Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
Άνοιξη 2006

ΗΥ463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

Web Searching

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

Γιάννης Τζίτζικας
Διάλεξη : 10
Ημερομηνία : 29 / 3 / 2006

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

Κίνητρο

- 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»

Web Spam

Ορισμός: **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

Αντιμετώπιση

Κλασσικός τρόπος αντιμετώπισης
- Seach 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

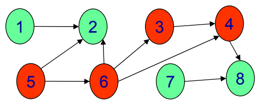
Μια ημιαυτόματη προσέγγιση

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*.

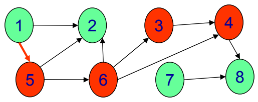
Ζητήματα:
- How we should implement the seed selection?
- How we can discover the good pages ?

Approximate isolation of the good set



- 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.

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

- 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 likehood 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 likehood of being good

Συνάρτηση Εμπιστοσύνης
(Trust Function)

- Desired Trust Property (relaxation of ITP)
  - T(p) < T(q) ⇔ Pr[O(p)=1] < Pr[O(q)=1]
  - T(p) = T(q) ⇔ Pr[O(p)=1] = Pr[O(q)=1]
- Threshold Trust Property (another relaxation of ITP)
  - T(p) > δ ⇔ O(q)=1

Υπολογισμός Εμπιστοσύνης:
The ignorant trust function

The **ignorant trust function T0**

- 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

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

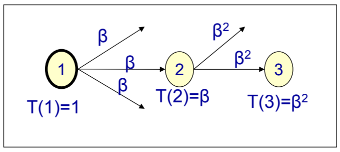
Διάδοση Εμπιστοσύνης
(Trust Propagation)

- 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:

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

- 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

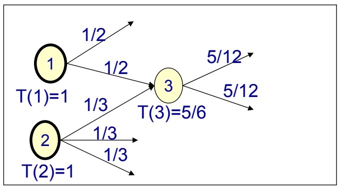
Εξασθένηση Εμπιστοσύνης (Trust Attenuation)
Trust dampening



Trust dampening

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

Εξασθένηση Εμπιστοσύνης (Trust Attenuation)
Trust splitting



Trust splitting

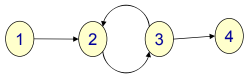
- motivation: the care with which people add links to their pages in often inversely proportional to the number of links on the page
- if page p has a trust score T(p) and it points to |out(p)| pages, each of them will receive a score fraction T(p)/ |out(p)| from p
- the actual score of a page will be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

- We could combine trust dampening and splitting

Ο Αλγόριθμος TrustRank

- In TrustRank we will combine **trust dampening** and **splitting**:
  - in each iteration, the trust score of a node is split among its neighbors and dampened by a factor of a_b
- We will compute TrustRank scores using a biased **PageRank** algorithm
  - the **oracle-provided scores replace the uniform distribution**

Επανάληψη: PageRank



Adjacency matrix M

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

Transition matrix T

$$T(p,q) = \begin{cases} 0 & if\,(q,p) \notin M \\ 1/|out(q)| & if\,(q,p) \in M \end{cases} \qquad T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{pmatrix}$$

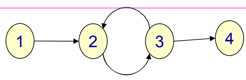
- The PageRank score R(p) of a page is defined as

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

- The equivalent matrix equation: $R = a \cdot T \cdot R + (1-a)\frac{1}{N} 1_N$

Επανάληψη: PageRank

$$R = a \cdot T \cdot R + (1-a)\frac{1}{N} 1_N$$



$$\begin{bmatrix} r1 \\ r2 \\ r3 \\ r4 \end{bmatrix} = a \cdot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} r1 \\ r2 \\ r3 \\ r4 \end{bmatrix} + (1-a)\frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} r1 \\ r2 \\ r3 \\ r4 \end{bmatrix} = a \cdot \begin{bmatrix} 0 \\ r1 + r3/2 \\ r2 \\ r3/2 \end{bmatrix} + (1-a)\frac{1}{4}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} r1 \\ r2 \\ r3 \\ r4 \end{bmatrix} = \begin{bmatrix} (1-a)/4 \\ a(r1 + r3/2) + (1-a)/4 \\ ar2 + (1-a)/4 \\ ar3/2 + (1-a)/4 \end{bmatrix}$$

