Journal of Advances in Science and Technology
Vol. 15, Issue No. 1, March-2018, ISSN 2230-9659

Algorithmic Web Usage Mining Approach (SSPRA)

Arti Pandey[1]* Rakesh Kumar Katare[2], Ajitesh S. Baghel[3]

Abstract – It is better in case of number of scanning which is compared with various pattern recognition algorithms like full scan. We also found that SSPRA is better and useful for finding the frequent data items that the user is searching for. Implementing web mining techniques on the processed data, should lead to concrete and meaningful suggestions for the betterment of the web site (fully or partially) in order to verify the impact of the changes suggested for improvising the success of the website, the access logs of the re-designed web site should be analyzed.

Keyword: SSPRA

- X -

I. INTRODUCTION

In Web Use mining, Apriori algorithm is an algorithm to learn the association-rules. Apriori is designed to operate on the databases which contain the transactions. Other algorithms are designed for finding association rules in data which do not have any transaction. It is usual for association-rule mining during a given set of things that the formula makes an attempt to seek out subsets, that square measure the intersection to a minimum variety of the item-sets (Srikant and Agrawal, 1996, Srikant and Yang, 2001, Zaki, 2000).

Apriori may be a powerful formula for mining frequent item sets for Boolean association-rules. The name of this formula relies on the very fact that the formula uses pre-knowledge of the frequent item-sets 'properties. Apriori employs an unvaried approach referred to as a level-wise search, wherever k-item sets square measure accustomed explore (a + 1)-item sets. First, the set of frequent 1-itemsets is found. This set is denoted L1. L1 is employed to seek out L2, the set of frequent 2- item sets that is employed to seek out L3, and so on, till no additional frequent k-item sets is found. The finding of every La needs one full scan of the information. In all these the very first step is to find out the maximum forward reference. When a user traverses a website, he or she can go to the hyperlinks which are provided in that particular page. The user clicks on these hyperlinks and move forward. So generally the paths are in the forward manner. But sometimes these paths may be reversed. In other words the user can go to the backward direction and go to another node. In this situation the path which has the maximum length in the forward direction are taken into account.

After calculating the maximal forward references, now there is need to scan the database tables (Pandey, et. al., 2017), (Pandey, et. al., 2017)

II. EXISTING APPROACH

M. Suman (2012) in her research paper suggested that Apriori uses a "bottom up" approach, wherever the frequent things square measure continuing with one item at a time and candidate's cluster square measure tested against this knowledge. This formula gets terminated once there are no additional booming extensions (Berendt, et. al., 2002, Pei, et. al., 2000, Zaki, 2001).

Apriori uses the breadth-first search and a hash tree structure to count candidate item-sets. This formula creates the candidate item-sets of length "a", from the item sets of length "a−1". Then it trims the candidates that have an in-frequent sub-pattern. In keeping with the "downward closure lemma", the candidate-set have all the frequent "a"-length item-sets. After that, it scans the dealing information to see frequent item sets among the candidates. For determinative frequent things quickly, the formula uses a hash tree to store candidate item sets. Apriori, whereas having historical significance, suffers from variety of deficiencies or the trade-offs, that have engendered different algorithms. Many Apriori Algorithms have been proposed by the researchers. Some of them are as follows; and also some of them are explained in the following section

A. Full Scan Approach

Chen .M .Park. J & et al., (1998) (Masseglia, et. al., 1999, Masseglia, et. al., 2003, Chen, et. al., 1998) **in**

his paper suggested a scanning method that scans the database table of the web usage logs. These web logs are collected either from the server or from the client side. The users' website traversal patterns sample is depicted in Figure 1.1.

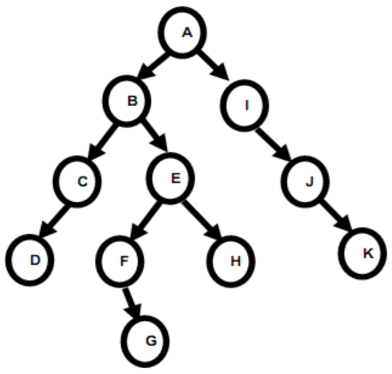


Figure 1.1 Website Structure

For example the paths traversed by a user are given in the tabular form in Table 1.1

Sr. No	Path
1	ABCD
2	ABEFG
3	ABEH
4	AIJK
5	ABC
6	AI
7	AIJ
8	ABE
9	ABEF
10	AB

Table 1.1 Input Paths Traversal

Firstly, I find out the support value. In here we you the word "support value, i.e.; show that the pages which are below this number are less in use. In the other word, it's a threshold value to decide that node can be avoided or not. This value is necessary because the data collected is very large and this data needs to be optimal in size, but to handle such kind of bigger data is very difficult. Now to find out the maximum forward reference, some levels of the refining are needed.

