Journal of Theoretical and Applied Information Technology
10th February 2014. Vol. 60 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

UTILITY, IMPORTANCE AND FREQUENCY DEPENDENT ALGORITHM FOR WEB PATH TRAVERSAL USING PREFIX TREE DATA STRUCTURE

[1]L.K.JOSHILA GRACE AND [2]Dr. V. MAHESWARI

[1]Research Scholar, Department of Computer Science and Engineering,
Sathyabama University, Chennai, India.
[2]Professor and Head , Department of Computer Application,
Sathyabama University, Chennai, India
E-mail: joshilagracejebin@gmail.com

ABSTRACT

Perceiving the navigational performance of website visitors is an essential factor for the success in the emerging systems of electronic commerce and mobile commerce. In this paper, we propose a utility and frequency dependent algorithm for web path traversal using prefix tree structure. Here, we take the web log files and extract the details such as time duration that the user visited the web page and the bookmark information about the web page. The extracted data from the web log file of different users are considered as input to construct the prefix tree. Using this prefix tree we mine the continuous sequential pattern and discontinuous sequential pattern and we calculate the weight value for each pattern mined using the continuous and discontinuous sequential pattern. The prefix tree constructed using the existing users can be updated further with the details of new users. We compare the path traversal efficiency of our proposed technique with the existing frequency dependent prefix span technique.

Keywords: *Continuous Sequential Pattern, Discontinuous Sequential Pattern, Time Duration, Bookmark, Prefix Tree*

1. INTRODUCTION

Web mining [1] is an area of data mining that manages the extraction of interesting knowledge from World Wide Web. More exactly [2], web content mining is a part of web mining that aims on the raw information existing in the web pages; source data primarily contains textual data in web pages (e.g. words, but also tags); typical applications are content based classification and content based ranking of web pages. Web structure mining is a part of web mining that aims on the structure of web sites; source data mainly contains structural information in web pages (e.g. links to other pages); typical applications are link based classification of web pages, ranking of web pages through a mixture of content and structure (e.g. [3]) and reverse engineering of web site models. Web usage mining is a part of web mining that manages the extraction of knowledge from server log files; source data primarily contains (textual) logs that are gathered when users access web servers and might be denoted in standard formats; typical applications are those based on user modeling techniques that are web personalization, adaptive web sites and user modeling [4].

In modern days, the research activities in web mining are based on the web usage mining. To discover interesting usage patterns from web server log files, web usage mining technique is widely used. To extract the user navigation patterns, mining algorithms are used. A navigation pattern denotes the relationship among web pages in a specific web site. In practice, mining web navigation patterns is useful and the extracted patterns can be used to predict and grasp visitors browsing behavior and intentions. It is helpful to improve the user experience, website configuration, efficiency and effectiveness of e-commerce. Website operators can apply web navigation patterns to analyze and predict user motivation and provide better recommendations and personalized services for their customers [5]. The issue for mining path traversal patterns from a large static web click dataset was proposed by Chen *et al.* [6]. Two multiple pass algorithms which are full scan and selective scan were proposed for mining set of path traversal patterns [7]. In most of the works

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

presented in the literature, the path traversal patterns were mined using data mining techniques such as association rule discovery and sequential pattern mining.

The concept of utility based web path traversal mining has been developed in [8]. The concept of utility for web path traversal first time was introduced in [9]. They adopted the descriptive measure for quality: the time spent on web pages (TSP) at particular web site as a utility of that web site. TSP is an appealing and interest indicators in diverse fields such as information retrieval, human-computer communication. It could easily be considered that TSP would be an explanatory measure of importance: the more time spend on a web page by the users, the more important the page assumed for them [10]. Incorporating TSP measure into the mining of path traversal pattern can lead to effective results to analyze and user forecast.

In this paper, we propose a utility, importance and frequency based algorithm for web path traversal using prefix tree data structure. Here, we use the web log file as input and we extract the details such as time duration, bookmark of that web page from the web log file. We construct a prefix tree based on the web pages visited by different users. Initially, we take the web pages visited by the first user to construct the prefix tree and thereafter we compare the web pages visited by the second user with the tree formed by first user. If the first web page of the second user is not same as that of the first user, a new root will form in the prefix tree to construct the nodes for the second user and if the first web page of the first and second users are same, the details will include with the first node of the first user and check for the second web page. If the second web page of the second user is not same as that of the first user's second web page, a new branch will form after the first node. Similarly, all the web pages of every user are checked and the prefix tree will be formed. When a new user comes, we can link those details with the already existing tree. We mine the continuous and discontinuous sequential patterns from the prefix tree we formed using the web pages visited by different users. Thereafter, we calculate the weight value for each patterns mined using continuous and discontinuous sequential patterns. This paper is organized as follows: second section shows some of the related work and the third section explains our proposed technique and the fourth section shows the outcome of our technique and fifth section concludes our technique.

