An introduction to Web Mining

part II

Ricardo Baeza-Yates, Aristides Gionis

Yahoo! Research

Barcelona, Spain & Santiago, Chile

ECML/PKDD 2008 Antwerp

Agenda

- Statistical methods: the size of the web
- Content mining
- Link analysis for spam detection

What is the size of the web?

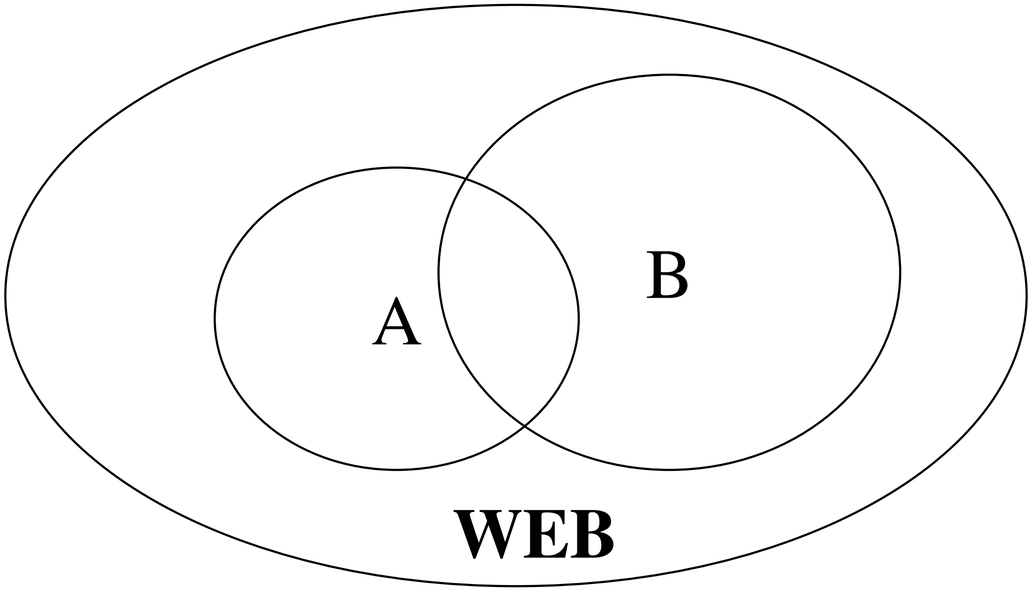
- Issues
  - – The web is really infinite
    - Dynamic content, e.g., calendar
    - Soft 404: www.yahoo.com/anything is a valid page
  - – Static web contains syntactic duplication, mostly due to mirroring (~20-30%)
  - – Some servers are seldom connected
- Who cares?
  - – Media, and consequently the user
  - – Engine design
  - – Engine crawl policy. Impact on recall

What can we attempt to measure?

- The relative size of search engines
  - The notion of a page being indexed is *still* reasonably well defined.
  - Already there are problems
    - Document extension: e.g. Google indexes pages not yet crawled by indexing anchor-text.
    - Document restriction: Some engines restrict what is indexed (first *n* words, only relevant words, etc.)

- The coverage of a search engine relative to another particular crawling process

Relative size and overlap of search engines



- [Bharat & Broder 98]
- Main idea:
- Pr[A&B | A] = s(A&B) / s(A)
- Pr[A&B | B] = s(A&B) / s(B)
- Thus:

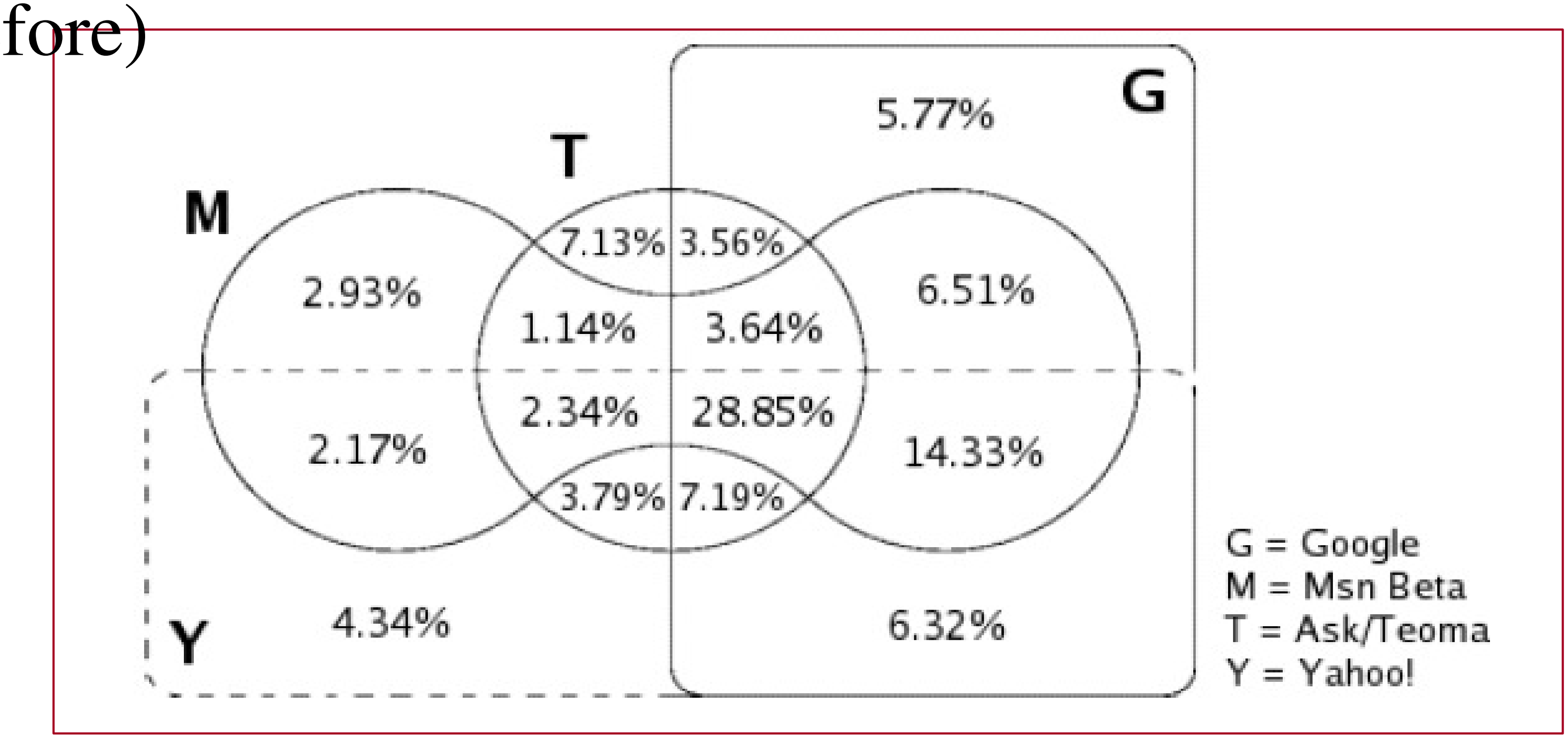
  s(A) / s(B) = Pr[A&B | B] / Pr[A&B | A]
- Need
  - **Sampling** a random page from the index of a SE
  - **Checking** if a page exists at the index of a SE

Sampling and checking pages

- Both tasks by using the public interface SEs
- **Sampling:**
  - Construct a large lexicon
  - Use the lexicon to fire random queries
  - Sample a page from the results
  - (introduces query and ranking biases)
- **Checking:**
  - Construct a *strong* query from the most k most distinctive terms of the page
  - (in order to deal with aliases, mirror pages, etc.)

Refinement of the B&B technique [Gulli & Signorini, 2005]

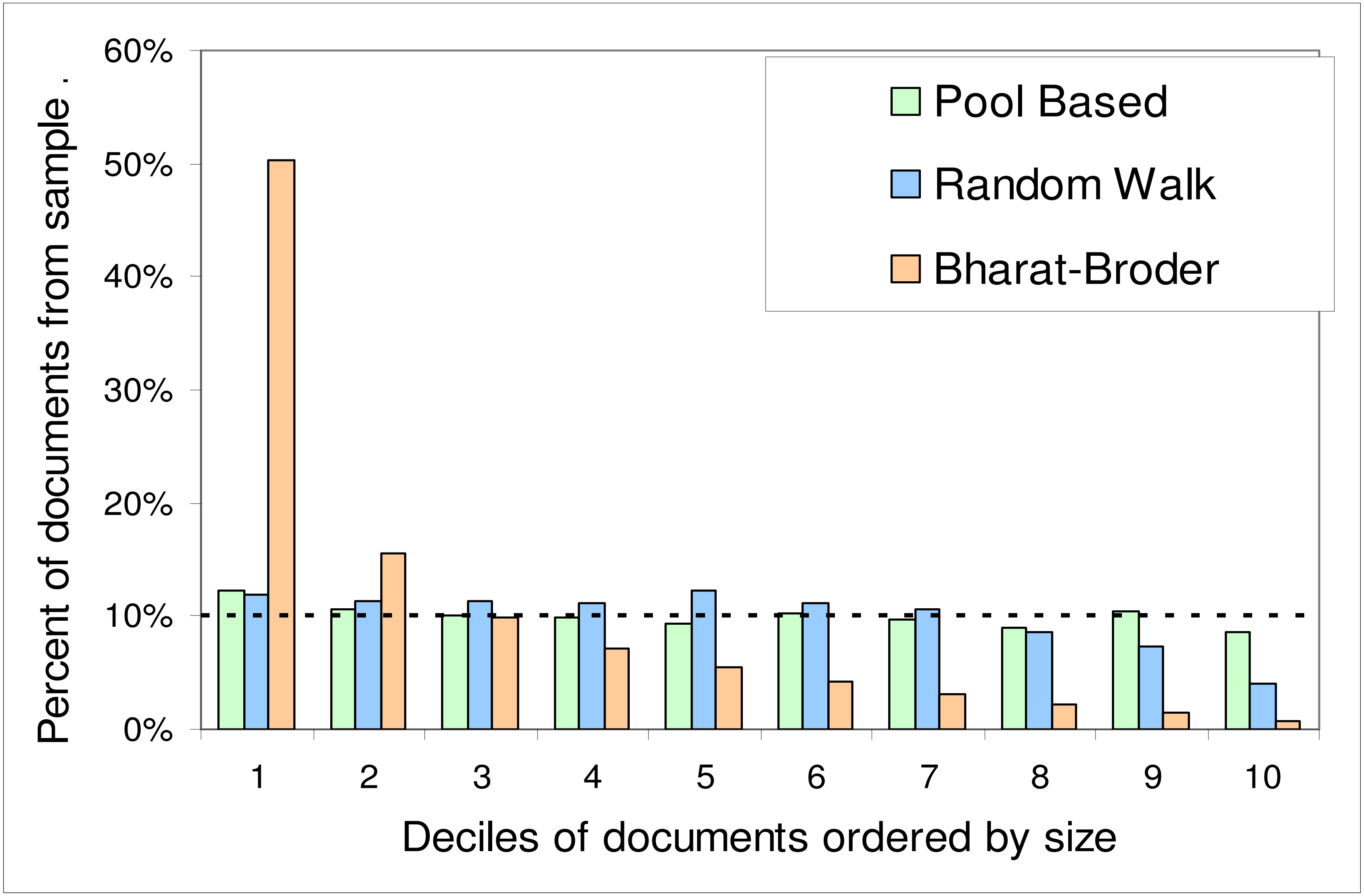
- Total web = 11.5 B
- Union of major search engines = 9.5 B
- Common web = 2.7 B (Much higher correlation than before)



