SemFreqSeq-Miner: Mining Semantic Frequent Sequence Patterns from Web Log File

Ms. R.Rooba, Research Scholar, Research and Development Cell, Bharathiar University, Coimbatore,India, rrooba@gmail.com

Dr. V.Valli Mayil, Principal, Palanisamy College of Arts, Perundurai, Erode, India. vallimayilv@gmail.com

Abstract: The semantic web and web mining are two emerging area in the research trends. Web usage mining is one of the process of web mining. The web usage mining analysis the web server log file and predict some patterns. Frequent Sequence Mining is one of the data mining technology which predicts frequently accessed patterns from the web log file. These frequently accessed patterns are used to identify the browsing behavior of the web users. The proposed work introduced the Semantic Frequent Sequence Miner (SemFreqSeq-Miner) to identify the semantic frequent sequences from the web log data. The proposed Semantic Frequent Sequence Mining supports to integrate the detailed semantic knowledge of the web pages to identify the semantic frequent patterns. This method introduced two steps to predict the semantic frequent patterns. The first step scan the input web log file and construct the SemFreqSeq- Tree. The second step mines the SemFreqSeq-Tree and extracted the semantic frequent sequence patterns from it. Thus our system SemFreqSeq-Miner identifies the semantic frequent sequence patterns from the web log files.

The SemFreqSeq-Miner has the ability to support the dynamic updations in the web log file. While the new sequences are added into the web log file the system supports to identify the revised semantic frequent sequence patterns without recompiling the entire tree construction process. An experiment was conducted on Bioportal dataset from Semantic web dog food website (http://data.semanticweb.org/usewod/2014/dataset2/ usewod2014. tar) and partially developed data set from Educational Institution website (www.kasc.ac.in). The results are compared and it reveals that the proposed SemFreqSeq-Miner performs by 15% - 20% of accuracy compared with other traditional model.

Keywords— Web Usage Mining, Web Log preprocessing, Frequent Patterns, Non Frequent Patterns, Semantic Data, Frequent Sequence Patterns.

I. INTRODUCTION

World Wide Web is one of the widespread data resources. Web mining is the process which analyses the web data. There are three different types of web mining technologies Web Content Mining, Web Structure Mining and Web Usage Mining [1]. Web Content mining is the techniques that help the users to identify web documents that meet a certain criteria. Web structure mining is used to rank search engines results by analyzing in-links and out-links of web pages. Web Usage Mining analyses the web server log data and predict the browsing behavior of the web users. Web usage mining consists of three processes such as Web log preprocessing, Pattern Discovery and Pattern Analysis. Web log preprocessing clean the log data collected from various sources. Pattern discovery identifies the interesting patterns from the cleaned web log data using statistical methods as well as data mining methods such as association rules, sequential patterns, cluster analysis and classification rules. Pattern Analysis process analysis the discovered patterns using OLAP tools, query processing and intelligent agent to remove the uninteresting patterns [2].

Now a days sequential pattern mining plays a vital role to identify the frequently accessed patterns from the web log data. Using sequential pattern mining, one can identify the paths that users frequently follow on a web site and hence it increases the prediction rate. Sequential pattern mining is well suited for log study due to the sequential nature of web users' activity [3]. Based on [4] frequent sequences are also called as Traversal Patterns. The traversal patterns can be categorized depends on the following features (1) Is the importance is given to the order of the page visits or not (2) Is the importance is given to the repeated page visits (refresh, reload) or not. (3) Is only the adjacent page visits present in the pattern or any order of page visits are allowed (4) Is the pattern that is not part of another pattern (maximal patterns) is allowed or not.

Commonly discovering sequence patterns from the large data set suffered cost, time and I/O complexity. The candidate generation based frequent sequent mining methods consumes major cost to generate candidate

 DOI : 10.18231/2454-9150.2018.0750

frequent item sets. Many sequential pattern mining methods which uses the Apriori based algorithms generates the candidate item sets [5, 6, 7], it requires multiple scans of the data set to identify the frequent sequence candidate item sets.

To solve this problem Han et, al [8] proposed a frequent pattern growth (FP – growth) algorithm which reduces multiple scan of the data base as well as prune steps of candidate generation. This method is tested on the transactional databases, but the order of page visit is not considered and not support for the sequence mining. To consider the order of page reference in the input transaction the FS- Miner [9] was introduced. The FS- Miner concentrates on the Traversal Patterns. It gives importance to the order of the page visits, repeated page visits and adjacent page visits. But the semantic concept of the page are not considered here.

