HYBRIDIZATION OF SOFT COMPUTING TECHNIQUE WITH APRIORI FOR ASSOCIATION RULE MINING

Rathi.R*, Assistant professor [senior], VIT University, Vellore.

Email: rathi.r@vit.ac.in

Received on 25-07-2016
Accepted on 28-08-2016

Abstract

The modern day rule generation by association rule mining (a-priori algorithm) and optimization of the same using genetic algorithms is rather a complex process. The crux of optimization problem is the complexity and accuracy of the methods used to either mine interesting rules or to extract interesting rules from the ones generated initially. This paper has been written with a simple agenda of unifying many elegant and useful techniques to create a less complex yet accurate algorithm for rule optimization and also reduce the computation time.

Keywords: Association rule mining, Genetic algorithms, Data mining, Multi-objective interestingness metrics.

1 Introduction

The associate rule mining (ARM) introduced in [1], is a mechanism to extract hidden, novel and interesting facts from database/dataset and to draw inferences on as to how different sets of items influence one another. The most widely used ARM method, which is also being widely used in this paper is the a-priori algorithm.

A-priori method is widely used for rule generation because of its elegant fashion and accuracy. Upon evaluation of each rule generated, we would find, not all rules are equally useful or in technical terms interesting. The rule interestingness of all the rules generated varies greatly.

To achieve the most interesting rules, we use many optimisation techniques, which optimise the rules and help to achieve sets of ‘optimal’ rules [2]. Optimisation could be carried out in two ways, one where only interesting rules are being generated (using some interestingness metrics), the other (used in this paper) is the one where the most interesting rules are extracted from the rules generated initially. Thus we require genetic algorithms, which are applied over these rules for the extraction of high-level prediction rules by performing a complete search over the database. Also it copies better with the attributes. The methods used in this paper generate interesting rules that are non-dominated (in the final set of rules). But these methods when used with the suggested modifications take
significantly lesser time to be computed. Furthermore, the paper consists of a brief introduction about the keywords, followed by proposed algorithm/methodology in section 2, result & evaluation in section 3 and conclusions & future work in section 4.

1.1 Association rule mining

Introduced in 1993 [1], ARM is widely used for knowledge mining in KDD. In mathematical terms, it can be represented as

\[ X \Rightarrow Y \]

Where \( X \) and \( Y \) are item sets.

It represents that item \( X \) influences item \( Y \) or,

The general meaning of this kind of rule could be explained using the famous bread and butter [3] scenario. To understand better, let us for example, take a database \( D \) which has let us say \( N \) number of tuples [8]. Let \( T \) be a transaction in \( D \). Now if \( X \Rightarrow Y \), is a rule, then it states that if an item \( X \) is present in \( T \), then the item \( Y \) is (let us assume) \( p\% \) probable to be present in \( T \). Thus in a purchase transaction \( B \) of bread, the customer is \( p\% \) probable to buy butter. Thus (let us assume \( X \) is bread and \( Y \) is butter, then according to the above rule) if a customer buys bread, then he is most probable to buy butter too, and hence both, bread and butter together, are also probable to be present in the transaction.

The direct applicability of ARM to business and its understandability even to a layman makes it a popular mining method. It is widely used in business, analytics and many more such areas.

Apart from various advantages, it has certain loopholes too. First is the complexity. As the number of item sets increases, the complexity proportionally increases. Secondly the problem of optimizing rules i.e. finding the best rules from a given set of rules and lastly dealing with negation of attributes [4].

1.2 Genetic Algorithms

As mentioned and discussed in [5], using GAs in high-level prediction rules is necessary and inevitable as they perform a global search and cope better with the attribute interaction. Various aspects being discussed about GAs here are-

Representation of rules

Genetic operators

Fitness Function
1.2.1 Representation of Rules

The representation of rules is being done as discussed in [5], using Michigan & Pittsburg approach along with binary coding.

