# Principles of Knowledge Discovery in Data

Winter 2007

**Chapter 8: Web Mining**

Dr. Osmar R. Zaïane

University of Alberta

# Course Content

- Introduction to Data Mining
- Association analysis
- Sequential Pattern Analysis
- Classification and prediction
- Contrast Sets
- Data Clustering
- Outlier Detection
- Web Mining

# Course Content

- Introduction to Data Mining
- Association Analysis
- Sequential Pattern Analysis
- Classification and prediction
- Contrast Sets
- Data Clustering
- Outlier Detection
- Web Mining
- Other topics if time permits (spatial data, biomedical data, etc.)

# Objectives

Understand the different knowledge discovery issues in data mining from the World Wide Web.

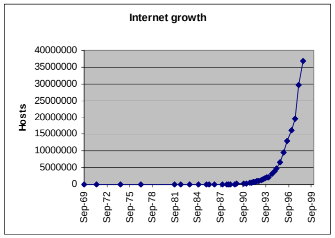
Distinguish between resource discovery and Knowledge discovery from the Internet.

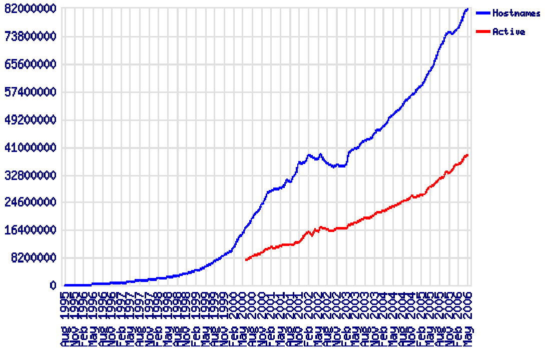
# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# WWW: Growth

- Growing and changing very rapidly
  - 5 million documents in 1995; 320 million documents in 1998; More than 1 billion in 2000.
  - Estimates in 2005: Google → 8 billion; Yahoo → 20 billion





- Number of web sites
  - One new Web server every 2 hours (1998)
  - Today, Netcraft survey says 82 million sites

**http://news.netcraft.com/archives/web_server_survey.html**

# WWW: Facts

- The web is the largest database ever built
- The Web is not a relational database.
  Some of it is structured, some is semi-structured and some is unstructured.
- No standards, unstructured and heterogeneous
- The size of the Web is technically infinite
- The content is dynamic and has duplicates and inconsistencies.
- Queries are non-deterministic

**Need for better resource discovery and knowledge extraction.**

**The Asilomar Report urges the database research community to contribute in deploying new technologies for resource and information retrieval from the World-Wide Web.**

# WWW: Incentives

- Enormous wealth of information on web
- The web is a huge, widely distributed collection of:
  - Documents of all sorts ( static as well as dynamically generated content and services)
  - Hyper-link information
  - Access and usage information
- Mine interesting nuggets of information leads to wealth of information and knowledge
- Challenge: Unstructured, huge, dynamic.

# WWW and its Problems

- Web: A huge, widely-distributed, highly heterogeneous, semi-structured, interconnected, evolving, hypertext/hypermedia information repository with no coordination in content creation and distribution.
- Problems:
  - the “*abundance*” problem:
    - 99% of info of no interest to 99% of people
  - *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

# Web Mining

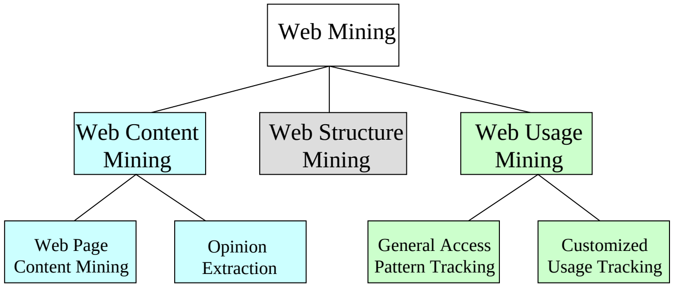
- Web mining is the application of data mining techniques and other means of extraction of knowledge for the integration of information gathered over the World Wide Web in all its forms: content, structure or usage. The integrated information is useful for either:
  - Understanding on-line user behaviour;
  - Retrieving/consolidating relevant knowledge/resources;
  - Evaluate the effectiveness of particular web sites or web-based applications;
- Web mining research integrates research from Databases, Data Mining, Information retrieval, Machine learning, Natural language processing, software agent communication, etc.

# Challenges for Web Applications

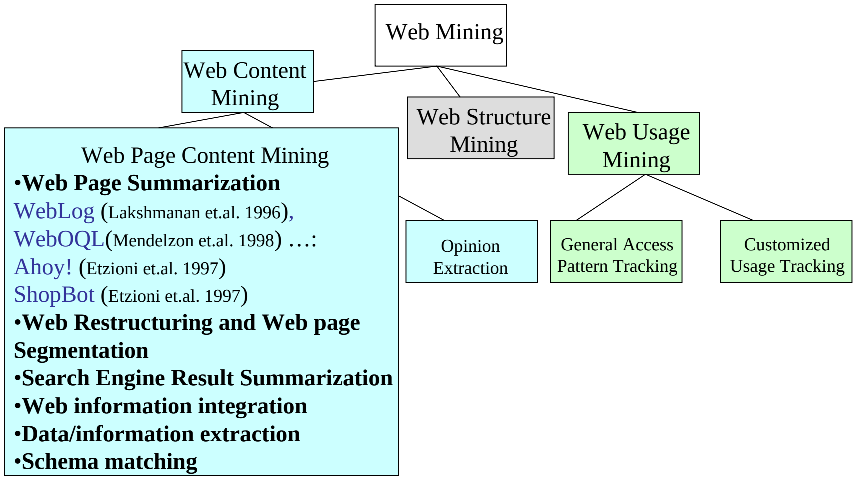
- Finding Relevant Information (high-quality Web documents on a specified topic/concept/issue.)
- Creating knowledge from Information available
- Personalization of the information
- Learning about customers / individual users; understanding user navigational behaviour; understanding on-line purchasing behaviour.

**Web Mining can play an important Role!**

# Web Mining Taxonomy



# Web Mining Taxonomy



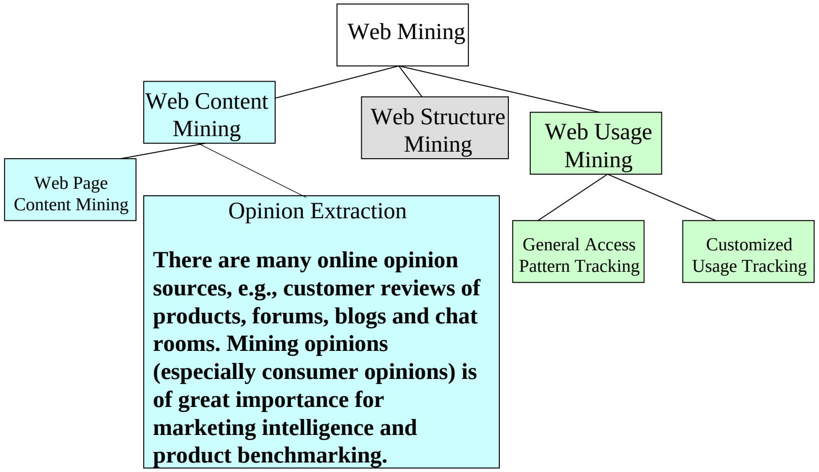
Web Page Content Mining

- **Web Page Summarization**

WebLog (Lakshmanan et.al. 1996),
WebOQL(Mendelzon et.al. 1998) …:
Ahoy! (Etzioni et.al. 1997)
ShopBot (Etzioni et.al. 1997)

- **Web Restructuring and Web page Segmentation**
- **Search Engine Result Summarization**
- **Web information integration**
- **Data/information extraction**
- **Schema matching**

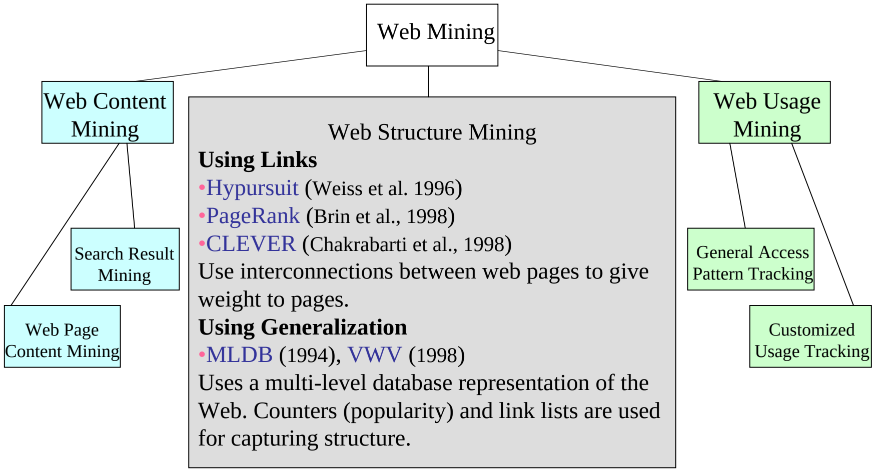
# Web Mining Taxonomy



Opinion Extraction

**There are many online opinion sources, e.g., customer reviews of products, forums, blogs and chat rooms. Mining opinions (especially consumer opinions) is of great importance for marketing intelligence and product benchmarking.**

# Web Mining Taxonomy



Web Structure Mining

**Using Links**

- Hypursuit (Weiss et al. 1996)
- PageRank (Brin et al., 1998)
- CLEVER (Chakrabarti et al., 1998)

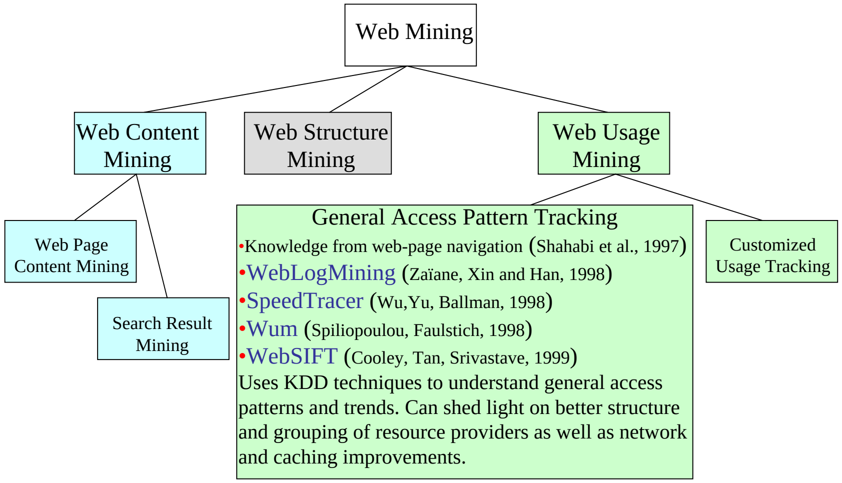
Use interconnections between web pages to give weight to pages.

