مجلة القادسية للعلوم الصرفة المجلد (12) العدد (1) لسنة 2007

Apply Genetic Algorithm To Generate Association Rule

Paper is presented by Lamia AbedNoor
Computer Science Department
College of Sciences
Al-Qadissiya University

Abstract

This paper presents a new approach for extracting the frequent itemsets that may be included in the frequent pattern using genetic algorithm. To construct an association rule pattern, we need to find the itemsets that are the most frequent in the data set. So the time that can be consumed to perform this task will grow exponential as the size of data is increased.

In this work, we attempt to develop an approach based on the genetic algorithm. However genetic algorithm can be used in evolving the itemsets in order to obtain the intended itemsets that achieve at least the minimum support and this can be done automatically.

Keywords: association rule, itemset, and genetic algorithm

1. Introduction

An association rule generated is considered one of different tools that are exploited to extract the unknown pattern from data set. The association rule consists of combination of data items that appears as the form (X=>Y) Where Y represents some items that depend on the X that represents another items.

To generate these rules, two stages must be performed. In first stage, the work concentrates in the searching the data set in order to find the item sets (groups of different combination items) that satisfy the minimum support that are determinate previously by the user. While in second stage, the relations between the items within the candidate itemset are recognized in

order to differentiate the independent items from the dependent items and then construct the rule as the form (X=>Y).

The work in the second stage is relatively considered simple. But the work in the first stage is not trivial. The generation of the frequent itemsets that satisfy the minimum support in the huge data set is not trivial task. So different algorithms were emerged in this field. Each one has its advantages and the suitable environment that it can be applied. We suggest a new algorithm (we call it **GenRul**) to generate the itemsets that constitute the association rules. We will discuss the details of GenRul in this paper to give a suitable view for this algorithm. This paper also contains some foundation items that support this view. The other sections of this paper are organized as follows: Section 2 discusses the related work. Section 3 gives the background of the association rule. Section 4 introduces simple view about genetic algorithms. Section 5 discusses the details of the proposed algorithm GenRul and the results that are produced after applying it on experiment data. At last section 6 concludes the related recommendations from this work.

2. Related Work

Related works seek about the algorithms that can discover the itemsets with efficient time. One of the important algorithms is PRIORI suggested by Agrawal&Srikant in 1994[1]. PRIORI algorithm contains number of iterations. At each iteration, large candidate itemsets are constructed from the itemsets that were generated in previous iteration and so on by combining the itemsets that have the sufficient support.

PHD algorithm was proposed in 1995[2]. It uses the technique of hashing to fitter out unnecessary itemsets for next candidate generation. It constructs hash table that contains the information about (k+1) itemsets. Each bucket in the hash table consists of a number to denote how many itemsets have been hashed to that bucket so far. Based on this hash table, a bit vectore is one if the number in the corresponding bucket is greater than or equal to minimum support. In the candidate generation stage,. After an

itemset is computed, each k-itemset is checked if hashed to bucket whose bit vector is one.

PARTITION algorithm that was suggested in 1995[3]. It scans the target dataset only twice. In scan 1, the database is partitioned and the local frequent pattern is found. While in scan 2, the global frequent is consolidated the SAMPLING algorithm [4] that select a sample of original database, mine frequent patterns within sample using APRIORI. Then scan database once to verify frequent itemsets found in sample, only borders of closure of frequent patterns are checked. At least scan database again to find missed frequent patterns.

The previous algorithms are cost. Where the mining long patterns need many passes of scanning and generates lots of candidates. Another type of algorithms that based on construct frequent itemsets with minimum database passes. Where the construct a frequent pattern tree by scanning the database for limited times. This approach also suffers from the overhead of saving frequent pattern tree in the memory because it's extensive sizes. Different approach was suggested that based on the constraints that are provided by the user. Using the user constraints means this approach is not performed automatically. Instead it is drove by the user to mine what he want to be mined [5].

3. Association Rule Mining

Association rule mining is the most commonly used data mining operation. It is an important class of regularities that exist in the data set. It entails generation of association rules based on the given set of data. These rules are defined using a stochastic approach, based on probabilities drawn up on the various attributes of the database.

