A new outlier detection approach to discover low hit web pages using sequential frequent pattern mining to improve website's design

S. Vasuki[1] Ph.D., Research Scholar, Asst.professor, Department of Computer Applications, J.J. College of Arts and Science, Pudukkottai, Tamil Nadu, India.

Dr. K. Subramanian[2] Research Guide , Asst.professor, Department of Computer Science, Govt. Arts and Science, kulithalai, Tamil Nadu, India.

ABSTRACT

The Internet offers huge volume of data to the users and grows rapidly every day. The web server creates log files regarding details about the page, IP address of the user, browser, and operating system used and time/date stamp regarding browsing patterns and this data is mined to extract useful information using web usage mining. The primary objective of this paper is to find the low hit pages of a website from the log files using finding outliers in sequential mining concept. To cater to the need of this objective, a new algorithm named "Detect Anomaly in Sequential Pattern Algorithm (DASPAT)" is proposed. The proposed algorithm creates candidates using Apriori like approach and discovers the unusual browsing behavior of the users, and the detected UBB are treated as outliers. This paper introduces a new approach to find the low hit web pages in tandem to enable the designers to understand how the user browses the site and allow them to redesign the web site.

Keywords

Outliers, Sequential mining, Frequent patterns, infrequent patterns, browsing behavior

1. INTRODUCTION

Web usage mining is the best technique for analyzing a user's browsing behavior in Clickstream data. The discovered browsing patterns not only be used to understand how a user navigates through a website, but also help to provide a better service to the user, to create an adaptive website and website personalization. Since website design is the most important success factor for a website, especially in E-commerce, the analysis of browsing behavior to discover usual and unusual browsing behavior plays a pivotal role in website design .Majority of the research work related to web usage mining techniques to find user's browsing behavior is based on the direct method which processes the weblog data directly to find either an interesting pattern or uninteresting patterns. Still different patterns of user's browsing behavior discovered can have distinctive meanings in different websites. An interesting pattern in a website may not be interesting in another. In this paper, we propose a new web usage mining approach to detect unusual browsing behaviors of the user to solve this glitch.

The proposed approach is useful for website designers to gauge how a user browses their website, especially for those designers who are keen to redesign their website. The rationale behind this approach of web usage mining is that the designer of the site must be in a position to define patterns of usual browsing behavior, and then by using this pattern, the designer should be able to discover any unexpected deviations.

The notion behind this is that the overall design concept of the site is best understood by the designer of the site and the designer is the best person to define a usual navigational pattern. Using this predefined patterns and the proposed DASPAT algorithm, browsing patterns that do not match the predefined are identified as patterns of unusual browsing behavior. The website designer can then use these discovered data to locate the weaker section of the site with low hits and act accordingly.

1.1 WEB MINING

Web mining is the application of data mining to the web data and traces user's visiting behaviors and extracts their interests using patterns. Since this area is applicable in e-commerce and Web analytics directly, web mining has become one of the important areas in computer science. Web Usage Mining uses mining methods in log data to extract the behavior of users which is used in various applications like personalized services, adaptive web sites, customer profiling, creating attractive web sites.

Web mining is classified into three areas, namely wen content mining, web usage mining and web structure mining.

1.2 Web Content

This is the evident data in the Web pages or the information which was meant to be displayed to the users. A major part of this data will consist of textual data and images.

1.3 Web Structure

The Data which describes the organization of the website, it is divided into two types. Intra-page structure information includes the arrangement of various HTML or XML tags within a given page.

1.4 Web Usage

The Data that describes the usage patterns of Web pages, such as IP addresses, page references, and the date and time of accesses and various other information's depending on the log format file.

Similar to all data mining task, the process of Web usage mining also comprises of three major steps (i) data pre-processing, (ii) pattern extraction and (iii) analysis. The input log data has to be pre-processed in order to have the appropriate input for the mining algorithms. The different methods need different input formats, thus the pre-processing phase can provide input data to various methodologies adapted. Pattern extraction is a process of discovering frequent or infrequent patterns from the weblog data related to the pages visited by the users.

