



ISSN 2278 – 0211 (Online)

## Generation of Direct and Indirect Association Rule from Web Log Data

**Kalpna Wani**

Department of Computer Engineering, Mumbai University/ PIIT New Panvel, Navi Mumbai, India

**Dr. J. W. Bakal**

Department of IT, Mumbai University/ SSJCOE Dombivali (E), Mumbai, India

### **Abstract:**

Nowadays all people are interested in shopping online as they don't have time to go in market and see the variety of products of different brands available in the market which will solve their purpose. Online shopping is best suited for such people it saves their time as well as money as money discount offers are given on online shopping items. To save more time of such user Web mining provides an option to study the search behaviour of the user and recommend the pages of their interest through web page recommendation system. The aim of our system is to find Direct and Indirect Association rule which will recommend more appropriate page to web page recommendation system.

**Key words:** Preprocessing, Association Rule, Indirect Association Rule, Parsing

### **1. Introduction**

There is a boom of online shopping in the market which motivates companies to analyse user behaviour to know what exactly users are purchasing, of what brand, of what price range etc. Server log is a file which generated automatically at server side which records all the visited pages information of the user. Server log is also called a web log file. We can create the session wise data from the server log files. Web log files consist of massive hidden data, if we mined it, it will help business analyst persons to take the right decision about their business improvement. We can generate association rule from the pre-processed data. In real world data are generally Incomplete i.e. lacking attribute values, lacking certain attributes of interest, or containing only aggregate data or noisy means containing errors or outliers. Sometimes data is inconsistent which contain discrepancies in codes or names.

Tasks in data pre-processing

- Data cleaning: fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- Data integration: using multiple databases, data cubes, or files.
- Data transformation: normalization and aggregation.
- Data reduction: reducing the volume but producing the same or similar analytical results.
- Data discretization: part of data reduction, replacing numerical attributes with nominal ones. [1]

We can find out frequently access sequence of the pages by using the different mining algorithm. Each algorithm is having its own advantages and disadvantages. In our system we are going to consider FP Tree and Eclat algorithm to find the frequently accessed pages. Once we got the frequently access pages from that information we can generate the association rule from that. Association rule is nothing but finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transactional databases, relational databases, and other information repositories.

### **2. Proposed System**

The aim is to develop the system [2] which will analyse user behaviour efficiently and effectively, so that it can be determined which web pages are more likely to be accessed by the user. Performance of the web page Recommendation System can be improved by using

- Algorithm for discovering frequent pattern which will save time and memory space.
- Discovering the Direct and Indirect Association Rules from Frequent Access Pages to recommend more no. of pages.
- Ranking the frequent access pages.

Proposed method contains generation of frequent patterns, finding direct and indirect association rule and ranking the pages for recommendation system.

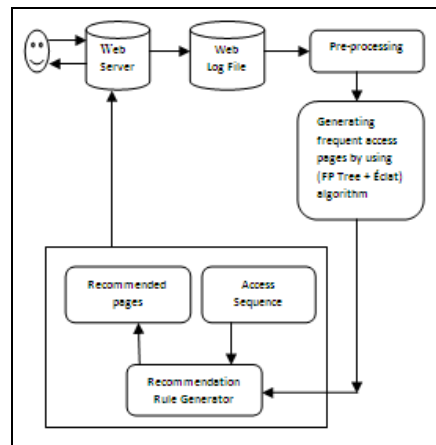


Figure 1: Architecture Diagram for Proposed System

Following steps are used for implementation:

- **Step 1: Data Cleaning & Preprocessing**

As we discussed in the introduction part of this paper Preprocessing is necessary, because Log file contain noisy & ambiguous data which may affect result of mining process. Some of web log file data are unnecessary for analysis process and could affect detection of web attack.

- **Step 2: Finding Frequent Pattern Using Data Mining algorithm**

Here we have used both FP Tree as well as Eclat algorithm to find the frequently access pages from web log data. For FP Tree algorithm first counts the occurrences of pages in database and then reorder the pages with higher count page first and so on. Draw the FP Growth tree and start mining FP tree by considering the suffix as page with lowest count first. Find the conditional pattern base for each page and if length of conditional pattern base is less than some threshold value then find the frequent pattern using Eclat algorithm and if length is greater than or equal to the threshold value then use FP Growth algorithm for finding the frequent pattern of visited pages.

