Web 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

Web Spam: defeating IR

Keyword stuffing and cloaking
Crawlers declare that it is a SE spider
They dish us an „optimized“ page
Users see a completely different page

But easy to detect for SE:
just detect keyword density

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 those bad sites...
Getting links: grabbing expired domains

Getting links: link exchange

Getting links: Mailing Lists

Getting Links: Guestbooks

Web Spam: Summary

Link Spam: General Scenario

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

Typical structure:
- boosting pages (spam farm) p₁,..., pₖ
- hijacked "leakage" links
- page p₀ to be "promoted"

Web transfers to p₀ the "hijacked" score mass ("leakage")

Theorem:
p₀ obtains the following PR authority:

\[
\text{PR}(p₀) = \frac{1}{1 - (1 - \epsilon) \lambda + \frac{\epsilon (1 - \epsilon) \lambda + 1}{n}}
\]

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)
- inspect PR distribution in graph neighborhood (SpamRank)
- detect spam pages based on statistical anomalies
- 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

### BadRank and TrustRank

**BadRank:**

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

**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

**Problems:**

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

### Web Advertising

**Banner ads (1995-2001):**

- Initial form of web advertising
- Popular websites charged \( X \$ \) for every 1000 "impressions" of ad
  - Called "CPM" rate
  - Modeled similar to TV, magazine ads
  - Untargeted to demographically targeted
  - 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

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Performance-based advertising works!</td>
<td>Performance-based advertising works!</td>
</tr>
<tr>
<td>5</td>
<td>Multi-billion-dollar industry</td>
<td>Multi-billion-dollar industry</td>
</tr>
<tr>
<td>19</td>
<td>Interesting problems</td>
<td>Interesting problems</td>
</tr>
<tr>
<td>20</td>
<td>What ads to show for a search?</td>
<td>What ads to show for a search?</td>
</tr>
<tr>
<td>21</td>
<td>If I’m an advertiser, which search terms should I bid on and how much to bid?</td>
<td>If I’m an advertiser, which search terms should I bid on and how much to bid?</td>
</tr>
</tbody>
</table>

### Web Advertising: Questions

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>Ads vs. Search Results</td>
<td>Ads vs. Search Results</td>
</tr>
<tr>
<td>22</td>
<td>Web Advertising: Questions</td>
<td>Web Advertising: Questions</td>
</tr>
<tr>
<td>23</td>
<td>Adwords problem</td>
<td>Adwords problem</td>
</tr>
<tr>
<td>24</td>
<td>Advertising: Simplified Model</td>
<td>Advertising: Simplified Model</td>
</tr>
<tr>
<td>25</td>
<td>Advertising: Justification (1)</td>
<td>Advertising: Justification (1)</td>
</tr>
<tr>
<td>26</td>
<td>Bad scenario for greedy</td>
<td>Bad scenario for greedy</td>
</tr>
</tbody>
</table>

### 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!

### Advertising: 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

### 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: \( xyyyyy \)

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

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

[Mehta, Saberi, Vazirani, and Vazirani]

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 = $\frac{3}{4}$

---

**Analyzing BALANCE**

Consider simple case: two advertisers, $A_1$ and $A_2$, each with budget $B$ (assume $B \leq 1$)
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 $A_2$’s budget

---

**Analyzing BALANCE**

Consider simple case: two advertisers, $A_1$ and $A_2$, each with budget $B$ (assume $B \leq 1$)
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 $A_2$’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 $B \leq N \leq 1$
$NB$ queries appear in $N$ rounds of $B$ queries each
Round 1 queries: bidders $A_1$, $A_2$, ..., $A_N$
Round 2 queries: bidders $A_2$, $A_3$, ..., $A_N$
Round $i$ queries: bidders $A_i$, $A_{i+1}$, ..., $A_N$
Optimum allocation: allocate round $i$ queries to $A_i$
- Optimum revenue $NB$

---

**BALANCE allocation**

After $k$ rounds, sum of allocations to each of bins $A_1$, ..., $A_N$ is $S_k = S_{k+1} = ... = S_N = \sum_{i=1}^{N} B/(N-i+1)$
If we find the smallest $k$ such that $S_k$, $B$, then after $k$ rounds we cannot allocate any queries to any advertiser
**BALANCE analysis**

\[
\begin{align*}
\text{B/1} & \text{ B/2} & \text{B/3} & \ldots & \text{B/(N-k+1)} & \ldots & \text{B/(N-1)} & \text{B/N} \\
\hline
\text{S}_1 \\
\text{S}_2 = B \\
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \ldots & \frac{1}{(N-k+1)} & \ldots & \frac{1}{(N-1)} & \frac{1}{N} \\
\hline
\text{S}_1 \\
\text{S}_2 = 1
\end{align*}
\]

\[H_n = \sum_{i=1}^{n} \frac{1}{i} \approx \log(n) \text{ for large } n\]

Result due to Euler

\[
\begin{align*}
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \ldots & \frac{1}{(N-k+1)} & \ldots & \frac{1}{(N-1)} & \frac{1}{N} \\
\hline
\log(N) & \ldots & \log(N) & \ldots & \log(N) & \ldots & \log(N) & \frac{1}{N} \\
\hline
\text{S}_1 = \frac{1}{1} & \frac{1}{2} = \log(N)-1 & \text{S}_2 = 1 \\
N-k = N/e & k = N(1-1/e)
\end{align*}
\]

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