LEVEL-1st

Frequency of the pages are calculated, which are referred by the user for his search. The frequency of the user to access any page is shown in the Table 1.2. Now, check that which node is having the frequency below the threshold support value. These item sets can be discarded for our further process. So, after applying this, the remaining data is shown in the Table 1.3.

Table 1.2 Initial Frequencies at Level 1

S. No.	Page	Frequency
1	A	10
2	B	7
3	C	2
4	D	1
5	E	4
6	F	2
7	G	1
8	H	1
9	I	3
10	J	2
11	K	1

Table 1.3 Frequencies after the minimization

S. No.	*Page*	*Frequency*
1	*A*	*10*
2	*B*	*7*
3	*C*	*2*
4	*E*	*4*
5	*F*	*2*
6	*I*	*3*
7	*J*	*2*

LEVEL-2nd

Now there is a need to make some combinations of these nodes as shown in the Table 1.4.

Further in the process of finding the patterns, these combinations are identified in the database provided in the initial phase.

Table 1.4 Frequency Combinations at the second level

AB
AC
AE
AF
AI
AJ
BC
BE
BF
BI
………
……

After this, the paths which are having the frequency below the support value are deleted out of the analysis. So by doing this comparison, the frequency table is obtained which is shown in Table 1.5.

Table 1.5 Remaining combinations after comparing with the support value

Path	Frequency
AB	7
AI	3
BC	2
BE	4

LEVEL-3rd

Now there is a need to make the combinations again and then comparing them with the initial path table values. After repeating this process once again and removing the path values which are having the frequencies below the support value the obtained table has got the results as shown in the Table 1.6.

Table 1.6 Remaining combinations at the 3rd level

Path	**Frequency**
ABC	2
ABE	4

So, finally there are two paths, which are the contenders of the frequently traversed paths. Among these two paths ABE is containing the higher frequency. So this path is traversed frequently by the user.

B. FP-GROWTH ALGORITHM:

FP-Growth Algorithm is an efficient and scalable way to mine the complete set of frequent-patterns, by using pattern fragment growth, by using an extended prefix-tree structure for storing suppressed and crucial information about the frequent-patterns named frequent-pattern tree (FP-tree). This algorithm allows the discovery of frequent item-set without any candidate item set generation. This is a two-step approach presented by Zhou, B.Y., Hui, S.C., and Fong, A.C.M. (2004), (Lu and Ezeife, 2003, Masseglia, et. al., 1999, Masseglia, et. al., 2003, Zhou, et. al., 2004). These steps are as follows.

Step 1:

1. Create a compact data structure named as FP-Tree.
2. Scan the data and find the support for each item in the raw data table.
3. Calculate the minimum support value.
4. Infrequent items are discarded.
5. Sort the infrequent items in the decreasing order.

Step 2: Construct the FP-Tree.

1. Read the transaction.
2. Create a node in the tree according to their connections.
3. Increase the count of these nodes by one.
4. Take another transaction.

Step 3: Continue until al l the transactions are taken into consideration.

1. To calculate the minimum support value, the minimum percentage of support should be decided, say 20% and the no. of transaction are 10.Minimum Support Value =10*(20/100) = 2
2. After calculating this value, this algorithm is applied and a tree is obtained.

This tree is called as FP-Tree. So at the end this FP-Tree states that which node as well as the path is most frequently visited by the user during the internet surfing. This may help in business decision making and the enhancements[9,10].

III. PROPOSED APPROACH:

In this section an attempt is made to solve our purpose in an efficient way. In this proposed work, the website structure itself is used during the analysis. The website structure means that the connectivity of the web pages that constitute the web data. In this algorithm, this architecture is considered only. This algorithm takes an array named node_path_table as an input. The basic entity must be as follows.

```
Struct node

{

Char x; //for transaction id

Char y[10]; //for storing the items

}
```

C. Proposed Algorithm:

The related algorithm SSPRA (Single Scan Pattern Recognition Algorithm) is as follows:

Algorithm:

Single Scan Pattern Recognition (SSPRA)

Input:

Database, Website structure, node_path_table

Method:

For each transaction in database d, IF transaction t is not starting from the root

Get the first node of the transaction t in S.

Repeat the node path table until s= node table [1].X.

