TBCSM- An Algorithmic Approach for Consecutive Access Pattern

Abstract

A TBCSMA (Time Base Contingent Sequence Mining Algorithm) is proposed to mine access patterns which are both periodic and sequential from a given database transaction. This new methodology has the ability to mined periodic sequential access patterns-it has an enhanced efficiency aspect to it, especially in terms of small support threshold and huge database. The proposed method work provides the predictive power to anticipate the use of the web page in a stipulated time period with the help of periodic successive access patterns which we incorporate as architecture of web recommendation system.

Keywords: WAP, TBCSMA, Periodic Pattern Mining.

Aarti Pandey
Researcher,
Deptt.of Computer Science, Awadhesh Pratap Singh University,
Rewa M.P.

Ajitesh S. Baghel
Researcher,
Deptt.of Computer Science,
Awadhesh Pratap Singh University,
Rewa, M.P.

Rakesh Kumar Katare
Researcher,
Deptt.of Computer Science,
Awadhesh Pratap Singh University,
Rewa, M.P.

Introduction

Data mining comprises of removing data from large dataset and put away in databases to comprehend the data or potentially take choices. The absolute most crucial data mining assignments are bunching, characterization, anomaly investigation, and pattern mining. Pattern mining comprises of finding fascinating, helpful, and unforeseen patterns in databases different sorts of patterns can be found in databases, for pattern, visit item sets, affiliations, sub diagrams, sequential guidelines, and occasional patterns. A consecutive Web access pattern is a line pattern in a set of web logs, which is sought by clients.

Review of Literature

The greater part of the past examinations for finding sequential patterns, for example, Apriori All and GSP are basically in light of the Apriori Algorithm [10] experience an indistinguishable issue that require expensive multiple outputs of databases with a specific end goal to figure out which of the hopefuls are really visit. Web log mining, intends to find intriguing and visit client access to patterns from the data gotten from the cooperation's of clients while surfing the Web. The web perusing data can be put away in web server logs, intermediary server logs or program logs. The mined learning would then be able to be utilized as a part of numerous down to earth applications, for example, enhancing the outline of sites, investigating client practices for customized administrations, and creating versatile sites as per diverse utilization situations.

Pei **et al, 2004 [8,9]** has proposed a novel and exceptionally compacted data structure known as Web Access Pattern Tree (or WAP-tree) which depends on the FP-tree structure to enhance the effectiveness of consecutive pattern mining algorithms,. The WAP-tree structure encourages the improvement of novel algorithms for mining consecutive access to patterns effectively from an extensive arrangement of web log pieces. The re-construction of intermediate conditional WAP-trees during mining is also costly as described earlier. Currently, there are a few extensions of the WAP-tree and the corresponding algorithms for mining sequential access patterns. The successive access patterns are also mined with periodic successive pattern.

Periodic Consecutive Access Pattern Miining

The concept of periodic pattern is depends on the notation of time period or time slot or time granules. A period Or Time granules is passed time between two occurrences of patterns [1, 2]. It can be counted in terms of a number of transactions which shows the length of periods or in terms of elapsed time. Let consider a transaction labeled T1 to T7 and its retail database can be represented by all kind of data. Let list of items can be represented as a set of item symbols and its transactions are ordered by time is shown below in table 1.1:

Table 1.1- Lengths Patterns of Transaction Database

TID	Transaction	P_1 {u,w}	P_2 {u,w}	P_3 {u,w}	P_4 {u,w}	P_5 {u,w}
T1	{u, w}	1				
T2	{w}		2			
T3	{u,v,w,x,y}					
T4	{v,w,x,y}			2		
T5	{u,w,x}				1	
T6	{u,w,y}					1
T7	{v,w,y}					

The periodic pattern can be determined by the periods that can count length of periods in terms of number of transactions. For the item sets {u, w} has five periods and each period represents the length period determines as number of transactions. The first period of {u, w} comes before the first occurrences of {u, w}. It is assumed that period is of 1 length because it appears in the first of the transaction of database. The periodic pattern can be determined by the periods that can count length of periods in terms of number of transactions. For the item sets {u, w} has five periods and each period represents the length period determines as number of transactions. The first period of {u, w} comes before the first occurrences of {u, w}. It is assumed that period is of 1 length because it appears in the first transaction of database. The second period of {u, w} represents the gap between first appearance and second appearance of {u, w} which is in transaction T1 and T3. Thus length of period is 2 is of 2 transaction. The third period of {u, w} represents the gap between second appearance (transaction T3) and third appearance (transaction T5) of {u, w}. Its length period is 2.

