Web Mining

Tutorial June 21, 2002

Pisa KDD Laboratory:

CNR-CNUCE **Dipartimento di Informatica**
Università di Pisa

Table of Content

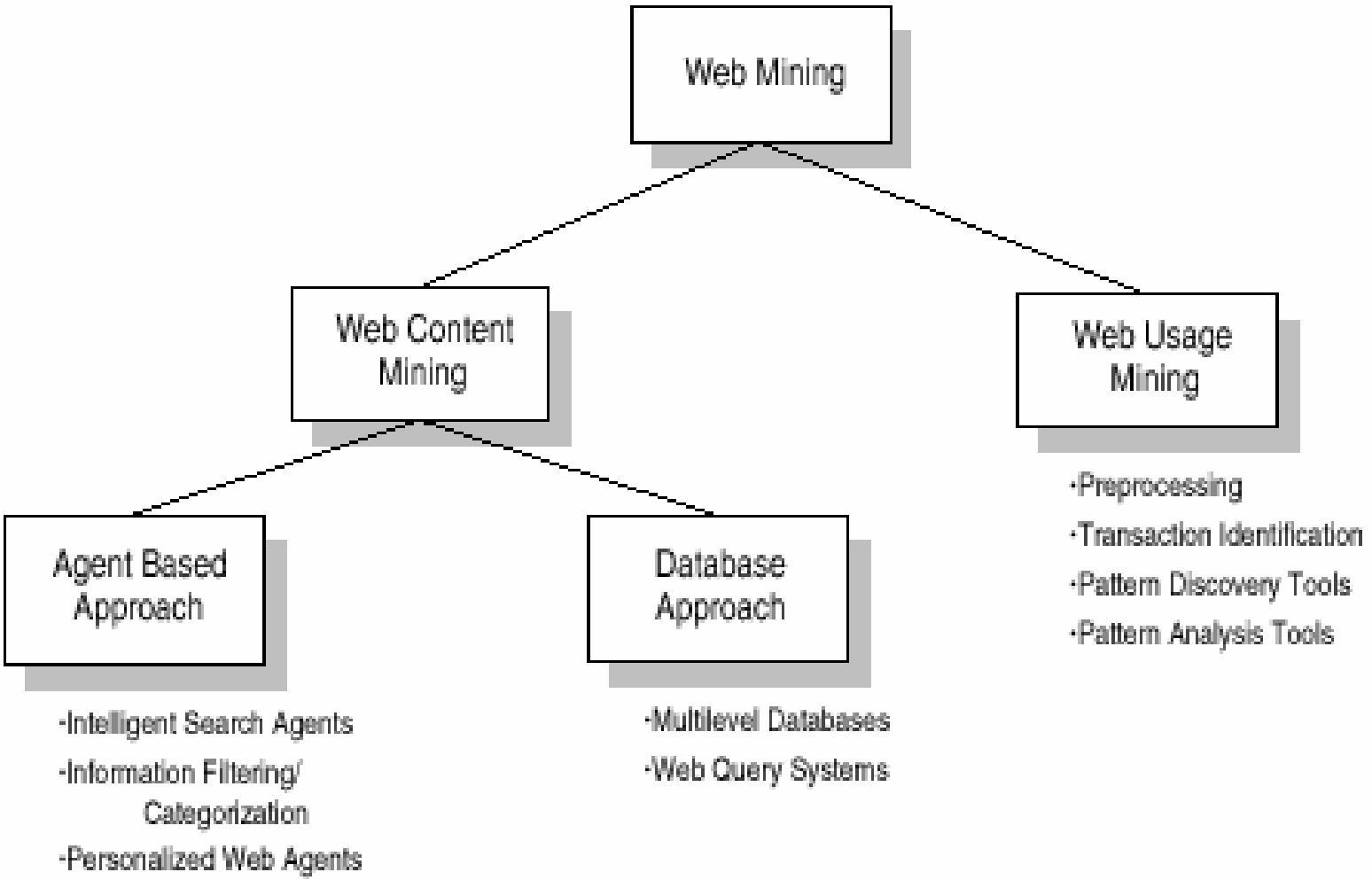
- Introduction
- The KDD cycle for the web
  - Preprocessing
  - Data mining tasks for the web
- Applications
  - Web caching
  - Personalisation
- A sample application step by step
- Research directions

Data mining and the web

Web Mining: *the discovery and analysis of useful information from the World Wide Web.*

- **web content mining**
  - aims at constructing higher-level models of organization of semi-structured data contained in web sites;
- **web structure mining** (also part of content mining)
  - aims at constructing models of web site structure in terms of page interconnections.
- **web usage mining** (or web log mining)
  - aims at discovering usage patterns from web logs of browsers, web servers and proxy servers;

