Mining the graph structures of the web

Aristides Gionis

Yahoo! Research, Barcelona, Spain, and
University of Helsinki, Finland

Summer School on Algorithmic Data Analysis (SADA07)
May 28 – June 1, 2007
Helsinki, Finland

What is on the Web?

Information + Porn + On-line casinos + Free movies + Cheap software + Buy a MBA diploma + Prescription - free drugs + V!-4-gra + Get rich now now now!!!

Graphic: www.milliondollarhomepage.com

Web spam

- Malicious attempts to influence the outcome of ranking algorithms
- Obtaining higher rank implies more traffic
- Cheap and effective method to increase revenue
- [Eiron et al., 2004] ranked 100 m pages according to PageRank: 11 out of 20 first were pornographic pages
- Spammers form an "active community"
- e.g., contest for who ranks higher for the query *"nigritude ultramarine"*

Web spam

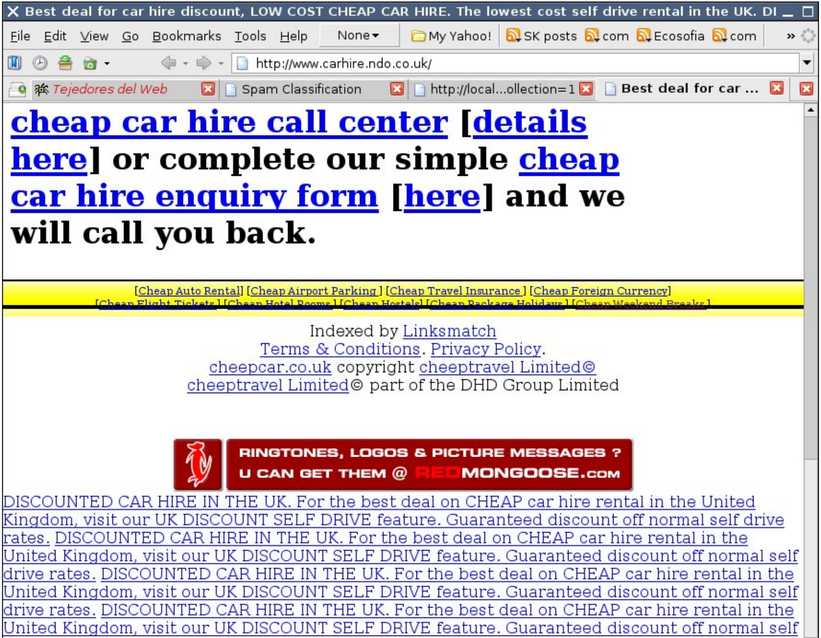
Adversarial relationship with search engines

- Users get annoyed
- Search engines waste resources

Web spam "techniques"

- ☑ Spamdexing
  - Keyword stuffing
  - Link farms
  - Scraper, "Made for Advertising" sites
  - Cloaking
- Click spam

