International Journal of Computer Networks and Communications Security

VOL.1, NO.7, DECEMBER 2013, 284–291
Available online at: www.ijcncs.org
ISSN 2308-9830

Improved Data Preparation Technique in Web Usage Mining

Mona S. Kamat[1], Dr. J.W.Bakal[2], Madhu Nashipudi[3]

[1, 3] Information Technology Department, Pillai Institute of Information Technology (PIIT), Panvel, Navi Mumbai, India

[2] Shivajirao S. Jondhale College Of Engineering (SSJCOE), Dombivali, Thane, India

E-mail: [1]monakamat@gmail.com

ABSTRACT

The process to discover and extract useful information is web usage mining. It helps to better understand and achieve the needs of web-based applications. To enhance the efficiency and to ease of the mining process, the data should be preprocessed. The proper study and analysis of web log file is useful to manage the websites effectively for administrative and user's perspective. Many a time the backward referencing used to track the reachability of the pages which consumes time and generates complete path from the root node even if the link has come from some other server page. This problem can be using two-way hash table structure as Access History List. In our system, session identification is done using AHL by considering immediate link analysis, backward referencing without searching the whole tree representing the server pages. Based on this study, it can be concluded that the system is complex but user session sequences are generated with less time and greater precision.

Keywords: *Session identification, Web usage mining, Preprocessing, Backward reachability.*

1 INTRODUCTION

Web Usage Mining is the type of Web Mining activity that involves the automatic discovery of user access patterns from various web log access records. This process consists of three main steps: 1. Preprocessing, 2. Pattern Discovery, 3. Pattern Analysis as shown in Fig 1.

Organizations often generate and collect large volumes of data coming from various sources in different format and follow different conventions. Hence preprocessing is necessary mainly to remove inconsistencies, eliminate duplicates, and extract useful data and to save it in a common format. Therefore, it is also considered very challenging and most important phase in Web usage mining.

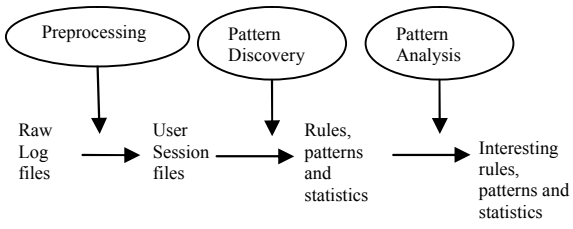

Fig. 1. Web Usage Mining

Pattern Discovery finds various rules and patterns using different web mining techniques and pattern analysis analyses and picks only the interesting patterns and rules for the end-users filtering the useless rules and patterns.

In the proposed system, we basically convert raw data log formats into formatted session files. We use optimal algorithms to generate user session sequences using data structures like Access History List represented by two-way hash table. The system will mainly have the data source as server log and also monitor client side data so as to overcome the problem of cache and proxy servers.

2 LITERATURE SURVAY

Identification of user session boundaries is very important to understand user's request based on their navigation behavior [9]. G.Arumugam and S.Suguna have presented new techniques to generate user session sequences by considering various factors. They have described the data structures used in the propose model and also given algorithms for User identification and session

identification. Session identification algorithm consists of three modules for the activities related to immediate link analysis, updating the AHL and analysis and backward reference and direct reference. Tanasa and Trousse in [10] state an additional step data fusion apart from data cleaning, data structurization and data summarization. Also they describe different heuristics on how to find web robot requests based on browsing speed. Theint Aye in [1] describes the data preprocessing activities like field extraction and data cleaning algorithms are presented.

3 STAGES IN OUR PROPOSED SYSTEM

1. Data collection
2. Parse the log by extracting the fields.
3. Store the data in a relational database
4. Merge data from various sources stored in intermediate files.
5. Data cleaning
6. User identification
7. Session Identification
8. Path completion
9. Data Summarization

