I.J. Information Technology and Computer Science, **2014, 11, 59-65**
Published Online October 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2014.11.09

A Novel Approach for Association Rule Mining using Pattern Generation

Deepa S. Deshpande
MGM's Jawaharlal Nehru Engineering College, Aurangabad, 431003, India
Email: deepadsd@yahoo.com

***Abstract*—**Data mining has become a process of significant interest in recent years due to explosive rate of the accumulation of data. It is used to discover potentially valuable implicit knowledge from the large transactional databases. Association rule mining is one of the well known techniques of data mining. It typically aims at discovering associations between attributes in the large databases. The first and the most influential traditional algorithm for association rule discovery is Apriori. Multiple scans of database, generation of large number of candidates item set and discovery of interesting rules are the main challenging issues for the improvement of Apriori algorithm. Therefore in order to decrease the multiple scanning of database, a new method of association rule mining using pattern generation is proposed in this paper. This method involves three steps. First, patterns are generated using items from the transaction database. Second, frequent item set is obtained using these patterns. Finally association rules are derived. The performance of this method is evaluated with the traditional Apriori algorithm. It shows that behavior of the proposed method is much more similar to Apriori algorithm with less memory space and reduction in multiple times scanning of database. Thus it is more efficient than the traditional Apriori algorithm.

***Index Terms*—**Data Mining, Association Rule Mining, Frequent Item Set

I. INTRODUCTION

Association rule mining [5] is used to find interesting trends and patterns in the large dataset. An association rule is an implication or if-then-rule which is supported by data. A typical association rule example is "90 percent of all customers who buy bread and butter also buy milk". The motivation for development of association rules is market basket analysis which analyzes customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. When a customer passes through a point of sale, the contents of his market basket are registered. This results in large collections of market basket data which provide information about which items were sold and, in particular, which combinations of items were sold. Thus it is the way of finding unsuspected relationship for decision making process by generating frequent item sets. Frequent item sets are those that occur at least a user-given number of times (referred as minimum support threshold) in the database. Such type of rules can be useful for important business decisions like product pricing, promotions, store layout etc. Association rule discovery will simply recover the likelihoods for any item to be purchased. Association rule mining algorithms are broadly characterized by its strategy to traverse the search space i.e. Breadth First Search (BFS)/Depth First Search (DFS) and strategy to determine support values of item sets i.e. counting occurrences or intersection. Apriori algorithm [18] is the most popular algorithm for mining association rules which comprises generation of frequent item set and discovery of association rules. Multiple scans of database, generation of large number of candidates item set and discovery of interesting rules are the main challenging issues for the improvement of Apriori algorithm. To address these issues many researchers developed several methods to improve the performance of association rule mining using different strategies [28] like BFS and counting occurrences, BFS and intersecting, DFS and counting occurrences, DFS and intersecting. In this paper, an attempt is made to propose a new method of association rule mining using pattern generation to address the issue of multiple scans of database. It uses strategy with both counting occurrences and intersection together to determine support values of item sets.

A. Mining Association Rules

Mathematical model used to describe association rule mining is represented by item sets and associations [5]. Let I = {i_1, i_2... i_m } be set of items. D be the set of transactions, where each transaction T is set of items such that T ℂ I. The quantities of items are not considered but the items are numbered as 1…d. Item sets of T are then sets of integers between 0 and d. The item sets are represented by bit vectors where item j is present in the corresponding item set of T if the j^{th} bit is set. Each transaction is associated with an identifier called Tid. Consider the example with the items pen, ink, milk and juice with item numbers 1, 2, 3 and 4 respectively.

