An Efficient Algorithm for Mining Web Navigation Patterns with a Path Traversal Graph

Ms. N. G. Sharma[#], S. R. Lomte[*]

[#]*Computer Engineering Department, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad*

ngsharma.comp@gmail.com
santoshlomte@hotmail.com

Abstract— With the expansion of e-commerce and mobile-based commerce, the role of web user on World Wide Web has become pivotal enough to warrant studies to further understand the user's intent, navigation patterns on websites and usage needs. Using web logs on the servers hosting websites, site owners and in turn companies, can extract information to better understand and predict user's needs, tailoring their sites to meet such needs.
The former mining algorithms do not provide a clear picture of the intentions of the visitors and suffer from drawback of either repetitive database scan or high memory load.
This paper uses the concept of throughout-surfing patterns(TSPs) and proposes an efficient algorithm for mining the patterns, that effectively predict and display the trends toward the next visited Web pages in a browsing session with a view to better understand the purposes of website visitors. It also uses a compact graph structure, termed a path traversal graph, to record information about the navigation paths of website visitors, required for mining TSPs. In addition, it proposes a new algorithm for graph traversal based on the prior graph structure to discover the TSPs. The experimental results show the proposed algorithm is highly efficient to discover TSPs, by improving the accuracy, reducing the execution time and memory requirements with a single scan on the database & avoiding generation of candidate sequence as like apriori.

Keywords— Web Log Mining, Path traversal graph, Throughout-surfing pattern, Browsing behavior, web Traversal pattern

I. INTRODUCTION

Initially, the world wide web and in-turn, the websites were only created with site owners interest; users' perspective and needs were not considered. Due to constant evolution of commerce from B2B to B2C, users became more important and centrifugal than companies serving or hosting the website. This relative shift from owner-centric to user-centric design has played an important role in improving the access efficiency of web pages by adaptive website system, dynamic re-organization of website, identification of target group of visitors, improving the performance of web search and prediction of user intent in web systems. Hence, the various techniques/strategies of web usage mining were developed to better understand user's intent, preference and interests. Mining Web navigation patterns is useful in practice, and the extracted patterns can be used to predict and understand visitors' browsing behaviour and intentions. It is helpful in improving user experience, website configuration, and the efficiency and effectiveness of e-commerce [1]. These patterns can be used by Website operators to analyze and predict user motivation so as to provide better recommendations and personalized services to their customers (Arotariteia & Mitra, 2004; El-Ramly and Stroulia; 2004; Pierrakos, Paliouras, Papatheodorou, & Spyropoulos, 2003; Schafer, Konstan, & Riedl, 2001).

A Web navigation pattern referred to as a Web access pattern (also known as clickstream) is a path through one or more Web pages in a website that is extracted from the access logs of the Web server. A series of Web pages in a website requested by a visitor in a single visit is referred to as a session. The process of discovering patterns from access logs is known as Web usage mining or Web log mining (Pei, Han, Mortazavi-Asl, & Zhu, 2000). Given a set of sessions, the support of a Web access pattern is defined as the ratio of the sessions containing the pattern to all sessions.

The former mining algorithms on weblogs suffer from drawback of either repetitive database scan or high memory load. For algorithms with a single database scan, special data structures are required to store the sequences in the database. However, it may be difficult to hold all sequences of the database in the data structure if it is too long.

As mentioned in [1] , mining Web navigation patterns with a path traversal graph does not generate the candidate patterns and it scans the database only once. This approach was applied to discover the throughout-surfing patterns(TSP's). In this paper, proposes an efficient mining algorithm to discover the throughout-surfing patterns. First, a compact structure is devised called the path traversal graph to portray the tracks of Web navigation. Second, an efficient algorithm for graph traversal is designed to discover the throughout-surfing patterns. The contributions of this paper are described as follows.

- The concept of TSP is used for understanding the purposes of website visitors.
- A compact graph is devised to store the information of Web navigation paths. The information of Web browsing and hyperlinks between Web pages are kept in the graph. The edges in the path traversal graph record both incoming and outgoing hyperlinks and the via-links hold "from-to-via" information in the graph that are necessary to predict where a visitor will go at any vertex by the vertex he comes from.
- This system defines a recursive algorithm(graph traverse) to find TSPs efficiently. A depth-first search (DFS) mechanism is adopted to traverse the path traversal graph.

