ISSN:2320-0790

An International Journal of Advanced Computer Technology

A Novel Prefetching Technique through Frequent Sequential Patterns from Web Usage Data

Veeragangadhara swamy T.M[1], Dr.G.T.Raju[2]

[1]Asst.Professor & Head, Department of IS&E, GMIT, Davanagere, Karnataka, India.
[2]Professor & Head, Department of CS & E, RNSIT, Bangalore, Karnataka, India.

Abstract: Frequent sequential patterns (fsp) from web usage data (wud) are very important for analyzing and understanding users behavior to improve the quality of services offered by the world wide web(www). Web prefetching is one of the techniques for reducing the web latency there by improve the web retrieval process. This technique makes use of prefetching rules that are derived from fsps. In this paper, we explore the different fsp mining algorithms such as spm, fp growth, and spade for extraction of fsps from wud of an academic website for a period that varies from weekly to quarterly. Performance analysis on all of these fsp algorithms has been made against the number of fsps they generate with a given minimum support. Experimental results shows that spade fsp mining algorithm perform better compared to spm and fp growth algorithms. Based on the fsps, we propose a novel prefetching technique that generate prefetching rules from the fsps and prefetch the web pages so as to reduce the users' perceived latency.

1. INTRODUCTION

Discovering FSPs means to find inter-transaction patterns such that the presence of a set of items is followed by another item in the time-stamp ordered transaction set. In WUD, a visit by a user is recorded over a period of time. The time stamp associated with a transaction in this case will be a time interval which is determined and attached to the transaction during the data preprocessing process. Extracted FSPs from WUD helps in understanding and predicting the users' behavior, so that the server performance may be improved through Prefetching techniques that helps in reducing user's perceived latency thereby improving the quality of Web services.

In this paper, we explore the different FSP mining algorithms to extract FSPs of a website for a period that varies from weekly to quarterly and propose an approach that generates the Prefetching rules based on association rules generation concept. Further, these Prefetching rules could be used to prefetch the web pages and keep them at web server cache memory, so that these pages may be served for the requested user at faster rate reducing the web latency. Firstly, we perform preprocessing upon the raw WUD to get the session database of each Web user that lists the Web users (IPs) along with their corresponding sessions. Then, we use FSP mining algorithms to extract FSPs from this session database and analyze them to find out the Periodicity. Finally, we propose a novel Prefetching technique that generate Prefetching rules from the FSPs and prefetch the Web pages so as to reduce the users' perceived latency.

2. Related Work

The concept of sequence Data Mining was first introduced by Rakesh Agrawal and R Srikant in the year 1995. The problem was first introduced in the context of market analysis. It aimed to retrieve frequent patterns in the sequences of products purchased by customers through time ordered transactions. Later on its application was extended to complex applications like web users' prediction, network detection, DNA research, etc., several techniques have been proposed for sequential pattern mining. They are mainly of two types: (i) *Apriori based* (ii) *Frequent Pattern growth (FP-growth) based*. *Apriori based* mining techniques such as Apriori-all, GSP [3], SPADE [4], LAPIN-SPAM [5], LAPIN [6], scan the database multiple times. A *n* size pattern requires *n* scans of the database and hence these mining techniques are generally inefficient. *FP-growth based mining* techniques

such as FreeSpan, BIDE [7], COBRA [8], PrefixSpan [9], UDDAG [1], etc., utilize a tree based representation that reflects the original database and two scans are required to construct the tree. From this tree, the sequential patterns are derived without reference to the original database. Changes in the original database can easily be reflected in the tree by incremental analysis. These sequential pattern mining algorithms are used for mining Web access patterns from WUD. Their variants are summarized as follows:

Cheng, et al., [10] proposed an approach that combines Apriori-all and clustering to identify the user access patterns and cluster users' path patterns for users in personalized service. Gaol [11] explored habits of users using Apriori-all algorithm which first stores the original web access sequence database for storing non-sequential data. This is based on the fact that the greater the number of combinations produced, the less likely the number of users who perform a combination of these and vice versa. While this approach is simple and straight-forward, such brute force tactics are obsolete as Apriori-all algorithms are found to be least efficient with respect to sequential pattern mining.

Pei, et al., [12] proposed Web Access Pattern tree (WAP-tree) for efficient mining of access patterns from Web logs. The Web access pattern tree stores highly compressed, critical information for sequential pattern mining. The WAP-tree registers all access sequence counts. There is no need for mining the original database any more as the mining process for all Web access patterns needs to work on the WAP-tree only. Therefore, WAP-mine needs to scan the access sequence database only twice. The height of the WAP-tree is one plus the maximum length of the frequent subsequences in the database. The width of the WAP-tree, i.e., the number of leaves of the tree, is the number of access sequences in the database. The size of the WAP-tree is much smaller than the size of access sequence database. It is shown that WAP-mine outperforms and has better scalability than GSP.

Xiaoqiu, et al., [13] proposed the improved WAP-tree in the form of highly compressed access sequences and introducing a sub-tree structure to avoid generation of conditional WAP tree repeatedly and to generate maximal sequences. Improved WAP-tree excels traditional WAP-tree in time and space, and shows better stability as the lengths of patterns vary. Also, mining frequent access sequences based on WAP-tree needs to scan transaction database only twice. Yang, et al., [14] designed an efficient algorithm Top Down Mine (TD-mine) which makes use of the WAP tree data structure for web access pattern mining. WAP tree can be traversed both top-down and bottom-up for the extraction of frequent access patterns. In TD-mine, a header table is used to traverse the tree from the root to the leaf nodes and mine patterns where the nodes are frequently accessed.

