Genetic Fuzzy Data Mining With Divide-And-Conquer Strategy

M.Kannan, P.Yasodha, V.Srividhya
CSA Dept., SCSVMV University, Enathur, Kanchipuram - 631 561.

Abstract:
Data mining is most commonly used in attempts to induce association rules from transaction data. Most previous studies focused on binary-valued transaction data. Transaction data in real-world applications, however, usually consist of quantitative values. This paper, thus, proposes a fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. A genetic algorithm (GA)-based framework for finding membership functions suitable for mining problems is proposed. The fitness of each set of membership functions is evaluated by the fuzzy-supports of the linguistic terms in the large 1-itemsets and by the suitability of the derived membership functions. Experiments are conducted to analyze different fitness functions and setting different supports and confidences. Experiments are also conducted to compare the proposed algorithm, the one with uniform fuzzy partition, and the existing one without divide-and-conquer, with results validating the performance of the proposed algorithm.

Keywords**:** *Genetic algorithm,Association rule,Fuzzy data-mining algorithm*

I. INTRODUCTION

Data mining, *the extraction of hidden predictive information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions.

One of the important problems in data mining is discovering association rules from databases of transactions where each transaction consists of a set of items. The most time consuming operation in this discovery process is the computation of the frequency of the occurrences of interesting subset of items (called candidates) in the database of transactions. Can one develop a method that may avoid or reduce candidate generation and test and utilize some novel data structures to reduce the cost in frequent pattern mining? This is the motivation of my study.

A fast algorithm has been proposed for solving this problem.The association rule mining has been one of the most popular data-mining subjects, which can be simply defined as finding interesting rules from large collections of data. Association rule mining has a wide range of applicability such Market basket analysis, Medical Diagnosis/ research, Website navigation analysis, Homeland security and so on.

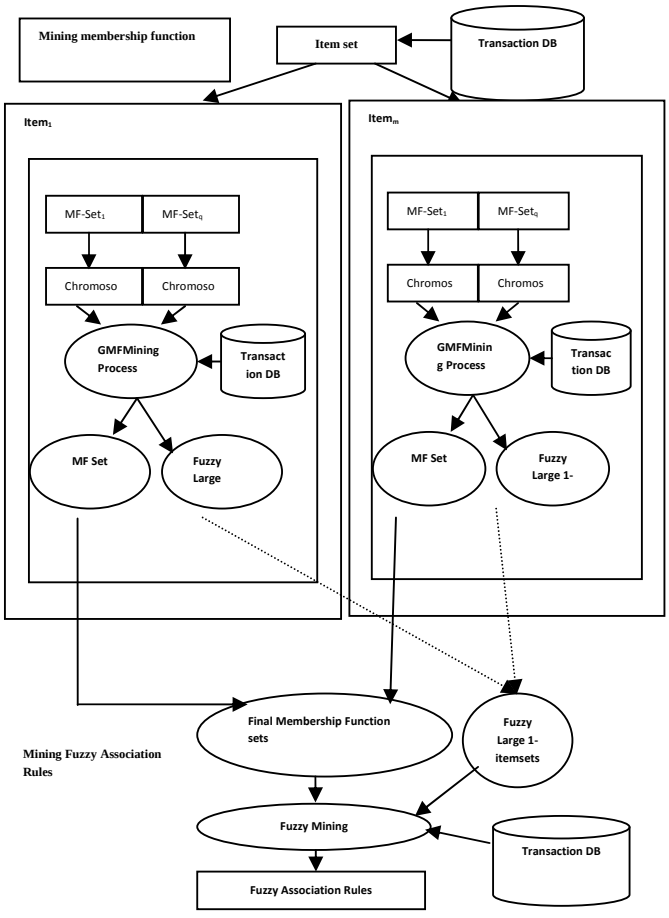
Association rules are used to identify relationships among a set of items in database. These relationships are not based on inherent properties of the data themselves (as with functional dependencies), but rather based on co-occurrence of the data items. The subsequent paper is considered as one of the most important contributions to the subject. Its main algorithm, Apriori, not only influenced the association rule mining community, but it affected other data mining fields as well.

Association rule and frequent itemset mining became a widely researched area, and hence faster and faster algorithms have been presented. Numerous of them are Apriori based algorithms or Apriori modifications

II.RELATED WORK:

In some works they dealt with the algorithmic aspects of association rule mining. We restricted ourselves to the "Classic" association rule problem, that is the generation of all association rules that exist in super market-like data with respect to minimal thresholds for support and confidence. In our work we use the genetic algorithm using association rule to find the support and confidence level of the itemset.

