Web Searching

I: History and Basic Notions, Crawling
II: Link Analysis Techniques
III: Web Spam Page Identification

Web Spam

Oρισμός: διασυνδεδεμένες σελίδες δημιουργημένες για παραπλάνηση των μηχανών αναζήτησης

Παραδείγματα:
- a pornography site page that contains thousands of keywords which are made invisible (to humans) by adjusting accordingly the color scheme
  - a search engine will include this page in the results of a query that contains some of these keywords
- creation of a large number of bogus web pages, all pointing to a single target page (that page will have high in-degree)
  - a search engine will rank high this page

Kίνητρο

- Web spam pages use various techniques to achieve higher-than-deserved rankings in a search engine’s results.
- Human experts can identify spam, but it is too expensive to manually evaluate a large number of pages
- Ανάγκη για αυτόματες ή ημιαυτόματες τεχνικές διαχωρισμού των «καλών» σελίδων από τις «κακές»

Antimetwpsi:

Κλασσικός τρόπος αντιμετώπισης
- Search engine companies typically employ staff members who specialize in the detection of web spam, constantly scanning the web looking for offenders
- In case a spam page is identified, the search engine stops crawling and indexing it.

Very expensive and slow spam detection process

Mia hmaioutma prosegignh

1) Selection of a small set of seed pages to be evaluated by an expert
2) After the manual selection of the reputable seed pages, the link structure of the web is exploited to discover other pages that are likely to be good.

Zetimata:
- How we should implement the seed selection?
- How we can discover the good pages?

Approximate isolation of the good set

Based on Z. Gyongyi, H. Garcia-Molina, J. Pedersen, Computing Web Spam with Trust Rank, SIGMOD’04

Empirical observation: Good pages seldom point to bad ones.
Approximate isolation of the good set

Exceptions

the creators of good pages can sometimes by «tricked» and add links to bad pages.

Examples:
- Unmoderated message boards where spammers post messages that include links to their spam pages
- Honey pots
  - pages that contain some useful resource but have hidden links to their spam pages (the honey pot attracts people to point to it)

Assessing Trust: Oracle Function

- We formalize the notion of a human checking a page by a binary «oracle» function $O$, over all pages $p$ in $V$.

\[
O(p) = \begin{cases} 
1 & \text{if } p \text{ is good} \\
0 & \text{if } p \text{ is bad}
\end{cases}
\]

- Oracle invocations are expensive and time consuming. We do not want to call the oracle function for all pages. Our objective is to be selective, i.e. to ask a human expert to evaluate only some of the pages.

Συνάρτηση Εμπιστοσύνης (Trust Function)

- To evaluate pages without relying on $O$, we will estimate the likelihood that a given page $p$ is good.

- Trust function yields values between 0 (bad) and 1 (good)
- Ideally, for any $p$, $T(p)$ should give us the probability that $p$ is good
- Ideal Trust Property (ITP)
  - $T(p) = \Pr[O(p) = 1]$
  - difficult to achieve
  - even if $T$ is not very accurate we could exploit it to order pages by their likelihood of being good

- Desired Trust Property (relaxation of ITP)
  - $T(p) < T(q) \iff \Pr[O(p) = 1] < \Pr[O(q) = 1]$
  - $T(p) = T(q) \iff \Pr[O(p) = 1] = \Pr[O(q) = 1]$

- Threshold Trust Property (another relaxation of ITP)
  - $T(p) > \delta \iff O(q) = 1$

Υπολογισμός Εμπιστοσύνης: The ignorant trust function

The ignorant trust function $T_0$

- We can select at random a seed set $S$ of $L$ pages and call the oracle on its elements.
- Let $S^+$ be the good pages and $S^-$ the bad ones. Since the remaining pages are not checked we can mark them with 1/2.
- This is the ignorant trust function $T_0$

\[
T_0(p) = \begin{cases} 
O(p) & \text{if } p \in S \\
1/2 & \text{otherwise}
\end{cases}
\]

Διάδοση Εμπιστοσύνης (Trust Propagation)

- We can exploit the empirical observation «Good pages seldom point to bad ones», and assign score 1 to all pages that are reachable from a page in $S^+$ in $M$ or fewer steps.
- Trust Function $T_M$:

\[
T_M(p) = \begin{cases} 
O(p) & \text{if } p \in S \\
1 & \text{if } p \notin S \text{ and } \exists q \in S^+ : q \rightarrow_M^* p \\
1/2 & \text{otherwise}
\end{cases}
\]