Liu, et al., [15] proposed the Breadth-First Linked WAP-tree (BFWAP-tree) to mine frequent sequences which reflects parent-child relationship of nodes. The proposed algorithm builds the frequent header node links of the original WAP-tree in a Breadth-First fashion and uses the layer code of each node to identify the parent-child relationships between nodes of the tree. It then finds each frequent sequential pattern through progressive Breadth-First sequence search, starting with its first Breadth-First subsequence event. BFWAP avoids re-constructing WAP-tree recursively and shows a significant performance gain.

Vijayalakshmi, et al., [16] designed an extended version of PrefixSpan called EXT-Prefixspan algorithm to extract the Constraint-based multidimensional frequent sequential patterns in web usage mining by filtering the dataset in the presence of various pattern constraints. EXT-PrefixSpan then mines the complete set of patterns but greatly reduces the efforts of candidate subsequence generation. This substantially reduces the size of projected database and leads to efficient processing. EXT-PrefixSpan can be used to mine frequent sequential patterns of multi-dimensional nature from any web server log file in the light of obtaining the frequent web access patterns. However, EXT-PrefixSpan does not specify any particular constraint for consideration.

Wu, et al., [17] proposed the CIC-PrefixSpan, a modified version of PrefixSpan that mines and generates Maximal Sequential patterns by combining PrefixSpan and pseudo-projection. First, preprocessing is done to categorize the user sessions into human user sessions, crawler sessions and resource-download user sessions for efficient Web sequential pattern mining by filtering out the non-human user sessions, leaving the human user sessions and finding the transactions using Maximum Forward Path (MFP). By utilizing CIC-PrefixSpan, the memory space is reduced and generating duplicate projections to find the most frequent path in the users' access path tree is also avoided. It is shown that CIC-PrefixSpan yields accurate patterns with high efficiency and low execution time compared to GSP and PrefixSpan. However, the frequent substructures within a pattern cannot be mined by CIC-PrefixSpan.

Verma et al., [18] designed a new pattern mining algorithm called Single Level Algorithm for extracting behavior patterns. These patterns are used to generate recommendations at run time for web users. Single Level Algorithm is designed keeping in mind the dynamic adaptation of focused websites that have a large number of web pages. It combines preprocessing, mining and analysis to eventually predict the users' behavior and hence is useful for specific websites and is highly scalable. It is shown to

be more efficient than Apriori algorithm. However, performing preprocessing on a very large Web log database can be time-consuming and too cumbersome to be integrated with mining and analysis.

Nasraoui, et al., [19] presented a framework for discovering and tracking evolving user profiles in real-time environment using Web usage mining and Web ontology. Preprocessing is first performed on the Web log data to identify user sessions. Then, profiles are constructed for each user and enriched with other domain- specific information facets that give a panoramic view of the discovered mass usage modes. This framework summarizes a group of users with similar access activities and consists of their viewed pages, search engine queries, and inquiring and inquired companies. By mapping some new sessions to persistent profiles and updating these profiles most sessions are eliminated from further analysis and focusing the mining on truly new sessions. However, this framework is not scalable.

Pitman, et al., [20] modified the Bi-Directional Extension (BIDE) algorithm for mining closed sequential patterns in order to identify domain-specific rule sets for recommendation of pages and personalization for web users in E-commerce. Individual supports are specified for each customer so that products can be recommended for individuals. Also, BIDE creates multidimensional sequences and further increase prediction for customers who do not explicitly specify their needs by using search functionality. However, additional strategies must be explored for identifying the most relevant sequential patterns without an exhaustive exploration of the search space bounded only by minimum support.

Masseglia, et al., [21] proposed a Heuristic based Distributed Miner (HDM), a method that allows finding frequent behavioral patterns in real time irrespective of the number of web users. Navigational schemas, that are completely adaptable to the changing Web log data, are provided by HDM for efficient frequent sequence pattern mining. Based on a distributed heuristic, these schemas provide solutions for problems such as (i) discovering interesting zones (a great number of frequent patterns concentrated over a period of time) (ii) discovering super- frequent patterns and (iii) discovering very long sequential patterns and interactive data mining. However, the quality of the schemas can further be improved by adapting the candidate population.

Zhou, et al., [22] designed an intelligent web recommender system known as Sequential Web Access based Recommender System (SWARS) for sequential access pattern mining. Conditional Sequence mining (CS-mine) algorithm is used to identify frequent sequential web access patterns. The access patterns are then stored in a compact tree structure, called Pattern- tree, which is then used for matching and generating web links for recommendations. SWARS has shown to achieve good performance with high satisfaction and applicability.

Yen, et al., [23] address the issue of re-discovery of dynamic web logs due to the obsolete web logs as a result of deletion of users' log data and insertion of new logs. Incremental mining utilizes previous mining results and finds new patterns from the updated (inserted or deleted) part of the web logs. A new incremental mining strategy called Incremental Mining of Web Traversal Patterns (IncWTP) is proposed which makes use of an incremental updating algorithm to maintain the discovered path traversal patterns when entries are inserted or deleted in the database. This is achieved by making use of an extended lattice structure which is used to store the previous mining results. However, the changes made to the website structure will not be reflected in the lattice structure.

Zhang et al., [24] applied the Galois lattice to mine Web sequential access patterns by representing the paths traversed using graphs and compare the performance with that of Apriori. Since the Apriori-like algorithms frequently scan entire transaction database to generate candidate patterns, Galois lattice reduces time complexity of closed sequential pattern mining as it needs only one scan. Jain, et al., [25] proposed a technique that employs Doubly Linked Tree to mine Web Sequential patterns. The web access data available is constructed in the form of doubly linked tree. This tree keeps the critical mining related information in compressed format based on the frequent event count. It is shown that for low support threshold and for large data base Doubly Linked Tree mining performance is better than conventional schemes such as Apriori-all and GSP. However, Doubly Linked Tree does not work well in a distributed environment.

