F3 Algorithm for Association Rules

Rina Raval
Assistant Professor,
Information Technology Department
Silver Oak College of
Engineering and Technology

ABSTRACT
Frequent pattern mining or association rule mining has been very fascinating research topic. It gives association rules which are nothing else but relationships amongst data. These relationships play vital role to make decision in market based analysis, medical applications, banking and many other organizations. Amongst several algorithms provided for frequent pattern mining, time necessitated is always very important aspect to be considered. The breakthrough approach named F3 algorithm, finds frequent patterns by considering quantity of individual item in single transaction rather than item's presence. Afterwards it finds supplementary appealing patterns from profit of items. This approach not only reduces the time for finding frequent patterns, but also endow with new effective pat-terns which act as a key to improve business utility.

General Terms
Apriori Algorithm, Association Rules, Frequent Pattern Mining

Keywords
PW-factor, Q-factor,F3 Algorithm

1. INTRODUCTION
Size of databases has been amplified in modern era. From these databases, an automatic discovery of use-ful patterns is a foremost effort of data mining which is also acknowledged as Knowledge Discovery"[13]. The systematic process designed to explore data in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected pat-terns to new subsets of data is known as data mining. Many techniques have been developed in data mining amongst which primarily frequent pattern mining is very imperative which results in association rules. Frequent pattern mining facilitates to take decisions about the data which are very important in terms of business viewpoint.

2. FREQUENT PATTERN MINING
Frequent pattern mining is an important technique which provides association rules which are used to unearth relationships between apparently unrelated data in a relational database[1]. These relationships are a central point to make decisions about dozens of data in databases, in the various fields like business organizations like banking, marketing, financing, super market and even in medical diagnosis . Frequent pattern mining was first proposed by Agrawal in 1993 for market basket analysis in the form of as-association rule mining[14]. Analysis was done of customer buying habits by finding associations between the different items that customers put in their "shop-ping baskets".

By and large, in the definition , important terms are D = database, T = {T1, T2, T3,…….T m} is the set of transactions, where each transaction is a collect ion of items and I = {i1, i2, i3….in} is set of items where Ti $\subseteq$ I. Item sets containing k items are identified as k-item sets.

Frequent items imp lies the number of times particular item's existence in the transaction which is not less than minimum threshold value given by user. This is also famous as support of item. In other words, support is the number of transactions in which the association rule holds. In terms of probability, it is the percentage of transactions that demonstrate the rule. Suppose the support of an item shampoo is 0.4%, it means only 0.4 percent of the transaction contain purchasing of item shampoo.

Confidence is the other important thing in frequent pattern mining which is a conditional probability that, given R present in transaction, S will also be present. For instance, association rule for "(shampoo, conditioner)→75%" states that, out of five transactions, three customers bought shampoo as well as conditioner, i.e. in three transactions , given item shampoo present, item conditioner will be present, too. Mini-mum confidence threshold will also be given by user.

3. PROBLEM OF FREQUENT PATTERN MINING
To find frequent patterns from a tiny database is quite swift but to find frequent patterns from hefty database is problematic. As database is huge, frequent patterns are also huge and time taken to find those patterns is even more. As minimum threshold support will be decreased, patterns will be myriad. From those patterns, association rules finding will be more complicated as relationships will be mo re as patterns are more. Apriori algorithm is used than time wastage will be done on generating candidate sets every time. In that, most of the time will be taken in scanning database itself. Eclat algorithm will not take as much time as Apriori, but for small databases it is not suitable. In FP Growth algorithm, candidate sets are not generated, so it requires less memory. But, for large databases it does take large execution time because of its complex data structure.

Ultimately, this will be true for any algorithm because it does not matter whichever algorithm is used, but the ultimate frequent patterns will be the same. The time taken to find frequent patterns will be dissimilar according to the type of algorithm but the need for decreasing execution time still remains as some algorithms are good in small database and some are good in larger databases [12]. Therefore, pruning unnecessary items and considering only important items is the unique goal of frequent pattern mining. In due course, the main aim is to focus on some approach which can trim down the time of execution which can be true for any frequent pattern mining algorithm. With that approach, if some

improvement can be done in algorithm itself, than it may give very effective and enhanced version.

Other problem is, in frequent pattern mining only item's existence is taken in to account, and from that support value is counted. But the factors like quantity of item in a single transaction, profit of individual item are not considered [15,16]. If these factors are considered, than it may play fundamental role to perk up business usefulness.

