PagePrompter: An Intelligent Agent for Web Navigation Created Using Data Mining Techniques

Y.Y. Yao, H.J. Hamilton, and Xuewei Wang

Abstract: Creating an intelligent agent for web navigation, which is an agent that dynamically gives recommendations to a web site's users by learning from web usage mining and users' behavior, is a challenge for web site designers. In this paper, we introduce a novel algorithm for creating an intelligent agent for navigating a web site based on combining web usage mining and machine learning. We describe the overall design of a system called PagePrompter that implements our idea. We elaborate on the usage mining module, recommendation module, and adaptive pages modules of the PagePrompter system.

Keywords: PagePrompter, Web Usage Mining, KDD, Data Mining, Web Mining, Log files, Association rules, Clustering, World Wide Web

1. Introduction

With the fast growth of information on the World Wide Web, finding and retrieving useful information becomes a very important issue. Web search engines offer a popular solution to this problem. Typically, a search engine returns a list of web pages according to their matches to the query. Little information is provided about the structure and access frequency of particular web site containing the web page. A web user may use the ranked web page list for navigating the web and finding relevant pages. In this paper, we propose another solution to this problem based on an intelligent agent. Instead of providing a list of web pages, an agent assists the user in navigating a particular web site while searching for useful information. The recommendations of the agent are based on results of mining web log data and observing user behavior.

Conceptually, the entire web may be interpreted as a graph, in which each web page is a node of the graph and each link is an edge of the graph connecting two web pages. The graph representation provides a good tool for describing relationships of web pages. It is a physical view of the web. It is not necessarily a good description of the semantic relationships among web pages. For effective and efficient retrieval, different logical views of web may be created by web users and web site designers. A web user may create a logical view of the web based on the his/her information needs. For example, a user may create bookmark files and personalized link pages, which reflect the user's personal interests. Alternatively, a web search engine may store user profiles representing the user's logical view of the web and search the web accordingly. A web site designer can also create different logical views of a web site for individual users or distinct groups of users. Web log data and the access patterns of a site, as well as user behavior, suggest information useful for this task. Data mining and machine learning techniques may be used to find such information. In this paper, we demonstrate that intelligent agent techniques can be combined with data mining and machine learning techniques to support web users and web site designers in creating logical views. For the user, an agent can be built to surf the web and construct logical views that are of interest. For the designer of a web site, an agent can be built to create various logic views of the site to assist users visiting it. The latter type of agent is the focus of this paper.

We have implemented an agent, called PagePrompter, which gives the recommendations to a web site's users. The agent acts likes a tour guide by assisting a user in navigating the web site. It can help the visitors find information quickly and efficiently by offering different logical views of the web site and providing additional information not available on the web pages. Such an agent may improve the performance of a web site and has great potential in E-commence. The knowledge of the PagePrompter is obtained from the web site designer, user behavior analysis and web mining. The main contribution of this paper is the novel combination of research ideas concerning intelligent agents and data mining. It demonstrates that data mining and machine learning can be used as knowledge acquisition methods for building intelligent agents.

The rest of this paper is organized as follows: Section 2 introduces some related research. We give the framework of PagePrompter in Section 3. Section 4 describes the usage module in PagePrompter. Section 5 describes the recommendation module in PagePrompter. In Section 6, we present the design of the adaptive pages module in PagePrompter. Section 7 gives conclusions.

2. Related research

Data mining is a step in the Knowledge Discovery in Databases (KDD) process consisting of applying data analysis and discovery algorithms that, within acceptable computational efficiency constraints, produce a particular enumeration of patterns over the data [22]. Data mining has been successfully applied in science, health, marketing, and finance. ***Web mining*** is the application of data mining techniques to large web data repositories [4]. Three major web mining methods are web content mining, web structure mining and web usage mining. ***Web content mining*** is the application of data mining techniques to unstructured data residing in web documents [19]. ***Web structure mining*** aims to generate structural summaries about web sites and web pages [19]. ***Web usage mining*** is the application of data mining techniques to discover usage patterns from web data [11].

