Discovery of Interesting Usage Patterns from Web Data

Robert Cooley[*], Pang-Ning Tan, Jaideep Srivastava[**]

{cooley,ptan,srivasta}@cs.umn.edu
Department of Computer Science and Engineering
University of Minnesota

Abstract. *Web Usage Mining* is the application of data mining techniques to large Web data repositories in order to extract usage patterns. As with many data mining application domains, the identification of patterns that are considered *interesting* is a problem that must be solved in addition to simply generating them. A necessary step in identifying interesting results is quantifying what is considered uninteresting in order to form a basis for comparison. Several research efforts have relied on manually generated sets of uninteresting rules. However, manual generation of a comprehensive set of evidence about beliefs for a particular domain is impractical in many cases. Generally, domain knowledge can be used to automatically create evidence for or against a set of beliefs. This paper develops a quantitative model based on *support logic* for determining the interestingness of discovered patterns. For Web Usage Mining, there are three types of domain information available; *usage, content*, and *structure*. This paper also describes algorithms for using these three types of information to automatically identify interesting knowledge. These algorithms have been incorporated into the Web Site Information Filter (WebSIFT) system and examples of interesting frequent itemsets automatically discovered from real Web data are presented.

1 Introduction and Background

The World Wide Web continues to expand at an amazing rate as a medium for conducting business and disseminating information. Even with evolving standards and technology, the ability to thoroughly analyze the usage of a Web site remains, and will grow, as an important capability for Web administrators. Design of a Web site centers around organizing the information on each page and the hypertext links between the pages in a way that seems most natural to the site users, to facilitate their browsing. For small sites, an individual Web designer's intuition along with some straightforward usage statistics may be adequate for predicting and verifying the users' browsing behavior. However, as the size and complexity of a Web site increases, the statistics provided by existing Web log analysis tools [1–3] may prove inadequate, and more sophisticated types

[*] Supported by NSF grant EHR-9554517
[**] Supported by ARL contract DA/DAKF11-98-P-0359

of analyses will be necessary. *Web Usage Mining*, which is the application of data mining techniques to large Web data repositories, adds powerful techniques to the tools available to a Web site administrator for analyzing Web site usage.

Web Usage Mining techniques developed in [8, 9, 11, 16, 19, 25, 27, 30] have been used to discover frequent itemsets, association rules [5], clusters of similar pages and users, sequential patterns [15], and perform path analysis [9]. Several research efforts [17, 13] have considered usage information for performing *Web Content Mining* [10]. An overview of some of the challenges involved in Web Content Mining is given in [28].

The notion of what makes discovered knowledge interesting has been addressed in [14, 18, 20, 26]. A common theme among the various criteria for interestingness is the concept of *novelty* or *unexpectedness* of a rule. Results that were previously known by the data analyst are not considered interesting. In Web Usage Mining, as with many data mining domains, thresholds for values such as *support* and *confidence* are often used to limit the number of discovered rules to a manageable number. However, high thresholds rarely discover any knowledge that was not previously known and low thresholds usually result in an unmanageable number of rules. The approach advocated by [14, 18] is to identify a set of *beliefs*, and use the set as a filter for identifying interesting rules. Rules that confirm existing beliefs are deemed uninteresting.

In a more general sense, both the discovered knowledge and any expectations defined from domain knowledge can be considered as pieces of evidence providing support *for* or *against* a particular belief. There can be multiple sources of evidence pertaining to any given belief about a domain, some of them possibly contradictory. Also, as pointed out in [14], evidence about some of the beliefs is likely to be imprecise or incomplete, requiring a framework with fuzzy logic [29] capabilities. A framework based on Baldwin's *support logic* [6] can be defined, which is specifically designed to handle reasoning about multiple sources of evidence with both boolean and fuzzy logic and includes an explicit accounting of ignorance regarding a belief. The framework is built around defining *support pairs* for every piece of evidence.[1]

Another problem that exists with the identification of interesting results is the generation of an initial set of evidence about beliefs from domain knowledge. Both [14] and [18] rely on manually generated sets of evidence. For [18], beliefs are only defined as interesting if there is conflicting evidence, so unless a fairly comprehensive set is created, many interesting results can be missed. [14] has a broader definition of interestingness that includes results that provide evidence about a belief not covered by domain knowledge. However, without a comprehensive set of evidence from domain knowledge, this method will end up misclassifying many results.

The Web Usage Mining domain has several types of information available that can be used as surrogates for domain knowledge. Using this information, a

[1] In order to avoid confusion with the standard data mining definition of support, Baldwin's support pairs will be referred to as *evidence pairs* for the rest of this paper.

large and fairly comprehensive set of evidence can be automatically generated to effectively filter out uninteresting results from the Web Usage Mining process.

The specific contributions of this paper are:

- Development of a general quantitative model of what determines the interestingness of discovered knowledge, based on Baldwin's support logic framework [6].
- Development of an approach for the automatic creation of an initial set of evidence about a belief set.
- Development of specific algorithms for automated discovery of interesting rules in the Web Usage Mining domain.
- Presentation of results from a Web Usage Mining system called the Web Site Information Filter (WebSIFT) system, using data collected from an actual Web site.

The rest of this paper is organized as follows: Section 2 defines the different types of Web data and information abstractions suitable for usage mining. Section 3 develops a general support logic based framework for defining and combining evidence about a domain. A formal definition of interestingness is also given in this section. Section 4 describes algorithms that can be used for automatically identifying interesting frequent itemsets and Section 5 presents an overview of the WebSIFT system. Section 6 summarizes some results from tests of the WebSIFT system on a Web server log. Finally, section 7 provides conclusions.