OVER ALL SYSTEM DESIGN STRUCTURE



III. THE PROPOSED MINING ALGORITHM

According to the above description, the proposed algorithm for mining both fuzzy association rules and membership functions based on the divide-and-conquer strategy is described next.

INPUT: A body of quantitative transaction data, a set of items, each with a number of predefined linguistic terms, a support threshold, a confidence threshold, and a population size.

OUTPUT: A set of fuzzy association rules with its associated set of membership functions.

Randomly generate populations, each for an item; each individual in a population represents a possible set of membership functions for that items.

Encode each set of membership functions into a string representation.

- Calculate the fitness value of each chromosome in each population.
- Execute crossover operations on each population.
- Execute mutation operations on each population.
- Using the selection operation to choose individuals in each population for the next generation
- If the termination criterion is not satisfied, go to Step 3; otherwise, do the next step. The termination criterion may be number of iterations, allowed execution time or convergence of the fitness values.
- Gather the sets of membership functions, each of which has the highest fitness value in its population. The sets of best membership functions gathered from all the populations are then used to mine fuzzy association rules from the given quantitative database

IV MINING FUZZY ASSOCIATION RULE:

This module gets the best membership functions gathered from all the Populations then they are used to mine fuzzy association rules.An example is given to illustrate the proposed mining algorithm. This is a simple example to show how the proposed algorithm can be used to mine membership functions and fuzzy association rules from quantitative data. Assume there are four items in a transaction database: milk, bread, cookies, and beverage. The data set includes the six transactions.Assume each item has three fuzzy regions: *Low*, *Middle*, and *High*. Thus, three fuzzy membership functions must be derived for each item.

Four populations are randomly generated, each for one item. Assume the population size is ten in this example. Each population then includes ten individuals. Each individual in the first population is a set of membership functions for item *milk*. Similarly, an individual in the other populations is a set of membership functions, respectively, for bread, cookies, and beverage.

- Each set of membership functions for an item is encoded into a chromosome according to the proposed representation. Assume the ten individuals in the four populations are randomly generated.
- The fitness value of each chromosome is then calculated by the following substeps. Take the chromosome in *Population* as an example. The membership functions in for *milk* are represented as (0 5 10 7 13 16 15 18 18).
- The quantitative value of each item in each transaction datum is transformed into a fuzzy set according to the membership functions represented by that chromosome. Take the first item in transaction as an example. The contents of include (milk, 5), (bread, 10), (cookies, 7), and (beverage, 7). The amount "5" of item *milk* is then converted into the fuzzy set ((1/Milk.Low) + (0/Milk.Medium) + (0/Milk.High)) using the membership functions C_1 in $Population_1$.
- The scalar cardinality of each fuzzy region in the transactions is calculated as the *count* value.Take the fuzzy region *milk.Low* as an example. Its scalar cardinality (1.0 + 0.6 + 0.0 + 0.4 + 0.0 + 0.0) = 2.0.
- The count of any fuzzy region is checked against the predefined minimum support value. Assume in this example, Alpha is set at 0.25. Since only the count value of *milk.Low* is larger than 0.25 * 6(= 1.5), *milk.Low* is then put in L_1.
- Only one large 1-itemset, *milk.Low*, is derived from the membership functions of in. The fuzzy support of *milk.Low* is 2/6 (=0.33) and its suitability is calculated as 1. The fitness value of is C_1, thus 0.33/1(=0.33). The fitness values of all the chromosomes in the four populations are calculated.
- The crossover operator is executed on the populations.
- Assume *d* is set at 0.35. Take C_1 and C_3 in $Population_1$ as an example. The following four candidate offspring chromosomes are generated:

$$\boldsymbol{C}_1 : 0\,5\,10, 7\,13\,16, 15\,18\,18;$$

$$\boldsymbol{C}_3 : 3\,6\,10, 6\,13\,16, 12\,20\,20;$$

$$1)\ C_1^{t+1} : 1.95\,5.65\,10, 6.35\,13\,16, 13.05\,19.3\,19.3;$$

$$2)\ C_2^{t+1} : 1.05\,5.35\,10, 6.65\,13\,16, 13.95\,18.7\,18.7;$$

$$3)\ C_3^{t+1} : 0\,5\,10, 6\,13\,16, 12\,18\,18;$$

$$4)\ C_4^{t+1} : 3\,6\,10, 7\,13\,16, 15\,20\,20.$$

- The fitness value of the above four candidates are then evaluated, the best two of the four candidate offspring chromosomes are chosen. Thus, and are chosen.
- The *mutation operator* is executed to generate possible offspring. The operation is the same as the traditional one except that rearrangement may need to be done.
- Assume the elitism selection strategy is used here. The best ten chromosomes in each population are, thus, selected as the next generation.
- Assume the number of generations is used as the termination criterion. The same procedure is then

executed until the predefined number of generations is achieved.

