Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 11, November 2014, pg.38– 45

RESEARCH ARTICLE

Identification of Best Algorithm in Association Rule Mining Based on Performance

Garima Sinha[1], Dr. S. M.Ghosh[2]

[1]Computrt Science & Engineering, Chhattisgarh Swami Vivekanand Technical University, Bhilai, India

[2]Computer Science & Engineering, Chhattisgarh Swami Vivekanand Technical University, Bhilai, India

[1] garimasinha03@gmail.com; [2] samghosh06@rediffmail.com

Abstract— Data Mining finds hidden pattern in data sets and association between the patterns. To achieve the objective of data mining association rule mining is one of the important techniques. Association rule mining is a particularly well studied field in data mining given its importance as a building block in many data analytics tasks. Many studies have focused on efficiency because the data to be mined is typically very large. This paper presents a comparison on three different association rule mining algorithms i.e. FP Growth, Apriori and Eclat. The time required for generating frequent itemsets plays an important role. This paper describes implementations of these three algorithms that use several optimizations to achieve maximum performance, w.r.t. execution time. The comparison of algorithms based on the aspects like different support and confidence values.

Keywords- Association Rule Mining, Apriori, FP growth, Eclat

I. INTRODUCTION

The extraction of hidden predictive information from vast amount of databases is called Data Mining, is a powerful new technology with great potential to help companies .It focus on the most vital information in their data warehouses. Data mining tools forebode future trends and behaviours, allowing businesses to make knowledge-driven, proactive decisions. Data mining tools can answer business queries that were too time consuming to resolve. Many companies already collect and refine large amount of data.

Frequent pattern mining is one of the most important research topics in data mining. The function is to discover the transactional data which describes the behaviour of the transaction. In an online shopping or an online business or the customers can buy items together. Frequent patterns are patterns such as item sets, sub sequences or substructures that appear in a data set frequently. We can examine the behaviour of the products purchased by the customers from the transactional database. For example a set of items Mobile and Sim card that appear frequently as well as together in a transaction set is a frequent item set. Subsequences means if a customer buys a Mobile he must also buy a Sim card and then head phone etc. From the history of the database these transactions are occurring sequentially is called sequential patterns. The Substructure means different structural forms such as sub tree, sub graphs, which may be used along with item sets or sequences.

II. PROBLEM STATEMENT

The problem can be stated as follows: Let I={$ì_1$, $ì_2$, $ì_3$….$ì_m$ }be a set of m. distinct literals called items, a set of variable D is length transactions over I. Each transaction contains a set. of items ($ì_1$, $ì_j$…….. $ì_k$) C I. A transaction also has an associated unique identifier called TID. An association rule is an implication- of the form A=>B, where A,B C I, and A∩B = ϕ. A is called the antecedent and B is called the consequent of the rule. In general, a set of items (such as the antecedent or the consequent of a rule) is called an itemset. The number of items in an itemset is called the length of an itemset. Itemsets of some length k are referred to as k-itemsets. For an itemset A.B, if B is an m-itemset then B is called an m-extension of A.

Each itemset has an associated measure of statistical significance called support. For an itemset A C I, *support* (A) =s if the fraction of transactions in D, containing A equals s. A rule has a measure of its strength called confidence defined as the ratio support (A U B) / support(A). The problem of mining association rules is to generate all rules that have support and confidence greater than some user specified minimum support and minimum confidence thresholds, respectively. This problem can be decomposed into the following subproblems:

1. All itemsets that have support above the user specified minimum support are generated. These itemset are called the large itemsets. All others are said to be small.
2. For each large itemset, all the rules that have minimum confidence are generated as follows: for a large itemset A and any B C A, if support (A)/support(A - B) ≥ minimum-confidence, then the rule A-B => B is a valid rule.

For example, let T1 = {P,Q,R}, T2 = {P,Q,S},T3 = {P,S,O} and T4 = {P,Q,S} be the only transactions in the database. Let, the minimum support and minimum confidence be 0.6 and 0.9 respectively. Then the large itemsets are the following :{P},{Q}, {S}, {P,Q}, {P,S} and {P,Q,S}. The valid rules are Q => P and S => P.