Jha, et al., [26] proposed a Frequent Sequential Traversal Pattern Mining based on dynamic Weights constraint of web access sequences (FSTPMW) to find the information gain of sequential patterns in session databases. The weight constraints are added into the sequential traversal pattern to control number of sequential patterns that can be generated in addition to minimum threshold. FSTPMW is efficient and scalable in mining sequential traversal patterns. But, it should be noted that FSTPMW does not consider levels of support along with the weights of sequential traversal patterns. Wang, et al., [27] proposed a Web personalization system that uses sequential access pattern mining based on CS-mine algorithm. The access patterns are stored in a compact tree structure called Pattern-tree which is then used for matching and generating web links for

recommendations. Pattern tree has shown to achieve good performance with accurate predictability.

Saxena, et al., [28] integrated mining and analysis by proposing the One Pass Frequent Episode discovery (FED) algorithm. In this approach significant intervals for each website are computed first and these intervals are used for detecting frequent patterns (Episodes). Analysis is then performed to find frequent patterns which can be used to forecast the user's behavior. The FED algorithm is very efficient as it finds out patterns within one cycle of execution itself. Oikonomopoulou, et al., [29] proposed a prediction schema based on Markov Model that extracts sequential patterns from Web logs using web site topology. Full coverage is achieved by the schema while maintaining accuracy of the prediction. Since Markov Models are infamous for their precision, the proposed prediction schema fails to deploy a more complex categorization method for each sequential pattern.

Rajimol, et al., [30] proposed First Occurrence List Maximum (FOLMax-mine) for mining maximal web access patterns based on FOL-Mine. It is a top-down method that uses the concept of first occurrence to reduce search space and improve the performance. This is achieved by finding out the Maximal Frequent Path in the patterns generated from the Web logs.

Fp growth algorithm for mining frequent patterns without candidate generation has been proposed by [31], FP- tree data has been used to store the compressed frequent patterns in transaction data base and mines in the form of projected Fp-tree. It is a pattern growth method for efficient mining of frequent patterns in large databases. Performance results show that Fp growth method is efficient and scalable for mining both short and long frequent patterns. It is faster than Apriori algorithm.

Although various researchers have contributed a lot towards extraction of FSPs for different applications, very few have applied on WUD. In this paper we explore Fp growth, SPM and SPADE. FSP mining algorithms on WUD and compare their performance analysis, the method that gives better results will be chosen for further processing.

3. Problem Formulation and Statement

Given Web usage data W, and the set of pages P = $\{p_i : 1<=i<=n\}$ of a website, a session is a subset of P, denoted by $(p_1, p_2, \ldots p_k)$, where $p_i \in P$, $i \in \{1, \ldots, k\}$. Here, the parentheses are omitted for a session with one page only. A Web access sequence *q* is a list of sessions, denoted by $< q_1 q_2 \ldots q_m >$, where q_i is a session and $q_i \subseteq P$, $i \{1, \ldots, m\}$. The number of sessions in q is called the length of q. Given two access sequences $x = < x_1 x_2 \ldots x_j >$ and $y = < y_1 y_2 \ldots y_k >$, *x* is said to be a subsequence of *y* and *y* a super sequence of *x*, if $k < j$ and there exists integers $1< i_1 < i_2 \ldots < i_j < k$, such that $x_1 \subseteq y_{i1}$, $x_2 \subseteq y_{i2}, \ldots, x_j \subseteq y_{ij}$. Here, x is also contained in *y* which is denoted by $x \subseteq y$. A session database is a set of tuples <*ip, q*>, where *ip* is the user's IP address which is used as a sequence identifier and *q* is the user's access sequence.

"Given a session database S, we construct Transaction database of WUD, and Apply FSP mining algorithms to extract FSPs, then apply association rule generation concepts for generating Prefetching rules".

Methodology

Given, WUD, we first perform the preprocessing and generate the session database *SD* as shown in Table 3. The session database consists of a set of tuples, with each tuple consisting of the user (IP) and the access sequence of that user. From Table 2, the user with IP address 1.0.1.2 has accessed the Web pages A, B, C and D with the sequence <A (A,C) B D>, where {A, B, C, D, E, F} is the set of unique Web pages accessed by different users. Here (A, C) in the above sequence denotes that in a single session, the two Web pages A and C were visited by the user in that order, otherwise the user is assumed to visit a single page per session if the parentheses () are not specified.

Table 1. Page Accesses

Pages	No. of Accesses
A	5
B	4
C	5
D	4
E	3
F	2
G	1
H	1

Table 2. Example Sessions Database.

User (IP)	Sequences
1.0.1.2	<A (A,C) B D>
1.0.1.3	<A D (E, F)>
1.0.1.4	<(B, D) C F>
1.0.1.5	<(C, E) (A, B, C, D) >
1.0.1.6	<A B C D E>

Web sequential patterns are mined from the session database *SD* using various FSP mining algorithms.

Database Transformation is used to remove infrequent pages, *i.e.*, those pages that do not have *MinSup* = 2.

From Table 1, the frequent items with *MinSup 2* are (A), (B), (C), (D), (E), (F), (A, C), (B, D). Since we require only these patterns, the remaining can be eliminated by substituting these patterns with non-negative integers, like : (A)-1, (A, C)-2, (B)-3, (B, D)-4, (C)-5, (D)-6, (E)-7, (F)-8. For the simplicity of representation, we assign each IP a unique identifier as well. The transformed database is as shown in the Table 4.

Table 3. After Database Transformation.

User (IP)	Sequences
P	<1 2 3 6>
Q	<1 6 (7, 8)>
R	<4 5 8>
S	<(5, 7) (1, 3, 5, 6) >
T	<1 3 5 6 7>

4. SYSTEM ARCHITECTURE

System Architecture of the proposed work is shown in figure 1. It consists of Pre processing, Sequential Pattern Mining and Pre-fetching Modules.

