WebAdaptor: Designing Adaptive Web Sites Using Data Mining Techniques

Howard J. Hamilton, Xuewei Wang, and Y.Y. Yao
Department of Computer Science
University of Regina, Regina, SK S4S 0A2, Canada
{hamilton,xwang,yyao}@cs.uregina.ca

Abstract

Creating an adaptive web site, which is a web site that automatically changes its contents and organization according to usage, is a challenge for web site designers. We introduce a novel algorithm for creating an adaptive web site based on combining web usage mining and adaptive web site generation. We outline the design of a system called WebAdaptor that implements our ideas. We elaborate on the usage mining module, adaptation module, and page generation module in the WebAdaptor system.

Keywords: WebAdaptor, Web Usage Mining, Adaptive Web Site, KDD, Data Mining, Web Mining, Log files, Association rules, Clustering, World Wide Web

1. Introduction

Automatically changing a web site to adapt to the typical behavior of users is a challenge for web site designers. An ***adaptive web site*** is a web site that automatically changes it contents and organization according to usage [9]. An adaptive web site provide novel useful features that aid visitors to thc site. For example, an adaptive web site may group relevant web pages based on recent access patterns. Because the user can find information more quickly and effectively, adaptation can improve overall performance of a web site.

The most accessible record of web site usage exists in the web server's log file. Analyzing a log file can answer questions such as: who has visited the site? what did they do? where did they come from? what pages are likely to be visited together?

We describe the WebAdaptor system for creating adaptive web sites with two main features: fresh pages and a site usage report. ***Fresh pages*** are new web pages automatically created by WebAdaptor, based on the results of web usage mining in combination with the existing structure and content of the web page. Fresh pages include topic group pages, access paths, and frequently accessed web pages. A ***site usage report*** is a series of suggestions and information for web site designers about the users' access patterns and behavior.

The rest of this paper is organized as follows. Section 2 introduces related research. Section 3 gives an overview of WebAdaptor. Sections 4 through 6 describe the design of usage module, the adaptation module and the page generation module, respectively. Section 7 gives conclusions.

2. Related Research

Web mining is the application of data mining techniques to large web data repositories [2]. ***Web content mining*** is the application of data mining techniques to unstructured data residing in web documents [15]. ***Web structure mining*** aims to generate structural summaries about web sites and web pages [15]. ***Web usage mining*** is the application of data mining techniques to discover usage patterns from web data [5].

Commercial software packages for web log analysis, such as Analog [1], WUSAGE [18], and Count Your Blessings [3] have been applied to many web servers. Common reports are a list of the most requested URLs, a summary report, and a list of the browsers used. Currently, these packages provide limited mechanisms for reporting user activity. They usually cannot provide adequate analysis of data relationships among log files. Another limitation of these packages is their slow speed.

Research in web usage mining has focussed on discovering access patterns from log files. Data mining techniques are appropriate because many web sites contain hundreds or thousands of pages and are accessed by millions of user sessions. A ***web access pattern*** is a recurring sequential pattern among the entries in a web log. For example, if various users repeatedly access the same series of pages, a corresponding series of log entries will appear in the web log file, and this series can be considered a web access pattern. Sequential pattern mining and clustering have been applied to discover web access patterns from log files [16]. Since the problem of finding sites visited together is similar to finding associations among itemsets in transaction databases, many web usage mining techniques search for association rules [2].

Current web usage mining research can be classified into personalization, system improvement, site modification, business intelligence, and usage characterization [5]. Making a dynamic recommendation to a web user, based on her/his profile in addition to usage behavior, is called ***personalization***. WebWatcher [17], SiteHelper [4], and analog [16] provide personalization for web site users.

Site modification is the automatic modification of a web site's contents and organization based on learning from web usage mining [5]. Previous research on adaptive web sites applied the SCML algorithm to adapt a web page to usage [10]. The SCML algorithm combines a Statistical Clustering algorithm with a Machine Learning algorithm. The clustering algorithm generates clusters of related pages, and the concept learning algorithm describes these clusters using expressions in the given concept description language.

3. The WebAdaptor System

In this section, the WebAdaptor system for web usage mining and adaptive web site generation is described.

3.1. Design Goals and Achievements

The main goal of WebAdaptor is to generate a suitable and flexible intelligent agent to help a user navigating a web site. By using several data mining techniques in the usage mining module, WebAdaptor provides a set of high quality recommendations for a web site. With the help of WebAdaptor, a user can find useful information easily. This greatly improves the web site's performance. In addition, a web site designer may use the information and suggestions given by WebAdaptor to redesign a web site to improve accessibility and performance.