Web usage mining research focuses on extracting patterns of browsing behavior of the users visiting a website. These patterns of browsing behavior are valuable to the site as they deliver accurate answers to questions like, how effective are the website in information delivery? Is the structure of the website user-friendly? Can we predict user's next page in the website? Can we increase the overall satisfaction of the users? Can we personalize the web content to attract specific groups frequently? Almost all the answer to these questions may come from the analysis of the data from log files stored in web servers.

1.5 OUTLIERS

An outlier is a data which is notably unusual from the remaining data. Hawkins formally defined the concept of an outlier as follows: *"An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism."*

Outliers sometimes referred to as *abnormalities*, *discordant*, *deviants*, or *anomalies* in the data mining vertical. The data created in many applications using one or more process could either reflect process activity or observations collected about entities. When the data creating process behaves in an unusual way, it results in the creation of outliers. Therefore, an outlier often contains useful information about unusual characteristics of the systems and entities, which impact the data generation process.

1.6 DATASET USED IN WEB MINING

Different kinds of data are being used for web usage mining and the dataset types are illustrated in Table1.1.

1.7 Web Server Logs

These are logs which maintain entire history of page requests given by the user at the server side. T

W3C maintains a standard format for web server log files [8], but other proprietary formats exist. More recent entries are typically appended to the end of the file. Information about the request, including client IP address, request data/time, page requested, HTTP code, bytes served, user agent, and referrer are typically added. These data can be combined into a single file, or separated into distinct logs, such as an access log, error log, or referrer log.

1.8 Proxy Server Logs

A Web proxy is a cache mechanism which interacts between client browsers and web server. It helps to decrease the load time of web pages and the network traffic load.

1.9 Browser Logs

The data which are collected at the browser client side after modifying the browsers or by using JavaScript and Java applets.

1.11 Typical web log data

A typical weblog file consists of many fields like IP address or hostname, User Agent, Referring URL, Method, Protocol, Path, Agent, Date, and Time. The web log file is preprocessed in such a way that it is ready to be used in the algorithm to fetch useful patterns. The usual web log file is shown in table 1 and this web log file is transformed into normalized weblog file as shown in table 2. The data is converted into records based on the IP address to identify the users browsing or navigational pattern. This normalized weblog file is fed as input to the proposed algorithm to test its workability.

Table 1.1: Sample Weblog file

IP	Method	Protocol	Page	Agent	OS	Date Time
192.157.55.78	GET	HTTP 1.1	Page 1	Opera	Win7	22.07.15:10.12.23
192.157.55.78	GET	HTTP 1.1	Page 3	Opera	Win7	22.07.15:10.13.51
192.157.55.78	GET	HTTP 1.1	Page 5	Opera	Win7	22.07.15:10.18.28
192.157.55.78	GET	HTTP 1.1	Page 9	Opera	Win7	22.07.15:10.21.37
122.87.23.61	GET	HTTP 1.1	Page 1	CHROME	Win7	22.07.15:11.102004
122.87.23.61	GET	HTTP 1.1	Page 3	CHROME	Win7	22.07.15:11.13.39
122.87.23.61	GET	HTTP 1.1	Page 4	CHROME	Win7	22.07.15:11.18.37
192.22.58.23	GET	HTTP 1.1	Page 1	IE	Win7	22.07.15:20.11.39
192.22.58.23	GET	HTTP 1.1	Page 6	IE	Win7	22.07.15:20.15.31
----	----	----	----	----	----	----:----

Table 1:2: Normalized Weblog file

IP	Pages Visited
192.157.55.78	1,3,5,9
122.87.23.61	1,3,4
192.22.58.23	1,6,3,5,15,17
90.76.02.19	1,3,5,15,17
201.23.12.61	1,3,4,3,17
156.76.72.65	1,3,5,15

2. RESEARCH PROBLEM

Mining sequential weblog data from a transactional database refers to the discovery of item sets with high or low frequency and support. The situation may become worse when the database contains lots of long transactions or if the dataset is very large in size. The main objective of the paper is to discover the infrequently visited pages in a site and to discover this we need to generate sequential candidates of the weblog data and prune away the frequently visited pages in tandem.

