Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MINING INTERESTING POSITIVE AND NEGATIVE ASSOCIATION RULE BASED ON GENETIC TABU HEURISTIC SEARCH

[1]HODA M. WAGUIH, [2]SAAD M. DARWISH, [1]MOHAMED A. OSMAN

[1]Computer and Information Systems Department, Sadat Academy for Management Sciences, Egypt
[2]Department of Information Technology, Institute of Graduate Studies and Research, University of Alexandria, Egypt
E-mail: [1]hoda.waguih@gmail.com, [2]saad.darwish@alex-igsr.edu.eg, [1]osmancooo1@yahoo.com

ABSTRACT

Association rule mining is a very efficient technique for finding a strong relation between correlated data. For the mining of positive and negative rules, a variety of algorithms are used such as Apriori algorithm. Practically the data set can contain a large number of data entities which form a large number of rules set. For the understanding of efficient relation, it is needed that rules should be kept as minimum as possible without losing the useful information. This is a difficult task and can be seen as optimization problem. To eliminate the multi-scan problem and reduce a large number of mined rules, in this paper, we propose a novel approach to mine interesting positive and negative rules from frequent and infrequent pattern set based on genetic Tabu heuristic search with key innovation to optimize genetic population of rules. Genetic Tabu Algorithm (GTA) follows procedures similar to the Genetic Algorithm (GA). However, for the selection process, GTA uses the Tabu search process to enhance the quality of mined rules in some way that is efficient in time and memory. The suggested algorithm is accomplished in two phases: (1) generate frequent and infrequent pattern set. (2) Efficiently generate optimized positive and negative rules by using useful frequent pattern set and GTA. Experiment results show that the proposed algorithm can efficiently generate positive and negative association rules and outperforms the traditional algorithms in terms of time and quality.

Keywords: *Association rule mining, Apriori Algorithm, Positive and negative rules, Genetic Tabu Search, Optimization.*

1. INTRODUCTION

Mining association rule (AR) is a data mining task that aims to discover relationships among items in a transactional database [1]. This task has been studied widely in the literature for its benefit in many application domains. A typical example of association rule mining application is the market basket analysis. In this process, the behavior of the customers is studied when buying different products in a shopping store. The discovery of interesting patterns in this collection of data can lead to important marketing and management strategic decisions. For instance, if a customer buys bread, what is the probability that he/she buys milk as well? Depending on the probability of such an association, marketing personnel can develop better planning of the shelf space in the store or can base their discount strategies on such associations or correlations found in the data [2].

An association rule is an implication or if-then-rule which is supported by data of the form $X \Rightarrow Y$, which can be used to predict that "If a customer buys itemset X, he/she will most likely buy itemset Y as well", where X and Y are frequent item sets in a transaction database and $X \cap Y = \emptyset$. The rule of the form $X \Rightarrow Y$ is considered a strong rule if its support (s) and confidence (c) satisfy minimum support (*mins*) and minimum confidence (*minc*) thresholds; these are called positive association rules (PARs). In addition to PARs, the relationship represented as "customers that buy itemset A do not buy itemset B" refers to a negative relation between two itemsets [3].

In general, there are three general characteristics that discovery of rules must satisfy; to have specifically a high-precision prediction, to be understandable and to be interesting. A measure to predict the association rule precision $X \Rightarrow Y$ is the confidence. This measures the reliability of interference made by the rule which is defined as $C = |X \cup Y|/X$; where $|X|$ is the number of examples that satisfy every condition in the antecedent X and $|X \cup Y|$ is the number of examples both of which satisfy the antecedent X and it has the class predicted by the consequent Y. But the confidence favors the rules

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

over-fitting the data. Due to this, it is necessary to determine the way a rule is applicable in the dataset, such as support. It is defined as $C = |X \cup Y|/N$; where N is the total number of examples. Support is often used to eliminate non-interesting rules. A measure to determine a rule interestingness is to find surprisingness of an attribute based on each attribute information gain [4].

Extraction of frequent item sets is a core step in many association analysis techniques. All generalized frequent pattern sets are not very efficient because a segment of the frequent pattern sets is redundant in the association rule mining. This is why traditional mining algorithm produces some uninteresting rules or redundant rules along with the interesting rule [1][5]. A common solution to deal with the complexity is to focus the search on special cases of interest. Some techniques employ domain knowledge to guide the search, some are focusing on a certain type of rules of interest, while others are considering interestingness measures to mine for statistically significant patterns [6].

Association rules are based on frequencies and count the transactions where items occur together. However, counting absences of items is prohibitive if the number of possible items is very large, which is typically the case. Nonetheless, knowing the relationship between the absence of an item and the presence of another can be very important in some applications. These rules are called negative association rules [6]. Negative association rules are useful in the market-basket analysis to identify products that conflict with each other or products that complement each other. Mining negative association rules is a difficult task, due to the fact that there are essential differences between positive and negative association rule mining. The researchers attack two key problems in negative association rule mining: (i) how to effectively search for interesting itemsets, and (ii) how to effectively identify negative association rules of interest [7].

Imagine a transaction in market basket analysis where a customer buys bread and milk. When mining for positive association rules only those two items are considered (i.e. bread and milk). However, when negative items are considered (i.e. items/products not present in a basket/transaction) the search space increases exponentially because of all the items in the collection, although not present in the transaction have to be considered. Not only is the problem complex, but also large numbers of negative patterns are uninteresting. The research of mining negative association patterns has to take into consideration both the complexity of the problem and the usefulness of the discovered patterns [6].

