A New Approach on Rare Association Rule Mining

N. Hoque
Department of Computer Science
Tezpur University

B. Nath
Associate Professor
Department of Computer Science
Tezpur University

D. K. Bhattacharyya
Professor
Department of Computer Science
Tezpur University

ABSTRACT

Association rule mining is the process of finding some relations among the attributes/attribute values of huge database based on support value. Most existing association mining techniques are developed to generate frequent rules based on frequent itemsets generated on market basket datasets. A common property of these techniques is that they extract frequent itemsets and prune the infrequent itemsets. However, such infrequent or rare itemsets and consequently the rare rules may provide valuable information. So, many applications demand to mine such rare association rules which have low support but higher confidence. This paper presents a method to generate both frequent and rare itemsets and consequently the rules. The effectiveness of the rules has been validated over several real life datasets.

General Terms:

Association Rule Mining, Rare Association Rule Mining

Keywords:

Association rule, rare rule, minimum constraint, confidence,multi-objective

1. INTRODUCTION

Association mining is one of the important tasks of data mining intended towards decision support. Basically it is the process of finding some relations among the attributes/attribute values of huge database. Inside the huge collection of data stored in a database, some kind of relationships among the various attributes may exist. Discovering such relationships may help in some decision making process significantly. However, extraction of such relationship from large dataset is not a trivial task. The process of extracting these relationships is termed as association rule mining and can be represented in IF-THEN form. Association rule mining problem was introduced by Agrawal[1] that works on a binary dataset, termed as market basket, where each attribute is termed as an item. In the last two decades several novel works have been evolved to handle the association rule mining problem[2][8][15]. All these algorithms are based on a support-confidence framework. They work in two phases, namely frequent itemset generation and rule generation. The first phase explains the concept of support to derive the frequent itemsets. Support of an itemset can be defined as the proportion of transactions in the dataset which contain the itemset. The confidence of a rule $X \Rightarrow$Y is defined as Conf($X \Rightarrow$Y)=sup($X\cup$Y)/sup(X). A major limitation of these frequent itemset generation techniques is that they can extract only those itemsets which are frequent with respect to a given threshold i.e support-count. However, in practical scenario, there can be some itemsets which have significance but their support counts are relatively less. To extract the relationship among those itemsets having less frequency of occurances, rare association rule mining came into existance. Rare association rules can provide useful knowledge[14] about those relatively infrequent or rare itemsets. However, a major difficulty with single *minsup* based association rule (or frequent itemset) mining approach is that they suffer from the dilemma called ”Rare Item Problem”[13]. If *minsup* is set too high, we miss the frequent itemsets involving rare items because rare items fail to satisfy high *minsup*. To find frequent itemsets consisting of both frequent and rare items, we have to set *minsup* very low. However, this may cause combinatorial explosion and produce too many frequent itemsets.

An association rule *r* is called a valid rare rule if its support is less than a given minimum support denoted by *minsup* i.e. $supp(r)$ <*minsup* and its confidence is greater than a given minimum confidence denoted by *minconf*, i.e. *conf(r)* >*minconf*. Rare association rules are usually required to satisfy a user specified minimum support and a user specified minimum confidence at the same time.

The Rare rules are very important for many applications such as medicine and biology[20]. But the major problem of rare rule generation is that a single *minsup* value can’t determine all the rare rules without generating uninteresting rules. To overcome this problem multiple minsup values are used [9] but still it suffers from the same problem and also dropped some of the rare rules. This has motivated us to develop a rare association rule mining technique to extract all the rare rules without generating uninteresting rules. The rest of the paper is organized as follows-section 2 reports a discussion on rare association rule generation methods. In section 3, the problem is defined followed by the proposed technique, in section 4. Experimental results are shown in section 5. Section 6 describes the multi-Objective rule generation followed by conclusion and future work in section 7.

2. RARE ASSOCIATION RULE GENERATION

In the past couple of years several novel algorithms[1][3][7][11][12][18] have been developed to extract strong association rules, fulfilling the minimum support and minimum confidence requirements. These algorithms are mainly focused on the frequent itemsets generation phase and capable of finding only the frequent itemsets from the dataset, and it drops the infrequent itemsets. But, the main goal here is to generate the rare rules which might give valuable information. Some algorithms were developed for extracting rare itemsets and/or frequent itemsets from a dataset. In this section some of those algorithms are discussed. To describe the algorithms, symbols and notations used are given in Table 1.