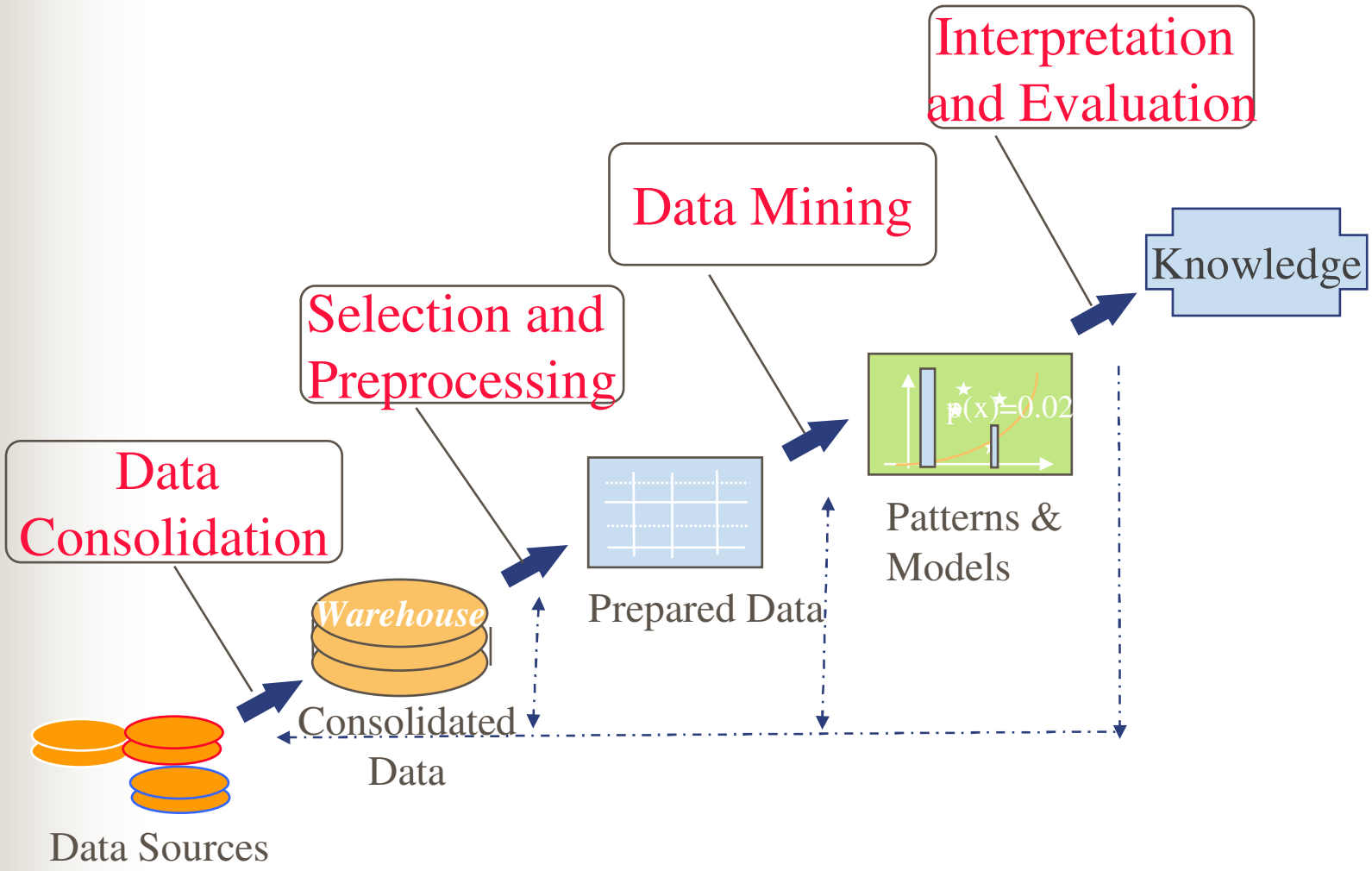
Web mining taxonomy



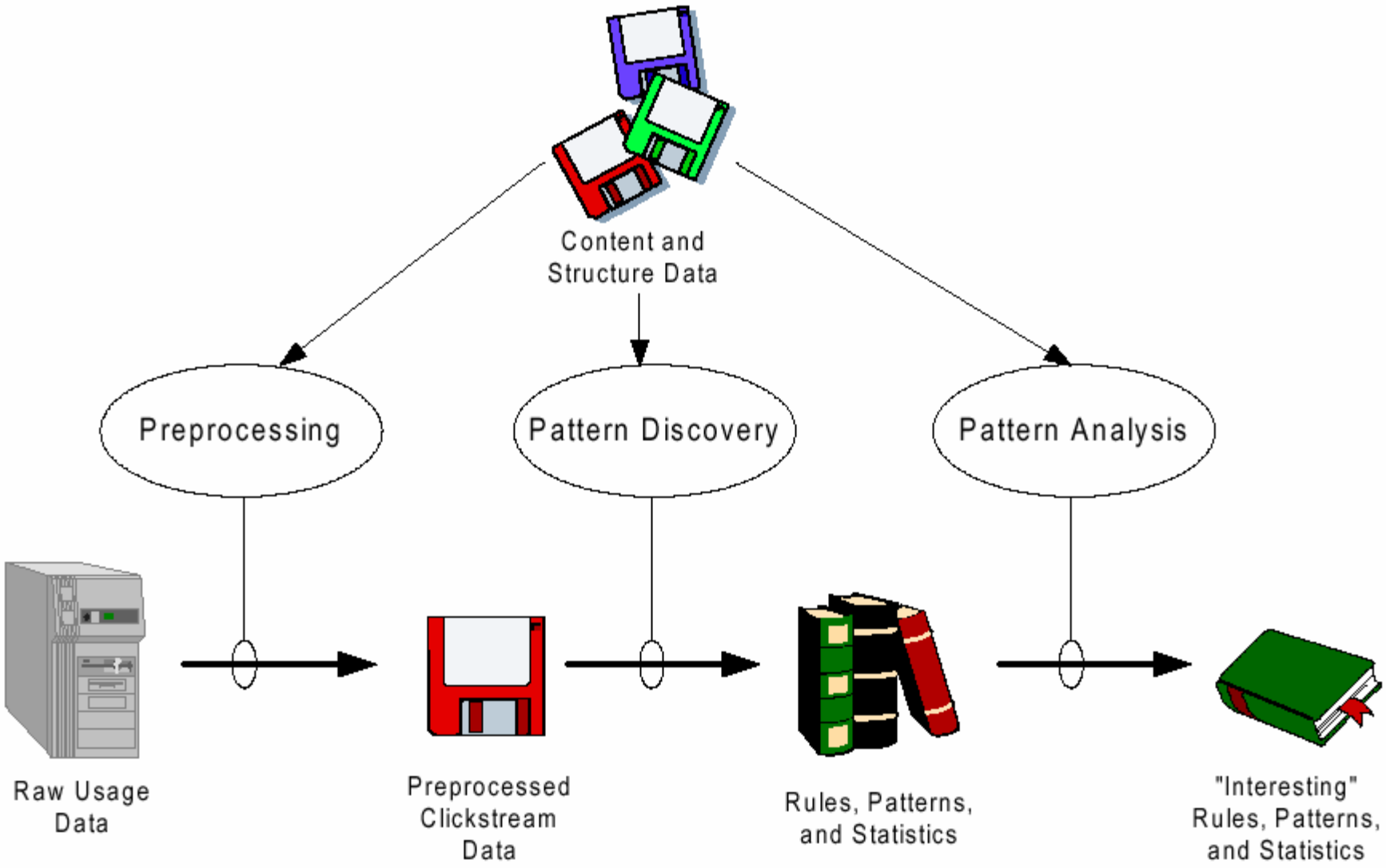
Web usage mining

- Focus on the analysis and discovery of usage patterns and models on the basis of historical data: the web log files which record previous access to web objects.
- Web log files provide a large source of data for DM, as web servers and proxy servers store order of millions logs every day.
- Various application domains:
  - web site redesign and restructuring,
  - self-adapting web sites,
  - recommendation systems in e-commerce

The KDD process



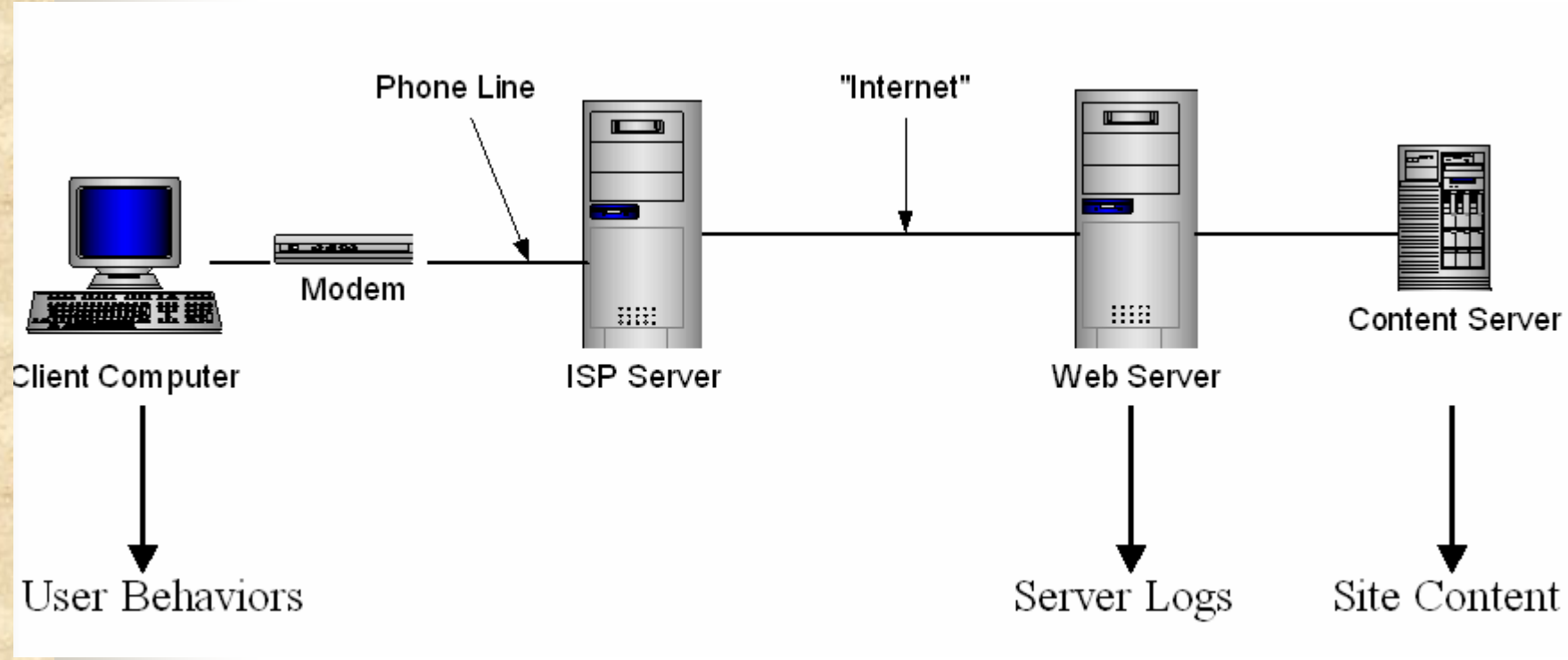
The Web Usage Mining Process



Web Mining:

Preprocessing

Client/Server Interaction



Web data sources

- Client
  - Registration data
  - Browser
  - Agents (e.g., applets, jscript, ecc.)
- Web Server
  - HTTP Clickstream
  - Content/Application Server (e.g., Vignette, Broadvision)
  - Packet Sniffer (e.g., Accrue)
  - Other (database log, file system log, ecc.)
- ISP o Proxy
  - HTTP Clickstream
  - Packet Sniffer

HTTP Clickstream: ECLF

- Extended Common Log Format (a row per HTTP request)

213.213.31.41 - - [15/Apr/2000:04:00:04 +0200] "GET images/h/h_home.gif HTTP/1.1" 200 1267 "http://www.di.unipi.it/" "MSIE 4.01; Windows 98“

- Host: **213.213.31.41** *(or reverse address lookup)*
- Ident e Authuser: **- -** *(IDs for IDENTD and HTTP/SSL)*
- Time: **[15/Apr/2000:04:00:04 +0200]** *(end-of-answer time)*
- HTTP Request: **"GET images/h/h_home.gif HTTP/1.1"**
- Status: **200** *(=OK, 3xx=redirection, 4xx=client error, etc.)*
- Bytes: **1267** *(number of transmitted to the Client)*
- Referrer: **"http://www.di.unipi.it/"**
- User agent: **"MSIE 4.01; Windows 98"**

HTTP Clickstream

Sometimes other data are available:

- Server Computer ID, Time-to-Serve, Content-Type, Expires, Last-Modified, No-cache, ...
- Cookies
  - Mechanism to handle the status of a session/to identify users
  - Cookie = string exchanged between client and server
    - Persistent / active only for actual session
    - Stored on the client side
    - Can be refused/cancelled by user

Terminology

- **Page file**: file accessed by a single HTTP request
- **Page view**: set of *page files* which compose a single page displayed in a browser.
  - Its page files can be retrieved from different servers
- **(local) User session:** set of page views retrieved by a user from the Web Server to achieve some goal.
  - Available at the Web Server level
  - Partially hidden by Browser and Proxy caching
- **Global user session**: set of page views retrieved by a user from the whole Web to achieve some goal.
  - Available only at the Client level
  - Partially available at the Proxy/ISP level because of Browser caching

Drinking from the Fire hose

- HTTP Clickstream is a “poor” data source
- Log preprocessing needed before data mining
- Preprocessing issues:
  1. Request Preprocessing
  2. User Identification
  3. Page Identification
  4. Page Content Identification
  5. Computing the Dwell time
  6. Session Identification
  7. Path Completion

1. Request Preprocessing

- URL are not unique IDs for web resources
  - E.g.: http://www.DI.unipi.IT = http://www.di.unipi.it, etc.
- Field extraction
  - Host, Path, Filename, File Extension, Query String
- Request selection
  - Filter out visits of robots
    - Several Heuristics: Known names/substrings, repeated accesses to the same page, too quick clickstream, data mining approaches, etc.
  - Analisys-dependent selection
    - E.g.: Only GET requests, or successful (code 200) requests, etc.

2. User Identification

- **IP** (+ User Agent)
  - Always possible, but not reliable (IP recycling by ISP + Proxy masking)
- **Cookies**
  - Can be refused by users, but are (quite) reliable and can be shared among different sites.
- **Embedded SessionID** (volatile cookies)
  - For a single visit: http://www.di.unipi.it/?session=123456
- **Client-side tracking** (modified browser)
  - Reliable, but invasive (privacy)
- **Authentication**
  - Maximal reliability, but invasive (privacy)
- **Others**:
  - Path analysis, ID Ethernet, ID CPU

3. Pageview Identification

Associates each request with its corresponding page view.

Referrer-based

- Page view = A + all requests having referrer A, till next request for A
  - Problems: multiframe pages (have several referrers); client and proxy caching can mask “next request for A”; external referrers
- **Analysis of HTML content**
  - Page View = A + all page files embedded in A (e.g.: <img>)
    - Problems: dynamic pages
- **Application/content server**
  - Access log at the application/content server
    - Very general but expensive

4. Page Content Identification

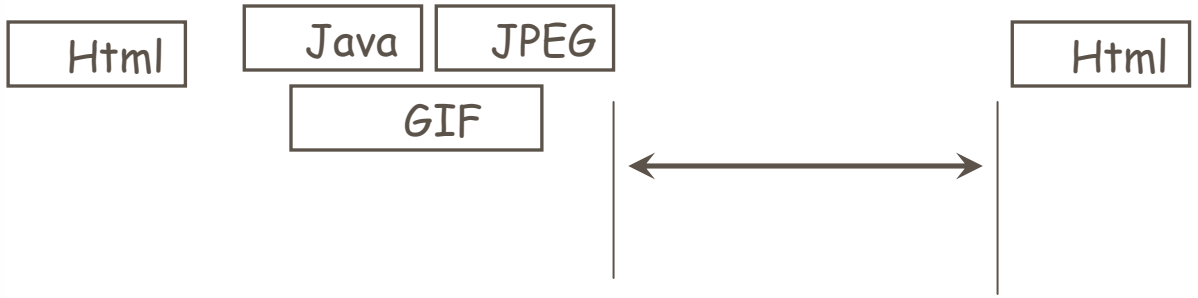
Associates each page view with a classification of its content

- E.g.: events in e-commerce = view, click-through, buy, bid, shoopping cart change

- **Explicit identification**
  - Made by web administrators
    - Expensive, not general
- **Analysis of HTML content**
  - Automatic natural language analysis techniques
    - Limits with dynamic pages, input pages
- **Application/content server log**
  - Content log at the application/content server level
    - Very general but expensive

5. Computing Dwell time

- **Dwel time:** time between the end of a page view loading and the beginning of next request from the same user



- Computing the PageView loading time:
  - Approximation based on PageView size
  - Experimental values
  - Special cases: streamining video
- Dwell time=0 ➔ loading aborted ➔ unsatified user

6. Session Identification

- A set of accesses from a user can be associated with different goals
- **User session** = *set of page views/files requested by a single user to the Web Server for a specific "goal".*
- **Objective**: given a sequence of page files/views

$$\langle p_1, \ldots, p_n \rangle$$

requested by a user, discover subsequences

$\langle p_1, \ldots, p_{n1} \rangle \quad \langle p_{n1+1}, \ldots, p_{n2} \rangle \quad \ldots \quad \langle p_{n(k-1)+1}, \ldots, p_{nk} \rangle$

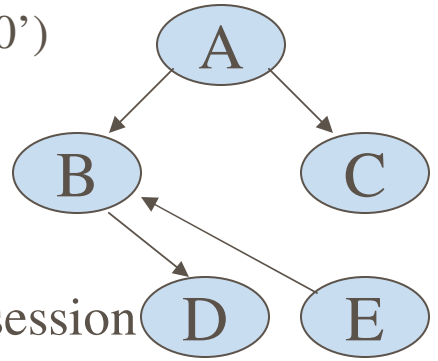
corresponding to different "goals".

6. Session Identification

Heuristics

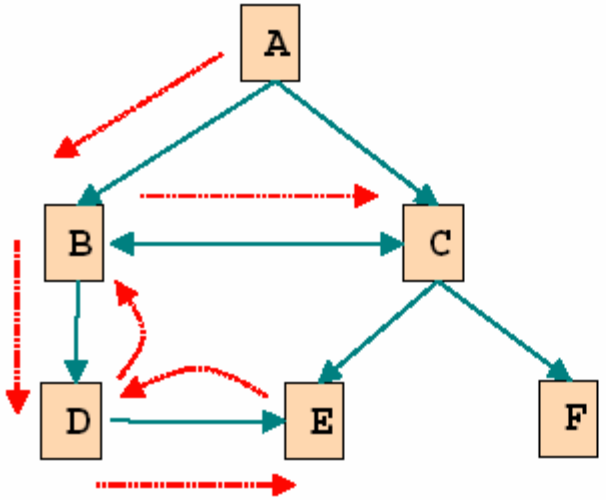
- **Time-oriented**
  - Subsequences must have limited time extensions:
    - Time between p_1 and p_{n1} must be ≤ t' (typical value: 30')
    - Time between p_i and p_{i+1} must be ≤t"
  - Navigational vs. Content pages
- **Navigation-oriented**
  - Linkage: "jump" to a non-reachable page → end of session
    - E.g.: Links=A→B, A→C, B→D, C→E

      The <A B D C E B> sequence is divided in <A B D C> and <E B>
    - Topology of the site is requested
  - Maximal Forward Reference: backtrack → end of session
    - E.g.: <A B D C B> yields: <A B D>, <A C> and <A B>



7. Path Completion

- **Objective**: deducing requests not logged because were satisfied by caches between user and web server.