Many literatures in intelligent data analysis show that negative association rules are as important as positive rules. In the traditional approach, finding negative association rules encounters a large search space and generates too many non-interesting rules. Therefore, an efficient and useful algorithm for finding negative association rules is very valuable. Our research focuses on reducing computing time and trying to find interesting positive and negative association rules. The proposed algorithm could speed up computing time efficiently by adopting Genetic-Tabu Algorithm.

This paper proposes a new positive and negative association rules mining approach which tackles the problems mentioned above (very large number of rules and expensive computing costs). Being different from other approaches of the same category, this new method creates a so called meta-heuristic association rules based on a combination of genetic algorithm and Tabu search to optimize mining parameters to enhance the quality of these rules in some way that is efficient in time and memory. The advantage of Tabu search is to eliminate a large number of useless rules and itemsets. The Genetic Tabu Search Algorithm (GTA) heuristic is a combination of the GA and TS techniques. In general, GTA follows procedures similar to the GA. However, for the selection process, GTA uses the Tabu search process. Such an approach has the advantage of being intuitive to users (no additional parameters required), and may serve as a "skeleton" for the development of more specialized mining algorithms.

The rest of the paper is organized as follows. Section 2 highlights the state-of-the-art related works. Section 3 gives the detailed process of the proposed algorithm. The experiment results and their discussion are presented in Section 4. Finally, in Section 5, we conclude this paper.

2. RELATED WORK

In order to solve positive association rules mining problem, numerous researchers have proposed genetic algorithm decision-making methods [4][5][9]. In this case, the GA-based strategy for identifying association rules without specifying actual minimum support is employed. GA is capable of discovering item sets that occur more frequently over a short time interval of a transaction. It does not exhaustively search the dataset or require any prior partitioning. In all approaches, the fitness function (based on support and confidence) is designed in such a way that to prioritize the rules based on the user preference. See

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[9] for a comprehensive survey regarding these methods.

While these approaches dealt with a challenging NP-Hard association rule mining problem of finding interesting association rules. They don't sufficient reliable for a large dataset, it needs to improve for the application in large data sets. Furthermore, these techniques need major modifications to improve the complexity reduction of association rule mining and genetic algorithms with the help of other optimization techniques.

Algorithms for discovering negative association rules are not widely discussed [5-12]. The discovery procedure of these algorithms can be decomposed into three stages: (1) find a set of positive rules; (2) generate negative rules based on existing positive rules and domain knowledge; (3) prune the redundant rules. But the problem with the negative association rule is it uses large space and can take more time to generate the rules as compared to the traditional mining association rule [1][3].

To solve this problem many authors combined positive frequent itemsets with domain knowledge in the form of taxonomy to mine negative associations. However, their algorithm is hard to generalize since it is domain dependent and requires a predefined taxonomy [10]. In this case, finding negative itemsets involve following steps: (1) first find all the generalized large itemsets in the data (i.e., itemsets at all levels in the taxonomy whose support is greater than the user specified minimum support). (2) next, identify the candidate negative itemsets based on the large itemsets and the taxonomy and assign them expected to support. (3) in the last step, count the actual support for the candidate itemsets and retain only the negative itemsets. Other authors introduced "*mininterest*" parameter to mine positive and negative association rules, but the authors do not discuss how to set it and what would be the impact on the results when changing this parameter.

A novel approach has suggested as stated in [10] in which mining both positive and negative association rules of interest can be decomposed into two sub problems, (1) generate the set of frequent itemsets of interest PL and the set of infrequent itemsets of interest (NL) (2) extract positive rules of the form $X \rightarrow Y$ in PL, and negative rules of the forms $X \rightarrow \neg Y$,$\neg X \rightarrow Y$ and $\neg X \rightarrow \neg Y$ in NL. Another technique relies on four steps to generating positive and negative association rules: (1) Generate all positive frequent itemsets L (P_1) (ii) for all itemsets I in L (P_1), generate negative frequent itemsets of the form $\neg$ (I_1 I_2) (iii) Generate all negative frequent itemsets $\neg I_1$ $\neg I_2$ (iv) Generate all negative frequent itemsets I_1 $\neg I_2$ and (v) Generate all valid positive and negative association rules. Authors generated negative rules without adding additional interesting measure(s) to support-confidence frame work.

Furthermore, an algorithm named SRM (substitution rule mining) discovers a subset of negative associations [10]. Their algorithm discovers first what they call *concrete items*, which are those itemsets that have a high chi-square value and exceed the expected support. Once these itemsets are discovered, they compute the correlation coefficient for each pair of them. From those pairs that are negatively correlated, they extract the desired rules (of the type $X \Rightarrow \neg Y$, where Y is considered as an atomic item).

In [3] the authors deal with an association rule mining problem for finding negative and optimized association rules. The frequent itemsets are generated using the Apriori association rule mining algorithm. Negative rules generating algorithm used modified negative coloration coefficients to generate all negative association rules. After all rule generation, the genetic algorithm is applying to optimize generate rule. The results reported in this paper are very promising since the discovered rules are of optimized rules.

In recent years, the genetic evolutionary algorithm has been received much attention from researchers to mine negative association rule. For instance, the paper in [11] introduced a genetic algorithm-based algorithm to find negative sequential patterns with novel crossover and mutation operations, which are efficient at passing good genes on to next generations without generating candidates. An effective dynamic fitness function and a pruning method are also provided to improve performance. Another paper presented in [12] in which the authors designed efficient mining method by obtaining positive and negative association rules in database and optimization of positive and negative association rule using genetic algorithm, the design of pruning strategies for reducing the search space and improving the usability of mining rules are used to correlate to association with mining methods. The approach is effective, efficient and promising.