2 Data Sources and Information Abstractions

Web Usage Mining analysis can potentially use many different kinds of data sources, as discussed in [21]. This paper classifies such data into the following broad types:

- **Content:** The *real* data in the Web pages, i.e. 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:** The data which describes the organization of the content. *Intra-page* structure information includes HTML or XML tags of various kinds, the sequence in which they appear, etc. The principal kind of *inter-page* structure information is hyper-links connecting one page to another.
- **Usage:** The data that describes the pattern of usage of Web pages, such as IP addresses, page references, and the date/time of accesses. This information can be obtained from Web server logs.

The World Wide Web Committe (W3C) Web Characterization Activity [4] has defined several data abstractions that are useful for Web Usage mining, such as *page view*, *server session*, and *click stream* that are based on the data types listed above. A page view is defined by 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 *user session* is the click-stream of page views for a single user across the entire Web. Typically, only the portion of each user session that is accessing a specific site can be used for analysis, since access information is not publicly available from the vast majority of Web servers. The set of page-views in a user session for a particular Web site is referred to as a *server session* (also commonly referred to as a *visit*). The term *user episode* refers to a subset of page views in a user session. In addition, Web pages can be classified into various types based on their content, structure and other attributes. For example, the ratio of the number of links in a page to the number of text units (say words) can be used as a measure for classifying pages into various types such as *navigational*, *content*, or *hybrid*. This issue is discussed in detail in [11].

Various kinds of analyses can be performed on these abstractions to extract knowledge useful for a variety of applications. A specific type of analysis is to make assertions about the aggregate usage behavior of all users who visit pages of a Web site. For example, the assertion can be made that a pair of pages that have structural proximity (due to hyperlinks between them) and/or content proximity (since they have information on closely related topics), are likely to be visited together often. Analysis of structure and content information can be used to make the initial assertion, and subsequent analysis of usage data can be used to examine the truth of such an assertion.

Note that in the above assertion, words such as *likely* and *often* are used rather than *will* and *always*. In an inductive analysis scenario with many sources of uncertainty, the first set of words more accurately captures the nature of assertions that can be made, making standard predicate logic too brittle a reasoning framework. Hence, the framework of *support logic* [6] is used for analysis, as described in the next section.

3 Evaluation of Beliefs in a Support Logic Framework

3.1 Measures of Interestingness

The ultimate goal of any data mining effort is to provide the analyst with results that are interesting and relevant to the task at hand. [26] defines two types of interestingness measures - *objective* and *subjective*. *Objective* measures rate rules based on the data used in the mining process. Thresholds on *objective* measures such as confidence, support, or chi-square [7] are invaluable for reducing the number of generated rules, but often fall well short of the goal of only reporting rules that are of potential interest to the analyst.

For *subjective* measures of interestingness, [26] defines two criteria to evaluate rules and patterns. A rule is *unexpected* if it is "surprising" to the data analyst, and *actionable* if the analyst can act on it to his advantage. The degree to which a rule is *actionable* depends on its application. Consider the use of association rules to restructure a Web site. Since the topology or content of a Web site can be modified based on any discovered information, all rules are *actionable* for this application. [18] formally defines the unexpectedness of a rule in terms of its deviation from a set of beliefs. [14] has a broader definition of interestingness that includes discovered rules that are not specifically covered by an initial set of beliefs. In other words, a rule that doesn't contradict an existing belief, but points out a relationship that hadn't even been considered is also interesting. While both [14] and [18] give examples of small sets of manually generated beliefs, neither addresses the issue of automated generation of a realistic belief set from a large amount of data.

3.2 Support Logic

A more general way to look at the problem of identifying the interestingness of discovered patterns is to consider each piece of information in terms of the evidence it gives *for* or *against* a given logical statement (belief). Baldwin's *support logic* [6, 23], which is an implementation of the Dempster-Schafer theory of evidence [24], provides a framework for this point of view. For a belief $\mathcal{B}$, evidence collected for or against $\mathcal{B}$ can be used to form an *evidence pair*, $[e_n, e_p]$, where:

$$
\begin{aligned}
e_n &= \text{necessary evidence in support of } \mathcal{B} && (1)\\
e_p &= \text{possible evidence in support of } \mathcal{B} && (2)\\
(1 - e_p) &= \text{necessary evidence in support of } \neg\mathcal{B} && (3)\\
(1 - e_n) &= \text{possible evidence in support of } \neg\mathcal{B} && (4)\\
(e_p - e_n) &= \text{uncertainty of } \mathcal{B} && (5)
\end{aligned}
$$

The values of e_n and e_p must satisfy the constraints:

$$
\begin{aligned}
e_n + (1 - e_p) &\leq 1 && (6)\\
e_n \geq 0, e_p &\geq 0 && (7)
\end{aligned}
$$

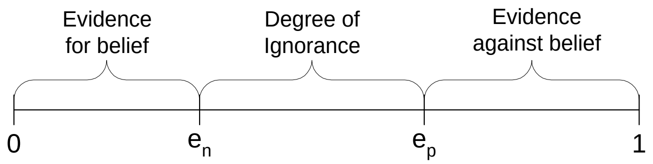

Fig. 1. Evidence pair values for a belief

Figure 1 shows the concepts that map to each region of a belief scale, given e_n and e_p. If e_n and e_p are equal, the situation reduces to probabilistic uncertainty.