The length period is 1 for fourth period of {u, w} which is gap between third appearance (transaction T5) and fifth appearance (transaction T6) of {u, w}. Now finding the fifth time period is interesting as it is the passed time between the last appearances (transaction T6) of {u, w} and last Transaction (T7). So the list of length period for item set {u, w} is 1, 2, 2, 1, 1. Thus a pattern is said to be periodic if it has min support is number of transaction set which is defined by the user.

As a continuation of the previously mentioned, Calendar patterns was proposed, this aided in the simple and better comprehension of transient affiliation rules. The preferred standpoint is that the work has fewer requirements for learning of data Apriori. The main essential for the previously mentioned is a pattern which depends on timetable which alludes to a specific date-book composition.[2,3]

Aim of the Study

The proposed TBCSM algorithmic approach has been applied to a web prescribed model that give personalized web services to access web pages that related more efficiently and effectively. The purpose of this model is to get future prediction of the web pages that are more same accessed by current user in future. This new methodology has the ability to mined periodic sequential access patterns-it has an enhanced efficiency aspect to it, especially in terms of small support threshold and huge database. It provides the predictive power to anticipate the use of the web page in a stipulated time period with the help of periodic successive access patterns which we incorporate as architecture of web recommendation system.

Calendar-Based Recurrent Time Constraints

This area we begin the characterizing new strategy a genuine time idea In the accompanying part, we recommend occasional timetable wearing time limitations is utilized for outline the constant pride .The centennial date-book whittling down time intimidation comprising of schedule steady loss layout and date-book weakening thing.[6,7]

Definition 1.1

Let an attribute D is a finite domain set of consist of positive integer attribute with time granularity in a calendar relational schema. A centennial logbook bases model is positively expressed as:

$CB_{RT} = (D_1\ int_1, D_2\ int_2, \cdots, D_n\ int_n)$.

Where, CBRT is the calendar base recurrent time and $D_1, D_2, \ldots, D_n \in D$ the date-book segments for pattern in a day, week, month, year and so forth are characterized by each D_j the limited interim for the honest to goodness time qualities are given by int_j of D_j for all attribute of positive integer number. End domain Dj is a finite subset has constraint valid i.e. combination of $D_n \times D_{n-1} \times \cdots \times D_1$are valid for Boolean function. A timetable model con notes a structure of date-book segments and real interims of time.

Let consider a calendar schema model can be seen in the format {Year: [2001,2002,···2009] , Month:[1,2,....,12],day:[1,2,....,31]} OR [Week Day[1,7],Time Hour[0,23]}. With constraint valid evaluate <y, m, d> true only if combination gives a valid date. Acknowledge a periodic wearing calendar model CB_{RT} = (D1 int1, D2 int2, …., D_n int n). , a *calendar deterrent pattern* is represented by (int1, int2, int3, ···, int n'), where *int Ij* ' represents a non negative integers set and *int j* ⊂ I_j, **or a wild-card symbol *** is a simple notion which denotes correct legitimate time specifications in I_j.

For example

To represent the real time scenario notion for an instance CBR_T = (*week day* [1, 7], *hour* [0, 23]), there is *PCJ* = (*, {5, 6, 7, 8}) which represents every day early morning time or *CJ* = ({6, 7}, {19, 20, 21}) represent every weekend's evening time. The real-time scenario notions for instance mornings and evenings are considered differently with respect to different people as per their individual interests and activities. Few precious instances of calendar based on CBT = (day-of-week [1, 7], hour [0, 23]) in Table 1.2 below.