- The best chromosome (with the highest fitness value) in each population is then output as the membership functions for deriving fuzzy association rules.
- Assume the final individuals in the four populations after the evolutionary process terminates. According to Table Generated, the best individuals in the four populations are C_8 in $Population_1$,C_1 in $Population_2$,C_1 in $Population_3$, C_8 and in Population .The final set of membership functions for each item is shown. After the membership functions are derived, the fuzzy mining method is then used to mine fuzzy association rules.

GENETIC ALGORITHM:

The fuzzy and GA concepts are used to discover both useful association rules and suitable membership functions from quantitative values. A GA-based framework with the divide-and-conquer strategy is proposed for searching for membership functions suitable for the mining problems. The final best sets of membership functions in all the populations are then gathered together to be used for mining fuzzy association rules.

The proposed framework is divided into two phases: mining membership functions and mining fuzzy association rules.

Assume the number of items is. In the phase of mining membership functions, it maintains populations of membership functions, with each population for an item.

Each chromosome in a population represents a possible set of membership functions for that item. The chromosomes in the same population are of the same length.

The proposed mechanism then chooses appropriate strings for gradually creating good offspring sets of membership functions. The offspring sets of membership functions undergo recursive "evolution" until a good set of membership functions has been obtained.

Next, in the phase of mining fuzzy association rules, the sets of membership function for all the items are gathered together and used to mine the fuzzy interesting association rules from the given quantitative database.

Steps of Genetic Algorithm is given below

Calculate Overlap Ratio.

Formula is = Overlap Length

$$\frac{\text{Overlap Length}}{\text{Min ([linguisticRegin]rightSpan,[linguisticRegin]LeftSpan)}}$$

Where overlaplength is (LeftspanMiddleValue – RightSpanFirstValue)

Eg.

Assume the newly created chromosome as {0 5 10 7 11 14 13 16}

For first two spans that is, Substitute values in formula

[Left Span] Low = {0 5 10} so Low [0] = 0, Low [1] =5, Low [2] =10

[Right Span] Mid = {7 11 14} so Mid [0] =8, Mid [1] =11, Mid [2] =14

Find out the Overlap Factor of these two regions first,

1. Formula: Overlap Length = Low [2] – Mid [0]

That is 7 – 10 = 3

So Overlap Length is 3 of there two spans.

Formula: Calculate Overlap ratio as specified in the STEP 1.

As per the spans considered

[Mid] Right Span = Mid [2] – Mid [1] = 14-11=3

[Low] Left Span = Low [1] – Low [0] = 5-0=5

$$\text{OverlapRatio} = \frac{3}{\text{Min } (3, 5)} = 1$$

Overlap Factor = Max (OverlapRatio, 1) - 1

Therefore (Max (1, 1)-1) =0

This Value must be 0.

Repeat the same for rest of the spans (Mid with High and High with Mid spans).

Finally three Overlap Factors will arrive .Sum all the three Factors.

Calculate Coverage Factor

$$\frac{\text{Mid [1] – Low [1]}}{\text{MaxTransQuantity}}$$

Assume Maximum Transaction is 12.

Therefore 11-5=6 -→ 6/12= 0.5

This must be < 2

4. Calculate Suitability

Sum (OverlapFactors) + CoverageFactor = Suitability.

There fore the result is 0 + 0.5 = 0.5

This must be < 2

Fuzzy Support Calculation.

Compares the Transaction Value of all transactions of a particular population to find out in which region the transaction falls. Finally calculate the count os each regions occurrences of a particular item. Whichever the region has maximum count that is called as fuzzy region and its count is called as FuzzySet value. Count can also be called as Scalar Cordinality. We need to calculate Fuzzy Support now. We will use Support value parameter given by the user.

Fuzzy Support = Fuzzy set / Total transactions

To find out fuzzy support we need to calculate minimum support.

(minimum support = Support * NoOfTrans)

From the fuzzy regions ScalarCordinality, check for whichever the value is less than Minimum support. Consider this value as Fuzzyset value.

Therefore Fuzzy Support = Fuzzy set / Total Transactions

Calculate Fittness Value of a chromosome.

$$\text{Fitness} = \frac{\text{FuzzySupport}}{\text{Sutability}}$$

ASSOCIATION RULE PROBLEM:

Frequent itemset mining came from efforts to discover useful patterns in customers' transaction databases. A customers' transaction database is a sequence of transactions (T= {t1, t2,….tn }), where each transaction is an itemset (ti T). An itemset with k elements is called a k-itemset. The support of an itemset X in T denoted as support(X), is the number of those transactions that contain X, i.e. support (X) = |{ti : X tj }| .

