2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A Cluster-Based Fuzzy-Genetic Mining Approach for Association Rules and Membership Functions

Chun-Hao Chen, Tzung-Pei Hong, and Vincent S. Tseng

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. Designing a sophisticated data-mining algorithm able to deal with various types of data presents a challenge to workers in this research field. In this paper, a cluster-based fuzzy-genetic mining algorithm is proposed for extracting both fuzzy association rules and membership functions from quantitative transactions. The proposed algorithm can dynamically adjust membership functions by genetic algorithms and uses them to fuzzify quantitative transactions. It can also speed up the evaluation process and keep good quality of solutions by clustering chromosomes. Experimental results show the effectiveness of the proposed approach.**

I. INTRODUCTION

Data mining is most commonly used in attempts to induce association rules from transaction data. An association rule is an expression X→Y, where X is a set of items and Y is a single item. It means in the set of transactions, if all the items in X exist in a transaction, then Y is also in the transaction with a high probability. 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.

Recently, fuzzy set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [14]. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [4, 14, 17, 25]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [6, 7, 8, 12, 13].

As to fuzzy data mining, Hong *et al.* proposed several algorithms to mine fuzzy rules from quantitative data [10, 11]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Cai *et al.* proposed weighted mining to reflect different importance to different items [2]. Each item was attached a numerical weight given by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue *et al.* then extended their concepts to fuzzy item vectors [26]. Besides, Lee *et al.* proposed a mining algorithm which used multiple minimum supports of different items to mine fuzzy association rules [16]. In the above approaches, the membership functions were assumed to be known in advance. Although many approaches for learning membership functions were proposed [3, 19, 20, 22, 23], most of them were usually used for classification or control problems. Wang *et al.* tuned membership functions for intrusion detection systems based on similarity of association rules [24]. Kaya *et al.* 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 [15]. Hong *et al.* also proposed a GA-based fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions [9]. That algorithm could dynamically adapt membership functions by genetic algorithms and used them to fuzzify quantitative transactions. Its fitness function was evaluated by the number of large 1-itemsets and the suitability of membership functions. The calculation for large 1-itemsets, however, took a lot of cost, especially when the database to be scanned could not totally fed into main memory.

In this paper, an enhanced approach, called cluster-based fuzzy-GA mining algorithm, is proposed to speed up the evaluation process and keep nearly the same quality of solutions. Each chromosome represents a set of membership functions used in fuzzy mining. The proposed algorithm first divides the chromosomes in a population into *k* clusters by using the *k*-means clustering approach. All the chromosomes in a cluster then use the number of large 1-itemsets derived from the representative chromosome in the cluster and their own suitability of membership functions to calculate the fitness values. The evaluation cost can thus be greatly reduced due to the time-saving in finding 1-itemsets. Experimental results also show the effectiveness of the proposed algorithm.

II. A CLUSTER-BASED FUZZY-GA MINING FRAMEWORK

In this paper, the fuzzy, GA and clustering concepts are used to discover both useful fuzzy association rules and suitable membership functions from quantitative transactions.

Chun-Hao Chen is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, 701, Taiwan. (e-mail: chchen@idb.csie.ncku.edu.tw).

Tzung-Pei Hong is with the Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, 811, Taiwan. (corresponding author; phone: +886+7+5919191; fax: +886+7 +5919374; e-mail: tphong@nuk.edu.tw).

Vincent S. Tseng is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, 701, Taiwan. (e-mail: tsengsm@mail.ncku.edu.tw).

0-7803-9489-5/06/$20.00/©2006 IEEE

A cluster-based fuzzy-GA mining framework is first proposed for searching membership functions suitable for mining problems and for using the final best set of membership functions to mine fuzzy association rules. The proposed framework is shown in Fig. 1.

