Using Divide-and-Conquer GA Strategy in Fuzzy Data Mining

Tzung-Pei Hong
Dept. of Electrical Engineering
National Univ. of Kaohsiung
tphong@nuk.edu.tw

Chun-Hao Chen, Yu-Lung Wu
Inst. of Info. Management
I-Shou University
m9122013@stmail.isu.edu.tw
wuyulung@isu.edu.tw

Yeong-Chyi Lee
Dept. of Info. Engineering
I-Shou University,
9003007d@stmail.isu.edu.tw

Abstract

Data mining is most commonly used in attempts to induce association rules from transaction data. Transactions 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 GA-based framework for finding membership functions suitable for mining problems is proposed. The fitness of each set of membership functions is evaluated using the fuzzy-supports of the linguistic terms in the large 1-itemsets and the suitability of the derived membership functions. The proposed framework thus maintains multiple populations of membership functions, with one population for one item's membership functions. The final best set of membership functions gathered from all the populations is used to effectively mine fuzzy association rules.

1. Introduction

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. Designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field.

Srikant and Agrawal then proposed a mining method [7] to handle quantitative transactions by partitioning the possible values of each attribute. Hong *et al.* proposed a fuzzy mining algorithm to mine fuzzy rules from quantitative data [4]. Wang *et al.* used GAs to tune membership functions for intrusion detection systems based on similarity of association rules [10]. Kaya *et al.* [6] proposed a GA-based clustering method to derive a predefined number of membership functions for getting a maximum profit within an interval of user specified minimum support values.

In [4], we proposed a mining approach that integrated fuzzy-set concepts with the apriori mining algorithm [1] to find interesting itemsets and fuzzy association rules in transaction data with quantitative values. In that paper, the membership functions were assumed to be known in advance. The given membership functions may, however, have a critical influence on the final mining results. This paper thus modifies the previous algorithm and proposes a new fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. The proposed algorithm can dynamically adapt membership functions by genetic algorithms and uses them to fuzzify the quantitative transactions. Our previous fuzzy mining approach [5] can thus be easily used to find fuzzy association rules.

2. A GA-based mining framework

In this section, the fuzzy and GA concepts are used to discover both useful association rules and suitable membership functions from quantitative values. A GA-based framework for achieving this purpose is proposed in Figure 1.

The proposed framework is divided into two phases: mining membership functions and mining fuzzy association rules. Assume the number of items is m. In the phase of mining membership functions, it maintains m populations of membership functions, with each population for an item I_j ($1 \leq j \leq m$). Each chromosome in a population represents a possible set of membership functions for that item. The proposed schema then encodes the chromosomes in the same population into fixed-length real strings. It then chooses appropriate strings for "mating", gradually creating good offspring sets of membership functions.

0-7803-8623-X/04/$20.00 ©2004 IEEE

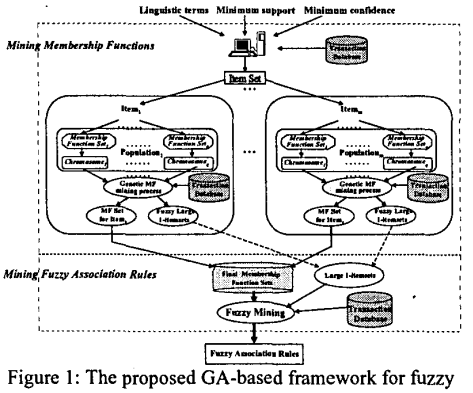

Figure 1: The proposed GA-based framework for fuzzy mining

The offspring sets of membership functions then 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 interesting rules from the given quantitative database. Our fuzzy mining algorithm proposed in [5] is adopted to achieve this purpose.

3 Chromosome representation

It is important to encode membership functions as string representation for GAs to be applied. Several possible encoding approaches have been described in [2, 8, 9]. In this paper, each set of membership functions for an item is encoded as a chromosome and handled as an individual with real-number schema.

Assume the membership functions are triangular. Three parameters are thus used to represent a membership function. Figure 2 shows an example for item I_j, where R_{jk} denotes the membership function of the k-th linguistic term and r_{jkp} indicates the p-th parameter of fuzzy region R_{jk}.

