An Evolutionary Data Mining Model for Fuzzy Concept Extraction

Mohammad Amin Rigi
Department of Computer Eng.
Ferdowsi University, Mashhad, Iran
Amin.rigi@gmail.com

Amin Milani Fard
Department of Computer Eng.
Ferdowsi University, Mashhad, Iran
milanifard@ieee.org

Mohammad –R. Akbarzadeh –T.
Department of Electrical Eng.
Ferdowsi University, Mashhad, Iran
akbarzadeh@ieee.org

Abstract

Considering the fast growth of data contents in terms of size as well as variety, finding useful information from collections of data have been extensively investigated in the past decade. In this paper a method is proposed for extracting useful information from a relational database using a hybrid of genetic algorithm and Fuzzy data mining approach to extract user desired information. The genetic algorithm is employed to find a compact set of useful fuzzy concepts with a good fuzzy support for the output of fuzzy data mining process. Experimental results show superiority of the proposed evolutionary system as compared to the common fuzzy grid-based data mining.

1. Introduction

Data mining methods aim at effectively helping users to get their desired information from large amounts of data [1][2]. Due to modern information technologies it is now possible to collect, store, transfer, and combine huge amounts of data at very low costs. Thus an ever-increasing number of users can afford building up large archives of documents and other data. However, exploiting the information contained in these archives in an intelligent way turns out to be fairly difficult. In contrast to the abundance of data there is a lack of tools that can transform these data into useful information and knowledge [3].

Since the fuzzy set theory deals with cognitive uncertainty, including vagueness and ambiguity [4], fuzzy data mining [5] is considered to be useful in discovering fuzzy concepts. As number of linguistic values for each attribute is required to generate useful fuzzy concepts, the genetic algorithm is employed to automatically determine the appropriate partition number for each attribute.

In the GA, the fitness function is to evaluate usefulness of fuzzy concepts with user specified dimension, which means number of attributes or linguistic variables that are extracted with fuzzy data mining method. In the GA fitness function we have a parameter that specifies this dimension. Here the proposed method consists of two phases to generate learning sequences. The first phase finds a needed competence set consisting of useful patterns by an algorithm to find frequent and necessary fuzzy grids, which is a significant part of the fuzzy grids based rules mining algorithm (FGBRMA) [5][6][7]. In common fuzzy data mining systems users do not have access to output of the system; therefore we have proposed a new method for the second phase to affect the output of such a system. In the second phase, the GA finds the best way that the information can be described (the way user wants the information to be), by choosing the way of partitioning the linguistic variables.

In the following sections, we first introduce the concept of fuzzy partition in the proposed data mining technique, and also the cases for partitioning attributes in Section 2. Section 3 presents the determination of useful patterns. In Section 4 we propose our evolutionary method. Implementation and results is described in Section 5, and finally conclusion is presented in Section 6.

2. Related works

Fuzzy sets concept and applications in approximate reasoning were originally proposed by Zadeh [8]. Formally a linguistic variable is denoted by (x; $T(x)$; U; G; M), where x is the variable name; $T(x)$ is the term set of x, that is a set of linguistic values or terms in natural language of x; U is the universe of discourse; G is syntactic rule for generating values of x; and M is

978-1-4244-1841-1/08/$25.00 ©2008 IEEE

semantic rule for associating a linguistic value with a meaning [9]. As an example for a linguistic variable of "*Age*", *T*(*Age*)={*young*, *close_to_30*, *close_to_50*, *old*}, *G* is a rule which generates the linguistic values in *Age*, *U*=[0,60], and *M*(*young*) assigns a membership function to "young" value.

A relational database is a collection of tables, each of which is assigned a unique name and consists of a set of attributes and stores a large set of records [10]. Each attribute is considered as a linguistic variable here. Let *d* be number of attributes in a database, then a *d*-dimensional feature space is constructed and each attribute is viewed as an axis of this space. Some cases for partitioning quantitative and categorical attributes by linguistic values are introduced as follows.

A quantitative attribute can be partitioned by $K \geq 2$ linguistic values. The value of K is dependent on the actual requirement or preference of system users. For example, $K = 2$ and $K = 3$ for the linguistic variable "Age" are depicted in Fig. 1. Triangular membership functions are used for each linguistic value defined in each quantitative attribute for simplicity. A linguistic value denoted by $A_{K,i}^{Age}$ can be described in a linguistic sentence.

