International Journal of Database and Theory and Application
Vol.9, No.2 (2016), pp.193-204
http://dx.doi.org/10.14257/ijdta.2016.9.2.21

ARM Amelioration Based On Artificial Bee Colony

Sourabh Sahota and Prince Verma

Dept. of CSE
CT Institute of Engg., Mgt & Tech.
Punjab(India)

Abstract

Association rule mining which is the most significance and use is one of a relevant approach for data mining. The fundamental of the association rule mining approach have been Apriori and introduce many access with changes in the apriori but though main idea continue to be the same that is use of support and confidence threshold (s). Conforming to the theory it is well know that no work has been done in the domain of Enhancing pruning step of Apriori. This paper introduces a new algorithm M-APRIORI. This algorithm advances to Enhance the Apriori algorithm by using mean support (supmean) rather than minimum support (supmin), to produce probable item-set instead of large item-set and Artificial bee Colony technique used to optimization the rules. In this paper Apriroi and M-Apriori are based On Artificial Bee Colony.

***Keywords** – Data Mining, KDD Process, Association Rule Mining, ABC (Artificial Bee Colony), Apriori, M-Apriori.*

1. Introduction

Data Mining is characterized as extracting information from extensive type of data. In another words, data mining is the technique of learning from data. It is the method to distinguish the learning or shrouded example frame extensive databases. Data Mining otherwise called Knowledge Discovery in Databases (KDD), allocated to the nontrivial are group of verifiable, effectively obscure and possibly supportive information from data in databases. As Data mining and learning revelation in databases (or KDD) are regularly regarded as option, So, data mining solidly is some portion of the data disclosure process. Data mining can be used by businesses in some manner. *Customer profiling* to Determine those subset of customers virtually beneficial to the business. *Targeting* to Determinative the characteristics of beneficial customers who have been recaptured by contender and *Market-basket analysis* in order to Determinative product subscribe by consumer which can be used for product orienting and get across –selling. These are not only the applications of data mining but are useful in businesses application also.

1.1 Association Rule Mining

Association Rule Mining is an data mining scope which asset the action of co-event of thing at value-based database conferred by Rakesh Agrawal in 1993. Affiliation standard mining, is a standout amongst the most vital and securely investigated strategies of information mining. It objective is to concentrate the fascinating relationships, incessant examples, affiliations or easygoing structures among sets of things in exchange databases or other data storehouses. There are two important basic scales for association rules, support(s) and Confidence(c).

Through the database is enormous and clients thought with respect to just those generally bought things, customarily edges of bolster and certainty are pre-characterized by clients to still those principles that are not all that fascinating or helpful. The two edges are called insignificant backing and negligible certainty serially, additional requirements

ISSN: 1975-0080 IJDTA

of intriguing standards likewise can be determined by the clients. The two crucial parameters of Association Rule Mining(ARM) are: support and confidence.

Support(s) of a association rule is defined like the percentage or fraction of transaction in D in order to contain X U Y. Support(s) can be calculated by the following formula:

$$Sup(X\ U\ Y) = \frac{\text{count (X U Y)}}{\text{count (D)}}$$

Confidence is an scale of power of the association rules. Confidence is defined at percentage or fraction of the number of transactions in D that comprise X also comprise Y. 'IF' component is Antecedent and 'THEN' component is consequent. It can be calculated by dividing the possibility of items decrescent together to the possibility of occurrence of antecedent. Confidence(c) is calculated by the following formula.

$$conf(X \Rightarrow Y) = \frac{\text{sup(X U Y)}}{\text{sup(x)}}$$

1.2 Artificial Beef Colony

Artificial Bee Colony Algorithm is an optimization algorithm introduced by the intelligent foraging behavior of honey bee swarm, introduced by Karaboga in 2005. Artificial Bee Colony research for natural behavior of real honey bees in food foraging. Honey bees used many structure like waggle dance to optimally detect food sources and to finding new ones. To generate a good candidate for developing new intelligent investigation algorithms. In the Artificial Bee Colony algorithm, the colony of artificial bees comprise three groups of bees: 1. Employed bees, 2. Onlookers and 3. Scouts. Each cycle of the search contain three step:

Step 1 : Inset the employed bees onto the food sources and then determine their nectar amounts.

