

## DISCOVERING FUZZY CLASSIFIERS BY GENETIC ALGORITHMS

**Hasanzade, M.,**  
M.S Student of Sharif University of  
Technology of Iran  
hasanzadeh@ce.sharif.edu

**Bagheri, S.**  
Assistant Professor of Sharif  
University of Technology of Iran  
Bagheri-s@sharif.edu

**Lucas, C.**  
Professor of Tehran University of  
Iran  
lucas@ipm.ir

### ABSTRACT

Today's solving most of application problems results in solving a data classification problem. Lots of solutions are proposed for classification problems. Most of them concentrate on reducing detection error of classifiers. For error reduction fuzzy logic can be useful. Obtaining error free and optimized classifiers could be done by evolutionary algorithms. Base on these, we proposed a machine learning based method for discovering fuzzy classifiers (a set of fuzzy rules) by genetic algorithms. The proposed method is tested by a number of benchmark data sets. Results in these tests are better than those of similar systems. This paper exhibits the obtained results.

### 1. INTRODUCTION

Today's, most of application problems could be defined as a case of classification problem. In such problems, correct separation and categorization of patterns of an event to gether with accurate prediction of not known patterns, is desired. There have been many approaches on data classification, including statistical approaches like LDA, MDA and PAL, machine learning approaches like C4.5, S-plus and CART and neural networks approaches like LVQ and RBF [1].

Solving a classification problem often starts by discovering and defining main attributes of patterns. Then the combination and relation of these features are utilized to define a special class.

As long as problem dimension increases, finding the correct interaction of features becomes more difficult. In this case prediction error of patterns may increase too. Such problems have attracted a lot of attention in machine learning and data mining techniques. Data mining output can be represented by If.. Then.. rules and discovering of best rules can be done in an evolutionary process ([4] ,[5]).

In general, the main motivation for using genetic algorithms in the discovery of high-level prediction rules is that they perform a global search and cope better with attribute interaction than the greedy rule induction algorithms often used in data mining [6].

On the other hand, in many applications, boundary of classes is not well defined as the definition of crisp classes. This fuzziness may cause errors. The error could be reduced by using fuzzy logic. Fuzzy logic is useful for definition of attributes as well as for determination of membership degree of a pattern to different classes.

There are many applications that discover fuzzy classifiers by genetic algorithms. Several methods are proposed for fuzzification, evaluating classifiers and doing genetic operators in these applications. We improved these methods in order to obtain higher detection rate.

This paper shows some results and reports the performance of evolved fuzzy classifiers for some data sets. The results are better than of similar systems.

The subsequent sections are organized as follows. Section 2 briefly describes the basic fuzzy logic and fuzzy classifier concepts used in this paper, section 3 reviews evolutionary data mining methods and presents the proposed approach, section 4 describes experiments and analysis of results and section 5 draws some conclusions and suggestions for future works.

### 2. FUZZY CLASSIFIER

In fuzzy logic, fuzzy sets define the linguistic notions and membership functions define the truth-value of such linguistic expressions. Table1 shows the difference between classic sets and fuzzy sets.

Membership function defines the membership degree to a fuzzy set. The domain is the universe of discourse (set of values the object can take) and the interval [0,1] is the

rang. Membership functions are often selected triangular or trapezoidal.

Table1: comparison between fuzzy sets and classic sets

CLASSIC SETS	FUZZY SETS
An object is entirely in a set or is not.	An object can partially be in a set.
The membership degree takes only two values 0 or 1.	The membership degree takes values between 0 and 1
1 means entirely in the set, 0 means entirely not in the set. Other values are not allowed.	1 means entirely in the set, 0 means entirely not in the set, other values mean partially in the set.

A collection of fuzzy sets, called fuzzy space, defines the fuzzy linguistic values or fuzzy classes that an object can belong to. A sample fuzzy space of five membership function is shown in figure 1.

With fuzzy spaces, fuzzy logic allows an object to belong to different classes at the same time. This concept is helpful when the difference between classes is not well defined.

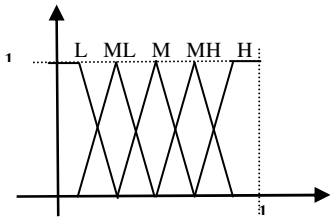


Fig.1 : A fuzzy space of five membership function .

