Proceedings of the 5th World Congress on Intelligent Control and Automation, June 15-19, 2004, Hangzhou, P.R. China

第五届全球智能控制与自动化大会
2004 年 6 月 15 - 19 日，中国杭州

An Intelligent Algorithm of Data Pre-processing in Web Usage Mining

Zhang Huiying, Liang Wei
School of Management
University of Tianjin
Tianjin, 300072, P.R.China
zhanghuiying@nankai.edu.cn

Abstract **- Web Usage Mining is the application of data mining techniques to usage logs of large Web data repositories in order to produce results used in some aspects, such as Web site design, Web server design, users classification, creating adaptive Web sites and Web site personalization. Data preprocessing is a critical step in Web Usage Mining. The results of data Preprocessing is relevant to the next steps, such as transaction identification, path analysis, association rules mining, sequential patterns mining, and so forth. An algorithm called "USIA" was presented and its advantage and disadvantage were analyzed, USIA is experimentally evaluated that not only its efficiency is high, but also it can identify user and session exactly.**

Index Terms - Web Usage Mining; Data Pre-processing; Users identification; Session Identification.

I. INTRODUCTION

The data source of Web usage mining is Web log files, from which we can realize users' browse patterns by Web usage mining. The patterns can be used in some aspects, such as:

1) Redesigning Web sites and some link.

2) Understanding users' interests and creating individual pages for them;

3) Classifying users and implement different sales promotion to different users to improve ROI;

4) Recommending Web pages to users, and some more other application.

The early methods of web usage mining can be seen in [1] and [2]. Data pre-processing is the first step of Web usage mining. The results of data pre-processing directly impact the results of next steps including transaction identification, path analysis, association rules mining and sequential patterns mining. In a word, if we get better result from the first step, we will improve the mined patterns' quality and save algorithm's running time. It is especially important to web log files, in respect that the structure of web log files are not the same as the data in database or data warehouse. They are not structured and complete due to various causations. So it is especially necessary to pre-process web log files in web usage mining. Through data pre-processing, web log can be transformed into another data structure, which is easy to be mined. Figure 1 shows an integrated process of web usage mining. It includes data pre-processing, mode mining, mode analysis and mode visualization. This paper will focus on the module of data pre-processing consisted of data cleaning, users identification, session identification and path completion, just as Figure 1 showed.

II. DATA PREPROCESSING

A. Data Cleaning

The purpose of data cleaning is to eliminate irrelevant items, and these kinds of technique are of importance for any type of web log analysis not only data mining. The discovered associations rules or reported statistics are useful none but the data represented in the server log gives an accurate picture of the user accesses to the Web site.

On account of that the HTTP protocol requires separate connections for every file requested from the Web server. A user's request to view a particular page often results in several log entries since graphics and scripts are downloaded in addition to the HTML file. In most cases, only the log entry of the HTML file request is relevant and should be kept to the user session file, for as much as, in general, a user does not explicitly request all of the graphics on a Web page, which are automatically downloaded due to the HTML tags. Since the main intent of Web Usage Mining is to get a picture of the user's behaviour, other than include file requests that the user did not explicitly request, elimination of the items deemed irrelevant can be reasonably accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as gif, jpeg, GIF, JPEG, jpg, JPG, and

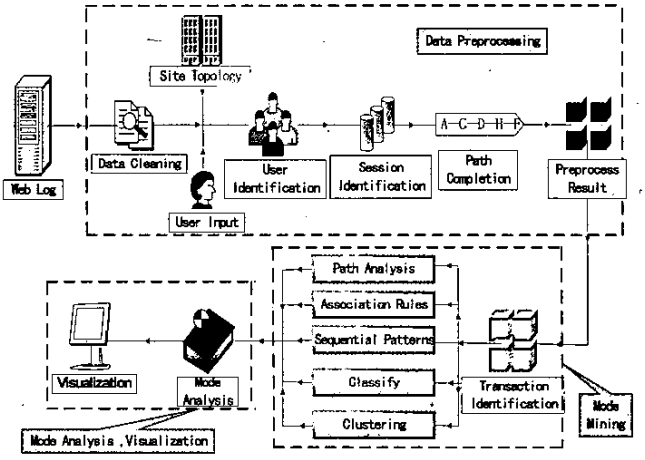

Fig. 1 Procedure of Web Usage Mining

map can be ignored. In addition, common scripts such as "count.cgi" can also be ignored.