2. RELATED WORKS: A BRIEF REVIEW

Literature presents several techniques for path traversal patterns mining based on either frequency or utility. In this section, we review some of the related techniques. Jieh-Shan Yeh *et al.* [15] have utilized the HITS values and PNT preferences as measures to mine users' favored traversal paths. They have structured mining based on HITS (hypertext induced topic selection) to rank the web pages. The preferred navigation path of the users' is found by an algorithm named PNT (preferred navigation tree). They have introduced PNTH (preferred navigation tree with HITS) algorithm which is an extension of PNT. Their algorithm used the concept of PNT and considered the relationship amid web pages using HITS algorithm. Their algorithm was suitable for e-commerce applications like improving web site design and web server performance. Yao-Te Wang and Anthony J.T. Leeb [17] have introduced the concept of throughout-surfing patterns (TSPs) and presented an efficient method for mining the patterns. The TSPs are more expressive to understand the purpose of website visitors. They suggested a compact graph structure, termed a path traversal graph to store information about the navigation paths of website visitors. The graph contained frequent surfing paths that were wanted for mining TSPs. In addition, they devised a graph traverse algorithm derived from the suggested graph structure to discover the TSPs.

To ascertain the user traversal pattern of web pages from the web log records, the web usage mining is used. Typically, an admired website may register hundreds of megabytes of web log records every day that offers rich information about web dynamics. From the web log databases, the repeated sequential web accessing patterns were determined by path traversal pattern mining. However, it fails to reflect the different impacts of different Web pages to different users. In internet information service applications, the variation amid web pages makes a strong impact on decision making. Therefore, Lin Zhou *et al.* [12] introduced "utility" into path traversal pattern mining problem. Utility is an estimation of how 'interesting' or 'useful' a web page. As an outcome, it allowed the web service providers to quantify the user preferences of diverse traversal paths. To recognize high utility path traversal pattern, two-phase utility mining model is exploited. They implemented their suggested 'high utility path traversal mining' algorithm on real world weblog database and contrast the high utility path traversal patterns with the frequent traversal patterns by traditional path

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

traversal technique. They demonstrated the interesting paths, as well as their significance to the decision making process. Chowdhury Farhan Ahmed *et al.* [14] have proposed an algorithm for utility-based web path traversal pattern mining. Their extensive experimental outcomes showed that their algorithm outperformed the existing algorithm.

A frequent sequential traversal pattern mining algorithm with weights constraint was recommended by Sisodia, M.S *et al.* [13]. Their foremost approach was to include the weight constraints into the sequential traversal pattern and maintain the downward closure property simultaneously. A weight range was expounded to maintain the downward closure property and pages were given diverse weights and traversal sequences allot a minimum and maximum weight. In scrutinizing a session database, the maximum and minimum weight in the session database was exploited to reduce infrequent sequential traversal subsequence by doing downward closure property was maintained. Their technique produced a few but considerable sequential traversal patterns in session databases with a low minimum support, by varying a weight range of pages and sequence.

Very useful data from the web logs with wide applications can be identified by mining web access sequence. More realistic data can be extracted by taking account of non-binary incidents of web pages as internal utilities in web access sequences. The available utility based technique has many disadvantages such as taking account only the forward references of web access sequences, not applicable for incremental mining, suffers in the level based candidate generation and test methodology, needs many database scans and does not show how to mine the web traversal sequences with external utility. Ahmed, C.F and Tanbeer, S.K [16] have recommended a technique to solve those issues and they proposed two tree structures named utility based web access sequence (UWAS) tree and incremental UWAS (IUWAS) tree to mine the web access sequences in static and dynamic databases respectively. Their technique handled both forward and backward references, static and dynamic data, excludes level wise candidate generation and test methodology, does not scan databases several times and considers both internal and external utilities of web page.

V.ValliMayil [11] has developed user web navigation sessions were inferred from log data and a Markov chain. The chain's higher probability trail was the most favored trail on the website. The algorithm applies a depth-first search that scrutinized the markov chain for high probability trails. Their technique result in the forecast of admired web path and user navigation behavior. Web link prediction is a method to forecast the web pages visited by the user derived from web pages visited by other users already.

3. PROPOSED UTILITY AND FREQUENCY DEPENDENT ALGORITHM

This section explains our recommended methodology. The input we give for our recommended technique is web log files. The web page, time duration, bookmark details are obtained from the web log file and we use those details for our recommended process. The Fig.1 shows the process takes place in our recommended technique.

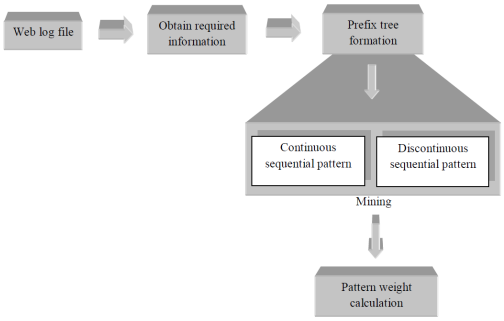


Fig.1 Sample Process Of Our Recommended Technique

The information we obtain from the web log file is used to construct the prefix tree. The continuous sequential pattern and discontinuous sequential pattern are derived from the prefix tree and the pattern weight for each pattern in the continuous sequential pattern and discontinuous sequential pattern are calculated.

3.1. Data Preprocessing

This section delineates the preprocessing of the web log file that we take as input of our recommended technique. Generally a web log file has the following structure: IP address, access time, HTTP request type used, URL of the referring page and browser name. An example for web log file is as follows: *192.164.31.18 [13/Jan/2013:10:18:52] "GET/HTTP/1.1" http://www.loganalyzer.net/ Mozilla/19.0 Windows 07*. We take the required fields from the web log file to process our recommended technique.

3.2 User Identification

This is an important section that makes to form a sequential database. To classify the users, we have to consider the IP addresses and the sessions. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IP address is divided as sessions based on certain time interval to make a unique user. For instance, if we set thirty minutes as one session for an IP address, the web pages visited within thirty minutes is taken as the web pages visited by a user. The next thirty minutes for that same IP address is taken as next user. Similarly, we have to make different users using different IP addresses.