Fuzzy classifier can be represented by a fuzzy rule. Fuzzy rules have the form:

*IF condition THEN consequent [weight]*

Condition is a complex fuzzy expression, i.e., that uses fuzzy logic operators and atomic fuzzy expressions. Consequent is an atomic expression, weight is a real number that defines the confidence of the rule.

### FUZZY C-MEAN CLUSTERING:

The fuzzy c-means (FCM) algorithm minimizes the objective function [8]:

$$J(U,V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2$$

$$u_{ik} : [0,1]$$

$$\forall k, \sum_{i=1}^c u_{ik} = 1 \quad (1)$$

U is the partition matrix that shows to what degree the k-th data point  $x_k$  belongs to each cluster as measured by its distance from the prototype of the i-th cluster,  $v_i$ . m is a weighting exponent. C is the number of clusters and n is the number of data points.

### 3. EVOLVING CLASSIFIERS

A genetic algorithm is the computational equivalent of the natural evolutionary process. In a genetic algorithm a set of chromosomes (population) is evolved using a set of genetic operators. Each chromosome codifies a possible solution for the given problem. Genetic operators are mutation, crossover and selection. Each chromosome has probability to be used by one of the genetic operators, and this probability depends on its fitness (efficiency of the chromosome to solve the given problem).

When we use genetic algorithms to solve a problem we should decide the appropriate method for three main steps: representing solutions by chromosomes, customizing genetic operators for defined chromosomes and defining the fitness function. After these steps we should decide for evolution process in details. We review the methods for these steps briefly and represent our proposed ones below.

#### A. RULE REPRESENTATION

Genetic algorithms (GAs) for rule discovery can be divided into two broad approaches based on how rules are encoded in the population of individuals. In the Michigan approach [6] each individual encodes a single prediction rule, whereas in the Pittsburgh approach [6] each individual encodes a set of prediction rules.

There are several approaches for representing a rule's condition part like: conjunctions of simple terms, disjunctions of simple terms, fixed condition structure, linear-Tree with precedence representation and complete expression tree [7].

We have used disjunctive normal form (DNF) representation. In this approach a simple rule 'IF' part constructed of conjunction of simple terms. Disjunction of these simple rules makes antecedent of a composed rule (chromosome). IF part of this rule is like below (order k) [7]:

$$DNF-k : (A_1 \wedge A_2 \wedge \dots \wedge A_k) \vee (A_{k+1} \wedge A_{k+2} \wedge \dots \wedge A_{2^*k}) \vee \dots \vee (A_p \wedge A_{p+1} \wedge \dots \wedge A_{p+k-1}) \quad (2)$$

Where  $p = k * [m/k]$  ([.] indicates the integer part). Represent ability, low disruption in genetic operators and the ability to define an appropriate fitness function are the reasons for selecting DNF for representation of Rules.

Representing 'Then' part of a rule in a chromosome depends on evolution method and will be discussed later.

#### B. GENETIC OPERATORS

In evolutionary data mining we need a crossover operator that causes local changes and does not produce incorrect chromosomes. In other words, we want that the crossover produces offspring similar to their parents. Mutation operator can be used simply in this application and does not need any extra consideration.

In selected representation method, the common genetic operators can be applied simply and no extra modification is needed.

We have used single point crossover that select crossover point between simple rules. Implementation of mutation operator is so that it can cover gene addition and gene deletion operators as well.

### C. FITNESS FUNCTION

The predictive performance of a rule can be summarized by a 2\*2 matrix, called a confusion matrix, as illustrated in Fig. 2 [6]. The elements of this matrix are True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). For example, TP shows the correct (T) firing (P) of the rule. We could define confidence factor (CF) and completeness measure (Comp) of the rule by confusion matrix elements as below [6]:

$$CF = TP / (TP + FP) \quad (3)$$

$$Comp_1 = TP / (TP + FN), \quad Comp_2 = TN / (TN + FP) \quad (4)$$

For completeness measure, we could consider both covering ratio of related patterns (comp1) and non-covering ratio of unrelated ones (comp2).

