IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 13, Issue 1 (Jul. - Aug. 2013), PP 73-82
www.iosrjournals.org

An Optimal Approach to derive Disjunctive Positive and Negative Rules from Association Rule Mining using Genetic Algorithm

Kannika Nirai Vaani.M [1], E Ramaraj [2]
[1](Training Division, TechMahindra Ltd, India)
[2](Dept of Computer Science and Engineering, Alagappa University, India)

***Abstract**: Mining frequent itemsets and association rules is a popular and well researched approach for discovering interesting relationships between variables in large databases. Association rule mining is one of the most important techniques of data mining that aims to induce associations among sets of items in transaction databases or other data repositories. There are various Algorithms developed and customized to derive the effective rules to improve the business. Amongst all, Apriori algorithms and FP Growth Algorithms play a vital role in finding out frequent item set and subsequently deriving rule sets based on business constraints. However there are few shortfalls in these conventional Algorithms. They are i) candidate items generation consumes lot of time in the case of large datasets ii) It supports majorly the conjunctive nature of association rules iii) The single minimum support factor not suffice to generate the effective rules iv) 'support/confident' alone not helping to validate the rules generated and v) Negative rules are not addressed effectively. Points from i) to iv) were addressed in the earlier works [10][13] . However identifying and deriving negative rules are still a challenge. The proposed work is considered to be the extended version of our earlier work [13]. It focuses how effectively negative rules can be derived with the help of logical rules sets which was not addressed in our earlier work. For this exercise the earlier work has been taken as the reference and the appropriate modifications and additions are updated into it where ever applicable. Hence by using this approach conjunctive & disjunctive; positive& negative rules can be generated effectively in an optimized manner.*
***Keywords** - Logical rule set, FP Growth Algorithm, Genetic Algorithm, Lift ratio, Multiple Minimum Support, Disjunctive Rules*

I. INTRODUCTION

Data mining is the task to mining the useful meaningful information from data warehouse. It is the source of inexplicit, purely valid, and potentially useful and important knowledge from large volumes of natural data [8].The selected knowledge must be not only precise but also readable, comprehensible and ease of understanding.Association rule basically use for finding out the useful patterns, relation between items found in the database of transactions [9].Association rule mining generally experimented to find all rules that satisfy user-specified minimum support and minimum confidence constraints [3].The important factor that makes association rule mining practical and useful is the minimum support. It is used to limit the number of rules generated. However, using only a single minsup implicitly assumes that all items in the data are of the same nature and/or have similar frequencies in the database. This is often not the case in real-life applications. In many applications, some items appear very frequently in the data, while others rarely appear. If the frequencies of items vary a great deal, we will encounter two problems,

1. If minsup is set too high, we will not find those rules that involve infrequent items or rare items in the data.
2. In order to find rules that involve both frequent and rare items, then minsup to be kept very low. However, this may produce too many rules.

So when one common support is fixed as minimum support for all the items, the rules which are not frequent occur but majorly contributing towards profit may get lost without notice.

For example, in a supermarket transaction data, in order to find rules involving those infrequently purchased items such as food processor and cooking pan (they generate more profits per item) very minimum support needs to be set ; but due to this the unwanted and rare items will not be get pruned. Hence fixing multiple minimum support for each items have become significant.

Multiple Minimum Supports to handle rare items:

In many data mining applications [4], some items appear very frequently in the data, while others rarely appear. If minsup is set too high, those rules that involve rare items will not be found. To find rules that involve both frequent and rare items, minsup has to be set very low. This may cause combinatorial explosion because those frequent items will be associated with one another in all possible ways. The disadvantage of support is the rare item problem. Items that occur very infrequently in the data set are pruned although they

would still produce interesting and potentially valuable rules. The rare item problem is important for transaction data which usually have a very uneven distribution of support for the individual items.

Algorithms dealing Association Rule Mining:

Apriori algorithm: In the earlier work [10] the Apriori algorithm was modified to include multiple supports, disjunctive and conjunctive rules and Genetic Algorithm was used to generate useful rules effectively. Major shortfall in modified Apriori was 'time taken' to generate frequent itemsets. In general Apriori-like algorithm may still suffer from the following two nontrivial costs, It is costly to handle a huge number of candidate sets [12]. It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching, which is especially true for mining long patterns. If one can avoid generating a huge set of candidates, the mining performance can be substantially improved. Hence the need of introducing an algorithm which takes considerably less time was realized. There comes the modified FP growth Algorithm. **FP-growth algorithm :**FP Growth approach is based on divide and conquers strategy for producing the frequent item sets [11]. FP-growth is mainly used for mining frequent item sets without candidate generation.