Removing the irrelevant items can reduce the data that will be analysed and increase the analysis's speed. It also can decrease the irrelevant items' negative influence to the mining process. For example, the size of total Web log file of Tianjin University's Web site (http://www.tju.edu.cn/) from Mar. 1, 2003 to Mar 7, 2003 is 105M byte, containing 1,174,093 records before data cleaning. After removing irrelevant items it remain 378,747 records, therewith we can see this step can remove a mass of irrelevant items.

B. Users Identification

Users identification is to identify who access Web site and which pages are accessed. If users have login of their information, it is easy to identify them. In fact, there are lots of user do not register their information. What's more, there are great numbers of users access Web sites through agent, several users use the same computer, firewall's existence, one user use different browsers, and so forth. All of problems make this task greatly complicated and very difficult to identify every unique user accurately. We may use cookies to track users' behaviours. But considering individual privacy, many users do not use cookies. So it is necessary to find other methods to solve this problem.

For users who use the same computer or use the same agent, how to identify them? As presented in [3], it uses heuristic method to solve the problem, which is to test if a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, the heuristic assumes that there is another user with the same computer or with the same IP address. Ref. [4] presents a method called navigation patterns to identify users automatically. But all of them are not accurate because they only consider a few aspects that influence the process of users identification. Considering this actuality, we presented a new algorithm called "USIA(User and Session Identification)". It analyses more factors, such as user's IP address, Web site's topology, browser's edition, operating system and referrer page. This algorithm possesses preferable precision and expansibility. It can not only identify users but also identify session. Session identification will be discussed in next section.

C. Session Identification

For logs that span long periods of time, it is very likely that users will visit the Web site more than once. The goal of session identification is to divide the page accesses of each user at a time into individual sessions. A session is a series of Web pages user browse in a single access.

The simplest method of achieving session is through a timeout, where if the time between page requests exceeds a certain time limit, it is assumed that the user is starting a new session. Many commercial products use 30 minutes as a default timeout. Because this method is easy and has been tested by some experiments, we use 30 minutes as timeout in our experiment.

D. Path Completion

Another critical step in data preprocessing is path completion. There are some reasons result in path's incompletion, for instance, local cache, agent cache, "post" technique and browser's "back" button can result in some important accesses not recorded in the access log file, and the number of URLs recorded in log maybe less than the real one. This problem is referred to path completion, which will influence next steps' efficiency and accuracy if it is not solved properly.

Methods similar to those used for users identification can be used for path completion. If a page request is made that is not directly linked to the last page a user requested, the log can be checked to see what page the request came from. If the page is in the user' s recent request history, the assumption is made that the user called up cached versions of the pages with the "back" button available on most browsers until a new page was requested. If the log is not clear, the site topology can be used to the same effect. If more than one page in the user's history contain a link to the requested page, it is assumed that the page closest to the previously requested page is the source of the new request [5]. Missing page references inferred through this method are added to the user session file. Although the method referred above cannot achieve 100 percent preciseness rate, but it was tested to obtain better result, and is available method. Another method is to use protocol HTTP／1.1 to avoid those problems come from local cache.

III. ALGORITHM - USIA

A. Algorithm's Thinking

USIA is an Algorithm about users identification and session identification. There are a lot of sequential records come from the same IP address in Web log files. If we use algorithm to check every record mentioned above, it will decrease algorithm's efficiency. If the current record's IP address is the same as previous record's, then we assume that the two record come from the same user. Now some definitions are given.

