Data Preparation for Mining World Wide Web Browsing Patterns

Robert Cooley*, Bamshad Mobasher, and Jaideep Srivastava

Department of Computer Science and Engineering
University of Minnesota
4-192 EECS Bldg., 200 Union St. SE
Minneapolis, MN 55455, USA

Abstract. The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexity of tasks such as Web site design, Web server design, and of simply navigating through a Web site have increased along with this growth. An important input to these design tasks is the analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site. *Web Usage Mining* is the application of data mining techniques to usage logs of large Web data repositories in order to produce results that can be used in the design tasks mentioned above. However, there are several preprocessing tasks that must be performed prior to applying data mining algorithms to the data collected from server logs. This paper presents several data preparation techniques in order to identify unique users and user sessions. Also, a method to divide user sessions into semantically meaningful transactions is defined and successfully tested against two other methods. Transactions identified by the proposed methods are used to discover association rules from real world data using the WEBMINER system [15].

1 Introduction and Background

The World Wide Web (WWW) continues to grow at an astounding rate in both the sheer volume of traffic and the size and complexity of Web sites. The complexity of tasks such as Web site design, Web server design, and of simply navigating through a Web site have increased along with this growth. An important input to these design tasks is analysis of how a Web site is being used. Usage analysis includes straightforward statistics, such as page access frequency, as well as more sophisticated forms of analysis, such as finding the common traversal paths through a Web site. Usage information can be used to restructure a Web site in order to better serve the needs of users of a site. Long convoluted traversal paths or low usage of a page with important site information could suggest that the site links and information are not laid out in an intuitive manner. The design

* Supported by NSF grant EHR-9554517.

of a phsysical data layout or caching scheme for a distributed or parallel Web server can be enchanced by knowledge of how users typically navigate through the site. Usage information can also be used to directly aide site navigation by providing a list of "popular" destinations from a particular Web page.

Web Usage Mining is the application of data mining techniques to large Web data repositories in order to produce results that can be used in the design tasks mentioned above. Some of the data mining algorithms that are commonly used in Web Usage Mining are association rule generation, sequential pattern generation, and clustering. Association Rule mining techniques [1] discover unordered correlations between items found in a database of transactions. In the context of Web Usage Mining a transaction is a group of Web page accesses, with an item being a single page access. Examples of association rules found from an IBM analysis of the server log of the Official 1996 Olympics Web site [7] are:

– 45% of the visitors who accessed a page about Indoor Volleyball also accessed a page on Handball.
– 59.7% of the visitors who accessed pages about Badminton and Diving also accessed a page about Table Tennis.

The percentages reported in the examples above are referred to as *confidence*. Confidence is the number of transactions containing all of the items in a rule, divided by the number of transactions containing the rule antecedants (The antecedants are Indoor Volleyball for the first example and Badminton and Diving for the second example).

The problem of discovering *sequential patterns* [17, 26] is that of finding inter-transaction patterns such that the presence of a set of items is followed by another item in the time-stamp ordered transaction set. By analyzing this information, a Web Usage Mining system can determine temporal relationships among data items such as the following Olympics Web site examples:

– 9.81% of the site visitors accessed the Atlanta home page followed by the Sneakpeek main page.
– 0.42% of the site visitors accessed the Sports main page followed by the Schedules main page.

The percentages in the second set of examples are referred to as *support*. Support is the percent of the transactions that contain a given pattern. Both confidence and support are commonly used as thresholds in order to limit the number of rules discovered and reported. For instance, with a 1% support threshold, the second sequential pattern example would not be reported.

Clustering analysis [12, 19] allows one to group together users or data items that have similar characteristics. Clustering of user information or data from Web server logs can facilitate the development and execution of future marketing strategies, both online and off-line, such as automated return mail to visitors falling within a certain cluster, or dynamically changing a particular site for a visitor on a return visit, based on past classification of that visitor.

As the examples above show, mining for knowledge from Web log data has the potential of revealing information of great value. While this certainly is an application of existing data mining algorithms, e.g. discovery of association rules or sequential patterns, the overall task is not one of simply adapting existing algorithms to new data. Ideally, the input for the Web Usage Mining process is a file, referred to as a *user session file* in this paper, that gives an exact accounting of who accessed the Web site, what pages were requested and in what order, and how long each page was viewed. A user session is considered to be all of the page accesses that occur during a single visit to a Web site. The information contained in a raw Web server log does not reliably represent a user session file for a number of reasons that will be discussed in this paper. Specifically, there are a number of difficulties involved in cleaning the raw server logs to eliminate outliers and irrelevant items, reliably identifying unique users and user sessions within a server log, and identifying semantically meaningful transactions within a user session.

This paper presents several data preparation techniques and algorithms that can be used in order to convert raw Web server logs into user session files in order to perform Web Usage Mining. The specific contributions include (i) development of models to encode both the Web site developer's and users' view of how a Web site should be used, (ii) discussion of heuristics that can be used to identify Web site users, user sessions, and page accesses that are missing from a Web server log, (iii) definition of several transaction identification approaches, and (iv) and evaluation of the different transaction identification approaches using synthetic server log data with known association rules.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 briefly discusses the architecture of the Web Usage Mining process and the WEBMINER system [15]. Section 4 presents a model for user browsing behavior and a method for encoding a Web site designer's view of how a site should be used. Section 5 gives a detailed breakdown of the steps involved in preprocessing data for Web Usage Mining. Section 6 presents a general model for identifying transactions along with some specific transaction identification approaches. Section 7 discusses a method used to generate Web server log data in order to compare the different transaction identification approaches. Section 8 presents the experimental results of using the WEBMINER system to mine for association rules with transactions identified with the different approaches. Finally, Section 9 provides conclusions.

2 Related Work

There are several commercially available Web server log analysis tools, such as [8, 10, 18], that provide limited mechanisms for reporting user activity, i.e. it is possible to determine the number of accesses to individual files and the times of visits. However, these tools are not designed for very high traffic Web servers, and usually provide little analysis of data relationships among accessed files, which is essential to fully utilizing the data gathered in the server logs. The concept of applying data mining techniques to Web server logs was first proposed in [6], [16], and [29]. Mannila et. al. [16] use page accesses from a Web server log as events for discovering frequent episodes [17]. Chen et. al. [6] introduce the concept of using the *maximal forward references* in order to break down user sessions into transactions for the mining of traversal patterns. A maximal forward reference is the last page requested by a user before backtracking occurs, where the user requests a page previously viewed during that particular user session. For example, if a user session consists of requests for pages `A-B-A-C-D-C`, in that order, the maximal forward references for the session would be `B` and `D`. Both [6] and [16] concentrate on developing data mining algorithms and assume that the server logs, after filtering out image files, represent an accurate picture of site usage. The Analog system [29] uses Web server logs to assign site visitors to clusters. The links that are presented to a given user are dynamically selected based on what pages other users assigned to the same cluster have visited.

The difficulty of identifying users and user sessions from Web server logs has been addressed in research performed by Pitkow [5, 21, 23]. Two of the biggest impediments to collecting reliable usage data are local caching and proxy servers. In order to improve performance and minimize network traffic, most Web browsers cache the pages that have been requested. As a result, when a user hits the "back" button, the cached page is displayed and the Web server is not aware of the repeat page access. Proxy servers provide an intermediate level of caching and create even more problems with identifying site usage. In a Web server log, all requests from a proxy server have the same identifier, even though the requests potentially represent more than one user. Also, due to proxy server level caching, a single request from the server could actually be viewed by multiple users throughout an extended period of time. The data mining results from the 1996 Olympics site [7] were obtained using *cookies* to identify site users and user sessions. Cookies are markers that are used to tag and track site visitors automatically. Another approach to getting around the problems created by caching and proxy servers is to use a remote agent, as described in [28]. Instead of sending a cookie, [28] sends a Java agent that is run on the client side browser in order to send back accurate usage information to the Web server. The major disadvantage of the methods that rely on implicit user cooperation stem from privacy issues. There is a constant struggle between the Web user's desire for privacy and the Web provider's desire for information about who is using their site. Many users choose to disable the browser features that enable these methods.