Step 2 : Choose the food sources by the onlookers after sharing the information of employed bee sand determining the nectar amount of the foods.

Step 3 : Characterizing the scout bees and Inset them onto the randomly characterizing food sources. In the Artificial Bee Colony, a food source situation produce a doable explanation to the problem to be optimized and the nectar amount of a food source match to the quality (Fitness) of the associated explanation.

1.3 APRIORI Algorithm

The Apriori algorithm produce the candidate item sets in one pass by using only the large item-set of the foregoing pass without consideration of the transactions of database. The basic concept used here is the whole subset of a large item-set is necessarily large. The kth pass candidate item-sets with k items is produce from previous pass by joining large item-sets of K-1 items (candidate descent), and deleting those item-sets that comprise any subset that is not large. This pruned procedure results in descent of a much smaller number of candidate item-sets.

- Algorithm Apriori(large 1 itemsets)
- L1={large 1 itemsets};
- **for**(k=2;Lk-1≠∅;k++) **do begin**
- Ck=apriori-gen(Lk-1); //New candidates
- **forall**transactions t ∈ D **do begin**
- Ct=subset(Ck,t); //Candidatescontained in *t*
- **forall**candidates c ∈ Ct **do**

- c.count++;
- **end**
- Lk={c ∈ Ck | c.count≥minsup}
- **end for**
- Answer=UkLk
- Algorithm 1:Apriori Algorithm

1.3.1 Apriori Explanation

As Apriori algorithm produce the candidate item-sets without seeing the transaction in the database. For Example: we are provide with a database D (Table 1) with some set of transaction, supmin =61% and conf min = 70%

Table 1. Database D

Transaction	Items	Transaction	Items
1	T_1,T_4,T_5	11	T_1,T_2,T_3,T_4
2	T_1,T_2,T_3,T_4	12	T_3,T_4,T_5
3	T_5	13	T_1,T_2,T_3,T_4
4	T_1,T_2,T_3,T_4	14	T_5
5	T_1,T_2,T_3,T_4	15	T_1,T_2,T_3,T_4
6	T_1,T_2,T_4	16	T_1,T_2,T_3,T_4,T_5
7	T_1,T_2,T_3,T_4,T_5	17	T_1,T_2,T_4,T_5
8	T_3,T_4	18	T_3,T_4,T_5
9	T_1,T_2,T_3,T_4	19	T_1,T_2,T_3,T_4,T_5
10	T_2,T_3,T_4	20	T_1,T_2

Before creating large item-sets (L1, L2 & so on) and candidate item-sets (C2 & so on) candidate, firstly Candidate item-set, C1 is generated from database (i.e. Table 2 from Table 1) and then, Large item-set, L1 is created from C1 using supmin=61% (i.e. Table 3 from Table 2)

Table 2. Candidate Item-set C1

Candidate Item-set, C1	Support
T_1	65%
T_2	70%
T_3	75%
T_4	80%
T_5	55%

Table 3. Large Item-set L1

Large Item-Set, L1	Support
T_1	65%
T_2	70%
T_3	75%
T_4	80%

Step 1: Candidate set is produce as follows:

- The candidate item-sets, C_k can be produce by joining large item-sets L_{k-1}items, and
- Deleting those that comprise any subset that is not large.
  In Example $C_2$is produce from L_1 items by join procedure (i.e. Table 4 from Table 3) and those item-sets are deleted that have some (k-1) subset of c is not in L_{k-1} where c € C_k.