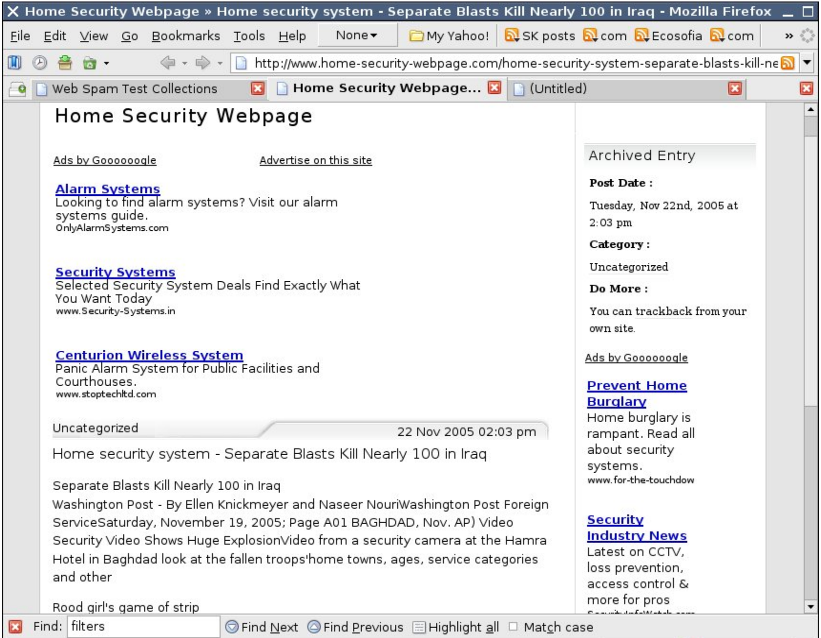
Typical web spam



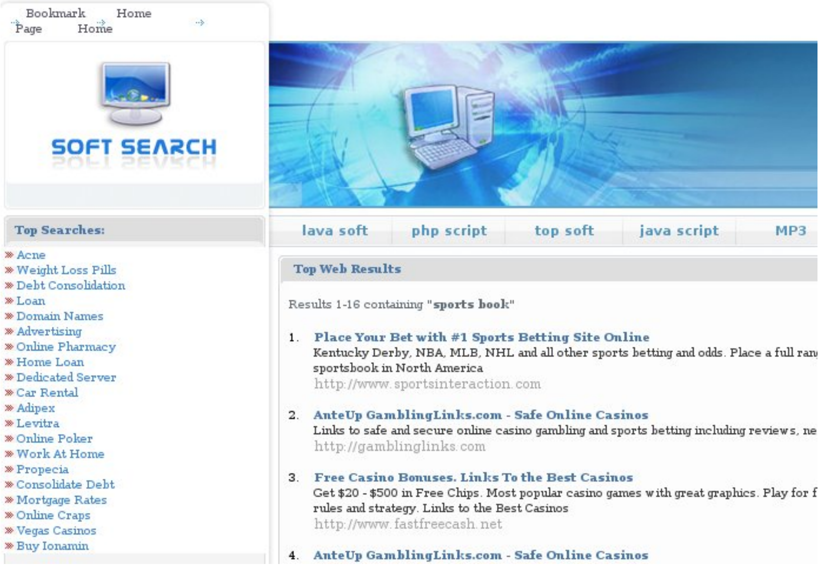
Hidden text


1 Stop Poker Games - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
None
My Yahoo! SK posts com Ecosofia com Diggs
http://www.1stoppoker.com/games.html
Web Spa... webspam ... http:...xhtml mp3 ware... 1 Stop P... Hidden Text "poker po...
Play Now! Million Dollar Guaranteed Tournament
Sign Up for 1 Stop's Newsletter!
Your E-Mail:
Subscribe!
Great Poker Tips, #8
There are only about 20 hands that are strong enough to play from an early position. Players are making a big mistake if they play weak or marginal hands without giving consideration to their position.
— Bill Burton, Get the Edge at Low Limit Texas Hold'Em, Bonus Books, 2002
Want the edge? Get the book!
Texas hold 'em (or simply hold 'em or holdem) is the most popular of the community card poker games. It is the most popular poker variant played in casinos in the western United States, and its no-limit form is used in the main event of the World Series of Poker (abbreviated WSOP), widely recognized as the world championship of the game.
Seven-card stud is a poker variant. Until the recent increase in popularity of Texas hold 'em, Seven-card stud was the most popular poker variant in home games across the United States, and in casinos in the eastern part of the country.
Omaha hold 'em (or Omaha holdem or simply Omaha) is a community card poker game based on Texas hold 'em. It was originally created as a high-hand only game, but a high-low split variant called "Omaha eight-or-better" has also become popular.
Five-card draw is often the first poker variant learned by most players, and is very common in home games although it is now rare in casino and tournament play. The lowball variations make more interesting games and are more commonly played in casinos. Two to eight players can play.
Source: Wikipedia, the free encyclopedia.
poker poker poker poker poker
poker poker poker poker poker poker poker poker poker
poker poker poker poker poker poker poker poker
pokerpoker poker poker poker poker poker poker poker
poker poker poker poker poker poker poker poker poker
pokerpoker poker poker poker poker poker poker poker
Scripts Currently Forbidden [<script>: 1] [J+F+P: 0]
Options...

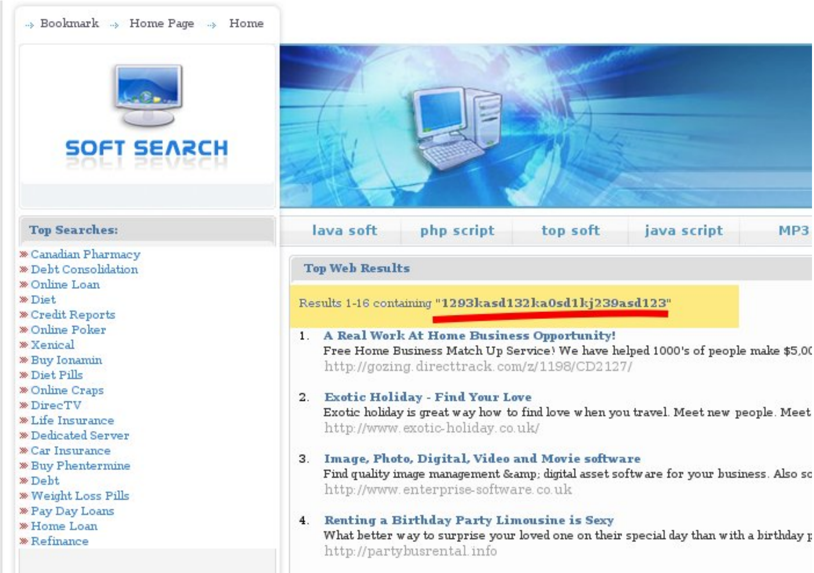
Made for advertising



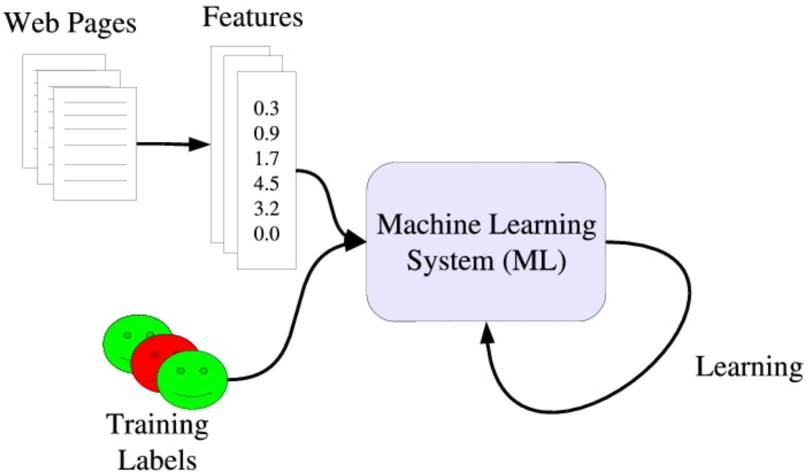
Search engine?