The proposed methodologies consider the semantic details of the web pages which will produce more accurate semantic frequent sequences compared to the existing methodologies. Our method concentrate on the following issues (1) semantic details of web pages (2) more importances are given to the order of page visits (3) repeated page visits (4) adjacent page visits (5) maximal and non-maximal patterns.

The remaining part of this paper is organized as follows. Section 2 analyses the related work. Section 3 describes about SemFreqSeq-tree construction. Section 4 explains the SemFreqSeq-mining algorithm. Section 5 describes the incremental feature of the proposed methodology. Section 6 shows the experimental analysis of the proposed method. Section 7 concludes this paper Section 8 shows the references cited in this paper.

II. LITERATURE SURVEY

In [10] Semantic Aware Sequential Pattern Mining was introduced. The semantic details in form of semantic distance is used during the pruning process, so that it reduced the search space and minimize the candidate generation as well as reduced the data base scans and support counting process.

The researchers in [11] proposed two algorithms S_P_M and Join_Apriori for sequence data base in which the semantic information from the preprocessor phase is used to prune candidate sequence and reduce the support count value.

The work in [12] used the ontology and semantic network module which represents the domain knowledge of that website. This work used the PLWAP Mine algorithm for generating frequently accessed patterns.

Here [13] a WAP-tree mine algorithm is used to identify the sequential patterns. It shows the working of WAP-Tree. This method assigns the binary code to the nodes in the tree and avoids the repeated scanning of every node. Thus it reduced the mining time.

In [14] the researchers used two methodologies Semantics-Aware Framework and Semantic-Aware PHS to identify the frequent semantic objects and semantic association rules. In Semantic-Aware framework, the cleaned web logs, domain ontology, maximum semantic distance and the semantic matrix are calculated. Then the Semantic Aware PHS is called to generate the frequent objects and semantic association rules.

The work in [15] discussed the CSB-mine algorithm used the lower support count to improve the efficiency of the algorithm. Based on the lower support count the pattern-tree is constructed, and used for online recommendations.

C.I. Ezeife and Y. Lu [16]& [17] in 2005 proposed the WAP-tree to predict the frequent sequences from the web log data. It avoids the repeated construction of the tree during the mining process. The trees are constructed using the pre-order method, then the prefix sequence search is used to mine the tree to identify the frequent sequential accessed

III. METHODOLOGY FOR SEMANTIC FREQUENT SEQUENCE TREE CONSTRUCION

If you are using *Word,* use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). "Float over text" should *not* be selected.

A. Identification of Frequent path and Frequent Sequences:

Consider P = {p1, p2, p3, p4…, pm } be the set of individual pages in a web site. A sequence Seq = < s1, s2, s3…, sn > is collection of page visits with si € P for $1 \leq p \leq n$. A web log file contains collection of sessions. These sessions are stored in the data base DB where each session is stored as a record. Each record is identified using the Record Identifier(RI) and each sequence is called as Session Sequence. In the session sequence when an page p appear immediately after another page pi, it implies there is path from page pi to pi + 1. The sequence can be can be represented as Seq = p-P, where p is the first element in the sequence and P is continuous elements or sub sequences in the session.

For a path T, the support value (Sup^{path} (T)) is the number of occurrences of the path in the input database.

For a Sequence Seq < s1, s2, s3, …, sn > the support value Sup^{seq}(Seq) is the number of occurrences of the sequence in the database either as a full sequence or as a subsequences of sessions.

B. Parameters used to construct Semantic Frequent Sequence Tree:

1. Sematic Similarity Matrix: The semantic similarity matrix is constructed for the web pages present in the web site to identify the semantic similarity between the web pages. The construction of semantic similarity matrix is already explained in our work [17].

The meta data such as keywords, entities , facts, socital tags present the web pages are extracted using the

 DOI : 10.18231/2454-9150.2018.0750

Alchemy API web services and it is stored in RDF data store in a machine readable format. We calculate the semantic similarity between the pages based on the different weights of the semantic components. Since it consists of 'n' different metadata in a web site, a web page is considered as a n-dimensional vector. The weight for different meta data items are calculated by using the Term Frequency and Inverse Document Frequency of meta data items in a web page.

- Term Frequency (TermF) is computed as follows:
  TermF = Frequency of each semantic meta data item in a web page / Total number of semantic meta data items in web page
- Inverse Document Frequency (DocF) is computed as follows:
  DocF= log (Total number of web pages in a website / number of web pages containing semantic meta data item f)
- Weight of meta data item 'w'in a page = TermF * DockF (for each metadata item in a web page)
- Similarity score of two web pages P1 and P2 are found by
  SimScore(P1,P2) = Dot product of weight of all semantic meta data items in pages (P1,P2) / Squared magnitude of vector minus the dot product.