[a]Apriori-Rare[20]: The main objective of this algorithm is to generate the frequent as well as rare itemsets. It is the modification of Apriori algorithm to generate minimal rare itemsets. It uses a sub-routine called *Supportcount* to find

Table 1. Symbols Used and their meaning

Symbols	Meaning	Symbols	Meaning
$Minsup$	Minimum support	$Maxsup$	Maximum support
$Minconf$	Minimum confidence	C_i	Candidate itemset
R_i	Rare itemset	F_i	Frequent itemset
LR_i	Large and Rare itemset	D	Dataset
S	Sporadic itemset	MIS	Minimum Item Support
LS	Least Support	NC_k	Not large candidate set
NLC_k	Both Large and not large candidates	MZG	Minimum Zero Generator
NL_k	Itemset satisfying second support	NLL_k	Both large & not large itemset

the support count of a given itemset. The main advantage of this algorithm is that it restores all the *minimal* rare itemsets. However, it fails to find *all* the rare itemsets.

Algorithm 1: Apriori-Rare

```
$C1$ ←All 1-itemsets
$i$ ←1
while ($C_i$ not Null) do
    Supportcount($C_i$)
    $R_i$ = Rare items ($Supportcount$ <Minsup)
    $F_i$ = Frequent items ($Supportcount$ >Minsup)
    $C_{i+1}$ = AprioriGen($F_i$).
    i = i+1
    End of While
end
F = $F_i$
MRI =$R_i$, where $i$=1,2,3.....
```

[b]Apriori-Inverse[10]: This algorithm determines only the sporadic rules using one *Minsup* value and one *Maxsup* value. The sporadic rules have the property that they fall below user define *Maxsup* but above the *Minconf* value. The main advantage of *Apriori-Inverse* is that it can find the sporadic itemsets much more quickly than apriori. However, a major limitation is that it is incapable of finding all the rare itemsets.

[c]MSapriori(Multiple Support Apriori)[9]: It uses multiple *minsup* values to determine the rare itemsets.

The working principle is same as *Apriori* except it uses multiple supports for the items in the dataset. If the actual support of the itemset is larger than the minimum of MIS values of the items present in the itemset then the itemset is called a frequent itemset. It attempts to overcome the rare item problem by altering the definition of minimum support with multiple misup values. In the extended model, it uses a user define minimum item support(MIS) for each item present in the dataset and minsup of an itemset is represented as the minimal MIS value among all its items. This way, the user expresses different support requirements for different rules. Let MIS(i) denotes the MIS value of item *i*. The lowest MIS value among the items in an itemset is the minimum support of a rule. With minimum item supports thus enable us to achieve the goal of having higher minimum supports for the rules that only involve frequent items, and having lower minimum supports for rules that involve less frequent items. The MIS values are calculated by the following formula[9]:

$$\begin{array}{rcll} \text{MIS}(i) & = & \text{M}(i), & \text{if } M(i) > LS \\ & = & LS & \text{otherwise} \end{array} \quad \text{where } \text{M}(i)=\beta * f(i),$$

where $0 \leq \beta \leq 1$ and LS is least support. This model tries to solve the rare item problem using MIS values but due to the value of β, it still suffers from the same rare item problem as *Apriori*. The value of β plays an important role in extraction of rare itemsets which is a major drawback because it does not depends on the frequency f of the itemsets. So the user specified β is an important factor. It is capable of solving the rare item problem of Apriori, however, here the rare itemsets are determined based on a user defined threshold β rather than the frequency of occurrence.