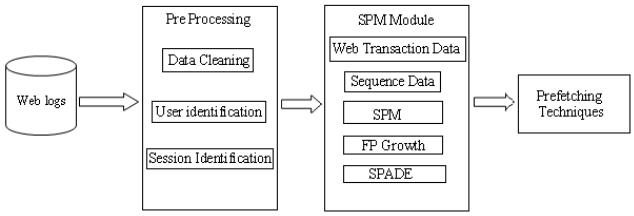


Figure 1: System Architecture

Weblogs

A Web log also called as WUD of a Web server contains of the transactions of Web users.

Preprocessing

- ***DataCcleaning:***

In this step the web usage data is cleansed, *i.e.*, useless records such as URLs containing images, multimedia, scripts and entries corresponding to crawlers are removed and only human initiated entries (*i.e.*, URLs ending with HTM, HTML, XHTML, PHP and JSP) are retained. Only a few of these fields are important for the mining process and hence after extracting or collecting only the important fields such as the User IP Addresses (referrers), URLs, Date, time spent the type of file in the URL (whether text, image, or script), the rest can be ignored.

- ***User Identification:***

The cleaned database (IP address) is grouped by different IP addresses and sorted by Date and Time for each IP for identifying different users and their transactions.

- ***Session Identification***

Session Identification is the process of partitioning the user activity records of each user into sessions in order to construct the actual sequence of actions done by each user. A session is a package of activities that consists of a user's navigation history. These sessions are then aggregated to create a session database. The user activity records are divided into sessions by assigning unique identifiers for each session. Each user is assigned a separate session and also a separate session is assigned for the same user if the user exceeds a certain threshold of time (15-20 minutes). Table 1 shows the sample user sessions for an academic website. Table 2 shows web transaction data for an academic website of 50 pages.

Table 4: Sample User Sessions for an Academic Web Site

Session Id	IP Address	Date &Time	URL Accessed
1	**117.216.148.95**	**2014-04-30 17:57:03**	**http://www.rnsit.ac.in/Invitations.html**
1	**117.216.148.95**	**2014-04-30 17:58:20**	**http://www.rnsit.ac.in/facilities.html**
2	**70.39.187.99**	**2014-04-30 18:03:38**	**http://www.rnsit.ac.in/hostel.html**
2	**70.39.187.99**	**2014-04-30 18:03:39**	**http://www.rnsit.ac.in /hostel.html**
3	**210.212.194.2166**	**2014-04-30 18:27:52**	**http://www.rnsit.ac.in rank_holders.html**
3	**210.212.194.2166**	**2014-04-30 18:27:53**	**http://www.rnsit.ac.in / rank_holders.html**
4	**115.118.176.83**	**2014-04-30 18:28:42**	**http://www.rnsit.ac.in /Admissions.html**
4	**115.118.176.83**	**2014-04-30 18:28:43**	**http://www.rnsit.ac.in /Admissions.html**
5	**117.208.191.46**	**2014-04-30 18:36:50**	**http://www.rnsit.ac.in /cse-dep.html**
5	**117.208.191.46**	**2014-04-30 18:36:51**	**http://www.rnsit.ac.in /cse-dep.html**

Table 5: Sequence data/ Web Transaction Data for Academic web site of 50 pages

Record No.	uid	pid
1	U5	P1,P9,P24
2	U23	P32,P24,P5
3	U34	P36,P14
4	U36	P23,P14
5	U4	P46,P14,P27
6	U27	P9,P48,P24
7	U17	p2,P16
8	U1	P31,P26,P22,P12
9	U2	P1,P9,P14
10	U35	P20,P47
11	U12	P44,P41
12	U35	P18,P44
13	U12	P23,P47

SPM Module

Pattern Mining is used to find hidden patterns from large database when a minimum threshold of occurrence (*MinSup*) has been specified. There we explore FP,SPM and SPADE algorithm for mining FSPs.

Prefetching module makes use of Association rules concept for generating Prefetching rules.

5. FSP Mining Algorithms

***SPM*:** The SPM algorithm also called as GSP (Generalized Sequential Pattern) algorithm [6] is an Apriori based sequential pattern mining algorithm. It is much faster than AprioriAll algorithm presented by Agarwal [2]. Two steps involved in GSP are Candidate Generation and Candidate pruning method. It has a very good scale up properties with respect to the number of transactions per data sequence and number of items per transaction. But it is not efficient in mining large sequence of databases having numerous patterns or long patterns as it cannot generate more candidate sequence and also multiple scans of database is needed because the length of each candidate grows by one at each database scan. Pseudo code of SPM is given below

SPM :

Input:

D = Database of Transactions;
U = Candidate Generation
P = Denotes the set of frequent 1- s
K = Length

Output: FSPs in D

1. Scan session Data base
2. Check for occurrence for particular Pid (after scan)
3. Let k=1;
4. Check for frequently visited pattern
5. Do while P(k)!=null;
6. Generate candidate sets U_{k+1} set of candidate k+1 sequences.
7. If U_{k+1} is not empty, find P_{k+1} i.e. the set of length (k+1) sequential patterns
8. k=k++;
9. End do

***SPADE*:** SPADE (Sequential Pattern Discovery using Equivalence classes) [7] is an Apriori based vertical format sequential pattern mining algorithm i.e. the sequences are given in vertical order instead of horizontal format. In addition, this algorithm uses the ID-List technique to reduce the cost for computing support counts. It consists of ID-List pairs where the first value stands for customer sequence and the second value refers to a transaction in it. The algorithm can use a breadth-first or a depth-first search method for finding new sequences. It needs multiple scans of database in mining. Mining long sequential patterns using SPADE is not possible as it needs an exponential number of short candidates .Pseudo code of SPADE is given below

