Web Mining Research at the Center for Web Intelligence

Bamshad Mobasher
Center for Web Intelligence
School of Computer Science, Telecommunication, and Information Systems
DePaul University, Chicago
What is Web Mining

- From its very beginning, the potential of extracting valuable knowledge from the Web has been quite evident
  - Web mining is the collection of technologies to fulfill this potential.

Web Mining Definition

application of data mining and machine learning techniques to extract useful knowledge from the content, structure, and usage of Web resources.
A Taxonomy for Web Mining

Web Mining
- Web Content Mining
  - IR / IA Approach
    - Intelligent Search Tools
    - Info. Filtering/Categorization
    - Personalized Info. Agents
    - Co-Citation Analysis
    - Content-Based Filtering
  - Database Approach
    - Web Query Systems
    - Concept Hierarchies
    - Structure Discovery
    - Ontology Learning
    - Web Info. Integration
- Web Usage Mining
  - E-Commerce Data Analysis
  - Site Evaluation/optimization
  - E-Metrics and Web Analytics
  - User Modeling/Profiling
  - Web Personalization
  - Web Communities

Web Structure Mining
Web Usage Mining

• The Problem
  - Analyze Web navigational data to find how the Web site is used by Web users

• Current Approaches
  - Statistical reports
  - OLAP
  - Data mining/Machine Learning techniques

• Applications
  - Web server caching
  - Link prediction and analysis
  - Web personalization
  - Web site adaptation
  - Web site evaluation and reorganization

• Challenge
  - Quantitatively capture Web users’ common interests and characterize their underlying tasks
Web Usage Mining as a Process

Data Preparation Phase

- Site Content & Structure
- Domain Knowledge

Pattern Discovery Phase

- Pattern Analysis
- Pattern Filtering
- Aggregation
- Characterization

- Aggregate User Models

Web & Application Server Logs

Data Preprocessing

- Data Cleaning
- Pageview Identification
- Sessionization
- Data Integration
- Data Transformation

User Transaction Database

Usage Mining

- User/Session Clustering
- Pageview Clustering
- Correlation Analysis
- Association Rule Mining
- Sequential Pattern Mining
Common Data Mining Tasks

• Clustering:
  ‣ Automatically group together users with similar purchasing or navigational patterns
    • User / customer segments
  ‣ Automatically group together items based on co-occurrence in user sessions
  ‣ Automatic creation of concept or functional hierarchies for the site

• Classification
  ‣ Categorizing pages or items into a concept hierarchy
  ‣ Classifying users into behavioral groups (e.g., browser, likely to purchase, loyal customer, etc.)
Common Data Mining Tasks

• Association Rules
  ‣ Associating presence of a set of items with other sets of items
    • $X \rightarrow Y$, where $X$ and $Y$ are sets of items
    • Support of the itemset $X \cup Y$: $Pr(X,Y)$; Confidence of rule: $Pr(Y|X)$
  ‣ Examples:
    • 30% of clients who accessed /special-offer.html, placed an online order in /products/software/

• Sequential Patterns / Path Analysis
  ‣ Finding common sequences of events/items appearing frequently in transactions
    • “x% of the time, when ABC appears in a transaction, C appears within z transactions”
    • 40% of people who bought the book “How to cheat IRS” booked a flight to South America 6 months later
    • 15% of visitors followed the path
      home > * > software > * > shopping cart > checkout
Example: Path Analysis for Ecommerce

Visit

10%

90%

Search (64% successful)

Search

Avg sale per visit: 2.2X

70%

30%

Last Search Succeeded

Avg sale per visit: 2.8X

No Search

Avg sale per visit: $X

Last Search Failed

Avg sale per visit: 0.9X
### Example: Association Analysis for Ecommerce

<table>
<thead>
<tr>
<th>Product</th>
<th>Association</th>
<th>Lift</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Reversible Mats</td>
<td>Egyptian Cotton Towels</td>
<td>456</td>
<td>41%</td>
</tr>
<tr>
<td>White Cotton T-Shirt Bra</td>
<td>Plunge T-Shirt Bra</td>
<td>246</td>
<td>25%</td>
</tr>
</tbody>
</table>

- **Confidence**: 41% who purchased Fully Reversible Mats also purchased Egyptian Cotton Towels
- **Lift**: People who purchased Fully Reversible Mats were 456 times more likely to purchase the Egyptian Cotton Towels compared to the general population
Web Usage Mining: clustering example

- **Transaction Clusters:**
  - Clustering similar user transactions and using centroid of each cluster as a usage profile (representative for a user segment)