Algorithm 2: Apriori-Inverse

```
$|D|$ ←Size of Dataset
Generate inverted index I of (item, [TID-list]) from D.
Generate sporadic itemsets of size 1:
$S1$ =NULL
for each item $i \in I$, do
    if (count($I$,i)/$|D|$) <maximum support and
    if $count$(I, $i$) >minimum absolute support
    $S1$ =S1+i
end
Find $S_k$ the set of sporadic k itemsets where $k \geq 2$
for $k$=2;$S_{k-1}$ is not null, do
    $S_k$=NULL
    for each $i \in I$ itemsets that are extension of $S_{k-1}$, do
        if all subsets of i of size k-1 belong's to $S_{k-1}$ and
        $count(I, i)$ >minimum absolute support then $S_k$ = $S_k$
        + i
    end
end
return $S_k$, for all k=1,2,3....
```

[d]RSAA algorithms (Relative Support Apriori Algorithm)[21]: The RSAA algorithm generates rules which involve significant rare itemsets. The main objective of this algorithm is to increase the support threshold values for the items having lower frequency and decrease the support threshold for items having higher frequency of occurrences. Like Apriori and MSapriori, RSAA is exhaustive in its generation of rules, so it spends a significant amount of time looking for the rules which are not rare. If the minimum permissible relative support count is set close to zero, then RSAA takes a similar amount of time to that taken by Apriori to generate low support rules. To generate candidate itemsets in RSAA, we should be able to construct the candidate itemset that contains rare data. The set of candidate itemsets in RSAA consists of two groups. One group includes the frequent items that satisfy the first support, and the other group includes the rare items that do not satisfy the first support count but satisfes the second support count. The former group is the same set as the one computed by Apriori. RSAA is exhaustive in its generation of rules, so it spends significant amount of time looking for rules which are not sporadic. However, it uses two thresholds, one is *Minsup* and the other is *Maxsup*.

[e]ARIMA (Another Rare Itemset Miner Algorithm)[19]: It initially calls the *Apriori-Rare* that generates the *Minimal Rare Itemsets*. ARIMA takes these MRIs and produces the rare itemsets. It uses the concept of zero generator to reduce the search space. The main advantage of ARIMA is that it can find rare itemset without generating zero itemsets. However, it is dependent on two threshold values, i.e. *minsup* and *maxsup*.

Table 2. Rare Association Mining Techniques: A General Comparison.

Method	Apriori-Inverse	Apriori-Rare	MSapriori	RSAA	ARIMA
Input parameter(s)	$Minsup$	$Minsup$	$Minsup$	$Minsup$	$Minsup$
proof of correctness	Yes	Yes	Yes	Yes	Yes
Proof of completeness	No	No	Yes	No	Yes
Type of Dataset	binary	binary	binary	binary	binary
No. of DB scans	multiple	multiple	multiple	multiple	multiple
Approach	bottom-up	bottom-up	bottom-up	bottom-up	bottom-up
Candidate generation	Yes	Yes	Yes	Yes	Yes
Type of Itemset	Sporadic	Minimal Rare, Frequent	Rare	Rare	Rare

Algorithm 3: RSAA

```
if($k$ ==2) then
Insert into $NC_2$
    select $p.item_1$; $q.item_1$ from $NL_1$.p;$NL_1$.q
insert into $NLC_2$
    select $p.item_1$; $q.item_1$ from $NL_1$.p; $L_1$.q
else
insert into $NC_k$
    select $p.item_1$; $p.item_2$;$\cdots$; $p.item_{k-1}$; $q.item_{k-1}$
from $NL_{k-1}$.p;$NL_{k-1}$.q
where $p.item_1$=$q.item_1$;
$p.item_2$=$q.item_2$;$\cdots$;$p.item_{k-1}$=$q.item_{k-1}$,
$p.item_{k-1} < q.item_{k-1}$;
insert into $NLC_k$
    select $p.item_1, p.item_2, \cdots, p.item_{k-1}, q.item_1$
from $NLL_{k-1}.p$, $NLL_{k-1}.q$
where $p.item_1 = q.item_1$; $p.item_2 = q.item_2$;$\cdots$;
$p.item_{k-1} = q.item_{k-1}$; $p.item_{k-1} < q.item_{k-1}$
```