By partitioning the universe of discourse for each linguistic variable, $K_1 \times K_2 \times \cdots \times K_d$ fuzzy grids with d dimensions in the pattern space can be obtained. Particularly, this paper views a fuzzy grid as a fuzzy concept. For example, consider two linguistic variables in a medical database "*Weight*" and "*Age*." We assume *T(Weight)*={*light*, *average*, *heavy*} and *T(Age)*= {*young*, *average*, *old*}. Fig. 2 shows the fuzzy grid partitioning of the above example. The shaded 2-D grid implies "heavy AND old" grid.

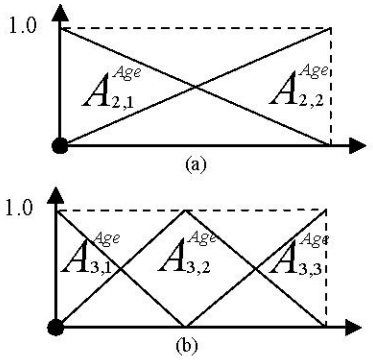

Fig. 1. Partitioning "Age" linguistic value (a) K=2, (b) K=3

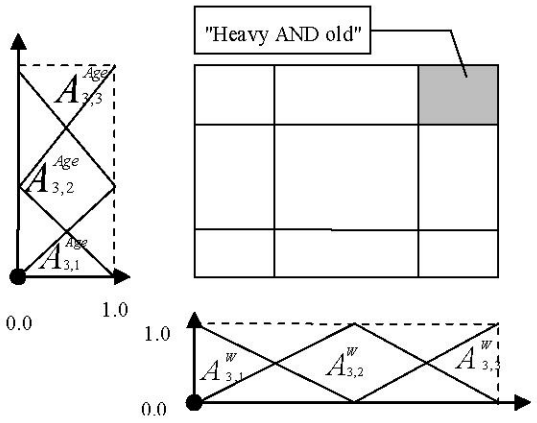

Fig. 2. Partitioning "Weight" and "Age" into three parts

3. Finding useful fuzzy concepts

In order to find frequent fuzzy grids we use part of the method proposed in [5]. Suppose each quantitative attribute X_m is partitioned by K linguistic values, and let the universe of discourse $U=\{t_1, t_2, \ldots, t_n\}$, then the linguistic value that is assigned to a candidate's k-dimensional ($1 \leq k \leq d$) fuzzy grid can be represented as $\left(A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}\right) = \sum_{p=1}^{n} \mu_{(t_p)} \left(A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}\right) / t_p$

The degree to which t_p belongs to the linguistic value $\left(A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}\right)$ is then computed using any T-norm operator. In this work we use algebraic product as follows: $\mu_{K,j1}^{x1}(t_p) * \mu_{k,j2}^{x2}(t_p) * \cdots * \mu_{k,in}^{xn}(t_p)$.

To check whether or not this fuzzy grid is frequent, we define the *fuzzy support concept* of $A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}$ as follows:

$$FS\left(A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}\right) = \left[\sum_{p=1}^{n} \mu_{K,j1}^{x1}(t_p) * \mu_{k,j2}^{x2}(t_p) * \cdots * \mu_{k,in}^{xn}(t_p)\right] / n$$

,where n is the number of samples.

When $FS\left(A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}\right)$ is higher than or equal to the user-specified minimum fuzzy support threshold (*minFS*), we can say that $A_{K,j1}^{x1} \wedge A_{K,j2}^{x2} \wedge \cdots \wedge A_{K,jk}^{xk}$ is a frequent k-dimensional fuzzy grid. Obviously, *FS* is between 0 and 1. A frequent fuzzy grid actually shows a useful pattern discovered from a relational database by the proposed data mining technique. As an example, consider having four patterns (or four personal information in a medical database), each of which has two linguistic variables. Similar to Fig. 1, for both of linguistic variables, three linguistic values are defined on interval [0,100] (the

universe of discourse is [0,100]). The corresponding data is summarized in Table 1.