Random-walk sampling

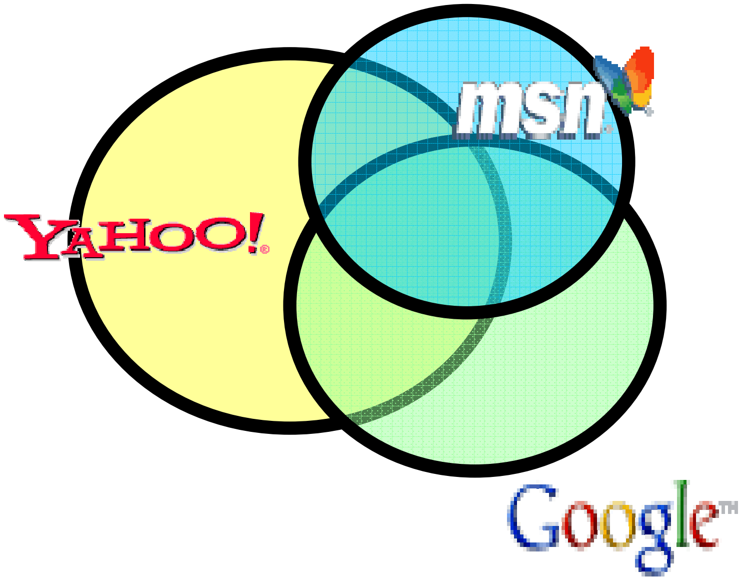
- [Bar-Yossef and Gurevich, WWW 2006]
- Define a graph on documents and queries:
  - Edge *(d,q)* indicates that document *d* is a result of a query *q*
- Random walk gives biased samples
- Bias depends on the degree of docs and queries
- Use Monte Carlo methods to unbias the samples and obtain uniform samples
- Paper shows how to obtain estimates of the degrees and weights needed for the unbiasing

Bias towards long documents



Relative size of major search engines

- [Bar-Yossef and Gurevich, 2006]



Google = 1

Yahoo! = 1.28

MSN Search = 0.73

Content mining

- Duplicate and near-duplicate document detection
- Content-based spam detection

Duplicate/Near-Duplicate Detection

- Duplication: Exact match with fingerprints
- Near-Duplication: Approximate match
  - Overview
    - Compute syntactic similarity with an edit-distance measure
    - Use similarity threshold to detect near-duplicates
      - E.g., Similarity > 80% => Documents are "near duplicates"
      - Not transitive though sometimes used transitively

Computing Similarity

- Features:
  - Segments of a document (natural or artificial breakpoints) [Brin95]
  - Shingles (Word N-Grams) [Brin95, Brod98]

    “a rose is a rose is a rose” =>

    a_rose_is_a

    rose_is_a_rose

    is_a_rose_is

    are all added in the bag of word representation

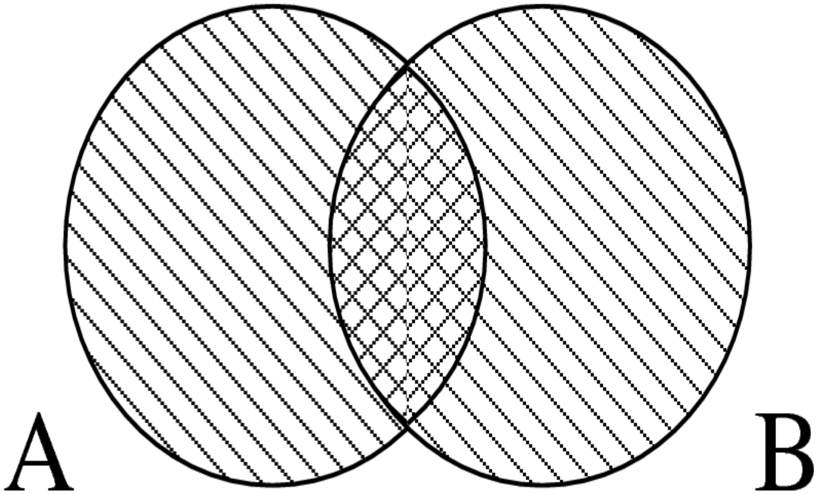
- Similarity Measure
  - TFIDF [Shiv95]
  - Set intersection [Brod98]

    (Specifically, Size_of_Intersection / Size_of_Union)

Jaccard coefficient

- Consider documents *a* and *b*
- Are represented by bag of words *A* and *B*, resp.
- Then:

J(a,b) = |A intersect B| / | A union B|



Shingles + Jaccard coefficient

- Computing exact Jaccard coefficient between all pairs of documents is expensive (quadratic)
- Approximate similarities using a cleverly chosen subset of shingles from each (a *sketch*)
- Idea based on hashing

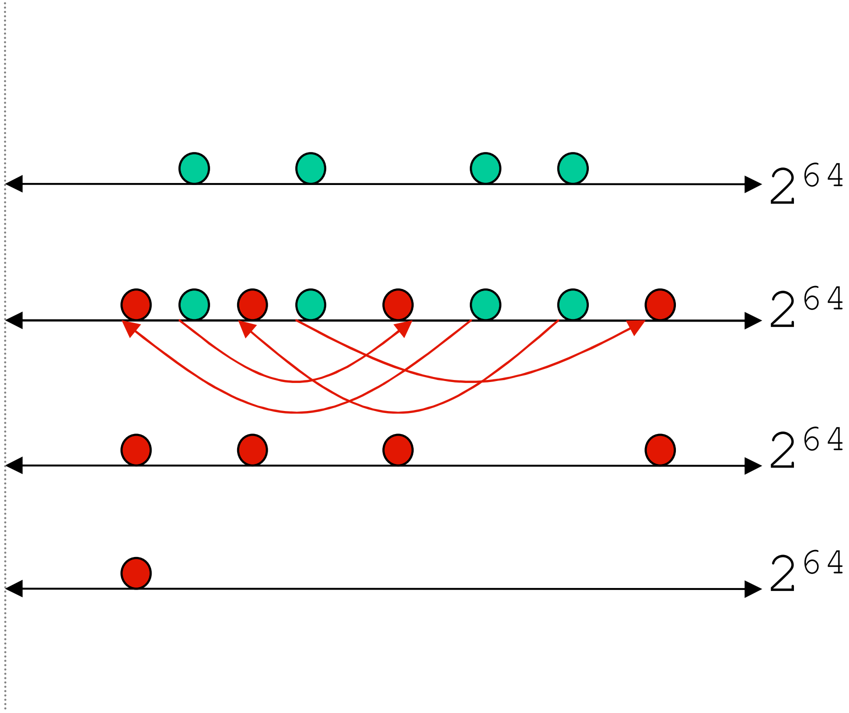
- Also known as locality-sensitive hashing (LSH)
    - A family of hash functions for which items that are similar have higher probability of colliding

Shingles + Jaccard coefficient

- Estimate size_of_intersection / size_of_union based on a short sketch ([Broder 97, Broder 98])
  - Create a “sketch vector” (e.g., of size 200) for each document
  - Documents which share more than t (say 80%) corresponding vector elements are similar
  - For doc D, sketch[i] is computed as follows:
    - Let f map all shingles in the universe to $0..2^m$ (e.g., f = fingerprinting)
    - Let π_i be a specific random permutation on $0..2^m$
    - Pick MIN π_i (f(s)) over all shingles s in D

Computing Sketch[i] for Doc1

Document 1

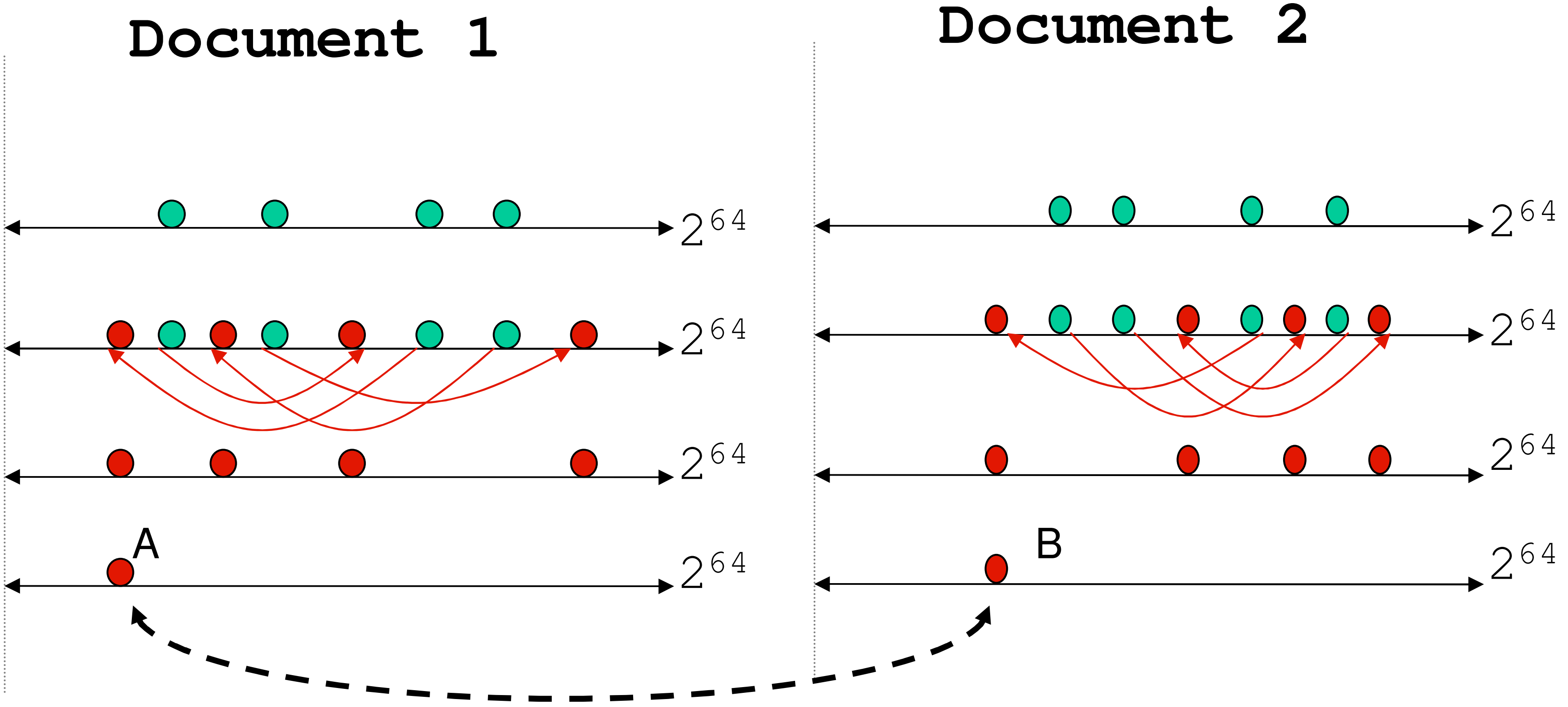


Start with 64 bit shingles

Permute on the number line with π_i

Pick the min value

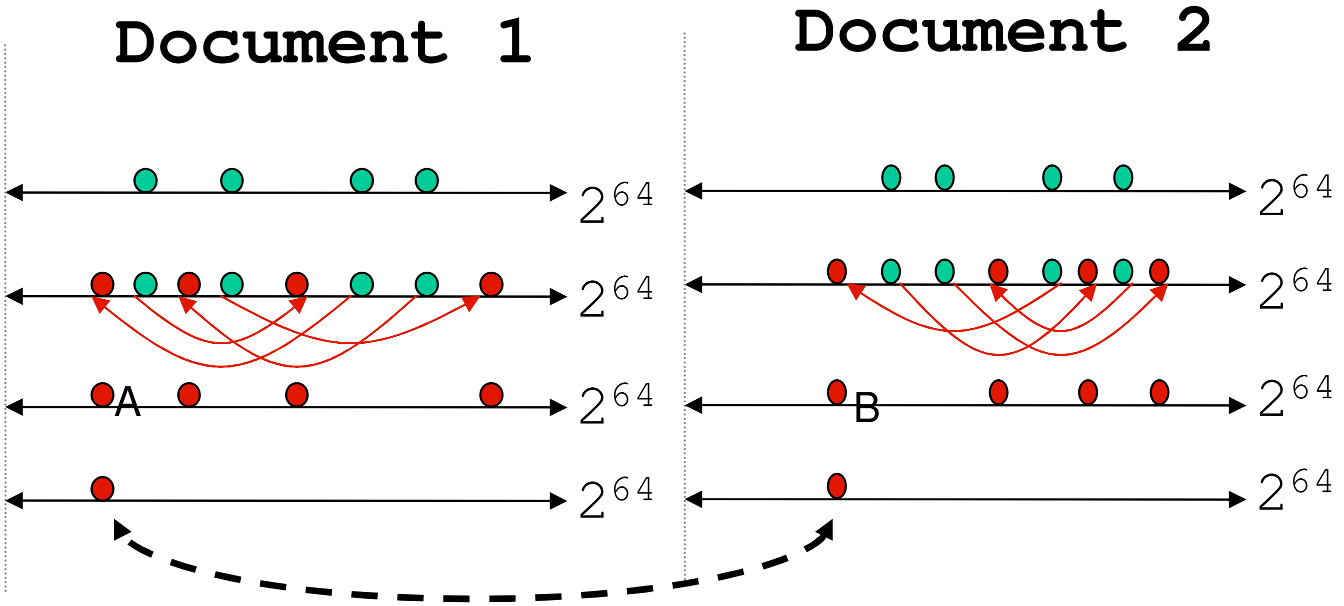
Test if Doc1.Sketch[i] = Doc2.Sketch[i]



Are these equal?

