Outline

- Motivation and taxonomy of crawlers
- Basic crawlers and implementation issues
- Universal crawlers
- Preferential (focused and topical) crawlers
- Crawler ethics and conflicts

Preferential crawlers

- Assume we can estimate for each page an importance measure, I(p)
- Want to visit pages in order of decreasing I(p)
- Maintain the frontier as a priority queue sorted by I(p)
- Possible figures of merit:
  - Precision ~
    | p: crawled(p) & I(p) > threshold | / | p: crawled(p) |
  - Recall ~
    | p: crawled(p) & I(p) > threshold | / | p: I(p) > threshold |

Preferential crawlers

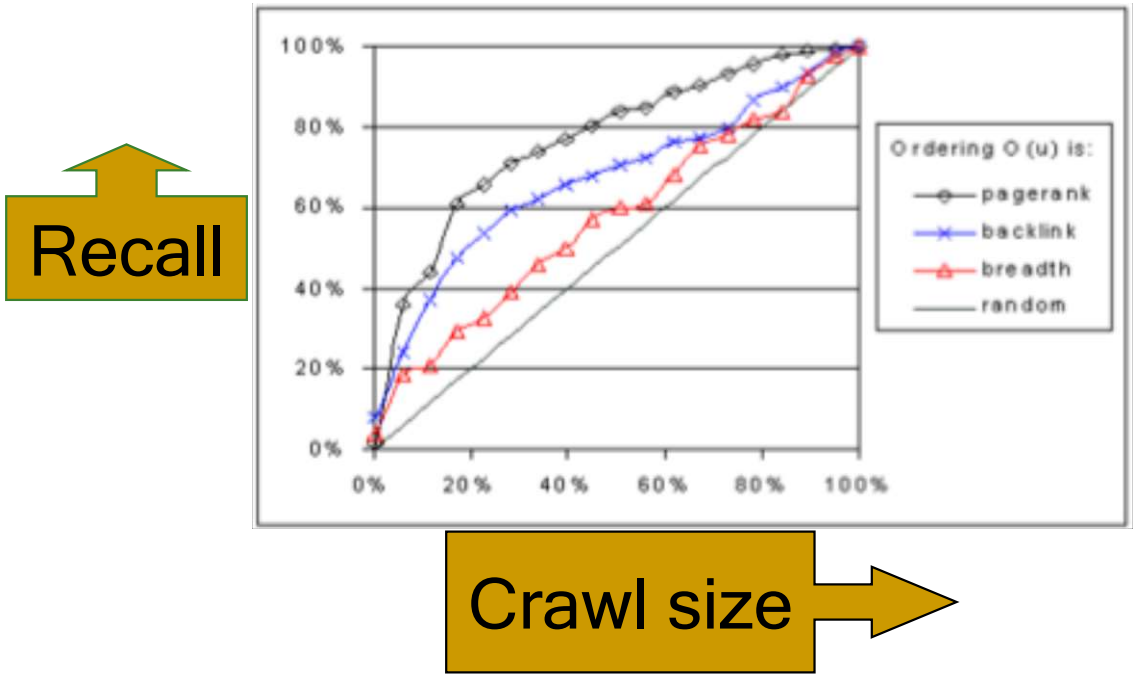
- Selective bias toward some pages, eg. most "relevant"/topical, closest to seeds, most popular/largest PageRank, unknown servers, highest rate/amount of change, etc…
- Focused crawlers
  - Supervised learning: classifier based on labeled examples
- Topical crawlers
  - Best-first search based on similarity(topic, parent)
  - Adaptive crawlers
    - Reinforcement learning
    - Evolutionary algorithms/artificial life

Preferential crawling algorithms: Examples

- Breadth-First
  - Exhaustively visit all links in order encountered
- Best-*N*-First
  - Priority queue sorted by similarity, explore top N at a time
  - Variants: DOM context, hub scores
- PageRank
  - Priority queue sorted by keywords, PageRank
- SharkSearch
  - Priority queue sorted by combination of similarity, anchor text, similarity of parent, etc. (powerful cousin of FishSearch)
- InfoSpiders
  - Adaptive distributed algorithm using an evolving population of learning agents

