International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:8, 2010

Using Swarm Intelligence for Improving Accuracy of Fuzzy Classifiers

Hassan M. Elragal

***Abstract*—**This paper discusses a method for improving accuracy of fuzzy-rule-based classifiers using particle swarm optimization (PSO). Two different fuzzy classifiers are considered and optimized. The first classifier is based on Mamdani fuzzy inference system (M_PSO fuzzy classifier). The second classifier is based on Takagi-Sugeno fuzzy inference system (TS_PSO fuzzy classifier). The parameters of the proposed fuzzy classifiers including premise (antecedent) parameters, consequent parameters and structure of fuzzy rules are optimized using PSO. Experimental results show that higher classification accuracy can be obtained with a lower number of fuzzy rules by using the proposed PSO fuzzy classifiers. The performances of M_PSO and TS_PSO fuzzy classifiers are compared to other fuzzy based classifiers

***Keywords*—**Fuzzy classifier, Optimization of fuzzy system parameters, Particle swarm optimization, Pattern classification.

I. INTRODUCTION

FUZZY inference systems have been successfully applied in many fields such as control, data classification, decision analysis, prediction, computer vision and expert systems [1-2]. Many approaches have been proposed for design of an optimal fuzzy system, such as statistical clustering methods, neural networks, evolutionary programming and swarm intelligence [3-11].

In data classification applications, fuzzy rules are derived from human experts as linguistic knowledge. Since it is not easy to derive fuzzy rules from human experts, many approaches have been proposed to generate fuzzy rules automatically from training patterns [12-16].

To generate fuzzy rules from training patterns, fuzzy partitions in the input space are generally considered. Two types of input space partitions have been used to model fuzzy systems. The first one is a grid fuzzy partition [13-16] and the second one is a scatter fuzzy partition [17, 18]. A Grid fuzzy partition has two problems. The first one is that the number of fuzzy sets for each input variable is determined in advance. The second problem is that the learning complexity suffers from an exponential explosion of number of rules as the number of inputs increases [13]. For scatter fuzzy partition, Simpson [17] proposed a method for generating the hyper box regions to construct a fuzzy classifier. In this approach, the value of the learning parameter is very critical and directly affects the structure of the fuzzy classifier and the classification performance.

Hassan M. Elragal is with Electrical Engineering Department, Faculty of Engineering, University of Bahrain, Bahrain (e-mail: helragal@eng.uob.bh)

Fuzzy C-mean (FCM) [19] and subtractive clustering algorithms [20] always been used in fuzzy partitioning. A critical problem for the fuzzy clustering algorithms is how to determine the optimal number of clusters. Too many clusters result in an unnecessarily complicated rule base, while too few clusters result in a poor performance.

Emami [21] show that, the essence of the fuzzy structure identification method is in clustering and projection. First, the output space is partitioned using a fuzzy clustering algorithm. Then, the partitions (clusters) are projected onto the space of the input variables. The output partition and its corresponding input partitions are the consequents and antecedents, respectively. Salehfar et al. [22] propose a new systematic and simple algorithm to build and tune fuzzy model directly from the input-output data. The new algorithm is called the Linguistic Fuzzy Inference (LFI) model. LFI use a projection method to determine membership functions and fuzzy rules then use a neural network to tune the parameters of the membership functions. Vachkov and Fukuda [23] introduce multilevel fuzzy modeling. They proposed a composite fuzzy model CFM that is an additive structure of one main fuzzy model and a number of correction models that try to gradually decrease the total inference errors.

Hwang [24] presents an automatic design of the optimal fuzzy rule base for modeling and control using evolutionary programming. Genetic algorithms (GAs) were applied to tune fuzzy membership functions or fuzzy rules separately. Homaifar and McCormick [25] encoded fuzzy membership functions and all possible rules into the chromosome. Cordon et al. [26] applied GAs to optimize the partition number of input variables. Ho et al. [27] applied an evolutionary scatter partition of feature space to design a compact fuzzy system, and used intelligent GAs to search for the optimal solution. Zhou and Khotanzad [28] propose a method for designing a fuzzy-rule-based classifier where fuzzy membership functions and the size and structure of fuzzy rules are extracted from the training data using GAs.

Swarm intelligence has been used as a tool in classification problems. Mirzayans [29] proposes a swarm system capable of extracting and exploiting the geometric properties of objects in images for fast and accurate recognition. The resulting feature profile is then processed by a classification subsystem to categorize the object. Ata [30] uses Particle Swarm Optimization (PSO) for feature selection in order to reduce the complexity of the classifier. Omran et al. [10] propose a dynamic clustering approach based on PSO. This approach is applied to unsupervised image classification. They try to determine the "optimum" number of clusters. The algorithm starts by partitioning the data set into a relatively large number of clusters then use binary PSO to find the "best" number of clusters. The centers of the chosen clusters is then refined via the K means clustering algorithm. Chia [11] proposes a method based on PSO for pattern classification to select a fuzzy classification system with an appropriate number of fuzzy rules.

In this paper, two fuzzy classifiers optimized with PSO are proposed. The first classifier is based on a Mamdani fuzzy inference system and the second one is based on a Takagi-Sugeno fuzzy inference system. The parameters of the two fuzzy classifiers including premise (antecedent) parameters, consequent parameters and structure of fuzzy rules are optimized using PSO. The performances of the two fuzzy-rule-based classifiers are tested on five real-world databases (Iris, phoneme, diabetes, heart and wine). Their performances are compared to other fuzzy classifiers.