Definition 1: Usersi=(User_ID, User_IP, User_Url, User_Time, User_Referer_Page, User_Agent), 0<i<n, where n is the number of total users; User_ID is users' ID have been identified; User_IP is user's IP address; User_Url is Web pages user accessed; User_Time is time user accessed; User_Referer_Page is the last page the user requested; User_Agent is agent user used.

We can identify a unique user through all of the factors mentioned above.

Definition 2: Sessionsi=(User_ID, Sj,[urlj1,urlj2,... urljk]), 0<i<n, where n stands for the number of total sessions; User_ID stands for users' ID that have been identified; Sj stands for one of the user's sessions; urljk stands for a aggregate of Web pages in session Sj.

Definition 3: Cube=(User_ID, Sj, User_IP,[(urlj1, tj1), (urlj2, tj2), ... (urljk, tjk)], where Cube is stored structure of users and sessions that have been identified by algorithm;

User_IP stands for user's IP address; tjk stands for the time user accessed urljk.

In USIA, User_ID, Sj and urljk are the same as definition 2. The detail expression of algorithm is given as Fig. 2.

B. Procedure of USIA

Input: Web log files; Timeout- TimeSpan.

Output: User's ID- User_ID; User's Session- Session

(Considering detailed algorithm will need more space, we only list framework of the algorithm.)

String[, ,] Cube; //Define a three-dimensional array. Using it to store User_ID ,r.IP, r.Url (Web pages' urls), r.Time(time which user access the Web page) and number of total Web pages in a single session.

Procedure foreach (Record r∈Log) //Where r is a record in Web log files.

```
{
    if (r.IP<>Last-r.IP) // Current record's IP address is not the same as previous record's IP address. Where Last-r.IP is previous record's IP address.
    {
        if ( isExistedIP ( Cube, r.IP ) = false ) //New IP address and there isn't the IP address in the user's aggregate that have been identified
        {   //Storing User_ID, r.IP , r.Url, r.Time
            User_ID++;
            SaveCube ( i, j, k, User_ID, r.IP, r.Url, r.Time );
        }
        else if ( isSameUser ( Usersi, r ) = true ) //Current user is the same user in the Usersi that have been identified..
        {
            if ( r.Time - Cube [ i, j, 1 ]< TimeSpan ) // If the time between page requests is not exceeds a certain limit, it is assumed that it is a single session.
            {
                j=j+1;
                Cube [ i, j, 0 ]= r.Url;
                Cube [ i, j, 1 ]= r.Time;
            }
            else    //If the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session.
            {
                Cube [ i, 0, 1 ]= j - 2;  //Where j - 2 is the number of total Web pages user accesses in a single session.
                i= i + 1;
                k ++ ;
                j=  0;   //j=0 because there is a new session.
                SaveCube ( i, j, k, User_ID, r.IP, r.Url, r.Time );
            }
        }
        else  //The current user is not the same as previous user.
        {
            User_ID = User_ID + 1;
            i= i+ 1;
            SaveCube ( i, j, k, User_ID, r.IP, r.Url, r.Time );
        }
    }
    else  // Current record's IP address is the same as previous record's IP address Then assume that the two records are made by one user.
    {
    j= j+ 1;
        Cube [ i, j, 0 ]= r.Url;
        Cube [ i, j, 1 ]= r.Time;
    }
}
```

Procedure isExistedIP (Cube, r.IP) //Searching User_Listi which is a user's aggregate has been identified, just like definition 3. If r.IP is exist in the User_Listi then return true, else return false.

Procedure isSameUser (Usersi, r) //Checking up Usersi . If current user is the same user in Usersi then return true and k which is number of the user's sessions else return false.