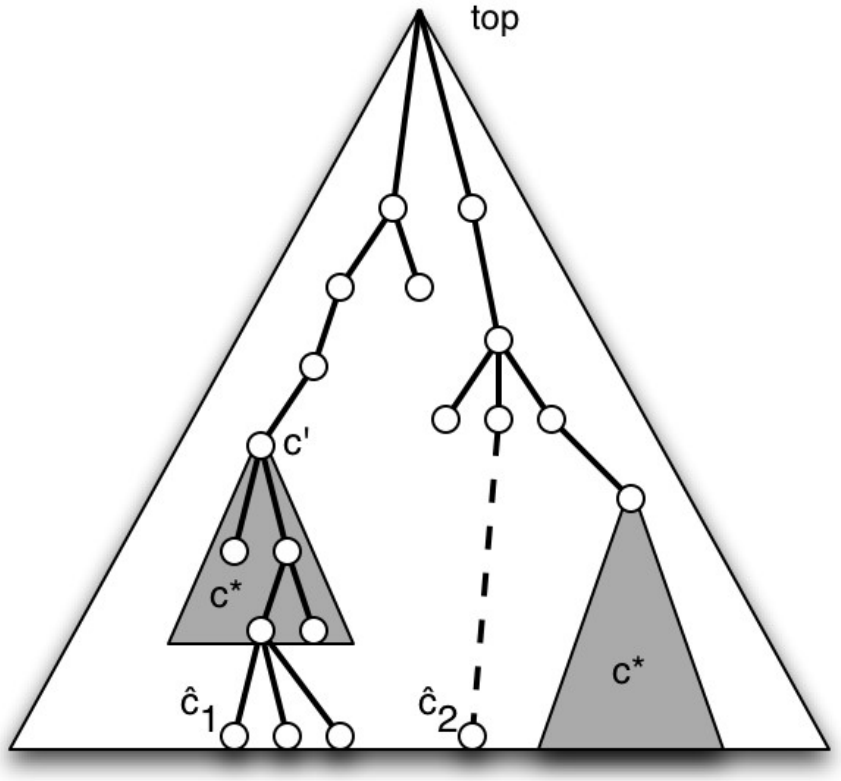
Preferential crawlers: Examples

- For I(p) = PageRank (estimated based on pages crawled so far), we can find high-PR pages faster than a breadth-first crawler (Cho, Garcia-Molina & Page 1998)



Focused crawlers: Basic idea

- Naïve-Bayes classifier based on example pages in desired topic, c*
- Score(p) = Pr(c*|p)
  - Soft focus: frontier is priority queue using page score
  - Hard focus:
    - Find best leaf $\hat{c}$ for p
    - If an ancestor c' of $\hat{c}$ is in c* then add links from p to frontier, else discard
  - Soft and hard focus work equally well empirically