The rest of this paper is organized as follows: Section II describes the structure of fuzzy systems. Section III discusses the two proposed PSO-based fuzzy classifiers. Section IV shows other techniques for optimizing the proposed fuzzy classifiers. Section V describes how classifiers can be compared based on average rank. Section VI shows the experimental results. Concluding remarks are included in Section VII.

II. Structure of Fuzzy Systems

The basic architecture of a fuzzy system is shown in Fig. 1. The main components are a fuzzification interface, a fuzzy rule base (knowledge base), an inference engine (decision-making logic), and a defuzzification interface.

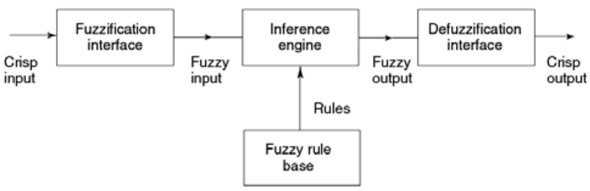


Fig. 1 Basic fuzzy system

The two most popular fuzzy inference systems that have been widely deployed in various applications are Mamdani fuzzy inference systems [31] and Takagi-Sugeno fuzzy inference systems [32]. The differences between these two fuzzy inference systems lie in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures differ accordingly.

Mamdani, fuzzy inference system is shown in Fig. 2. The rule antecedents and consequents are defined by fuzzy sets and have the following structure:

Rule i: ***IF*** *x is* A_i ***AND*** *y is* B_i ***THEN*** Z_i *is* C_i

There are several defuzzification techniques. The most widely used defuzzification technique is centroid of area.

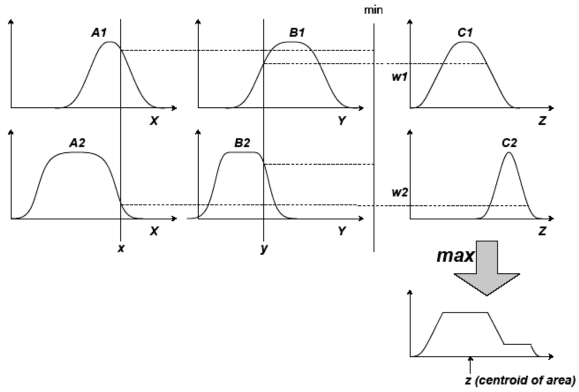


Fig. 2. Mamdani fuzzy inference system

Takagi–Sugeno fuzzy inference system is illustrated in Fig. 3. Takagi and Sugeno proposed an inference scheme in which the conclusion of a fuzzy rule is constituted by a weighted linear combination of the crisp inputs rather than a fuzzy set. The fuzzy rule has the following structure:

Rule i: ***IF*** *x is* A_i ***AND*** *y is* B_i ***THEN***

$$Z_i = p_{i,1}\, x + p_{i,2}\, y + p_{i,0}$$

Where $p_{i,1}$, $p_{i,2}$, and $p_{i,0}$ are the consequent parameters.

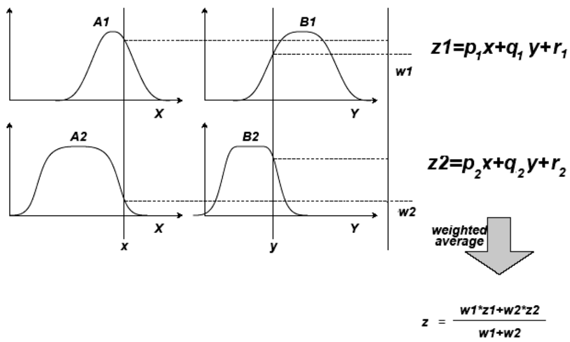


Fig. 3. Takagi–Sugeno fuzzy inference system

III. The Proposed Fuzzy Classifiers

In this work, a particle swarm optimization based approach is used to optimize the parameters of fuzzy classifiers from the training data. PSO is a swarm intelligence method for global optimization [33-35]. In a PSO fuzzy based approach, each particle in the swarm is considered to represent a fuzzy classification system. Then, a fitness function is evaluated to guide the search and select an appropriate fuzzy classification system such that the number of incorrectly classified patterns is minimized.

A. Particle Swarm Optimization Algorithm

The basic PSO algorithm consists of three steps, namely, generating particles positions and velocities, velocity update, and finally, position update. First, the positions, $\boldsymbol{x}^i(k)$ and velocities, $\boldsymbol{v}^i(k)$ of the initial swarm of particles are randomly generated. The positions and velocities are given in a vector format for the i^{th} particle at time k. The second step is to update velocities of all particles at time k +1 using the particles objective or fitness values, which are, function of the particles current positions in the design space at time k. The fitness function value of a particle determines which particle has the best global value in the current swarm, $\boldsymbol{p}^g(k)$, and determines the best position of each particle over time, $\boldsymbol{p}^i$, i.e. in current and all previous moves. The velocity update formula uses these two pieces of information for each particle in the swarm along with the effect of current motion, $\boldsymbol{v}^i(k)$, to provide a search direction, $\boldsymbol{v}^i(k+1)$, for the next iteration.

$$v^i(k+1) = w*v^i(k) + c_1*rand*(p^i - x^i(k)) + c_2*rand*(p^g(k) - x^i(k)) \quad (1)$$

Where w is inertia factor, $c1$ is self confidence factor; $c2$ is swarm confidence factor and *rand* is uniformly distributed random variables to ensure good coverage and avoid entrapment in local minimum.