Test for 200 random permutations: $\pi_1, \pi_2, \ldots \pi_{200}$

However…



A = B iff the shingle with the MIN value in the union of Doc1 and Doc2 is common to both (I.e., lies in the intersection)

This happens with probability:

Size_of_intersection / Size_of_union

Mirror detection

- Mirroring is systematic replication of web pages across hosts.
  - Single largest cause of duplication on the web
- Host1/α and Host2/β are mirrors iff

  For all (or most) paths p such that when

  http://Host1/ α / p exists

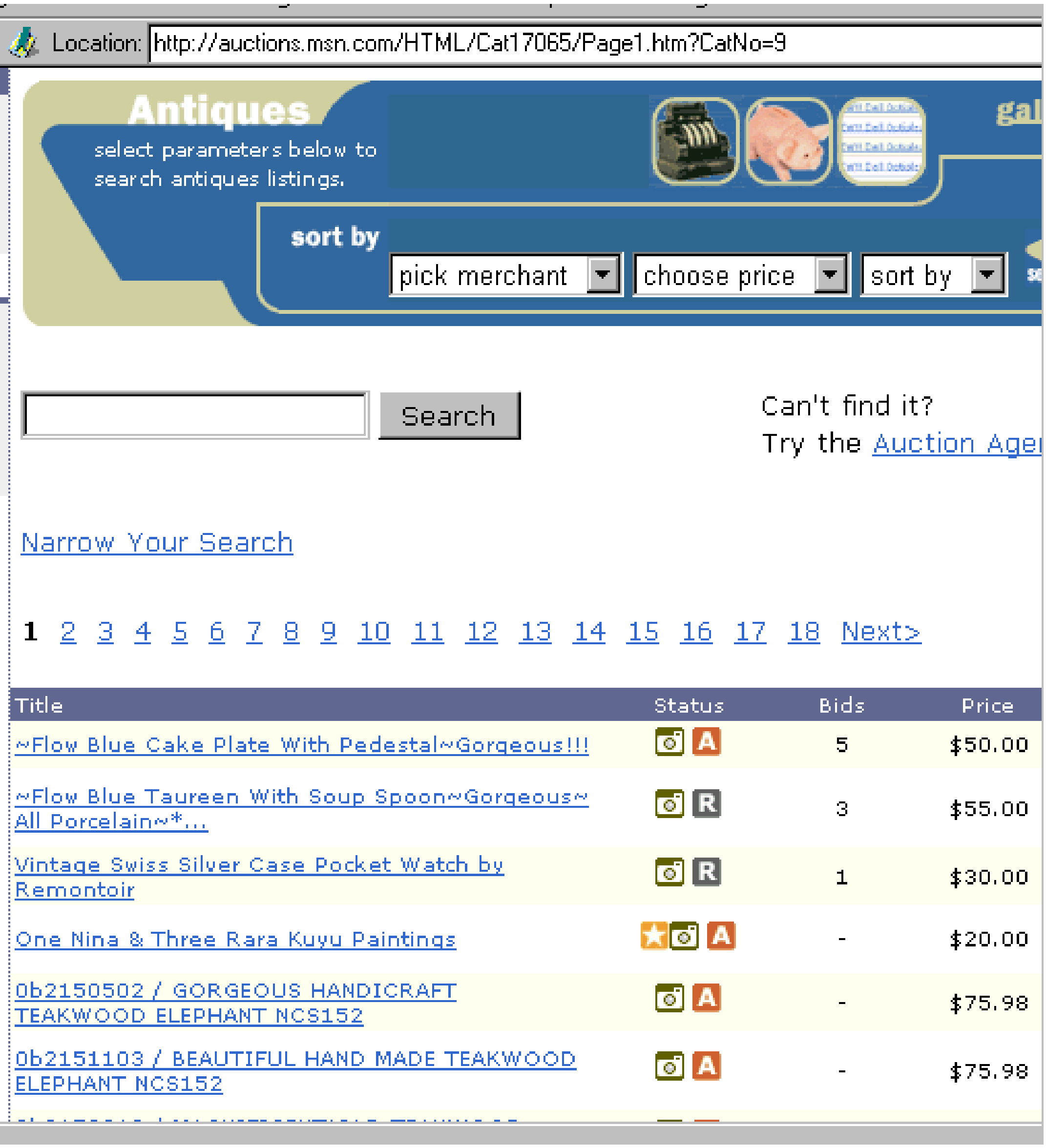
  http://Host2/ β / p exists as well

  with identical (or near identical) content, and vice versa.
- E.g.,
  - http://www.elsevier.com/ and http://www.elsevier.nl/
  - Structural Classification of Proteins
    - http://scop.mrc-lmb.cam.ac.uk/scop
    - http://scop.berkeley.edu/
    - http://scop.wehi.edu.au/scop
    - http://pdb.weizmann.ac.il/scop
    - http://scop.protres.ru/

Repackaged Mirrors

Auctions.msn.com

Auctions.lycos.com



Location: http://auctions.msn.com/HTML/Cat17065/Page1.htm?CatNo=9

Antiques

select parameters below to search antiques listings.

sort by: pick merchant ▾ | choose price ▾ | sort by ▾

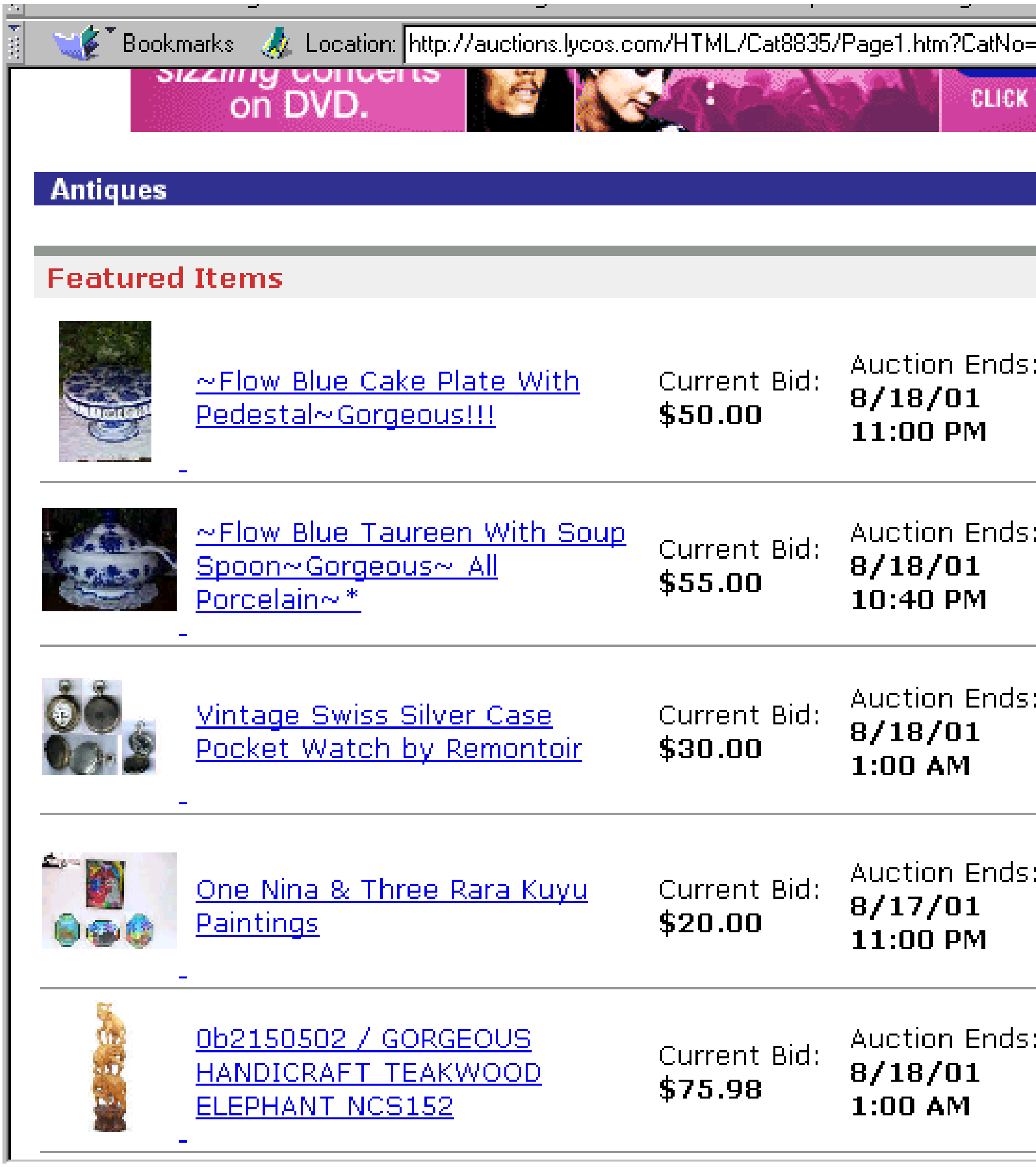
[ ] Search

Can't find it?
Try the Auction Ager

Narrow Your Search

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 Next>

Title	Status	Bids	Price
~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	📷 A	5	$50.00
~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*...	📷 R	3	$55.00
Vintage Swiss Silver Case Pocket Watch by Remontoir	📷 R	1	$30.00
One Nina & Three Rara Kuyu Paintings	★ 📷 A	-	$20.00
0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	📷 A	-	$75.98
0b2151103 / BEAUTIFUL HAND MADE TEAKWOOD ELEPHANT NCS152	📷 A	-	$75.98



Bookmarks Location: http://auctions.lycos.com/HTML/Cat8835/Page1.htm?CatNo=

Antiques

Featured Items

	Item	Current Bid	Auction Ends
	~Flow Blue Cake Plate With Pedestal~Gorgeous!!!	Current Bid: $50.00	Auction Ends: 8/18/01 11:00 PM
	~Flow Blue Taureen With Soup Spoon~Gorgeous~ All Porcelain~*	Current Bid: $55.00	Auction Ends: 8/18/01 10:40 PM
	Vintage Swiss Silver Case Pocket Watch by Remontoir	Current Bid: $30.00	Auction Ends: 8/18/01 1:00 AM
	One Nina & Three Rara Kuyu Paintings	Current Bid: $20.00	Auction Ends: 8/17/01 11:00 PM
	0b2150502 / GORGEOUS HANDICRAFT TEAKWOOD ELEPHANT NCS152	Current Bid: $75.98	Auction Ends: 8/18/01 1:00 AM