When e_n and e_p are not equal, the difference between the values represents the amount of ignorance about a belief. Note that ignorance, or lack of evidence, is fundamentally different than uncertainty. The uncertainty of a fair coin flip coming up heads is known to be 0.5. However, not enough is known about many real life situations to attach a definitive probabilistic value. Instead of assigning a default probability, Dempster-Schafer theory allows the assignment of an interval indicating that there is missing evidence about a particular belief. Another way to think of ignorance is the lack of confidence in the probabilistic values assigned to a belief. As an example, assume that evidence has been collected about the belief $\mathcal{B}(X, Y)$, that Web pages X and Y are related. If all of the evidence is in support of $\mathcal{B}(X, Y)$ and one is completely confident in the evidence, the evidence pair is $[1, 1]$. On the other extreme, if all of the evidence is against $\mathcal{B}(X, Y)$, the evidence pair is $[0, 0]$. If the data leads to a 25% degree of belief that $\mathcal{B}(X, Y)$ is true,and a 40% degree of belief that $\mathcal{B}(X, Y)$ is false, then $[0.25, 0.6]$ would represent the appropriate evidence pair. This says that the degree of ignorance about $\mathcal{B}(X, Y)$ is 35%. Finally, if there is no evidence pertaining to $\mathcal{B}(X, Y)$, the evidence pair is $[0, 1]$, giving a complete lack of confidence, or a degree of ignorance of 100%. Independent of the type of the source for generating an evidence pair, pairs can be combined per Baldwin's *support logic programming calculus* [6] to obtain a single evidence pair per belief. The basic rules are as follows:

If $\mathcal{B}$:$[e_{1n}, e_{1p}]$ AND $\mathcal{B}$:$[e_{2n}, e_{2p}]$ are two independent evidence pairs from different sources about belief $\mathcal{B}$, then conclude $\mathcal{B}$:$[e_n, e_p]$, where

$$e_n = [e_{1n}e_{2n} + e_{1n}(e_{2p} - e_{2n}) + e_{2n}(e_{1p} - e_{1n})]/K \tag{8}$$

$$\begin{aligned} 1 - e_p = [&(1 - e_{1p})(1 - e_{2p}) + (e_{1p} - e_{1n})(1 - e_{2p}) + \\ &(e_{2p} - e_{2n})(1 - e_{1p})]/K \end{aligned} \tag{9}$$

$$K = 1 - e_{1n}(1 - e_{2p}) - e_{2n}(1 - e_{1p}) \tag{10}$$

All beliefs have a default evidence pair value of [0,1] until some data is introduced that pertains to that belief. As subsequent data relevant to a belief is analyzed, the evidence pair can be updated using equations 8, 9, and 10. For any data mining domain, the rules that are generated can be used to initialize a set of evidence pairs. A second set of evidence pairs can be generated from domain knowledge or from another knowledge discovery algorithm. Building on the support logic framework, an interesting result can be defined as either a belief with a combined evidence pair that is significantly different from one of the original evidence pairs, or original evidence pairs that are significantly different from each other. "Significantly different" can be determined by setting a threshold value, $\mathcal{T}$, for differences in both e_n and e_p. A formal definition of interesting can be defined as follows[2]:

[2] While this definition uses the familiar L2-norm, other norms could be substituted as appropriate.

For a belief, $\mathcal{B}$ with an interestingness pair $\mathcal{I}(n_i, p_i)$, where

$$n_i = |e_n^{(x)} - e_n^{(y)}| \tag{11}$$

$$p_i = |e_p^{(x)} - e_p^{(y)}| \tag{12}$$

$\mathcal{B}$ is interesting if:

$$\mathcal{T} \leq \sqrt{n_i^2 + p_i^2} \tag{13}$$

In the definition above, the x and y superscripts designate values from different evidence pairs. Since the interestingness of a belief is defined by a real-value, an ordering among interesting beliefs can also be established. In the simplest case, all evidence is either 100% for a belief, 100% against a belief, or there is no evidence about a belief. This leads to nine different "boundary" cases that can occur when comparing evidence generated from two separate sources. These are shown in Table 1, along with the three types of comparisons that can be made. For the two cases where the evidence pairs are in complete disagreement, the combined evidence pair is "Null." This is because completely contradictory evidence can not, and should not be automatically reconciled. Comparing one of the original evidence sources with the combined evidence identifies beliefs with conflicting evidence along with evidence only represented in the other source as interesting. This is useful when one set of evidence is considered to be established or "known" and a second set of evidence is "new." By comparing the known evidence pairs to the combined evidence pairs, all of the previously unknown and conflicting results will be labeled as interesting. If the two evidence sources are directly compared, all beliefs that have evidence from only one of the sources will be declared interesting in addition to any conflicting beliefs. This may be desirable for situations when both sources of evidence are considered to be new. By setting an appropriate threshold $\mathcal{T}$ and choosing which evidence pairs will be compared, any combination of the following situations can be automatically labeled as interesting:

- Beliefs with conflicting evidence.
- Beliefs with evidence from source 1 but not source 2.
- Beliefs with evidence from source 2 but not source 1.

Note that the definitions of interestingness from both [18] and [14] are included in this framework.

3.3 Generation of Belief Sets for Web Usage Mining

For Web Usage Mining, there are two additional sources from which evidence pairs can be automatically created; the content and structure data (as discussed earlier, evidence can also be manually generated by a domain expert). The task of reconciling conflicting evidence from the content and structure data falls under the category of *Web Content Mining*, which is beyond the scope of this paper.

Table 1. Comparison of Boolean Evidence Pairs from Separate Sources

Evidence			Interestingness ($\mathcal{T} = 1$)		
Source 1	Source 2	Combined	Source 1 vs. Combined	Source 2 vs. Combined	Source 1 vs. Source 2
[0,0]	[0,0]	[0,0]	No	No	No
[0,0]	[0,1]	[0,0]	No	Yes	Yes
[0,0]	[1,1]	Null	Yes	Yes	Yes
[0,1]	[0,0]	[0,0]	Yes	No	Yes
[0,1]	[0,1]	[0,1]	No	No	No
[0,1]	[1,1]	[1,1]	Yes	No	Yes
[1,1]	[0,0]	Null	Yes	Yes	Yes
[1,1]	[0,1]	[1,1]	No	Yes	Yes
[1,1]	[1,1]	[1,1]	No	No	No

Table 2. Examples of Web Usage Information that can be automatically flagged as Interesting

Source 1	Source 2	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 frequent itemset contains pages that are not directly linked
Usage Clusters	Site Content	A usage cluster contains pages from multiple content clusters

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 e_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 e_n and e_p based on the calculated "distance" between pages. Table 2 gives some examples of the types of interesting beliefs that can be identified in the Web Usage Mining domain using the framework described in the previous section.