The problem in this weblog data is discovering the outliers (i.e.) the individual page in the transaction dataset is not enough to provide the required details about the browsing behavior [9] of the users. For example if a user visits page 3 and from there he moves to page 9 frequently, it is a frequent path chosen by the users but if the user moves from page 3 to page 5 rarely, this is the actual outlier in this research work. We need to discover such navigational pages to help the designer of the website to redesign or restructure the site.

The major hindrance or the glitch present in most of the existing techniques is lack of execution speed and huge memory space cost. The challenge of web sequential frequent item set mining is in restricting the size of the candidate set and simplifying the computation and complexities related to time and memory.

2.1 RELATED WORKS

A sequence dataset consists of ordered elements or items, archived with or without a concrete notion of time such as consumer shopping sequences, web click streams, and biological sequence data. Sequential pattern mining, the mining of frequently occurring events or subsequences as patterns, was first introduced by Agarwal et al [1].

GSP Algorithm (*Generalized Sequential Pattern* algorithm) is an algorithm used for sequence mining [4]. The algorithms for resolving sequence mining related problems are mostly centered on the *a priori* [level-wise] algorithm. The apriori use the level-wise paradigm to first extract all the frequent items in a level-wise approach. Level wise paradigm means counting the occurrences of all singleton items in the dataset. Next, the transactional items are filtered by eliminating the non-frequent unpromising items. At the end of this phase, each transaction consists of only the frequent items. This modified dataset with frequent item set becomes an input to the GSP algorithm. This process requires only one pass over the whole database.

SPADE [5] is an algorithm that is based on lattice theory and applies temporal join operation to find sequential patterns. This algorithm is based on apriori approach and performs better than GSP.

The key features of the algorithm are as follows:

1. Spade use a *vertical id-list* database format, where it associates with each sequence a list of objects in which it occurs, along with the time-stamps. Spade show that all frequent sequences can be enumerated via simple temporal joins (or intersections) on id-lists.
2. Spade uses a lattice-theoretic approach to decompose the original search space (lattice) into smaller pieces (sub-lattices) which can be processed independently in main-memory. SPADE approach usually requires three

database scans, or only a single scan with some pre-processed information, thus minimizing the I/O costs.

2.2 PRELIMINARY ON SUPPORT CONSIDERATION

Consider I = {I_1, I_2, … , I_n} be a set of 'n' distinct attributes, T be transaction that contains a set of items such that T ⊆ I, Ds be a database with different transaction records T. An association rule is an implication in the form of A → B, where A, B⊆ I are sets of items called item sets, and A n B = ø. A is called antecedent while B is called consequent.

Assumption:

Sequential Web pages symbolizes that the antecedent Web page should be navigated or browsed before the descendant Web page.

This assumption is very crucial due to the fact that Web pages archived in the server log files are sequential in nature and the order of the viewed Web pages in a website is decisive in the prediction process of discovering the navigational patterns of the users [2][3].

Definition:

An item set is called an infrequent item set if its support count value is less than a user-specified minimum support threshold which is denoted as min_sup [6]. Else, it is called a frequent item set.

Support(s) of an association rule [10] is defined as the percentage/fraction of transaction records that contain A U B to the total number of transactional records present in the database. The support is one of the major parameter used in association rule mining and this value is used to filter out the rules which are not so interesting or useful for the decision making process.

Definition:

Let Ds be the transaction dataset over a set of distinct items I. For the computed mean min_sup value, the problem of ***Non Frequent Item set Mining NFI*** *is to find the complete set of Frequent Item sets as shown below*

{AB | AB ⊆ I & Sup(AB) ≤ mean support Mρ}

Pruning Property

"A sequential item set is considered as an outlier, if the support count is less than the mean min_Sup , else a promising item set."