The transaction containing pen, ink and juice is then represented by the bit vector (1 1 0 1). From the bit vector it is clear which elements are in item set of T and which are not. The database D is represented as a bit matrix where each row corresponds to an item set for transaction T and the columns correspond to the items. For example transactional database for purchase shown in "Table 1" can be represented by the bit matrix shown in "Table 2". The rule x → y holds in transaction set D with confidence c if c% of transactions in D that contain x also contain y. The rule has support s in transaction set D if A of transactions in D contain x Ù y. Confidence denotes the strength of implication and support indicates the frequencies of the occurring patterns in the rule. The proportion of transactions in D for which this rule holds is called the confidence and it is defined formally as C (x → y) = s(x V y) / s(x), where C represents confidence and s represents support. A strong rule is given by a rule x → y for which x V y is frequent, i.e., s(x V y) ≥ min-supp and c(x → y) ≥ min-conf. The constants min-supp and min-conf are nonzero values provided by the user and their careful choice is crucial to the detection of sensible rules. The problem of mining association rules is decomposed into two steps:

Table 1. Transaction Database

Tid	Items
1	{pen, ink, milk, juice}
2	{pen, ink, milk}
3	{pen, milk}
4	{pen, ink, juice}

Table 2. Bit Matrix

Tid	Pen	ink	milk	juice
1	1	1	1	1
2	1	1	1	0
3	1	0	1	0
4	1	1	0	1

1. Discover the frequent item sets i.e. sets of item sets that have transaction support above user pre-defined min-supp.
2. Use frequent item sets to generate association rules for the database.

Example of frequent item set for above purchase database considering min-supp as 50% is given here

L = {pen | s(pen) > min-supp ;
ink | s(ink) > min-supp ;
milk | s(milk) > min-supp}

Example of association rule derived is {pen → ink} because (pen V ink) is frequent i.e. s (pen V ink) ≥ min-supp and confidence of rule is

C (pen → ink) = s (pen V ink) / s (pen) = ¾ = 75%

Overall performance of mining association rules is determined by the first step because after finding large item set the corresponding association rules can be derived in straightforward manner. Therefore this paper focuses on finding frequent item set using traditional and proposed algorithm of association rule mining.

B. Traditional Apriori Algorithm

The traditional Apriori algorithm [5] uses downward closure property which states that each subset of frequent item set must also be frequent. This algorithm was suggested by R. Aggrawal and R. Srikant in 1994 [18] and is one of the most important data mining algorithms. It uses a breadth first search approach, first finding all frequent 1-item set and then discovering frequent 2-itemset and continues by finding increasingly larger frequent item sets. It is named so because it uses prior knowledge of frequent item set property. It takes database D of t transactions and min-supp threshold represented as fraction of t as input. Apriori generates all possible frequent item set L_1, L_2 $L_3 \ldots L_k$ as output. The algorithm proceeds iteratively. Item sets with single item are considered for generating frequent item set in the first pass. In the subsequent passes frequent item sets identified in the previous pass are extended with another item to generate frequent item sets. The algorithm terminates after k passes if no frequent k- item set is found.

```
Algorithm: Apriori Algorithm
Input: D-Transactional Database, min-supp- Minimum Support Threshold
Output: L- Large item set
Begin
Step 1: k =1
Step 2: Find frequent item set Lk from Ck /* Ck is the set of all candidate item sets */
        Scan database D and count each item set in Ck
        If the count > min-supp then
            Add that item set to Lk
Step 3: Form Ck+1 from Lk
        For k = 1, C1 = all item sets of lengrth-1
        For k >1, generate Ck from Lk-1 as
        Ck = k-2 way join of Lk-1 with itself;        /* joining step */
        If both {I1 ,…,Ik-2 , Ik-1} and { I1 ,…,Ik-1 , Ik } are in L k-1
        then
        Add {I1… Ik-2 , Ik-1, Ik } to Ck
        Remove {I1… Ik-2 , Ik-1, Ik } if it does not contain a large
        (k-1) subset; /* prune step */
Step 4: k = k + 1
Step 5: Repeat steps 2, 3 and 4 until Ck is empty
End
```

Fig. 1. Traditional Apriori algorithm.