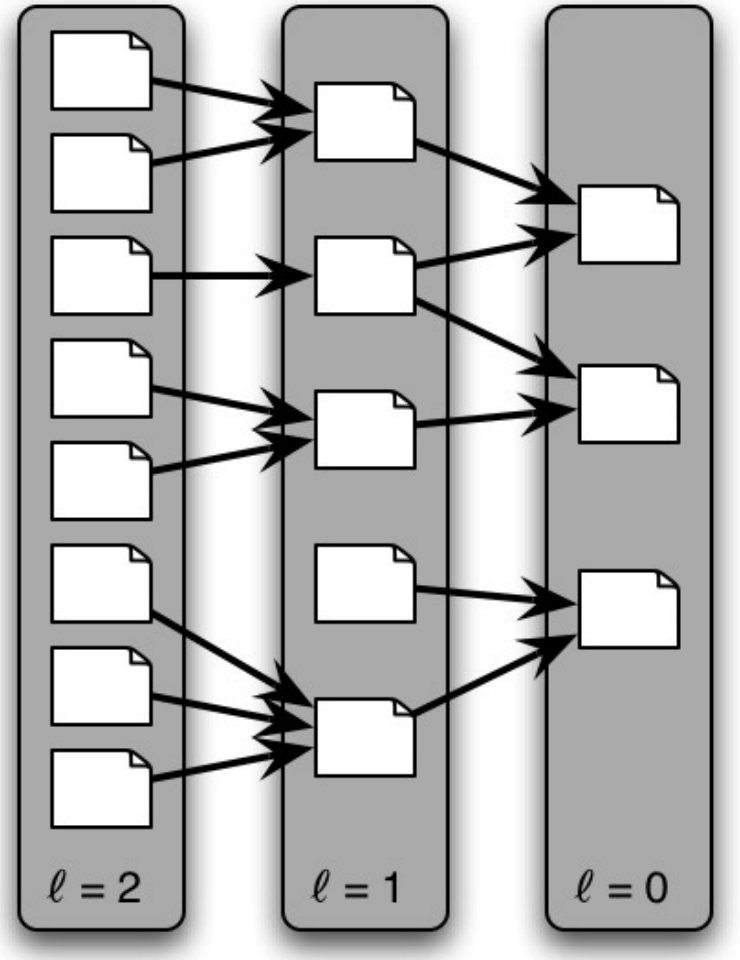


Example: Open Directory

Focused crawlers

- Can have multiple topics with as many classifiers, with scores appropriately combined (Chakrabarti et al. 1999)
- Can use a distiller to find topical hubs periodically, and add these to the frontier
- Can accelerate with the use of a critic (Chakrabarti et al. 2002)
- Can use alternative classifier algorithms to naïve-Bayes, e.g. SVM and neural nets have reportedly performed better (Pant & Srinivasan 2005)

Context-focused crawlers



Context graph

- Same idea, but multiple classes (and classifiers) based on link distance from relevant targets
  - ℓ=0 is topic of interest
  - ℓ=1 link to topic of interest
  - Etc.
- Initially needs a back-crawl from seeds (or known targets) to train classifiers to estimate distance
- Links in frontier prioritized based on estimated distance from targets
- Outperforms standard focused crawler empirically

Topical crawlers

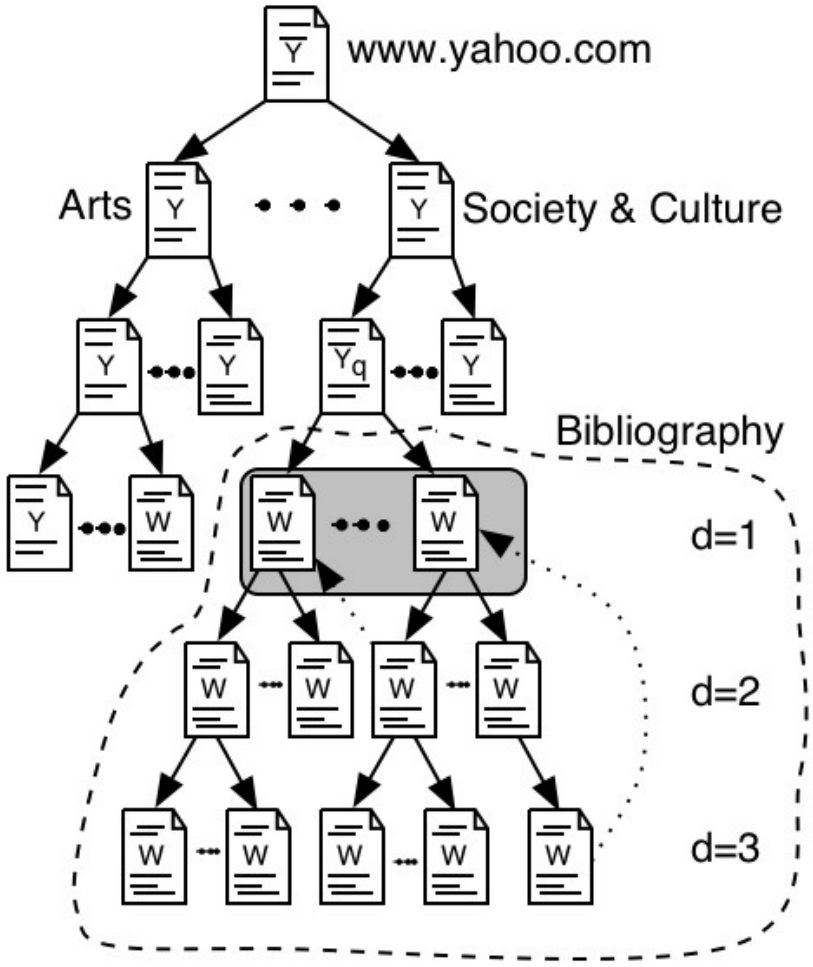
- All we have is a topic (query, description, keywords) and a set of seed pages (not necessarily relevant)
- No labeled examples
- Cosine similarity may be used
- Original idea: Menczer 1997, Menczer & Belew 1998