4. ENHANCED APPROACH

Main drawback of frequent pattern mining frequent pattern mining algorithm is discussed in above section. To remove these problems, proposed approach is given in this paper. The attributes like frequency and profit of items are very essential in business stand point[17]. In the proposed approach both of these attributes are considered to find frequent pat-terns in less time . The approach given here is stated as F3 algorithm (Frequent pattern mining Fro m Frequency of items).These algorithm is having main three steps as following:

1. Calculate Profit ratio of items given profit.
2. Find frequent items from quantitative dataset using proposed Apriori algorithm
3. Find interesting patterns from previous frequent patterns using PW-factor

The first step calculates profit ration from given profit of items. Second step finds frequent patterns using proposed Apriori algorithm. This step gives us the frequent patterns and association rules. After finding association rules, the third step finds other interesting relationships from previous patterns . This pro-vides abbreviated patterns than previous patterns and is also useful in business perspective.

4.1 Important terms used in algorithm

Basic terms: T = {T1, T2, T3,…….Tm} is the set of transactions, where each transaction is a collection of item's quantity(frequency) in that transaction. i.e. the number of times the single item is purchased by customer in single transaction and I = {i1, i2, i3….in} is set of items.

Q-factor: Pro fit ratio of particular item is stated as Q-factor. When customer purchases item from m super market each item has its own unique transaction id and profit of item. Profit of item is awfully significant factor through which one can identify which item's selling is higher. In the approach given here,

Q-factor will be used further to calculate PW - factor. This profit of item will be profit margin, not profit mark up. Equation for calculating profit rat ion is given below:

$$\textbf{Q – Factor} = P / \Sigma Pi \qquad (1)$$

Where, P = profit of item

ΣPi = sum of all items profit

PW-factor: PW-factor: Profit and weight factor is stated as PW-factor. Using PW-factor other patterns can be found from frequent patterns which can be helpful to identify which item is frequent, as well as which item's profit is more. The new schemes can be made from these to catch the attention of customers. Initially minimum PW-factor threshold value will be given like it is given for support. The patterns satisfying minimum PW -factor will be selected as final frequent patterns.PW-factor can be calculated as follow

$$PW = \sum_{i=1}^{n} \text{frequency} * \text{Q – Factor} \qquad (2)$$

Where, Σ frequency from 1 to n = total frequency of item calculated, i.e. . it is support for k-item set at different level.

4.2 Enlightenment of algorithm

The first step: The very first step is to calculate the profit ratio i.e. Q-factor of individual items given profit of items using equation(1). After calculating the Q-factor, store it to the database. Table 3 shows the item's profit and calculated Q-factor.

Table 4.1 Q-factor of items

Items	Profit	Q-factor
I1	40%	0.153846
I2	20%	0.076923
I3	45%	0.173077
I4	80%	0.307692
I5	75%	0.28846

The second step: In the customary frequent pattern mining algorithms, the support of the item is counted by checking whether the item is present in the trans-action or not. If item is present than support count is incremented. In this case it takes more time because if for k item set, any element is not present than also scanning should be done to count support. For example, for 3-item set (shampoo, conditioner, hair oil) and for five transactions, scanning must be done in all five transactions for these three items, even though any of them is not present .i.e. if in transaction one, shampoo is not present, then also checking for conditioner and hair oil must be done, which is wastage of time. To overcome this negative aspect, the new F3 approach is given which counts support of items from item's quantity i.e. frequency in the single transaction. To use item's quantity is the center idea of the proposed algorithm.

Commonly, when customer purchases any item from super market, in the database the item's information is added in the unique transaction id for that customer with item name, quantity and price in that transaction id. So in real life, the database with transaction id and items are not given. The database is having quantity of item in the transaction id as customer can buy single number of times in one transaction. For example, one can acquire five packets of bread in one transaction. Using this originality, the proposed algorithm works.

Table 4.2 Database

Transaction id	Item	Quantity	Price
1	I1	2	68.50
1	I2	3	50
1	I4	1	60
1	I5	6	45.50
2	I2	3	50

2	I4	4	60
2	15	1	45.50
3	I1	1	68.50
3	I2	1	50
3	I3	1	15.05
3	I4	6	60
4	I1	3	68.50
4	I2	4	50
4	I4	7	60
5	I4	2	60

The Table 4.3 shows the database with transaction id, item, item's quantity in single transaction and price.