Commercial software packages for web log analysis, such as Analog [3], WUSAGE [24], and Count Your Blessings [5] 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.

Research in web usage mining has focussed on discovering access patterns from log files. 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 [8, 20]. The problem of finding sites visited together is similar to finding associations among itemsets in transaction databases [17]. Therefore, many web usage mining techniques search for association rules [4].

Current web usage mining research can be classified into personalization, system improvement, site modification, business intelligence, and usage characterization [9]. Making a dynamic recommendation to a web user, based on the user profile in addition to usage behavior, is called ***personalization***. WebWatcher [21], SiteHelper [7], and analog [20] provide personalization for web site users. Web usage data can be combined with marketing data to give information about how visitors use a web site for E-commerce [1]. For usage characterization, some researchers focused on Xmosaic [12] and self-configuring benchmarks [18]. ***Site modification*** is the automatic modification of a web site's contents and organization based on learning from web usage mining.

3. The PagePrompter System

3.1. Design Goal of PagePrompter

The main goal of PagePrompter 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, PagePrompter provides a set of high quality recommendations for a web site. With the help of the PagePrompter, 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 PagePrompter to redesign a web site to improve accessibility and performance.

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

- 50% 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.
- 20% of people who accessed web page http://www.cs.uregina.ca/~xwang/study.htm, also searched the web site by accessing http://www.cs.uregina.ca/~xwang/link.htm,

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

3.2. Architecture of PagePrompter

As shown in Figure 3.1, PagePrompter has three main modules: the usage mining module, the recommendation module, and the adaptive pages module. The ***usage mining module*** performs data cleaning and transaction identification. It uses the Apriori algorithm to generate the frequent itemsets and association rules. It also uses the leader clustering algorithm and the C4.5 machine learning algorithm to generate page clusters. The ***recommendation module*** captures the user's action and connects to the database to obtain suggestions. The ***adaptive pages module*** generate adaptive pages. It also manages and queries the database, which contains all data in the system. PagePrompter consists of a collection of data mining algorithms, web pages, CGI scripts, C language functions, Perl programs, JAVA applications and JAVA Applets as well as a central database.