**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.

# Web Mining Taxonomy

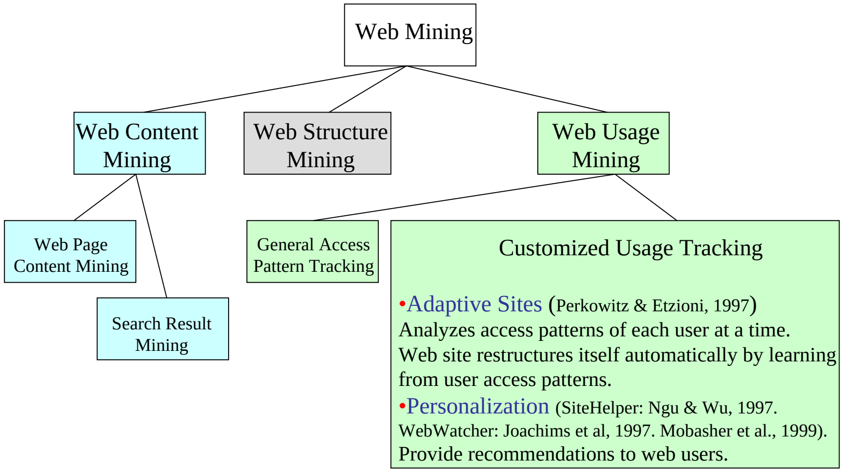


General Access Pattern Tracking

- Knowledge from web-page navigation (Shahabi et al., 1997)
- WebLogMining (Zaïane, Xin and Han, 1998)
- SpeedTracer (Wu,Yu, Ballman, 1998)
- Wum (Spiliopoulou, Faulstich, 1998)
- WebSIFT (Cooley, Tan, Srivastave, 1999)

Uses KDD techniques to understand general access patterns and trends. Can shed light on better structure and grouping of resource providers as well as network and caching improvements.

# Web Mining Taxonomy



Web Mining

Web Content Mining

Web Structure Mining

Web Usage Mining

Web Page Content Mining

Search Result Mining

General Access Pattern Tracking

Customized Usage Tracking

•Adaptive Sites (Perkowitz & Etzioni, 1997)
Analyzes access patterns of each user at a time.
Web site restructures itself automatically by learning from user access patterns.

•Personalization (SiteHelper: Ngu & Wu, 1997. WebWatcher: Joachims et al, 1997. Mobasher et al., 1999).
Provide recommendations to web users.

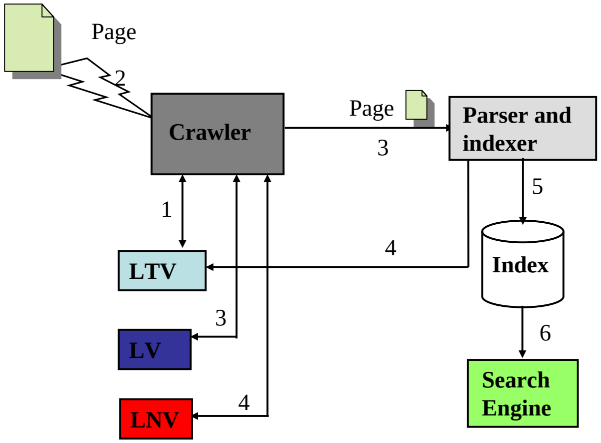
# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# Web Content Mining: a huge field with many applications

- **Data/information extraction**: Extraction of structured data from Web pages, such as products and search results. Extracting such data allows one to provide services. Two main types of techniques, machine learning and automatic extraction exist.
- **Web information integration and schema matching**: Although the Web contains a huge amount of data, each web site (or even page) represents similar information differently. How to identify or match semantically similar data is a very important problem with many practical applications.
- **Opinion extraction from online sources**: There are many online opinion sources, e.g., customer reviews of products, forums, blogs and chat rooms. Mining opinions (especially consumer opinions) is of great importance for marketing intelligence and product benchmarking.
- **Knowledge synthesis**: Concept hierarchies or ontology are useful in many applications. However, generating them manually is very time consuming. A few methods that explores the information redundancy of the Web exist. The main application is to synthesize and organize the pieces of information on the Web to give the user a coherent picture of the topic domain.
- **Segmenting Web pages and detecting noise**: In many Web applications, one only wants the main content of the Web page without advertisements, navigation links, copyright notices. Automatically segmenting Web page to extract the main content of the pages is an interesting problem. A number of interesting techniques have been proposed in the past few years.

# Search engine general architecture

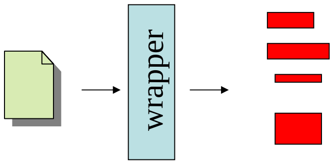


# Search Engines are not Enough

- Most of the knowledge in the World-Wide Web is buried inside documents.
- Search engines (and crawlers) barely scratch the surface of this knowledge by extracting keywords from web pages.
- There is text mining, text summarization, natural language statistical analysis, etc., but not the scope of this tutorial.

# Web page Summarization or Web Restructuring

- Most of the suggested approaches are limited to known groups of documents, and use custom-made wrappers.



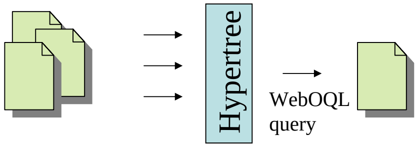
Ahoy!
WebOQL
Shopbot
…

# Discovering Personal Homepages

- Ahoy! (shakes et al. 1997) uses Internet services like search engines to retrieve resources a person’s data.
- Search results are parsed and using heuristics, typographic and syntactic features are identified inside documents.
- Identified features can betray personal homepages.

# Query Language for Web Page Restructuring

- WebOQL (Arocena et al. 1998) is a declarative query language that retrieves information from within Web documents.
- Uses a graph hypertree representation of web documents.



•CNN pages
•Tourist guides
•Etc.

# Shopbot

- Shopbot (Doorendos et al. 1997) is shopping agent that analyzes web page content to identify price lists and special offers.
- The system learns to recognize document structures of on-line catalogues and e-commerce sites.
- Has to adjust to the page content changes.

# Mine What Web Search Engine Finds

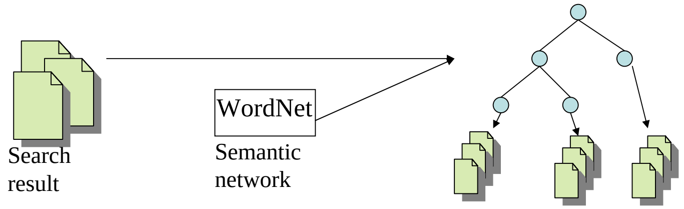
- Current Web search engines: convenient source for mining
  - keyword-based, return too many answers, low quality answers, still missing a lot, not customized, etc.
- Data mining will help:
  - coverage: “Enlarge and then shrink,” using synonyms and conceptual hierarchies
  - better search primitives: user preferences/hints
  - linkage analysis: authoritative pages and clusters
  - Web-based languages: XML + WebSQL + WebML
  - customization: home page + Weblog + user profiles

# Refining and Clustering Search Engine Results

- WebSQL (Mendelzon et al. 1996) is an SQL-like declarative language that provides the ability to retrieve pertinent documents.
- Web documents are parsed and represented in tables to allow result refining.
- [Zamir et al. 1998] present a technique using COBWEB that relies on snippets from search engine results to cluster documents in significant clusters.

# Ontology for Search Results

- There are still too many results in typical search engine responses.
- Reorganize results using a semantic hierarchy (Zaïane et al. 2001).



# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# Web Structure Mining

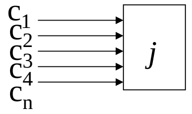
- Hyperlink structure contains an enormous amount of concealed human annotation that can help automatically infer notions of “authority” in a given topic.
- Web structure mining is the process of extracting knowledge from the interconnections of hypertext document in the world wide web.
- Discovery of influential and authoritative pages in WWW.

# Citation Analysis in Information Retrieval

- Citation analysis was studied in information retrieval long before WWW came into the scene.
- Garfield's *impact factor* (1972): It provides a numerical assessment of journals in the journal citation.
- Kwok (1975) showed that using citation titles leads to good cluster separation.

# Citation Analysis in Information Retrieval

- Pinski and Narin (1976) proposed a significant variation on the notion of impact factor, based on the observation that not all citations are equally important.
  - A journal is influential if, recursively, it is heavily cited by other influential journals.
  - *influence weight:* The influence of a journal *j* is equal to the sum of the influence of all journals citing *j*, with the sum weighted by the amount that each cites *j*.



$$IW_j=\sum_{n}^{i=1} \alpha_i c_i$$

## HyPursuit

- Hypursuit (Weiss et al. 1996) groups resources into clusters according to some criteria. Clusters can be clustered again into clusters of upper level, and so on into a hierarchy of clusters.
- Clustering Algorithm
  - Computes clusters: set of related pages based on the semantic info embedded in hyperlink structure and other criteria.
  - abstraction function

## Search for Authoritative Pages

A good authority is a page pointed by many good hubs, while a good hub is a page that point to many good authorities.

This mutually enforcing relationship between the hubs and authorities serves as the central theme in our exploration of link based method for search, and the automated compilation of high-quality web resources.

## Discovery of Authoritative Pages in WWW

- Hub/authority method (Kleinberg, 1998):
  - Prominent authorities often do not endorse one another directly on the Web.
  - Hub pages have a large number of links to many relevant authorities.
  - Thus hubs and authorities exhibit a mutually reinforcing relationship:

## Hyperlink Induced Topic Search (HITS)

- Kleinberg’s HITS algorithm (1998) uses a simple approach to finding quality documents and assumes that if document A has a hyperlink to document B, then the author of document A thinks that document B contains valuable information.
- If A is seen to point to a lot of good documents, then A’s opinion becomes more valuable and the fact that A points to B would suggest that B is a good document as well.

## General HITS Strategy

HITS algorithm applies two main steps.

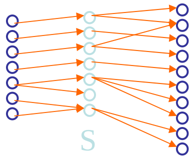
- A sampling component which constructs a focused collection of thousand web pages likely to be rich in authorities.

- A weight-propagation component, which determines the numerical estimates of hub and authority weights by an iterative procedure.

