ADAPTIVE WEB SITES BY WEB USAGE MINING

Yongjian Fu   Mario Creado   Ming-Yi Shih
Department of Computer Science
University of Missouri-Rolla
Rolla, MO 65409-0350
{yongjian, mcreado, mingyi}@umr.edu

Abstract *An approach for reorganizing a Web site based on user access patterns is proposed. The Web server's log files and the Web pages on the site are first preprocessed to obtain the access statistics of the Web pages and the links among the pages. Using the access information, the Web pages are classified into two major categories: index pages and content pages. The Web site is evaluated and a reorganization of the Web site is presented based on the access information. Our experiments on a large real data set show that the approach is efficient and practical for adaptive Web sites.*

Keywords: Web usage mining, adaptive Web site, Web page classification, Web mining.

1 Introduction

With the increasing popularity of the World Wide Web, building and maintaining a quality Web site has become a serious concern for many organizations. A static Web site that does not change over time will not achieve its goal to serve the needs of its users, because of the dynamic nature of the Web and the Web users. A Web site needs to adapt to the changing needs of its users. Hopefully, the adaption can be done automatically or semi-automatically. An adaptive Web site is defined as a Web site that semi-automatically improves its organization and presentation by learning from visitor access patterns[9].

To make a Web site adaptive, Web mining, in particular, Web usage mining, has been introduced to learn patterns in Web usage. Web usage mining is the process to identify interesting patterns from Web server logs. It has been proven to be a powerful tool for adaptive Web sites, Web personalization, user clustering, user profiling, and network optimization.

We aim to build an adaptive Web site that will reorganize its pages so that its users can find the information they want with minimum effort. When a Web site is designed initially, its intended usage at that time is reflected in its structure, i.e., how the pages are organized and linked. Over the time, however, its usage may change and the original organization is no longer appropriate. Basically, we want the Web site to evolve so that its users can find their current targets with minimum number of clicks. In the meantime, we want to preserve the original structure if possible since it may contain some semantics and a radical change of organization can confuse users. For examples, a book at an on-line bookstore may become popular and we can make it easier to access by having a link from home page to it, instead of hiding it at several levels deep from home page.

In this paper, an approach for the reorganization of a Web site based on user access patterns is proposed. First, access information of the Web users is extracted from the Web server's log files. Second, the Web pages on the Web server are classified into index pages and content pages, based on the characteristics and access statistics of the pages. Finally, the whole Web site is analyzed and a reorganization of the Web site is presented based on access information and page classification. The approach has been implemented and tested on a real Web site. Our experiments show that the approach is efficient and practical for adaptive Web sites.

2 Background

Most commonly used Web servers maintain a server log of page requests in the form of Common Log Format. The Common Log Format specifies that a record in a log file contains the IP address of the user, the date and time of the request, the URL of the page, the protocol, the return code of the server, and the size of the page if the request is successful.

In Web usage mining, the server logs are preprocessed to group requests from the same user into server sessions. A server session is defined as the pages requested from a single visit of a user to the Web site. During the preprocessing, irrelevant information for Web usage mining such as background images and unsuccessful requests is ignored. The users are identified by the IP addresses in the log and all requests from the same IP address are put into a session. Different heuristics have been developed to deal with the inaccuracy due to caching, IP sharing or blocking, and network congestion.

A lot of studies have been conducted in Web usage mining. Some focus on the mining of association rules and navigation patterns in the user access paths [1, 3, 14]. A session is viewed as a transaction in association rule mining and algorithms for association rule mining are employed to find frequent paths that are followed by many users. Others build data cubes from Web server logs for OLAP and data mining [2, 15]. The statistics along pages, IP domains, geographical location of users, and access time are calculated from sessions. Some others cluster users based on their access patterns [5, 8, 12]. There is also research on data preparation [4] and query language [13] for Web usage mining.

Recently, research into adaptive Web site has been proposed by some researchers. An initial definition of the problem was presented in [9]. Clustering of pages based on access patterns has been studied in [10]. Web pages that are not directly linked but are frequently accessed together are clustered and an index page can be synthesized to link these pages together. In [7], pages are clustered based on their occurrences in frequent paths that are found through association rule mining.