III. RELATED WORK

Several algorithms for mining associations have been proposed in the literature [1, 3, 6, 7, 11, 12, 14, 19, 15]. The *Apriori* algorithm [3, 2] forms the core of almost all of the current algorithms. The key observation used is that all subsets of a frequent itemset must themselves be frequent. During the initial pass over the database the support for all single items (1-itemsets) is counted. The frequent 1-itemsets are used to generate candidate 2-itemsets. The database is scanned again to obtain their support, and the frequent 2-itemsets are selected for the next pass. This iterative process is repeated for n=3;4_until there are no more frequent n-itemsets to be found. However, if the database is too large to fit in memory, these algorithms incur high I/O overhead or scanning it in each iteration. Paper, we analyzed and studied various existing improved apriori algorithm to mine frequent itemsets. Mainly common drawbacks are found in various existing apriori algorithm which is improved by using different approaches. It can be applied to many different applications like market basket analysis, telecommunication, network analysis, banking services and many others.

A new method for generating frequent itemsets by using frequent itemset tree (FI-tree)[5]. The analysis of total execution time for generating frequent itemsets denoted with standard dataset wine. Method execution time is better compare to SaM method. At 60% support threshold, two methods nearly matches the execution time.

The comparison of 2 new algorithms Apriori and Apriori-hybrid with the previously known algorithms, the AIS and SETM algorithms have done [11].

Eclat is vertical data format algorithm. Basic Features of Eclat have introduced in [13] i.e. Transaction Recoding, Types of Incidence Structures, Incidence Matrix Derivation etc. Survey on three different association rule mining algorithms[16] -AIS, FP-tree and Apriori algorithms have done. It describe their drawbacks which would be helpful to find new solution for the problems found in these algorithms.

[19] Described an algorithm which is not only efficient but also fast for discovering association rules in large databases in the (n-1)th pass are used to generate the candidate item sets Cn, using the Apriori-gen function Next, the database is scanned and the support of candidates in Cn is counted. This process illustrate in fig. 1, which is derived from Table-1. The working of Apriori algorithm is fairly depends upon the Apriori property which states that" All nonempty subsets of a frequent itemsets must be frequent"(table 2).

IV. METHOD DESCRIPTION

A. Apriori Algorithm

Apriori algorithm was first proposed by Agrawal .Apriori is more efficient during the candidate generation process .To count the support of item sets it uses a breadth-first search strategy and uses a candidate generation function. It exploits the downward closure property of support. To avoid measuring certain item sets, Apriori uses pruning techniques ,while guaranteeing completeness. These are the item sets that the algorithm can prove will not turn out to be large. The algorithm simply counts item occurrences to determine the large 1- itemsets in the first pass. A subsequent pass, say pass n, consists of two phases. First, the large item sets Ln-1 found in the (n-1)th pass are used to generate the candidate item sets Cn, using the Apriori-gen function Next, the database is scanned and the support of candidates in Cn is counted. This process illustrate in Fig-1, which is derived from Table-I. The working of Apriori algorithm is fairly depends upon the Apriori property which states that" All nonempty subsets of a frequent itemsets must be frequent".

TABLE I. TRANSACTIONAL DATA

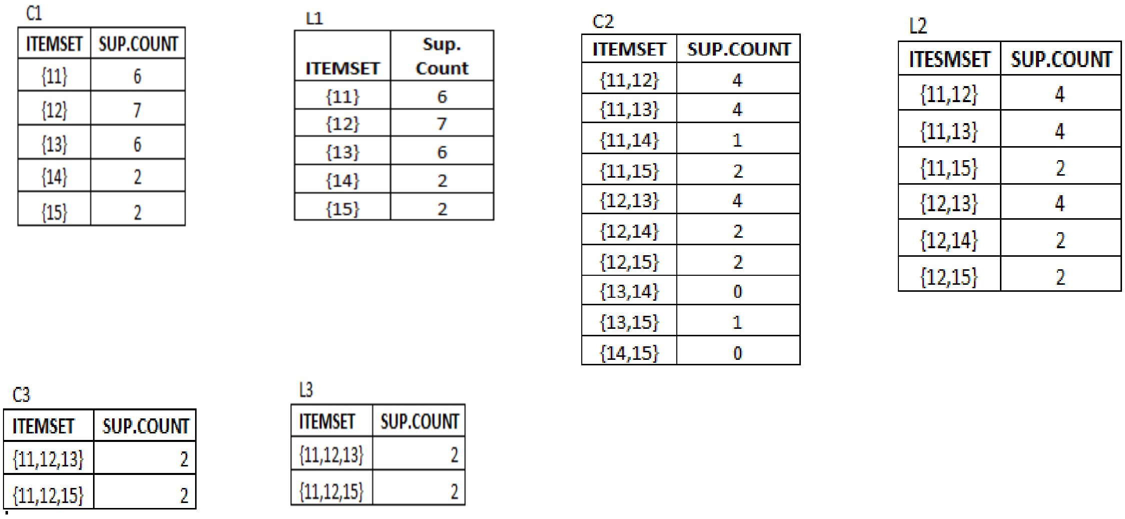
TID	LIST OF ITEMS
T101	I1,I2,I5
T201	I2,I4
T301	I2,I3
T401	I1,I2,I4
T501	I1,I3
T601	I2,I3
T701	I1,I3
T801	I1,I2,I3,I5
T901	I1,I2,I3

