A New Approach for Extracting Closed Frequent Patterns and their Association Rules using Compressed Data Structure

Vimal Kishor Tiwari
Department of Computer Science and Engineering
BUIT, BU, Bhopal, India

Anju Singh,Ph.D
Department of Information Technology
BUIT, BU, Bhopal, India

ABSTRACT
In data mining, term frequent pattern extraction is largely used for finding out association rules. Generally association rule mining approaches are used as bottom-up or top-down approach on compressed data structure. In the past, different works proposed different approaches to mine frequent patterns from giving databases. In this paper, we propose a new approach by applying the closed & intersection approach using compressed data structure. We have used closed as bottom-up and intersection as top-down approach. This combined approach allows diminishing the search time by reducing database scan for finding out closed frequent patterns and their association rules. The time complexity of the proposed algorithm is less while the classical approach like a priori has taken more time for given items in the dataset. Experimental results show that our approach is more efficient and effective than a traditional apriori algorithm.

Keywords
Closed Approach, Intersection approach, Apriori algorithm, Closed Frequent Pattern, Data Mining, Compressed Data Structure.

1. INTRODUCTION
As with the new innovation of the IT technologies, the amount of accumulated data is also increasing. It has resulted in a large amount of data stored in databases, warehouses and other repositories. Thus the Data mining comes into representation to explore and analyze the databases to extract the interesting and previously unknown patterns and rules known as association rule mining.
In data mining, Association rule mining is one of the important descriptive tasks which can be defined as discovering meaningful patterns from the huge collection of databases. Mining frequent itemset is very crucial part of association rule mining.

There are many approaches have been planned for the previous several decades including horizontal design based techniques, vertical design based techniques, and projected design based techniques, Other than a large amount of the approaches go through from repeated database scan, Candidate generation (Apriori Algorithms), memory consumption problem (FP-tree Algorithms) and many more for mining frequent patterns. As many industries transactional databases contain the same set of transactions many times, to apply this thought, this paper presents a new technique which requires less time and guarantees that, give better performance than a traditional apriori algorithm.

2. LITERATURE SURVEY
This section gives literature survey of different frequent pattern mining approaches. This section also gives a survey for Data mining and frequent pattern mining.
As data mining deals with the extraction of interesting knowledge, the conception of interestingness is very important for extracting knowledge patterns & their association rules. Several interestingness measures have been used to extract knowledge patterns from the data. For example, the interestingness measure, "minimum support" is used to my frequent itemsets and both "minimum support" and "minimum confidence" is used to mine association rules [2].
In the field of Association Rules Mining, some research work carried out by various researchers for generating association rules and related to the extracting frequent item sets in the large data sets is discussed.
Efficient Mining of Intertransaction Association Rules is proposed by the Anthony K.H. Tung et al [21]**,** in this study introduce the notion of intertransaction association rule, and describe its dimensions minimum support and minimum confidence.
Vicente et al [22] proposed An Associative Memory for Association Rule Mining, In this work focus on investigating if a type of mapping neural network, better known as an associative memory, is suitable for association rule mining.
Zhang Hui et al [23]**,** proposed a method Study on Association Rules Mining Based on Searching Frequent Free Item Sets using Partition, they divide the database into multiple partitions and then find frequent free item sets in each partition, then merge the several partitions to generate other frequent free item sets and count the support.
Weimin Ouyang et al [24]**,** proposed a Mining Direct and Indirect Association Patterns with Multiple Minimum Supports.
In the research work done by Sarjon Defit [25]**,** proposed a method Intelligent Mining Association Rules called IMAR. IMAR is described by three most significant phases, i.e., pre-process, process and post-process. The based on various research they proposed the various methods and algorithms for mining association rules, based on this study we improve the performance of association rules mining by using closed approach, Intersection Approach and generating of the algorithm.
In this section, first explain the model of association rules. Next, discuss the related work pertaining to the research efforts made in the literature to confront the item problem. Here also summarize the issues when selecting an appropriate interestingness measure to extract association rules.

2.1 Data mining

Data mining, also referred to as knowledge discovery from data (KDD), is the automated or convenient extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the Web, other massive information repositories, or data streams [26].

Data mining should be regarded as a strategic and competitive move. So before the Data mining process starts, the goal which is in focus of the analysis should be clarified. Otherwise it's not possible to search for new valuable information if the necessary parameters cannot be defined as there are different models for the data mining process based on the task at hand. The following description is based on the model of Fayyad [27].

