The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns

ROBERT COOLEY
KXEN, Inc.

The discipline of Web Usage Mining has grown rapidly in the past few years, despite the crash of the e-commerce boom of the late 1990s. Web Usage Mining is the application of data mining techniques to Web clickstream data in order to extract usage patterns. Yet, with all of the resources put into the problem, claims of success have been limited and are often tied to specific Web site properties that are not found in general. One reason for the limited success has been a component of Web Usage Mining that is often overlooked—the need to understand the content and structure of a Web site. The processing and quantification of a Web sites content and structure for all but completely static and single frame Web sites is arguably one of the most difficult tasks to automate in the Web Usage Mining process. This article shows that, not only is the Web Usage Mining process enhanced by content and structure, it cannot be completed without it. The results of experiments run on data from a large e-commerce site are presented to show that proper preprocessing cannot be completed without the use of Web site content and structure, and that the effectiveness of pattern analysis is greatly enhanced.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*data mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*information filtering*

General Terms: Experimentation, Measurement

Additional Key Words and Phrases: Data mining, World Wide Web, Web usage mining

1. INTRODUCTION

The discipline of Web Usage Mining has grown rapidly in the past few years, despite the crash of the e-commerce boom of the late 1990s. Web Usage Mining is the application of data mining techniques to Web clickstream data in order to extract usage patterns. Instead of questioning the need for sophisticated Web site analysis tools, those responsible for maintaining Web sites are

The work discussed in this article was performed while the author was attending the University of Minnesota and was supported by National Science Foundation (NSF) grant EHR-9554517.

Author's address: KXEN, Inc., 650 Townsend Street, Suite 300, San Francisco, CA 94103-6232; email: rob.cooley@kxen.com.

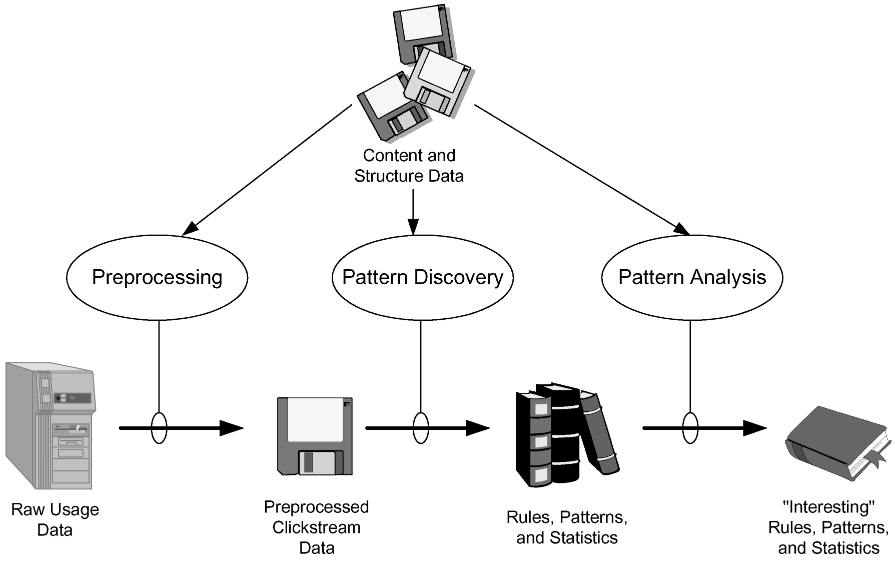
Content and Structure Data
Preprocessing
Pattern Discovery
Pattern Analysis
Raw Usage Data
Preprocessed Clickstream Data
Rules, Patterns, and Statistics
"Interesting" Rules, Patterns, and Statistics

Fig. 1. High level *Web Usage Mining* process.

clamoring for them. Yet, with all of the resources put into the problem and the obvious demand, claims of success have been limited, and are often tied to specific Web site properties that are not found in general. Academic [Cooley et al. 1999; Spiliopoulou and Faulstich 1998] and commercial (http://www.netiq.com, http://www.accrue.com) tools for parsing, cleaning, and sessionizing Web server logs are abundant, as are data mining algorithms for discovering patterns or trends from a clean set of usage data [Agrawal and Srikant 1994; Chen et al. 1996; Getoor and Sahami 1999]. One reason for the limited success has been a component of Web Usage Mining that is often overlooked—the need to understand a Web sites content and structure. The processing and quantification of a Web sites content and structure for all but completely static and single frame Web sites is arguably one of the most difficult tasks to automate.

Figure 1 shows the high-level steps involved in the Web Usage Mining process. The content and structure of a Web site are used as inputs to every major step of the process. For the purposes of this paper, the following definitions will be used for the various types of data involved with Web Usage Mining:

—*Content*. The *real* data in the Web pages, that is, the data the Web page was designed to convey to the users. This usually consists of, but is not limited to, text and graphics.

—*Structure*. Data that describes the organization of the content. *Intrapage* structure information includes the arrangement of various HTML or XML tags within a given page. The principal kind of *interpage* structure information is hyperlinks connecting one page to another.

—*Usage*. Data that describes the pattern of usage of Web pages, such as IP addresses, page references, and the date and time of accesses. Typically, the usage data comes from an Extended Common Log Format (ECLF) Server log.

The information provided by the data sources listed above can all be used to construct a data model consisting of several data abstractions, notably *users, page views, click-streams*, and *server sessions*. In order to provide some consistency in the way these terms are defined, the W3C Web Characterization Activity (http://www.w3c.org/WCA) has published a draft of Web term definitions relevant to analyzing Web usage. A *page view* is defined as all of the files that contribute to the client-side presentation seen as the result of a single mouse "click" of a user. A *click-stream* is then the sequence of page views that are accessed by a user. A *server session* is the click-stream for a single visit of a user to a Web site. A brief overview of the necessary steps for processing Web server logs will be provided in Section 5 and further details can be found in Cooley et al. [1999] and Cooley [2000].

Processing the structure and content of a Web site are interrelated tasks. The answer to the question of what links are available from a given page view depends on how the page view is defined. The degree of difficulty in performing content and structure processing is highly dependent on the technology used to create the Web site content. Conceptually, the problem is simple—create a map or graph of the Web site. However, for dynamically generated Web sites, especially ones utilizing personalization technologies, the question of equality for served content is not trivial. Jeff Bezos, the CEO of Amazon.com, has been quoted as saying, "The potential is there to completely redecorate your storefront for every customer that comes to your site." This means that the design or content of a "home page" for a Web site may be different depending on the user. Should different versions of the home page be considered to be different pieces of content for analysis purposes? A simple text matching algorithm for the served files may result in an explosion in the number of "unique" pieces of content in a site map. It may even result in a forest of site maps, instead of a single graph.

There are two major tasks associated with content and structure processing–determining what constitutes a unique page file, and determining how to represent the content and structure of the page file in a quantifiable form. The first task relates to the discussion in the paragraph above. Once the first task is solved, the structure and content of a Web site need to be encoded in a way that is useful for performing the various steps of Web Usage Mining. The task of defining a unique page file in a way that is semantically meaningful is highly dependent on the Web site. In current practice, defining the site content is a manual process that not only depends on the technology used to create the Web site, but also the goals of the analysis.

This article shows that, not only is the Web Usage Mining process greatly enhanced by content and structure, it cannot be completed without it. Specifically, this article:

- Identifies the problems associated with processing Web site content and structure for the purpose of Web Usage Mining.
- Describes how these problems are handled by the Web Site Information Filter (WebSIFT) system.

—Presents experimental results from the WebSIFT system demonstrating the benefits and necessity of incorporating Web site content and structure into the Web Usage Mining process.

The rest of this article is organized as follows: Section 2 discusses related work, Sections 3 and 4 explain the challenges of content and structure processing, Section 5 gives a brief overview of Web Usage Mining, and Section 6 presents the results of experiments using the WebSIFT system. Finally, Section 7 provides a conclusion.

2. RELATED WORK

Monitoring and understanding how the Web is used is an active area of research in both the academic and commercial worlds. A survey of Web Usage Mining projects has been published in Srivastava et al. [2000]. Several projects [Buchner and Mulvenna 1998; Cooley et al. 2000; Pirolli et al. 1996; Spiliopoulou et al. 1999; Heer and Chi 2001] have shown the usefulness of content or structure when performing Web Usage Mining; however, none have focused on the fact that the process cannot be completed without it. Applying data mining to the structure and content of Web sites is an interesting area of research in its own right. However, in the context of Web Usage Mining, the structure and content of a site is a critical input to preprocessing algorithms, can be used as a filter before and after pattern discovery algorithms, and can provide information about expected user behaviors for pattern analysis. Results from Web Structure and Content Mining projects, such as ParaSite [Spertus 1997], the authoritative source and hub work of Kleinberg [Gibson et al. 1998], LIRA [Balabanovic and Shoham 1995], WebKB [Craven et al. 1998], or WebACE [Moore et al. 1997] can be used as part of the preprocessing phase to cluster or classify Web pages in order to enhance a Web Usage Mining project. A review of Web Content and Web Structure mining research can be found in Kosala and Blockeel [2000].