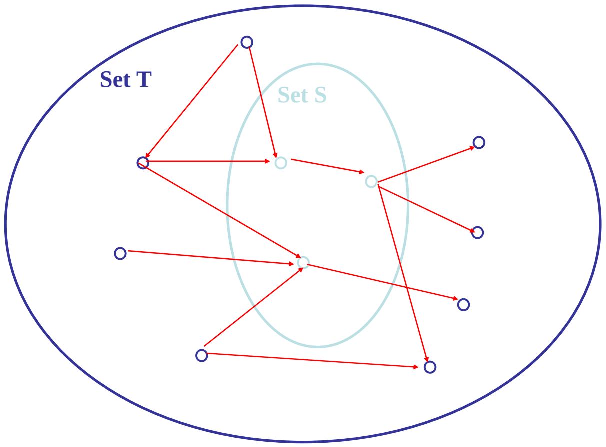
## Steps of HITS Algorithm

- Starting from a user supplied query, HITS assembles an initial set S of pages:

  The initial set of pages is called root set. These pages are then expanded to a larger root set T by adding any pages that are linked to or from any page in the initial set S.



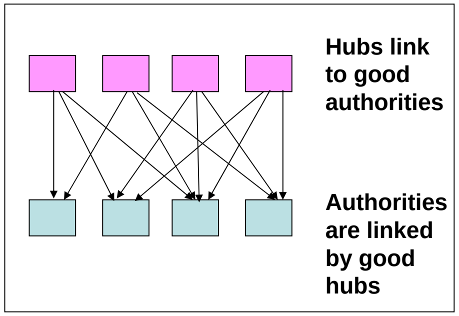
- HITS then associates with each page p a hub weight h(p) and an authority weight a(p), all initialized to one.



- HITS then iteratively updates the hub and authority weights of each page. Let p → q denote “page p has an hyperlink to page q”. HITS updates the hubs and authorities as follows:

$$h(p) = \sum_{p \to q} a(q)$$

$$a(p) = \sum_{q \to p} h(q)$$



Hubs link to good authorities

Authorities are linked by good hubs

# Further Enhancement for Finding Authoritative Pages in WWW

- The CLEVER system (Chakrabarti, et al. 1998-1999)
  - builds on the algorithmic framework of extensions based on both content and link information.
- Extension 1: mini-hub pagelets
  - prevent "*topic drifting*" on large hub pages with many links, based on the fact: Contiguous set of links on a hub page are more focused on a single topic than the entire page.
- Extension 2. Anchor text
  - make use of the text that surrounds hyperlink definitions (href's) inWeb pages, often referred to as *anchor* text
  - boost the weights of links which occur near instances of query terms.

# CLEVER System

- The output of the HITS algorithm for the given search topic is a short list consisting of the pages with largest hub weights and the pages with largest authority weights.
- HITS uses a purely link-based computation once the root set has been assembled, with no further regard to the query terms.
- In HITS all the links out of a hub page propagate the same weight, the algorithm does not take care of hubs with multiple topics.

# Extensions in CLEVER

The CLEVER system builds on the algorithmic framework of extension based on content and link information.

Extension 1: mini-hub pagelets

Prevent “topic drifting” on large hub pages with many links, based on the fact: Contiguous set of links on a hub page are more focused on a single topic than the entire page.

# Extensions in CLEVER

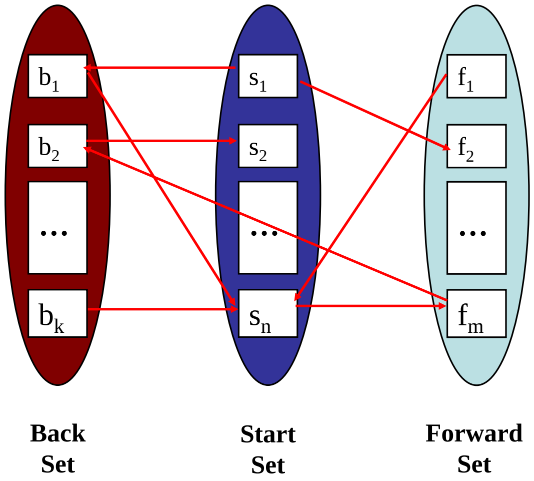
Extension 2. Anchor text

- Make use of the text that surrounds hyperlink definitions (href’s) in Web pages, often referred as anchor text.
- Boost the weights of links which occurs near instance of the query term.

# Connectivity Server

- Connectivity server (Bharat et al. 1998) also exploit linkage information to find most relevant pages for a query.
- HITS algorithm and CLEVER uses the 200 pages indexed by the AltaVista search engine as the base set.
- Connectivity Server uses entire set of pages returned by the AltaVista search engines to find result of the query.

- Connectivity server in its base operation, the server accept a query consisting of a set L of one or more URLs and returns a list of all pages that point to pages in L (predecessors) and list of all pages that are pointed to from pages in L (successors).
- Using this information Connectivity Server includes information about all the links that exist among pages in the neighborhood.

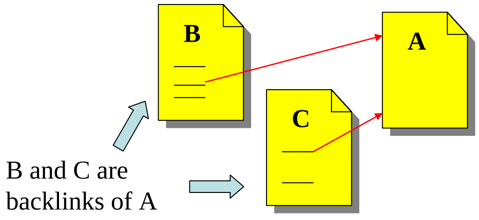


- The neighborhood graph is the graph produced by a set L of start pages and the predecessors of L, and all the successors of L and the edges among them.
- Once the neighborhood graph is created, the Connectivity server uses Kleinberg’s method to analyze and detect useful pages and to rank computation on it.
- Outlier filtering (Bharat & Henzinger 1998-1999) integrates textual content: nodes in neighborhood graph are term vectors. During graph expansion, prune nodes distant from query term vector. Avoids contamination from irrelevant links.

# Ranking Pages Based on Popularity

- Page-rank method ( Brin and Page, 1998): Rank the "importance" of Web pages, based on a model of a "random browser.“
  - Initially used to select pages to revisit by crawler.
  - Ranks pages in Google’s search results.
- In a simulated web crawl, following a random link of each visited page may lead to the revisit of popular pages (pages often cited).
- Brin and Page view Web searches as random walks to assign a topic independent “rank” to each page on the world wide web, which can be used to reorder the output of a search engine.
- The number of visits to each page is its PageRank. PageRank estimates the visitation rate ➔ popularity score.

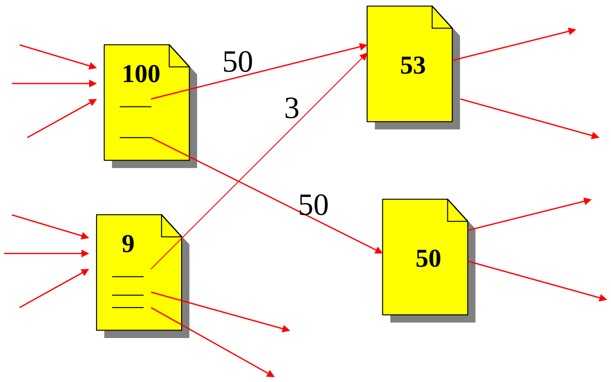
# Page Rank: A Citation Importance Ranking



B and C are backlinks of A

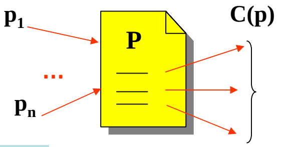
- Number of backlinks (~citations)

# Idealized PageRank Calculation



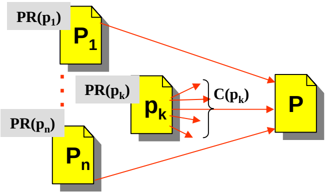
# PageRank

Each Page *p* has a number of links coming out of it C(*p*) (C for citation), and a number of pages pointing to it $p_1, p_2 \ldots.., p_n$.



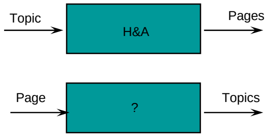
PageRank of P is obtained by

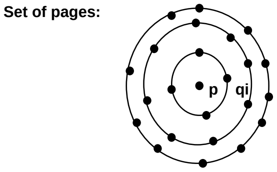
$$PR(p) = (1-d) + \left(\sum_{k=1}^{n} \frac{PR(p_k)}{C(p_k)}\right)$$



## Reputation of a Page: The TOPICS Method

Inverting H&A Computation





**Set of terms:** all terms *t* that appear in *p* or some of the *qi*'s.

$$R(p,t) = \frac{d}{N_t}$$

```
For i =1,2, …,k
    For each path q_1 → q_2 → … → q_i → p
        For each term t in q_i
            $R(p,t) = R(p,t) + \left( \frac{(1-d)^i}{\prod_{j=1}^{i} O(q_i)} \right) \frac{d}{N_t}$
```

## Simplification for real time Implementation of Topics

- k=1, O(q)=7.2 , d=0.1 (use of snippets from 1000 pages linking to p)

$$\boldsymbol{R(p,t) = C \times \sum_{q \to p} \frac{1}{N_t}}$$ (q contains t)

- That is, R(p,t) ~I(p,t)/$N_t$

## Comparaison

- Google assigns initial ranking and retains them independently of any queries. This makes it faster.
- CLEVER and Connectivity server assembles different root set for each search term and prioritizes those pages in the context of the particular query.
- Google works in the forward direction from link to link.
- CLEVER and Connectivity server looks both in the forward and backward direction.
- Both the page-rank and hub/authority methodologies have been shown to provide qualitatively good search results for broad query topics on the WWW.
- Hyperclass (Chakrabarti 1998) uses content and links of exemplary page to focus crawling of relevant web space.

## Nepotistic Links

- Nepotistic links are links between pages that are present for reasons other than merit.
- Spamming is used to trick search engines to rank some documents high.
- Some search engines use hyperlinks to rank documents (ex. Google) it is thus necessary to identify and discard nepolistic links.
- Recognizing Nepotistic Links on the Web (Davidson 2000).
- Davidson uses C4.5 classification algorithm on large number of page attributes, trained on manually labeled pages.

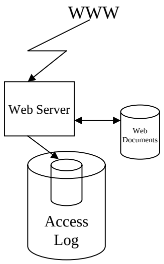
# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# Existing Web Log Analysis Tools

- There are many commercially available applications.
  - Many of them are slow and make assumptions to reduce the size of the log file to analyse.
- Frequently used, pre-defined reports:
  - Summary report of hits and bytes transferred
  - List of top requested URLs
  - List of top referrers
  - List of most common browsers
  - Hits per hour/day/week/month reports
  - Hits per Internet domain
  - Error report
  - Directory tree report, etc.
- Tools are limited in their performance, comprehensiveness, and depth of analysis.