Επανάληψη: Ο Αλγόριθμος PageRank

function **PageRank**
Input T: transition matrix, N: number of pages,
a_b: decay factor for biased PageRank, M_b: number of biased PageRank iterations
output **t*** : PageRank scores

(3) **d** =1/N * $\mathbf{1}_N$ // initial score for all pages is 1/N

(5) **t* = d**
for i=1 to M_b do // evaluates PageRank scores
t* = a_b T t* + (1 - a_b) **d**
return **t***

Ο Αλγόριθμος TrustRank

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.
// E.g. **s**(p) is the desirability for page p
(2) **σ** = Rank({1,…,N}, s) // orders in decreasing order of s-value all pages
(3) **d** = $\mathbf{0}_N$ // initial score for all pages is 0
for i=1 to L do // invokes oracle function on the most desirable pages
if O(**σ**(i)) =1 then **d**(σ(i))=1
(4) **d** := **d** / |**d**| // normalize static distribution score (to sum up to 1)
(5) **t*** = **d**
for i=1 to M_b do // evaluates TrustRank scores using a biased PageRank
t* = a_b T t* + (1 - a_b) **d** **// note that d replaces the uniform distribution**
return **t***

Ο Αλγόριθμος TrustRank

Remarks:
- Step 5 implements a particular version of **trust dampening** and **splitting**: in each iteration, the trust score of a node is split among its neighbors and dampened by a factor of a_b
- The good seed pages have no longer a score of 1, however they still have the highest scores

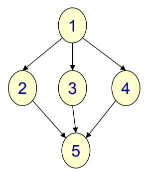
(4) **d** := **d** / |**d**| // normalize static distribution score (to sum up to 1)
(5) **t*** = **d**
for i=1 to M_b do // evaluates TrustRank scores using a biased PageRank
t* = a_b T t* + (1 - a_b) **d** **// note that d replaces the uniform distribution**
return **t***

Selecting Seeds

(1) **s** = **SelectSeed** () // seed-desirability: returns a vector.
// E.g. **s**(p) is the desirability for page p
(2) **σ** = Rank({1,…,N}, s) // orders in decreasing order of s-value all pages

Πιθανές Στρατηγικές
- α) Random selection
- β) **High PageRank**

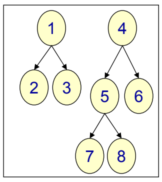
  Επιλέγουμε τις σελίδες με **υψηλό PageRank σκορ** διότι αυτές οι σελίδες συχνά εμφανίζονται στην κορυφή των απαντήσεων
- γ) **Inverse PageRank**

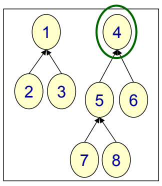


Although 5 has the highest PageRank it is not a good seed

Selecting Seeds: (γ) **Inverse PageRank**

- επειδή η εμπιστοσύνη διαχέεται από τις καλές σελίδες, είναι λογικό να επιλέξουμε εκείνες τις σελίδες από τις οποίες μπορούμε να φτάσουμε σε πολλές άλλες
  - άρα μια ιδέα είναι να επιλέξουμε τις σελίδες με πολλά outlinks
  - Επιλογή των p1, p4, p5
- γενίκευση: επιλέγουμε τις σελίδες που δείχνουν σε πολλές σελίδες οι οποίες με τη σειρά τους δείχνουν σε πολλές σελίδες, **κ.ο.κ**
  - Επιλογή της p4
- Τρόπος: Αφού η σπουδαιότητα μιας σελίδας εξαρτάται από τα outlinks της (και όχι από τα inlinks της), μπορούμε να χρησιμοποιήσουμε την PageRank **αντιστρέφοντας την φορά των ακμών**





Experimental Evaluation

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

Experimental Evaluation: Seed Selection

(1) **s = SelectSeed** ()
(2) **σ** = Rank({1,…,N}, s)
(3) d = 0_N
(4) for i=1 to L do
  if O(**σ**(i)) =1 then **d**(σ(i))=1