Effective pruning reduces the volume of candidates generated considerably and decreases the execution time consumed and the memory usage of the machine. The main over heads in data mining are memory consumption and time consumption and these are the main factors that increases the overall computation cost of the algorithm.

Anti-monotonicity property of unpromising items:

If i_m is an unpromising item, then i_m and all its supersets are not promising items. Only the supersets of promising items have the maximum possibility [7] to be a high promising item set.

The proposed algorithm DASPAT utilizes this anti-monotonicity property to reduce the volume of candidate generation during second phase of the algorithm and thereby reduces the memory and time consumption considerably.

Since the proposed algorithm utilizes stringent pruning techniques, the candidate generation will be very small when compared to other existing algorithms and which in turn will reduce the memory consumption to a greater extent. The voluminous candidates, iterations and passes to mine through these candidates will pile the memory footprint and the computational cost of the machines considerably. But the proposed algorithm addresses this issue quite well and reduces the memory footprints considerably.

3. PROPOSED APPROACH

The proposed approach is to discover the navigational paths visited infrequently by the users. To accomplish

this, a new algorithm named "Detect Anomaly in Sequential Pattern Algorithm (DASPAT)" is proposed. The first step in the proposed approach is to clean the noisy data present in the weblog dataset to ensure accurate detection of outliers since noise is different from outliers. Next step is to generate sequential candidates with customized pruning technique to remove the very frequent sequential sets and to reduce the dimension of the data. Next step is to detect the outliers present in the normalized data and discover the infrequent sequential pages. To accomplish these steps separate procedures are proposed and the working concept behind these procedures is enumerated with examples.

3.1 NOISE REMOVAL

This procedure helps to remove the false hits present in the weblog dataset as many of the pages will be repeated in succession in the transactional row of the dataset. This has to be removed to provide accurate results.

PROCEDURE Remove Noise(Dataset Đ)

INPUT: Sequential Dataset Đ
OUTPUT: Noiseless Dataset Đ
BEGIN:

1. Find the total Transactional Rows Ř ∈ Đ
2. For all Row Ř ∈ Đ do
3. Find the Total Elements Ī ∈ Row Ř
4. For all Elements Ī ∈ Ř do
5. CHECK IF (Elements (Ī) = Elements (Ī +1)) then
6. Remove Elements at (Ī +1)
7. End IF
8. End FOR
9. Return Đ

END PROCEDURE

Figure1.1: Pseudo code of Remove Noise

Let us consider a transactional row T2 which contains 8 items {I1, I2, I3, I4, I5, I6, I7, I8} and the values corresponding to these items are {3, 2, 2, 2, 5, 5, 6, 8}. Here I2 is data and I3, I4 are noises which has to be cleaned before the candidate generation. The RemoveNoise procedure checks the I^{th} item with the $(I+1)^{th}$ item and if found to be equal, the (I+1) item will be removed from the transactional row. The resultant cleaned transactional row T2= {3, 2, 5, 6, 8}

3.2 SEQUENTIAL CANDIDATE GENERATION

PROCEDURE CreateItemsets (Dataset Đ)

Input: Dataset Đ = {T_1, T_2, T_3..... T_n }
Output: Sequential Itemsets SeqCand
BEGIN:

1. Scan Dataset Đ
2. Compute the individual item support count ρ
3. Find the MeanSupport Mρ
   $M\rho = \frac{1}{n}\sum_{1=1}^{n} (\rho_i)$
4. For all TransactionRow Ř ∈ Đ do
5. For all Elements Ī ∈ Ř do
6. Compute Elements count
7. While [SeqCand count ≤ Elements count] do
8. Combine Elements[Ī] ∪ Elements[Ī + 1] => SeqCand
9. IF [Element in SeqCand Support > = Mρ]
10. Prune SeqCand Set
11. End While
12. End For
13. End For
14. Return SeqCand

END PROCEDURE

Figure 1.2: Pseudo code of CreateItemsets

The main problem here in the paper is pruning the unpromising items when they does not possess value more than mean support, single items are not pruned away after finding the individual support. Instead the itemset is formed and the mean support value is compared with the item set and if all the elements in the itemset possesses values more than the mean support

then the itemset is pruned away. This technique is employed to reduce the dimension of the candidates.