Step 1: Data selection

Out of a data base the needed data were selected according to its objects and characteristics.

Step 2: Pre-Processing

In this step a cleaning of the selected data is done. This means for example the filling of missing values.

Step 3: Transformation

In the transformation phase the data are transformed in new formats, if necessary.

Step 4: Data Mining

In this step of the process identifies the patterns and relationships between the data.

Step 5: Interpretation and Evaluation

In the last step the result has to be interpreted and evaluated to come up with suitable actions.

2.2 Association Rules

Association rules are an important class of regularities that exists in a database. Since the introduction of association rules in [6], the problem of mining association rules from transaction databases has been actively studied in the data mining community [12, 13, 14]. The common application is market basket analysis, where association rule mining analyses how the items purchased by consumers are associated. An example of an association rule is as follows, soap $\Rightarrow$ shampoo [support = 20%; confidence = 75%]:

The above rule says that 20% of customers buy soap and shampoo together, and those who buy soap also buy shampoo 75% of the time. The basic model of association rules is as follows [1]: Let I = $\{i_1; i_2; \ldots; i_n\}$ be a set of items and T be a set of transactions (dataset). Each transaction t is a set of items such that t $\subseteq$ I. An itemset (or a pattern) X is a set of items such that X $\subseteq$ I. A pattern containing k number of items is called a k-pattern. An implication of the form A $\Rightarrow$ B, where A $\subset$ I, B $\subset$ I and A $\cap$ B = / 0 is called an association rule iff,

(i) The support of A $\Rightarrow$ B, denoted as S(A $\cup$ B) = P(A $\cup$ B) = f (A $\cup$ B)|T|, is not less than the user specified minimum support threshold, minsup.

(ii) The confidence of A $\Rightarrow$ B, denoted as C (A $\cup$ B) = P (B|A) =S (A $\cup$ B) S (A), is not less than the user specified minimum confidence threshold, minconf.

Where, f (A $\cup$ B) represents the frequency of the pattern, A $\cup$ B in T and |T| represents the total number of transactions in T. The itemsets which satisfy the minimum support are called frequent itemsets.

2.3 Apriori Algorithm

The first algorithm for mining all frequent itemsets and strong association rules by algorithm given in [1]. Shortly after that, the algorithm was improved and renamed apriori. Apriori algorithm is, the most classical and important algorithm for mining frequent itemsets. Apriori is used to find all frequent itemsets in a given database DB.

The key idea of Apriori algorithm is to make multiple passes over the database. It employs an iterative approach known as a breadth-first search (level-wise search) through the search space, where k-itemsets are used to explore (k+1)-itemsets.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" [6].

Apriori Algorithm: (by Agrawal et al at IBM Almaden Research Centre)

Pass 1

1. Generate the candidate itemsets in C_1
2. Save the frequent itemsets in L_1

Pass *k*

1. Generate the candidate itemsets in C_k from the frequent itemsets in L_{k-1}
   1. Join L_{k-1} *p* with L_{k-1}q, as follows:
      insert into C_k
      select *p*.item$_1$, *p*.item$_2$, . . . , *p*.item$_{k-1}$, *q*.item$_{k-1}$
      from L_{k-1} *p*, L_{k-1}q
      where *p*.item$_1$ = *q*.item$_1$, . . . *p*.item$_{k-2}$ = *q*.item$_{k-2}$, *p*.item$_{k-1}$ < *q*.item$_{k-1}$
   2. Generate all (*k*-1)-subsets from the candidate itemsets in C_k
   3. Prune all candidate itemsets from C_k where some (*k*-1)-subset of the candidate itemset is not in the frequent itemset L_{k-1}
2. Scan the transaction database to determine the support for each candidate itemset in C_k
3. Save the frequent itemsets in L_k ($\cup_k L_k$).

Now present a simple example of how Apriori works.

