Ανάκτηση Κειμένου και Εξόρυξη από τον Παγκόσμιο Ιστό

Ανάκτηση Κειμένου (εισαγωγικά θέματα)

- Βάσεις Κειμένων (document databases)
  - Μεγάλη συλλογή από κείμενα από διάφορες πηγές: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
  - Τα δεδομένα δεν ακολουθούν κάποιο αυστηρό μοντέλο – ημι-δομημένα *semi-structured*
- Information retrieval – Ανάκτηση Πληροφορίας
  - Η πληροφορία οργανώνεται σε (ένα μεγάλο αριθμό) από κείμενα - documents
  - Information retrieval problem: εντοπισμός των σχετικών κειμένων (documents) με βάση την είσοδο του χρήστη όπως λέξεις κλειδιά ή παραδείγματα κειμένου

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database systems
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

- Βασικές έννοιες
  - Ένα κείμενο (αρχείο) document μπορεί να περιγράφει από ένα σύνολο αντιπροσωπευτικών λέξεων-κλειδιά (keywords) που ονομάζονται όροι δεικτοδότησης - index terms.
  - Διαφορετικοί όροι με διαφορετικό βαθμό σχετικότητας μπορούν να χρησιμοποιηθούν για την περιγραφή κειμένων με διαφορετικό περιεχόμενο
  - Αυτό επιτυγχάνεται με την ανάθεση αριθμητικών βαρών (numerical weights) σε κάθε όροι δεικτοδότησης του κειμένου (π.χ.: συχνότητα, tf-idf)
- Αναλογία με ΣΔΒΔ:
  - Όροι Δεικτοδότησης → Γνωρίσματα
  - Βάρη → Τιμές γνωρισμάτων

Αναζήτηση με μια λέξη κλειδί (keyword queries)

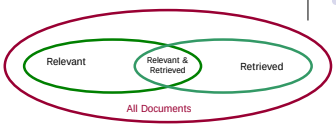
Αίτημα Boole

$(t_{11} \vee t_{12} \vee \ldots \vee t_{i1}) \wedge (t_{21} \vee t_{22} \vee \ldots \vee t_{2i2}) \wedge \ldots. (t_{j1} \vee t_{j2} \vee \ldots \vee t_{jij})$

Όπου τα t_{ij} είναι όροι

Αίτημα Διαβάθμισης (Ranking)

Βασικές Μετρικές



Precision – Ακρίβεια : το ποσοστό των ανακτημένων documents που είναι σχετικά με την ερώτηση (δηλαδή, το ποσοστό των «σωστών» απαντήσεων»)

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

Recall – Ανάκληση: το ποσοστό των σχετικών documents που ανακτούνται

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Συνήθως, κατασκευάζονται ευρετήρια που περιέχουν ζεύγη <όρος, id-αρχείου> με πιθανών επιπλέον πεδία όπως η συχνότητα εμφάνισης του όρου στο αρχείο

Παρόμοια, δουλεύουν και οι μηχανές αναζήτησης

Μια ταξινομημένη λίστα (ανεστραμμένη λίστα) (inverted file, inverted list) για κάθε όρο

Παράδειγμα

Rid	Λέξεις-Κλειδιά
1	agent James Bond
2	agent mobile computer
3	James Madison movie
4	James Bond movie

Agent <1,2>
Bond <1,4>
Computer <2>
James <1,3,4>
Madison <3>
Mobile <2>
Movie <3,4>

Παράδειγμα ερωτήσεων

Ευρετήριο Λεξιλογίου:

Για τον ταχύτερο εντοπισμό της λίστας για κάθε όρο: Το σύνολο των όρων μπορεί να οργανωθεί με τη χρήση μιας δομής ευρετηρίου (π.χ. B+-δέντρο)

Στα φύλλα, δείκτες προς την αντίστοιχη ανεστραμμένη λίστα

Παράδειγμα

Ένας όρος, σύζευξη, διάζευξη

Υπογραφή εγγράφου (File Signature) Μια εγγραφή ευρετηρίου για κάθε έγγραφο στη βάση δεδομένων

Κάθε εγγραφή σταθερό μέγεθος b bits, εύρος της υπογραφής

Κατασκευή της υπογραφής ενός αρχείου: Σε κάθε όρο που υπάρχει στο αρχείο, εφαρμόζεται μια *συνάρτηση κατακερματισμού*, που επιστρέφει ένα αριθμό από το 1 ως το b και το αντίστοιχο bit της υπογραφής του αρχείου γίνεται 1

Για μια ερώτηση, φτιάχνουμε την υπογραφή της και σαρώνουμε τις υπογραφές των αρχείων για να βρούμε κάποια που ταιριάζει

False positives

Αποφυγή σάρωσης όλου του αρχείου υπογραφών:

Αρχείο υπογραφών με κατακόρυφο διαμερισμό σε μονοψήφιες στήλες:

Διαμερίζουμε ένα αρχείο υπογραφών σε ένα σύνολο κατακόρυφων δυαδικών στηλών

Για κ άσσους ανάκτηση κ-στηλών

Boolean Model – Δυαδικό Μοντέλο

- Το μοντέλο που είδαμε μέχρι στιγμής θεωρεί ότι οι όροι δεικτοδότησης είναι είτευπάρχουν είτε δεν υπάρχουν στο αρχείο (κείμενο)
- Δηλαδή, τα βάρη είναι όλα δυαδικά (0 ή 1)
- Οι ερωτήσεις είναι όροι συνδεδεμένοι με : not, and, και or
  - πχ.: car *and* repair, plane *or* airplane
- Το δυαδικό μοντέλο προβλέπει ότι ένα αρχείο είναι είτε σχετικό είτε μη σχετικό με βάση ένα ταίριασμα της ερώτησης με το αρχείο