C. Outline of the Paper

In this paper, the algorithmic aspect of association rule mining is discussed. During last two decades, variety of efficient algorithms has been developed to mine association rules. Hence Section 2 gives general survey of few of these algorithms. The proposed method of association rule mining along with example is discussed in Section 3. Furthermore Section 4 describes the performance evaluation of the proposed method of association rule mining and traditional Apriori algorithm. Finally conclusion is given in Section 5 with a short summary of our results.

II. RELATED WORK

Many efficient and scalable algorithms [28-35] have been developed for association rule mining in last two decades. These algorithms can be classified into three categories. i) Apriori like algorithms which uses candidate generation and test approach. ii) Frequent pattern growth based algorithms such as FP-growth. iii) Algorithms that use vertical database format. Huge number of candidates and multiple scans of transaction database are the challenging issues for improvement of association rules algorithms. R. Agrawal, T. Imielinski and A. Swami [22] introduced AIS algorithm for association rule discovery over market basket data. It requires many passes over the database. Candidate item sets are generated where most of them turn out to be not frequent. Another algorithm SETM [11] was presented by M. Houtsma and A. Swami for the same task. It saves a copy of candidate item set together with transaction ID of the generating transaction. Like AIS it also generates candidate item sets based on transaction scans from the database. However the fundamental problem of SETM algorithm is size of candidate item sets together with transaction ID. In 1994, R. Agrawal and R. Srikant proposed the most popular Apriori algorithm for mining association rules [18]. It generates candidate item sets C_k in a pass k using only large item sets L_{k-1} in the previous pass. The idea rests on the fact that any subset of a frequent item set must be frequent. AprioriTid [18] an improvement to Apriori algorithm also generates candidate item sets like Apriori algorithm. But database is not used for counting supports after first pass; rather vertical format of set C_k is used for this purpose. Repetitive scanning of transactional database and generation of large candidate item sets are the major drawbacks of Apriori like algorithms. So improvement to Apriori is essential concern. Different methods have been developed by different researchers in order to address the problem of repetitive scanning and generation of large candidate item sets. Savasere et al. [24] introduced Apriori like PARTITION algorithm to implement parallel processing of datasets and uses set intersections to determine support values. DHP using hash technique for partitioning to reduce repetitive scans of database is developed by J.S. Park et al. [7]. In 1997, new algorithm Eclat [13] was discovered by M. J. Zaki for fast discovery of association rules. Latter he studied the use of lattice theory and introduced an algorithm [12] based on vertical database format which performs significantly better than Apriori. DIC [23] is further variation of Apriori algorithm proposed by S.Brin et al. Prefix tree is employed for strict separation between counting and generating candidates. Interlocking support determination and candidate generation decreases the database scans. In 1998 C. C. Aggarwal and P.S. Yu [2] introduced a method, which uses adjacency among item sets and does not use vertical database format. In this method special threshold called primary threshold is defined. It avoids generating redundant association rules however it does not perform better for support thresholds lower than the primary threshold. CARMA [3] continuous association rule mining algorithm developed by C. Hidber allows user to change the support threshold and continuously displays the resulting association rules in first phase and in second phase it determines precise support of item sets to extract frequent item set. CARMA is faster than Apriori at very low support threshold only. New way of mining association rules using trie structure to preprocess the database was presented by A. Amir et al. [1]. It is based on Rymon's set enumeration tree search. All transactions are mapped into trie structure with all support counts of items. Frequent item sets are generated by traversing trie structure using depth-first search. F. Coenen et al. [4] improved the method proposed by A. Amir et al. by considering trie structure that contains partial totals of the support counts of item sets. However it requires another step to sum up partial counts in order to obtain the actual support of an item set. Yew-Kwong Woon et al. [25] introduced an algorithm with structure called SOTrieIT (Support-ordered Trie Item set) for the fast discovery of frequent item set. The fast association rule mining algorithm for dynamic updated databases is proposed by Ni Tain-quan et al. [16] to overcome the difficulty of updating frequent item sets in the dynamic database. Many attempts have been done with bitmap techniques in the association rule algorithm [9, 10, 13-15, 17] to improve the performance to find association rules. Intersection and count operations of bitmap offer fast computation with efficient storage. The well known Frequent Pattern growth (FP-growth) algorithm [6] also gives good results. It maintains Frequent Pattern-tree (FP-tree) of database. This generates frequent item set without generation of candidate item set and reduces multiple times scanning of database. It is compact as all infrequent items are removed and highly frequent items share nodes in the tree. The implementation of FP-tree is very much complex. Therefore FP-growth only gives better performance at low support thresholds.