Aiming to fill the large search space and many non-interesting rules gaps for finding association rules, this paper proposes a hybrid genetic-Tabu algorithm for mining both of positive and negative association rules. Existing approaches have tried to address these problems by incorporating attribute correlations or rule interestingness measures to filter out unimportant

rules, or by relying on additional background information concerning the data. As opposed to this, the aim of this paper is to propose a computationally workable approach that stays within the strict bounds of the original support-confidence framework. This approach could speed up execution time efficiently and extract interesting rules easily, the advantage of Tabu search algorithm is to eliminate a large number of useless rules. Using Tabu search algorithm`s strong climbing ability to avoid the "premature" phenomenon exist in genetic algorithm effectively. Meanwhile, use the genetic algorithm to get an initial solution to improve the solution quality. It modifies the core operator in GA-cross operator and mutation operator according to the ideal of Tabu search algorithm [13-15].

3. STRUCTURE AND METHODOLOGY

3.1 Problem Formulation

Formally, association rules are defined as follows: Let $I = \{i_1, i_2, \ldots i_n\}$ be a set of items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with a unique identifier TID. A transaction T is said to contain X, a set of items in I, if $X \subseteq T$. An association rule is an implication of the form "$X \rightarrow Y$", where $X \subseteq I$; $Y \subseteq I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has support s in the transaction set D if s% of the transactions in D contain $X \cup Y$. In other words, the support of the rule is the probability that X and Y hold together among all the likely current cases. It is said that the rule $X \rightarrow Y$ holds in the transaction set D with confidence c if c% of transactions in D that contain X also contain Y. In other words, the confidence of the rule is the conditional probability that the consequent Y is true under the condition of the antecedent X. The problem of discovering all association rules from a set of transactions D consists of generating the rules that have a support and confidence greater than given thresholds. These rules are called strong rules, and the framework is known as the support confidence framework for association rule mining [8] [16].

A negative association rule is an implication of the form $X \rightarrow \neg Y$ (or $\neg X \rightarrow Y$ or $\neg X \rightarrow \neg Y$), where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \emptyset$ (Note that although rule in the form of $\neg X \rightarrow \neg Y$ contains negative elements, it is equivalent to a positive association rule in the form of $Y \rightarrow X$. Therefore, it is not considered as a negative association rule.) In contrast to positive rules, a negative rule encapsulates relationship between the occurrences of one set of items with the absence of the other set of items. The rule $X \rightarrow \neg Y$ has support s % in the data sets if s % of transactions in T contain itemset X while do not contain itemset Y. The support of a negative association rule, supp $(X \rightarrow \neg Y)$, is the frequency of occurrence of transactions with item set X in the absence of item set Y. Let U be the set of transactions that contain all items in X. The rule $X \rightarrow \neg Y$ holds in the given data set (database) with confidence c % if c% of transactions in U do not contain item set Y. The confidence of negative association rule, conf $(X \rightarrow \neg Y)$, can be calculated with $P(X \neg Y)/P(X)$, where $P(.)$ is the probability function. The support and confidence of itemsets are calculated during iterations. However, it is difficult to count the support and confidence of non-existing items in transactions. To avoid counting them directly, we can compute the measures through those of positive rules [9] [12].

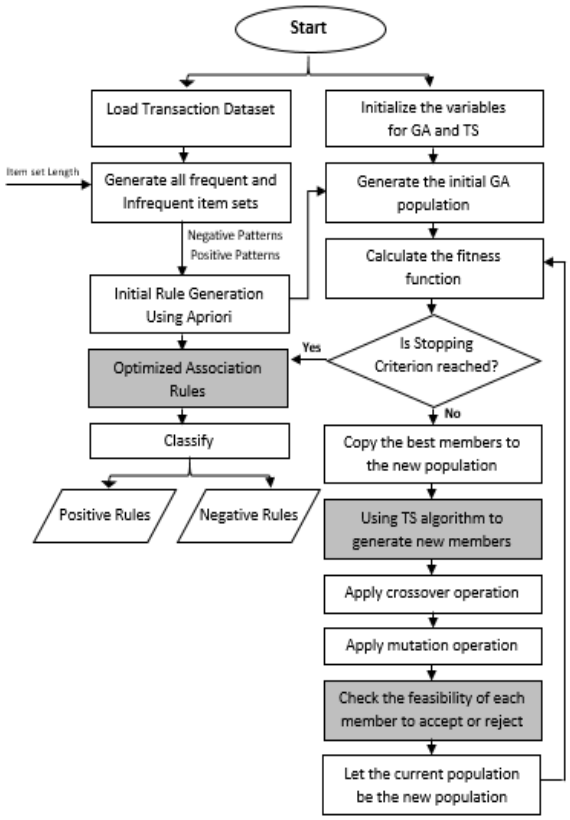
3.2 Motivation and The Proposed Solution

As mentioned above, mining negative association rules, however, raises a number of critical issues. First of all, in typical supermarket transaction databases, there are thousands of items (wares) and each record (customer transaction) only contains a few of them. For a database containing 10,000 items, such that each customer on average buys 10 of them, the density of the database is 0.1%. From the perspective of negative patterns (indicating the absence of items), the density is a soaring 99.9%, leading to explosive amounts of, mostly uninteresting, rules. Secondly, the complexity of AR mining algorithms is exponential in the number of items; if for each item from the database a corresponding negated item (indicating the absence of the original item) is considered, the computational costs will skyrocket. However, intuitive mining algorithms like Apriori suffers from some weakness in spite of being clear and simple [12]. Apriori will be very low and inefficient when memory capacity is limited with a large number of transactions (has exponential complexity [6].