3.3 Prefix Tree Construction

The prefix tree construction is explained as follows: a set of web pages which were visited by different users are taken for our process to construct the prefix tree. A sample web pages visited by different users are shown in Table.1.

Table.1 Sample Web Pages Visited By Different Users

Users	Visited Web Pages
$U1$	b,d,g,k,n,r
$U2$	b,c,e,g,h
$U3$	a,d,k,a,n
$U4$	f,c,s,e,g
$U5$	b,d,g,n,s

In this table the web pages visited by the users are in sequential order and the time duration of each web page and bookmark details is shown in the Table.2. The time duration, bookmark details and frequency of the web pages is used to calculate the weight value for each pattern in the continuous sequential pattern and the discontinuous sequential pattern.

Table.2 Sample Web Pages With Time Duration And Bookmark

Users	Visited Web Pages with Time Duration and Bookmark
$U1$	$(b,35,1),(d,26,0),(g,31,1),(k,43,1),(n,28,0),(r,27,1)$
$U2$	$(b,21,1),(c,44,0),(e,49,1),(g,38,1),(h,29,0)$
$U3$	$(a,33,0),(d,40,0),(k,27,1),(a,31,0),(n,29,1)$
$U4$	$(f,23,1),(c,27,1),(s,38,0),(e,45,1),(g,32,1)$
$U5$	$(b,36,1),(d,29,1),(g,32,0),(n,19,1),(s,23,0)$

In Table.2 the time duration that the users visited for a particular web site and the details about the bookmark i.e. whether the user bookmarked that web page or not is given followed by the web page. For instance, in $(d,29,1)$, d denotes the web page and 29 denotes the duration of time taken by the user for a particular website and 1 denotes that the web page is bookmarked and if it is 0 instead of 1, then it denotes the webpage is not bookmarked. The Fig.2 shows the prefix tree formed from the details given in Table.2.

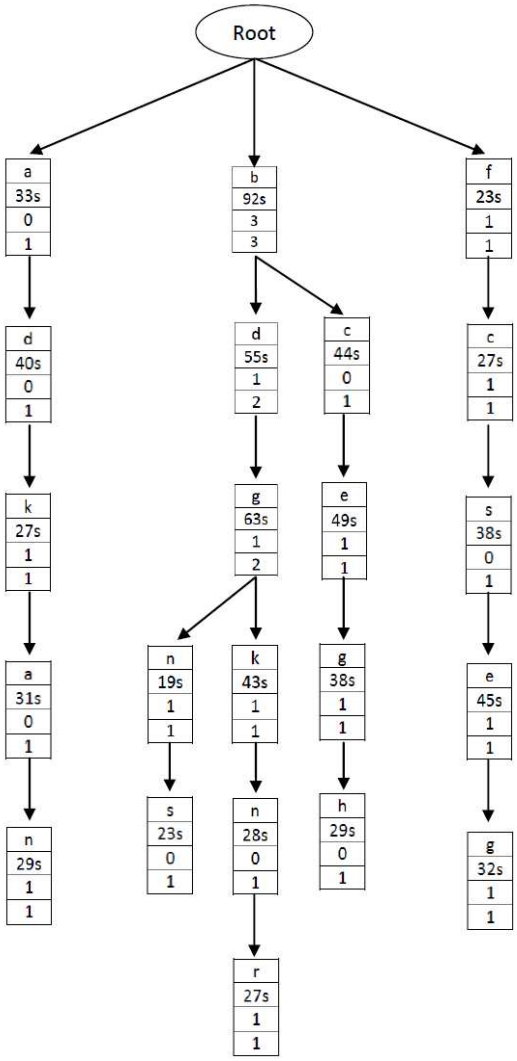


Fig.2 Prefix Tree

The prefix tree shown in Fig.2 is formed from the time duration and the bookmark details of the web pages given in Table.2. Here, initially the tree is plotted using the first user and thereafter the second user is considered and so on. For instance, the web pages visited by the first user are b,d,g,k,n,r. We generate the tree based on the first user initially and we take the web pages visited by the second user b,c,e,g,h to link it with the prefix tree construction. Comparing the web pages visited by the second user with the first user, the first web page i.e. b for both the users are same and the second web page visited by both are different. So while linking the second user in the prefix tree, a different branch would form in the

Journal of Theoretical and Applied Information Technology
10th February 2014. Vol. 60 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

tree after the first page which is common in both the users. Here, the frequency of the first webpage b would get increase. When considering the third user, the first web page a visited by the third user is different from both the first user and the second user; so a new branch is formed from the root itself. Similarly, we have to form the tree using the web pages visited by different users. The Fig.3 shows the details contain in the cells of a web page in the prefix tree.

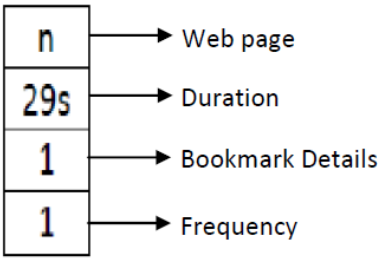


Fig.3 Representation Of Each Cell

The prefix tree formed eventually would get update using incremental concept. The incremental concept is that adding the details of web pages visited by new users. So, whenever a new user comes, the prefix tree will get update i.e. the details of the web pages visited by the new user will get link with the prefix tree already formed. The Fig.4 shows the algorithm for the construction of prefix tree.