4 Filtering of Knowledge based on Interestingness

4.1 Evidence from Structure Information

The use of structure information to guide the knowledge discovery process in Web Mining has been discussed by several authors [16, 19, 21, 25]. Most of their work is focused on using the site structure to perform clustering on Web pages or user path profiles. However, the utilization of structure information during knowledge analysis (in particular, for automated filtering of uninteresting results) has been largely ignored.

There are several ways to accommodate structure information into the filtering phase of a Web Mining system. This section introduces one approach for incorporating structural evidence into the support logic framework for filtering uninteresting frequent itemsets. The goal is to obtain a structural evidence pair $E^{(s)} = [e_n^{(s)}, e_p^{(s)}]$ that will represent the belief that a set of pages are related. Any suggested approach for quantifying $E^{(s)}$ must satisfy the following minimum requirements :

- **Consistency:** The values of $e_n^{(s)}$ and $e_p^{(s)}$ are subjected to the constraints given in equations 6 and 7 (Namely, the values are between 0 and 1, and the sum of $e_n^{(s)} + e_p^{(s)}$ is not greater than 1).
- **Reducibility:** The structural evidence pair for a large itemset can be calculated from the evidence pairs of its constituent itemsets. Furthermore, the rules for combining the evidence pair for the smaller itemsets must be consistent, i.e. the combined evidence pair for the larger itemset must be the same irrespective of the order in which the itemsets are combined.
- **Monotonicity:** The necessary structural evidence, $e_n^{(s)}$, should increase monotonically as the number of links connecting the pages within an itemset increases.
- **Connectivity:** If the graph representing an itemset is connected, its $e_n^{(s)}$ should be large compared to one that is not connected (when both graphs contain the same number of links and vertices).

One method for calculating $e_n^{(s)}$ that meets the requirements listed above is to use a combination of the following two parameters: the link factor and connectivity factor. Link factor (lfactor) is a normalized measure for the number of links present among the pages in an itemset.

$$\text{lfactor} = \frac{L}{N(N-1)} \tag{14}$$

where N is the number of pages in the itemset and L is the number of direct hyperlinks between them. The denominator ensures that the consistency requirement is satisfied. Furthermore, one can verify that both monotonicity and reducibility requirements are obeyed by the lfactor measure.

The connectivity requirement can be captured in a simple way by introducing a connectivity factor (cfactor) which is defined as

$$\text{cfactor} = \begin{cases} 1, & \text{if G(I) is connected;} \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

where G(I) is the graphical representation for itemset I. The necessary evidence can now be defined as :

$$e_n^{(s)} = \text{lfactor} \;\; \text{x} \;\; \text{cfactor} \tag{16}$$

$e_p^{(s)}$ can be set anywhere between $e_n^{(s)}$ and 1, depending on the desired degree of ignorance. The experiments described in Section 6 use $e_n^{(s)} = e_p^{(s)}$.

4.2 Evidence from Usage Information

Mined results in the form of frequent itemsets can be used to provide evidence for pages being related. In order to derive an evidence pair from a frequent itemset, a single measure of the strength of the relationship between the pages is needed. This is normally done by breaking an N item frequent itemset up into N separate association rules, and reporting the confidence for each rule. However, this method results in several rules about the same set of pages, all with potentially different confidence levels. Since the order and number of page accesses for a user session have been removed from frequent itemsets, this expansion of the discovered rules does not make sense. A measure other than support that can be calculated for frequent itemsets is the *coverage*. The coverage of a rule is the fraction of the total number of transactions that contain *at least one* of the items in the itemset (as opposed to support, which measures the fraction of transactions that contain *all* of the items). Support, $\mathcal{S}$, and coverage, $\mathcal{C}$, for a frequent itemset with items i_1 through i_n are defined as follows, where Count(predicate) is the number of transactions containing the predicate, and N_T is the total number of transactions:

$$\mathcal{S} = \frac{\text{Count}(i_1 \wedge i_2 \ldots \wedge i_n)}{N_T} \tag{17}$$

$$\mathcal{C} = \frac{\text{Count}(i_1 \vee i_2 \ldots \vee i_n)}{N_T} \tag{18}$$

Notice that both support and coverage are highly dependent on the total number of transactions. By taking the ratio of support to coverage, this dependency is eliminated. The support-to-coverage ratio (SCR) gives a single measure of the strength of a frequent itemset that is independent of the total number of transactions in the database. Essentially, the SCR is the support of a frequent itemset when only considering the transactions that contain at least one item in the itemset. The SCR for a frequent itemset can be calculated using the algorithm shown in Table 3, which is called after the completion of each level of a frequent itemset generation algorithm, such as Apriori [5] (or one of its variants). The Table 3 algorithm is based on the fact that for any frequent itemset, the supports or counts for all of the contributing subset itemsets have already been calculated.

For a given frequent itemset, the SCR provides evidence for, and (1-SCR) provides evidence against a set of pages being related to each other. Therefore, a simple evidence pair for usage evidence that does not take any degree of ignorance into account is [SCR,SCR].