Additional information of item can also be present in practical life. This Table gives the perfect idea about how many items are purchased by which customer. According to the frequent pattern mining algorithms, to find frequent patterns , another database is to be created, which has transaction id and item names, so that one can have idea that this many items are purchased in one transaction. But this process is quite complex and takes time . And moreover, frequent items finding still remains. To conquer this problem, directly the dataset is created which has transaction id and individual item's frequency (quantity) in single transaction. Table 4.3 shows the dataset with these functionalities. In relation to this Table, each and every item's quantity is displayed in the transaction id.

The name of this dataset id given as Quantitative dataset as it shows quantity of items. For example, from Table 4.1 it can be clearly seen that item I1 is purchased in transaction 1 and items quantity is 2, so in Table 4.2 it is shown that quantity is 2 of item I1. Similarly for items I2 and I5 have quantity 3 and 6 respectively. The rest of the items have quantity zero as they are not purchased in that transaction. The term frequency is used in place of quantity in the proposed algorithm.

Table 4.3 shows Quantitative dataset which will be used to find the support of item set. Minimum support threshold value is 60% here. In the other algorithms, from m transactions and items support is calculated. So if in any transaction, some/all item elements of item set are not present, then also, those item elements in the transactions, which takes more time are also to be checked.

Table 4.3 Quantitative dataset

TID	**Items**				
	I1	**I2**	**I3**	**I4**	**I5**
1	2	3	0	1	6
2	0	3	0	4	1
3	1	1	1	6	0
4	3	4	0	7	0
5	0	0	0	2	0

But, as here the support is calculated from the Quantitative dataset, if in the item set, the individual item element's frequency in greater than zero or not that is to be checked. If it is greater than zero then item is present in the transaction. For example, for item set {I1, I2, I3}, it will be checked that item I1, I2 and I3 all have frequency greater than zero in each transaction. In transaction 1, I1 has frequency 2, so check for I2 which is 3 and for I3 is 0, so as for one item element's frequency is zero, the transaction must skip.

Similarly if it is checked in transaction 2 for item set {I1, I2, I3} than I1 has frequency zero, that means, no need to check further for other two item elements i.e. for I2 and I3, as if any of them is not present than support cannot be incremented for that item set and immediately skip that transaction. Accordingly in the other item sets, many transactions will be skipped and time for the execution will be diminished.

The process given above is for calculating support. But, the algorithm used for this is proposed Apriori algorithm. Classical Apriori makes use of an iterative approach known as breath-first search, where k-1 item set are used to search k item sets. There are two main steps in Apriori.

1)Join - The candidates are generated by joining among the frequent item sets level-wise.

2)Prune-Discard items set if support is less than minimum threshold value and discard the item set if its subset is not frequent[9].

Amongst which here change is made in pruning step to reduce time complexity.

4.3 Proposed method in Apriori

In join step candidate set Ck is found by joining Lk-1 elements with itself as denoted in classical algorithm. In prune step it can use efficient method for candidate set selection from Lk-1. According to the property if (k-1) subset of any item set Is $\subseteq$ Ck is not element of Lk-1 , then this (k-1) subset is not frequent and so Is is also not frequent. The algorithm needs to search level Lk-1 for k times for each element Is in Ck. So here an efficient method is proposed which only searches Lk-1 once to complete deletion of element Is in Ck.

The idea behind this is, If (k-1) item sub sets of Is belongs to Lk-1 are less than k, then (k-1) sub set is not frequent ,so Is is also not frequent item set of Ck. Total number of k -1 item sub sets belongs to Lk-1 is k. So the count is set which must be equal to k after scanning the previous level Lk-1.If any of these (k-1) item sub set is not part of Lk-1, then total number of (k-1) item sub sets are less than k. So that (k-1) item subset is not frequent, results in Is to be infrequent.

For example, if 3-item set are there,{I1, I2, I3} then the (k-1) subsets are {I1,I2},{I2,I3} and {I1,I3}. Ac-cording to the property all three (k-1) subsets must be present in L2 and their presence is checked for k times i.e. individually for all (k-1) subsets. Total number of (k-1) subsets are k=3. So in the proposed method, the checking of existence of (k-1) sub-sets only once in the previous level. The count must be equal to 3 after scanning L2. If count is equal to 3, then 3-item set {I1,I2,I3} should not be discarded from C3. At last, frequent patterns are getting done as a result. Association rules are also found. Minimum confidence threshold is 80%.

