Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2014, *8*, 405-409 405

Open Access

Algorithm for Mining Web Access Patterns Based on User Access Sequences

Sun Jinhua*, Meng Zhaorui and Xie Yanqi

School of Computer and Information Engineer, Xiamen University of Technology, Xiamen, Fujian 361024, P.R. China

Abstract: Through analysis of user access sequences, this research studies the browsing behaviors of web users and proposes a user access pattern mining algorithm based on the maximal forward reference sequence to determine frequent access paths. Experiments have shown that this effective algorithm could obtain a satisfactory result. This algorithm is superior to the traditional Full Scan algorithm in terms of execution time and performance.

Keywords: Data mining, maximal forward reference sequence, user access pattern.

1. INTRODUCTION

The access log of WWW is kept in each Web server. Web usage mining is the process of analyzing Web logs to determine user access pattern and to extract information or pattern in which users are interested [1]. This process helps to identify users' loyalty, preference, and satisfaction; find potential users; and enhance the service competitiveness of the sites, which is significant in the field of e-commerce [2, 3].

Web access pattern mining is an important direction of Web usage mining. Through the data mining technique and algorithm analysis of Web logs, behavior characteristics of the users can be determined and judged; thus, the website's structure can be adjusted, and personalized service can be provided [4, 5]. Among studies on mining algorithm, the most representative Apriori algorithm [6], the DHP algorithm using the hash table [7], and the improved Full Scan algorithm based on the DHP algorithm [8] are all expected to mine the important information hidden in the Web logs in a more effective way. Despite the careful design, these algorithms still have some problems, such as low execution efficiency because of repeated database scanning when being applied to Web access pattern mining. Therefore, their performance should be improved.

Based on the characteristics of Web browsing, this research studies the mining process of Web access patterns frequently browsed by users from Web logs and designs a mining algorithm based on the maximal forward reference sequence to determine the user access pattern. Experiments have shown that this algorithm could effectively mine the user access pattern.

2. PROBLEM DESCRIPTION AND RELEVANT DEFINITIONS

The Web access pattern mining aims to mine the Web pattern frequently browsed by users from Web logs. The user access pattern is also a sequential pattern, that is, an ordered set of data items. The sequence of an access page varies in different access patterns [8, 9].

For convenience of describing the algorithm, relevant definitions are provided below.

Definition 1: The maximal forward reference sequence is generated after identifying the preprocessed log data by Transaction (t). This sequence is a user access pattern.

Definition 2: The reference sequence length is defined as the number of pages in the user access pattern t. A t with k pages is known as k-reference sequence.

Definition 3: If the reference sequence satisfies the minimum support conditions set by the user, the sequence is known as the large reference sequence.

Based on the above definitions, the Web access pattern mining process can be summarized into two steps:

(1) The original user access sequence is transferred into many maximal forward reference sequences.

The forward reference sequence is the primary consideration in designing the algorithm; the backward reference is the action performed by the user for convenience of browsing some web pages. Therefore, the backward reference is insignificant to the Web access pattern mining and should be removed [6].

(2) The frequently browsed patterns, which are all large reference sequences, are found.

In both steps, the second is the key.

3. ALGORITHM DESIGN

3.1. Finding the Maximal Forward Reference Sequence

This study uses a method similar to that in [8, 10] to generate all maximal forward reference sequences. This method assumes that user access page sequences are $P_1P_2...P_n$, which can be expressed in composition of sequence pairs $(NULL, P_1)$, (P_1, P_2), (P_2, P_3), …, (P_{n-1}, P_n). These sequence pairs are generated according to the sorting of access times.

1874-110X/14 2014 Bentham Open

For convenience of describing the algorithm, the sequence pairs are expressed as (S_1 D_1), (S_2 D_2), …, (S_n D_n). The starting and ending nodes are represented by NULL. Using the method similar to the depth-first search, the original user access sequences can be divided into several maximal forward reference sequences [6]. The following algorithm is used to obtain the maximal forward reference sequence briefly known as GMFRS.

Algorithm: GMFRS: solve the maximal forward reference sequence

Input: Original user access sequence

Output: maximal forward reference sequence

Step 0:

$R = NULL$; $flag = 1$; $i = 1$; // initialization of variable

Step 1:

read Sequence (S_i D_i);

$S=S_i$; $D=D_i$;

if (S==null)

{

$R=D$;

goto Step 3;

}

Step 2:

if (the k^{th} terms in D and R are the same)

{

if ($flag$ == 1)

{

write R to the database DB;

delete all terms after the k^{th} term in R;

$flag = 0$;

}

else

{

Attach D to the end of R;

if ($flag$ == 0)

$flag = 1$;

}

}

Step 3:

if (the access sequence has not been completely scanned)

{

$i = i + 1$;

go to Step 1;

}

else

write R to DB;

In the GMFRS algorithm, the variable R stores the current forward reference sequence with the initial value of NULL. DB stores all maximal forward reference sequences; i represents the number of currently considered sequence pairs with the initial value of 1; identifier $flag$ = 1 refers to a forward browsing path; conversely, $flag$ = 0 refers to a backward browsing path; and the initial value of flag is set to 1.