Μοντέλο με βάρη

Συχνότητα όρου- term frequency : πόσες φορές εμφανίζεται ένας όρος σε ένα έγγραφο
Κανονικοποιημένο ώστε να αποφύγουμε να δώσουμε μεγαλύτερο βάρος σε μεγάλα έγγραφα
Σημασία του όρου t_i σε ένα έγγραφο

$$tf_i = \frac{n_i}{\sum_k n_k}$$

Ανεστραμμένη συχνότητα εγγράφου *inverse document frequency* μετρά πόσο είναι γενικά σημαντικός ένας όρος

$$idf_i = \log \frac{|D|}{|\{d : d \in t_i\}|}$$

|D| αριθμός εγγράφων

Έγγραφα στα οποία ανήκει ο όρος t_i

$$tfidf = tf * idf$$

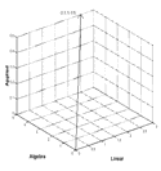
Μεγάλη τιμή όταν μεγάλη συχνότητα εμφάνισης (σε ένα συγκεκριμένο έγγραφο) και μικρή συχνότητα εμφάνισης του όρου ε όλη τη συλλογή
Βάρος χρήσιμο για να αποφύγουμε κοινούς όρους

Άλλα Μοντέλα

- Ένας πίνακας με τη συχνότητα των όρων (term frequency table)
  - Κάθε εγγραφή *frequent_table(i, j)* = # of occurrences of the word t_i in document d_j
  - Συνήθως, το *ποσοστό* (*ratio*) αντί του πραγματικού αριθμού εμφανίσεων
- Similarity metrics – μετρική ομοιότητας: μεταξύ ενός κειμένου και μιας ερώτησης (συνόλου από λέξεις-κλειδιά - όρους)
  - Relative term occurrences $sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$
  - Cosine distance:

Vector Model – Μοντέλο Διανυσμάτων

- Τα αρχεία και οι ερωτήσεις αναπαρίστανται ως m-διάστατα διανύσματα, όπου m είναι ο συνολικός αριθμός όρων στη συλλογή
- Ο βαθμός ομοιότητας ενός αρχείου d και μιας ερώτησης q υπολογίζεται ως η συνέλιξη τους, χρησιμοποιώντας μετρικές όπως η Ευκλείδεια απόσταση ή το συνημίτονο της γωνίας των δύο διανυσμάτων



Latent Semantic Indexing

- Βασική Ιδέα
  - Similar documents have similar word frequencies
  - Difficulty: the size of the term frequency matrix is very large
  - Use a singular value decomposition (SVD) techniques to reduce the size of frequency table
  - Retain the *K* most significant rows of the frequency table

Άλλα Θέματα

- Ρίζα λέξεων - Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug, drugs, drugged*
- Συνώνυμα - Synonymy: *A* keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
- Πολυσημία - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

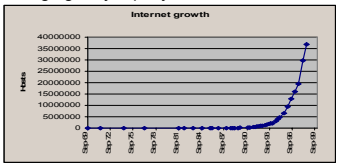
Άλλα Θέματα

- Stop list
  Set of words that are deemed "irrelevant", even though they may appear frequently, πχ, *a, the, of, for, to, with*, etc.
- Οντολογίες - Wordnet

Εξόρυξη στον Παγκόσμιο Ιστό

- The WWW is huge, widely distributed, global information service center for
  - ΥΠΗΡΕΣΙΕΣ - Information services: news, advertisements, consumer information, financial management, education, government, e-commerce, etc.
  - ΣΥΝΔΕΣΜΟΙ - Hyper-link information
  - ΠΛΗΡΟΦΟΡΙΑ ΧΡΗΣΗΣ - Access and usage information
- WWW provides rich sources for data mining
- Challenges
  - Too huge for effective data warehousing and data mining
  - Too complex and heterogeneous: no standards and structure

- Growing and changing very rapidly

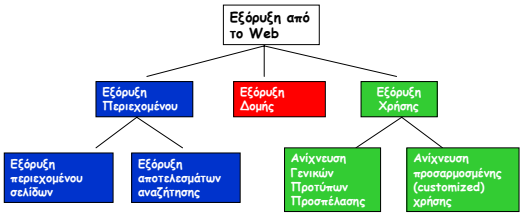


- Broad diversity of user communities
- Only a small portion of the information on the Web is truly relevant or useful
  - 99% of the Web information is useless to 99% of Web users
  - How can we find high-quality Web pages on a specified topic?

Εξόρυξη από το Web

- Ψάχνουμε για
  - Web access patterns
  - Web structures
  - Regularity and dynamics of Web contents
- Problems
  - The "abundance" problem: ο αριθμός των σελίδων που συσχετίζονται με έναν όρο μπορεί να είναι πολύ μεγάλος
  - Limited coverage of the Web: hidden Web sources, majority of data in DBMS
  - Limited query interface based on keyword-oriented search
  - Limited customization to individual users

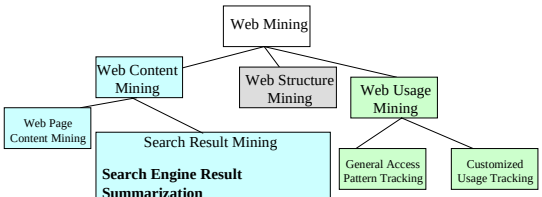
Κατηγορίες Εξόρυξης από το Web



Κατηγορίες Εξόρυξης από το Web

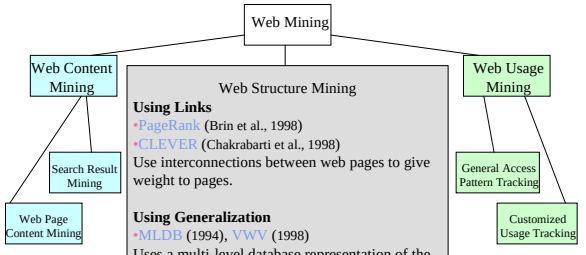
Web Page Content Mining
Web Page Summarization
WebLog (Lakshmanan et.al. 1996), WebOQL(Mendelzon et.al. 1998):
Web Structuring query languages;
Can identify information within given web pages
Ahoy! (Etzioni et.al. 1997):Uses heuristics to distinguish personal home pages from other web pages
ShopBot (Etzioni et.al. 1997): Looks for product prices within web pages