This study proposes a new approach (see Fig. 1) to solve a mining positive and negative association rule problem with the help of integrating GA and TS. For efficiency, the system scans the database once and transform transactions into a space-reduced structure called encoded transactions stored in main memory. In the mining steps, the proposed algorithm uses negative interestingness and negative confidence to increase the accuracy of mined results. Pruning technique through genetic-Tabu search is used to remove non-interesting negative association rules. This approach achieves a better result with the help of efficient chromosome representation, powerful crossover strategies, and neighborhood strategies. The Proposed technique can be described as follows and the next subsections describe each step in details [4] [5] [7] [8] [15-19].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 1: The proposed AR mining system based on Genetic-Tabu heuristic search



- Take an input dataset which contains a number of attributes (items) and instance (records), and convert the data into numeric values.
- Initialize the data with a length of the item sets k = 2, 3, 4 and pass support and confidence. Similarly, initialize all the general parameters involved in genetic algorithm and Tabu search.
- Generate all the frequent and Infrequent item sets based on step 1 from Apriori algorithm for an item set of length k =2, 3, 4. In this case, a frequent itemset I is satisfied when $sup(I) \geq minsupp$; whereas, an infrequent itemset J is fulfilled when $sup(J) \leq minsupp$.
- Apply Apriori algorithm to generate positive association rules from frequent items sets and negative association rules from infrequent item sets.
- Define the fitness function for specific support & confidence values in the case of positive association rules; and negative interestingness in the case of negative association rules.
- GA searches the solution from a population of points instead of a single point. The algorithm is computationally simple and powerful. This step uses the GA to generate the child chromosomes of the positive and negative association rules and calculate the fitness value of each individual child chromosomes. Compare the individual fitness value of each child with the average fitness value and regenerate positive and negative association rules.
- During the hybrid search process, GA starts with a set of initial solution and generates a set of new solutions. On each set of the new solution, TS performs a local search to improve them. Then GA uses the improved solution of TS to continue with parallel evolution. Tabu Search works on the individual string, which are points on the solution space. TS guides the iterations from one neighborhood point to another by locally improving the solution's quality and has the ability to avoid poor local minima. Integration of GA and TS using their own strengths has a good chance of providing a reasonable solution to global combinatorial optimization problems.
- Crossover and mutate the remaining child chromosomes and reinitialize the fitness value and recalculate and regenerate final positive and negative rules.

A. Load Transaction Dataset

The association rules generated from the proposed algorithm needs datasets containing a number of transaction values with a different average number of items for the transaction (T=10,12,15), different average length of maximal frequent patterns (I=6,8,12), and diverse the total number of transactions (D=100,50000,100000). So, the performance of the proposed methodology is tested for each dataset.

B. Parameters Initialization

For the genetic algorithm, it is not too much difficult to set the parameters. In genetic algorithm, we have parameters as follow; (1) number of generations, (2) number of populations, (3) mutation rate, (4) mutation percentage of population, (5) chromosome length in case of binary encoding, and (6) crossover percentage of population [4]. Among these parameters, mutation rate must be very low, as low as 0.05 or even smaller. Because the higher value could destroy the solution. Mutation percentage and crossover percentage are depending on the problem and their own efficiency that you could find an optimal value for them by different runs. If you could settle an optimal value for these three parameters, the other first two parameters would be settled simply [5].

The elements of the TS algorithm in connection with data mining are defined as follows [13] [14]:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

- *Initial solution:* The initial solution can be obtained by various methods such as the priority dispatching rules, the diverse insertion and random methods, etc. even artificial intelligence. Many researchers show that the initial solution method affects the scheduling solution quality; such that the better initial solution is the better TS solution is. In this paper, the association rules generated from the GA are used as an initial solution method.
- *Neighborhood structure:* A neighborhood structure is a mechanism which can obtain a new set of neighbor solutions by applying a simple modification to a given solution. Each neighbor solution is reached from a given solution by a move. Neighborhood structure is directly effective on the efficiency of TS because TS proceeds iteratively from one neighbor to another in problem solution space. Therefore, a neighborhood structure must eliminate unnecessary and infeasible moves if it is possible.
- *Move:* The best neighbor which is not Tabu or satisfies a given aspiration criterion is selected as new seed solution. "The best" neighbor is one whose objective function, C_{max}, is minimum. If all neighbor is Tabu or no neighbor satisfies the aspiration criterion then the oldest neighbor, entering the Tabu list at first, is selected as new seed solution.
- *Tabu list and updating:* There are two common ways of the use of memory in TS: short- and long-term memory. The scholars state that the effect of both types of memories may be viewed a modifying the neighborhood of a given solution. The short-term memory keeps track of the solutions attributes that have changed during the recent history. It is exploited as a Tabu list. In this work, we use only a single Tabu list. The elements added on the list are attributive. The main aim of using an attributive representation is to save computer memory. Tabu list is updated after each move in so far as the strategic forgetting occurs. The move selected replaces to the top of the list and the elements on the list go down one apiece. The element on the bottom of the list is removed from the list, i.e., the temporary memory is updated. The length of Tabu list determines the time limit of remaining on memory for elements. Therefore, it can change the course of the search.
- *Aspiration criterion:* The aim of the aspiration criterion is to override the Tabu status of a neighbor. The aspiration criterion used in this work is as follows: If the move yields a solution better than the best obtained so far then the move is performed even as it is Tabu.
- *Termination criterion:* When the number of the dis-improving moves reaches to a maximum value, set to 2000, or no neighbor is generated or an infeasible solution is encountered, the TS algorithm terminates.