Table 1: Four patterns with two linguistic variables

	Attributes							
	Age				Weight			
Patterns	value	$\mu_{1,1}$	$\mu_{1,2}$	$\mu_{1,3}$	value	$\mu_{2,1}$	$\mu_{2,2}$	$\mu_{2,3}$
t_1	82	0.0	0.36	0.64	35	0.30	0.70	0.00
t_2	65	0.00	0.70	0.30	70	0.00	0.60	0.40
t_3	43	0.12	0.88	0.00	49	0.02	0.98	0.00
t_4	20	0.60	0.40	0.00	12	0.76	0.24	0.00

If we assume *x1* as *Age* and *x2* as *Weight* then $FS(A_{3,j}^{xi})$ can be easily computed, where i=1,2 and j=1,2,3,. For example: $FS(A_{3,2}^{x1})$ = (0.36 + 0.70 + 0.88 + 0.40) / 4 (i.e., 0.585). If the *minFS* is 0.20, then $A_{3,2}^{x1}$ is considered as a useful fuzzy concept. $(A_{3,2}^{x1} \wedge A_{3,1}^{x2})$ is a valid 2-D fuzzy concept by joining two useful 1-D fuzzy concepts, and its fuzzy support FS($A_{3,2}^{x1} \wedge A_{3,1}^{x2}$) is computed as follows:
(0.36*0.30+0.70*0.00+0.88*0.02+0.40*0.76)/4=107.

It is notable that not all fuzzy grids can be joined to make higher dimensional grids. For example $(A_{3,1}^{x1} \wedge A_{3,2}^{x2} \wedge A_{3,3}^{x2})$ is an invalid fuzzy concept since both $A_{3,2}^{x2}$ and $A_{3,3}^{x2}$ are defined in X_2.

4. The proposed approach

As mentioned above, it is necessary for users to specify the number of linguistic values K_i, to determine useful fuzzy concepts. However, users might not be specialized enough to determine those parameters. Sometimes it is more important for the user to get high dimensional information and sometimes not. For example we have these two concepts:

a. Age of customer is about 30 years.
b. Income of employee is about 30000, experience is about 10 years and sexuality is male.

We assume that both of the above fuzzy concepts have suitable fuzzy support. We can see that the first case consists of one attribute (linguistic variable), and the second one has three attributes. Sometimes it is important for system users (decision makers) to extract information like case "a" and sometimes like case "b". But as mentioned before, the fuzzy data mining task is to extract concepts like these. In fuzzy data mining system the inputs are minFS, linguistic values of an attribute, and a database for mining. Obviously, dimensions of output fuzzy concepts are not specified by the user. To determine the dimension of output Genetic Algorithm (GA) is used.

The basic concept of the GAs were originally developed by Holland [12] and later revised by Goldberg [13]. Goldberg showed that GAs are independent of any assumption about the search space and are based on the mechanism of natural genetics. The first step to model this problem as a GA problem, is determining the interpretation function (chromosome design), GA operators, and fitness function.

4.1. Chromosome Design

An initial population containing N_{pop} chromosomes is generated. Each gene in the binary chromosome is randomly assigned as either 1 or 0, with a probability of 0.5. The j[th] chromosome ($1 \le j \le N_{pop}$) is represented by $l_j^1 l_j^2 l_j^3 \cdots l_j^d$, where l denote the substrings that encode the number of linguistic values K_i in i[th] attribute, and d is number of linguistic variables, respectively. So each chromosome shows a way of fuzzy grid partitioning in the database. For example the j[th] chromosome l_j^1 implies number of first attribute in the database.

4.2 GA operations

Since the chromosomes are binary coded, simple one point crossover has been used as crossover operation. One point in the selected chromosome would be selected along with a corresponding point in another chromosome and then the tails would be exchanged. For each gene of the newly generated binary chromosomes in N_{pop}, the mutation operation with probability P_m, is performed on each bit or gene of each binary string. Mutation processes causes some bits to invert and produce some new information. The only problem of mutation is that it may cause some useful information to be corrupted. Therefore we used elitism which means the best individual will go forward to the next generation without undergoing any change to keep the best information. For reproduction operation, $N_{pop}/2$ of chromosomes with higher fitness will be selected to make next population. We use weighted average method [14] for generating the next population.