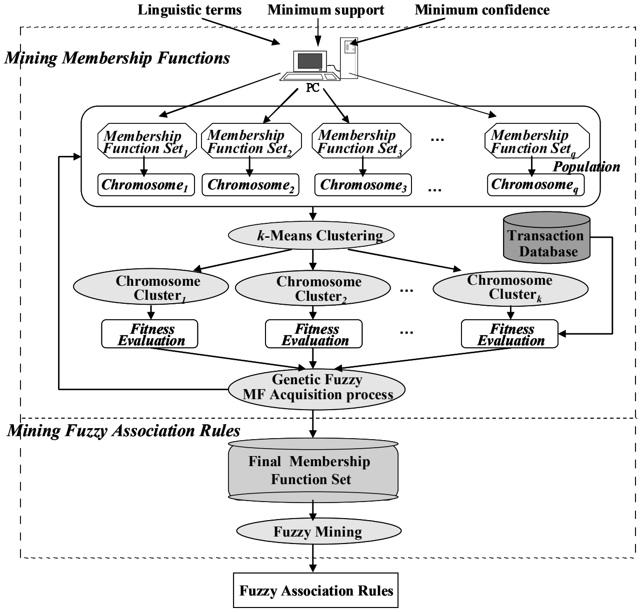

Fig. 1. The proposed cluster-based fuzzy-GA mining framework

The proposed framework maintains a population of sets of membership functions, and uses the genetic algorithm to automatically derive the resulting one. It first transforms each set of membership functions into a fixed-length string. It then uses the *k*-means clustering approach to gather similar chromosomes into groups. All the chromosomes in a cluster use the number of large 1-itemsets derived from the representative chromosome in the cluster and their own suitability of membership functions to calculate their fitness values. Since the number for scanning a database decreases, the evaluation cost can thus be reduced. The evaluation results are utilized to choose appropriate chromosomes for mating in the next generation. The offspring membership function sets then undergo recursive "evolution" until a good set of membership functions has been obtained. Finally, the derived membership functions are used to mine fuzzy association rules.

III. 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 [3, 18, 22, 23]. In this paper, each set of membership functions is encoded as shown in Fig. 2. It is the same representation as that in our previous paper [9] for a fair comparison later.

In Fig. 2, each membership function is assumed to be isosceles-triangle and represented by a pair (c, w), with c indicating the center abscissa and w representing half the spread. R_{jk} denotes the membership function of the k-th linguistic term of *item* I_j. All pairs of (c, w)'s for a certain item are concatenated to represent its membership functions. Since c and w are both numeric values, a chromosome is thus encoded as a fixed-length real-number string rather than a bit string.

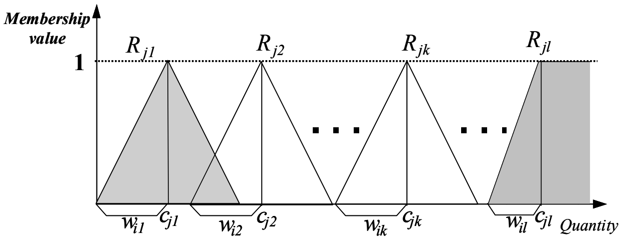

Fig. 2. Membership Functions of item I_j

Note that other types of membership functions (e.g. non-isosceles trapezes) can also be adopted in our method. For coding non-isosceles triangles and trapezes, three and four points are needed instead of two for isosceles triangles.

IV. MINING MEMBERSHIP FUNCTIONS AND ASSOCIATION RULES

A. 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 isosceles-triangular membership functions. Each membership function corresponds to a linguistic term in a certain item. The initial set of chromosomes is randomly generated with some constraints for forming feasible membership functions.

B. Fitness and Selection

In order to develop a good set of membership functions from an initial population, the genetic algorithm selects *parent* membership function sets with its probability values for mating. An evaluation function is then used to qualify the derived membership function sets. Our previous fitness function defined in [9] is used for a fair comparison. It is shown as follows:

$$f(C_q) = \frac{|L_{1q}|}{suitability(C_q)},$$

where $|L_{1q}|$ is the number of large 1-itemsets obtained by using the set of membership functions in chromosome C_q and suitability(C_q) represents the shape suitability of C_q. Suitability(C_q) is defined as:

$$\sum_{j=1}^{m}[\,overlap\_factor(C_{qj}) + coverage\_factor(C_{qj})],$$

where m is the number of items. Overlap_factor(C_{qj}) represents the overlap factor of the membership functions for an item I_j in the chromosome C_q and is defined as:

$$overlap\_factor(C_{qj}) = \sum_{k \neq i}[\max((\frac{overlap(R_{jk}, R_{ji})}{\min(w_{jk}, w_{ji})}), 1) - 1],$$

where *overlap*(R_{jk}, R_{ji}) is the overlap length of R_{jk} and R_{ji}. Coverage_factor(C_{qj}) represents the coverage ratio of a set of membership functions for an item I_j and is defined as:

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

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 factor used in the fitness function can reduce the occurrence of the two bad kinds of membership functions shown in Fig. 3, where the first one is too redundant, and the second one is too separate.