Han et. al. [30] have loaded Web server logs into a data cube structure [9] in order to perform data mining as well as traditional On-Line Analytical Processing (OLAP) activities such as roll-up and drill-down of the data. While both [29] and [30] acknowledge the difficulties involved in identifying users and user sessions, no specific methods are presented for solving the problem. Smith et. al. [27] use "path profiles" in order to generate dynamic content in advance of a user request. The problems that caching causes are also identified in [27], but because of the focus on dynamic content, caching is assumed to have minimal impact, and is hence ignored.

The SiteHelper system [20] learns from information extracted from a Web site's server log in tandem with search cues explicitly given by a user in order to recommend pages within the Web site. Several research efforts, such as [3, 11, 13, 22], maintain logs of user actions as part of the process of using the content of Web pages and hypertext links in order to provide page recommendations to users. These projects can be classified as *Web Content Mining* as defined by [4], since the learning and inference is predominantly based on the content of the pages and links, and not strictly the behavior of the users. Because users explicitly request the services of these recommendation systems, the problem of identifying user sessions is greatly simplified. Effectively, a request for a page recommendation on the part of a user serves as a registration, enabling the tracking of the session. While users are often willing to register in order to receive certain benefits (which in this case is in the form of navigation assistance), for tasks such as usage analysis that provide no immediate feedback to the user, registration is often looked upon as an invasion of privacy, similar to the use of cookies and remote agents.

3 The WEBMINER System

The WEBMINER system [4, 15] divides the Web Usage Mining process into three main parts, as shown in Figs. 1 and 2. Input data consists of the three server logs - access, referrer, and agent, the HTML files that make up the site, and any optional data such as registration data or remote agent logs. The first part of Web Usage Mining, called preprocessing, includes the domain dependent tasks of data cleaning, user identification, session identification, and path completion. Data cleaning is the task of removing log entries that are not needed for the mining process. User identification is the process of associating page references, even those with the same IP address, with different users. The site topology is required in addition to the server logs in order to perform user identification. Session identification takes all of the page references for a given user in a log and breaks them up into user sessions. As with user identification, the site topology is needed in addition to the server logs for this task. Path completion fills in page references that are missing due to browser and proxy server caching. This step differs from the others in that information is being added to the log. The other preprocessing tasks remove data or divide the data up according to users and sessions. Each of these tasks are performed in order to create a user session file which will be used as input to the knowledge discovery phase.

As shown in Fig. 2, mining for association rules requires the added step of transaction identification, in addition to the other preprocessing tasks. Transaction identification is the task of identifying semantically meaningful groupings of page references. In a domain such as market basket analysis, a transaction has a natural definition - all of the items purchased by a customer at one time. However, the only "natural" transaction definition in the Web domain is a user

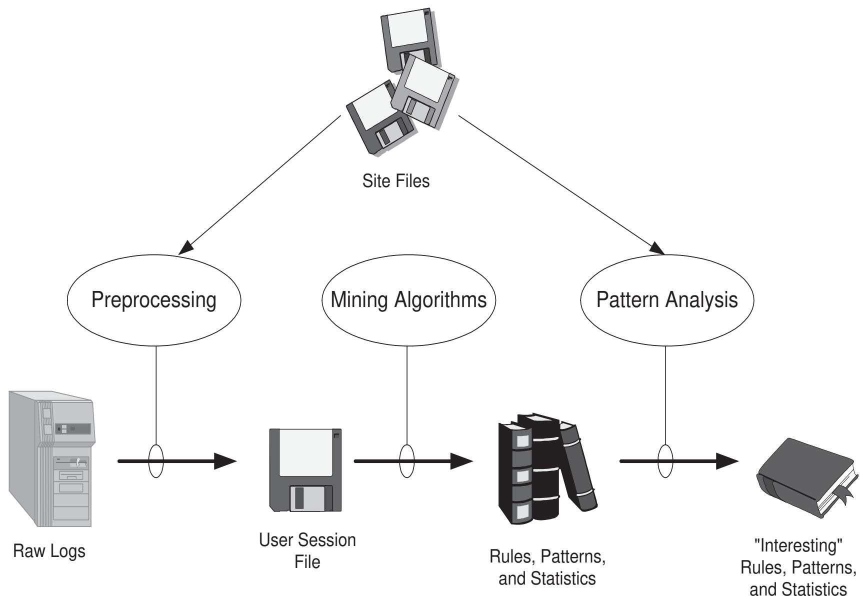

Fig. 1. High Level Usage Mining Process

session, which is often too coarse grained for mining tasks such as the discovery of association rules. Therefore, specialized algorithms are needed to refine single user sessions into smaller transactions.

The knowledge discovery phase uses existing data mining techniques to generate rules and patterns. Included in this phase is the generation of general usage statistics, such as number of “hits” per page, page most frequently accessed, most common starting page, and average time spent on each page. Association rule and sequential pattern generation are the only data mining algorithms currently implemented in the WEBMINER system, but the open architecture can easily accommodate any data mining or path analysis algorithm. The discovered information is then fed into various pattern analysis tools. The *site filter* is used to identify interesting rules and patterns by comparing the discovered knowledge with the Web site designer’s view of how the site should be used, as discussed in the next section. As shown in Fig. 2, the site filter can be applied to the data mining algorithms in order to reduce the computation time, or the discovered rules and patterns.

4 Browsing Behavior Models

In some respects, Web Usage Mining is the process of reconciling the Web site developer’s view of how the site should be used with the way users are actually browsing through the site. Therefore, the two inputs that are required for the