- $q$-$M$-$p$: there is a path of maximum length $M$ from $q$ to $p$
- The bigger $M$ the further we are from good pages, the less certain we are that a page is good
The PageRank score $R(p)$ of a page is defined as

$$R(p) = \frac{1}{N} \left( \frac{1-a}{n} \right) + \sum_{q \in \text{in}(p)} \frac{T(p,q)}{|\text{out}(q)|} R(q)$$

**Trust dampening**

- assign a score $\beta (<1)$ to pages reachable at 1 step
- assign the score $\beta^2$ to pages reachable at 2 step, and so on
- pages with multiple inlinks: maximum score or average score

**Trust splitting**

- motivation: the care with which people add links to their pages in often inversely proportional to the number of links on the page
- if page $p$ has a trust score $T(p)$ and it points to $|\text{out}(p)|$ pages, each of them will receive a score fraction $T(p)/|\text{out}(p)|$ from $p$
- the actual score of a page will be the sum of the score fractions received through its inlinks

*We could combine trust dampening and splitting*

**Oracle-provided scores replace the uniform distribution**

**Algorithm**

- the oracle-provided scores replace the uniform distribution

**Splitting**

$$\frac{R(p)}{N} = \frac{1-a}{n} + \frac{1}{N} \sum_{q \in \text{in}(p)} \frac{T(p,q)}{|\text{out}(q)|} R(q)$$

**Adjacency matrix $M$**

$$M = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}$$

**Transition matrix $T$**

$$T = \begin{bmatrix}
0 & 1/2 & 0 & 1/2 \\
1/2 & 0 & 0 & 1/2 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix}$$

**Function PageRank**

```
function PageRank
    input: T: transition matrix, N: number of pages, a: decay factor for biased PageRank, M: number of biased PageRank iterations
    output: T*: PageRank scores

    (3) d = 1/N * n
        // initial score for all pages is 1/N

    (5) T* = d
        for i = 1 to M do
            // evaluates PageRank scores
            T* = a * T* + (1 - a) * d
        return T*
```

**Empirical evidence**

- The PageRank scores converge to the long-term PageRank scores

**Empirical evidence**

- We can use the PageRank scores to estimate the importance of web pages

**Empirical evidence**

- The PageRank scores can be used to estimate the importance of web pages

**Empirical evidence**

- The PageRank scores can be used to estimate the importance of web pages

**Empirical evidence**

- The PageRank scores can be used to estimate the importance of web pages
Selecting Seeds

function TrustRank
input T: transition matrix, N: number of pages, L: limit of oracle invocations, a_b: decay factor for biased PageRank, M_b: number of biased PageRank iterations
output t*: TrustRank scores

(1) s = SelectSeed ()  // seed-desirability: returns a vector.
(2) σ = Rank(1, ..., N, s)  // E.g. a(p) is the desirability for page p
(3) d = σ_p  // initial score for all pages is 0
for i = 1 to L do // invokes oracle function on the most desirable pages
  for i = 1 to M_b do   // evaluates TrustRank scores using a biased PageRank
    d = σ_p  // seed-desirability: returns a vector.
    σ = Rank(1, ..., N, s)  // orders in decreasing order of s-value all pages
    if O(σ_p) = 1 then
      d = 0
    end
    for i = 1 to L do // invokes oracle function on the most desirable pages
      d = σ_p  // seed-desirability: returns a vector.
      σ = Rank(1, ..., N, s)  // orders in decreasing order of s-value all pages
    end
    if O(σ_p) = 1 then
      d = 0
    end
  end
end
return t*

Experimental Evaluation

• Experiments on the complete set of pages crawled and indexed by AltaVista (Aug. 2003)
• Reduce computational cost: work at the level of web sites (instead of web pages)
  • grouping of the (billions of) pages into 31 millions sites
  • websiteA points to websiteB if one or more pages from websiteA point to one or more pages of websiteB
  • So at most 1 link may start from website A and point to website B
• Observations
  • 1/3 of the websites are unreferenced
  • So TrustRank cannot differentiate between them because they all have |in(p)|=0
  • However they are low scored anyway (e.g. by PageRank) so they do not appear high in answers
• However they are low scored anyway (e.g. by PageRank) so they do not appear high in answers

Πιθανές Στρατηγικές

a) Random selection

β) High PageRank

Επέλεξαν τις σελίδες με υψηλό PageRank αφού διότι αυτές οι σελίδες συχνά εμφανίζονται στην κορυφή απαντήσεων.