User's navigation path:
A => B => D => E
=> D => B => C

URL	Referrer
A	--
B	A
D	B
E	D
C	B

- Missing path: from E to C
  - Shorthest solution: E → D → B → C
  - It is not unique!

Web Mining

Data mining tasks

Data Mining

- General objective: Search for common patterns in a data set.
- NOT a replacement for:
  - Session Analysis
  - Static Aggregation and Statistics (Reports)
  - OLAP

Common Data Mining Tasks

- Frequent Itemsets
- Association Rules
- Clustering
- Classification
- Sequential Patterns

Frequent Itemsets

- Find groups of items that appear together in a "transaction" with some frequency.
- Similar to statistical correlation.
- Standard measure used is "support", which gives the percentage of transactions that an itemset appears in.
- For example, "Items `A` and `B` appear together in *s%* of the transactions."

Frequent Itemset

Example

- *The "Home Page" and "Shopping Cart Page" are accessed together in 20% of the sessions.*
- *The "Donkey Kong Video Game" and "Stainless Steel Flatware Set" product pages are accessed together in 1.2% of the sessions.*

NB: transactions are defined as sets of visited web pages (e.g.: user sessions).

Association Rules

- Similar to Frequent itemsets, except a directionality is introduced to each rule.
- Instead of just *"A and B appear together frequently (with support s%)"*, we get:

  "When A appears, B also appears x% of the time"

  "When B appears, A also appears y% of the time"
- The *x* and *y* values are referred to as Confidence.

Association Rules

Example

- *When the "Shopping Cart Page" is accessed in a session, the "Home Page" is also accessed 100% of the time.*
- *When the "Shipping conditions" page is accessed in a session, a purchase is performed 75% of the time.*

Clustering

- Form groups of similar items.
- Often difficult to define similarity.
- Two types of clustering
  - Number of groups are predetermined.
  - Number of groups are automatically determined by the algorithm.

Clustering

Page Clustering: Example

Usage-based frequent itemsets are clustered:

- *"Donkey Kong Video Game", "Pokemon Video Game", and "Video Game Caddy" product pages are related.*

Graph partitioning of web site structure:

- *"DM homepage", "DM bibliography", "DM teaching" link only (and are linked only by) each other.*

Based on content dissimilarity between pages:

- *"Donkey Kong Video Game", "Pocket Donkey Kong", and "Donkey Kong 2" (on different sites) have similar contents.*

Clustering

User session Clustering: Example

User Transaction Clustering:

- *Transactions=sets of URLs (pages in a session) as binary vectors*
- *K-means on transactions*

Association rule hyper-graph partitioning:

- *Arcs A→B represent associations between URLs A and B*
- *Clustering by partitioning the graph*

Classification

- Discover rules that will predict what group an item belongs to.
- Similar to clustering, with the concept that every item belongs to a group or class.
- Usually requires “training”, where a classifier is “shown” examples of different groups in order to learn how to classify new items as they arrive.

Classification

Example

Classify pages on content

- *“Donkey Kong Video Game”, “Pokemon Video Game”, and “Video Game Caddy” product pages contain words “game”, “console”, … => they are all part of the Video Games product group.*

Classify sessions/users

- *The user who visits both the "Delivery options" and "How to pay" pages, falls in the "Probable customers" category.*

Sequential Patterns

- Sequential patterns add an extra dimension to frequent itemsets and association rules - time.
- Items can appear before, after, or at the same time as each other.

Sequential Patterns

Example

Time-forward associations:

- *50% of users who visited the "Video Game Caddy" page, later visited also the "Donkey Kong Video Game" page. This occurs in 1% of the sessions.*

Time-backward associations:

- *30% of clients who visited* **/products/software/** *had done a search in* **Yahoo** *using the keyword "software" before their visit.*

Sequences:

- *In 2% of sessions the user visited "Institute homepage" then "Contact info" then "Tom's personal page".*

Data Mining: Pro and cons

- Advantages:
  - Discovery complex patterns that were not obvious through other tools.
  - Reduce vast amounts of data to a small number of rules or patterns.
- Disadvantages:
  - Requires the most resources.
  - No “out of box” solutions.
  - Discovered patterns are often obvious and already known.

Filtering interesting patterns

Means:

- web structure information

Hypothesis:

- Domain knowledge can be derived from content and structure of a site, to define "expected patterns"

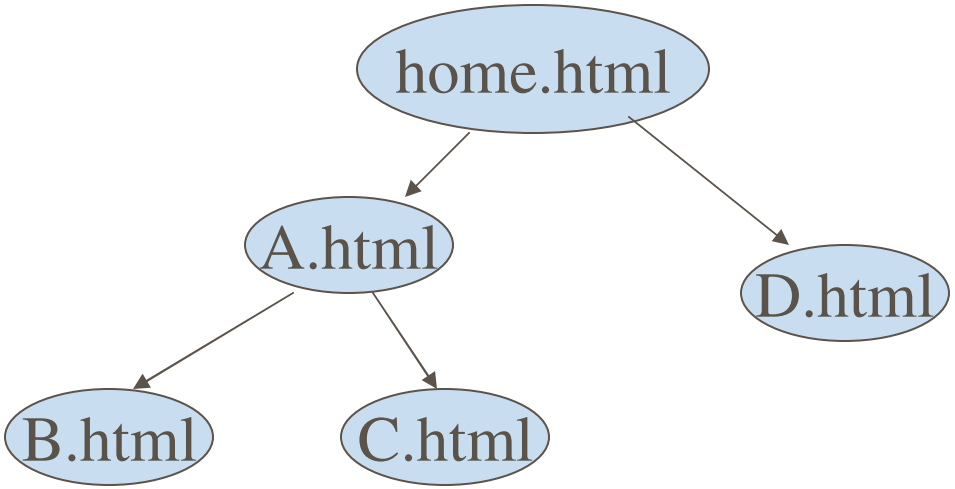
"Subjective interestingness":

- Measures deviation from expected patterns

Filtering interesting patterns

Expected patterns: Example

Site structure:



Rule **B.html → A.html** is expected (thus not interesting).

Rule **D.html → C.html** is unexpected (thus interesting).