Mining the World-Wide Web



Search Engine Result Summarization
•Clustering Search Result (*Leouski and Croft, 1996, Zamir and Etzioni, 1997*):
Categorizes documents using phrases in titles and snippets

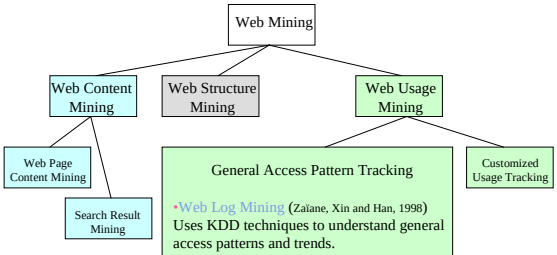
Mining the World-Wide Web



Web Structure Mining
Using Links
•PageRank (Brin et al., 1998)
•CLEVER (Chakrabarti et al., 1998)
Use interconnections between web pages to give weight to pages.

Using Generalization
•MLDB (1994), VWV (1998)
Uses a multi-level database representation of the Web. Counters (popularity) and link lists are used for capturing structure.

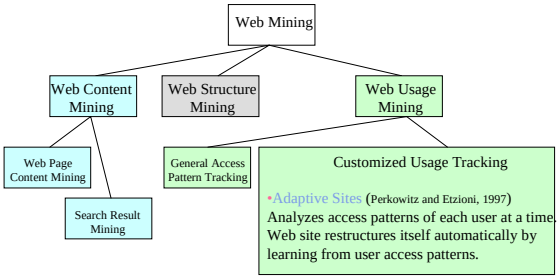
Mining the World-Wide Web



General Access Pattern Tracking

•Web Log Mining (Zaïane, Xin and Han, 1998)
Uses KDD techniques to understand general access patterns and trends.
Can shed light on better structure and grouping of resource providers.

Mining the World-Wide Web



Web Usage Mining

- Mining Web log records to discover user access patterns of Web pages
- Applications
  - Target potential customers for electronic commerce
  - Enhance the quality and delivery of Internet information services to the end user
  - Improve Web server system performance
  - Identify potential prime advertisement locations
- Web logs provide rich information about Web dynamics
  - Typical Web log entry includes the URL requested, the IP address from which the request originated, and a timestamp

Techniques for Web usage mining

- Construct multidimensional view on the Weblog database
  - Perform multidimensional OLAP analysis to find the top *N* users, top *N* accessed Web pages, most frequently accessed time periods, etc.
- Perform data mining on Weblog records
  - Find association patterns, sequential patterns, and trends of Web accessing
  - May need additional information,e.g., user browsing sequences of the Web pages in the Web server buffer
- Conduct studies to
  - Analyze system performance, improve system design by Web caching, Web page prefetching, and Web page swapping

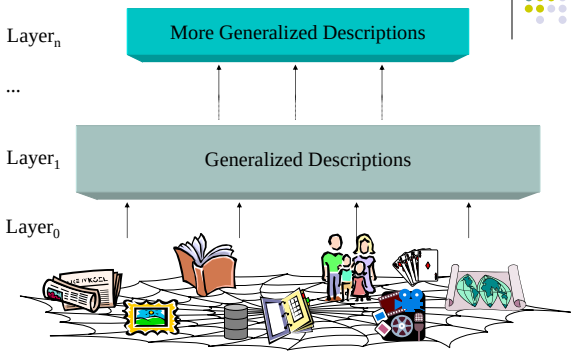
Automatic Classification of Web Documents

- Assign a class label to each document from a set of predefined topic categories
- Based on a set of examples of preclassified documents
- Example
  - Use Yahoo!'s taxonomy and its associated documents as training and test sets
  - Derive a Web document classification scheme
  - Use the scheme classify new Web documents by assigning categories from the same taxonomy
- Keyword-based document classification methods
- Statistical models

Multilayered Web Information Base

- $Layer_0$: the Web itself
- $Layer_1$: the Web page descriptor layer
  - Contains descriptive information for pages on the Web
  - An abstraction of $Layer_0$: substantially smaller but still rich enough to preserve most of the interesting, general information
  - Organized into dozens of semistructured classes
    - *document, person, organization, ads, directory, sales, software, game, stocks, library_catalog, geographic_data, scientific_data*, etc.
- $Layer_2$ and up: various Web directory services constructed on top of $Layer_1$
  - provide multidimensional, application-specific services

Multiple Layered Web Architecture



Mining the World-Wide Web

Layer-0: Primitive data

Layer-1: dozen database relations representing types of objects (metadata)

document, organization, person, software, game, map, image,…

• **document**(file_addr, authors, title, publication, publication_date, abstract, language, table_of_contents, category_description, keywords, index, multimedia_attached, num_pages, format, first_paragraphs, size_doc, timestamp, access_frequency, links_out,...)

• **person**(last_name, first_name, home_page_addr, position, picture_attached, phone, e-mail, office_address, education, research_interests, publications, size_of_home_page, timestamp, access_frequency, ...)