The rest of this paper is organized as follows. Section 2 describes the related work on Web usage mining. Section 3 displays the structure of the proposed algorithm for path traversal graph construction. Also, a graph traverse algorithm based on the path traversal graph is introduced in this section. In Section 4, gives experimental results and Section 5 gives performance evaluations. Finally, conclusions are made in Section 6.

II. RELATED WORK

There are numerous studies on the navigation behaviour of website visitors. Most are conducted by the techniques of mining Web access patterns, such as improving the access efficiency of Web pages by the adaptive website system, reorganizing a website dynamically, identifying the target group of Web visitors,

strengthening the performance of Web searches, and predicting user behaviour patterns in mobile Web systems[1]. Research efforts to discover Web access patterns focus on three main paradigms (Fac-ca & Lanzi, 2005): association rules, sequential patterns, and clustering. Some well-known algorithms for mining association rules have been modified to extract sequential patterns, for instance used AprioriAll and GSP, two extensions of the Apriori algorithm for association rules mining, argues that algorithms for association rule mining (e.g., Apriori) are not efficient when applied to long sequential patterns, which is an important drawback when working with Web logs. Accordingly, [J. Pei, J. Han, B. Mortazavi-asl, H. Zhu] proposes an alternative algorithm in which tree structures (WAP-tree) are used to represent navigation patterns. The algorithm (WAP-mine) and the data structure (WAP-tree), specifically tailored for mining Web access patterns, WAP-mine outperforms other Apriori-like algorithms like GSP. Tree structures are also used . The mining method using the WAP-tree alleviates both problems of scanning the database repeatedly and generating tremendous candidate sequences. However, to use the conditional search strategies in WAP-tree-based mining algorithms, it requires reconstructing a large number of intermediate conditional WAP-trees during mining processes, which is rather costly (Zhou et al., 2006). In addition, the serious problem of performance degradation arises when the capacity of the main memory cannot hold the entire structure of the WAP-tree.

Tao, Hong, and Su (2007) in paper "Web usage mining with intentional browsing data"addressed another interesting topic of Web usage mining with intentional browsing data (IBD). IBD is a category of on-line browsing actions, such as ''copy'', ''scroll'', or ''save as,'' which is not recorded in Web log files. To make IBD available like Web log files, they proposed an on-line data collection mechanism for capturing IBD.`

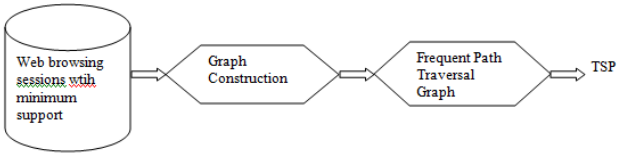
Algorithms for mining sequential patterns are common in Web navigation pattern mining (Agrawal & Srikant, 1995; Lee & Wang, 2003; Lin & Lee, 2005; Masseglia, Poncelet, & Teisseire, 2009; Pei et al., 2004). Apriori algorithm (Agrawal & Srikant, 1995) in paper "Mining Sequential Patterns" introduces a level-wise iterative search to discover all maximal frequent sequential patterns. The concept of mining frequent closed sequences is derived (Wang, Han, & Li, 2007; Yan, Han, & Afshar, 2003). The frequent closed sequences are regarded as a pattern closure of all frequent sequential patterns. This proves more efficient to discern the pattern closure instead of all frequent patterns; however, it consumes a lot of memory and leads to a huge search space for pattern closure checking. Yen and Chen (2001) adopted a graph-based approach to mine both association rules and sequential patterns. As mentioned in their conclusions, the graph structure may not fit in the main memory when the database is very large. In general, the database is huge and the algorithm is inadequate. Li, Lee, and Shan (2006) presented an incremental mining algorithm, termed DSM-PLW, to find the maximal reference sequences in one database scan.