Algorithm for prefix tree construction:

1. ***Input:*** Time duration, bookmark details of the web page extracted from web log file
2. Separate users based on specific time period
3. Construct tree using first user initially
4. **For** each user check web pages with the sequence of nodes in tree
5. **If** any sequence of nodes in the tree is same as the web pages of user we check
6. Add the details of the web pages of user with the nodes in the tree
7. **Else**
8. **If** some of nodes in the sequence is same from the beginning and differ in the middle
9. Add the details until the sequence and form a new branch from the node it differ
10. **Else**
11. Construct a new branch from the root
12. **End if**
13. **End for**
14. ***Output:*** Prefix tree based on the sequence of web pages visited by the users

Fig.4 Algorithm For Prefix Tree Construction

3.4 Continuous Sequential Pattern

It is a pattern that comes in sequential order. The sequential order is taken in different length and we calculate the weight value for each length of continuous sequential pattern considered from the prefix tree. Some sample continuous sequential pattern from the prefix tree in Fig.2 is $b \rightarrow c \rightarrow e, d \rightarrow k \rightarrow a$, etc. The given sample patterns are three length continuous sequential pattern. Similarly, we can take any length of continuous sequential patterns and we have to calculate the weight value for each pattern. The pattern $b \rightarrow c \rightarrow g$ is not a continuous sequential pattern, because the web page e is missing and this pattern is considered from the link of second user in the prefix tree. The Fig.5 shows the algorithm for the continuous sequential pattern.

Algorithm for continuous sequential pattern:

1. ***Input:*** Prefix tree we constructed
2. Generate set of patterns with different length
3. **For** each pattern check with prefix tree
4. **If** sequence is same
5. Consider the sequence and calculate weight value
6. **Else**
7. Ignore the sequence
8. **End if**
9. **End for**
10. ***Output:*** Weight value for the pattern that satisfies the sequence in the prefix tree

Fig.5 Algorithm For Continuous Sequential Pattern

3.5 Discontinuous Sequential Pattern

It is a pattern that comes in sequential order except the threshold length we set to ignore the web pages. Some of the sample discontinuous sequential patterns are $b \rightarrow c \rightarrow g$, $b \rightarrow g \rightarrow n$, etc. Here, the patterns given in sample discontinuous sequence are one length threshold based two length discontinuous sequential patterns. In $b \rightarrow c \rightarrow g$ pattern one web page e is missing amid the web pages c and g in the prefix tree; and in $b \rightarrow g \rightarrow n$ pattern, the web page d is missing between b and g, and the web page k is missing between g and n in the prefix tree shown in Fig.2. When considering the fifth user's link in the prefix tree for the pattern $b \rightarrow g \rightarrow n$, the web page d is missing amid the web pages b and g. Similarly, we can set any length as threshold for the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discontinuous sequential pattern. If we consider the threshold length as two, some of the sample three length discontinuous sequential patterns from the prefix tree are $b \rightarrow k \rightarrow n, b \rightarrow g \rightarrow h, f \rightarrow c \rightarrow g$, etc. We have to calculate weight value for each pattern formed from the discontinuous sequential pattern based on different threshold length. An algorithm is shown in Fig.6 for discontinuous sequential pattern.

Algorithm for discontinuous sequential pattern:

1. ***Input:*** Prefix tree we constructed
2. Generate set of patterns with different length
3. **For** each pattern check with prefix tree
4. Set discontinuous length
5. **If** sequence is same that satisfies discontinuous length we set
6. Consider the sequence and calculate weight value
7. **Else**
8. Ignore the sequence
9. **End if**
10. Change the length we set for discontinuous
11. **End for**
12. ***Output:*** Weight value for the pattern that satisfies the discontinuous length we set

Fig.6 Algorithm For Discontinuous Sequential Pattern

3.6 Pattern Weight Calculation

The pattern weight value of a pattern is the sum of utility of a pattern and importance of pattern and frequency of pattern. It is shown by an equation below:

$$P_W = P_u + P_{imp} + P_{freq}$$

Where,

P_W → Weight value of a pattern

P_u → Utility of a pattern

P_{imp} → Importance of the pattern

P_{freq} → Frequency of the pattern

Utility of a pattern: The utility of a pattern P_u is calculated based on the time duration td that the users taken for the web pages in the pattern. The utility of a pattern P_u is the ratio of total time duration of the web pages P_{td} in the pattern to the product of total number of nodes TN in the prefix tree and the maximum time duration $N_{td}^{\max}$ of a node in the prefix tree. The node in the prefix tree is the web page visited by the users. The formula to calculate the utility of a pattern is given below:

$$P_u = \frac{P_{td}}{TN \times N_{td}^{\max}}$$

Where,

P_{td} → Time duration of total web pages in a pattern

TN → Total number of nodes in the prefix tree

$N_{td}^{\max}$ → Maximum time duration of a node in prefix tree

The total time duration of the web pages P_{td} in a particular pattern is calculated by adding the time duration of all the web pages in that particular pattern. It is shown by the formula below:

$$P_{td} = \sum_{i=1}^{n} td_i$$

Where,

td_i → Time duration of i^{th} web page in a pattern

n → Total number (i.e. length) of web pages in a pattern

Importance of a Pattern: The importance of a pattern is based on the bookmark details of the web pages in the pattern. It is the ratio of total number of bookmarked web pages in the pattern to the total number of nodes in the prefix tree. The formula to calculate the importance of a pattern is as follows:

$$P_{imp} = \frac{P_{bc}}{TN}$$

In the above equation, P_{bc} indicates the total number of bookmarked web pages in a pattern and TN indicates the total number of nodes in the prefix tree. The nodes in the prefix tree are the web pages visited by the users.