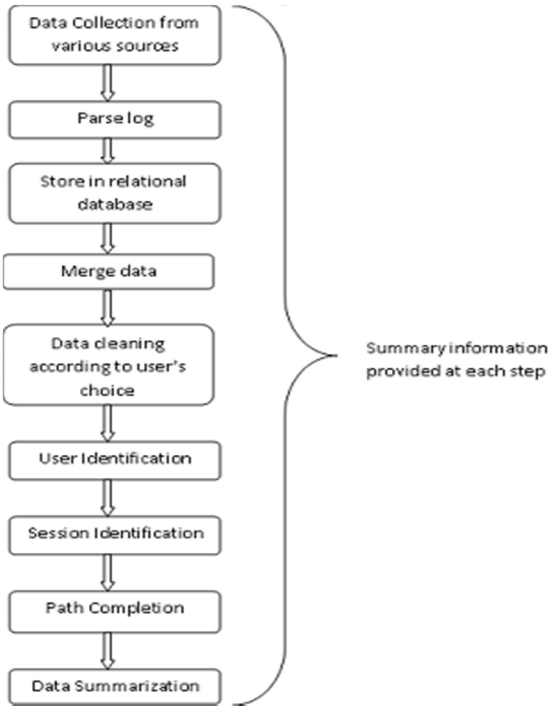

Fig. 2 Proposed System for Preprocessor

4 ACCESS HISTIRY LIST

When a user uses backward reference the User Identification algorithm searches the server pages tree. This backward reference leads to the following problems in [9]: i) Checking of backward reachability between two pages in the tree consumes time ii) Always a path is generated from the root node to the node that leads to backward reference. But the pages that are referred directly from some other server not through the root have not been considered.

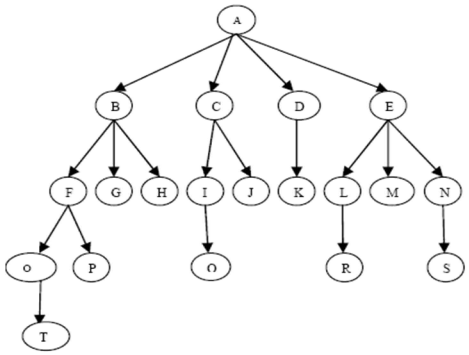

Fig. 3. Tree structure of server pages

To handle these issues a two-way hashed structure called AHL depicted in Fig.4 is introduced.

Case 1: Time reduction in the backward reachability process

The existing algorithms search the unvisited pages also to check the backward reachability traversing from the root to frame a complete path. But in the proposed system, the unvisited pages are not considered. Fig. 3 is used for the discussion.

- Let user0 visit O.html through the page sequence A.html->B.html->F.html->O.html and later refer H.html page.
- The <referrer> of H.html is used to analyze the backward reachability.
- AHL entries of user0 are searched to know the backward reachability and to generate a complete path using the primary index key U0S0 and secondary index Key B.html. Using a single search with AHL one can directly locate the page sequence pointed by B.html.
- Then a complete path is framed by appending a current page with the page sequence pointed by B.html as A.html->B.html->H.html.

- Now, increase the count of access for page sequence A.html->B.html by one and create an entry with U0S1 as primary index key and H.html as secondary index key for the page sequence A.html->B.html->H.html as depicted in Fig.4.

In the existing systems, the nodes F, P, B, G, and H are searched to check the backward reachability between H and O where P and G are unvisited by user0 thus taking more time.

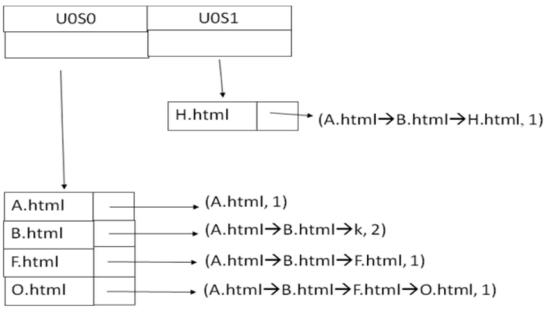

Fig. 4. AHL when the user is at H.html after accessing O.html

Case 2: To analyze the issue of accessing a node not through the root but from some other server

In [9] a complete path is framed from the root, even though the pages are referred from some other server. AHL avoids going through the root when the pages are directly referred from some other page.

- Let the next reference by user0 be L.html, directly from another server page. So, the session should start from L.html.
- AHL is searched using the primary index key U0S0 and U0S1 and secondary index key L.html. Search result is nil.
- Therefore U0S2 starts with L and one can deduce the fact that this reference is not due to backward reference. But the existing processes generate a complete path as A.html->E.html->L.html which is not a correct prediction as the page E.html is not a cached page.