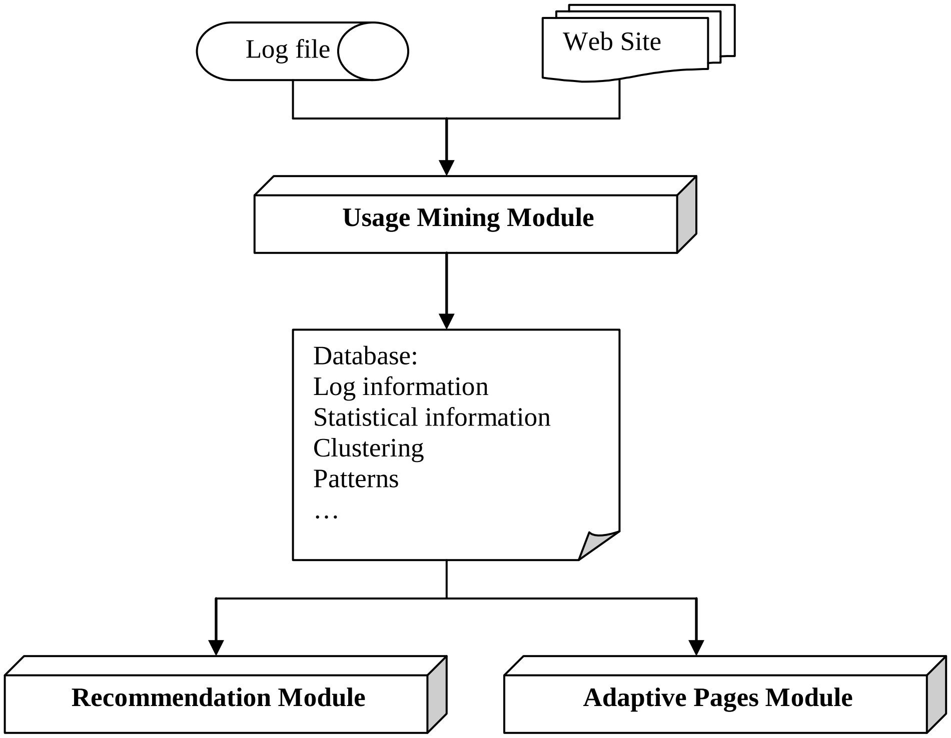


Figure 3.1. Architecture of PagePrompter

4. Usage Module

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

4.1. Data Preparation

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. Common components of a log file 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. 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)"
```

The server log files contain many entries that are irrelevant or redundant for the data mining tasks. For example, all entries relating to image files and map files are irrelevant to identifying user behavior. Therefore, PagePrompter cleans the raw data to remove unneeded data.

After cleaning the data, PagePrompter identifies transactions. For usage mining, individual entries for page accesses are grouped into meaningful transactions. The PagePrompter uses the IP address, time, web page, browser software, and operating system to group entries. First, PagePrompter uses the IP address to identify unique users. Any access from different IP addresses is identified as a different transaction. Secondly, as different users may use the same IP address, PagePrompter uses the browser software and operating system to further classify the accesses. A different browser or operating system is taken to indicate a different transaction. Finally, because the same user may visit the web site at different times. PagePrompter uses a time period of 6 hours to further divide the information into individual transactions.

4.2. Data Mining

During data mining, PagePrompter finds association rules, page clusters, and standard statistics.

4.2.1. Association Rules

Once the user transactions have been identified, we search for association rules [17] to

```
procedure AprioriAlg()
begin
    $L_1$ := {Frequent 1-itemsets};
    for ( k := 2; $L_{K-1} \neq 0$; k++ ) {
        $C_k$= Apriori-Gen($L_k$-1) ; // new candidates
        For all transactions t in the dataset {
            $C_t$ = subset($C_k$, t);
            for all candidates $c \in C_k$ contained in t do
                    c:count++
        }
        $L_k = \{ c \in C_k$ | c:count >= min-support}
    }
    Answer := $\bigcup_k L_K$
end

function Apriori-Gen( )
insert into $C_k$
select p.$item_1$, p.$item_2$, …p.$item_{k-1}$, q.$item_{k-1}$
from $L_{k-1}$ p, $L_{k-1}$ Q
where p.$item_1$ = q.$item_1$,… p.$item_{k-2}$ = q.$item_{k-2}$, p.$item_{k-1}$ < q.$item_{k-1}$
```

Figure 4.1. The Apriori Algorithm

find relationships among these data. The Apriori algorithm can find ***frequent itemsets***, which are groups of items occurring frequently together in many transactions.

The Apriori algorithm is given in Figure 4.1. With the Apriori algorithm, the problem of mining association rules is decomposed into two parts: finding all frequent itemsets, i.e., all combinations of items that have transaction support above a support threshold, and generating the ***association rules*** from these frequent itemsets. Table 4.1 shows the confidences of some association rules in one example.

For PagePrompter, the items are URLs, and the frequent itemsets are combinations of pages that are often accessed together. The set U of n unique URLs appearing in the log files:

$$U = \{ url_1, url_2, \ldots, url_n \},$$

and the set of m user transactions is

$$T = \{t_1, t_2, \ldots, t_m\}.$$

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

$$\text{Support}(url_i) = \frac{|\{t \in T: u \subseteq t\}|}{|\,T\,|}$$

X ==> Y	Support (X U Y)	Support (X)	Confidence
{/~xwang} ==> {/~xwang/photo.htm}	50%	100%	50%
{/~xwang} ==> {/~xwang/linc.htm}	25%	100%	25%
…	…	…	…
{/~xwang, /~xwang/study.htm} ==> {/~xwang/R1.htm}	50%	50%	100%
…	…	…	…
{/~xwang, /~xwang/study.htm, /~xwang/R1.htm} ==> {/~xwang/linkc.htm}	25%	50%	50%
…	…	…	…
…	…	…	…
…	…	…	…

Table 4.1. Confidence Values for Selected Association Rules

4.2.2. Finding Page Clusters

- **LCSA Algorithm Schema**

To find suitable clusters, we propose the ***LCSA*** (Leader, C4.5, and web Structure for Adaptive web site) algorithm to generate adaptive web pages. The LCSA algorithm is shown in Figure 5.1.

In the 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. A typical cluster of pages often had little in common. To give the users high quality suggestions, the clusters of pages should be related by content or location in the web site's structure as well. PagePrompter combines the clustering of pages with information about the contents and the web site structure to generate the adaptive web site.