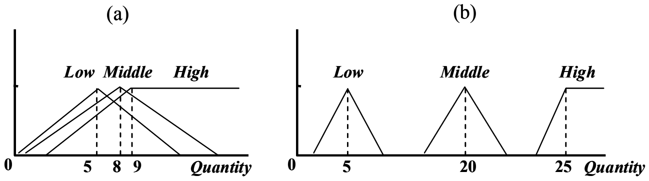

Fig. 3. Two bad membership functions

C. Clustering Chromosomes

Although the evaluation by 1-itemsets is much faster than that by all itemsets or interesting association rules, it is still time-consuming since the database must be scanned once for each chromosome. In this paper, we thus propose a new method based on clustering technique to reduce the evaluation time of large 1-itemsets. The process proceeds as follows. The coverage factors and overlap factors of all the chromosomes are used to form appropriate clusters. The *k*-means clustering approach is adopted here to cluster chromosomes. Since the chromosomes with similar coverage factors and overlap factors will form a cluster, they will have nearly the same shape of membership functions and induce about the same number of large 1-itemsets. For each cluster, the chromosome which is the nearest to the cluster center is thus chosen to derive its number of large 1-itemsets. All chromosomes in the same cluster then use the number of large 1-itemsets derived from the representative chromosome as their own. Finally, each chromosome is evaluated by this number of large 1-itemsets divided by its own suitability value. The details of the process will be further illustrated later.

D. 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 [5] and the *one-point mutation*, are used in the genetic-fuzzy mining framework. Assume there are two parent chromosomes:

$$C_u^t = (c_1, ..., c_h, ..., c_Z) \text{, and}$$

$$C_w^t = (c_1^{'}, ..., c_h^{'}, ..., c_Z^{'}) .$$

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 (between $-w_{jk}$ to $+ w_{jk}$) to the center or to the spread of an existing linguistic term, say R_{jk}. Assume that c and w represent the center and the spread of R_{jk}. The center or the spread of the newly derived membership function will be changed to $c + \varepsilon$ or $w + \varepsilon$ by the mutation operation. Mutation at the center 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 center values.