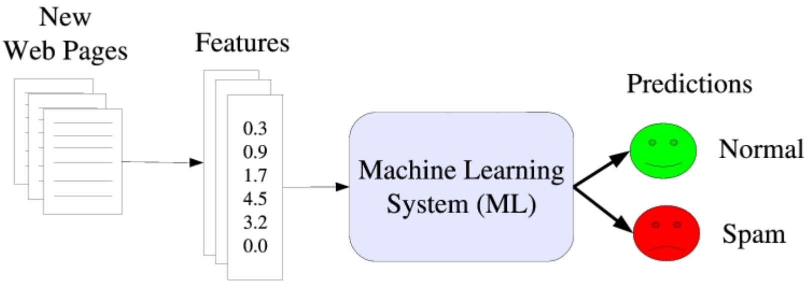
Fake search engine



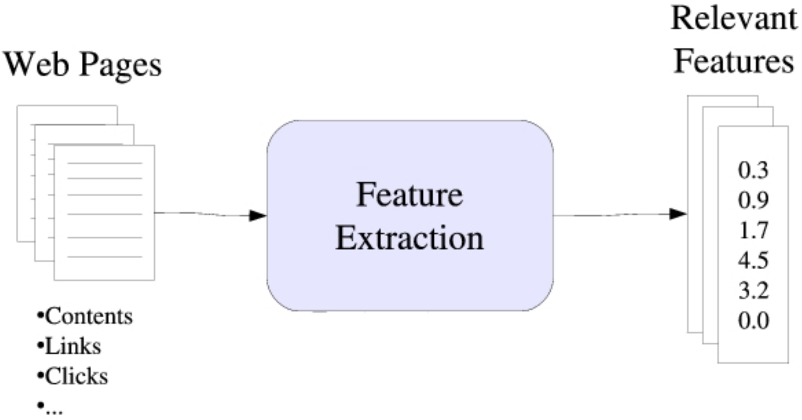
Machine learning



Machine learning



Feature extraction



Challenges: machine learning

Machine learning challenges:

- Learning with interdependent variables (graph)
- Learning with few examples
- Scalability

Challenges: information retrieval

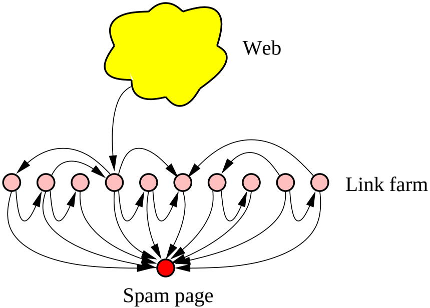
Information retrieval challenges:

- Feature extraction: which features?
- Feature aggregation: page/host/domain
- Recall/precision tradeoffs
- Scalability