5 ALGORITHMS

A. Parsing / Field Extraction

The process of separating field from the single line of the log file is known as field extraction or parsing the log. The server uses different characters which work as separators and most used separator character is ',' or 'space '.

Algorithm:

1) Start
2) Create a table to store log data
3) Use space or comma as delimiter to separate the fields in the line as per the type of log
4) Seperated fields are stored in variable IP, date, method, referrer, status, byte, agent, url
5) Stop

B. Data Cleaning

The data cleaning module is intended to clean web log data by deleting irrelevant and useless records in order to retain only usage data that can be effectively exploited to recognize users' navigational behavior. Hence we prune following requests: 1. Failed and corrupt requests 2. Requests for multimedia objects 3. Requests from web robots

Algorithm:

1) Start
2) For each log entry
   If request was unsuccessful then move to
anomaly table(check status)
Else If the access method not equals "GET" then move to anomaly table(check method)
Else If the request if for multimedia objects then move to anomaly table(check suffix)
Else If the request is from web robot then move to anomaly table (find browsing speed or
   "robots.txt" in filename)
3) Keep a count of each kind of requests pruned and saved in anomaly table.
4) Stop

C. User Identification

It means identifying each user accessing website and the goal is to mine every user's access characteristic and then make user clustering and provide personalized service to them. It is greatly complicated by the existence of local caches, corporate firewalls, and proxy servers[9]. If login is provided on the website then the user can directly be identified based on the login identification, but in absence of login information, following heuristics are considered.

1) *IP Address:* Each IP address represents one user. Since there might be multiple users with same IP address, only IP address information not reliable.
2) *User Agent:* For more logs, if the IP address is the same, but the agent log shows a change in browser software or operating system, an IP address represents a different user. Hence IP address along with the user agent is matched to identify unique users.

D. Session Identification

Session is the time between the logged in and logged out and finds the sequence of clickstream to trace the user.