III. Proposed Method of Association Rule Mining

Like Apriori algorithm, initially transaction database D is represented by bit matrix in which each row corresponds to an item set for transaction and the columns correspond to the items. This transaction database D and user defined minimum support threshold are used as input to this algorithm. First, frequent item set with length one i.e. L_1 is obtained from transaction database D. Then the pattern table is derived by using items in L_1. The frequency for each pattern from pattern table is counted as logical AND of pattern with transactions in the database which gives output as true. The patterns having frequency count equal to zero are pruned. Next, k item sets (k >= 1) are used to generate (k+1) item sets. For each item set the support is calculated as sum of all frequency count of matched patterns from the pattern table and check if it is frequent. The procedure is repeated until no more frequent item set can be found. Algorithm is given in "Fig. 2"

Working Example

To illustrate the working of proposed method we used transactional database D shown in "Table 3". The transactional database contains 15 transactions with an item set I = {A, B, C, D, E} of five items and bit matrix representation of D is given in "Table 4"

Input: D: Transaction Database, min_sup : user defined minimum support .
Output: L: frequent feature set , R: Association Rules
Begin
Step 1: For each item, count the occurrence from transactional database and add it to frequent one item set L_1 if it satisfies minimum support constraint.
Step 2: P = Generate_ Pattern_Table (L_1)
Step 3: For each i in P do
Assign N = 0
For each j in D do
flag = Pi AND Dj
If flag is True Then N++
End for
Assign Pi.Count = N
If Pi.Count = 0 then ignore Pi
End for
Step 4: k = 2
Step 5: L_{k+1} = Ø;
Step 6: C_{k+1} = Generate_Candidate_Itemset(L_k);
Step 7: For each j in C_{k+1} do
Count = 0;
For each i in P do
If match found (Pi. pattern, j) is True then
Count = Count + Pi.Count
End for
If Count >= min-sup then L_{k+1} = {j}
End for
Step 8: Set k= k +1; Repeat steps 5 to 8 until no more frequent item set is found i.e. L_{k+1} = null
Step 9: L = Ù L_k
Step 10: R = Ø
Step 11: For each l from L_k do
For each x in l do
R = R U {x => (l - x)}
End for
End for
End

Fig. 2. Algorithm for proposed method of association rule mining.

Table 3. Transaction Database Example

Tid	Items
1	{ A , B , E }
2	{ B , D }
3	{ B , C }
4	{ A , B , C , D }
5	{ A , C }
6	{ B , C }
7	{ B , C }
8	{ A , B , C }
9	{ A , B , C }
10	{ A , B , D }
11	{ A , B , C }
12	{ B , C }
13	{ A , B , D }
14	{ A , B , C }
15	{ B , C }

Table 4. Bit Matrix Representation

Tid	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	1	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	0	1	1	0	0
8	1	1	1	0	0
9	1	1	1	0	0
10	1	1	0	1	0
11	1	1	1	0	0
12	0	1	1	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	0	1	1	0	0