# What Is Weblog Mining?



- Web Servers register a log entry for every single access they get.
- A huge number of accesses (hits) are registered and collected in an ever-growing web access log.

•Weblog mining:

- –Enhance web server and system performance
- –Improve web site navigation (i.e. improve design of sites & web-based applications)
- –Target customers for electronic commerce
- –Identify potential prime advertisement locations
- –Facilitates personalization (user profiling)
- –Intrusion and security issues detection

# Web Server Log File Entries

| IP address | User ID | Timestamp | Method | URL/Path | Status | Size | Referrer | Agent | Cookie |
|---|---|---|---|---|---|---|---|---|---|

```
dd23-125.compuserve.com - rhuia [01/Apr/1997:00:03:25 -0800] "GET /SFU/cgi-bin/VG/VG_dspmsg.cgi?ci=40154&mi=49 HTTP/1.0 "
    200 417
```

```
129.128.4.241 – [15/Aug/1999:10:45:32 – 0800] " GET /source/pages/chapter1.html " 200 618 /source/pages/index.html
    Mozilla/3.04(Win95)
```

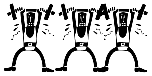
## Diversity of Web access log Mining

- Web access log provides rich information about Web dynamics
- Multidimensional Web access log analysis:
  - – disclose potential customers, users, markets, etc.
- Plan mining (mining general Web accessing regularities):
  - – Web linkage adjustment, performance improvements
- Web accessing association/sequential pattern analysis:
  - – Web cashing, prefetching, swapping
- Trend analysis:
  - – Dynamics of the Web: what has been changing?
- Customized to individual users

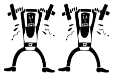
## More on Log Files

- Information NOT contained in the log files:
  - – use of browser functions, e.g. backtracking within-page navigation, e.g. scrolling up and down
  - – requests of pages stored in the cache
  - – requests of pages stored in the proxy server
  - – Etc.
- Special problems with dynamic pages:
  - – different user actions call same cgi script
  - – same user action at different times may call different cgi scripts
  - – one user using more than one browser at a time
  - – Etc.

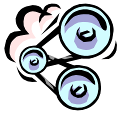
## Main Web Mining steps

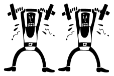


- Data Preparation

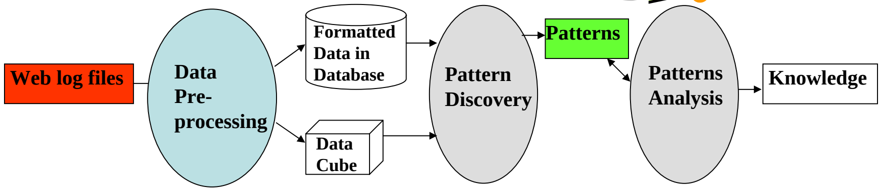


- Data Mining





- Pattern Analysis



## Data Pre-Processing

Problems:

- Identify types of pages: content page or navigation page.
- Identify visitor (user)
- Identify session, transaction, sequence, episode, mission, action
- Inferring cached pages

- Identifying visitors:
  - – Login / Cookies / Combination: IP address, agent, path followed
- Identification of session (division of clickstream)
  - – We do not know when a visitor leaves ➔ use a timeout (usually 30 minutes)
- Identification of user actions
  - Parameters and path analysis

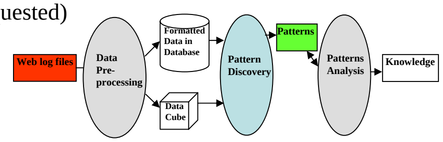
# Use of Content and Structure in Data Cleaning

- Structure:
    - The structure of a web site is needed to analyze session and transactions.
    - Hypertree of links between pages.
- Content
    - Content of web pages visited can give hints for data cleaning and selection.
    - Ex: grouping web transactions by terminal page content.
    - Content of web pages gives a clue on type of page: navigation or content.

# Data Mining: Pattern Discovery

Kinds of mining activities (drawn upon typical methods)

- Clustering (Cluster users based on browsing patterns - Cluster pages based on content – Cluster navigational behaviours based on browsing patterns similarity)
- Classification (classify users, pages, behaviours)
- Association mining (Find pages that are often viewed together)
- Sequential pattern analysis (Find frequent sequences of page visits)
- Prediction (Predict pages to be requested)



# What is the Goal?

- Personalization
- Adaptive sites
- Banner targeting
- User behaviour analysis
- Web site structure evaluation
- Improve server performance (caching, mirroring…)
- …

# Traversal Patterns

- The traversed paths are not explicit in web logs
- No reference to backward traversals or cache accesses
- Mining for path traversal patterns
- There are different types of patters:
    - Maximal Forward Sequence: No backward or reload operations: abcdedfg ➔ abcde + abcdfg
    - Duplicate page references of successive hits in the same session
    - contiguously linked pages

# Clustering

- Clustering

  Grouping together objects that have "similar" characteristics.
  - Clustering of transactions

    Grouping same behaviours regardless of visitor or content
  - Clustering of pages and paths

    Grouping same pages visited based on content and visits
  - Clustering of visitors

    Grouping of visitors with same behaviour

# Classification

- Classification of visitors
- Categorizing or profiling visitors by selecting features that best describe the properties of their behaviour.
- 25% of visitors who buy fiction books come from Ontario, are aged between 18 and 35, and visit after 5:00pm.
- The behaviour (ie. class) of a visitor may change in time.

# Association Mining

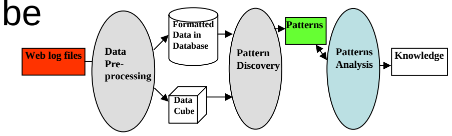
- Association of frequently visited pages
- What pages are frequently accessed together regardless of the ordering
- Pages visited in the same session constitute a transaction. Relating pages that are often referenced together regardless of the order in which they are accessed (may not be hyperlinked).
- Inter-session and intra-session associations.

# Sequential Pattern Analysis

- Sequential Patterns are inter-session ordered sequences of page visits. Pages in a session are time-ordered sets of episodes by the same visitor.
- Sequences of one user across transactions are considered at a time.
- (<A,B,C>,<A,D,C,E,F>, B, <A,B,C,E,F>)
- <A,B,C> <E,F> <A,*,F>,…

# Pattern Analysis

- Set of rules discovered can be very large
- Pattern analysis reduces the set of rules by filtering out uninteresting rules or directly pinpointing interesting rules.
  - SQL like analysis
  - OLAP from datacube
  - Visualization



# Discussion

- Analyzing the web access logs can help understand user behavior and web structure, thereby improving the design of web collections and web applications, targeting e-commerce potential customers, etc.
- Web access log entries do not collect enough information.
- Data cleaning and transformation is crucial and often requires site structure knowledge (Metadata).
- OLAP provides data views from different perspectives and at different conceptual levels.
- Web access Log Data Mining provides in depth reports like time series analysis, associations, classification, etc.

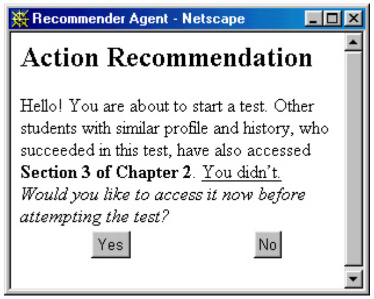
# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# Introduction to Recommendation

- Recommender Systems suggests products to buy.
  - Popularly being used in e-Commence to encourage customers to purchase more products.
  - *Amazon.com*$^{TM}$ (www.amazon.com) *CDNOW*$^{TM}$ (www.cdnow.com), etc.
- Recommender Systems suggest on-line Resources
  - There are too many resources. It is hard to find what we want when we want it.
  - Let users find web pages or resources interesting to them more easily.
- Recommender Systems suggest products close to the specified ones
  - Query relaxation when original query was not satisfied
  - K-nearest neighbours when answer is not enough

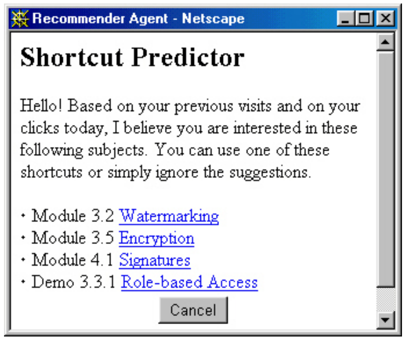
# Examples:
# Recommendation Based on Usage



Recommender Agent - Netscape

## Action Recommendation

Hello! You are about to start a test. Other students with similar profile and history, who succeeded in this test, have also accessed **Section 3 of Chapter 2**. You didn't. *Would you like to access it now before attempting the test?*

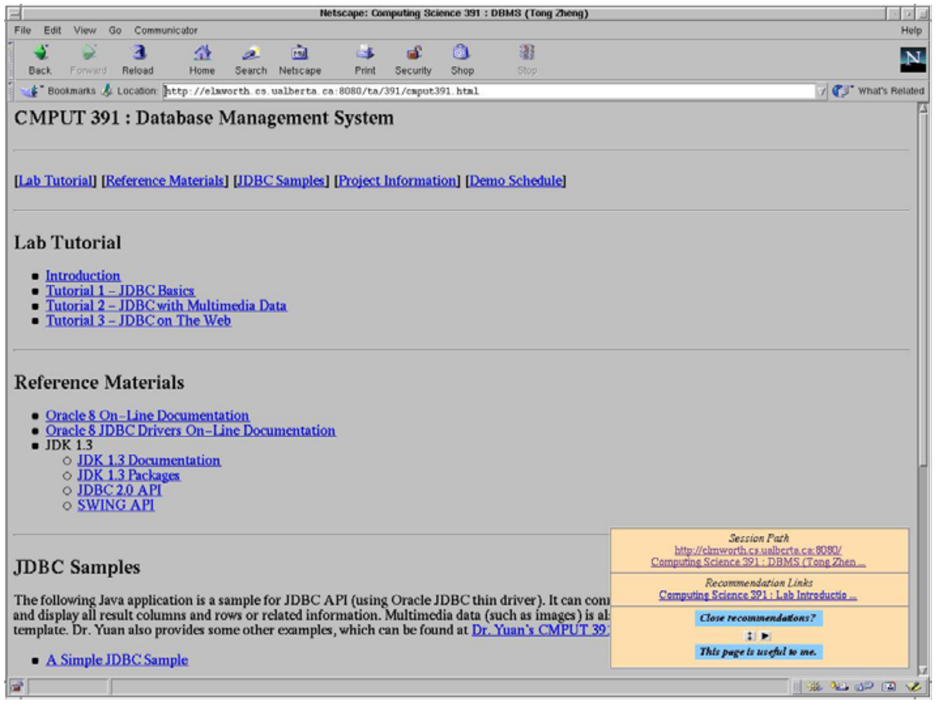


Recommender Agent - Netscape

## Shortcut Predictor

Hello! Based on your previous visits and on your clicks today, I believe you are interested in these following subjects. You can use one of these shortcuts or simply ignore the suggestions.