From Table 4.2 frequent patterns can be given as Follow:

Table 4.4 Frequent Patterns

Item	Support
I1	60 %
I2	80 %
I4	100 %
I1, I2	60 %
I1, I4	60 %
I2, I4	80 %
I1, I2, I4	60 %

Table 4.5 Strong Rules

Association	Confidence
I1→i2	100%
I1→I4	100%
I4→I2	80%
I1, I2→ I4	100%
I1, I4→ I2	100%
I1→ I2, I4	100%

Table 4.5 shows association between items i.e. strong rules from Table 4.4.

The third step: After getting frequent items, the third step is to check whether they satisfy minimum PW-factor or not. Items satisfying minimum Pw-factor are selected as frequent patterns. Value for minimum PW-factor is taken as 2. To find minimum PW - factor, frequent items are used. To find PW-factor equation (2) is used. After finding Pw -factor of each and every item, those items are selected who are satisfying minimum PW -factor. The calculated Pw-factor for frequent items is shown in Table 6. Only those patterns are selected whose PW -factor is >= 1.5 which is shown in Table 4.7.

Table 4.6 PW-factor

Frequent Items	PW-factor
I1	0.462
I2	0.308
I4	1.538
I1, I2	0.692
I1, I4	1.385
I2, I4	1.538
I1, I2, I4	1.615

Table 4.7 Selected Patterns

Frequent Items	PW-factor
I4	1.53`8
I2, I4	1.538
I1, I2, I4	1.615

From Table 4.7, conclusion can be given that the selected items are frequent, as well as their profit is also high. This kind of information's can be helpful for the business organizations, especially for super markets to increase their selling. After the comprehensive description of the algorithm, the following section gives the algorithm and is its final steps.

4.4 F3 Algorithm

Input: D = Database, minimum support threshold, minimum confidence threshold, minimum PW-factor threshold

Begin

1) Given profit of items, calculate profit ratio of the

items by using equation (1) .

2) Store profit ratio of each item.

3) Scan database.

4) Create a Quantitative dataset which has TID and

each item's frequency associated with it

5) Generate candidate set Ck by 'join' step of Apriori algorithm.

6) For each item set Is $\in$ Ck and c $\subset$ Is, where c is (k-1) subset of Is, check if previous level Lk-1 contains c or not.

7) If it contains c then increment count and go to step 6 else go to step 7.

8) If count is equal to k(i.e. number of elements in Is $\in$ Ck) keep items in candidate set and go to step 10 else go to step 9.

9) Delete item set Is from Ck candidate set.

10) If support count is greater than minimum support, then go to step 12 else go to step 11.

11) Consider item as infrequent item.

12) Consider item as frequent item if it satisfies minimum confidence.

13) Find profit and weighted factor from frequent items using equation (2).

14) If PW-factor is greater than minimum PW-factor, go to step 16 else go to step 15.

15) Reject infrequent patterns.

16) Select frequent patterns

End

5. PERFORMANCE EVALUATION

The algorithm is implemented on the massive data-bases Super market and Customer Care in C#.NET with SQL Server 2008 and MS Access as backend. The F3 algorithm trims down the time of execution and improves the time complexity. The testing of the F3 algorithm and some other traditional algorithms is done on the same database. The resultant chart in Fig 1,2,3 and 4 shows that the time taken by the F3 algorithm for different support values keeping confidence and PW-factor same, is less than the other algorithms

Table 5.1 Time comparison for Supermarket data-base using SQL server

	Time taken by SQL Server in (seconds)			
Support (in %)	**F3 Algorithm**	**Existing Approach**	**Apriori**	**Eclat**
40	8.01	12.01	18.5	11.2
50	7.50	11	11.9	10.15
60	6.76	10	12.01	9
70	4.5	9.8	12	8.57
80	3	8.9	11.08	7.7

