Using Data Mining Techniques on Web Access Logs to Dynamically Improve Hypertext Structure

F. Masseglia[1,2] – P. Poncelet[2]– M. Teisseire[2]

[1] Laboratoire PRiSM
Université de Versailles
45 Avenue des Etats–Unis
78035 Versailles Cedex, France

[2] LIRMM UMR CNRS 5506
161, Rue Ada
34392 Montpellier Cedex 5, France

E–mail: {massegli,poncelet,teisseire}@lirmm.fr

Abstract

With the growing popularity of the World Wide Web (Web), large volumes of data such as user address or URL requested are gathered automatically by Web servers and collected in access log files. Discovering relationships and global patterns that exist in such files can provide significant and useful information for performance enhancement, restructuring a Web site for increased effectiveness, and customer targeting in electronic commerce. In this paper, we propose an integrated system (WebTool) for applying data mining techniques such as association rules or sequential patterns on access log files. Once interesting patterns are discovered, we illustrate how they can be used to customize the server hypertext organization dynamically.

1 Introduction

Applying data mining techniques to the Web is called Web mining and can be broken in two main categories: Web content mining and Web usage mining [CoMo97]. The former deals with discovering and organizing Web–based information whereas Web usage mining addresses the problem of exhibiting behavioural patterns from one or more Web servers collecting data about their users. Even if Web analysis tools offer various facilities (e.g. list of top requested URLs, address of users, etc) [Hype98], relationships among accessed resources or user accesses are not provided by such tools which are still limited in their performance [ZaXi98]. The considered relationships are merely observations of individual behaviours of visitors. If they are frequent, i.e. sufficiently repeated by individuals to be relevant, it could be an interesting knowledge for decision makers who attempt to draw general lessons from particular cases. For instance, relationships and global patterns embedded in servers can provide significant and useful information for restructuring a Web site for increased effectiveness. Furthermore it is now possible to automatically discover user profile in order to anticipate the needs of the user and provide the right information. Existing Web analysis tools try to provide user profiles but they have the following drawbacks: (i) the input is often a subjective description of the users by the users themselves and (ii) the profile is static [MoCo99].

We show in this paper that data mining techniques such as association rules or sequential patterns are very efficient for automatically discovering user profiles. For concreteness, consider a very simple association rule *"50 % of visitors who accessed URLs* `/info-f.html` *and* `labo/infos.html` *also visited* `situation.html`*"*. The discovered relationship can be used to improve the efficiency of information retrieval by prefetching documents for the users. In our case, the Web server may react to a customer request for `labo/infos.html` by prefetching the `situation.html` page. Mining sequential patterns with time constraints are very useful to capture typical behaviours over time, i.e. behaviours sufficiently repeated by individuals to be relevant for decision maker. Sequential patterns could provide temporal relationships such as: *"60 % of clients who visited* `/jdk1.1.6/docs/Package-java.io.html` *and*

/jdk1.1.6/docs/ja-va.io.Buffered-Writer.html in the same transaction, also accessed /jdk1.1.6/docs/relnotes/deprecatedlist.html during the following month" or "*34 % of clients visited /relnotes/deprecatedlist.html between September the 20th and October the 30th and /jdk1.1.6/docs in november*". Such temporal relationships could be very useful to improve the hypertext organization. We can use the information "a visitor accessed two or three pages in less than 5 seconds" to group together these pages since it appears that there is no useful information for visitors. Furthermore, in order to provide an efficient navigation to the visitor, the organization of the server can be customized and navigational links can be dynamically added.

The groundwork of the approach presented in this paper is Web usage mining. We propose an integrated system, called WebTool, for mining either association rules or sequential patterns. Our proposal pays particular attention to time constraint handling. Once interesting patterns are discovered, we illustrate how they can be used to dynamically customize the hypertext organization. More precisely, the user current behaviour can be compared to one or more sequential patterns and navigational hints can be added to the pages proved to be relevant for this category of users.

The rest of the paper is organized as follows. In Section 2 we present an overview of the WebTool System. Section 3 addresses the problem of using discovered patterns in order to update the hypertext organization. Related work is presented in Section 4. Finally Section 5 concludes with experiments and future directions.

2 Overview of the WebTool System

For presenting our approach, we adopt the chronological viewpoint of data processing: from collected raw data to exhibited knowledge. Like in [CoMo97], we consider that the mechanism for discovering relationships and global patterns in Web servers is a 2–phase process. From data automatically gathered by Web servers and collected in access log files, the preprocessing phase removes irrelevant data and performs a clustering of entries driven by time considerations. It yields in a populated database containing the meaningful remaining data. In the second phase, data mining techniques are applied in order to extract useful patterns or relationships and a visual query langage is provided in order to improve the mining process. Our approach is supported by an integrated system enforcing the described capabilities. Its functional architecture, close to that of the WebMiner [CoMo97], is depicted in figure 1.