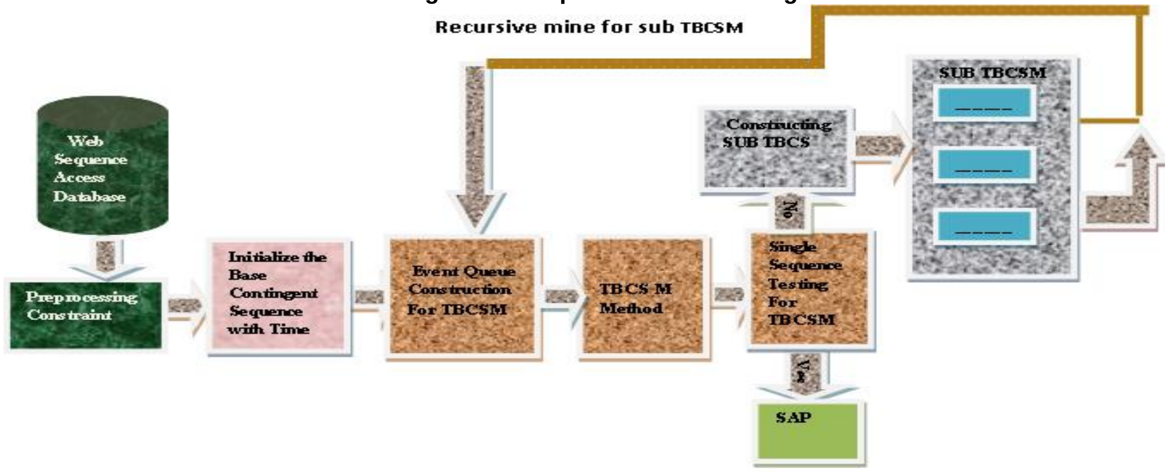
Table 1.2- Few Precious Instances Calendar

Time Concept	Calendar Instances
Early morning	(*,{4,5,6,7})
Morning	(*,{7,8,9,10,11})
Noon	(*,{12})
Afternoon	(*,{13,14………,17})
Evening	(*,{18,19,20,21})
Night	(*,{22,23,0…….,4})
Weekdays	({1,2,………5}),*) /({1,2,………6}), *)
Weekend	({6,7}), *) /({7}), *)

The Tbcsma Approach

TBCSMA (Time Base Contingent Sequence Mining Algorithm) have the ability to mined periodic sequential access patterns-it has an enhanced efficiency of WAP(Web Access Pattern) tree based CS Mine Algorithm [4,5] aspect to it. It employed straightly the conditional successive origin of each frequent event without creating any WAP –Tree. The current research work provides the predictive power to anticipate the use of the web page in a stipulated time period with the help of periodic sequential access patterns. To store the successive data in WAP tree is a highly compressed data structure, so we need to recover the unzipped sequences during mining process in practice that's why our proposed algorithm is not based on WAP tree. In our proposed approach there is no need for reconstruction of large number of intermediate WAP conditional tree.

Figure: 1- Proposed TBCSMA Stages



The figure 1 proposed a TBCSMA Steps depicted below gives a framing of proposed approach which consists of following steps:

A. Preprocessing Constraints
B. Event Queue Construction for TBCSM
C. Single Sequence Testing for Time Base Contingent Successive
D. Constructing Sub contingent sequence Base
E. Recurrent mining for sub contingent Base.

Problem Formulation

A web log is a collection of sequences of access events from one user or session in timestamp ascending order. The Preprocessing tasks as such data cleaning, client recognition, session recognition and transaction recognition can be implemented to the original web log files to obtain the web access transactions.

Working Methods

A sequence of database is taken from web repository then a pre processing constraint with time is defined at initially. Once the origin is constructed from database then a method is created name event construction where a header table for event is created and a queue linked structure is created which is recorded through header table .The irrelevant frequent items are deleted from event. A sub contingent sequence base is constructed for sequence of web access which is tested by method of single sequence testing for time base contingent. Finally the main method of whole TBCSM algorithm is called to mine the consecutive patterns for a given web access sequence database

Preprocessing Constraints

TBCSMA initially origin of conditional sequence is constructed from database of web access sequence which does not meet the need of time recurrent calendar based constraint. An initial contingent successive chain is constructed by using constraints satisfied events (C_{ST}) with time. The init base contingent successive chain and base contingent successive chain is stated as follows:

Definition 1.2

The init TBCS is a set of web access transaction in terms of database of web access transaction catenation which included recurrent time based calendar constraint.