The process of mining patterns is proceeding on the SP-forest in the main memory. The space upper bound of O(2k) where k is the number of frequent references will obscure the method as k is greater than 30. Lee and Yen (2007) used the lattice structure to store the previous mining results for incremental Web traversal patterns. The patterns may be obtained rapidly when the database or the website structure is updated. Again, as stated in their conclusions, the size of the lattice structure may become too large to be loaded into the main memory.

The former mining algorithms suffer from either repetitive database scan or high memory load. For algorithms with a single database scan, they build special data structures to store the sequences in the database. However, it is impracticable to hold all sequences of the database in the data structure. On the contrary, proposed scheme for mining TSPs is realistic, in which the memory is loaded with the hyperlink structure of the website instead of the sequence database.

III. PROPOSED SYSTEM

Proposed system is based on graph data structure for storing and retrieving the session information. In the proposed system information from user login or surfing session from web server is collected. Flat file system data schema is used to store this session like csv or txt file system. Proposed system is mainly divide into two parts the first being graph construction and second is graph traversal. Given below is the architectural diagram of the approach for mining TSP.



In the first part the user logs are extracted from the data file and filtering process is applied so as to remove unwanted and duplicate data. Result of filtering module contain user navigation patterns in the form of hyperlinks. Each hyperlink is treated as node or vertex. By using this session information a graph structure is plot. Consider G is graph of session (s1,s2,s3,...sn) then each s1, s2 are treated as vertex of graph. To build this graph first the root node is determined, after that link between each node is found and these links are treated as edges of graph. Pattern matching algorithm is used to parse the data. After forming of edges via links are found and using these via link session surfing graph is plotted. Fig.1 show the user login session and their navigation patterns. And Fig 2 show the constructed graph using these links. Following are the steps of graph construction:

1) Retrieve session values form the web server.
2) Filter duplicate and unwanted data.
3) Sort the session according to the website pages
4) Find session home page as a root node of graph
5) Identify vertex and edges
6) Identify direct links and via links
7) Join this direct link and via links.

session ID	Web Browseing path
s1	(1,3,5,6,9,11)
s2	(1,2,5,6,8,11)
s3	(1,2,5,7,9,11)
s4	(1,4,5,6,9,10)

Fig. 1 User login session and their navigation patterns

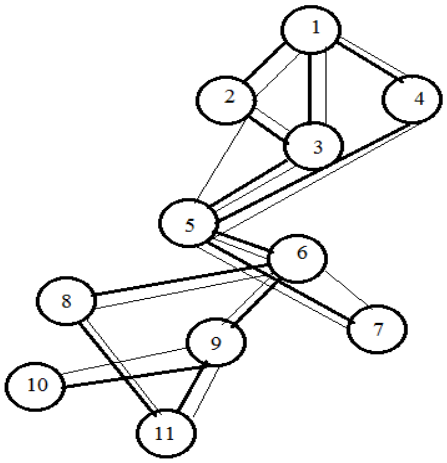


Fig. 2 Solid line shows direct links and thin line shows via links

Second step of proposed system contain path traversal and calculate minimum support for finding frequent path. BSF method is used for traversing the constructed graph. Following are the steps of path traversal.

Steps of the path traversal:

1) Select root node and traverse decedent vertex.
2) Get the decedent vertex of root node .
3) Dynamically calculate the minimum graph support to find frequent paths using distance finding methods.
4) Select the node of each via links having maximum value then the support vector.
5) By using this select node draw frequent path
6) Replete graph construction method
7) Repeat 3rd step
8) Calculate via links using support values
9) Used recursive tsp() method to calculate tsp of graph

Following code show the tsp function to calculate throughout surfing patterns.

```
Tsp()
{
 for (start = 0; start <rootlinks; start++)
      {

        for (nodes= 0; nodes <totlnodes.size(); nodes++)
          {
              string []vailinks = totalnodes[j];

                 for (int x = 0; x < vailinks.Length; x++)
                   {
                     string  vailink = vailinks[x].Replace("(", "");
                     vailink = vailink.Replace(",", "");
                     int index = vailink.IndexOf(last);
                     if (vailink.Trim().Contains(last) && (index+1) != vailink.Length
                     && vailink.Contains (rootnode)==false)
                     {
                       string decendent = arr[i];
                       decendent = decendent + vailink.Substring(index + 1, 1);
                       result[i] = decendent;
                     }
                   }
          }

      }

          if (secarrat.Length != 0)
             tsp(secarrat);

 }
```