Aug 2001

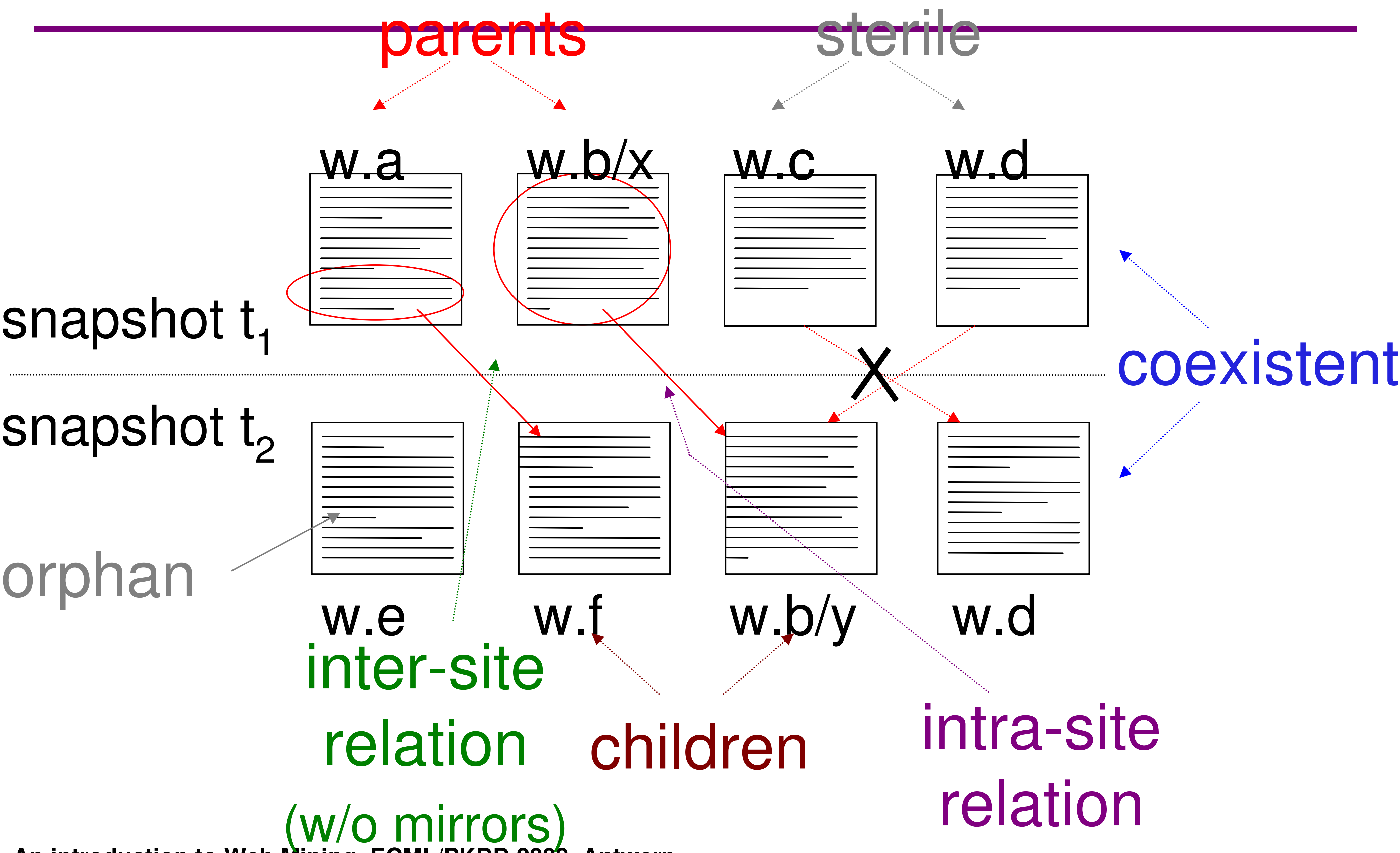
Motivation of near-duplicate detection

- Why detect mirrors?
  - Smart crawling
    - Fetch from the fastest or freshest server
    - Avoid duplication
  - Better connectivity analysis
    - Combine inlinks
    - Avoid double counting outlinks
  - Redundancy in result listings
    - “If that fails you can try: <mirror>/samepath”
  - Proxy caching

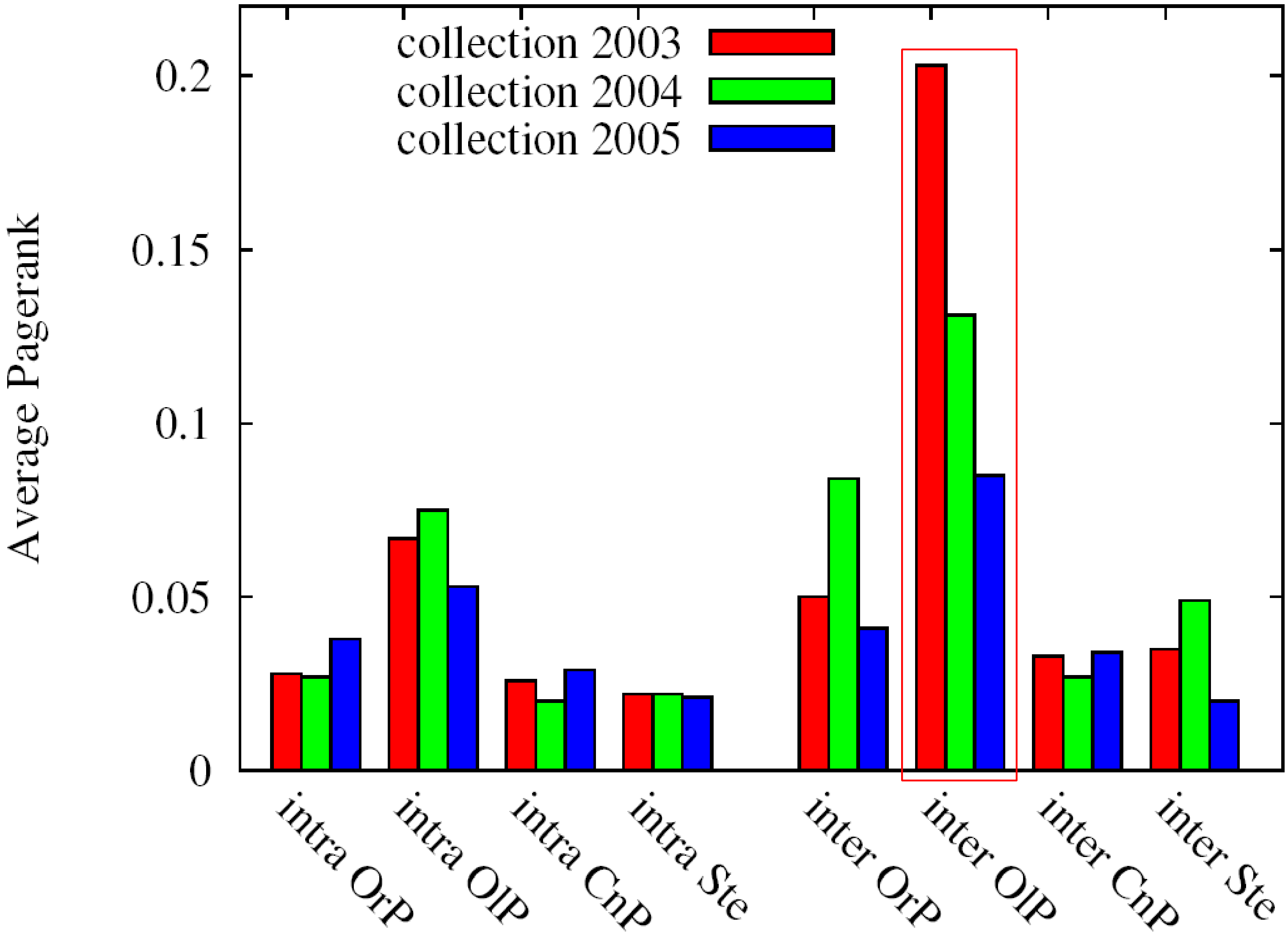
Study genealogy of the Web

- [Baeza-Yates et al., 2008]
- New pages copy content from existing pages
- Web genealogy study:
  - How textual content of source pages (parents) are reused to compose part of new Web pages (children)
  - Not near-duplicates, as similarities of short passages are also identified
- How can search engines benefit?
  - By associating more relevance to a parent page?
  - By trying to decrease the bias?