```
 Input: log file L, web structure tree T, with description D for each node of T.
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
      (1) for each page P in c
          A) Derive the complete set P' of prefixes from the pathname of  page P.
          B) Use k shortest prefixes in P' as condition attributes for a C4.5
               training instance.
         C) Use the name of c as decision attribute for 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

- **Clustering**

The leader algorithm [10] 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 [20], which uses the leader algorithm, to create our clustering module. The Leader algorithm is given in Figure5.2. Beginning with no clusters, the input vectors are examined one by

```
 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
         remove c from C
return C
```

Figure 5.2. The Leader Algorithm

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. 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 PagePrompter is to find if the contents of these pages are related using C4.5.

- **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 [11]. The C4.5 algorithm generates a classification-decision tree for the given data-set by recursively partitioning the data into smaller subsets, based on the value of an attribute.

The task of PagePrompter is to generate a set of recommendations that should suit both the access patterns and web page content. We already have the clusters of access patterns. Therefore, we use the cluster id as the distance attribute for C4.5. The condition attributes are described from the web site's structure, i.e., its arrangement in a series of directories/folders of files. We create one training instance for C4.5 and generate a decision tree.

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

4.2.3. Statistical methods

By analyzing the log entries, PagePrompter also provides statistic analysis analogous to that provided by web site traffic analysis tools. The information generated 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. Recommendation Modules

The recommendation module is the main part of PagePrompter. By using the data from usage module, this module can give the user best assistant for web navigation. Figure 5.1 shows the structure of the recommendation module.

When a user enter the web site, his action is captured by PagePrompter and sent back to the PagePrompter server. The server connects to a database and queries the database for any relevant recommendations for the current page. If the PagePrompter find any relevant information, it pops up a small window, which contains the recommendations.

As shown in Figure 5.2, the recommendation window include several parts: next choice, group of related pages, hottest pages, recently created or modified pages, user feedback, and some information about visitors.

6. Adaptive pages module

As data are generated, we store them in a relational database. The adaptive pages module manages and queries this database and generates the adaptive web pages.