. Input:

D = Denotes list of sequences
SID = Sequence_ID
EID = Event_ID
P = Page sequence
S = Support

Output: FSPs in D

1. First Scan S and transforms it into vertical format.
2. Let P1= frequent 1-sequences
3. Let P2=Frequent 2-sequences
4. Check equivalence classes belongs for all 1 sequences (with minimum support 2)
5. Second scan check equivalence classes for all 2 sequences.
6. For each [S] belongs and do
7. Find frequent sequences P;
8. End

Fp Growth: There are two methods to find FSPs, first one is FP tree and second is vertical data format. FP growth, which mines the complete set of frequent web pages without candidate generation and this method adopts a divide-and-conquer strategy, first it compresses the database representing frequent pages into frequent pattern tree, which maintain the page set association information. Then divides the compressed database into a set of conditional database. Mining frequent pages using vertical format, the FP growth methods more frequent patterns from a set of transaction in TID page set format, where pages is an web pages and TID set is set of transactions identifiers containing the pages. The main advantage of vertical format is , better than apriori algorithm. Pseudo code of Fp growth is given below

FP Growth:

Input :

D: Denotes database of Transactions.

Output: FSPs in D

1. Let P denotes the set of page set
2. Let k=1;

3. Do while p(k)!=Null;
4. Construct candidate set U_{k+1}
5. TID set pf P[k] && P[k+1] as same Transaction ID's
6. If U_{k+1} is not empty, find T_{k+1} , i.e the set of sequences
7. k=k+1;
8. End do;

6. Prefetching Rule Generation Technique

This component consisting of Prefetching rules pertaining to the requested Web pages. After the pattern analysis, Prefetching rules generated and are triggered when the first page in the sequence is accessed by a user.

The association rule mining is most popular and authentic instance appropriate advance for finding interesting relations between items i.e web pages.

Let $I=\{I_1,I_2,\ldots,I_m\}$ be a set of web pages. Let D, the task relevant data, bea set of database session Transactions where each transaction T is a set of items such that $T \subseteq I$. each transactub us associated with an identifier, called TID. Let P_1 be a set of pages. A transaction T is said contain P_1 if and only if $P_1 \subseteq T$. An association rule is an implication of the form $P_1 \Rightarrow P_2$, Where $P_1 \subset I$, $P_2 \subset I$, and $P_1 \cap P_2 = \phi$. The rule $P_1 \Rightarrow P_2$ holds in the transaction set D with support s, where s is the percentage of transactions in D that contain $P_1 U P_2$ and this is taken to be the probability, $P(P_1 U P_2)$. The rule $P_1 \Rightarrow P_2$ has confidence c in the transaction set D, where c is the percentage of transactions in D containing P_1 that also contain P_2 and this is to be the conditional probability, $P(P_2|P_1)$. That is

Support $(P_1 \Rightarrow P_2) = P(P_1 U P_2)$
Confidence $(P_1 \Rightarrow P_2) = P(P_2|P_1)$

Rules that satisfy both minimum support threshold (*min_sup*) and a minimum confidence threshold (min-conf).

1. $P_1 \wedge P_9 \Rightarrow P_{24}$ confidence= 2/2 = 100%
2. $P_1 \wedge P_{24} \Rightarrow P_9$ confidence = 2/1 = 20%
3. $P_9 \wedge P_{24} \Rightarrow P_1$ confidence = 2/2= 100%
4. $P_1 \Rightarrow P_9 \wedge P_{24}$ confidence = 2/2 =100%
5. $P_9 \Rightarrow P_1 \wedge P_{24}$ confidence =2/3 = 66%
6. $P_{24} \Rightarrow P_1 \wedge P_9$ confidence = 2 /3 = 66%

7. Experimental Results and Discussions

The proposed algorithms have been implemented using Java on WUD of academic website rnsit.ac.in. Each sequence in the dataset corresponds to page views of a user during period: weekly to quarterly. Each event in the sequence corresponds to a user's request for a page. The page requests served *via* caching mechanism are not recorded in the server logs and hence, not present in the data. Experiments are pursued to compare the efficiency of various FSP mining algorithms.Proposed algorithms demonstrated the satisfactory scale-up properties with respect to various parameters such as the total number of Web access sequences, the total number of pages, the average lengths of sequences. Table 6 shows the statistics for a website of 25 web pages with minimum support of 2 and Table 7 shows the statistics for a website of 50 web pages with minimum support of 2.

.

Table 6. Statistics for a website of 25 web pages with minimum support=2

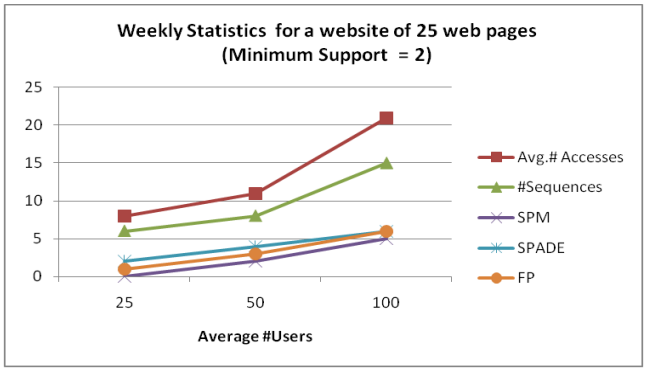
Period	Avg. # Users	Avg.# Accesses	#Sequences	FSPs		
				SPM	SPADE	FP
Weekly	25	8	6	0	2	1
	50	11	8	2	4	3
	100	21	15	5	6	6
Fortnightly	50	12	9	2	6	4
	100	27	20	5	8	6
	200	52	24	9	13	12
Monthly	100	24	18	8	13	12
	200	51	23	22	28	27
	500	134	25	51	56	53
Quarterly	200	54	23	24	38	26
	500	132	24	54	56	53
	750	213	26	72	80	71
	1000	262	30	113	114	113