Web genealogy

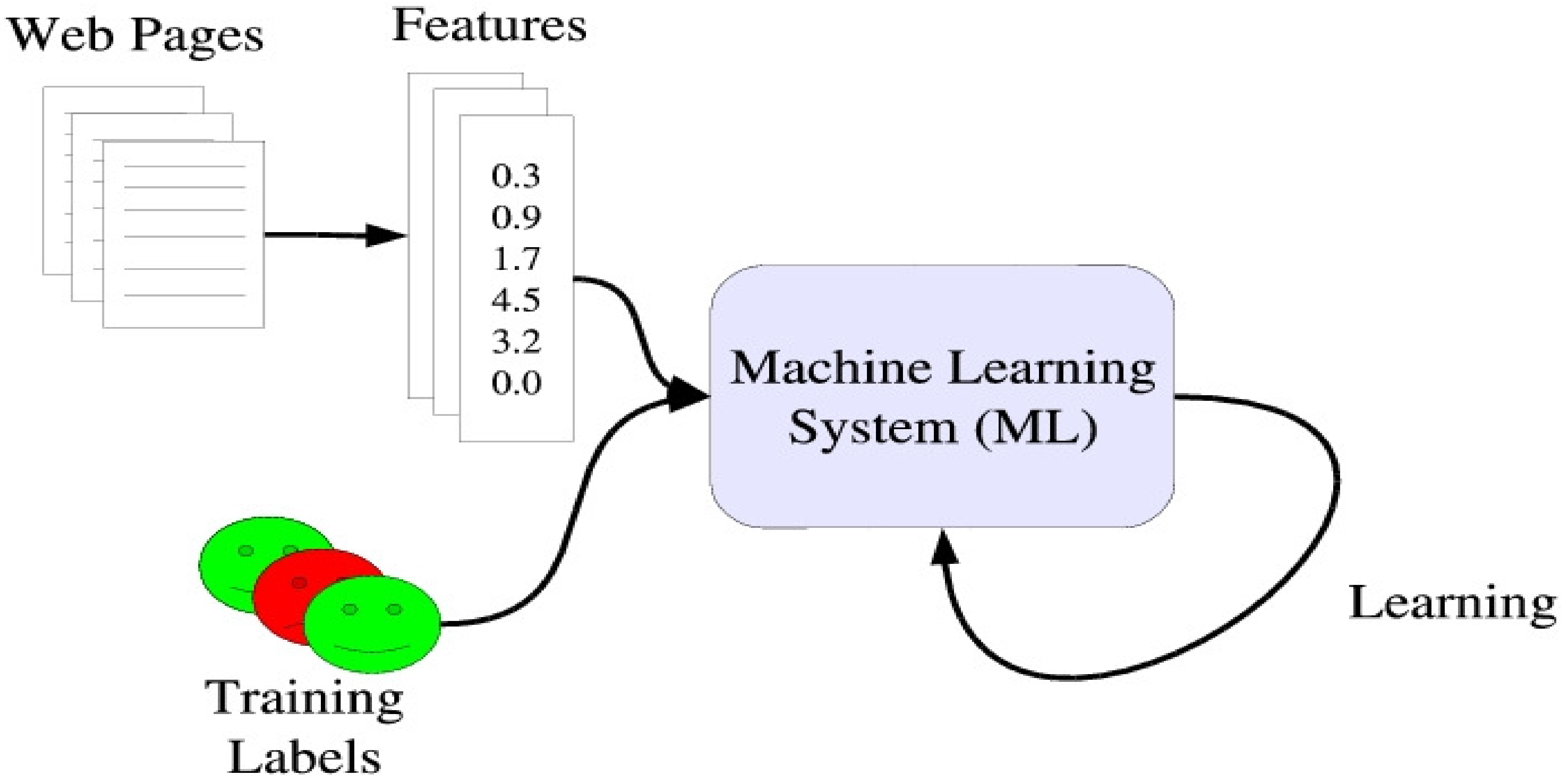


Pagerank for each component



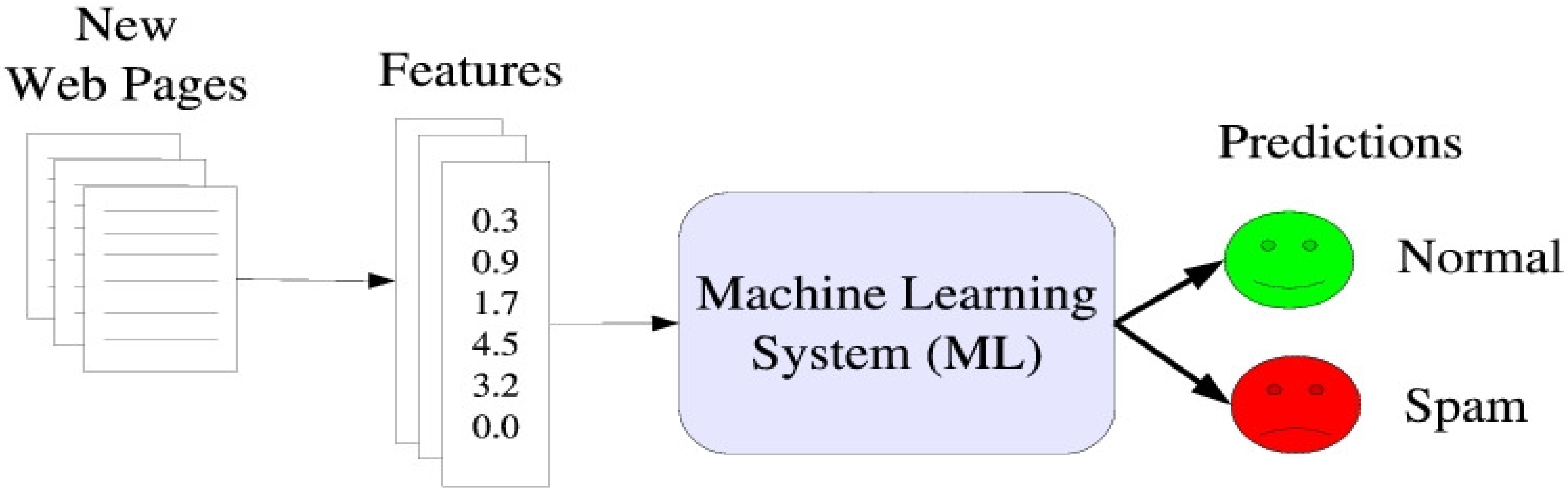
Content-based spam detection

- **Machine-learning approach --- training**



Content-based spam detection

- **Machine-learning approach --- prediction**



The dataset

- Label “spam” nodes on the host level
  - agrees with existing granularity of Web spam
- Based on a crawl of .uk domain from May 2006
- 77.9 million pages
- 3 billion links
- 11,400 hosts

The dataset

- 20+ volunteers tagged a subset of host
- Labels are “spam”, “normal”, “borderline”
- Hosts such as .gov.uk are considered “normal”
- In total 2,725 hosts were labelled by at least two judges
- hosts in which both judges agreed, and “borderline” removed
- Dataset available at

  http://www.yr-bcn.es/webspam/

Content-based features

- Number of words in the page
- Number of words in the title
- Average word length
- Fraction of anchor text
- Fraction of visible text

See also [Ntoulas et al., 06]

Content-based features
Entropy related

- Let $T = \{ (w_1, p_1), ..., (w_k, p_k) \}$ the set of trigrams in a page, where trigram w_i has frequency p_i
- Features:

✓ Entropy of trigrams: $H = -Sum_i\ p_i\ log(p_i)$

✓ Independent trigram likelihood: $- (1/k)\ Sum_i\ log(p_i)$

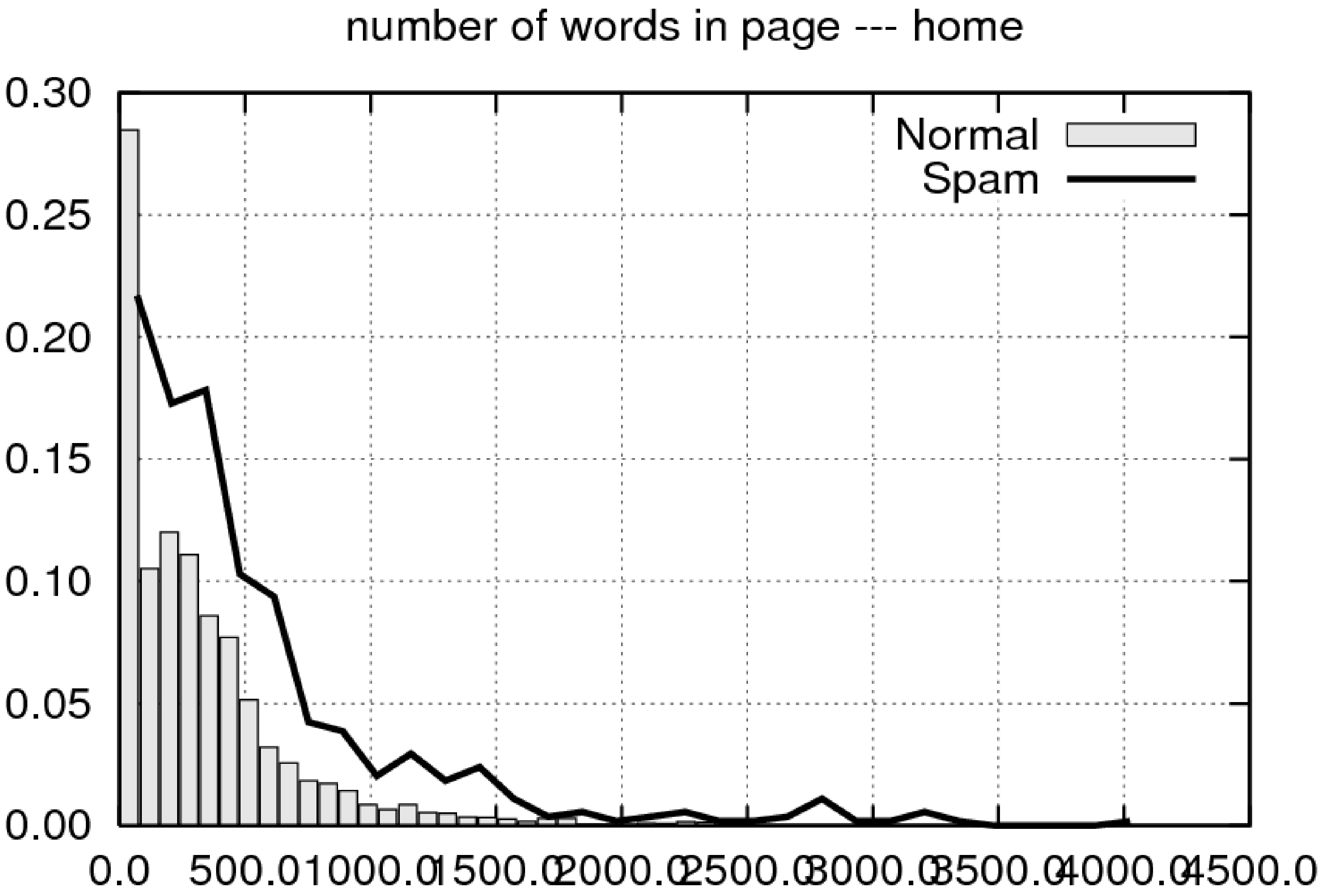
✓ Also, compression rate, as measured by bzip

Content-based features related to popular keywords

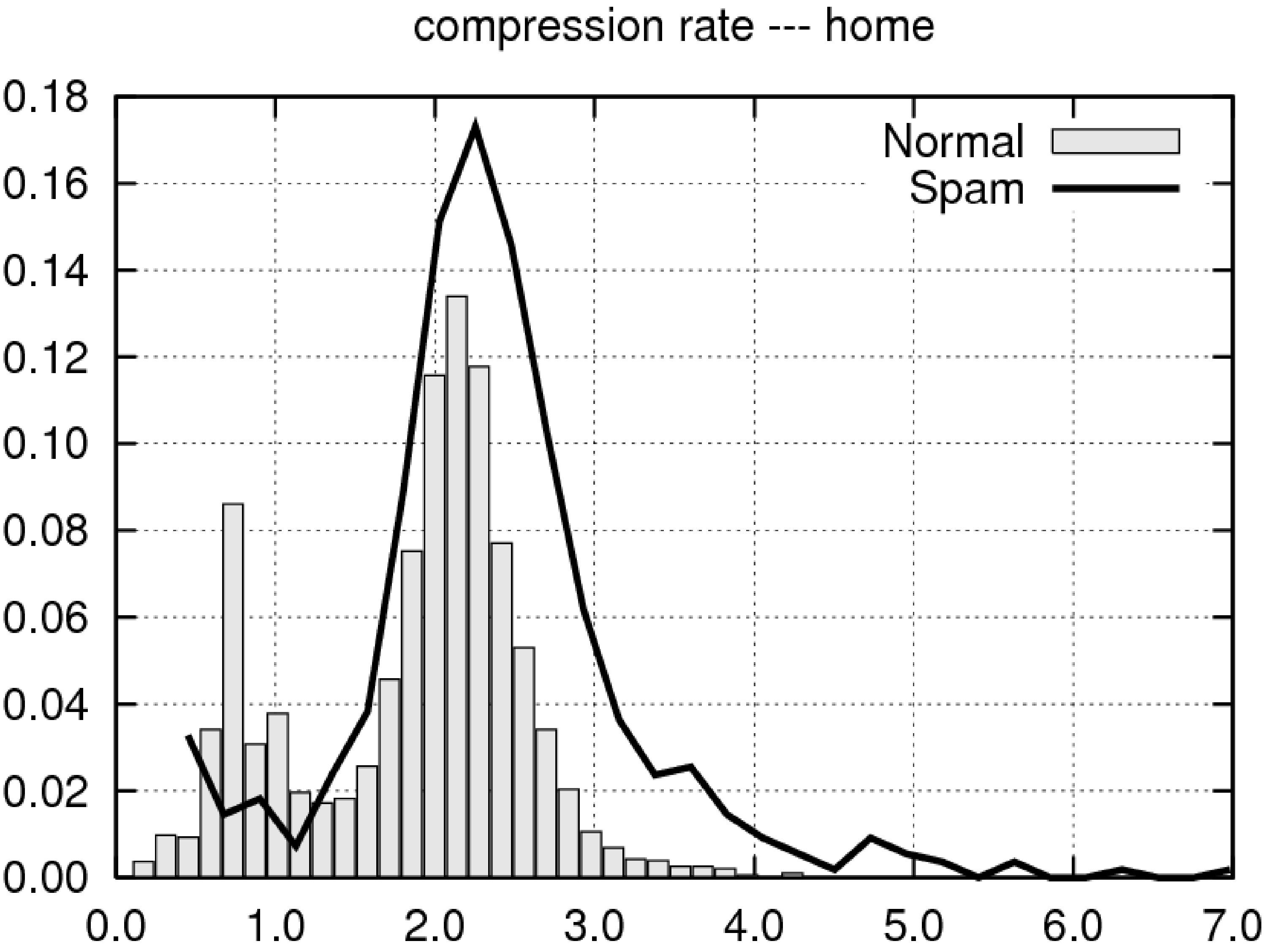
- *F* set of most frequent terms in the collection
- *Q* set of most frequent terms in a query log
- *P* set of terms in a page
- Features:

✓ Corpus “precision”	*| P intersect F | / | P |*
✓ Corpus “recall”	*| P intersect F | / | F |*
✓ Query “precision”	*| P intersect Q | / | P |*
✓ Query “recall”	*| P intersect Q | / | Q |*