- **Step 3: Generating Direct Association Rule between the Frequent Access Pages**

After getting the frequently visited page sequence filter the pattern by applying some minimum confidence and support value we are going to consider minimum support of 20% and minimum confidence 40% and form the one page association rule between pages of frequent pattern and calculate its support and confidence values and filter it with respect to minimum confidence and support value.

- **Step 4: Indirect Association Rule Generation between the Frequent Access Pages**

To generate the indirect association rule of pages and for further steps of implementation we have referred the paper [3]. There are many pages which are connected to each other indirectly via some transitive pages. Find such pages which are indirectly connected and filter it with respect to minimum confidence of 40% and support of 20%.

- **Step 5: Complex Association Rule Generation**

A complex association rule exists if at least one of two component rules exists, i.e., either direct or complete indirect or both of them. The main quality features of both direct and indirect rules—confidences—are combined within complex association rules.

A complex association rule is characterized by the *complex confidence*,  $con*(di \rightarrow *dj)$ , as follows

$$Con*(di \rightarrow *dj) = \alpha \cdot Con(di \rightarrow dj) + (1 - \alpha) \cdot Con\#(di \rightarrow \#dj)$$

Where  $\alpha$  is the direct confidence reinforcing factor  $\alpha \in [0, 1]$

Calculate the complex confidence value for different values of  $\alpha$ . Setting  $\alpha$  we can emphasize or damp the direct confidence at the expense of the complete indirect one. The greater the value of  $\alpha$ , the closer the complex confidence to the direct one. Depending on highest value of  $\alpha$  rank is assigned to the pages and sends it to recommendation system.

### 3. Frequent Pattern Mining Algorithm

#### 3.1. FP-Growth Algorithm [4]

FP stands for frequent pattern. In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root.

Recursive processing of this compressed version of main dataset grows large item sets directly, instead of generating candidate items and testing them against the entire database. Growth starts from the bottom of the header table (having longest branches), by finding all instances matching given condition. New tree is created, with counts projected from the original tree corresponding to the set of

instances that are conditional on the attribute, with each node getting sum of its children counts. Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree.

Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins.

### 3.2. Eclat (Equivalence Class Transformation) Algorithm [4]

The Eclat algorithm is used to perform itemset mining. Itemset mining let us find frequent patterns in data like if a consumer buys milk, he also buys bread. This type of pattern is called association rules and is used in many application domains.

The basic idea for the éclat algorithm is use tidset intersections to compute the support of a candidate itemset avoiding the generation of subsets that does not exist in the prefix tree.

### 3.3. Algorithm

The Eclat algorithm is defined recursively. The initial call uses all the single items with their tidsets. In each recursive call, the function IntersectTidsets verifies each itemset-tidset pair  $(X, t(X))$  with all the others pairs  $(Y, t(Y))$  to generate new candidates  $N_{xy}$ . If the new candidate is frequent, it is added to the set  $P_x$ . Then, recursively, it finds all the frequent Itemsets in the X branch. The algorithm searches in a DFS manner to find all the frequent sets.

## 4. Proposed Algorithm

In our system we are going to combine the features of FP-Tree and Éclat algorithm. FP –Tree algorithm is efficient when large numbers of items are there and Éclat is suitable for small size of items. It uses FP-tree to store the compact database in memory and recursively mine the frequent patterns from this data structure similar to the FP-growth approach. In addition, it will automatically switch from mining FP-trees using FP-growth to mining TID-lists using Éclat Algorithm depending on the structure of the currently processed data.

In order to enable the mining task using Éclat method, during the pattern growth process, the conditional pattern base of a frequent item will be transformed into TIDlists. If its size is small enough for better mining on the vertical data structure then Éclat Algorithm is used. For e.g. consider the web session database as shown below in Table number 1.

Session Id	Pages
1	P1,P2,P4
2	P1,P4
3	P1,P2,P4
4	P1,P3
5	P2,P4,P5,P6
6	P2,P4
7	P4,P5,P6
8	P2,P4,P5,P6
9	P1,P6
10	P1,P3

Table 1: Session Database

Now we are going to apply the FP-Tree algorithm to find the frequently accessible pages. First step is to count the occurrences of pages in entire database its summary is given in the table.

<b>P4:7</b>
<b>P1:6</b>
<b>P2:5</b>
<b>P6:4</b>
<b>P5:3</b>
<b>P3:2</b>

Table 2: Count of Pages in Database

After getting page count rearrange the page sequence of session database as per order of pages given in Table 2. Rearrangement of pages is shown in Table. 3.