4.3 Fitness function

Defining fitness function is one of the most important steps in designing a GA-based method, which can guide the search toward the best solution. We consider dimension factor and the fuzzy support of a chromosome, as two important parameters of the

fitness function. The dimension factor shows the interest of user to extract information with fewer or more attributes. We can see the algorithm for the fitness function in Fig. 3.

```
1 DO fuzzy grid partitioning with the
      chromosome values
2 dimension factor=user defined value
3 Fitness=0
4 factor =1
5 DO
6  Fitness = Fitness+
    (factor*average of fuzzy support of
     useful fuzzy concepts)
7  factor = factor * dimension factor
8  make higher dimension fuzzy
   grids
9 WHILE higher dimension fuzzy grids exists
```

Fig. 3. Algorithm for computing fitness function

The formula of fitness function is shown below:

$$\text{Fitness} = \sum_{1}^{m} \left(\frac{f^m}{n} \sum_{1}^{n} FS(m-D) \right)$$

where m is number of dimension of useful concepts that we have extracted or number of do-while loop in figure 5. f is dimension factor and n is number of fuzzy concepts. And $\frac{1}{n}\sum_{1}^{n} FS(m-D)$ is the average of m-dimensional useful fuzzy concepts.

It is obvious that if the dimension factor has a higher value, higher dimensional information will be more important to extract from database. In the first stage the fitness value equals the average of one-dimension grids fuzzy support. In the next iteration of the do-while loop, the dimension factor affects the fitness value. If the value of dimension factor is larger than 1 it means that making higher dimension fuzzy concepts are more important. The bigger dimension factor is, the more important the higher dimension fuzzy grids would be. If the dimension factor is smaller than 1 it means low dimension concepts are more important.

4.4 Algorithm design

In the following algorithm each chromosome indicates a way of partitioning. First an initial population is generated and then for each chromosome the fitness is computed via fuzzy data mining. After calculating the fitness function value for each parent chromosome the algorithm will generate N children. The higher a parent chromosome's fitness function value is the higher probability it has to contribute one or more offspring in the next generation. After performing genetic operations, if some chromosomes do not satisfy the fitness, the algorithm discards this process and gets M ($M \leq N$) children chromosomes. The algorithm then selects N chromosomes with the lower fitness value from the $M+N$ chromosomes (M children and N parents) to be parents of the next generations. This process would repeat until a certain number of generations are processed, after which the best chromosome is chosen. Fig. 4 shows our fuzzy grid partitioning optimizer GA approach.

```
input: system output dimension
output: optimized fuzzy grid partitioning

 1 initialize PRNG with shared secret
 2 produce N initial parent chromosomes
 3 while done ≠ yes do
 4  produce N random children chromosomes
 5  pass the best individual to next generation
 6  randomly mating
 7  exchange parts of chromosomes
 8  mutate with rate = 2/100
 9  evaluate chromosome by fitness function
10  if fitness satisfied then done = yes
11  else produce next generation chromosomes
12 end while
```

Fig. 4: GA-based fuzzy grid partitioning optimizer

5. Implementation and results

The data mining system is implemented in *Java* and *MySQL* database management system. In our database there are 25000 records of individuals in a club; each record consisting of three linguistic variables (*Age*, *Income*, and *Experience*) and four categorical variables (*Marriage*, *Gender*, *Owning_house*, *Owning_car*). It means in our database there are seven fields that can be used for mining (of course there exist some other fields such as name and address that are not relevant to our work). Table 2 shows the assigned parameter values for this implementation.

Table 2: Parameters Settings for ga-based approach

Parameter	Value
Population size	20
Mutation probability	0.02
Crossover probability	0.9
Elitism probability	0.5
Number of GA iteration	40
Minimum Fuzzy Support	0.3

The results of the experiments show that the proposed method works correctly. Table 3 shows the the number of useful grids with respect to dimension factor. It is notable that *minFS* equals 0.3 in all experiments. Results show that when dimension factor has a higher value the number of dimensional fuzzy concepts grows.

Table 3: Number of useful fuzzy grids with respect to dimension factor

	Useful Grids Dimension		
Dimension factor	1-D	2-D	3-D
0.01	9	27	104
0.5	9	24	96
1	9	30	108
5	10	36	140

Figure 5 shows plots of the best individuals' fitness in each generation when the dimension factor is 0.5, 1 or 5. When the dimension factor is 0.5, the most effective parameter in the fitness is the average of 1-D fuzzy concepts, average of 2-D fuzzy concepts multiplied by 0.5, and the average of 3-D fuzzy concepts multiplied by 0.25. When dimension factor is 5, the most effective parameter is average of 3-D fuzzy concepts as the average of 3-D fuzzy concepts multiplied by 25 and average of 2-D fuzzy concepts multiplied by 5. Since our database attributes are few in this example (only three linguistic variables and four categorical variables), the results are shown only in 1-D, 2-D and 3-D. It is obvious that if there were more attributes in our database then we could discuss fuzzy concepts with more information (attributes).