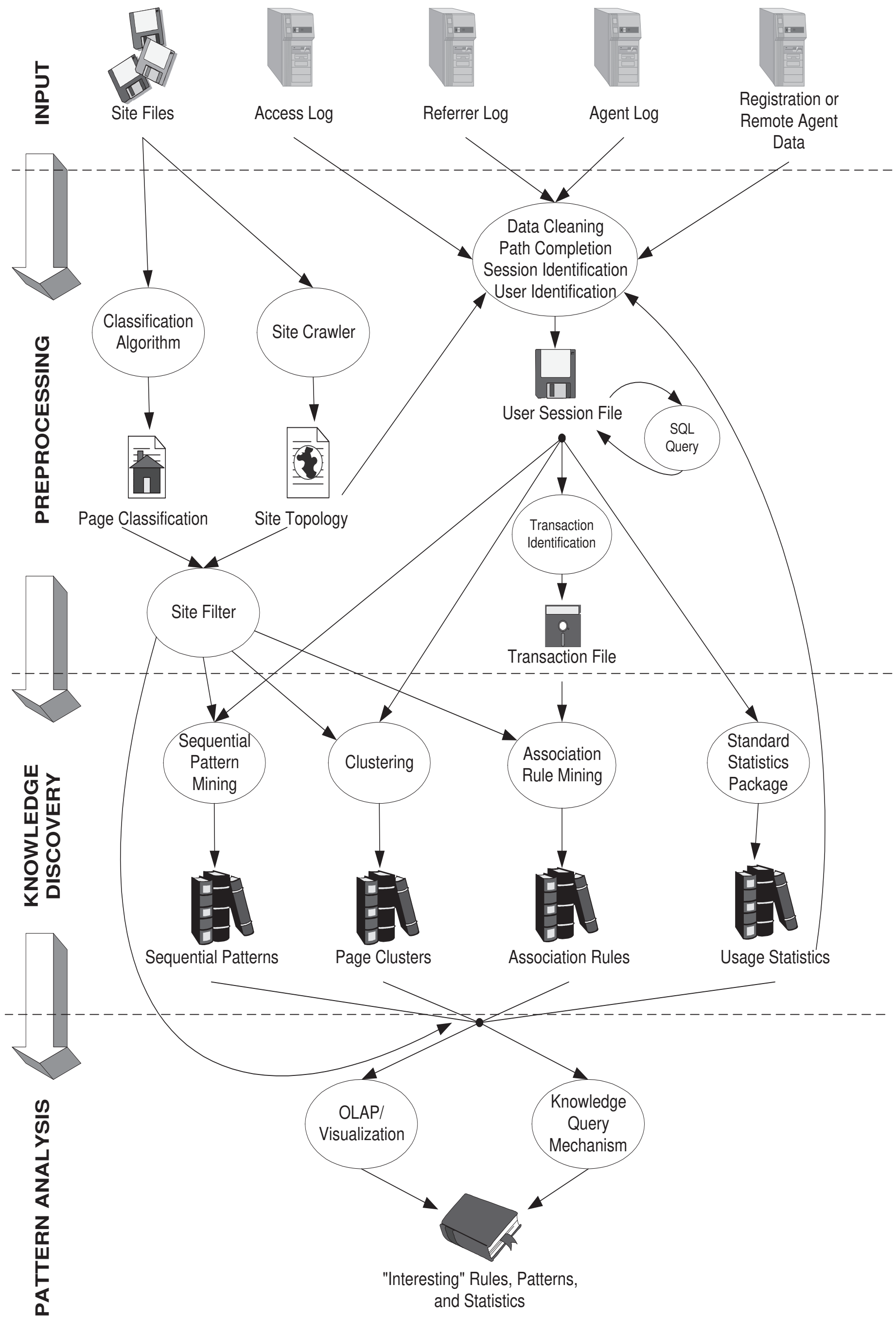

Fig. 2. Architecture for WEBMINER

Table 1. Common Characteristics of Web Pages

Page Type	Physical Characteristics	Usage Characteristics
Head	• In-links from most site pages • Root of site file structure	• First page in user sessions
Content	• Large text/graphic to link ratio	• Long average reference length
Navigation	• Small text/graphic to link ratio	• Short average reference length • Not a maximal forward reference
Look-up	• Large number of in-links • Few or no out-links • Very little content	• Short average reference length • Maximal forward reference
Personal	• No common characteristics	• Low usage

Web Usage Mining process are an encoding of the site developer's view of browsing behavior and an encoding of the actual browsing behaviors. These inputs are derived from the site files and the server logs respectively.

4.1 Developer's Model

The Web site developer's view of how the site should be used is inherent in the structure of the site. Each link between pages exists because the developer believes that the pages are related in some way. Also, the content of the pages themselves provide information about how the developer expects the site to be used. Hence, an integral step of the preprocessing phase is the classifying of the site pages and extracting the site topology from the HTML files that make up the web site. The topology of a Web site can be easily obtained by means of a site "crawler", that parses the HTML files to create a list of all of the hypertext links on a given page, and then follows each link until all of the site pages are mapped. The pages are classified using an adaptation of the classification scheme presented in [23]. The WEBMINER system recognizes five main types of pages:

- Head Page - a page whose purpose is to be the first page that users visit, i.e. "home" pages.
- Content Page - a page that contains a portion of the information content that the Web site is providing.
- Navigation Page - a page whose purpose is to provide links to guide users on to content pages.
- Look-up Page - a page used to provide a definition or acronym expansion.
- Personal Page - a page used to present information of a biographical or personal nature for individuals associated with the organization running the Web site.

Each of these types of pages is expected to exhibit certain physical characteristics. For example, a head page is expected to have in-links from most of the other pages in the site, and is often at the root of the site file structure. Table 1 lists some common physical characteristics for each page type. Note that these are only rules-of-thumb, and that there will be pages of a certain type that do not match the common physical characteristics. Personal pages are not expected to exhibit any common characteristics. Each personal page is expected to be a combination of one of the other page types. For example, personal pages often have content in the form of biographical information followed by a list of "favorite" links. The important distinction for personal pages is that they are not controlled by the site designer, and are thus not expected to contribute heavily to discovered rules. This is because usage is expected to be low for any given personal page compared to the overall site traffic. Thresholds such as support would be expected to filter out rules that contain personal pages. There are several valid combinations of page types that can be applied to a single page, such as a head-navigation page or a content-navigation page. The page classifications should represent the Web site designer's view of how each page will be used. The classifications can be assigned manually by the site designer, or automatically using supervised learning techniques. In order to automate the classification of site pages, the common physical characteristics can be learned by a classification algorithm such as C4.5 [25] using a training set of pages. Another possibility is that a classification tag can be added to each page by the site designer, using a data structure markup language such as XML (Extensible Markup Language) [2].

4.2 Users' Model

Analogous to each of the common physical characteristics for the different page types, there is expected to be common usage characteristics among different users. These are also shown in Table 1. The reference length of a page is the amount of time a user spends viewing that particular page for a specific log entry. Some of the challenges involved in calculating reference lengths and maximal forward references will be discussed in Sect. 6.

In order to group individual Web page references into meaningful transactions for the discovery of patterns such as association rules, an underlying model of the user's browsing behavior is needed. For the purposes of association rule discovery, it is really the content page references that are of interest. The other page types are just to facilitate the browsing of a user while searching for information, and will be referred to as auxiliary pages. What is merely an *auxiliary* page for one user may be a *content* page for another. Transaction identification assumes that user sessions have already been identified (see Sect. 5.2).

Using the concept of auxiliary and content page references, there are two ways to define transactions, as shown in Fig. 3. The first would be to define a transaction as all of the auxiliary references up to and including each content reference for a given user. Mining these *auxiliary-content* transactions would essentially give the common traversal paths through the web site to a given content page. The second method would be to define a transaction as all of the content references for a given user. Mining these *content-only* transactions would give associations between the content pages of a site, without any information as to the path taken between the pages. It is important to note that results generated from content-only transactions only apply when those pages are used as content references. For example, an association rule, `A` $\Longrightarrow$ `B`, is normally taken to mean that when `A` is in a set, so is `B`. However, if this rule has been generated with content-only transactions, it has a more specific meaning, namely that `A` implies `B` only when both `A` and `B` are used as content references. This property allows data mining on content-only transactions to produce rules that might be missed by including all of the page references in a log. If users that treat page `A` as an auxiliary page do not generally go on to page `B`, inclusion of the auxiliary references into the data mining process would reduce the confidence of the rule `A` $\Longrightarrow$ `B`, possibly to the point where it is not reported. Depending on the goals of the analysis, this can be seen as an advantage or a disadvantage. The key is that auxiliary-content transactions can be used when this property is undesirable. The challenge of identifying transactions is to dynamically determine which references in a server log are auxiliary and which are content. Three different approaches are presented and compared in Sects. 6 and 8.