Lift Ratio

A high value of confidence suggests a strong association rule. However this can be deceptive because if the antecedent and/or the consequent have a high support, we can have a high value for confidence even when they are independent. A better measure to judge the strength of an association rule is to compare the confidence of the rule with the benchmark value where we assume that the occurrence of the consequent item set in a transaction is independent of the occurrence of the antecedent for each rule. This benchmark can be found out from the frequency counts of the frequent item sets. This enables to compute the lift ratio of a rule. The lift ratio is the confidence of the rule divided by the confidence assuming independence of consequent from antecedent. A lift ratio greater than 1.0 suggests that there is some usefulness to the rule. The larger the lift ratio, the greater is the strength of the association. With the lift value, the importance of a rule can be validated in an effective manner.

Formula List:

Confidence and Life factors are calculated as below,

$$Confidence = \frac{No.of\ transactions\ contain\ all\ the\ items\ in\ A\ \&C}{No.of\ transctions\ contain\ the\ items\ in\ A} = (support\ of\ AUC)/(support\ of\ A)$$

$$Expected\ Confidence = \frac{No.\ of\ transactions\ having\ the\ consequent\ items}{Total\ no.of\ transactions}$$

$$Lift = \frac{Confidence}{Expected\ Confidence}$$

The above factor can also be included while validating any rule set.

Disjunctive rules:

Association rule mining deals conjunctive rules majorly. But using disjunctive rules, extensive rule sets which are very much useful in mining the dataset can be found out effectively. At times disjunctive rule sets are also preferred to infer interesting rules. For example ideal rule set for the below query "If classes B or C or D are committed, what is the chance that A is also committed?" would be B OR C OR D →A.

Let X = {i1, i2...in} be a set of items. XC = {i1 AND i2 AND … AND in} or simply {i1i2...in} is a conjunctive term of X, and XD = {i1 OR i2 OR … OR in} is a Disjunctive term of X. The possible rule is of one of the following types, involving conjunctive and disjunctive terms from the set of items X U Y| X **Ω** Y = Φ[2][5].

The below table contains the set of possible positive rule sets which contain Conjunctive and Disjunctive nature of rules.

Table I :Ruleset of Conjuction and Disjuntion for Positive Rules

Type	Support	Confidence
$X_C \Rightarrow Y_C$	$s = P(X_C \cup Y_C)$	$c = P(Y_C \mid X_C)$
$X_C \Rightarrow Y_D$	$s = P(X_C \cup Y_D)$	$c = P(Y_C \mid X_D)$
$X_D \Rightarrow Y_C$	$s = P(X_D \cup Y_C)$	$c = P(Y_D \mid X_C)$
$X_D \Rightarrow Y_D$	$s = P(X_D \cup Y_D)$	$c = P(Y_D \mid X_D)$

Rule sets List1:

1. XY Z ⟹VW - type $X_C \Rightarrow Y_C$
2. XY Z ⟹V OR W - type $X_C \Rightarrow Y_D$
3. X OR Y OR Z ⟹ VW - type $X_D \Rightarrow Y_C$
4. X OR Y OR Z ⟹ V OR W - type $X_D \Rightarrow Y_D$

The number of rules that are found by the Associations mining function can be reduced by using rule filters if the number of frequent item sets is high. Rule filters are a powerful way to limit the amount of rules to be generated or the content of the rules. This parameter determines the maximum number of items that occur in an association rule.In the example the frequent itemset (E,A,C) has been used for analysis and in this case the maximum rule length is 3 and maximum antecedent allowed is 2; the maximum consequents allowed are 2.

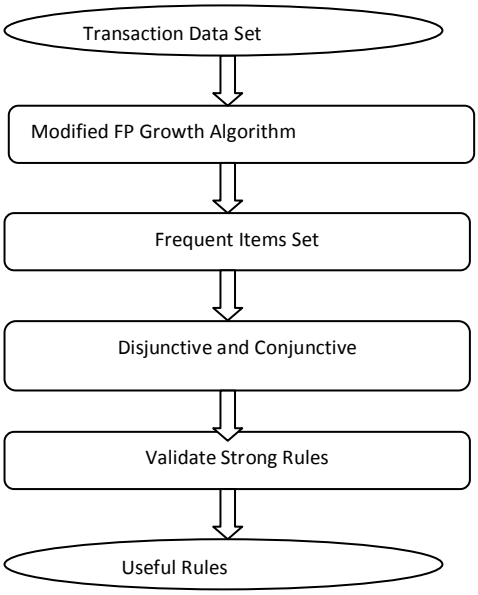
Disjunctive rules with Multiple minimum Supports using Modified FP growth Algorithm using E-Rules:

'E-Rules'[13] is an algorithm to derive positive conjunctive & disjunctive with the help of genetic algorithm where in FP growth approach was modified accordingly. It is an integrated algorithm for useful and effective association rule mining to capture even useful rare items; Conjunctive& Disjunctive rules sets using Genetic Algorithm; Lift Factor to analyse the strength of derived rules. The previous work was predominantly worked on 'time taken' to generate frequent item sets.It was found that there was a drastic difference in 'time taken' as modified FP growth algorithm was used in the pace of Apriori algorithm. The salient features of our earlier were i) Apriori Algorithm has been replaced with modified FP algorithm to reduce the time in generating candidate item generation ii) Genetic Algorithm is used in generating rule set iii) An integrated approach is proposed in combining Multiple minimum support, Conjunction and disjunction rule generation ,Lift factor to validate the result set.

A. Summary of earlier work:

General Framework of E-Rules:

From the below diagram the overall design framework of our earlier work can be understood easily.



As the data set containing various transactions is given as input to E-Rules, it will generate the frequent item set for all possible LFH and RHS rule sets out of it using modified FP growth algorithm. All possible positive rule sets (Disjunction and Conjunction) are generated as per proposed rule guides [5].For the rules generated confident and Lift factors are calculated to validate the strength of the rules. The output is considered as Useful Rules.

Example:An example is given to demonstrate 'ERules' .Table II shows dataset that contains 10 transactions and 7 items and the ordered items as per their frequency.

Table II :Dataset for 10 transactions:

TID	Items	Ordered Items
1	ABDG	BADG
2	BDE	EBD
3	ABCEF	EBACF
4	BDEG	EBDG
5	ABCEF	EBACF
6	BEG	EBG
7	ACDE	EADC
8	ABE	EBA
9	ABEF	EBAF
10	ACDE	EADC

Modified FP growth Algorithm has been taken for generating Frequent Item set to reduce the time taken to generate frequent item sets. Major steps in FP-growth are,

Step1: It firstly compresses the database showing frequent item set in to FP-tree. FP-tree is built using 2 passes over the dataset.

Step2: It divides the FP-tree in to a set of conditional database and mines each database separately, thus extract frequent item sets from FP-tree directly. FP-tree is a frequent pattern tree can be defined below.

FP Tree for the dataset in Table II:

Figure :1 FP Tree generation:

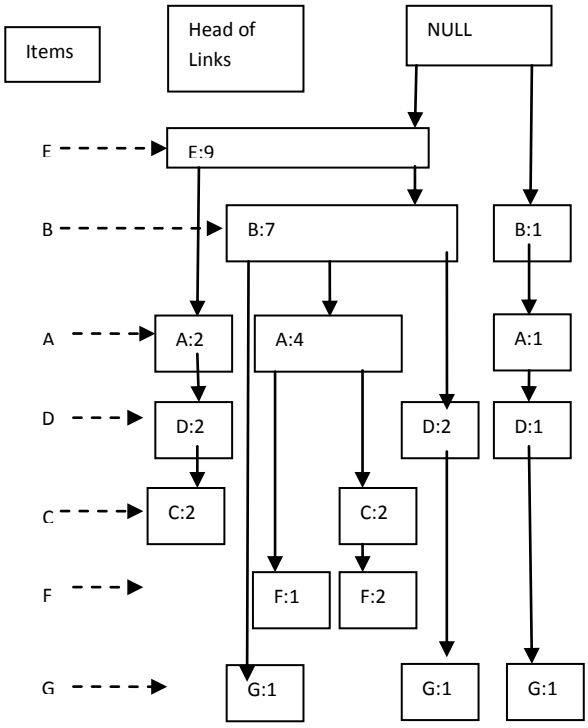


Table III: Conditional Pattern Bases and Frequent items generation:

Item	Conditional Path	Conditional Pattern Base	Conditional FP tree
G	{(E:9,B:7),(E:9,B:7,D:2),(B:1,A:1,D:1)	{(E:1,B:1), (E1,B:1),(B1,A1,D1)}	{(E:2,B:3)}|G}
F	{(E:9,B:7,A:4),(E:9,B:7,A:4,C:2)}	{(E:1,B:1,A:1),(E:2,B:2,A:2)}	{(E:3,B:3,A:3)|F}
C	{(E:9,A:2,D:2),(E:9,B:7,A:4)}	{(E:2,A:2,D:2),(E:2,B:2,A:2)}	{(E:4,A:4)|C}
D	{(E:9,A:2),(E:9,B:7),(B:1,A:1)}	{(E:2,A:2),(E:2,B;2),(B:1,A:1)}	{(E:4,A:3,B:3)}|D
A	{(E:9),(E:9,B:7),(B:1)}	{(E2),(E:4,B:4),(B:1)}	{(E:6,B:5)}|A
B	{(E:9)}	{(E:7),(ø)}	{(E:7),(ø:1)}|B
E	ø	ø	ø

From the FP-tree construction process, it is seen that one needs exactly two scans of the transaction database. Let us see how the FP tree is constructed for node C. For node C, it derives a frequent pattern (E:4, A:4) and two paths in the FP-tree (E:9,A:2,D:2) and (E:9,B:7,A:4) .The first path indicates that string (E,A,D and C) appears twice in the database. To study which string appear together with C, only C's prefix path {E:2; A:2; D:2} counts. Similarly, the second path indicates string (E,B and A) appears twice in the set of transactions in DB, or C's prefix path is {E2,B:2,A:2).These two prefix paths of C form C's sub-pattern base, which is called C's conditional pattern base (i.e., the sub-pattern base under the condition of C's existence). Construction of an FP-tree on this conditional pattern base (which is called C's conditional FP-tree) leads to branch (E:4,A:4). Hence the frequent patterns are generated as per Algorithm 2. They are {E,A,C,(E,A),(E,D),(E,C),(E,A,D),(E,A,C),(E,A,D,C)} . The search for frequent patterns associated with C terminates. Frequent itemset (E, A, C) has been taken for the demonstration purpose. The possible association rules can be generated as per below table. First column in the below table denotes whether it is Antecedent or Consequent; 1-Antecedent 0-Consequent.

Table IV : Encoded Frequent Item sets for Antecedent and Consequents:

Antecedent/Consequent	A	C	E
1	1	0	0
1	0	1	0
1	0	0	1
1	1	1	0
1	0	1	1
1	1	0	1
0	1	0	0
0	0	1	0
0	0	0	1
0	1	1	0
0	0	1	1
0	1	0	1

1) Bitwise items storage:

In order to read and scan the item sets for the calculation, each item needs to get encoded with respect to the transaction into bits. Here the encoding style could be 1 for presence of item in transaction and 0 for absence. Find below the Bit pattern for items.

Table V: Encoding pattern for each Item:

Item	Transaction ID	Bit Pattern	Count
A	{TID1,ITD3,TID5, TID7,TID8,TID9, TID10}	1010101111	7
B	{TID1,TID2,TID3, TID4,TID5,TID6, TID8,TID9}	1111110110	8
C	{TID3,TID5,TID7, TID10}	0010101001	4
D	{TID1,TID2,TID4, TID7,TID10}	1101001001	5
E	{TID2,TID3,TID4, TID5,TID6,TID7, TID8,TID9,TID10}	0111111111	9
F	{TID3,TID5,TID9}	0010100010	3
G	{TID1,TID4,TID6}	1001010000	3

Using the above encoding Genetic Algorithm is deriving possible positive rules for Frequent Item sets. It can be generated as per Rule sets List 1 in Table I."If A is bought then C and E also bought"A⟹CE; "If A is bought then C or E also bought " A⟹C or E Similarly the below possible rules can be derived.

Table VI: Set of Positive conjunctive and disjunctive rules:

Rule			Confident	Lift
Antecedent	Symbol	Consequent		
E	⟹	AC	0.444444444	1.111111
C+E	⟹	A	0.666666667	0.952381
A	⟹	CE	0.571428571	1.428571
AE	⟹	C	0.666666667	1.666667
A	⟹	C+E	0.857142857	0.952381
E	⟹	A+C	0.666666667	0.952381
A+E	⟹	C	0.4	1
A+C	⟹	E	0.857142857	0.952381
C	⟹	AE	1	1.666667
CE	⟹	A	1	1.428571
AC	⟹	E	1	1.111111
C	⟹	A+E	1	1