V. THE PROPOSED MINING ALGORITHM

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

The cluster-based fuzzy-genetic mining algorithm:

```
INPUT: A body of $n$ quantitative transactions, a set of $m$
       items, each with a number of linguistic terms, a
       parameter $k$ for $k$-means clustering, a population size
       $P$, a crossover rate $P_c$, a mutation rate $P_m$, a support
       threshold $\alpha$, and a confidence threshold $\lambda$.
OUTPUT: A set of fuzzy association rules with its associated
        set of membership functions.
STEP 1: Randomly generate a population of $P$ individuals;
        each individual is a set of membership functions for
        all the $m$ items.
STEP 2: Encode each set of membership functions into a
        string representation according to the schema stated
        in Section III.
STEP 3: Calculate the coverage_factor and the overlap_factor
        of each chromosome using the formulas defined in
        Section IV-B.
STEP 4: Divide the chromosomes into $k$ clusters by the
        $k$-means clustering approach based on the two
        attributes (coverage_factors and overlap_factors);
        Find out the representative chromosome in each
        cluster, which is the nearest to the center.
STEP 5: Calculate the number of large 1-itemsets for each
        representative chromosome by the following
        substeps.
  STEP 5.1: 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 representative chromosome, where $v_j^{(i)}$ is the quantitative value of I_j in D_i, 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 5.2: For each item region R_{jk}, calculate its count from the transactions as follows:

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

STEP 5.3: 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 5.4: Set $|L_1|$ as the number of large 1-itemsets for the representative chromosome.

STEP 6: Calculate the fitness value of each chromosome using the number of large 1-itemsets of its representative chromosome and the suitability value of its own according to the formula defined in Section IV-B.

STEP 7: Execute the crossover operation on the population.

STEP 8: Execute the mutation operation on the population.

STEP 9: Use the Roulette-wheel selection operation to choose appropriate individuals for the next generation.

STEP 10: If the termination criterion is not satisfied, go to Step 3; otherwise, do the next step.

STEP 11: Get the set of membership functions with the highest fitness value.

STEP 12: Mine fuzzy rules using the set of membership functions.

The set of membership functions are then used to mine fuzzy association rules from the given database. Our fuzzy mining algorithm proposed in [11] is then adopted to achieve this purpose.

VI. EXPERIMENTAL RESULTS

In this section, experiments made to show the performance of the proposed approach are described. They were implemented in Java on a personal computer with Intel Pentium IV 3.20GHz and 512MB RAM. 64 items and 10000 transactions were used in the experiments. In each data set, the numbers of purchased items in transactions were first randomly generated. The purchased items and their quantities in each transaction were then generated. An item could not be generated twice in a transaction. The initial population size P is set at 50, the number of cluster k is set at 3, the crossover rate p_c is set at 0.8, and the mutation rate p_m is set at 0.001. The parameter d of the crossover operator is set at 0.35 according to Herrera *et al.*'s paper [5] and the minimum support α is set at 0.05 (5%).

After 500 generations, the final membership functions are apparently much better than the original ones. For example, the initial membership functions of some four items among the 64 items are shown in Fig. 4.

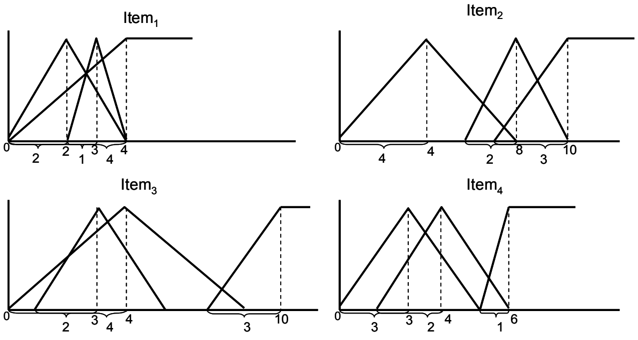

Fig. 4. The initial membership functions of some four items

In Fig. 4, the membership functions have the two bad types of shapes according to the definition in the previous section. The membership functions for $Item_1$, $Item_2$ and $Item_4$ overlap too much. After 500 generations, the final membership functions for the same four items are shown in Fig. 5.

It is easily seen that the membership functions in Fig. 6 is better than those in Fig. 4. The two bad kinds of membership functions didn't appear in the final results.