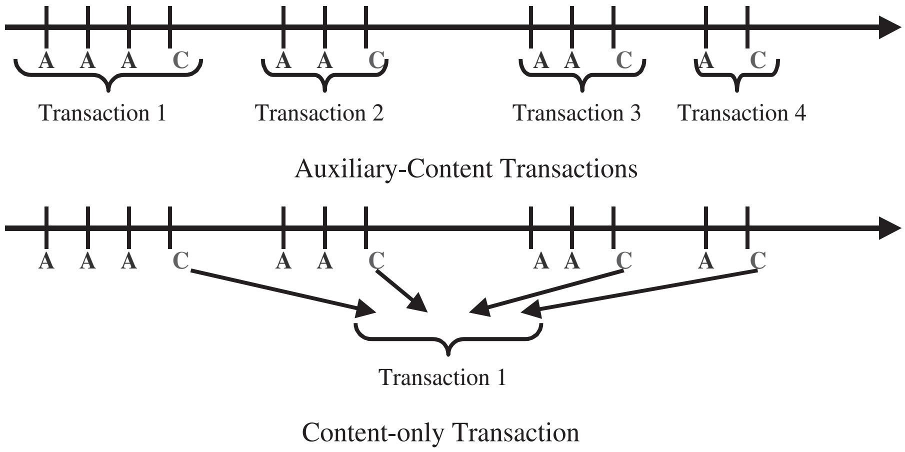

Fig. 3. Transaction Types: Auxiliary and Content page references are labeled along the time axes with an A or C respectively

4.3 Site Filter

As shown in Fig. 2, the site topology is used during the user identification and path completion preprocessing tasks. Both site topology and page classifications will be needed for applying a site filter to the mining algorithms, or the generated rules and patterns during the pattern analysis phase. The site filter method used in the WEBMINER system compares the web site designer's view of how the site should be used with the way users are actually browsing through the site, and reports any discrepancies. For example, the site filter simply checks head pages to see if a significant number of users are starting on that page, or if there are other pages in the site which are a more common starting point for browsing. The start statistics for the head page are only flagged as interesting if it is not being treated as such by the users. Conversely, any page that is being used as a head page, but is not classified as one is also reported.

The site filter also uses the site topology to filter out rules and patterns that are uninteresting. Any rule that confirms direct hypertext links between pages is filtered out. The sensitivity of the task can be adjusted by increasing the length of the paths that are considered uninteresting. In other words, rules confirming pages that are up to `n` links away can be pruned, instead of just those that are directly linked. In addition, rules that are expected but not present can also be identified. If two pages are directly linked and not confirmed by a rule or pattern, this information is as important as the existence of rules that are not supported by the site structure. By using the site filter, the confidence and support thresholds of the standard data mining algorithms can be lowered significantly to generate more potential rules and patterns, which are then culled to provide the data analyst with a reasonable number of results to examine.

5 Preprocessing

Figure 4 shows the preprocessing tasks of Web Usage Mining in greater detail than Figs. 1 and 2. The inputs to the preprocessing phase are the server logs, site files, and optionally usage statistics from a previous analysis. The outputs are the user session file, transaction file, site topology, and page classifications. As mentioned in Sect. 2, one of the major impediments to creating a reliable user session file is browser and proxy server caching. Current methods to collect information about cached references include the use of cookies and cache busting. Cache busting is the practice of preventing browsers from using stored local versions of a page, forcing a new down-load of a page from the server every time it is viewed. As detailed in [21], none of these methods are without serious drawbacks. Cookies can be deleted by the user and cache busting defeats the speed advantage that caching was created to provide, and is likely to be disabled by the user. Another method to identify users is user registration, as discussed in section 2. Registration has the advantage of being able to collect additional demographic information beyond what is automatically collected in the server log, as well as simplifying the identification of user sessions. However, again due to privacy concerns, many users choose not to browse sites that require registration and logins, or provide false information. The preprocessing methods used in the WEBMINER system are all designed to function with only the information supplied by the *Common Log Format* specified as part of the HTTP protocol by CERN and NCSA [14].

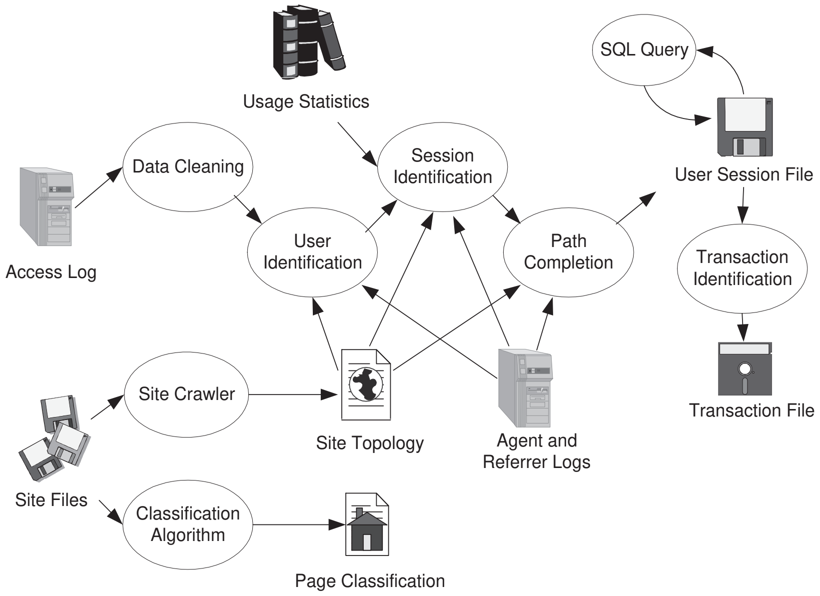

Fig. 4. Details of Web Usage Mining Preprocessing

5.1 Data Cleaning

Techniques to clean a server log to eliminate irrelevant items are of importance for any type of Web log analysis, not just data mining. The discovered associations or reported statistics are only useful if the data represented in the server log gives an accurate picture of the user accesses to the Web site. The HTTP protocol requires a separate connection for every file that is requested from the Web server. Therefore, a user's request to view a particular page often results in several log entries since graphics and scripts are down-loaded in addition to the HTML file. In most cases, only the log entry of the HTML file request is relevant and should be kept for the user session file. This is because, in general, a user does not explicitly request all of the graphics that are on a Web page, they are automatically down-loaded due to the HTML tags. Since the main intent of Web Usage Mining is to get a picture of the user's behavior, it does not make sense to include file requests that the user did not explicitly request. Elimination of the items deemed irrelevant can be reasonably accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG, and map can be removed. In addition, common scripts such as "count.cgi" can also be removed. The WEBMINER system uses a default list of suffixes to remove files. However, the list can be modified depending on the type of site being analyzed. For instance, for a Web site that contains a graphical archive, an analyst would probably not want to automatically remove all of the GIF or JPEG files from the server log. In this case, log entries of graphics files may very well represent explicit user actions, and should be retained for analysis. A list of actual file names to remove or retain can be used instead of just file suffixes in order to distinguish between relevant and irrelevant log entries.

5.2 User Identification

Next, unique users must be identified. As mentioned previously, this task is greatly complicated by the existence of local caches, corporate firewalls, and proxy servers. The Web Usage Mining methods that rely on user cooperation are the easiest ways to deal with this problem. However, even for the log/site based methods, there are heuristics that can be used to help identify unique users. As presented in [23], even if the IP address is the same, if the agent log shows a change in browser software or operating system, a reasonable assumption to make is that each different agent type for an IP address represents a different user. For example, consider the Web site shown in figure 5 and the sample information collected from the access, agent, and referrer logs shown in figure 6. All of the log entries have the same IP address and the user ID is not recorded. However, the fifth, sixth, eighth, and tenth entries were accessed using a different agent than the others, suggesting that the log represents at least two user sessions. The next heuristic for user identification is to use the access log in conjunction with the referrer log and site topology to construct browsing paths for each user. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, again, the heuristic assumes that there is another user with the same IP address. Looking at the figure 6 sample log again, the third entry, page `L`, is not directly reachable from pages `A` or `B`. Also, the seventh entry, page `R` is reachable from page `L`, but not from any of the other previous log entries. This would suggest that there is a third user with the same IP address. Therefore, after the user identification step with the sample log, three unique users are identified with browsing paths of `A-B-F-O-G-A-D`, `A-B-C-J`, and `L-R`, respectively. It is important to note that these are only heuristics for identifying users. Two users with the same IP address that use the same browser on the same type of machine can easily be confused as a single user if they are looking at the same set of pages. Conversely, a single user with two different browsers running, or who types in URLs directly without using a sites link structure can be mistaken for multiple users.