3. WEB SITE CONTENT

As will be shown in the next section, the hypertext link structure of a Web site naturally lends itself to a directed graph. However, quantifying the content in the page files is not as straightforward. Even for simple forms of analysis such as static aggregation of page usage, the preprocessing step of *Uniform Resource Identifier (URI) translation* or mapping must be performed. URI translation is the first piece of a larger preprocessing step referred to as *page view identification* [Cooley et al. 1999]. For static Web sites, URI translation is relatively simple. There is a one-to-one mapping between URIs and content. However, for dynamic sites, several URIs can map to a single piece of content, and several pieces of content can potentially map to the same URI.

A common example of more than one URI mapping to a single piece of content is an embedded session identifier. To facilitate the tracking of sessions, a session ID is often embedded in the dynamic URIs served by a Web site. For this case, a simple string match would end up declaring every URI to be unique, except for repeat visits to a page within a single session. Another example is when

Table I. Dynamic URI Translation Example

Sample URIs	www.umn.edu/script?ID=8376&loc=Store&dest=page1 www.umn.edu/script?ID=8376&loc=Store&dest=prod&prod=item1 www.umn.edu/script?ID=4596&loc=Store&dest=page1 www.umn.edu/script?ID=9432&loc=Store&dest=prod&prod=item2
Regular Expressions	.*dest=prod&prod=\(.*)\ .*dest=\(.*)\
Translations	page1 item1 page1 item2

referring information is embedded in a URI. For this case, the same piece of content will have a different URI for each link to the content. The single content to multiple URI problem can often be solved through the use of an ordered set of regular expressions. The set of regular expression indicate which sections of a URI are responsible for identifying the content, and which sections are used for other purposes such as session tracking or user specific variables. This is the solution used by the WebSIFT system. A simple example is shown in Table I. For the example, only two simple regular expression are required, but many sites require dozens of expressions to correctly translate all of the possible URIs. While the expressions may look simple once they are written, an understanding of the site content is required in order to write them. Currently, there are no tools available for automatically creating a URI translation scheme for a Web site. This method requires that all of the necessary information for translating a URI is present in the logged string. As discussed in the next paragraph, this is not always the case.

If a single URI maps to several pieces of content, the situation cannot be resolved using Web server logs. Examples of when this can occur are dynamic pages formed from hidden POST requests, or pages formed from content servers that maintain state for each user or session. For these cases, usage data must be collected from a source other than the Web server. Even packet sniffer technologies, such as the kind employed by Accrue (http://www.accrue.com) or WebTrends (http://www.netiq.com) are not capable of detecting this problem if state is being maintained by the content server. A logging mechanism that is "content server aware" is required for this case, such as the solution provided by Blue Martini [Ansari et al. 2000].

In addition to URI translation, content hierarchies, page classifications, or page clusters are extremely useful for filtering the input to the pattern discovery stage, or providing context for pattern analysis. Often, a content hierarchy exists for a Web site in the form of a product hierarchy that can be made use of once the URI translation is completed. For Web sites without an existing hierarchy, the two techniques that seem to be the most useful for supporting Web Usage Mining are clustering and classification of the page files based on the text in the file. While it would be interesting to take into account the graphics and other multimedia files, research into automated multimedia clustering and classification is still relatively new and very resource intensive. In order to make use of any data mining algorithm, the text in the page files must first be preprocessed. Once the text is preprocessed, any number of clustering or classification algorithms can be run. Unless XML tags are used to add semantic structure to a Web site, the problem of quantifying page files is essentially the same as the quantification of unstructured text documents. While some experiments have attempted to make use of HTML display tags to infer semantic meaning, these methods do not work in the general case. *Wrapper* projects, such as Gruser et al. [1998] and Sahuguet and Azavant [1999], have had some success in using regularities in the HTML tag structure of Web sites, but methods such as these are extremely brittle, and any deviations from the expected tag structure corrupts the algorithm. The WebKB project [Craven et al. 1998] goes beyond simple wrappers by utilizing what are referred to as *sufficiently redundant features* from the content of Web pages to add to the information gained from the HTML tag structure. However, the WebKB project relies on the existence of a domain specific ontology for the content information.

4. WEB SITE STRUCTURE

The structure of a Web site is created by the hypertext links between page views and the frame and image tags that populate a particular page view (referred as intrapage structure). Several usage preprocessing steps can not be completed without the site structure. In addition, the site structure is useful for identifying potentially interesting rules. As has been described in Cooley et al. [1999] and Cooley [2000], the Web site structure is required for page view identification, and may be needed to identify users in the absence of a unique user identifier such as cookies. Due to the presence of *frames*, the number of potential page views for a Web site can be vast. It is not uncommon for every page view on a site to consist of two or three frames. Often, there is a top frame for general site navigation, a left frame for more specific navigation, and a main frame with some content. Assuming that there are x top frames, y left frames, and z main frames for a site, the number of unique page views could be as high as $x * y * z$. This number can quickly become intractable. Therefore, the structure of a Web site needs to be stored as a set of frames, $\mathcal{F}$, with a list of associated links and targets. A target is the page area that the link should be loaded into in the browser display. To further complicate the situation, a single link can lead to the replacement of between one and all of the frames in a page view. A formal definition of a site structure map for use in Web Usage Mining is as follows, where $\mathcal{M}$ is a site map, h_i is an HTML file, r is a link type, and g_i is a target area:

$$\mathcal{M} = [<\mathcal{F}_1; \ldots ; \mathcal{F}_n>] \quad (1)$$

$$\mathcal{F} = \{h_f, \mathcal{L}_1, \ldots, \mathcal{L}_m\} \quad (2)$$

$$\mathcal{L} = <r, (h_1, g_1)| \cdots |(h_p, g_p)> \quad (3)$$

The link type indicates how the page file will be requested from the Web server. The most common is *get*, which means the standard GET method will be used to request the page file. Other common types include *post*, *hidden post*, *frame*, *ftp*, and *mail*. The post and hidden post types both use the POST HTTP method for sending data back to the Web server. Although the POST method is theoretically a one way transfer of data from the client to the server, in practice,

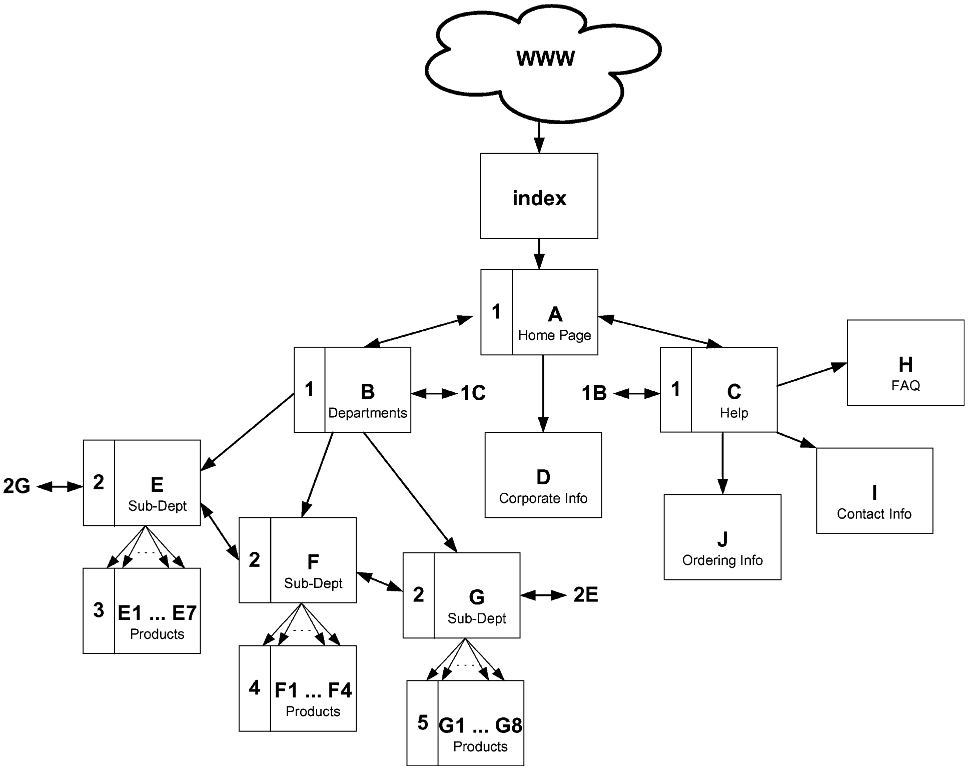

Fig. 2. Example Web site.