An association rule is a rule which implies certain association relationships among a set of objects (such as "occur together or one implies the other") in a database. Given a set of transactions where each transaction is a set of literal (called items), an "association rule" is an expression of the form

“X=>Y has support S and confidence C “ where “X” and “Y” are two attributes or set of attributes (called itemsets) is the given database[6].So Association rules specify correlation between frequent itemsets.

The task of association analysis is typically performed in two steps. First, all frequent itemsets are found. Where an itemset is frequent if it appears in at least a given support which is called minimum support of all transaction. Next association rules are found the form X=>Y that means the percentage of transactions containing X also containing Y passes a threshold.

Support: is a numeric value that refers to the number of the occurrence of the item in the database. While the support of a rule “X=>Y” is the probability of attributes (or attribute sets) X and Y occurring together in the same transactions. Suppose there are (n) transactions in the database, X and Y occurs together in exactly (m) of them, then support of the rule ““X=>Y” is: m/n.

Confidence: The confidence of a rule “X=>Y” is defined as the probability of occurrence of X and Y together in all transactions in which X already occurs. If there are (n) transactions in which X occurs, and in exactly (m) of them X and Y occur together, then confidence of the rule is: m/n

Minimum Support: Minimum support is an input parameter to the algorithm for generating association rules. It defines the threshold support, and rules that have support greater than minimum support only are generated.

Minimum Confidence: Minimum confidence is an input parameter that defines the minimum level of confidence that a rule must posses.

Candidate and Large Itemsets:

Association rules are very closely linked to the concept of itemsets. A k-itemset is defined as a tuple in which there k-attributes (item). Therefore any collection of tuples(transactions) in which each tuple has exactly k attributes is said to constitute a set of k-itemsets.

4. Genetic Algorithm (GA)

Genetic algorithms are stochastic search methods that mimic the metaphor of natural biological evolution. Genetic algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better approximations to a solution.

At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breading them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

Genetic algorithm differentiates from other evolutionary computation methods. However, it can be said that most methods called "GA" have at least the following elements in common: population of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutation of new offspring.

The chromosomes in a GA population typically take the form of bit string. Each locus in the chromosome has two possible alleles: 0 and 1. Each chromosome can be thought of as a point in the search space of candidate solutions. The GA processes populations of chromosome, successively replacing one such population with another.

General Schema of GA

1- initialize population;

The population is a set of chromosomes. An individual chromosome represents a candidate solution for the problem.

2- evaluate each individual in population;

Calculate the fitness of each individual according to some function.

3- *Repeat*

4- select parents;

The selection of parents is done according to their fitness.

5- recombination pairs of parents;

Recombination is achieved using genetic operator that is called (crossover), which performs the exchange, between subsequences of mating chromosomes.

6- mutate the resulting offspring;

Mutation is another genetic operator, which randomly flips some bits in chromosomes according to some probability.

7- evaluate offspring;

Calculate the fitness for new generation of chromosomes.

8- insert offspring in the populations;

The population is reconstructed with new chromosomes

9- until (stopping criteria)

The algorithm is stopped according to some constraints.

10- *extract solution from population;*

After the algorithm is stopped, the desired solutions are extracted from the population according to some conditions.

The Need for Genetic Algorithm

The evolution property of genetic algorithm is the most important to use it in computational problems for different reasons. One of these reasons is that many computational problems require searching through a huge number of possibilities for solutions [7]. The problem in this paper is characterized by large search space. While the goal is to discover the itemsets with high frequencies in database, this means, the database must be scanned based on all the possible combinations of attributes' states. It is obvious, the required time for this task will be grown exponentially if the state or/and attributes are increased and in result it affects on the growth in the number of combinations.

