A New Method for Generating All Positive and Negative Association Rules

Rupesh Dewang
School of Information Technology,
Rajiv Gandhi Proudyogiki Vishwavidyalaya,
Bhopal, Madhya Pradesh, India

Jitendra Agarwal
School of Information Technology,
Rajiv Gandhi Proudyogiki Vishwavidyalaya,
Bhopal, Madhya Pradesh, India

Abstract— Association Rule play very important role in recent scenario of data mining. But we have only generated positive rule, negative rule also useful in today data mining task. In this paper we are proposing "A new method for generating all positive and negative Association Rules" (NRGA).NRGA generates all association rules which are hidden when we have applied Apriori Algorithm. For representation of Negative Rules we are giving new name of this rules as like: CNR, ANR, and ACNR. In this paper we are also modify Correlation coefficient (CRC) equation, so all generate results are very promising. First we apply Apriori Algorithm for frequent itemset generation and that is also generate positive rules, after on frequent itemset we apply NRGA algorithm for all negative rules generation and optimize generated rules using Genetic Algorithm

Keywords- **Association Rule, Data Mining, Genetic Algorithm, Negative Rules Generating Algorithm** (NRGA).

I. INTRODUCTION

Data mining is the task to mining the useful meaningful information from data warehouse. It is the source of in-explicit, purely valid, and potentially useful and important knowledge from large volumes of natural data [7]. The selected knowledge must be not only precise but also readable, comprehensible and ease of understanding. There are a many of data mining task such as ARs, sequential patterns, Classification, clustering, time series, etc., and there have been lots of techniques and algorithms for these tasks and different types of data in data mining. When the data consist continuous values, it becomes hard to mine the data and some special techniques need to be prepared. Association rule basically use for finding out the useful patterns, relation between items found in the database of transactions [2]. For example, consider the sales database of a Music CD store, where the records represent customers and the attributes represent Music CD. The mined patterns are the set of Music CDs most frequently bought together by the customer. An example could be that, 70% of the people who buy old song cds also buy guzzle cds. The store can use this information for future sales, self restore of records etc. There are many application areas for association rule mining techniques, which include catalog design, store layout, customer segmentation, and telecommunication alarm diagnosis and so on.

Most of the research has only point out for positive association rule but negative association rule also play very important role in Data mining task. But, mining negative association rules is a difficult task, due to the fact that there are prerequisite differences between positive and negative association rule mining. We will make attention on two key problems in negative association rule mining:

(1) How to effectively finding out for negative frequent itemsets.
(2) How to effectively locate negative association rules.

Although importance of negative associations, only some of researchers ([3], [9], etc.) proposed an algorithm to mine these types of associations rules. In this research paper we are proposed and a New Method (NARG) for finding out (extract) all negative association rule and optimize these generated rule using genetic algorithm.

II. ASSOCIATION RULE

A-Apriori Algorithm

Apriori [5][11] is the most popular and effective algorithm to find all the frequent itemsets in dataset. It is proposed by Agrawal and Srikant in 1994. Let I = {I1, I2……, Ik} be a set of k distinct attributes, also called literals. Ai = s is an item, where s is a domain value is attributing, Ai in a relation, R (A1… An). A is an itemset if it is a subset of I. DT =. {ti, ti+1… tn) is a set of transactions, called the transaction (tid, itemset).

A transaction t contains an itemset A if and only if, for all items IA, i is in t-itemset. An itemset A in a transaction database DT has a support, denoted as Supp (A) (we also use p(A) to stand for Supp (A)), that is the ratio of transactions in DT contain A. Supp (A) = |A(t)| / |DT|, Where A(t) = {t in DT/t contains A}. An itemset A in a transaction database DT is called a large (frequent) itemset if its support is equal to, or greater than, a threshold of minimal support (minsupp), which is given by users or experts. An association rule is an expression of the form IF A THEN B (or A -> B), A∩B = Ø, where A and B are sets of items. The meaning of this expression is that transactions of the databases, which contain A, tend to contain B. Each association rule has two quality measurements: support and confidence, defined as:

(1) The support of a rule A->B is the support of A ∪ B, where A ∪ B means both A and B occur at the same time in same transaction.
(2) The confidence or predictive accuracy [2] of a rule A->B is conf (A->B) as the ratio:| (A ∪ B)(t)| / |A(t) or Supp(A ∪ B) / Supp(A).