Sample cluster centroid from dept. Web site (cluster size = 330)

<table>
<thead>
<tr>
<th>Support</th>
<th>URL</th>
<th>Pageview Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>/courses/syllabus.asp?course=450-96-303&amp;q=3&amp;y=2002&amp;id=290</td>
<td>SE 450 Object-Oriented Development class syllabus</td>
</tr>
<tr>
<td>0.97</td>
<td>/people/facultyinfo.asp?id=290</td>
<td>Web page of a lecturer who thought the above course</td>
</tr>
<tr>
<td>0.88</td>
<td>/programs/</td>
<td>Current Degree Descriptions 2002</td>
</tr>
<tr>
<td>0.85</td>
<td>/programs/courses.asp?depcode=96&amp;deptmne=se&amp;courseid=450</td>
<td>SE 450 course description in SE program</td>
</tr>
<tr>
<td>0.82</td>
<td>/programs/2002/gradds2002.asp</td>
<td>M.S. in Distributed Systems program description</td>
</tr>
</tbody>
</table>
Latent Variable Models

- Assume the existence of a set of latent (unobserved) variables (or factors) which “explain” the underlying relationships between two sets of observed variables.

Probabilistic Latent Semantic Analysis (PLSA):

- Probabilistically determine the association between each latent factor and pages, or between each factor and users.
- Latent factors correspond to distinguishable patterns usually associated with performing certain navigational task.
“Task-Oriented” User Tracking
Example from CTI Data

A real user session (page listed in the order of being visited)

<table>
<thead>
<tr>
<th>PageName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admission main page</td>
</tr>
<tr>
<td>Welcome information – Chinese version</td>
</tr>
<tr>
<td>Admission info for international students</td>
</tr>
<tr>
<td>Admission - requirements</td>
</tr>
<tr>
<td>Admission – mail request</td>
</tr>
<tr>
<td>Admission – orientation info</td>
</tr>
<tr>
<td>Admission – F1 visa and I20 info</td>
</tr>
<tr>
<td>Application – status check</td>
</tr>
<tr>
<td>Online application - start</td>
</tr>
<tr>
<td>Online application – step 1</td>
</tr>
<tr>
<td>Online application – step 2</td>
</tr>
<tr>
<td>Online application - finish</td>
</tr>
<tr>
<td>Department main page</td>
</tr>
</tbody>
</table>

Top factors given this user

<table>
<thead>
<tr>
<th>Task</th>
<th>Factor Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 3</td>
<td>0.4527</td>
</tr>
<tr>
<td>Task 25</td>
<td>0.3994</td>
</tr>
<tr>
<td>Task 20</td>
<td>0.0489</td>
</tr>
<tr>
<td>Task 26</td>
<td>0.0458</td>
</tr>
</tbody>
</table>

PageName

<table>
<thead>
<tr>
<th>PageName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Department main page</td>
</tr>
<tr>
<td>Admission requirements</td>
</tr>
<tr>
<td>Admission main page</td>
</tr>
<tr>
<td>Admission costs</td>
</tr>
<tr>
<td>Programs</td>
</tr>
<tr>
<td>Online application – step 1</td>
</tr>
<tr>
<td>…</td>
</tr>
<tr>
<td>Admission – international students</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PageName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online application – start</td>
</tr>
<tr>
<td>Online application – step1</td>
</tr>
<tr>
<td>Online application – step2</td>
</tr>
<tr>
<td>Online application - finish</td>
</tr>
<tr>
<td>Online application - submit</td>
</tr>
<tr>
<td>…</td>
</tr>
<tr>
<td>Department main page</td>
</tr>
</tbody>
</table>
Web Personalization

• The Problem
  ‣ dynamically serve customized content (pages, products, recommendations, etc.) to users based on their profiles, preferences, or expected interests

• Current Approaches
  ‣ collaborative filtering
    • give recommendations to a user based on ratings of “similar” users
    • most successful and wide-spread approach
    • but, suffers from scalability problems given many items and users
  ‣ content-based filtering
    • track pages the user visits and recommend other pages with similar content
    • may miss relationships among objects not based on content similarity
  ‣ rule-based filtering
    • provide content to users based on predefined rules (e.g., “if user has clicked on A and the user’s zip code is 90210, then add a link to C”)
    • relies on static profiles for users obtained through explicit feedback
Web Mining Approach to Personalization

• Basic Idea
  ‣ generate *aggregate user models* (usage profiles) by discovering user access patterns through Web usage mining (offline process)
  ‣ match a user’s active session against the discovered models to provide dynamic content (online process)