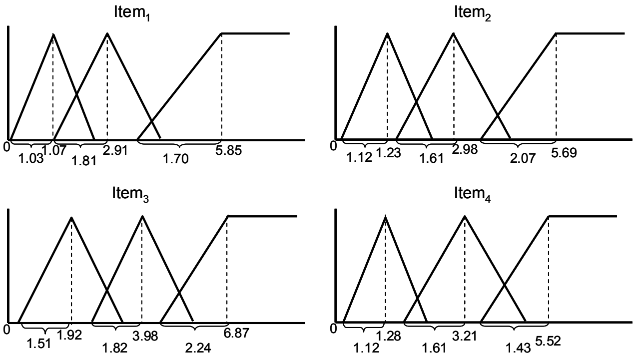

Fig. 5. The final membership functions of some four items after 500 generations

The average fitness values of the chromosomes along with different numbers of generations are shown in Fig. 6. As expected, the curve gradually goes upward, finally converging to a certain value.

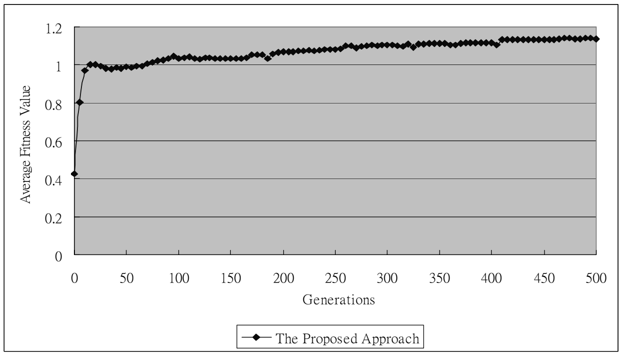

Fig. 6. The average fitness values along with different numbers of generations

Next, experiments were made by using only suitability and only $|L_1|$ as the fitness functions to show the validity of the proposed one. For the same experimental environments and data, the membership functions of the above four items after 500 generations by using only suitability as the fitness function are shown in Fig. 7, and by using only the number of large 1-itemsets are shown in Fig. 8.

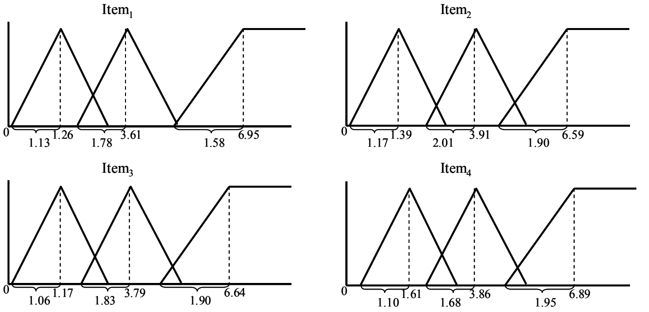

Fig. 7. The final membership functions when only the suitability is considered

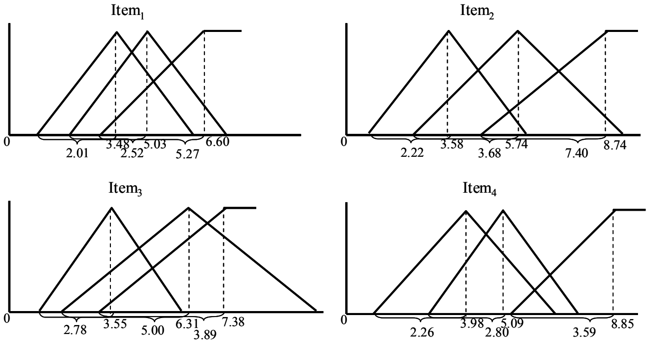

Fig. 8. The final membership functions when only $|L_1|$ is considered

It can be easily seen from Fig. 7 that the derived membership functions by considering only suitability are satisfactory because the suitability measure is designed for getting good shapes of membership functions. Its number of large 1-itemsets is, however, less than the original one. On the contrary, it is very natural for the derived membership functions by considering only the number of large 1-itemsets to have a bad shape from Fig. 8. Their overlapping degrees are quite high. Next, experiments were made to compare the proposed method with our previous one [9] for showing the effect of using clusters in evaluation. The average fitness values of the chromosomes along with different numbers of generations are shown in Fig. 9.

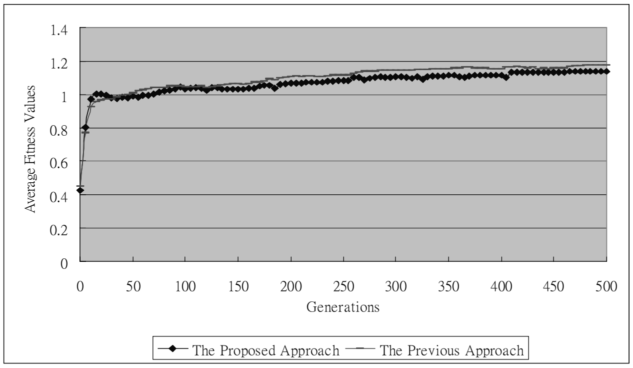

Fig. 9. The comparison results between the proposed and previous approach

From Fig. 9., it can be easily observed that the average fitness values of the proposed approach was only a little less than the previous one, but very close. The results are reasonable since the proposed approach used the clustering technique to estimate the number of large 1-itemsets for the chromosomes in the same cluster. The comparison for the execution time of the two approaches is shown in Table I.

TABLE I

COMPARISON FOR THE EXECUTION TIME OF THE TWO APPROACHES

	Population Size	Cluster Number	Execution Time (Minutes)	Speed-up Ratio
The proposed approach	50	3	42.95	3.83
The previous approach	50	null	164.71	1

From Table I, it can be seen that the proposed approach ran nearly four times faster than the previous one. It can be concluded that the proposed approach can not only get nearly the same fitness values as the previous approach but also greatly reduce the execution time. The proposed approach can thus get a good trade-off between accuracy and execution time.

At last, experiments were made for providing a comparative analysis of the proposed approach with the fuzzy mining approach in [11] with uniform fuzzy partition. The relationship between the numbers of large 1-itemsets and the minimum supports for the proposed approach and the one with uniform fuzzy partition is shown in Fig. 10.

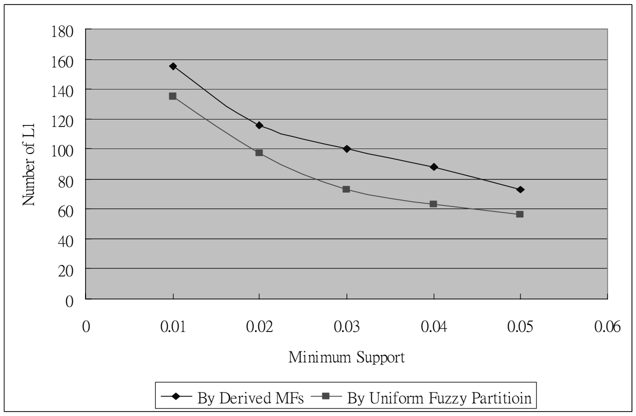

Fig. 10. The relationship between the numbers of large 1-itemsets and the minimum supports for the two approaches

It can be observed from Fig. 10. that the number of large 1-itemsets derived by the proposed algorithm was larger than the one with uniform fuzzy partition. It is consistent with the previous discussion since the adopted fitness function will help the proposed approach search for a bigger number of large 1-itemsets.