5.3 Session Identification

For logs that span long periods of time, it is very likely that users will visit the Web site more than once. The goal of session identification is to divide the page accesses of each user into individual sessions. The simplest method of achieving this is through a timeout, where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session. Many commercial products use 30 minutes as a default timeout, and [5] established a timeout of 25.5 minutes based on empirical data. Once a site log has been

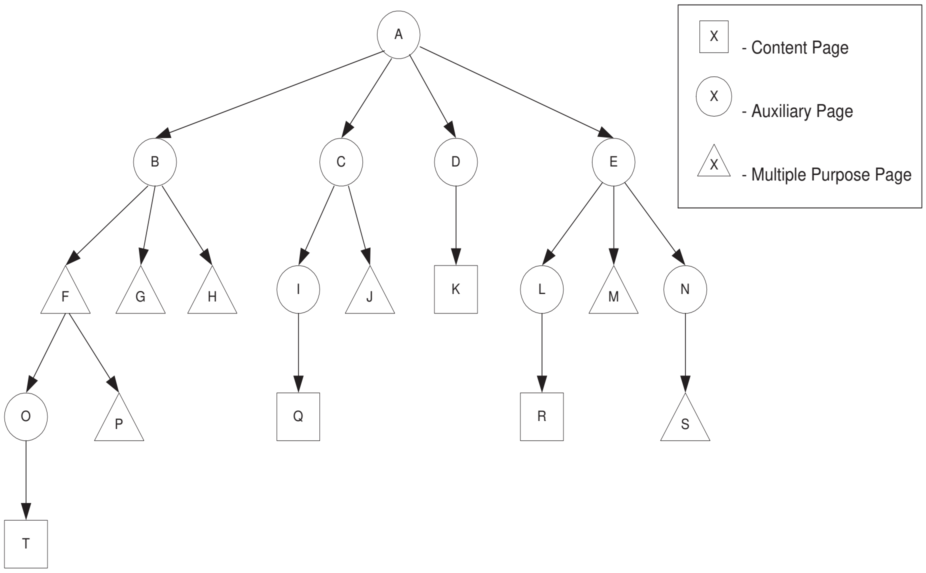

Fig. 5. Sample Web Site - Arrows between the pages represent hypertext links

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referred	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7	123.456.78.9	-	[25/Apr/1998:03:07:55 -0500]	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04 (Win95, I)
8	123.456.78.9	-	[25/Apr/1998:03:09:50 -0500]	"GET C.html HTTP/1.0"	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9	123.456.78.9	-	[25/Apr/1998:03:10:02 -0500]	"GET O.html HTTP/1.0"	200	2270	F.html	Mozilla/3.04 (Win95, I)
10	123.456.78.9	-	[25/Apr/1998:03:10:45 -0500]	"GET J.html HTTP/1.0"	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11	123.456.78.9	-	[25/Apr/1998:03:12:23 -0500]	"GET G.html HTTP/1.0"	200	7220	B.html	Mozilla/3.04 (Win95, I)
12	123.456.78.9	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
13	123.456.78.9	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

Fig. 6. Sample Information from Access, Referrer, and Agent Logs (The first column is for referencing purposes and would not be part of an actual log).

analyzed and usage statistics obtained, a timeout that is appropriate for the specific Web site can be fed back into the the session identification algorithm. This is the reason usage statistics are shown as an input to session identification in Fig. 4. Using a 30 minute timeout, the path for user 1 from the sample log is broken into two separate sessions since the last two references are over an hour later than the first five. The session identification step results in four user sessions consisting of `A-B-F-O-G`, `A-D`, `A-B-C-J`, and `L-R`.

5.4 Path Completion

Another problem in reliably identifying unique user sessions is determining if there are important accesses that are not recorded in the access log. This problem is referred to as *path completion*. Methods similar to those used for user identification can be used for path completion. If a page request is made that is not directly linked to the last page a user requested, the referrer log can be checked to see what page the request came from. If the page is in the user's recent request history, the assumption is that the user backtracked with the "back" button available on most browsers, calling up cached versions of the pages until a new page was requested. If the referrer log is not clear, the site topology can be used to the same effect. If more than one page in the user's history contains a link to the requested page, it is assumed that the page closest to the previously requested page is the source of the new request. Missing page references that are inferred through this method are added to the user session file. An algorithm is then required to estimate the time of each added page reference. A simple method of picking a time-stamp is to assume that any visit to a page already seen will be effectively treated as an auxiliary page. The average reference length for auxiliary pages for the site can be used to estimate the access time for the missing pages. Looking at Figs. 5 and 6 again, page `G` is not directly accessible from page `O`. The referrer log for the page `G` request lists page `B` as the requesting page. This suggests that user 1 backtracked to page `B` using the back button before requesting page `G`. Therefore, pages `F` and `B` should be added into the session file for user 1. Again, while it is possible that the user knew the URL for page `G` and typed it in directly, this is unlikely, and should not occur often enough to affect the mining algorithms. The path completion step results in user paths of `A-B-F-O-F-B-G`, `A-D`, `A-B-A-C-J`, and `L-R`. The results of each of the preprocessing steps are summarized in Table 2.

5.5 Formatting

Once the appropriate preprocessing steps have been applied to the server log, a final preparation module can be used to properly format the sessions or transactions for the type of data mining to be accomplished. For example, since temporal information is not needed for the mining of association rules, a final association rule preparation module would strip out the time for each reference, and do any other formatting of the data necessary for the specific data mining algorithm to be used.

Table 2. Summary of Sample Log Preprocessing Results

Task	Result
Clean Log	• `A-B-L-F-A-B-R-C-O-J-G-A-D`
User Identification	• `A-B-F-O-G-A-D` • `A-B-C-J` • `L-R`
Session Identification	• `A-B-F-O-G` • `A-D` • `A-B-C-J` • `L-R`
Path Completion	• `A-B-F-O-F-B-G` • `A-D` • `A-B-A-C-J` • `L-R`

6 Transaction Identification

6.1 General Model

Each user session in a user session file can be thought of in two ways; either as a single transaction of many page references, or a set of many transactions each consisting of a single page reference. The goal of transaction identification is to create meaningful clusters of references for each user. Therefore, the task of identifying transactions is one of either *dividing* a large transaction into multiple smaller ones or *merging* small transactions into fewer larger ones. This process can be extended into multiple steps of merge or divide in order to create transactions appropriate for a given data mining task. A transaction identification approach can be defined as either a merge or a divide approach. Both types of approaches take a transaction list and possibly some parameters as input, and output a transaction list that has been operated on by the function in the approach in the same format as the input. The requirement that the input and output transaction format match allows any number of approaches to be combined in any order, as the data analyst sees fit. Let L be a set of user session file entries. A session entry $l \in L$ includes the client IP address $l.ip$, the client user id $l.uid$, the URL of the accessed page $l.url$, and the time of access $l.time$. If the actual user id is not available, which is often the case, an arbitrarily assigned identifier from the user session file is used. There are other fields in user session file entries, such as the request method used (e.g., POST or GET) and the size of the file transmitted, however these fields are not used in the transaction model. A General Transaction t is a triple, as shown in (1).

$$\begin{aligned} &t = < ip_t, uid_t, \{(l_1^t.url, l_1^t.time), \ldots, (l_m^t.url, l_m^t.time)\} > \\ &\text{where, for } 1 \leq k \leq m,\ \ l_k^t \in L,\ \ l_k^t.ip = ip_t,\ \ l_k^t.uid = uid_t \end{aligned} \tag{1}$$

Table 3. Summary of Sample Transaction Identification Results (Transaction type is not defined for the Time Window Approach)

Approach	Transactions	
	Content-only	Auxiliary-Content
Reference Length	F-G, D, L-R, J	A-B-F, O-F-B-G, A-D L, R, A-B-A-C-J
Maximal Forward Reference	O-G, R, B-J, D	A-B-F-O, A-B-G, L-R A-B, A-C-J, A-D
Time Window	A-B-F, O-F-B-G, A-D, L-R, A-B-A-C-J	

Since the initial input to the transaction identification process consists of all of the page references for a given user session, the first step in the transaction identification process will always be the application of a divide approach. The next sections describe three divide transaction identification approaches. The first two, *reference length* and *maximal forward reference*, make an attempt to identify semantically meaningful transactions. The third, *time window*, is not based on any browsing model, and is mainly used as a benchmark to compare with the other two algorithms. The results of using the three different approaches on the sample Fig. 6 data are shown in Table 3.