Prefix node path table by [i].y into transaction table t.

// now we have to work on count.

For each node in the transaction t Increase the node value of count by 1.

Result:

Get the node with maximum no. of count and preceding sequence from node_path_table.

Because, this scans the database only once it may be named as Single Scan Pattern Algorithm.

Let us suppose that we have the website structure as shown in Figure 1.1 & Table 1.1 where the transaction id of the user transaction and the path followed by the user is given. The initially website structure shows the way that web pages are connected to each other via the hyperlinks. Now by analyzing the transactions we can get the Table 1.1. This table shows the transaction id and the path that the user followed.

In this algorithm, the first transaction is taken and then that path is followed. Initially each node has count as 0. This count value is increased by one, whenever the node is traversed in between any path. The maximal paths are identified by using the algorithm named maximal forward reference. Initially each node is having the count as 0, as shown in Figure 1.2(a). This is the very initial situation at the log file sources. Here the circles represent the web pages and the links represent the hyperlinks that connect the nodes in the data.

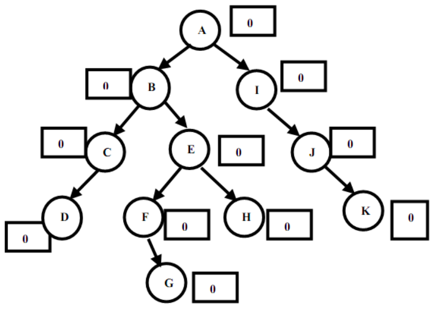


Figure 1.2(a) Initial Situation.

After this, the database table is scanned again. Here the path ABCD is scanned first. This is the path which is traversed during the web site access. Initially, nodes A, B, C and D are having the count value as 0. These nodes are now updated to one, as shown in Figure 1.2(b).

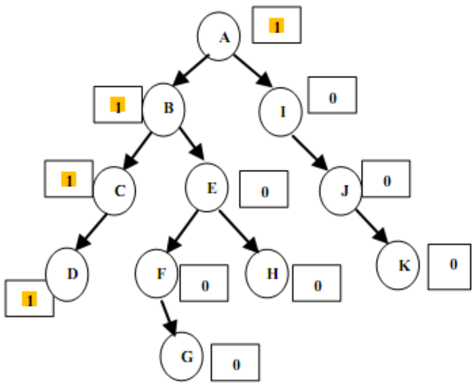


Figure 1.2(b) ABCD path.

Now comes to the path ABEFG, in the reference table. It can be noticed that the nodes A and B are repeated again, so their values are increased to 2, while the nodes E, F and G will have the count value as 1. This situation can be seen in Figure 1.2(c)

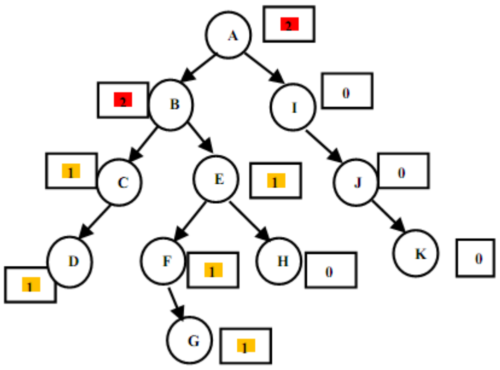


Figure 1.2(c) For ABEFG Path.

The next path is ABEH. Here, the nodes A and B are repeated again, so their values are increased to 3, while the node E, which is also repeated, will have the count value as 2 and count value of H is 1, as shown in the Figure 1.2(d).

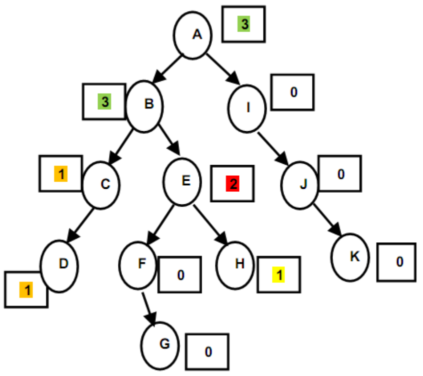


Figure 1.2(d) ABEH Path.

Next is the path AIJK. Here, only the nodes A is repeated again, so it's value is increased to 4, while the node I, J and K are not repeated at all, will have the count value as 1, as shown in the Figure 1.2(e).