• **image**(image_addr, author, title, publication_date, category_description, keywords, size, width, height, duration, format, parent_pages, colour_histogram, Colour_layout, Texture_layout, Movement_vector, localisation_vector, timestamp, access_frequency, ...)

Mining the World-Wide Web

Layer-2: simplification of layer-1

•**doc_brief**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, key_words, major_index, num_pages, format, size_doc, access_frequency, links_out)

•**person_brief** (last_name, first_name, publications,affiliation, e-mail, research_interests, size_home_page, access_frequency)

Layer-3: generalization of layer-2

•**cs_doc**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, keywords, num_pages, form, size_doc, links_out)

•**doc_summary**(affiliation, field, publication_year, count, first_author_list, file_addr_list)

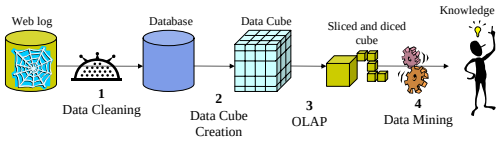
•**doc_author_brief**(file_addr, authors, affiliation, title, publication, pub_date, category_description, keywords, num_pages, format, size_doc, links_out)

•**person_summary**(affiliation, research_interest, year, num_publications, count)

Benefits of Multi-Layer Meta-Web

- Benefits:
  - Multi-dimensional Web info summary analysis
  - Approximate and intelligent query answering
  - Web high-level query answering (WebSQL, WebML)
  - Web content and structure mining
  - Observing the dynamics/evolution of the Web
- Is it realistic to construct such a meta-Web?
  - Benefits even if it is partially constructed
  - Benefits may justify the cost of tool development, standardization and partial restructuring

- Design of a Web Log Miner
  - Web log is filtered to generate a relational database
  - A data cube is generated form database
  - OLAP is used to drill-down and roll-up in the cube
  - OLAM is used for mining interesting knowledge



- Βασισμένες σε ευρετήρια: Αναζητούν σελίδες, τις δεικτοδοτούν και κατασκευάζουν τεράστια ευρετήρια βασισμένα σε λέξεις κλειδιά
- Χρήσιμες για τον εντοπισμό σελίδων που περιέχουν συγκεκριμένες λέξεις κλειδιά
- Προβλήματα
  - Ένα θέμα μπορεί να περιέχει χιλιάδες έγγραφα
  - Πολλά σχετικά με κάποιο θέμα έγγραφα μπορεί να μην περιέχουν τις λέξεις κλειδιά που το προσδιορίζουν

- Θα δούμε
  - Page Rank
  - HITS

Και οι δύο εκμεταλλεύονται την ύπαρξη links

PageRank: Capturing Page Popularity (Brin & Page'98)

Ο αρχικός αλγόριθμος του google, παρουσιάστηκε στην κλασική εργασία:
"The Anatomy of a Large-Scale Hypertextual Web Search Engine", Sergey Brin and Lawrence Page

Η εργασία περιλαμβάνει μια πολύ ενδιαφέρουσα «ιστορικής σημασίας» εισαγωγή

"We chose our system name, Google, because it is a common spelling of googol, or 10^{100} and fits well with our goal of building very large-scale search engines. "

The verb, "google", was added to the *Merriam Webster Collegiate Dictionary* and the *Oxford English Dictionary* in 2006, meaning, "to use the Google search engine to obtain information on the Internet." (source: Wikipedia)

Βασική Ιδέα

Ακόμα και αν ένα τεράστιο ευρετήριο με όλες τις λέξεις -> αυτό που έχει σημασία είναι οι σημαντικές precision vs recall

Τι είναι σημαντικό
οι συνδέσεις
Μια σελίδα που δέχεται πολλές αναφορές περιμένει κανείς να είναι γενικά πιο σημαντική

- Οι Web pages δεν είναι όλες το ίδιο "σημαντικές"
  www.joe-schmoe.com v www.stanford.edu
- Αναφορές (Inlinks) ως «ψήφοι» as votes
  www.stanford.edu 23,400 inlinks
  www.joe-schmoe.com 1 inlink

Ο PageRank βασίζεται στην «μέτρηση αναφορών» "citation counting", αλλά με μια βελτίωση:

Βασική Ιδέα

Δεν είναι όλες οι αναφορές το ίδιο σημαντικές!

Θεωρεί «έμμεσες αναφορές» "indirect citations" (αναφορές από σελίδες που επίσης έχουν πολλές αναφορές θεωρούνται πιο σημαντικές)

Αναδρομικό πρόβλημα!

Απλή Αναδρομική Διατύπωση

- Η ψήφος κάθε ακμής (αναφοράς) είναι ανάλογη της σημαντικότητας (PR)της σελίδας από την οποία προέρχεται
-
- Αν μια σελίδα P με σημαντικότητα (PR) x έχει n outlinks, κάθε link παίρνει x/n ψήφους

Παράδειγμα

Υπάρχει μια γενική ποσότητα PR που μοιράζεται στις σελίδες του συστήματος.
Έστω 4 σελίδες: A, B,C και D.
Αρχική προσεγγιστική τιμή για καθεμία PR = 0.25

Έστω B, C, και D έχουν link μόνο στο A, τότε όλα το PageRank PR( ) τους θα μαζευόταν στο A

$$PR(A) = PR(B) + PR(C) + PR(D).$$

Έστω τώρα ότι η B έχει link στη C, και η D έχει links και στο B και στο C
Η τιμή του PR μιας σελίδας μοιράζεται ανάμεσα στις εξωτερικές ακμές της
Άρα η ψήφος της B έχει αξία για την A 0.125 και 0.125 για την C.
Αντίστοιχα, μόνο το 1/3 του PageRank του D μετρά για PageRank του A (περίπου 0.083).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

Γενικός ορισμός του PageRank για μια σελίδα A:

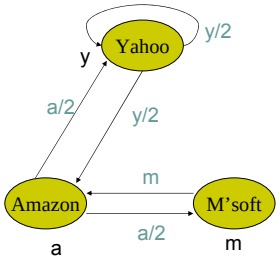
Έστω ότι η A έχει τις σελίδες T1...Tn που δείχνουν σε αυτήν (δηλαδή, αναφορές)

Έστω C(T) ο αριθμός των εξωτερικών ακμών μιας σελίδας A

PR(A) = PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)

Απλό μοντέλο «ροής» -"flow" model

Το web το 1839



$y = y/2 + a/2$
$a = y/2 + m$
$m = a/2$

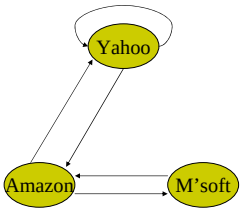
Λύση των εξισώσεων ροής

- 3 εξισώσεις, 3 άγνωστοι, όχι σταθερές
  - Μη μοναδική λύση
  - Οι λύσεις ισοδύναμες με scale factor
- Επιπρόσθετος περιορισμός για μοναδικότητα της λύσης
  - y+a+m = 1 (το συνολικό PR που μοιράζεται στις σελίδες)
  - y = 2/5, a = 2/5, m = 1/5

Διατύπωση με την μορφή πίνακα

- Ο πίνακας **M** έχει μια γραμμή και μια στήλη για κάθε web σελίδα (πίνακας γειτνίασης)
- Έστω ότι η σελίδα j έχει n outlinks
  - Αν j -> i, τότε $M_{ij}=1/n$
  - Αλλιώς, $M_{ij}=0$
- **M** είναι column stochastic matrix
  - Οι στήλες έχουν άθροισμα 1

Διατύπωση με την μορφή πίνακα (παράδειγμα)



$y = y/2 + a/2$
$a = y/2 + m$
$m = a/2$

	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

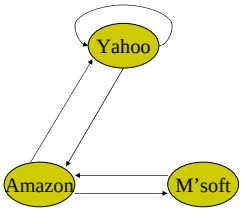
Άθροισμα 1 (οι ψήφοι του y)

Διατύπωση με την μορφή πίνακα

- Έστω **r** ένα διάνυσμα με μια εγγραφή web σελίδα
  - r_i είναι η σημαντικότητα (PR) της σελίδας i
  - r: rank vector

[PR(y)
PR(a)
PR(m)]

PR Διάνυσμα (παράδειγμα)



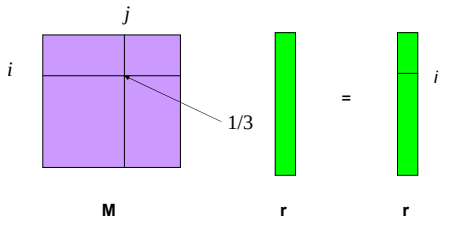
$y = y/2 + a/2$
$a = y/2 + m$
$m = a/2$

	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

r = Mr

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Έστω ότι η σελίδα j έχει links σε 3 σελίδες, συμπεριλαμβανομένου του i



Ιδιοδιανύσματα (eigenvectors)

- Οι εξισώσεις ροής μπορούν να γραφούν

$$r = M\,r$$

- Δηλαδή, ο rank vector είναι ένα ιδιοδιάνυσμα (eigenvector) του στοχαστικού πίνακα γειτνίασης του web
  - Συγκεκριμένα είναι το βασικό ιδιοδιάνυσμα (αυτό που αντιστοιχεί στην ιδιοτιμή 1)

Power Iteration method – Επαναληπτική Μέθοδο

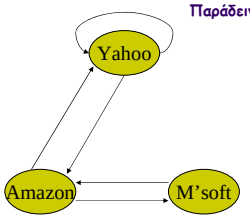
Ένα απλό επαναληπτικό σχήμα (aka relaxation)
Έστω N web σελίδες

Αρχικοποίηση: $\mathbf{r}^0 = [1/N,\ldots,1/N]^T$
Επανάληψη: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
Τερματισμός όταν $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$

$|\mathbf{x}|_1 = \Sigma_{1 \le i \le N}|x_i|$ είναι L_1 norm

Μπορεί να χρησιμοποιηθούν και άλλες μετρικές, πχ Ευκλείδεια

Παράδειγμα



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

y	1/3	1/3	5/12	3/8		2/5
a =	1/3	1/2	1/3	11/24	...	2/5
m	1/3	1/6	1/4	1/6		1/5

Μοντέλο Τυχαίου Surfer (random walk)

Το PageRank μιας σελίδας μπορεί επίσης να θεωρηθεί ότι εκφράζει την πιθανότητα ένας τυχαίος surfer να φτάσει σε αυτήν (δηλαδή, εκφράζει πόσο δημοφιλής είναι)

Ένας τυχαίος surfer ξεκινά από μια τυχαία σελίδα και συνεχίζει να κάνει click σε links, χωρίς να επιστρέφει σε προηγούμενη σελίδα

- Τη χρονική στιγμή t, ο surfer είναι σε κάποια σελίδα P
- Τη χρονική στιγμή $t+1$, ο surfer ακολουθεί ένα εξωτερικό link- outlink του P τυχαία - uniformly at random
- Φτάνει σε κάποια σελίδα Q του P
- Συνεχίζει την παραπάνω διαδικασία επ άπειρω

Έστω $\mathbf{p}(t)$ το διάνυσμα του οποίου το i^{th} στοιχείο είναι η πιθανότητα ο surfer να είναι στη σελίδα i τη χρονική στιγμή t

- $\mathbf{p}(t)$ probability distribution - κατανομή πιθανότητας στις σελίδες

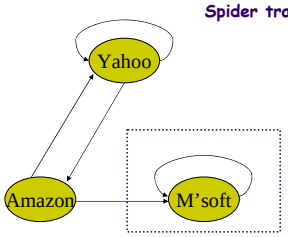
The stationary distribution

- Where is the surfer at time t+1?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t)$ M (transition probabaility from state j (web page) state I
- Suppose the random walk reaches a state such that $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t) = \mathbf{p}(t)$
  - Then $\mathbf{p}(t)$ is called a stationary distribution for the random walk
- Our rank vector $\mathbf{r}$ satisfies $\mathbf{r} = \mathbf{M}\mathbf{r}$
  - So it is a stationary distribution for the random surfer

A central result from the theory of random walks (aka Markov processes):

For graphs that satisfy certain conditions, the stationary distribution is unique and eventually will be reached no matter what the initial probability distribution at time t = 0.