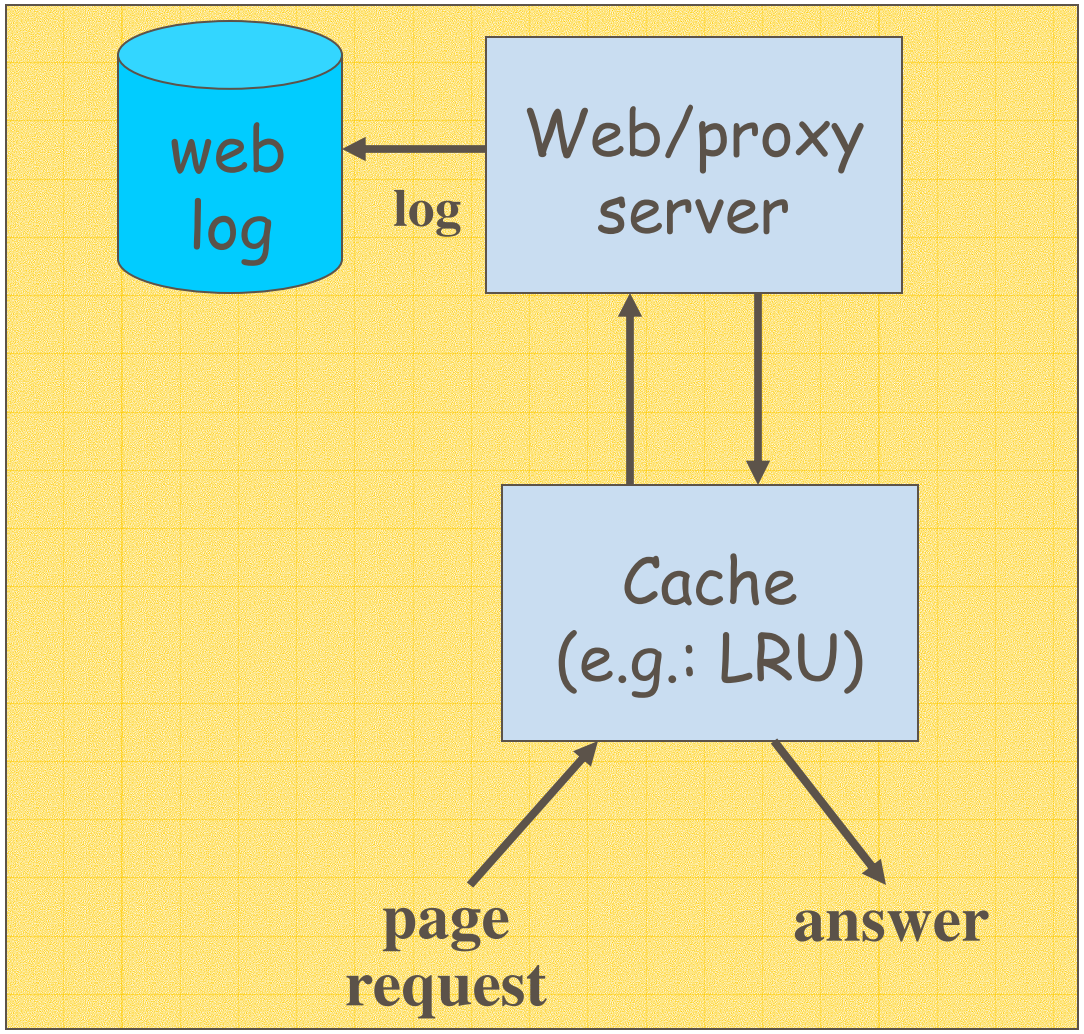
Applications

1. Web mining for Web caching

Web caching

- **Web caching is the temporary storage of web objects for later retrieval**
- **It can be performed at:**
  - client level
  - proxy level
  - server level
- **Caching may**
  - **reduce bandwidth**
  - **reduce server load**
  - **reduce latency**
  - **improve reliability**

Web caching: basic schema



Objectives

- Intelligent web caching:

  → *extend the (LRU) policy of web/proxy servers by making it sensible to web access models extracted from history log data using data mining techniques*

- Design of an intelligent web caching system
  - *model extraction*
  - *definition of a cache replacement policy that exploit extracted data mining models*

Data mining techniques adopted

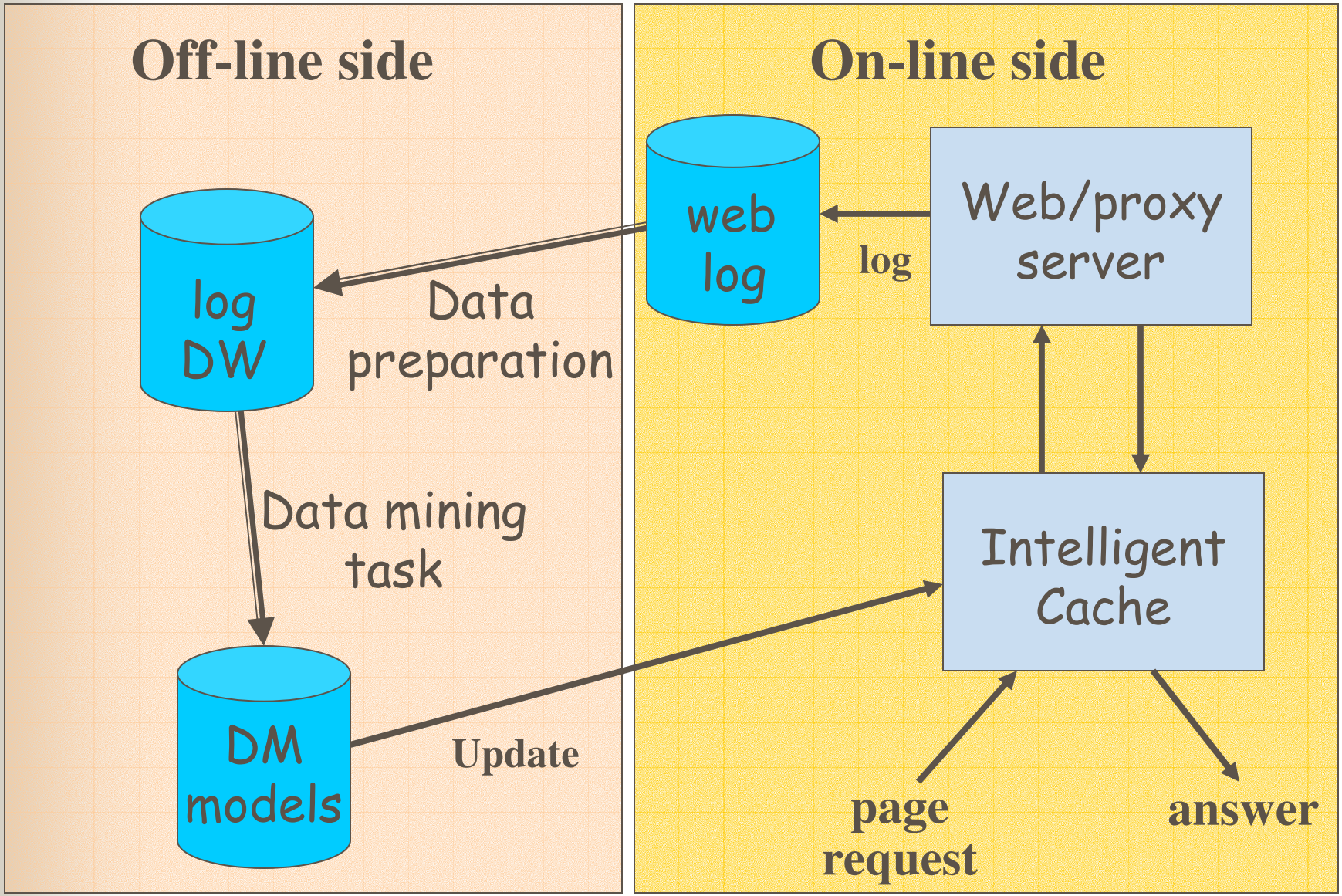
- Association rules
  - we extract from the web logs rules of the form A => B, where A and B are web documents
  - rule means that when the web document A is requested, then B is also likely to be requested within the same user session.
  - in the cache replacement algorithm: if A is requested, and therefore kept in the cache if present according to the LRU policy, then also B is treated analogously.

Data mining techniques adopted

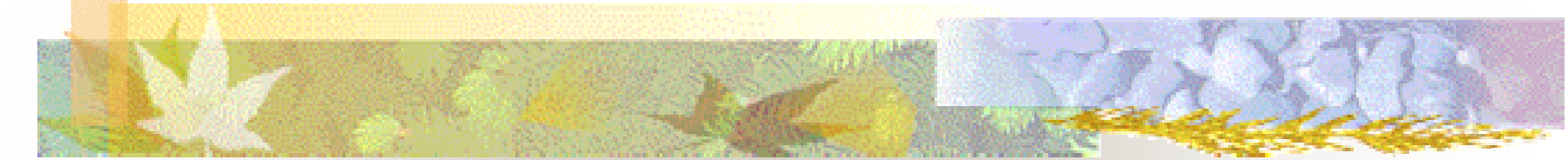
- Classification
  - we develop a decision tree:

    A model capable to predict, given a request of a web document A, the time of the next future request of A given other properties of A itself.
  - the prediction is based on the historical data contained in the web logs.
  - the prediction is used to assign a weight to A in the cache.