Let the Database, D = { T_1 = (1,4,5), T_2 = (1,2), T_3 = (3,4,5), T_4 = (1,2,4,5)}. Let the minimum support value minsup = 2. This Dataset is running by the above algorithm,

C_1 = {{1},{2},{3},{4},{5}}

L_1 = {{1},{2},{4},{5}}

C_2 = {{1,2},{1,4},{1,5},{2,4},{2,5},{4,5}}

L_2 = {{1,2},{1,4},{1,5},{4,5}}

C_3 = {{1, 4, 5}}

L_3 = {{1, 4, 5}}

Note that while forming C_3 by joining L_2 with itself, we get three potential candidates, {1,2,4}, {1,2,5} and {1,4,5}. However only {1,4,5} is a true candidate, and the first two are eliminated in the pruning step, since they do not satisfy the condition of minimum support.

Generally, frequent-pattern mining results in a huge number of patterns of which most can be found to be insignificant according to application and/or user requirements. As a result, there have been efforts in the literature to mine constraint-based and/or user-interest based frequent patterns [7, 8, 9, 10]. In recent times, temporal periodicity of frequent patterns has been used as an interestingness criterion to discover a class of user-interest based frequent patterns, called periodic-frequent patterns [11]. A pattern is said to be periodic-frequent if it satisfies both the minimum support (minsup) and the minimum confidence (minconf) constraints. Minsup constraint controls the minimum number of transactions that a pattern must cover in a database. Minconf constraint controls the minimum number of items that a pattern must cover in all the transactions.

Since a single minsup and a single minconf constraint are used for all items in the database, this model implicitly assumes that all items have similar frequencies and occurrence behavior. However, this is not the case in real-world datasets. Real-world datasets are non-uniform in nature containing both frequent items.

The first algorithm for mining all frequent itemsets and strong association rules was the AIS algorithm by [1]. Shortly after that, the algorithm was improved and renamed Apriori. Apriori algorithm is, the most classical and important algorithm for mining frequent patterns. Traditional Apriori is used to extract all frequent patterns in a given database (DB).

The key idea of Apriori algorithm is to make multiple passes over the database. It employs an iterative technique known as a (BFS) breadth-first search through the search space, where k-itemsets are used to explore (k+1) -itemsets. BFS is a level-wise search in a hierarchical order i.e. from root node to a leaf node.

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" [6]. It also described the non monotonic property which states that if the system cannot pass the minimum support test, all its supersets will fail to pass the test [2, 6]. Therefore if the one set is infrequent then all its supersets are also frequent and vice versa. This property is used to prune the non-frequent candidate elements.

It is absorbed that reducing the candidate items from the database is one of the important tasks for increasing the efficiency. Thus a DHP technique was proposed [15] to reduce the number of candidates in the early phases C_k for k>1and thus the size of database.

Several transactions in a database may contain the same set of items. Even if two transactions are originally different, early pruning of infrequent items from them can make the remaining set of items identical. We can reduce the transaction volume by replacing each set of identical transactions by a single transaction and a count of its occurrences. This could be done using a modified prefix tree or by sorting transactions. We found that using the prefix tree was more efficient compared to sorting [16].

The compression scheme described in the Compact tree reduces the number of nodes in the transaction tree and allows further grouping of transactions that share some common items. A complete prefix tree will have many identical sub trees in it [16].

FP-tree algorithm [17, 18, 19] is based upon the recursively divide and conquer strategy; first the set of frequent 1-itemset and their counts is discovered. To start from each frequent itemsets, construct the conditional pattern base, then its conditional FP-tree is constructed. This tree is a prefix tree.

A compact tree structure, called a CT - tree, to compress the original transaction data. This allows the CT-Apriori algorithm, which is revised from the classical Apriori algorithm, to generate frequent patterns quickly by skipping the initial database scan and reducing a great amount of I/O time per database scan [5].

Another compact data structure named Compressed FP-Tree (CFP-Tree) [20]. The number of nodes in a CFP-Tree can be up to half less than in the corresponding FP-Tree. CFP-Tree is a bottom-up approach that generates the frequent itemsets following the pattern growth approach non-recursively.

3. PROPOSED WORK

The main objective of this work is to develop and propose a new scheme for mining the closed frequent itemsets and their association rules out of transaction data set. The proposed scheme is based on two approaches: closed approach and intersection approach. The proposed scheme is more efficient than Apriori algorithm, as it is based on two of the most efficient approaches.