Definition 1.3

The base contingent Web access successive catenation of an event e_i is based on prefix sequence PWAS $_{Prefix}$ (Pattern of web access sequence) denoted as TBCS(C_{ST}) where C_{ST} = PWAS $_{Prefix}$ + e_i is consecutive chain sequence set of event e_i in certain web log data included calendar time constraint.

Let we consider a UASE unique access Successive event described by web resources of different browser i.e. web pages URL and PWAS = pattern for web access sequence. Let a group of consecutive access pattern is PWAS= $e_1, e_2, \ldots, e_k$, e_{k+1}….. en ($e_j \in$ UAE ∀ 1≤j≤n) then the length of PWAS is denoted as length |PWAS|=n. It is not necessary that $e_j \neq e_k$ for j ≠ k in PWAS i.e. repetition of articles is allowed.

Let EWAT is event for web access transaction contains transaction time T and pattern for web access sequence PWAS, where EWAT = (T, PWAS).

The web access chain transaction is fetch from transaction database of particular user which may be single or multiple users i.e. logs of server side .The portend algorithm is not dependent on web log which consist of catenation of web access transaction. It is already described in section 3.2 of consecutive pattern mining. Let assume that WADBTE is a web access database of transaction events where WADBTE= {(T1, S1), (T2, S2).........,(Tm, S $_m$)} where S1,S2,... S $_m$ are sub suffix successive chain of PWAS$_J$, ∀ (1≤j≤m) and Tj represent transaction time for web access database.

Let a calendar period based time constraint is {PC $_T$}.WADBTE (PC $_T$) = {(T_j , PWAS $_j$) | $T_j \sqsubset$ PC,1≤j≤ m} is a set which is included in another set Where PC $_T$ is defined as the length of WATDE. The threshold strain of PWAS in WADBTE is PC_T. The algorithm for pre-processing constraints for transaction events from the web access transaction database EWATDB;

Algorithm

Pre Processing Constraint Algorithm:

Lemma 1

a. PC_T = [*CBPT*, C_J] – A Calendar-Based Time Recurrent periodic Constraint consisting of Model for Calendar Based Model *CBPT* and Calendar Deterrent Example C_J.
b. WADBTE = { WADBTE *i* | WADBTE *j* = (T *j*, PWAS *j*), 1 ≤ *j* ≤ *n*} – Web Access Database Transaction, and WADBTE *j* is a Web Access database Transaction consisting of Time T *j* for Transactions and web access consecutive chain Sequence PWAS .

Proof

1. Assign Init TBCS = Ø.
2. For all WADBTE *j* ϵ WADBTE, if T *j* is included in PC_T, insert PWAS *j* into *Init* TBCS.
3. Return *Init* TBCS.

Result

1: Init TBCS - consecutive Chain Base for Initial Contingent Patterns for WADBTE

Event Queues Construction for Contingent Base Sequence

In this subsection an event queue is constructed to mine the TBCSM algorithm for TBCS (C_{ST}) where C_{ST} =Ø. The tasks perform the following steps:

Conditional Frequent Event is Determined from BCS (C_{ST})

This is the event whose support value not less than from threshold support Min Sup for given base contingent sequence. The formula to identify those event whose support or strain is greater than or equal to min sup

$$\textbf{Strain(S)} = \frac{\left|\{S_j \middle| e_j \in S_j, S_j \in TBCS\ (C_{ST}\}\right|}{|Init - TBCS|} \geq \textbf{Min sup}$$

Designing the Event Queue

A linked list structure for contingent event e_i created called e_i-Queue. Each article of called e_i-queue in sequence of (TBCS (C_{ST}) is labelled e_i. This event queue is recorded in Header table.