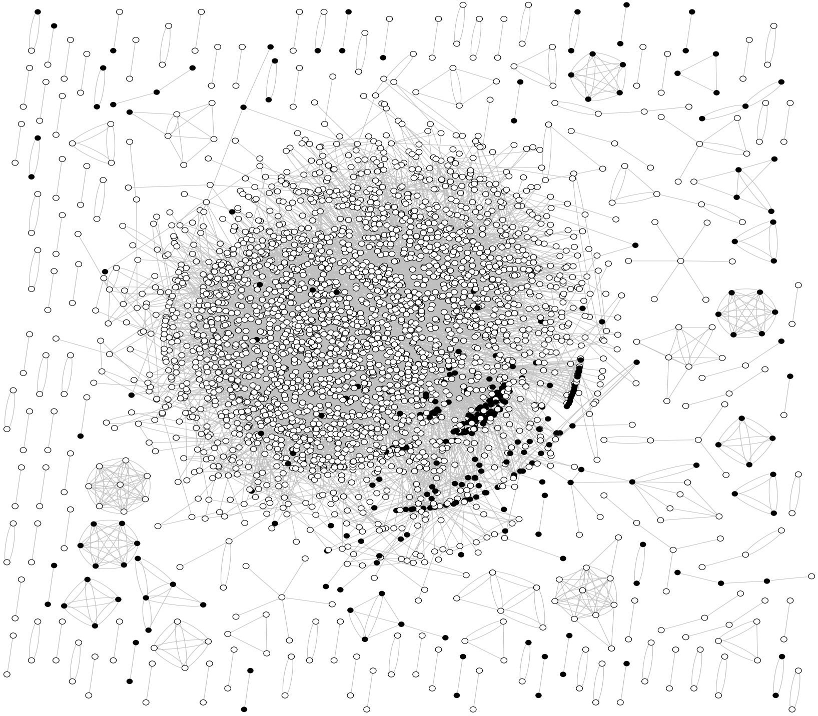
Learning with dependent variables

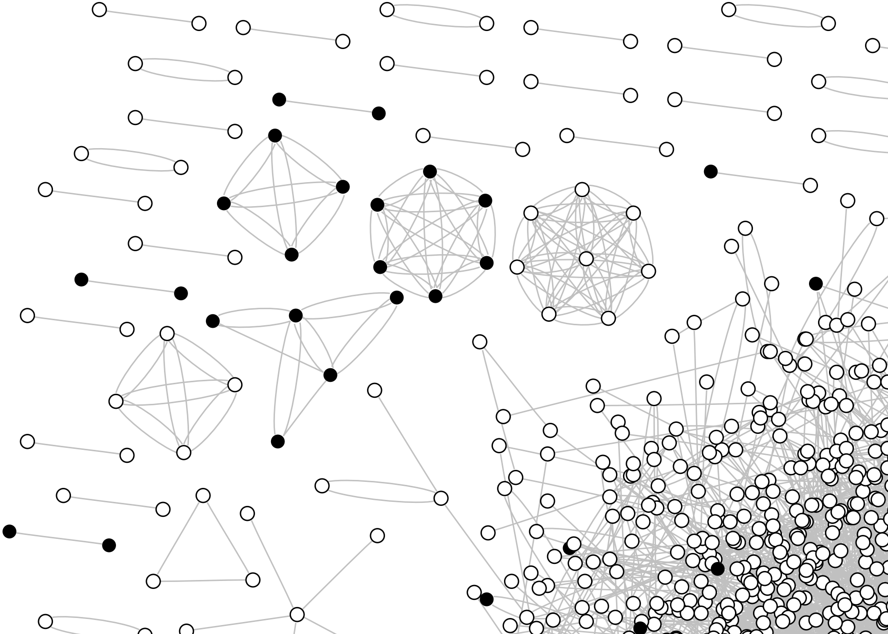
Dependency among spam nodes

- Link farms used to raise popularity of spam pages

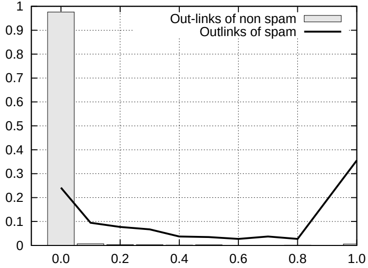


- Single-level link farms can be detected by searching for nodes sharing their out-links [Gibson et al., 2005]
- In practice more sophistocated techniques are used

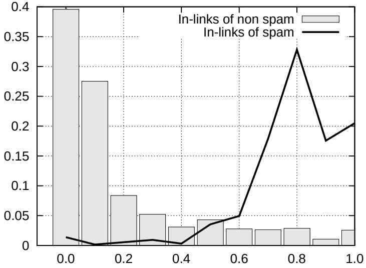




Dependencies among spam nodes



Spam nodes in out-links



Spam nodes from in-links

Overview of spam detection

- 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

The dataset

- Label “spam” nodes on the host level
  agrees with existing granularity of Web spam
- Based on a crawl of `.uk` domain done in 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 labeled by at least two judges, hosts in which both judges agreed, and "borderline" removed
- Dataset available at `http://www.yr-bcn.es/webspam/`

Features

- Link-based features extracted from the host graph
- Content-based extracted from individual pages
- Aggregate content features at the host level

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., 2006]

Content-based features (entropy related)

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

Features:

- Entropy of trigrams $H = -\sum_{w_i \in T} p_i \log p_i$
- Independent trigram likelihood $I = -\frac{1}{k} \sum_{w_i \in T} \log p_i$
- Also, compression rate, as measured by `bzip`

Content-based features (related to popular keywords)

F set of most frequent terms in the collection

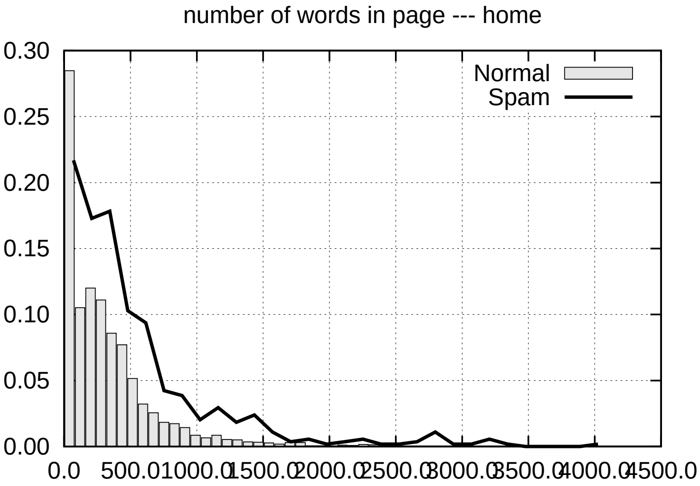
Q set of most frequent terms in a query log

P set of terms in a page

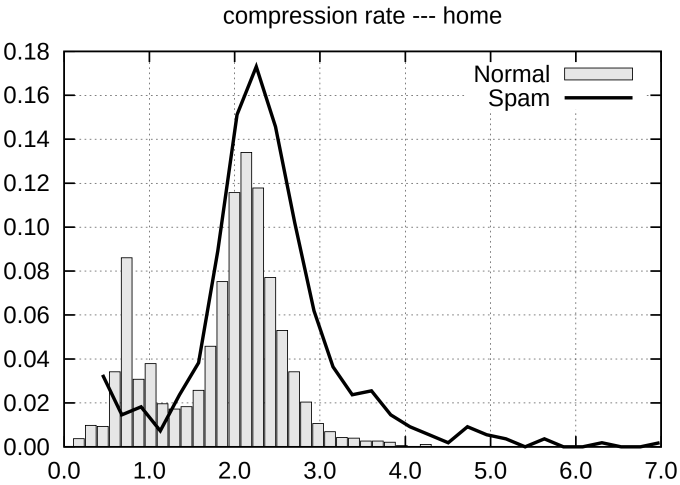
Features:

- Corpus “precision” $|P \cap F| / |P|$
- Corpus “recall” $|P \cap F| / |F|$
- Query “precision” $|P \cap Q| / |P|$
- Query “recall” $|P \cap Q| / |Q|$

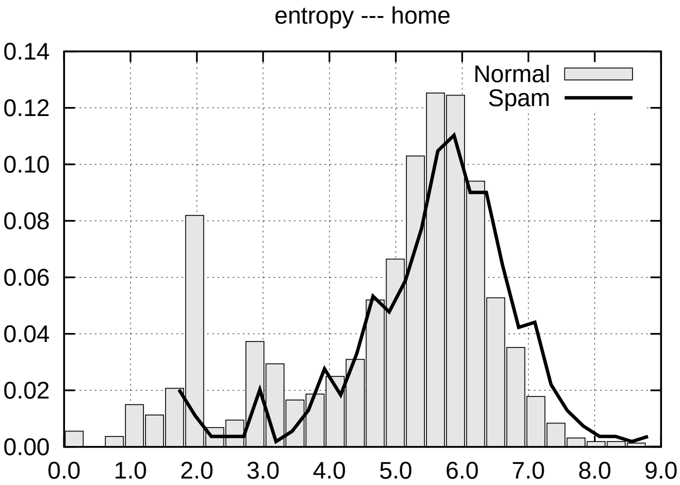
Content-based features – Number of words in the host home page