IV. EXPERIMENTAL RESULTS

Consider a data set consisting of 30 Web browsing sessions as shown in Table 1 and the corresponding Web structure is depicted in Fig 3. The web pages are named as numbers as 1,2,3,4,..... Fig 4 show the corresponding initial path traversal graph and the frequent path traversal graph respectively. Duplicate and unwanted sessions are eliminated for simplicity. Then the vertex and edges are determined and their via links are calculated. Based on minimum support vector calculated using distance finding method, the nodes having values more than minimum support are selected. Then an initial frequent path graph is constructed.

Table 1
The data set of 30 Web browsing sessions.

Session ID	Web browsing session	Session ID	Web browsing session
S001	<1, 2>	S016	<3, 7,10>
S002	<1, 3,4>	S017	<3, 7,12,17>
S003	<1, 3,4,10>	S018	<3, 7,12,16>
S004	<1, 3,7,10,14,19>	S019	<3,7,12,16,21>
S005	<1, 3,7,11>	S020	<4,10,14,19>
S006	<1, 3,7,12>	S021	<7, 10,14,19>
S007	<1,3,7,12,16,20,7,12>	S022	<7,12,16,14,19>
S008	<1, 3,7,13>	S023	<10,14,18>
S009	<1, 4,8>	S024	<12,16,14,19>
S010	<1, 4,9>	S025	<12,16,20,7>
S011	<1,4,10,15>	S026	<20,7,12>
S012	<1, 4,10,14,19>	S027	<23,24,25,26>
S013	<1, 5>	S028	<25,26,23,24>
S014	<1, 6>	S029	<24,25,26,23,24>
S015	<3, 4,10>	S030	<23,24,25>

Vertex	Via-link
3	<1,3,4> <1,3,7>
4	<1,4,10> <3,4,10>
7	<3,7,10> <3,7,12> <20,7,12>
10	<4,10,14> <7,10,14>
12	<7,12,16>
14	<10,14,19> <16,14,19>
16	<12,16,14> <12,16,20>
20	<16,20,7>
23	<26,23,24>
24	<23,24,25>
25	<24,25,26>
26	<25,26,23>

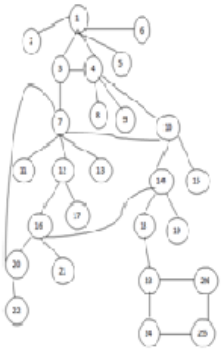


Fig. 3 Corresponding Web Structure

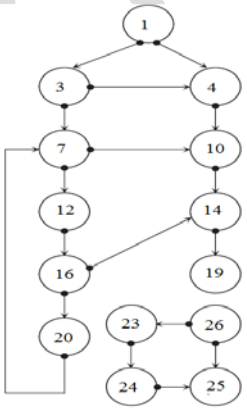


Fig. 4 Corresponding Frequent Path Traversal Graph

P1	<1 3 4 10 14 19>
P2	<1 3 7 10 14 19>
P3	<1 3 7 12 16 14 19>
P4	<1 3 7 12 16 20 7>
P5	<1 4 10 14 19>
P6	<23 24 25 26 23 24>

Fig. 5 Final output of the system as a TSP

Following table shows the via links for each vertex which are calculated from the above fig of frequent path traversal graph. It can be seen that vertex 23,24,25,26 cannot be reached from the root vertex or from any other vertex. So these vertices have been excluded from rest of the connected graph. Fig 5 contains the final output of the system as a TSP.

V. PERFORMANCE EVALUATION

This section, deals with a comparison of our proposed algorithm with Apriori and Graph Traverse algorithms.

On different settings of minimum support thresholds, the experimental results are shown in below Fig 6. While the minimum support becomes lower, the number of frequent patterns increases and the frequent patterns get longer. In Fig. 6, the execution time of the proposed mining approach remains almost at the same level because it discovers the TSP by traversing the path traversal graph and almost all the run time is spent on constructing the path traversal graph.