Initially, consider the candidate item set of size one C_1 and scan the database to determine the support i.e. occurrence count of each item of C_1. Take item A as an example. Support of A is 9 as occurrence count of A is 9 in transaction database. The support of each item is compared with predefined minimum support threshold i.e. min-supp. Here min-supp is considered as 15%. Since the support of items A, B, C and D are respectively larger than the predefined min-supp, these four items are considered as frequent and added in the frequent 1-itemset L_1 shown in “Table 5”. Further pattern table (“Table 6”) is generated by extending all items in L_1 with all other items from L_1. The frequency for each pattern from pattern table is counted as logical AND of pattern with transactions in the database which gives output as true. The patterns having frequency count equal to zero are pruned. Hence patterns P_3, P_6 and P_9 from “Table 6” are pruned. Next, frequent -2 item set L_2 is obtained using L_1 and pattern table. For each item set from L_2 the support is calculated as sum of all frequency count of matched patterns from the pattern table and check if it is frequent. Matched patterns are found as logical AND of item set with each patterns in the pattern table which gives output as the item set itself. The item set having support count greater than user defined minimum support threshold (min-supp) are considered as frequent item set. Taking item set {A B} as an example, P_1, P_7, P_8 and P_{10} are found as matched patterns. Therefore sum of support counts of all these patterns is counted as support value of item set {A B} which is frequent item set as it satisfies min-supp condition. Item sets {A B}, {A C}, {A D}, {B C} and {B D} obtained from L_1 are frequent as their support counts are respectively larger than pre-defined mini-supp. These five items sets are added to L_2. Same procedure is repeated to obtain frequent 3-item set L_3 using L_2 and pattern table. Item sets {A B C} and {A B D} are found to be frequent hence added to L_3. It is found that L_4 is empty and it is impossible to generate a frequent item set with length five L_5. Since no more frequent item sets can be found the mining process is terminated. Frequent 2-item set and frequent 3-item set are given in “Table 7” and “Table 8” respectively.

Frequent item set = L_1 Ù L_2 Ù L_3.

Frequent item set = {A, B, C, D, {A B}, {A C}, {A D}, {B C}, {B D}, A B C}, {A B D}}

Total Number of items in Frequent item set = 11

Table 5. Frequent 1-ItemSet L_1

Feature	Count
A	9
B	14
C	11
D	4

Table 6. Pattern Table

PID	Pattern	Items				TID	Count
		A	B	C	D		
P_1	{A B}	1	1	0	0	1	1
P_2	{A C}	1	0	1	0	5	1
P_3	**{A D}**	**1**	**0**	**0**	**1**	**Nil**	**0**
P_4	{B C}	0	1	1	0	3,6,7,12,15	5
P_5	{B D}	0	1	0	1	2	1
P_6	**{C D}**	**0**	**0**	**1**	**1**	**Nil**	**0**
P_7	{A B C}	1	1	1	0	8,9,11,14	4
P_8	{A B D}	1	1	0	1	10,13	2
P_9	**{B C D}**	**0**	**1**	**1**	**1**	**Nil**	**0**
P_{10}	{A B C D}	1	1	1	1	4	1

Table 7. Frequent 2-ItemSet L_2

Item Set	Occurrence	Count
{A B}	P_1 +P_7 + P_8 + P_{10}	08
{A C}	P_2+ P_7+P_{10}	06
{A D}	P_8+P_{10}	03
{B C}	P_4+P_7+P_{10}	10
{B D}	P_5+P_8+P_{10}	04

Table 8. Frequent 3-ItemSet L_3

Item Set	Occurrence	Count
{A B C}	P_7+P_{10}	05
{A B D}	P_8+P_{10}	03

Association rules are derived using this frequent item set. Sample rules are shown in “Table 9”. Confidence is calculated for each rule to decide strength of implication. It is desirable to consider only those rules as strong rules having confidence closer to one. These rules can be further evaluated to obtain valuable knowledge.

Table 9. Sample Association Rules

Sr. No.	Association Rule	Confidence
1	A B → C	0.625
2	A C → B	0.833
3	B C → A	0.5
4	A → B C	0.55
5	B → A C	0.35
6	C → A B	0.45
7	A B → D	0.37
8	A D → B	1
9	B D → A	0.75
10	A → B D	0.33
11	B → A D	0.21
12	D → A B	0.75

IV. Performance Evaluation

Both Apriori and proposed algorithm of association rule mining are applied to transaction database given in “Table 3” with user define minimum support threshold as 15% and analyzed their performance as shown in “Table 10”.