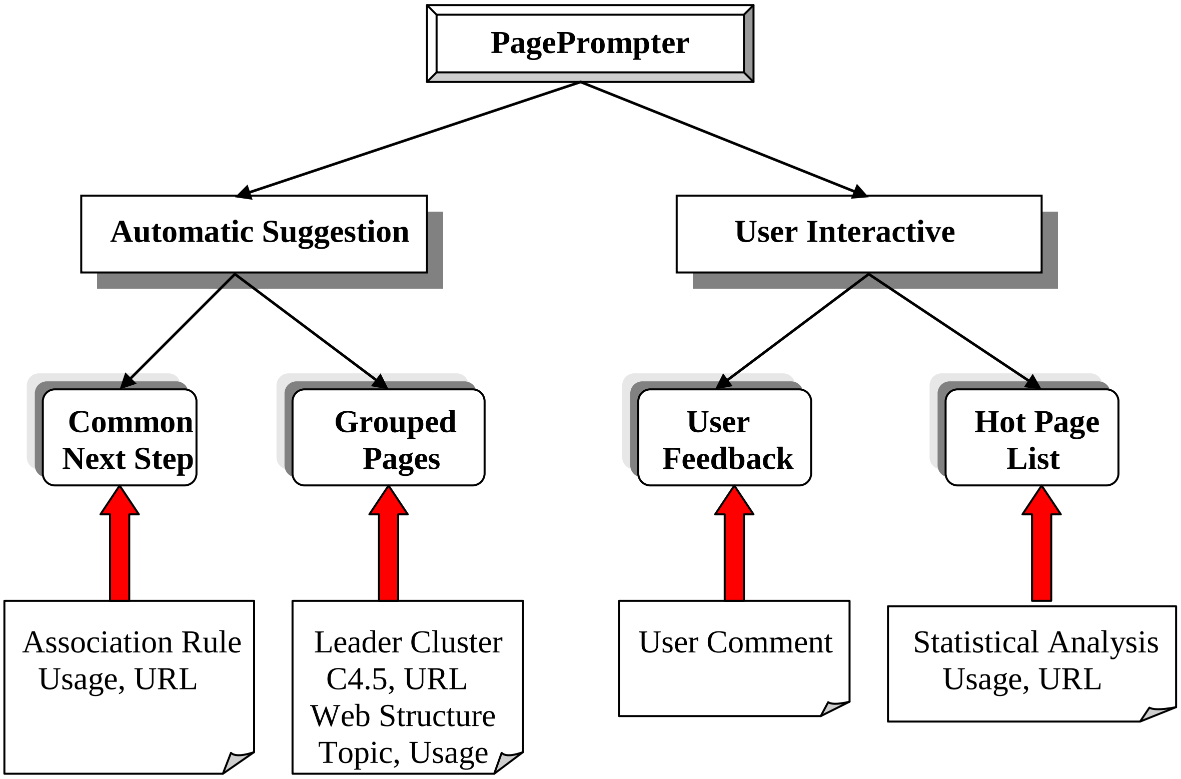
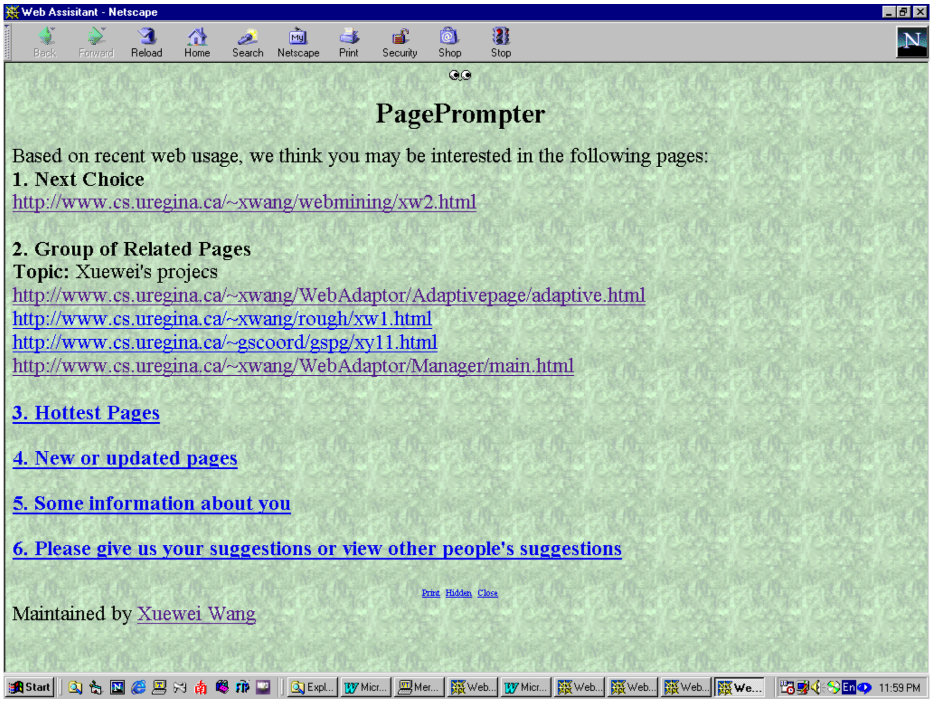