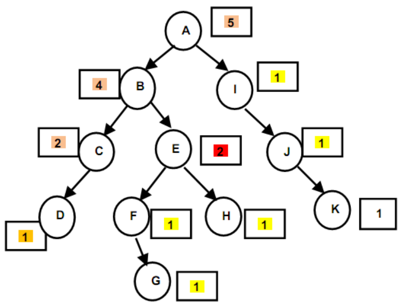


Figure 1.2(e) AIJK Path.

The next path is ABC. Here, the nodes A, B and C are repeated again, so their values are increased to 5, 4 and 2 respectively. There is no node which is repeated. The current situation can be seen in the Figure 1.2(f).

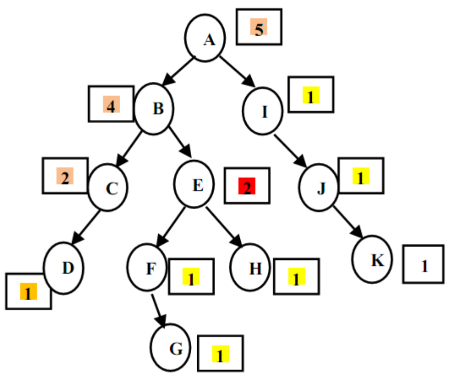


Figure 1.2(f) ABC Path.

Now there is a path AI. In this case A and I both are repeated, so these both will get there count value increased by 1, as shown in Figure 1.10.

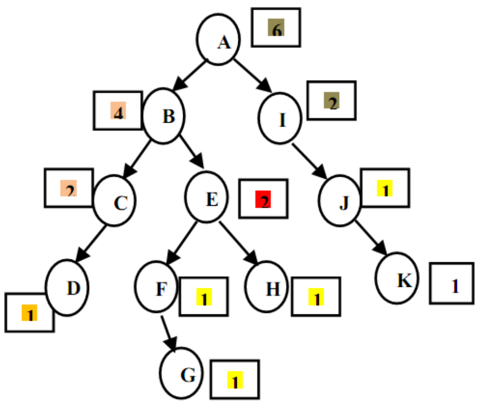


Figure 1.2(g) For AI Path.

In path AIJ, the entire node A, I and J are repeated, so their count value will also get increased by 1. So the count value will be A-7, I-3 and J-2, as shown Figure 1.11.

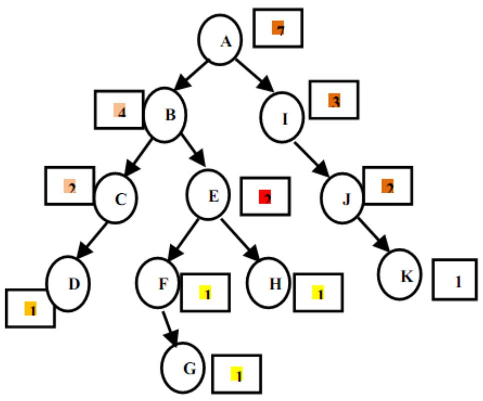


Figure 1.2(h) AIJ Path.

Now there is a path named as ABE. In this path the user goes to A, then B and then E. so the count values will be updated according to Figure 1.3(i).

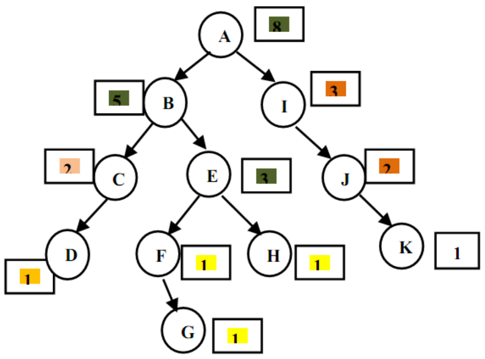


Figure 1.3(i) ABE Path.

ABEF is the path which the user traversed next. The count value of A, B, E and F is increased by one. So the situation is like Figure 1.2(j).

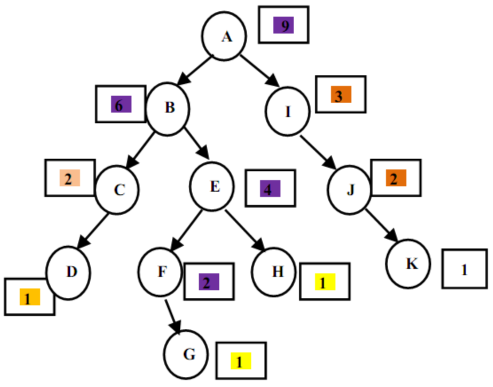


Figure 1.3(j) ABEF Path.