Figure 1: An overview of the WebTool process

Data Preprocessing

An entry in the log file generally respects the Common Log Format specified by the CERN and the NCSA [W3C98]. During the data processing, three types of manipulations are carried out on the entries of the server log. First of all, a data filtering step is performed in order to filter out irrelevant requests (for instance, this step can filter out requests encompassing graphics or sound). Then the remaining access log file is sorted by address and time. Finally, we propose like in [MoJa96] to cluster together entries sufficiently close over time. Indeed, an entry in the access log file could be seen as a single transaction. In order to better understand temporal relationships embedded in the access files, transactions are built up by grouping, for each visitor, URLs accessed in the same window size according to a maximum time gap. For example, let us consider a time gap value of 3 seconds and the following visitor navigation from an access log file:

```
192.70.76.73––[22/Nov/1998:11:06:11 +0200] "GET /info/program.html HTTP/1.0"
192.70.76.73––[22/Nov/1998:11:06:12 +0200] "GET /info/matiere.html HTTP/1.0"
```

The two entries can be grouped together since the delay between the two transactions is less than 3 seconds. The preprocessing phase results in a new database containing coded transactions. Each transaction, provided with a relative data, concerns a visitor, and groups together URLs visited during a common time range.

Knowledge Discovery

From the transformed data yielded by the preprocessing stage, two techniques of knowledge discovery can be applied for fully meeting the analyst needs: association rules and sequential patterns. Association rules mining provides the end user with correlations among references to various pages and sequential patterns can be used to determine temporal relationships among pages. Furthermore, mining sequences with time constraints allows a more flexible handling of the visitor transactions, insofar the end user is provided with the following advantages:

1. To gather page accesses when their dates are rather close. For example it does not matter if pages in a sequential pattern were present in two different transactions, as long as the transaction–times of those transactions are within some small time window. In other words, by introducing a time window we relax the original transaction cutting and could consider that all URLs accessed during a small range (few seconds) are grouped together. With a small size of time window, we may consider that these URLs are mainly concerned by navigation.
2. To regard sets of pages as too close or distant to appear in the same frequent sequence. For example, the end user probably does not care if a visitor accessed `/java-tutorial/ui/anuimLoop.html`, followed by `/relnotes/deprecatedlist.html` three months later.

Visualization Tool

The analyst is provided with a query language for expressing his preferences. In this language, regular expressions over URLs are used to select antecedent or consequent of the rule. Furthermore, in order to discard irrelevant inputs for a particular analysis, restriction parameters refering dates or IP address domains are also provided.

We implemented the WebTool system using GNU C++ with STL and the user interface module is implemented using Java which gives several benefits both in terms of added functionality and in terms of easy implementation. This module also concerns the preprocessing phase, i.e. the mapping from an access log file to a database of data–sequences according to the user defined time window, as well as the visualization tool. Interested reader may refer to [MaPo99] where a complete description of the WebTool system is proposed.

3 Updating the Hypertext Organization Dynamically

Jointly with the WebTool system, we developed a generator of dynamic links in Web pages using the rules generated from sequential patterns or association rules. The generator is intended for recognizing a visitor according to his navigation through the pages of a server. When the navigation matches a rule, the hypertext organization is dynamically modified.

Since we are only interested in navigation through pages, we assume, in the following, that the hypertext document is defined as a directed weakly connected labeled multigraph. Insofar a navigation through the hypertext corresponds to a sequence of stops in several pages. In other words, a navigation in the hypertext document is an alternated sequence of nodes and edges $n^{t0}e^{t0}n^{t1}e^{t1}...n^{ti-1}e^{ti-1}n^{ti}$, beginning and ending with nodes, in which each edge is incident with the two nodes immediately preceding and following it.
In the following we no longer consider edges between pages, and an hypertext navigation for a visitor C is a tuple $E_C = <id_c, \{n^{t1}, n^{t2}, ... n^{tm} \}>$ where, for $1 \leq k \leq m$, n^{tk} is the kth node accessed by the visitor (URL of the reached page) with its associated time stamp.

Let us assume user defined parameters standing for time constraints. A rule R generated from sequential patterns or association rules, is a tuple R = <<a_1 a_2 ... a_i>, <c_1 c_2 .... c_j>> where, for $1 \leq k \leq i$, a_k stands for a set of URLs in the antecedent part and, for $1 \leq k \leq j$, c_k stands for a set of URLs in the consequent part. Furthermore the antecedent as well as the consequent part respect time constraints. As illustration, considering the second example of the Section 1, the rule R is the following R = <<(`/jdk1.1.6/docs/Package-java.io.html/jdk1.1.6/docs/java.io.BufferedWriter.html`)>, <(`/jdk1.1.6/docs/rel-notes/deprecated-list.html`)>>

For performing the insertion of a dynamic link from the antecedent part of a rule, let us introduce the interesting subset notion: considering a user defined parameter *minPages*, standing for the minimal number of pages from which a link can be added. The *interesting subset* of R, noted Is_R, is defined as follows: for all a_k in {a_1 a_2 ... a_i}, $a_k \in Is_R$ if and only if $k \leq$ minPages.