C. Generate all Frequent and Infrequent Itemsets

The problem of frequent item set mining consists at looking only for patterns with support at least equal to a fixed threshold. However, in infrequent item set mining we are exclusively looking for non-common patterns: patterns that are present in the data transactions with a frequency smaller than a fixed threshold. In the general Apriori algorithm the frequent item sets are found by generating the candidate item sets based on minimum support threshold. For finding the infrequent item sets, the support is to be considered that is minimum negative count is considered here. If the support frame work is considered let us take the minimum support threshold that is minimum negative count as 3 (as illustrated in Table 1), 1, 4, 5, 6, 8 are considered as infrequent. The algorithm used to extract infrequent itemsets is as follows [20]:

D. Initial rule generation using Apriori

As is common in association rule mining, given a set of itemsets the algorithm attempts to find subsets which are common to at least a minimum number of the itemsets (frequent and infrequent). Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. the suggested system uses Apriori algorithm with support - confidence framework inside frequent itemsets to create initial positive association rules. For initial negative association rules, the system employs Apriori algorithm with negative interestingness inside infrequent itemsets.

To demonstrate this operation, given the binary relation between items X and Y in Table 2. Each item in the database has two conditions, that is, presence or absence. Therefore, such a four-state table is created for item X and Y. From Table 2, we can easily deduce support and confidence for mining positive (traditional) association rules. For instance, support of rule $X \rightarrow Y$ can be expressed by [8]:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: Infrequent 1-itemsets.

Database

TID	ITEMSET
10	1,3,4,7
20	2,3,5,6,8
30	1,2,3,5,7
40	2,5,8
50	1,3,7,8
60	2,3,7

Infrequent 1-itemsets

ITEM	INFRE-QUENCY	TID
1	3	20,40,60
2	2	10,50
3	1	40
4	5	20,30,40,50,60
5	3	10,50,60
6	5	10,30,40,50,60
7	2	20,40
8	3	10,30,60

Table 2: Binary relations.

Item Y / Item X	Presence	Absence
Presence	N_a	N_b
Absence	N_c	N_d

$$\frac{N_a}{N_a + N_b + N_c + N_d} \quad (1)$$

where N_a represents the number of transactions containing X and Y, N_b is the number of transactions contain only X, N_c indicates the number of transactions contain only Y, N_d specifies the number of transactions containing neither X or Y. whereas the confidence of rule $X \rightarrow Y$ can be expressed as:

$$\frac{N_a}{N_a + N_b} \quad (2)$$

In order to extract interesting negative association rules from large databases, a proper measure for negative association rule mining must be defined. From Table 2, we find that attribute N_a is the condition that X and Y occur at the same time. The others have at least one negative (Absence) factor. Therefore, instead of using traditional measures in equations 1 and 2, a measure for mining interesting negative association rules must be used as follows [8]:

$$X \rightarrow -Y : \frac{N_b}{N_a + N_b} \quad (3)$$

$$-X \rightarrow Y : \frac{N_c}{N_c + N_d} \quad (4)$$

$$-X \rightarrow -Y : \frac{N_d}{N_c + N_d} \quad (5)$$

E. Generate the initial GA population

Generally, the goal of finding the optimal set of such general rules is not an easy task. Many evolutionary learning algorithms have been used to automate association rule mining. Given the above generated positive and negative rules, the following steps are employed to optimize the generated rules. The genetic algorithm requires a population of feasible solutions to be initialized for both positive and negative rules and updated during the evaluation process of the proposed approach; the initial population of rules comes from the rules extracted by means of the Apriori algorithm.

Input: A Transactional Database, minimum negative-count, *k*-item set length.
Output: Infrequent k-Item sets.
Method:

```
start
  Scan the given transactional data base
    for (k=1 to n) do   //n is the max no. of items
      for (each k-item set i) do
        for (each transaction T) do
          if (i does not belongs to T)
              return TID;
        end for
      end for
    end for
    for (each k-item set i) count number of TIDs
        //count gives infrequency
        if (count>= minimum negative count)
        return infrequent k-item set;
    end for
  end
```

The proposed system adapts Pittsburgh approach in order to encode rules within a population of individuals. In which each individual represents a rule set (chromosome = set of rules). The main advantage of the method is that the entire rule base is coded; therefore, it is not necessary to do the quantitative analysis of indispensable rules to see if the method functions properly, because, unlike the Michigan method, all possible rules take part in the working time of the genetic algorithm. Although the Pittsburgh approach involves more computational overhead, the competition among complete rule bases, as opposed to single-rules, induces better cooperation among the rules. In general, coding in the Pittsburgh method is binary coding, in which "1" means that a knowledge base rule will be in a knowledge base, whereas "0" means it will not be used. The binary code is adopted because it has advantages of better searching, simpler coding, and decoding, easier implementation of crossing and variation [9] [10].

F. Calculate the fitness function

Several criteria are used to quantify the quality, or fitness, of a chromosome. Some of these criteria are highly qualitative and, in some cases, subjective.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

However, in the context of generic search, we must formulate a single numerical quantity that encapsulates the desirable features. To develop a compact rule base from an initial population; the genetic algorithm selects chromosomes with high fitness values for mating. An evaluation function is a set of test objects including the instances and historical records, which is then used to qualify the derived rule set.

Ideally, the discovered rules should: (a) have a high predictive accuracy; (b) be comprehensible, and (c) be interesting [5] [11]. The fitness function should be customized to the specific search spaces; thus, choice of fitness function is very important to get the desired results. The population is ranked with the help of fitness function. The system applies the genetic algorithm on the selected population from the initial rules and computes the fitness function after each step until the genetic algorithm is terminated. The performance of the derived rule sets their fitness value is fed back to the genetic algorithm to continue. Now, how the solution space is searched to promote the quality rules. It is known that higher the values of N_a and N_d and lower the values of N_b and N_c, the better is the rule [7].

$$\text{Confidence Factor CF} = \{N_a / (N_a + N_c)\} \text{ Mod1} \quad (6)$$

The system also relies on another factor, completeness measure for computing the fitness function.

$$\text{Fitness} = (\text{CF} * \text{Comp}) \text{ Mod1} \quad (7)$$

$$\text{Comp} = \{N_a/(N_a + N_b)\} \text{ Mod1} \quad (8)$$

the fitness function shows that how much we near the generate the rule. In this fitness function, we are using Mod operation with 1 in order to ensure that it will not exceed the range of fitness function, which is [0….1]. GA then compares the individual fitness value of each child with the average fitness value and regenerate positive and negative association rules. The best members are copied to the new generation. After that the genetic operators find out the search capability and convergence of the algorithm.