6.2 Transaction Identification by Reference Length

The reference length transaction identification approach is based on the assumption that the amount of time a user spends on a page correlates to whether the page should be classified as a auxiliary or content page for that user. Figure 7 shows a histogram of the lengths of page references between 0 and 600 seconds for a server log from the Global Reach Internet Productions (GRIP) Web site [24].

Qualitative analysis of several other server logs reveals that like Fig. 7, the shape of the histogram has a large exponential component. It is expected that the variance of the times spent on the auxiliary pages is small, and the auxiliary references make up the lower end of the curve. The length of content references is expected to have a wide variance and would make up the upper tail that extends out to the longest reference. If an assumption is made about the percentage of auxiliary references in a log, a reference length can be calculated that estimates the cutoff between auxiliary and content references. Specifically, given a percent of auxiliary references, the reference length method uses a maximum likelihood estimate to calculate the time length `t` as shown in (2).

$$
\begin{aligned}
t &= \tfrac{-ln(1-\gamma)}{\lambda} \\
\text{where } \gamma &= \ \% \text{ of auxiliary references,} \\
\lambda &= \ \text{reciprocal of observed mean reference length.}
\end{aligned} \qquad (2)
$$

The definition of (2) comes from integrating the formula for an exponential distribution, from γ to zero. The maximum likelihood estimate for the exponential

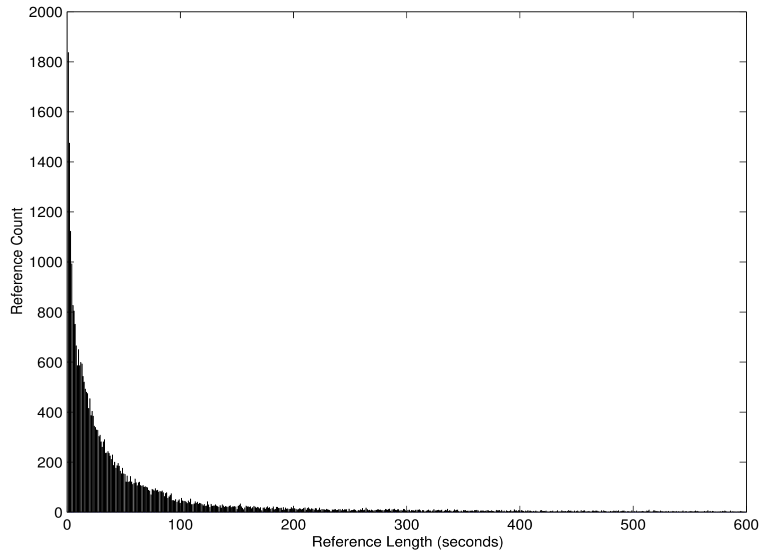

Fig. 7. Histogram of Web Page Reference Lengths (seconds)

distribution is the observed mean. The time length `t` could be calculated exactly by sorting all of the reference lengths from a log and then selecting the reference length that is located in the $\gamma\times$`log_size` position. However, this increases the complexity of the algorithm from linear to O(nlogn) while not necessarily increasing the accuracy of the calculation since the value of γ is only an estimate. Although the exponential distribution does not fit the histograms of server log data exactly, it provides a reasonable estimate of the cutoff reference length. It would be interesting to examine the cost-benefit tradeoffs for distributions that fit the histograms more accurately.

The definition of a transaction within the reference length approach is a quadruple, and is given in (3). It has the same structure as (1) with the reference length added for each page.

$$\begin{gathered} t_{rl} = < ip_{t_{rl}}, uid_{t_{rl}}, \{(l_1^{t_{rl}}.url, l_1^{t_{rl}}.time, l_1^{t_{rl}}.length), \\ \ldots, (l_m^{t_{rl}}.url, l_m^{t_{rl}}.time, l_m^{t_{rl}}.length)\} > \\ \text{where, for } 1 \le k \le m,\ \ l_k^{t_{rl}} \in L,\ \ l_k^{t_{rl}}.ip = ip_{t_{rl}},\ \ l_k^{t_{rl}}.uid = uid_{t_{rl}} \end{gathered} \tag{3}$$

The length of each reference is estimated by taking the difference between the time of the next reference and the current reference. Obviously, the last reference in each transaction has no "next" time to use in estimating the reference length. The reference length approach makes the assumption that all of the last references are content references, and ignores them while calculating the cutoff time. This assumption can introduce errors if a specific auxiliary page is commonly used as the exit point for a Web site. While interruptions such as a phone call or lunch break can result in the erroneous classification of a auxiliary reference as a content reference, it is unlikely that the error will occur on a regular basis for the same page. A reasonable minimum support threshold during the application of a data mining algorithm would be expected to weed out these errors.

Once the cutoff time is calculated, the two types of transactions discussed in Sect. 4 can be formed by comparing each reference length against the cutoff time. Depending on the goal of the analysis, the auxiliary-content transactions or the content-only transactions can be identified. If C is the cutoff time, for auxiliary-content transactions the conditions,

$$\begin{gathered}\text{for} \quad 1 \leq k \leq (m-1) : l_k^{t_{rl}}.length \leq C \\ \text{and} \quad k = m : l_k^{t_{rl}}.length > C\end{gathered}$$

are added to (3), and for content-only transactions, the condition,

$$\text{for} \quad 1 \leq k \leq m : l_k^{t_{rl}}.length > C$$

is added to (3). Using the example presented in Sect. 5 with the assumption that the multiple purpose pages are used as content pages half of the time they are accessed, a cutoff time of 78.4 seconds is calculated (Of course, with such a small example, the calculation is statistically meaningless). This results in content-only transactions of `F-G`, `D`, `L-R`, and `J`, as shown in Table 3. Notice that page `L` is classified as a content page instead of an auxiliary page. This could be due to any of the reasons discussed above.

The one parameter that the reference length approach requires is an estimation of the overall percentage of references that are auxiliary. The estimation of the percentage of auxiliary references can be based on the structure and content of the site or experience of the data analyst with other server logs. The results presented in Sect. 8 show that the approach is fairly robust and a wide range of auxiliary percentages will yield reasonable sets of association rules.

6.3 Transaction Identification by Maximal Forward Reference

The maximal forward reference transaction identification approach is based on the work presented in [6]. Instead of time spent on a page, each transaction is defined to be the set of pages in the path from the first page in a user session up to the page before a backward reference is made. A forward reference is defined to be a page not already in the set of pages for the current transaction. Similarly, a backward reference is defined to be a page that is already contained in the set of pages for the current transaction. A new transaction is started when the next forward reference is made. The underlying model for this approach is that the maximal forward reference pages are the content pages, and the pages leading up to each maximal forward reference are the auxiliary pages. Like the reference length approach, two sets of transactions, namely auxiliary-content or content-only, can be formed. The definition of a general transaction shown in

(1) is used within the maximal forward reference approach. Again, using the Sect. 5 example, auxiliary-content transactions of `A-B-F-O`, `A-B-G`, `L-R`, `A-B`, `A-C-J`, and `A-D` would be formed. The content-only transactions would be `O-G`, `R`, `B-J`, and `D`. The maximal forward reference approach has an advantage over the reference length in that it does not require an input parameter that is based on an assumption about the characteristics of a particular set of data.