At the last the use traversed the path as AB. The count value of A becomes 10 and the count value of B becomes 7, as shown in Figure 1.2(k).Figure 1.2 (k) analyzes, when all the paths are identified and the reference count of each node. Here the numbers shown are the reference counts based on this, it can be identified that user traversed what kind of products and what kind of improvements are needed in the business layouts, or may recommend users for any suitable purchase.

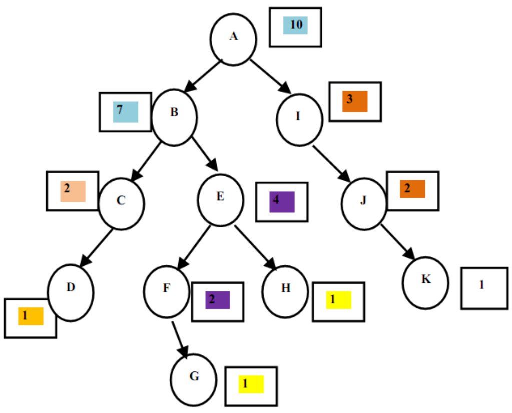
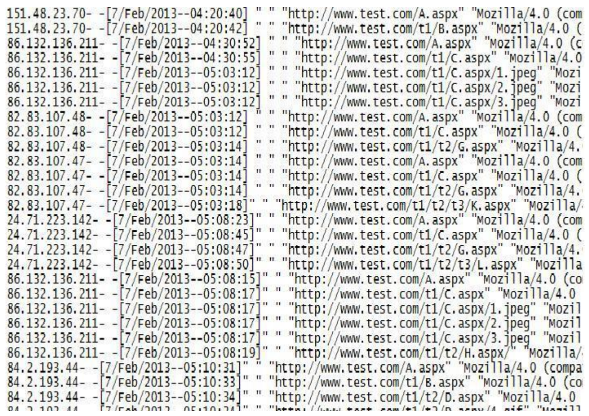


Figure 1.2(k) AB Path.

IV. TESTING AND RESULTS

Now the implementation part is discussed with the testing portion here. The results are also compared by using different-different cases. The log files, as shown in Figure1.4, is the main content of any web usage mining techniques. This file may reside in the server side. This file may be at the client side but in that situation it may be difficult to extract that log file.