Figure 5.1: Structure of Recommendation Module



PagePrompter

Based on recent web usage, we think you may be interested in the following pages:

1. Next Choice

http://www.cs.uregina.ca/~xwang/webmining/xw2.html

2. Group of Related Pages

Topic: Xuewei's projecs

http://www.cs.uregina.ca/~xwang/WebAdaptor/Adaptivepage/adaptive.html
http://www.cs.uregina.ca/~xwang/rough/xw1.html
http://www.cs.uregina.ca/~gscoord/gspg/xy11.html
http://www.cs.uregina.ca/~xwang/WebAdaptor/Manager/main.html

3. Hottest Pages

4. New or updated pages

5. Some information about you

6. Please give us your suggestions or view other people's suggestions

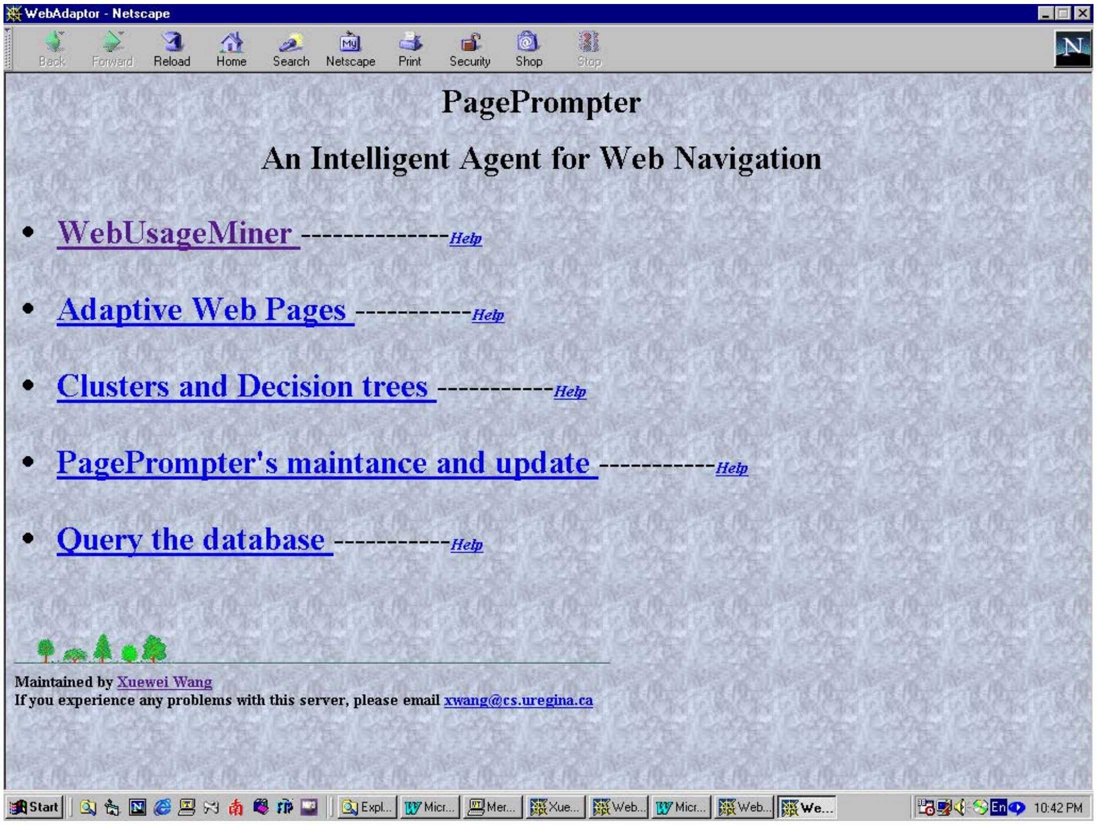
Print Hidden Close

Maintained by Xuewei Wang

Figure 5.2: Interface of PagePrompter

6.1. Database Querying and Management

Using a web browser, an end user can query and manage PagePrompter through database querying and management part. We use a flexible graphical interface to provide a friendly user interface. By using CGI scripts, the