Sim-score will falls between 0 and 1. In this 0 represents no similarity and 1 represents high similarity and 0.50 represents the medium similarity. The threshold mechanism is used to categorize the similarity of two pages for low, high and medium. Sim-average is found to calculate threshold values. Semantic Similarity Matrix (SSM) is calculated by comparing all continuous two pages in a web site.The threshold values are identified depends on the average similarity score value between all web pages in web site. Let P = {p1, p2, p3, p4, …., p5 } be a set of all web pages in a website, the average similarity score value between all pages is calculated as

$$Sim_{avg} = \frac{\sum_{px\ py \in P} Similarity\ (p_x p_y)}{|P|}$$

Based on the average similarity score value, preset threshold for similarity is calculated by using the equation

Minimum Semantic Similarity Support Value (MinSSSV) = 0.50 * Sim_{avg}. Here medium similarity is considered for identifying the semantic frequent sequence paths.

1. **Minimum Semantic Similarity Support Value (MinSSSV**) is the minimum semantic similarity value between the pages in the path to be considered potentially frequent. This parameter is set by the system and not set by the user. The semantic similarity values between the web pages are obtained from the semantic similarity matrix.

2. **Minimum Path Support Value (MinPSV):** Minimum value that a path should satisfy to be potentially frequent. This is obtained by multiplying total number of paths in the data base with minimum path support threshold value (MinPST). This parameter is set by the system and it is used to construct the Semantic Frequent Sequence Tree.

3. **Minimum Path Support Threshold (MinPST):** MinPST is the number of occurrences of the path in the input database to the total number of paths in the database (Sup^{path}(T) / total number of paths in the data base). This parameter is set by the system.

4. **Minimum Sequence Support Value (MinSSV):** Minimum number of times that a sequence needs to appear in the database to be considered frequent. This is obtained by multiplying the total number of paths in the database by the minimum support sequence threshold value (MinSST). This parameter is set by the user. It is used by Semantic Frequent Sequence Miner algorithm

5. **Minimum Sequence Support Threshold (MinSST):** Frequency of the sequence in the data base to the total number of paths in the database(Sup^{seq}(Seq) / total number of paths in the database).

Definitions: Frequent Sequences: Any sequence in the input data base that has Sup^{seq}(Seq) ≥ MinSSV, that sequences are called as frequent sequence or frequent pattern.

Semantic Frequent path: Any path T in the input data base that has a Sup^{path} (T) ≥ MinSSV and Semantic Similarity Score (P_x , P_y) ≥ MinSSSV that path is considered as a semantic frequent path.

Potentially Semantic Frequent Path: A path which satisfy Sup^{path} (T) ≥ MinPSV and Sup^{path} (T) < MinSSV and Semantic Similarity Score (P_x , P_y) ≥ MinSSSV, that path is called as a potentially semantic frequent path.

Non Potentially Semantic Frequent Path: A path which does not satisfy MinPSV and MinSSV and satisfies Semantic Similarity Score (P_x , P_y) ≥ MinSSSV is called as Non Potentially Semantic Frequent Path.

C: Semantic Frequent Sequence Tree Construction

Definition: A Semantic Frequent Sequence tree is a structure that consists the following components:

A **Tree Structure** which consists of root node. Each node in the SemFreqSeq-Tree has children. The nodes in the tree have node label which specifies the page name from the input database sequence. The edges in the tree have edge-label, edge-value and edge-path. Edge-label specifies the form and to nodes that are linked using this edge. The edge-

 DOI : 10.18231/2454-9150.2018.0750

value represents the number of sequences that share this edge in the particular tree path. The tree path is the path that starts from the root node to the current node.

A **Semantic Path Table (SPT)** which stores information about Sematic frequent and potentially semantic frequent paths in the input data base. This table has three fields. Path field specifies the name of the path. Value field stores the number of occurrences of the path in the input database. Semantic Similarity field stores the semantic similarity value between the web pages.

A **Non- Frequent Semantic Path Table (NFSPT)** which consists of information about non –frequent semantic paths in the input data base. This is referred during the implementation of the incremental feature of the system. This table consists of the path, value and RI values.