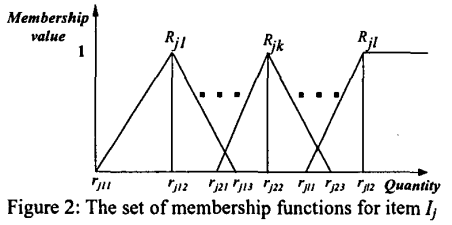

Figure 2: The set of membership functions for item I_j

Each fuzzy region R_{jk} has three parameters. The membership functions of item I_j can be represented as a string of $r_{j11}r_{j12}r_{j13}r_{j21}r_{j22}r_{j23}$ ... $r_{jl1}r_{jl2}r_{jl3}$, where $r_{jl3} = \infty$. Thus, a chromosome is encoded as a real-number string rather than a bit string. All the chromosomes in the same population have the same string length. Below, an example is given to demonstrate the process of encoding membership functions.

Example 1: Assume there are four items in a transaction database: milk, bread, cookies and beverage. Also assume a possible set of membership functions for Item *milk* is given as shown in Figure 3.

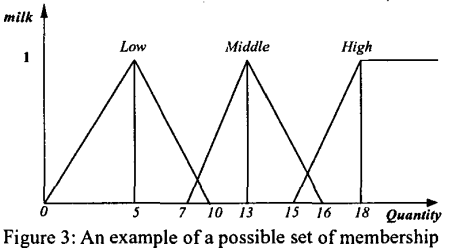

Figure 3: An example of a possible set of membership functions for Item *milk*

There are three linguistic terms, *Low*, *Middle*, and *High*, for this item. According to the proposed encoding scheme, the chromosome for representing the set of membership functions in Figure 3 is encoded as shown in Figure 4.

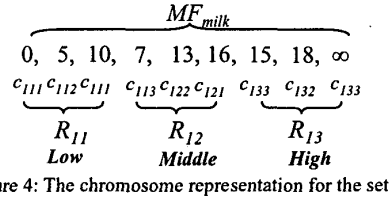

Figure 4: The chromosome representation for the set of membership functions in Figure 3

The membership function of *Low* for *milk* is encoded as (0, 5, 10) according to Figure 3. Similarly, the membership functions for *Middle* and *High* are respectively encoded as (7, 13, 16) and (15, 18, ∞). The chromosome is then the catenation of the three tuples.

According to the proposed representation, each chromosome will consist of $3*|I_j|$ real numbers for Item I_j, where $|I_j|$ is the number of linguistic terms for I_j. Since the length is short, when compared with the

other approaches in witch each chromosome consists of the membership functions for all the items, the convergence of the solutions can be easily obtained. This representation thus allows genetic operators to search for appropriate solutions quickly.

4. Mining membership functions and fuzzy association rules

4.1. Initial population

A genetic algorithm requires a population of feasible solutions to be initialized and updated during the evolution process. As mentioned above, each individual within the population is a set of triangular membership functions for a certain item. Each membership function corresponds to a linguistic term in the item. The initial set of chromosomes is randomly generated with some constraints of forming feasible membership functions.

4.2. Fitness and selection

In order to develop a good set of membership functions from an initial population, the genetic algorithm selects parent sets of membership functions with high fitness values for mating. An evaluation function is defined to qualify the derived sets of membership functions. The performance of membership function sets is then fed back to the genetic algorithm to control how the solution space is searched to promote the quality of the membership function sets. Before the fitness of each set of membership functions is formally described, several related terms are first explained below.

The overlap ratio of two membership functions R_{jk} and R_{ji} ($k < j$) is defined as the overlap length divided by the minimum of the right span of R_{jk} and the left span of R_{ji}. That is,

$$overlap\_ratio(R_{jk}, R_{ji}) = \frac{overlap(R_{jk}, R_{ji})}{\min(c_{jk3} - c_{jk2}, c_{ji2} - c_{ji1})},$$

where $overlap(R_{jk}, R_{ji})$ is the overlap length of R_{jk} and R_{ji}.