Considering the table 1.2, the individual support of the elements in the transaction table is computed as shown in the table 1.3

Table 1.3: Computed individual support count

ITEM	COUNT	ITEMS	COUNT
1	6	6	1
3	6	9	1
4	2	15	3
5	4	17	3

The mean support is computed for the transaction table using the mean formula,

$$M\rho = \frac{1}{n}\sum_{1=1}^{n} (\rho_i)$$

Where n is number of items, ρ is individual support count.

$$M\rho = 3.25$$

The candidates are created for rows and the pruning mechanism is applied to reduce the dimension of the candidate generated. Let us consider the first row in the table 1.2,

T1= { 1, 3, 5, 9}

Itemsets = { [1,3], [3,5], [5,9], [1,3,5], [3,5,9], [1,3,5,9]}

[1,3] = support count of 1 is 6 and 3 is 6, since both are greater than mean support Mρ, Item set [1,3] is pruned.

[3,5] = support count of 3 is 6 and 5 is 4, since both individual support count is greater than mean support Mρ, item set [3,5] is pruned away.

[5,9] = support count of 5 is 4 and 9 is 1, here the support count of 9 is less than mean support Mρ, so this itemset is retained.

[3,5,9] = support count of 3 is 6, 5 is and 9 is 1. The support count of 9 is less than the mean support Mρ, so this itemset is retained.

Table 1.4: Candidate generation after pruning

ITEM	TID	ITEM	TID
5,9	T1	15,17	T4
3,5,9	T1	3,5,15	T4
3,4	T2	5,15,17	T4
1,6	T3	1,3,5,15	T4
6,3	T3	3,5,15,17	T4
5,15	T3	3,4	T5
15,17	T3	4,3	T5
1,6,3	T3	3,17	T5
6,3,5	T3	1,3,4	T5
3,5,15	T3	5,15	T6
5,15,17	T3	3,5,15	T6
1,6,3,5	T3		
6,3,5,15	T3		
3,5,15,17	T3		
5,15	T4		

The mean support value based pruning reduces the candidate generation to a certain extent and the reduction percentage is calculated using the formula,

$$\text{Candidate reduction} = \frac{\text{Pruned candidates}}{\text{Total}} \times 100$$

$$\frac{\text{Pruned candidates}}{\text{Actual candidates}} \times 100$$

$$= (26 / 38) \times 100 = 68.72\ \%$$

3.3 OUTLIER DETECTION

The resultant candidates produced by the createItemset procedure are used to detect the outliers present in the itemset and to identify the low hit pages of a website. Here the count of every itemset is computed and compared with the mean support value found earlier in the createItemset procedure, if the count is more than the mean support value Mρ, the itemset is pruned away and the remaining itemsets are considered as outliers.

PROCEDURE DetectOUTLIERS (CandidateItemset C_i, Mean Support Mρ)

Input: Dataset Đ = {T_1, T_2, T_3…… T_n }

Output: Outliers

BEGIN:

1. Compute the total itemsets totItems
2. For all Itemset I ∈ C_i do
3. Initialize iC = 0
4. For all Itemset (I+1) ∈ C_i do
5. IF (Itemset[I] = Itemset[I+1]) then
6. Increment iC=iC+1
7. REPLACE Itemset[I+1] by "****"
8. END IF
9. END FOR
10. IF(iC ≤ Mρ) then
11. STORE Itemset as Outliers
12. END IF
13. END FOR
14. RETURN Outliers

END PROCEDURE

Figure 1.3: Pseudo code of detectOUTLIERS

This procedure clearly detects the outliers present in the transaction database and portrays the result to the site owners enabling them to remodel or redesign the site according to the results to fetch more hits to the low hit pages in the website. The proposed algorithm uses these procedures to discover the low hit pages as shown in the following section.