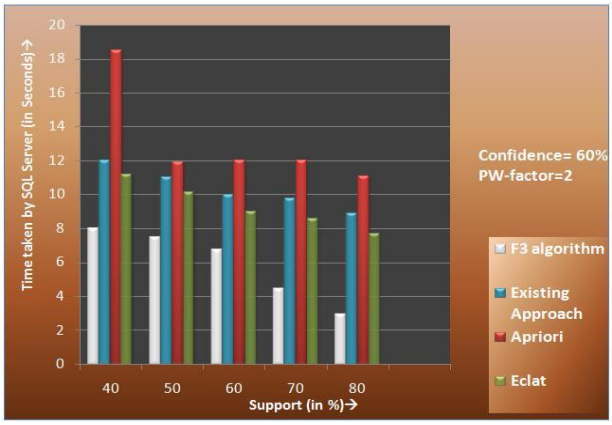

Fig 5.1 Comparison chart for Supermarket database using SQL server as a back end

Table 5.2 Time comparison for Supermarket database using MS Access

	Time taken by MS Access in (seconds)			
Support (in %)	**F3 Algorithm**	**Existing Approach**	**Apriori**	**Eclat**
40	9.01	14.01	19.01	12.01
50	7.70	12	15.9	11.20
60	7.02	11.30	14.01	9.8
70	4.59	10.10	13	9
80	3.3	9.001	12.08	7.90

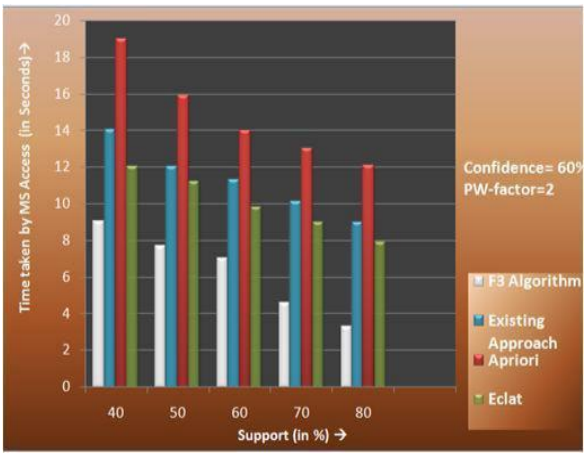

Fig 5.2 Comparison chart for Supermarket database using MS Access as a back end

Table 5.3 Time comparison for Mobile Customer Care database using SQL Server

Time taken by SQL Server in (seconds)				
Support (in %)	**F3 algorithm**	**Existing Approach**	**Apriori**	**Eclat**
40	12.01	18.28	23.32	15.2
50	11.4	16.32	18.22	14.3
60	11	13.35	16.72	15.25
70	9.3	11.21	15.43	14.29
80	5.4	11.02	10.21	9.21

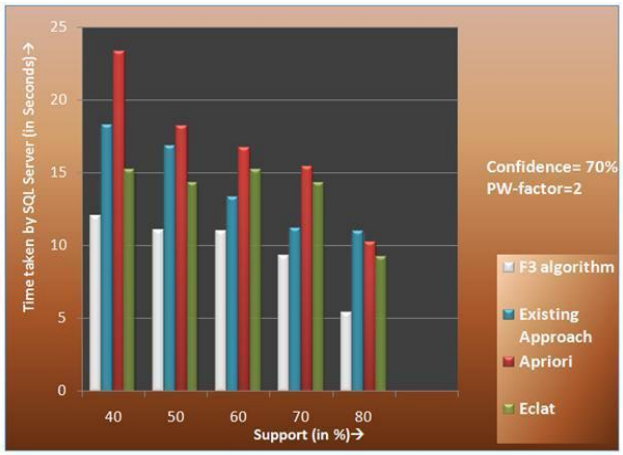

Fig 5.3 Comparison chart for Mobile Customer Care database using SQL Server as a back end

Table 5.4 Time comparison for Mobile Customer care database using MS Access

Time taken by MS Access in (seconds)				
Support(in %)	**F3 Algorithm**	**Existing Algorithm**	**Apriori**	**Eclat**
40	12.5	18.9	24.43	16.21
50	11.04	17.72	19.21	15.21
60	12.2	14.4	16.89	15.74
70	9.9	11.3	15.88	14.9
80	6.5	11.01	11.48	9.32

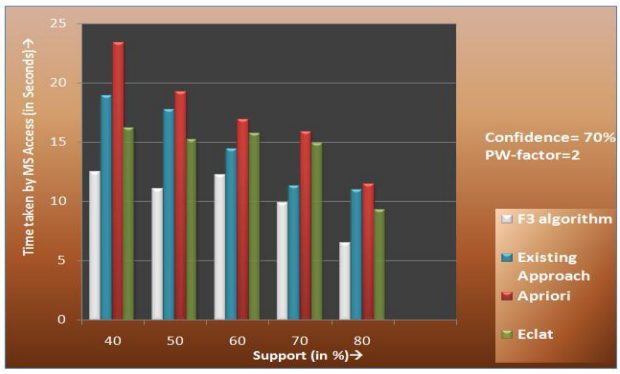

Fig 5.4 Comparison chart for Mobile Customer Care database using MS Access as a back end

6. CONCLUSION

To recapitulate, it can be analyzed that, the proposed F3 algorithm reduces time for the execution by finding frequent items from quantity of items in a transaction. The future work on this algorithm is to work on the memory utilization of it as here that part is not considered. Also, the algorithm can be combined with some other association techniques to optimize even more time.