Figure 2 shows the statistics such as average number of users, average of accesses, average number of sequences for a website of 25 pages and the number of FSPs generated by the different FSP using algorithms such as SPADE,SPM and FP growth, for the period that vary from Weekly to Quarterly

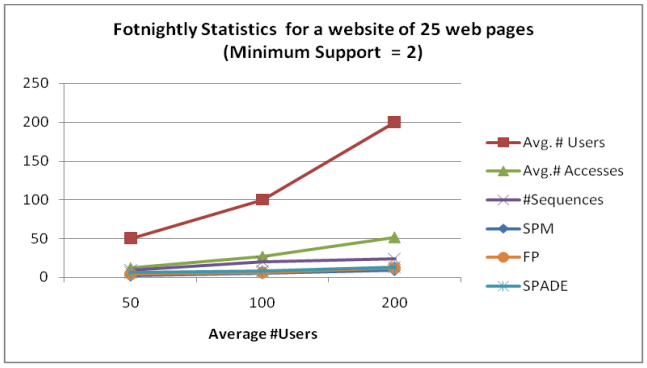
Table 7. Statistics for a website of 50 web pages with minimum support=2

Period	Avg. #Users	Avg.# Accesses	#Sequences	FSPs		
				SPM	SPADE	FP
Weekly	25	7	3	2	6	4
	50	13	5	4	10	8
	100	24	11	11	14	13
Fortnightly	50	12	8	6	9	8
	100	31	19	10	2	10
	200	53	47	13	16	15
Monthly	100	29	21	8	7	9
	200	47	33	15	18	16
	500	274	43	45	45	44
Quarterly	200	54	36	13	14	13
	500	141	50	56	58	47
	750	193	52	76	77	74
	1000	262	53	115	117	113

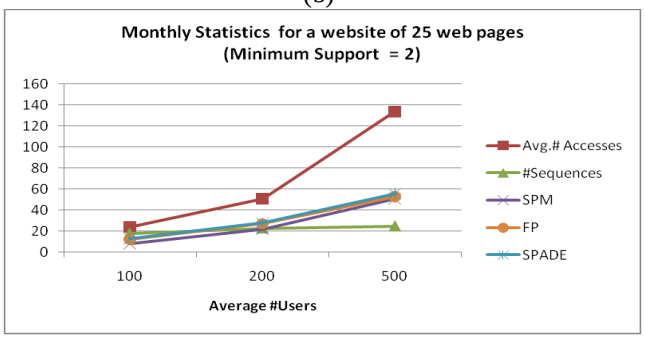
Figure 3 shows the statistics such as average number of users, average of accesses, average number of sequences for a website of 50 pages and the number of FSPs generated by the different FSP using algorithms such as SPADE,SPM and FP growth, for the period that vary from Weekly to Quarterly



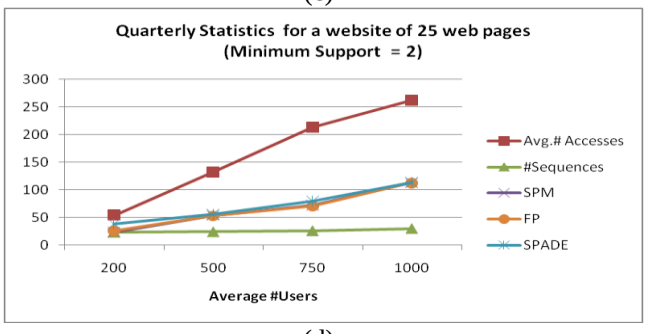
(a)



(b)

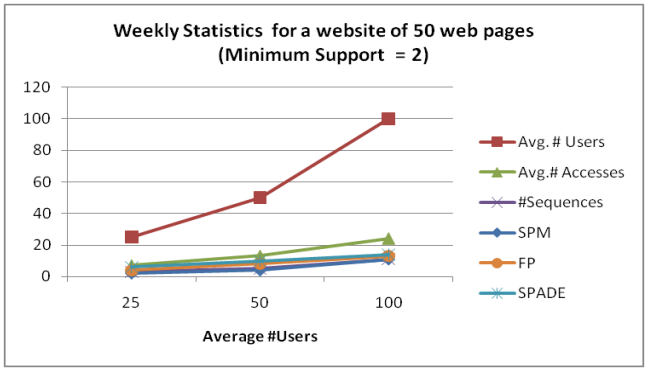


(c)

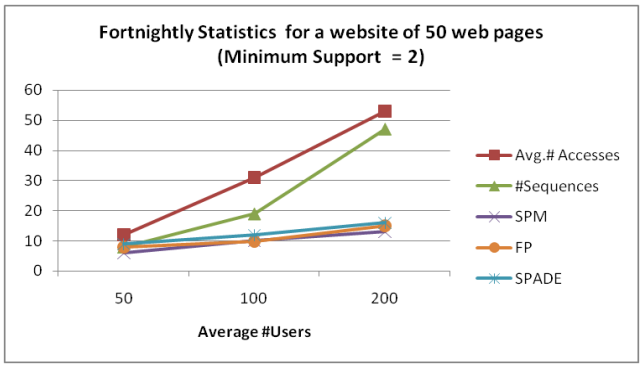


(d)