6.4 Transaction Identification by Time Window

The time window transaction identification approach partitions a user session into time intervals no larger than a specified parameter. The approach does not try to identify transactions based on the model of Sect. 4, but instead assumes that meaningful transactions have an overall average length associated with them. For a sufficiently large specified time window, each transaction will contain an entire user session. Since the time window approach is not based on the model presented in Sect. 4, it is not possible to create two separate sets of transactions. The last reference of each transaction does not correspond to a content reference, the way it does for the auxiliary-content transactions of the reference length and maximal forward reference approaches. If W is the length of the time window, definition 1 applies for transactions identified with the time window approach with the following added condition:

$$(l_m^t.time - l_1^t.time) \leq W$$

Since there is some standard deviation associated with the length of each "real" transaction, it is unlikely that a fixed time window will break a log up appropriately. However, the time window approach can also be used as a merge approach in conjunction with one of the other divide approaches. For example, after applying the reference length approach, a merge time window approach with a 10 minute input parameter could be used to ensure that each transaction has some minimum overall length.

7 Creation of Test Server Log Data

In order to compare the performance of the transaction identification approaches for the mining of association rules presented in Sect. 6, a server log with known rules is needed. As can be seen in Table 3, the three different approaches result in different sets of transactions, even for a simple example. Mining of association rules from actual Web server logs also tends to result in different lists of rules for each approach, and even for the same approach with different input parameters. There is no quantitative method for determining which of the results are better than the others. It was decided to create server logs with generated data for the purpose of comparing the three approaches. The user browsing behavior model presented in Sect. 4 is used to create the data. The data generator takes a file with a description of a Web site as a directed tree or graph and some embedded association rules. The embedded association rules become the "interesting" rules that are checked for during experiments with different transaction identification approaches. The interesting rules in this case are associations that are not born out by the site structure, which will be identified by the site filter discussed in Sect. 4. For each "user", the next log entry is one of three choices, a forward reference, backward reference, or exit. The probability of each choice is taken from the input file, and a random number generator is used to make the decision. If the page reference is going to be a content reference, the time is calculated using a normal distribution and a mean value for the time spent on a page taken from the input file. The times for auxiliary hits are calculated using a gamma distribution. The reason for using a gamma distribution for the auxiliary references and a normal distribution with different averages for the content references is to create an overall distribution that is similar to those seen in real server logs. A pure exponential distribution does not model the initial steep slope of the histogram particularly well. Figure 8 shows a histogram of the reference lengths for a generated data set, which is very similar to the histogram of the real server log data shown in Fig. 7.

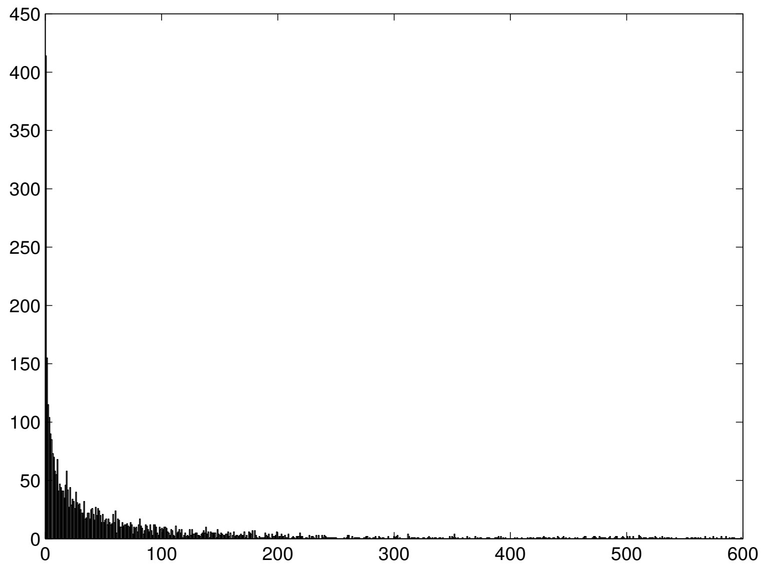

Fig. 8. Histogram of Generated Data Reference Lengths (seconds)

Besides prior knowledge of the interesting association rules, the other advantage of using the generated data for testing the transaction identification approaches is that the actual percentage of auxiliary references is also known. The obvious disadvantage is that it is, after all, only manufactured data and should not be used as the only tool to evaluate the transaction identification

Table 4. Number of Interesting Rules Discovered (number discovered/total possible)

Approach	Parameter	Sparse	Medium	Dense
Time Window	10 min.	0/4	0/3	0/3
	20 min.	2/4	2/3	1/3
	30 min.	2/4	2/3	2/3
Reference Length	50%	4/4	3/3	3/3
	65%	4/4	3/3	3/3
	80%	4/4	3/3	3/3
M. F. R.		4/4	2/3	1/3

approaches. Since the data is created from the user behavior model of Sect. 4, it is expected that the transaction identification approach based on the same model will perform best.

Three different types of web sites were modeled for evaluation, a sparsely connected graph, a densely connected graph, and a graph with a medium amount of connectivity. The sparse graph, with an average incoming order of one for the nodes, is the site used for the examples in Sects. 5 and 6, and is shown in Fig. 5. The medium and dense graphs use the same nodes as the sparse graph, but have more edges added for an average node degree of four and eight, respectively.

8 Experimental Evaluation

8.1 Comparison using Synthetic Data

Table 4 shows the results of using the three transactions identification approaches discussed in section 3 to mine for association rules from data created from the three different web site models discussed in Sect. 7. The reference length approach performed the best, even though it uses a pure exponential distribution to estimate the cutoff time, while the synthetic data was created with a combination of normal and gamma distributions. The maximal forward reference approach performs well for sparse data, but as the connectivity of the graph increases, its performance degrades. This is because as more forward paths become available, a content reference is less likely to be the "maximal forward reference." For dense graphs, auxiliary-content transactions would probably give better results with the maximal forward reference approach. The performance of the time window approach is relatively poor, but as the time window increases, so does the performance.

Table 5 shows the average ratio of reported confidence to actual confidence for the interesting rules discovered. The differences between the reference length and maximal forward reference approaches stand out in Table 5. The reported confidence of rules discovered by the reference length approach are consistently close to the actual values. Note that even though the created data has an actual auxiliary page ratio of 70%, inputs of 50% and 80% produce reasonable results.

Table 5. Ratio of Reported Confidence to Actual Confidence

Approach	Parameter	Sparse	Medium	Dense
Time Window	10 min.	null	null	null
	20 min.	0.82	0.87	0.87
	30 min.	0.98	0.90	0.88
Reference Length	50%	0.99	0.95	0.96
	65%	1.0	0.99	0.96
	80%	0.97	0.99	0.96
M. F. R.		0.79	0.47	0.44

Table 6. Transaction Identification Run Time (sec) / Total Run Time (sec)

Approach	Parameter	Sparse	Medium	Dense
Time Window	10 min.	0.81/4.38	0.82/4.65	0.75/3.94
	20 min.	0.84/7.06	0.80/7.06	0.73/4.42
	30 min.	0.79/7.16	0.77/9.95	0.72/5.17
Ref. Length	50%	1.82/4.62	1.66/4.46	1.47/4.09
	65%	1.68/4.29	1.72/4.35	1.45/4.02
	80%	1.62/4.14	1.66/4.26	1.48/4.03
M. F. R.		1.26/3.98	1.30/3.95	1.20/3.87

The reported confidence for the rules discovered by the maximal forward reference approach is significantly lower than the actual confidence, and similar to the results of Table 4, it degrades as the connectivity of the graph increases.

Table 6 shows the running time of each transaction identification approach, and the total run time of the data mining process. The total run times do not include data cleaning since the data was generated in a clean format. Although the data cleaning step for real data can comprise a significant portion of the total run time, it generally only needs to be performed once for a given set of data. The time window approach shows the fastest run time, but a much slower overall data mining process due to the number of rules discovered. The reference length approach has the slowest approach times due to an extra set of file read/writes in order to calculate the cutoff time. All three of the approaches are O(n) algorithms and are therefore linearly scalable.