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a cluster-based fuzzy-genetic mining algorithm for extracting both fuzzy association rules and membership functions from quantitative transactions. The proposed algorithm can dynamically adjust membership functions by genetic algorithms and uses them to fuzzify quantitative transactions. It can also speed up the evaluation process and keep nearly the same quality of solutions by clustering chromosomes. Each chromosome represents a set of membership functions used in fuzzy mining. The proposed algorithm first divides the chromosomes in a population into k clusters by using the k-means clustering approach. All the chromosomes in a cluster then use the number of large 1-itemsets derived from the representative chromosome in the cluster and their own suitability of membership functions to calculate the fitness values. The evaluation cost can thus be greatly reduced due to the time-saving in finding 1-itemsets. Experimental results first show that the adopted fitness function can achieve a good trade-off between numbers of large itemsets and suitability of membership functions. They then show that using the clustering technique to speed up the evaluation process can not only get nearly the same fitness values as the previous approach but also greatly reduce the execution time. The proposed approach can thus get a good trade-off between accuracy and execution time. In the future, we will continuously attempt to enhance the GA-based mining framework for more complex problems.

ACKNOWLEDGMENT

This research was supported by the National Science Council of the Republic of China under contract NSC94-2213-E-390-005.

REFERENCES

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

[2] C. H. Cai, W. C. Fu, C. H. Cheng and W. W. Kwong, “Mining association rules with weighted items,” *The International Database Engineering and Applications Symposium*, pp. 68-77, 1998.

[3] 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.

[4] I. Graham and P. L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision*, Chapman and Computing, Boston, pp.117-158, 1988.

[5] 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.

[6] T. P. Hong and J. B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems*, Vol.103, No. 3, pp.389-404, 1999.

[7] T. P. Hong and J. B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," *Fuzzy Sets and Systems*, Vol. 112, No. 1, pp. 127-140, 2000.

[8] R. H. Hou, T. P. Hong, S. S. Tseng and S. Y. Kuo, “A new probabilistic induction method, ” *Journal of Automatic Reasoning*, Vol. 18, pp. 5-24, 1997.

[9] T. P. Hong, C. H. Chen, Y. L. Wu and Y. C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions", accepted and to appear in *Soft Computing*.

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

[11] 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, pp. 587-604, 2001.

[12] T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems*, Vol. 84, pp. 33-47, 1996.

[13] T. P. Hong and S. S. Tseng, “A generalized version space learning algorithm for noisy and uncertain data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 2, pp. 336-340, 1997.

[14] A. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, pp.8-19, 1992.

[15] 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*, pp. 881 -886, 2003.

[16] Y. C. Lee, T. P. Hong and W. Y. Lin, "Mining fuzzy association rules with multiple minimum supports using maximum constraints", *Lecture Notes in Computer Science*, Vol. 3214, pp. 1283-1290, 2004.

[17] E. H. Mamdani, “Applications of fuzzy algorithms for control of simple dynamic plants, “ *IEEE Proceedings*, pp.1585-1588, 1974.

[18] A. Parodi and P. Bonelli, "A new approach of fuzzy classifier systems," *Proceedings of Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, Los Altos, CA, pp. 223-230, 1993.

[19] H. Roubos and M. Setnes, “Compact and transparent fuzzy models and classifiers through iterative complexity reduction,” *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 4, pp. 516-524, 2001.

[20] M. Setnes, H. Roubos, “GA-fuzzy modeling and classification: Complexity and performance,” *IEEE Transaction on Fuzzy System*, Vol. 8, No. 5, pp. 509-522, 2000.

[21] M. Srinivas and L.M. Patnaik, “Genetic algorithms: a survey,” *Computer*, Vol. 27, No. 6, pp. 17-26, 1994.

[22] 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.

[23] 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.

[24] 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*, pp. 131-134, 2000.

[25] R.Weber, “Fuzzy-ID3: a class of methods for automatic knowledge acquisition,” *The second International Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp. 265-268, 1992.

[26] S. Yue, E. Tsang, D. Yeung and D. Shi, “Mining fuzzy association rules with weighted items,” *The IEEE International Conference on Systems, Man and Cybernetics*, pp. 1906-1911, 2000.