Session Id	Pages	Ordered Pages
1	P1,P2,P4	P4,P1,P2
2	P1,P4	P4,P1
3	P1,P2,P4	P4,P1,P2
4	P1,P3	P1,P3
5	P2,P4,P5,P6	P4,P2,P6,P5
6	P2,P4	P4,P2
7	P4,P5,P6	P4,P6,P5
8	P2,P4,P5,P6	P4,P2,P6,P5
9	P1,P6	P1,P6
10	P1,P3	P1,P3

Table 4: Session Database with ordered pages

Next step is to form the FP-Tree with reference to the Table 2.as shown in fig.2.

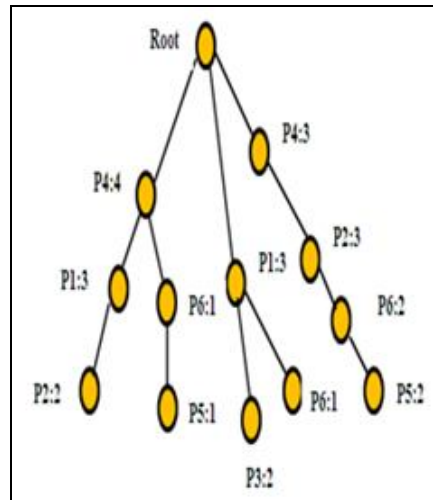


Figure 2: FP-Tree

Next step is mining the FP – Tree .Start mining FP tree by considering the suffix as page with lowest count first. Find the conditional pattern base for each page and if length of conditional pattern base is less than some threshold value then find the frequent pattern using Éclat algorithm and if length is greater than or equal to the threshold value then use FP Growth algorithm for finding the frequent pattern of visited pages. Sample session dataset is shown below.

Item	Condition Pattern Base	Conditional Pattern	Frequent Pattern
P3	{P1:2}	{P1:2}	{P1,P3:2}
P5	{P4:1,P6:1} {P4:2,P2:2,P6:2}	{P4,P6:1} {P4 P2 P6:2}	{P4 P6 P5:3} {P4 P2 P6 P5:2} {P4 P2 P5:2} {P4 P5:3} {P2 P5:2} {P5 P6:3}
P6	{P4:1} {P1:1} {P4,P2:2}	{P4:1} {P4 P2:2}	{P4 P6:3} {P2 P4:2} {P4 P2 P6:2}

P2	{P4 P1:2} {P4:3}	{P4:5,P1:2}	{P4 P1 P2:2} {P4 P2:5} {P1 P2:2}
P1	{P4:3}	{P4:3}	{P4 P1:3}

Table 5: FP – Tree Mining

For implementation of our project for finding frequent pattern using FP-Tree we have considered minimum length of Conditional pattern Base as less than equal to 2 elements. If length of Conditional pattern Base is less than equal to 2 elements then frequents patterns are generated by Éclat algorithm and if it is more than two then frequents patterns are generated by FP-Tree algorithm. As shown in Table 5, for prefix P6, P3, P2, P1 and for one combination of P5 frequents patterns are generated by Éclat algorithm and for rest by FP-Tree algorithm. In FP-Tree algorithm frequents patterns are generated by taking combinations of conditional patterns and Item (Prefix) as shown above in Table.5.Consider page P6 ,it has conditional pattern base as shown below.

{P4:1}
{P1:1}
{P4,P2:2}

Table 6: Conditional pattern base of page P6

His conditional pattern base can be used to construct the conditional FP-tree (Fig. 3).

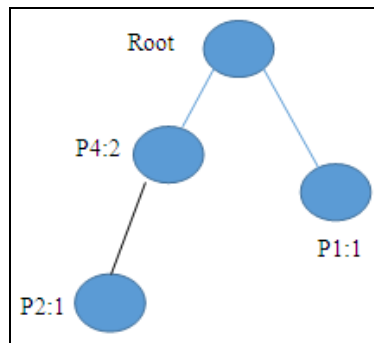


Figure 3: Conditional FP-tree

In proposed algorithm, we consider these sets as transactions (Table 7) and transform them into TID-lists for mining using Éclat approach. The transformation is executed by assigning each set with an ID and grouping IDs into lists. In above example, three TID-lists (Fig. 4) can be generated including {1, 3} of page P4, {2} of page P1 and {3} of page P2. To save memory and benefit bitwise operation efficiency, we represent TID-lists in bit-vector form (Fig.5). In other side, each set in the conditional pattern base has a frequency value indicating the number of its occurrence. We combine all the frequency values into a weight vector which will be used to compute the support of the TID-list. The weight vector in the given example is {1, 1, 2}.