1) Steps involve in Apriori Algorithm

a) Candidate item sets are generated using only the large item sets of the previous pass without considering the transactions in the database.

b) The large item set of the previous pass is joined with itself to generate all item sets whose size is higher by 1.

c) Each generated item set that has a subset which is not large is deleted. The remaining item sets are the candidate ones.



C1

ITEMSET	SUP.COUNT
{11}	6
{12}	7
{13}	6
{14}	2
{15}	2

L1

ITEMSET	Sup. Count
{11}	6
{12}	7
{13}	6
{14}	2
{15}	2

C2

ITEMSET	SUP.COUNT
{11,12}	4
{11,13}	4
{11,14}	1
{11,15}	2
{12,13}	4
{12,14}	2
{12,15}	2
{13,14}	0
{13,15}	1
{14,15}	0

L2

ITESMSET	SUP.COUNT
{11,12}	4
{11,13}	4
{11,15}	2
{12,13}	4
{12,14}	2
{12,15}	2

C3

ITEMSET	SUP.COUNT
{11,12,13}	2
{11,12,15}	2

L3

ITEMSET	SUP.COUNT
{11,12,13}	2
{11,12,15}	2

Fig. 1 Apriori Mining Process

B. Fp growth Algorithm

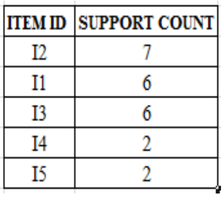
Among from the many algorithms suggested like Apriori algorithms. Apriori algorithm based upon the anti-monotone property. Due to their two main problems i.e. high computational cost and repeated database scan , there is need of compact data structure for mining frequent item sets. FP-Growth algorithm is an efficient algorithm for producing the frequent itemsets without generation of candidate item sets which based upon the divide and conquers strategy. It needs a 2 database scan for finding all frequent item sets. This approach compresses the large database of frequent itemsets into frequent pattern tree structure recursively in the same order of magnitude as the numbers of frequent patterns, then in next step divide the compressed database into set of conditional databases.

1) Construction of Frequent Pattern Tree

First it scans the database and manages the items appearing in the transaction. The items are considered infrequent whose support is less than minimum support. These items are deleted from consideration. All other remaining items are considered as frequent items and arrange in the descending order of their frequency. This list is known as header table when store in table. All the respective support of the items is stored using pointers in the frequent pattern tree. Then construct the frequent pattern tree which is also known as compact tree. In header table, The sorted items according to frequency are used to build the FP-tree (Fig. 2.).Essentially all transaction restored in a tree data structure. This requires a complete database scan. when the item insert in the tree checks if it exist earlier in tree as in same order then increment the counter of support by 1 which is mentioned along with each item in the tree which separated by comma. Using pointers a link is maintained which same item and its entry in header table. Pointer points to the first occurrence of each item in header table. It uses the tree data structure which stores all frequent elements in a compact form.

Suppose minimum support is 2. So delete all infrequent items whose support is less them 2.The remaining transactions arranged in descending order of their frequency. Create a FP- tree for Each Transaction create a node of an items whose support is greater than minimum support, as same node encounter just increment the support count by 1.Table II denote the support count of items derived from Table-I.