That is, support = frequencies of occurring patterns; confidence = strength of implication. Support-confidence framework [5][11]: Let I be the set of items in database D, A, B I be itemset, A ∩B = f, p(A) is not zero and p(B) is not zero. Minimal support minsupp) and minimal confidence (minconf) are given by users or experts. Then A->B is a valid rule if

1. Supp (A ∪ B) is greater or equal to minsupp,
2. Conf (A->B) is greater or equal to minconf.

Mining association rules can be broken down into the following two sub-problems[5]:

1. Generating all itemsets that have support greater than, or equal to, the user specified Minimal support. That is, generating all large itemsets.
2. Generating all the rules that have minimum confidence.

B-Negative Association Rules

The negation of an itemset A is represented by ¬A, which means the absence of the itemset 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 [10].In this paper work we have create a meaning for these type rule like:

Positive Rule (PR) = A=> B
Consequent Negative Rule (CNR) = A=> ¬B
Antecedent Negative Rule (ANR) = ¬A=>B
Antecedent and Consequent Negative (ACNR) = ¬A=> ¬B

The support and Confidence for CNR, ANR and ACNR rule is given by the following formulas:

The support and Confidence for CNR, ANR.and ACNR rule is given by the following formulas:

I- Consequent Negative Rule (CNR):

$$\text{Supp}(A \Rightarrow \neg B) = \text{supp}(A) - \text{supp}(A \cup B) \quad (1)$$

$$\text{Conf}(A \Rightarrow \neg B) = \frac{\text{supp}(A) - \text{supp}(A \cup B)}{\text{Supp}(A)} \quad (2)$$

II- Antecedent Negative Rule (ANR):

$$\text{Supp}(\neg A \Rightarrow B) = \text{supp}(B) - \text{supp}(A \cup B) \quad (3)$$

$$\text{Conf}(\neg A \Rightarrow B) = \frac{\text{supp}(B) - \text{supp}(A \cup B)}{1 - \text{supp}(A)} \quad (4)$$

III- Antecedent and Consequent Negative (ACNR):

$$\text{Supp}(\neg A \Rightarrow \neg B) = 1\text{-supp}(A)\text{-supp}(B) + \text{supp}(A \cup B) \quad (5)$$

$$\text{Conf}(\neg A \Rightarrow \neg B) = \frac{1\text{-supp}(A)\text{-supp}(B) + \text{supp}(A \cup B)}{1\text{-supp}(A)} \quad (6)$$

The negative association rules discovery seeks rules of the three forms with their support and confidence greater than, lesser then or equal to, user-specified minsupp and minconf thresholds respectively. These rules are referred to as an interesting negative association rule. The algorithm uses the correlation coefficient (CRC) between itemsets to find negative association rules. The correlation coefficient (CRC) between itemsets can be defined as:

$$CRC = \left\{ \frac{\text{Supp}(AUB)}{\text{Supp}(A)*\text{supp}(B)} \right\} \text{ Mod1} \quad (7)$$

A and B are itemsets.
When CRC (A, B) = 1, A and B are independent.
When CRC (A, ⌐B) <1, A and B have negative correlation.
When CRC (⌐A, B) <1, A and B have negative correlation.
By using Mod1 in eq.7, CRC values not exceed more then 1.it is providing benefit in negative rule generation.

III. NRGA(NEAGTIVE RULE GENRATING ALGORITHM)

NRGA mining is algorithm for generated all Association rule like CNR, ANR and ACN. Let DB is database that contain training dataset T, minsupp, minconf given by the user. Our algorithm for extracting both positive and negative association rules as follows:

Algorithm
Input: A training dataset T, minsupp, minconf.
Output: frequent itemsets (FI), CNR, ANR, ACNR.