Position update is the last step in an iteration. The position of each particle is updated using its velocity vector as:

$$x^i(k+1) = x^i(k) + v^i(k+1) \quad (2)$$

The three steps of velocity update, position update, and fitness calculations are repeated until a desired convergence criterion is met.

B. Mamdani Based PSO (M_PSO) Fuzzy Classifier

The first proposed fuzzy classifier is based on Mamdani fuzzy inference system. In this model, PSO is used to optimize the parameters of input membership functions, output membership functions and structure of the fuzzy rules. The number of fuzzy rules is gradually changed from 2 to 15 (this can be extended to any number of rules).

The steps for designing of M_PSO fuzzy classifier are as follows:

Step 1: Divide the database into training data set and testing data set. Each input will be represented by three Gaussian membership functions. Output will be represented by number of Gaussian membership functions equal to the number of classes. Start with number of rules equal 2.

Step 2: Create a population of N particles (classifiers). Each particle consists of:
   i. Parameters of input and output membership functions (mean and variance)
   ii. Random rules generated according to the assigned number of rules.

Step 3: For each particle, construct Mamdani fuzzy system, apply training data set and calculate the corresponding fitness value (classification accuracy).

Step 4: Apply PSO algorithm and produce new population.

Step 5: Repeat steps 2 to 4 until a solution results in perfect classification of all training samples or until a maximum number of fitness function evaluation is reached.

Step 6: Record the final fitness value with the corresponding number of rules.

Step 7: Increase number of rules by one and check if number of rules greater than 15. If yes go to step 8 else go to step 2.

Step 8: Select the best performing fuzzy classifier based on the training data set results recorded in step 6. The best performing classifier is the one with maximum classification accuracy and minimum number of rules.

Step 9: Apply testing data set on the selected fuzzy classifier and record classification accuracy.

To identify the size of each particle, assume there are Ni inputs and one output. Each one of the inputs is represented by three Gaussian membership functions. The output is represented by number of Gaussian membership functions equal to number of classes Nc. Each membership function is represented by two variables (mean and variance). The total number of variables for input and output membership functions will be (Ni*3+Nc)*2. Assume there are Nr rules and each rule has (Ni+1) variables. Then, the total number of variables that need to be optimized (particle size) will be (Ni*3+Nc)*2 + (Ni+1)* Nr.

C. Takagi-Sugeno Based PSO (TS_PSO) Fuzzy Classifier

The second proposed classifier is based on Takagi-Sugeno fuzzy inference system. In this model, PSO is used to optimize parameters of the input membership functions (antecedent or premise) and consequent parameters. The number of rules is gradually changed from 2 to 15 (this can be extended to any number of rules).

The steps for design of TS_PSO fuzzy classifier are as follows:

Step 1: Divide the database into training data set and testing data set. Start with number of rules equal 2.

Step 2: Create a population of N particles (classifiers). Each particle consists of:
   i. Parameters of input membership functions (mean and variance), where each input will be represented by number of membership functions equal to assigned number of rules
   ii. Consequent parameters of Takagi-Sugeno fuzzy model according to assigned number of rules.

Step 3: For each particle, construct Takagi-Sugeno fuzzy system, apply training data set and calculate the corresponding fitness value (classification accuracy).

Step 4: Apply PSO algorithm and produce a new population.

Step 5: Repeat steps 2 to 4 until a solution results in perfect classification of all training samples or until a maximum number of fitness function evaluation is reached.

Step 6: Record the final fitness value with the corresponding number of clusters.

Step 7: Increase number of rules by one and check if number of clusters greater than 15. If yes go to step 8 else go to step 2.

Step 8: Select the best performing fuzzy classifier based on the training data set results recorded in step 6. The best performing classifier is the one with maximum classification accuracy and minimum number of rules.

Step 9: Apply testing data on the selected fuzzy classifier and record classification accuracy.

To identify the size of each particle, assume there are *Ni* inputs and one output. Assume there are *Nr* rules. Each input will be represented by number of membership functions equal to assigned number of rules. Each membership function is represented by two variables (mean and variance). Then the number of premise parameters are $Ni*Nr*2$. For the consequent parameters we have $(Ni+1)*Nr$ variables. The total number of variables that need to be optimized (particle size) will be $(3*Ni + 1)*Nr$.

IV. Comparison with Other Optimization Techniques

The proposed fuzzy inference system is optimized with the PSO technique. Two other commonly used optimization techniques will be used to optimize the proposed classifiers. The first one is genetic algorithm (GA) and the second one is differential evolving (DE) algorithm.

Genetic Algorithms (GAs) [36] use the concept of Darwin's theory of evolution which basically stressed the fact that the existence of all living things is based on the rule of "survival of the fittest". The idea starts by creating different possible solutions to a problem. These solutions are then tested for their performance. Among all possible solutions, a fraction of the good solutions is selected, and the others are eliminated (survival of the fittest). The selected solutions undergo the process of reproduction, crossover, and mutation to create a new generation of possible solutions (which are expected to perform better than the previous generation). This process of production of a new generation and its evaluation is repeated until there is a convergence within a generation.