ALGORITHM DASPAT (Dataset Đ)

Input: Dataset Đ = {T_1, T_2, T_3…… T_n }

Output: Outliers

BEGIN:

1. Load dataset **Đ**
2. RemoveNoise(Dataset Đ)
3. Candidates= CreateItemsets (Dataset Đ)
4. DetectOutliers(Candidates)
5. Return Outliers

END PROCEDURE

Figure 1.4: Pseudo code of DASPAT Algorithm

The step 2 of the DASPAT algorithm removes the duplicate items present in succession in the transaction dataset. The step 3 generates candidates after employing mean support based pruning technique to reduce the dimension of the candidate generated. The step 4 detects the outliers present in the dataset and finally the low hit pages are identified.

Final Output

Table 1.5: Final output as outlier

ITEMSET	SUPPORT
1,6	0.16
3,7	0.16
4,3	0.16
5,9	0.16
6,3	0.16
1,6,3	0.16

From the table 1.5, the output is interpreted as the users navigates from page 1 to page 6 very rarely and this signifies the site owners an alarm to the site designers and owners to restructure the site to compel the users to navigate through the low hit pages or the weak pages where visibility of the pages are not good.

4. EXPERIMENTAL EVALUATION

The proposed algorithm DASPAT is executed to test the performance and the experiments were performed on a dual core 2.66 GHz processor with 1 GB RAM running on windows 7 ultimate platform. The algorithm is implemented in visual VB6.0 and executed on a sparse BMS Web View dataset comprising of 59601 sequences, 497 distinct items, and average length of 2.51. The proposed algorithm detects the outliers quite well with minimum memory usage, faster speed and less overall computational speed.

The proposed algorithm was experimentally evaluated with the existing algorithms like GSP and from the results it is observed that the proposed algorithm outperformed the existing GSP algorithm by a large magnitude with respect to speed, memory and candidates. But the GSP needs to be provided with a user specified minimum support value whereas the proposed algorithm automatically computes the mean support.

From the graph shown in figures 1.5(a)and (b), it is evident that the proposed algorithm performed better than the existing GSP algorithm in terms with the speed and candidate generation.

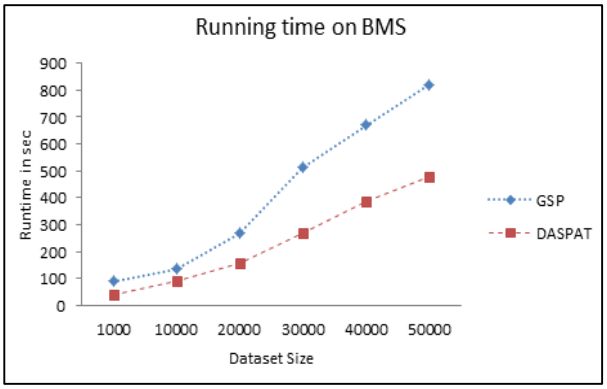

Figure 1.5: (a)Comparison of DASPAT with GSP

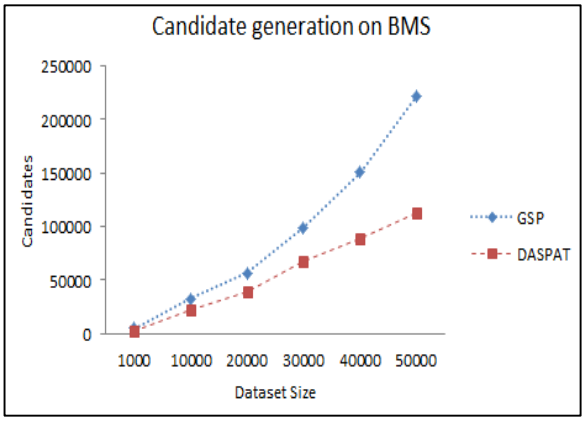

Figure 1.5(b): Comparison of DASPAT with GSP

5. CONCLUSION AND FUTURE WORK

"What does ALL my data implies to the user?" This is the foremost inquiry that a web usage mining [11] system attempts to answer. However, as the complexity of the web applications and user's interface grows, we need to either exploit new techniques or optimize an existing approach in order to uncover a scalable and precise result. Recently, semantic enhancement of weblogs has been perhaps the most assuring advancement in the area of web usage mining. As emphasized in Cooley [12], "not only is the web usage mining enriched by the content and structure, it cannot be accomplished without it."

