Chapter 4
Web Spam & Advertising

Sergej Sizov
Information Retrieval
Summer term 2008

7.1 Web Spam

..not just for email anymore

Users follow search results
- Money follows users... Spam follows money...
There is value in getting ranked high
- Funnel traffic from SEs to Amazon/eBay/...
Make a few bucks
- Funnel traffic from SEs to a Viagra seller
Make $6 per sale
- Funnel traffic from SEs to a porn site
Make $20-$40 per new member
- Affiliate programs

7.2 Spam: Motivation

Let’s do the math...
- Assume 500M searches/day on the web
- All search engines combined
- Assume 5% commercially viable

Much more if you include „adult-only” queries
- Assume $0.50 made per click (from 5c to $40)
- $12.5M/day or about $4.5 Billion/year

Keyword stuffing and cloaking
Crawlers declare that it is a SE spider
They dish us an “optimized” page
But easy to detect for SE: just detect keyword density

Spam: defeating IR

Spam: query flooding

easy to detect for SE: just detect the page is not about the query

Spam: defeating IR/NLP

Ideally, links should help: no one should link to these bad sites...
Getting links: grabbing expired domains

Getting links: link exchange

Getting links: Mailing Lists

Getting Links: Guestbooks

Spam prevention: CAPTCHA

Web Spam: Summary

Content spam:
- repeat words (boost tf)
- weave words/phrases into copied text
- manipulate anchor texts

Link spam:
- copy links from Web dir. and distort
- create honeypot page and sneak in links
- infiltrate Web directory
- purchase expired domains
- generate posts to Blogs, message boards, etc.
- build & run spam farm (collusion) + form alliances

Hide/cloak the manipulation:
- masquerade href anchors
- use tiny anchor images with background color
- generate different dynamic pages to browsers and crawlers

Link Spam: General Scenario

Typical structure:

Web transfers to p0 the „hijacked“ score mass („leakage“)

\[
\lambda = \sum_{q \in \text{IN}(p_0) \setminus \{p_1, \ldots, p_k\}} \frac{\text{PR}(q)}{\text{outdegree}(q)}
\]

Theorem:
p0 obtains the following PR authority:

\[
\text{PR}(p_0) = \frac{1}{1-(1-\varepsilon)\lambda + \varepsilon(1-\varepsilon)k+1}
\]

The above spam farm is optimal within some family of spam farms (e.g. letting hijacked links point to boosting pages).
Spam Countermeasures

Basic Ideas:
- compute negative propagation of blacklisted pages (BadRank)
- compute positive propagation of trusted pages (TrustRank)
- detect spam pages based on statistical anomalies
- inspect PR distribution in graph neighborhood (SpamRank)
- learn spam vs. ham based on page and page-context features
- spam mass estimation (fraction of PR that is undeserved)
- probabilistic models for link-based authority (overcome the discontinuity from 0 outlinks to 1 outlink)

BadRank and TrustRank

BadRank:
start with explicit set B of blacklisted pages
define random-jump vector \( r \) by setting \( r_i = 1/|B| \) if \( i \in B \) and 0 else
propagate BadRank mass to predecessors

\[
BR(p) = \beta r_p + (1-\beta) \sum_{q \in \text{in}(p)} BR(q) / \text{indegree}(q)
\]

TrustRank:
start with explicit set T of trusted pages with trust values \( t_i \)
define random-jump vector \( r \) by setting \( r_i = t_i \) if \( i \in B \) and 0 else
propagate TrustRank mass to successors

\[
TR(q) = r q + (1-r) \sum_{p \in \text{out}(q)} TR(p) / \text{outdegree}(p)
\]

Problems:
maintenance of explicit lists is difficult
difficult to understand (and guarantee) effects

7.2 Web Advertising

Banner ads (1995-2001)
- Initial form of web advertising
- Popular websites charged XS for every 1000 “impressions” of ad
  - Called “CPM” rate
  - Modeled similar to TV, magazine ads
- Untargeted to demographically tagged
- Low clickthrough rates
  - low ROI for advertisers

Performance-based advertising

Introduced by Overture around 2000
- Advertisers “bid” on search keywords
  - When someone searches for that keyword, the highest bidder’s ad is shown
  - Advertiser is charged only if the ad is clicked on
- Similar model adopted by Google with some changes around 2002
  - Called “Adwords”
Ads vs. Search Results

Web Advertising: Questions

Performance-based advertising works!

- Multi-billion-dollar industry

Interesting problems

- What ads to show for a search?
- If I’m an advertiser, which search terms should I bid on and how much to bid?

Web Advertising: Questions

Adwords problem

A stream of queries arrives at the search engine

- \( q_1, q_2, \ldots \)

Several advertisers bid on each query

When query \( q \) arrives, search engine must pick a subset of advertisers whose ads are shown

Goal: maximize search engine’s revenues

Clearly we need an online algorithm!

Simplest algorithm is greedy...

... the greedy algorithm is actually optimal!

Adwords problem

A stream of queries arrives at the search engine

- \( q_1, q_2, \ldots \)

Several advertisers bid on each query

When query \( q \) arrives, search engine must pick a subset of advertisers whose ads are shown

Goal: maximize search engine’s revenues

Clearly we need an online algorithm!

Simplest algorithm is greedy...

... the greedy algorithm is actually optimal!