Differential Evolution (DE) [37] is a parallel direct search heuristic approach. The initial vector population is chosen randomly. DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector. Let this operation be called mutation. The mutated vector's parameters are then mixed with the parameters of another predetermined vector, the target vector, to yield the so-called trial vector. Parameter mixing is often referred to as "crossover". If the trial vector yields a lower cost function value than the target vector, the trial vector replaces the target vector in the following generation. This last operation is called selection. Each population vector has to serve once as the target vector so that competitions take place in one generation.

V. Comparisons of Multiple Classifiers

In this paper, classifiers are compared to each other based on average ranks. Ranks are based on overall classification accuracy. Ranking strategies are as follows, the best performing algorithm getting the rank of 1, the second best rank 2 and so on. In case of ties, average ranks of the tied ranks will be assigned.

VI. Experimental Results

The performance of the two proposed PSO fuzzy classifiers is studied using five widely used real-world databases (Iris, Phoneme, Diabetes, Heart and Wine). Four different fuzzy classifiers are compared. Classifier 1 and 2 are based on Mamdani fuzzy system, classifier 3 and 4 are based on Takagi-Sugeno fuzzy system.

A. *Mamdani Type Classifiers*

Classifier 1:

The first classifier is a Mamdani type fuzzy classifier with fuzzy partition of input and output space using FCM clustering algorithm. In this classifier, the user assigns number of clusters (In this paper, number of clusters are optimized on the training data set and changed from 2 to 15 clusters). Number of rules and number of Gaussian membership functions for each input and output is the same as the assigned number of clusters. FCM is used to optimize parameters of membership functions.

Classifier 2:

The second classifier is the M_PSO fuzzy classifier discussed before.

B. *Takagi-Sugeno Type Classifiers*

Classifier 3:

The third classifier is a Takagi-Sugeno type fuzzy classifier where the input space is partitioned using FCM clustering algorithm and the consequent parameters are optimized using least square.

Classifier 4:

The fourth classifier is the TS_PSO fuzzy classifier discussed before.

1. *Experiments with Iris Database*

The Iris database contains three classes (Setosa, Versicolor and Virginica). Each class contains 50 patterns. Each pattern is represented by four inputs. Training data set contain 100 patterns and testing data set contains 50 patterns.

Classifier 1 gets 79.40% overall correct classification accuracy with 8 rules. Classifier 2 (M_PSO) gets 95.83% with 4 rules. The optimized fuzzy rules using M_PSO fuzzy classifier are as follows:

Rule	Inputs				Output
	1	2	3	4	
1	2	2	1	1	2
2	2	2	3	2	3
3	2	2	3	1	3
4	1	0	1	0	2

Rule 1 can be interpreted as, if input 1 is mf 2, and input 2 is mf 2, and input 3 is mf 1, and input 4 is mf 1, then output is mf 2. (mf stand for membership function)

Fig. 4 shows the optimized fuzzy membership functions using M_PSO fuzzy classifier. Classifier 3 gets 93.75% with 7 rules. Classifier 4 (TS_PSO) gets 100% with 2 rules. Fig. 5 shows the optimized fuzzy membership functions using TS_PSO fuzzy classifier.

2. *Experiments with Phoneme Database*

The phoneme database contains vowels coming from 1809 isolated syllables. Five different attributes were chosen to characterize each vowel. The aim is to distinguish between nasal and oral vowels. This database contains 5404 pattern presented in a random order. Class nasal contain 3818 patterns and class oral contain 1586 patterns. Training data set contains 2000 patterns and testing data set contains 3404 patterns.

Classifier 1 gets 71.07% overall correct classification accuracy with 8 rules. Classifier 2 (M_PSO) gets 78.79% with 6 rules. The optimized fuzzy rules using M_PSO fuzzy classifier are as follows:

Rule	Inputs					Output
	1	2	3	4	5	
1	1	2	2	1	1	2
2	2	2	1	1	1	2
3	2	1	2	2	2	1
4	2	1	1	1	1	1
5	1	1	2	2	2	1
6	2	2	1	1	2	2

Classifier 3 gets 77.75% with 10 rules. Classifier 4 (TS_PSO) gets 81.55% with 8 rules.

3. Experiments with Pima Indians Diabetes Database

The diabetes database contains two classes (tested positive or tested negative). Each pattern is represented by eight inputs. This database contains 768 patterns. Class tested positive contain 268 patterns and class tested negative contain 500 patterns. Training data set contain 300 patterns and testing data set contains 468 patterns.

Classifier 1 gets 59.11% overall correct classification accuracy with 8 rules. Classifier 2 (M_PSO) gets 72.72% with 4 rules. The optimized fuzzy rules using M_PSO fuzzy classifier are as follows:

Rule	Inputs								Output
	1	2	3	4	5	6	7	8	
1	1	1	1	1	1	1	1	3	1
2	2	1	0	1	0	1	1	1	2
3	2	0	3	1	1	3	3	2	1
4	2	1	1	2	2	3	1	0	2

Fig. 6 shows the optimized fuzzy membership functions using M_PSO fuzzy classifier. Classifier 3 gets 66.41% with 10 rules. Classifier 4 (TS_PSO) gets 68.51% with 9 rules. Fig. 7 shows the optimized fuzzy membership functions using TS_PSO fuzzy classifier.