Intelligent web caching: Architecture



Applications



2. Web Mining for e-commerce: Personalization

Web Personalization

Three aspects of a Web site affect its utility in providing the intended service to its users:

- Content
- Layout of individual pages
- Structure of the entire Web site itself

Personalization/Recommendation

- The Problem
  - dynamically serve customized content (pages, products, recommendations, etc.) to users based on their profiles, preferences, or expected interests
- Personalization v. Customization
  - Customization: user controls and customizes the site or the product based on his/her preferences
    - usually manual, but sometimes semi-automatic based on a given user profile
  - Personalization: done automatically based on the user's actions, the user's profile, and (possibly) the profiles of others with "similar" profiles

Customization Example

my.yahoo.com

Personalization Example

amazon.com

(Traditional) Personalization Methods

- Currently, the most used technique for web personalization is *collaborative filtering*.

  E.g.: k-Nearest-Neighbor approach:
  - Each visitor is mapped to the *k* most similar past users (similar=same ratings to items, same page accesses, etc.)
  - A set of items is proposed to the visitor, obtained from the analysis of her/his *neighborhood*'s past activity
- Limitations of this method:
  - not scalable to large number of items (slow on-line kNN)
  - does not integrate additional site information such as content/navigational pages

Web Mining for Personalization

- Web mining approach: dividing the process
  1. (slow) offline pattern discovery
  2. (fast) online application of discovered patterns
- It provides the tools to analyze Web log data in a user-centric manner such as segmentation, profiling, and clickstream discovery.
- Data mining results to create decision rules for customizing Web site content based on an individual user's behavior.

Association-based Personalization

Basic Idea

- Match left-hand side of rules with the active user session and recommend items in the rule's consequent
- Ordering of accessed pages is not taken into account
- Good recommendation *accuracy*, but the main problem is *coverage*
- Tradeoff: Coverage vs. Computational cost:
  - high support thresholds lead to low coverage and may eliminate important, but infrequent items from consideration
  - low support thresholds result in very large model sizes and computationally expensive pattern discovery phase

Association-based Personalization

The approach of Mobasher et al.

- Avoid offline generation of all association rules; generate recommendations directly from itemsets
  - discovered frequent itemsets of are stored into an “itemset graph” (an extension of lexicographic tree structure of Agrawal, et al 1999)
  - recommendation generation can be done in constant time by doing a directed search to a limited depth
- Frequent itemsets are matched against a user's active session *S* by performing a search of graph to depth |S|
- A recommendation *r* is an item at level |S+1| whose recommendation score is the confidence of rule *S* ==> *r*

Sequence-based Personalization

Basic Idea

- Take the ordering of accessed items into account
- Two basic approaches:
  - use contiguous sequences (e.g., Web navigational patterns)
  - use general sequential patterns
- Contiguous sequential patterns are often modelled as Markov chains:
  - Usually applied to prefetching
  - In context of recommendations, they can achieve higher accuracy than other methods, but may be difficult to obtain reasonable coverage

Sequence-based Personalization

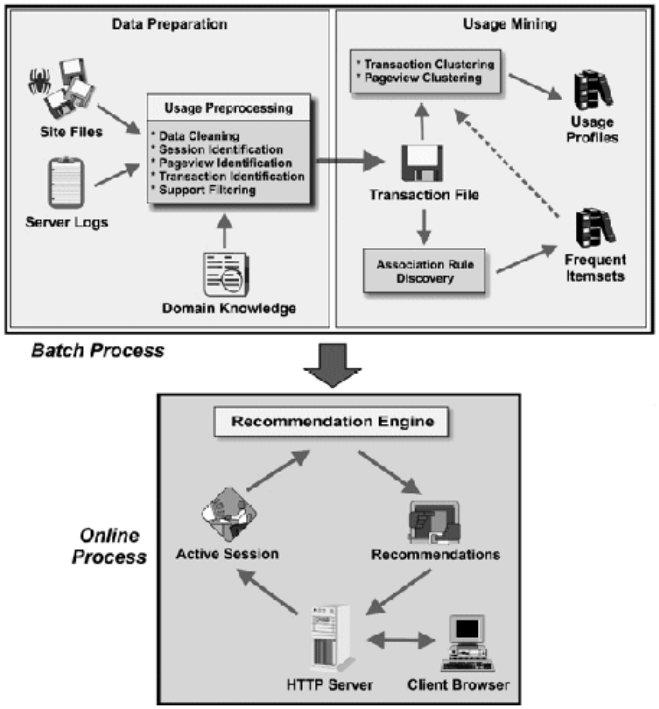
The approach of Gaul and Schmidt-Thieme

- Recommendations are based on frequent patterns of past behaviour
- A recommender is a predictor for a class of events (access to pages, form submissions, etc.)
- A *navigation history* is a set, a sequence or a more complex structure of events
- A collection of recommenders provide suggestions and a combination/selection is proposed to the user

Usage profiles for Personalization

Mobasher, Dai, Luo, Nakagawa

- Clustering to identify transaction clusters.
- New technique to identify user profiles from transaction clusters
- The general architecture presented has two components:



← (offline) Web usage analysis

← (online) Recommendation

Other Personalization Tasks

Spiliopoulou et al.: Improving Sites

- Effectiveness of a Web site in providing users with the content they need in the most optimized manner is the key to retaining them.
- Web Usage Miner: a navigation pattern Q.L.

```
SELECT  t       FROM NODE AS x y z
TEMPLATE    x*y*z   AS   t
WHERE x.support>=20  AND  (y.support/x.support) >= 0.5
        AND (z.support/y.support) >= 0.15
```

  - *Conversion* rates over different kinds of patterns are used to understand the site usage.

→ Suggestions to improve site content/structure

Other Personalization Tasks

Perkowitz & Etzioni: Adaptive Web Sites/1

The **IndexFinder** consists of three phases:

- *Log processing*: Establishment of sessions as sets of page requests
- *Cluster mining*: Grouping of co-occuring non-linked pages with help of the site graph
- *Conceptual clustering*:
  - The representative concept of each cluster is identified.
  - Cluster members not adhering to this concept are removed from the cluster.
  - Pages adhering to this concept and not appearing in the cluster are attached to the cluster.

Other Personalization Tasks

Perkowitz & Etzioni: Adaptive Web Sites/2

- For each cluster, the **IndexFinder** presents to the Web designer:
  - An index page with links to all pages of a cluster
- The Web designer decides:
  - whether the new page should indeed be established
  - what its label should be
  - where it should be located in the site

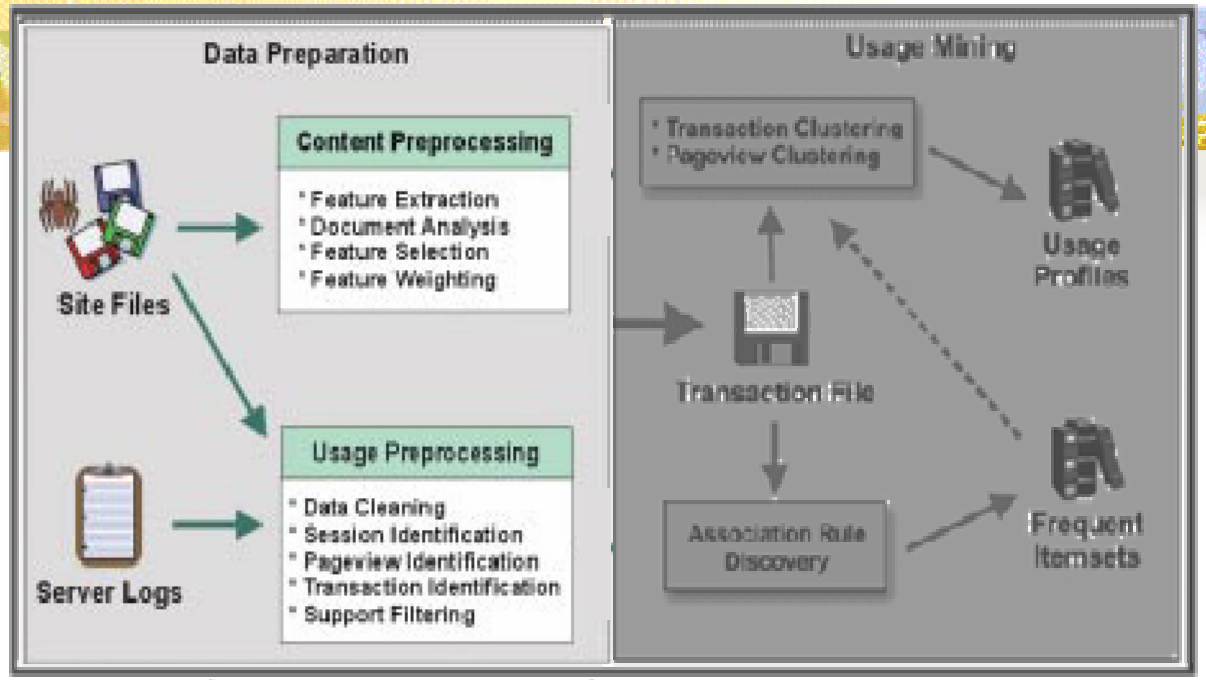
Usage profiles for Personalization

step by step

(Mobasher, Dai, Luo, Nakagawa)