An itemset is frequently if its support is greater than a support threshold, originally denoted by min_supp. The frequent itemset mining problem is to find all frequent itemset in a given transaction database. The algorithms were judged for three main tasks: all frequent itemsets mining, closed frequent itemset mining, and maximal frequent itemset mining.A frequent itemset is called closed if there is no superset that has the same support (i.e., is contained in the same number of transactions). Closed itemsets capture all information about the frequent itemsets, because from them the support of any frequent itemset can be determined.

A frequent itemset is called maximal if there no superset that is frequent. Maximal itemsets define the boundary between frequent and infrequent sets in the subset lattice. Any frequent itemset is often also called free itemset to distinguish it from closed and maximal ones.

Obviously, the collection of maximal frequent itemset is a subset of the collection of closed frequent itemset which is a subset of the collection of all frequent itemsets, the supports of all their subsets is not available, while this might be necessary for some applications such as association rules. On the other hand, the closed frequent itemsets from a lossless representation of all frequent itemsets since the support of those itemsets that are not closed is uniquely determined by the closed frequent itemsets.Through our study to find patterns problem

we can divide algorithms into two types: algorithms respectively with and without candidate generation. Any Aprioi-like instance belongs to the first type. Eclat may also be considered as an instance of this type.

The FP-growth algorithm is the best–known instance of the second type. Comparing the two types, the first type needs several database scans. Obviously the second type performs better than the first.