In this research, we attempt to use the results from Web usage mining to reorganize the Web site. Page access information of users is coupled with the knowledge of how the Web site is organized and is expected to function, for the purpose of discovering and recommending suitable changes to the site organization. The main difference between our approach and those in [7, 10] is that we do not create clusters of pages, rather we let the Web site evolve as the usage evolves.

Our approach is comprised of three major parts: preprocessing, page classification, and site reorganization. In preprocessing, the Web site is analyzed and an internal representation is built. The server log files should also be transformed into sessions and the access statistics of the pages are computed. In page classification, the pages are classified into two categories: index pages and content pages. In site reorganization, the Web site is examined to find suitable changes of the Web site based on page categories and access statistics. These changes are selected such that they will reduce the number of clicks a user needs in order to find the information the user wants. These reorganizations may be recommended to the Webmaster for possible actions.

3 Adaptive Web Sites By Web Usage Mining

3.1 Preprocessing

The preprocessing takes a server log file(s) and the Web site URL as its input. It will result in an internal representation of the Web site with access statistics attached. Specifically, there are three sub-tasks in preprocessing as follows.

1. Construction of an internal representation of the Web site.

   From the URL of the Web site, the pages are explored. The links in each HTML page (identified by the tag <A HREF>) are followed. The result is an internal data structure that represents the whole Web site. All references to other Web sites are ignored. We use a directed graph as our internal data structure in which a page is a node and a link is an arc.

2. Transformation of logs into sessions.

   The server log file is first scanned to filter out background images and unsuccessful requests. Requests from the same IP address are grouped into a session. A timeout of 30 minutes is used to decide the end of a session, i.e., if the same IP address does not occur within a time range of 30 minutes, the current session is closed. The server logs are transformed into a set of sessions. Each session contains a session ID and a set of (page-ID, time) pairs, where time is the time the user spent on the page. It is determined by the difference between two consecutive requests. For the last page, the time is unknown.

3. Computation of page access statistics.

   The sessions are read to gather statistics about page accesses. The following statistics, among others, are computed for each page, and stored in the internal date structure.

   - Number of times the page is accessed.
   - Total amount of time that has been spent on the page.
   - Number of sessions in which the page is accessed.

3.2 Page Classification

Web pages can be classified into index pages and content pages [6, 11]. An index page is a page that is used mostly for navigation of the Web site. It normally contains little information with the exception of links. A content page is a page that holds information which users will be interested in. The Web pages are classified in to one of the categories using the following heuristics.

- Reference length.

  The reference length of a page is the average amount of time a user spent on the page. It is expected that the reference length of an index page is typically small while the reference length of a content page will be larger. Based on this assumption, the reference length of a page can hint whether the page should be categorized as an index or content page.

  A cut-off value is computed by a function which takes two input parameters, the average reference length of all pages and an estimation of the overall percentage of pages that are index pages. If a page's reference length is less than the cut-off value, it is more likely an index page, otherwise, it is more likely a content page.

- Number of links in a page.

  Generally, an index page has more links than a content page. A threshold is set such that the number of links in a page is compared with the threshold. A page with more links than the threshold is probably an index page. Otherwise, it is probably a content page.

- File type.

  The file type of a page also implies its category. If the page is a non-HTML file, it is marked as a content page, otherwise its category is marked as UNKNOWN and further processing must be done to determine its category.

The algorithm for page classification is given below.

Algorithm 3.1 Input: (1) a Web page p with its mean reference length, and (2) threshold t_l for number of links.

Output: Category of p.