1) Initialize FI=NULL and CNR=NULL, ANR=NULL, ACNR=NULL.
2) Generate frequent itemsets from T.
FI ἐ T
3) for (any frequent itemset A and ⌐B in FI)
3.1 Calculate supports value of A => ⌐B.
3.2 Calculate confidence value of A => ⌐B
3.3 if (supp (A=> ⌐B) ≥ minsupp and conf (A=> ⌐B) ≥ minconf)
if (CRC (A, ⌐B) <1)
{
CNR=CNR Ụ (A=> ⌐B).
}
4) for (any frequent itemset ⌐A and B in FI)
4.1 Calculate supports value of ⌐ A => B
4.2 Calculate confidence value of ⌐A => B
4.3 if (supp (⌐A=> B) ≥ minsupp and conf (⌐A=> B) ≥ minconf)
if (CRC (⌐A, B) <1)
{
ANR=ANR Ụ (⌐A=> B).
}
5) for (any frequent itemset ⌐A and ⌐B in FI)
5.1 Calculate supports value of ⌐ A => ⌐B
5.2 Calculate confidence value of ⌐A => ⌐B
5.3 **if** (supp (⌐A=> ⌐B) ≥ minsupp and conf (⌐A=> ⌐B) ≥ minconf)
if (CRC (⌐A, ⌐B) <1
{
ACNR=ACNR Ụ (⌐A=>⌐ B).
}
6) Return CNR, ANR, ACNR.

NRGA generate all positive and negative rules.

IV. GENETIC ALGORTIHM

Genetic Algorithm (GA) is general purpose search algorithm which use principles inspired by natural genetic populations to evolve solutions to problems [8]. All GAs typically starts from a set, called population, of random solutions (candidate). These solutions are evolved by the repeated selection and variations of more fit solutions, following the principle of survival of the fittest. The elements of the population are called individuals or chromosomes, which represent candidate solutions. Chromosomes are typically selected according to the quality of solutions they represent. To measure the quality of a solution, fitness function is assigned to each chromosome in the population. Hence, the better the fitness of a chromosome, the more possibility the chromosome has of being selected for reproduction and the more parts of its genetic material will be passed on to the next generations. Genetic Algorithms are very easy to develop and validate, which makes them highly attractive, if they applied. The algorithm is parallel; it can be applied to large populations efficiently, so if it begins with a poor original solution it can rapidly progress to good solutions. Use of mutation makes the method capable of identifying global optimal, even in very difficult problem domains. The technique does not need prior knowledge about the distribution of the data, this way Gas can efficiently explore the space of possible solutions. This space is called search space, and it contains all the possible solutions that can be encoded [4].

V. OPTIMIZATION OF ASSOCIATION RULE 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.

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 a fixed length chromosome structure. Here we are using three bit binary encoding for representation Fig.1 show the attribute representation and Fig.2 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.

A	B	C	D	E	F
001	010	011	100	101	110

Fig. 1. Representation of attribute in binary encoding.

Presence of Attribute	Absence of Attribute
111	000

Fig.2. Presence and Absence of attribute

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.3

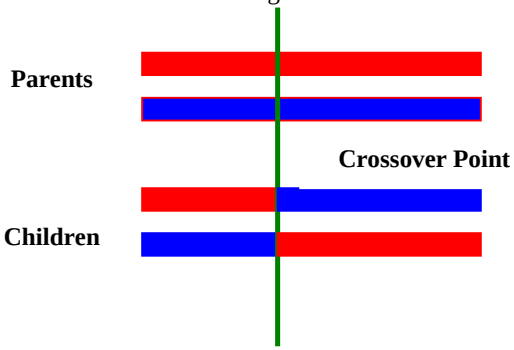


Fig.3.Single Point Crossover

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. 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:

IF A THEN B

Where A is the antecedent and C is the consequent. The rules performance can be shown in figure.4 by a 2×2 matrix, which is called confusion matrix.