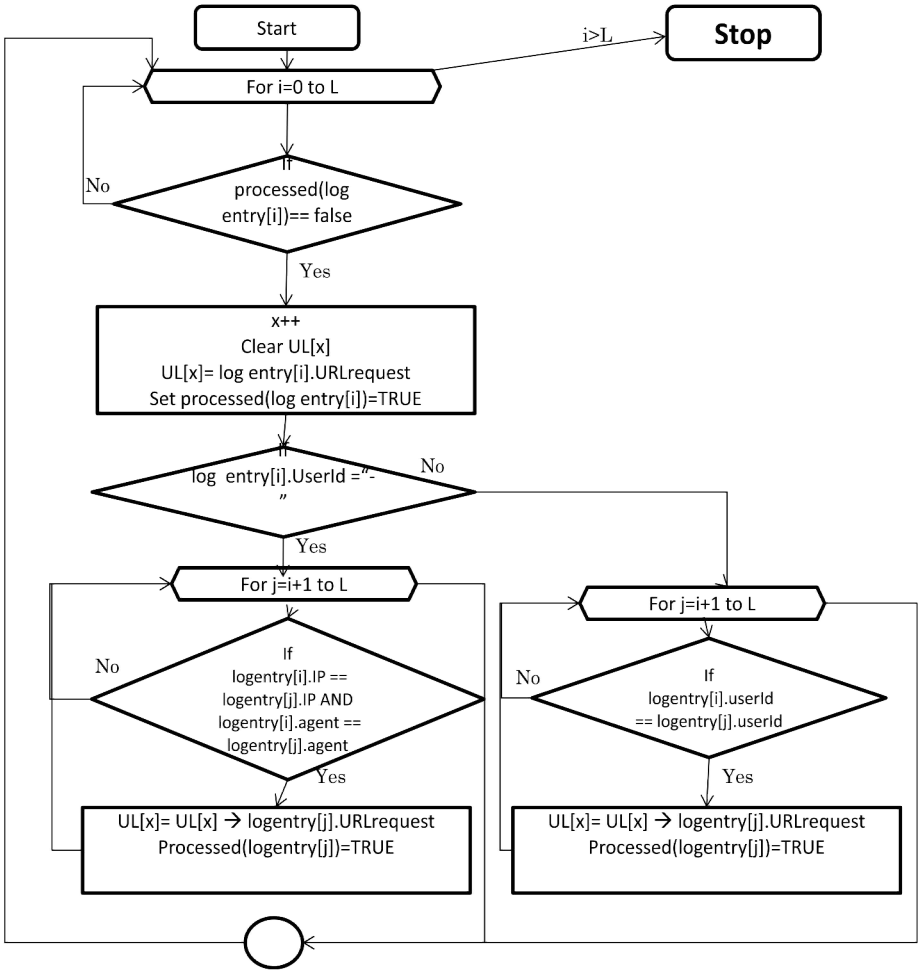

Fig. 5. User Identification Algorithm

Activity. Following rules are briefed to identify user session in the project:

i. If there is a new user, there is a new session.
ii. In one user session, if the refer page is null, there is a new session
iii. If the time between page requests exceeds a limit of 30 minutes (default timeout for session). It is assumed that the user is starting a new session

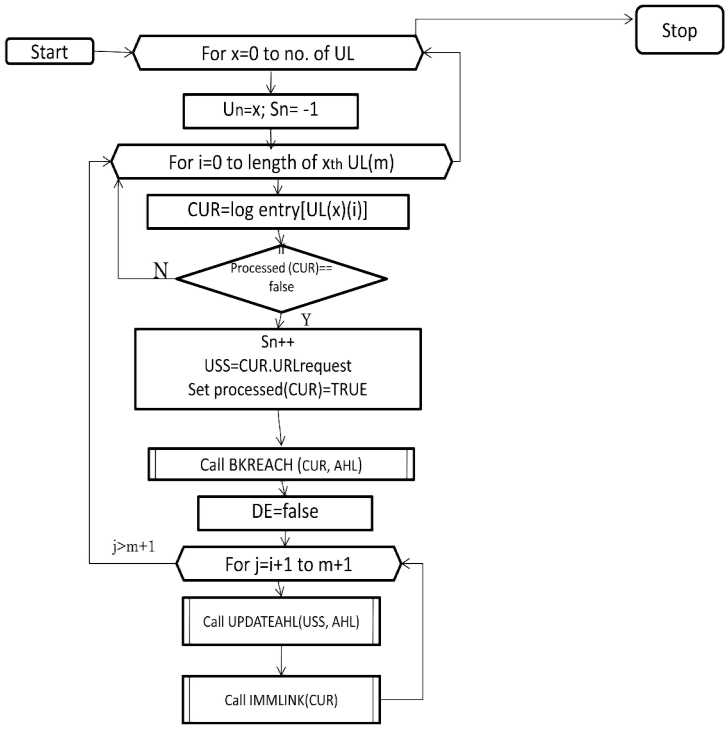

Fig. 6. Session Identification Algorithm

Path Completion: Some of reference information will be lost due to the existence of caching mechanisms. To discover user's travel pattern, the missing pages in the user access path should be appended. For example, if a user goes back to a page A during the same session, the second access to A will likely result in viewing the previously cached version of A and therefore, no request is made to the server. If the hyperlink between the current request page and a page next to the last request does not exist, then the path is incomplete, and needs to be added and should be invoked to check the log. [5]

Session Identification algorithm takes input as User List and the algorithm loops through the user lists. While looping through the items in each List, it does the following:

1) BKREACH()- checks if the current node has backward.

Reference to any node already in AHL as in Fig 7.