Table 3. SCR Algorithm

```
Algorithm SCR
let F = $\{I_1, I_2, \cdots, I_n\}$ denote the discovered frequent itemset
cover = 0
for level $l$ = 1 to $n$
    lcount = CountSum(itemsets $\subseteq$ F)
    cover = cover + $(-1)^{l+1}$ * lcount
SCR = Count(F)/cover
end;
```

4.3 Evidence Combination

The remaining issue before using the support logic calculus to combine the structural and usage evidence is scaling. Since the two sets of evidence are derived in different manners from different sets of data, the scales do not necessarily match. For the usage data, a factor that has not been considered in the generation of the evidence pairs is user attrition. Several studies summarized in [22] have found that the mean path length of user sessions is typically about 3 pages with a heavy tailed distribution. Therefore, as the number of pages in a belief increases, the less likely it is that a corresponding frequent itemset will be discovered, simply because of user attrition. However, the strength of the corresponding domain evidence pair does not necessarily decrease as the size of the set increases. In order to account for this, one set of evidence pairs needs to be scaled based on the size of the page set. The WebSIFT information filter simply uses the number of pages in the set as its scaling factor as follows:

$$\text{sfactor} = n \tag{19}$$

Once the evidence pairs are scaled, the evidence combination rules presented in Section 3 are used to calculate the combined evidence pairs. Either the mined or domain evidence pair can be taken as the "existing" evidence to be compared with the combined evidence. The algorithm for creating, combining, and comparing evidence pairs is shown in Table 4.

5 The WebSIFT System

The WebSIFT system[3] divides the Web Usage Mining process into three main parts, as shown in Figure 2. For a particular Web site, the three server logs - access, referrer, and agent (often combined into a single log), the HTML files, template files, script files or databases that make up the site content , and any optional data such as registration data or remote agent logs provide the information to construct the different information abstractions defined in Section 2. The preprocessing phase uses the input data to construct a server session file

[3] Based on the WEBMINER prototype [10].

Table 4. Frequent Itemset Filter Algorithm

```
Algorithm Filter
for each F in the discovered frequent itemsets
    $e_p^{(m)} = e_n^{(m)} = \mathrm{SCR(F)} * \mathrm{sfactor(F)}$
    $e_p^{(s)} = e_n^{(s)} = \mathrm{lfactor(F)} * \mathrm{cfactor(F)}$
    $[e_p^{(c)}, e_n^{(c)}] = \mathrm{BaldwinCombine}(e_p^{(m)}, e_n^{(m)}, e_p^{(s)}, e_n^{(s)})$
    let x = m or s per user input
    If $\mathrm{Interest}(e_p^{(x)}, e_n^{(x)}, e_p^{(c)}, e_n^{(c)}) \geq \mathcal{T}$
        Add F to InterestingSets
end;
```

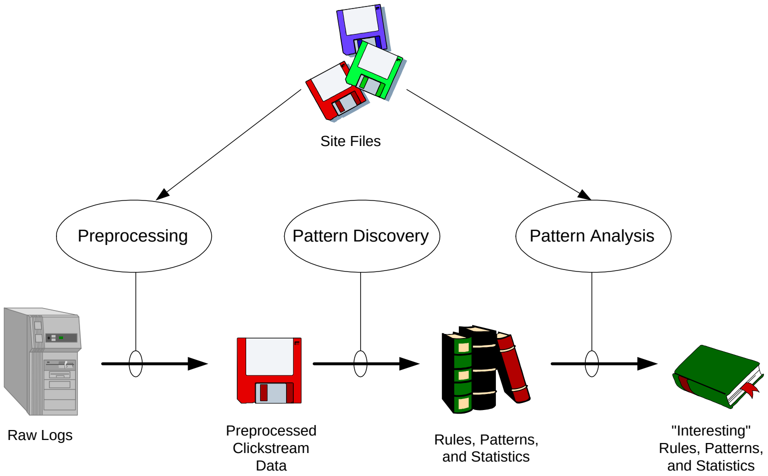

Fig. 2. High Level *Web Usage Mining* Process

based on the methods and heuristics discussed in [11]. In order to preprocess a server log, the log must first be “cleaned”, which consists of removing unsuccessful requests, parsing relevant CGI name/value pairs and rolling up file accesses into page views. Once the log is converted into a list of page views, users must be identified. In the absence of cookies or dynamically embedded session IDs in the URIs, the combination of IP address and user agent can be used as a first pass estimate of unique users. This estimate can be refined by using the referrer field, as described in [11]. The click-stream for each user is divided up into sessions based on a simple thirty minute timeout. Finally, path completion is performed by again looking at the referrer information for each request. These steps are shown in Figure 3. The preprocessing phase of the WebSIFT system allows the option of converting the server sessions into *episodes* prior to performing knowledge discovery. In this case, episodes are either all of the page views in a server session that the user spent a significant amount of time viewing (assumed to be a content page), or all of the navigation page views leading up to each content page view. The details of how a cutoff time is determined for classifying a page view as content or navigation are also contained in [11].

Preprocessing for the content and structure of a site involves assembling each page view for parsing and/or analysis. Page views are accessed through HTTP requests by a "site crawler" to assemble the components of the page view. This handles both static and dynamic content. In addition to being used to derive a site topology, the site files are used to classify the pages of a site. Both the site topology and page classifications can then be fed into the *information filter*. While classification of the site content is really a data mining process in its own right, because it is being used in a supporting role for Web Usage mining, it has been included in the preprocessing phase.

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. Clustering can be performed on either the users or the page views. The discovered information can then be fed into various pattern analysis tools. The current implementation includes the information filter, an association rule graph/visualization tool, and querying of the results through SQL. The WebSIFT system has been implemented using a relational database, procedural SQL, and the Java programming language. Java Database Connectivity (JDBC) drivers are used to interface with the database. Although algorithms have been identified and tested for individual portions of the system, only the generation and filtering of frequent itemsets, association rules, and general statistics is fully automated at this time.

6 Experimental Evaluation

The experiments described in this section were performed on Web server logs from February 1999 at the University of Minnesota Department of Computer Science and Engineering Web site; http://www.cs.umn.edu/.