Spider traps (παράδειγμα)



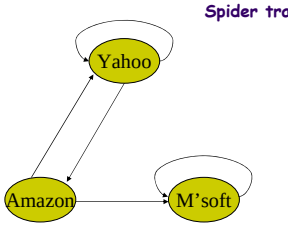
	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

y		1	1	3/4	5/8	0
a	=	1	1/2	1/2	3/8	. . . 0
m		1	3/2	7/4	2	3

Spider traps

- Μια ομάδα σελίδων είναι μια αραχνο-παγίδα spider trap αν δεν υπάρχουν ακμές – από την ομάδα σε σελίδες εκτός της ομάδας
  - Ο τυχαίος surfer παγιδεύεται
- Οι συνθήκες που χρειάζονται για το θεώρημα των τυχαίων περιπάτων παύουν να ισχύουν

Spider traps (παράδειγμα)



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

y		1	1	3/4	5/8	0
a	=	1	1/2	1/2	3/8	. . . 0
m		1	3/2	7/4	2	3

Επέκταση Μοντέλου

Σε κάθε βήμα, ο τυχαίος surfer έχει δύο δυνατότητες:
- Με πιθανότητα β, ακολουθεί ένα τυχαίο link
- Με πιθανότητα 1-β πετάγεται σε κάποια άλλη σελίδα τυχαία
- Τιμές για το β 0.8 - 0.9

Καταφέρνει να βγει από την παγίδα μετά από κάποιες χρονικές στιγμές

Επέκταση Μοντέλου

Αρχικός ορισμός του PageRank για μια σελίδα A:

PR(A) = PR(T1)/C(T1) + ... + PR(Tn)/C(Tn)

Ορισμός με τον παράγοντας απόσβεσης d (damping factor) μεταξύ του 0 και του 1

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

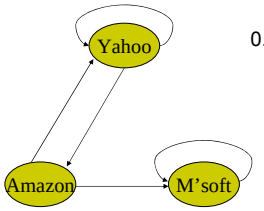
Ώστε το άθροισμα να είναι 1 -> 1-d/N
Ο πρώτος παράγοντας με την ίδια πιθανότητα διαλέγω οποιαδήποτε σελίδα

Μοντέλο Τυχαίου Surfer

Ένας τυχαίος surfer ξεκινά από μια τυχαία σελίδα και συνεχίζει να κάνει click σε links, χωρίς να επιστρέφει σε προηγούμενη σελίδα αλλά τελικά *βαριέται* και ξεκινά από κάποια άλλη τυχαία σελίδα

Το d (ο παράγοντας απόσβεσης) είναι η πιθανότητα σε κάθε σελίδα ο τυχαίος surfer να βαρεθεί και να αρχίσει από κάποια άλλη τυχαία σελίδα

Παράδειγμα (d=0.8)



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

y		1	1.00	0.84	0.776	7/11
a	=	1	0.60	0.60	0.536	5/11
m		1	1.40	1.56	1.688	21/11

Matrix formulation

- Έστω N σελίδες
  - Έστω σελίδα j, με ένα σύνολο outlinks O(j)
  - $M_{ij} = 1/|O(j)|$ αν j -> i and $M_{ij} = 0$ otherwise
  - Η τυχαία μεταπήδηση είναι ισοδύναμη με το
    - Να προσθέσουμε ένα τυχαίο link από το j το σε οποιαδήποτε άλλη σελίδα με $(1-\beta)/N$
    - Ελάττωση της πιθανότητας να ακολουθήσουμε ένα outlink από $1/|O(j)|$ σε $\beta/|O(j)|$
    - Ή ισοδύναμα: χρέωσε σε κάθε σελίδα ένα ποσοστό $(1-\beta)$ της τιμής της και κάνε κατανομή αυτού ισοδύναμα

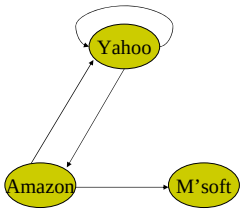
- Κατασκευή του Nxn πίνακα *A*
  - $A_{ij} = \beta M_{ij} + (1-\beta)/N$
- *Ο* ***A*** *είναι στοχαστικός πίνακας*
- Το page rank διάνυσμα **r είναι το βασικό ιδιοδιάνυσμα αυτού του πίνακα**

  r = Ar

  Ισοδύναμα, **r** είναι stationary distribution των τυχαίων περιπάτων με μεταπηδήσεις (random walk with teleports)

Αδιέξοδα

Οι σελίδες χωρίς outlinks για τον τυχαίο surfer



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 1/15 \end{bmatrix}$$

Non-stochastic!

y		1	1	0.787	0.648	0
a	=	1	0.6	0.547	0.430	0
m		1	0.6	0.387	0.333	0

Ο Αλγόριθμος PageRank

Dealing with dead-ends

- Μεταπήδηση
  - Για αδιέξοδα, ακολούθησε τυχαία μεταπήδηση με πιθανότητα 1
  - Τροποποίησε τον πίνακα
- Ψαλίδισε της -- Prune and propagate
  - Preprocess the graph to eliminate dead-ends
  - Might require multiple passes
  - Compute page rank on reduced graph
  - Approximate values for deadends by propagating values from reduced graph

Ο Αλγόριθμος PageRank

Μια σελίδα μπορεί να έχει υψηλό PR αν υπάρχουν πολλές σελίδες που δείχνουν σε αυτήν ή όταν κάποιες σελίδες που δείχνουν σε αυτήν έχουν υψηλό PR

Και οι δύο περιπτώσεις έχουν σημασία:
Πχ στη δεύτερη περίπτωση αν υπάρχει link από πχ Yahoo!