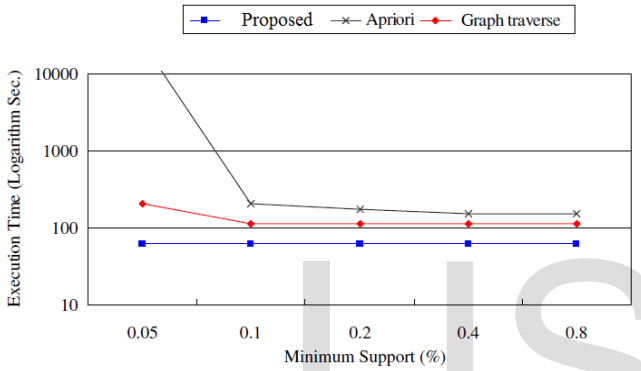


Fig. 6 Execution time for various settings of minimum support thresholds

Fig. 7 and 8 illustrate the scalability of the algorithms by varying the mean length and the number of Web browsing sessions respectively. The execution time is expected to be proportional to the mean length as well as the number of Web browsing sessions. As the length of the Web browsing sessions or the size of the data set increases, the cost of scanning the data set also raises. The experimental results confirm to the expectation.

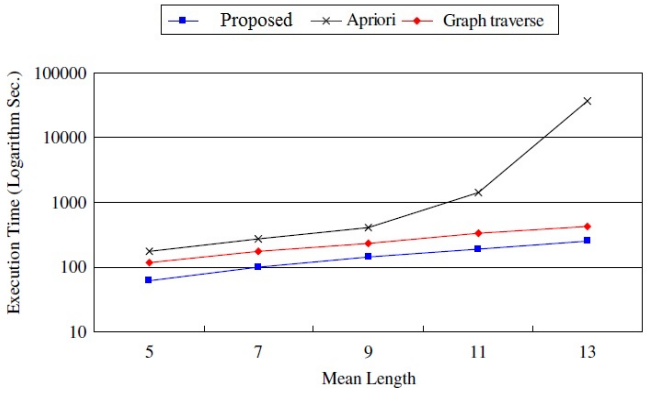


Fig. 7 Execution time for varying mean length of Web Browsing Sessions

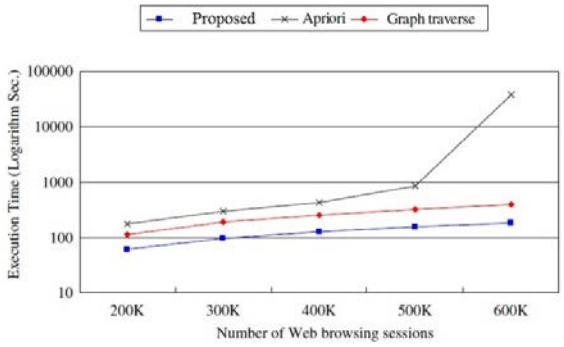


Fig. 8 Execution time for different size of data sets

In Fig. 9, the number of fan-outs is varied from eight to thirteen. The number of via-links associated with a vertex is proportional to the number of fan-outs. Therefore, the execution time grows with a larger number of fan-outs. Because the Graph Traverse algorithm must examine all combinations of large sequences obtained in the (k -1)th iteration to produce candidates of length k, its execution time grows exponentially. The number of fan-outs has a great effect on the number and the size of the projection databases. As the number of fan-outs increases, both the number and the size of the projection databases increase. Therefore, the Apriori algorithm does not perform well as the number of fan-outs increases.

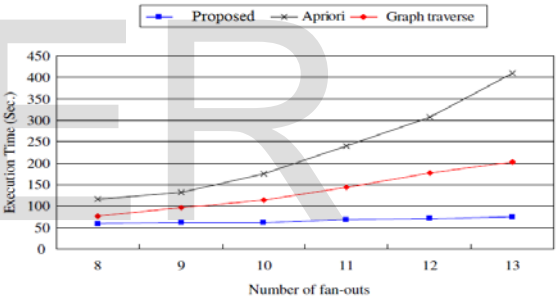


Fig. 9 Execution time for various number of fan-outs

Fig. 10 presents the results of experiments conducted on the real data. As the minimum support varies from 0.14% to 0.01%, the execution time of our method grows slightly from 25 to 27 seconds, which confirms to the results shown in Fig. 6.