Algorithm 4: ARIMA

```
$MZG$ =Null
$S$ =all attributes in D
$i$ =length of smallest itemset in MRI
$C_i$ =i-long itemsets in MRI
$MZG$ =i-itemsets having support count zero
$R_i$=i-itemsets having Support larger than zero
while ($R_i$ not NULL), do
    loop over the elements of $r$ in $R_i$
        Cand = All possible supersets of $r$ using $S$
          loop over the element of Cand(c)
                if $c$ has a proper subset in MZG then delete $c$
    from $Cand$
          $C_{i+1}$=$C_{i+1}$ + Cand
          $Cand$ = NULL
    Supportcount($C_{i+1}$)
    $C_{i+1}$=$C_{i+1}$ +(i+1) long itemsets in MRI
    $MZG$ =MZG + (i+1) itemsets having support count zero
    $R_{i+1}$=(i+1) itemsets having Support larger than zero
    i=i+1
end
$I_R$=Union of all rare itemsets
```

2.1 Discussion and Motivation

Based on our limited experimental study on the existing algorithms for rare association mining, as reported in Table 2, it can be observed that-

—All the algorithms are capable of generating rare itemsets but except *ARIMA* and *MSapriori*, the rest algorithms do not guarantee *proof of completeness*;

—*ARIMA* can find all the rare itemsets but it is dependent on *Apriori-Rare* for the MRIs;

—*Apriori-Inverse* finds only a subset of rare itemsets;

—*MSapriori* ensures the *proof of completeness*, but the rules it generates are not all interesting;

Agrawal's algorithm generates all the frequent itemsets and drops the rare itemsets. However, it also generates a special superset of rare itemsets called the *Minimal Rare Itemsets* (MRIs). Following this algorithm, several variants have been proposed, but most of them generate only frequent itemsets and consequently the frequent rules. *Apriori-Rare*, another variant of *Apriori*, retains MRIs instead of dropping them and generates the rare itemsets. But a major limitation of this algorithm is that it cannot determine all the rare itemsets. *Apriori-Inverse* finds the perfectly rare itemsets i.e. rare itemsets such that all their subsets are rare, but again it cannot generate all the rare itemsets. To overcome this rare item problem, the *MSapriori* was introduced which uses multiple *Minsup* values to find the rare itemsets. Again, it also cannot determine all the rare rules and often found to generate uninteresting rules. *ARIMA* uses the MRIs generated by *Apriori-Rare* and uses the concept of *Minimum Zero Generator* (MZG) to find the rare itemsets. Thus, based on our limited experimental study on these algorithms, it is observed that most algorithms suffer from the limitations of huge memory requirements and execution time. Another limitation of these algorithms is their dependency on multiple user parameters, which are difficult for appropriate assessment.

So, we are motivated to introduce (i) an algorithm (referred here as *NBD-Apriori-FR*) for generation of both frequent as well as rare itemsets and (ii) a method to generate the rules without loss with minimum number of database scans and by fulfilling the three objectives: confidence, comprehensibility and interestingness.

3. PROBLEM FORMULATION

For a given dataset say D, with reference to a user-defined threshold '*Minsup*', the problem is to find all the frequent and rare itemsets without violating the *proof of correctness* with minimum database scans and to find all the rare rules by fulfilling three objectives, namely *confidence*, *comprehensiveness* and *interestingness*.

4. NBD-APRIORI-FR:PROPOSED METHOD FOR FREQUENT AND RARE ITEMSET FINDING

Our method i.e *NBD-Apriori-FR* uses the same downward closure property and botton-up approach of *Apriori* algorithm. It takes D and *Minsup* as inputs and produces both rare and frequent itemsets as outputs. In an initial step, it makes one database scan to find the support counts of all the single-itemsets. Then based on the support counts, it categorizes the itemsets as: *zero itemsets* having support count zero,*frequent itemsets* having support counts greater than *Minsup* and *rare itemsets* having support counts less than *Minsup*. Next, it generates *three* candidate lists. The *first* candidate list is generated from the frequent single-itemset, *second* candidate list is generated from the rare itemset and the *third* list is generated by combining frequent and rare itemsets. Now, these three lists are combined to make one *single* *list* and make one database scan to find the *zero*, *frequent* and *rare itemsets* of size 2. This procedure continues for itemsets of larger size until no more frequent or rare itemsets are produced. But, before scanning the database for k itemsets where $k \geq 2$, this algorithm first generates the $(k-1)$ subsets of the candidate k itemsets. If any $(k-1)$ subsets belong to the $(k-1)$ zero list then that k itemset is put into the zero k list. Finally, from the rare itemsets it generates rare rules using three objectives: confidence, comprehensibility and interestingness based on pareto optimal solution. The steps of the algorithm are given next.