Step 2: Large item-set, L_k is produce form candidate item-set, C_k using $sup_{min.}$
The above two steps are repeated until Large Item-set came to be empty.
Here from the candidate item-sets, C2 elements with sup ≥supmin, Large item-set, L2 is created (Table 5). As all candidate item-sets, C3 ha sup ≥ supmin, so all became Large item-set, L3 (Table 5 from Table 6)

Table 4. Candidate Item-set C2

Candidate Item-set, C1	Support
T_1,T_2	65%
T_1,T_3	50%
T_1,T_4	55%
T_2,T_3	60%
T_2,T_4	65%
T_3,T_4	70%

Table 5. Large Item-set L2

Large Set L2	Support
T_1,T_2	65%
T_3,T_4	70%

Large Item-set: (T_1,T_2), (T_3,T_4) & Association Rules are:

Table 6. Association Rules

Large Set	Association Rules	Confidence
T_1,T_2	T_1=>T_2	65/65=100%
	T_2=>T_1	65/70=92.8%
T_3,T_4	T_3=>T_4	70/75=93.3%
	T_4=>T_3	70/80=87.5%

2. Proposed Approach (M-APRIORI)

The apriori algorithm must be user defined threshold values, i.e. minimum support (sup min) and minimum confidence (Confmin). Supmin is must be to produce the large item-set from candidate set and confmin is in need to produce required set of association rule form produce large item-set. The procedure to build large item-set in apriori by cut out of not so expected candidate sets is reliant on Supmin threshold implement by user. But a few vital rules get pruned due to this user-defined threshold as user doesn't know that at which threshold valuable association rules get generated.

So, an algorithm is suggested named "M-APRIORI". The prospective algorithm targets on pruning in Enhance the process of Apriori algorithm by using mean support (supmean) rather than minimum support (supmin), to produce probable item-set rather than large item-set. Here mean support is not user defined value but it is calculated using the formula.

$$\text{sup mean} = \frac{\sum_{k=1}^{n}(\text{sup}(k))}{n}$$

Where n is the number of items involved. So this formula would lead to better item-sets and this increases the number of preferable association rules than Apriori algorithm.

2.1 M-APRIORI Algorithm Explanation

The prospective algorithm target to Enhance the process of Apriori algorithm by using mean support (supmean) instead of minimum support (supmin), to produce probableitemset instead of large item-set. For Example: we are provided with a database D (Table 1) with some set of transaction and confmin=70%. Before crating Probable item-sets (P1,P2& so on) and candidate item-sets (C2, C3 & so on) candidate, firstly candidate item-set, C1 is generated from database (i.e. Table 8 form Table 7). Then, Would-be Probable item-set, P1 is created from C1 (i.e. Table 9 from Table 8). The item-sets with sup ≥ sup mean are inserted into Probable item-set. i.e.

$$Using\ supmean = \frac{65 + 70 + 75 + 80 + 55}{5} = 69\%$$

Now emphasis are performed on item-sets with sup ≤ supmean, item-set and sup ≥ Sp (here Sp = supmean/1.2 = 69/1.2=57.5%) i.e. item A. So, items with sup ≥ Sp is {A} with sup=65%). From these, items with probable > 80% are taken into Probable Item-set. i.e. If items with sup ≥ Sp have maximum sup, say Sp-max then items with support greater than (80% of Sp-max) are inserted into probable item-set, while others are pruned. (here Sp-max=65% and 80% of Sp-max=52%). So, Probable Item-set P1 is given in Table 10.

Table 7. Database D

Transaction	Items	Transaction	Items
1	T_1,T_4,T_5	11	T_1,T_2,T_3,T_4
2	T_1,T_2,T_3,T_4	12	T_3,T_4,T_5
3	T_5	13	T_1,T_2,T_3,T_4
4	T_1,T_2,T_3,T_4	14	T_5
5	T_1,T_2,T_3,T_4	15	T_1,T_2,T_3,T_4
6	T_1,T_2,T_4	16	T_1,T_2,T_3,T_4,T_5
7	T_1,T_2,T_3,T_4,T_5	17	T_1,T_2,T_4,T_5
8	T_3,T_4	18	T_3,T_4,T_5
9	T_1,T_2,T_3,T_4	19	T_1,T_2,T_3,T_4,T_5
10	T_2,T_3,T_4	20	T_1,T_2

