International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

Mining & Optimization of Association Rules Using Effective Algorithm

Sanat Jain[1], Swati Kabra[2]

[1]M.Tech Scholar, [2]Asst. Professor, CS Dept.Medicaps Institute of Technology & Management, Indore, RGPV University M.P.

[1]jain.sanat@gmail.com
[2]swativkkabra@gmail.com

***Abstract*— Association Rule Mining is generally performed in Generation of frequent item sets & Rule generation. Mining association rules is not full of reward until it can be utilized to improve decision-making process of an organization. This paper is concerned with discovering positive and negative association rules. We present an Apriori-based algorithm that is able to find all valid positive and negative association rules in a support confidence framework. The algorithm can find all valid association rules quickly and overcome some limitations of the previous mining methods. The complexity and large size of rules generated after mining have motivated researchers and practitioners to optimize the rule, for analysis purpose. This optimization can be done using Genetic Algorithm.**

***Keywords*—association rule, frequent item set, correlation coefficient, optimization, genetic algorithm.**

I. INTRODUCTION

Data Mining is one of the fastest growing research areas for Knowledge discovery. In Knowledge discovery Association Rule Mining plays a vital role. Association Rules Mining introduced by R. Agrawal [1] is an important research topic among the various data mining problems. Association rules have been extensively studied in the literature for their usefulness in many application domains such as market basket analysis, recommender systems, and diagnosis decisions support, telecommunication, intrusion detection, and etc.

Mining Association rules is not full of reward until it can be utilized to improve decision-making process of an organization. When mining association rules, we adopt another minimum support threshold to mine frequent item sets. With a correlation coefficient measure and pruning strategies, the algorithm can find all valid association rules quickly and overcome some limitations of the previous mining methods. The complexity and large size of rules generated after mining have motivated researchers and practitioners to optimize the rule, for analysis purpose. All the traditional association rule mining algorithms were developed to find positive associations between item sets.

Several algorithms have been developed to cope with the popular and computationally expensive task of association rule mining. With the increasing use and development of data mining techniques and tools, much work has recently focused on finding negative patterns, which can provide valuable information. However, mining negative association rules is a difficult task, due to the fact that there are essential differences between positive and negative association rule mining.

II. CONCEPTS

A. Definitions

Let I= {i1, i2… in} be a set of n distinct literals called items. Let Database be a set of transactions, where each transaction T is a set of items, and each transaction is associated with a unique identifier called TID. Let A, called an item set, be a set of items in I. The number of items in an item set is the length (or the size) of an item set. Item sets of length k are referred to as k item sets. A transaction T is said to contain an if A ⊂ T. An association rule is an implication of the form A⇒ B, where A⊂I, B⊂I, and A∩B=φ. We call A the antecedent of the rule, and B the consequent of the rule. The rule A⇒ B has a support (denoted as supp) s in DB if s% of the transactions in DB contains A∪B. In other words, the support of the rule is the probability that A and B hold together among all the possible presented cases. I.e. supp (A⇒ B) =supp (A∪B) =P (A∪B). (1)The rule A ⇒ B has a measure of its strength called confidence (denoted as conf) c if c% of transactions in DB that contain A also contain B. In other words, the confidence of the rule is the conditional probability that the consequent B is true under the condition of the antecedent A. i.e. conf (A⇒ B) = P (B|A) =supp (A∪B)/supp (A) [13].

B. Support and Confidence

Support count: The support count of an item set X, denoted by X. count, in a data set T is the number of transactions in T that contain X. Assume T has n transactions.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

Then,

$$\text{Support} = \frac{(X \cup Y).\text{count}}{n}$$

$$\text{Confidence} = \frac{(X \cup Y).\text{count}}{X.\text{count}}$$

C. Negative Association Rules

The negation of an item set A is indicated by ¬A, which means the absence of the item set A. We call a rule of the form A⇒ B a positive association rule, and rules of the other forms (A⇒ ¬B, ¬A⇒ B and ¬A⇒ ¬B) negative association rules. The support and confidence of the negative association rules can make use of those of the positive association rules [2].