By using the Apriori algorithms [17], leader clustering algorithm [10], and C4.5 [11], the WebAdaptor can discovery association rules and clustering to extract relationships from large web log files. For example, WebAdaptor can find the following relationship by using association rules:

> 38% of people who accessed the web page
> http://www.cs.uregina.ca/~xwang/study.htm,
> also accessed
> http://www.cs.uregina.ca/~xwang/photo.htm.

Since WebAdaptor is accessible via any web browser, visitors to a web site can use it at any time. In addition, by using flexible graphical interface, it provides a friendly user interface. By using CGI scripts, the WebAdaptor provides user control over data mining process and allows users to extract only relevant and useful rules.

3.2. Architecture of WebAdaptor

To improve the design of web sites, analyze system performance, understand user reaction, and build adaptive web site, we analyze the web log files by finding association rules, sequential patterns, and clusters.

As shown in Figure 3.1, WebAdaptor has three main modules. The ***usage mining module*** performs data cleaning and session identification. It uses the Apriori algorithm to generate the large itemsets and association rules. The ***adaptation module*** uses the Leader clustering algorithm and the C4.5 machine learning algorithm to generate adaptive web pages. The ***page generation module*** manages and queries the database, which contains all data in the system. This architecture is described in detail in the next three sections.

In the current prototype implementation, WebAdaptor consists of a collection of data mining programs, web pages, CGI scripts, C language functions, Perl programs, JAVA applications and JAVA Applets, together with a central database.

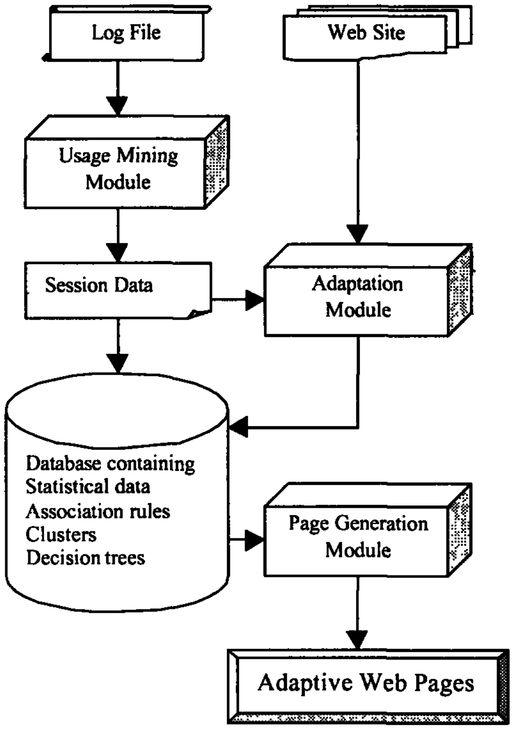


Figure 3.1. Architecture of WebAdaptor

4. Usage Mining Module

The two main tasks in the usage module are data preparation and usage mining.

4.1. Data Preparation

Before the data mining algorithms can be used, the log file data is prepared by cleaning it and identifying sessions.

Web servers commonly record an entry in a web log file for every access; most accessible information for web site usage exists in this log file. The relevant information for web usage is stored in files that can be dissected in a variety of ways and can be used for detailed analysis.

With long lines wrapped, a typical log file looks like:

```
net-ppp65.cc.uregina.ca - - [10/Feb/2000:11:19:56 -0600] "GET
/~xwang/gif/but/b4a.gif HTTP/1.0" 304 -
net-ppp65.cc.uregina.ca - - [10/Feb/2000:11:19:56 -0600] "GET
/~xwang/gif/but/b4a.gif HTTP/1.0" 304 -
```

"http://www.cs.uregina.ca/~xwang/" "Mozilla/4.04 [en] (Win95; I)"

Common components of a log file format include: Internet Protocol (IP) address or Domain Name for the user, host name, user authentication, date/time, request or command, Universal Resource Locator (URL) path for the item, Hyper Text Transfer Protocol (HTTP) method, completion code, and number of bytes transferred.

The server log files contain many entries that are irrelevant or redundant for the data mining tasks. These entries are removed by cleaning the data. For example, all entries relating to image files and map files are irrelevant.