Table 8. Candidate Item-set C1

Candidate Item-set, C1	Support
T_1	65%
T_2	70%
T_3	75%
T_4	80%
T_5	50%

Table 9. Probable Item-set P1

Probability Item-set, P1	Support
T_2	70%
T_3	75%
T_4	80%

Table 10. Probable Item-set P1

Probability Item-set, P1	Support
T_1	65%
T_2	70%
T_3	75%
T_4	80%

Table 11. Candidate Item-set C2

Candidate Item-set, C2	Support
T_1,T_2	65%
T_1,T_3	50%
T_1,T_4	55%
T_2,T_3	60%
T_2,T_4	60%
T_3,T_4	70%

Candidate item-set C2 is generated from P1 (Table 9 from Table 10). Here from the candidate item-sets, C2 elements with sup ≥supmin, Probable item-set, P2 is created (Table 12 from Table 11). And for Probable item-set generation same process as above is repeated.

Table 12. Probable Item-set P2

Probability Item-set, C2	Support
T_1,T_2	65%
T_1,T_4	55%
T_2,T_3	60%
T_2,T_4	60%
T_3,T_4	70%

Table 13. Candidate Item-set C3

Candidate Item-set, C3	Support
T_1,T_2,T_4	55%
T_2,T_3,T_4	50%

From here P3 is created containing (T_1,T_2,T_4). As its sup > 52% and (T_2,T_3,T_4) is pruned (see Table 14).

Table 14. Probable Item-set P3

Probability Item-set, C3	Support
T_1,T_2,T_4	55%

Probable Item-set: {(A,B), (B,C), (B,D), (C,D), (A,B,D)}. So, (B,C), (B,D) are more in Probable item-set than large item-set of Apriori algorithm in Section and association rule are prescribed in Table 15.

Table 15. Association Rules

Large Set	Association Rules	Confidence
T_1,T_2	T_1=>T_2	65/65=100%
	T_2=>T_1	65/70=92.8%
T_1,T_4	T_1=>T_4	55/70=78.5%
	T_4=>T_1	55/80=68.7%
T_2,T_3	T_2=>T_3	60/70=85.7%
	T_3=>T_2	60/75=80%
T_2,T_4	T_2=>T_4	60/70=85.7
	T_4=>T_2	60/80=75%
T_3,T_4	T_3=>T_4	70/75=93.3%
	T_4=>T_3	70/80=87.5%
T_1,T_2,T_4	(T_1,T_2)=>T_4	55/65=84.6%
	(T_1,T_4)=>T_2	55/55=100%
	(T_2,T_4)=>T_1	55/60=91.6%

2.2 Designing Artificial Bee Colony (ABC) Optimization for Apriori & M-Aprior

Artificial Bee Colony has not been utilized on ARM's Apriori Algorithm for the optimization of results. So, is implemented ABC on Apriori & then on M-Apriori for the better optimization of results. The approach utilized ABC over Aprior & M-Apriori as follows. Each rule have antecedent(left side) and consequent (right side) parts respectively. We assume consequent part of rule as bees and antecedent part's first attribute as Food. Each rule has some occurrence the most occurrences are assumed as fitness value for employee bees from which group's best rule is found this occurrence is assumed as the information which is passed futher to onlooker bees for better results. The onlooker bee finds occurrences & perform any updation to rules that are discovered having same food. E.g. Suppose we see first rule, second rule and third rule has same food then among them that rule is best which have highest occurrence value in transaction set.