TABLE II. SUPPORT COUNT



ITEM ID	SUPPORT COUNT
I2	7
I1	6
I3	6
I4	2
I5	2

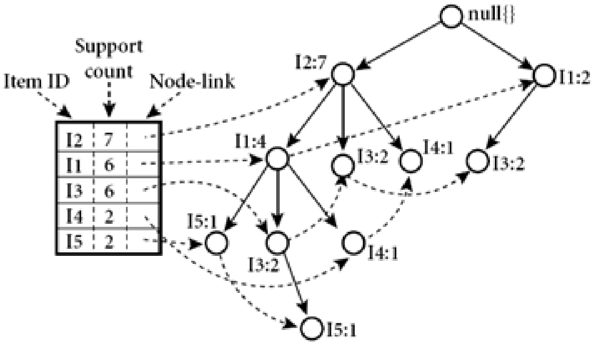


Fig. 2 FP Tree construction of Table-I.

C. *Eclat Algorithm*

Using vertical data format, frequent itemsets can also be mined efficiently, which is the essence of the ECLAT (Equivalence CLASS Transformation) algorithm. It is developed by Zaki.

TABLE 3
THE VERTICAL DATA FORMAT OF THE TRANSACTIONAL DATASET

ITEMSET	TID SET
I1	{T101,T401,T501,T701,T801,T901
I2	{T101,T201,T301,T401,T601,T801,T901}
I3	{T301,T501,T601,T701,T801,T901}
I4	{T201,T401}
I5	{T101,T801}

1) Mining frequent itemsets using vertical data format.

Consider the horizontal data format of the transaction database, of Table.1This can be transformed into the vertical data format shown in Table -3 by scanning the data set once. Consider the horizontal data format of the transaction database, of Table.1This can be transformed into the vertical data format shown in Table -3 by scanning the data set once. Mining can be performed on this data set by intersecting the TID sets of every pair of frequent single items. The minimum support count is 2. Because every single item is frequent in Table 5, In total there are 10 intersections performed .It generate 8 nonempty 2-itemsets as shown in Table 4. The itemsets {I1, I4} and {I3, I5}, do not belong to the set of frequent 2-itemsets. The candidate generation process here will generate only two 3-itemsets: {I1, I2, I3} and {I1, I2, I5}. By intersecting the TID sets of any two corresponding 2-itemsets of these candidate 3-itemsets, Shown in Table 5. where there are only two frequent 3-itemsets: {I1, I2, I3: 2} and {I1, I2, I5: 2}.
It involve following steps:

- First, By scanning the data set once we transform the horizontally formatted data to the vertical format. The support count of an itemset is simply the length of the TID set of the itemsets
- Starting with n = 1, the frequent n-itemsets can be used to construct the candidate (n+1)-itemsets based on the Apriori property. The computation is done by intersection of the TID sets of the frequent n-itemsets to compute the TID sets of the corresponding (n+1)-itemsets.
- This process repeats, with n incremented by 1 each time, until no frequent itemsets or no candidate itemsets can be found. Besides taking advantage of the Apriori property in the generation of candidate (n+1)-itemset from frequent n-itemsets,

TABLE 4
THE 2-ITEMSET S IN VERTICAL DATA FORMAT

ITEMSET	TID SET
{I1,I2}	{T101,T401,T801,T901}
{I3,I3}	{T501,T701,T801,T901}
{I1,I4}	{T401}
{I1,I5}	{T301,T601,T801,T901}
{I2,I3}	{T201,T401}
{I2,I5}	{T101,T801}
{I3,I5}	{T801}

TABLE 5
THE 3-ITEMSETS IN VERTICAL DATA FORMAT

ITEMSET	TID SET
{I1,I2,I3}	{T801,T901}
{I1,I2,I5}	{T101,T901}

V. RESULT AND DISCUSSION

Data set: The dataset was taken from the UCI repository of machine learning databases [18]. The characteristics of datasets selected for the experiment (Table 6)

TABLE 6
CHARACTERISTICS OF DATASETS.

The characteristics of Datasets		
Dataset Name	No. of Columns	No. of Records
Pima Dataset	9	768

Result analysis: A detailed study to assess the performance of Apriori,Fp growth and Eclat algorithms. The performance metrics is the total execution time taken and the number of frequent itemsets generated using pima datasets. For this comparison also same dataset were selected as for the experiments with minimum support and confidence ranging from 1% to 17%.(Table-7)

TABLE 7
TOTAL EXECUTION TIME USING PIMA DATASET

Support (%)	Confidence (%)	Total Execution Time in second		
		Apriori	Fp Growth	Eclat
1	5	0.115	179.23	0.065
1	10	0.102	123.54	0.055
2	5	0.1025	84.79	0.023
2	10	0.06	25.58	0.0335
5	13	0.018	8.46	0.0075
6	15	0.028	6.58	0.007
7	17	0.036	5.12	0.0037