PagePrompter provides user control over data mining process and allows users to extract only relevant and useful rules. The main interface of PagePrompter is shown in Figure 6.1. As well, the adaptive pages module also gives the user control over the data. The data is updated regularly.



PagePrompter

An Intelligent Agent for Web Navigation

- WebUsageMiner -------------Help
- Adaptive Web Pages -----------Help
- Clusters and Decision trees -----------Help
- PagePrompter's maintance and update -----------Help
- Query the database -----------Help

Maintained by Xuewei Wang
If you experience any problems with this server, please email xwang@cs.uregina.ca

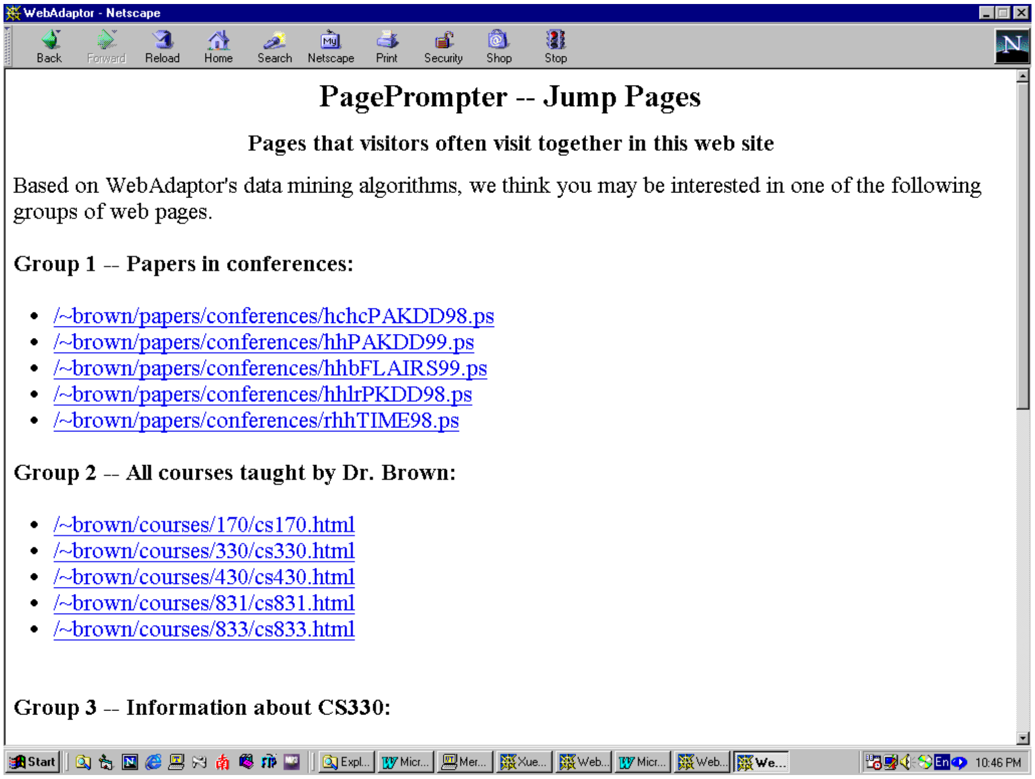
Figure 6.1. Main interface of PagePrompter

6.2. Adaptive Web Page – Fresh Pages

PagePrompter 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. They include three types: jump pages, access paths, and frequently accessed web pages. A fresh page may become the favorite starting point of site visitors.