After cleaning the data, we identify the sessions. To use association rules, the Leader clustering algorithm, and C4.5 machine learning to discovery patterns and relationship from log files, we must group the individual page accesses into meaningful sessions. The WebAdaptor uses the IP address, time, web page, and agent to classify several entries into a single session. First, use IP address to identify unique users. Any access from different IP addresses is identified as a different session. Secondly, because different users may use the same IP address, WebAdaptor uses the browser software and operating system to further classify the accesses. A differ browser or operating system is taken to indicate a different session. Finally, because the same user may visit the web site at different times, we use a time period of 6 hours to further divide the entries into individual sessions.

4.2. Data Mining

Once the user sessions have been identified, we search for association rules [13] to find relationships among these data. The Apriori algorithm can find ***frequent itemsets***, which are groups of items occurring frequently together in many sessions. With the Apriori algorithm, the problem of mining association rules is decomposed into two parts: find all combinations of items that have session support above a threshold level minimum support (frequent itemsets), and generate the ***association rules*** from these frequent itemsets. Table 4.1 shows the support and confidence values for some association rules in one example.

X ⇒ Y	Support(X ∪ Y)	Support(X)	Confidence
{/~xwang} ⇒ {/~xwang/photo.htm}	50%	100%	50%
{/~xwang} ⇒ {/~xwang/link.htm}	25%	100%	25%
...	...	...	...
{/~xwang, /~xwang/study.htm} ⇒ {/~xwang/R1.htm}	50%	50%	100%
...	...	...	...
{/~xwang, /~xwang/study.htm, /~xwang/R1.htm} ⇒ {/~xwang/link.htm}	25%	50%	50%
...	...	...	...
...	.	...	...
...	...	...	...

Table 4.1. Confidence Values for Selected Association Rules

The input is a set U of n unique URLs appearing in the log file:

$$U = \{url1, url2, \ldots, urln\}$$

and a set S of m user sessions:

$$S = \{s1, s2, \ldots, sm\}$$

The support of a set of URLs $u \subseteq U$ is defined as:

$$\text{Support} = \frac{|\{s \in S: u \subseteq s\}|}{|S|}$$

By analyzing, WebAdaptor can also generate statistics analogous to those from web site traffic analysis tools. They include:

- Summary information, such as period of time, number of accesses, number of hits, number of visitors, and number of hosts.
- Access information for each page.
- Visitor IP addresses.
- Information about visitors' browser and operating system platforms.

5. Adaptation Module

5.1. LCSA Algorithm

In the adaptation module, we propose the ***LCSA*** (Leader, C4.5, and web Structure for creating Adaptive web sites) algorithm to generate adaptive web pages. The LCSA algorithm is shown in Figure 5.1. We used 5 conditional attributes in the C4.5 training instances (i.e., k was set to 5 in step 4B).

```
Input: logfile L, web structure tree T, with a
          description in D for each node of T, and
          k, the number of conditional attributes.
1 Run  cleaning program on L and generate
          cleaned data CD.
2 Identify sessions in CD to produce the set of sessions S
3 Run Leader algorithm on S to generate a set C of
          clusters.
4 for each cluster c in C
     for each page P in c
        A) Derive the complete set P' of prefixes from
           page P's pathname.
        B) Use k shortest prefixes in P' as conditional
           attributes for a C4.5 training instance.
        C) Use name of c as decision attribute for training
           the C4.5 training instance
6 Run C4.5 algorithm to generate decision tree DT.
7 Combine DT and D to form adaptive pages.
```

Figure 5.1. LCSA Algorithm Schema

In LCSA algorithm, the Leader algorithm is used to generate page clusters and C4.5 is used to generate rules. ***Clustering*** seeks to identify a finite set of categories or clusters to describe the data. It divides a data set so that records with similar content are in the same group, and groups are as different as possible from each other. We choose to use clustering based on user navigation patterns, whereby site users with similar browsing patterns are grouped in the same cluster.

Although we can use the clusters to generate adaptive web pages, preliminary experiments showed that the quality of the resulting pages was poor because the cluster of pages had little in common. To give the users high quality suggestions, the clusters of pages should be related not only by access patterns but also by content or location in the web site's structure. WebAdaptor combines the clustering of pages with information about the contents and the web site structure to generate the adaptive web site.