3.1 Closed Approach (bottom-up)

An itemset I am said to be closed if and only if close (I) = x (y (I)) = x o y (I) = I, Where the composite function close = x or y is called Galois operator or closure operator. The closure operator defines a set of equivalence classes over the lattice of frequent itemsets: two itemsets belong to the same equivalence class if and only if they have the same closure, i.e. they are supported by the equivalent set of transactions. This can also show that an itemset I is closed if no superset of I with the same support exists. Therefore mining the maximal elements of all the equivalence class corresponds to mining all the closed itemsets [4].

In the bottom-up process, closed frequent patterns are extracted. We explain closed approach later by using an example.

3.2 Intersection Approach (top-down)

Here the top down approach that we have used is intersection approach. The main reason we introduce this approach is due to the problem of candidate generation. General algorithms like the pincer search generate the candidate by decomposition technique. This may result in a lot of candidates. Decomposition technique that is used in pincer search is very expensive to scan the database for every one of these candidates' itemsets [3].

In the intersection approach we will only have an item set that is produced by the intersection of two largest itemsets see in Figure 1. Using the intersection approach as a top down approach we can greatly reduce the database scanning and the number of candidates.

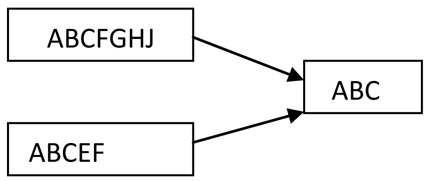

Fig 1: Intersection Approach

When, start the top-down process in the first time. It must check that itemset is sorted in lexicographic order means that sort the itemsets by its transaction length. In this process check the longest itemset in the sorted table first. When check the support of an itemset, it only needs to check support of the itemset whose length is longer than the current one. If the support of an itemset satisfies the condition of the minimum support then keep this process for the bottom-up process. If not, intersect any two transactions that are infrequent to generate the candidate set. After using this approach, get an itemset, if itemset is frequent than this itemset is used to discard the candidates that I met during bottom-up process. If not, then check frequency of all the itemsets with the longest length itemset [3].

3.3 Proposed Algorithm

One disadvantage of the apriori algorithm has this is more time consuming & taken large space. The proposed algorithm is time efficient. In the apriori algorithm very small sample size may generate many false rules, and thus degrade the performance. To remove this drawback and for practical purposes we can use our new approach for finding out closed frequent itemsets and their association rules of the data mining.

Steps of Proposed work:

Input:

1. A source database
2. Minimum Support
3. Minimum Confidence

Output: Set of closed frequent items and their associated rule

Step 1: scan the input database to get 1- frequent item sets & sort them

Step 2: Find 2- frequent closed item sets.

Step 3: Take two largest itemset and find out single item set from these itemsets using the intersection.

Step 4: remove all the subsets of sets obtained in step 3

Step 5: Repeat the steps 2 to 4 until the all closed items are not found out.

Step 6: Merge all closed frequent itemset (1-itemset to n-itemset) in order to get the final set of closed frequent itemset.

Step 7: find out association rules for all closed frequent itemsets.

3.4 Example

In the proposed algorithm we use two approaches closed as bottom-up and intersection as top-down approach. The following data set in table I has been used to show the algorithm. In this table 1 shows the original transaction database.

TABLE I
ORIGINAL TRANSACTON DATABASE

TID	List of Items
1	ABC
2	ABD
3	ABCFG
4	AC
5	AC
6	BC
7	BC
8	ABCE
9	BCD

In the table II shows the compressed data structure with head & the body part of the database. In this example we are taking minimum support is 2 (minsup = 2). The head of the compressed data structure is a list of 2-tuples (Items, Count), where Items is the name of an item and Count is the frequency count of item in the Original Transactional Database; and all items in the head are ordered in frequency descending order. The body of the compressed data structure is a set of 2-tuples (TC, LOI), where LOI is a unique transaction, TC is the occurrence count of LOI in Original Transactional Database, and the items in each transaction of the body are ordered in frequency descending order.

TABLE II
COMPRESSED DATA STRUCTURE

HEAD							
Items	C	B	A	D	E	F	G
Count	8	7	6	2	1	1	1

BODY	
List of Items	Count
ABCFG	1
ABCE	1
ABC	1
ABD	1
BCD	1
AC	2
BC	2

Table III shows the closed 1-itemset and check if itemset are frequent (satisfies the minimum support) then keep this itemset otherwise prune the itemset. In this process itemset E, F & G are pruned. Remaining itemsets are kept in Table IV.