If the overlap length is larger than the minimum of the above two half spans, then these two membership functions are thought of as a little redundant. Appropriate punishment must then be considered in this case. Thus, the overlap factor of the membership functions for an item Ij in the chromosome Cq is defined as:

$$overlap\_factor(C_q) = \sum_{k \neq i} [\max((\frac{overlap(R_{jk}, R_{ji})}{\min(c_{jk3} - c_{jk2}, c_{ji2} - c_{ji1})}), 1) - 1].$$

The coverage ratio of membership functions for an item I_j is defined as the coverage range of the functions divided by the maximum quantity of that item in the transactions. The more the coverage ratio is, the better the derived membership functions are. Thus, the coverage factor of the membership functions for an item I_j in the chromosome C_q is defined as:

$$coverage\_factor(C_q) = \frac{1}{\frac{range(R_{j1}, ..., R_{jl})}{max(I_j)}},$$

where $range(R_{j1}, R_{j2}, \ldots, R_{jl})$ is the coverage range of the membership functions, l is the number of membership functions for I_j, and $max(I_j)$ is the maximum quantity of I_j in the transactions.

The suitability of the set of membership functions in a chromosome C_q is thus defined as:

$$suitability(C_q) = overlap\_factor(C_q) + coverage\_factor(C_q).$$

The fitness value of a chromosome C_q is then defined as:

$$f(C_q) = \frac{\sum_{X \in L_1} fuzzy\_support(X)}{suitability(C_q)},$$

where L_1 is the set of large 1-itemsets obtained by using the set of membership functions in C_q, and $fuzzy\_support(X)$ is the fuzzy support of the 1-itemset X from the given transaction database.

The suitability factor used in the fitness function can reduce the occurrence of the two bad kinds of membership functions shown in Figure 5, where the first one is too redundant, and the second one is too separate.

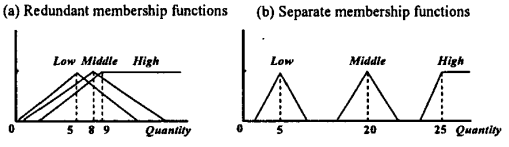

Figure 5: Two bad sets of membership functions

The overlap factor in *suitable*(C_q) is designed for avoiding the first bad case, and the coverage factor is for the second one. Below, an example is given to illustrate the above idea.

Example 2: Continuing from Example 1, assume $max(I_1) = 13$. The suitability of the chromosome C_1 for item *milk* is calculated as follows:

Suitability(C_1) =

$$\sum_{k \neq 1}[\max((\frac{overlap(R_{jk}, R_{j1})}{\min(c_{jk3} - c_{jk2}, c_{j12} - c_{j11})}), 1) - 1] + \frac{1}{\frac{range(R_{j1}, R_{j2}, R_{j3})}{\max(I_j)}}$$

$= 0 + 0 + 0 + 1$
$= 1.$

Besides, using the fuzzy-supports of the linguistic terms in the large 1-itemsets can achieve a trade-off between execution time and rule interestingness. Usually, a linguistic term of an item with a larger fuzzy-support in the 1-itemsets will usually result in its appearance in itemsets of more items with a higher probability, which will thus usually imply more interesting association rules. The evaluation by the fuzzy supports in 1-itemsets is, however, faster than that by considering all itemsets or interesting association rules.

4.3. Genetic operators

Genetic operators are very important to the success of specific GA applications. Two genetic operators, the *max-min-arithmetical (MMA) crossover* proposed in [3] and the *one-point mutation*, are used in the genetic fuzzy mining framework. Assume there are two parent chromosomes $C_u^t = (c_1, ..., c_h, ..., c_Z)$ and $C_w^t = (c_1^{'}, ..., c_h^{'}, ..., c_Z)$. The *max-min-arithmetical (MMA)* crossover operator will generate the following four candidate chromosomes from them.

