Web Mining and Searching

Approaches to Efficient Automated Knowledge Discovery from Semi-structured Web sources

Lecture's Outline

- Brief Introduction
- Web Content Mining
- CASE STUDY: Extracting Patterns & Relations
  - "Books and Authors": an intriguing early experiment to mine the Web for relational data
- Web Usage Mining
- CASE STUDY: Dynamic Itemset Counting
  - discovering interesting sets of items in a space too large to even consider each pair of items

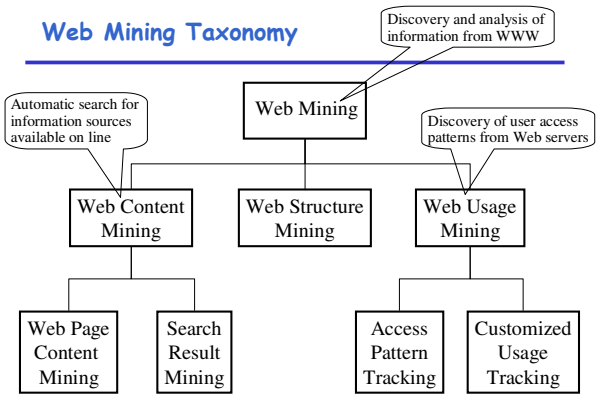
Web Mining Issues (slightly outdated)

- Size of Web in 1999 (estimates)
  - >350 million pages
  - was growing at about 1 million pages a day
- In 2001, Google announced that it was indexing 3 billion documents
- Diverse types of data

Web Data

- Web pages
- Intra-page structures
- Inter-page structures
- Usage data
- Supplemental data
  - Profiles
  - Registration information
  - Cookies

Web Mining Taxonomy



Web Content Mining

Web Content Mining

- Extends work of basic search engines
- Search Engines
  - IR application
  - Keyword based
  - Similarity between query and document
  - Crawlers
  - Indexing
  - Profiles
  - Link analysis

Crawlers

Robot (spider) traverses the hypertext structure in the Web.

- Collect information from visited pages
- Used to construct indices for search engines

Traditional Crawler – visits entire (?) Web and replaces index

Periodic Crawler – visits portions of the Web and updates subset of index

Incremental Crawler – selectively searches the Web and incrementally modifies index

Focused Crawler – visits pages related to a particular subject

Focused Crawler

- Only visits links from a page if that page is determined to be relevant.
- The classifier is static after learning phase.
- Components:
  - Classifier which assigns relevance score to each page based on crawl topic.
  - Distiller to identify *hub pages.*
  - Crawler visits pages to based on classifier and distiller scores.

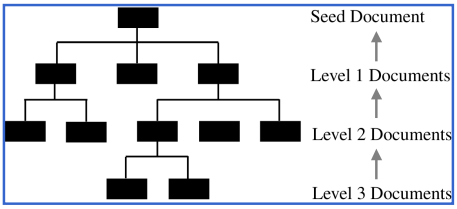
Focused Crawler

- Classifier to relate documents to topics
- Classifier also determines how useful outgoing links are
- ***Hub Pages*** contain links to many relevant pages. Must be visited even if not high relevance score.

Context Focused Crawler

- Context Graph:
  - Context graph created for each seed document.
  - Root is the seed document.
  - Nodes at each level show documents with links to documents at next higher level.
  - Updated during crawl itself.
- Approach:
  - Construct context graph and classifiers using seed documents as training data.
  - Perform crawling using classifiers and context graph created.

Context Graph

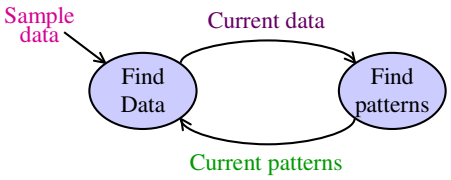


Extracting Patterns and Relations from the WWW

Case Study of Web Content Mining

Extracting Patterns and Relations from WWW

- General idea is to search the Web for data of a given type (i.e., data that are somehow related)



Extracting Patterns and Relations from WWW

Let R = r1, r2, …, rN the target relation.

If we compute an approximation R' of R, then
- the coverage is |R'∩R| / |R|
- the error rate is |R'-R| / |R'|

Goal is to maximize coverage and minimize error rate.