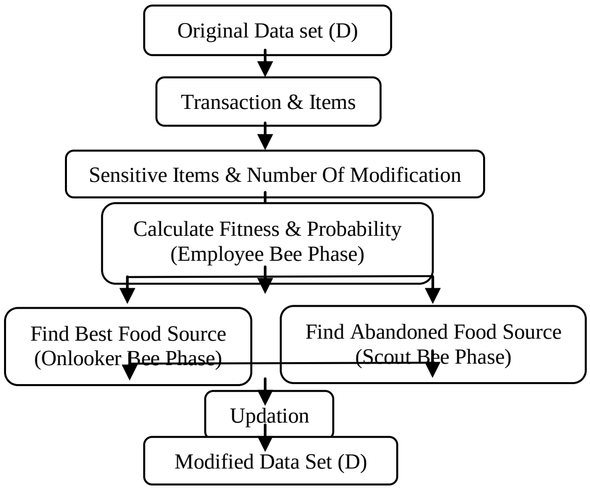

Figure 1. Artificial Bee Colony Optimization

So that path is mostly followed by bees and it is shortest path for food source. At the end we get only those rules which are optimized by Artificial Bee Colony.

3. Experimental Results

To estimate the efficiency of proposed algorithm the performance is broadly analyzed in comparison to the Apriori algorithm. The parameters for comparison between Apriori, Apriori with ABC(Artificial Bee Colony) and M-Apriori, M-Apriori with ABC(Artificial Bee Colony) would be:

1. Minimum, Maximum & Mean confidence
2. Time Consumed
3. Number of Rules generated
    a) Total
    b) With confidence≥0.9
    c) With confidence≥0.8
    d) With confidence≥0.7
    e) With confidence≥0.7

The Test Database to be used for the objective of comparison with the Apriori algorithm is accurate and recognized i.e. Smoking Dataset and Weather dataset.

3.1 SMOKING DATASET

The Dataset helps to know that patient should be fitted with hard contact lenses, soft contact lenses or none.

Smoking Dataset are shown in Table (Min-Max Confidence, Total Rules & Time required) & Table (Rules with Conf≥90%, Conf≥80, Conf≥70, Conf≥70).

From the results it is clear that Minimum , Maximum & Mean confidence is same in case of M-Apriori as some cases of apriori Algorithm (with minsup = 0.11,0.12, 0.14, 0.15).The rules generated in M-Apriori are also same as these above defined cases of apriori. But in M-Apriori is better in case than Apriori when minsup=0.11. Also it is not needed to find the minimum support for finding the best rules in M-Apriori as in the case of Apriori.

This Table Show the Smoking Dataset Results Before ABC

Smoking Dataset Results before ABC

Smoking Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .11	*Min sup=0 .12*	*Min sup=0 .14*	*Min sup=0 .15*	*Mean sup=0 .1308*
Confmin	0.1610	0.1610	0.1610	0.1722	0.1610
Confmax	1.0	1.0	1.0	1.0	1.0
Confmean	0.5303	0.5322	0.5343	0.5208	0.5303
Total Rules	162	160	158	142	162
Time Required (Milliseconds)	178	150	148	154	328
Total Time Required(Milliseconds)=630					

Number of Rules (Smoking Dataset) before ABC

Smoking Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0. 11	*Min sup=0. 12*	*Min sup=0. 14*	*Min sup=0. 15*	*Mean sup=0 .1308*
Rules with conf>= 0.9	14	14	14	13	14
Rules with conf>= 0.8	13	13	14	12	13
Rules with Conf>= 0.7	10	10	10	10	10
Rules with conf>= 0.7	125	123	121	107	125

This Table Show the Smoking Dataset Results After ABC.

Smoking Dataset Results after ABC

Smoking Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .11	*Min sup=0 .12*	*Min sup=0 .14*	*Min sup=0 .15*	*Mean sup=0 .1308*
Confmin	0.1610	0.1610	0.1610	0.1722	0.3753
Confmax	0.9	0.9	0.9	0.9	0.6
Confmean	0.4840	0.4864	0.4891	0.5047	0.3753
Total Rules	92	90	88	82	52
Time Required (Milliseconds)	211	231	221	215	731
Total Time Required(Milliseconds)=878					

Number of Rules (Smoking Dataset) after ABC

Smoking Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .11	*Min sup=0 .12*	*Min sup=0 .14*	*Min sup=0 .15*	*Mean sup=0 .1308*
Rules with conf>= 0.9	6	6	6	5	0
Rules with conf>= 0.8	9	9	10	8	0
Rules with Conf>= 0.7	1	1	1	2	0
Rules with conf>= 0.7	76	74	72	67	52

3.2 Weather Dataset

Weather Dataset are shown in Table (Min-Max Confidence, Total Rules & Time required) & Table (Rules with Conf≥90%, Conf≥80, Conf≥70, Conf≥70).