Frequency of a pattern: The frequency of a pattern depends on the total number of similar pattern in the prefix tree. It is defined as the ratio of similar patterns in the prefix tree to the total number of nodes in the prefix tree. It is shown by an equation below:

$$P_{freq} = \frac{TP_{count}}{TN}$$

Where,

TP_{count} → Total number of similar pattern in the prefix tree

TN → Total number of nodes in the prefix tree

Journal of Theoretical and Applied Information Technology
10th February 2014. Vol. 60 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. RESULT AND DISCUSSION

This section explains the result we obtained for our proposed technique. We have used the synthetic dataset and real dataset for implementation and we evaluated the number of patterns mined by setting different threshold and we evaluated the efficiency.

4.1 Experimental setup and Dataset description

Our proposed technique is implemented in java (jdk 1.6) that has the system configuration as i5 processor with 4GB RAM. The synthetic dataset is generated as same format of the real dataset and it is divided as training dataset and testing dataset for our implementation. The real dataset is taken by the procedure as follows: we installed 'CC Proxy' software on the server to monitor the web usage of the users. The users are connected to the server and we checked the navigation of every user from the website "http://www.infrauniv.com/". The bookmark details are taken from each user separately. The Fig.7 shows the sample web log file and the web log files are converted to our process.

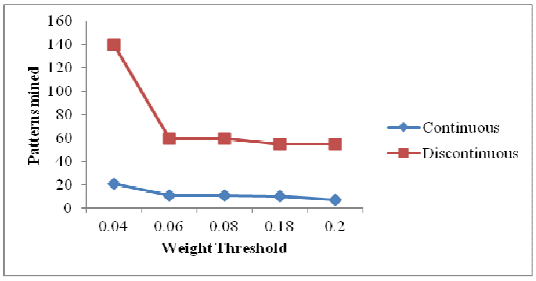
29/Jul/2013:17:18:08 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/aboutus.php HTTP/1.1
29/Jul/2013:17:18:23 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/solutions.php HTTP/1.1
29/Jul/2013:17:18:33 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/contactus.php HTTP/1.1
29/Jul/2013:17:19:27 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/events.php HTTP/1.1
29/Jul/2013:17:20:01 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/phd.php HTTP/1.1
29/Jul/2013:17:20:10 +0530	192.168.1.120	Akshara	HTTP	GET	http://www.infrauniv.com/index.php HTTP/1.1
29/Jul/2013:17:28:18 +0530	192.168.1.122	Akshara	HTTP	GET	http://www.infrauniv.com/infrastructure.php HTTP/1.1
29/Jul/2013:17:28:49 +0530	192.168.1.122	Akshara	HTTP	GET	http://www.infrauniv.com/aboutus.php HTTP/1.1
29/Jul/2013:17:28:54 +0530	192.168.1.122	Akshara	HTTP	GET	http://www.infrauniv.com/infrastructure.php HTTP/1.1
29/Jul/2013:17:29:12 +0530	192.168.1.122	Akshara	HTTP	GET	http://www.infrauniv.com/events.php HTTP/1.1
29/Jul/2013:17:29:17 +0530	192.168.1.122	Akshara	HTTP	GET	http://www.infrauniv.com/contactus.php HTTP/1.1

Fig.7 Sample Web Log File

4.2 Performance Comparison

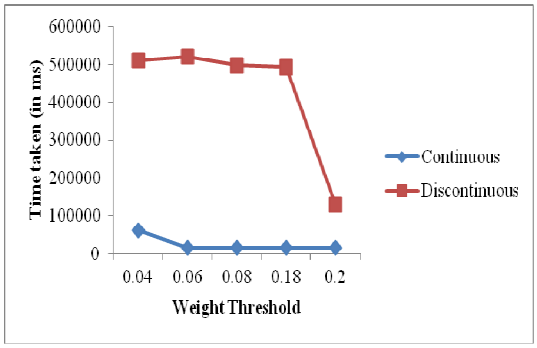
4.2.1. Comparison of patterns mined based on synthetic dataset

This section shows the performance of the continuous sequential pattern mining and discontinuous sequential pattern mining and the existing technique using synthetic dataset. The Graph.1 shows the comparison between the continuous and discontinuous sequential pattern mining based on the patterns mined for different pattern weight threshold we set.



Graph.1 Comparison Of Patterns Mined For Varying Weight Threshold Using Synthetic Dataset

The threshold we set to obtain the number of patterns mined is based on pattern weight. We set the pattern weight value as threshold and check the number of patterns mined for continuous pattern mining and discontinuous pattern mining. Here, when we set the threshold as 0.04, the number of patterns mined using continuous sequential pattern mining is 21 and the number of patterns mined using discontinuous sequential mining is 140. When the threshold is 0.06, the number of patterns mined is 11 using continuous sequential pattern mining and it is 60 using discontinuous sequential pattern mining. When we set the threshold as 0.08, we get the same number of patterns as we got using threshold 0.06 for both the continuous sequential pattern and discontinuous sequential pattern. When the threshold is 0.18, the number of patterns mined using continuous sequential pattern is 10 and the number of patterns mined using discontinuous sequential pattern is 55. When we set the threshold as 0.2, the number of patterns mined is 7 for the continuous sequential pattern and it is 55 for the discontinuous sequential pattern.