V.RESULTS

FIG1.SALES STATUS REPORT

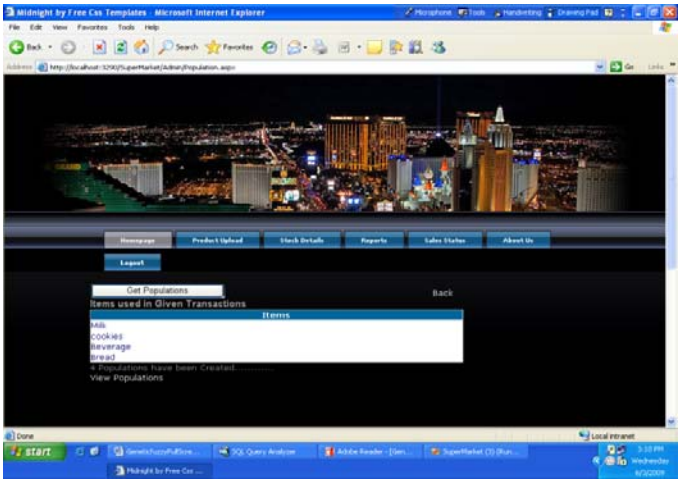


FIG 2 GET POPULATION REPORT

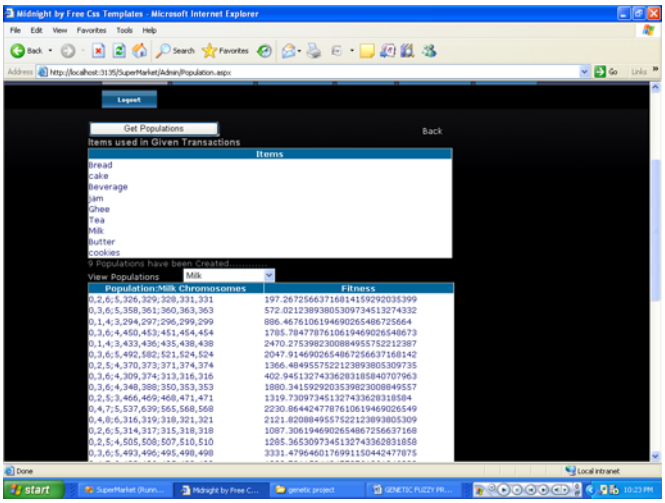


FIG 3. VIEW POPULATION REPORT

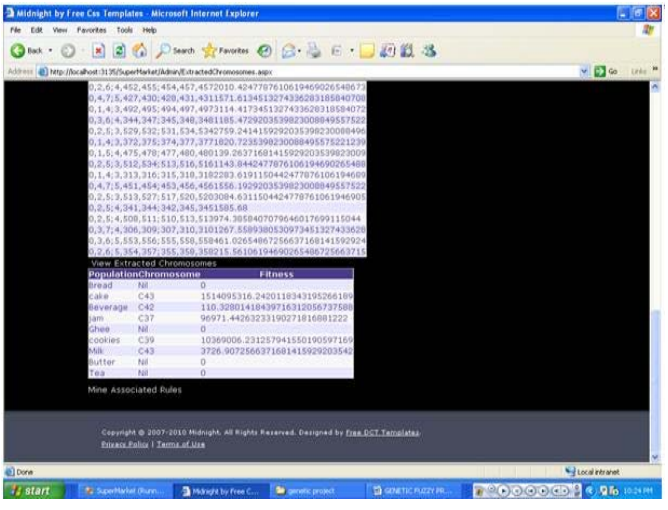


FIG 4.VIEW EXTRACTED CHROMOSOMES

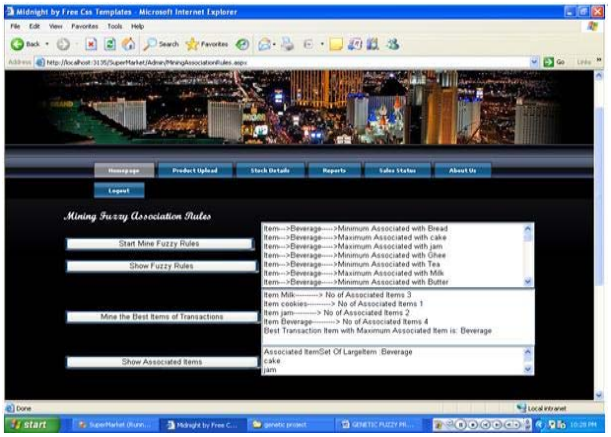


FIG 5 MINE ASSCIOTATION RULE

VI CONCLUSION

In this paper, we have proposed a GA-based fuzzy data mining algorithm for extracting both association rules and membership functions from quantitative transactions. Since the fitness of each set of membership functions is evaluated by fuzzy-support values in large 1-itemsets and by suitability of derived membership functions, the derivation process can easily be done using the divide-and-conquer strategy. We have thus proposed a mining framework, which maintains multiple populations, each for an item's membership functions.

The final best sets of membership functions gathered from all the populations are used to mine fuzzy association rules. The results have also shown that the proposed genetic-fuzzy mining algorithm can get a good tradeoff between fuzzy supports of large 1-itemsets and suitability values of membership functions. Note that if the density of values focuses on a certain interval, the proposed approach can still work. The generated membership functions will have their centers gathered around the interval.

FUTURE APPLICATION OF THE PAPER

The leftmost and the rightmost membership functions may extend to the two end points of the possible range. The membership value, thus, becomes smaller when an item quantity is closer to the end points of the possible range. Although the proposed divide-and-conquer method has fast convergence speed, its disadvantage is that only the supports of large 1-itemsets are considered. If itemsets with more than one item are considered, then the proposed approach may need to be greatly modified. In the future, we will continuously attempt to enhance the GA-based mining framework for more complex problems.

REFERENCES

[1] R.Agarwal, A.Swami. "Mining association rules between sets of items in large databases," Proc.of ACM SIGMOD,pp 207-216,1993.

[2] W.Zhang, "Mining fuzzy quantitative rules,"Proc.of IEEE ICTAI, pp. 99-102,1999.

[3] T.P.Hong, M.J.Chiang and S.L.Wang, "Mining from quantitative data with linguistic minimum supports and confidences, "Proc. Of IEEE-FUZZ, 2002,PP.494-499.

[4] C.M.Kuok, A.W. Fu, and M.H. Wong. "Mining fuzzy association rules in databases, " SIGMOD Record, vol.17, no.1, pp. 41-46, 1998.

[5] K.C.C Chan and W.H. Au, "Mining Fuzzy Association Rules" Proc. Of ACM CIKM, 1997, pp. 209-215.

[6] G.Chen and Q. Wei. Fuzzy data mining: Discovering of fuzzy Databases, pages 45-66. Springer, 2000

[7] Hong TP, Kuo CS, Chi SC (1999) Mining association rules from quantitative data:363-376

[8] R.Uday kiran, P.Krishna Reddy : Mining Rare Association Rules in the Databases with widely Varying Items Frequencies

[9] Brijs, T., Swinnen, G. Vanhoof, G.: The use of association rules for product assortment decisions – a case study. In: Knowledge Discovery and Data Mining

[10] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," in *Proc. Int. Conf. Very Large Databases*, 1994, pp. 487–499.

[11] W. H. Au, K. C. C. Chan, and X. Yao, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 532–545, Dec. 2003.

[14] T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets Syst.*, vol. 84, pp. 33–47, 1996.

[15] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets Syst.*, vol. 141, pp. 59–88, 2004.