Deleting Irrelevant Frequent Event

At last all irrelevant frequent are discarded which are not needed any more in article items of sequences.

The A-queue, B-queue and C-queue are starting for the Header Table. The items labeled as non-frequent events D, E and F in each sequence are deleted. In similar manner, for any subsequent contingent base sequence, using the Constructing Event Queue Algorithm the Header Table and event queues can also be constructed.

Algorithm

Constructing Event Queue Algorithm

Lemma2

1. *Sup Min* – Minimum Strain Threshold.
2. TBCS (C_{ST}) – Transactions Base Contingent Web Access Sequential Chain of C_{ST}.
3. UASE = {e_j|1 ≤ *j* ≤ *n*} ∀ web access successive events in TBCS (C_{ST}).

Result

Event queues and Head Table *HT* along with TBCS (C_{ST}).

Constructing Sub-Contingent Successive Base

The definition for the Sub-Contingent Successive Base sequence is defined as.

Definition 1.7 *TBCS* (PWAS *prefix+ e j*) is known as Sub-Contingent Successive Base sequence of *TBCS* (PWAS *prefix*), if *ej ≠ Ø ∀ ej,* web access transaction event in the Head Table of TBCS (C_{ST}), the Sub-TBCS-Construction algorithm for creating

TBCS (C_{ST} +*ej*) which is based on TBCS (C_{ST})is as shown in Fig:

Algorithm

TBCS-Construction Algorithm

Input

1. TBCS (C_{ST}) - Base Contingent Successive Chain Web Access (Sequence) of C_{ST}
2. e_j –An event in Head Table of TBCS (C_{ST})

Output

1. TBCS (C_{ST} + e_j) - Conditional Web Access Successive Chain (Sequence) Base of ej based on CWSB(STc)

Method

1. Assign TBCS (C_{ST} + e_j)= Ø.
2. ∀ Web access event article sequence in ej-queue of TBCS (C_{ST}), insert its suffix web access sequence into TBCS (C_{ST} + e_j).
3. Return TBCS (C_{ST} + e_j).

Example

For the Init-TBCS, The all suffix of sequences; A by following the A-queue as TBCS (A), which is one of the sub-contingent sequence base of Init-TBCS. The result is TBCS (A) contains {BAC-1, BCAC-1, BA-1, BACC-1}. Note that BAC-1 is the abbreviated from (B-1) (A-1) (C-1).

Single Sequence Testing for Contingent Base Sequence

TB*CS (C_{ST})* can be terminated when all the web access sequences in TB*CS (C_{ST})* are merged to form a single web access sequence. A part of the resultant recurrent successive chain access patterns can be formed using single web access sequence. In contrast, we can also build the Sub-Contingent Successive Base sequence for TB*CS (C_{ST})* and carry out repeated mining. The Contingent Base Sequence Testing Algorithm is for checking the all web access sequences that can be merged to form a single web access sequence and the algorithm is given in below in figure:

Algorithm

Single Contingent-Sequence-Base- Testing-Lemma

1. TB*CS (C_{ST})* – Time Base Contingent Web Access Successive Chain (Sequence) of C_{ST}
2. H_T – Head Table of TB*CS (C_{ST})*

Proof

1. Assign Mono-*Sequence* = Ø.
2. If TB*CS (C_{ST})* = Ø,
   Return
   flag- successful and Unique-*Sequence* = Ø.
3. For *j* = 1 to web access sequences of max length ϵ TB*CS (C_{ST})* Do
   If, all the jth elements in whole web access catenation TB*CS (C_{ST})* are the same advent e.
   And, if absolute enumerate of these advent elements ≥ Min Sup X |Init-
   TBCS|, create another advent element e with the enumeration and insert it
   Into Single-catenation b)
   Else, rebound flag- failed and Single-catenation = Ø.
4. Rebound valid and successful *Single-Sequence.*

Result

1. *outcome: flag- successful,* or flag – *failed*
2. Unique-*Sequence* - single sequence of TB*CS (C_{ST}) .*

The Whole TBCSM Algorithm

The whole TBCSM proposed algorithm for a given web access sequence database to mine the consecutive patterns is shown in fig.