Table 10. Performance Evaluation- Apriori Vs Proposed Algorithm

Method	Large Item Set	No. of Items	No of database scans
Apriori Algorithm	{A,B,C,D,{A,B},{A,C}, {A,D}, {B,C},{B,D}, {A,B,C}{A,B,D}}	11	04
Proposed Algorithm using pattern generation	{A,B,C,D,{A,B},{A,C}, {A,D}, {B,C},{B,D}, {A,B,C}{A,B,D}}	11	02

It is found that behavior of the proposed method is much more similar to Apriori algorithm with reduction in multiple times scanning of database. Apriori algorithm uses iterative search. It is simple and easy to implement also. But there are some shortcomings

1. To generate frequent item set with length k, the database should be scanned for k times and system I/O load is high which affects the performance of Apriori algorithm. To improve the performance, the proposed method adopts simple and fast boolean operations for pattern generation to replace complex computations required for generation of frequent item set. This results reduction in multiple times scanning of transaction database.
2. With increasing amount of records Apriori algorithm takes longer time to generate frequent item set as compared with proposed method. Experiment is carried out with increase number of records by 50 for fixed 5 items.

Table 11. Performance Evaluation- Apriori Vs Proposed Algorithm

No of records	Apriori Algorithm (Time in Sec)	Proposed Algorithm (Time in Sec)
100	35	6
150	53	9
200	71	12
250	88	14
300	105	17

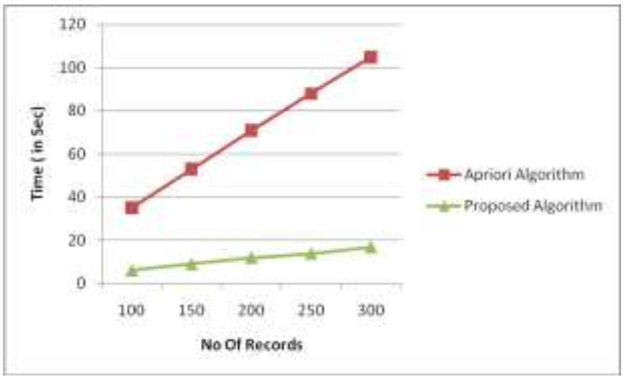

Fig. 3. Graph 1 :No. of Records Vs. Time

3. If minimum support threshold set to high, data covered will be less and less associations can be formed. If minimum support set to low, data covered will be more and many associations can be formed.

"Table 12" shows comparative study for 15 transactions with increase in user defined minimum support threshold.

Table 12. Performance Evaluation- Apriori Vs Proposed Algorithm

Minimum Support	No. of iterations required (Apriori Algorithm)	No. of iterations required (Proposed Algorithm)
15%	360	225
30%	184	115
60%	184	115
75%	104	90
80%	104	90

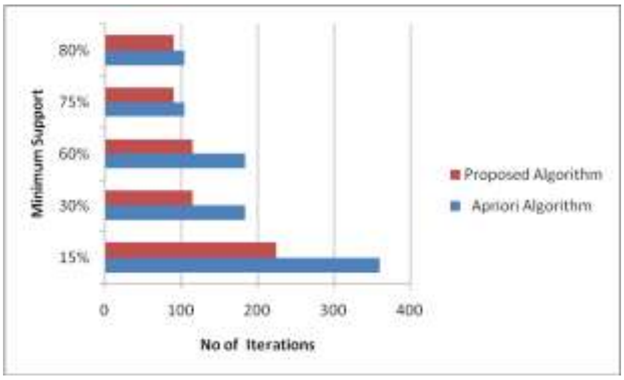

Fig. 4. Graph 2 :No. of Iterations Vs. Minimum Support

V. Conclusion

In this paper a novel association rule mining algorithm based on pattern generation is proposed and manifests good performance in the preliminary experiment. It doesn't need complex computation to deal with frequent item set generation process. The proposed method adopts the simple and fast boolean operations for pattern generation to discover frequent feature set. We evaluated the performance of proposed method with traditional Apriori algorithm. Result shows that behavior of the proposed method is much more similar to Apriori Algorithm with significantly reduced I/O overhead due to reduction in repetitive passes over transaction database.