TABLE III
CLOSED 1-ITEMSETS

Item	Closed	Support
A	A	6
B	B	7
C	C	8
D	BD	2
E	ABCE	1
F	ABCFG	1
G	ABCFG	1

TABLE IV
CLOSED 1-ITEMSETS AFTER PRUNING

Item	Closed	Support
A	A	6
B	B	7
C	C	8
D	BD	2

Table V shows that closed 2-itemset. In this table itemset {A, D}, {C, D} are pruned because of infrequent. Remaining itemsets are kept in table VI.

TABLE V
CLOSED 2-ITEMSETS

Item	Closed	Support
AB	AB	4
AC	AC	4
AD	ABD	1
BC	BC	6
BD	BD	2
CD	BCD	1

TABLE VI
CLOSED 2-ITEMSETS AFTER PRUNING

Item	Closed	Support
AB	AB	4
AC	AC	4
BC	BC	6
BD	BD	2

Table VII from Table VI shows that closed 3-itemset. Itemset {B, C, D} is pruned. Table VIII shows the final closed 3- itemset.

TABLE VII
CLOSED 3-ITEMSETS

Item	Closed	Support
ABC	ABC	3
BCD	BCD	1

TABLE VIII
CLOSED 3-ITEMSETS AFTER PRUNING

Item	Closed	Support
ABC	ABC	3

Choose two largest Itemset from the body of the database and apply the intersection approach on these itemset and find out new itemset. Check if this itemset are frequent then we use this itemset to remove scanning the subsets find out by bottom up approach because their subsets are also frequent. On the above tables (Table V to Table VIII) dark rows indicate the frequent itemset find out by using an intersection approach see in Figure 1.
Check the support of {A, B, C} if it is greater than the minimum support than it is frequent & by the property if any set is frequent than its subset is also frequent. By applying Intersection Approach on Closed -2 Itemset, we are finding closed frequent itemset.

TABLE IX
INTERSECTION APPROACH APPLY ON CLOSED
2- ITEMSET

Closed	Support
AB	4
AC	4
BC	6

Check support for each item set on the above table by support of {A, B, C} and we find out list of closed frequent item.(For closed frequent item support must be greater than its immediate superset then the item set is closed). Finally we merge all closed itemset in Table X.

TABLE X
ALL CLOSED ITEMSETS

Closed	Support
A	6
B	7
C	8
BD	2
ABC	3
AB	4
AC	4
BC	6

This two way approach works faster than the apriori algorithm. At the last we find an association rule for all close frequent itemsets by using the following way. First of all set the support and confidence:
Minimum support = 2 and Minimum Confidence = 50 %.
Association rule find out for frequent item set is shows in

Table XI

TABLE XI
ASSOCIATION RULES

D ⇒ B	A ⇒ BC	AB ⇒ C	AC ⇒ B
BC ⇒ A	A ⇒ B	B ⇒ A	A ⇒ C
C ⇒ A	B ⇒ C	C ⇒ B	

4. EXPERIMENTAL RESULTS

The Experiments performed on computer with Core 2 Duo 2.20 GHZ Processor, 2.0 GB RAM and hard disk 320 GB. Both the algorithms have developed in JAVA language. For unit of measurement this work considers time in seconds.
The experimental result of proposed work is shown in figure 2 reveals that the proposed scheme outperforms the Apriori approach for different minimum support and minimum confidence value. Our algorithm takes minimum time during pattern generation also takes more than 1400 secs to generate frequent pattern for dataset size 35568 transitions so it's difficult to show graph in this range, hence here this work show maximum limit 90 sec for graphical representation for comparison time taken to generate frequent pattern to understand comparison easily. Actual time taken for dataset is mention in Table XI. Here dataset size is measured as total number of transaction in respective dataset.