Low error rate is much more critical than high coverage:
- given a large database, coverage of 20% is acceptable;
- an error of 10% might make the result useless.

Extracting Patterns and Relations from WWW

Start with a (representative) sample of the tuples;

Repeat // until fixpoint or some other criterion is met
// (e.g., timeout, out-of-memory, good results)
- Find where the data appears on the Web;
- Search this data for patterns;
  - If a found pattern identifies several examples of known tuples and is sufficiently specific that is unlikely to identify too much, keep it;
- Given the set of accepted patterns, find the data that appears in these patterns, and add it to the set of known data.

What is a Pattern?

- An indication of the *order* (whether a title appears prior to the author or vice-versa);
- The *URL prefix*;
- The *prefix* of the text;
- The *middle* (text between the two elements);
- The *suffix* of the text.

Example: Title->Author

user.it.uu.se/~kostis/Teaching/DM-05/

...<LI><I>Introduction to Data Mining</I> by Tan, Steinbach, and Kumar</LI>...

Finding Accurate Patterns

Specificity of a pattern

|prefix| x |middle| x |suffix| x |URL prefix|

roughly measures how likely we are to find a pattern; higher specificity ⇒ fewer occurrences we expect.

A pattern P must meet 2 conditions to be accepted:
- There must be at least 2 known items that appear in P;
- The product of the specificity of P and the number of occurrences of data items in P must exceed a certain threshold (user specified; not estimated).

Data Occurrences

- An occurrence of a tuple is associated with the pattern in which it occurs.
- It consists of:
  - Particular title and author;
  - The complete URL of occurrence;
  - The order, prefix, middle, and suffix of the pattern in which the title and author occurred.

Finding Data Occurrences given Data

We assume that there is an index of the Web.

- Find (pointers to) all pages containing any known author.
- Find (pointers to) all pages containing any known title. Start by pages with each word of a title, and then check that the words appear in the correct order on the page.
- Intersect the sets of pages that have an author and a title on them.

Note that the method is essentially a-priori.

Building Patterns from Data Occurrences

1. Group the data occurrences according to their order and middle;
2. For each group find the longest common prefix, suffix, and URL prefix;
3. If the specificity test for this pattern is met, accept it;
4. Else, try to split the group into two by extending the length of the URL pattern by one character. Goto step 2.

Finding Occurrences given Patterns

- Find all URL's that match the URL prefix in at least one pattern;
- For each of those pages, scan the text using a regular expression built from the pattern's prefix, middle, and suffix;
- Extract from each match the title and author, according to the order specified in the pattern.

Web Usage Mining

Web Usage Mining Applications

- Personalization: developing users' profiles
- Improve structure of a site's Web pages
- Aid in caching and prediction of future page references (prefetching)
- Improve design of individual pages
- Improve effectiveness of e-commerce
  - boost sales and advertising

Personalization

- Tuning Web access or contents to better fit the desires of each user.
- Manual techniques identify user's preferences based on profiles or demographics.

Collaborative filtering identifies preferences based on ratings from similar users.

Content based filtering retrieves pages based on similarity between pages and user profiles.

Web Usage Mining Activities

1. Preprocessing Web log (clickstream)
   - Cleanse
   - Remove extraneous information
   - Sessionize
     - Session: Sequence of pages referenced by one user at a sitting
2. Pattern Discovery
   - Count patterns that occur in sessions
   - Pattern is sequence of pages references in session
   - Similar to association rules
     - Transaction: session
     - Itemset: pattern (or subset)
     - Order is important
3. Pattern Analysis: interpretation of results

Association Rules in Web Mining

- Web Mining:
  - Content
  - Structure
  - Usage
- Frequent patterns of sequential page references in Web searching.
- Uses:
  - Caching
  - Clustering users
  - Develop user profiles
  - Identify important pages

Web Usage Mining Issues

- Identification of exact user not possible.
- Determining the exact sequence of pages referenced by a user is often not possible due to client-level caching, use of proxy servers, ISP's, firewalls, etc.
- Session is not a well defined notion
- Security, privacy, and legal issues

Web Log Cleansing

1. Replace source IP address with unique but non-identifying ID.
2. Replace exact URL of pages referenced with unique but non-identifying ID.
3. Delete error records and records containing not page data (such as figures and code).