A **Semantic Similarity Matrix (SSM)**, represents the semantic similarity value between the pages present in the input data base, it is shown below

$$SSM = \begin{bmatrix} & p1 & p2 & p3 & p4 & p5 & p6 & p7 & p8 & p9 \\ p1 & 0.00 & 0.56 & 0.23 & 0.10 & 0.32 & 0.55 & 0.23 & 0.41 & 0.32 \\ p2 & 0.56 & 0..00 & 0.59 & 0.51 & 0.30 & 0.19 & 0.31 & 0.32 & 0.37 \\ p3 & 0.23 & 0.59 & 0.00 & 0.61 & 0.10 & 0.23 & 0.32 & 0.21 & 0.33 \\ p4 & 0.10 & 0.51 & 0.61 & 0.00 & 0.72 & 0.41 & 0.59 & 0.31 & 0.21 \\ p5 & 0.32 & 0.21 & 0.10 & 0.70 & 0.00 & 0.12 & 0.20 & 0.64 & 0.17 \\ p6 & 0.55 & 0.19 & 0.23 & 0.41 & 0.12 & 0.00 & 0.13 & 0.22 & 0.31 \\ p7 & 0.23 & 0.31 & 0.32 & 0.59 & 0.20 & 0.13 & 0.00 & 0.31 & 0.63 \\ p8 & 0.41 & 0.32 & 0.21 & 0.31 & 0.64 & 0.22 & 0.31 & 0.00 & 0.66 \\ p9 & 0.32 & 0.37 & 0.33 & 0.21 & 0.17 & 0.31 & 0.63 & 0.66 & 0.00 \end{bmatrix}$$

Fig.3.1: Semantic Similarity Matrix

RI	Session Sequence
1	$P_4 \rightarrow P_7 \rightarrow P_9$
2	$P_4 \rightarrow P_7$
3	$P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_8 \rightarrow P_9$
4	$P_3 \rightarrow P_4 \rightarrow P_5$
5	$P_3 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_7$
6	$P_3 \rightarrow P_2$
7	$P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_7 \rightarrow P_9$
8	$P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$
9	$P_2 \rightarrow P_4 \rightarrow P_5 \rightarrow P_8 \rightarrow P_9$
10	$P_2 \rightarrow P_4 \rightarrow P_5 \rightarrow P_8$
11	$P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_2 \rightarrow P_6 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3$
12	$P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3$
13	$P_1 \rightarrow P_9 \rightarrow P_3$
14	$P_4 \rightarrow P_9 \rightarrow P_5$
15	$P_9 \rightarrow P_7 \rightarrow P_4 \rightarrow P_2 \rightarrow P_1$

Fig. 3.2: Input Web log File

Path	Value	Semantic Similarity
$P_4 \rightarrow P_7$	4	0.59
$P_7 \rightarrow P_9$	2	0.63
$P_3 \rightarrow P_4$	7	0.61
$P_4 \rightarrow P_5$	6	0.72
$P_5 \rightarrow P_8$	3	0.64
$P_8 \rightarrow P_9$	2	0.66
$P_3 \rightarrow P_2$	5	0.59
$P_2 \rightarrow P_3$	2	0.59
$P_1 \rightarrow P_2$	4	0.56
$P_2 \rightarrow P_4$	2	0.51
$P_6 \rightarrow P_1$	2	0.55

Fig. 3.3 : Semantic Path Table

Path	Value	RI
$P_5 \rightarrow P_2$	1	11
$P_2 \rightarrow P_6$	1	11
$P_5 \rightarrow P_6$	1	12
$P_1 \rightarrow P_9$	1	13
$P_9 \rightarrow P_3$	1	13
$P_4 \rightarrow P_9$	1	14
$P_9 \rightarrow P_5$	1	14
$P_9 \rightarrow P_7$	1	15
$P_7 \rightarrow P_4$	1	15
$P_4 \rightarrow P_2$	1	15
$P_2 \rightarrow P_1$	1	15

Fig.3.4 : Non Frequent Semantic Path Table

Semantic Frequent Sequence Tree Construction: Consider the above Semantic Similarity Matrix, Web log file, Frequent Path Table and Non Frequent Path Table. For the given web log file, assuming Minimum Path Support (MinPSV) = 2 and Minimum Sequence Support Value(MinSSV) = 3 and Minimum Semantic Similarity Support Value (MinSSSV) = 0.5. Now the Semantic Frequence Sequence Tree is constructed as follows:

1) First go through the input web log file and identify the value of each path present in the log file.
2) From the web log file those paths that satisfy , Sup^{path} (T) $\geq$ MinPSV and Semantic Similarity Score (P_x , P_y) $\geq$ MinSSSV are inserted into the semantic path table along with the path occurrence value. The paths those are not satisfy the predefined MinPSV value and satisfy Semantic Similarity Score (P_x , P_y) $\geq$ MinSSSV, that are inserted into the Non frequent semantic path table.
3) Create the Root node for the Semantic Frequent Sequence Tree.