5.GenRul Algorithm

This algorithm is suggested in order to evolving the itemsets that constitute the association rules. Initially generate random itemsets from the database as in the example that is shown in

table (1). Then evolve these itemsets in order to obtain the suitable itemsets that passes the threshold of the minimum support. Notice this algorithm deals with categorical attributes. GenRul can be done in multiple steps:

Outlook	Temp(F)	Humidity(%)	Windy?	Class
sunny	75	70	true	Play
sunny	80	90	true	Don't Play
sunny	85	85	false	Don't Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
overcast	72	90	true	Play
overcast	83	78	false	Play
overcast	64	65	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play
rain	65	70	true	Don't Play
rain	75	80	false	Play
rain	68	80	false	Play
rain	70	96	false	Play

Table.1: A small training data set [1]

Encoding the problem

As we know the type of encoding that is used to represent the problem is related with the application of this problem [7]. The problem here is the itemset that consists of number of categorical attributes. The suggested encoding is to represent the chromosome that consists of the attributes. Where the chromosome is divided into multiple areas (that is analog to bit string). Each area is allocated to one attribute (for example as shown in figure 1). The value that can be associated to each one can be constrained to the domain of the attributes. The number of the possible values is called cardinality of the attribute. The size of domain of each area in chromosome differs from one to other. For example the initial population that can be generated is shown in figure(2).It consists of four chromosomes. Then fitness function that is represented by support value of the object that hold these attributes in data set as shown in table (2).

Outlook	windy	class

Figure(1)

(The intended itemsets consists of three attributes (outlook, windy, class). That means the desired chromosome contains three partitions that represent locus.)

Chromosome1	rain	false	Play
Chromosome2	rain	true	Play
Chromosome3	sunny	true	Play
Chromosome4	overcast	false	Don't play

Figure(2)

(The initial population that consists of four chromosomes)

No. of Chromosome	Fitness(support)

1	3
2	0
3	1
4	0

Table(2)
(Fitness values for the above example)

Input Parameter

GenRul require to some parameters that constraint the evolving process before its beginning like: -

-The size of chromosome that refers to the number of divisions of the chromosome must be defined.

-The minimum support for the itemsets.

-The stopping criteria to end the evolving process.

-The mutation rate.

Search Space

The search space (P) contains all the possible combinations of attributes' states that can be founded in data set. The individuals are constructed from several attribute with specific states in order to represent some combinations as shown in figure (1). The number of possible solutions can be computes as following:

$$Z = C_r^n = \frac{n!}{(n-r)!r!} \qquad (1)$$

Z: The number of combinations of the desired number of attributes (r) which represent itemsets from available attributes (n) in data set.

C={C1,C2,…….Cz}

Where C is a set of combinations according to (1), each combination Ci contains a specific group of attributes. An attribute may hold multiple states (S), so in results combination Ci produces some possible choices.

Ci=S1*S2*……Sr (2)

Where Sj is the number of states which attribute j holds.

As a result and according to equations (1), (2), the possible solutions can be extracted from the data set are (P):

$$P = \sum_{i=1}^{z} Ci \qquad (3)$$

Fitness function

The goal from the GenRul is to find the itemsets (combination of attributes) with high frequencies (support). So the fitness of an individual is calculated by occurrence times of the possible combination of attributes' states that represent in data set as shown in table (2).

GenRul Algorithm

-Input the size of chromosome (number of attributes that evolve).
-Build the initial population of chromosomes.
-Repeat until (stopping condition).
{
-For each individual chromosome (itemsets), compute its fitness (support value for the itemsets).
Extract the itemsets that satisfy the minimum support to constitute the association rules.
Choose randomly the number of the chromosomes (itemsets) that satisfy the minimum support to be parent for reproduce offspring.
Select the divisions of the pairs of chromosomes to be exchanges.
Mutate produced offspring if the time of mutation operation is less than mutation rate.
Evaluate the resulted offspring.
Insert offspring in the population.
}

6. The Experiment and results

Experiment Data

The proposed algorithm was applied on a specific data set that is obtained from the internet site for (UCI Repository [8]) that contains different data sets that are dedicate for researches. The selected data set is university data set. It contains the data about universities. It consists of 200 records and (14) attributes. Some of the attributes are nominal and others are continuous. While the algorithm deals with the nominal values, therefore it was used the specific technique that is known discretizing [9] in order to convert the continuous values into nominal. The states of attributes are two for some attributes and three for others as shown in table (3).