1. $C_1^{t+1} = (c_{11}^{t+1}, ..., c_{1h}^{t+1}, ..., c_{1Z}^{t+1})$,
   where $c_{1h}^{t+1} = dc_h + (1-d)c_h^{'}$,
2. $C_2^{t+1} = (c_{21}^{t+1}, ..., c_{2h}^{t+1}, ..., c_{2Z}^{t+1})$,
   where $c_{2h}^{t+1} = dc_h^{'} + (1-d)c_h$,
3. $C_3^{t+1} = (c_{31}^{t+1}, ..., c_{3h}^{t+1}, ..., c_{3Z}^{t+1})$,
   where $c_{3h}^{t+1} = min\{c_h, c_h^{'}\}$,
4. $C_4^{t+1} = (c_{41}^{t+1}, ..., c_{4h}^{t+1}, ..., c_{4Z}^{t+1})$,
   where $c_{4h}^{t+1} = max\{c_h, c_h^{'}\}$,

where the parameter d is either a constant or a variable whose value depends on the age of the population. The best two chromosomes of the four candidates are then chosen as the offspring.

The one-point mutation operator will create a new fuzzy membership function by adding a random value ε (may be negative) to one parameter of an existing linguistic term, say R_{jk}. Assume that r_{jkp} represents a parameter of R_{jk}. The parameter of the newly derived membership function may be changed to $r_{jkp} + \varepsilon$ by the mutation operation. Mutation at a parameter of a fuzzy membership function may, however, disrupt the order of the resulting fuzzy membership functions. These fuzzy membership functions then need rearrangement according to their values. An example is given below to demonstrate the mutation operation.

Example 3: Continuing from Example 1, assume the mutation point is set at c_{122} and the random value ε is set at 4. The mutation process is shown in Figure 6.

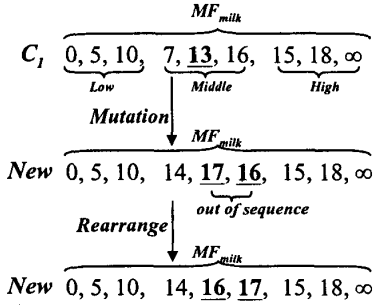

Figure 6: A mutation operation

5. The proposed mining algorithm

According to the above description, the proposed algorithm for mining both membership functions and fuzzy association rules is described below.

The proposed mining algorithm:

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

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

STEP 1: Randomly generate m populations, each for an item; Each individual in a population

represents a possible set of membership functions for that items.

STEP 2: Encode each set of membership functions into a string representation.

STEP 3: Calculate the fitness value of each chromosome in each population by the following substeps:

STEP 3.1: For each transaction datum D_i, i=1 to n, and for each item I_j, j=1 to m, transfer the quantitative value $v_j^{(i)}$ into a fuzzy set $f_j^{(i)}$ represented as:

$$\left(\frac{f_{j1}^{(i)}}{R_{j1}}+\frac{f_{j2}^{(i)}}{R_{j2}}+....+\frac{f_{jl}^{(i)}}{R_{jl}}\right),$$

using the corresponding membership functions represented by the chromosome, where R_{jk} is the k-th fuzzy region (term) of item I_j, $f_{jl}^{(i)}$ is $v_j^{(i)}$'s fuzzy membership value in region R_{jk}, and l (= $|I_j|$) is the number of linguistic terms for I_j.

STEP 3.2: For each item region R_{jk}, calculate its scalar cardinality on the transactions as follows:

$$count_{jk}=\sum_{i=1}^{n} f_{jk}^{(i)}.$$

STEP 3.3: For each R_{jk}, $1 \le j \le m$ and $1 \le k \le |I_j|$, check whether its $count_{jk}$ is larger than or equal to the minimum support threshold α. If R_{jk} satisfies the above condition, put it in the set of large 1-itemsets (L_1). That is:

$$L_1 = \{R_{jk} \mid count_{jk} \ge \alpha,\ 1 \le j \le m \text{ and } 1 \le k \le |I_j|\}.$$

STEP 3.4: Set the fitness value of the chromosome as the sum of the fuzzy supports (the scalar cardinalities / n) of the fuzzy regions in L_1 divided by $suitability(C_q)$. That is:

$$f(C_q)=\frac{\sum_{X \in L_1} fuzzy\_support(X)}{suitability(C_q)}.$$

STEP 4: Execute crossover operations on each population.