an application such as a content server will send a page file to the client as a response to the data sent in the POST. The difference between a regular post and a hidden post is in how the client data are sent back to the Web server. A regular post appends the data to the URI in CGI format (a series of text name/value pairs delimited by "&"). A hidden post passes the data in the HTTP header. This is an important distinction since the URI is logged as part of the CLF or ECLF formats, but hidden POST data are not. If the source of the usage data contains the hidden POST parameters, such as a packet sniffer log, then there is no distinction between the *post* and *hidden post* types. The frame type refers to the use of the HTML "frame" tag. In this case, page files listed as frames are automatically requested from the Web server. The ftp and mail link types are just two common types that do not lead to HTML page files being served. An example of a very simple Web site is given in Figure 2. The site is essentially a tree structure, with each page view consisting of one or two frames. The site map for Figure 2 is as follows, where the possible frame areas are *left, main,* and *top*:

$$\begin{aligned}\mathcal{M} = {} & [\{index, (frame, 1, left|frame, A, main)\};\\ & \{1, (get, A, main), (get, B, main), (get, C, main)\};\\ & \{2, (get, E, main), (get, F, main), (get, G, main)\};\end{aligned}$$

$$
\begin{aligned}
&\{3, (get, E1, main), \ldots, (get, E7, main)\};\\
&\{4, (get, F1, main), \ldots, (get, F4, main)\};\\
&\{5, (get, G1, main), \ldots, (get, G8, main)\};\\
&\{A, (get, D, top)\};\\
&\{B, (get, 2, left|E, main), (get, 2, left|F, main), (get, 2, left|G, main)\};\\
&\{C, (get, H, top), (get, I, top), (get, J, top)\};\\
&\{D\};\\
&\{E, (get, 3, left|E1, main), \ldots, (get, 3, left|E7, main)\};\\
&\{F, (get, 4, left|F1, main), \ldots, (get, 4, left|F4, main)\};\\
&\{G, (get, 5, left|G1, main), \ldots, (get, 5, left|G8, main)\};\\
&\{E1\}; \ldots; \{E7\}; \{F1\}; \ldots; \{F4\}; \{G1\}; \ldots; \{G8\};\\
&\{H\}; \{I\}; \{J\}]
\end{aligned}
\tag{4}
$$

The site map lends itself very naturally to an object-oriented schema, where a site object contains a list of frame objects, and each frame object is made up of a file, plus a list of link objects. In order to form page views from the site map, an initial page view needs to be specified. In the case of the example site, the natural initial view is the home page, `1-A`, which will be automatically served as a result of a request for `index.html`. By default, most Web servers fulfill a request for a directory with a file named index.html or home.html. e.g. A request for www.sample.com will get index.html from the base directory for the Web site. All other page views for the site can be formed by replacing one or more frame areas with any of the candidate links in the active frames. For example, page view `1-B` is formed by following the link to `B` from frame `1`. Page view `2-F` is formed from a single link in `B`, that replaces both frame `1` with `2` and frame `B` with `F`. With 5 left frames and 25 main frames, a maximum of 125 page views could be theoretically formed. However, since in this simple example each main frame is only accessible from one left frame, there are only 25 page views resulting from combinations of left and main frames. With the four single-frame page views, this site has 29 possible page views.

There are several options involving page views and page files when preparing for pattern discovery. For a multiframed site, patterns can be discovered based on page views, the individual page files, or the set of page files for a particular frame location. Page-view-based patterns prevent the problem of discovering patterns that simply represent the different components of the same page view. For example, the "Home Page" view may be made up of two page files, `HomeLeft.html` and `HomeBody.html`. Performing pattern discovery with page view identifiers such as "Home Page" prevents cluttering the results with associations between `HomeLeft.html` and `HomeBody.html`. However, in the case where a page file in one frame may be associated with several page files in another frame, page view pattern discovery has two potential problems. Because the occurrences of a page file may be spread across several page views, each page view with the page file may not appear enough times to meet a minimum support criteria, whereas the individual page file may have sufficient support.

Second, patterns can be discovered that only contain page views with the same page file. For example, consider a product page file that can be viewed along with ten different left navigation page files. This means that there are ten page views that differ only by the left navigation frame. The product page may appear in 10% of the sessions, but when split among the ten page views, may never appear in more than 1% of the sessions. Even if all of the page views meet the minimum support, it is likely that a pattern relating all ten of the page views will be discovered. This is not much better than discovering patterns of page view components.

A third option is to perform pattern discovery on page files from a single frame, such as the “main” or ”body” frame. This prevents the discovery of page view components in addition to preventing the dilution of a page file among multiple page views. The problem with patterns discovered for a single frame location comes during the pattern analysis phase. When using a technique such as pattern filtering based on site structure, patterns that do not contain all of the page view components can be missing the links that tie the page files together. Consider a set of product pages that are all listed on the same left navigation frame. Taken individually, none of the product pages are linked to each other, but when the entire page views are considered, they are all doubly linked. For single-frame sites, the page files and page views are equivalent, preventing any of the problems discussed above.

5. WEB USAGE MINING

Figure 3 shows the details of the Web Usage Mining process for the WebSIFT system. Web Usage Mining preprocessing consists of converting the usage, content, and structure information contained in the various available data sources into the data abstractions necessary for pattern discovery. Once the raw usage data has been preprocessed into server sessions, a number of techniques can be used to discover patterns such as association rules [Agrawal and Srikant 1994], clusters of similar pages or users, or sequential patterns [Mannila et al. 1995]. The discovered patterns can then be used for a variety of applications, such as:

—*Site Design*—Reorganization of the link structure or content of the pages to reflect actual usage.

—*Business Marketing Decision Support*—Determination of common behaviors or traits of users who perform certain actions, such as purchasing merchandise.

—*Personalization*—Customization of page views based on information gained about each user. This can include dynamic pricing or real-time promotions to encourage “cross-sells” and “up-sells.”

—*Usability Studies*—Determination of interface quality.

—*Security*—Detection of “unusual” accesses to secure data.

—*Network Traffic Analysis*—Determination of equipment requirements and data distribution in order to efficiently handle site traffic.

Finally, pattern analysis is performed to convert the discovered patterns into useful or interesting knowledge.