Algorithm 5: NBD-Apriori-frequent-rare

```
$D$ =Dataset
$i$ =1
$C_i$ =All 1 itemsets
Supportcount($C_i$)
$L_i$=$i$ itemsets having supportcount larger minsup
$R_i$=$i$ itemsets having supportcount smaller minsup
$MZG$=$i$ itemsets having supportcount zero
while ($L_i$ or $R_i$ is not Null), do
    $L_{i+1}$=CandidateGen($L_i$)
    $R_{i+1}$=CandidateGen($R_i$)
    $LR_{i+1}$=CandidateGen($L_i$,$R_i$)
    $i$=$i$+1
    $C_i$=$L_i$ + $R_i$ + $LR_i$
    Supportcount($C_i$)
    $L_i$=$i$ itemsets having supportcount larger minsup
    $R_i$=$i$ itemsets having supportcount smaller minsup
    $MZG$=$i$ itemsets having supportcount zero
end
$L$=Union of all large itemsets
$R$=Union of rare itemsets
```

4.1 Complexity Analysis

The overall complexity of the algorithm is basically depends on the size of the dataset and the user defined parameter *Minsup*. For any dataset of n attributes and m records, the approximate complexity is $O(n^{k+1} \times m)$, where k is the maximum length itemset.

4.2 Proof of Correctness

In this section we establish that *NBD-Apriori-FR* is correct in generating both frequent and rare itemsets. Following lemma provides the proof of correctness of our *NBD-Apriori-FR*.
Lemma 1: *NBD-Apriori-FR* is correct i.e the itemsets generated by the algorithm are either frequent or rare with reference to the user defined threshold *Minsup*.
Proof: The correctness of *NBD-Apriori-FR* can be established from the fact that it generates the final list of frequent and rare itemsets based on three candidate lists i.e *zero, frequent* and *rare* with reference a user defined threshold i.e *Minsup*. An itemset is put in the final list of frequent and rare itemset iff it satisfies the *Minsup* condition, hence the proof.

5. EXPERIMENTAL RESULTS

To implement the method we used C++ in a linux environment on a 32-bit workstation having 2.94 Ghz core2 Due processor, 4GB RAM and 360GB Secondary storage.

5.1 Datasets Used

To evaluate the performance of the proposed *NBD-Apriori-FR* we used three synthetic and four real-life benchmark UCI datasets with various dimensinality and number of instances. The characteristics of the datasets used are reported in Table 3.

Table 3. Datasets used for evaluation.

Dataset	Type	Attributes	Records
cancer	Real	4	32
Monk1	Real	5	423
Monk3	Real	5	423
Mushrooms	Real	128	8413
T20D10000k	Synthetic	5	20
T100D10000k	Synthetic	10	100
T1000D90000k	Synthetic	12	1000

5.2 Results

The peformance of the proposed method was compared with *Apriori* and two other well known rare itemset finding techniques, viz *Apriori-Rare* and *ARIMA* and the results are reported for each dataset in tables 4 through 6.

Table 4. Results on *Monk1* and *Monk3* dataset for Minsup 80%.

Algorithm	*Monk1*		*Monk3*	
	FIs	RIs	FIs	RIs
Apriori	4	0	4	0
Apriori-Rare	4	5	4	5
ARIMA	0	28	0	28
Proposed	4	6	4	6

Table 5. Results on *Cancer* and *Mushrooms* dataset for Minsup 50%

Algorithm	*Cancer*		*Mushrooms*	
	FIs	RIs	FIs	RIs
Apriori	7	0	163	0
Apriori-Rare	7	0	163	147
ARIMA	0	0	0	43,907
Proposed	7	0	163	47,767