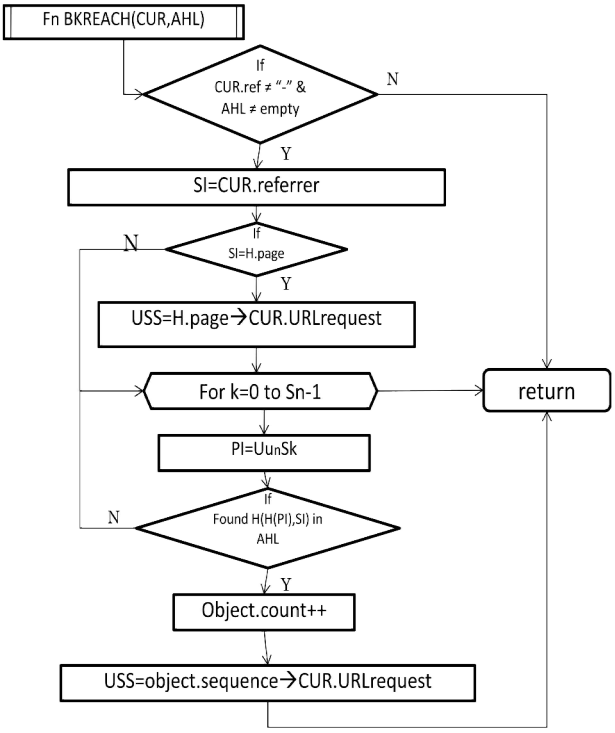

Fig. 7. Flowchart for BKREACH function

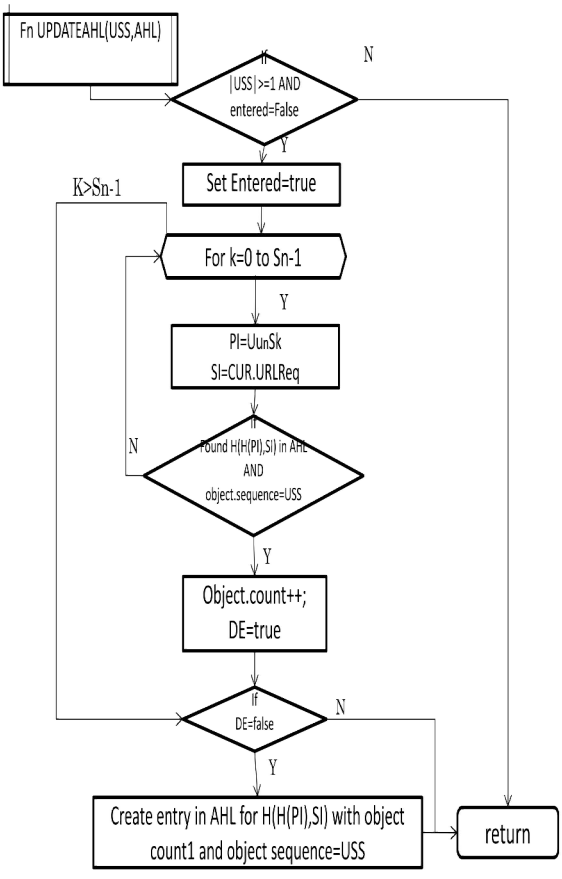

Fig. 8. Flowchart for UPDATEAHL function

2) Loops through the rest of unprocessed items in the list and compares it to the current node.

a. Call function UPDATEAHL---checks if USS exists in AHL and if not add in AHL and accordingly set the count as in Fig 8.

b. Call IMMLINK()—does the immediate link analysis to check if the next item is in the same session as the current node and whether it is reachable from the current node as in Fig 9.