6.1 Preliminary Experiments

To test the feasibility of filtering discovered rules based on site structure, two simple preliminary tests were run. All of the discovered itemsets were assigned an evidence pair of [1,1] (100% belief that the pages are related), and sets of pages without a frequent itemset were assigned an evidence pair of [0,0]. Any frequent itemset that represented a set of pages not directly connected by hypertext links was declared to be potentially interesting. This is analogous to the boundary case of one source providing evidence for a belief with no corresponding evidence from the second source . The second test took all of the connected pairs of pages that had sufficient individual support, and looked for corresponding frequent itemsets. Pairs of pages that did not have a corresponding frequent itemset were also declared to be interesting. This is the boundary case where there is

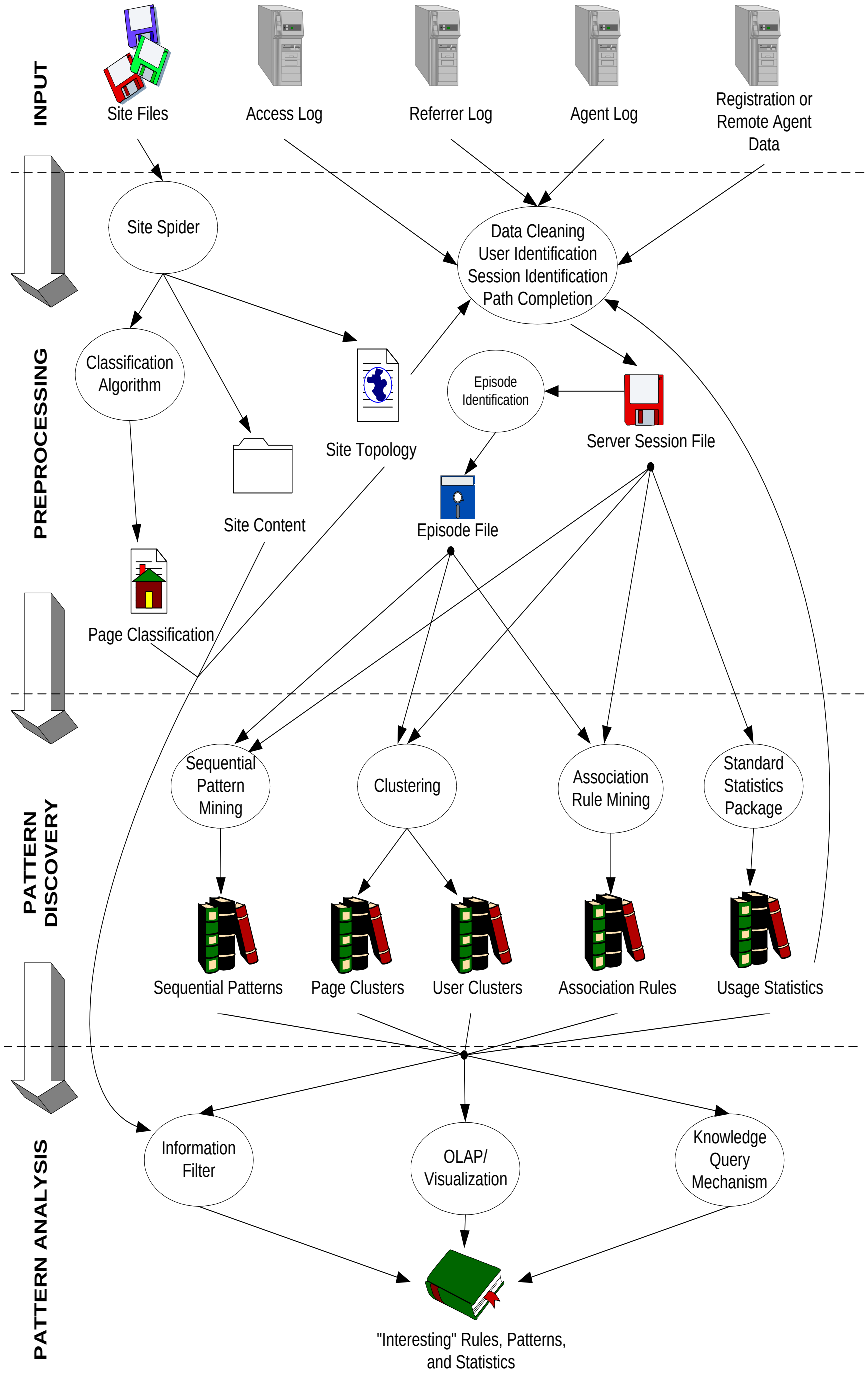

Fig. 3. WebSIFT Architecture

Table 5. Frequent Itemsets with Usage Evidence but no Structural Evidence

#	Mined Support(%)	Related Pages
1	0.10	/Research/, /tech_reports/
2	0.10	/employment/, /newsletter/
3	0.10	/faculty/, /newsletter/
4	0.10	/icra99/ICRA99-Index.htm, /icra99/Notice.html, /icra99/TechnProgram.htm, /icra99/advanceprogram2.htm
5	0.10	/new/, /sem-coll/
6	0.10	/reg-info/98-99_schedule.html, /reg-info/ss1-99.html, /reg-info/ss2-99.html
7	0.11	/Research/Agassiz/, /faculty/
8	0.11	/icra99/Notice.html, /icra99/best.html
9	0.11	/icra99/Proceeding-Order.htm, /icra99/Registration.htm
10	0.22	/grad-info/, /grad-info/97-98-grad-handbook.html
11	0.25	/grad-info/, /grad-info/96-97-grad-handbook.html

Table 6. Itemsets with Conflicting Evidence

#	Web Pages
1	/Research/Agassiz/agassiz_pubs.html, /Research/Agassiz/agassiz_people.html
2	/Research/GIMME/tclprop.html, /Research/GIMME/Nsync.html
3	/Research/airvl/minirob.html, /Research/airvl/loon.html
4	/Research/mmdbms/home.shtml, /Research/mmdbms/group.html
5	/newsletter/kumar.html, /newsletter/facop.html
6	/newsletter/letter.html, /newsletter/facop.html
7	/newsletter/letter.html, /newsletter/kumar.html
8	/newsletter/newfac.html, /newsletter/facop.html
9	/newsletter/newfac.html, /newsletter/kumar.html
10	/newsletter/newfac.htm, /newsletter/letter.html

conflicting evidence. These tests are referred to as the BME (Beliefs with Mined Evidence) and BCE (Beliefs with Conflicting Evidence) in [12].