Procedure SaveCube (i, j, k, User_ID, r.IP, r.Url, r.Time) //Storing User_ID, r.IP, r.Url, r.Time.

```
{
    Cube [ i, j, 0 ]= User_ID + "-" + k;  // k is number of the user's sessions.
    Cube [ i, j+1, 0 ]= r.IP;
    Cube [ i, j+2, 0 ]= r.Url;
    Cube [ i, j+2, 1 ]= r.Time;
}
```

The hypothesis of above algorithm is based on two preconditions as followed:

1) Acceptable Users identification's Precision: In order to examine users identification's precision, we chose 200 sequential records from Web log files randomly. Every sequential part comes from different IP address. Using hypothesis it is assumed those records are made by 200 different users. Using IsExistedIP and IsSameUser algorithms to analyze those records, we identified 202 different users. Comparing the two results, the hypothesis is acceptable.

2) Algorithm's Efficiency: Using the hypothesis may avoid checking up lots of records stage by stage and it can save a lot of time.

C. Algorithm's Advantage and Disadvantage

1) Good Precision: In traditional methods, IP address was used as criterion to identify user and algorithm's precision was unacceptable. On the contrary, we check User_ID, User_IP, User_Url, User_Time, User_Referer_Page and User_Agent to identify unique user. So there is a good precision.

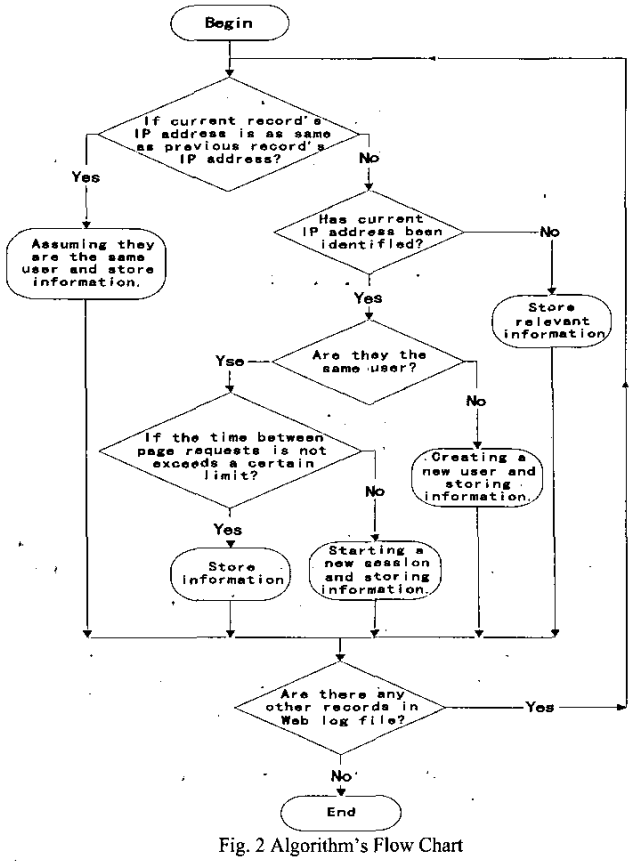

Fig. 2 Algorithm's Flow Chart

2) High Efficiency: The algorithm may identify unique user and session at the same time and avoid low efficiency by accomplishing that separately.

3) Favourable Structure of Store: For users and sessions identified by USIA, we construct a three-dimensional dynamic array to store User_ID, r. IP, r. Url and r. Time, avoiding store space's waste. In Figure 3, the axis of Users - Sessions are users and sessions, where n is number of total users identified and k is the user's the k-th session. 1-1, 2-1, 3-1, n-k stands for different user's sessions. The same user may have more than one session.

From Fig. 3 we can see that user 3 have two sessions: 3-1 and 3-2. Where IPn is the n-th user's IP address. The axle of IP - Urls is a sequence of Web pages' url in a single session, such as A, B, C, D, E, F and so forth. The axle of time is the time user accesses Web page. For example, 2003-3-1 10:21:36 is the time user 1 accessed Web page A. Where "12" is a number of total Web pages user 1 accessed in session 1-1. All of the results identified have been stored in the cube and it will facilitate next analysis remarkably.