The user access sequence to a Web station could be ABCBD (Fig. **1**). Subsequently, this pattern is used as an example to illustrate the transfer of the original user access sequence to the corresponding maximal forward reference sequence based on the previous GMFRS algorithm.

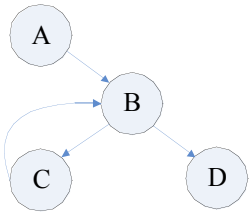

Fig. (1). Example for user access sequence.

The user access sequence (ABCBD) is expressed as (NULL, A), (A, B), (B, C), (C, B), (B, D). Changes in the relevant data of the execution process are shown in Table **1**.

Table 1. Execution process of the algorithm.

i	*R*	Data in DB	*flag*
1	A	NULL	1
2	AB	NULL	1
3	ABC	NULL	1
4	AB	ABC	0
5	ABD	ABC, ABD	1

Table **1** shows that the values of R and flag indicate the status after performing Step 2; the value of DB is the status after performing Step 3.

In the example, the final result of the maximal forward reference sequence with GMFRS is {ABC, ABD}.

3.2. Access Pattern Mining Algorithm

3.2.1. Theoretical Basis of the Algorithm

Mining of the Web browsing path can begin after transferring the original user access sequence to the maximal forward reference sequence (also known as the target sequence). This process aims to determine all the large reference sequences [8, 10], which satisfy the minimum support conditions set by the user.

Based on the above definition and assumption, two inferences can be drawn.

(1) If TC (J_1 J_2 … J_k) is greater than or equal to the minimum support number, TC(J_1 J_2 … J_k) is a large k-reference sequence.

(2) With the method similar to AprioriAll [11], the generated large (k-1)-reference sequence can be used to deduce the candidate k-reference sequence.

Assuming that L_{k-1} is the set of all large ($k-1$)-reference sequences, $J = (J_1\ J_2\ \ldots\ J_{k-1})$ and $P = (P_1\ P_2\ \ldots\ P_{k-1})$ $P\ L_{k-1}$ and $J_i = P_{i-1}$, $2 \le i \le k-1\}$ are the large ($k-1$)-reference sequences, and C_k is the set of the candidate k-reference sequences, then:

$C_k = \{(J_1\ J_2\ \ldots\ J_{k-1}\ P_{k-1})\ |J\ L_{k-1}$ and $P\ L_{k-1}$ and $J_i = P_{i-1}, 2 \le i \le k-1\}$

3.2.2. Algorithm Implementation

Based on the previous analysis, a method of mining the large reference sequence known as FLRS is designed to determine the large reference sequences. This algorithm mainly has three steps:

Step 1: All target sequences are scanned once, the frequency of occurrence of each web page and adjacent linked web page [such as (BD)] is recorded, and the sequence number is written down.

Step 2: The large 1-sequence and large 2-sequence are determined according to Step 1 and the minimum support number.

Step 3: The candidate k-reference sequence ($k \ge 3$) is generated according to Inference 2, and the intersection operation is performed with the TS value of the combined reference sequences.

The algorithm is described as follows:

Algorithm FLRS: find the large reference sequence

Input: user access target sequence

Output: all large reference sequences

Scan all target sequences once and record the frequency of occurrence of each web page; write down the frequency of occurrence of the adjacent linked web page and the sequence number.

$L_1 = \{X|X$ is a web in the target sequence and $TC(X) \ge$ the minimum support number$\}$;

$L_2 = \{Y|Y$ represents the sequence of the adjacent node web pages in the target sequence and TC(Y) $\ge$ the minimum support number$\}$;

for ($k = 3$; $|L_{k-1}| > 1$; k++) //$|L_{k-1}|$ is the number of elements in L_{k-1}

{

According to Inference (2), let L_{k-1} generate C_k

for (the candidate reference sequence $C \in C_k$)

{

maximum possible TS (C) = TS ($S1$) studying TS ($S2$);

maximum possible TC (C) = Card (TS (C));

if (maximum possible TC (C) $\ge$ minimum support number)

{

check whether the target sequence in Maximum Possible TS (C) contains Sequence C;

if ($TC(C) \ge$ the minimum support number)

$L_k = L_k \cup \{C\}$;

}

}

}

Once the candidate reference sequence is generated according to the values recorded by TS and TC, a large reference sequence can be distinguished easily.