From the results we came to know that Minimum , Maximum & Mean confidence is same in case of M-Apriori as some cases of apriori Algorithm (with minsup = 0.14,0.15, 0.17, 0.18).The rules generated in M-Apriori are also same as these above defined cases of apriori. But in M-Apriori is better in case than Apriori when minsup=0.14. Also needed to find the minimum support for finding the best rules in M-Apriori as in the case of Apriori.

The Dataset helps to know that the temperature .The temperature is clear or not in the sky.

This Table Show the Smoking Dataset Results Before ABC

Weather Dataset Result before ABC

Weather Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .14	*Min sup=0 .15*	*Min sup=0 .17*	*Min sup=0 .18*	*Mean sup=0 .1654*
Confmin	0.1972	0.3115	0.3115	0.3214	0.1972
Confmax	1.0	1.0	1.0	1.0	1.0
Confmean	0.7237	0.7452	0.7452	0.7657	0.7237
Total Rules	656	490	490	398	656
Time Required (Milliseconds)	392	282	286	231	832
Total Time Required(Milliseconds)=1191					

Number of Rules (Weather Dataset) before ABC

Weather Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .14	*Min sup=0 .15*	*Min sup=0 .17*	*Min sup=0 .18*	*Mean sup=0 .1654*

Rules with conf>= 0.9	283	194	194	160	283
Rules with conf>= 0.8	11	11	11	11	11
Rules with Conf>= 0.7	62	62	62	58	62
Rules with conf>= 0.7	300	223	233	169	300

This Table Show the Weather Dataset Results After ABC

Weather Dataset Result after ABC

Weather Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .14	*Min sup=0 .15*	*Min sup=0 .17*	*Min sup=0 .18*	*Mean sup=0 .1654*
Confmin	0.1972	0.3115	0.3115	0.3214	0.1972
Confmax	1.0	1.0	1.0	1.0	1.0
Confmean	0.7355	0.7264	0.7264	0.7600	0.6682
Total Rules	180	143	143	111	105
Time Required (Milliseconds)	575	453	528	361	1561
Total Time Required(Milliseconds)=1917					

Number of Rules (Weather Dataset) after ABC

Weather Apriori Dataset	*Apriori*				*M-Apriori*
	Min sup=0 .14	*Min sup=0 .15*	*Min sup=0 .17*	*Min sup=0 .18*	*Mean sup=0 .1308*
Rules with conf>= 0.9	81	55	55	46	34
Rules with conf>= 0.8	3	4	4	4	1
Rules with Conf>= 0.7	16	16	16	14	14
Rules with conf>= 0.7	80	68	68	47	56

4. Conclusion

In this paper the effective adjustment of the Apriori for rules generation and optimization is displayed by applying ABC technique with mean support is based on ABC(Artificial Bee Colony) unique in several means than actual apriori. In addition , the simulation results show that the proposed technique converges better quality solutions in case of confidence (max,min or mean) and in number of rules (total rules, rules with confidence >90%). In the future the focus would be to implement the ABC(Artificial Bee Colony) for ECLAT algorithm to produce better quality solutions.

References

[1] K. Wang, M. Stan and K. Skadron, "Association Rule Mining with the Micron Automata Processor", IEEE International Parallel & distributed Processing Symposium, **(2015)**.

[2] B. D. Dubey, M. Sharma and R. Shah, "Comparative Study of Frequent Item Set In Data Mining", International Journal of Programming Languages and Applications (IJPLA), vol. 5, no. 1, **(2015)**.