①

Preprocessing



Batch Process

Online Process

Preprocessing/1

- Data cleaning
  - Requests for .gif, .GIF, .jpg, .JPG, ... (editable list) are filtered out
- User identification
  - IP+Agent name to distinguish users
  - Referrers and site topology: access to a page not reachable from visited pages ➔ new user
- Session identification
  - Time-oriented: maximum threshold between contiguous requests. Default:30 minutes.

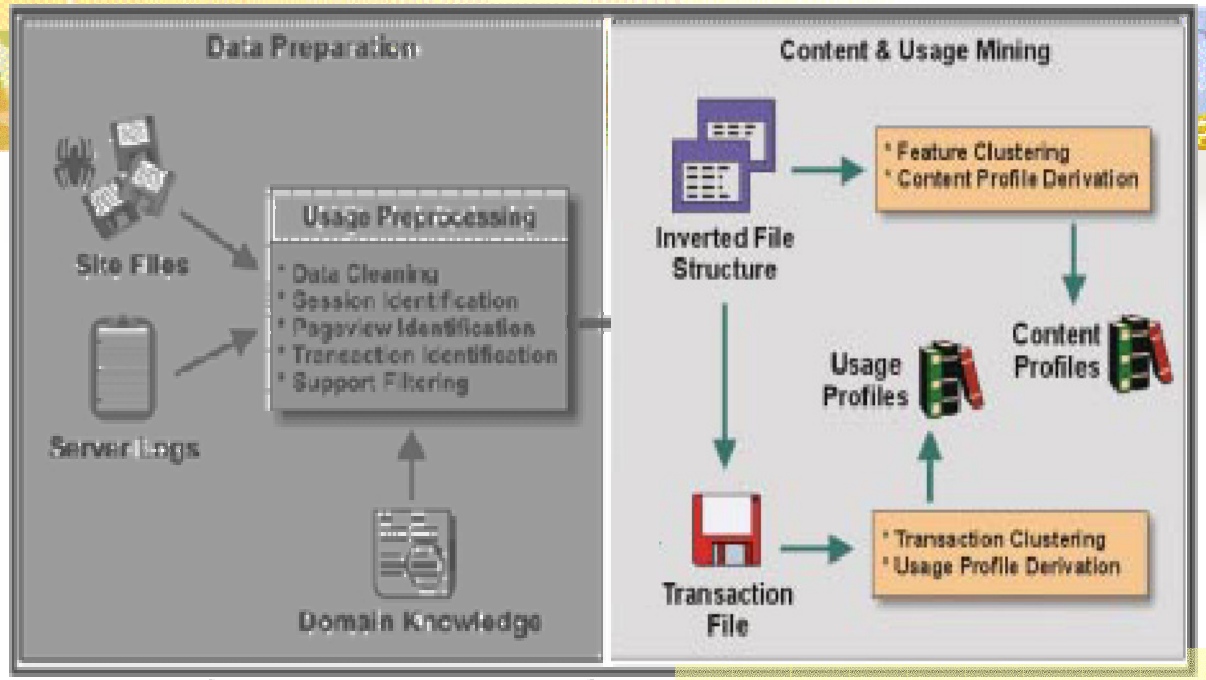
Preprocessing/2

- Path completion
  - Standard approach, selecting minimal-length paths
- Transaction identification (= sub-parts of sessions)
  - 3 Techniques, analogous to session identification:
    - **Reference lenght**: time-out between contiguous requests
    - **Maximal forward reference**: backtrack ➔ new transaction
    - **Time window**: time-out between first and last requests

Preprocessing/3

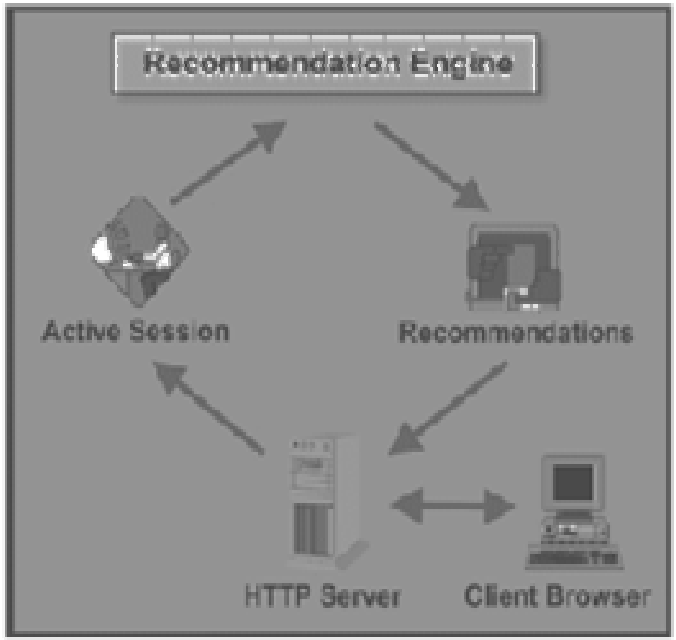
Format:

- Transactions are represented as vectors
  - Dimensions: one for each possible page view
  - Values: for each page view *p*, its *weight*:
    - w(*p*)∈ [0,1] If *p* is in the transaction
    - 0 Otherwise
  - Computing *w(p)*:
    - From site structure analysis
    - By domain experts