Now again go through the data base and calling the ***insertnode*** function for each input path present the input web log file.

SemFreqSeq – Tree Construction Algorithm
Input : Input Web log file, Minimum Path support Value, Minimun Sematic Similarity Support Value
Output : Semantic Frequent Sequence Tree of input web log file
Method:
Step 1: Go through the input web log file to identify the Number of occurrences for all paths present in the input web log file.
Step 2: insert the paths in to the semantic path table which satisfy Sup^{path} (T) $\geq$ MinPSV and Semantic Similarity

 DOI : 10.18231/2454-9150.2018.0750

Score (P_x , P_y) ≥ MinSSSV

Step 3:Insert the paths that does not satisfy predefined MinPSV value and satisfy Semantic Similarity Score (P_x , P_y) ≥ MinSSSV into the NFPT.
Step 4: Create Root node (R) for Semantic Frequent Sequence Tree
Step 5: For (Each input sessions in web log file get input path)
Call Insertnode (R, input path)
Step 6: Return SemFreqSeq- Tree.

Procedure insertnode (Root node RN, path p-P)
Step 1: if (Path p – P is in SPT)
Step 2: if (RN has Child C and C.node Label = p) {
Step 3: increment RN-C.edge value by 1 }
Step 4: Else { Create Node C with C.Nodelabel= p
Step 5: Create edge RN-C with RN-C.Edgevalue = 1
Step 6: Add edge RN-C to the Semantic path table RN –C }
Step 7: If (P is non empty) { call insertnode(C,P)}}
Step 8: Else if path p –P in NFSPT){
Step 9: IF (P is non-empty) { call insertnode(R,P)}}
Step 10: if P is last page in input path , and input path was not cut , store inputpath.ID (RID) in SessEnd.ID. // (To identify that the path is completed or not)

Figure3. 5: Semantic Frequent Sequence Tree

Compressed Semantic Frequent Sequence Tree: SemFrqSeq–Tree is compressed in three ways.

1) Not all the paths presnt in the web log files are stored in the SemFreqSeq- Tree. Only potentially semantic frequent path are stored in the tree. Non – potential semantic paths are pruned and not stored in the tree.
2) Inserting of the paths in to the tree shares the all possible existing nodes and edges in the tree. This will automatically reduce the tree size and search space.
3) Compared to the tradiational non – semantic frequent tree algorithms SemFreqSeq–Tree reduce more search space because even though the path satisfy the minimum support value , the pages present in the path are not semantically similar means that is not included in the semantic path table.

IV. SEMANTIC FREQUENT SEQUENCE MINER

Depends upon the MinPSV and MinSSV the paths are divided into three types.

1) Semantic Frequent Paths: Paths which satisfy support vale Sup ^{path} ≥ MinSSV ≥ MinPSV and Semantic Similarity Score (P_x , P_y) ≥ MinSSSV are called semantic frequent paths. These paths are represented in the semantic Frequent Sequence Tree and it can be the part of the semantic frequent sequences.

2) Potentially Semantic Frequent Paths: A path which satisfy Sup^{path} (T) ≥ MinPSV and Sup^{path} (T) < MinSSV and Semantic Similarity Score (P_x , P_y) ≥ MinSSSV, that path is called as a potentially semantic frequent path. These paths are stored in Semantic Path Table and are not part of the semantic frequent sequence tree.

3) Non – Potentially Semantic Frequent Paths: Paths with support value Su p^{path} (T) < MinPSV and Semantic Similarity Score (P_x , P_y) ≤ MinSSSV, these paths are stored in NFSPT and not represented in the the SemFreqSeq- Tree.

Only the Frequent semantic paths are part of the semantic frequent sequences. During the mining process only semantic frequent paths are considered.

Features of the Semantic frequent Sequence Tree:

1) The session sequence which contains the non-frequent path(s) is pruned during the construction of the semantic Frequent Sequence tree.
2) If MinPSV < MinSSV, the SemFreqSeq – Tree contains some required details for the mining process.
3) The branches in the semantic frequent sequence tree will provide all possible subsequences that ends with a given frequent path.
4) To extract a sequence that suffix with a certain path P from the semantic frequent sequence tree , it is necessary to examine the branch prefix path that ends with a path (P) backward up to (maximum) the root of the tree.

Semantic Frequent Sequence Tree Mining Process:

We take Minimum Path Support(MinPSV) = 2 and Minimum Sequence Support Value(MinSSV) = 3 and Minimum Semantic Similarity Support Value (MinSSSV) = 0.5 for the mining process.