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

Figure.4.Confusion matrix for a rule

It is known that higher the values of TP and TN and lower the values of FP and FN, the better is the rule. Confidence Factor, CF = {TP / (TP + FN)} Mod1

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

$$Comp= \{TP/ (TP+FP)\} \text{ Mod1}$$

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

The fitness function shows that how much we near to generate the rule. 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 and Methodology

Now we are presenting algorithm structure. In this paper the genetic algorithm is applied over the rules fetched from Apriori association Rule mining. The proposed method for generating association rule by genetic algorithm is as follows:

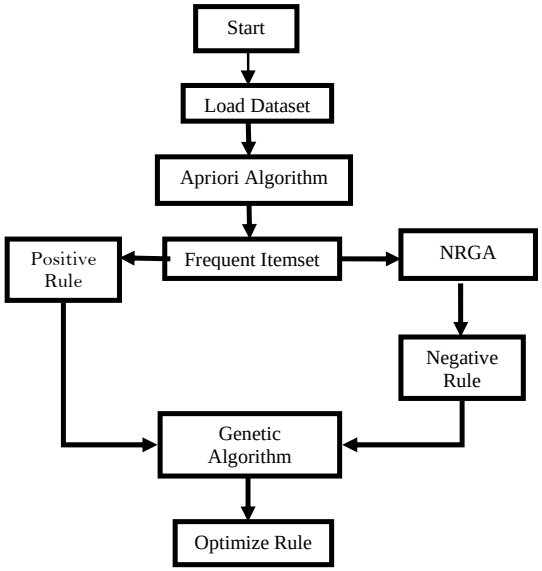


Fig.5. Algorithm flow Chart

1. Start

2. Load a sample of records from the database that fits into the memory.

3. Apply Apriori algorithm to find the frequent itemsets with the minimum support. Suppose S is set of the frequent item set generated by Apriori algorithm.

4. Apply NRGA (Negative rules generating Algorithm) for generation of all rules.

5. Set Q=Ǿ where Q is the output set, which contains the all association rule.

6. Set the Input termination condition of genetic algorithm.

7. Represent each frequent item set of S as binary encoding, string using the combination of representation specified in method above.

8. Select the two members (string) from the frequent item set using Roulette Wheel sampling method.

9. Apply GA operators, crossover and mutation on the selected members (string) to generate the association rules.

10. Find the fitness function for x=>y each rule.

11. If generated rule is better then previous rule then

12. Set Q = Q U {x =>y}

13. If the desired number of generations is not completed, then go to Step 3.

14. Stop.

VI. EXPERIMENT RESULTS

In this paper we are using Zoo database obtained from UCI machine learning repository. The data set has 101 Instances. It is consisting of 18 attributes (animal name, 15 Boolean attributes, and 2 numeric). Here we are using six attribute (Boolean) for our paper work. we only mined such the setting of parameters: The size of evolutionary population N=100, crossover rate=0.1 mutation rate=0.005.The experiment was executed on

Celeron(R) CPU 3.0GHz and 2.00 GB of main memory using Microsoft Windows XP operating system machine and software was java1.6.4, Net beans, Microsoft excel. Following table1 shows the rule generated after apply algorithm. In the following tables, *L*k is denoted as all frequent *k* itemset.In table1, table2 and table3 show rule generated by GA.

Table.1. L1 Rule generate by Genetic Algorithm

X1	=>	Y1	Supports	Confidence	Fitness
tail	=>	toothed	0.519608	0.697368	0.596138
backbone	=>	⌐ eggs	0.401961	0.488095	0.366865
milk	=>	toothed	0.401961	0.97619	0.645545
backbone	=>	toothed	0.607843	0.738095	0.738095
⌐ eggs	=>	toothed	0.205882	0.35	0.118548
backbone	=>	⌐ milk	0.411765	0.5	0.5
backbone	=>	tail	0.735294	0.892857	0.881109
milk	=>	tail	0.352941	0.857143	0.406015
⌐ eggs	=>	tail	0.352941	0.857143	0.350877

Table.2. L2 Rule generate by Genetic Algorithm

X1	X2	=>	Y	Support	Confidence	Fitness
backbone	milk	=>	toothed	0.401961	0.97619	0.645545
backbone	milk	=>	⌐ tail	0.392157	0.666667	0.406015
backbone	eggs	=>	tail	0.392157	0.930233	0.489596
milk	tail	=>	toothed	0.343137	0.972222	0.548835
backbone	eggs	=>	⌐toothed	0.215686	0.511628	0.165416
backbone	tail	=>	toothed	0.519608	0.706667	0.604086

Table.3. L3 Rule generate by Genetic Algorithm

X1	X2	X3	=>	Y	Supports	Confidence	Fitness
backbone	milk	tail	=>	toothed	0.343137	0.972222	0.548835

Figure.6 shows rule generate by apriori algorithm and there confidence value. Figure.7 shows Rules generated through NRGA method in which confidence value lie between 0 to 1,that shows some rules are very interesting for association rule mining, which not generated by apriori algorithm.