The proposed algorithm proves to be efficient in both run time and memory consumption, but there is always room for further research and improvement. Improvements can be made in the pruning strategies to ensure minimum number of candidates is generated with less running time and with less memory consumption. Clustering techniques can be employed to test the results to fetch the low hit pages and the outlier based on clustering might be a better solution to the site designers.

The proposed algorithm DASPAT clearly performed well with respect to running time, candidate generation and memory footprints. This paper work analyzed the three major hindrances which blocks the smooth performance of the sequential mining namely, candidate, memory usage and running time. The experimental evaluation of the proposed algorithr `

provided a detailed comparative analysis and proved that the proposed algorithm fared better than the existing algorithm GSP.

REFERENCES

[1] R. Agrawal, and R. Srikant, *Mining sequential patterns*. In ICDE'95, Taipei, Taiwan, Mar. 1995.

[2] F. Masseglia, F. Cathala, and P. Poncelet, *The psp approach for mining sequential patterns*. In PKDD'98, Nantes, France, Sept. 1995.

[3] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining," Proc. 2000 ACM SIGKDD Int'l Conf. Knowledge Discovery in Databases (KDD '00), pp. 355-359, Aug. 2000.

[4] R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Per-formance Improvements. Research Report RJ 9994, IBM Almaden Research Center,San Jose, California, December 1995.

[5] SPADE: An Efficient Algorithm for Mining Frequent Sequences, MOHAMMED J. ZAKI*Computer Science Department, Rensselaer Polytechnic Institute, Troy NY 12180-3590, machine learning,42, 31-60,2001*

[6] Mannila, H., & Toivonen, H. (1996). Discovering generalized episodes using minimal occurences. In *2nd Intl.Conf. Knowledge Discovery and Data Mining*.

[7] Mannila, H., Toivonen, H., & Verkamo, I. (1995). Discovering frequent episodes in sequences. In *1st Intl. Conf.Knowledge Discovery and Data Mining*.

[8] O. R. Zaiane, M. Xin, and J. Han, "Discovering web access patterns and trends by applying olap and data mining technology on web logs," in *ADL '98: Proceedings of the Advances in Digital Libraries Conference*. Washington, DC, USA: IEEE Computer Society, 1998, pp. 1-19

[9] Albanese, M., Picariello, A., Sansone, C. & Sansone, L. (2004), 'Web personalization based on static information and dynamic user behavior', *WIDM'04,*
USA pp. 80–87.

[10] Liu, B., Hsu, W. & Ma, Y. (1999), 'Mining association rules with multiple minimum support', *KDD, San Diego* pp. 337–341.

[11] Kousalya, Suguna, Saravanan "Improving the Efficiency of Web Usage Mining Using K-Apriori and FP-Growth Algorithm ",March-2013

[12] R. Cooley, B. Mobasher, and J. Srivastava, "Data preparation for mining world wide web browsing patterns," *Knowledge and Information Systems*, Vol. 1, No. 1, pp. 5-32, 1999

S. Vasuki, the educational qualification of author is M.phil in computer science done in Algappa University, karaikudi, tamilnadu, India. in the year of April 2008.P.G degree M.S(IT&M) in Ayya nadir janaki amma college, sivakasi, tamilnadu., India in the year of April 2003. The author's major area of interest is data mining. She presented and participated in various colleges International and national conferences. Currently she is working as a Assistant professor in J.J college of Ats and Science(Autonomous),pudukkottai,,Tamilnadu, India.

Dr.K.Subramanin earned his Ph.D degree alagappa university in 2012.Now he is guiding 8 research scholars in bharathidasan university tiruchirappalli,tamilnadu,india. He is having more than 18 years teaching experience.Currently he is working in Governement Arts college,kulithalai,Tamilnadu, india .