Batch Process

Profile extraction



Online Process

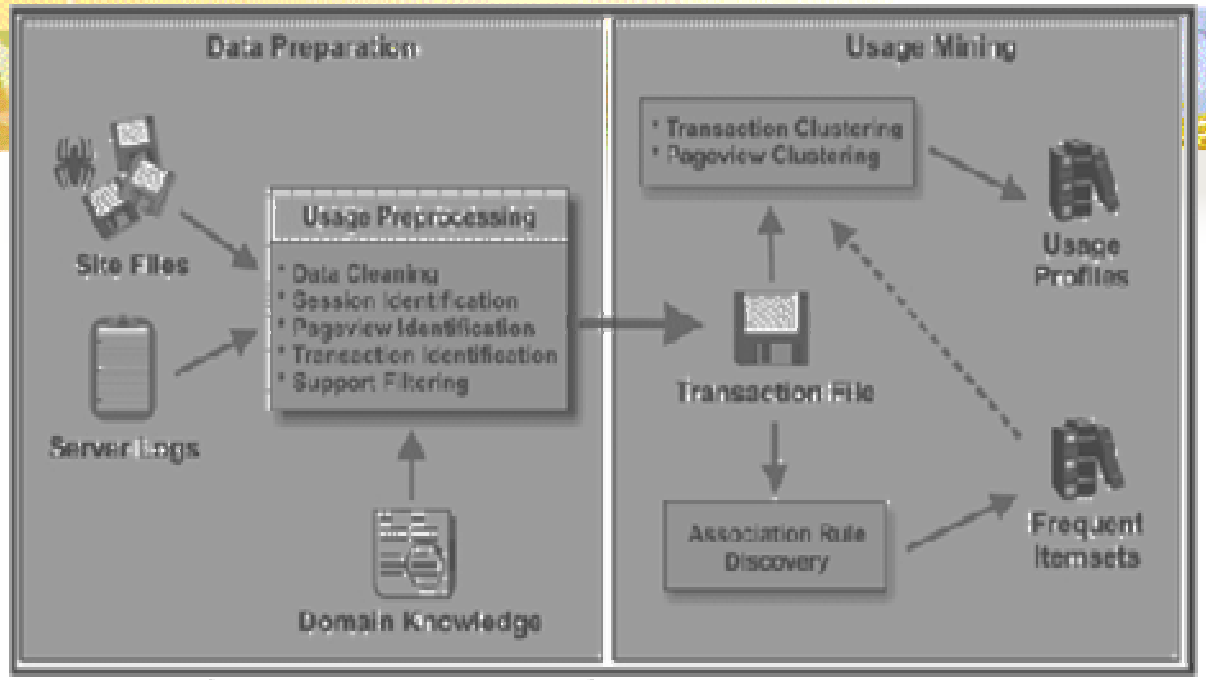
Computing aggregate profiles

- Transaction clustering
  - Standard k-means clustering over the vector representation of transactions
  - Result: set of clusters TC= $\{c_1, \ldots, c_n\}$
- For each cluster *c*, extract its corresponding profile:
  - compute the *mean vector mc* of *c*
  - normalise values so that min=0 and max=1
  - components $< \mu$ are set to 0 ➔ the page view is discarded

Computing aggregate profiles

Example

Table 1. User behavior profiles.		
	Weight	**Pageview URI**
Profile 1	0.78	Call for Papers
	0.67	CFP: ACR 1999 Asia-Pacific Conference
	0.64	CFP: Society For Consumer Psychology Conference
	0.61	ACR 1999 Annual Conference
	0.55	CFP: ACR 1999 European Conference
	0.52	CFP: Int'l Conference on Marketing and Development
	0.50	Conference Update
Profile 2	0.82	CFP: Journal of Psychology and Marketing II
	0.71	CFP: Society For Consumer Psychology Conference
	0.68	Conference Update
	0.68	CFP: Journal of Consumer Psychology II
	0.56	CFP: Conference on Gender, Marketing and Consumer Behavior
	0.52	Online Archives



Batch Process

Recommendation

③

Online Process

Computing Recommendations

- The active session ***s*** of a user is considered
  - Sliding window: take only last *n* page views
  - Compute its vector representation, with weights
- Match *s* with all profiles *mc* (clusters C) *:*
  - Normalised cosine similarity: $$match(S,C) = \frac{\sum_k w_k^C \cdot S_k}{\sqrt{\sum_k (S_k)^2 \times \sum_k (w_k^C)^2}}$$
- Compute the *recommendation scores*, for all $p \in C$:

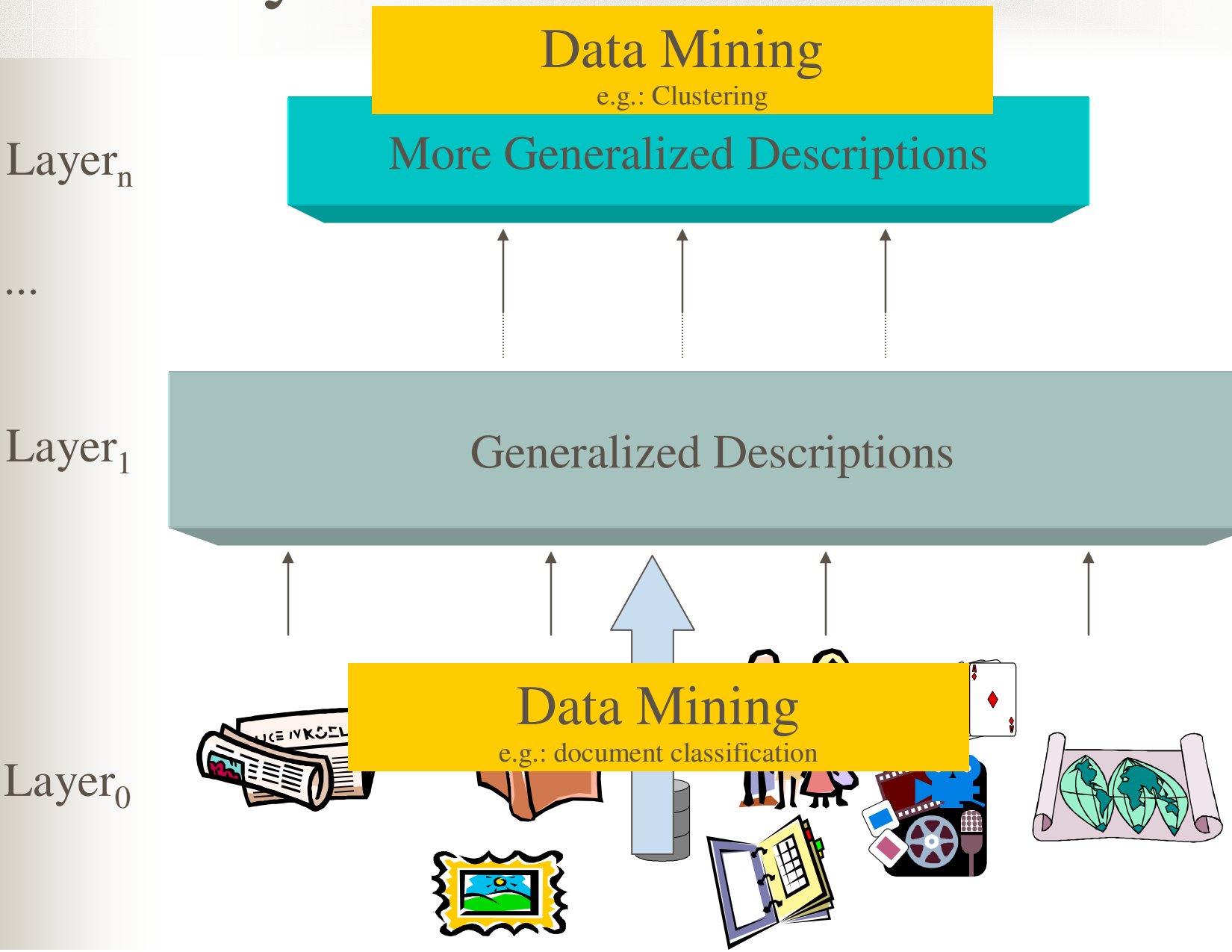
$$Rec(S, p) = \sqrt{weight(p, C) \cdot match(S, C)}$$

- Add *k* best recommendations to the last page requested

Research topics

- All steps of the process can be improved
  - Preprocessing: user identification means, better session heuristics, etc.
  - Mining: new algorithms for web data, ...
  - Pattern analysis: more effective filtering strategies, qality evalution methodologies, ...
- XML: exploiting DTD semantic information
- Web wharehousing: multi-abstraction level organisation of the data on the web (→next slide)
- ...

Multilayered Web Information Base



Short bibliography

1. Special Issue of "Data Mining and Knowledge Discovery" on "*Applications of Data Mining to E commerce*" - vol 5 num 2/3
2. Special Issue of "Data Mining and Knowledge Discovery" on "*Web Mining*" - vol 6 num 1
3. Special Issue of Communication of the ACM, August 2000 on Personalization
4. Robert Cooley, Bamshad Mobasher, and Jaidep Srivastava. Data preparation for mining world wide web browsing patterns. Journal of Knowledge and Information Systems , 1(1), 1999.
5. W. Gaul and L. Schmidt-Thieme. Recommender systesms based on navigation path features. KDD'2001 Workshop WEBKDD'2001. ACM.
6. Mobasher, B., Cooley, R., and Srivastava, J. (2000). Automatic personalization based on web usage mining. Communications of the ACM, 43(8), 142-151.
7. B. Mobasher, H. Dai, T. Luo, M. Nakagawa. Effective personalization based on association rule discovery from Web usage data. Technical Report 01-010, Deaprtment of Computer Science, DePaul University.
8. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, Y. Sun, and J. Wiltshire. Discovery of aggregate usage profiles for web personalization. KDD'2000 Workshop WEBKDD'2000. ACM.
9. M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In Proc. Of AAAI/IAAI'98, pages 727-732, 1998.