5.3 Discussion

From the experimental results it can be observed that the proposed method can find both frequent and rare itemsets without any loss of frequent and rare itemsets. As can be seen from the tables that in case of all the datasets, our method can find all the frequent itemsets generated by *Apriori*. In case of *cancer* dataset, none including our method finds a rare itemset. However, in case of other datasets, our method consistently finding more rare itemsets than *Apriori-rare*. However, in case of three datasets, *ARIMA* finds more number of rare itemsets than ours. Also, it has been observed that it takes almost same amount of time and same number of database scan like Apriori. Since, it maintains a zero itemsets list so that if any subset of a candidate list belongs to the zero lists then its support will be zero and it does not require any database scan to count the support. Though *ARIMA* tries to find all the rare itemsets but it needs twice the time as required by *Apriori*. Because, first it calls *Apriori-Rare* algorithm to generate the Minimal Rare Itemsets and from those MRIs, finally it generates the rare itemsets.

6. RARE RULE GENERATION USING MULTI-OBJECTIVE GENETIC ALGORITHM

Several methods have been introduced in the past couple of years either by using bottom-up or top-down or by using hybridization of both to find the frequent rules using the support-confidence framework. However, a major limitation of those methods is that it contains only one item in the consequent part. It is resolved by

Table 6. Results on *T20D10000K*,*T100D10000K* and *T1000D90000K* dataset for Minsup 50%

Algorithm	*T20D10000K*		*T100D10000K*		*T1000D10000K*	
	FIs	RIs	FIs	RIs	FIs	RIs
Apriori	31	0	511	0	15	0
Apriori-Rare	31	0	511	60	15	62
ARIMA	0	92	0	10,976	0	844092
Proposed	31	40	511	38,143	15	903768

the Srikant's first algorithm [17] that practically it may contain any number of items in the consequent part. Again most methods check some candidate rules unnecessarily that waste significant amount of time. Srikant's second algorithm [17] overcomes this problem by eliminating unnecessary checking of candidate rules. But in rare rule generation we cannot directly use these algorithms because rare itemsets may have zero support value. So, specifically to address this rare rule generation problem using single objective based support-confidence framework may not be the right choice. So, here we introduce a multi-objective pareto[16] based rule generation method to generate the rare rules.

6.1 Multiobjective Pareto based rule generation method

Multi-objective problems have multiple objectives and hence multiple solutions.So, it is always very difficult to find out a single optimal solution from a set of multiple solutions. In such problems, it is natural to find out a set of solutions depending on non-dominance criterion[4][5][6][22]. The decision maker takes a decision based on the solution that seems to fit better depending on the circumstances can be chosen from the set of these candidate solutions. A solution, say a, is said to be dominated by another solution, say b, if and only if the solution b is better or equal with respect to all the corresponding objectives of the solution a, and b is strictly better in terms of at least one objective. Here the solution b is called a non-dominated solution. The *NBD-Apriori-MOFR* algorithm uses three objective functions mainly *confidence, comprehensibility* and *interestingness* for rare rule generation. The confidence of a rule (A →C) is support($A \cup C$)/support(A). Comprehensibility and interestingness are defined as follows: Comprehensibility=log(1+P)/log(1+Q)
Interestingness=[SUP($A \cup C$)/SUP(A)][SUP($A \cup C$)/SUP(C)] [1-(SUP($A \cup C$)/R)]. Here, P, Q and R are the size of consequent part, size of the whole rule and total number of records in the database respectively. A is the antecedent and C is the consequent. Next we report our MOGA (Multi-objective GA) based rare rule generation method.
Algorithm *NBD-Apriori-MOFR*:

(1) Load a sample of records from the database that fits in the memory.
(2) Generate N chromosomes randomly.
(3) Decode them to get the values of the different attributes.
(4) Scan the loaded sample to find the support of antecedent part, consequent part and the rule
(5) Find the confidence, comprehensibility and interestingness values.
(6) Rank the chromosomes depending on the non-dominance property.
(7) Assign fitness to the chromosomes using the ranks, as mentioned earlier.
(8) Bring a copy of the chromosomes ranked as 1 into a separate population, and store them if they are non-dominated in this population also. If some of the existing chromosomes of this population become dominated, due to this insertion, then remove the dominated chromosomes from this population.
(9) Select the chromosomes, for next generation, by roulette wheel selections scheme using the fitness calculated in Step 7.
(10) Replace all chromosomes of the old population by the chromosomes selected in Step 9.
(11) Perform multi-point crossover and mutation on these new individuals.
(12) If the desired number of generations is not completed, then go to Step 3.
(13) Decode the chromosomes in the final stored population, and get the generated rules.