Algorithm

TBCSM Algorithm

Input

1. PC_T = [*CBPT*, *CJ*] – A Calendar-Based Time Recurrent periodic Constraint consisting of Model for Calendar Based Model *CBPT* and Calendar Deterrent Example *CJ.*
2. WADBTE = { WADBTE *i* | WADBTE *j* = (T *j*, PWAS *j*), 1 ≤ *j* ≤ *n*} – Web Access Database Transaction, and WADBTE *j* is a Web Access database Transaction consisting of Time T *j* for Transactions and web access consecutive chain Sequence PWAS *j.*
3. *Sup Min* – Minimum Strain Threshold.
4. *TE* = {*tej*|1 ≤ j ≤ *n*},∀ Access concatenation bond events transaction in WADBTE.

Output

1. *PTSAPE* – is the Periodical Time Successive Access Pattern Events set

Method

1. Assign *PTSAPE* = Ø.
2. Make Use of Pre-Processing-Constraint-Algorithm to construct
   Init- TB*CS (C_{ST})* , C_{ST} = Ø).
3. Make Use of Event-Queue-Construction to build event queues for TB*CS(C_{ST})*.
4. Make Use of Contingent- Base -Sequence-Testing to check single-sequence for TB*CS (C_{ST})*.
   a. If result is successful, insert all ordered combinations of transaction events article in sequence frequent items SF*I* = C_{ST} +*Single-Sequence* to PTS*APE*.
   b. Otherwise, ∀ event in T ej in Head Table of TB*CS(C_{ST})*, use Sub_CWSB_Construction_ Algorithm to build TB*CS(C_{ST}* + t_{ej}). Set C_{ST} = C_{ST} +T*ej* and repeatedly mine TB*CS (C_{ST})* from step3.
5. Return *PTSAPE*.

Performance Evaluation

Let considering the complete periodic consecutive access patterns with C = [(day-of-week [1, 7], hour [0, 23]), ({6, 7}, {20, 21})] and Min Sup = 75% is shown in Table 1.3

Table 1.3 Periodic Patterns Sample Data base

Length of Patterns	Periodic Consecutive Access Patterns
1	a:4, b:4, c:3
2	aa:4, ab:4, ac:3, ba:4, bc:3
3	aac:3, aba:4, abc:3, bac:3
4	abac:3

This fragment talks about the execution of the proposed approach with the customary methodologies for consecutive get to mining of the patterns. In the execution of the TBCSM is contrasted and the ordinary i.e. the conventional form of the web get to example mine calculation i.e. TWAPM (Temporal Web access pattern mine) calculation for repetitive progressive chain gets to examples mining. TBCSM is the best and execution situated calculation which mines the general successive web get to examples utilizing a powerful data structure additionally called as Web Access Pattern (WAP) tree[10,11]. The execution of the WAP mine calculation is speedier than the customary Apriori situated in the request of greatness. To handle the time repetitive timetable based imperatives, the Pre-processing Constraint Algorithm is utilized over TWAPM to get all the limitation fulfilled web get to exchanges from the genuine web get to exchange database. Suppose PCT = [(week-day [1, 7], hour-time [0, 23]), ({6, 7}, *)], speaks to each hour of consistently. Around 22,717 Coercion interest web get to catenation exchanges are utilized for the accomplishment estimation. In order to deal with calendar-based periodic time constraints, the step on Constraint Pre processing is applied to TWAPM for extracting all the constraint satisfied transactions from the original web access transaction database. The Web Access Pattern-tree is then constructed from the constraint-satisfied transactions, and the Web Access Pattern mine algorithm is used to mine the periodic sequential access patterns.

Web Prescribed Model

The proposed TBCSM algorithm has been applied to a web prescribed model that give personalized web services to access web pages that related more efficiently and effectively. The purpose of this model is to get future prediction of the web pages that are more same accessed by current user in future. J. Konstanz et al. (1997) and T.W. Yan, (1999) applied most Traditional techniques to web recommendation such as collaborative filtering and C. Shahabi (2001) and B. Mobasher (2001) applied hybrid approaches that have major drawback for proxy server user to visit the websites and identities of such users are hidden and difficult to achieve. Association rule mining and clustering are some of Web usage mining techniques have also been applied for prescribed web model in last few coming years. Consecutive access pattern for prescribed web model is slightly different from existing web recommended techniques.