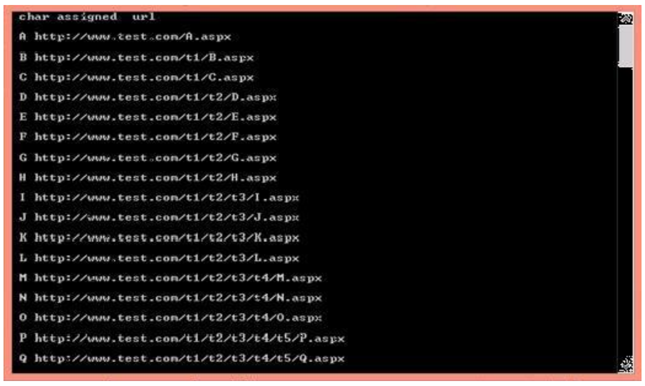
Case 1:



```
151.48.23.70- -[7/Feb/2013--04:20:40] " " "http://www.test.com/A.aspx" "Mozilla/4.0 (com
151.48.23.70- -[7/Feb/2013--04:20:42] " " "http://www.test.com/t1/B.aspx" "Mozilla/4.0 (
86.132.136.211- -[7/Feb/2013--04:30:52] " " "http://www.test.com/A.aspx" "Mozilla/4.0 (c
86.132.136.211- -[7/Feb/2013--04:30:55] " " "http://www.test.com/t1/C.aspx" "Mozilla/4.0
86.132.136.211- -[7/Feb/2013--05:03:12] " " "http://www.test.com/t1/C.aspx/1.jpeg" "Mozi
86.132.136.211- -[7/Feb/2013--05:03:12] " " "http://www.test.com/t1/C.aspx/2.jpeg" "Mozi
86.132.136.211- -[7/Feb/2013--05:03:12] " " "http://www.test.com/t1/C.aspx/3.jpeg" "Mozi
82.83.107.48- -[7/Feb/2013--05:03:12] " " "http://www.test.com/A.aspx" "Mozilla/4.0 (com
82.83.107.48- -[7/Feb/2013--05:03:12] " " "http://www.test.com/t1/C.aspx" "Mozilla/4.0 (
82.83.107.48- -[7/Feb/2013--05:03:14] " " "http://www.test.com/t1/t2/G.aspx" "Mozilla/4.
82.83.107.47- -[7/Feb/2013--05:03:14] " " "http://www.test.com/A.aspx" "Mozilla/4.0 (com
82.83.107.47- -[7/Feb/2013--05:03:14] " " "http://www.test.com/t1/C.aspx" "Mozilla/4.0 (
82.83.107.47- -[7/Feb/2013--05:03:14] " " "http://www.test.com/t1/t2/G.aspx" "Mozilla/4.
82.83.107.47- -[7/Feb/2013--05:03:18]" " "http://www.test.com/t1/t2/t3/K.aspx" "Mozilla/
24.71.223.142- -[7/Feb/2013--05:08:23]" " "http://www.test.com/A.aspx" "Mozilla/4.0 (com
24.71.223.142- -[7/Feb/2013--05:08:45]" " "http://www.test.com/t1/C.aspx" "Mozilla/4.0 (
24.71.223.142- -[7/Feb/2013--05:08:47]" " "http://www.test.com/t1/t2/G.aspx" "Mozilla/4.
24.71.223.142- -[7/Feb/2013--05:08:50]" " "http://www.test.com/t1/t2/t3/L.aspx" "Mozilla
86.132.136.211- -[7/Feb/2013--05:08:15]" " "http://www.test.com/A.aspx" "Mozilla/4.0 (co
86.132.136.211- -[7/Feb/2013--05:08:17]" " "http://www.test.com/t1/C.aspx" "Mozilla/4.0
86.132.136.211- -[7/Feb/2013--05:08:17]" " "http://www.test.com/t1/C.aspx/1.jpeg" "Mozil
86.132.136.211- -[7/Feb/2013--05:08:17]" " "http://www.test.com/t1/C.aspx/2.jpeg" "Mozil
86.132.136.211- -[7/Feb/2013--05:08:17]" " "http://www.test.com/t1/C.aspx/3.jpeg" "Mozil
86.132.136.211- -[7/Feb/2013--05:08:19]" " "http://www.test.com/t1/t2/H.aspx/" "Mozilla/
84.2.193.44- -[7/Feb/2013--05:10:31]" " "http://www.test.com/A.aspx" "Mozilla/4.0 (compa
84.2.193.44- -[7/Feb/2013--05:10:33]" " "http://www.test.com/t1/B.aspx" "Mozilla/4.0 (co
84.2.193.44- -[7/Feb/2013--05:10:34]" " "http://www.test.com/t1/t2/D.aspx" "Mozilla/4.0
```

Figure 1.3 Web Log File Sample.

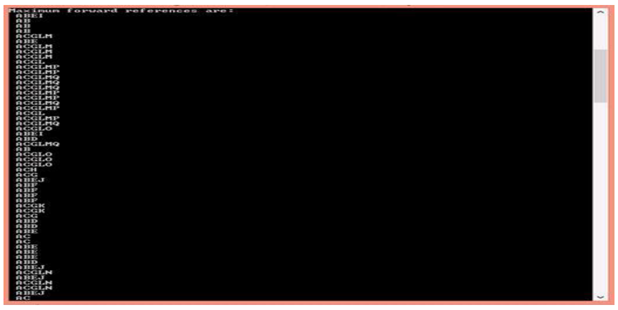
In the Figure 1.5, the various web pages are shown. Here is the list of URL of the web pages are also shown. There may be hundreds of pages in a single website; these are the sub-pages in a single web site. So these are the web pages which are open for the user to be accessed.



```
char assigned  url
A http://www.test.com/A.aspx
B http://www.test.com/t1/B.aspx
C http://www.test.com/t1/C.aspx
D http://www.test.com/t1/t2/D.aspx
E http://www.test.com/t1/t2/E.aspx
F http://www.test.com/t1/t2/F.aspx
G http://www.test.com/t1/t2/G.aspx
H http://www.test.com/t1/t2/H.aspx
I http://www.test.com/t1/t2/t3/I.aspx
J http://www.test.com/t1/t2/t3/J.aspx
K http://www.test.com/t1/t2/t3/K.aspx
L http://www.test.com/t1/t2/t3/L.aspx
M http://www.test.com/t1/t2/t3/t4/M.aspx
N http://www.test.com/t1/t2/t3/t4/N.aspx
O http://www.test.com/t1/t2/t3/t4/O.aspx
P http://www.test.com/t1/t2/t3/t4/t5/P.aspx
Q http://www.test.com/t1/t2/t3/t4/t5/Q.aspx
```

Figure 1.4 List of URL's of the web pages.