4. Experiments with Cleveland Heart Database

The heart database contains two classes (tested positive or tested negative). Each pattern is represented by thirteen inputs. This database contains 303 patterns. Class tested positive contain 139 patterns and class tested negative contain 164 patterns. Training data set contain 150 patterns and testing data set contains 153 patterns.

Classifier 1 gets 57.34% overall correct classification accuracy with 6 rules. Classifier 2 (M_PSO) gets 72.69% with 4 rules. The optimized fuzzy rules using M_PSO fuzzy classifier are as follows:

Rule	Inputs													Output
	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	2	2	1	2	1	2	1	2	1	3	2	1	2	1
2	1	2	1	2	2	2	1	2	1	0	1	1	3	1
3	1	3	2	1	0	3	2	2	2	2	3	0	3	1
4	1	2	2	0	1	3	2	1	2	1	2	2	2	2

Classifier 3 gets 63.65% with 10 rules. Classifier 4 (TS_PSO) gets 74.28% with 5 rules.

5. Experiments with Wine Database

The wine database contains three classes. Each pattern is represented by thirteen inputs. This database contains 178 patterns. Class one contains 59 patterns, class two contains 71 patterns and class three contain 48 patterns. Training data set contain 90 patterns and testing data set contains 88 patterns.

Classifier 1 gets 49.43% overall correct classification accuracy with 15 rules. Classifier 2 (M_PSO) gets 64.37% with 4 rules. The optimized fuzzy rules using M_PSO fuzzy classifier are as follows:

Rule	Inputs													Output
	1	2	3	4	5	6	7	8	9	10	11	12	13	
1	2	2	1	1	3	2	0	1	2	2	2	2	2	1
2	1	1	1	1	1	2	1	2	1	1	2	2	1	1
3	3	2	1	1	1	2	2	2	3	1	1	2	3	1
4	1	0	2	1	1	0	0	2	3	1	1	1	3	1

Classifier 3 gets 91.45% with 3 rules. Classifier 4 (TS_PSO) gets 95.60% with 3 rules.

Based on testing the four fuzzy classifiers on testing data sets, Table 1 shows:

- Percentage of correct classification for each class.
- Overall classification accuracy.
- Ranking for comparing classifier 1, classifier 2 (M_PSO), classifier 3 and classifier 4 (TS_PSO).
- Number of fuzzy rules for best-optimized classifier.

Table 2 shows a comparison between the two proposed fuzzy classifiers and the same classifiers when optimized with genetic algorithm and when optimized with differential evolving algorithm. It also show ranking for comparing different optimization techniques. Note that average rank is assigned if two or more classifiers have the same classification accuracy.

From Table 1 we can conclude that:

- For Mamdani type fuzzy system, the proposed fuzzy classifier (M_PSO) has better classification accuracy with a lower number of rules.
- For Takagi-Sugeno type fuzzy system, the proposed fuzzy classifier (TS_PSO) has better classification accuracy with a lower number of rules.
- Based on average rank of overall classification accuracy (5 datasets and 4 classifiers), classifier 1 gets 4, classifier 2 gets 2, classifier 3 gets 2.8 and classifier 4 gets 1.2. This means that, the proposed classifiers have better ranks than other classifiers.
- Based on average rank of overall classification accuracy classifier 4 (TS_PSO) has better classification accuracy than classifier 2 (M_PSO).

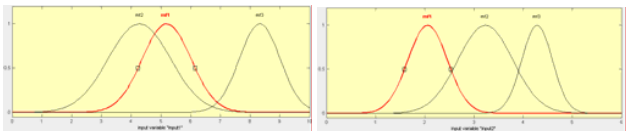
From Table 2 we can conclude that:

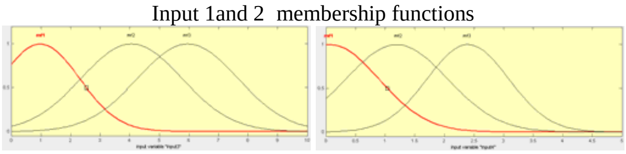
- For Mamdani type classifier, based on average rank of overall classification accuracy (5 datasets), M_PSO gets 1.2, M_GA gets 2.6 and M_DE gets 2.2. This means M_PSO has better rank than M_GA and M_DE.

- For Takagi-Sugeno type classifier, based on average rank of overall classification accuracy (5 datasets), TS_PSO gets 1, TS_GA gets 2.2 and TS_DE gets 2.8. This means TS_PSO has better rank than TS_GA and TS_DE.

VII. CONCLUSION

This paper discusses a method for designing a fuzzy-rule-based classifiers using particle swarm optimization (PSO). Two different fuzzy classifiers are considered and optimized. The first proposed fuzzy classifier is based on Mamdani fuzzy inference system (M_PSO). In this model, PSO is used to optimize the parameters of the input membership functions, output membership functions and structure of the fuzzy rules. The second proposed fuzzy classifier is based on Takagi-Sugeno fuzzy inference system (TS_PSO). In this model, PSO is used to optimize premise parameters and consequent parameters. The performance of the two proposed fuzzy classifiers are tested on five real-world databases (Iris, Diabetes, Phoneme, Heart and Wine). Experimental results show that higher classification accuracy can be obtained with lower number of fuzzy rules by using the two proposed PSO fuzzy classifiers. Experimental results also show that, PSO has better performance in optimizing parameters of the two proposed fuzzy classifiers than genetic algorithm and deferential evolving algorithm.