The theoretical model architecture of the prescribed system is - First, all users' web access activities of a website are recorded by the Web server of the website and stored into the Web Server Logs. The TBCSM Algorithm component is then applied to mine the sequential access patterns from the Web Server Logs off line. The Pattern-tree Construction component constructs the recommendation model or the Pattern-tree from the mined sequential access patterns. Both the TBCSMA algorithm and the Pattern-tree construction processes are carried out off-line. When a user visits the website, the user's HTTP requests in the current browsing session are recorded in order, and the current access sequence is constructed. Matching the user's current access sequence from the recommendation model of the Pattern-tree, the Recommendation Rules Generation component will generate recommendation rules. The recommended links will then be inserted into the current requested page dynamically.

Evaluating the Performance of TBCSM Algorithm

The two algorithms, TBCSM and TWAPM, were performed on a 2.40 GHz Intel (R) Core TM (i3)6006u CPU 2.0 GHZ 64 bit operating system x64 based processor with 4.00 GB memory, running on Microsoft Windows 10 Pro. One sample datasets are used to test the two algorithms. The offline Dataset_1 is the real web server logs collected from the server log file dated from 19 February 2004 until 13 March 2004 has been selected for further analysis. The server log files are retrieved from the UUM Educare server. The total amount of the server log file between that duration is about 650 MB and the large amount of data becomes the most challenging problem to handle during the Data Pre processing phase. This dataset contains a total of 6,828 web access sequences, with each sequence containing from 1 up to 140 records from a total of 57 categories. Only 5,175 web access sequences of the Dataset_1 have more than one article were used for the measurement. Figure 2 shows the run time of the TWAPM increases sharply, when the support threshold decreases, and the TBCSMA always costs less time than the TWAPM.

Figure: 2 - Run Time of TBCSMA Over TWAPM

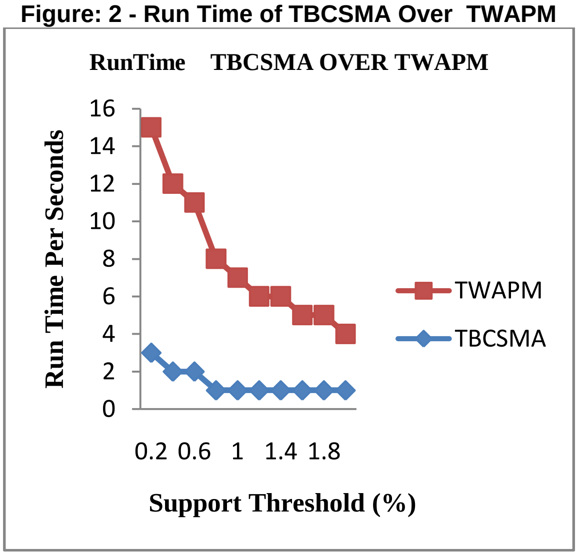
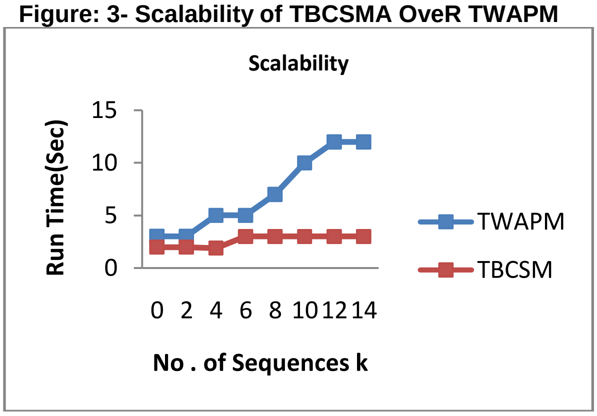


Figure: 3- Scalability of TBCSMA OveR TWAPM



In the figure 3, we have measured the scalability of the two algorithms with respect to different sizes of the constraint satisfied web access sequences. We use different databases (5,000 to 21,717 constraint-satisfied web access sequences with sizes variation) with a fixed support threshold (0.2%). This result shows that the TBCSMA has better scalability than the TWAPM while the size of input database becomes larger.