G. Tabu Search for new members generation

In this step, Tabu search algorithm is utilized to generate new members in the new population. Tabu search uses a local or neighborhood search procedure to iteratively move from a solution *x* to a solution *x*' in the neighborhood of *x*, until some stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure, Tabu search modifies the neighborhood structure of each solution as the search progresses. The solutions admitted to $N^*(x)$, the new neighborhood, are determined through the use of memory structures. The search then progresses by iteratively moving from a solution *x* to a solution *x*' in $N^*(x)$ [13] [19].

This approach contains two basic mechanisms: *Tabu restrictions* and *aspiration criteria*. Mapping to our work, the former represents the best solutions found now have already been recorded in the Tabu list memory before. The latter is similar to the former one, but its solution quality is the best in all the iterations found so far. So, they can be put into the Tabu list memory again to test Tabu restrictions in the next iterations. Perhaps the most important type of memory structure used to determine the solutions admitted to $N^*(x)$ is the Tabu list. In its simplest form, a Tabu list is a short-term memory which contains the solutions that have been visited in the recent past (less than n iterations ago, where n is the number of previous solutions to be stored (n is also called the Tabu tenure). Tabu search excludes solutions in the Tabu list from $N^*(x)$. A variation of a Tabu list prohibits solutions that have certain attributes. Selected attributes in solutions recently visited are labeled "Tabu-active.". Solutions that contain Tabu-active elements are “Tabu”. This type of short-term memory is also called "recency-based" memory. Algorithm 2 show the pseudo code of the Tabu search

In this case, the system used the same representation as GA approach to encode the rules. The best of the new solution is then selected again to test by Tabu restrictions and aspiration criteria. They may be put into the Tabu list memory depending on whether they pass the former test. If they fail, but satisfy the latter test, they can also be put in the Tabu list memory. Herein, the length of the Tabu list is initially assigned according to the size of the problem and it will be decreased and increased during the construction of the solution so as to achieve better exploration of the search space. Furthermore, the simplest and most commonly used aspiration criterion consists of allowing a move, even if it is in Tabu and results in a solution with an objective value better than that of the current best-known solution. Also, the most commonly used stopping criterion in our case is after some number of iterations without an improvement in the objective function value [19].

The advantage of TS over other methods is to use its memorized ability to prevent from searching previous visited areas. Therefore, it is easier to escape from local optimum and approach the global or near global optimum in a short time. Here f_x is the evaluation function for rules that is the same as used in the GA procedure. Also, the rules are sorted according to their quality. In this way, we can always pick the best one first to test from the beginning of the list. Herein, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Tabu search choses the best neighborhood's based on the same fitness function as in Eq. (8).

K: = 1.
generate initial solution
WHILE the stopping condition is not met **DO**
Identify *N*(s). (Neighborhood set)
Identify *T* (s, k). (Tabu set)
Identify *A* (s, k). (Aspirant set)
Choose the best *s*' | f_x: *N* (s, k) = {*N*(s) - *T* (s, k)} + *A* (s, k). Memorize *s*' if it improves the previous best-known solution
s: = *s*'. *k*: = *k*+1.
END WHILE

H. Apply crossover operation

Given the optimized population generated from the Tabu search step (selection operator), the crossover operator is employed that involves the swapping of genetic material (bit-values) between the two parent strings. Two parents produce two offspring. There is a chance that the chromosomes of the two parents are copied unmodified as offspring. There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring. Generally, the chance of crossover is between 0.6 and 1.0 [19]. The new two offspring created from this mating are put into the next generation of the population. By recombining portions of good individuals, this process is likely to create even better individuals. Herein, the system chooses dynamic one-point crossover [15] as the crossing operation of GA. The process of crossing is shown as follows: Selected a chromosome from the population according to crossover probability PC, select a breakpoint in the chromosome randomly, exchange the right end of the breakpoints in parents to generate a new offspring.

I. Apply mutation operation

It may be possible that crossover operation may produce degenerate population. In order to undo this, mutation operation is performed. Mutation operation can be inversion, insertion, reciprocal exchange or others. In case of inversion two random points are selected and the string between them is reversed. In case of insertion, a node is inserted at random position in the string. In reciprocal exchange, nodes at two random positions are exchanged, this paper uses insertion. With some low probability Pm between 0.1 to 0.3, a portion of the new individuals will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence.

In GA, mutation operators are mostly used to provide exploration and crossover operators are widely used to lead the population to converge on one the good solutions find so far (exploitation). Consequently, while crossover tries to converge to a specific point in the landscape, the mutation does its best to avoid convergence and explore more areas.

J. Check the member feasibility

As mentioned before, the TS purpose of classifying certain moves as forbidden – i.e. "Tabu" is basically to prevent cycling. Moreover, TS permits backtracking to previous solutions, which may ultimately lead, via a different direction, to better solutions [14] [15] [18]. Herein, after reproduction operations, the members of the new population are tested based on its fitness function; so as to treat them as members of the current population. After that, the termination condition is checked. If the termination condition is reached, the process is stopped, if not the next step is pursued. There are several possible stopping conditions for the search. In our hybrid algorithm, the system stops the process if one of the following two conditions is satisfied: (1) The number of iterations performed since the best solution last changed is greater than a predefined maximum number of iterations, or (2) If the maximum allowable number of iterations (generations) is reached. After this process is completed, the current system extracts the optimal association rules for both positive and negative frequent patterns.

