Speeding Up Web Access Using Weighted Association Rules

Abhinav Srivastava, Abhijit Bhosale, and Shamik Sural

School of Information Technology,
Indian Institute of Technology, Kharagpur
{abhinavs, abhijitb, shamik}@sit.iitkgp.ernet.in

Abstract. In this paper, we propose a data mining technique for finding frequently used web pages. These pages may be kept in a server's cache to speed up web access. Existing techniques of selecting pages to be cached do not capture a user's surfing patterns correctly. We use a Weighted Association Rule (WAR) mining technique that finds pages of the user's current interest and cache them to give faster net access. This approach captures both user's habit and interest as compared to other approaches where emphasis is only on habit.

Keywords: Data Mining, Web Server, Weighted Association Rule (WAR).

1 Introduction

Increasing popularity of the web has resulted in heavy traffic in the Internet. The effect of this growth is a significant increase in the user perceived delay. The reasons of delay are network congestion, low bandwidth, improper utilization of bandwidth and heavy load on web servers and propagation delay. The most obvious solution is to increase the bandwidth. But, it is not that easy because of required infrastructure and cost. The propagation delay cannot be reduced after certain stage, since it depends on the physical distance between source and destination. One way to speed up the web access is to prefetch those pages that show user's habit and interest and put them into the cache. When user requests for these pages, it will be served from the cache.

URLs requested by users are stored in the web logs. These URLs show information accessed by the users and give their surfing pattern. Data mining techniques can be used to mine these logs and extract association rules between the URLs requested by the users. The association rules will be of the form $X \rightarrow Y$ where X and Y are URLs. It means if a user accesses URL X then he would be accessing URL Y most likely.

Previous works in this field concentrate only on finding the user's habit. Srivastava et al [1] has attempted to provide a survey of web usage mining and its application in analyzing web usage data to better serve users. Nanopoulos et al [2] focus on the predictive prefetching and develop a new algorithm for the same. Woon et al [3] propose a novel and more holistic version of web usage mining

named TRAnSACTIONized Logfile Mining (TRALOM) to effectively and correctly identify transactions as well as to mine useful knowledge from web access logs. Vaisman et al [4] study the data mining techniques (association rules, sequential patterns) and present a complete architecture of data mining proxy. However, most of these approaches only consider the user's habit and not the interest. User's interest keeps on changing while his habit typically remain constant. The existing approaches fail to capture this information.

In this paper, we show that both user's habit and interest can be captured by a Weighted Association Rule Mining (WAR) technique. We assign weights to all the URLs stored in a web log to distinguish recently accessed URLs from the old ones. The rest of the paper is organized as follows: In Section 2, we introduce the WAR technique and give salient features of our work. In Section 3, we present details of the experiment that we have carried out and give results. Finally, we conclude the paper with some discussions.

2 WAR Approach

User surfing patterns can be categorized into two types. (i) Regular accesses such as news, email etc. (ii) Occasional accesses based on the events such as Olympics. The former is known as habit and the later is known as interest. Users interest changes based on the events happening at any point of time.

Before we present our approach in detail, let us consider following situation: UserA logs on to Internet everyday for reading news and checking emails. He visits mail.yahoo.com and rediffmail.com in any order. However, because of the Olympics season, he starts visiting Olympics website to see the latest events and scores. In this case, Plain Association Rule (PAR) would give rules (UserA, mail.yahoo.com) $\rightarrow$ (UserA, rediffmail.com) $\wedge$ (UserA, rediffmail.com) $\rightarrow$ (UserA, mail.yahoo.com). PAR will not detect association rules that contain the Olympics website because UserA has started visiting this website recently and to become the rule, it should be visited enough so that its support can go above the minimum support. On the other hand, using WAR scheme following rules will also be generated: (UserA, mail.yahoo.com) $\rightarrow$ (UserA, www.Olympics.com) $\wedge$ (UserA, rediffmail.com) $\rightarrow$ (UserA, www.Olympics.com). These rules will provide real activity of UserA and captures both the habit (emails, news) and interest (Olympics) of the user. Hence, to come up with good frequent set of URLs accessed, it is important to consider user's interest also. Here are some salient features of the WAR technique:

For each URL stored in the web log, we assign a weight based on how recently it was accessed. The purpose of this weighting process is to categorize URLs into most recently accessed and less recently accessed. To achieve this, we divide the log file into different windows and assign each window a number known as windowIndex starting from the last window. The window with windowIndex 1 contains most recently accessed URLs, which are at the end of the log file. We assign all the URLs, which are in window 1, a highest weight $hWeight$, where $0 < hWeight < 1$. Let i , $weight_i$ and N be the windowIndex, weight of i^{th} window

and total number of windows (i.e. maximum value of i) respectively. $weight_i$ is calculated as follows:

$$weight_i = (hWeight)^i \quad (1)$$

It is evident from equation (1), that as we move from i to j where i and j are windowIndex and $i < j$, $weight_j < weight_i$. Since weight of a window represents weight of each URL in that window, weight assigned to a URL also decreases. So, the total weight of any URL u can be calculated as follows:

$$totalURLWeight = \sum_{i=1}^N weight_i \quad (if \ u \in i^{th} \ window) \quad (2)$$

Once the weight is calculated for all the URLs present in the web log, a minimum value of support is required. Unlike other approaches [2][3][4] where support is chosen only on the basis of count of URLs, we consider both weight and count of URLs for calculating support. So, for $X\%$ minimum support the weight would be:

$$(X * \sum_{i=1}^N weight_i) / 100 \quad (3)$$

Finally, we modify the existing apriori algorithm [5] to support weighted URLs. In WAR, modified apriori algorithm works by counting the weight of each URL in all the windows in each phase using equation (2). Once we calculate the total weight of each URL, we eliminate those URLs which are below the minimum support. We iterate this process until no more frequent URLs set can be generated.