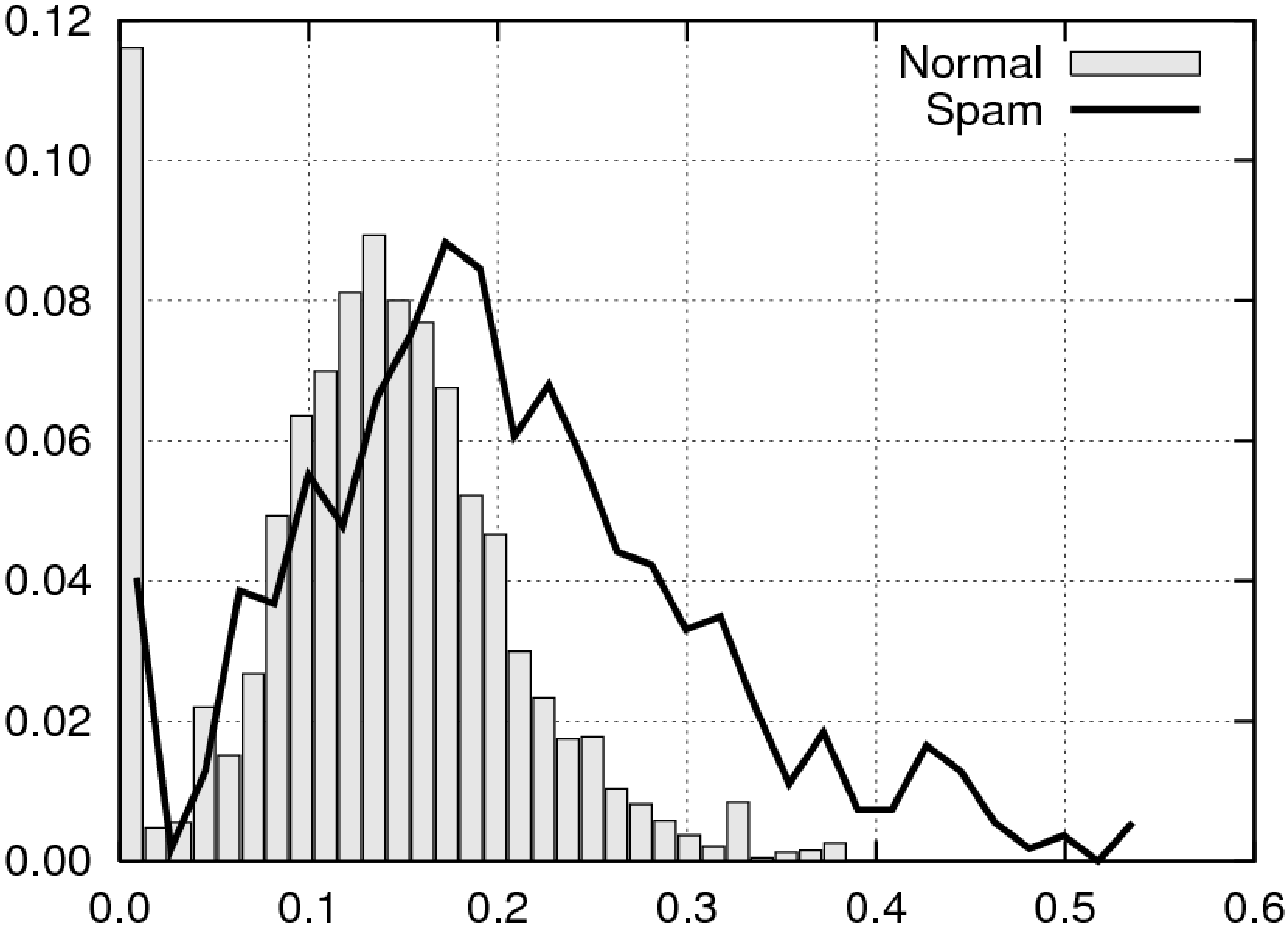
Content-based features number of words in home page



Content-based features compression rate



Content-based features
Query precision



The classifier

- C4.5 decision tree with bagging and cost weighting for class imbalance

- With content-based features achieves:

– True positive rate:	64.9%
– False positive rate:	3.7%
– F-Measure:	0.683

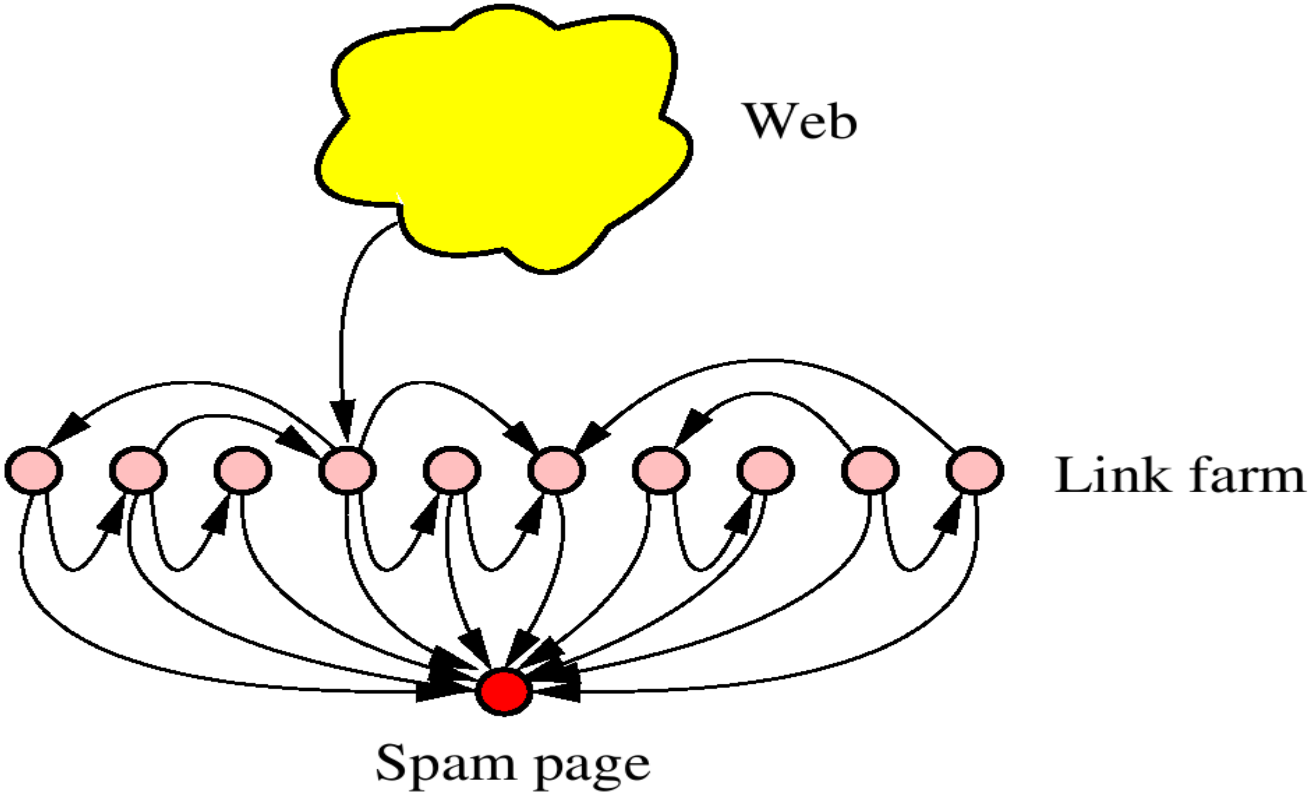
Structure and link analysis

- Link-based spam detection

Link-based spam detection

- Link farms used by spammers to raise popularity of spam pages
- Link farms and other spam strategies leave traces on the structure of the web graph
- Dependencies between neighbouring nodes of the web graph are created
- Naturally, spammers try to remove traces and dependencies

Link farms



- Single-level link farms can be detected by searching for nodes sharing their out-links
- In practice more sophisticated techniques are used

Link-based features
Degree related

- in-degree
- out-degree
- edge reciprocity
  - – number of reciprocal links
- assortativity
  - – degree over average degree of neighbors

Link-based features
PageRank related

- PageRank
- indegree/PageRank
- outdegree/PageRank
- ...
- Truncated PageRank [Becchetti et al., 2006]
  - A variant of PageRank that diminishes the influence of a page the PageRank score of its neighbors
- TrustRank [Gyongyi et al., 2004]
  - As PageRank but with teleportation at Open Directory pages

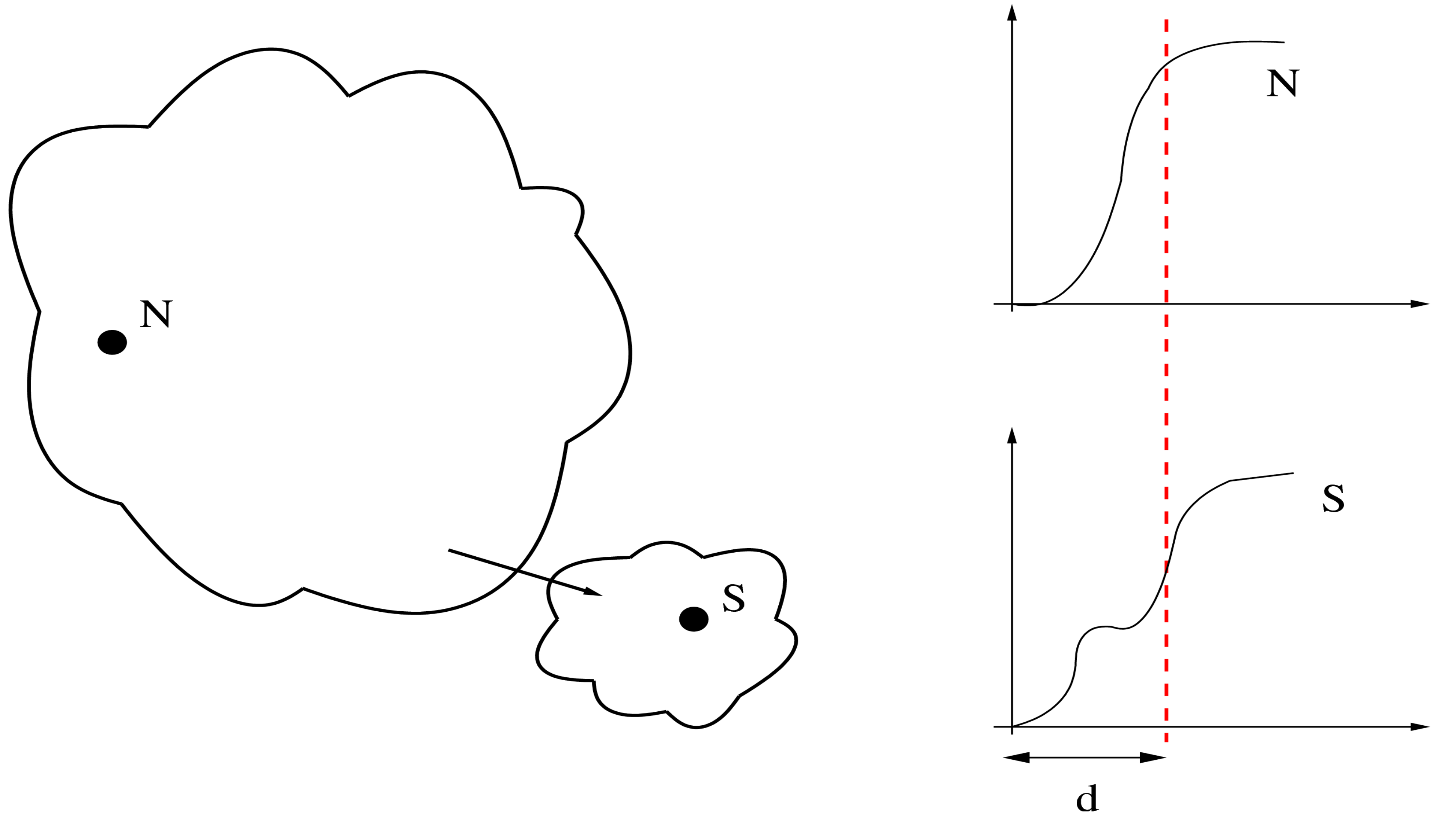
Link-based features
Supporters

- Let *x* and *y* be two nodes in the graph
- Say that *y* is a *d*-supporter of *x*, if the shortest path from *y* to *x* has length at most *d*
- Let $N_d(x)$ be the set of the *d*-supporters of *x*
- Define bottleneck number of *x*, up to distance *d* as