In the above table the Confident and its relative lift factors are calculated as per the below formula.Confident and Lift factors can be calculated using Formula List as mentioned above.In the above table there are only the conjunctives and disjunctive positive rules are observed. Hence the scope of introducing negative rules is realized.As a final step the ultimate useful rules can be identified using its confidence and lift factors. The below table shows the final rules for the frequent item set (E,C and A) .The predefined minimum confident was 0.75 hence the rules whose confident <0.75 can be pruned by 'E-Rules' .Hence the result set is as follows,

Table VII: Pruned Rules:

Rule			Confident	Lift
Antecedent	**Symbol**	**Consequent**		
A	⇒	C+E	0.857142857	0.952381
A+C	⇒	E	0.857142857	0.952381
C	⇒	AE	1	1.666667
CE	⇒	A	1	1.428571
AC	⇒	E	1	1.111111
C	⇒	A+E	1	1

This is the time to validate the reliability and usefulness. Comparing Lift with Confidence, the Lift of a rule is a relative measure in the sense that it compares the degree of dependence in a rule versus independence between the consequent items and the antecedent items. The rules that have higher Lift will have higher dependence in them.A lift ratio greater than 1.0 suggests that there is some usefulness to the rule. Hence the useful rules were finalized by checking lift value(>1) as well. 'E-Rules' generates conjunctive; disjunctive and positive rules effectively. But the negative rules are not addressed.

II. THE PROPOSED APPROACH: NEGATIVE ASSOCIATION RULES

In much of association rule mining algorithms, result sets are derived for positive rules. And the useful rules are decided based on factors like pre defined Confidence, Lift etc. In the previous work Table VII shows such useful rules which are having confidence greater than 0.75 and Lift almost 1 or more. These rules are only focusing the itemset or combination of itemsets which are present in the dataset. But the absence of these item sets is not dealt. Observing the absence of items from a transaction record will produce a more complete result [14]. Hence to mine negative association rules, the information of all rules sets regardless of their confidence/Lift needs to be maintained to produce negative rule set for each of its positive rule set. Otherwise, the relationships involving the absence of these item sets cannot be discovered. If an infrequent rule set is discarded, then it is not possible to further discover negative association rules that involve the absence of this rule set; for example, $X \Rightarrow \neg Y$ and $\neg X \Rightarrow \neg Y$. Consequently, the opportunity to report a large number of negative association rules involving the absence of item sets will be lost. Thus, avoidance of infrequent patterns is a major hindrance in discovering negative association rules.The loss of negative association rules involving the same item sets used by the positive association rules can be analyzed as below. Among two item sets X and Y used by a positive association rule $X \Rightarrow Y$, there can exist three other negative association rules $X \Rightarrow \neg Y$, $\neg X \Rightarrow Y$ and $\neg X \Rightarrow \neg Y$.

A. Negative rule set list for Conjunctive and Disjunctive rules:

Assume that union of item sets X and Y are present in data set.The association rule of the same can be represented as Xc ⇒Yc where c represents the conjunction nature of antecedent and consequent. The support for these item sets can be expressed using probability, P(Xc U Yc). The support of not observing the same item sets in the same transaction records, also known as absence of an item sets, is given by P ¬(Xc U Yc). And it is nothing but P ¬(Xc U Yc) $= 1 - P$(Xc U Yc). We can analyse for the loss of negative association rules involving the same item sets used by the positive association rules. Among two item sets X and Y used by a positive association rule $X \Rightarrow Y$, there can exist three other negative association rules $X \Rightarrow \neg Y$, $\neg X \Rightarrow Y$ and $\neg X \Rightarrow \neg Y$.Hence for the set of items the conjunctive and disjunctive negative rules can be derived as per the below table.

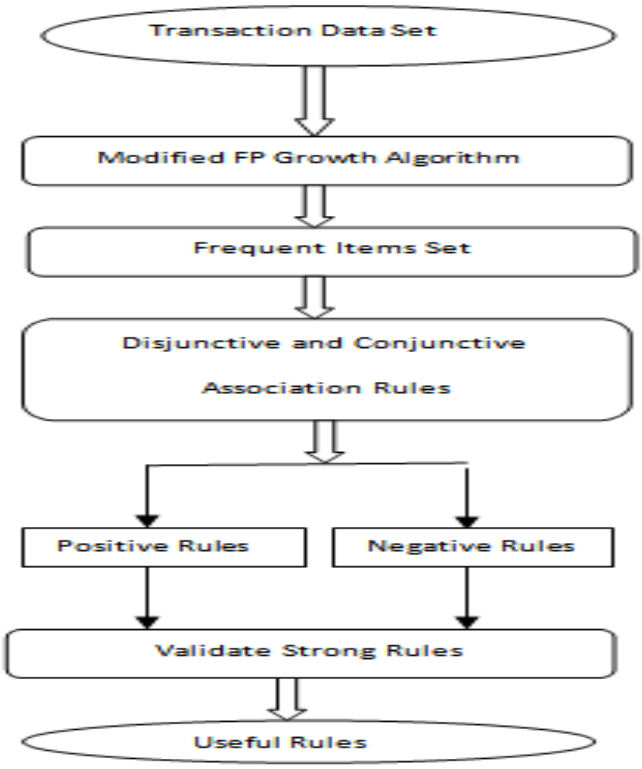
Table VIII: The modified List of Conjunctive and disjunctive for Positive and Negative Ruleset