The processed log consisted of 43,158 page views divided among 10,609 user sessions. A threshold of 0.1% for support was used to generate 693 frequent itemsets, with a maximum set size of six pages. There were 178 unique pages represented in all of the rules. Both methods described in the previous section were run on the frequent itemsets. The first method resulted in 11 frequent itemsets being declared as potentially interesting, and the second method resulted in 10 missing page pairs being declared as potentially interesting. Tables 5 and 6 show the interesting results identified by each algorithm.

Of the frequent itemsets shown in Table 5, the two including the graduate handbook (numbers 10 and 11) are of note because these pages are out-of-date. A page with the 1998-99 graduate handbook exists, and the links from the /grad-info/ page to the older handbooks have been removed. However, since the pages were not actually removed from the site and other pages in the site reference them (or users have old bookmarks), the older handbooks are still accessed. The supports of these itemsets are 0.25% and 0.22% respectively. Had the support threshold been set higher to limit the total number of itemsets discovered, the rules would have been missed.

In Table 6, the fourth pair of pages is of note because the first page functions solely as an entry page to a particular research group's pages. However, the link from the first page is flashing and located fairly low on the page. This indicates a design problem since not all of the visitors from the first page are visiting the second.

6.2 Interesting Frequent Itemsets

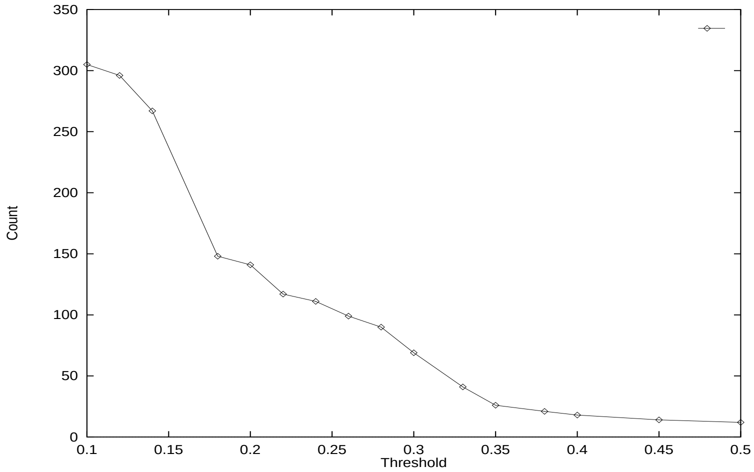

Fig. 4. Number of Interesting Itemsets with different Threshold Values

For this set of experiments, the processed log consisted of 31,584 page views divided among 8175 user sessions. A threshold of 0.1% for support was used to generate 1345 frequent itemsets, with a maximum set size of nine pages. There were 363 unique pages represented in all of the rules. The usage and structure evidence pairs were calculated and combined as described in Section 4. Figure 4 shows the number of rules that are declared to be interesting for a range of

Table 7. Interesting Frequent Itemsets Comparing Structure Evidence with Combined Evidence

#	Structure Evidence	Combined Evidence	Interestingness	Related Pages
1	(0.5,0.5)	(0.092,0.092)	0.577	/Research/,/Research/arpa/
2	(0.5,0.5)	(0.083,0.083)	0.589	/Research/,/Research/cais/
3	(0.5, 0.5)	(0.196,0.196)	0.430	/Research/airvl/loon.html, /Research/airvl/minirob.html
4	(0.5,0.5)	(0.096,0.096)	0.572	/contact-info.html, /systems-staff/contact-info.hml
5	(0.5,0.5)	(0.179,0.179)	0.453	/help/,/help/configure/
6	(0.5,0.5)	(0.146,0.146)	0.500	/help/,/help/security/
7	(0.5,0.5)	(0.128,0.128)	0.523	/help/,/help/setup/cs-setup.html
8	(0.5,0.5)	(0.139,0.139)	0.510	/help/,/help/software/
9	(0.5,0.5)	(0.190,0.190)	0.439	/help/,/help/support.html
10	(0.5,0.5)	(0.185,0.185)	0.445	/help/,/help/web/
11	(0.5,0.5)	(0.179,0.179)	0.453	/newsletter/, /newsletter/kumar.html
12	(0.5,0.5)	(0.128,0.128)	0.526	/newsletter/, /newsletter/relations.html
13	(0.5,0.5)	(0.141,0.141)	0.508	/newsletter/, /newsletter/robfac.html

thresholds. Notice that at an interestingness threshold of 0.1, only about 300 of the 1345 discovered rules are declared to be interesting. This indicates that the methods for calculating evidence pairs for usage and structure evidence result in similar values for most of the itemsets.

Two lists of potentially interesting rules were identified by comparing the structure evidence with the combined evidence, and then comparing the usage evidence with the combined evidence. The lists of potentially interesting frequent itemsets are shown in Tables 7 and 8. Table 7 basically contains pages that are used together less than would be expected from the structure of the site (using an interestingness threshold value of 0.4). The first two rules are of note because /Research/arpa/ and /Research/cais/ aren't actually HTML pages, but are only directories, which might explain why they are not accessed as often as expected. The results presented in this table are consistent with the theoretical arguments presented in the previous section. Table 8 contains rules with pages that aren't directly connected by links but have relatively high support (with the threshold set at 0.5). Despite not having directly connected hyperlinks, these pages are somewhat related by their common URL structure. This is an artifact of the choice of a binary cfactor for computing the structural evidence pair. A cfactor that assigns non-zero values for pages that are close to each other but not directly connected would most likely filter out many of the rules listed in Table 8. Nevertheless, both tables verify the ability of WebSIFT's information filter to identify rules with conflicting evidence in accordance with the support logic framework.