TID	Item	Weight
1	P4	1
2	P1	1
3	P4P2	2

Table 7: Dataset equivalent to table 6

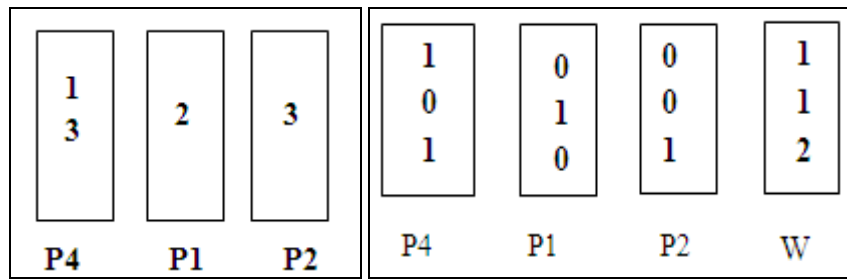


Figure 4: TID-lists

Figure 5: TID bit vectors & Weight Vector

TID-List and Bit vectors transformed from conditional pattern base of page P6 during the *FP-tree mining* stage, thousands or even millions of conditional pattern bases are processed. However, only those whose size is considerably small are transformed into TID bit vectors. This generated TID bit vectors and weight vector W is used further to generate the frequent pattern using Éclat algorithm. To generate the pattern AND operation is performed between P4 and P1 and then between P4 and P2, if result of AND operation gives 1 and corresponding weight vector value is 2(i.e. minimum support) then that pair is considered as frequent pages pair as shown below.

Consider result of P4 AND P1

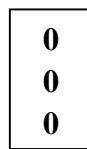


Figure 6: Result of P4 AND P1

Here none of the bit vector is having value 1 so P4P1 is not the frequent pattern.

Consider result of P4 and P2

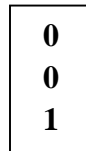


Figure 7: Result of P4 AND P2

In above result third value is 1 which is having corresponding value 2 for weight vector (i.e. Minimum support value) so pattern P4P2 is frequent pattern. Similarly do for other combination to get frequent pattern by Éclat algorithm. Next task to be done is finding the direct and indirect association rule between the frequent access pages to give ranking to them and then recommending for web page recommendation system. Consider the example given below to understand the working of rest of the steps of implementation. After finding the frequent pattern find the confidence and support value for each frequent pattern as shown below in table 8.

After getting the frequently visited page sequence, filter the pattern by applying some minimum confidence and support value and form the one page association rule between pages of frequent pattern and calculate its support and confidence values and filter it with respect to minimum confidence and support value as shown below in table 9.

In direct association rule of pages there are many pages which are connected to each other indirectly via some transitive pages. Find such pages which are indirectly connected and filter it with respect to minimum confidence and support value.