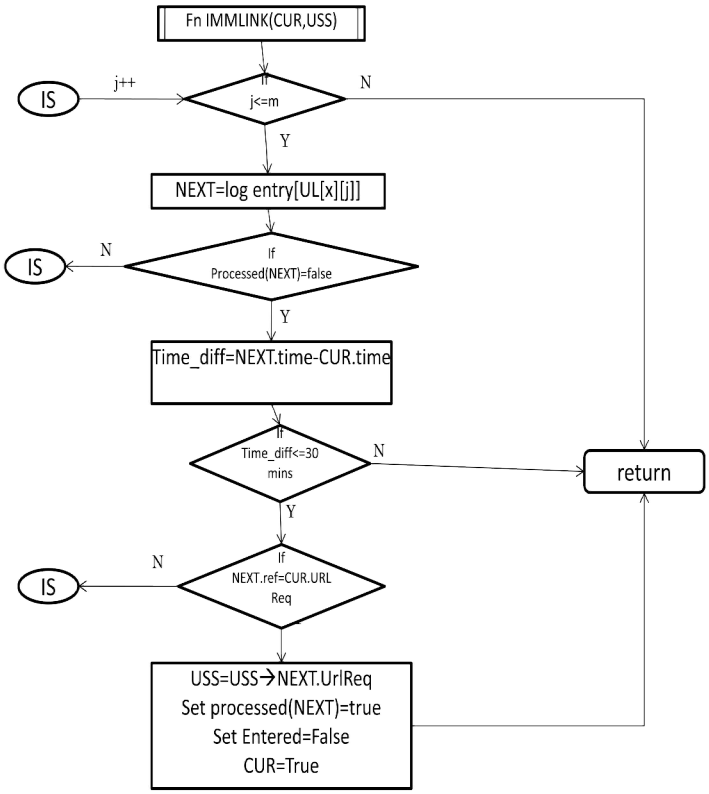

Fig. 9. Flowchart for function IMMLINK

E. Data summarization

This module provides a set of statistics that summarizes the results of the performed preprocessing step. Data summarization generates reports summarizing the information obtained after the application of pre-processing step. This gives a schematic and concise description of the usage data mined from the analyzed log file. Precisely, the created statistical reports provide the necessary information to detect some particular aspects related to the user browsing behavior or to the traffic of the considered site (such as how many images, videos, etc are downloaded; the volume of the requests made; how many requests are generated by robots, etc).

A preliminary summary is generated as soon as the log data are loaded into the pre-processor and contains information about the total number of requests of the analyzed log file, the number of the satisfied requests, the number of failed or corrupt requests, the volume of transferred bytes, etc.

6 CONCLUSION AND FUTURE WORK

The proposed system gives many data preparation techniques to clean the data and identify users and sessions. Also client side data is handled to overcome the problem of caching and existence of proxy servers. The system mainly handles websites where users may not be comfortable revealing their identities by logging in. Hence the system will identify users and then identify sessions using two-way hash structure. Due to the hash structure used in storing user session sequence, backward referencing is done without the entire server pages tree being searched. Hence backward referencing

takes much lesser time and also identifies session with higher precision.

In future the system could be modified so as to recognize if the users in the new batch of log file already exists in the earlier batches of log files processed. The system could be further extended to real time so as to generate a learning graph to predict and prefetch the user's next request.

ACKNOWLEDGMENT

I would like to thank Dr. J.W. Bakal Sir and Madhu Madam for facilitating all the necessary inputs, study material and resources and guiding me with their rich experience. I would especially like to thank Bakal Sir for his unconditional support and confidence in my work.

7 REFERENCES

[1] Theint Aye, "Web cleaning for mining of web usage patterns", International Conference on Computer research and Development(ICCRD), pages 490-494, Vol. 2, May 2011

[2] Asha Khilrani, Prof. Shishir K. Shandilya, "Implementation of User's Browse Log Monitoring Tool for Effective Web Usage Mining", International Journal of Computer Science and Information Technologies 2011, Vol. 2 (3) p. 1061-1064

[3] Sanjay Bapu Thakare, Sangram Gawali, "A Effective and Complete Preprocessing for Web Usage Mining", International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, pp.848-851

[4] G. Arumugam, S.Suguna, "Optimal Algorithms for Generation of User Session Sequences Using Server Side Web User Logs", Network and Service Security, 2009. N2S '09. International Conference, pp. 1 – 6

[5] Sumian Peng, Qingqing Cheng, "Research on Data Preprocessing process in the Web Log Mining", Information Science and Engineering (ICISE), 2009 1st International Conference 2009 pp. 942 – 945

[6] Mehdi Heydari, Raed Ali Helal, Khairil Imran Ghauth, "A Graph-Based Web Usage Mining Method Considering Client Side Data", Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference Vol. 1 pp. 147 – 153

[7] G. Castellano, A. M. Fanelli, M. A. Torsello, "Log Data Preparation For Mining Web Usage Patterns", IADIS International Conference Applied Computing 2007, pp. 371-378

[8] R.M. Suresh, R. Padmajavalli, "An Overview of Data Preprocessing in Data and Web Usage Mining", IEEE 2006, pp.193-198

[9] Li Chaofeng, Research and development of data preprocessing in Web Usage Mining, Journal of south-central university for nationalities, 2005(4):82-85

[10] Tanasa, D., Trousse, B. , "Advanced Data Preprocessing for Intersites Web Usage Mining", Intelligent Systems, IEEE, Mar-Apr 2004,Vol 19, Issue: 2, pp.59 - 65

[11] J.Srivastava, R. Cooley "Web Usage Mining : Discovery and Application of Usage Patterns from Web Data", SIGKDD Explorations. 2000 Vol 1(2):1223-1247

[12] R.Cooley, B. Mobasher. J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns", Journal of Knowledge and Information System, 1999.