Spamdexing

Content spam – Link spam

Google bombing:

Προσθήκη αναφορών που επηρεάζουν άμεσα το PR

Link farms:

Σελίδες που αναφέρονται η μία στην άλλη

Google: Άλλα στοιχεία

Anchor Text

Το κείμενο που υπάρχει στα links έχει διαφορετική αντιμετώπιση
Οι περισσότερες μηχανές αναζήτησης το συσχέτιζαν με τη σελίδα στην οποία εμφανίζεται
Google και με τη σελίδα στην οποία δείχνει

- Ποιο ακριβή πληροφορίες για τις σελίδες που δείχνουν παρά για τις σελίδες στις οποίες εμφανίζονται
- Μπορεί να δείχνουν σε σελίδες που δεν έχουν κείμενο αλλά εικόνες, προγράμματα, κλπ

Google: Αρχιτεκτονική

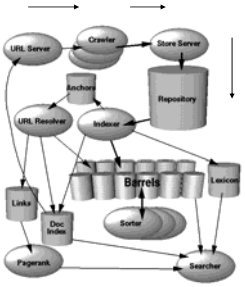
Most of Google is implemented in C or C++ for efficiency and can run in either Solaris or Linux.

The web crawling (downloading of web pages) is done by several distributed crawlers.

There is a URLserver that sends lists of URLs to be fetched to the crawlers.

The web pages that are fetched are then sent to the storeserver.

The storeserver then compresses and stores the web pages into a repository.



Google: Αρχιτεκτονική

Every web page has an associated ID number called a **docID** which is assigned whenever a new URL is parsed out of a web page.

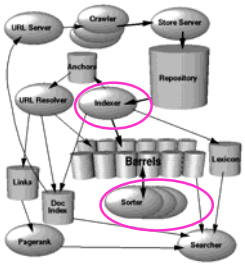
The indexing function is performed by the indexer and the sorter.

The indexer reads the repository, uncompresses the documents, and parses them.

document -> a set of word occurrences called **hits**.

Hits: word, position in document, an approximation of font size, and capitalization.

The indexer distributes these hits into a set of "barrels", creating a partially sorted forward index.

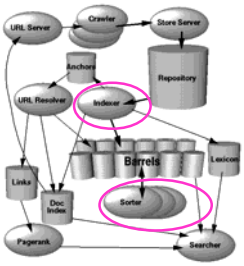


Google: Αρχιτεκτονική

Indexer:

It parses out all the links in every web page and stores important information about them in an anchors file.

This file contains enough information to determine where each link points from and to, and the text of the link.

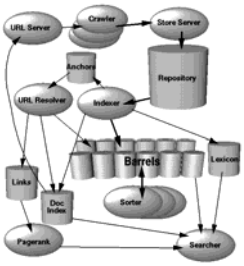


Google: Αρχιτεκτονική

URLresolver relative URLs -> absolute URLs -> docIDs.

The sorter takes the barrels, which are sorted by docID and resorts them by wordID to generate the inverted index.

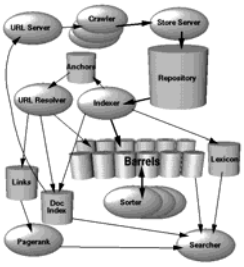
+ lexicon



Google: Αρχιτεκτονική

The searcher is run by a web server

uses the lexicon built by DumpLexicon together with the inverted index and the PageRanks to answer queries.



Ο Αλγόριθμος HITS

Problems with the Web linkage structure

- Not every hyperlink represents an endorsement
  - Other purposes are for navigation or for paid advertisements
  - If the majority of hyperlinks are for endorsement, the collective opinion will still dominate

Μια αυθεντία (authority) για κάποιο θέμα σπάνια θα έχει link σε αντίπαλη αυθεντία στον ίδιο τομέα

Οι αυθεντικές σελίδες σπάνια είναι περιγραφικές/αντιπροσωπευτικές

Ο Αλγόριθμος HITS

Ο αλγόριθμος HITS (Hyperlink-Induced Topic Search)

Για κάθε θέμα: δύο τύποι σελίδων

Αυθεντική: Μια σελίδα που είναι αυθεντία σε ένα θέμα και αναγνωρίζεται ως τέτοια από άλλες σελίδες (δηλαδή, υπάρχουν πολλοί σύνδεσμοι σε αυτήν)

Κομβικοί: Μια σελίδα που αναφέρεται σε μια αυθεντική σελίδα

Βασική ιδέα:
Οι σελίδες που αναφέρονται από άλλες σελίδες συχνά πρέπει να είναι αυθεντίες (Authorities)
Οι σελίδες που αναφέρουν πολλές άλλες σελίδες πρέπει να είναι καλά κομβικά σημεία (hubs)