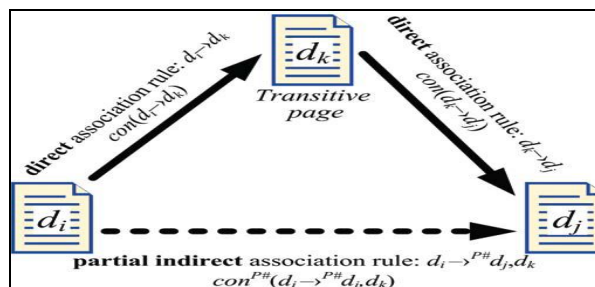


Figure 8: Indirect Association between Two Web Pages

Frequent Pattern	Confidence	Support
{P1,P3}	30%	20%
{P4 P6 P5}	P4 P6->P5 = 100% P4 P5->P6 =100% P5 P6->P4 =100%	30%
{P4 P2 P6 P5}	P4 ->P2 P6 P5 = 28% P2 ->P4 P6 P5 = 40% P6 ->P4 P2 P5 = 5% P5 -> P4 P2 P6 = 60% P2 P6 P5->P4 =100% P4 P6 P5->P2 = 100% P4 P2 P5->P6 = 100% P4 P2 P6->P5 = 100% P4 P2->P6 P5 =40% P4 P6->P2 P5 = 60% P4 P5->P2 P6 =60% P6 P5->P4 P2 =60% P2 P5->P4 P6 =100% P2 P6->P4 P5 =100%	20%
{P4 P2 P5}	P4 P2->P5 =40% P4 P5->P2 = 60% P2 P5->P4 = 100%	20%
{P4 P5}	42%	30%
{P2 P5}	40%	20%
{P5 P6}	100%	30%
{P4 P6}	42%	30%
{P2 P6}	40%	20%
{P4 P2 P6}	P4 P2->P6 = 40% P4 P6->P2 = 60% P6 P2->P4 = 100%	20%
{P4 P1 P2}	P4 P1->P2 = 60% P4 P2->P1 = 40% P1 P2->P4 = 100%	20%
{P4 P2 }	71%	50%
{P1 P2 }	33%	20%

Table 8: Frequent Patterns with Support and Confidence

Sr. No	Rule	Confidence	Support
1	P1->P4	0.5	0.2
2	P2->P4	1	0.5
3	P2->P5	0.4	0.2
4	P2->P6	0.4	0.2
5	P4 ->P2	0.6	0.5
6	P4->P6	0.5	0.4
7	P5->P2	0.6	0.2
8	P6->P2	0.4	0.2
9	P6->P4	0.8	0.4
10	P6->P5	0.4	0.2

Table 9: Formation of Direct Association Rule

The complete indirect association rule  $d_i \rightarrow \# d_j$  aggregates all partial indirect association rules from  $d_i$  to  $d_j$  with respect to all existing transitive pages  $d_k \in T_{ij}$  and is characterized by complete indirect confidence  $con\#(d_i \rightarrow \# d_j)$

$$con\#(d_i \rightarrow \# d_j) = \frac{1}{max_T} \sum_{d_k \in T_{ij}} con^{P\#}(d_i \rightarrow^{P\#} d_j, d_k)$$

Where

$$max_T = \max_{d_i, d_j \in D} (card(T_{ij}))$$

L1 (i-k)	L2 (k-j)	Transitive pages	Lk (k-j)	Complete Indirect Rule
P4->P2 P5->P2 P6->P2	P2->P4 P2->P5 P2->P6	P2	P2->P4 P2->P5 P2->P6	P4->#P5,P2 P4->#P6,P2 P5->#P4,P2 P5->#P6,P2 P6->#P5,P2
P1->P4 P2->P4 P6->P4	P4->P2 P4->P6	P4	P4->P2 P4->P6	P1->#P2,P4 P1->#P6,P4 P2->#P6,P4 P6->#P2,P4
P2->P5 P6->P5	P5->P2	P5	P5->P2	P6->#P2,P5



P2->P6 P4->P6	P6->P2 P6->P4 P6->P5	P6	P6->P2 P6->P4 P6->P5	P2->#P4,P6 P2->#P5,P6 P4->#P2,P6 P4->#P5,P6
------------------	----------------------------	----	----------------------------	--

Table 10: Formation of Indirect Association Rule

Sr. No	Rule	Confidence
1	P1-->#P2,P4	0.3
2	P1-->#P6,P4	0.25
3	P2-->#P4,P6	0.32
4	P2-->#P5,P6	0.16
5	P2- ->#P6,P4	0.5
6	P4-->#P2,P6	0.2
7	P4-->#P5,P2	0.24
8	P4-->#P5,P6	0.2
9	P4-->#P6,P2	0.24
10	P5-->#P4,P2	0.6
11	P5-->#P6,P2	0.24
12	P6-->#P4,P2	0.4
13	P6-->#P5,P2	0.16
14	P6->#P2,P4	0.48
15	P6->#P2,P5	0.24

Table 11: Indirect Association Rule with Confidence

A complex association rule exists if at least one of two component rules exists, i.e., either direct or complete indirect or both of them. The main quality features of both direct and indirect rules—confidences—are combined within complex association rules.