Input 1and 2 membership functions


Input 3 and 4 membership functions

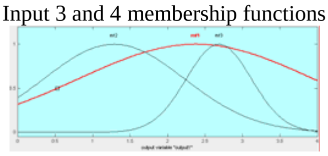
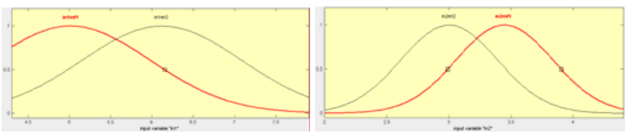

Output membership functions

Fig. 4. The optimized fuzzy membership functions using M_PSO fuzzy classifier for Iris database


Input 1 and 2 membership functions

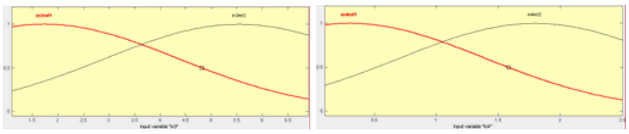
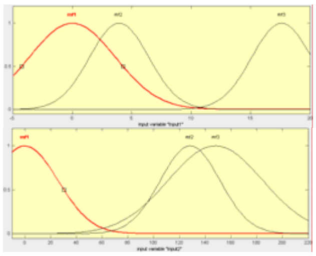
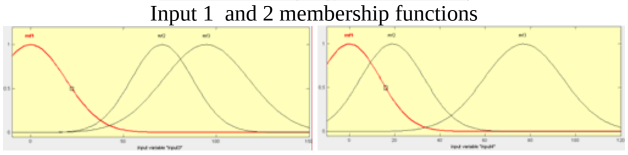
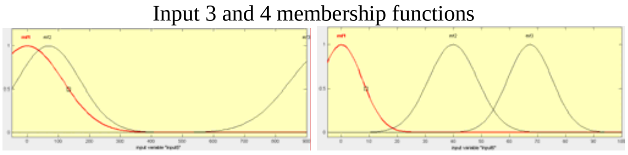
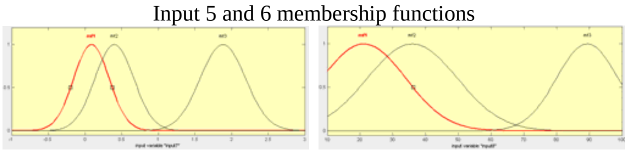

Input 3 and 4 membership functions

Fig. 5. The optimized fuzzy membership functions using TS_PSO fuzzy classifier for Iris database


Input 1 and 2 membership functions


Input 3 and 4 membership functions


Input 5 and 6 membership functions


Input 7 and 8 membership functions

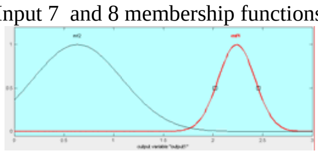
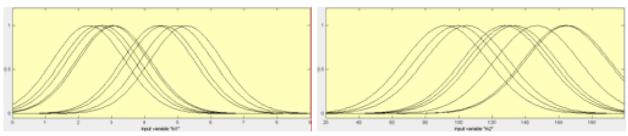
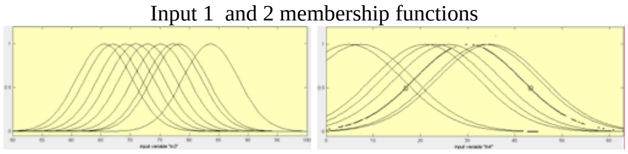
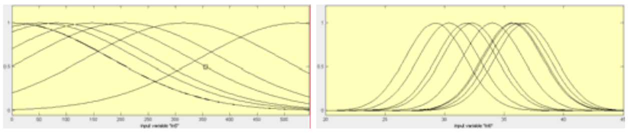

Output membership functions

Fig. 6. The optimized fuzzy membership functions using M_PSO fuzzy classifier for Diabetes database

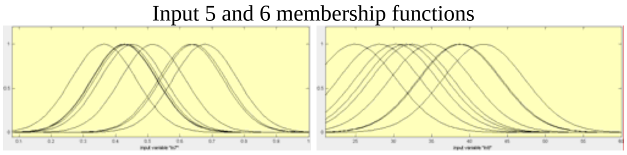

Input 1 and 2 membership functions


Input 3 and 4 membership functions

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:8, 2010



Input 5 and 6 membership functions



Input 7 and 8 membership functions

Fig. 7. The optimized fuzzy membership functions using TS_PSO fuzzy classifier for Diabetes database