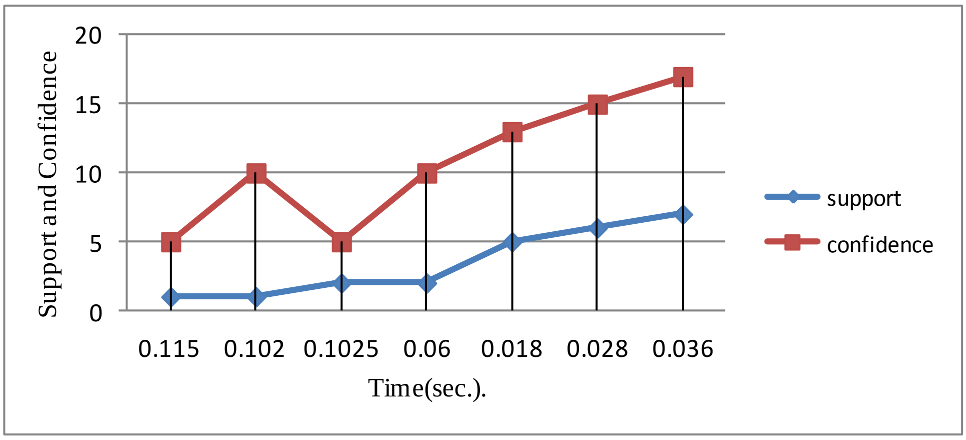


Fig:3 Graph for Apriori

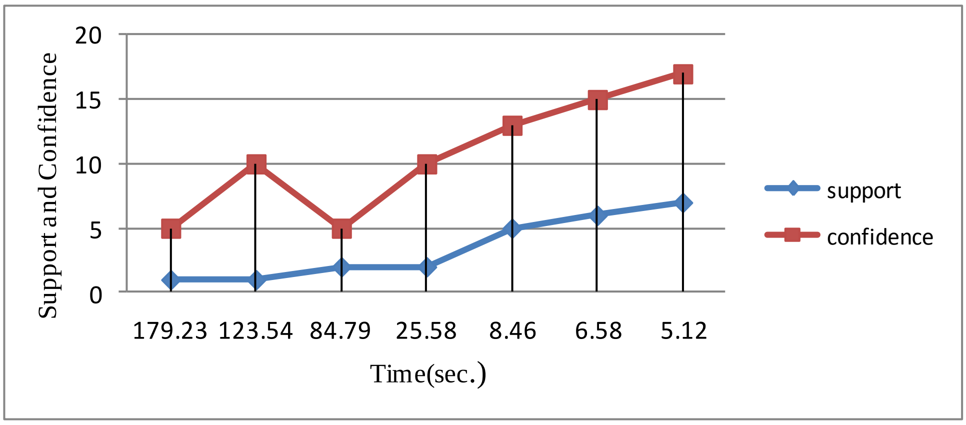


Fig:4 Graph for FP Growth

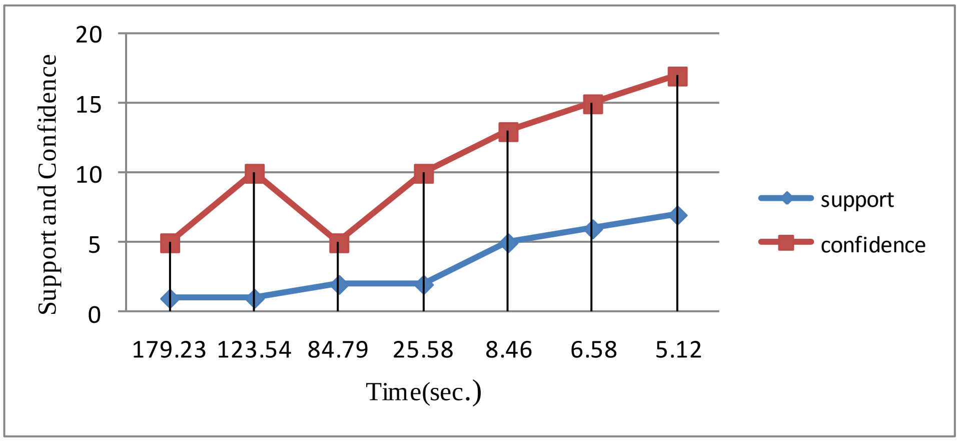


Fig:5 Graph for Eclat

From the above graph(Fig. 4, Fig. 5, Fig. 6) obtained by taking the pima dataset between the execution time verses support and confidence .It is depicted that Eclat is the best among three algorithms.