Method: A combination of the heuristics mentioned above.

```
if type of $p$ is not HTML
  mark $p$ as a content page
else
  $\lambda \leftarrow$ mean reference length of all pages
  $t \leftarrow -\ln(1-\gamma) \times \lambda$ // cut-off point
  // $\gamma$ is the estimated percentage
  // of index pages
  if ($p$'s mean reference length $< t$) or
    (number of links in $p > t_l$)
    mark $p$ as an index page
  else
    mark $p$ as a content page
```

3.3 Reorganization of Web Site

The goal of this phase is to reorganize the Web site such that users will be able to access the information they desire with fewer clicks. A Web site provides a better service to users by cutting down on their navigation time, thus making the Web site more user friendly. Therefore, the general idea of reorganization is to cut down on the number of intermediate index pages a user has to go through.

To achieve the goal, we need to place the frequently accessed pages higher up in the Web site structure, i.e., closer to the home page, while pages that are accessed infrequently should be placed lower in the structure. However, we want to preserve the original site structure whenever possible, since it may bear business or organizational logics. Besides, dramatic changes of the site structure may confuse users. As a compromise between these two conflicting requirements, we introduce an evolutionary approach to Web site reorganization. The basic idea is to locally adjust the site when a frequently accessed page should be promoted.

Three threshold parameters are introduced in our approach. The minimum frequency threshold, F, is used to determine if a page is frequently accessed. If the number of sessions that contains a page is greater than F, the page is frequently accessed. Two other thresholds are, maximum number of links in an index page (I) and maximum number of links in a content page(C).

During the reorganization, pages are processed from top down starting at the home page. For each page, we consider its children, where a child is any page that the current page has a link to. Depending upon its category and the number of children it has, different actions may be taken.

For each page, we consider three cases depending upon the number of children it has. In case it doesn't have any children, the processing of that particular page is terminated. If a page has multiple parents, where a parent is a page that has a link to the current page, we consider one parent at a time. We denote the parent page as p and the current page as n.

- n has one child c.

  If n is an index page, obviously the page is redundant since it only serves as a link from p to page c. The most appropriate action will be to delete n and create a direct link from p to page c.

  If n is a content page and c is more frequent than n, c should be promoted by adding a direct link from p to c. However, in case p has used its links to full capacity, it will be necessary to demote n to be a child of c. The maximum number of links in p is determined by its category and the thresholds I and C.

- n has two children, c_1 and c_2.

  For simplicity, we assume that the frequency of c_1 is greater than or equal to that of c_2.

  If n is an index page, the most appropriate action will be to delete n and create in p a direct link to c_1 and another to c_2. However, since this time two links will have to be added in p while only one is deleted, it can only be possible if p has an extra link to spare. If not, either c_1 and c_2 have to be merged or p will link to c_1 which will in turn link to c_2. Two or more pages are mergable if they have the same parents, the total number of links in these pages does not exceed the threshold, and the pages are all HTML pages.

If n is a contend page, c_1 is promoted only if p has a free link since we cannot delete n.

- n has three or more children, $c_1, c_2, \ldots, c_k$.

  Again, we assume that the children are sorted in decreasing order of their frequency. That is, c_1's frequency is the largest, followed by c_2 and so on until c_k.

  If c_1 is significant (its frequency is larger than the sum of all other's) and p has a free link, c_1 is promoted by adding a link from p to c_1 and deleting the link from n to c_1.

  If c_1 is significant but p does not have a free link, we switch n and c_1 if c_1 has a free link.

  For remaining infrequent children, c_i, ..., c_k, they are merged whenever possible to free links in n.

The algorithm for reorganization is outlined as follows.

Algorithm 3.2 Input: (1) an internal representation of a Web site with frequency and category for every page, and (2) thresholds I, C, and F.

Output: A reorganized Web site.

Method: A top-down approach with localized adjustments.

```
Initialize a queue $Q$
Put children of the home page into $Q$
Mark the home page
While $Q$ not empty
  $n \leftarrow pop(Q)$
  Mark $n$
  For each parent $p$ of $n$
    adjust the site according to the number of children of $n$ as discussed above
  for each child $c$ of $n$
    add $c$ into $Q$ if $c$ is not marked
```

4 Experiments

Our approach has been implemented using C++ on a Sun Microsystems Ultra 10 workstation with 256MB of memory running Solaris 2.6. The data set (log files) used was obtained from http://www.cs.washington.edu /homes/map/adaptive/download.html. These logs were of the Hyperreal Web site (*http://www.hyperreal.org*), taken during the period from September 1997 to October 1997. The server logs have already been put into sessions. The log files are approximately 79MB in size containing over 78,000 sessions spanning over 700,000 page requests.