A complex association rule is characterized by the *complex confidence*,  $con^*(di \rightarrow dj)$ , as follows

$$Con^*(di \rightarrow dj) = \alpha \cdot Con(di \rightarrow dj) + (1 - \alpha) \cdot Con\#(di \rightarrow \#dj)$$

Where  $\alpha$  is the direct confidence reinforcing factor

$$\alpha \in [0, 1]$$

Calculate the complex confidence value for different values of  $\alpha$ . Setting  $\alpha$  we can emphasize or damp the direct confidence at the expense of the complete indirect one. The greater the value of  $\alpha$ , the closer the complex confidence to the direct one. Depending on highest value of  $\alpha$  rank is assigned to the pages and sends it to recommendation system.

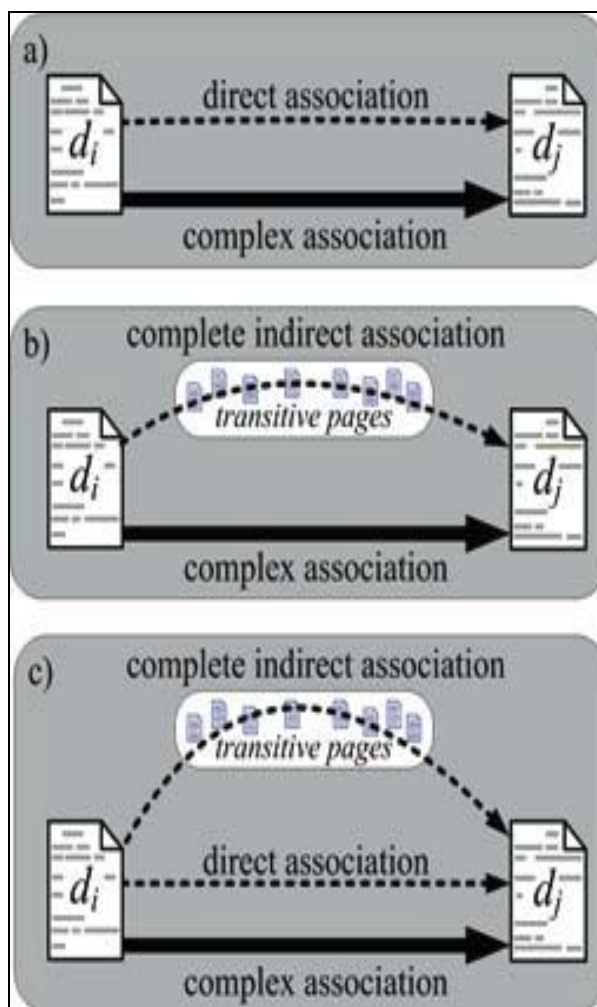


Figure 9: Complex association results from either a direct association (a) or a complete indirect one (b) or both (c)

Page	Direct $P_i \rightarrow P_j$	Indirect $P_i \rightarrow \#P_j$	Complex: $\text{con}^*(P_i \rightarrow P_j)$							
			$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
P1	P4	P2 P6	P2P6P4		P2P4P6	P4P2P6		P2P4P6		P4P2P6
P2	P4P5P6	P4P5P6	P2 P6 P4							
P4	P2P5	P2P5P6	P4P2P6			P2P4P6			P4P2P6	
P5	P2	P4P6	P4P6P2	P4P2P6		P2P4P5				
P6	P2P4P5	P4P5	P6P2P4					P4P6P2		P2P4P6

Table 12: Page Ranking List

Sr.No	Rule	Direct Pi->Pj	Indirect Pi->#Pj	Complex:con*(Pi->Pj)								
				$\alpha= 0.2$	$\alpha= 0.3$	$\alpha= 0.4$	$\alpha= 0.5$	$\alpha= 0.6$	$\alpha= 0.7$	$\alpha= 0.8$	$\alpha= 0.9$	
1	P1->*P2	✗	✓	0.24	0.21	0.18	0.15	0.12	0.9	0.6	0.03	
2	P1->*P3	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
3	P1->*P4	✓	✗	0.1	0.15	0.2	0.25	0.30	0.35	0.40	0.45	
4	P1->*P5	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
5	P1->*P6	✗	✓	0.2	0.175	0.026	0.125	0.1	0.075	0.05	0.025	
6	P2->*P1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
7	P2->*P4	✓	✓	0.456	0.524	0.592	0.66	0.728	0.796	0.864	0.932	
8	P2->*P5	✓	✓	0.208	0.013	0.015	0.016	0.015	0.013	0.010	0.005	
9	P2->*P6	✓	✓	0.32	0.042	0.048	0.05	0.048	0.42	0.032	0.018	
10	P3->*P1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
11	P4->*P1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
12	P4->*P2	✓	✓	0.019	0.025	0.028	0.030	0.288	0.252	0.192	0.010	
13	P4->*P5	✗	✓	0.176	0.154	0.132	0.11	0.088	0.066	0.044	0.022	
14	P4->*P6	✓	✓	0.019	0.025	0.028	0.03	0.028	0.025	0.019	0.010	
15	P5->*P1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
16	P5->*P2	✓	✗	0.12	0.18	0.24	0.30	0.36	0.42	0.48	0.54	
17	P5->*P4	✗	✓	0.48	0.42	0.36	0.30	0.24	0.18	0.12	0.06	
18	P5->*P6	✗	✓	0.192	0.168	0.144	0.12	0.096	0.072	0.048	0.024	
19	P6->*P1	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	
20	P6->*P2	✓	✗	0.08	0.12	0.16	0.20	0.24	0.28	0.32	0.36	
21	P6->*P4	✓	✓	0.051	0.067	0.076	0.08	0.076	0.067	0.512	0.028	
22	P6->*P5	✓	✓	0.208	0.232	0.256	0.28	0.304	0.328	0.352	0.005	