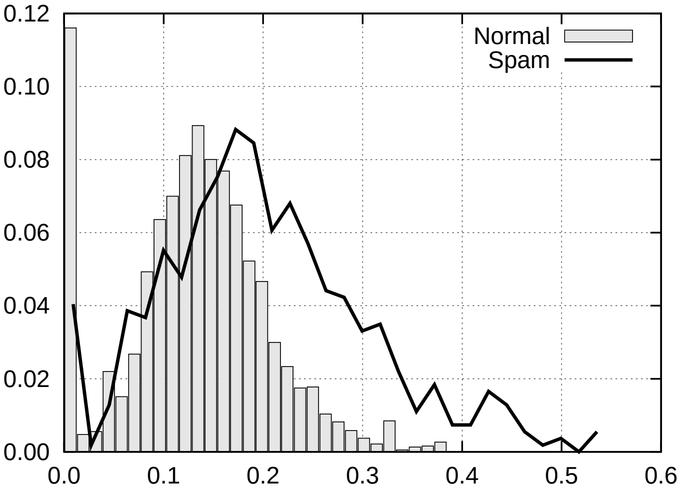
Content-based features – Compression rate



Content-based features – Entropy



Content-based features – Query precision



Link-based features – Degree related

On the host graph

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

Link-based features – PageRank related

- PageRank
- Truncated PageRank [Becchetti et al., 2006]
  - a variant of PageRank that diminishes the influence of a page to the PageRank score of its neighbors
- TrustRank [Gyöngyi et al., 2004]
  - as PageRank but deportation vector at `Open Directory` pages

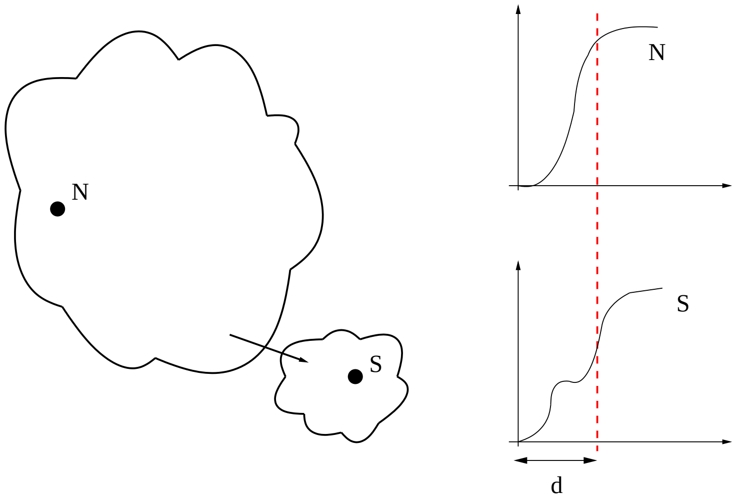
Link-based features – Supporters

- 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

$$b_d(x) = \min_{j \leq d} \{ \frac{N_j(x)}{N_{j-1}(x)} \}$$

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

Link-based features – Supporters



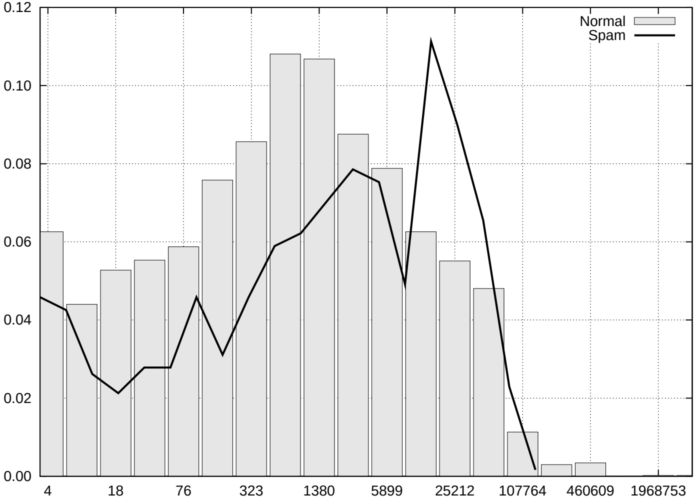
Link-based features – Supporters

- How to compute the supporters?
- Remember *Neighborhood function*

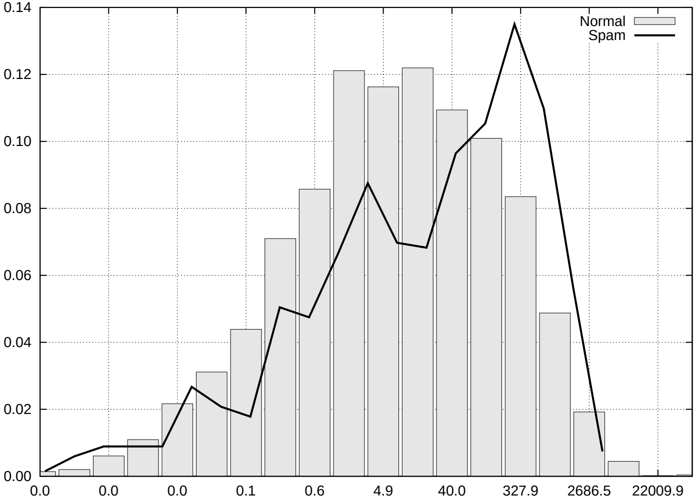
$$N(h) = |\{(u,v) \mid d(u,v) \leq h\}| = \sum_u N(u,h)$$

  and ANF algorithm
- Probabilistic counting using basic Flajolet-Martin sketches or other data-stream technology