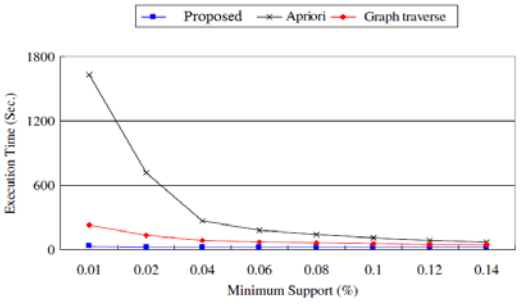


Fig. 10 Execution time on real data

Table 2 exhibits the numbers of TSPs and WTPs (TSP count and WTP count respectively) as well as the precision and recall measures gathered from the experiments on 750 K real data. Precision is defined as the ratio of mined Web traversal patterns to all TSP. Recall is defined as the ratio of mined Web traversal patterns to the Web traversal pat-terns contained in the data set. Both equations of the precision and recall are listed below.

Precision = number of WTP in TSP/number of TSP (1)

Recall = number of WTP in TSP/number of WTP in data sets

(2)

The ‘‘Missed’’ column records the number of WTPs that are neither found nor contained in TSP. The ‘‘Matched’’ column presents the number of Web traversal patterns that match throughout-surfing patterns. The ‘‘Contained’’ column shows the number of Web traversal patterns that are contained in some TSPs. As shown in Table 2, the discovered TSPs hold all Web traversal patterns, that is, all the Web traversal patterns are contained in the TSP, and hence the values in the column of ‘‘Missed’’ are all zero. Theorem 1 is verified by the columns of ‘‘Matched,’’ ‘‘Contained,’’ and ‘‘Missed.’’ While the minimum support is considerably low, the precision and recall are degraded. Nevertheless, it is unrealistic to set the minimum support at such low levels to generate a large number of patterns. On the other hand, the higher minimum sup-port setting acquires high degrees of precision and recall measures.

Min_sup(%)	TSP count	WTP count	Matched	Contained	Missed	Precision	Recall
5%	6	30	4	27	0	0.666	0.133
10 %	4	11	2	8	0	0.5	0.181
15 %	3	7	2	6	0	0.66	0.285

VI. CONCLUSION

In this paper, the problem of mining Web navi-gation patterns is investigated. Two primary issues involved in mining Web navigation patterns are the effectiveness and the efficiency of the mining approaches. First, concept of through-out-surfing patterns is used, which are effective to predict website visitor's surfing paths and destinations. Second, a path traversal graph and graph traverse algorithm is devised to increase the efficiency of mining throughout-surfing patterns. The research results show that throughout-surfing patterns are more effective for content management and they are applicable to providing surfing paths recommendation and personalized configuration of dynamic websites.

In addition, a path traversal graph structure is suitable for incremental mining of sequential patterns. The compact graph structure retained in the main memory may be output to permanent storage. While mining patterns from the database with new added data, the path traversal graph is restored in the main memory and the new data is retrieved and appended to the graph. Then, the processes for mining TSPs are performed in the graph. The proposed mining algorithm can be extended for mining TSP from incremental databases in the future study. Moreover, TSP only features consecutive click sequences. It is another interesting issue to mine non consecutive browsing patterns.

The algorithms proposed in this study will be advanced to discover the discontinuous browsing patterns in a website.

REFERENCES

[1] Yao-Te Wang, Anthony J.T. Lee "Mining Web navigation patterns with a path traversal graph" Expert Systems with Applications: An International Journal Volume 38 Issue 6, June, 2011

[2] Facca, F. M., & Lanzi, P. L. (2005). Mining interesting knowledge from Weblogs: A survey. Data and Knowledge Engineering, 53, 225–241.

[3] Lee, Y.-S., & Yen, S.-J. (2007). Incremental and interactive mining of Web traversal patterns. Information Sciences, 178(2), 278–306.