In order to illustrate when an hypertext satisfies a rule, let us consider a visitor navigation <X^{t0} A^{t1} Y^{t2} B^{t3} Z^{t4} C^{t5}> where X, A, Y, B, Z, C are URLs accessed by the visitor. By introducing time constraints, one of the visitor path could be: p = <(X^{t0}) (A^{t1} Y^{t2} B^{t3}) (Z^{t4} C^{t5})>. Now, let us consider a rule R where the set of URLs of the antecedent part is: a= <(A B) (C) (D E)>. Let us assume that minPages=3, thus to be considered as interesting three pages must be accessed by the same visitor. The *interesting subset*, Is_R, is the following <(A B) (C)>. The visitor satisfies the rule since (A B) $\subseteq$ (A^{t1} Y^{t2} B^{t3}) and (C) $\subseteq$ (Z^{t4} C^{t5}).

Implementation Issues

The technique presented so far was implemented using the functional architecture depicted in figure 2.

Figure 2: General architecture

The Web server (*http* daemon) reacts to a customer request returning an applet encharged of the connection to the *visitor manager module* in order to transmit visitor IP address, required URL and a cookie encompassing the visitor navigation. The visitor manager module is a Java application running on the Web server site and using a client/server mechanism. When receiving IP address and required URL, the *visitor manager* examines the customer behaviour by using the *rule base* through the *correspondence module*.
The latter checks if the customer behaviour, i.e. the client navigation, satisfies a rule previously extracted by the data mining process. When an input satisfies a rule in the *correspondence module*, the required page is modified by the *page manager* which dynamically adds links towards the consequent of the recognized rule. The applet then recovers the URL and displays page on the navigator. If no rule corresponds to the current behaviour of the customer, the URL towards the required page is turned over to the applet which can display it.

Figure 3: Part of the hypertext organization and dynamically inserted link

In order to illustrate how a dynamic link is inserted, let us consider the following example. In the different rules obtained from the IUT access log file, we have noticed that 85% of visitors who visited "`/index.html`" and "`/info/index.html`" pages in the same transaction, followed by "`/info/program.html`" within 2 days, request the server on the "`/info/opportunity.html`" URL after an additional visit to the "`/info/index.html`" (C.f. Figure 3).

Let us consider a client accessing the pages <(`index.html info/genera.html`) (`info/program.html`)> during his navigation. Let us consider that the navigation satisfies the previous rule. A link corresponding to each consequent of this rule is added to the page. In our case, a link to the page "opportunities" (`/info/opportunity.html`) is dynamically inserted in the URL concerning the Program (C.f. Figure 3).

4 Related Work

Analyzing user access log for exhibiting useful access patterns has been studied in some interesting approaches. Among them, we quote the approach presented in [CoMo97,MoJa96]. A flexible architecture for Web mining, called WEBMINER, and several data mining functions (clustering, association, etc) are proposed. For instance, even if time constraints are not handled in the system, an approach for mining sequential patterns is addressed. Furthermore various constraints can be specified using a SQL–like language with regular expression in order to provide much more control all along the discovery process. In [MoCo99], the authors address the problem of automatic personalization, i.e. taking into account the user's taste to provide automatically the right information.

The WUM system proposed in [SpLu98] is based on an "aggregated materialized view of the Web log". Such a view contains aggregated data on sequences of pages requested by visitor. A query processor is incorporated to the miner in order to identify navigation patterns satisfying properties (existence of cycles, repeated access, etc) specified by the user.

On–line analytical processing (OLAP) and multi–dimensional Web log data cube are proposed by [ZaXi98]. In the WebLogMiner project, the data is split up into the following phases. In the first phase, the data is filtered to remove irrelevant information and it is transformed into a relational database in order to facilitate the following operation. A multi–dimensional array structure, called a data cube is built, each dimension representing a field with all possible values described by attributes. OLAP technology is used in the third phase. Finally data mining techniques can be used on the Web log data cube and Web log database.

The use of access patterns for automatically classifying users on a Web site is discussed in [YaJo96]. In this work, the authors identify clusters of users that access similar pages using user access log entry. This lead to an improved organization of the hypertext documents. In this case, the organization can be customized on the fly and links can be dynamically suggested.

5 Conclusion

In this paper, we presented an architectural framework for Web usage mining. We showed that association rules and sequential patterns extracted from Web server access logs allow to predict user visit patterns as well as a dynamic hypertext organization.