4) Disadvantage: As the algorithm need check several factors when it judge whether the current user is the same user in Usersi or not, it results in more time needed. But analysing Web log files in real time is not needed, so the algorithm's speed is not the most important. Comparing speed and precision, we think precision is more important.

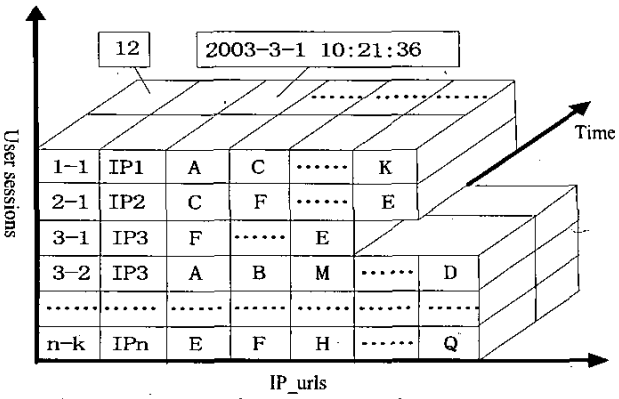

Fig. 3 Representation of storage structure of Users and Session

IV. EXPERIMENTAL RESULT

Our data source is Web log file of Tianjin University's Web site (http://www.tju.edu.cn/), which is from Mar. 1, 2003 to Mar. 7, 2003. The size of total Web log file is 105M bytes. After data cleaning it remain 378,747 records. Experimental condition is AMD Athlon (tm) XP 1700+ and 256M memory.

USIA identified 55,625 users and 78,566 sessions. If we only use IP address as criterion to do it, we identified 52,870 users. That is to say there are 55,625 - 52,870 = 2,755 users neglected. It is obvious that USIA possess better precision. From marketing manager's angle, if the 2,755 users are identified precisely, it will bring great benefit to company. Because they can adopt corresponding methods to attract their attention then turn them into loyal customers.

V. CONCLUSION

Our work focus on data pre-processing of Web usage mining and we also laid particular emphasis on algorithm's realization. USIA checks User_ID, User_IP, User_Url, User_Time, User_Referer_Page and User_Agent to identify unique user. So there is a better precision than traditional methods. We hope our work can be helpful to other researchers. We will consider user clustering [6] in the future and we think it will help us to get more precise experiment's result.

There are lots of researchers are studying Web usage mining. But few of them make great progress. We are sure there will be more and more researchers and fund go into this area because this area has great commercial value, wide application prospect and relevant technique's development prospect. The emphases of study will continue focus on mode analysis, result's visualization and man-machine interaction.

REFERENCES

[1] H. Mannila, H Toivonen, and A. I. Verkamo, "Discovering Frequent Episodes in Sequences", *Proc. of the 1st Int. Conf. on Knowledge Discovery and Data Mining*, Montreal, Canada, August 1995.

[2] J. Pitkow, "In search of reliable usage data on the www", *Proc. 6th Int. WWW Conf.*, Santa Carla, CA, pp. 451–463, 1997.

[3] P. Pirolli, J. Pitkow, and R. Rao, "Silk from a sow's ear: Extracting usable structures from the Web", *Proc. of 1996 Conference on Human*

Factors in Computing Systems (CHI-96), Vancouver, British Columbia, Canada, 1996.

[4] R.Cooley, B.Mobasher, and J.Srivastava. "Grouping Web page references into transactions for mining world wide Web browsing patterns", *Proc. of the IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)*, 1997.

[5] W. Gaul and L. Schmidt-Thieme. "Mining Web navigation path fragments". *Proceedings of the Workshop on Web Mining for E-Commerce -- Challenges and Opportunities*, Boston, MA, Aug. 2000.

[6] Ypma, A., Heskes, T. "Categorization of Web Pages and User Clustering with mixtures of Hidden Markov Models", *Proceedings of the International Workshop on Web Knowledge Discovery and Data Mining, WEBKDD'02*, July 23 2002, Edmonton, Canada.