Optimization of Association Rule Mining using Improved Genetic Algorithms*

Manish Saggar
Undergraduate Student
4th Year, B.TECH
Indian Institute of Information Technology, Allahabad, India .
msaggar_01@iiita.ac.in

Ashish Kumar Agrawal
Undergraduate Student
4th Year, B.TECH
Indian Institute of Information Technology, Allahabad, India .
ashish_agr82@yahoo.co.in

Abhimanyu Lad
Undergraduate Student
4th Year, B.TECH
Indian Institute of Information Technology, Allahabad, India .
alad_01@iiita.ac.in

Abstract: *In this paper, the main area of concentration was to optimize the rules generated by Association Rule Mining (apriori method), using Genetic Algorithms. In general the rule generated by Association Rule Mining technique do not consider the negative occurrences of attributes in them, but by using Genetic Algorithms (GAs) over these rules the system can predict the rules which contains negative attributes. The main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. The improvements applied in GAs are definitely going to help the rule based systems used for classification as described in results and conclusions.*

Keywords: Genetic Algorithms, Data Mining, Association Rule Mining.

1 Introduction

In today's jargon, the amount of data stored in databases continues to grow very fast. This large amount of data contains latent knowledge, which can be utilized to improve decision making process of an organization. This knowledge discovery can be done in various ways available today, like Decision Tree, Association Rule Mining, Bayesian Classifier and so on. The form of this latent knowledge also varies to a large extent from different kind of rules to prediction values. In this paper the authors have considered Association Rule Mining and tried to improve this technique by applying Genetic Algorithms on the rules generated by Association Rule Mining.

A brief introduction about Association Rule Mining and GA is given in the following sub-sections, followed by methodology in section 2, which will describe the basic implementation details of Association Rule Mining and GAs. In section 3 the authors will discuss the results followed by conclusion in the last section.

1.1 Association Rules

Introduced in 1993 [5], association rule mining has gained great deal of attention. Even today people use it for mining in KDD. In brief, an association rule is an expression X=>Y, where X and Y are item sets.

The meaning of this kind of rule is : Given a database D containing say N tupples or transactions, where say T belongs to D is a transaction, then X=>Y expresses that whenever a transaction T contains X than T probably also contains Y. This probability or confidence is defined as the percentage of transactions containing Y in addition to X with regard to overall number of transactions containing X. Thus the authors can represent can represent this probability as conditional probability p(Y∈T|X∈T). The thrust behind introduction of these rules was there similarity with market-based data where rules like "A customer buys milk and Bread will also buy butter with a probability, say x %" is a famous example. Also, their direct applicability to business problems together with their inherent understandability, even for non-experts, made them a popular mining method. Further, it was also determined that their applications can be further extended from general dependency based rules to a wide range of business applications.

Mining Association rules is not full of advantages; it has some limitations too, first of all the algorithmic complexity. The number of rules grows exponentially with the number of items. But this complexity is tackled with some latest algorithms which can efficiently prune the search space. Secondly, the problem of finding rules from rules, i.e. picking interesting rules from set of rules. The

work tackling the second problem mainly support the user when browsing the rule set, e.g. [4] and the development of further useful quality measures on the rules, e.g. [2;6;7]. Thirdly, the problem that is being discussed in this paper is that, association rules do not utter the rules in which the negation of attributes is there. Like, say there are three attributes in the database X1, X2, X3, than rules like "If a customer takes X1 and not X2 than he will take X3 with a confidence of say c %" will not be provided by normal association rule mining. In order to generate these kinds of rules and also to tackle the second problem discussed above, i.e. to evolve quality rules, this paper is using Genetic Algorithms.

1.2 Genetic Algorithms

As discussed in [1], in general the main motivation for using GAs in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining. This section of the paper discusses several aspects of GAs for rule discovery. The main areas of discussion include individual representation of rules, Genetic Operators involved and the choice of Fitness function.