Table 8. Interesting Frequent Itemsets Comparing Usage Evidence with Combined Evidence

#	Usage Evidence	Combined Evidence	Interestingness	Related Pages
1	(0.409,0.409)	(0,0)	0.579	/Research/airvl/people.html, /Research/airvl/postdoc.html
2	(0.5,0.5)	(0,0)	0.707	/Research/arpa/, /Research/neural/
3	(0.391,0.391)	(0,0)	0.553	/employment/fac-positions/soft-sys.html, /employment/msse/
4	(0.370,0.370)	(0,0)	0.524	/employment/msse/, /employment/temporary/
5	(0.643,0.643)	(0,0)	0.909	/employment/other/naog.html, /employment/other/ncs.html
6	(0.393,0.393)	(0,0)	0.556	/help/configure/, /help/offsite/cs-offsite.html
7	(0.435,0.435)	(0,0)	0.615	/icra99/TechnProgram.htm, /icra99/advanceprogram.htm
8	(0.391,0.391)	(0,0)	0.553	/icra99/best.html, /icra99/bestk.html
9	(0.474,0.474)	(0,0)	0.670	/labs/1-214.html, /labs/downtime/
10	(0.474,0.474)	(0,0)	0.670	/labs/1-260.html, /labs/2-216.html
11	(0.5,0.5)	(0,0)	0.707	/labs/1-260.html, /labs/downtime/
12	(0.409,0.409)	(0,0)	0.579	/labs/2-216.html, /labs/downtime/
13	(0.556,0.556)	(0,0)	0.786	/labs/CCIE/cost.html, /labs/CCIE/description.html
14	(0.707,0.707)	(0,0)	1.000	/reg-info/ss1-99.html, /reg-info/ss2-99.html
15	(0.583,0.583)	(0,0)	0.825	/sem-coll/cray.html, /sem-coll/seminar/seminar.html

7 Conclusions

Using the support logic model, this paper has developed a general framework for determining the interestingness of mined knowledge. The framework leverages the power of a robust logic system based on fuzzy logic and evidential reasoning, that is capable of reasoning about evidence from multiple sources about a given belief. Both reinforcing and conflicting pieces of evidence can be handled. In addition, automated methods for generating evidence in support of beliefs have been defined and tested for the Web Usage Mining domain.

Future work will include the incorporation of content data and development of information filter algorithms for use with sequential patterns and clusters of pages and users. In addition, tests will be run with various degrees of ignorance built into the calculated evidence pairs.

References

1. Funnel web professional. http://www.activeconcepts.com.
2. Hit list commerce. http://www.marketwave.com.
3. Webtrends log analyzer. http://www.webtrends.com.
4. World wide web committee web usage characterization activity. http://www.w3.org/WCA.
5. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th VLDB Conference*, pages 487–499, Santiago, Chile, 1994.
6. J. F. Baldwin. Evidential support logic programming. *Fuzzy Sets and Systems*, 24(1):1–26, 1987.
7. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD International Conference on Management of Data*, 1997.
8. Alex Buchner and Maurice D Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *SIGMOD Record*, 27(4):54–61, 1998.
9. M.S. Chen, J.S. Park, and P.S. Yu. Data mining for path traversal patterns in a web environment. In *16th International Conference on Distributed Computing Systems*, pages 385–392, 1996.
10. Robert Cooley, Bamshad Mobasher, and Jaideep 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, 1997. IEEE.
11. Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1), 1999.
12. Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Discovery of interesting usage patterns from web data. Technical Report TR 99-022, University of Minnesota, 1999.
13. T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *The 15th International Conference on Artificial Intelligence*, Nagoya, Japan, 1997.
14. Bing Liu, Wynne Hsu, and Shu Chen. Using general impressions to analyze discovered classification rules. In *Third International Conference on Knowledge Discovery and Data Mining*, 1997.
15. 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.
16. Olfa Nasraoui, Raghu Krishnapuram, and Anupam Joshi. Mining web access logs using a fuzzy relational clustering algorithm based on a robust estimator. In *Eighth International World Wide Web Conference*, Toronto, Canada, 1999.
17. 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*, Santa Clara, CA, 1997.
18. Balaji Padmanabhan and Alexander Tuzhilin. A belief-driven method for discovering unexpected patterns. In *Fourth International Conference on Knowledge Discovery and Data Mining*, pages 94–100, New York, New York, 1998.
19. Mike Perkowitz and Oren Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *Fifteenth National Conference on Artificial Intelligence*, Madison, WI, 1998.

20. G. Piatetsky-Shapiro and C. J. Matheus. The interestingness of deviations. In *AAAI-94 Workshop on Knowledge Discovery in Databases*, pages 25–36, 1994.
21. Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: Extracting usable structures from the web. In *CHI-96*, Vancouver, 1996.
22. James E Pitkow. Summary of www characterizations. In *Seventh International World Wide Web Conference*, 1998.
23. A. L. Ralescu and J. F. Baldwin. Concept learning from examples and counter examples. *International Journal of Man-Machine Studies*, 30(3):329–354, 1989.
24. G. Schafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
25. Cyrus Shahabi, Amir M Zarkesh, Jafar Adibi, and Vishal Shah. Knowledge discovery from users web-page navigation. In *Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997.
26. A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Eng.*, 8(6):970–974, 1996.
27. Myra Spiliopoulou and Lukas C Faulstich. Wum: A web utilization miner. In *EDBT Workshop WebDB98*, Valencia, Spain, 1998. Springer Verlag.
28. Shivakumar Vaithaynathan. Data mining on the internet - a kdd-98 exhibit presentation. http://www.epsilon.com/kddcup98/mining/, 1998.
29. L. A. Zadeh. A theory of approximate reasoning. *Machine Intelligence*, 9:149–194, 1979.
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. In *Advances in Digital Libraries*, pages 19–29, Santa Barbara, CA, 1998.