5.2. Clustering

The Leader algorithm [6] for clustering is used because the number of entries in a web log file is large and efficiency is essential. We adapted the Analog software package [16], which uses the Leader algorithm, to create our clustering module. The Leader algorithm is given in Figure 5.2. Beginning with no clusters, the input vectors are examined one by one. Each vector is added to the closest cluster, if the distance is less than MaxDistance. If no such cluster exists, the vector forms a new cluster.

```
Input: a set of vectors V.
Output: a set of cluster C
  set C to empty
  for each v ∈ V
       if the cardinality of v is greater than
              MinNumPages then
          find cluster c in C such that the distance
          between the median of c and v is the minimum
          (set d to this minimum) among all clusters in C
     if the distance d is less than MaxDistance then
          add v to c
     else add {v} to C
  for each c in C
     if the size of c is less than MinClusterSize then
          remove c from C
  return C
```

Figure 5.2. The Leader Algorithm

The output of this algorithm is a set of page clusters that indicate the web pages frequently visited together by users. The next step in WebAdaptor is to check whether the contents of these pages are related based on their pathnames in the web site.

5.3. C4.5

The C4.5 software package implements the C4.5 concept learning algorithm for finding decision trees or decision rules from attribute-value data [7]. We use the prefixes of a page's path name as conditional attributes, and the cluster id as the decision attribute. For example, for the URL ~brown/courses/170/cs170.html, each of ~brown, ~brown/courses, and ~brown/courses/cs170 are treated as values of conditional attributes.

The resulting decision tree describes the access patterns with respect to structure information. We combine it with web site content to produce a set of adaptive web pages, which we call ***topic group pages***. Then we put those topic group pages in our database.

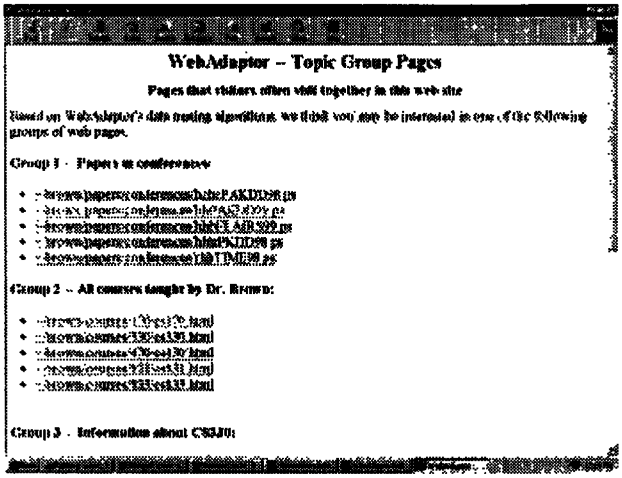
6. Page Generation Module

As data are generated, they are stored in a relational database. The page generation module manages and queries this database. In addition, the page generation module generates adaptive web pages.

WebAdaptor creates an adaptive web site by generating fresh pages and a site usage report. The ***site usage report*** provide web designers with a general description of visitors, association rules, clusters, decision trees, and some simple statistical information.

A ***fresh page*** is a web page that is automatically created either for each visitor or periodically, such as once a day, based on web usage mining. The fresh pages form the main interface for the adaptive web site. A fresh page lists one of three types of URLs: topic groups, access paths, or frequently accessed web pages. A fresh page may become the favorite starting point of a site visitor.

A topic group page lists the URLs of other pages, grouped under textual headings. As shown in Figure 6.2, each group contains web pages on a common subject that have been frequently visited together by users recently. The topic group pages are formed by the Leader algorithm, C4.5 concept learning algorithm, web site content, and web site structure. The topic group pages are the most common of the adaptive web pages.



WebAdaptor -- Topic Group Pages

Pages that visitors often visit together in this web site

Based on WebAdaptor's data mining algorithms, we think you may be interested in one of the following groups of web pages.

Group 1 - Papers at conferences

- ~brown/papers/conferences/[illegible]KDD98.ps
- [illegible]
- ~brown/papers/conferences/[illegible]LAIR99.ps
- ~brown/papers/conferences/[illegible]KDD98.ps
- ~brown/papers/conferences/[illegible]TIME98.ps

Group 2 -- All courses taught by Dr. Brown:

- [illegible]
- ~brown/courses/330/cs330.html
- ~brown/courses/430/cs430.html
- [illegible]
- ~brown/courses/335/cs335.html

Group 3 - Information about CS330:

Figure 6.2. Topic group Page

An ***access path*** is a sequence of page references (URLs) such that the probability that a visitor will visit one page given that he has visited another page is great than a threshold. Access paths reveal relationships between the web pages that users access. Given the association rules in Table 4.1, the access paths are as follows:

Path1: www.cs.uregina.ca/~xwang
⇒www.cs.uregina.ca/~xwang/photo.html
Path2: www.cs.uregina.ca/~xwang
⇒www.cs.uregina.ca/~xwang/linc.html
Path3: www.cs.uregina.ca/~xwang
⇒www.cs.uregina.ca/~xwang/study.html
⇒www.cs.uregina.ca/~xwang/R1.html
Path4: www.cs.uregina.ca/~xwang
⇒www.cs.uregina.ca/~xwang/study.html
⇒www.cs.uregina.ca/~xwang/R1.html
⇒www.cs.uregina.ca/~xwang/linc.html

The ***frequently accessed web pages*** are web pages that are most frequently visited by users, as measured by simple statistical measures.

WebAdaptor extracts data from a database to dynamically generate the adaptive web pages. The fresh pages are given to web site visitors when they visit this web site. The user can decide whether to use these suggestions or not. The site usage report is intended primarily for the web designers for use when redesigning the web site.

7. Conclusion

In this paper, we presented a framework for combining web usage mining and adaptive web site generation. We introduced the LCSA algorithm to combine the Leader clustering algorithm, the C4.5 concept learning algorithm, web site content, and web site structure. The implementation shows that the results are not only in keeping with the actual access patterns, but also consistent with the web pages' content.

WebAdaptor providers information to the web site design on how to improve the design of a web site, analyze performance, understand user behavior, and generate an adaptive web site without changing the original web site.

References

[1] Analog http://www.statslab.cam.ac.uk/~sret1/analog/.

[2] B. Mobasher, N. Jain, J. Han, J. Srivastava. "Web Mining: Pattern Discovery From World Wide Web Transaction." In *International Conference on Tools with Artificial Intelligence*, pp. 558-567, Newport Beach, 1997.

[3] Count Your Blessings. http://www.internetworld.com/print/montly/1997/06/iwlabs.html.

[4] D. S. W. Ngu, and X. Wu. "SiteHelper: A Localized Agent that Helps Incremental Exploration of the World Wide Web." In *Proceedings of 6th International World Wide Web Conference*, Santa Clara, CA, 1997.

[5] J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data." *SIGKDD Explorations*, Vol. 1, Issue 2, 2000.

[6] J. Hartigan. *Clustering Algorithms*. John Wiley. 1975.

[7] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann. San Mateo, CA, 1993.

[8] L. Catledge and J. Pitkow. "Characterizing Browsing Behaviors on the World Wide Web." *Computer Networks and ISDN Systems*, 27(6), 1995.

[9] M. Perkowitz, and O. Etzioni. "Adaptive Web Sites: Automatically Synthesizing Web Pages." In *Proceedings of Fifteenth National Conference on Artificial Intelligence (AAAI'98)*. Madison, WI, 1998.

[10] M. Perkowitz, and O. Etzioni. "Adaptive Web Sites: Conceptual Cluster Mining." In *Proceedings of Sixteenth International Conference on Artificial Intelligence (IJCAI'99)*, Stockholm, Sweden, 1999.

[11] O. Zaiane, M. Xin, and J. Han. "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs." In *Proc. Advances in Digital Libraries Conf. (ADL'98)*, Melbourne, Australia, pp. 144-158, April 1998.

[12] R. Agrawal and R. Srikant. "Mining Sequential Pattern." In *Proc. 1995 International Conference Data Engineering*, Taipei, Taiwan, pp. 3-14, March 1995.

[13] R. Agrawal, and R. Srikant. "Fast Algorithms for Mining Association Rules." In *Proceedings of the 20th VLDB Conference*, Santiago, Chile, pp. 487-499, 1994.

[14] S. L. Manley. *An Analysis of Issues facing World Wide Web Servers*. Undergraduate thesis, Harvard, 1997.

[15] S. K. Madria, S. S. Bhowmick, W. K. Ng, E. Lim: "Research Issues in Web Data Mining." In *First International Conference on Data Warehousing and Knowledge Discovery*, Florence, Italy, 1999: 303-312

[16] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. "From User Access Patterns to Dynamic Hypertext Linking." In *Proceedings of the 5th International World Wide Web Conference*, Paris, France, 1996.

[17] 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.

[18] WUSAGE http://www.boutell.com/wusage/.