- Module 3.2 Watermarking
- Module 3.5 Encryption
- Module 4.1 Signatures
- Demo 3.3.1 Role-based Access



Netscape: Computing Science 391 : DBMS (Tong Zheng)

File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://elmworth.cs.ualberta.ca:8080/ta/391/cmput391.html What's Related

CMPUT 391 : Database Management System

[Lab Tutorial] [Reference Materials] [JDBC Samples] [Project Information] [Demo Schedule]

Lab Tutorial

- Introduction
- Tutorial 1 – JDBC Basics
- Tutorial 2 – JDBC with Multimedia Data
- Tutorial 3 – JDBC on The Web

Reference Materials

- Oracle 8 On–Line Documentation
- Oracle 8 JDBC Drivers On–Line Documentation
- JDK 1.3
  - JDK 1.3 Documentation
  - JDK 1.3 Packages
  - JDBC 2.0 API
  - SWING API

JDBC Samples

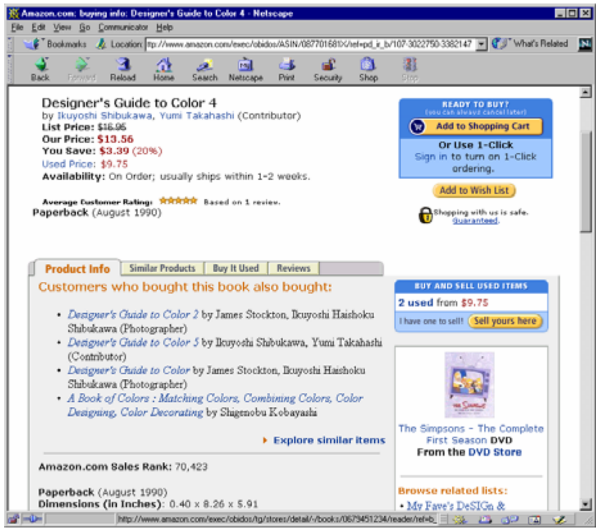
The following Java application is a sample for JDBC API (using Oracle JDBC thin driver). It can con and display all result columns and rows or related information. Multimedia data (such as images) is al template. Dr. Yuan also provides some other examples, which can be found at Dr. Yuan's CMPUT 39

- A Simple JDBC Sample

*Session Path*
http://elmworth.cs.ualberta.ca:8080/
Computing Science 391 : DBMS (Tong Zhen...

*Recommendation Links*
Computing Science 391 : Lab Introductio...

# Other Recommender Systems



Amazon.com: buying info: Designer's Guide to Color 4 - Netscape

File Edit View Go Communicator Help
Bookmarks Location: ttp://www.amazon.com/exec/obidos/ASIN/087701681X/ref=pd_ir_b/107-3022750-3382147 What's Related
Back Forward Reload Home Search Netscape Print Security Shop Stop

Designer's Guide to Color 4
by Ikuyoshi Shibukawa, Yumi Takahashi (Contributor)
**List Price:** ~~$16.95~~
**Our Price:** $13.56
**You Save:** $3.39 (20%)
Used Price: $9.75
**Availability:** On Order; usually ships within 1-2 weeks.

**Average Customer Rating:** ★★★★★ Based on 1 review.
**Paperback** (August 1990)

READY TO BUY?
(you can always cancel later)

Or Use 1-Click
Sign in to turn on 1-Click ordering.

Shopping with us is safe. Guaranteed.

Customers who bought this book also bought:

- *Designer's Guide to Color 2* by James Stockton, Ikuyoshi Haishoku Shibukawa (Photographer)
- *Designer's Guide to Color 5* by Ikuyoshi Shibukawa, Yumi Takahashi (Contributor)
- *Designer's Guide to Color* by James Stockton, Ikuyoshi Haishoku Shibukawa (Photographer)
- *A Book of Colors : Matching Colors, Combining Colors, Color Designing, Color Decorating* by Shigenobu Kobayashi

Explore similar items

**Amazon.com Sales Rank:** 70,423

**Paperback** (August 1990)
**Dimensions (in inches)**: 0.40 x 8.26 x 5.91

BUY AND SELL USED ITEMS
2 used from $9.75
I have one to sell!

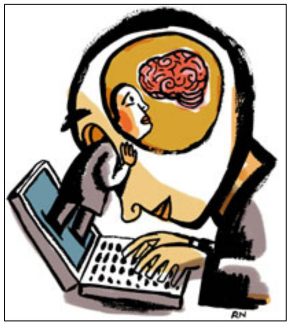
The Simpsons - The Complete First Season DVD
From the DVD Store

Browse related lists:
- My Fave's DeSIGn &

http://www.amazon.com/exec/obidos/tg/stores/detail/-/books/0679451234/reader/ref=b_

Amazon.com is a typical example but there are other recommender systems for books (ratingZone,…), for music CDs (CDNOW…), for movies (MovieCritic…) etc.

# I know what you'll read next summer (Amazon, Barnes&Noble)



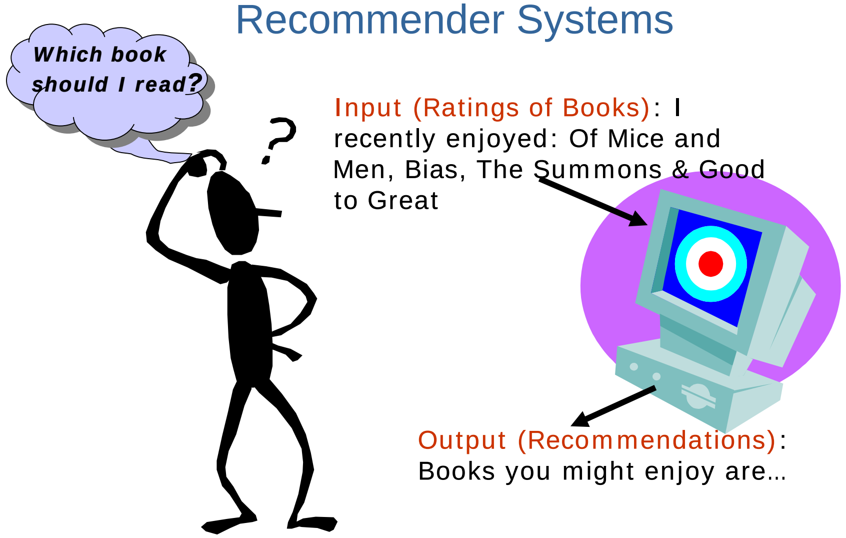
- what movies you should watch… (Reel, RatingZone, Amazon)
- what music you should listen to… (CDNow, Mubu, Gigabeat)
- what websites you should visit (Alexa)
- what jokes you will like (Jester)
- & who you should date (Yenta)

Source: Rashmi Sinha

# Collaborative Filtering: the Basic Idea

- *The basic idea of collaborative filtering is people recommending items to one another*.

# Basic Interaction Paradigm of Recommender Systems



Input (Ratings of Books): I recently enjoyed: Of Mice and Men, Bias, The Summons & Good to Great

Output (Recommendations): Books you might enjoy are…

*Popularity of Recommender Systems*

# All automated collaborative filtering algorithms use the following steps to make a recommendation

1. Construct a profile for a user: This profile normally consists of a user's rating of some items in the domain. The ratings are normally captured on some numerical scale.
2. Compare user's profile with profiles of other users: Compare this profile to all (or some subset of) the other users in the system and calculate a similarity between each pair of profiles compared. The actual algorithm used to determine similarity of users profiles may vary.
3. Construct the set of nearest neighbours for this user: These are the N most similar user profiles for a particular user. These form this users nearest neighbours. Weight each profile in the nearest neighbour set by the degree of similarity to the user profile.
4. Use the Nearest Neighbour set to make recommendation: Use the nearest neighbour set of weighted user profiles to calculate a predicted rating for a new item in the domain for a user. If the predicted rating exceeds a given threshold value, recommend this item to the user.

At the heart of Recommender Systems are Collaborative Filtering Algorithms that rely on correlation between individuals

| Ratings of Books | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Jane | 5 | 3 | 3 | 4 | 2 | 1 | | |
| Alexander | 3 | 4 | 2 | 3 | 4 | 5 | 1 | 3 |
| Amelia | 4 | 3 | 1 | 2 | 4 | 2 | 4 | 1 |
| Duncan | 4 | 2 | 1 | 3 | 4 | 1 | 5 | 2 |

- Jane & Duncan: correlation = .52
- Jane & Alexander: correlation = -.67
- Jane & Amelia: correlation = .23

Recommendations for Jane: Book 7

# Recommender with Association Rules

- What if we have no ratings?
- Based on transactions user$_i$ bought $<i_1, i_2, \ldots>$
- If User$_x$ buys $i_a$ and $<i_a, i_b>$ is frequent itemset in the purchase logs and user x never bought $i_b$ then suggest $i_b$
- Association rule based recommenders need to be trained. ➔ training set ➔ updated often

# Issues with Previous Approaches

- Most consider exclusively web usage data.

There are other channels to exploit

- Transactions assume information needs are fulfilled sequentially.

Not true in reality

- Newly added pages are never recommended.

The new pages may contain the needed data

- Buried and difficult to reach pages are never recommended.

Defeats the purpose of recommending

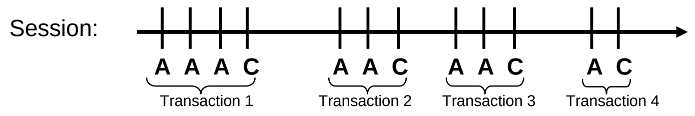
- Recommended lists are long and unordered.

Carefully ranking recommendation is important

# Transaction Identification

- Two standard approaches
  - *Reference Length* Approach
  - *Maximal Forward Reference* Approach
- Same underlying assumption:

A visitor may have different information needs during a visit, but all the information needs must be fulfilled **in the sequence**.