[3] N. J. Ghalehsefidi and M. N. Dehkordi, "A Survey on privacy preserving association rule mining", ACSJJ Advances in Computer Science: an International Journal, vol. 4, no. 14, **(2015)**.

[4] H. Jafarzadeh and R. R. Torkashvand, "Provide a New Approach for Mining Fuzzy Association Rules Using Apriori Algorithm", Indian Journal of Science and Technology, vol. 8, no. 8, **(2015)**, pp. 707-714.

[5] Ms. Dh. Patel and Prof. K. Sarvakar, "Algorithms for Frequent Pattern Mining An Analysis", International Journal of Enhanced Research in Science Technology & Engineering ISSN: 2319-7463, vol. 4, no. 4, **(2015)**.

[6] W. Zakaria, Y. Kotb and F. Ghaleb, "MCR-Miner: Maximal Confident Association Rules Miner Algorithm for Up/Down-Expressed Genes", Application Math Information Science, vol. 8, no. 2, **(2014)**, pp. 799-809.

[7] J. Singh, H. Ram and Dr. J. S. Sodhi, "Improving Efficiency of Apriori Algorithm Using Transaction Reduction", International Journal of Scientific and Research Publications, vol. 3, no. 1, **(2013)**.

[8] R. P. N. S. Mehta, "A Survey on Mining Algorithms", International Journal of Soft computing and Engineering, vol. 2, no. 6, **(2013)**, pp. 460-46.

[9] S. Nagpal, "Improved Apriori Algorithm using logarithmic decoding and pruning", International Journal of Engineering Research and Applications, vol. 2, no. 3, **(2012)**, pp. 2569-2572.

[10] Y. Ramamohan, K. Vasantharao, C. K. Chakravarti and A. S. K.Ratnam, "A Study of Data Mining tools In Knowledge Discovery Process", International Journal of Soft Computing and Engineering(IJSCE) ISSN:2231-2307, vol. 2, no. 3, **(2012)**.

[11] S. A. Abaya, "Association Rule Mining Based On Apriori Algorithm in Minimizing Candidate Generation", International Journal of Scientific & Engineering Research, vol. 3, no. 7, **(2012)**.

[12] S. A Abaya, "Association Rule Mining Based on Apriori Algorithm in Minimizing Candidate Generation", International Journal of Scientific & Engineering Research, vol. 3, no. 7, **(2012)**.

[13] S. Vijayarani and M. S. Prabha, "Association Rule Hiding Using Artificial Bee Colony Algorithm", International Journal of Computer Application, (1975-8887), vol. 33, no. 2, **(2011)**.

[14] F. H. A. Zawaidah, Y. H. Jbara and M. A. A. A. Zanana, "An improved Algorithm for mining Association rule in large database", World of Computer and Information technology, vol. 1, no. 7, **(2011)**, pp. 311-316.

[15] Z. Chen, S. Cai, Q. Song and C. Zhu, "An Improved Apriori Algorithm based on pruning Optimization and transaction reduction", IEEE transaction on evolution computation, **(2011)**, pp. 1908-1911.

[16] W. Y. Quing, Y. R. Hua and L. P. Yu, "An improved apriori algorithm for association rule mining", IEEE International Symposium, **(2009)**, pp. 942-946.

[17] H. Wu, Z. Lu, L. Pan, R. S. Xu and W. Jiang, "An improved Apriori based algorithm for association rule mining", IEEE Sixth international conference on fuzzy systems and knowledge discovery, **(2009)**, pp. 51-55.

[18] D. Sun, S. Teng, W. Zhang and H. Zhu, "An Algorithm to Improve the Effectiveness of Apriori", IEEE Conference, **(2007)**, pp. 385-390.

[19] J. Gao, S. Li and F. Qian, "Optimization on Association Rules Apriori Algorithm", IEEE Conference, vol. 2, **(2006)**, pp. 5901-5905.

[20] C. Borgelt, "Efficient Implementations of Apriori and Eclat", FIMI '03' Proceedings of the IEEE ICDM Workshop on, **(2003)**.