D. Horizontal database and Vertical database

A set of transactions in TID-item set format (that is, {TID : item set }) ,where TID is a transaction–id and item set is the set of items bought is transaction TID .This data format is known as horizontal data format**.** Alternatively data format can also be presented in item-TID_set format (that is {item: TID_set}), where item is an item name, and TID_set is the set of transaction identifiers containing the item. This is an item name, and TID_set is the set of transaction identifiers containing the item. This format is known as vertical data format [3].

TABLE I: HORIZONTAL DATABASE

Transaction ID or TID	Item sets
1	a, c, d, e, f
2	a, b, e
3	c, e, f
4	a, c, d, f
5	c, e, f

TABLE II: VERTICAL DATABASE

Items	TID of Transactions containing the Item or TID sets.
a	1, 2, 4
b	2
c	1, 3, 4, 5
d	1, 4
e	1, 2, 3, 5
f	1, 3, 4, 5

Association rule can be generated by using vertical data format also. It become faster to generate the rules and easy to generate the rules. In this dissertation work we are using vertical data format to generate the Association rules.

E. Advantage of using Vertical data format in Association rules

Firstly, computing the support with the vertical layout since it involves only the intersection of the TID sets.
Secondly, there is an automatic "reduction" of database before each scan in that only those item sets that are relevant to the following scan of the mining process are accessed from disk Finally the vertical layout is more versatile in supporting various search strategies.

III. METHODOLOGY

In this dissertation work, we tried to generate Association rules using vertical data format and after the rules are generated we tried to reduce the rules on the bases of support value and confidence value.

Generate Association Rules using Vertical Data Format:

The Association rule can be generated by using vertical data format. The steps for generation of association rules are convert the horizontal data base into vertical database. Take the intersection of item sets and union of respective transaction id. Repeat the steps till all frequent items are generated. After the frequent items are found out, generate the association rules. In this paper, we have generated positive and negative association rules. The figure shows all types of association rules

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

TABLE III: ASSOCIATION RULES

	B	~ B
A	TP	FP
~ A	FN	TN

The symbol ~ shows negation or we can say that the absence of the item sets in the transactions.

Optimization of Association Rules Using GA:

In this section describes the GA algorithm for optimization of association rule associated .First, explanation of how GA algorithm represents the rule individually and encodes scheme and the chromosome structure (Representation of rule) shown. After that, description of genetic operators and fitness function assignment and selection criteria are listed. Finally, the algorithmic structure is given [12].

A. Representation of Individually in Rule and Encoding Scheme

Representation of generated rule in GA is play very important role. Mainly two Methods are mostly based on how rules are encoded in the population of individuals ("Chromosomes") as discussed in [6] Michigan and Pittsburgh, In the Michigan Approach each individual encodes a *single* prediction rule, whereas in the Pittsburgh approach each individual encodes *a set of* prediction rules. In this paper we are only interested to generate single rule so, here we are using Michigan approach. GA use various encoding scheme like tree encoding, permutation encoding, binary encoding etc., here we adopt binary encoding.
Consider following example,

If paper and pencil then eraser not Ink

Now, following Michigan's approach and binary encoding, for simplicity usage, this rule can be represented as **001** 111 **010** 111 0**11** 111 **100** 000 where, the bold tri-digits are used as attribute id, like **001** for paper, **010** for pencil, **011** for eraser and **100** the normal tri-digits are 000 or 111 which shows absence or presence respectively. Now this rule is ready for further computations.

B. Chromosome Structure (Representation of Attribute of Dataset)

GA algorithms are a fixed length chromosome structure. Here we are using three bit binary encoding for representation table IV show the attribute representation and table V show the Presence and absence of rule, in this paper we are only interested to take 6 attribute like for example, A,B,C,D,E, and F.

TABLE IV: REPRESENTATION OF ATTRIBUTE IN BINARY ENCODING

A	B	C	D	E	F
100	010	011	100	101	110

TABLE V: PRESENCE & ABSENSE OF ATTRIBUTE

Presence of Attribute	Absence of Attribute
111	000

C. Genetic operator

Genetic Algorithm uses genetic operators 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.GA algorithm used single-point crossover operation with probability 0.1.