4. EXPERIMENT AND PERFORMANCE EVALUATION

4.1. Experiment Environment

The algorithm is realized with Java (JDK1.5) on a PC with a Pentium 2.8 GHz processor, DDR400 1 GB memory, Windows Server 2003 operating system, and MySQL database. Web logs are the daily operation data of small- and medium-sized enterprises in Sanming of Fujian and are saved as text files. The number of records is about 100,000.

4.2. Experimental Process

This experiment mainly verifies the validity of the algorithm and tests its performance. Based on the general Web log mining process [12] and combined with the algorithm described in this paper, the experimental process is designed (Fig. **2**).

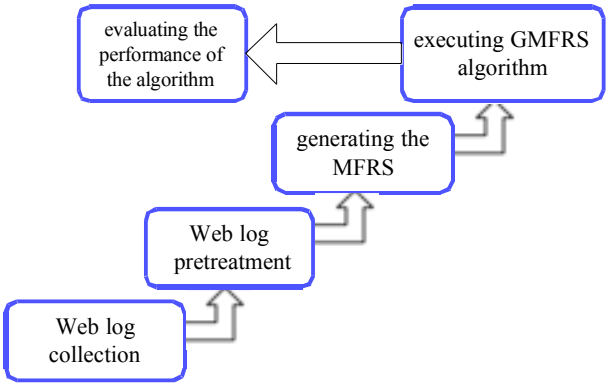

Fig. (2). Experimental process.

The experimental process includes Web log collection, log pretreatment, maximal forward reference sequence generation according to the algorithm GMFRS, mining algorithm process, and performance evaluation of the algorithm.

4.3. Analysis of Experimental Results and Performance Evaluation of the Algorithm

To evaluate the performance of the algorithm, the daily operation date of small- and medium-sized enterprises is loaded in the experiment system to conduct the actual operation test and compare the test results with the classical Full Scan algorithm [8].

(1) Comparison of performance between FLRS and FS algorithms

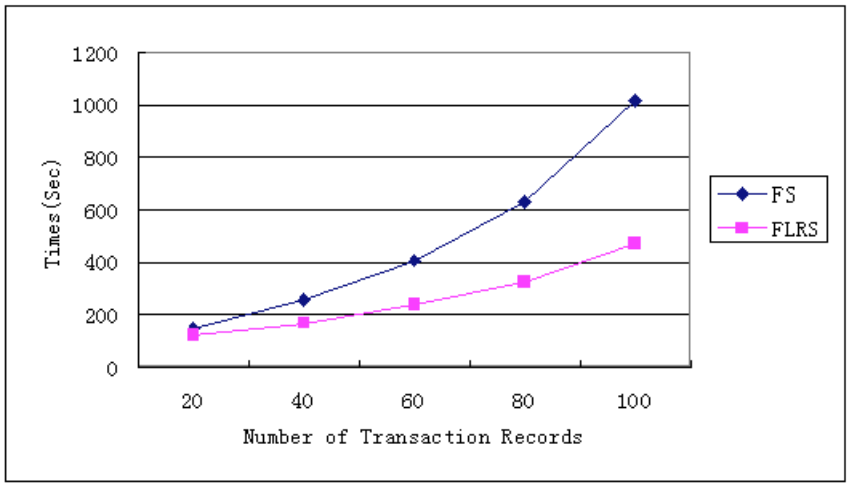

Fig. (3). Comparison of execution times between two algorithms under different amounts of data.

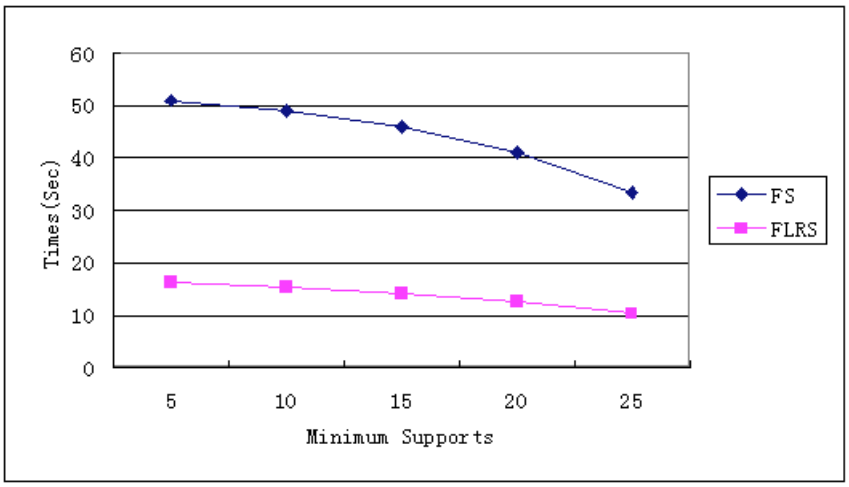

Fig. (4). Comparison of execution times between two algorithms under different supports.