• Advantages
  ‣ no explicit user ratings or interaction with users
  ‣ helps preserve user privacy, by making effective use of anonymous data
  ‣ enhance the effectiveness and scalability of collaborative filtering
  ‣ more accurate and broader recommendations than content-only approaches
Automatic Web Personalization:

Recommendation Engine

Web Server

Client Browser

Active Session

Recommendations

Integrated User Profile

Stored User Profile

Domain Knowledge

Aggregate Usage Profiles

<user,item1,item2,…>
Two important factors in evaluating recommendations

- **Precision**: measures the ratio of “correct” recommendations to all recommendations produced by the system
  - low precision would result in angry or frustrated users

- **Coverage**: measures the ratio of “correct” recommendations to all pages/items that will be accessed by user
  - low coverage would inhibit the ability of the system to give relevant recommendations at critical points in user navigation

Transactions Divided into **Training & Evaluation** Sets

- training set is used to build models (generation of aggregate profiles, neighborhood formation)
- evaluation set is used to measure precision & coverage
- 10-Fold Cross Validation generally used in the experiments
Problems with Web Usage Mining

- **New item problem**
  - Patterns will not capture new items recently added
  - Bad for dynamic Web sites, and in the context of collaborative filtering

- **Poor machine interpretability**
  - Hard to generalize and reason about patterns
    - No domain knowledge used to enhance results
    - E.g., knowing a student is interested in a particular program, we could recommend the prerequisites, core or popular courses in this program, related courses in other programs, etc.

- **Poor insight into the patterns themselves**
  - The nature of the relationships among items or users in a pattern is not directly available

- **Solution**: Integrate Semantic Knowledge with Web Usage Mining
Ontology-Based Usage Mining

- **Approach 1: Ontology-Enhanced Transactions**
  - Initial transaction vector: \( t = <\text{weight}(p_1,t), ..., \text{weight}(P_n,t)> \)
  - Transform into content-enhanced transaction:
    \[ t = <\text{weight}(o_1,t), ..., \text{weight}(o_n,t)> \]
  - The structured objects \( o_1, ..., o_r \) are instances on ontology entities extracted from pages \( p_1, ..., p_n \) in the transaction
  - Now mining tasks can be performed based on ontological similarity among user transactions

- **Approach 2: Ontology-Enhanced Patterns**
  - Discover usage patterns in the standard way
  - Transform patterns by creating an aggregate representation of the patterns based on the ontology
    - Requires the categorization of similar objects into ontology classes
    - Also requires the specification of different aggregation/combination function for each attribute of each class in the ontology
Ontology-Based Pattern Aggregation

<table>
<thead>
<tr>
<th>Name</th>
<th>Genre</th>
<th>Actor</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie 1: {A}</td>
<td>Genre-All</td>
<td>{S: 0.7; T: 0.2; U: 0.1}</td>
<td>{2002}</td>
</tr>
<tr>
<td></td>
<td>Romance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comedy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Romantic Comedy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kid&amp;Family</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Movie 2: {B}</td>
<td>Genre-All</td>
<td>{S: 0.5; T: 0.5}</td>
<td>{1999}</td>
</tr>
<tr>
<td></td>
<td>Romance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Comedy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Movie 3: {C}</td>
<td>Genre-All</td>
<td>{S: 0.6; W: 0.4}</td>
<td>{2000}</td>
</tr>
<tr>
<td></td>
<td>Comedy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usage profile

- 0.50 Movie1.html
- 0.35 Movie2.html
- 0.15 Movie3.html

Object Extraction

Ontology-Based Aggregation

Semantic Usage pattern
Example: Semantically Enhanced Collaborative Filtering

• **Basic Idea:**
  - Extend item-based collaborative filtering to incorporate both similarity based on ratings (or usage) as well as semantic similarity based on domain knowledge

• **Structured semantic knowledge about items**
  - Extracted automatically from the Web based on domain-specific reference ontologies
  - Used in conjunction with user-item mappings to create a combined similarity measure for item comparisons
  - Singular value decomposition used to reduce noise in the semantic data

• **Semantic combination threshold**
  - Used to determine the proportion of semantic and rating (or usage) similarities in the combined measure
Web Content Mining

• Discovery of meaningful patterns from page content and meta data across the Web or in a particular Web site

• Examples of Techniques Used
  ‣ text mining
  ‣ document categorization and clustering
  ‣ semantic analysis