Topical locality

- Topical locality is a necessary condition for a topical crawler to work, and for surfing to be a worthwhile activity for humans
- Links must encode semantic information, i.e. say something about neighbor pages, not be random
- It is also a sufficient condition if we start from “good” seed pages
- Indeed we know that Web topical locality is strong :
  - Indirectly (crawlers work and people surf the Web)
  - From direct measurements (Davison 2000; Menczer 2004, 2005)

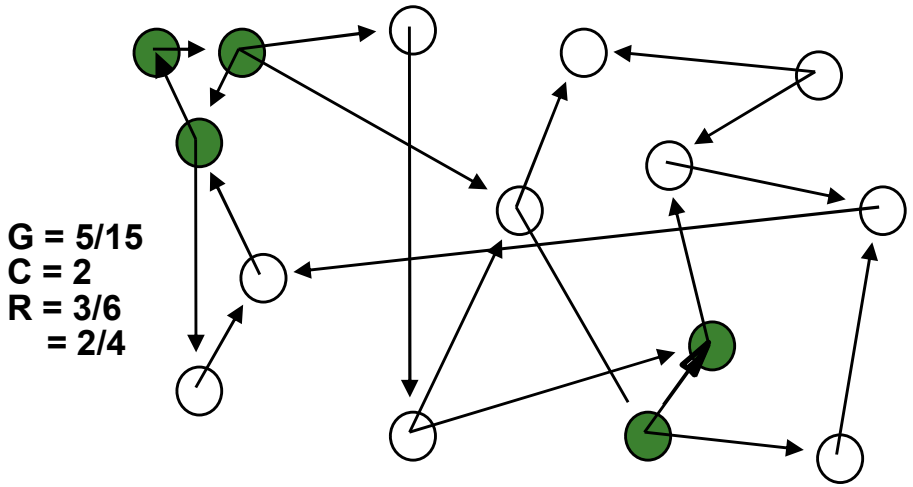
Quantifying topical locality

- Different ways to pose the question:
  - How quickly does semantic locality decay?
  - How fast is topic drift?
  - How quickly does content change as we surf away from a starting page?
- To answer these questions, let us consider exhaustive breadth-first crawls from 100 topic pages



The "link-cluster" conjecture

- Connection between semantic topology (relevance) and link topology (hypertext)
  - G = Pr[rel(p)] ~ fraction of relevant/topical pages (topic generality)
  - R = Pr[rel(p) | rel(q) AND link(q,p)] ~ cond. prob. Given neighbor on topic
- Related nodes are clustered if ***R > G***
  - Necessary and sufficient condition for a random crawler to find pages related to start points
  - Example: 2 topical clusters with stronger modularity within each cluster than outside



Link-cluster conjecture

- Stationary hit rate for a random crawler:

$$\eta(t+1) = \eta(t) \cdot R + (1 - \eta(t)) \cdot G \geq \eta(t)$$

$$\eta \xrightarrow{t \to \infty} \eta^* = \frac{G}{1 - (R - G)}$$

Conjecture

$$\eta^* > G \Leftrightarrow R > G$$

Value added of links

$$\frac{\eta^*}{G} - 1 = \frac{R - G}{1 - (R - G)}$$

where η(t) is the probability that the crawler hits a relevant page at time t

Outline

- Motivation and taxonomy of crawlers
- Basic crawlers and implementation issues
- Universal crawlers
- Preferential (focused and topical) crawlers
- Evaluation of preferential crawlers
- Crawler ethics and conflicts

Crawler ethics and conflicts

- Crawlers can cause trouble, even unwillingly, if not properly designed to be “polite” and “ethical”
- For example, sending too many requests in rapid succession to a single server can amount to a Denial of Service (DoS) attack!
  - Server administrator and users will be upset
  - Crawler developer/admin IP address may be blacklisted

Crawler etiquette (important!)