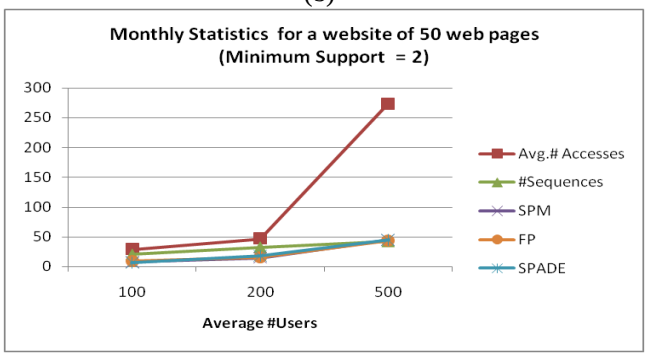
Figure 2. Statistics showing the Average number of Users, Accesses, Sequences for a website of 25 web pages and #FSPs generated by FP, SPADE and SPM algorithms with minimum support= 2 for a period a) Weekly b) Fortnightly c) Monthly d) Quarterly



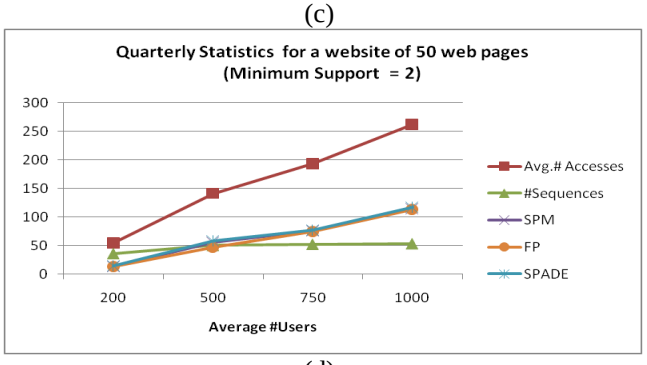
(a)



(b)



(c)



(d)

Figure 3. Statistics showing the Average number of Users, Accesses, Sequences for a website of 50 web pages and #FSPs generated by FP, SPADE and SPM algorithms with minimum support= 2 for a period a) Weekly b) Fortnightly c) Monthly d) Quarterly

It is observed from Figure 2 and Figure 3 that SPADE FSPs algorithm performs better than the other two methods in terms of #patterns generated. Also observed that, compare to weekly statistics, fortnightly statistics gives the better results and as the #users increases , the #accesses also increases. resulting in better number of sequential patterns and lead to better number of Prefetching rules.

After generating the FSPs, we derive the Prefetching rules using association rule generation concept. The Prefetching rules generated from each FSP algorithms for a minimum support that varies from 2% to 3% and the minimum confidence threshold value is 2 indicating the particular page should occur at least 2 times. Using association rule, we generated Prefetching rules from frequent web pages, once the frequent web pages sets from transactions in database have been found. It is without delay to generate strong Prefetching rules from database, where strong Prefetching rules satisfy both minimum support and minimum confidence threshold, it is based on the transactional data for all web pages. Let take l={p_1,p_9,p_{24}} and generate Prefetching rules for non empty sets l are {p_1,p_9}, {p_1,p_{24}} {p_9,p_{24}} {p_1} {p_9} {p_{24}}. if the minimum confidence threshold is say 70% above, then only the rec.no 1,2,4,and 6 Prefetching rules are the outputs, these 100% shows that strong occurrences. Table 8 shows the Prefetching rules generated for academic web site of 50 pages.

Table 8. Prefetching rules for Academic web site of 50 pages

rec.no	Rules	Support	Confidence
1	P1,p9 ->P24	0.1	100%
2	P1,p24-> P9	0.1	20%%
3	P9,p24->P1	0.1	100%
4	P1-> P9,P24	0.1	100%
5	P9 -> P1,P24	0.2	66%
6	P24 -> P1,P9	0.1	66%

Conclusion

The Web sequential patterns using different FSP mining algorithms to generate Web Prefetching rules has been presented in this paper. Sequential Pattern Mining techniques based on Apriori require multiple scans of the sequence database and generate huge number of candidate sets for long web access sequences. This problem makes these algorithms ineffective and inefficient in dealing with large sequence sets with long sequences, especially with web logs. Among all, SPADE algorithm has more advantages like speed, less database scans and high performance. Prefetching rules are then generated based on FSPs. Further, the Prefetching rules generated using proposed algorithm are used in effectively reducing the web latency.

REFERENCES

[1]. Jinlin Chen, (2010) "An UpDown Directed Acyclic Graph Approach for Sequential Pattern Mining", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 7, pp. 913-928.

[2]. Ding-An Chiang, Cheng-Tzu Wang, Shao-Ping Chen. & Chun-Chi Chen, (2009) "The Cyclic Model Analysis on Sequential Patterns", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 11, pp. 1617 – 1628.

[3]. Yu Hirate. & Hayato Yamana, (2006) "Generalized Sequential Pattern Mining with Item Intervals", *Journal of Computers*, Vol. 1, No. 3, pp. 51-60.

[4]. Mohammed J. Zaki, (2001) "Spade: An Efficient Algorithm For Mining Frequent sequences", *Machine Learning*, Vol. 42, pp. 31-60.

[5]. Zhenglu Yang. & Masaru Kitsuregawa, (2005) "LAPIN-SPAM: An Improved Algorithm for MiningSequential Pattern", *IEEE International Conference on Data Engineering Workshops*, pp. 1222-1226.

[6]. Zhenglu Yang, Yitong Wang. & Masaru Kitsuregawa, (2007) "LAPIN: Effective Sequential Pattern Mining Algorithms by Last Position Induction for Dense Databases", *International Conference on Database systems for advanced applications*, pp. 1020-1023.

[7]. Jianyong Wang, Jiawei Han. & Chun Li, (2007) "Frequent Closed Sequence Mining without Candidate Maintenance", *IEEE Transactions on Knowledge and Data Engineering*, Vol.19, No. 8, pp. 1042-1056.