Step 1: Retrieving the derived paths : Take a path from the Path table with Sup ^{path}(P) ≥ MinSSV, we identify its prefrequent sequences by following the Path P from the path table to edges in the SemFreqSeq- Tree. For each path in the SemFreqSeq- Tree that consists P we extract its prefix paths from this edge to the root node. We call these paths are prefrequent paths.

Step 2: Developing Prefrequent sequence Structure: For the prefrequent sequences of the path P identified in the previous step develop the prefresequent structure for P by splitting the frequency to each prefrequent sequences. Here we remove path P from the end of the each prefrequent sequences.

Step 3: Developing the Prefrequent Sequence Tree : In the Prefrequent sequence Structure for path P, we create a PreFrequent Sequence Tree and insert each of the path from

 DOI : 10.18231/2454-9150.2018.0750

the prefrequent sequence structure of P in a backward manner.

We create necessary nodes and edges. This is called as Prefrequent Sequence Tree.

Step 4: Identifying Semantic Frequent Sequences: Perform the depth first traversal in the Prefrequent sequence Tree structure and return the sequences which satisfying te MinSSV. These sequences are called as semantic frequent Sequences. Now add the path P at the end of the sequence to get the complete semantic frequent sequences.

Semantic Frequent Sequence Algorithm (SemFreqSeq – Miner) Algorithm
// Semantic Frequent Sequences
For all path (P_i in SPT and P_i. value >= MinSSS)
{ // PreSemantic Frequent Sequences
For (all links PL_j in SemFreqSeq-Tree reachable from PT.Ptr(P_i)
{ retrieve PL_j , remove the lastlink, assign PLj. Value = lastpathvalue}
// Pre Semantic Frequent Sequence Tree
Construct PSFS – tree
For (all sequences Seq_l in PSFS –Tree)
{ Concatenate (Seq_L, P_i) } }

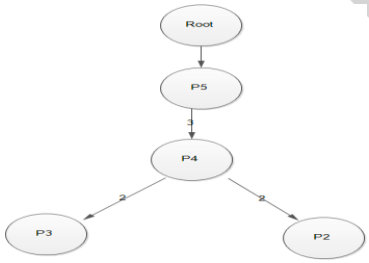
1.Prefrequent Sequences for the Path p5 – p8 in the Path table is show++n below:

Prefrequent Sequence1 : < p3 -p4: 4, p4 - p5: 4, p5 - p8: 1>
Prefrequent Sequence2 : <p2 - p4: 3, p4- p5: 2, p5 - p8 :2 >

2. Prefrequent Sequence Structure :

- For Prefrequent Sequence1 : < p3 -p4: 1, p4 - p5: 1>
- Prefrequent Sequence2 : <p2 - p4: 2, p4- p5: 2 >

3. Prefrequent Sequence Tree



4. Semantic Frequent Sequences for the Path p5 – p8:
Concatenate the Path P5 –P8 at the end of the paths which satisfies the MinSSV.
Semantic frequent Sequence for path < P5- P8: P4 P5 P8 : 3 >

V. ELABORATION FEATURE OF PROPOSED METHOD

The goal of this methodology is the update of the input data bases should be implemented and generating frequent sequences in minimum cost. Elaboration feature of the SemFreSeq-Miner address how to keepthe SemFreqSeq-Tree incrementally without reconstructing the entire tree and how to mine semantic frequent sequences.
This method maintains two parameters Non-Frequent Path Table(NFPT) and RID of the input sequences from the input web log file.

A:Elaborative Nature of SemFreqSeq – Tree:

Elaborative nature of the tree consists of the following steps.
1.SemFreqSeq – Tree supports insertion and deletion of the sequences in to the input web log file. This method takes input as the SemFreqSeq – Tree that represents the input web log file state before update. Then it inserts(or deletes) sequences from the tree.
2. In some situations the tree may be partially reconstructed, at this time some of the branches might be pruned and may be moved from one place to another place in the tree.
3. From the web log file the count of the paths are identified.
4. The path counts in the Path Table(PT) and Non Frequent Path Table (NFPT) are incremented are decremented.
5. The value of MinPSV and MinSSV are updated if necessary.
6. The paths present in the NFPT and which is converted from Non Frequent path to frequent path are transferred to NFPT to PT.
7. Paths in PT which is converted to non frequent paths are transferred from PT to NFPT. (The Paths which does not satisfy updated MinPSV and MinSSV). These paths are no longer present in the SemFreqSeq – Tree. So, we prune them from the tree.
8. For the paths which are transferred from NFPT to PT, we get the RID of that path from the NFPT and get the input session sequences from the original web log file. Insert this new sequence to the SemFreqSeq- Tree. This insertion process compose the sequences that were previously decomposed by the InsertNode Algorithm. Then we insert the remaining subsequences starting from the current node.
9. Now we delete the same subsequences from the top of the tree , if it present.
Now the Elaborated SemFreqSeq- Tree is constructed with out reconstruction of the entire tree structure.