4. EXPERIMENTAL RESULTS & DISCUSSION

This section shows the performance of the suggested algorithm for mining both interesting positive and negative rules. Experiments are performed on a computer Intel ® Core™ i5-2450M CPU @ 2.50 GHz, running on a Windows 10, 64-bit operating system, x64-based processor and 6 GB of memory. All codes are implemented under the MATLAB version 7.8.0. Experiments were conducted on 3 benchmark datasets from UCI learning machine laboratory website (http://archive.ics.uci.edu/ml/index.php).

The proposed system was compared with base Apriori [20] and genetic-Apriori algorithms [5] for mining interesting positive and negative rules with different input parameter (support, confidence, itemset length). The results are representing in Table 3 to 5 where the number of interesting positive ($A \rightarrow B$) and negative rules are denoted as ($A \rightarrow \neg B$, $\neg A \rightarrow B$, $\neg A \rightarrow \neg B$). Note that since Apriori needs additional passes over the database to derive all required supports, obviously it is more time-consuming.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Total number of generated rules using Apriori algorithm with support (65%) confidence (55%) and item length 2.

Datasets	Apriori Algorithm				
		$A \rightarrow B$	$A \rightarrow \neg B$	$\neg A \rightarrow B$	$\neg A \rightarrow \neg B$
Heart Disease	Frequent	8	3	1	8
	In-Frequent	0	16	20	14
Breast Cancer	Frequent	33	0	0	42
	In-Frequent	0	17	17	23
Iris	Frequent	7	0	0	8
	In-Frequent	0	12	12	16

Table 4: Total number of generated rules using Genetic-Tabu algorithm with support (65%) confidence (55%) and item length 2.

Datasets	Apriori Algorithm with GA				
		$A \rightarrow B$	$A \rightarrow \neg B$	$\neg A \rightarrow B$	$\neg A \rightarrow \neg B$
Heart Disease	Frequent	3	0	0	2
	In-Frequent	0	7	9	10
Breast Cancer	Frequent	12	0	0	16
	In-Frequent	0	6	6	8
Iris	Frequent	2	0	0	3
	In-Frequent	0	5	5	8

Table 5: Total number of generated rules using Genetic-Tabu algorithm with support (65%) confidence (55%) and item length 2.

Datasets	Apriori Algorithm with GTA				
		$A \rightarrow B$	$A \rightarrow \neg B$	$\neg A \rightarrow B$	$\neg A \rightarrow \neg B$
Heart Disease	Frequent	3	0	0	2
	In-Frequent	0	6	7	7
Breast Cancer	Frequent	10	0	0	15
	In-Frequent	0	5	5	7
Iris	Frequent	2	0	0	3
	In-Frequent	0	4	3	6

From the tables, it can be concluded that the proposed system mines the least number of positive and negative association rules that achieve both of optimal support and confidence according to the fitness function formula. The role of the Tabu in reducing the rules search spaces inside GA is clearly shown. With the increase in the number of database transactions the proposed system reduces the number of generated rules clearly ranging from 10 - 20% reduction from the instance of which depends on the GA algorithm only.

Note that the increase in support values with the damping of the confidence value with decrease the number of extracted rules for both positive and negative itemsets by a large percentage. In similar, the increase in the values of confidence with stability of support value will as decrease the number of extracted rules but in this time with small percentage. From this it can be concluded that the support has a stronger effect to reduce the number of mined rules.

Table 6: The relation between support, number of rules and execution time for Chase UCI dataset

Support	2	4	6	7	8	9	10
Confidence	.1	.2	.3	.4	.5	.6	.7
Execution Time	2.3	2.2	1.6	1.6	1.1	1.0	0.25
No. of Rules	1423	1423	850	850	850	850	301

The second experiment investigates the relationship between minimum support and the time required by the proposed system to extract all the rules, whether positive or negative. It can be observer from Table 6 that by increasing minimum support, the number of rules decreases and thus the time required to extract those decreases. These findings correspond to the other methods found in the surveys. We note that the proposed system needs more time (computational cost) to perform the Tabu search operations, which contribute to reduce the search space. This leads to increase the efficiency of the proposed system for the extraction of both positive and negative rules.

The third experiment investigates the extent to which the proposed system is affected by some important variables within GA such as crossover rate and mutation rate. First of all, the increase in the ratio of both crossover rate and mutation rate increases the time taken by GA to extract the rules. We compared different mutation rates from 0% to 20% on Heart Disease and Breast Cancer. The results show that the mutation rate will not have an outstanding effect, but if it is set to 0%, it will result in missing a lot of interesting rules. A Mutation rate of 5-10% is a good choice because it can produce around 80% rules for Heart Disease and above 90% rules for Breast Cancer. When the mutation rate is 5%, the average runtime per rule extraction is lower. We, therefore, set a mutation rate of 5% for the following experiments.

In a similar way, we compared different crossover rates from 60% to 100% on both Heart Disease and Breast Cancer datasets. With low crossover rates, such as 60%, we obtained almost the same rules as with high crossover rates. So, 60% is the best choice for the two datasets in our experiments. For the rest of the variables within GA such as a number of generations and population size. The effect of these variables may be absent in the case of Tabu search algorithm, which is used to improve the selection of population within

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

each iteration. It is clear from these experiments that the suggested system has good scalability for the different parameters, and that the efficiency gain thanks to the proposed optimizations is significant.