Chromosomes can be created as in Fig.1

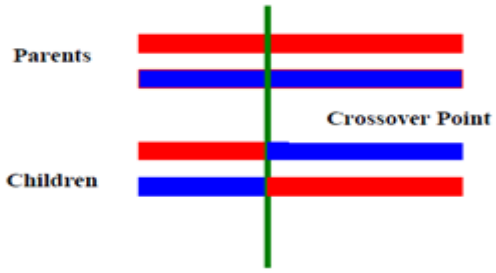

Figure 1: Single Point Crossover

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.

D. Fitness Function

Ideally the discovered rules should: (a) have a high predictive accuracy; (b) be comprehensible; and (c) be interesting.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

The fitness function should be customized to the specific search spaces, thus choice of Fitness function [6] is very important to get the desired results. The population is ranked with the help of fitness function. We apply genetic algorithm on the selected population from the database and compute the fitness function after each step until the genetic algorithm is terminated. Rules generally define as [5]:

IF A THEN B

Where A is the antecedent and C is the consequent. The rules performance can be shown in table VI by a 2×2 matrix, which is called confusion matrix.

TABLE VI: CONFUSION MATRIX FOR A RULE

Predicted/actual class	Item set A	Not Item set A
Item set B	TP	FP
Not item set B	FN	TN

It is known that higher the values of TP and TN and lower the values of FP and FN, the better is the rule [2].

Confidence Factor, CF = {TP / (TP + FN)} Mod1

We also introduce another factor completeness measure for computing the fitness function.

Comp= {TP/ (TP+FP)} Mod1

Fitness= (CF *Comp) Mod1

In this fitness function we are using Mod operation with 1 in order to insure that it will not exceed the range of fitness function, which is [0...1]. The fitness function shows that how much we near to generate the rule.

E. Algorithm Structure

In this paper we are presenting flow chart of algorithm structure [12]. The genetic algorithm is applied over the rules fetched from Apriori association Rule mining. The proposed method for generating optimized association rule by genetic algorithm is as follows:

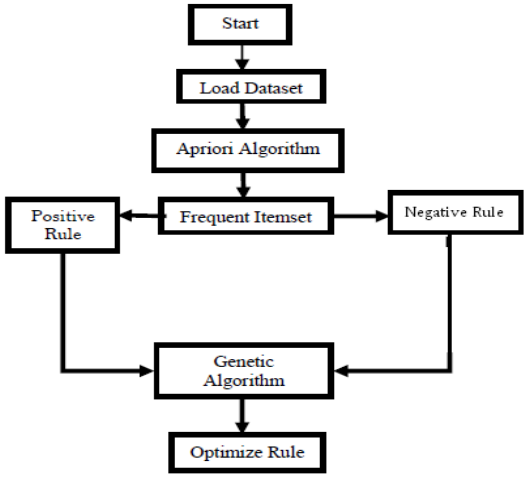

Figure 2: Algorithm Flow Chart

1. Start
2. Load a sample of records from the database that fits into memory.
3. Apply Apriori algorithm to find the frequent item sets with the minimum support. Suppose S is set of the frequent item set generated by Apriori algorithm.
4. Set Q=Ǿ where Q is the output set, which contains the all association rule.
5. Set the Input termination condition of genetic algorithm.
6. Represent each frequent item set of S as binary encoding.
7. Select the two members (string) from the frequent item set.
8. Apply GA operators, crossover and mutation on the selected members (string) to generate the association rules.
9. Find the fitness function for x=>y each rule.
10. If (fitness function > min confidence) then
11. Set Q = Q U {x =>y}
12. If the desired number of generations is not completed, then go to Step 3.
13. Stop

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 4, April 2012)

IV. EXPERIMENTAL RESULT

For the convenience of comparison, we conducted our experiments on the synthetic dataset to study the behaviours of the algorithm.

Example: Let us consider a small transactional table with 10 transactions and 8 items. In Table VII a small transactional database is given.

TABLE VII: A TRANSACTION DATABASE TD

TID	ITEM SET	TID	ITEM SET
1	Tea	5	Apple
2	Mango	6	Milk
3	Sugar	7	Bread
4	Oranges	8	Jam

Graph shows the comparision between non optimized and optimized methods.