A jump page lists the URLs of other pages, grouped under textual headings. Each group contains web pages are on a common subject that have been frequently visited together by users recently. The jump pages are formed by the leader algorithm, C4.5 concept learning algorithm, web site content, and web site structure. The jump pages are the main parts of adaptive web pages. Figure 6.2 shows a part of jump pages.



PagePrompter -- Jump 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 in conferences:

- /~brown/papers/conferences/hchcPAKDD98.ps
- /~brown/papers/conferences/hhPAKDD99.ps
- /~brown/papers/conferences/hhbFLAIRS99.ps
- /~brown/papers/conferences/hhlrPKDD98.ps
- /~brown/papers/conferences/rhhTIME98.ps

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

- /~brown/courses/170/cs170.html
- /~brown/courses/330/cs330.html
- /~brown/courses/430/cs430.html
- /~brown/courses/831/cs831.html
- /~brown/courses/833/cs833.html

Group 3 -- Information about CS330:

Figure 6.2. Jump pages

An ***access path*** is an association rule describing the probability that a visitor will visit one page given that he has visited another page. Access paths reveal relationships between the web pages that users access. Given the association rules in Table 4.1, the access paths based on those association rules 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.htm

Path3: www.cs.uregina.ca/~xwang ° www.cs.uregina.ca/~xwang/study.htm
° www.cs.uregina.ca/~xwang/R1.html

Path4: www.cs.uregina.ca/~xwang ° www.cs.uregina.ca/~xwang/study.htm
° www.cs.uregina.ca/~xwang/R1.html ° www.cs.uregina.ca/~xwang/linc.htm

Frequently accessed web pages are web pages that are most frequently visited be users. They will indicate what are the hottest pages in this web site. Frequently web pages are created based on the usage statistics.

PagePrompter extracts data from database to dynamically generate the adaptive web pages. The fresh pages are given to a web site visitor when he/she visits this web site. The user can decide whether to use these suggestions or not. The site usage report is provided to web designers to allow redesign of the web site.

7. Conclusion

In this paper, we presented a practical framework for designing an intelligent agent that dynamically gives the recommendations to the web site's users by learning from web usage data and users' behavior. Like a tour guide, the agent assists a user in navigating the web site. Knowledge obtained by the agent may also be used to improve the design of web sites.

A novel algorithm is introduced for creating an intelligent agent for navigating web We introduce the LCSA algorithm to combine the leader cluster algorithm, C4.5 concept learning algorithm, web site content, and web site structure.

PagePrompter can also be used as a tool by a web site designer for improving the design of web sites, analyzing system performance, understanding user behavior, and generating an adaptive web site without changing the original web site.

References

[1] A. Buchner and M. D. Mulvenna. "Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining." *SIGMOD Record*, 27(4): 54-61, 1998.

[2] Analysis of Data Mining Algorithms
http://www.gl.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm.

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

[4] 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.

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

[6] C.T. Tu. "A Clustering Algorithm Based on User Queries." *Journal of the American Society for the Information Science*. July-August 1974.

[7] 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.

[8] D. Florescu, A.Y. Levy, and A.O. Mendelzon. "Database Techniques for the World-Wide Web: A Survey." *SIGMOD Record*, 27(3): 59-74, 1998.

[9] 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.

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

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

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

[13] 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.

[14] 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.

[15] 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, pages 144-158, April 1998.

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

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

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

[19] 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

[20] 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.

[21] 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.

[22] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[23] V. Almeida, A. Bestavros, M. Crovella, and A. D. Oliveira. *Characterizing Reference Locality in the WWW*. Technical Report TR-96-11, Boston University, 1996.

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