TABLE I
COMPARISON BETWEEN DIFFERENT FUZZY CLASSIFIERS USING DIFFERENT DATABASES

		Mamdani Type				Takagi-Sugeno Type			
		Classifier 1		Classifier 2 (M_PSO)		Classifier 3		Classifier 4 (TS_PSO)	
Iris	Class 1	94.44		100.00		100.00		100.00	
	Class 2	100.00		87.50		93.75		100.00	
	Class 3	43.75		100.00		87.50		100.00	
	Over all / Rank	**79.40**	**4**	**95.83**	**2**	**93.75**	**3**	**100.00**	**1**
	No. of Rules	8		4		7		2	
Phoneme	Class 1	84.19		74.54		88.74		78.41	
	Class 2	57.95		83.04		66.76		84.69	
	Over all / Rank	**71.07**	**4**	**78.79**	**2**	**77.75**	**3**	**81.55**	**1**
	No. of Rules	8		6		10		8	
Diabetes	Class 1	73.24		73.24		81.94		69.57	
	Class 2	44.97		72.19		50.89		67.46	
	Over all / Rank	**59.11**	**4**	**72.72**	**1**	**66.41**	**3**	**68.51**	**2**
	No. of Rules	8		4		10		9	
Heart	Class 1	53.49		66.28		61.63		81.40	
	Class 2	61.19		79.10		65.67		67.16	
	Over all / Rank	**57.34**	**4**	**72.69**	**2**	**63.65**	**3**	**74.28**	**1**
	No. of Rules	6		4		10		5	
Wine	Class 1	48.28		93.10		89.66		96.55	
	Class 2	100.00		100.00		90.24		90.24	
	Class 3	0.00		0.00		94.44		100.00	
	Over all / Rank	**49.43**	**4**	**64.37**	**3**	**91.45**	**2**	**95.60**	**1**
	No. of Rules	15		4		3		3	

TABLE II
COMPARISON BETWEEN DIFFERENT OPTIMIZATION TECHNIQUES FOR THE PROPOSED MAMDANI FUZZY CLASSIFIER

		Mamdani Type					
		Classifier 2 (M_PSO)		Classifier 2 (M_GA)		Classifier 2 (M_DE)	
Iris	Class 1	100.00		100.00		100.00	
	Class 2	87.50		87.50		87.50	
	Class 3	100.00		100.00		100.00	
	Over all / Rank	**95.83**	**2**	**95.83**	**2**	**95.83**	**2**
	No. of Rules	4		4		4	
Phoneme	Class 1	74.54		70.28		66.32	
	Class 2	83.04		70.54		82.27	
	Over all / Rank	**78.79**	**1**	**70.41**	**3**	**74.29**	**2**
	No. of Rules	6		4		4	
Diabetes	Class 1	73.24		72.24		78.93	
	Class 2	72.19		69.82		60.95	
	Over all / Rank	**72.72**	**1**	**71.03**	**2**	**69.94**	**3**
	No. of Rules	4		10		6	
Heart	Class 1	66.28		69.77		76.74	
	Class 2	79.10		73.13		67.16	
	Over all / Rank	**72.69**	**1**	**71.45**	**3**	**71.95**	**2**
	No. of Rules	4		10		6	
Wine	Class 1	93.10		68.97		0.00	
	Class 2	100.00		100.00		92.68	
	Class 3	0.00		0.00		100.00	
	Over all / Rank	**64.37**	**1**	**56.32**	**3**	**64.23**	**2**
	No. of Rules	4		6		6	

TABLE III
COMPARISON BETWEEN DIFFERENT OPTIMIZATION TECHNIQUES FOR THE PROPOSED TAKAGI-SUGENO FUZZY CLASSIFIER

		Takagi-Sugeno Type					
		Classifier 2 (M_PSO)		Classifier 2 (M_GA)		Classifier 2 (M_DE)	
Iris	Class 1	100.00		100.00		100.00	
	Class 2	100.00		93.00		87.50	
	Class 3	100.00		100.00		100.00	
	Over all / Rank	**100.00**	**1**	**97.92**	**2**	**95.83**	**3**
	No. of Rules	2		2		2	
Phoneme	Class 1	78.41		79.89		80.73	
	Class 2	84.69		83.04		81.59	
	Over all / Rank	**81.55**	**1**	**81.47**	**2**	**81.16**	**3**
	No. of Rules	8		8		8	
Diabetes	Class 1	69.57		75.27		69.23	
	Class 2	67.46		60.27		64.50	
	Over all / Rank	**68.51**	**1**	**68.27**	**2**	**66.86**	**3**
	No. of Rules	9		9		8	
Heart	Class 1	81.40		76.74		75.58	
	Class 2	67.16		70.15		70.15	
	Over all / Rank	**74.28**	**1**	**73.45**	**2**	**72.87**	**3**
	No. of Rules	5		5		6	
Wine	Class 1	96.55		96.55		86.21	
	Class 2	90.24		95.12		95.12	
	Class 3	100.00		88.89		100.00	
	Over all / Rank	**95.60**	**1**	**93.52**	**3**	**93.78**	**2**
	No. of Rules	3		3		3	

REFERENCES

[1] Yager, R. R. and Filev, D. P. "Essentials of Fuzzy Modeling and Control," John Wiley, New York, U.S.A. (1994).

[2] Jang, J. S., Sun, C. T. and Mizutani, E., " Neuro-Fuzzy and Soft Computing," Prentice Hall, New Jersey, U.S.A. (1997).

[3] J.C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithm," Plenum Press, New York, 1981.

[4] J.C. Bezdek, "On the relationship between neural networks, pattern recognition and intelligence," *Int. J. Approx. Reasoning* 6 (2) (1992) 85–107.

[5] C.T. Lin, C.S.G. Lee, "Neural network-based fuzzy logic control and decision system," *IEEE Trans. Comput*. 40 (12) (1991) 1320–1336.

[6] C.F. Juang, J.Y. Lin, C.T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans. Syst. Man Cybern.* Part B 30 (2) (2000) 290–302.