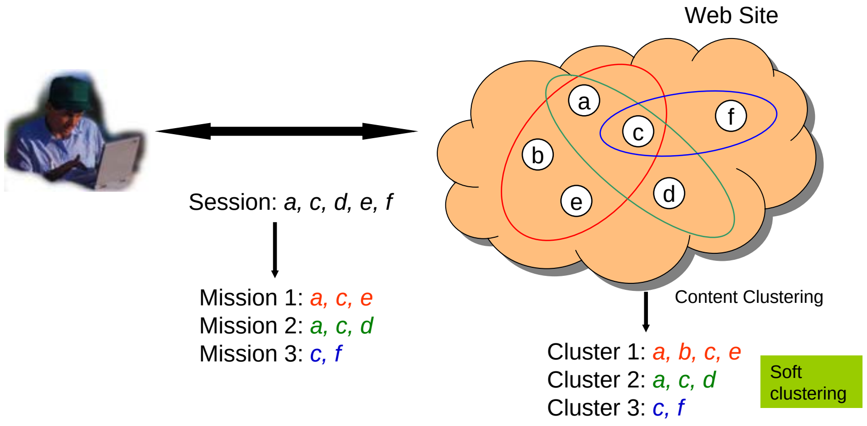
# Mission vs. Transaction

- More often than not, we open several browsers to surf a site, looking for different information at the same time. Moreover, we may sometimes interrupt our current goal and start another in the middle, and then return to the original one later on.
- In these scenarios, the transaction identification approaches mistakenly group pages to fulfill users' different information needs into one transaction.
- Because the transaction is the base of any data mining algorithm for pattern discovery, this misclassification would obviously compromise the effect of the data mining task, or even cause it to fail.

# Mission Identification

- Mission Identification – an improved transaction identification approach.
  - Acknowledging that users may visit a website with multiple goals, i.e., different information needs.
  - Making no assumption on the sequence in which these needs are fulfilled.
- *Mission*: a sub-session related to one of these information needs
  - Allowing overlap between missions
  - Representing a concurrent search in the site

# How to Identify Missions



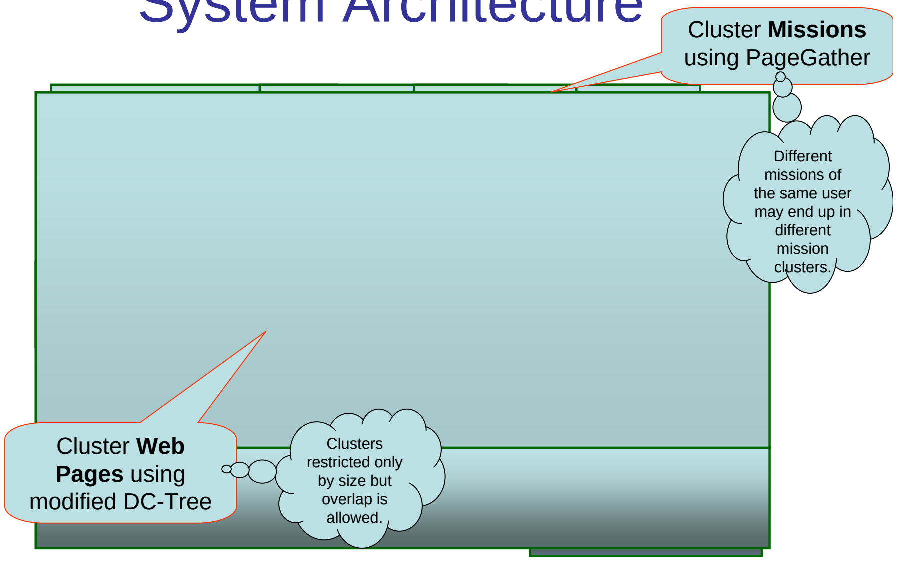
Session: *a, c, d, e, f*

Mission 1: *a, c, e*
Mission 2: *a, c, d*
Mission 3: *c, f*

Cluster 1: *a, b, c, e*
Cluster 2: *a, c, d*
Cluster 3: *c, f*

Soft clustering

# System Architecture



Cluster **Missions** using PageGather

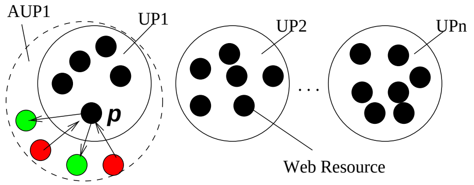
Different missions of the same user may end up in different mission clusters.

Cluster **Web Pages** using modified DC-Tree

Clusters restricted only by size but overlap is allowed.

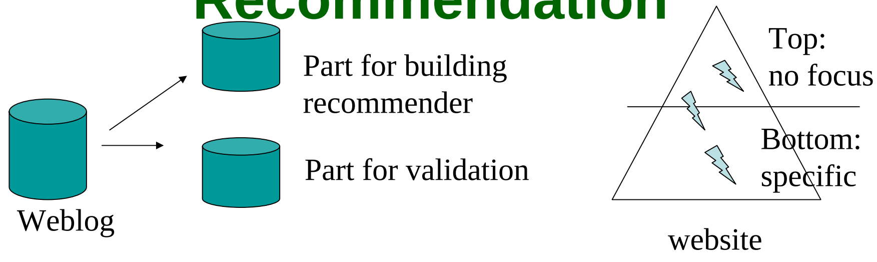
# User Profile Improvement

- Providing an opportunity for these rarely visited or newly added pages to be recommended.
- User profile improvement is done in a two-step process.
  - Augmentation



  - Pruning

# Validating page Recommendation



Weblog

Part for building recommender

Part for validation

Top: no focus

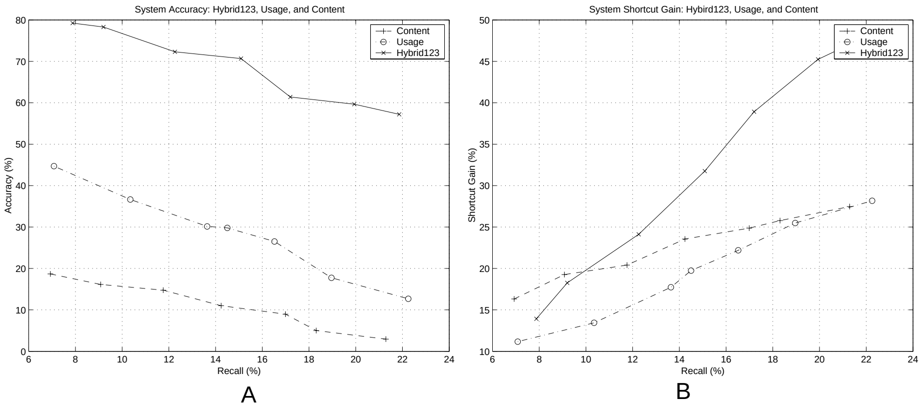
Bottom: specific

website

Measures: (1) precision / recall; (2) length of short cut

However: recommending a short shortcut is useless if page link-out is small. It is still useful if page link-out is large.

# One Experimental result Example



# Tightening or Relaxing Queries

- In many on-line applications such as hotel reservation, flight scheduling, or product selection by description, a user is provided with the means to specify their needs by way of describing constraints and submitting queries.
- What happens when there is no answer to the specified query?
- What happens if there are too many answers to the specified query?
- An intelligent system can recommend to relax the original query (or tighten it).

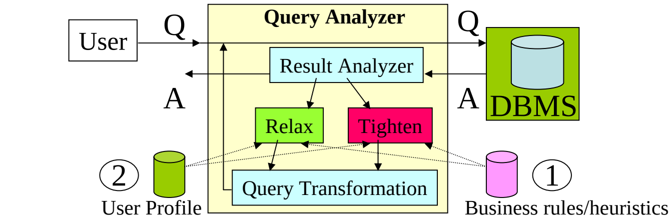
# Examples of Relaxed Queries

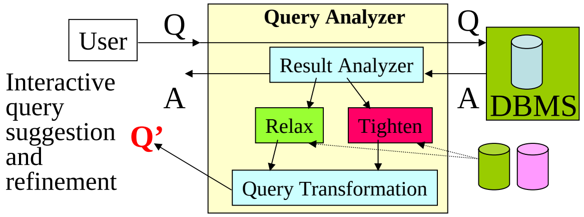
- Original Query: List all hotels with a price range [$75..$100] with swimming pool and Internet access
- There are hotels with swimming pool and Internet access but the prices are above $100 ➔ **relax price range constraint to [$75..$150]**
- There are hotels between $75 and $100 but without swimming pool
  ➔ **relax swimming pool constraint.**

# Query modification

- Query relaxation or tightening can be done based on:
  - Heuristics
  - Business rules
  - Interactively with the user
  - User profile with weights on constraints (preferences)

# Architecture scenarios





# What convinces a user to sample the recommendation

- Judging recommendations:
  - What is a good recommendation from the user's perspective?
- Trust in a Recommender System:
  - What factors lead to trust in a system?
- System Transparency:
  - Do users need to know why an item was recommended?

Source: Rashmi Sinha

# Design Recommendations: Justification

- Justify your Recommendations
  - Adequate Item Information: Providing enough detail about item for user to make choice
  - System Transparency: Generate (at least some) recommendations which are clearly linked to the rated items
  - Explanation: Provide an Explanation, why the item was recommended.
  - Community Ratings: Provide link to ratings / reviews by other users. If possible, present numerical summary of ratings.

Source: Rashmi Sinha

## Design Recommendations: Accuracy vs. Less Input

- Don’t sacrifice accuracy for the sake of generating quick recommendations. Users don’t mind rating more items to receive quality recommendations.
  - A possible way to achieve this: have multilevel recommendations. Users can initially use the system by providing one rating, and are offered subsequent opportunities to refine recommendation
  - One needs a happy medium between too little input (leading to low accuracy) and too much input (leading to user impatience)

Source: Rashmi Sinha

## Design Recommendations: New Unexpected Items

- Users like Rec. Systems as they provide information about new, unexpected items.
  - List of recommended items should include new items which the user might not find out in any other way.
  - List could also include some unexpected items (e.g., from other topics / genres) which the user might not have thought of themselves.

Source: Rashmi Sinha

## Design Recommendations: Trust Generating Items

- Users (especially first time users) need to develop trust in the system.
  - Trust in system is enhanced by the presence of items that the user has already enjoyed.
  - Generating some very popular (which have probably been experienced previously) in the initial recommendation set might be one way to achieve this.

Source: Rashmi Sinha

## Design Recommendations: Mix of Items

- Systems need to provide a mix of different kinds of items to cater to different users:
  - Trust Generating Items: A few very popular ones, which the system has high confidence in
  - Unexpected Items: Some unexpected items, whose purpose is to allow users to broaden horizons.
  - Transparent Items: At least some items for which the user can see the clear link between the items he /she rated and the recommendation.

Question: Should these be presented as a sorted list / unsorted list/ different categories of recommendations?