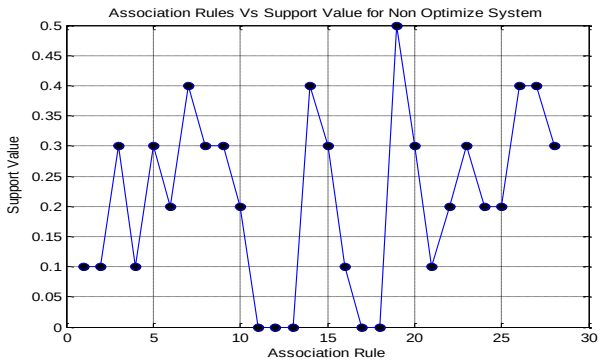

Figure 3: Association Rule For Non Optimize System

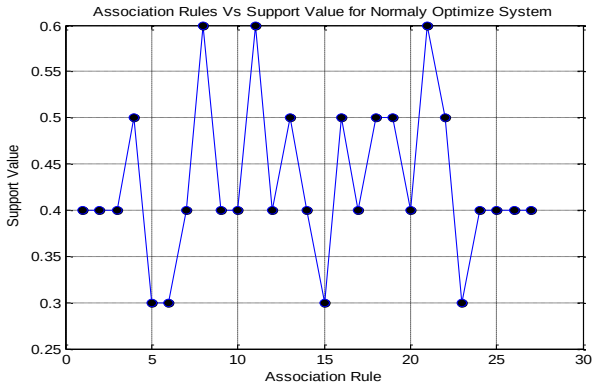

Figure 4: Association Rule For Optimize System

V. CONCLUSION

In this paper, we have designed a new algorithm for efficiently mining positive and negative association rules in databases and optimization of positive and negative association rule using genetic algorithm. Our approach is novel and different from existing research. We have designed pruning strategies for reducing the search space and improving the usability of mining rules, and have used the correlation coefficient to judge which form association rule should be mined. It is shown by empirical studies that the proposed approach is effective, efficient and promising.

REFERENCES

[1] R. Agrawal, T. IMIELINSKI, and A. SWAMI, "Mining association rules between sets of items in massive databases," In Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, ACM, Washington D.C., 1993, pp. 207-216.

[2] Alex A. Freitas, "Understanding the crucial differences between classification and discovery of association rules - a position paper" *ACM SIGKDD Explorations*, 2(1):65-69, 2000.

[3] A. Savasere, E. Omiecinski, and S. Navathe, "Mining for strong negative associations in a large database of customer transactions," In Proc. of ICDE, 1998, pp. 494-502.

[4] Das Sufal and Saha Banani" Data Quality Mining using Genetic Algorithm" International Journal of Computer Science and Security, (IJCSS) Volume (3): Issue (2)

[5] Anandhavalli M." Optimized association rule mining using genetic algorithm" Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp-01-04

[6] Manish Saggar and Agarwal Ashish Kumar "Optimization of Association Rule Mining using Improved Genetic Algorithms" 2004 IEEE Computer Society Press.

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

[8] Wook J. and Woo S... *New Encoding/Converting Methods of Binary GA/Real-Coded GA.* IEICE Trans, 2005 Vol.E88-A, No.6, 1545-1564.

[9] W. Teng, M. Hsieh, and M. Chen, "On the mining of substitution rules for statistically dependent items," In Proc. of ICDM, 2002, pp.442-449.

[10] X. Dong, S. Wang, H. Song, and Y. Lu, "Study on Negative Association Rules," Transactions of Beijing Institute of Technology, Vol. 24, No. 11, 2004, pp. 978-981.

[11] X. Wu, C. Zhang, and S. Zhang, "Efficient Mining of Both Positive and Negative Association Rules," ACM Transactions on Information Systems, Vol. 22, No. 3, 2004, pp. 381–405.

[12] Rupesh Dewang et al." A New Method for Generating All Positive and Negative Association Rules" International Journal on Computer Science and Engineering (IJCSE), ISSN: 0975-3397 Vol. 3 No. 4 Apr 2011.

[13] Honglei Zhu, Zhigang Xu, "An Effective Algorithm for Mining Positive and Negative Association Rules" International Conference on Computer Science and Software Engineering 2004 IEEE Computer Society Press.