VI. CONCLUSION AND FUTURE SCOPE

This paper presents a comparison on three different association rule mining algorithms i.e. FP Growth, Apriori and Eclat based on execution time verses support and confidence. The comparison of algorithms is done using pima dataset where it has been identified that with a particular itemset. The Eclat algorithm is fastest among three. It was also identified that the execution time decreases with increasing confidence and support.

The above algorithms can be used in other domains to bring out interestingness among the data present in the repository. Association rules produced by these three algorithms can be combined to form efficient algorithms for better results for any real life application. Algorithms can also be combined to for an efficient algorithm.

REFERENCES

[1] Patel Tushar S.et.al.,“ An Analytical Study of Various Frequent Itemset Mining Algorithms”, *Research Journal of Computer and Information Technology Sciences* , Vol. 1(1) ,February (2013)

[2] Ekta Garg et.al. “A Survey on Improved Apriori Algorithm”, *International Journal of Engineering Research and Technology* ,Volume 2,July 2013

[3] Ms. Shweta et.al., “ Mining Efficient Association Rule Through Apriori Algorithm Using Attribute and Comparative Analysis of Various Association Rule Algorithms”,*International Journal of Advanced Research in Computer Science and Software Engineering* ,Volume 3,June 2013

[4] Harendra singh et.al. “A modified FP Tree Algorithm for Generating Frequent Access Patterns” *Journal of Environmental Science, Computer Science and Engineering & Technology*, Volume 2,August 2013

[5] Damor Nirali N. et.al. “A new Method to Mine Frequent itemset Using Frequent Itemset Tree”,*Research Journal of Computer and Information Technology Science*,Volume1,April 2013

[6] saravanan Suba et.al. “ A Study on Milestone of Association Rule Mining Algorithm in Large Data Bases”*Inetnational Journal of Computer Application* ,Volume 47,June 2012

[7] Hamman W.Sammuel et.al. “Fastest Association Rule mining algorithm Predictor(FARM-AP)”*ACM* ,May 2011

[8] Xindong Wu et.al. “Top 10 Algorithm in Data Mining”,*Springer*,December-2007

[9] Jiawei Han et.al. “ Frequent pattern mining: current status and future Directions”, *Springer*, January 2007

[10] Jiawei Han et.al. “Data Mining Concepts and Techniques ”,*Elsevier* ,San Francisco, Second Edition,2006

[11] Rakesh Agrawal et.al. “ Fast Algorithm for Mining Association Rule”, *Proceeding of 20th VLDB Conference*,Santiago,chile,1994

[12] Aakansha Saxena “A Survey on Frequent Pattern Mining Methods- Apriori, Eclat, Fp growth”, *International Journal of Engineering Development and Research*,2014

[13] Thieme, S.L. “Algorithmic Features of Eclat”. *FIMI, Volume 126 of CEUR Workshop Proceedings*, CEUR-WS.org, 2004.

[14] vManisha Girotra et.al. “Comparative Survey on Association Rule Mining Algorithms” International Journal of Computer Applications (0975 – 8887) Volume 84 – No 10, December 2013.

[15] Khurana, K., and Sharma, S. “A comparative analysis of association rule mining algorithms ”.*International Journal of Scientific and Research Publications*, Volume 3, Issue 5, May 2013.

[16] R.Divya and S.Vinod kumar “survey on AIS,Apriori and FP Growth Algorithms ”,*International Journal of Computer Science and Management Research,* Vol 1, Issue 2 ,September 2012

[17] B. Goethals and M. J. Zaki, editors. *Proceedings of the IEEE ICDM,* Workshop on Frequent Itemset Mining Implementations,Melbourne, Florida, USA, November 19, 2003.

[18] UCI Machine Learning Repository,http://archive.ics.uci.edu/ml/datasets/adult

[19] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for Mining Association in large databases.In *21st VLDB conf.*,1995.

Raorane A.A., Kulkarni R.V. and Jitkar B.D., Association Rule –Extracting knowledge using Market Basket Analysis,*Res.J.Recent Sci.*,1(2),19-27 (2012)