Source: Rashmi Sinha

# Outline

- Introduction to Web Mining
  - What are the incentives of web mining?
  - What is the taxonomy of web mining?
- Web Content Mining: Getting the Essence From Within Web Pages.
- Web Structure Mining: Are Hyperlinks Information?
- Web Usage Mining: Exploiting Web Access Logs.
- Recommender Systems
- Warehousing the Web (if time permits)

# Warehousing a Meta-Web: An MLDB Approach

- *Meta-Web:* A structure which summarizes the contents, structure, linkage, and access of the Web and which evolves with the Web
- Layer$_0$: the Web itself
- Layer$_1$: the lowest layer of the Meta-Web
  - an entry: a Web page summary, including class, time, URL, contents, keywords, popularity, weight, links, etc.
- Layer$_2$ and up: summary/classification/clustering in various ways and distributed for various applications
- Meta-Web can be warehoused and incrementally updated
- Querying and mining can be performed on or assisted by meta-Web (a multi-layer digital library catalogue, yellow page).

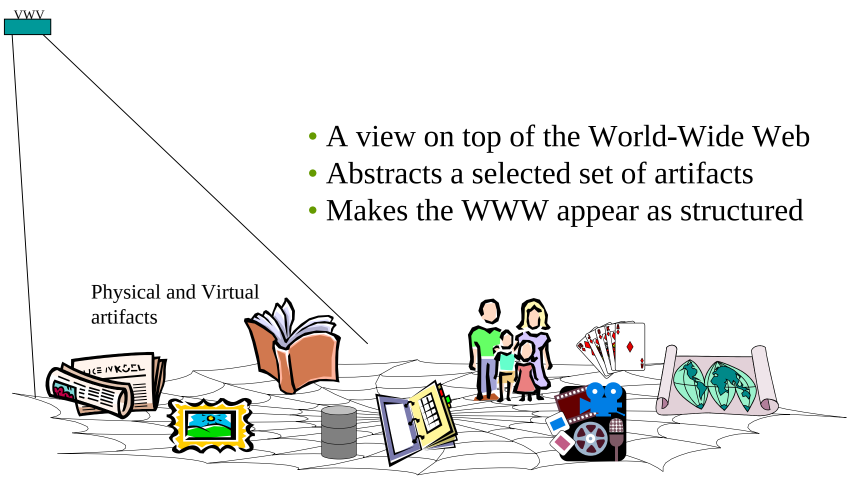
# Construction of Multi-Layer Meta-Web

- XML: facilitates structured and meta-information extraction
- Hidden Web: DB schema “extraction” + other meta info
- Automatic classification of Web documents:
  - based on Yahoo!, etc. as training set + keyword-based correlation/classification analysis (IR/AI assistance)
- Automatic ranking of important Web pages
  - authoritative site recognition and clustering Web pages
- Generalization-based multi-layer meta-Web construction
  - With the assistance of clustering and classification analysis

# Use of Multi-Layer Meta Web

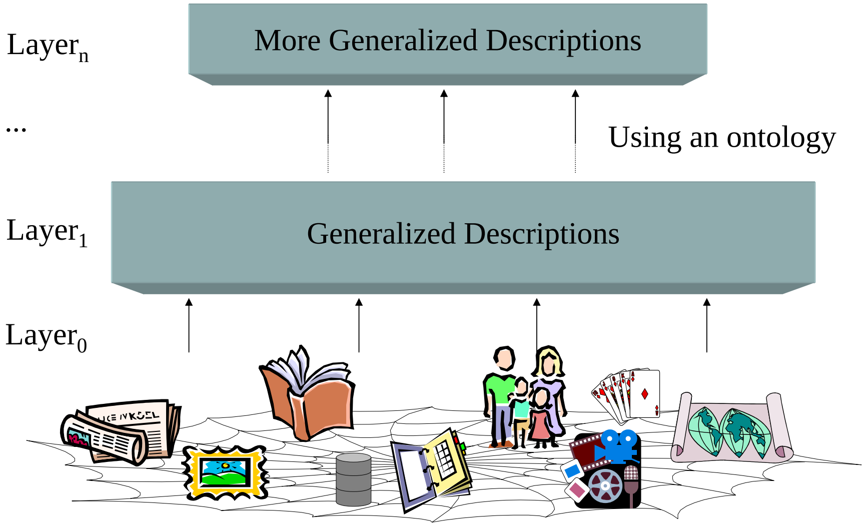
- Benefits of Multi-Layer Meta-Web:
  - Multi-dimensional Web info summary analysis
  - Approximate and intelligent query answering
  - Web high-level query answering (WebSQL, WebML)
  - Web content and structure mining
  - Observing the dynamics/evolution of the Web
- Is it realistic to construct such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

# Virtual Web View



- A view on top of the World-Wide Web
- Abstracts a selected set of artifacts
- Makes the WWW appear as structured

# Multiple Layered Database Architecture



Layer$_n$ More Generalized Descriptions

...

Using an ontology

Layer$_1$ Generalized Descriptions

Layer$_0$

# Observation

| key | Price | broker | age | exterior | roof | arft | mbr | br1 | br2 | lr | dr | kt | atr | pk | add | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12345 | $95,000 | Sussex | 22 | Stucco | Gravel | 911 | 13x9 | 13x8 | 0 | 14x12 | 12x9 | 9x7 | Y | N | ... | ... |
| 12346 | $110,000 | Sutton | 16 | Mixed | Tar/Gr | 939 | 13x10 | 13x9 | 6x5 | 11x13 | 12x11 | 9x5 | Y | Y | ... | ... |
| 12347 | $114,000 | Rennie | 10 | Wood | Tar/Gr | 933 | 11x13 | 10x10 | 0 | 12x13 | 12x9 | 10x7 | N | Y | ... | ... |
| 12348 | $119,900 | Rennie | 10 | Wood | Tar/Gr | 974 | 11x13 | 10x10 | 0 | 13x12 | 12x10 | 9x9 | N | Y | ... | ... |
| 12349 | $116,900 | P.George | 12 | Stucco | Tar/Gr | 901 | 12x12 | 11x10 | 8x3 | 15x12 | 11x9 | 9x7 | Y | Y | ... | ... |
| 12350 | $99,000 | P.George | 17 | Stucco | Tar/Gr | 879 | 13x10 | 12x9 | 0 | 13x11 | 10x10 | 6x11 | Y | N | ... | ... |
| 12351 | $119,500 | Sutton | 14 | Mixed | Tar/Gr | 815 | 14x11 | 14x9 | 0 | 13x12 | 7x9 | 9x7 | N | Y | ... | ... |
| 12352 | $115,000 | Homelife | 6 | Mixed | Tar/Gr | 911 | 14x11 | 14x9 | 0 | 14x12 | 13x9 | 7x7 | Y | Y | ... | ... |
| 12353 | $116,900 | Rennie | 10 | Wood/atc | Tar/Gr | 964 | 11x13 | 14x9 | 0 | 14x11 | 12x9 | 9x7 | N | Y | ... | ... |
| 12354 | $110,500 | Rennie | 16 | Mixed | Tar/Gr | 990 | 13x11 | 13x8 | 0 | 12x13 | 10x10 | 17x5 | N | Y | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

| Area | Class | Type | Price | Size | Age | Count |
|---|---|---|---|---|---|---|
| Richmond | Aprt | 1 bdr | $75,000-$85,000 | 500-700 | 10-12 | 23 |
| Richmond | Aprt | 1 bdr | $85,000-$95,000 | 701-899 | 5-10 | 18 |
| Richmond | Aprt | 2 bdr | $95,000-$110,000 | 900-955 | 10-12 | 12 |
| ... | ... | ... | ... | ... | ... | ... |

Transformed and generalized database

•User may be satisfied with the abstract data associated with statistics
•Higher layers are smaller. Retrieval is faster
•Higher layers may assist the user to browse the database content progressively

# Multiple Layered Database Strength

- Distinguishes and separates meta-data from data
- Semantically indexes objects served on the Internet
- Discovers resources without overloading servers and flooding the network
- Facilitates progressive information browsing
- Discovers implicit knowledge (data mining)

# Multiple Layered Database First Layers

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_in, 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, ...)

# Examples

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| URL | title | set of authors | pub_data | format | language | size | set of keywords | set of media | set of links-out | set of links-in | access-freq | timestamp |

Documents

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| URL | format | size | height | width | Start_frame | duration | set of keywords | set of parent pages | visual feature vectors | access-freq | timestamp |

Images and Videos

# Multiple Layered Database Higher Layers

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_in, 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_in, links_out)

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

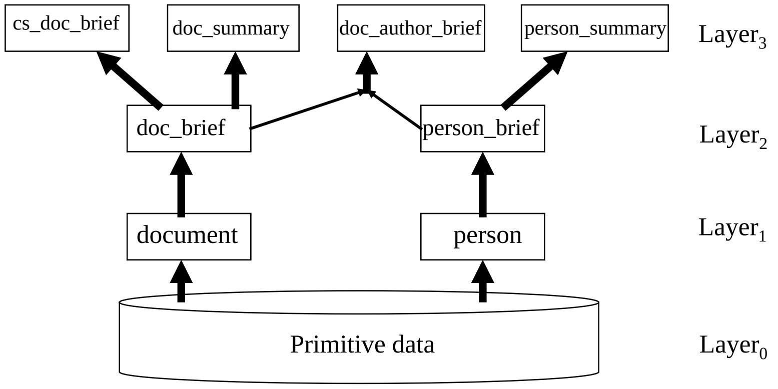
•**doc_author_brief**(file_addr, authors, affiliation, title, publication, pub_date, category_description, keywords, num_pages, format, size_doc, links_in, links_out)

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

# Multiple Layered Database doc_summary example

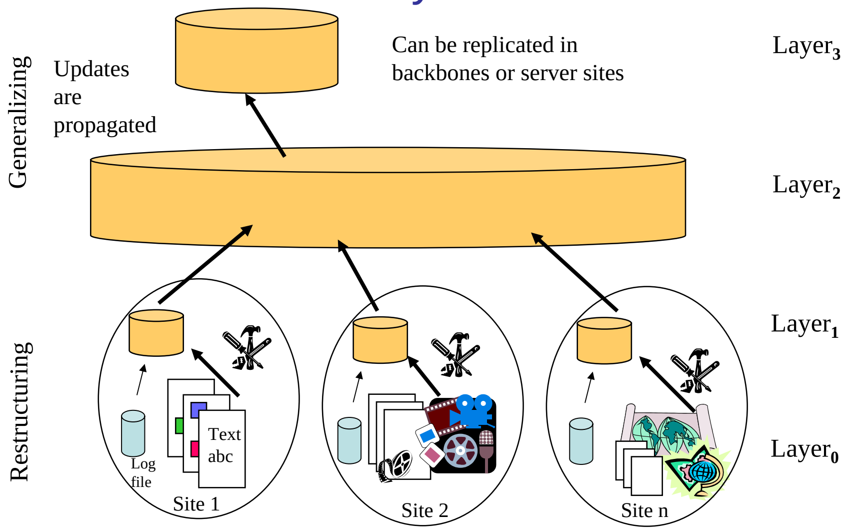
| **affiliation** | **field** | **pub_year** | **count** | **first_author_list** | **file_addr_list** | … |
|---|---|---|---|---|---|---|
| Simon Fraser Univ. | Database Systems | 1994 | 15 | Han, Kameda, Luk, ... | … | … |
| Univ. of Colorado | Global Network Systems | 1993 | 10 | Danzig, Hall, ... | … | … |
| MIT | Electromagnetic Field | 1993 | 53 | Bernstein, Phillips, ... | … | … |
| … | … | … | … | … | … | … |