References

[1] A. Amir, R. Feldman, and R. Kashi,"A new and versatile method for association generation", Information Systems, Vol. 22, no. 6, pp.333-347, 1999
[2] C. C. Aggarwal and P.S.Yu, "Online generation of association rules", Proc.14th Int'l Conf. Data Engg. pp 402-411, 1998.
[3] C. Hidber ,"Online Association Rule Mining", proc ACM SIGMOD Conf. pp 145-154,1999.
[4] F. Coenen, G. Goulbourne and P.H. Leng, "Computing Association Rules Using Partial Totals", proc. 5th European Conf. Principles and Practice of Knowledge Discovery in Databases ,pp.54-66,2001
[5] J. Han, M. Kamber (2001), *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, CA,SIGMOD Conf.,pp-1-12,2000
[6] J. Han, J. Pei, and Y. Yin,"Mining frequent patterns without candidate generation", Proc. ACM SIGMOD International Conference on Management of Data, May 16-18, Dallas, Texas, pp 1-12
[7] J. S. Park, M.S. Chen P.S. Yu," Using Hash-based Method with Transaction Trimming for Mining Association Rules", IEEE Trans. Knowledge and Data Engg., vol 9,no. 5, pp. 813-824, Sept/Oct 1997.
[8] K. Wang, Y. H, J. Han, "Mining frequent item sets using support constraints", in Proceedings of the 26th International Conference on Very Large Data Bases, 2000, pp. 43-52
[9] Lin T.Y.," Data Mining and Machine Oriented Modeling: A Granular Computing Approach", Journal of Applied intelligence, Oct 2000.