Representation of rules plays a major role in GAs, broadly there are two approaches based on how rules are encoded in the population of individuals ("Chromosomes") as discussed in [1] Michigan and Pittsburgh; The pros and cons as discussed in [1] is as follows, Pittsburgh approach leads to syntactically-longer individuals, which tends to make fitness computation more computationally expensive. In addition, it may require some modifications to standard genetic operators to cope with relatively complex individuals. By contrast, in the Michigan approach the individuals are simpler and syntactically shorter. This tends to reduce the time taken to compute the fitness function and to simplify the design of genetic operators. However, this advantage comes with a cost. First of all, since the fitness function evaluates the quality of each rule separately, now it is not easy to compute the quality of the rule set as a whole - i.e. taking rule interactions into account. In this paper Michigan's approach is opted i.e. each individual encodes single rule. The encoding can be done in a number of ways like, binary encoding or expression encoding etc. For example let's consider a rule " If a customer buys milk and bread then he will also buy butter", which can be simply written as

If milk **and** bread **then** butter

Now, following Michigan's approach and binary encoding, for simplicity sake, this rule can be represented as **00** 01 **01** 01 **10** 01 where, the bold di-digits are used as product id, like 00 for milk, 01 for bread and 10 for butter and the normal di-digits are 00 or 01 which shows absence or presence respectively. Now this rule is ready for further computations.

Second, area of concern is Genetic Operators. Mainly three operations are to be performed, selection, cross-over and mutation to robustly search the rule space for various options. Selection involves selecting two fit parents for evolving new children rules which are fit than the parents, and in this manner the average fitness of the rules can be increased. Cross-over and mutation provides the ways to evolve new rules.

Third area of concern is fitness function. Since, the discovered rules should: (a) have a high predictive accuracy; (b) be comprehensible; and (c) be interesting, thus choice of this function is very important to get the desired results.

2 Methodology

In this paper the genetic algorithms are applied over the rules fetched from association rule mining. Now for demonstration its utility, the database is produced synthetically. This database contains the choice of electives by students during their 3rd year of course studies at Indian Institute of Information Technology, Allahabad, India. Students have to choose four subjects from eight based on their liking and area of interest. Now, the authors firstly implemented Association Rule mining (using a-priori technique) by the help of their toolkit [3]. And then the GAs are applied to evolve the rules which contains the negations in attributes and are of richer quality. In this section the paper discusses each step in detail.

2.1 Association Rule Mining (a-priori)

The Algorithm for its implementation is same as described in section 1.1. The rules came out of it looks like:

IF NFC **&** RIA **THEN** BI
IF QC **&** VLSI **THEN** IPR
...

Where NFC, RIA, QC, VLSI, etc. are of the eight subjects out of which student has to choose four. The rules above shows that if a student takes NFC and RIA then the probability is high that he will choose BI too, similarly if he chooses QC and VLSI then the probability is high that he will take IPR.

There is no boundation on the number of antecedents in the rules, but there is a constraint on the number of consequents, and i.e.

number of consequents = 1

This boundation doesn't make any harm, because if in case the user wants to see the confidence value of a rule that contains the more than one consequents can do the same by taking two rules from our system and then by doing the intersection of it.

2.2 Genetic Algorithms with Modifications

The GAs implemented over here involves the basics learnt from Goldenberg's book on Genetic Algorithms. The following subsection will tell in more detail various choices,

(a) Individual Representation

The individuals are represented using the *Michigan's approach,* i.e. each individual encodes single rule. As discussed in section 1.2, about its merits and de-merits.

(b)Representing the rule antecedent (a conjunction of conditions)

Here the authors have employed binary encoding, for the rules. An example encoding is as follows, since the antecedent can contain all the subjects, thus it needs to have space for all eight subjects. Thus in antecedent each subject has a predefined location and each subject needs two bits for representation, now the two bits can represent four different states, but in this paper the authors have used just three out of them.

00	**10**	**00**	**10**	**01**	**00**	**10**	**01**
1	2	3	4	5	6	7	8

Where, the lower numbereing (from 1 to 8) defines the slots for various subjects, like 1 is for NFC, 2 is for RIA etc. in this manner till eighth slot. The upper bold row gives the status of each subject defined by the student, like, if a student chooses the subject than its particular slot will contain a di-digit of 01, similarly if he do not chooses the subject than 00, and if the subject do not matter in this rule that 10 (while the option of 11 is not used here).

For the consequent part the same encoding is used except that only one consequent is allowed.