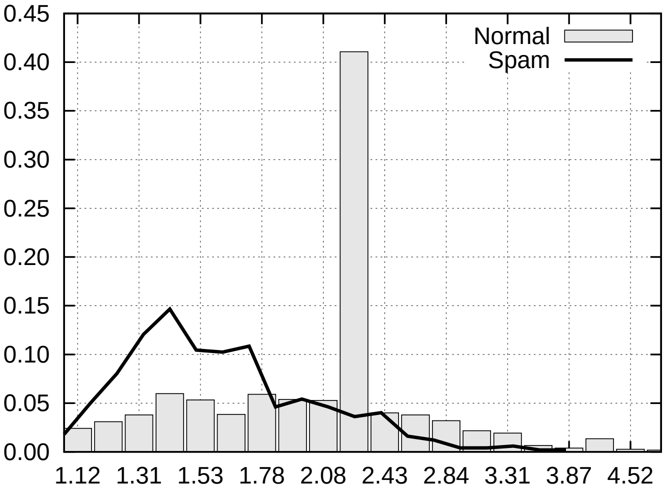
Link-based features – In degree



Content-based features – Assortativity



Content-based features – Supporters



Putting everything together

- 140 link-based features for each host
- 24 content-based features for each page
- aggregate content features at the host level by considering features of
  - host home page
  - host page with max PageRank
  - average and standard deviation of the features of all pages in the host
- $140 + 4 \times 24 = 236$ features in total

The measures

		Prediction	
		Non-spam	Spam
True Label	Non-spam	a	b
	Spam	c	d

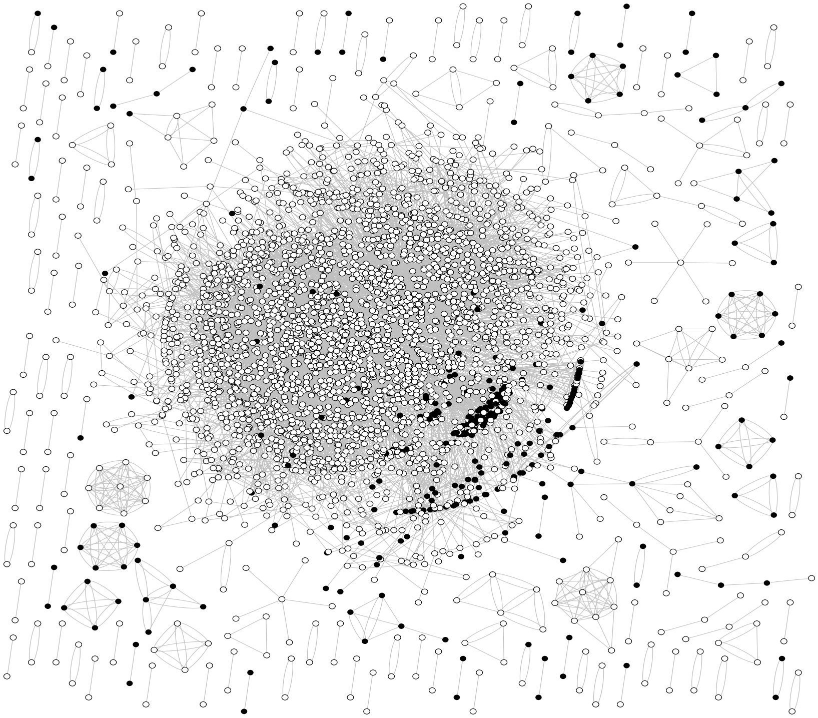
- Recall: $R = \frac{d}{c+d}$
- False positive rate: $P = \frac{b}{b+a}$
- F-measure: $F = 2\frac{PR}{P+R}$

The classifier

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

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

The resulting tree uses 45 features (18 content)



Exploit topological 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**

Exploit topological dependencies – Propagation

- Perform a random walk on the graph
- With probability α follow a link
- With probability $1 - \alpha$ jump to a random node labeled as spam
- Relabel as spam every node whose stationary-distribution component is higher than a threshold
  - threshold learned from the training data

Improvement

	Baseline	Fwds.	Backwds.	Both
True positive rate	78.7%	76.5%	75.0%	75.2%
False positive rate	5.7%	5.4%	4.3%	4.7%
F-Measure	0.723	0.716	**0.733**	0.724

Exploit topological 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

Exploit topological dependencies – Stacked learning

- Let $p(h) \in [0..1]$ be the prediction of a classification algorithm for a host h
- Let $N(h)$ be the set of pages related to h (in some way)
- Compute

$$f(h) = \frac{\sum_{g \in N(h)} p(g)}{|N(h)|}$$

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

Exploit topological dependencies – Stacked learning

	Baseline	Avg. of in	Avg. of out	Avg. of 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	First pass	Second pass
True positive rate	78.7%	85.2%	88.4%
False positive rate	5.7%	6.1%	6.3%
F-Measure	0.723	0.750	**0.763**

Spam detection – Conclusions

- Spam detection as a problem of learning in a graph
- Same framework has other applications, e.g., topical classification of documents in a hyper-linked environment