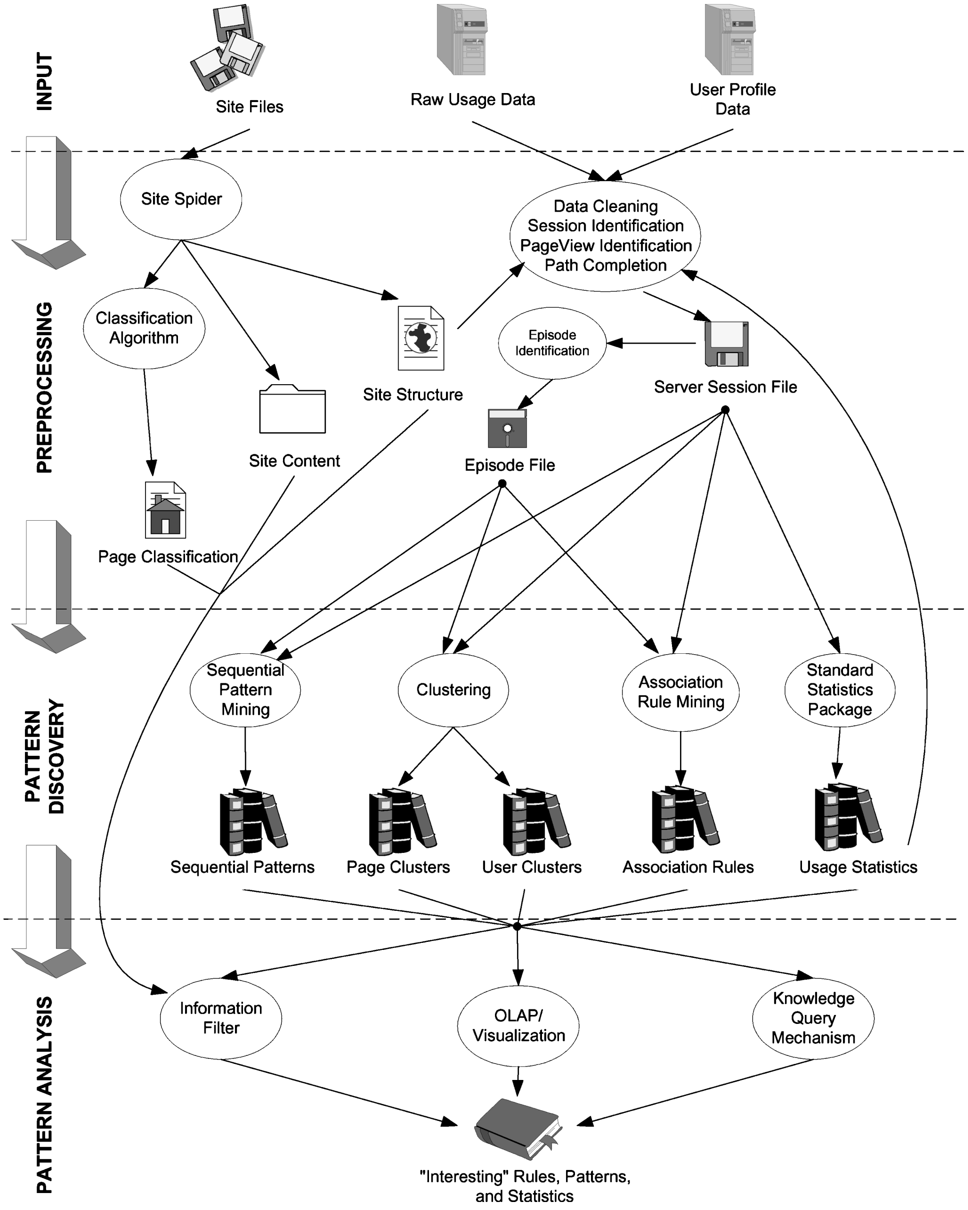

Fig. 3. *Web Usage Mining* process.

5.1 Preprocessing

The practical difficulties in performing preprocessing are a moving target. As the technology used to deliver content over the Web changes, so do the preprocessing challenges. While each of the basic preprocessing steps remains constant, the difficulty in completing certain steps has changed dramatically as Web sites have moved from static HTML served directly by a Web server, to dynamic scripts created from sophisticated content servers and personalization tools. Both client-side tools (e.g., browsers) and server-side tools (e.g., content servers) have undergone several generations of improvements since the inception of the Web. *Data cleaning* is a site specific step that involves mundane tasks such as merging logs from multiple servers and parsing the log into data fields. Typically, graphics file requests are stripped out at this stage. Next, user and *session identification* is performed through one of several methods, the most common being the use of cookies, user registration, or embedded sessiond IDs in the URIs. *Page view identification* determines which page file requests are part of the same page view and what content was served. As discussed in the previous two sections, this step is highly dependent on knowledge of the Web site structure and content. Finally, *path completion* fills in page references that are missing due to local browser caching. This step differs from the others in that information is being added to the log. Each of these tasks are performed in order to create a server session file that will be used as input to the pattern discovery phase. Note that the server session file is usually only an estimate of what actually occurred due to techniques that obscure the data collection such as proxy servers and caching that are common in today's browsing environment. Details of the heuristics and algorithms used by the WebSIFT system to handle these problems are contained in Cooley [2000].

Episode identification is an optional preprocessing step that can be performed after the required preprocessing steps. An episode is defined by the W3C as a *semantically meaningful* subset of a user session. Any number of definitions for episodes can be manually created for a Web site based on the content. For example, episodes of only the product pages viewed on an e-commerce site, or sports page views from a news site can be created. Another popular episode definition is the subset of page views that deal with the "shopping cart checkout" process. E-commerce sites often use separate definitive tracking mechanisms available through commercial content servers for tracking critical events such as this.

5.2 Pattern Discovery

While the WebSIFT system can make use of a number of pattern discovery algorithms, the two that were used to provide the Section 6 experimental results were Apriori [Agrawal and Srikant 1994] for frequent itemsets and Concept Indexing [Karypis and Han 2000] for clusters.

5.2.1 *Frequent Itemsets.* Only a subset of the information contained in the server sessions or episodes is necessary for frequent itemset discovery. The order and number of occurrences of a page view or page file in a session is not required. Therefore, sessions must be stripped down to a list of unique session ID/page pairs. A minimum support cutoff must be identified in order to limit the number of discovered patterns and the computation time. The support of an itemset is the fraction of the total sessions that the set appears in together. Support, S, is defined as follows for a set of n items, where D is the data set and i is an item:

$$S = \frac{count(\{i_1, \ldots, i_n\} \in D)}{count(D)}. \tag{5}$$

Once the frequent itemsets are discovered, *lift* can be used to objectively rank the sets. Lift [Brin et al. 1997] is defined as the *support* of a frequent itemset divided by the probability of all of the items appearing together in a set if the items are randomly and independently distributed:

$$L = \frac{S(i_1, \ldots, i_n)}{\prod_{j=1}^{n} S(i_j)}. \tag{6}$$

An lift measure that is greater than one indicates that the items in the itemset appear together more often than what would be expected through a random distribution. Items that have very high support will often appear in frequent itemsets simply because they randomly appear together in sessions. However, these itemsets tend to have a low lift measure. If the itemsets are broken out into association rules, the confidence of each rule can be calculated. The confidence of a rule is the fraction of sessions where the subsequent is present if the antecedent is also present. Typically, association rules are limited to single item antecedents (otherwise, the number of rules for an n item itemset can be much greater than n). Confidence is defined as follows for a rule $i_a \Rightarrow \{i_{s1}, \ldots, i_{sn}\}$:

$$C = \frac{S(i_a, i_{s1}, \ldots, i_{sn})}{S(i_a)}. \tag{7}$$

The support, confidence, and lift measures will be used for the generation of experimental results in Section 6.

5.2.2 *Concept Index Clusters.* The full Concept Indexing algorithm is actually a method for dimensionality reduction in order to achieve better performance for traditional classification algorithms. However, the first step of the algorithm is to compute k clusters from weighted vectors. For Web Usage Mining, the vectors are made up of page view accesses, with the weights being access counts for a session. The clustering algorithm starts with k seeds, and then uses recursive bisection to compute the final clusters. The full details of the algorithm are reported in Karypis and Han [2000].

5.3 Pattern Analysis

The primary form of pattern analysis used by the WebSIFT system is the *information filter*, which is broken into two parts—an *objective filter* and a *subjective filter*. The objective filter handles the numeric measures that are associated with the various pattern discovery methods—such as support and lift. The subjective filter handles beliefs about the usage of pages in a Web site formed by analyzing the content and structure. The WebSIFT information filter uses Dempster–Shafer (DS) methods to encode and combine specific beliefs about the use of a Web site. For a belief $\mathcal{B}$, evidence collected for or against $\mathcal{B}$ can be used to form a *support pair*, $[s_n, s_p]$, where:

$$\begin{aligned} s_n &= \text{necessary support of } \mathcal{B} && (8)\\ s_p &= \text{possible support of } \mathcal{B} && (9)\\ (1-s_p) &= \text{necessary support of } \neg\mathcal{B} && (10)\\ (1-s_n) &= \text{possible support of } \neg\mathcal{B} && (11)\\ (s_p-s_n) &= \text{uncertainty of } \mathcal{B}. && (12)\end{aligned}$$

The values of s_n and s_p must satisfy the constraints:

$$s_n + (1-s_p) \leq 1 \qquad (13)$$

$$s_n \geq 0, s_p \geq 0. \qquad (14)$$

As an example, assume that evidence has been collected about the belief B, that Web pages `A` and `C` are related. If all of the evidence is in support of B, the DS pair is [1, 1]. On the other extreme, if all of the evidence is against B, the DS pair is [0, 0]. If the data leads to a 25% degree of belief that B is true, and a 40% degree of belief that B is false, then [0.25, 0.6] would represent the appropriate DS pair. This says that the degree of uncertainty about B is 35%. Finally, if there is no evidence pertaining to B, the DS pair is [0, 1], giving an uncertainty of 100%. Independent of the type of the source for generating a DS pair, pairs can be combined per Dempster's rule of combination [Shafer 1976] to obtain a single DS pair per belief. The basic rule is as follows:

If $\mathcal{B}$:$[s_{1n}, s_{1p}]$ AND $\mathcal{B}$:$[s_{2n}, s_{2p}]$ are two independent DS pairs from different sources about belief $\mathcal{B}$, then conclude $\mathcal{B}$:$[s_n, s_p]$, where

$$k = 1 - s_{1n}(1-s_{2p}) - s_{2n}(1-s_{1p}) \qquad (15)$$

$$s_n = [s_{1n}s_{2n} + s_{1n}(s_{2p}-s_{2n}) + s_{2n}(s_{1p}-s_{1n})]/k \qquad (16)$$

$$\begin{aligned}1-s_p = [&(1-s_{1p})(1-s_{2p}) + (s_{1p}-s_{1n})(1-s_{2p}) +\\ &(s_{2p}-s_{2n})(1-s_{1p})]/k \qquad (17)\end{aligned}$$

5.3.1 *Subjective Filter.* For Web Usage Mining, the assumption is that content and structure data can be used as surrogates for the Web site designer's domain knowledge. Links between pages provide evidence in support of those pages being related. The stronger the topological connection is between a set of pages, the higher the value of s_n is set for the evidence pair. Evidence pairs based on the site content can also be automatically generated by looking at content similarity, and assigning values of s_n and s_p based on the calculated "distance" between pages. Table II gives some examples of the types of interesting beliefs that can be identified in the Web Usage Mining domain using the subjective filter.

Notice that domain knowledge in the examples is fairly fuzzy, and no specific values are given for the beliefs. This is because while it is fairly easy to identify beliefs such as, "pages `A` and `B` will be used together often," it is much more difficult to reliably estimate the exact parameters of a frequent itemset, cluster of pages, or some other usage pattern in advance. Especially since the number

Table II. Information Filter Examples

Mined Knowledge	Domain Knowledge Source	Interesting Belief Example
General Usage Statistics	Site Structure	The head page is not the most common entry point for users
General Usage Statistics	Site Content	A page that is designed to provide content is being used as a navigation page
Frequent Itemsets	Site Structure	A set of pages is frequently accessed together, but not directly linked
Usage Clusters	Site Content	A usage cluster contains pages from multiple content categories

Table III. Combination and Interestingness of Boundary Cases

#	New	Existing	Combined	δ	$1-k$	$\mathcal{I}_s$
1	[0.1,0.1]	[0.9,0.9]	[0.5,0.5]	0.82	0.57	1.39
2	[0.9,0.9]	[0.1,0.1]	[0.5,0.5]	0.82	0.57	1.39
3	[0.1,0.1]	[0.1,0.9]	[0.1,0.1]	0.10	0.80	0.90
4	[0.9,0.9]	[0.1,0.9]	[0.9,0.9]	0.10	0.80	0.90
5	[0.1,0.1]	[0.1,0.1]	[0.01,0.01]	0.17	0.13	0.30
6	[0.9,0.9]	[0.9,0.9]	[0.99,0.99]	0.17	0.13	0.30
7	[0.1,0.9]	[0.1,0.9]	[0.17,0.83]	0.02	0.10	0.12
8	[0.1,0.9]	[0.1,0.1]	[0.1,0.1]	0.1	0.00	0.10
9	[0.1,0.9]	[0.9,0.9]	[0.9,0.9]	0.1	0.00	0.10

and values of the parameters vary greatly with the pattern discovery method. The Web site structure and content will indicate which pages are *likely* to be used together (or not used together), but not necessarily how often, or to what numerical certainty. This is why the DS inference mechanism has been chosen for the subjective filter. The issue of estimating parameter values (or even deciding which of several available parameters should be estimated at all) is avoided by attaching the weaker DS semantics to the belief pairs.

The subjective interestingness, $\mathcal{I}_s$ of a belief after the introduction of new evidence is defined as follows, where k is as defined in Eq. (15), and δ is the Euclidean distance between the existing and combined belief pairs :

$$\mathcal{I}_s = \delta + (1 - k) \tag{18}$$

In the simplest case, all evidence is either 100% for a belief, 100% against a belief, or there is no evidence about a belief. However, belief pairs of [1,1] or [0,0] are special cases because they can not be revised by the introduction of new evidence. While this is a desirable quality in the case of "hard" beliefs that are known to be true, it is not a good idea for "soft" beliefs that are subject to change. Therefore, initial soft beliefs will bounded by [0.1,0.1] and [0.9,0.9] to allow for revision in the face of new evidence. (This does not prevent combined values from exceeding the bounds, as shown in Table III). There are nine different "boundary" cases that can occur when combining evidence from domain and usage sources for soft beliefs. These are shown in Table III along with amount of subjective interestingness, $\mathcal{I}_s$.

The first two boundary conditions are for fully conflicting evidence, which gives the highest measure of interestingness. The next two conditions are where there is no existing knowledge about a belief (full ignorance) and the new evidence is strongly positive or negative. Boundary conditions five and six show evidence that is in complete agreement, and the last three conditions are where the new evidence has nothing to say about the belief. Notice that both the change and conflict terms are necessary to create an interestingness measure that makes intuitive sense. The change term (δ) ends up ranking the conditions with evidence in complete agreement (conditions 5 and 6) as more interesting than the conditions where new strong evidence is replacing ignorance (conditions 3 and 4). However, use of the conflict measure alone would rank the strong new evidence conditions over the conditions of complete conflict (conditions 1 and 2). Another property of the DS semantics and combination rule is that the evidence gets appropriately weighted. As discussed, multiple sets of evidence in favor of a belief will result in a belief pair approaching [1,1] (demonstrated in condition 6 of Table III). The closer an existing belief pair is to the upper or lower bounds, the less a single set of conflicting set of evidence will change the combined pair. This prevents one anomalous set of evidence from distorting the overall belief. If the anomalous evidence becomes the norm, the belief pair will start to be pulled more and more towards the new "corrected" values.

5.3.2 *Objective Filter.* The *objective filter* is activated for subsequent sets of evidence pertaining to discovered usage patterns. While the subjective filter picks up interesting additions, deletions, and large scale changes to the discovered patterns, the objective filter is designed to monitor changes in the actual parameters that are part of any pattern discovery algorithm. For example, consider an association rule relating pages A and B with some support, S and confidence, C. Even if the association rule is regularly discovered each time the usage patterns are analyzed, significant changes to the support or confidence should also be considered interesting. A measure like support may not factor into the calculation of an evidence pair supporting the belief that pages A and C are related through usage. However, a significant drop or increase in the support is certainly of interest. Changes in the parameters associated with a pattern discovery method are best tracked with statistical inference. Maintenance of a mean and variance for each parameter is relatively easy and allows for testing of a null hypothesis that the existing mean and the new value are equivalent. In Silberschatz and Tuzhilin [1996] it is pointed out that hypothesis testing cannot handle the situation where it is *expected* that a value is going to change, only the case where the expectation is for a constant mean. However, if a change is expected by the analyst for some reason, this can always be formulated as a subjective belief, and handled in the subjective filter instead.

5.3.3 *Evidence Quantification.* In order to make use of the subjective information filter, the different sources of evidence need to be transformed into belief pairs. By the nature of *subjective* evidence, there is no one correct method for quantifying the evidence. For belief pairs created by the analyst that are not a direct result of structure or content evidence, the quantification is whatever the analyst feels comfortable with. The task of converting fuzzy concepts, such as *unlikely* or *extremely likely*, into numerical ratings has been used successfully in a number of research efforts [Bonissone and Decker 1986; Resnik et al. 1994]. However, the main goal of the information filter is to use the existing data sources to automatically generate belief pairs in order to provide standardized measures for similar situations and avoid the labor associated with manual quantification.

The appropriate method for quantifying usage evidence is completely dependent on the type of pattern discovered and the algorithm used for the discovery. The number and types of parameters associated with pattern discovery methods is vast. However, since the beliefs are in the form of what sets of pages will be related through usage, any discovered usage pattern is a direct measure of the true relation. To create a single evidence pair for a frequent itemset, the maximum confidence from the corresponding set of association rules can be used:

$$B_u^1[s_n, s_p] = [C_{max}, C_{max}], \tag{19}$$

where C_{max} is taken from the set of confidence levels, $\{C_1, \ldots, C_n\}$, associated with an n item frequent itemset.