Κομβικοί Αυθεντικοί

Ο Αλγόριθμος HITS

Βασική ιδέα του HITS

Καλές αυθεντίες είναι αυτές στις οποίες αναφέρονται καλά κομβικά σημεία

Καλά κομβικά σημεία είναι αυτά τα οποία αναφέρονται σε καλές αυθεντίες

Αναδρομική έκφραση

Ο Αλγόριθμος HITS

Το web ως ένας κατευθυνόμενος γράφος

Κόμβοι: ιστοσελίδες

Ακμή από Α στον Β: η ιστοσελίδα Α έχει έναν υπερ-σύνδεσμο στην ιστοσελίδα Β

Ο αλγόριθμος σε 2 φάσεις:

Φάση Ι: (δειγματοληπτικό στάδιο) ένα σύνολο σελίδων που αποτελεί το βασικό σύνολο για κάποιο θέμα

Φάση ΙΙ: (επαναληπτικό στάδιο) επεξεργασία του βασικού συνόλου για τον εντοπισμό καλών αυθεντικών και κομβικών ιστοσελίδων

Ο Αλγόριθμος HITS

Φάση Ι: Υπολογισμός βασικού συνόλου

1. Υπολογισμός αρχικού συνόλου: σύνολο-ρίζα

Κλασικοί μέθοδοι: πχ ανάκτηση όλων των σελίδων που περιέχουν τις λέξεις κλειδιά

(περιμένουμε ότι θα περιέχει (τουλάχιστον) αναφορές προς σχετικές σελίδες)

Ο Αλγόριθμος HITS

Φάση Ι: Υπολογισμός βασικού συνόλου

2. Σελίδες-σύνδεσμοι:

σελίδα που είτε συμπεριλαμβάνει σύνδεσμο που να αναφέρεται σε έναν κόμβο p στο σύνολο ρίζα (p είναι αυθεντία) είτε

Ένας κόμβος p στο σύνολο ρίζα (p είναι κομβικό σημείο) περιέχει σύνδεσμο που αναφέρεται σε αυτήν

Βασικό Σύνολο: διεύρυνση του συνόλου-ρίζα ώστε να περιλαμβάνει και τις σελίδες συνδέσμους – *Βασικές ιστοσελίδες*

Ο Αλγόριθμος HITS

Φάση ΙΙ: Ποιες βασικές ιστοσελίδες είναι κόμβοι και αυθεντίες

Κάθε βασική σελίδα p δύο τιμές:

h_p - Συντελεστής Κομβικού Ρόλου (πολλούς δείκτες σε αυθεντικές)

a_p - Συντελεστής Αυθεντικότητας (πολλοί δείκτες από κομβικές σε αυτήν)

Ο Αλγόριθμος HITS

Βασική διαφορά από τον Page Rank

- Δύο τιμές ανά σελίδα (αυθεντία – κομβικό σημείο)
- Θεματικά υποσύνολα του web γράφου - ξεκινάμε από το βασικό σύνολο

Ο Αλγόριθμος HITS

Φάση ΙΙ: Ποιες βασικές ιστοσελίδες είναι κόμβοι και αυθεντίες

Αρχικοποίηση, $\forall$ p, $h_p = 1$ και $a_p = 1$

Επαναληπτικά, αυξάνεται

$a_p = \Sigma\ h_q$

Βασικές σελίδες q που δείχνουν στην p

$h_p = \Sigma\ a_q$

Βασικές σελίδες q στις οποίες δείχνει η p

Αναπαράσταση με πίνακες

Έστω το βασικό σύνολο σελίδων {1, 2, …, n}

Πίνακας Γειτνίασης (adjacency matrix) B: n x n

B[i, j] = 1 αν η σελίδα i περιέχει σύνδεσμο που δείχνει στη σελίδα j

Έστω h = <h_1, h_2, …, h_n> το διάνυσμα συντελεστών κομβικών ρόλων και a = <a_1, a_2, …, a_n> το διάνυσμα συντελεστών αυθεντικότητας

(αντίστοιχο του r vector)

Οι κανόνες ενημέρωσης

Αρχικά

h = B a a = B^T h

1η επανάληψη

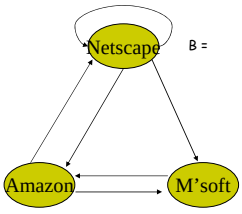
h = B B^T h = (B B^T) h a = B^T B a = (B^T B) a

2η επανάληψη

h = (B B^T)² h a = (B^T B)² a

Σύγκλιση στα ιδιοδιανύσματα του BB^T και B^TB αν κανονικοποιηθούν αρχικά οι συντελεστές

Διατύπωση με την μορφή πίνακα (παράδειγμα)



B =

n	m	a
1	1	1
0	0	1
1	1	0

B^T =

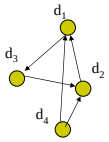
n	m	a
1	0	1
1	0	1
1	1	0

B B^T =

n	m	a
3	1	2
1	1	0
2	0	2

$$\begin{bmatrix} 3 & 1 & 2 \\ 1 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 2 \\ 4 \end{bmatrix} \quad \ldots$$

h = BB^T h



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

"Adjacency matrix"

Initial values: a=h=1

$$h(d_i) = \sum_{d_j \in OUT(d_i)} a(d_j)$$
$$a(d_i) = \sum_{d_j \in IN(d_i)} h(d_j)$$

Iterate

Normalize:

$$\sum_i a(d_i)^2 = \sum_i h(d_i)^2 = 1$$

$$\vec{h} = A\vec{a}; \quad \vec{a} = A^T\vec{h}$$
$$\vec{h} = AA^T\vec{h}; \quad \vec{a} = A^TA\vec{a}$$

Again eigenvector problems…

Προβλήματα

- Drifting: όταν ένα κομβικό σημείο περιέχει πολλά θέματα
- Topic hijacking: όταν πολλές σελίδες από το ίδιο web site δείχνουν στο ίδιο δημοφιλές