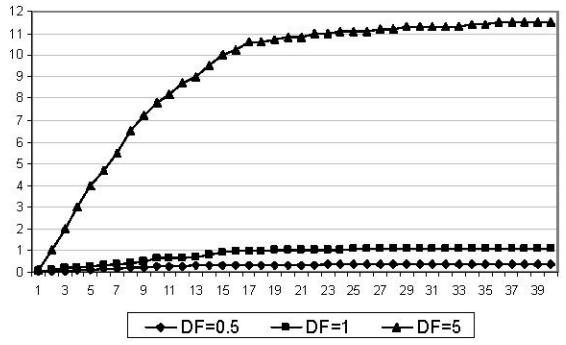

Fig. 5: Fitness value of the best individuals in GA w.r.t generation for dimension factor equal to 0.5, 1, and 5

According to Table 3, when dimension factor is 1 the system would be a common fuzzy data mining method in which we could not affect the output. For instance when dimension factor is 5, our 3-D information (information with three attributes) is more than when the dimension factor is 1. This indicates the superiority of our method compared to the common simple fuzzy grid based data mining method.

6. Discussion and future works

This paper is focused to extract useful information from a given database using a GA-optimized fuzzy data mining method. The output of the common fuzzy mining system is constant. But sometimes users want more information from database, perhaps information with higher dimensions. So we used genetic algorithm to find information with more attributes. The GA has one input which decides the dimension of output of system. The output of GA is number of linguistic values for each attribute in the database. We use this input to get fuzzy information with lower/higher attributes. However, experience shows that determining this value is non-trivial. Our future work, therefore, may consider a method for determining the dimension factor.

References

[1] S.M. Chen, "Techniques and applications of fuzzy theory in document retrieval systems", in:C.T. Leondes (Ed.), Fuzzy Theory Systems: Techniques and Applications, vol. 2, AcademicPress, San Diego, CA, 1999, pp. 691–715.

[2] S. Miyamoto, "Two approaches for information retrieval through fuzzy associations", *IEEE Transactions on Systems, Man, and Cybernetics* 19 (1) (1989) 123–130.

[3] Rudolf Kruse, Detlef Nauck, and Christian Borgelt "Data Mining with Fuzzy Methods: Status and Perspectives", Department of Knowledge Processing and Language Engineering Otto-von-Guericke-University of Magdeburg Universit¨atsplatz 2, D-39106 Magdeburg, Germany

[4] Y. Yuan, M.J. Shaw, "Induction of fuzzy decision trees", Fuzzy Sets and Systems 69 (1995) 125–139

[5] C. Hu, R.S. Chen, G.H. Tzeng, "Generating learning sequences for decision makers through data mining and competence set expansion", IEEE Transactions on Systems, Man, and Cybernetics 32 (5) (2002) 679–686.

[6] Y.Chung Hu, "Finding useful fuzzy concepts for pattern classification using genetic algorithm" ,Department of Business Administration, Chung Yuan Christian University, Chung-Li 320, Taiwan, ROC Information Sciences 175 (2005) 1–19

[7] Y. C. Hu, R. S. Chen, and G. H. Tzeng, Discovering fuzzy association rules using fuzzy partition methods, in Knowledge-Based Systems, Volume 16, Number 3, April 2003, pp. 137-147(11)

[8] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353

[9] J-S.R. JANG *Neouro-Fuzzy and Soft computing* , Prentice Hall 1997.

[10] J. W. Han and M. Kamber, *Data Mining: Concepts and Techniques*.San Mateo, CA: Morgan Kaufmann, 2001.

[11] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 260–270, Aug. 1995.

[12] J. H. Holland, "Adaptation in natural and artificial systems", *Ann Arbor, MI University of Michigan Press 1975.*

[13] D. E. Goldberg, "The genetic algorithms in search, optimization, and machine learning", New Y7ork: Addison-Wesley, 1989.

[14] R. L. Haupt, S. E. Haupt, Practical Genetic Algorithm, John Wiley &. Sons, Inc., 1998.