$$b_d(x) = min_{j <= d} \; N_j(x)/N_{j-1}(x)$$

- minimum rate of growth of the neighbors of *x* up to a certain distance

Link-based features
Supporters



Link-based features
Supporters

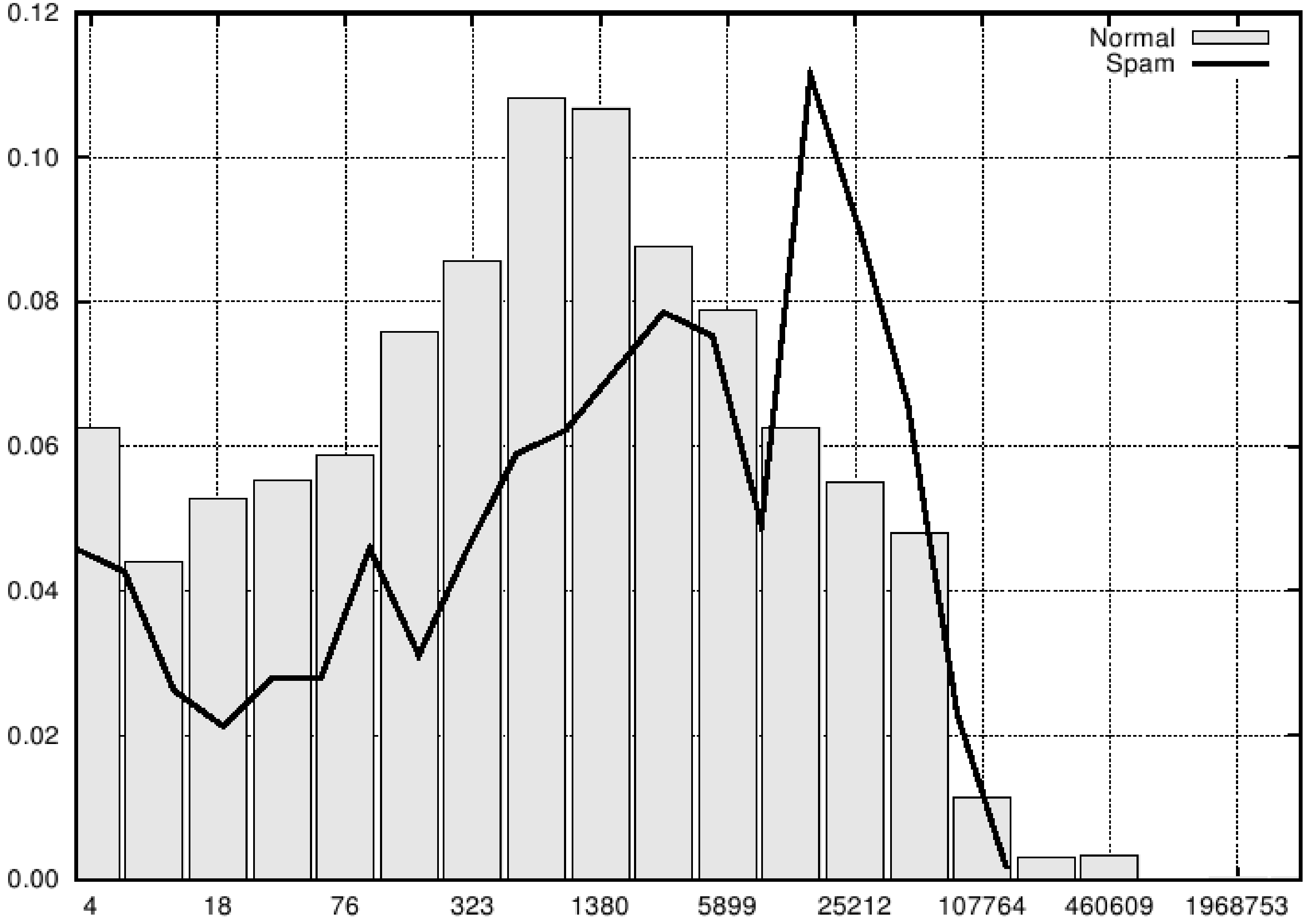
- How to compute the supporters?
- Utilize *neighborhood function*

$$N(h) = |\{ (u,v) \mid d(u,v) <= h \}| = \sum_u N(u,h)$$

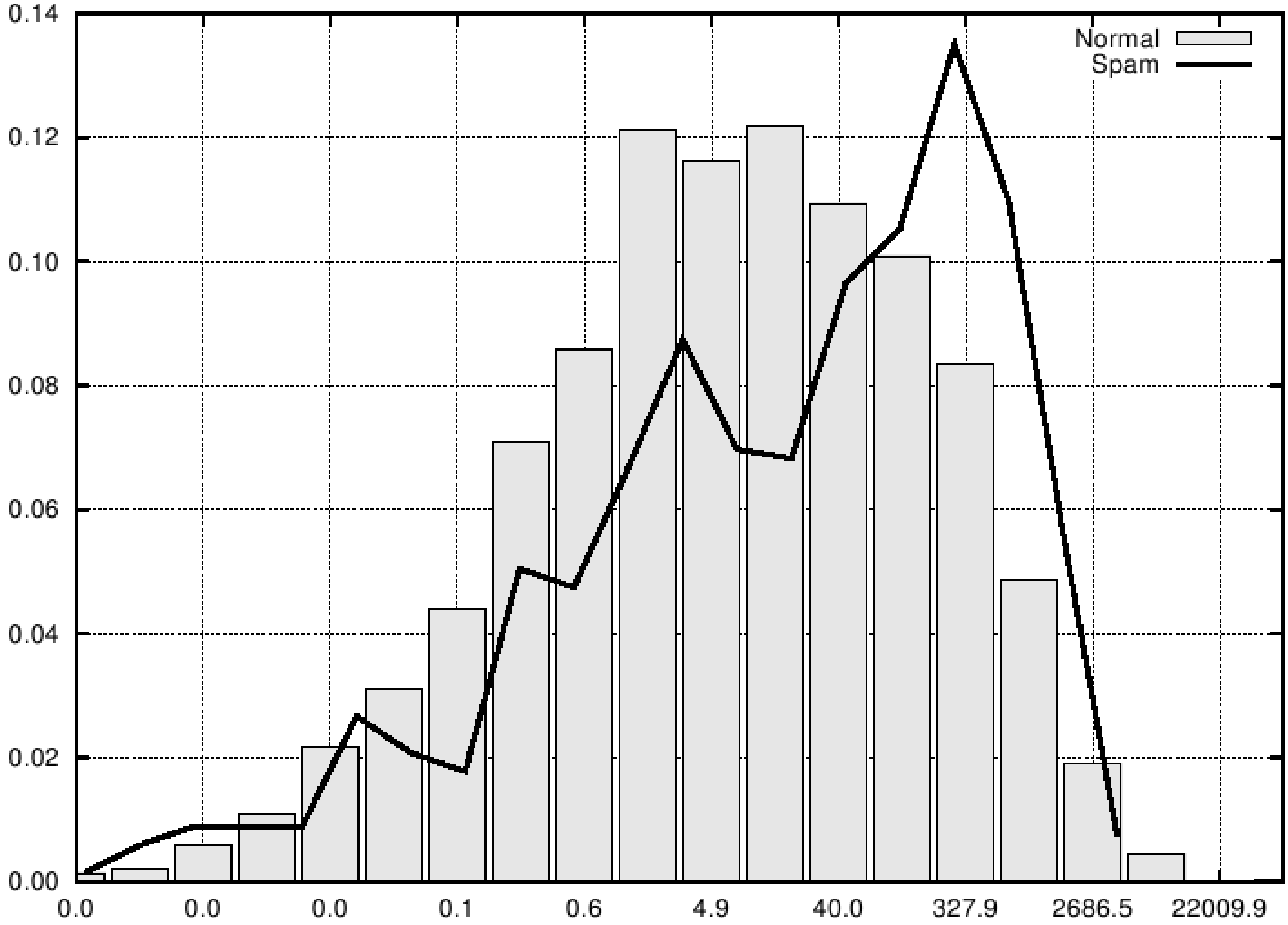
Palme

- and ANF algorithm [Palmer et al., 2002]
- Probabilistic counting using Flajolet-Martin sketches or other data-stream technology
- Can be done with a few passes and exchange of sketches, instead of executing BFS from each node