Positive Rule List	Possible Negative Rules	Support for Presence of ruleset	Support for absence of ruleset
Xc ⇒Yc	Xc⇒¬Yc	s = P(X_C U ¬Y_C)	1-P(Xc U ¬Yc)
	¬Xc⇒Yc	s = P(¬X_C U Y_C)	1-P(¬Xc U Yc)
	¬Xc⇒¬Yc	s = P(¬X_C U ¬Y_C)	1-P(¬Xc U ¬Yc)
Xc ⇒Y_D	Xc⇒¬Y_D	s = P(X_C U ¬Y_D)	1- P(X_C U ¬Y_D)
	¬Xc⇒ Y_D	s = P(¬X_C U Y_D)	1- P(¬X_C U Y_D)
	¬Xc⇒¬ Y_D	s = P(¬X_C U ¬Y_D)	1- P¬ (X_C U ¬Y_D)
X_D ⇒Yc	X_D⇒¬Y_C	s = P(X_D U ¬Y_C)	1- P(X_D U ¬Y_C)
	¬X_D⇒ Y_c	s = P(¬X_D U Y_C)	1- P(¬X_D U Y_C)
	¬X_D⇒¬ Y_C	s = P(¬X_D U ¬Y_C)	1- P(¬X_D U ¬Y_C)
X_D ⇒Y_D	X_D⇒¬Y_D	s = P(X_D U ¬Y_D)	1- P(X_D U ¬Y_D)
	¬X_D⇒ Y_D	s = P(¬X_D U Y_D)	1- P(¬X_D U Y_D)
	¬X_D⇒¬ Y_D	s = P(¬X_D U ¬Y_D)	1- P(¬X_D U ¬Y_D)

Rule sets List of the type **Xc ⟹Yc:**

1. XY Z ⟹VW - Positive Rule
2. XY Z ⟹¬VW - Negative Rule
3. ¬XY Z ⟹VW -Negative Rule
4. ¬XY Z ⟹¬VW -Negative Rule

The above shows the list of possible rule set for itemsets XYZ and VW of the type Xc ⟹Yc , where Xc(Antecedent of conjunctive nature) :XYZ and Yc(Consequent of Conjunctive nature) :VW.



III. FRAME WORK OF THE PROPOSED TASK:

The general overall framework of the proposed work can be represented by the below diagram.

For any given transaction dataset, the proposed algorithm generates the positive, negative rule set in an effective and optimized manner. As per algorithm mentioned in section V, the set of positive and negative rules are generated before the rules are going to be pruned as per its minimum confident and its Lift factor. Hence the chances of losing any useful rules are avoided. So the output of the Algorithm is ensuring the useful and reliable rules.

IV. GENETIC ALGORITHM:

Genetic Algorithm (GA)[1] incorporates Darwinian evolutionary theory with sexual reproduction. GA is stochastic search algorithm modeled on the process of natural selection, which underlines biological evolution. GA has been successfully applied in many search, optimization, and machine learning problems.GA works in an iteration manner by generating new populations of strings from old ones. Every string is the encoded binary, real etc., version of a candidate solution .An evaluation function associates a fitness measure to every string indicating its fitness for the problem. This type of representation is relative to position. Presence of 1 at ith position indicates occurrence of the item in transaction[i]. Similarly presence of 0 at jth position indicates absence of item in transaction[j].For example, Bit pattern for A can be given as 1010101111 based on its availability in the transactions in below table. Similarly ¬A gives 0101010000 based on its absence in each transaction. Hence conjunction and disjunction of few rules can be evaluated and its relative count could be calculated as follows.