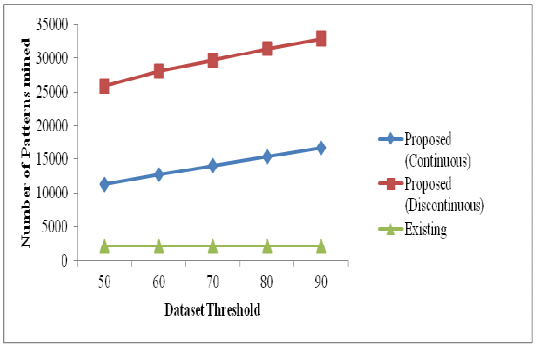


Graph.2 Execution Time For Patterns Mined By Varying Weight Threshold Using Synthetic Dataset

The Graph.2 shows the time taken to execute for finding the number of patterns mined by varying the pattern weight value as threshold. When the threshold is 0.04, the time taken to execute is 60.962s for continuous sequential pattern and 510.020s for discontinuous sequential pattern. When we set the threshold as 0.06, the time taken to execute is 16.367s for continuous sequential pattern and 522.008s for discontinuous sequential pattern. When the threshold is 0.08, the time taken to execute for continuous sequential pattern is 15.841s and for discontinuous pattern is 498.980s. When the threshold is 0.18, the execution time is 16.253s for continuous sequential pattern and it is 492.646s for discontinuous sequential pattern; and when we set the threshold as 0.2, the execution time

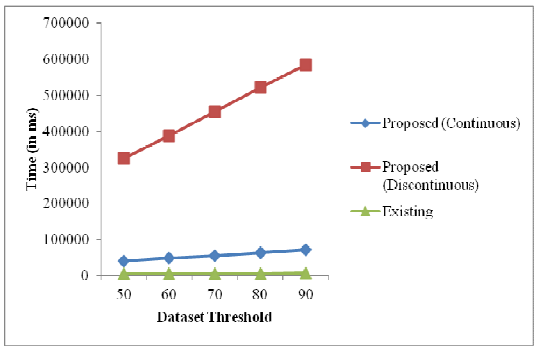
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is 15.814s for continuous sequential pattern and it is 129.789s for discontinuous sequential pattern.



Graph.3 Patterns Mined By Varying Percentage Level Of Dataset As Threshold Using Synthetic Dataset

The Graph.3 shows the number of patterns mined using our proposed technique (continuous sequential pattern and discontinuous sequential pattern) and the existing technique by varying the percentage level of dataset as threshold i.e. the percentage level of dataset we taken for training. When we take fifty percentages of dataset for training and remaining for testing, the number of patterns we obtained for continuous sequential pattern is 11267 and the number of patterns we obtained for discontinuous pattern is 25881 and the patterns we obtained using the existing technique is 2146. When we set sixty percentages of dataset for training and remaining for testing, we obtained 12724 patterns using continuous sequential pattern and 28120 patterns using discontinuous sequential pattern and 2115 using the existing technique. When the dataset we give for training as seventy percentages, the number of patterns we obtained is 14091 for continuous sequential pattern and it is 29672 for discontinuous sequential pattern and 2096 for the existing technique. When we give eighty percentages of dataset for training, we obtained 15407 patterns using continuous sequential pattern and 31452 patterns using discontinuous sequential pattern and 2102 patterns using the existing technique. When we give ninety percentages of dataset for training, we obtained 16723 patterns using continuous sequential pattern and 32933 patterns using discontinuous sequential pattern and 2093 patterns using the existing technique.



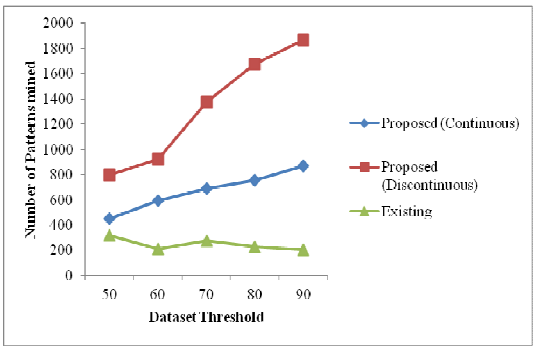
Graph.4 Execution Time By Varying The Percentage Level Of Dataset Using Synthetic Dataset

The Graph.4 shows the time taken to execute to find the number of patterns mined by varying the percentage level of dataset taken for training. When the dataset threshold is 50, the execution time is 40.645s for continuous sequential pattern and it is 324.483s for discontinuous sequential pattern and 5.583s for the existing technique; and when the dataset threshold is 60, the execution time is 48.182s using continuous sequential pattern and it is 388.203s for discontinuous sequential pattern and 5.482s for the existing technique; and when the dataset threshold is 70, the execution time is 55.288s for continuous sequential pattern and 455.164s for discontinuous sequential pattern and 5.777s for the existing technique. When we take eighty percentages of dataset for training, the execution time is 62.992s using continuous sequential pattern and it is 522.396s using discontinuous sequential pattern and 5.102s for the existing technique; and when we take ninety percentages of dataset for training, the time taken to execute is 71.113s using continuous sequential pattern and it is 585.649s using discontinuous sequential pattern and 6.574s using existing technique.