5. CONCLUSION

A considerable body of work has been carried out on the problem of positive association rule mining, but negative association rule mining has received very little attention. In this paper, we proposed an efficient method of mining generalized positive and negative association rules. This paper proposed a novel method for optimization of interesting positive and negative association rule. The defined algorithm is a combination of A priori algorithm and integrated genetic- Tabu algorithm. Instead of mining negative association rules with an intuitive method, the suggested system uses negative interestingness to characterize the property of negative association rules and justify the effectiveness.

With integrated genetic-Tabu search algorithm, the system reduces the search space of the mining process and a useful representation of generalized association rules is obtained that exploits the upward closure property of association rules. Compared to a naive implementation, the suggested system is very efficient in finding all positive and negative ARs. The simulation results show that the expected goals are achieved because the rules selected by the proposed method having much better support and confidence values with a reduction in processing time. In future, we wish to conduct experiments on large real datasets and compare the performance of our algorithm with other related algorithms. Furthermore, we plan to incorporate other useful factors into our implementation inside the fitness function in order to improve the quality and usefulness of the mined negative association rules and to further reduce the computational workload.

REFERENCES

[1] S. Jain, and R. G. Vishwakarma, "Generating Positive & Negative Rules using Efficient Apriori Algorithm", Proceedings of the Third International Road Federation (IRF) Conference, pp. 34-38, India, 2015.

[2] C. Kaur, "Association Rule Mining using Apriori Algorithm: A Survey", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 2, No. 6, pp. 2081-2084, 2013.

[3] R. Dewang, and J. Agarwal, "A New Method for Generating All Positive and Negative Association Rules", International Journal on Computer Science and Engineering, Vol. 3, No. 4, pp. 1649-1657, 2011.

[4] R. Saxena, S. Shrivastava, and A. Mathur, "Association Rules Mining using Modified Genetic Algorithm", International Journal of Scientific Engineering and Technology, Vol. 1, Issue 4, pp. 35-38, 2012.

[5] N. Rai, S. Jain, and A. Jain, "Mining Interesting Positive and Negative Association Rule based on Improved Genetic Algorithm (MIPNAR_GA)", International Journal of Advanced Computer Science and Applications, Vol. 5, No. 1, pp. 160-165, 2014.

[6] L. Antonie, J. Zaiane, and O. Zaiane, "Frequent Pattern Mining", Negative Association Rules, Chapter 6, pp. 135-145, Springer International Publishing Switzerland, 2014.

[7] Shivakeshi, H. M. Mahantesh, B. N. Kumar, and B. M. Pampapathi, "An Optimized Method for Generating All Positive and Negative Association Rules", International Journal of Computer Science Trends and Technology, Vol. 4, Issue 3, pp. 117-123, 2016.

[8] L. Tsaim S. Lin, and D. Yang, "Efficient Mining of Generalized Negative Association Rules", International Conference on Granular Computing, pp. 471-476, USA, 2010.

[9] A. Sharma, and N. Tivari, "A Survey of Association Rule Mining Using Genetic Algorithm", International Journal of Computer Applications & Information Technology, Vol. 1, Issue 2, pp. 5-11, 2012.

[10] N. Jain, V. Sharma and M. Malviya, "Reduction of Negative and Positive Association Rule Mining and Maintain Superiority of Rule using Modified Genetic Algorithm", International Journal of Advanced Computer Research, Vol. 2, pp. 31-36, 2012.

[11] B. Alatas and E. Akin, "An Efficient Genetic Algorithm for Automated Mining of both Positive and Negative Quantitative Association Rules", Soft Computing, Vol. 10, Issue 3, pp. 230-237, 2006.

[12] N. Suryawanshi, S. Jain and A. Jain, "A Review of Negative and Positive Association rule mining with multiple constraint and correlation factor", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, pp. 778-781, 2012.

[13] R. Thamilselvan and P. Balasubramanie, "A Genetic Algorithm with a Tabu Search (GTA) for Traveling Salesman Problem", International Journal of Recent Trends in Engineering, Vol. 1, pp. 607-610, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] R. Thamilselvan, and P. Balasubramanie, "Integrating Genetic Algorithm with a Tabu Search (GTA) for Network Traffic Scheduling", International Journal of Computer Applications, Vol. 72, pp. 21-24, 2013.

[15] W. Qing-rong, Z. Chang-sheng, H. Hong-yong, Y. Yuan, and W. Qing-rong, "Improved Genetic-Tabu search Algorithm in Bus Scheduling Study", International Conference on Computer and Electrical Engineering, pp. 1-6, China, 2012.

[16] R. Sumalatha, and B. Ramasubbareddy, "Mining Positive and Negative Association Rules", International Journal on Computer Science and Engineering, Vol. 2, No. 9, pp. 2916-2920, 2010.

[17] R. V. Prakash, Govardhan and S.S.V.N. Sarma, "Mining Frequent Itemsets from Large Data Sets using Genetic Algorithms", Artificial Intelligence Techniques Novel Approaches & Practical Applications, pp.38-43, 2011.

[18] V. R. Babu and A. R. Samanthula, "Tenet Production with Apriori and Genetic Algorithms", International Research Journal of Engineering and Technology (IRJET), Vol. 2, Issue 5, pp. 1262-1269, 2015.

[19] R. Thamilselvan and P. Balasubramanie, "Integration of Genetic or Job Shop Scheduling with Unordered Subsequence Exchange Crossover Algorithm", Journal of Computer Science, Vol. 8, pp. 681-693, 2012.

[20] S. Kamepalli, R. Kurra, and S. Krishna, "Apriori Based: Mining Infrequent and Non-Present Item Sets from Transactional Data Bases", International Journal of Electrical & Computer Science, Vol. 14, No. 23, pp. 31-35, 2014.