The motivation for developing a multi-objective genetic algorithm(GA) for rule generation was that (i) GAs are a robust search method, capable of effectively exploring the large search spaces often associated with attribute selection problem; (ii) GAs perform a global search, (iii) GAs already work with a population of candidate solutions, which make them naturally suitable for multi-objective problem solving where the search algorithm is required to consider a set optimal solutions at each iteration.

6.2 Proof of Correctness

Following lemma provides the proof of correctness of our *NBD-Apriori-MOFR*.
Lemma 2: The *NBD-Apriori-MOFR* algorithm is correct i.e the rules generated from the rare itemsets are ranked according to their optimal solutions of confidence, comprehensibility and interestingness to find the best set of rules.
Poof: Our algorithm can be proved as correct from the fact that the confidence, comprehensibility and interestingness of a rule say A is greater than or equal to another rule say B and at least one objective measure value among confidence, comprehensibility and interestingness must be strictly greater than the objective measures of other rule.

6.3 Experimental Results

To implement the proposed MOGA based method for rare rule generation, we used the similar programming platform and environment as reported in *section* 5.1. To evaluate the performance of our method we used three benchmark UCI datasets, namely *Monk1, Monk3* and *Mushrooms* dataset, and the results are reported for various *Minsup* values in Tables 7 through 9. Rules generated from different datasets are shown below.

Table 7. Rules generated for *Monk1* Dataset and their effectiveness

Rule	Minsup in %	Confidence	Comprehen sibility	Interes tingness
6, 7 →3	25	0.333333	0.682606	0.069959
1, 4 →8	50	0.500000	0.682606	0.051988
2 →8	50	0.666666	0.792481	0.185969
2, 5 →8	50	0.510638	0.682606	0.053094
6, 7 →2	25	0.333333	0.682606	0.069959

Table 8. Rules generated for *Monk3* Dataset and their effectiveness

Rule	Minsup in %	Confidence	Comprehen sibility	Interes tingness
6, 7 →3	25	0.333333	0.682606	0.069959
1, 4 →8	50	0.500000	0.682606	0.051988
2 →8	50	0.666666	0.792481	0.185969
2, 5 →8	50	0.510638	0.682606	0.053094
6, 7 →2	25	0.333333	0.682606	0.069959

Table 9. Rules generated for *Mushrooms* Dataset and their effectiveness

Rule	Minsup in %	Confidence	Comprehen sibility	Interes tingness
$2, 65 \rightarrow 95$	8	0.300000	0.682606	0.299049
$21, 117 \rightarrow 10$	4	0.500000	0.682606	0.499762
$36, 80 \rightarrow 4$	2	0.250000	0.682606	0.124970
$4 \rightarrow 10$	1	0.250000	0.792481	0.062493

6.4 Discussion

Association rule mining problem can be viewed as a multi-objective problem rather than single objective one. Using confidence, comprehensibility and interestingness as the various objective measures, rare rules are generated from the rare itemsets. Similarly, frequent rules are also generated from the frequent itemsets using the same objective measures. The *confidence* of a rule determines the reliability of the rule. *Comprehensibility* determines the significance of a rule based on number of items present in both the antecedent and consequent part. *Interestingness* helps in knowledge gathering. Based on these measures an attempt has been shown to determine the best set of rules. However, along with the meaningful rare rules, generation of redundant rules is a limitation of our method.

7. CONCLUSION AND FUTURE WORK

Several frequent and rare association rule mining techniques have been studied and reported in this paper. A general comparison among these techniques also has been reported to highlight their pros and cons. To address the limitations of these techniques, an effective Apriori based frequent and rare itemset finding technique has been presented. The superiority of the technique has been established in terms of seven datasets while comparing with its other competing algorithms. Finally, to address the limitation of any support-confidence based single objective rare rule mining method, this paper introduces a multi-objective GA based rare rule generation method. The effectiveness of the method has been shown over three publicly available UCI datasets. Work is going on for further enhancement of our MOGA (Multi-Objective GA) based rare association mining method by considering measures other than confidence, interestingness and comprehensiveness, and for datasets with large number of high-dimensional instances to help finding only meaningful rare association rules. Attempt is also going on to explore the possibility of applying both the methods in network anomaly detection towards identification of known as well as unknown attacks.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. pages 206–216. In ACM SIGMOD International Conference on Management of Data, 1993.