4.2.2. Comparison of patterns mined based on real dataset

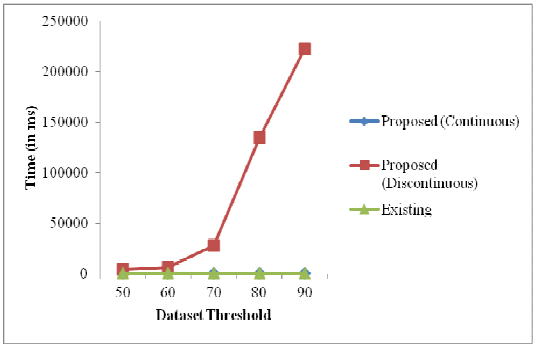
This section shows the performance comparison of the patterns mined using continuous sequential pattern, discontinuous sequential pattern and the existing technique based on real dataset. The Graph.5 shows the number of patterns mined using our proposed technique (continuous sequential pattern and discontinuous sequential pattern) and the existing technique by varying the percentage level of dataset for training and remaining for testing.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



Graph.5 Patterns Mined By Varying Percentage Level Of Dataset As Threshold Using Real Dataset

In this Graph.5, our proposed continuous and discontinuous patterns performed well than the existing technique. While comparing the continuous and discontinuous pattern, the discontinuous sequential pattern mined better compared to the continuous sequential pattern. The Graph.6 shows the time taken to execute to mine the patterns using real dataset.



Graph.6 Execution Time Using Real Dataset

In this Graph.6, the execution time is less for the existing technique compared to our proposed continuous and discontinuous sequential pattern mining techniques. While comparing the execution time of our techniques, the continuous sequential pattern mining obtained less execution time compared to the discontinuous sequential pattern mining.

4.2.3.Efficiency Comparison with Existing Technique

This section shows the comparison of our proposed technique with the existing technique in terms of efficiency. The Table.3 shows the comparison of our proposed technique with the existing technique based on path traversal efficiency calculation.

Table.3 Efficiency Comparison Using Synthetic Dataset

Path Traversal Efficiency			
Patterns	**Proposed Technique**		**Existing Technique**
	Continuous	***Discontinuous***	
1 length	1	1	1
2 length	1	1	1
3 length	1	1	1
4 length	0.787	0.892	0.36
5 length	0.33	0.663	0.055

The path traversal efficiency is calculated by dividing the number of patterns matched with the total number of patterns taken for comparison. It is shown by a formula below:

$$Pathtraversalefficiency = \frac{no.of\ patternsmatched}{Total\,no.of\ patternstaken\,for\,comparison}$$

The efficiency values shown in the Table.3 is calculated by giving eighty percentages of dataset for training and remaining dataset for testing. For instance, if there has fifty data in the testing dataset and if we need to check four length pattern, the first four nodes from each data are taken and checked with the tree we formed; if twenty patterns are exist in the tree, then the number of patterns matched is twenty and the total number of patterns we taken for comparison is fifty. In this Table.3, we have checked the path traversal efficiency of our proposed technique and existing technique for one length pattern, two length patterns, three length patterns, four length patterns and five length patterns. Here, the path traversal efficiency is hundred percentages for both the proposed and existing techniques for one length pattern, two length patterns and three length patterns. When evaluating the path traversal efficiency using four length patterns, the continuous sequential pattern of our proposed technique achieved 78.7 percentages and the discontinuous of our proposed technique achieved 89.2 percentages and the existing prefix span technique achieve 36 percentages. When calculating the path traversal efficiency using five length patterns, the continuous sequential pattern of our proposed technique achieved 33 percentages and the discontinuous sequential pattern of our proposed technique achieved 66.3 percentages and the existing prefix span technique achieved five percentages.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table.4 Efficiency Comparison Using Real Dataset

Path Traversal Efficiency			
Patterns	**Proposed Technique**		**Existing Technique**
	Continuous	***Discontinuous***	
1 length	0.875	0.875	0.75
2 length	0.875	0.875	0.75
3 length	0.25	0.255	0.25
4 length	0	0	0
5 length	0	0	0

The Table.4 shows the path traversal efficiency we obtained for our technique (continuous sequential pattern and discontinuous sequential pattern) and the existing technique using real dataset.

Top References of continuous pattern		
1 length pattern	**2 length pattern**	**3 length pattern**
http://www.infrauniv.com/favicon.ico	http://www.infrauniv.com/favicon.ico, http://www.infrauniv.com/index.php	http://www.infrauniv.com/contactus.php, http://www.infrauniv.com/aboutus.php, http://www.infrauniv.com/feedback.php
http://www.infrauniv.com/solutions.php	http://www.infrauniv.com/solutions.php, http://www.infrauniv.com/events.php	http://www.infrauniv.com/favicon.ico, http://www.infrauniv.com/index.php , http://www.infrauniv.com/infrastructure.php
http://www.infrauniv.com/	http://www.infrauniv.com/, http://www.infrauniv.com/favicon.ico	
Top References of discontinuous pattern		
1 length pattern	**2 length pattern**	**3 length pattern**
http://www.infrauniv.com/favicon.ico	http://www.infrauniv.com/favicon.ico, http://www.infrauniv.com/index.php	http://www.infrauniv.com/contactus.php, http://www.infrauniv.com/aboutus.php, http://www.infrauniv.com/feedback.php
http://www.infrauniv.com/solutions.php	http://www.infrauniv.com/solutions.php, http://www.infrauniv.com/events.php	http://www.infrauniv.com/favicon.ico, http://www.infrauniv.com/index.php , http://www.infrauniv.com/infrastructure.php
http://www.infrauniv.com/	http://www.infrauniv.com/, http://www.infrauniv.com/favicon.ico	
http://www.infrauniv.com/contactus.php	http://www.infrauniv.com/contactus.php, http://www.infrauniv.com/guidance.php	
Top References of prefix span		
1 length pattern	**2 length pattern**	**3 length pattern**
http://www.infrauniv.com/contactus.php	http://www.infrauniv.com/contactus.php, http://www.infrauniv.com/guidance.php	http://www.infrauniv.com/contactus.php, http://www.infrauniv.com/aboutus.php, http://www.infrauniv.com/solutions.php
http://www.infrauniv.com/favicon.ico	http://www.infrauniv.com/favicon.ico, http://www.infrauniv.com/index.php	