[8]. Kuo-Yu Huang, Chia-Hui Chang, Jiun-Hung Tung. & Cheng-Tao Ho, (2006) "COBRA: Closed Sequential Pattern Mining Using Bi-phase Reduction Approach", *International Conference on Data Warehousing and Knowledge Discovery*, pp. 280-291.

[9]. Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal. & Mei-Chun Hsu, (2004) "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 11, pp. 1424-1440.

[10]. Xiaorong Cheng. & Hong Liu, (2009) "Personalized Services Research Based On Web Data Mining Technology", IEEE International Symposium on Computational Intelligence and Design, pp. 177-180.

[11]. Ford Lumban Gaol, (2010) "Exploring The Pattern of Habits of Users Using Web Log Sequential Pattern", *IEEE International Conference on Advances in Computing, Control and Telecommunication Technologies*, pp. 161-163.

[12]. Jian Pei, Jiawei Han, Behzad Mortazavi-asl. & Hua Zhu, (2000) "Mining Access Patterns Efficiently from Web Logs", *Pacific-Asia Conference on Knowledge Discovery and Data Mining Current Issues and New Applications*, pp. 396-407.

[13]. Tan Xiaoqiu, Yao Min. & Zhang Jianke, (2006) "Mining Maximal Frequent Access Sequences Based on Improved WAP-tree", *IEEE International Conference on Intelligent Systems Design and Applications*, pp. 616-620.

[14]. Sen Yang, Jiankui Guo. & Yangyong Zhu, (2007) "An Efficient Algorithm for Web Access Pattern Mining", *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 726 – 729.

[15]. Lizhi Liu. & Jun Liu, (2010) "Mining Web Log Sequential Patterns with Layer Coded Breadth-First Linked WAP-Tree", *IEEE International Conference on Information Science and Management Engineering*, pp. 28-31.

[16]. V. Mohan, S. Vijayalakshmi . & S. Suresh Raja, (2009) "Mining Constraint-based Multidimensional Frequent Sequential Pattern in Web Logs", *European Journal of Scientific Research*, Vol. 36, No. 3, pp. 480-490.

[17]. Hai-yan Wu, Jing-jun Zhu. & Xin-yu Zhang, (2009) "The Explore of the Web-based Learning Environment based on Web Sequential Pattern Mining", *IEEE International Conference on Computational Intelligence and Software Engineering*, pp. 1-6.

[18]. Bhupendra Verma, Karunesh Gupta, Shivani Panchal. & Rajesh Nigam, (2010) "Single Level Algorithm: An Improved Approach for Extracting User Navigational Patterns to Technology", *International Conference on Computer & Communication Technology*, pp. 436-441.

[19]. Olfa Nasraoui, Maha Soliman, Esin Saka, Antonio Badia. & Richard Germain, (2008) "A Web Usage Mining Framework for Mining Evolving User Profiles in Dynamic Web Sites", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 20, No. 2, pp. 202-215.

[20]. Arthur Pitman. & Markus Zanker, (2010) "Insights From Applying Sequential Pattern Mining To E-Commerce Click Stream Data", *IEEE International Conference on Data Mining Workshops*, pp.967-975.

[21]. Florent Masseglia, Maguelonne Teisseire. & Pascal Poncelet, (2002) "Real Time Web Usage Mining with a Distributed Navigation Analysis", *International Workshop on Research Issues in Data Engineering*, pp. 169-174.

[22]. Baoyao Zhou, Siu Cheung Hui. & Kuiyu Chang, (2004) "An Intelligent Recommender System using Sequential Web Access Patterns", *IEEE Conference on Cybernetics and Intelligent Systems*, pp. 393-398.

[23]. Show-Jane Yen, Yue-Shi Lee. & Min-Chi Hsieh, (2005) "An Efficient Incremental Algorithm for Mining Web Traversal Patterns", *IEEE International Conference on e-Business Engineering*, pp. 274-281.

[24]. Zhennan Zhang, Xu Qian. & Yu Zhao, (2008) "Galois Lattice for Web Sequential Patterns Mining", *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, pp. 102-106.

[25]. Dr. Suresh Jain, Ratnesh Kumar Jain. & Dr. R. S.

Kasana, (2009) "Efficient Web Log Mining using Doubly Linked Tree", *International Journal of Computer Science and Information Security*, Vol. 3, No. 1, pp. 1-5.

[26]. Dhirendra Kumar Jha, Anil Rajput, Manmohan Singh. & Archana Tomar, (2010) "An Efficient Model for Information Gain of Sequential Pattern from Web Logs based on Dynamic Weight Constraint", *IEEE International Conference on Computer Information Systems and Industrial Management Applications*, pp. 518-523.

[27]. Xiaogang Wang, Yan Bai. & Yue Li, (2010) "An Information Retrieval Method Based on Sequential Access Patterns", *IEEE Asia-Pacific Conference on Wearable Computing Systems*, pp. 247-250.

[28]. Kanak Saxena. & Rahul Shukla, (2010) "Significant Interval and Frequent Pattern Discovery in Web Log Data", *IJCSI International Journal of Computer Science Issues*, Vol. 7, No. 3, pp. 29-36.

[29]. Diamanto Oikonomopoulou, Maria Rigou, Spiros Sirmakessis. & Athanasios Tsakalidis, (2004) "Full-Web Prediction based on Web Usage Mining and Site Topology*", IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 716-719.

[30]. Rajimol A. & Raju G, (2011) "Mining Maximal Web Access Patterns- A New Approach", *International Journal of Machine Intelligence*, Vol. 3, No. 4, pp. 346-348.

[31]. J. Han, J.Pei, and Y. Yin," Mining Frequent Patterns without Candidate Generation," Proc. 2000 ACM-SIGMOD Int'l Conf. Management of Data (SIGMOD '00),pp.1-12, May 2000.