Advertizing: Justification (1)

Each ad has a different likelihood of being clicked

- Advertiser 1 bids $2, click probability = 0.1
- Advertiser 2 bids $1, click probability = 0.5
- Clickthrough rate measured historically

Simple solution

- Instead of raw bids, use the “expected revenue per click”

Each advertiser has a limited budget

- Search engine guarantees that the advertiser will not be charged more than their daily budget

Advertizing: Simplified Model

Assume all bids are 0 or 1

Each advertiser has the same budget \( B \)

One advertiser per query

Let’s try the greedy algorithm

- Arbitrarily pick an eligible advertiser for each keyword

BAD SCENARIO FOR GREEDY

Two advertisers A and B

A bids on query \( x \), B bids on \( x \) and \( y \)

Both have budgets of $4

Query stream: \( xxxyyyy \)

- Worst case greedy choice: \( BBBB \ldots \)
- Optimal: \( AAAABBBB \)
- Competitive ratio = \( \frac{1}{2} \)

... formal analysis shows this is the worst case
**BALANCE algorithm [MSVV]**

For each query, pick the advertiser with the largest unspent budget

- Break ties arbitrarily

**Analyzing BALANCE**

Consider simple case: two advertisers, A₁ and A₂, each with budget B

(assume B → 1)

Assume optimal solution exhausts both advertisers' budgets

BALANCE must exhaust at least one advertiser's budget

- If not, we can allocate more queries
- Assume BALANCE exhausts A₁'s budget

**General Result**

In the general case, worst competitive ratio of BALANCE is

\[
1 - \frac{1}{e} = \text{approx. 0.63}
\]

Interestingly, no online algorithm has a better competitive ratio

Won't go through the details here, but let's see the worst case that gives this ratio

**Worst case for BALANCE**

N advertisers, each with budget B ≥ N ≥ 1

NB queries appear in N rounds of B queries each

Round 1 queries: bidders A₁, A₂, ..., Aₙ

Round 2 queries: bidders A₂, A₃, ..., Aₙ

Round i queries: bidders Aᵢ, ..., Aₙ

Optimum allocation: allocate round i queries to Aᵢ

- Optimum revenue NB

**BALANCE allocation**

After k rounds, sum of allocations to each of bins A₁,...,Aₙ is

\[
S_k = S_{k+1} = \ldots = S_n = \sum_{i=0}^{N-1} \frac{B}{i} = \frac{B}{N(N-1)}
\]

If we find the smallest k such that \(S_k ≥ B\), then after k rounds we cannot allocate any queries to any advertiser

---

**Analyzing BALANCE**

For each query, pick the advertiser with the largest unspent budget

- Break ties arbitrarily

Two advertisers A and B

A bids on query x, B bids on x and y

Both have budgets of $4

Query stream: xxxxyyyy

BALANCE choice: ABABBB...

- Optimal: AAAABBBB

Competitive ratio = \(\frac{3}{4}\)

**General Result**

In the general case, worst competitive ratio of BALANCE is

\[
1 - \frac{1}{e} = \text{approx. 0.63}
\]

Interestingly, no online algorithm has a better competitive ratio

Won't go through the details here, but let's see the worst case that gives this ratio
**BALANCE analysis**

\[
\begin{array}{cccccc}
B/1 & B/2 & B/3 & \ldots & B/(N-k+1) & B/(N-1) & B/N \\
S_1 & & & & & & \\
S_2 & & & & & & \\
S_k = B & & & & & & \\
1/1 & 1/2 & 1/3 & \ldots & 1/(N-k+1) & 1/(N-1) & 1/N \\
S_1 & & & & & & \\
S_2 & & & & & & \\
S_k = 1 & & & & & & \\
\end{array}
\]

Fact: \( H_n = \sum_{i=1}^{n} \frac{1}{i} = \text{approx. } \log(n) \text{ for large } n \)

- Result due to Euler

\[
\begin{array}{cccccc}
1/1 & 1/2 & 1/3 & \ldots & 1/(N-k+1) & 1/(N-1) & 1/N \\
\log(N) & & & & & & \\
\log(N)-1 & & & & & & \\
S_k = 1 \text{ implies } H_{N-k} = \log(N)-1 = \log(N/e) \\\nN-k = N/e & & & & & & \\
k = N/(1-1/e) & & & & & & \\
\end{array}
\]

So after the first \( N(1-1/e) \) rounds, we cannot allocate a query to any advertiser

Revenue = BN(1-1/e)

Competitive ratio = 1-1/e

**General version of problem**

Arbitrary bids, budgets

Consider query \( q \), advertiser \( i \)

- Bid = \( x_i \)
- Budget = \( b_i \)

BALANCE can be terrible

- Consider two advertisers \( A_1 \) and \( A_2 \)
  - \( A_1: x_1 = 1, b_1 = 110 \)
  - \( A_2: x_2 = 10, b_2 = 100 \)

**Generalized BALANCE**

Arbitrary bids; consider query \( q \), bidder \( i \)

- Bid = \( x_i \)
- Budget = \( b_i \)
- Amount spent so far = \( m_i \)
- Fraction of budget left over \( f_i = 1 - m_i/b_i \)
- Define \( \psi_i(q) = x_i(1-e^{-f_i}) \)

Allocate query \( q \) to bidder \( i \) with largest value of \( \psi_i(q) \)

Same competitive ratio (1-1/e)