To validate our system, we carried out a number of experiments. First, the log was taken from the « IUT d'Aix en Provence » web site. The site hosts a variety of information including for instance the home pages of 10 departments, course information or job opportunities. During experiments, the access log file covered a period of six months and there were 10, 384 requests in total. Experiments on this log were mainly concerned by association rules, i.e. We would like to know more about accessed pages but we didn't care about temporal aspects. We thus conduct experiments using different minimum support values. Empirical evaluations have shown that the end user was provided with rules in less than 40 seconds with a support value of 20 %. In other words, assuming that the preprocessing phase is done, we obtain all the navigation paths followed by at least 20 % of visitors in less than 40 seconds. The two following rules are examples of discovered rules:

```
(/iut/imgs/veille3.jpg)→(/iut/pages/sommaire.html) (/iut/pages/format.html)
support 0.5
(/iut/pages/prog.html) → (/iut/pages/info.html) (/iut/mq/pages/biblio.html)
support 0.582
```

Second, we were interesting in experiments handling time constraints with the access log file obtained from the Lirmm Home Page. The log contains about 150K entries corresponding to the requests made during March of 1998 and its size is about 85M Bytes (before pre–processing). There are 1500 distinct URLs referenced in the transactions and 2000 visitors. Let us consider the following rule extracted by the mining process on the log considering time constraints: <(/jdk1.1.6 /jdk1.1.6/docs/index.html) (/jdk1.1.6/docs/api/packages.html) <(/jdk1.1.6/docs/relnotes/deprecatedlist.html)> indicating that 67% of visitors who accessed the URLs (/jdk1.1.6 /jdk1.1.6/docs/index.html) the same day, the URL (/jdk1.1.6/docs/api/packages.html) more than 2 days after, also visited the URL (/jdk1.1.6/docs/relnotes/deprecatedlist.html) within the 5 days and 2,3% of visitors have accessed the four URLs.

Third, in order to assess the relative performance of the system when updating the hypertext dynamically, we have generated synthetic data simulating access log files. Experiments were performed on a mirror site of the IUT d'Aix en Provence. We observed that cookies and client/server mechanisms do not slow down page access (reaching a page through an applet requires less than one second). Nevertheless, we are currently investigating how to efficiently insert several links into a page. Inserting a dynamic link is, for the moment, achieved using a script writen in Perl and we consider that all pages are provided with a set of comment tags which are removed when the page is updated.

Future work on Web usage mining will be done in various directions. First, we aim to improve the analysis of visitor behaviour encompassing frequent back–tracks. Even if applet–based, site topology [Pitk97] or "client–site" log files [ZaXi98] solutions exist, they are not easy to handle and need an access to the visitor site. Second, we are currently studying how to improve the process extraction using an incremental mining. This problem is very important in the Web mining context since the log files (access log, error log, etc) are always growing. We think that an incremental approach focusing on relationships previously extracted could be very efficient.

References

[ChKa97] D.W. Cheung, B. Kao, and J. Lee. Discovering User Access Patterns on the World–Wide Web. In Proceedings of the 1st Pacific–Asia Conference on Knowledge Discovery and Data Mining (PAKDD'97), February 1997.

[W3C98] World Wide Web Consortium. httpd–log files. In http://lists.w3.org/Archives, 1998.

[CoMo97] R. Cooley, B. Mobasher, and J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.

[Hype98] HyperNews. Httpd log analyzers.
In http://www.hypernews.org/HyperNews/get–www/loganalyzers.html, 1998

[MaPo99] F. Masseglia, P. Poncelet, and R. Cicchetti. WebTool: An Integrated Framework for Data Mining. In Proceedings of the 10th International Conference on Database and Expert Systems Applications

(DEXA'99), Florence, Italy, August 1999.

[MoCo99] B. Mobasher, R. Cooley, and J. Srivastava. Automatic Personalization Based on Web Usage Mining. Technical report, Depaul University, 1999.

[MoJa96] B. Mobasher, N. Jain, E. Han, and J. Srivastava. Web Mining: Pattern Discovery from World Wide Web Transactions. Technical Report TR–96–050, Department of Computer Science, University of Minnesota, 1996.

[Pitk97] J. Pitkow. In Search of Reliable Usage Data on the WWW. In Proceedings of the 6th International World Wide Web Conference, pages 451–463, Santa Clara, CA, 1997.

[SpLu98] M. Spiliopoulou and L.C. Faulstich. WUM: A Tool for Web Utilization Analysis. In Proceedings of EDBT Workshop WebDB'98, Valencia, Spain, March 1998.

[YaJo96] 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, May 1996.

[ZaXi98] O. Zaïane, M. Xin, and J. Han. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In Proceedings on Advances in Digital Libraries Conference (ADL'98), Santa Barbara, CA, April 1998.