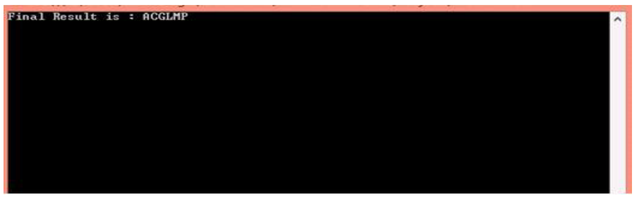
Now there is a need to calculate the maximum forward reference by using the web log files. The algorithm to calculate the MFR is same as the other algorithms uses it. These are shown in the Figure 1.6.



```
Maximum forward references are:
 ABEI
 AB
 AB
 AB
 ACGLM
 ABE
 ACGLM
 ACGLM
 ACGLM
 ACGL
 ACGLMP
 ACGLMP
 ACGLMQ
 ACGLMQ
 ACGLMQ
 ACGLMP
 ACGLMP
 ACGLMQ
 ACGLMP
 ACGL
 ACGLMP
 ACGLMQ
 ACGLO
 ABEI
 ABD
 ACGLMQ
 AB
 ACGLO
 ACGLO
 ACGLO
 ACH
 ACG
 ABEJ
 ABF
 ABF
 ABF
 ABF
 ACGK
 ACGK
 ACG
 ABD
 ABD
 ABE
 AC
 AC
 ABE
 ABE
 ABE
 ABD
 ABEJ
 ACGLN
 ABEJ
 ACGLN
 ACGLN
 ABEJ
 AC
```

Figure 1.5 Maximum Forward References

After employing the algorithm in JAVA, the results obtained are shown in the Figure 6.7.



```
Final Result is : ACGLMP
```

Figure 1.6 Most Frequent Paths that the User Traversed.

This is the path which is traversed by the user most of the time; he or she traverses at the time of searching anything on the internet.

Case 2:

The results can be compared in graphical representation also, as shown in Figure 1.8.

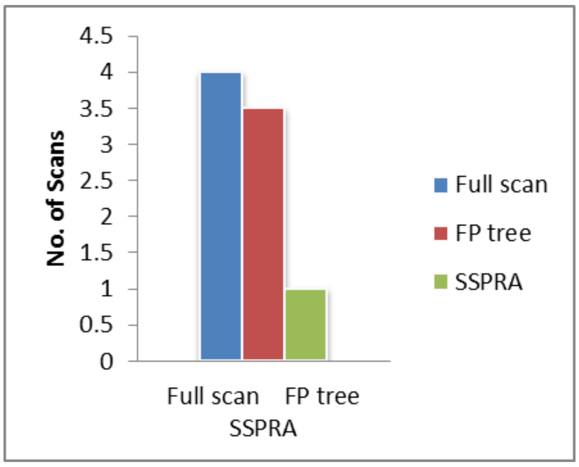


Figure 1.7 Comparison between the existing algorithm and SSPRA.

As it is already said that the proposed algorithm is better in terms of number of comparisons and the number scans. Now say number of level in a tree are N and the tree contains M nodes and the no of branches a node may have at a time is X, then we can have the equation (1).

$N = \log_X M$ (1)

Where, N: no. of levels in a tree

M: no. of nodes in the tree

X: The maximum no. of branches in a tree

This graph tells that in full scan the no of database scans needed are 4. So the no. of nodes in the tree are 81, the levels are 4 and the maximum branches a node can have are 3. In case of selective scan the levels in the tree are 3.67 (about 4), the no. of nodes are 13 (approx.) and the maximum allowable branches that a node may have are 2. For the FP tree algorithm and the SSPRA algorithm the no. of scans are constant i.e. 2 and 1 respectively.

There is no need to think that FP-tree algorithm is far better than any other previous designs, but in this SSPRA algorithm no tree is created. Creating an FP tree becomes an overhead sometimes, especially when there is large sized web structure. But in this proposed algorithm only the web structure is used and we can mine the data to find out the behavior of the user and help in business decision making.

Case 3:

Comparison between other method & the proposed algorithms; as shown in the Table 1.8 the existing algorithms employ tree creation and the candidate generation.