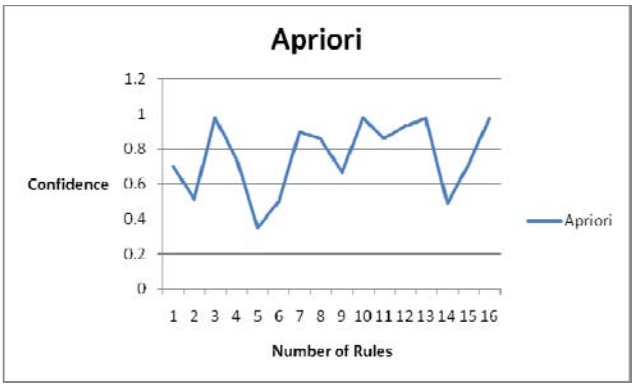


Figure.6 Rule generated through Apriori

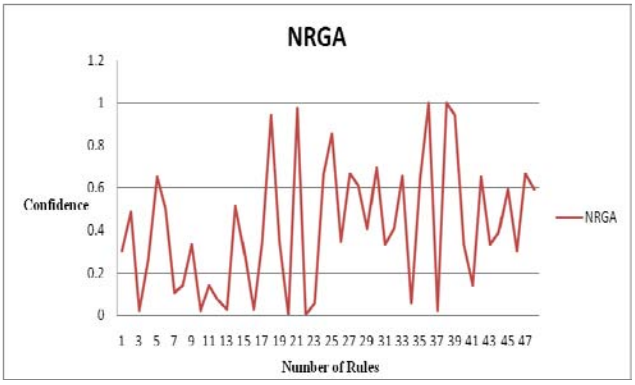
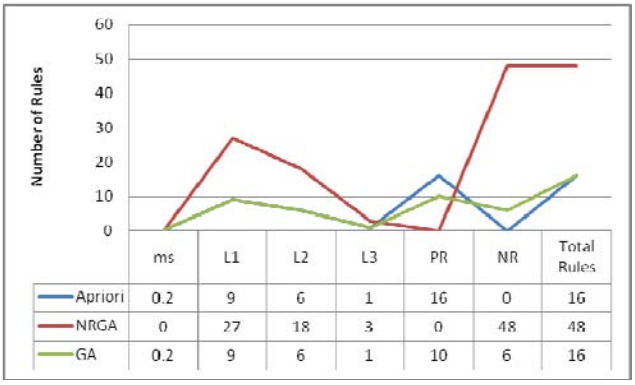


Figure.7 Rules generated by NRGA.

Figure,8 shows the comparison between numbers of rule generated by three algorithms in which Apriori Algorithm generate only Positive Rule(PR),NRGA algorithm generate all Negative Rules(NR).finally GA generate interesting PR and NR optimize Rules.



	ms	L1	L2	L3	PR	NR	Total Rules
Apriori	0.2	9	6	1	16	0	16
NRGA	0	27	18	3	0	48	48
GA	0.2	9	6	1	10	6	16

Figure.8 Comparison of Rules Generated by Three Algorithm.

VIII. CONCLUSIONS AND FUTIRE WORK

For resent scenario in market base analysis, Negative rules play very important role in decision making. We have 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. NRGA use modified CRC to generate all negative association rules. After all rule generation, GA are apply to optimize generate rule. The results reported in this paper are very promising since the discovered rules are of optimized rules. In future we

can use other technique to minimize the complexity of the genetic algorithm. We can also use other evolutionary algorithm like PSO and ACO for optimization of association rules.

IX. REFERENCES

[1] 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.

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

[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] Colombetti M. and Dorigo M... *"Training Agents to Perform Sequential Behavior"*., Italian National Research Council, 1993, pp: 93-023.

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

[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.