B: Mining the Elaborative SEmFreqSeq – Tree

After the reconstruction of the SemFreqSeq- Tree, mining should be done for the paths in PT, those are updated during the elaboration process. Mining should be done for the following aspects:

1. Mine the paths if the path was in input web log file, or the path was one of the subsequences that were

 DOI : 10.18231/2454-9150.2018.0750

deleted from the tree during the tree reconstruction process.

2. Mine the paths which are converted from Potentially Frequent path to Frequent Path and converted from Non Potentially Frequent Path to Frequent Path.
3. Delete the paths from the tree which are converted from Frequent Paths to Potentially frequent paths and frequent path to Non – FrequentPaths. These links are previously discovered links.
4. Further we did not do any updation for the paths which are in the following categories after elaboration.

i) The paths which are converted from Potentially Frequent Path to Non – Potentially Frequent Path.
ii) The path which are converted from Non – Frequent Paths to Potentially Frequent Paths
iii) The Paths which remains in the Potentially Frequent Paths
iv) The Paths Which remains in the Non Frequent paths.

Now mining process applied to all the paths present in the Updated Path Table (PT) which satisfies the MinSSV value.

VI. EXPERIMENTAL EVALUATION

We take two types of data set. The first data set is the Semantic Web Dog Food Data set which consists of the DBPedia web site log data. It consists of a random selection of 20 days of dbpedia logs from the period July - November 2015.This data set is obtained from http:// data.semanticweb.org/usewod/2014/dataset2/ usewod2014. Tar We collected 5,000 sessions from the log file. These sessions totally containing 24 page references.

The second data set is Educational institute (Kongu Arts and Science College) web site log data recorded in the college server. This log containing data for the period of 3 months period. This log data is preprocessed and then totally 7000 sessions are identified. These sessions containing the 18 page references.

We compare the performance of the proposed method with the existing FS-Miner [9] algorithm where the semantic concepts are not used. And also compare with the Semantics- Aware Sequential Pattern mining [10] where the semantic details are used in form of ontology distances

In Figure 6.1 the new methodology is compared with the existing systems. It shows that when the number of sessions increased the increase in time clearly explained for all the methods.

In Figure 6.2 clearly explain the incremental nature of the system automatically reduces the reconstruction time of the system.

In Figure 6.3 the support values of the existing systems and the proposed methodology are compared.