Unlike frequent itemsets, usage clusters do not necessarily represent actions that occur within single sessions. A set of pages from a usage cluster may not appear in any one session. Also, the available parameters for creating evidence pairs depends on the algorithm used to create the clusters. There are no universal parameters like support and confidence to help construct an evidence pair. The beliefs that relate usage cluster evidence are that pages related by content or structure will be represented by a single usage cluster. If a usage cluster maps to several sets of pages related by content/structure, or several usage clusters map to the same content/structure cluster or category, this would be unexpected and therefore interesting. These beliefs will be of the following form for a set of usage clusters, $\mathcal{U}$ and domain sets $\mathcal{D}$:

$$\begin{aligned} &B[s_n, s_p] \text{ where} \\ s_n = \ & \text{evidence for } \left\{\mathcal{U}_i \bigcap \mathcal{D}_j\right\} = \begin{cases} \mathcal{D}_j & \text{for } i = j \\ \emptyset & \text{for } i \neq j \end{cases} \\ 1 - s_p = \ & \text{evidence for } \left\{\mathcal{U}_i \bigcap \mathcal{D}_j\right\} \neq \begin{cases} \mathcal{D}_j & \text{for } i = j \\ \emptyset & \text{for } i \neq j \ . \end{cases} \end{aligned} \tag{20}$$

The full details and justification for the information filter and evidence quantification can be found in [Cooley 2000].

6. EXPERIMENTAL RESULTS

Data from a large e-commerce site was used to run experiments with the WebSIFT system. The data set consisted of a single day's traffic, which had 608,823 page file requests once the graphics requests were removed. A simplified map of the Large E-commerce site is shown in Figure 4. The map is not complete, but shows the core set of pages and links that are used most often. The page files outlined in bold are the key pages in the site related to the display and

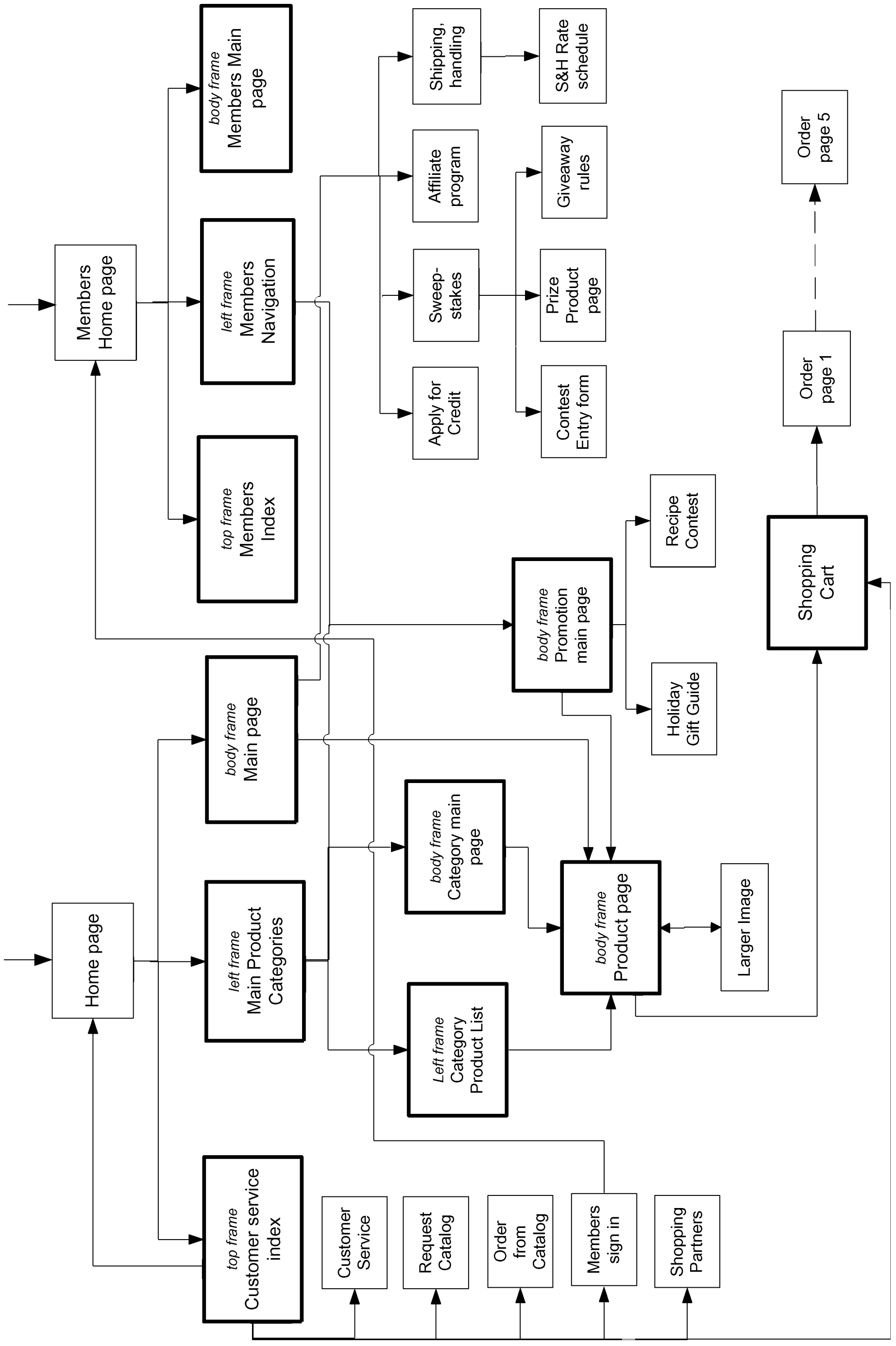

Fig. 4. Large E-commerce site map.

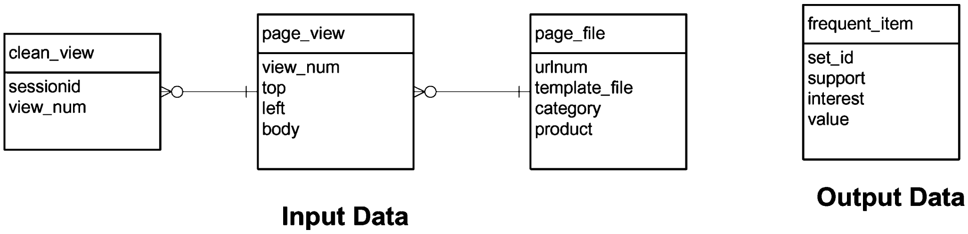

Fig. 5. Schema for frequent itemsets.

sale of products. There are two main entry points, one for members and one for anonymous users. The site includes 571 different categories and 5757 different products leading to between 40,000 and 50,000 distinct page views each day.

Two sets of experiments were run on the data. The first works with frequent itemsets to demonstrate that it is impossible to perform any Web Usage Mining without some use of Web content or structure. The second experiment demonstrates how the use of Web content in the form of a hierarchy can be used in the pattern analysis stage to pinpoint subjectively interesting results among many discovered patterns.

6.1 Frequent Itemsets

The input for this experiment, shown in Figure 5, is the output of the page view identification task for usage preprocessing. The data cleaning and session identification tasks were performed by parsing an ECLF log, removing all graphics requests, removing all log entries that did not contain an embedded session ID, and setting the sessionid field equal to the sessionid CGI value for the request. The output of this step yielded 50,581 sessions. The *template_file*, *category*, and *product* CGI values for both the request and referrer were replaced with the appropriate numeric identifier from the *page_file* table. The page view identification algorithm described in Sections 4 and 5 was then run to form the *page_view* table. All of the page views for the Web site have three frames, so each page view identified had a top, left, and body page file. The path completion heuristic was not called since the Apriori algorithm does not require the time or number of occurrences of an item within a session.

As discussed in Section 4, patterns can be based on the page view identifiers, page files from a single frame, or page files from multiple frames. For page view patterns, the *clean_view* data were the input to the pattern discovery algorithms. For single and multiframe patterns, the *clean_view* data were joined with the *page_view* data in order to access the individual components of the page views. The final table, *page_file* was used to translate the identifiers back into human understandable labels once the patterns had been discovered.