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**

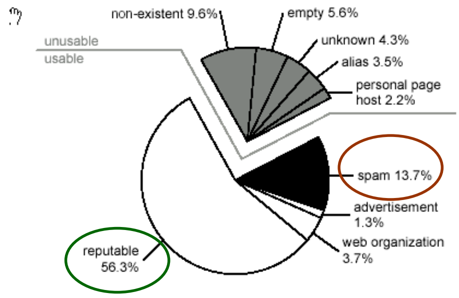
Experimental Evaluation: Evaluation Sample

- 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 <u>was not selected at random</u>.
  - Recall that we are mainly interetested 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 backet equals 5% of the total PageRank score
  - |backet1|=86, |backet2|= 665, … , |backet20|= 5 millions pages
- select 50 sites at random from each bucket (20 * 50 =1000)

Experimental Evaluation: Evaluation Sample

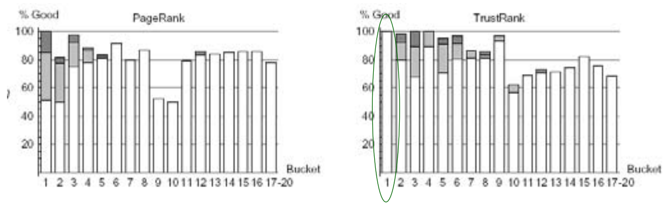
The results of the manual evaluation (oracle invocation) of the pages in the sample set of 1000 sites



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

Evaluation Results: Comparing PageRank with TrustRank

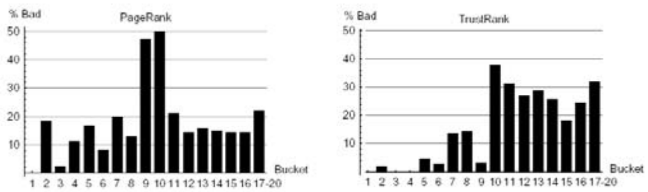
Good sites



Reputable=white, advertisement=gray, webOrganization=dark gray

Evaluation Results: Comparing PageRank with TrustRank

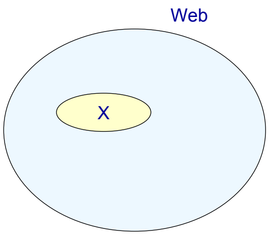
Bad sites



TrustRank is a reasonable spam detection tool

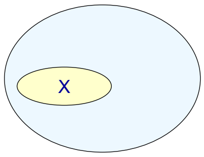
Μέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης (Evaluation Metrics for the Trust Function)

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



Μέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης: **Precision and Recall**

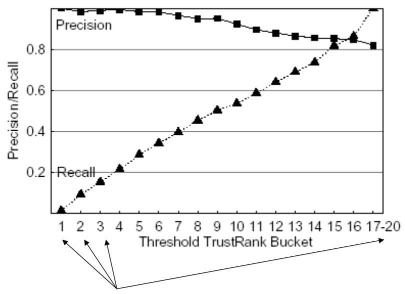
- We can define precision and recall based on the **threshold** trust property:



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

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

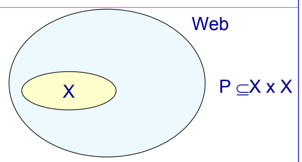
Μέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης: Precision & Recall: Πειραματική Αξιολόγηση



δ: such that to separate backets

Μέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης: **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.



- The following metric can signal a violation of the ordered trust property

$$I(T,O,p,q) = \begin{cases} 1 & \text{if } T(p) \geq T(q) \text{ and } O(p) < O(q) \\ 1 & \text{if } T(p) \leq T(q) \text{ and } O(p) > O(q) \\ 0 & \text{otherwise} \end{cases}$$

$$pairord(T,O,P) = \frac{|P| - \sum_{(p,q) \in P} I(T,O,p,q)}{|P|}$$

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

Μέτρα Αξιολόγησης της Συνάρτησης Εμπιστοσύνης: Pairwise Orderedness: Πειραματική Αξιολόγηση

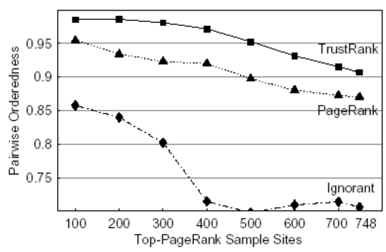


Figure 12: Pairwise orderedness.

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, Compating Web Spam with Trust Rank, SIGMOD'04