• Applications
  ‣ content-based filtering
  ‣ Information extraction
  ‣ intelligent interface agents
  ‣ discovery of user profiles
  ‣ ontology acquisition and learning
  ‣ recommendation systems
  ‣ opinion extraction from the Web
  ‣ contextual information access
Contextual Information Access

- Integrating multiple sources of evidence
  - Short-term user context; Long-term user profiles; Domain knowledge

```
  Short-term User Context
  Long-term User Profile
  Domain Knowledge

  Information Access Context

  Information Access Operations (Searching, Browsing)

  Other Information Sources
```
ARCH Background

- ARCH - Adaptive Agent for Retrieval Based on Concept Hierarchies

  - Essence of the system is to incorporate domain-specific concept hierarchies with interactive query formulation

  - Query enhancement in ARCH uses two mutually-supporting techniques:
    - **Semantic** – using a concept hierarchy to interactively disambiguate and expand queries
    - **Behavioral** – observing user’s past browsing behavior for user profiling and automatic query enhancement
Domain Knowledge & Search Context

How about a pet python?

python

ARCH

Concept Hierarchy

Entertainment
- Comedy
  - Monty Python
- Movies

Science
- Reptiles
- Mammals
- Pythons

Computers & Internet
- Programming & Development
  - Languages
  - Web Directories
- Python

Enhanced Query

Search Engine
Overview of the System

- **The system consists of an offline and an online component**

- **Offline component:**
  - Handles the learning of the concept hierarchy
  - Handles the learning of the user profiles

- **Online component:**
  - Displays the concept hierarchy to the user
  - Allows the user to select/deselect nodes
  - Generates the enhanced query based on the user’s interaction with the concept hierarchy
Query Enhancement Mechanism

User Enters Keyword Query

User Interaction

Retrieve Relevant Portions of the Concept Classification Hierarchy

Display Concept Hierarchy

User Selects and/or Deselects Concepts

Enhanced Query Generation

User Profile Module

Concept Classification Hierarchy
Example from Yahoo Hierarchy

Term Vector for "Genres:")

- music: 1.000
- blue: 0.15
- new: 0.14
- artist: 0.13
- jazz: 0.12
- review: 0.12
- band: 0.11
- polka: 0.10
- festiv: 0.10
- celtic: 0.10
- freestyl: 0.10
Aggregate Representation of Nodes in the Hierarchy

• **A node is represented as a weighted term vector:**
  - centroid of all documents and subcategories indexed under the node

\[
T_n = \left[ \frac{\left( \sum_{d \in D_n} T_d \right) / |D_n| + \sum_{s \in S_n} T_s}{|S_n| + 1} \right]
\]

- \( n \) = node in the concept hierarchy
- \( D_n \) = collection of individual documents
- \( S_n \) = subcategories under \( n \)
- \( T_d \) = weighted term vector for document \( d \) indexed under node \( n \)
- \( T_s \) = the term vector for subcategory \( s \) of node \( n \)
Online Component – User Interaction with the Hierarchy

• **The initial user query is mapped to the relevant portions of hierarchy**
  - user enters a keyword query
  - system matches the term vectors representing each node in the hierarchy with the keyword query
  - nodes which exceed a similarity threshold are displayed to the user, along with other adjacent nodes.

• **Semi-automatic derivation of user context**
  - ambiguous keyword might cause the system to display several different portions of the hierarchy
  - user **selects** categories which are relevant to the intended query, and **deselects** categories which are not
Generating the Enhanced Query

- Based on an adaptation of Rocchio's method for relevance feedback
  - Using the selected and deselected nodes, the system produces a refined query $Q_2$:
    \[ Q_2 = \alpha \cdot Q_1 + \beta \cdot \sum T_{\text{sel}} - \gamma \cdot \sum T_{\text{delsey}} \]
    - each $T_{\text{sel}}$ is a term vector for one of the nodes selected by the user,
    - each $T_{\text{delsey}}$ is a term vector for one of the deselected nodes
    - factors $\alpha$, $\beta$, and $\gamma$ are tuning parameters representing the relative weights associated with the initial query, positive feedback, and negative feedback, respectively such that $\alpha + \beta - \gamma = 1$. 
An Example

Initial Query: “music, jazz”

Selected Categories: “Music”, “jazz”, “Dixieland”

Deselected Category: “Blues”

Portion of the resulting term vector:

music: 1.00, jazz: 0.44, dixieland: 0.20, tradition: 0.11, band: 0.10, inform: 0.10, new: 0.07, artist: 0.06
Another Example – ARCH Interface