# Construction of the Stratum

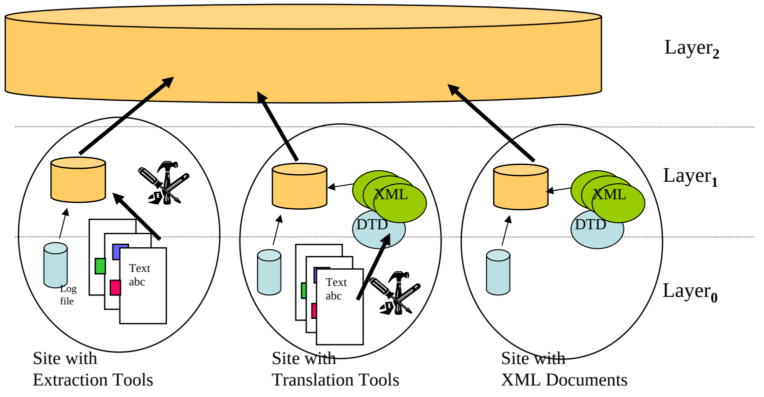

cs_doc_brief
doc_summary
doc_author_brief
person_summary
Layer$_3$
doc_brief
person_brief
Layer$_2$
document
person
Layer$_1$
Primitive data
Layer$_0$

- The multi-layer structure should be constructed based on the study of frequent accessing patterns
- It is possible to construct high layered databases for special interested users ex: *computer science documents, ACM papers, etc.*

# Construction and Maintenance of Layer-1


Generalizing
Restructuring
Updates are propagated
Can be replicated in backbones or server sites
Layer$_3$
Layer$_2$
Layer$_1$
Layer$_0$
Log file
Text abc
Site 1
Site 2
Site n

# Options for the Layer-1 Construction


Layer$_2$
Layer$_1$
Layer$_0$
Log file
Text abc
XML
DTD
Site with Extraction Tools
Site with Translation Tools
Site with XML Documents

# The Need for Metadata

Can XML help to extract the right needed descriptors?

<NAME> eXtensible Markup Language</NAME>
<RECOM>World-Wide Web Consortium</RECOM>
<SINCE>1998</SINCE>
<VERSION>1.0</VERSION>
<DESC>Meta language that facilitates more meaningful and precise declarations of document content</DESC>
<HOW>Definition of new tags and DTDs</HOW>

**Dublin Core Element Set**

TITLE
CREATOR
SUBJECT
DESCRIPTION
PUBLISHER
CONTRIBUTOR
DATE
TYPE
FORMAT
IDENTIFIER
SOURCE
LANGUAGE
RELATION
COVERAGE
RIGHTS

XML can help solve heterogeneity for vertical applications, but the freedom to define tags can make horizontal applications on the Web more heterogeneous.

# Concept Hierarchy

| | | |
|---|---|---|
| All | **contains**: | Science, Art, … |
| Science | **contains**: | Computing Science, Physics,Mathematics,… |
| Computing Science | **contains**: | Theory, Database Systems, Programming Languages,… |
| Computing Science | **alias**: | *Information Science, Computer Science, Computer Technologies, …* |
| Theory | **contains**: | Parallel Computing, Complexity, Computational Geometry, … |
| Parallel Computing | **contains**: | Processors Organization, Interconnection Networks, RAM, … |
| Processor Organization | **contains**: | Hypercube, Pyramid, Grid, Spanner, X-tree,… |
| Interconnection Networks | **contains**: | Gossiping, Broadcasting, … |
| Interconnection Networks | **alias**: | *Intercommunication Networks, …* |
| Gossiping | **alias**: | *Gossip Problem, Telephone Problem, Rumour, …* |
| Database Systems | **contains**: | Data Mining, Transaction Management, Query Processing, … |
| Database Systems | **alias**: | *Database Technologies, Data Management, …* |
| Data Mining | **alias**: | *Knowledge Discovery, Data Dredging, Data Archaeology, …* |
| Transaction Management | **contains**: | Concurrency Control, Recovery, ... |
| Computational Geometry | **contains**: | Geometry Searching, Convex Hull, Geometry of Rectangles, Visibility, ... |

# WebML

Since concepts in a MLDB are generalized at different layers, search conditions may not exactly match the concept level of the inquired layers. Can be too general or too specific.

Introduction of new operators

Primitives for additional relational operations

| WebML primitive | Operation | Name of the operation |
|---|---|---|
| covers | $\supset$ | Coverage |
| covered-by | $\subset$ | Subsumption |
| like | $\approx$ | Synonymy |
| close-to | $\sim$ | Approximation |

User-defined primitives can also be added

# Top Level Syntax

*<WebML>* ::= *<Mine Header>* **from** relation_list
  [**related-to** name_list] [**in** location_list]
  **where** where_clause
  [**order by** attributes_name_list]
  [**rank by** {inward | outward | access}]

*<Mine Header>* ::= {{**select** | **list**} {attribute_name_list | *}
  | *<Describe Header>* | *<Classify Header>*}

*<Describe Header>* ::= **mine description**
  **in-relevance-to** {attribute_name_list | *}

*<Classify Header>* ::= **mine classification**
  **according-to** attribute_name_list
  **in-relevance-to** {attribute_name_list | *}

## WebML Example: Resource Discovery

Locate the documents related to “computer science” written by “Ted Thomas” and about “data mining”.

```
select     *
from       document
related-to “computer science”
where      “Ted Thomas” in authors and one of keywords like “data mining”
```

Discovering Resources

Returns a list of URL addresses together with important attributes of the documents.

## WebML Example: Resource Discovery

Locate the documents about “data mining” linked from Osmar’s web page and rank them by importance.

```
select   *
from     document
where    exact “http://www.cs.sfu.ca/~zaiane” in links_in
         and one of keywords like “data mining”
rank by inward, access
```

Discovering Resources

Returns a list of URL addresses together with important attributes of the documents.

## WebML Example: Resource Discovery

Locate the documents about “Intelligent Agents” published at SFU and that link to Osmar’s web pages.

```
select   *
from     document
in “http://www.sfu.ca”
related-to “computer science”
where    “http://www.cs.sfu.ca/~zaiane” in links_out
         and one of keywords like “Agents”
```

No “**exact**” ⇒ prefix substring

Discovering Resources

Returns a list of URL addresses together with important attributes of the documents.

## WebML Example: Resource Discovery

List the documents published in North America and related to “data mining”.

```
list       *
from       document
in         “North_America”
related-to “computer science”
where      one of keywords covered_by “data mining”
```

Discovering Resources

Returns a list of documents at a high conceptual level and allows browsing of the list with slicing and drilling through to the appropriate physical documents.

## WebML Example: Knowledge Discovery

Inquire about European universities *productive* in publishing on-line *popular* documents related to database systems since 1990.

```
select   affiliation
from     document
in       “Europe”
where    affiliation belong_to “university” and
         one of keywords covered-by “database systems”
         and publication_year > 1990 and count = “high”
         and f(links_in) = “high”
```

Weight (heuristic formula)

Discovering Knowledge

Does not return a list of document references, but rather a list of universities.

## WebML Example: Knowledge Discovery

Describe the general characteristics in relevance to authors' affiliations, publications, etc. for those documents which are popular on the Internet (in terms of access) and are about "data mining".

**mine description**
**in-relevance-to** author.affiliation, publication, pub_date
**from** document related-to Computing Science
**where** **one of** keywords **like** "database systems"
and access_frequency = "high"

Discovering Knowledge

Retrieves information according to the 'where clause', then generalizes and collects it in a data cube for interactive OLAP-like operations.

## WebML Example: Knowledge Discovery

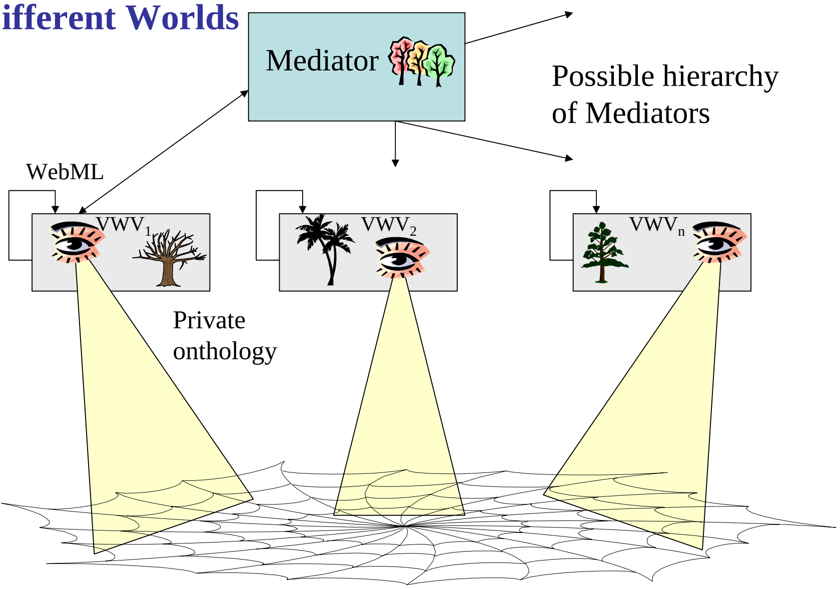
Classify, according to update time and access popularity, the documents published on-line in sites in the Canadian and commercial Internet domain after 1993 and about IR from the Internet.

**mine classification**
**according-to** timestamp, access_frequency
**in-relevance-to** *
**from** document **in** Canada, Commercial
**where** **one of** keywords **covered-by** "Information Retrieval"
**and one of** keywords like "Internet"
**and** publication_year > 1993

Discovering Knowledge

Generates a classification tree where documents are classified by access frequency and modification date.

## Different Worlds



Possible hierarchy of Mediators

Private onthology