For example,

If Physics and Chemistry then Maths

The above rule represents the fact that if a student chooses Physics and Chemistry as his subjects, he is let us say x% probable to choose Maths too. Following the approaches in [5] and binary coding, the above rule can be represented as-

00 11 01 11 10 11

where the underlined digits are subject ids for Physics, Chemistry and Maths. Digits like 00 and 11 shows absence or presence respectively.

1.2.2 Genetic Operators

Selection process involves three main operations to be performed, selection, cross-over & mutation for global search.

Selection in other words could be described as a process to choose 2 fit rules which could be then evolved into a new better and fitter rule from the former one. The latter increases the average fitness and provides us with a better rule.

Cross-over and mutation are used to evolve new rules.

1.2.3 Fitness Functions

Fitness functions measure how fit or novel or interesting a rule is. There are many ways to calculate the fitness, and they are chosen according to the need.

1.3 Data Mining

Data mining or knowledge mining is the process of mining certain data and then performing further optimisation techniques on them to evolve better information.

1.4 Multi-objective interestingness metrics


2. The proposed algorithm

To present the algorithm proposed & its detailed step wise working, let us consider a synthetic database depicting a hypothetical scenario. The database D, taken into consideration in this paper, consists of the yearly production &
The sales of the stationary items produced by XYZ Company for two consecutive years, 2010 & 2011. The database is actually broken into many sets called the datasets or records.

Thus each dataset for the 2 years contains many more records. Records on the basis of selling data for each item in the four zones of a city, inter alia North, East, West and South, records on the basis of various salesmen in each area and the number of products sold by each of them, etc.

The general methods when performed with the suggested modifications evolve into rules which take lesser time to be computed than the methods without modifications. Therefore, we generate interesting, non dominated rules in a lesser time.

2.1 Association Rule Mining (a-priori)

The a-priori method was implemented on the database, resulting in the following rules,

If Pencil and Desk then Pen

If binder then pen

………. followed by many such rules

Here we conclude that there is no need for any boundaries on the antecedent part. It could be any number (< total number of items), but the consequent part must be greater than at least one, else the rule holds no significance. Thus there is a bound on the consequence which is, Number of consequence >= 1

2.2 Representation

2.2.1 Individual Representation

Using Michigan’s approach [5], each individual item could be encoded.

2.2.2 Representation of rules

Using [7] and binary coding, the rules could be in binary format. For example, let us take the rule

If Pencil and Desk then Pen

The items pencil, desk, pen, penset, and binder of the database, D, could be represented using 00, as item id. The bi-digit of 01 represents that a customer buys the item, 00 represents the customer does not buy the item and 10 represents the item does not matter in this rule, (the option of 11 left unused).

For the representation of consequent part, similar encoding is used except for the fact that only one consequent is allowed in this rule & accordingly the number of consequents to be encoded is selected.
2.3 Genetic Operators

It is proposed that 2 groups: internal population P, & a pareto-store, P’, are maintained from where the individuals would be selected/updated from/after genetic operations.

2.3.1 Selection

For selection, Roulette wheel sampling procedure has been used in this paper. In the given procedure, the two parent rules for crossover and mutation are selected on the basis of their fitness value, i.e., the more fit the rule, the better is the chances of getting selected. In this procedure, the values of all the candidates/items is normalised first (to make them lie between 1 & 0), further a random item is chosen and evaluated and compared to the rest normalised values of candidates and selected accordingly[7].

2.3.2 Mutation

In mutation, firstly, using probability it is determined whether the mutation has to be done or not and later it is determined that at which point the mutation has to be performed.

In binary encoding, simple toggling operation is required for mutation i.e.

\[
1 \rightarrow 0 \text{ and } 0 \rightarrow 1
\]

Thus 110111 becomes after mutation 110100 is performed at a particular point.

Mutation is modified by [6] enabling the mutation operator to either generalise/specialize a rule condition by changing attribute value or condition’s interval value.

2.3.3 Crossover

Like mutation, there are probabilities whether crossover would be performed and point where it would be performed.