Various experiments have been performed to study the performance of the page classification and site reorganization algorithms.

4.1 Page Classification Experiments

To evaluate the page classification algorithm, we study the precision of classification for various values of γ and t_l. The precision of classification for a category is calculated as the percentage of pages in the category which are correctly classified.

Figure 1 shows the precision of classification for various values of t_l. The value of γ is 60%. As shown in Figure 1, when t_l increases, the precision for index pages decreases and the precision for content pages increases, because less pages are classified as index pages. The best overall precision is achieved when t_l is 20.

Figure 2 shows the precision of classification for various values of γ. The value of t_l is 20. As shown in Figure 2, when γ increases, the precision for index pages increases and the precision for content pages decreases, because more pages are classified as index pages. the best overall precision is achieved when when γ is between 40% and 70%.

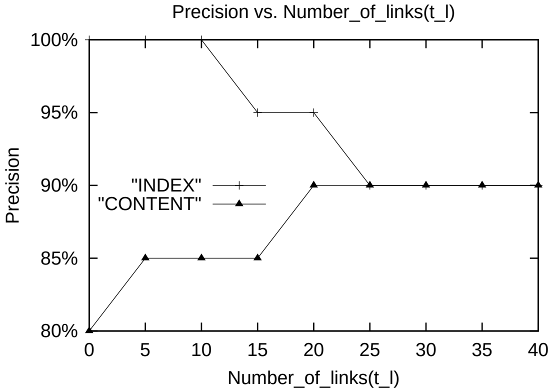

Figure 1: Precision for different values of t_l

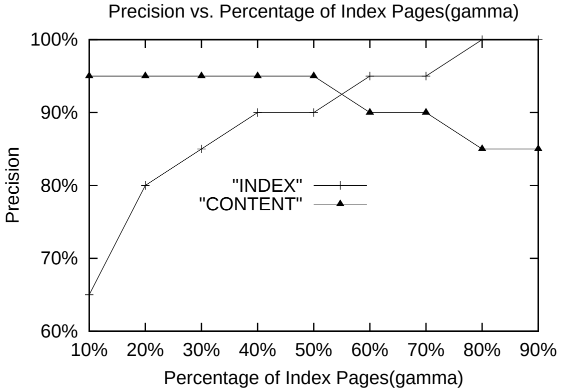

Figure 2: Precision for different values of γ

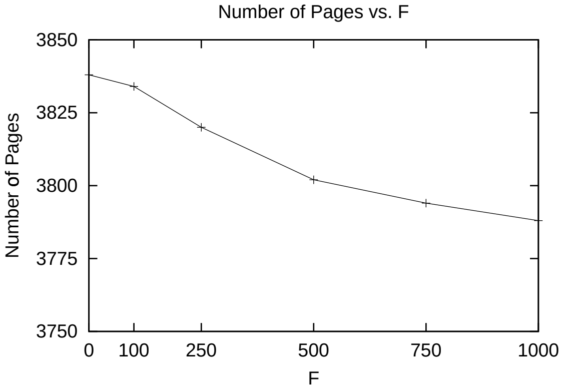

Figure 3: Number of pages for different Fs

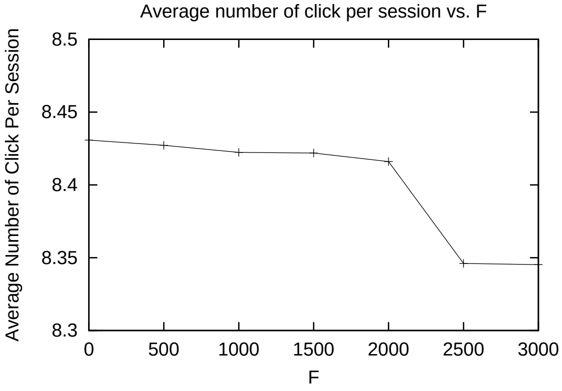

Figure 4: Average number of clicks per session

4.2 Reorganization Experiments

To evaluate the effectiveness of the reorganization algorithm, we examine the number of pages and links, as well as the average number of clicks in a session, before and after the reorganization. The values of I and C are set to 30 and 10, respectively. Results from most other values of I and C show only minor differences.