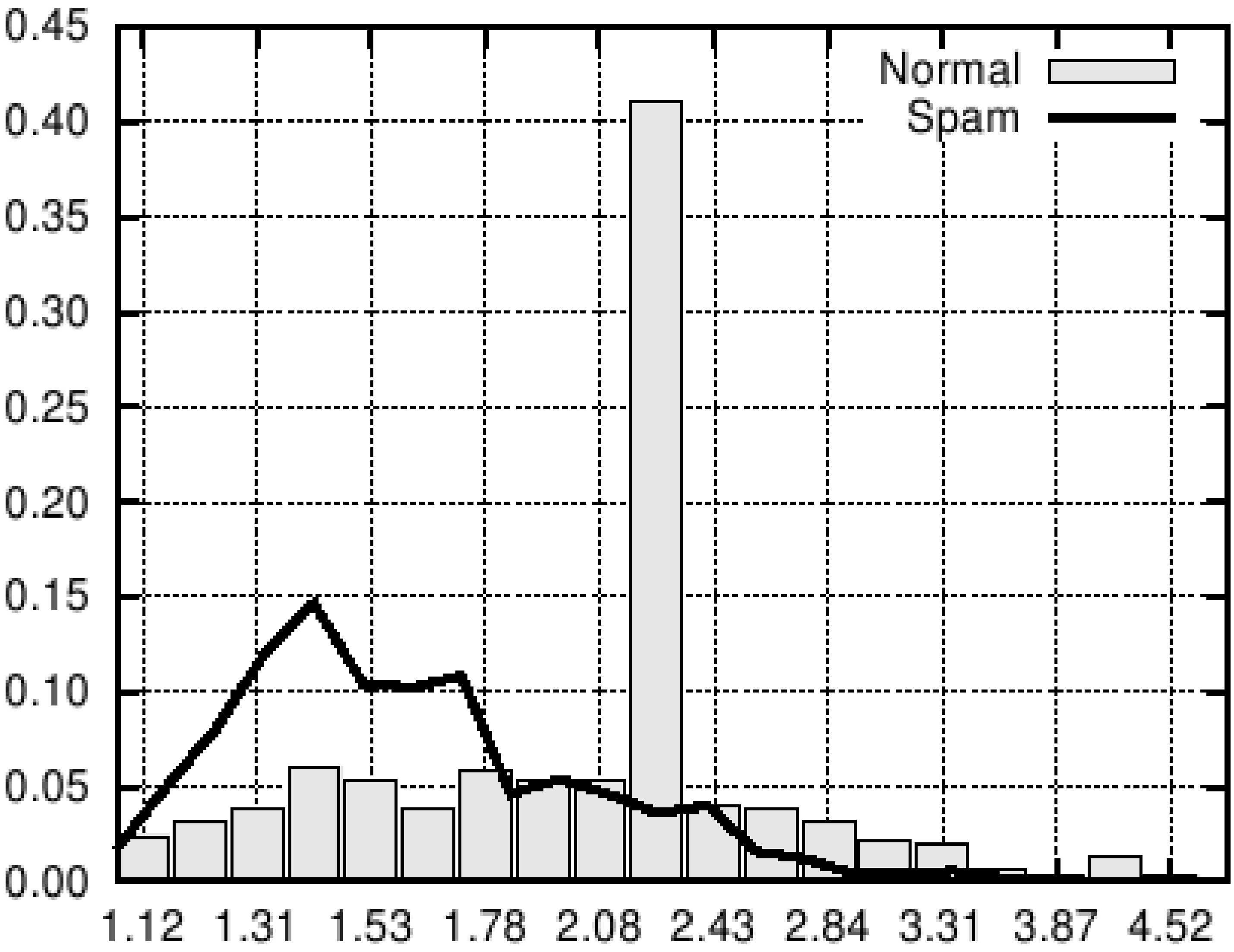
Link-based features - In-degree



Link-based features - Assortativity



Link-based features - Supporters

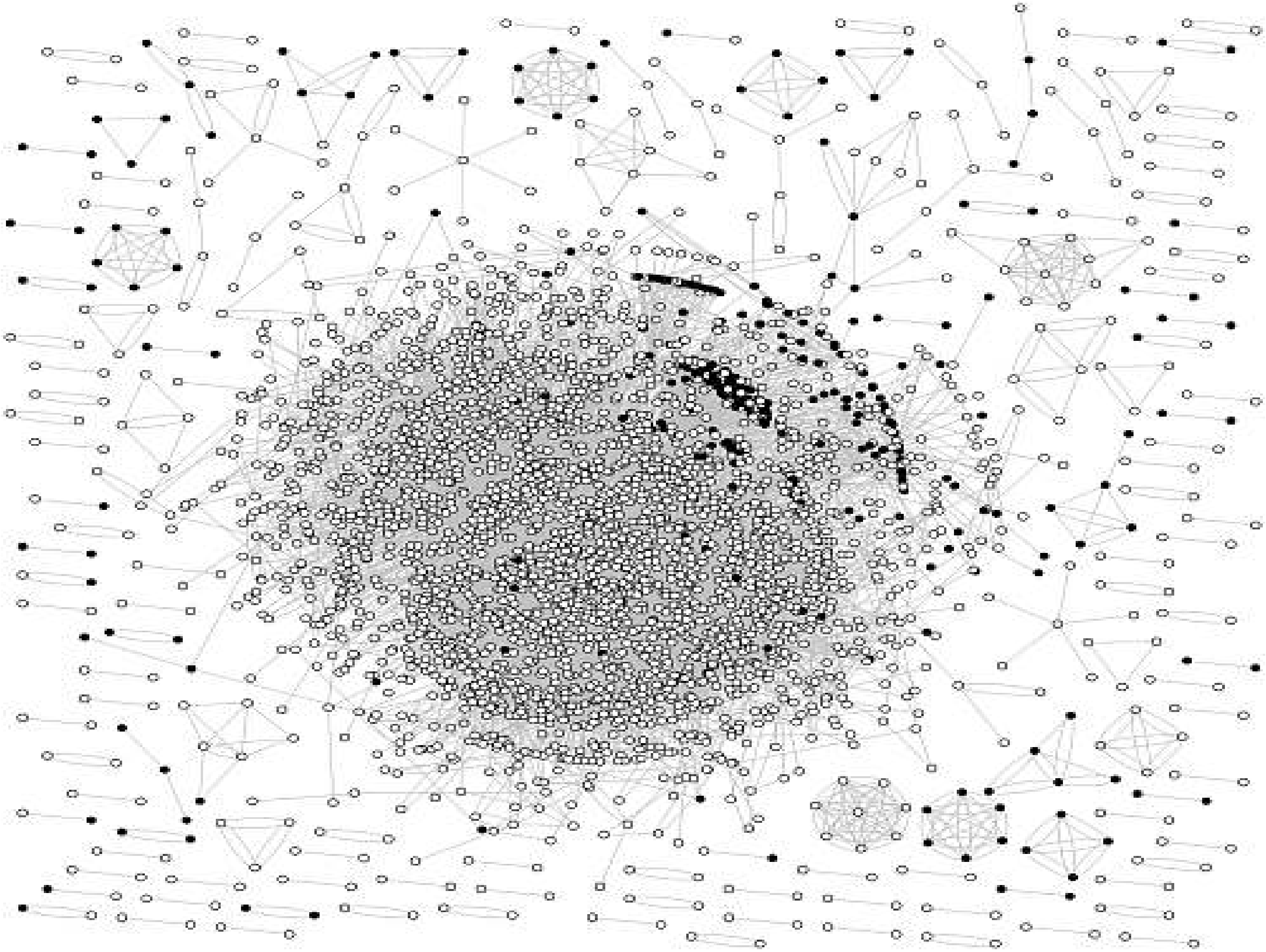


The classifier
Combining features

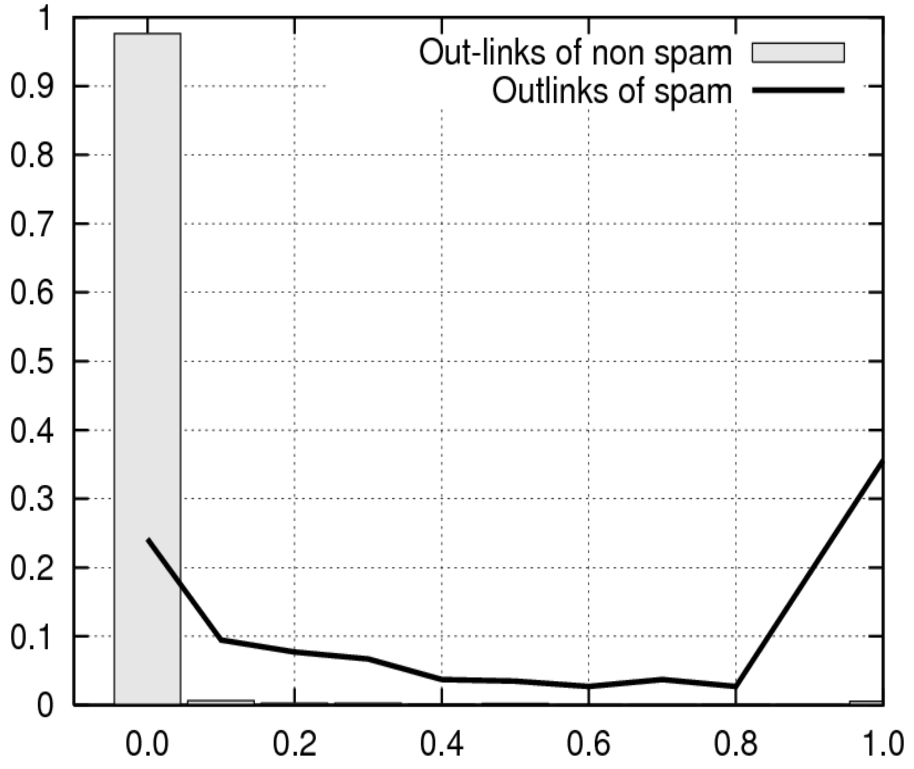
- C4.5 decision tree with bagging and cost weighting for class imbalance

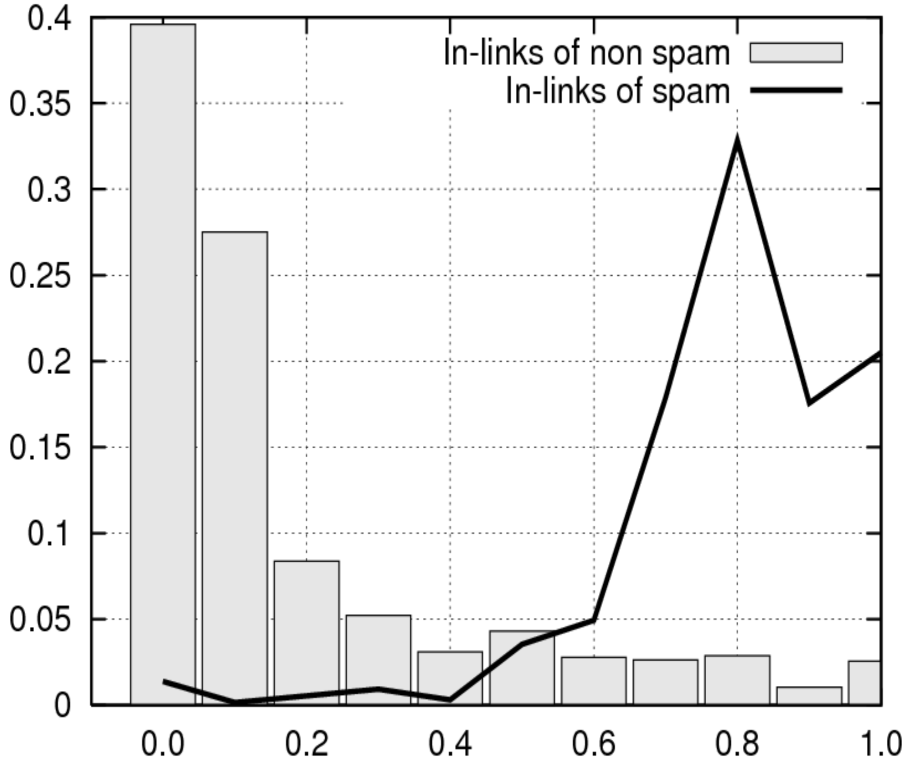
features:	**Content**	**Link**	**Both**
True positive rate:	64.9%	79.4%	78.7%
False positive rate:	3.7%	9.0%	5.7%
F-Measure:	0.683	0.659	**0.723**

Dependencies among spam nodes



Dependencies among spam nodes





- **Spam nodes in out-links**
- **Spam nodes from in-links**

Exploiting dependencies

- Use a dataset with labeled nodes
- Extract content-based and link-based features
- Learn a classifier for predicting spam nodes independently
- Exploit the graph topology to improve classification
  - Clustering
  - Propagation
  - Stacked learning

Exploiting dependencies Clustering

- Let $G=(V,E,w)$ be the host graph
- Cluster G into m disjoint clusters $C_1,\ldots,C_m$
- Compute $p(C_i)$, the fraction of nodes classified as spam in cluster C_i
  - if $p(C_i) > t_u$ label all as spam
  - if $p(C_i) < t_l$ label all as non-spam
- A small improvement:

	Baseline	***Clustering***
True positive rate:	*78.7%*	*76.9%*
False positive rate:	*5.7%*	*5.0%*
F-Measure:	***0.723***	***0.728***

Exploiting dependencies Propagation

- Perform a random walk on thegraph
- With probability a follow a link
- With prob *1-a* jump to a random node labeled spam
- Relabel as spam every node whose stationary distribution component is higher than a threshold

- Improvement:

	Baseline	***Propagation***
True positive rate:	*78.7%*	*75.0%*
False positive rate:	*5.7%*	*4.3%*
F-Measure:	*0.723*	***0.733***

Exploiting dependencies
Stacked learning

- Meta-learning scheme [Cohen and Kou, 2006]
- Derive initial predictions
- Generate an additional attribute for each object by combining predictions on neighbors in the graph
- Append additional attribute in the data and retrain
- Let *p(h)* be the prediction of a classification algorithm for *h*
- Let *N(h)* be the set of pages related to *h*
- Compute:

$$f(h) = Sum_{g \text{ in } N(h)} \, p(g) \, / \, |N(h)|$$

- Add *f(h)* as an extra feature for instance *h* and retrain

Exploiting dependencies
Stacked learning

- **First pass**:

	Baseline	***in***	***out***	***both***
True positive rate:	*78.7%*	*84.4%*	*78.3%*	*85.2%*
False positive rate:	*5.7%*	*6.7%*	*4.8%*	*6.1%*
F-Measure:	***0.723***	***0.733***	***0.742***	***0.750***

- ***Second pass:***

	Baseline	***1st pass***	***2nd pass***
True positive rate:	*78.7%*	*85.2%*	*88.2%*
False positive rate:	*5.7%*	*6.1%*	*6.3%*
F-Measure:	***0.723***	***0.750***	***0.763***

Overall summary

- Open problems and challenges:
  - Modeling web graph and other web data
  - Model evolution
  - Data cleaning and anonymization
  - Improve IR relevance
  - Manage and integrate highly heterogeneous information: content, links, social links, tags, feedback, usage logs, wisdom of crowd, etc.
  - Design improved web applications
  - Battle adversarial attempts and collusions

Special thanks

- Carlos Castillo
- Alessandro Tiberi
- Barbara Poblete
- Alvaro Pereira

Thank you!