γ) Inverse PageRank

Although 5 has the highest PageRank it is not a good seed.
Experimental Evaluation: Seed Selection

Seed Set Selection
- Inverse PageRank applied on the graph of websites worked better than High PageRank (for the seed selection process)
- Parameters: \(a:0.85\), iterations: 20
- With 20 iterations the relative ordering stabilized
- Manual inspection of the top 1250 sites (\(|S|=1250\))
- From these only 178 were used as good seeds, i.e. \(|S^+|=178\) sites

\[
\begin{align*}
(1) \quad & s = \text{SelectSeed()} \\
(2) \quad & \sigma = \text{Rank}([1, \ldots, N], s) \\
(3) \quad & d = a_0 \\
(4) \quad & \text{for } i=1 \text{ to } L \text{ do} \\
& \quad \text{if } O(\sigma(i)) = 1 \text{ then } d(\sigma(i)) = 1
\end{align*}
\]

Experimental Evaluation: Evaluation Sample

- To test the effectiveness of TrustRank we need a Reference Collection (e.g. something like TREC)
- A sample set \(X\) of 1000 sites was selected and evaluated manually, i.e. the oracle function was invoked (i.e. a person inspected them and decided whether they are spam or not)
- The Sample set \(X\) was not selected at random.
  - Recall that we are mainly interested in spam pages that appear high in answers
  - The following sample selection method was followed:

- Generate list of sites in decreasing order of their PageRank scores
- Segment them into 20 buckets so that the sum of the scores in each bucket equals 5% of the total PageRank score
  - |bucket1|=86, |bucket2|= 665, \ldots, |bucket20|= 5 millions pages
- select 50 sites at random from each bucket (20 * 50 = 1000)

The results of the manual evaluation (oracle invocation) of the pages in the sample set of 1000 sites

This collection (i.e. \(X\)) was used for evaluating TrustRank versus PageRank

Evaluation Results: Comparing PageRank with TrustRank

Good sites
- Reputable=white, advertisement=gray, webOrganization=dark gray

Bad sites
- TrustRank is a reasonable spam detection tool
Mέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης (Evaluation Metrics for the Trust Function)

• Assume a sample set $X$ of web pages for which we can invoke both $T$ and $O$

![Diagram showing a sample set $X$ of web pages](image)

**Precision and Recall**

• We can define precision and recall based on the threshold trust property:

$$\text{prec}(T,O) = \frac{|\{p \in X \mid T(p) > \delta \text{ and } O(p) = 1\}|}{|\{q \in X \mid T(q) > \delta\}|}$$

$$\text{rec}(T,O) = \frac{|\{p \in X \mid T(p) > \delta \text{ and } O(p) = 1\}|}{|\{q \in X \mid O(q) = 1\}|}$$

**Precision & Recall: Πειραματική Αξιολόγηση**

![Graph showing precision vs. recall](image)

5: such that to separate buckets

**Pairwise Orderedness**

• We can generate from $X$ a set $P$ of pairs and we can compute the fraction of the pairs for which $T$ did not make a mistake.

$$\text{Pairord}(T,O,P) = \frac{|\{p \in X \times X \mid T(p) > T(q) \text{ and } O(p) < O(q)\}|}{|P|}$$

• Pairord$(T,O,P)=1$ if $T$ does not make any mistake
• Pairord$(T,O,P)=0$ if $T$ makes always mistakes

**Pairwise Orderedness: Πειραματική Αξιολόγηση**

![Graph showing pairwise orderedness](image)

Figure 12: Pairwise orderedness.

Συμπέρασμα Πειραματικής Αξιολόγησης

TrustRank can effectively filter out spam from a significant fraction of the Web, based on a good seed set of less than 200 sites
References

- Z. Gyongyi, H. Garcia-Molina, J. Pedersen, Computing Web Spam with Trust Rank, SIGMOD'04