Web Log Sessionizing

- Divide Web log into sessions.
- Two common techniques:
  - Number of consecutive page references from a source IP address occurring within a predefined time interval (e.g. 25 minutes).
  - All consecutive page references from a source IP address where the inter-click time is less than a predefined threshold.

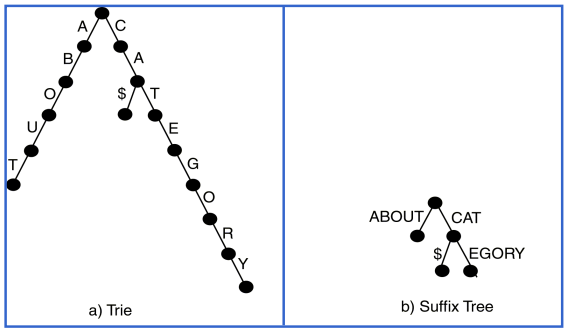
Data Structures

- Used to keep track of patterns identified during Web usage mining process
- Common techniques:
  - Trie
  - Suffix Tree
  - Generalized Suffix Tree
  - WAP Tree

Tries vs. Suffix Trees

- ***Trie:***
  - Rooted tree
  - Edges labeled which character (page) from pattern
  - Path from root to leaf represents pattern.
- ***Suffix Tree:***
  - Single child collapsed with parent.
  - Edge contains labels of both prior edges.

Tries and Suffix Trees



Generalized Suffix Tree

- Suffix tree for multiple sessions.
- Contains patterns from all sessions.
- Maintains count of frequency of occurrence of a pattern in the node.
- ***WAP (Web Access Pattern) Tree:***
  Compressed version of generalized suffix tree

Types of Patterns

- Algorithms have been developed to discover different types of patterns.
- Properties:
  - Ordered – Characters (pages) must occur in the exact order in the original session.
  - Duplicates – Duplicate characters are allowed in the pattern.
  - Consecutive – All characters in pattern must occur consecutive in given session.
  - Maximal – Not subsequence of another pattern.

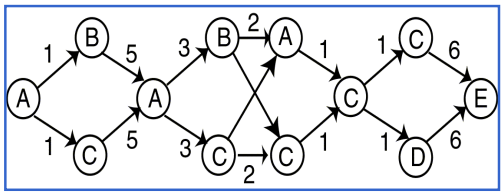
Pattern Types

- Association Rules
  - None of the properties hold
- Episodes
  - Only ordering holds
- Sequential Patterns
  - Ordered and maximal
- Forward Sequences
  - Ordered, consecutive, and maximal
- Maximal Frequent Sequences
  - All properties hold

Episodes

- Partially ordered set of pages
- *Serial episode* – totally ordered with time constraint (however, not contiguous)
- *Parallel episode* – partial ordered with time constraint
- *General episode* – partial ordered with no time constraint

DAG for Episode Discovery



Dynamic Itemset Counting

Finding Unusual Itemsets

Find words that appear together "unusually often"
("New" + "York" or "Dutchess" + "of" + "York")

Unusually often:

- The number of documents containing the words is much greater than that expected from a random distribution.
- One appropriate measure is *entropy per word in the set.* Formally, the *interest* in a set of words S is:

$$\frac{\log_2\left(\frac{prob(S)}{\prod_{w\in S} prob(\{w\})}\right)}{|S|}$$

Finding Unusual Itemsets: Technical Problems

- Interest is not monotone, or "downwards closed", the way support is.
  - I.e., we can have a set S with high interest value, yet some, or even all, of its intermediate proper subsets are not interesting.
- With more than 10^8 words appearing in the Web, it is not possible even to consider all pairs of words.

DICE: Dynamic Itemset Counting Engine

- Visits Web pages in a round-robin fashion
  - counts occurrences of certain sets of words and of the individual words in that set;
  - number of words small enough to fit in main memory
- From time to time, the words are reconsidered by throwing away low interest words.

Heavy edge property:

- An experimentally justified observation that words appearing in a high-interest set are more likely than others to appear in other high-interest sets.

DICE: Dynamic Itemset Counting Engine

Other constructions used to create new sets:

- Two random words; (*)
- One word from an interesting set + a random word;
- Two words from two different interesting pairs;
- The union of two interesting sets whose intersection is of size 2 or more;
- {a,b,c} if all of {a,b}, {a,c}, and {b,c} are interesting.

(*) Heavy-edge property independent construction