8.2 Association Rules from Real Data

Transactions identified with the reference length and maximal forward reference approaches were used to mine for association rules from a server log with data from the Global Reach Web site. The server log used contained 20.3 Mb of raw data, which when cleaned corresponded to about 51.7K references. Because the Global Reach Web server hosts Web sites for other companies, the server log is really a collection of smaller server logs and the overall support for most discovered association rules is low. Accordingly, the association rule generation

Table 7. Examples of Association Rules from www.global-reach.com

Approach Used	Conf.(%)	Supp.(%)	Association Rules
Reference Length (content-only)	61.54	0.18	`/mti/clinres.htm /mti/new.htm` $\Longrightarrow$ `/mti/prodinfo.htm`
Reference Length (content-only)	100.00	0.15	`/mti/Q&A.htm /mti/prodinfo.htm /mti/pubs.htm` $\Longrightarrow$ `/mti/clinres.htm`
Reference Length (content-only)	26.09	0.14	`/cyprus-online/dailynews.htm` $\Longrightarrow$ `/mti/Q&A.htm`
Maximal Forward Reference (content-only)	52.17	0.14	`/cyprus-online/Magazines.htm /cyprus-online/Radio.htm` $\Longrightarrow$ `/cyprus-online/News.htm`
Maximal Forward Reference (aux-content)	73.50	1.32	`/mti/clinres.htm /mti/new.htm` $\Longrightarrow$ `/mti/prodinfo.htm`

algorithm was run with thresholds of 0.1% support and 20% confidence. This led to a fairly large number of computed rules (1150 for the reference length approach and 398 for the maximal forward reference approach). The site filter was not applied to the discovered rules. Table 7 shows some examples of association rules discovered.

The first two rules shown in Table 7 are straight forward association rules that could have been predicted by looking at the structure of the web site. However, the third rule shows an unexpected association between a page of the *cyprus-online* site and a page from the *MTI* site. Approximately one fourth of the users visiting `/cyprus-online/dailynews.htm` also chose to visit `/mti/Q&A.htm` during the same session. However, since the *cyprus-online* page no longer exists on the Global Reach server, it is not clear if the association is the result of an advertisement, a link to the *MTI* site, or some other factor. The fourth rule listed in Table 7 is one of the 150 rules that the maximal forward reference approach discovered that was not discovered by the reference length approach. While the reference length approach discovered many rules involving the *MTI* web site, the maximal forward reference approach discovered relatively few rules involving the *MTI* site. An inspection of the *MTI* site revealed that the site is a fully connected graph. Consistent with the results of Sect. 8.1, the maximal forward reference approach does not perform well under these conditions. The association rule algorithm was run with auxiliary-content transactions created from the maximal forward reference approach to confirm the theory that the rules missed by the content-only transactions would be discovered. The last rule listed in Table 7 is the same as the first rule listed, and shows that the auxiliary-content transactions from the maximal forward reference approach can discover rules in a highly connected graph. However, at thresholds of 0.1% support and 20% confidence, approximately 25,000 other rules were also discovered with the auxiliary-content approach.

9 Conclusions

This paper has presented the details of preprocessing tasks that are necessary for performing *Web Usage Mining*, the application of data mining and knowledge discovery techniques to WWW server access logs. This paper also presented experimental results on synthetic data for the purpose of comparing transaction identification approaches, and on real-world industrial data to illustrate some of its applications (Sect. 8). The transactions identified with the *reference length* approach performed consistently well on both the real data and the created data. For the real data, only the reference length transactions discovered rules that could not be reasonably inferred from the structure of the Web sites. Since the important page in a traversal path is not always the last one, the *content-only* transactions identified with the *maximal forward reference* approach did not work well with real data that had a high degree of connectivity. The *auxiliary-content* transactions led to an overwhelmingly large set of rules, which limits the value of the data mining process. Future work will include further tests to verify the user browsing behavior model discussed in Sect. 4 and a more rigorous analysis of the shape of reference length histograms in order to refine the reference length transaction identification approach.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, pages 487–499, Santiago, Chile, 1994.
2. T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible markup language (XML) 1.0 W3C recommendation. Technical report, W3C, 1998.
3. M. Balabanovic and Y. Shoham. 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*, 1995.
4. R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the World Wide Web. In *International Conference on Tools with Artificial Intelligence*, pages 558–567, Newport Beach, CA, 1997.
5. L. Catledge and J. Pitkow. Characterizing browsing behaviors on the World Wide Web. *Computer Networks and ISDN Systems*, 27(6), 1995.
6. M.S. Chen, J.S. Park, and P.S. Yu. Data mining for path traversal patterns in a Web environment. In *Proceedings of the 16th International Conference on Distributed Computing Systems*, pages 385–392, 1996.
7. S. Elo-Dean and M. Viveros. Data mining the IBM official 1996 Olympics Web site. Technical report, IBM T.J. Watson Research Center, 1997.
8. e.g. Software Inc. Webtrends. *http://www.webtrends.com*, 1995.
9. J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. In *IEEE 12th International Conference on Data Engineering*, pages 152–159, 1996.
10. Open Market Inc. Open Market Web reporter. *http://www.openmarket.com*, 1996.
11. T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the World Wide Web. In *Proc. of the 15th International Conference on Artificial Intelligence*, pages 770 – 775, Nagoya, Japan, 1997.

12. L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley & Sons, 1990.
13. H. Lieberman. Letizia: An agent that assists Web browsing. In *Proc. of the 1995 International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995.
14. A. Luotonen. The common log file format. *http://www.w3.org/pub/WWW/*, 1995.
15. B. Mobasher, N. Jain, E. Han, and J. Srivastava. Web Mining: Pattern discovery from World Wide Web transactions. Technical Report TR 96-050, University of Minnesota, Dept. of Computer Science, Minneapolis, 1996.
16. H. Mannila and H. Toivonen. Discovering generalized episodes using minimal ocurrences. In *Proc. of the Second Int'l Conference on Knowledge Discovery and Data Mining*, pages 146–151, Portland, Oregon, 1996.
17. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovering frequent episodes in sequences. In *Proc. of the First Int'l Conference on Knowledge Discovery and Data Mining*, pages 210–215, Montreal, Quebec, 1995.
18. net.Genesis. net.analysis desktop. *http://www.netgen.com*, 1996.
19. R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.
20. D.S.W. Ngu and X. Wu. Sitehelper: A localized agent that helps incremental exploration of the World Wide Web. In *6th International World Wide Web Conference*, pages 691–700, Santa Clara, CA, 1997.
21. J. Pitkow. In search of reliable usage data on the WWW. In *Sixth International World Wide Web Conference*, pages 451–463, Santa Clara, CA, 1997.
22. M. Pazzani, L. Nguyen, and S. Mantik. Learning from hotlists and coldlists: Towards a WWW information filtering and seeking agent. In *IEEE 1995 International Conference on Tools with Artificial Intelligence*, 1995.
23. P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the Web. In *Proc. of 1996 Conference on Human Factors in Computing Systems (CHI-96)*, Vancouver, British Columbia, Canada, 1996.
24. Global Reach Internet Productions. GRIP. *http://www.global-reach.com*, 1997.
25. J. Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.
26. R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proc. of the Fifth Int'l Conference on Extending Database Technology*, Avignon, France, 1996.
27. S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict HTTP requests. In *7th International World Wide Web Conference*, Brisbane, Australia, 1998.
28. C. Shahabi, A. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users Web-page navigation. In *Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997.
29. T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Fifth International World Wide Web Conference*, Paris, France, 1996.
30. O. R. Zaiane, M. Xin, and J. Han. Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs, 1998.