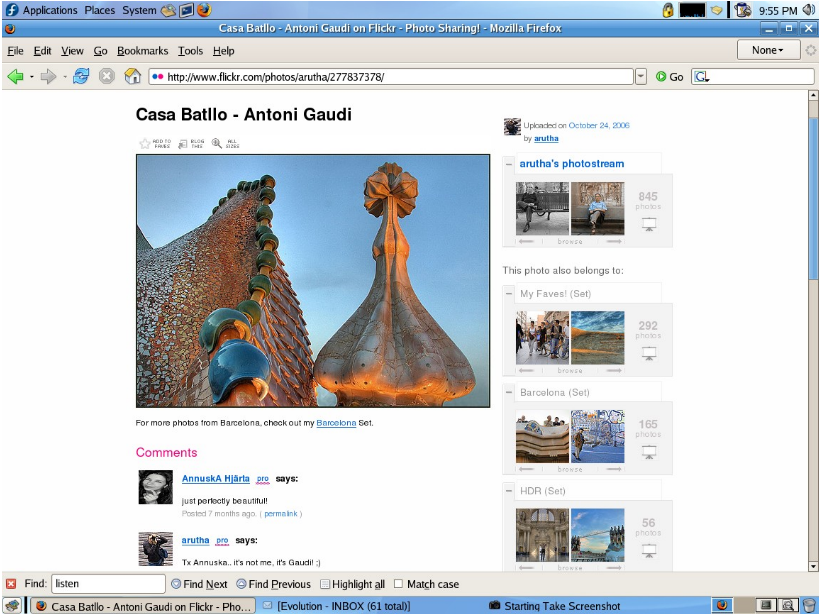
1. Web spam
2. Web spam detection
3. Predicting popularity

Predicting popularity

- Dynamic environment in which new items are published
- Items are published by “authors”
- Authors provide feedback to other authors’ items
- Feedback can be either explicit or implicit

  positive or negative vote, link, citation
- Natural notion of successful items
- Question: Can we predict which items will be successful?

Application I – Photo sharing



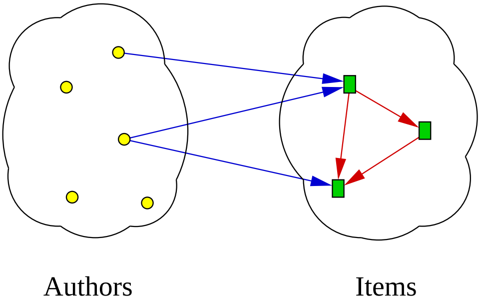
Application I – Photo sharing

- Flickr
- Users (authors)
  - upload photos
  - tag photos
  - comment on photos
  - mark favorites
  - create friendship links
  - form an online community
- Can we predict the popularity of a newly uploaded photo?
- e.g., estimate the number of “favorites” in the next few months

Application II – Academic bibliography

- Database of scientific articles, e.g., CiteSeer
- Authors publish papers
- Existing papers accumulate reputation by citations
- Can we predict the popularity of a newly published paper?
- e.g., estimate the number of citations after a few years

The abstract graph model



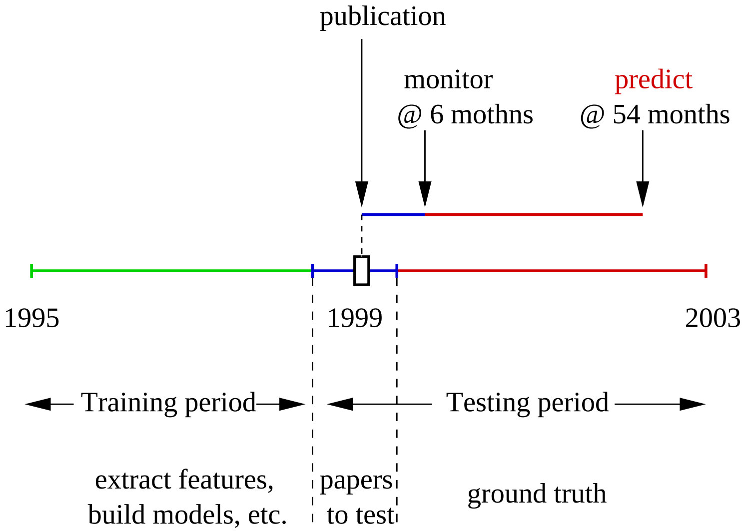
Other information:

- content of items
- a social network on authors

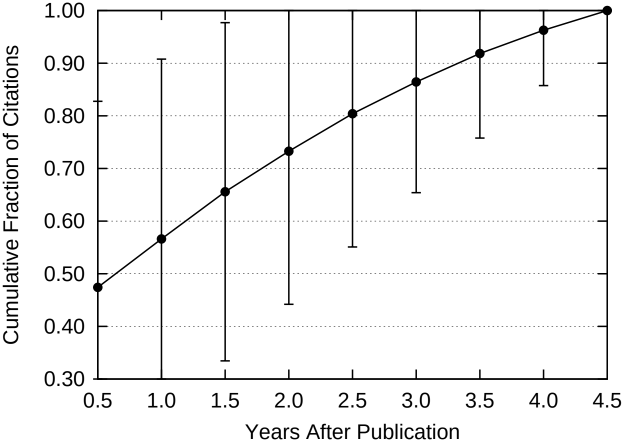
The dataset

- CiteSeer database of scientific articles
- `http://citeseer.ist.psu.edu/`
- 581 866 papers published from 1995 to 2003 (inclusive)
- Keep only papers for which at least one of the authors had three papers or more in the dataset
- Prune 11% of the dataset

The prediction task

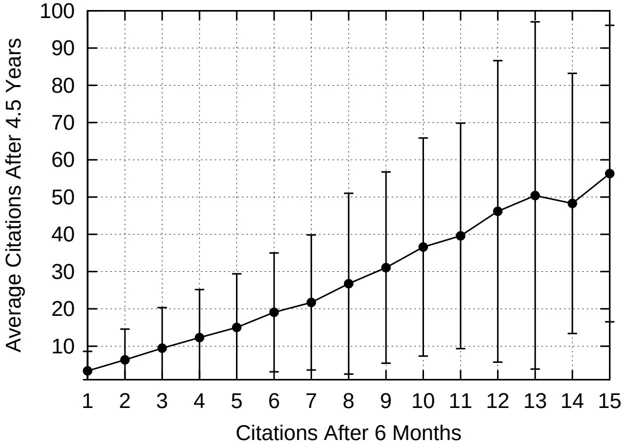


The challenges – Large variance



Cumulative fraction of citations over time

The challenges – Large variance



Citations at 6 months vs. average citations at 54 months

The baseline

- Citations at 6 months and citations at 54 months have correlation coefficient 0.57
- Can be a basis for a prediction, but not so accurate
- How to improve it?