- Identify yourself
  - Use 'User-Agent' HTTP header to identify crawler, website with description of crawler and contact information for crawler developer
  - Use 'From' HTTP header to specify crawler developer email
  - Do not disguise crawler as a browser by using their 'User-Agent' string
- Always check that HTTP requests are successful, and in case of error, use HTTP error code to determine and immediately address problem
- Pay attention to anything that may lead to too many requests to any one server, even unwillingly, e.g.:
  - redirection loops
  - spider traps

Crawler etiquette (important!)

- Spread the load, do not overwhelm a server
  - Make sure that no more than some max. number of requests to any single server per unit time, say < 1/second
- Honor the Robot Exclusion Protocol
  - A server can specify which parts of its document tree any crawler is or is not allowed to crawl by a file named ‘robots.txt’ placed in the HTTP root directory, e.g. http://www.indiana.edu/robots.txt
  - Crawler should always check, parse, and obey this file before sending any requests to a server
  - More info at:
    - http://www.google.com/robots.txt
    - http://www.robotstxt.org/wc/exclusion.html

More on robot exclusion

- Make sure URLs are canonical before checking against robots.txt
- Avoid fetching robots.txt for each request to a server by caching its policy as relevant to this crawler
- Let's look at some examples to understand the protocol…

www.apple.com/robots.txt

```
# robots.txt for http://www.apple.com/

User-agent: *
Disallow:
```

All crawlers…

…can go anywhere!

www.microsoft.com/robots.txt

```
# Robots.txt file for http://www.microsoft.com

User-agent: *
Disallow: /canada/Library/mnp/2/aspx/
Disallow: /communities/bin.aspx
Disallow: /communities/eventdetails.mspx
Disallow: /communities/blogs/PortalResults.mspx
Disallow: /communities/rss.aspx
Disallow: /downloads/Browse.aspx
Disallow: /downloads/info.aspx
Disallow: /france/formation/centres/planning.asp
Disallow: /france/mnp_utility.mspx
Disallow: /germany/library/images/mnp/
Disallow: /germany/mnp_utility.mspx
Disallow: /ie/ie40/
Disallow: /info/customerror.htm
Disallow: /info/smart404.asp
Disallow: /intlkb/
Disallow: /isapi/
#etc…
```

All crawlers…

…are not allowed in these paths…

www.springer.com/robots.txt

```
# Robots.txt for http://www.springer.com (fragment)

User-agent: Googlebot
Disallow: /chl/*
Disallow: /uk/*
Disallow: /italy/*
Disallow: /france/*

User-agent: slurp
Disallow:
Crawl-delay: 2

User-agent: MSNBot
Disallow:
Crawl-delay: 2

User-agent: scooter
Disallow:

# all others
User-agent: *
Disallow: /
```

Google crawler is allowed everywhere except these paths

Yahoo and MSN/Windows Live are allowed everywhere but should slow down

AltaVista has no limits

Everyone else keep off!

More crawler ethics issues

- Is compliance with robot exclusion a matter of law?
  - No! Compliance is voluntary, but if you do not comply, you may be blocked
  - Someone (unsuccessfully) sued Internet Archive over a robots.txt related issue
- Some crawlers disguise themselves
  - Using false User-Agent
  - Randomizing access frequency to look like a human/browser
  - Example: click fraud for ads

More crawler ethics issues

- Servers can disguise themselves, too
  - Cloaking: present different content based on User-Agent
  - E.g. stuff keywords on version of page shown to search engine crawler
  - Search engines do not look kindly on this type of "spamdexing" and remove from their index sites that perform such abuse

Gray areas for crawler ethics

- If you write a crawler that unwillingly follows links to ads, are you just being careless, or are you violating terms of service, or are you violating the law by defrauding advertisers?
  - Is non-compliance with Google's robots.txt in this case equivalent to click fraud?
- If you write a browser extension that performs some useful service, should you comply with robot exclusion?