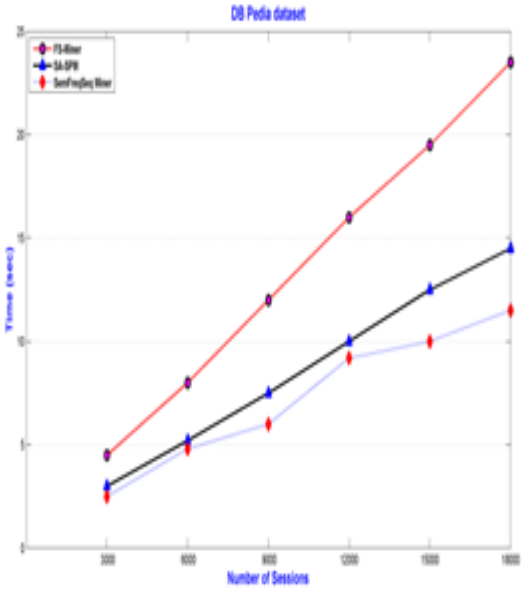


Fig.6.1 Comparison of different number of sessions with time

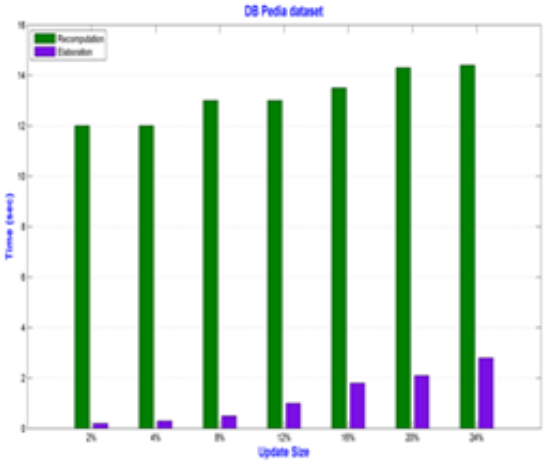


Fig. 6.2 Comparison of incremental feature with reconstruction of the system

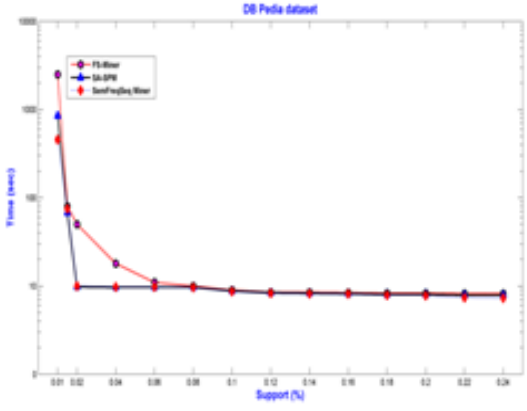


Fig. 6.3 Comparison of different support values of the system with existing system

 DOI : 10.18231/2454-9150.2018.0750

VII. CONCLUSION

The Proposed system developed the SemFreqSeq – Miner. It uses the Semantic details of the web pages to prune the paths in the input web log file. By combining the semantic details in the pruning process it reduces the search space. The Compressed SemFreqSeq – Tree is constructed. From the tree the semantic frequent Sequences are identified. This method also supports the elaboration nature of the data base. When the new input sequence is added in the data base it automatically update the tree without the full reconstruction of the tree. By adding the semantic details prune many paths when handling data with large number of distinct items and give semantically relevant frequent sequences.

REFERENCES

[1] R.Cooley, J. Srivastava, and B.Mobasher, Web mining: Information andPattern Discovery on the world Wide Web. In CTAI, Pages 558 – 567, 1997.

[2] R. Cooley, J. Srivastava, and M. Deshpande, And Tan, P-N(2000), Web Usage Mining: Discovery and applications of usage pattern from web data, SIGKDD Explorations, New York: Springer-Verlag, Pages 12 – 23, 1985, ch.4.

[3] Laura Hollink, Peter Mika, Roi Blano, Web Usage Mining with Semantic Analysis, ACM 978-1-4503-2035-1, May 2013.

[4] M.H.Dunham, Data Mining: Introductory and Advanced Topics, Prentice Hall, 2003.

[5] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In VLDB, pages 487–499, 1994.

[6] R. Ng, L. Lakshmanan, J. Han, and Pang. Exploratory mining and pruning optimization of constrained association rules. In SIGMOD Conf., pages 13–24, 1998.

[7] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: alternatives and implications. In SIGMOD Conf., pages 343–354, 1998.

[8] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In SIGMOD, pages 1–12, May 2000.

[9] Maged EI – Sayed, Carolina Ruiz, and Elke A. Rundensteiner. FS-Miner: Efficient and Incremental Mining of Frequent Sequence Pattern in web logs. ACM 1-5811-978-0/04/0011, Pages 128 – 135, WIDM'04, November 2004.

[10] Nizar R.Mabroukeh and Christie I.Ezeife. Using Domain Ontology for Semantic Web Usage Mining and Next Page Prediction. CIKM'09, Hong Kong China, ACM 978-1-60558-512-3/09/11, Pages 2 – 9, November 2009.

[11] Jypsy Jain, Kapil Sahu.Next Web Page Prediction Using Sematically Enhanced Patterns and markov Mdoel, International journal of Research in Science & Engineering . Volume: 2, Issue:5 pages : 92 – 100.

[12] Thi Thanh Sang Nguyen, Hai Yan Lu, and Jie Lu, Web-Page Recommendation Based on Web Usage and Domain Knowledge,IEEE transaction on knowledge and data engineering ,October 2014.

[13] Yi Lu,C.I.Ezeife,Position Coded Pre-order Linked WAP- Tree for WebLog Sequential Pattern Mining, Springer 2003.

[14] Nu War Has, Semntic Web Usage Mining to Develop Prediction System,International Journal of Computer Applications, Volume 107 – No 13, December 2014.

[15] B. Zhou, S. C. Hui, and A. C. M. Fong, "Efficient sequential access pattern mining for web recommendations", Int. J. Knowl.-BasedIntell. Eng. Syst., vol. 10, no. 2, pp. 155–168, Mar. 2006.

[16] B. Mobasher, "Data mining for web personalization", in The Adaptive Web, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl,Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 90–135.

[17] Ms. R.Rooba, Dr. V.ValliMayil, Semantic Aware Future page Prediction based on Domain Intelligent Markov Model for recommendation, International journal of pure and Appled Mathematics, Volume 118 No.9, 2018, pages 911 – 919.

 DOI : 10.18231/2454-9150.2018.0750