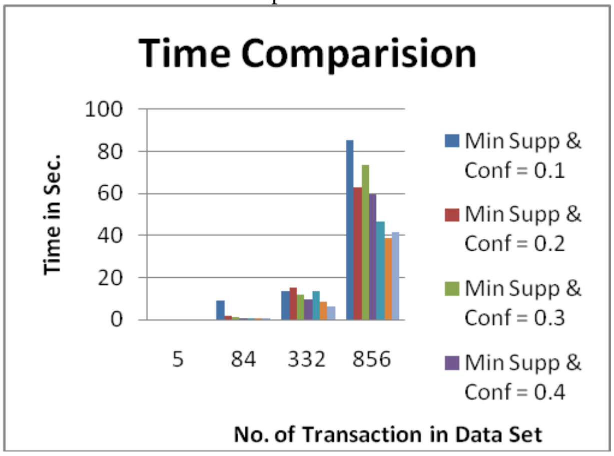

Fig 2: Time comparison for different Minsup and Minconf value

TABLE XII

TIME COMPARISION

Trans. In Dataset	MS 0.1, MC 0.1 Sec	MS 0.2, MC 0.2 Sec	MS 0.3, MC 0.3 Sec	MS 0.4, MC 0.4 Sec	MS 0.5, MC 0.5 Sec	MS 0.6, MC 0.6 Sec	MS 0.7, MC 0.7 Sec
5T.I5.D1K	0.32	0.28	0.21	0.19	0.16	0.22	0.09
84T.I13.D6K	9.38	2.03	1.4	0.94	0.82	0.63	0.6
332T.I13.D19K	13.52	15.62	11.89	9.67	13.81	8.76	6.33
856T.I55.D30K	85.63	62.86	73.51	59.56	46.56	38.63	41.52
35568T.I44.D1123	1453.65	1259.23	1153.41	1213.89	986.53	899.01	865.26

Figure 3 shows the analysis between Closed Intersection Approach with apriori algorithm. Here dataset size is measured according to number of transactions in respective dataset for generating frequent patterns.

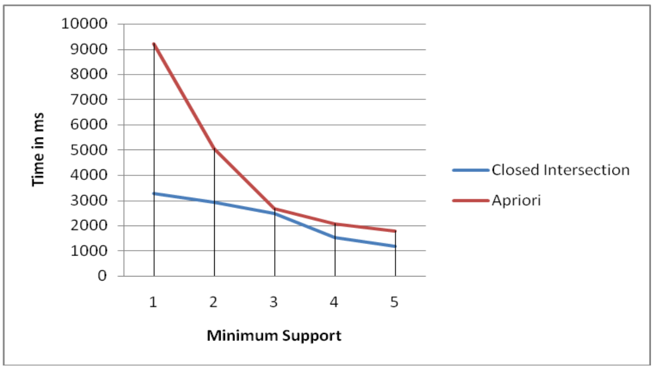

Fig 3: Time comparison for different Minsup

Analysis of above figure shows that proposed algorithm is more efficient than previous algorithm.

5. CONCLUSION & FUTURE SCOPE

The key idea behind finding closed frequent pattern is to get common string between current combinations of itemset from transactional dataset. Traditional algorithm such as apriori algorithm is used for finding frequent itemsets present in dataset. New proposed algorithm is used for finding closed frequent itemsets and their corresponding association rule from transition dataset. This proposed algorithm reduces time complexity a lot. This work simulation result also supports this statement. This proposed algorithm is efficient to reduced time almost 50% in compare to tradition algorithm for calculating closed frequent pattern. Overall time complexity can be observed by given time comparison in result and discussion chapter. Therefore this work employed it in our scheme to guarantee the time saving considered as an efficient method as proved from the results.

This work can summarize the main contribution of this research as follows:

- To study and analyze various existing approaches to mine frequent itemsets.
- To devised a new better scheme than Apriori algorithm using closed approach and intersection approach using compressed data structure as combined approach for mining closed frequent itemsets and their association rules.

There are a number of future research directions based on the work presented in this paper.

- When Database is large enough that is not fit in to the main memory then this algorithm creates a problem .To make this algorithm space efficient is an interesting field for future work.
- Other top-down approach such as partition algorithm is used with proposed approach is an good aspect for future work.
- New data structure is used with proposed approach is another future aspect.
- This scheme was applied in wine and retailer industry application, trying other industry is an interesting field for future work.

6. REFERENCES

[1] Xiaobing Liu, Kun Zhai, Witold Pedrycz, “An improved association rules mining method”, Pages 1362-1374, ACM 2012.

[2] Xin Zhang, Pin Liao, Huiyong Wang, “A New Association Rules Mining Algorithm”, Page(s): 429 – 432, IEEE 2009.

[3] Xain-hui Chang & Don-Lin Yang “Efficient mining of maximal frequent itemsets with the closed & intersection approach”, Department of Information Engineering & Computer Science,Feng Chia University Taiwan.