[2] R. Agrawal, T. Imielinski, and A. Vu. Mining association rules with item constraints. pages 67–73. In the Third International Conference on Knowledge Discovery in Databases and Data Mining, 1997.

[3] S. Brin, R.Motwani, J.D.Ullman, and S.Tsur. Dynamic itemset counting and implication rules for market basket data. volume 26, pages 255–268. in Proc. of the 1997 ACM SIGMOD Int'n Conf. on Management of data, 1997.

[4] C.A.C Coello. A comprehensive survey of evolutionary-based multi-objective optimization technique. pages 269–308. Knowledge and information systems, 1999.

[5] C.M. Fonseca and P.L. Fleming and. An overview of of evolutionary algorithms in multi-objective optimization. pages 1–16. Evolutionary Computation 3, 1995.

[6] A.A. Freitas. A survey of evolutionary algorithm for data mining and knowledge discovery. pages 819–845. Advances in Evolutionary Computing, Springer-Verlag,New York, 2003.

[7] J. Han, J. Pei, Y. Yin, and R. Mao. Mining frequent patterns without candidate generation:a frequent-pattern tree approach. volume 8, pages 53–87. Data Mining and Knowledge Discovery, 2004.

[8] R. U. Kiran and P. K. Reddy. An improved multiple minimum support based approach to mine rare association rules. pages 340–347. IEEE Symposium on Computational Intelligence and Data Mining, 2009.

[9] R. U. Kiran and P. K. Reddy. Mining rare association rules in the datasets with widely varying items' frequencies. The 15th International Conference on Database Systems for Advanced Applications Tsukuba, Japan, April 1-4,, 2010.

[10] Y. S. Koh and N. Rountree. Finding sporadic rules using apriori-inverse. pages 97–106. Springer-Verlag Berlin Heidelberg, 2005.

[11] T. S. Kumar, V. Kavita, and T. Ravichandran. Efficient tree based distributed data mining algorithm for mining frequent patterns. volume 10. International Journal of Computer Applications, 2010.

[12] D. I. Lin and Z. M. Kedem. Pincer-search: an efficient algorithm for discovering the maximal frequent set. pages 105–219. In Proc. Of 6th European Conference on Extending DB Tech, 1998.

[13] B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. pages 337–341. ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations, 1999.

[14] H. Mannila. Methods and problems in data mining. pages 41–55, 1997.

[15] B. Nath, D. K. Bhattacharyya, and A. Gosh. Faster generation of association rules. volume 1, pages 267–279. IJITKM, 2008.

[16] B. Nath and A. Ghosh. Multi-objective rule mining using genetic algorithm. pages 123–133. Information Science 163, 2004.

[17] R.Srikant and R. Agrawala. Mining generalized association rules. pages 407–419. Proceedings of the 21st VLDB Conference Zurich, Swizerland, 1995.

[18] A. Savesere, E. Omiecinski, and S. Navathe. An effective algorithm for mining asociation rules in large database. pages 432–443. In proceedings of International Conference on VLDB95, 1995.

[19] L. Szathmary and P. Valtchev. Towards rare itemset mining. Soutenue publiquement le.

[20] L. Szathmary, P. Valtchev, and A. Napoli. Generating rare association rules using the minimal rare itemsets family. volume 4, pages 219–238. International Journal on Software Informatics, 2010.

[21] H. Yun, D. Ha, B. Hwang, and K. H. Ryu. Mining association rules on significant rare data using relative support. volume 67, pages 181–191. The Journal of Systems and Software, 2003.

[22] E. Zitzler, K. Dev, and L. Thiele. Comparision of multi-objective evolutionary algorithms: empirical results. pages 125–148. Evolutionary Computation 8, 2000.