[10] Lin T.Y., Lounie E,"Finding Association Rules using Fast bit Computation: Machine-Oriented Modeling",IS-MIS-2000.
[11] M. Houtsma and A. Swami," Set oriented mining of association rules", Research report RJ 9567, IBM Almaden Research Center, San Jose, California, Oct 1993.
[12] M. J. Zaki, "Scalable algorithms for association mining", IEEE Trans. Knowledge Data Eng. ,vol. 12,no. 3,pp 372-390 May/June 2000.
[13] M. J. Zaki et al., "New Algorithms for fast discovery of Association Rules", in KDD-97
[14] M. J. Zaki ," Generating non-redundant association rules", in KDD-2002
[15] Morzy T., Zakrzewiez M.," Group Bitmap Index: A Structure for Association Rule Retrieval", Prod. of the 4th International Conf. on Knowledge Discovery and Data Mining (KDD-98).
[16] Ni Tian-quan, Wang Jain-dong, Peng Xiao-bing and Liu Yian, "A fast association rules mining algorithm for dynamic updated databases", Inform. Tech. J. 8 (8); 1235-1241, 2009, ISSN 1812-5638.
[17] P. Shenoy, J.R. Haritsa, S. Sudarshan, G.Bhalotia, M. Bawa and D. Shah ,"Turbo-Charging Vertical Mining of Large Databases", Proc. ACM SIGMOD Conf.,pp.22-33,2000
[18] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", Proc. 20th Int'l. Conf. Very Large Databases, pp.487-499,1994
[19] R. Agrawal, R. Srikant, "Mining sequential patterns", The Eleventh IEEE International Conference on Data Engineering, 1995, pp. 3-14
[20] R. Agrawal, R. Srikant, Q, Vu, "Mining association rules with item constraints", The Third International Conference on Knowledge Discovery in Databases and Data Mining, 1997, pp.67-73
[21] R. Agrawal, T. Imielinski, A. Swami, "Database mining: a performance perspective", IEEE Transactions on Knowledge and Data Engineering.5 (6) (1993) 914-925.
[22] R. Agrawal, T. Imielinski, A. Swami, "Mining association rules between sets of items in large databases", In Proc of the ACM SIGMOD International Conference on Management of Data, Jun 1993,Washington, D.C.,USA,pp.207-216.
[23] S. Brin, R. Motwani, J.D. Ullman and S. Tsur, " Dynamic itemset counting and implication rules for market basket data", In Proc. of ACM SIGMOD Int'l conference on Management of Data ,1997,26(2),255-264
[24] Savasere A.,E. Omiecinski and S. Navathe," An efficient algorithm for mining association rules in large database", proc. of 21st international Conf. on Very Large Database,(ICLD 1995), Zurich, Swizerland,pp. 432-443.
[25] Yew-Kwong Woon, Wee-Keong Ng, and Ee-Peng Lim,"A support ordered trie for fast frequent item set discovery", IEEE Trans. On Knowledge And Data Engineering", Vol.16, No.7, July 2004
[26] Predrag Stansic and Savo Tomovic," Mining association rules from transaction databases and Apriori multiple algorithm",ISBN : 978-972-8924-68-3 @ 2008 IADIS
[27] Hinbing Liu and Baisheng Wang," An association rule mining algorithm based on Boolean matrix", Data Science Journal, Vol. 6, Supplement, 9, Sept 2007
[28] Ulrich Guntzer, Jochen Hipp and Gholamreza Nakhaeizadeh, "Algorithms for Association Rule Mining-A General Survey and Comparison", ACM SIGKDD, Vol.2,Issue 1,July 2000,pp.58-64
[29] Thabet Slimani and Amor Lazzez," Efficient Analysis of Pattern and Association Rule Mining Approaches", International Journal of Information Technology and Computer Science,2014,03,70-81
[30] Liu X., Zhai K., & Pedrycz W. "An improved association rules mining method", *Expert Systems with Applications*, 2012 *39*(1):1362–1374. doi:10.1016/j. eswa.2011.08.018
[31] Gouda, K. and Zaki,M.J. GenMax, "An Efficient Algorithm for Mining Maximal Frequent Itemsets", Data Mining and Knowledge Discovery, 2005, 11: 1-20.
[32] Grahne G. and Zhu G, "Fast Algorithms for frequent itemset mining using FP-trees", in IEEE transactions on knowledge and Data engineering, 2005,17(10):1347-1362.
[33] Kamrul Shah, Mohammad Khandakar, Hasnain Abu. "Reverse Apriori Algorithm for Frequent Pattern Mining", Asian Journal of Information Technology, 2008, :524-530, ISSN: 1682-3915.
[34] Praksh S., Parvathi R.M.S, "An enhanced Scalling Apriori for Association Rule Mining Efficiency", European Journal of Scientific Research, 2010, 39:257-264, ISSN: 1450-216X.
[35] Rao, S., Gupta, P. "Implementing Improved Algorithm Over Apriori Data Mining Association Rule Algorithm", IJCST.2012, 3 (1), 489-493.

Authors' Profile

Deepa S. Deshpande has obtained her Bachelor of computer engineering in 1995 and M.Tech in computer engineering in 2006 from Pune University. Currently she is pursuing Phd. in the area of image mining at SRTMU University. Presently she is working as Associate Professor in Computer Science and Engineering department of MGM's Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India. She has industrial experience of 1 year and teaching experience of 16 years. Her area of interest is Database Management Systems, Data Warehousing, Data Mining, Computer Vision, Image Mining. She is life member of professional societies like CSI, ISTE and QCFI.

How to cite this paper: Deepa S. Deshpande,"A Novel Approach for Association Rule Mining using Pattern Generation", International Journal of Information Technology and Computer Science(IJITCS), vol.6, no.11, pp.59-65, 2014. DOI: 10.5815/ijitcs.2014.11.09