(c) Generic Operators

For *selection* the authors used Roullete Wheel Sampling procedure, in this procedure, the parents for crossover and mutations are selected based on their fitness, i.e. if a candidate has more fitness function value more will be its chance to get selected. The implementation of Roullete Wheel Sampling is done by first normalizing the values of all candidates so that, there probabilities lie between 0 and 1, and then by using Java's random number function, a random number is evaluated, and then corresponding to this value and the fitness normalized value, the candidate is selected.

Mutation : This part of the genetic algorithms, require great care, here there are two probabilities, one usually called as p_m, this probability will be used to judge whether mutation has to be done or not, when the candidate fulfills this criterion it will be fed to another probability and that is, locus probability that is on which point of the candidate the mutation has to be done.

In the case of database provided, binary encoding is used thus simple toggling operator is requird for mutation, i.e. mutate 0->1 and 1->0.

Crossover : Same as the case with Mutation here two probabilities are there, one for the whether crossover has to be performed or not, i.e. p_c, and other for finding the location, the point where, crossover must be done.

This paper has used single point crossover technique for mutation as described above.

(d) Fitness function

A general problem of over-fitting is occurred if simple confidence factor is used as described in [1]. Thus the authors used the following method; described in [1]. Let a rule be of the form:

IF A THEN C,

where A is the antecedent (a conjunction of conditions) and C is the consequent (predicted class). The predictive performance of a rule can be summarized by a 2 x 2 matrix, sometimes called a confusion matrix, as illustrated in the following fig.

		actual class	
		C	not C
predicted class	C	TP	FP
	not C	FN	TN

Fig 1 : Confusion Matrix for a classification rule

The labels in each quadrant of the matrix have the following meaning:

TP = True Positives = Number of examples satisfying A and C

FP = False Positives = Number of examples satisfying A but not C

FN = False Negatives = Number of examples not satisfying A but satisfying C

TN = True Negatives = Number of examples not satisfying A nor C

Clearly, the higher the values of TP and TN, and the lower the values of FP and FN, the better the rule.

$$\text{Confidence Factor, } CF = TP / (TP + FP) \qquad (1)$$

Now measure the predictive accuracy of a rule by taking into account not only its CF but also a measure of how "complete" the rule is, i.e. what is the proportion of examples having the predicted class C that is actually covered by the rule antecedent. The rule completeness measure, denoted Comp, is computed by the formula:

$$Comp = TP / (TP + FN) \qquad (2)$$

In order to combine the CF and Comp measures one can define a fitness function such as:

$$Fitness = CF \times Comp \qquad (3)$$

Although this fitness function does a good job in evaluating predictive performance, it has nothing to say about the comprehensibility of the rule. This fitness function can be extended (or any other focusing only on the predictive accuracy of the rule) with a rule comprehensibility measure in several ways. A simple approach is to define a fitness function such as

$$Fitness = w1 \times (CF \times Comp.) + w2 \times Simp \qquad (4)$$

where Simp is a measure of rule simplicity (normalized to take on values in the range 0..1) and w1 and w2 are user-defined weights. The Simp measure can be defined in many different ways, depending on the application domain and on the user. In general, its value is inversely proportional to the number of conditions in the rule antecedent – i.e., the shorter the rule, the simpler it is.

3 Results

As described earlier, in this paper the implementation of GAs are applied to the rules obtained by applying association rule mining on synthetic database which is based on the selection procedure of electives in 3rd year of B.Tech course. The database was made at random.

The columns in the database are the subjects and their value is either one or zero depending on whether they are selected or not. Following figure shows a database glimpse.