Table 1.8 Comparison between other methods and the proposed Algorithm

Factor	Full scan	Proposed Algorithm
Consecutive Pattern	Perfect for Consecutive pattern	Those hyperlinks are also consider which are not even consecutive
Tree Creation	Needed	Not Needed
Data Base Scan	With each pass database scan need to be done and it is equal to the level of the tree in this algorithm	Single scan needed
Database size	It is efficient for small size database	Both the small and larger size database is handled in this algorithm
I/O Overhead	I/O overhead cost is higher	I/O overhead cost is less than other algorithm
Candidate Generation	Candidate generation is performed	No candidate generation is needed

These two steps are very hard to execute. While in the proposed algorithm there is no need of either candidate generation or the tree creation. The Full Scan algorithm at least *n* number of database scans is required. Here *n* is the level of the tree in these algorithms. While in the proposed algorithm, there is no need to scan the database *n* times at all, only one scan is enough for the analysis. Because there is no need of any kind of tree creation, only one scan is needed. Only the Website structure is used for the purpose of analysis. By using the website structure, each node gets a count variable. Each time the node is traversed; this variable gets increased by one. This is the variable *"count"*, which plays an important role in analysing the patterns. The algorithms which are already in existence suffer from the I/O overhead and induce more cost to the execution. But the proposed algorithm does not suffer from the I/O overhead.

Therefore, it can be inferred from Case 1, Case 2 and Case 3 that the proposed algorithm is better than the existing algorithms such as FP-Tree algorithm, Full Scan algorithm and the Reference Scan algorithm. The numbers of scans in the proposed algorithm are reduced, so the I/O overhead is minimized.

V. CONCLUSION AND FUTURE ASPECTS:

Various pattern recognition algorithms have been discussed and compared with the proposed algorithm. By the comparative Apriori analysis, it can be seen that this SSPRA algorithm is working better than the existing algorithms. In case of scanning it takes only one scan for finding the most frequent data item the user is searching for. SSPRA is better than other algorithms in terms of the number of scans and would be useful for finding the frequent data item, the user is searching for. The work done in the thesis is useful in identifying the frequent patterns. In future, this work can be implemented in the big data environment. The proposed algorithm is suitable for the large data scans. As it is known that in the cloud environment the data becomes so much big. The web hosting is also supported by the clouds. So to analyze the data and get the user behavior this algorithm may get a good place.

REFERENCES

Agrawal R., and Srikant R. (1995). "Mining Sequential Patterns", In Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan, pp. 3- 14.

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.

Chen, M.S., Park, J.S. and Yu, P.S. (1998). "Efficient data mining for path traversal patterns", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No.2, pp.209-221.

Lu, Y., and Ezeife, C. I. (2003). "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.

Masseglia, F., Poncelet, P. and Cicchetti, R. (1999). "An efficient algorithm for web usage mining", Networking and Data Networks Journal (NIS), pp. 571-603.

Masseglia, F., Poncelet, P. and Teisseire, M. (2003). "Incremental mining of sequential patterns in large databases", Data and Knowledge Engineering, Vol. 46, No. 1, pp. 97-121.

Pandey Aarti , Pandey Prabhat & Baghel Ajitesh S. (2017). "A Comprehensive Literatyre Rview on Sequential Pattern Algorithms in Web Usage Mining", *Journal of Advances and Scholary Researches in Allied Education Vol. XIV, Issue No. 1, October-2017, ISSN 2230-7540.*

Pandey Aarti , Pandey Prabhat & Baghel Ajitesh S. (2017). "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.*

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.

Pei, J., Han, J., Mortazavi-Asl, B. and Zhu, H. (2000). "Mining access patterns efficiently from web logs. Knowledge Discovery and Data Mining", Current Issues and New Applications (LNCS) Vol. 1805, pp. 396-407.

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.

Srikant, R. and Yang, Y. (2001). "Mining web logs to improve website organization", Proceedings of the Tenth International World Wide Web Conference, Hong Kong, pp. 430-437.

Suman M. (2012). "A Frequent Pattern Mining Algorithm Based on FP- Tree Structure and Apriori Algorithm", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 2, No. 1, pp. 114-116.

Zaki, M.J. (2000). "Scalable algorithms for association mining" IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 3, pp. 372-390.

Zaki, M.J. (2001). "SPADE: An Efficient Algorithm for Mining Frequent Sequences", Machine Learning, Vol. 42, pp. 31-60.

Zhao, Q. and Bhowmick, S.S. (2003). "Sequential Pattern Mining: A Survey", Technical Report Center for Advanced Data Networks, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.

Zhou, B.Y., Hui, S.C., and Fong, A.C.M. (2004). CS-mine: An Efficient WAP-tree Mining for Web Access Patterns. In Proceedings of the 6th Asia Pacific Web Conference (APWeb'04), Hangzhou, China, Lecture Notes in Computer.

Corresponding Author

Arti Pandey*

E-Mail – **aarti.tiwari10@gmail.com**