		Actual Class	
		C	Not C
Predicted Class	C	TP	FP
	Not C	FN	TN

Figure 2: confusion matrix

We can define several fitness functions by confusion matrix and extracted relations of it. For example weighted sum of two completeness factors[4] and product of confidence and completeness factors[6] have been used. We have defined main part of fitness function as below:

$$Fitness1 = comp1 \times comp2$$

$$= (TP / (TP + FN)) \times (TN / (TN + FP)) \quad (5)$$

Since the goal in optimization of fuzzy classifiers is the simultaneous satisfaction of the optimization objectives (completeness factors) t-norms must be used. In other words, because we are interested in a rule that covers its class patterns while not covering other class patterns, the t-norm (product) of completeness factors is used. Using this approach, the risk of local optimum (empty or general rules) convergence probability is too much decreased.

In proposed approach confusion matrix elements are calculated according to the following set of equations:

$$TP = \sum_{i=1}^p \text{predicted} ( \text{class}_1 \text{data}_i )^{1/2}$$

$$FP = \sum_{i=1}^q \text{predicted} ( \text{class}_2 \text{data}_i )^{1/2}$$

$$TN = \sum_{i=1}^q (1 - \text{predicted} ( \text{class}_2 \text{data}_i ))^{1/2}$$

$$FN = \sum_{i=1}^p (1 - \text{predicted} ( \text{class}_1 \text{data}_i ))^{1/2} \quad (6)$$

The above relations are for classifier of class1 in a binary classification. Predicted is the fuzzy value of the condition part of the codified rule. Predicted (classjdatai) shows the prediction degree of the i-th data of j-th class with the classifier. p and q are the number of class1 and the samples of other classes of training data sets respectively.

For calculating confusion matrix elements we could use sum of predictions (as in [4]) instead of sum of square roots. But using sum of square roots causes rules to have general prediction on their samples. In other words by proposed relations, the rule that predicts all the patterns partially (~0.5) is better than a specialized rule that predicts some of samples very high (~1) and others very low (~0).

For considering comprehensibility of the rule in fitness function we should add another factor. This factor is often selected as simplicity. We have selected representation of rules method so that simplicity could be evaluated well. In this equation simplicity is a function of number of simple rules (Rc) in a chromosome and total number of conditions (Cc) in the rule as below:

$$Simp = 1/Cc + Rc/Cc \quad (7)$$

By this definition the final fitness function is:

$$Fitness = w_1 * (Comp_1 * Comp_2) + w_2 * Simp \quad (8)$$

### D. EVOLUTION PROCESS

Evolution process depends on representation of 'Then' part of rule (predicted class). There are several ways for this representation. One possibility is to encode it in the genome of an individual and making it subject to evolution. The second possibility is to associate all individuals of the population with the same predicted class. In this way we would need to run the evolutionary algorithm one time for each class. The third possibility is to choose the predicted class most suitable for a rule in an evolution process. The above first and third possibilities avoid the need to perform multiple runs of the evolutionary algorithm to discover rules predicting different classes, which is the case in the above second possibility. Overall, the third possibility seems more sound than the first one [6]. In several applications, where the number of classes are low the second approach is used (e.g. [4], [5]). We have also used this approach.

### E. DETECTION METHOD

We have used confidence of a rule to determine detection of patterns. First we multiply this factor (equation 3) by prediction of rule antecedent, then the class of pattern determined according to the result.

Detection threshold is set to 0.5 for detecting pattern by only one classifier. When we use all classifiers together, the most fired rule determines the class of pattern. These two methods can both be used in binary classification.

#### 4. EXPERIMENTATION

In order to evaluate the performance of proposed approach, tests were conducted using the IRIS, WINE and VOTE [13] data sets. The data sets are described in Table2. On preprocessing step numerical attributes have been normalized between 0 and 1. C- Mean clustering is used to define three triangular membership function for continuous attributes. For non-numerical ones, we used the categorical values as crisp sets (fuzzy sets without overlapping). We have compared our system with two other evolutionary systems named here system1 [7], system2 [5]. Both of these systems discover fuzzy rules in distinct evolution processes. In system2 classifiers co-evolved with membership functions.

The following parameters were used: Population size: 200, Maximum number of iterations: 200, Selection strategy: tournament selection. The size of tournament and the mutation rate is shown in table2 for each test. Fitness of individuals is evaluated according to equation (9) with parameters:  $w_1 = 0.999$ ,  $w_2=0.001$ .

After 10 iterations in evolution process, rule pruning has applied. In this process simple duplicated rules in a classifier will be omitted. To lower running cost of this process, it could be applied on elitisms.

The tests results have been shown in Table 3, 4, and 5 for IRIS, VOTE and WINE data sets respectively. The result of ESIA and BGP system are taken from report of system2.