What is missing

- Past information about the authors
- Exploiting the network structure:
- Good authors tend to write good papers
- Good authors tend to cite good papers
- Papers written and cited by good authors tend to be successful

Machine learning approach

- Extract a set of features and use it to build a better model

Author-based features

For each author compute:

- Total number of citations received
- Total number of papers (co)authored
- Average number of citations per paper
- Total number of co-authors
- Average number of co-authors per paper
- ...

For each paper compute:

- aggregate of the features of its authors
  (using `sum`, `avg`, `max`)

Link-based features

- EigenRumor algorithm [Fujimura and Tanimoto, 2005]
- Inspired by HITS [Kleinberg, 1999]

Eigenrumor algorithm

- P: *provision matrix* (authors $\times$ papers)

  $P_{ij} = 1$ if author i has provided paper j and 0 otherwise

  E: *evaluation matrix* (authors $\times$ papers)
- $E_{ij} = 1$ if author i has evaluated paper j and 0 otherwise
- $\mathbf{r}$: *reputation* scores of papers
- $\mathbf{a}$: *authority* scores of authors
- $\mathbf{h}$: *hub* scores of authors

Eigenrumor algorithm

- High-reputation papers are written by high-authority authors and cited by high-hub authors
- High-authority authors write high-reputation papers
- High-hub authors cite high-reputation papers
- In equations

$$\mathbf{r} = \alpha P^T \mathbf{a} + (1 - \alpha) E^T \mathbf{h}$$

$$\mathbf{a} = P\mathbf{r}$$

$$\mathbf{h} = E\mathbf{r}$$

Link-based features

For each author compute:

- Authority score
- Hub score

For each paper compute:

- Reputation score
- Aggregate of authority score and hub score of its authors (using `sum`, `avg`, `max`)

Prediction tasks

1. Regression: predict the number of citations of a paper
2. Classification: predict if a paper will be *successful* (defined as being in the top 10%)

Results

Effect of monitoring period

A posteriori citations	Predicting Citations r	Predicting Success F
6 months	0.57	0.15
1.0 year	0.76	0.54
1.5 years	0.87	0.63
2.0 years	0.92	0.71
2.5 years	0.95	0.76
3.0 years	0.97	0.86
3.5 years	0.99	0.91
4.0 years	0.99	0.95

Results

Effect of different type of features

A priori features	*A posteriori* features			
	First 6 months		First 12 months	
	r	*F*	*r*	*F*
None	0.57	0.15	0.76	0.54
Author-based	0.78	0.47	0.84	0.54
Hubs/Auth	0.69	0.39	0.80	0.54
Host	0.62	0.46	0.77	0.57
EigenRumor	0.74	**0.55**	0.83	**0.64**
ALL	**0.81**	**0.55**	**0.86**	0.62

Conclusions

- Predicting reputation as a link-analysis task
- Can we improve performance?
- Can we solve the problem in more “noisy” environments?

New and challenging graph datasets

- Social networks
- Yahoo! answers
- Users ask questions, provide answers, vote for best answers, mark "good" questions, report abuses, try to collect points, etc.
- Problems:
  - search for answers to questions already asked
  - build reputation mechanisms for users
  - predict quality of questions or answers
  - find "expert" users
  - suggest questions to users interested in answering

New and challenging graph datasets

- Query logs
- Users make queries
- Queries are related if they
  - return similar results
  - return results with similar content
  - return urls that user click
  - etc..
- Problems:
  - find similar queries
  - find generalizations and specializations of queries
  - query suggestion and personalization

Acknowledgments

The following people have contributed directly or indirectly to some of the content in this presentation

- Ricardo Baeza-Yates
- Carlos "Chato" Castillo
- . . .

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., and Baeza-Yates, R. (2006).

Link-based characterization and detection of Web Spam.

In *Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Seattle, USA.

Cohen, W. W. and Kou, Z. (2006).

Stacked graphical learning: approximating learning in markov random fields using very short inhomogeneous markov chains.

Technical report.

Eiron, N., Curley, K. S., and Tomlin, J. A. (2004).

Ranking the web frontier.

In *Proceedings of the 13th international conference on World Wide Web*, pages 309–318, New York, NY, USA. ACM Press.

Fujimura, K. and Tanimoto, N. (2005).

The EigenRumor Algorithm for Calculating Contributions in Cyberspace Communities.

Gibson, D., Kumar, R., and Tomkins, A. (2005).
Discovering large dense subgraphs in massive graphs.
In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 721–732. VLDB Endowment.

Gyöngyi, Z., Garcia-Molina, H., and Pedersen, J. (2004).
Combating Web spam with TrustRank.
In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Toronto, Canada. Morgan Kaufmann.

Kleinberg, J. M. (1999).
Authoritative sources in a hyperlinked environment.
Journal of the ACM, 46(5):604–632.

Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D. (2006).
Detecting spam web pages through content analysis.
In *Proceedings of the World Wide Web conference*, pages 83–92, Edinburgh, Scotland.