The URLs which a user accesses as a part of his habit, should be present in most of the windows including recent windows also. So, they must have high weights and will be part of frequent URLs set. The URLs which the user accesses as part of his interest should only present in the recent windows. Since recent windows have high weights, it is more likely that the total weight of a URL is above the minimum support and will be part of the frequent sets. Hence, the frequent URLs set generated by WAR contains mix of URLs capturing both habit and interest of the users. These frequent URLs can be used to speed up the web access by prefetching the URLs and present to the user whenever it is requested.

3 Experimental Results

A log file of a proxy server was used for the analysis of our work. The log was generated for a period of 8 working hours between 9 am and 5 pm. First, the log file is cleaned by removing the URLs of dynamic pages like jsp, asp etc. Then the log entries are separated as per requesting IP addresses. Here an IP address is considered as identifier for a user, assuming that only one user is using a particular computer, for accessing the web. The log entries of each IP address are processed separately. A window size is decided for a transaction. For experimental purpose, we have selected three different time windows of 1 min,

5 min and 10 min. Hence the log is partitioned in a group of URLs falling in a window, forming a transaction. Now, these transactions are used to find the frequent item sets of URLs.

We compare performance of our approach with Vaisman et al's approach that use PAR [4]. Here, due to size restriction we have included a few sample results of our experiments. We plot the Support vs. Number of Transactions graphs for different window sizes. Figures 1(a), 1(b) show graphical results for window size of 1 min, Figures 2(a), 2(b) show for 5 min and Figures 3(a) , 3(b) show for 10 min window. The bold line is the minimum support value while others are for the sites accessed by the user. The Figures 1(a), 2(a) and 3(a) are results of PAR approach. In these Figures, access count of each URL is treated as the support. The Figures 1(b), 2(b) and 3(b) are results of WAR approach. In every graph a particular line represents how the support value changes with number of transactions. For example let us consider the regular continuous line, which represents the support values for site <http://10.105.8.42>. In Figure 2(a), at the time when total number of transactions generated from the log file is 50, the URL <http://10.105.8.42> is accessed 8 times since beginning. While from Figure 2(b), we can observe that at the same time the calculated weight for the URL <http://10.105.8.42> is little less than 1.

Consider Figure 2(a), if we notice the accessed site <http://10.105.8.42> (regular continuous line), we can see that it is always above the minimum support and hence will be always become a part of the frequent item set. We can see that it is actually not accessed during the period of time from transactions 40 to 65. This means that user is not interested in this site, at least for some period. But still it is considered as a frequent item. Consider <http://10.112.5.205> (dashed line with cross mark) , which is accessed by the user for the first time in transaction number 60. After that, the user frequently accesses it since he gets interested in that side. But it still falls below the minimum support value and not considered as a frequent item.

(a) Non-weighted support

(b) Weighted support

Fig. 1. Comparison of support vs. transactions for 1 minute window

Fig. 2. Comparison of support vs. transactions for 5 minutes window

Now, let us see how WAR helps in finding the user's interest that is not considered by PAR. In Figure 2(b), we can see that the weight for the site `http://10.105.8.42` falls below minimum support at transaction number 46. Hence it will not be considered as frequent item during the period when the user is not interested in visiting that site. If we consider site `http://10.112.5.205`, we can observe that site is able to cross the minimum support at transaction number 66, just after 6 transactions of its first access. Hence a user's interest is better tracked in WAR.

The most critical part in the weighted support system is the selection of window size and minimum support, which need fine tuning as per the situation. In Figures 1(a), 1(b), where window size is 1 min, the support percentage needs

Fig. 3. Comparison of support vs. transactions for 10 minutes window

to be lowered to get the similar results as we have obtained for 5 min window. On the other hand, as in case of Figures 3(a), 3(b) where the window size is 10 min, support percentage needs to be increased to obtain similar results. We can include a learning phase in our framework in which a sample log file will be used to find out the appropriate values of window size and minimum support percentage.

4 Conclusions and Discussions

In this paper, we have presented a new approach in which a weight is assigned to each URL for capturing its recency. We have used a weighted apriori algorithm to generate the frequent URL set. We also have suggested how this result can be used to speedup web access and reduce a user's perceived latency. This result can be improved by adding a learning phase in which window size and minimum support values can be tuned. Our work highlights the need for some mechanism to distinguish between frequently and recently accessed items by a user to predict his current interest and habits.

References

1. J. Srivastava, R. Cooley, M. Deshpande, P. N. Tan, *Web usage mining: Discovery and applications of usage patterns from web data*, SIGKDD Explorations, (2000) 12-23.
2. A. Nanopoulos, D. Katsaros, Y. Manolopoulos, *Effective prediction of web user access*, WEBKDD'01, (2001).
3. Y. Woon, W. Ng, E. Lim, *Online and incremental mining of separately-grouped Web access logs*, 3rd International Conference on Web Information Systems Engineering. (2002) 53-62.
4. A. A. Vaisman, G. Dandretta, M. Sapia, *Enhancing Web access using datamining techniques*, 14th International Workshop on Database and Expert Systems Application, (2003) 327-331.
5. A. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Verkamo, *Fast Discovery of Association Rules*, 20th International Conference on Very Large Database, (1996) 307-328.