[4] Li, H.-F., Lee, S.-Y., & Shan, M.-K. (2006). DSM-PLW: Single-pass mining of path traversal patterns over streaming Web click-sequences. Computer Networks, 50,1474–1487.

[5] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., et al. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. IEEE Transactions on Knowledge and Data Engineering, 16(10), 1–17.

[6] Tseng, V.-S., & Lin, K.-W. (2006). Efficient mining and prediction of user behavior patterns in mobile Web systems. Information and Software Technology, 48,357–369.

[7] Wang, J., Han, J., & Li, C. (2007). Frequent closed sequence mining without candidate maintenance. IEEE Transactions on Knowledge and Data Engineering, 19(8), 1042–1056.

[8] Xing, D., & Shen, J. (2004). Efficient data mining for Web navigation patterns.Information and Software Technology, 46, 55–63.

[9] Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining closed sequential patterns in large datasets. In Third SIAM international conference on data mining (SDM), SanFrancisco. CA (pp. 166–177).

[10] Yen, S.-J., & Chen, A. L. P. (2001). A graph-based approach for discovering various types of association rules. IEEE Transactions on Knowledge and Data Engineering, 13(5), 839–845.

[11] J. Pei, J. Han, B. Mortazavi-asl, H. Zhu, Mining access patterns efficiently from web logs, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 396–407.

[12] X. Huang, N. Cercone, A. An, Comparison of interestingness functions for learning web usage patterns, in:Proceedings of the Eleventh International Conference on Information and Knowledge Management, ACM Press,2002, pp. 617–620.\

[13] B. Mortazavi-Asl, Discovering and mining user web-page traversal patterns, Master_s thesis, Simon Fraser University, 2001

[14] E.S. Nan Niu, M. El-Ramly, Understanding web usage for dynamic web-site adaptation: A case study, in: Proceedings of the Fourth International Workshop on Web Site Evolution (WSE_02), IEEE, 2002, pp. 53–64.

[15] Bo Wu, Defu Zhang, Qihua Lan, Jiemin Zheng An Efficient Frequent Patterns Mining Algorithm based on Apriori Algorithm and the FP-tree Structure Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on (Volume:1)

[16] Suchita A.Chavan " Different Approaches of Mining Web Navigation Pattern: Survey" International Journal of Computer Applications (0975 – 8887) International Conference on Recent Trends in engineering & Technology - 2013(ICRTET'2013)

[17] Rakesh Agrawal and Ramakrishnan Srikant "Mining Sequential Patterns" IBM research paper IEEE.

[18] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 11, NOVEMBER 2004

[19] K. Balog P. Hofgesang W. Kowalczyk "Modeling Navigation Patterns of Visitors of Unstructured Websites" Research and Development in Intelligent Systems XXII Proceedings of AI-2005, the Twenty-fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, December 2005

[20] Yu-Hui Tao, Tzung-Pie Hong, Yu-Ming-Su "Web Usage mining with intentional browsing data" Expert Systems with Applications: An International Journal Volume 34 Issue 3, April 2008.

[21] Huxing Zhang _, Gang Wu _, Kingsum Chow y, Zhidong Yu y, and XueZhi Xing "Detecting Resource Leaks through Dynamical Mining of Resource Usage Patterns" DSNW '11 Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops.

[22] E.S. Nan Niu, M. El-Ramly, Understanding web usage for dynamic web-site adaptation: A case study, in: Proceedings of the Fourth International Workshop on Web Site Evolution (WSE_02), IEEE, 2002, pp. 53 – 64.

[23] Configuration file of W3C httpd, http://www.w3.org/Daemon/User/Config/ (1995).

[24] W3C Extended Log File Format, http://www.w3.org/TR/WD-logfile.html (1996).

[25] D.M. Kristol, Http cookies: standards, privacy, and politics, ACM Transactions on Internet Technology (TOIT) 1 (2) (2001) 151 – 198.

[26] Pilot Software, Web site analysis, Going Beyond Traffic Analysis http://www.marketwave.com/productssolutions/hitlist.html (2002).