Table2: Data set features and test parameters

Test data sets	Test data set features			Test parameters	
	Size	No. of Classes	No. of Attribute	Mutation Rate	Tournament size
WINE	178	3	14	0.9	4
IRIS	150	3	4	0.5	2
VOTE	435	2	16	0.65	4

Table3: Test results for IRIS data set.

system	Detection rate	Rule
Our system	<b>97.11</b>	Fuzzy : DNF3
System2	95.3	Fuzzy : DNF
System1	94.84	Fuzzy: complete tree
System1: DNF3	93.99	Fuzzy: DNF3
BGP	94.1	Crisp
ESIA	95.33	Crisp

Table4: Test results for VOTE data set.

system	Detection rate	Rule
Our system	<b>95.33</b>	Fuzzy : DNF3
System1	95.42	Fuzzy: complete tree
System1: DNF3	95.43	Fuzzy: DNF3

A ten-fold testing was employed [9]. That is, data set is divided randomly in ten groups; each group was taken as testing set for the fuzzy classifier trained with the others

nine groups. The process is repeated five times. The proposed result of our system is the average of three times testing as above.

Table5: Test results for WINE data set.

system	Detection rate	Rule
Our system	<b>94.06</b>	Fuzzy : DNF3
System1	92.22	Fuzzy: complete tree
System1: DNF3	90.55	Fuzzy: DNF3

Our system works best in IRIS and WINE data sets. They have continuous attributes, so they use the benefits of fuzzy rules and using the sum of square roots in fitness function. The VOTE data set consist of only discrete attributes so we did not expect improving the result by proposed methods.

The results of binary classification of VOTE data set are for the case of using only one classifier that works better than the case using both of the classifiers.

#### 5. CONCLUSIONS

We proposed a method for discovering fuzzy classifier by genetic algorithms in this paper. In evolutionary data mining, finding a suitable fitness function that estimates prediction of rules well is difficult. We proposed a fitness function based on confusion matrix that works well enough. We guess that using aggregation functions in calculation of confusion matrix elements will improve the result. This way, we may also prevent over training.

In this paper we extract fuzzy classifiers in separate evolution processes. We hope that in binary classification, co-evolution of classifiers will improve the prediction rate. This will prevent from resulting in contradictory classifiers in the classification cases when we may have more than exactly two classes. Now, we are testing these proposed suggestions and we hope the improvement of results.

#### REFERENCES

- [1] Lim, T., Loh, W., A Comparison of Prediction, Accuracy, Complexity and Training Time of Thirty-Three Old and New Classification Algorithms. Technical Report, Department of Statistics, University of Wisconsin-Madison, No. 979, 1997.
- [2] Bridges, S., M., Vaughn, R., B., Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection, Proceedings of the Twenty-third National Information Systems Security Conference, Baltimore, MD, October 2000.
- [3] Lee ,w., "Mining audit data to build intrusion detection models", Proc. Int. conf. Knowledge discovery and Data Mining (KDD'98), pages 66-72.1998.
- [4] Gomez, J., Dasgupta, D., Evolving Fuzzy Classifiers for Intrusion Detection, proceedings of the 2002 IEEE Workshop on Information Assurance.
- [5] Mendes, R., R., F., Voznika, F., de B., Freitas, A., A., Nievola, J. C., Discovering Fuzzy Classification Rules

with Genetic Programming and Co-Evolution, In Principles of Data Mining and Knowledge Discovery (Proc. 5th European Conference PKDD 2001) - Lecture Notes in Artificial Intelligence, Springer-Verlag, 2001.

[6] Freitas, A., A., A survey of evolutionary algorithms for data mining and knowledge discovery. To appear in: A. Ghosh and S. Tsutsui. (Eds.) Advances in Evolutionary Computation. Springer- Verlag, 2001.

[7] Gomez, J., Gonzalez, F., Dasgupta, D., Complete Expression Trees for Evolving Fuzzy Classifier Systems

with Genetic Algorithms, Submitted to the Evolutionary Computation Conference GECCO02, 2002.

[8] Zimmermann, H., J., Fuzzy Set Theory and Its Application, Kluwer Academic Publishers, 1996.

[9] Blake, C., L., Merz, C., J., UCI Repository of machine learning databases Irvine, CA: University of California, Department of Information and Computer Science. <http://www.ics.uci.edu/~mlern/MLRepository.html> (1998).