The crossover operator is proposed to be modified either to generalize the crossover operator if the rule is too specific (using bitwise OR), or to specialize if the rule is too general (using bitwise AND). [6]

These operators are applied to the antecedent part to obtain resultant consequent part.

2.3.4 Fitness Function

According to [4] & [6], the fitness function can be performed using the following methods,

First, consider the rule \( \text{If A then C} \), where A is antecedent part & C is consequent part. The predicted performance can be calculated as given in [5], thus we conclude that

Confidence factor, \( \text{CF} = \frac{TP}{TP + FP} \)

Rule completeness, \( \text{Comp} = \frac{TP}{TP + FN} \)
Thus, fitness function = (F X Comp) i.e. the product of confidence factor & completeness of the rule. It could be further extended by using w1, w2 & Simp.

Where w1 & w2 – user defined weights & Simp – simplicity of rule i.e. lesser the number of conditions, simpler the rule and similarly, more the conditions, more complex the rule.

The fitness function is further proposed to be modified according to algorithms in [6].

For the modification presentation, let us understand the concept of transactional superiority & syntactic superiority. Rules can be superior to each other in many ways. They could be superior or inferior on the basis of their novelty, easy to understand factor, usefulness etc. As defined in [6], transactional superiority is a measure of the number of records that satisfy a given rule, r, i.e. a rule r is said to be transaction-wise superior to another rule r’, if rule r holds true for more records than rule r’.

If there is no difference in the number of records that satisfy the rules r and r’, we use concept of syntactic superiority.

Syntactic superiority measures the difference in rules in the attribute space. Thus rule r is said to be superior to r’ if it covers more attributes than those covered by r’.

The transactional superiority is calculated as

$$Ts (r, r’) = |s(r1) - s(r2)|/ |r_c|$$  \[7\]

& syntactic superiority is calculated as

$$Ss (r, r’) = \sum_{i=0}^{n} (p(R1ai | c) - P(R2Ai | c))$$

Overall superiority of rule r over r’ is calculated as

$$S (r, r’) = Ts (r, r’) + Ss (r, r’), \text{ which is the modified fitness function}$$

### 2.3.5 Pareto store update

The Pareto store is populated by firstly copying all the non-dominated rules to the store. Further, the rules found to be dominated by other rules are removed iteratively until we reach a set of rules that are completely non-dominated.

### 3. Results

This paper implements the genetic algorithm and other related modifications on the database showing yearly production & sales of stationary items for different years, different zones and again for different sales by different salesman. The database used was synthetically designed.
Derived from the various records of the database, a second version of the database was created where the columns represented various items and the values 1 or 0 were used to represent the sales of the items for all the four zones and different salesmen in the year 2010. The same could be done for the year 2011 & similarly extended for other records.

Thus the GA operators used on the database & their results could be depicted in a tabular form as follows –

<table>
<thead>
<tr>
<th>GA Population</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operators/Methods</td>
<td>Association Rule Mining Genetic Operators- Selection -Mutation -Crossover -Fitness Function (Inclusion of many Modifications)</td>
</tr>
<tr>
<td>Accuracy of the rules</td>
<td>100%</td>
</tr>
</tbody>
</table>

Also when compared, the modified methods took significantly lesser time than the general methods, for rule generation & optimisation.

4 Conclusion & Future Work

Though a lot has been published in this field, but this paper has been written only to unify various powerful, useful, efficient and accurate GA operators to find the best possible set of rules in a minimum time that remain and are novel,
useful and easy to understand to the users. Also this paper helps to realize the importance of ARM and GAs and their usefulness in fields like business, marketing, business analysis etc. It also showcases various modifications that help achieve optimal rules in a minimum time.

Though the area is becoming indispensable to the modern day computing applications, a lot is yet to be achieved. The complexity of the scenarios is still not handled very well. The author is trying to study more methods to solve the complexity issue and is optimistic about developing methods that could handle the complexity well and more efficiently.

References

   {write year}.

Corresponding Author:
Rathi.R*,
Email: rathi.r@vit.ac.in