Table 13: Values of Complex Confidence

## 5. Conclusion

In today's world there is boom of online shopping and because of that if companies want to increase their business it's required to analyze the user behavior. This is possible only with help of web mining and its further processing to find frequent pattern and the association rule between them. This system will help to recommend appropriate page to user. In future we are planning to complete remaining work by providing security to log file

## 6. References

1. [http://www.cs.ccsu.edu/~markov/ccsu\\_courses/DataMining-.html](http://www.cs.ccsu.edu/~markov/ccsu_courses/DataMining-.html)
2. Kalpana Wani, Madhu N. Association Rule Mining for Large Database. In International Journal of Engineering Research and Technology in ISSN:2278-0181(ISO 3297:2007)volume-2 Issue10 Oct 2013
3. Przemyslaw Kazienko." Mining Indirect Association Rules for Web Recommendation", Int. J. Appl. Math. Comput. Sci., 2009, Vol. 19, No. 1, 165–186
4. [http://en.wikipedia.org/wiki/Association\\_rule\\_learning](http://en.wikipedia.org/wiki/Association_rule_learning)
5. J. Han, J. Pei, Y. Yin, "Mining frequent patterns without candidate generation", in Proc. of the Int. Conf.on Management of Data, 2000.
6. Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207. Do i: 10.1145/170035.170072. ISBN 0897915925.
7. Bina Kotiyalt, Ankit Kumar, Bhaskar Pant, R.H. Goudar, Shiv ali Chauhan and Sonam Junee." User Behavior Analysis in Web Log through Comparative Study of Éclat and Apriori", Proceedings of 7'h International Conference on Intelligent Systems and Control (ISCO 2013)
8. N. Venkatesan, Ramaraj," Frequent Itemset Mining with Bit Search", Journal of Theoretical and Applied Information Technology, 15 July 2012. Vol. 41 No.1
9. Ravi Bhushan and Rajender Nath," Recommendation of Optimized Web Pages to Users Using Web Log Mining Techniques", 978-1-4673-4529-3/12/\$31.00c 2012 IEEE
10. Bamshad Mobasher, Robert Cooley, Jaideep Srivastava,"Automatic Personalization Based on Web Usage Mining", Communications of the ACM, New York, Volume 43, Issue 8, Aug 2000.
11. Yan Li, Boqin FENG and Qinjiao MAO, "Research on Path Completion Technique in Web Usage Mining", International Symposium on Computer Science and Computational Technology, IEEE, 2008.
12. Robert. Cooley, Bamshed Mobasher and Jaideep Srinivastava,"Web mining:Infonnation and Pattern Discovery on the World Wide Web", In International conference on Tools with Artificial Intelligence, pages 558-567, Newport Beach, IEEE,1997.
13. Perkowitz, M., Etzioni, Adaptive sites: Automatically learning from user access patterns. In: Proc. of the Sixth International WWW Conference, Santa Clara, CA. (1997).
14. Cooley, R., Mobasher, B., & Srivastava, J.: Data preparation for mining World Wide Web browsing patterns. Knowledge Information Systems, 1(1), pp. 5-32. (1999).
15. H. Mannila, H. Toivonen. Discovering generalized episodes using minimal occurrences. In: Proc.Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 1996