Boolean Operation on bit pattern for few positive and negative rules:

Table IX: Bit Pattern for All rules:

Items	Operation	Result	Count
A and C	{1010101111} and {0010101001}	{0010101001}	4
A or C	{1010101111} or {0010101001}	{1010101111}	7
(A or C) and E	({1010101111} or {0010101001}) and {0111111111}	{0010101111}	6
¬(A and C)	¬ ({1010101111} and {0010101001})	{1101010110}	6
¬(A or C)	¬ ({{1010101111} or {0010101001})	{0101010000}	3
(A or C) and ¬E	({1010101111} or {0010101001}) and {1000000000}	{1000000000}	1
¬(A or C) and E	({1010101111} or {0010101001}) and {0111111111}	{0101010000}	3
¬(A or C) and ¬E	({1010101111} or {0010101001}) and {1000000000}	{0000000000}	0

From the above table it is clear that the P(negative rule)=1-P(Positive rule)Where P denotes the probability (refer table)For instance, the frequency of positive rule (A and C) is 4 in the data set, and the frequency of its negative rule ¬ (A and C) is proved to be 6. It is found to be correct in the above table where the number of transactions is 10.The generated rules can be analysed with its confident and Lift factors. Below are few positive & negative rules and its confident & Lift factors for the rule set (A+C)=>E. Similarly the negative rule sets for the other positive rule sets can be generated.

Table X: List of Positive and Negative Rules

Rule			Confident	Lift
Antecedent	Symbol	Consequent		
E	⇒	AC	0.444444444	1.111111
C+E	⇒	A	0.666666667	0.952381
A	⇒	CE	0.571428571	1.428571
AE	⇒	C	0.666666667	1.666667
A	⇒	C+E	0.857142857	0.952381
E	⇒	A+C	0.666666667	0.952381
A+E	⇒	C	0.4	1
A+C	**⇒**	**E**	**0.857142857**	**0.952381**
C	⇒	AE	1	1.666667
CE	⇒	A	1	1.428571
AC	⇒	E	1	1.111111
C	⇒	A+E	1	1
(A + C)	**⇒**	**¬E**	**0.142857143**	**1.42**
¬(A + C)	**⇒**	**E**	**1**	**1.1**
¬(A+ C)	**⇒**	**¬E**	**0**	**0**

From the above it is understood that the positive rules having confidence 1 and lift factor more than 1 are most useful and promising rules. But in the case of negative rules we can understand that the rules are validated if its confident is 1 or lift factor more than 1.Because the absence of any item in a transaction set will also mean that items may not be available for purchase. Hence in that case we get the highlighted rules as useful and promising rules which are going to be playing vital role in analyzing and improving the business.

A. **Genetic operators:**

Genetic Algorithm uses genetic operators [7] to generate the offspring of the existing population. This section describes three operators of Genetic Algorithms that were used in GA algorithm: selection, crossover and mutation.

1) Selection: The selection operator chooses a chromosome in the current population according to the fitness function and copies it without changes into the new population.GA algorithm used route wheel selection where the fittest members of each generation are more chance to select.

2) Crossover: The crossover operator, according to a certain probability, produces two new chromosomes from two selected chromosomes by swapping segments of genes.

3) Mutation: The mutation operator is used for maintaining diversity. During the mutation phase and according to mutation probability, 0.005 in GA algorithm, value of each gene in each selected chromosome is changed.

Genetic algorithms are playing vital role in this overall algorithm. They are involved in, i)generating bit pattern s for each items& rule set and finding out the count ii)generating positive and negative rules based on proposed algorithm iii)calculating confident and lift factors for each rule iv) pruning the useless rules.

V. TO GENERATE NEGATIVE RULES

In order to make 'E-Rules' to generate negative rules, they are modifications needed in the existing algorithm [13]. Each frequent item set in frequent item set derived can be given as one of the input for the below algorithm to derive the respective conjunctive and disjunctive rules.(E,A,C) frequent set has been taken for the demonstration purpose.

Pseudo code for generating Positive and Negative Rules:

Step 1: Create function whose parameters are Dataset, list_of_antecedents-A , list_of_consequent –C

List of A and C can be generated using the following conditions using Genetic Algorithm.

Min (A) =Cnt (FIS)-[Cnt (FIS)-1];Max (A) =Cnt (FIS)-[(Cnt (FIS)-(Cnt (FIS)-1))]

Example, If Frequent Item Set (FIS) =3 then Maximum Antecedes=2 and Minimum Antecedents =1

This holds true for consequents as well, hence, Min(C) = Cnt (FIS)-[Cnt (FIS)-1];Max(C) = Cnt (FIS)-[(Cnt (FIS)-(Cnt (FIS)-1))]

[A- Antecedent; C-Consequent; Cnt- number of frequent item set;Bit pattern for the frequent item set along with operators are like in Table IV.And the function returns interesting rules as per Rule sets like in Table X]

Step 2: Find out allowed antecedents (A) and Consequent(C). Here A and C contains list of FIS.