Figure 3 shows total number of resulting pages at the Web site after reorganization using various values of F. For the original Web site, F is 0. When F increases, more pages will be removed or merged since more pages become infrequent. Similar patterns are found for the number of links.

Figure 4 shows the average number of clicks per session on the Web site after reorganization using various values of F. For the original Web site, F is 0. The reorganization algorithm reduces the average number of clicks per session. However, it is not reduce by many since the majority of pages are content pages and thus not changed.

5 Conclusions and Future Work

An approach for the reorganization of a Web site based on user access patterns has been presented. The algorithms proposed have been successfully implemented and tested. The results show a high precision in page classification and a decrease in the number of clicks needed by users to navigate for the information they desire. The approach is efficient and practical for adaptive Web sites.

Our experiments show that the parameters play an important role in site reorganization. The setting of their values is done empirically. A further investigation on this issue to find a general model for selecting appropriate values is interesting.

Currently, we assume one page consists of a single file on the server. Another interesting extension to our work will be to apply our approach on page views. A page view is defined as all of the files that contribute to the client-side view of the requested information as the result of a single mouse click of a user. If a page view contains several files, such as a framed page, we need to consider the individual files in the page view as well as the relationships among the files.

References

[1] J. Borges and M. Levene. Mining association rules in hypertext databases. In *Proc. 1998 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'98)*, pages 149–153, August 1998.

[2] A. Büchner and M. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *SIGMOD Record*, 27, 1998.

[3] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *Proc. Int. Conf. on Tools with Artificial Intelligence*, pages 558–567, Newport Beach, CA, 1997.

[4] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1, 1999.

[5] Y. Fu, K. Sandhu, and M. Shih. Clustering of web users based on access patterns. In *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*, San Deigo, CA, August 1999.

[6] S. Madria, S. Bhowmick, W. K. Ng, and E. P. Lim. Research issues in web data mining. In *Proc. DAWAK'99*, Florance, Italy, September 1999.

[7] B. Mobasher, R. Cooley, and J.Srivastava. Creating adaptive web sites through usage-based clustering of urls. In *Proc. IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, November 1999.

[8] G. Paliouras, C. Papatheodorou, V. Karkaletsis, and C. D. Spyropoulos. Clustering the users of large web sites into communities. In *Proc. Inter. Conf. on Machine Learning (ICML)*, pages 719–726, Stanford, CA, 1999.

[9] M. Perkowitz and O. Etzioni. Adaptive web sites: An ai challenge. In *Proc. Int. Joint Conf. on AI (IJCAI)*, pages 16–23, 1997.

[10] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *Proc. 15th National Conf. on Artificial Intelligence (AAAI/IAAI'98)*, pages 727–732, Madison, Wisconsin, July, 1998.

[11] A. Scime and L. Kerschberg. Websifter: An ontology-based personalizable search agent for the web. In *Proc. Inter. Conf. Digital Libraries*, pages 439–446, Kyoto, Japan, November, 2000.

[12] C. Shahabi, A. Z. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. In *Proc. of 1997 Int. Workshop on Research Issues on Data Engineering (RIDE'97)*, Birmingham, England, April 1997.

[13] M. Spiliopoulou and L. Faulstich. Wum: A web utilization miner. In *Proc. EDBT Workshop WebDB'98*, Valencia, Spain, 1998.

[14] M. Spiliopoulou, L. Faulstich, and K. Winkler. A data miner analyzing the navigational behaviour of web users. In *ACAI'99 Int. Conf., Workshop on Machine Learning in User Modelling*, Florance, Italy, July 1999.

[15] O. R. Zaïane, X. 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*, pages 19–29, 1998.