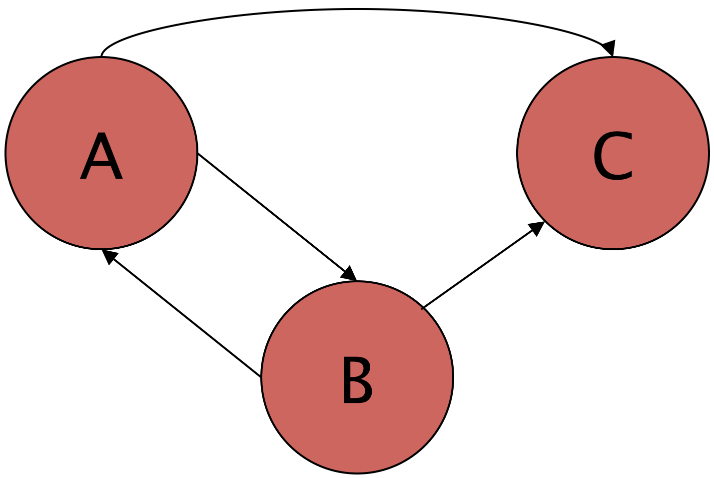
Ergodic Markov chains

- 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



  - With teleportation, our Markov chain is ergodic
  - Theorem: With an ergodic Markov chain, there is a stable long term visit rate.

Markov chains (2nd Try)



Try this: Calculate the matrix A_{ik} using a 10% chance of teleportation

A_{ik}:	A	B	C
A			
B			
C			

Probability vectors

- A probability (row) vector **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*.
  1 *i* *n*

More generally, the vector 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 **x** = *(x_1, … x_n)* at this step, what is it at the next step?
- Recall that row *i* of the transition prob. Matrix **A** tells us where we go next from state *i*.
- So from **x**, our next state is distributed as **xA**.

Pagerank algorithm

- 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 xA
  - After two steps at xA^2 , then xA^3 and so on.
  - "Eventually" means for "large" k, $xA^k = a$

- Algorithm: multiply x by increasing powers of A until the product looks stable

Steady State

- For any ergodic Markov chain, there is a unique long-term visit rate for each state
  - Over a long period, we'll visit each state in proportion to this rate
  - It doesn't matter where we start

Eigenvector formulation

- The flow equations can be written

$$\mathbf{r} = \mathbf{A}\mathbf{r}$$

- So the rank vector is an eigenvector of the adjacency matrix
  - In fact, it's the first or principal eigenvector, with corresponding eigenvalue 1

PageRank summary

- Pre-processing:
  - Given graph of links, build matrix **A**
  - From it compute **a**
  - The pagerank a_i is a scaled number between 0 and 1
- Query processing:
  - Retrieve pages meeting query
  - Rank them by their pagerank
  - Order is query-*independent*

PageRank issues

- Real surfers are not random surfers.
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!

  → Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query [video service].
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
  - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.
  - Clearly not desirable.

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - PageRank in its original form (as presented here) has a negligible impact on ranking.
  - However, variants of a page's PageRank are still an essential part of ranking.
  - Addressing link spam is difficult and crucial.

Summary

- Crawling – Obtaining documents for indexing
  - Need to be polite
- PageRank – A G(d) for asymmetrically linked documents

- Chapters 20 and 21 of IIR
- Resources
  - Paper on Mercator crawler by Heydon et al.
  - Robot exclusion standard

"PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results"