Step 3: Generation of Negative rules:

P←ø, N←ø AR←ø ---initially null

P-returns Positive rules;N-returns Negative rules;AR- union of Positive and negative rules

For each Item set A and for each item set C, Create an offspring when bit pattern (A) is not equal to bit pattern(C).

If Bit pattern (A) and/or Bit pattern(C)>= min_predefined_min_support then return the rule P U A→C

i. If Bit pattern (A) and/or Bit pattern ¬ (C)>=min_predefined_min_support then return the rule N UA→¬C
ii. If Bit pattern ¬ (A) and/or Bit pattern (C)>=min_predefined_min_support then return the rule N U ¬A→C
iii. If Bit pattern ¬ (A) and/or Bit pattern ¬(C)>=min_predefined_min_support then return the rule N U ¬A→¬C

b. Else Prune the rule.

Step 4: Return P U N

The output of this algorithm is a combination of positive (P) and negative rules (N).

VI. CONCLUSION

'E-Rules' has become a complete algorithm which will generate conjunctive&disjunctive, positive&negative ruleset effectively. Negative rule sets are generated using logical rule set mapping for each positive rule set. Hence it is useful and effective association rule mining algorithm to capture even useful rare items using Genetic Algorithm in more optimal manner. This work is unique as it is dealing an algorithm to generate negative rulesets for disjunctive and conjunctive rules, which was not addressed in the earlier work [13].

REFERENCES

[1] Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K. "Optimized association rule mining using genetic algorithm", Advances in Information Mining, ISSN: 1(2), 2009, 0975–3265.

[2] Marcus C. Sampaio, Fernando H. B. Cardoso, Gilson P. dos Santos Jr.,Lile Hattori "Mining Disjunctive Association Rules" 15 Aug. 2008

[3] Bing Liu, Wynne Hsu and Yiming Ma "Mining Association Rules with Multiple Minimum Supports"; ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99), August 15-18, 1999, San Diego, CA, USA.

[4] Yeong-Chyi Lee a, Tzung-Pei Hong b,_, Wen-Yang Lin , Mining "Association Rules with Multiple Minimum Supports Using Maximum Constraints"; Elsevier Science, November `22, 2004.

[5] Michelle Lyman, "Mining Disjunctive Association Rules Using Genetic Programming" The National Conference On Undergraduate Research (NCUR); April 21-23, 2005

[6] Farah Hanna AL-Zawaidah, Yosef Hasan Jbara, Marwan AL-Abed Abu-Zanona, "An Improved Algorithm for Mining Association Rules in Large Databases" ; Vol. 1, No. 7, 311-316, 2011

[7] Rupesh Dewang, Jitendra Agarwal, "A New Method for Generating All Positive and Negative Association Rules"; International Journal on Computer Science and Engineering, vol.3,pp. 1649-1657,2011

[8] Olafsson Sigurdur, Li Xiaonan, and Wu Shuning."Operations research and data mining,in": European Journal of Operational Research 187 (2008) pp:1429–1448.

[9] Agrawal R., Imielinksi T. and Swami A. "Database mining: a performance perspective", IEEETransactions on Knowledge and Data Engineering 5 (6), (1993), pp: 914–925.

[10] Kannika Nirai Vaani.M, Ramaraj E "An integrated approach to derive effective rules from association rule mining using genetic algorithm", Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference, (2013), pp: 90–95.

[11] Jiawei Han, Jian Pei, and Yiwen Yin. "Mining Frequent Patterns without Candidate Generation", Data Mining and Knowledge Discovery (8), (2004), pp: 53-87.

[12] Oskar Kohonen ,Popular Algorithms in Data Mining and Machine Learning (http://www.cis.hut.fi/Opinnot/T-61.6020/2008/fptree.pdf ;),2008.
[13] Kannika Nirai Vaani.M, Ramaraj E "E-Rules: An Enhanced Approach to derive disjunctive and useful Rules from Association Rule Mining without candidate item generation",VOL 72,NO 19,(June 2013).
[14] Alex Tze Hiang Sim, Maria Indrawan, Samar Zutshi, Bala Srinivasan "Logic-Based Pattern Discovery", IEEETransactions On Knowledge And Data Engineering, VOL. 22, NO. 6,(JUNE 2010).