7. ACKNOWLEDGMENTS

I sincerely want to thank my parents and my brother for the constant motivation during the research paper work. I would like to thank my husband too for helping me out in the implementation work.

8. REFERENCES

[1] Karl Aberer,(2007-2008), Data mining-A short introduction[Online],Available:http://lsirwww.epfl.ch/courses/dis /2003ws/lecturenotes/week13Dataminng print.pdf

[2]Agrawal, R. and Srikant, R. 1995." Mining sequential patterns", P. S. Yu and A. S. P. Chen, Eds. In:*IEEE Computer Society Press, Taipei, Taiwan, 3{14}.*

[3] R.Divya, S.Vinod kumar ,"Survey on AIS,Apriori and FP-Tree algorithms",In: *International Journal of Computer Science and Management Research Vol 1 Issue 2 September 2012, ISSN 2278-733X*

[4] Goswami D.N., Chaturvedi Anshu.,Raghuvanshi C.S.," An Algorithm for Frequent Pattern Mining Based On Apriori", In: Goswami D.N . et. al./ (IJCSE) International Journal on Computer Science and Engineering ,,Vol. 02, No. 04, 2010, 942 -947, ISSN : 0975-3397

[5] Sheila A. Abaya, "Association Rule Mining based on Apriori Algorithm in Minimizing Candidate Generation",In:*International Journal of Scientific & Engineering Research Volume 3, Issue 7, July-2012*

[6] Zhang Changsheng, Li Zhongyue, Zheng Dongsong," An Improved Algorithm for Apriori",In: *IEEE,First International Workshop on Education Technology and Computer Science,2009*

[7] Ms. Sanober Shaikh, Ms. Madhuri Rao,Dr. S. S. Mantha," A New Association Rule Mining Based On Frequent Item Set",In : *CS & IT-CSCP 2011*

[8]Mamta Dhanda," An Approach To Extract Efficient Frequent Patterns From Transactional Database",In: *International Journal of Engineering Science and Technology (IJEST), Vol.3 No.7 July 2011, ISSN:0975-5462*

[9]Andrew Kusiak, Association Rules -The Apriori algorithm[Online],Available:http://www.engineering.uiowa.edu/~comp/Public/ Apriori.pdf

[10]Mamta Dhanda, Sonali Guglani , Gaurav Gupta, "Mining Efficient Association Rules Through Apriori Algorithm Using Attributes ", In: *International Jour-nal of Computer Science and Technology Vol 2,Issue 3,September 2011,ISSN:0976-8491*

[11] Hilderman R. J., Hamilton H. J.,"Knowledge Discovery and Interest Measures ",In: *Kluwer Aca-demic Publishers, Boston, 2002*

[12]Ku mar, Association analysis: basic concepts and algorithms [Online], Available: http:// www-users.cs.umn.edu ~kumar/dmbook/ch6.pdf

[13]Tan, Steinbach, Ku mar, (2004, April18), Introduction to Data mining[Online],Available:http://wwwusers.cs.umn.edu/~kumar/dmbook/dmsli des/chap6_basic_association_analysis.pdf

[14]R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", In: *Proceedings of the 1993 Inter-national Conference on Management of Data (SIGMOD 93), pages 207-216, May 1993*

[15]Lu,S.,Hu , H., Li, F., "Mining Weighted Association Rules", In: Intelligent Data Analysis, vol.5 , no. 3, pp.211 - 225, August 2001

[16]Parvinder S. Sandhu, Dalvinder S. Dhaliwal.S.N. Panda,Atul Bisht,"An Improvement in Apriori Algorithm Using Profit and Quantity" , In: *2010 Second International Conference on Computer and Network Technology, April 23-April 25,ISBN: 978-0-7695-4042-9,Bangko k, Thailand*

[17]Jianying Hu, Aleksandra Mojsilovic, "High - utility pattern mining: A method for discovery of high-utility item sets",In: *Pattern Recognition vol. 40, no. 11, pp.3317-3324, November 2007*