A 1% support threshold was used to generate frequent itemsets for the Web site with the Apriori algorithm for the page views, left frame, body frame, and both the left and body frame. An attempt was made to discover frequent itemsets using all three frames, which is what would occur if no knowledge of the site structure was available. However, since there are only two distinct page files for the top frame location, the Apriori algorithm was unable to complete at

Table IV. Large E-commerce Site Frequent Itemsets

Pattern Type	Total Number of Patterns	Number of Non-subset Patterns	Number of Distinct Page Files
Page View	5691	110	69
Left Frame	442	49	35
Body Frame	1173	89	43
Both Frames	76,655	121	78

Table V. Page View Frequent Itemsets

Page View	Support	Lift
Home Page, Sweepstakes Home Sweepstakes Entry	4.29	21.7
Home Page, Main Domestics	2.27	19.6
Home Page, Main Jewelry	2.00	27.9
Home Page, Main Apparel	1.91	24.5
Home Page, Main Home Accents	1.90	22.3
Home Page, Main Home Accents, Furniture	1.39	22.4
Home Page, Main Kids, Video Games, Nintendo	1.31	576.2
Holiday Sweepstakes, Holiday Sweepstakes Entry	1.17	65.2
Home Page, Main Sports	1.13	39.2
Order 1, Order 2, Order 3 Order 4, Order 5	1.10	3.23×10^6

the 1% support level. At 5% support, the use of all page files resulted in 4910 frequent itemsets, of which only 10 were not subsets of another itemset. The 10 nonsubset itemsets consisted mainly of components of the same page views.

The number of discovered patterns for each input type is shown in Table IV. Since any n item frequent itemset has $\sum_{x=1}^{n-2}\binom{n}{n-x}$ frequent subsets, the number of sets that are not subsets of any other frequent itemset are also shown.

Table V shows the top ten discovered page view rules ordered by support with an lift of at least one standard deviation above the average lift for all discovered non-subset itemsets.

Manual inspection shows that none of the discovered rules are particularly interesting from a subjective point of view. Most of the frequent itemsets confirm that the Home Page and one of the high level categories are related through usage. The order stream also appears in a few of the rules.

In order to show the value of site structure for pattern analysis, a 0.5% support threshold was used to generate frequent itemsets for both the left and body frame. The nontrivial, nonsubset itemsets were input into the subjective information filter and compared against the structure of the Web site. 155,951 frequent itemsets were discovered by the experiment, of which 209 were nontrivial and nonsubset itemsets. The five itemsets with the highest subjective interestingness are shown in Table VI and the five itemsets with the lowest subjective interestingness are shown in Table VII. The confidence value reported in the tables is the maximum confidence of the n possible association

Table VI. Frequent Itemsets with High Subjective Interestingness

Page View	Support	Maximum Confidence	Lift	Subjective Interest
Home Page, Main Holiday, Holiday Sweepstakes	0.55	10.2	10	1.26
Home Page, Main Holiday, Sweepstakes	0.56	11.5	2	1.23
Home Page, Main Housewares, Main Holiday	0.50	13.0	50	1.19
Home Page, Main Holiday, Member Order Status	0.51	9.35	21	1.18
Home Page, Main Domestics, Main Housewares, Main Electronics	0.50	12.44	25484	1.14

Table VII. Frequent Itemsets with Low Subjective Interestingness

Page View	Support	Maximum Confidence	Lift	Subjective Interest
Home Page, Main Electronics, Computers, Peripherals	0.59	95.4	11281	0.58
Home Page, Main Home Accents, Wall Decor, Misc.	0.56	96.8	100547	0.58
Home Page, Main Kids, Video Games, Handheld	0.57	98.6	31025	0.58
Home Page, Main Kids, Video Games, Nintendo	0.87	98.8	31102	0.58
Main Holiday, Main Domestics, Main Home Accents	0.50	13.0	41	0.32

rules that can be generated from an *n* item frequent itemset. The maximum confidence was used to calculate a value for the belief that the pages in a frequent itemset are related. This is then compared to a belief value calculated from the connectivity of the interpage structure.

All but one of the frequent itemsets represented connected pages, resulting in most of the patterns being assigned a low subjective interestingness measure. In other words, the pages were directly connected and therefore expected to be used together. However, the one unconnected frequent itemset was declared to be the least interesting of all the discovered patterns. This is because the

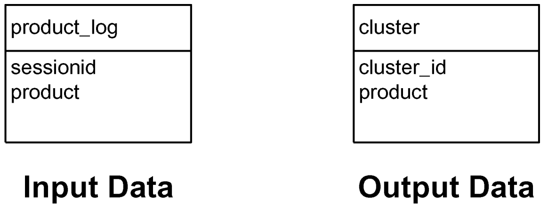

Fig. 6. Schema for product cluster experiments.

maximum confidence for the itemset was quite low, meaning that more often than not, the pages were not accessed together as expected due to the lack of connectivity. The rules with the highest subjective interestingness are all rules that have much lower confidence than would be expected due to the connectivity of the pages. The first two rules indicate that there is not very much cross-traffic between the sweepstakes pages and the Home Page. It turns out that there is a direct site entry point into the sweepstakes pages. A possible explanation for the low confidence is that users are coming to the site to register for the sweepstakes, and then leaving without browsing the site. This could be confirmed by performing sequential pattern analysis on the Home Page and sweepstakes pages. The last three rules are also potentially interesting due to a lack of cross-traffic between the page views. It is not immediately clear why this is the case for the third and fourth patterns, but a possible explanation for the fifth pattern is that users that browse Housewares and Domestics are not interested in Electronics (or vice-versa). The first four itemsets in Table VII are all levels of a single category hierarchy. The last itemset is the one unconnected itemset already mentioned. Note that four of the five subjectively interesting patterns have a very low lift measure, and they all have low support and confidence. The Table VII patterns have just the opposite characteristics—the confidence and lift measures are quite high.

6.2 Concept Clusters

The input and output schema for this experiment are shown in Figure 6. All product pages for the Web site are identified by a product code contained in the *product* CGI variable for a URI. The *product_log* table was created by selecting the unique non-null product values for each session ID. The Karypis and Han [2000] concept clustering method was used to discover product clusters from the *product_log* table.

One hundred and twenty-seven multiproduct clusters were discovered containing 1228 unique products. Twenty-seven of the 127 clusters were equally ranked with the highest amount of interestingness. For all 27 clusters, at least one of the products was in a completely different branch of the hierarchy than the other products in the cluster. Figure 7 shows five of these clusters. Fifty-five clusters contained products from the same low level category, and 21 clusters contained products from the same second-level category. In most cases, the height of the content hierarchy was four.

Most of the 27 interesting clusters are similar to the first cluster listed in Figure 7, where a quick manual inspection is sufficient to notice the difference

	Product	Category
1	10" Cuckoo Clock	Clocks
	Diamond Hoop Earrings	Ladies Jewelry
	9-Pc. Rose Clock Set	Clocks
	NBA Showtime	Sega
	Diamond Pendant Earrings	Ladies Jewelry
	GameBoy Colors	Nintendo
	N64, Resident Evil 2	Nintendo
	PlayStation; Jet Moto 3	Nintendo
	Super Nintendo w/Tetris Attack	Nintendo
	Super Nintendo;Space Invaders	Nintendo
2	40-Pc. Kitchen Linen Set	Kitchen Ensembles
	5-Pc. Inflatable Bath Set	Accessories
	Bathroom 3-Shelf Corner Caddy	Accessories
	Bathroom Spacesavers Scale	Scales
	Bedding for Kids	Children
	Camouflage Bedding	Children
	Country Kitchen Towel Sets	Table Linens
	Dazey Turbospa	Massagers
	Doral 4-Pc. Rug Set	Accessories
	Flannel Sheet Sets	Flannel Knits
	Graphics Chenille Spreads	Chenille
	Hearts Chenille Bedspreads	Chenille
	Kenya Comfortspread Bedding	Comfortspreads
	Chenille Bedroom Ensemble	Chenille
	Leaves Rugs	Oriental
	Poppin' Fresh Kitchen Linen Sets	Kitchen Ensembles
	Satin Comforter and Sheet Sets	Satin
	Scroll Chenille Bedspread	Chenille
	St. Croix Comforter and Sheet Set	Satin
	St. Croix Kitchen Textiles	Kitchen Ensembles
	St. Croix Satin Bed Set	Satin
3	Rechargeable Master Craft Drill	Cordless
	Black Decker Cordless Drill	Cordless
	Black Decker Cordless Screwdriver	Cordless
	Black Decker Socket/Bit Set	Cordless
	Black Decker Hand Power Tools	Power Tools
	Deluxe Tire Changing Kit	Accessories
	Master Craft 101-Pc. Tool Set	Hand Tools
	Master Craft Storage Chest	Hand Tools
	Master Craft 119-Pc. Tool Set	Hand Tools
	Master Craft 127-Pc. Tool Set	Hand Tools
	Highway Emergency Kit	Accessories
	NASCAR Fleece Shirt	Active
	NASCAR Denim Shirt	Active
	NASCAR Jersey	Active
4	Compact Metal Detector	Misc
	Highway Emergency Kit	Accessories
	Metal Detector	Misc
	Protek I Metal Detector	Misc
5	12-Pc. Bath Towel Set	Bath Sets
	20-Pc. All Cotton Bath Towels	Bath Sets
	20-Pc. Printed Towel Set	Bath Sets
	24 Piece Towel Set	Bath Sets
	24-Pc. Bath Towel Set	Bath Sets
	26-Pc. All-Cotton Towels	Bath Sets
	5-Pc. Bath Set	Bath Rugs
	50-Pc. Bath/Kitchen Towels	Bath Sets
	6-Pc. Bath Towel Set	Bath Sets
	Robe/Towel Sets	Bath Sets
	Solid Color Bath Towel Set	Bath Sets
	St. Croix Towel Sets	Bath Sets
	Bathroom Storage Accessories	Bath