The minimum support is set to 3. A comparison of the execution times under different amounts of data is shown in Fig. **3**.

Under different numbers of records (Fig. **3**), the FLRS algorithm is obviously superior to the classical FS algorithm in terms of execution time and performance; the gap is also gradually widened with the increase of the amount of data mainly because the FS algorithm needs to scan the data records many times during browsing pattern mining. With the increase of the amount of data, the reading time will also expand, but the FLRS algorithm needs to scan the data records only once.

(2) Algorithm execution under different supports

To determine the influence of different supports on the algorithm performance, the number of data records in the experiment is set to 10,000 to observe the execution time of the algorithms under different supports (Fig. **4**).

The change in the minimum support (support number) slightly affects the execution time of the FLRS algorithm because this algorithm needs to scan the data records only once during browsing pattern mining (Fig. **4**). The worst situation is that some data are read twice; thus, the execution efficiency is relatively stable. The FS algorithm will decrease the execution time with the increase of the minimum support degree because a large amount of candidate item sets occupies large memory space, and the I/O frequency increases when the FS algorithm generates the reference sequence. With the increase of the support, the candidate term sets will decrease gradually.

5. CONCLUSION

This paper discusses Web access pattern mining technology and suggests a maximal forward reference sequence-based Web access pattern mining algorithm for users. Therefore, this sequence designs the corresponding experimental process and realizes an experimental system. Experiments have shown that the algorithm could effectively mine the Web pattern that the users frequently access.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial subsidy provided by the Educational Commission of Fujian Province of China under project no. JB12184.

REFERENCES

[1] P. E. Román, G. L'Huillier, J. D. Velásquez, "Web usage mining", *Advanced Techniques in Web Intelligence-I*. Springer Berlin Heidelberg, pp.143-165, 2010.

[2] C. J. Carmona, S. Ramírez-Gallego, F. Torres, E. Bernal, M.J. del Jesus, S. García, "Web usage mining to improve the design of an e-commerce", *Expert Systems with Applications,* vol. 39, no.12, pp. 11243-11249, 2012.

[3] K. Sharma, G. Shrivastava, and V. Kumar, "Web mining: Today and tomorrow", *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*. vol. 1, 2011.

[4] D. Nie, A. Li, Z. Guan, T. Zhu "Your Search Behavior and Your Personality." *Pervasive Computing and the Networked World*. vol. 8351, pp. 459-470, 2014.

[5] R. Leung, J. Rong, G. Li, Rob Law, "Personality differences and hotel web design study using targeted positive and negative association rule mining", *Journal of Hospitality Marketing & Management,* vol. 22, no.7, pp. 701-727, 2013.

[6] R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules", *Proc. 20th International Conference very large data bases, VLDB*, vol. 1215, 1994.

[7] J. S. Park, M. Chen, and P. S. Yu, *An effective hash-based algorithm for mining association rules ACM*, vol. 24. no. 2, 1995.

[8] M. Chen, J. S. Park, and P. S. Yu. "Efficient data mining for path traversal patterns", *Knowledge and Data Engineering,* vol. 10, no. 2, pp. 209-221, 1998.

[9] Y. Wang, and A. J. T. Lee, "Mining Web navigation patterns with a path traversal graph", *Expert Systems with Applications*, vol. 38, no. 6, pp. 7112-7122, 2011.

[10] P. S. M. Tsai, Y. S. Lee, "The technique for discovering web access patterns and applications", *Minghsin Journal*, vol. 33, no. 7 pp. 239-255, 2007.

[11] M. J. Zaki, "SPADE: An efficient algorithm for mining frequent sequences", *Machine Learning*, vol. 42, no. 1-2, pp. 31-60, 2001.

[12] T. Aye, "Web log cleaning for mining of web usage patterns."*Computer Research and Development (ICCRD), 2011 3rd International Conference on*, IEEE, vol. 2, 2011.

Received: September 22, 2014 Revised: November 30, 2014 Accepted: December 02, 2014