Fig.8 Web Pages Mined While Testing Using Real Dataset

The Fig.8 shows the web pages mined while testing using the real dataset for our technique (continuous sequential pattern and discontinuous sequential pattern) and the existing technique.

5. CONCLUSION

In this paper we have proposed a utility, importance and frequency dependent algorithm for web path traversal using prefix tree data structure. We used the web log files as input for our technique and we extracted the time duration of the websites that the users used and the bookmark information about the web pages and we constructed the prefix tree based on the web pages visited by different users. From this prefix tree, we have mined the continuous sequential pattern and discontinuous sequential pattern. We compared the number of patterns mined based on continuous and discontinuous sequential pattern by setting different thresholds. We also evaluated the efficiency of our technique and compared with the existing frequency dependent prefix span technique. The path traversal efficiency comparison showed that our proposed technique is better compared to the existing technique.

REFERENCES

[1] Oren Etzioni, “The world-wide web: Quagmire or gold mine?”, Communications of the ACM, vol. 39, no.11, pp.65–68, 1996.

[2] Kosala and Blockeel, “Web mining research: A survey”, ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 1–15, 2000.

[3] Sergey Brin and Lawrence, “Page. The anatomy of a large-scale hypertextual Web search engine”, Computer Networks and ISDN Systems, vol. 30, pp.107–117, 1998.

[4] Federico Michele Facca and Pier Luca Lanzi, "Recent Developments in Web Usage Mining Research", Lecture Notes in Computer Science, vol. 2737, pp. 140-150, 2003.

[5] Yao-Te Wanga, Anthony J.T. Lee, " Mining Web navigation patterns with a path traversal graph”, Expert Systems with Applications, vol. 38, pp. 7112–7122, 2011.

[6] M.-S. Chen, J.-S. Park, P.S. Yu, “Efficient data mining for path traversal patterns”, IEEE Trans. Knowl. Data Eng.(TKDE), vol. 10, no. 2, pp. 209–221, 1998.

[7] Hua-Fu Li, Suh-Yin Lee, Man-Kwan Shan, “DSM-PLW: Single-pass mining of path raversal patterns over streaming Web click-sequences”, Computer Networks, vol. 50, pp.1474–1487, 2006.

[8] Liu Y., W.-keng. Liao, Choudhary A. N.: A fast high utility itemsets mining algorithm. In Proceedings of the first International Conference on Utility-Based Data Mining, pp. 90-99, 2005.

[9] Zhou L., Liu Y., Wang J., Shi Y.: Utility-Based Web Path Traversal Pattern Mining. In Proceedings 7th International Conference on Data Mining Workshops, pp. 373-378, 2007.

[10] Istvan K. Nagy and Csaba Gaspar-Papanek, “User Behaviour Analysis Based on Time Spent on Web Pages”, Web Mining Applications in E-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

commerce and E-services, pp. 117-136 , January 2009.

[11] V.ValliMayil, "Web Navigation Path Pattern Prediction using First Order Markov Model and Depth first Evaluation," International Journal of Computer Applications, Vol.45, no.16, 2012.

[12] Lin Zhou,Ying Liu,Jing Wang,Yong Shi,"Utility-Based Web Path Traversal Pattern Mining," Proceedings of the Seventh IEEE International Conference on Data Mining Workshops,pp.373-380,2007.

[13] Sisodia, M.S. Pathak, M.; Verma, B.; Nigam, R.K."Design and Implementation of an Algorithm for Finding Frequent Sequential Traversal Patterns from Web Logs Based on Weight Constraint," 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), pp.317 – 322, 2009.

[14] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, Young-Koo Lee, "Efficient mining of utility-based web path traversal patterns, "Proceedings of the 11th international conference on Advanced Communication Technology,Vol.3,pp.2215-2218, 2009.

[15] Jieh-Shan Yeh ,Ying-Lin Lin,Yu-Cheng Chen," Mining Preferred Traversal Paths with HITS, "Proceedings of the International Conference on Web Information Systems and Mining,pp.98-107, 2009.

[16] Ahmed, C.F.,Tanbeer, S.K. ; Byeong-Soo Jeong ,"Mining High Utility Web Access Sequences in Dynamic Web Log Data ,"11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), 2010.

[17] Yao-Te Wang, Anthony J.T. Leeb," Mining Web navigation patterns with a path traversal graph", Journal Expert Systems with Applications," Vol.38, no.6, pp.7112-7122, June, 2011.