Web Usage Mining

Web Usage Mining

Web Usage Mining: automatic discovery and analysis of patterns in click-stream and associated data collected as a result of user interactions with Web resources on one or multiple web sites.

Automatic discovery of patterns.

- Data collection/pre-processing.
- Data fusion/cleaning.
- Pageview identification.
- User identification.
- Session identification.
- Path completion.

Analysis of patterns.

- Visitor segmentation.
- Association and correlation analysis.
- Analysis of navigational patterns.
- Classification and prediction.
- Opinion mining.
- User community discovery.

Pattern Discovery

Data Collection and Pre-processing

Data sources.

- **Usage data.**
  - Web server logs.
  - Web/On-line Application logs.
- **Content data.**
  - Web page content.
  - On-line databases. (dynamic content generation)
- **Structure data.**
  - Site structure. (via `<a href>` tags).
  - Site architecture/Site map.
  - On-line DBs. (dynamic content generation)
- **User data.**
  - Web server logs.
  - Cookies.
  - User databases (for authenticated users).

Web Server Logs

A **Web server log** is a collection of entries recording activities of an HTTP web server.

Web server log record. Content and format of web server log records varies from web server to server. However, some information is present in **all log entries** while some other information, while optional, is commonly included.

Typical fields found in a web server log record are:

Field	Mandatory/Optional
IP Address of the client	*mandatory*
Type of HTTP request	*optional?*
URL of requested page	*mandatory*
Timestamp of the request	*mandatory*
Return code	*?*
User agent	*optional*
Referring Web page URL	*optional*
Bytes transferred	*optional*
Cookies	*optional*

Pageviews and Sessions

Pageview. A **pageview** is an aggregate representation of a collection of objects generated/delivered to client on a signle "click"/request.

- Abstraction of a notion of a *web page*.
- Simple case: a single HTML document
- "Moderate" case: a single HTML document + graphics/multimedia files associated with it.
- "Frames" case: multiple HTML documents (with embedded scripts), multiple multimedia files.
- "Dynamic content" case: multiple HTML documents, back-end database requests/results, scripts, stylesheets and data/multimedia files.

Session. A **session** is a **sequence** of **pageviews** experienced by a **single user** during a single visit.

Data Pre-Processing

(We mostly concentrate on web usage/web server log data here, other information may also require pre-processing though).

Data Fusion. Combine web server logs from multiple servers.

Data Cleaning. Filtering web server logs to keep **only important HTTP requests** corresponding to "cores" of important **pageviews**.

- Filter out requests for image files (as part of web page loading process).
- Filter out crawlers/spiders.
  - Via self-Id (from HTTP fields);
  - Via announced behavior (e.g., reading `robots.txt`);
  - Via discovery of stealthy behavior (classification?);

Pageview Identification.

- Request-based. (each HTML page request = pageview).
- Content-based. (dealing with multiple frames).
- Structure-based. (dealing with dynamic content).
- Filter pageviews. (not all may be needed for consideration).

User Identification.

- Anonymous Users.
  - From IP addresses (simplistic).
  - From IP address/user agent pairs (a bit better).
  - From cookies (not always possible).
- Known Users. Map user acitivities to user account profiles.

In both cases, create **user activity record**.

Session Detection

Session detection is the process of determining which observed pageviews correspond to a single session.

Usually, **session detection** is a **heusristic process** that tries to approximate the real collection of user sessions R with a collection C_h specified by some heuristic function h. Ideally $C_h = R$.

Total time heuristic. Total session time no more than some threshold amound Θ. (Θ can be 2-3 hours, e.g.)

Time between pageviews heuristic. Two *consecutive pageviews* p_1 and p_2 with timestamps $t(p_1)$ and $t(p_2)$ are put in the same session **iff**

$$t(p_2) - t(p_1) <= \delta.$$

δ, selected in advance represents the maximum allowed time between pageviews (e.g., 30 mins, or 1 hour).

Sessions by referral. If referring web page information is stored in the web server logs, then a pageview p with a referring URL $r(p)$ is put in session S, **iff** $r(p)$ is a URL of some pageview in S.

This allows for *non-linear* session detection, but requires referral information.

Path Completion. **Path Completion** is the task of determining pageviews that *must have occurred* during a session, and *reinserting them* into the session.

Pageviews can be omitted from a session due to

- Misdetection (error of heuristic);
- Caching (a proxy server or a client has a cached version of the page);
- HTTP POST requests (the "dynamic" part of the request is not kept in the log).

Web Usage Pattern Analysis

- Visitor segmentation.
- Association and correlation analysis.
- Analysis of navigational patterns.
- Classification and prediction.
- Opinion mining.
- User community discovery.

Sessions and Transactions

There are two ways to view the collected session data:

- Sequential: order of pageviews matters. Used for analysis of navigational patterns, prediction, association analysis.
- Vector: order of pageviews does not matter, only whether a page was visited. Used for visitor segmentation, opinion mining, classification and prediction, community discovery.

Session as a sequence. $S = (p_1, \ldots, p_n)$, where p_i is a pageview, and $t(p_{i-1}) < t(p_i) < t(p_{i+1})$. ($t(p)$ is the timestamp of the pageview p).

Session as a vector. $S = (w_1, \ldots w_N)$. Here N is the number of **unique** pages/pageviews on the web site, and w_i is the **weight** of a pageview p_i in session S. If we consider each session to have a unique user u_j, then, we obtain the following **user-pageview matrix**:

	Pageviews			
Users	p_1	p_2	$\ldots$	p_N
user1	w_{11}	w_{12}	$\ldots$	w_{1N}
user2	w_{21}	w_{22}	$\ldots$	w_{2N}
user3	w_{31}	w_{32}	$\ldots$	w_{3N}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
userM	w_{M1}	w_{M2}	$\ldots$	w_{MN}

Weights w_{ij} can be determined in different ways:

- Boolean: $w_{ij} = 1$ if user_i visited p_j and 0 otherwise.
- Frequency: w_{ij} is the frequency of visits of user_i to page p_j.
- Normalized frequency: w_{ij} is the frequency of visits of user_i to page p_j normalized by some factor (e.g., maximum number of visits to a page during a session).

Pageview-feature matrix. A **pageview-feature** matrix is a mapping of a **preset** set of **features** to each pageview.

Content-enhanced transaction matrix. A mapping between sessions/users and **features** constructed via user-pageview matrix and pageview-feature matrix.

Analysis

Visitor Segmentation

Problem. Given a list of sessions, group them by percieved similarity (in the absence of other knowledge). Represent each cluster by an aggregate object.

Solution(s) . Unsupervised learning. Usually, partitional methods (k-center clustering) are used.

Profile constuction. Use cluster centroids.

Association and Correlation Analysis

Problem. Find pages that are commonly accessed together.

Solution(s) . Association rules mining techniques (Apriori algorithm modifications).

Analysis of Sequential and Navigational Patterns

Opinion Mining

Behavior Prediction