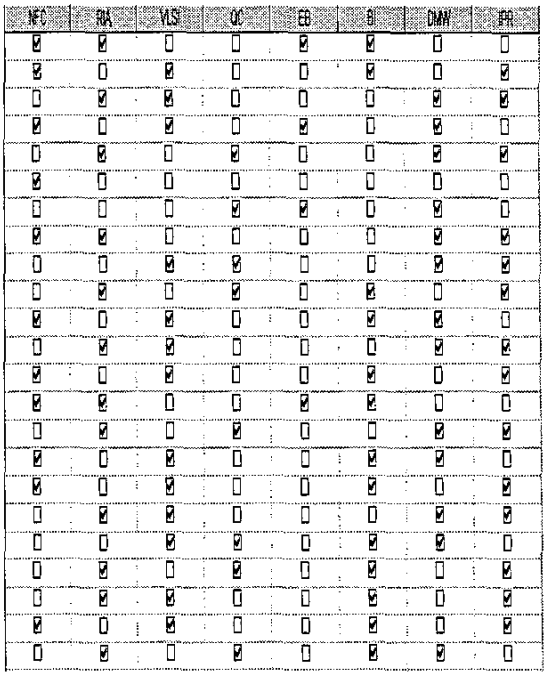

Fig 2 : Glimpse of Database

The parameters used for GAs in this paper are as follows,

Table 1 : GA parameters

GA Population	100
Selection Rule	Roullete Wheel Sampling with DeJong's Crowding Procedure
Cross Over Probability	0.1
Mutation Probability	0.005
Fitness Function	As described in section 2.2.4
Accuracy of the rules	100 %

The rules evolved out of the system after application of apriori –association rule mining and GAs in succesion contains some rules with negations in the attributes as predicted and desired.

The following rules were evolved from 100 rows of the synthetic database created.

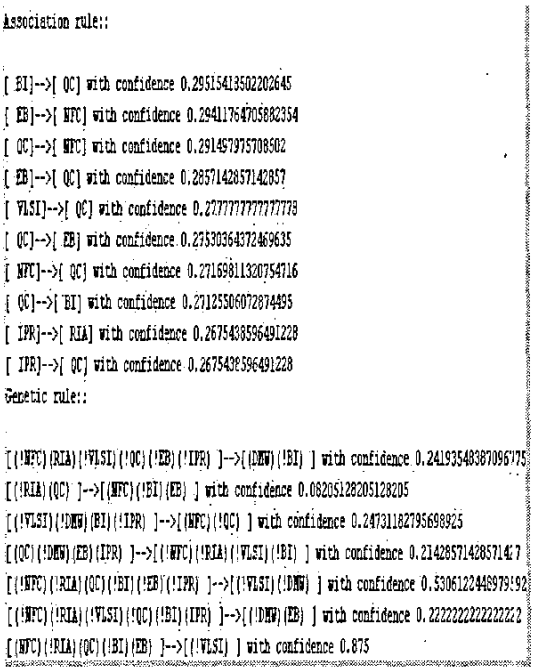

Fig 3 : Showing Association rules and new Genetically evolved rules

4 Conclusions and Future Work

Although a number of works are already published in this field, but in this paper the authors have tried to use the enormous robustness of GAs in mining the Association Rules. The results generated when the technique applied on the synthetic database, includes the desired rules, i.e. rules containing the negation of the attributes as well as the general rules evolved from the Association Rule Mining.

The authors believe that the toolkit can also handle other databases, after minor modifications. As for future work, the authors are currently working on the complexity reduction of Genetic Algorithms by using distributed computing.

References

[1] Alex A. Freitas, "A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery" Postgraduate Program in Computer Science, Pontificia Universidade Catolica do Parana Rua Imaculada Conceicao, 1155. Curitiba - PR. 80215-901. Brazil.

[2] C. Silverstein, S. Brin, R. Motwani and J.D. Ullan. Scalable techniques for mining causal structures. In the Proc. of 1998 ACM SIGMOD Int'l Conf. on Management of Data, Seattle, Washington, USA, June 1998.

[3] Manish Saggar, Ashish K. Agarwal, Abhishek Agarwal; Discovery- A Data Mining Tookit, Under graduate mini-project in 4th semester of 4 year B.TECH course from IIIT-A.

[4] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In Proc. of the 3rd Int'l Conf. on Information and Knowledge Management, Gaithersburg, Maryland, 29. Nov - 2. Dec 1994.

[5] R.Agrawal, T. Imielinski, and A.Swami. Mining association rules between sets of items in large databases. In the Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (ACM SIGMOD '93), Washington, USA, May 1993.

[6] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalising association rules to correlations. In the Proc. of the ACM SIGMOD Int'l Conference on Management of Data (ACM SIGMOD '97).

[7] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In the Proc. of the ACM SIGMOD Int'l Conf. on Management of Data, 1997.