- Initial query = *python*
- Intent for search = *python as a snake*
- User selects *Pythons under Reptiles*
- User deselects *Python under Programming and Development and Monty Python under Entertainment*

**Enhanced query:**

<table>
<thead>
<tr>
<th>Burmes: 1</th>
<th>Infect: 0.494</th>
<th>Constrictor: 0.42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python: 0.937</td>
<td>Vet: 0.472</td>
<td>Pet: 0.41</td>
</tr>
<tr>
<td>Snake: 0.733</td>
<td>Lizard: 0.472</td>
<td></td>
</tr>
<tr>
<td>Reptile: 0.499</td>
<td>Egg: 0.43</td>
<td></td>
</tr>
</tbody>
</table>
Experimental Evaluation

- Performed testing using a small subset of the Yahoo hierarchy
  - Approximately 50 nodes
  - Domains: Business and Economy, Computers and Internet, Entertainment, Science

- Search scenarios
  - Single keyword: python, bat, hardware, bug, mouse
  - Two keywords: python snake, bat mammal, baseball bat, computer hardware, home hardware, bug surveillance, computer mouse
  - Selection and deselection of nodes based on specific search intent

- Ten signal documents and ten noise documents collected for each search scenarios
  - indexed using tf.idf weights
Experiment Evaluation, cont.

- **Two types of search**
  - Simple Query Search
    - Used the user’s original query
  - Enhanced Query Search
    - Used the enhanced query that was generated by ARCH
    - Selection and deselection of nodes based on search scenarios

- **Matching** - cosine similarity measure
- **Comparison** - precision and recall
Results

Average precision for Enhanced Query versus Simple Query Search using two keywords

![Graph showing precision vs. similarity threshold](image)

**Precision** = proportion of retrieved documents actually relevant

Average recall for Enhanced Query versus Simple Query Search using two keywords

![Graph showing recall vs. similarity threshold](image)

**Recall** = proportion of relevant documents actually retrieved
Can user profiles replace the need for user interaction?

- Instead of explicit user feedback, the user profiles are used for the selection and deselection of concepts.
- Each individual profile is compared to the original user query for similarity.
- Those profiles which satisfy a similarity threshold are then compared to the matching nodes in the concept hierarchy.
  - Matching nodes include those that exceeded a similarity threshold when compared to the user’s original keyword query.
- The node with the highest similarity score is used for automatic selection; nodes with relatively low similarity scores are used for automatic deselection.
Generation of User Profiles

• Profile Generation Component of ARCH
  ▸ passively observe user’s browsing behavior
  ▸ use heuristics to determine which pages user finds “interesting”
    ▫ time spent on the page (or similar pages)
    ▫ frequency of visit to the page or the site
    ▫ other factors, e.g., bookmarking a page, etc.
  ▸ implemented as a client-side proxy server

• Clustering of “Interesting” Documents
  ▸ ARCH extracts feature vectors for each profile document
  ▸ documents are clustered into semantically related categories
    ▫ we use a clustering algorithm that supports overlapping categories to capture relationships across clusters
    ▫ algorithms: overlapping version of k-means; hypergraph partitioning
  ▸ profiles are the significant features in the centroid of each cluster
Results Based on User Profiles

Simple vs. Enhanced Query Search

- Simple Query Single Keyword
- Simple Query Two Keywords
- Enhanced Query with User Profiles

Precision vs. Threshold (%)

Recall vs. Threshold (%)

Simple vs. Enhanced Query Search
Future Work on ARCH

• Integrate the system with a specific search engine on the World Wide Web
  ▸ Weighted term vector queries must be translated into Boolean queries

• Experiment using other concept hierarchies
  ▸ Specific domains (e.g., medical)
  ▸ Design is intended to be independent of specific concept hierarchy

• Fully integrate and evaluate the User Profile Module

• Additional information
  ▸ http://www.ahusieg.com/arch
Some Ongoing Projects at CWI

• Web usage mining and Web analytics
  ‣ Intelligent techniques for sessionization and pageview identification
  ‣ Integration of usage, content, and structure
  ‣ Ontology-based Web usage mining (Semantic Web Usage Mining)

• Web personalization
  ‣ Data mining techniques for personalization
  ‣ Integrating domain knowledge (ontologies) in user modeling
  ‣ Hybrid recommendation systems
  ‣ Secure Personalization

• Web content mining / Information Agents
  ‣ Automatically learning domain-specific ontologies from Web sites
  ‣ Integrating domain ontologies, learned user profiles, and short-term user interests for “contextual information access” on the Web
    • The ARCH project