Fig. 7. Interesting product clusters.

in the listed products. The most likely explanation for these clusters is holiday shoppers looking for gifts for a number of different people, since the log was taken from the 1999 holiday season. For the other four clusters listed in Figure 7, it is not immediately apparent why they have been ranked as interesting. However, for each cluster, there is at least one product that is part of a completely different high level category than the rest. For cluster 2, the “Dazey Turbospa” is under Housewares and the rest of the products are under Domestics. In cluster 3, the three pieces of “NASCAR” clothing are under Apparel and for cluster 4 the metal detectors are listed under the Sporting Goods category. All of the other products from clusters 3 and 4 are listed under Hardware. For cluster 5, the last product, “Bathroom Storage Accessories” is listed under the Home Accents hierarchy, while the other products are under Domestics. Because the site structure somewhat mirrors the content hierarchy, in each of these cases, several links had to be followed to get to the desired products in a different category. These interesting clusters indicate an opportunity to increase cross-sells for the e-commerce site by directly linking the products from different categories.

7. CONCLUSIONS

Some amount of information about the content and structure of a Web site is used in every Web Usage Mining project. Even if the knowledge is simply that the site is static and each page view is single framed, the preprocessing of the Server logs cannot be completed without knowing these facts. In the case of the frequent itemsets discovered in Section 6, even getting to the point of being able to generate the frequent itemsets at reasonable level of support required knowledge of the site content to perform the URI translation, and site structure to build the page views. It turned out to be the absence of user behaviors that were the most interesting. This type of pattern analysis cannot occur without an understanding of how the Web site should be used. The results of the concept clustering also show that knowledge of the content of a Web site can be effectively utilized to perform pattern analysis. The interesting product clusters could have easily been lost among the hundred or so discovered clusters. Not to mention that the ability to mine solely on product page accesses was only possible through intelligent URI translation and content based filtering. The automatic translation of URIs is an open issure that is currently preventing Web Usage Mining systems from gaining widespread use. However, this article has shown that the amount of effort involved in processing and quantifying the structure and content of a Web site is well worth it in terms of the usefulness of the results.

REFERENCES

AGRAWAL, R. AND SRIKANT, R. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference* (Santiago, Chile). Jorge Bocca, Matthias Jarke, and Carlo Zaniolo, Eds. Morgan-Kaufmann, Reading, Mass., pp. 487–499.

ANSARI, S., KOHAVI, R., MASON, L., AND ZHENG, Z. 2000. Integrating e-commerce and data mining: Architecture and challenges. In *Workshop on Web Mining for E-Commerce* (Boston, Mass). pp. 49–60.

BALABANOVIC, M. AND SHOHAM, Y. 1995. Learning information retrieval agents: Experiments with automated web browsing. In *On-line Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments*.

BONISSONE, P. P. AND DECKER, K. S. 1986. Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity. *Uncert. Artif. Intell.* 2217–2247.

BRIN, S., MOTWANI, R., AND SILVERSTEIN, C. 1997. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, New York.

BUCHNER, A. AND MULVENNA, M. D. 1998. Discovering internet marketing intelligence through online analytical web usage mining. *SIGMOD Record 27*, 4, 54–61.

CHEN, M. S., PARK, J. S., AND YU, P. S. 1996. Data mining for path traversal patterns in a web environment. In *Proceedings of the 16th International Conference on Distributed Computing Systems*. pp. 385–392.

COOLEY, R. 2000. Web usage mining: discovery and application of interesting patterns from web data. PhD. dissertation. Univ. Minnesota.

COOLEY, R., MOBASHER, B., AND SRIVASTAVA, J. 1999. Data preparation for mining world wide web browsing patterns. *Knowl. Inf. Syst. 1*, 1.

COOLEY, R., TAN, P.-N., AND SRIVASTAVA, J. 1999. Websift: The web site information filter system. In *WEBKDD* (San Diego, Calif).

COOLEY, R., TAN, P.-N., AND SRIVASTAVA, J. 2000. Discovery of Interesting Usage Patterns from Web Data. Lecture Notes in Computer Science/Lecture Notes in Artificial Intelligence, Myra Spiliopoulou, Ed. Springer-Verlag, New York.

CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T., NIGAM, K., AND SLATTERY, S. 1998. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the 15th Conference of the American Association for Artificial Intelligence* (Madison, Wisc.). AAAI Press, pp. 509–516.

GETOOR, L. AND SAHAMI, M. 1999. Using probabilistic relational models for collaborative filtering. In *WEBKDD* (San Diego, Calif.).

GIBSON, D., KLEINBERG, J., AND RAGHAVAN, P. 1998. Inferring web communities from link topology. In *Proceedings of the Conference on Hypertext and Hypermedia* (Pittsburgh, Pa.). ACM, New York, pp. 225–234.

GRUSER, J.-R., RASCHID, L., VIDAL, M. E., AND BRIGHT, L. 1998. Wrapper generation for web accessible data sources. In *Conference on Cooperative Information Systems* (New York, N.Y.). IEEE Computer Society, Los Alamitos, Calif., pp. 14–23.

HEER, J. AND CHI, E. H. 2001. Identification of web user traffic composition using multi-modal clustering and information scent. In *Proceedings of the 1st SIAM International Conference on Data Mining Workshop on Web Mining* (Chicago, Ill.). ACM, New York, pp. 51–58.

KARYPIS, G. AND HAN, E.-H. (SAM). 2000. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Tech. Rep. TR-00-0016. Univ. Minnesota.

KOSALA, R. AND BLOCKEEL, H. 2000. Web mining research: A survey. *SIGKDD Explorations 2*, 1, 1–15.

MANNILA, H., TOIVONEN, H., AND VERKAMO, A. I. 1995. Discovering frequent episodes in sequences. In *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining* (Montreal, Que., Canada). AAAI Press, pp. 210–215.

MOORE, J., HAN, E.-H. (SAM), BOLEY, D., GINI, M., GROSS, R., HASTINGS, K., KARYPIS, G., KUMAR, V., AND MOBASHER, B. 1997. Web page categorization and feature selection using association rule and principal component clustering. In *Proceedings of the 7th Workshop on Information Technologies and Systems*.

PIROLLI, P., PITKOW, J., AND RAO, R. 1996. Silk from a sow's ear: Extracting usable structures from the web. In *CHI-96* (Vancouver, B.C., Canada).

RESNIK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. 1994. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the Conference on Computer Supported Cooperative Work*. ACM.

SAHUGUET, A. AND AZAVANT, F. 1999. Building light-weight wrappers for legacy web data sources using w4f. In *International Conference on Very Large Databases* (Edinburgh, Scotland). Morgan-Kaufmann, Reading Mass., pp. 738–741.

SHAFER, G. 1976. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, N.J.

SILBERSCHATZ, A. AND TUZHILIN, A. 1996. What makes patterns interesting in knowledge discovery systems. *IEEE Trans. Knowl. Data Eng. 8*, 6, 970–974.

SPERTUS, E. 1997. Parasite: Mining structural information on the web. *Comput. Netw. ISDN Syst.: The Int. J. Comput. Telecommun. Netw. 29*, 1205–1215.

SPILIOPOULOU, M. AND FAULSTICH, L. C. 1998. WUM: A web utilization miner. In *EDBT Workshop WebDB98* (Valencia, Spain). Springer Verlag, Reading Mass.

SPILIOPOULOU, M., POHLE, C., AND FAULSTICH, L. C. 1999. Improving the effectiveness of a web site with web usage mining. In *WEBKDD* (San Diego, Calif.).

SRIVASTAVA, J., COOLEY, R., DESHPANDE, M., AND TAN, P.-N. 2000. Web usage mining: Discovery and application of usage pattens from web data. *SIGKDD Explor. 1*, 2, 12–23.

Received November 2000; revised February 2002 and October 2002; accepted November 2002