[4] Panida Songram and Veera Boonjing “Closed multidimensional sequential pattern mining”, International Journal Knowledge Management Studies, Vol. 2, No. 4, 2008. pp.460 – 479

[5] Qian Wan and Aijun “Compact Transaction Database for Efficient Frequent Pattern Mining” An Department of Computer Science and Engineering York University, Toronto, Ontario, M3J 1P3, Canada.

[6] Agrawal R., Imielinski T., and Swami A. N., "Mining association rules between sets of items in large databases", In SIGMOD Conference, pages 207–216, 1993.

[7] Hu T., Sung S. Y., Xiong H., and Fu Q., "Discovery of maximum length frequent itemsets". Inf. Sci., 178:69–87, January 2008.

[8] Quang T. M. , Oyanagi S. , and Yamazaki K., "Mining the k-most interesting frequent patterns sequentially". In E. Corchado, H. Yin, V. J. Botti, and C. Fyfe, editors, IDEAL, volume 4224 of Lecture Notes in Computer Science, pages 620–628. Springer, 2006.

[9] Kumar, R. ; Dixit, M., Analysis on probabilistic and binary datasets through frequent itemset mining, Page(s): 263 - 267 Conference Publications, IEEE 2012.

[10] Schmidt Jana and Kramer Stefan , “The Augmented Itemset Tree: A Data Structure for Online Maximum Frequent Pattern Mining”, pp 277-291 Springer 2011.

[11] Tanbeer S. K., Ahmed C. F., Jeong B.-S., and Lee Y.-K., "Discovering periodic-frequent patterns in transactional databases". In PAKDD ’09: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, pages 242–253, Berlin, Heidelberg, Springer-Verlag 2009.

[12] Ceglar A. and Roddick J.F., "Association mining. ACM Computing", Survey, 38, July 2006.

[13] Han J., Cheng H., Xin D., and Yan X., "Frequent pattern mining: Current status and future directions", Data Mining and Knowledge Discovery, 14(1), 2007.

[14] Stankovic, S.V., Rakocevic, G., “A classification and comparison of Data Mining algorithms for Wireless Sensor Networks”, Page: 265 – 270, IEEE 2011.

[15] Clifton C., "Encyclopedia britannica: Definition of data mining", 2010.

[16] Yudho Giri Sucahyo Raj P. Gopalan CT-ITL: Efficient Frequent Item Set Mining Using a Compressed Prefix Tree with Pattern Growth.

[17] Haiying Ma,Dong Gang, “Generalized association rules and decision tree”, Page(s) 4600 – 4603, IEEE, 2011.

[18] Kim Hyea Kyeong, Kim Jae Kyeong , “A product network analysis for extending the market basket analysis”, Pages 7403–7410, Elsevier 2012.

[19] Brijs T., Swinnen G., Vanhoof K., and Wets G., "Using association rules for product assortment decisions: A case study", In KDD, pages 254–260, 1999.

[20] Yudho Giri, Sucahyo Raj, P. Gopalan, CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure.

[21] Anthony K.H. Tung, Hongjun Lu, Jiawei Han and and Ling Feng, “Efficient Mining of Intertransaction Association Rules”, IEEE Transaction on knowledge and data engineering, VOL.15, NO.1, pages 46-56, 2003.

[22] Vicente Baez-Monroy and Simon O’Keefe, “An Associative Memory for Association Rule Mining”, IEEE Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, pages 2227-2232, 2007.

[23] Zhang Hui, LuYu and Zhou jinshu, “Study on Association Rules Mining Based on Searching Frequent Free Item Sets using Partition”, Asia-Pacific Conference on Information Processing, pages 343-346,2009.

[24] Weimin Ouyan gand Qinhua Huang, “Mining Direct and Indirect Association Patterns with Multiple Minimum Supports”, Computational Intelligence and Software Engineering (CISE), International Conference on IEEE, pages 1-4, 2010.

[25] Sarjon Defit, “Intelligent Mining Association Rules”, International Journal of Computer Science & Information Technology (IJCSIT), Vol. 4, No 4, pages 97-106, 2012.

[26] Wim Le Page “Mining Patterns in Relational Databases”, Universiteit Antwerpen, pages 1-21, 2009.

[27] Fayyad Usama, “Advances in Knowledge Discovery and Data Mining”. Cambridge:MIT Press, 1996.