An attribute	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Number of states	3	3	2	2	3	2	3	3	3	3	3	3	3	2

Table(3)
Number of states in each attribute

The Practical work

The algorithm was experimented for several times on the data. At each time, different size of itemsets was experimented starting from 2 attributes and then 3,4,5,6,7. The data set that was experimented has no itemsets over 7 attributes therefor the algorithm was not next to 7 attributes. The minimum support that was used in this experiment varies according to the size of itemsets as shown in table (4) and were drive from the experiments. The population that was used is 8 chromosomes. This size of population is suitable for our problem based on the experiments we applied.

No. of attributes	Minimum support
2	60

3	50
4	40
5	30
6	20
7	10

Table(4)
Minimum Support for each size of itemsets

The Results

The results of the practical work were compared with brute-force technique (that tests all the possible solutions) in relation with the no. of solutions that are evaluated in order to discover the best solutions. While brute-force technique scanned the data set for all possible solutions that can be expected according to equations (1),(2),(3) as shown in table (5). In turn when using the proposed algorithm (GenRul), the no. of candidate solutions that are evaluated based on the size of population and no. of iterations required to reach the desired solutions. The results that are produced from GenRul are shown in table (6). It describes the no. of iterations required to obtain a good solution.

Size of itemset	Possible solutions (P)
2 attributes	669
3 =	≈9000
4 =	≈60000
5 =	≈200000
6 =	≈1000000
7 =	≈3000000

Table(5)
The number of all possible solutions that are available

Size of itemset	No. of iterations

2 attributes	50
3 =	66
4 =	75
5 =	76
6 =	102
7 =	117

Table(6)

No. of iterations that gave good solutions that are available

7. Discussion and Conclusion

-The results that were obtained from the experimented evaluated according to the no. of solutions that were scanned in order to reach to the best solutions because the evaluation the candidate solutions impose scanning the whole data set in order to compute the fitness of them. The scanning the whole data set. These factor plays an important role in real word data set that is characterized by huge amount. So reducing the no. of access data set cycles results in good execution time for algorithm .

- From the practical work, many of possible solutions are not available in the real data set, so the blind search as in brute-force is unacceptable. In turn, the genetic algorithm works with more interact with the intended data set. This means many possible solutions that are not existing in data set can be avoided. As a result a good execution time can be obtained with applying GenRul.
- The growth of the no. of states or attributes extended the possible solutions as we see in table (5), which use limited number of attributes and state, so the search space become incredible if the states or attributes may increase. Therefore it is suitable to seek about the techniques that decrease the

scanning of all the possible solution like the proposed algorithm GenRul.

-The suitable number of iterations that GenRul algorithm requires to give the desired solutions with different size of itemsets as shown in table (6) , this encourage to use with different size of itemset and in the same time it can be applied with a huge databases.

References

[1] Agrawal, R. and Srikant, R. "Fast algorithms for association mining association rules", VLDB-94, p.p.487-499, 1994.

[2] J.Park, M. Chen, and P.Yu, "An effective hash based algorithm for mining association rules", SIGMOD, 1996.

[3] A. Sarasere, E. Omiccinskiy, and S. Navathe."An efficient algorithm for mining association rules in large databases", web page: www.gunther.smeal.psu.edu/context/15954/0.

[4] H. Toivonen, "Sampling large databases for association rule", In 22th International Conference on Very Large Databases (VLDB'96), 134-145, Mumbay, India, September 1996.

Web page: -www.cs.Helsinki.FI/research/fdk/datamining.

[5] C. Chris, "Introduction to data mining", web page:-www.cs.purdue.edu/homes/clifton/cs490d

[6] W. Gauhar, and P. Nagender, "Association Rule mining : Extension of direct hashing and pruning concept to quantitative database", web page:-www.education.vsul.con/gauthar/extendeddhpmining.pdf.

[7]M. Mitchell, "An Introduction to Genetic Algorithms",book,1997.

[8] UCI Repository site :-www.ics.uci.edu/pup/machine-learning-database/university/.

[9] W.Ian H. and F.Eibe; "Data Mining: Practical Machine Learning tools and techniques with JAVA implementaion",2000, Morgan Kaufmann Publishers.