Conclusion

The performance of TBCSMA is compared with Temporal Web Access Pattern mine (TWAPM) algorithm for mining periodic sequential access patterns. Thus TBCSMA is one of the most efficient algorithms that mine common sequential access patterns from a highly compressed data structure known as Web Access Pattern-tree. The performance of the proposed TBCSMA has been evaluated in comparison with the TWAPM algorithm. When the support threshold becomes smaller and the number of web access sequences gets larger the TBCSMA performs much more efficient than the TWAPM algorithm.

References

1. *Berendt, B., Mobasher, B., Nakagawa, M. and Spiliopoulou, M. (2002) "The Impact of Site Structure and User Environment on Session reconstruction in Web Usage Analysis," In Proceedings of the Forth Web KDD 2002 Workshop, At the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'2002), Edmonton, Alberta, Canada, pp.1-13,2002.*
2. *Fournier P. V., Lin J.C.W. and et al. (2017),"A Survey of Sequential Pattern Mining" ,Data Science and Pattern Recognition ,c 2017 ISSN XXXX-XXXX, Ubiquitous International Volume 1, Number 1, February 2017.*
3. *Lu, Y., and Ezeife, C. I. "Position Coded Pre-Order Linked WAP-tree for Web Log Sequential Pattern Mining", In proceedings of the seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining, Seoul, Korea, April 30-May 2, 2003, published in LNCS by Springer Verlag, pp. 337-349, 2003.*
4. *Masseglia, F., Poncelet, P. and Cicchetti, R. "An efficient algorithm for web usage mining", Networking and Data Networks Journal (NIS), pp.571-603, 1999.*
5. *Masseglia, F., Poncelet, P. and Teisseire, M. "Incremental mining of sequential patterns in large databases", Data and Knowledge Engineering, Vol. 46, No. 1, pp. 97-121, 2003.*
6. *Pandey Aarti , Pandey Prabhat & Baghel Ajitesh S., "A Comprehensive Literatyre Rview on Sequential Pattern Algorithms in Web Usage Mining", Journal of Advances and Scholarly Researches in Allied Education Vol. XIV, Issue No. 1, October-2017, ISSN 2230-7540.*
7. *Pandey Aarti , Pandey Prabhat & Baghel Ajitesh S., "Architecture in Web Use Mining for Data Pre-Processing", Journal of Advances and Scholarly Researches in Allied Education Vol. XIV, Issue No. 1, October-2017, ISSN 2230-7540.*
8. *Pei, J., Han, J., Mortazavi- Asl, B. and Zhu, H.(2000) "Mining access patterns efficiently from web logs", in PADKK '00: Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications. London, UK: Springer- Verlag, pp. 396-407, 2000.*
9. *Pei, J., Han, J., Mortazavi-Asl, B. and Zhu, H. "Mining access patterns efficiently from web logs. Knowledge Discovery and Data Mining", Current Issues and New Applications (LNCS) Vol. 1805, pp. 396-407, 2000.*
10. *Srikant, R. And Agrawal, R. (1996), "Mining Sequential Patterns: Generalizations and Performance Improvements", In Proceedings of the 5th Int'l Conference on Extending Database Technology: Advances in Database Technology, LNCS 1057, pp. 3-17, 1996.*
11. *Srikant, R. and Yang, Y. "Mining web logs to improve website organization", Proceedings of the Tenth International World Wide Web Conference, Hong Kong, pp. 430-437, 2001.*
12. *Zaki, M.J. "Scalable algorithms for association mining" IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 3, pp. 372-390, 2000.*
13. *Zhao, Q. and Bhowmick, S.S "Sequential Pattern Mining: A Survey", Technical Report Center for Advanced Data Networks, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.*