STEP 5: Execute mutation operations on each population.

STEP 6: Using the selection criteria to choose individuals in each population for the next generation.

STEP 7: If the termination criterion is not satisfied, go to Step 3; otherwise, do the next step.

STEP 8: Gather the sets of membership functions, each of which has the highest fitness value in its population.

The sets of the best membership functions gathered from each population are then used to mine fuzzy association rules from the given quantitative database. Our fuzzy mining algorithm proposed in [5] is then adopted to achieve this purpose.

6. An example

In this section, 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 shown in Table 1.

Table 1. Six transactions in this example

TID	Items
T1	(milk, 5); (bread, 10); (cookies, 7); (beverage, 7).
T2	(milk, 7); (bread, 14); (cookies, 12).
T3	(bread, 15); (cookies, 12); (beverage, 10).
T4	(milk, 2); (bread, 5); (cookies, 5).
T5	(bread, 9).
T6	(milk, 13); (beverage, 12).

Assume each item has three fuzzy regions: *Low*, *Middle*, and *High*. Thus, three fuzzy membership functions must be derived for each item. Assume the population size is 10. For the data shown in Table 1, the proposed mining algorithm generates the membership functions shown in Figure 7.

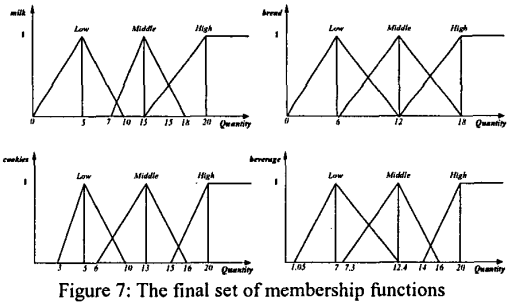

Figure 7: The final set of membership functions

After the membership functions are derived, the fuzzy mining method proposed in [5] is then used to mine fuzzy association rules from the quantitative database.

7. Conclusion and future works

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 the fuzzy-supports of the linguistic terms in the large 1-itemsets and the suitability of the derived membership functions, the derivation process can easily be done by the divide-and–conquer strategy. Our approach can reduce human experts' intervention during the mining process, thus saving much acquisition time. In the future, we will continuously attempt to enhance the GA-based mining framework for more complex problems.

Acknowledgment

The authors would like to thank the anonymous referees for their very constructive comments. This research was supported by the National Science Council of the Republic of China under contract NSC92-2213-E-390-001.

References

[1] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," *The International Conference on Very Large Databases*, 1994, pp. 487-499.

[2] O. Cordón, F. Herrera, and P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, 2001.

[3] F. Herrera, M. Lozano and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets and Systems*, Vol. 92, No. 1, pp. 21–30, 1997.

[4] T. P. Hong, C. S. Kuo and S. C. Chi, "Mining association rules from quantitative data", *Intelligent Data Analysis*, Vol. 3, No. 5, 1999, pp. 363-376.

[5] T. P. Hong, C. S. Kuo and S. C. Chi, "Trade-off between time complexity and number of rules for fuzzy mining from quantitative data," *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 9, No. 5, 2001, pp. 587-604.

[6] M. Kaya, and R. Alhajj, "A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining," The IEEE International Conference on Fuzzy Systems, 2003, pp. 881 -886.

[7] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," The 1996 ACM SIGMOD International Conference on Management of Data, Monreal, Canada, June 1996, pp. 1-12.

[8] C. H. Wang, T. P. Hong and S. S. Tseng, "Integrating fuzzy knowledge by genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 2, No.4, pp. 138-149, 1998.

[9] C. H. Wang, T. P. Hong and S. S. Tseng, "Integrating membership functions and fuzzy rule sets from multiple knowledge sources," *Fuzzy Sets and Systems*, Vol. 112, pp. 141-154, 2000.

[10] W. Wang and S. M. Bridges, "Genetic algorithm optimization of membership functions for mining fuzzy association rules," The International Joint Conference on Information Systems, Fuzzy Theory and Technology, 2000, pp. 131-134.