[7] C.L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," *Proceedings of the Fourth International Conference on Genetic Algorithm*, San Diego, July 12–16, 1991, pp. 450–457.

[8] C.L. Karr, E.J. Gentry, "Fuzzy control of pH using genetic algorithm," *IEEE Trans. Fuzzy Systems.* 1 (1) (1993) 46–53.
[9] M. Setnes, H. Roubos, "GA-Fuzzy modeling and classification: complexity and performance," *IEEE Trans. Fuzzy Systems.* 8 (4) (2000) 509–522.
[10] Mahamed G.H. Omran, Andries P Engelbrecht, and Ayed Salman, "Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification," *Proceeding of world academy of science. Engineering and technology*, volume 9, November 2005 ISSN 1307-6884.
[11] Chia-Chong Chen "Design of PSO-based Fuzzy Classification Systems", *Tamkang Journal of Science and Engineering*," Vol. 9, No 1, (2006) pp. 63_70
[12] Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H., "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms," *IEEE Trans. Fuzzy Systems*, Vol. 3, (1995) pp. 260_270.
[13] Jang, J. S., "ANFIS: Adaptive-Network-Based Fuzzy Inference systems," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 23, (1993) pp. 665_685.
[14] Nozaki, K., Ishibuchi, H. and Tanaka, H., "Adaptive Fuzzy Rule-Based Classification Systems," *IEEE Trans. on Fuzzy Systems*, Vol. 4, No. 3, Aug., (1996) pp. 238_250.
[15] Wang, L. X. and Mendel, J. M., "Generating Fuzzy Rules by Learning from Examples," *IEEE Trans. On Systems, Man and Cybernetics*, Vol. 22, (1992) pp. 1414_ 1427.
[16] Wong, C. C. and Chen, C. C., "A GA-Based Method for Constructing Fuzzy Systems Directly from Numerical Data," *IEEE Trans. on Systems, Man and Cybernetics- Part B: Cybernetics*, Vol. 30, (2000) pp. 904_911.
[17] Simpson, P. K., "Fuzzy Min-Max Neural Networks- Part 1: Classification," *IEEE Trans. Neural Networks*, Vol. 3, Sep., (1992) pp. 776_786.
[18] Wong, C. C. and Chen, C. C., "A Hybrid Clustering and Gradient Descent Approach for Fuzzy Modeling," *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 29, (1999) pp. 686_693.
[19] Xie, X.L., and G.Beni, "A validity measure for fuzzy clustering," *IEEE trans. Pattern Anal. Mach. Intell*. 3(8): 1991 pp. 841-846.
[20] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent and Fuzzy Systems*, 1994 pp. 267-278,.
[21] Emami, M.R. Turksen, Burhan. Goldenberg A.A.,. "Development of A Systematic Methodology of Fuzzy Logic Modeling," *IEEE transaction on Fuzzy Systems*, 6(3) 1998.
[22] Salehfar, H. et al., "A systematic approach to linguistic fuzzy modeling based on input-output data," *IEEE Proceedings of the 2000 Winter Simulation Conference*, 2000.
[23] Vachkov, G. and Fukuda, T. , "Multilevel composite fuzzy models," *The Ninth IEEE International Conference on Fuzzy Systems* Volume 1, Issue , 2000 pp. 453 – 458.
[24] Hwang, H.S., "Automatic design of fuzzy rule base for modeling and control using evolutionary programming," *IEE Proc-Control Theory Appl*, Vol 146, 1999.
[25] A. Homaifar, E. McCormick, "Simultaneous design of membership functions and rule sets for fuzzy controller using genetic algorithms," *IEEE Trans. Fuzzy Systems.* 3 (2) (1995) 129–139.
[26] O. Cordon, F. Herrera, P. Villar, "Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the data base," *IEEE Trans. Fuzzy Systems.* 9 (4) (2001) 667–674.
[27] S.Y. Ho, H.M. Chen, S.J. Ho, T.K. Chen, "Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space," *IEEE Trans. Syst. Man Cybern*. Part B 34 (2) (2004) 1031–1044.
[28] Enwang Zhou, Alireza Khotanzad, "Fuzzy classifier design using genetic algorithms", *Pattern Recognition* 40 (2007) 3401 – 3414.
[29] Tanya Mirzayans, Nitin Parimi, Patrick Pilarski, Chris Backhouse,Loren Wyard-Scott, Petr Musilek , "A Swarm Based System for Object Recognition," *Neural Network World* 3/05, 2005, pp. 243-255.
[30] Ata E. Zadeh Shermeh and Reza Ghaderi, "An Intelligent System for Classification of the Communication Formats Using PSO," *Informatica* 32 (2008) pp. 213-218.
[31] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, Vol. 7, No. 1, 1975 pp. 1-13.
[32] Sugeno, M., Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.
[33] James Kennedy and Russell C. Eberhart, "Swarm Intelligence", Morgan Kaufmann Publishers, 2001.
[34] Mahamed G. H. Omran, "Particle Swarm Optimization Methods for Pattern Recognition and Image Processing", PhD. thesis , University of Pretoria, Pretoria, South Africa, November, 2004.
[35] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization," *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia (1995), 1942-1945.
[36] R. Forrest, " Genetic Algorithms: Principles of Natural Selection Applied to Computation," *Science*, Vol. 261, pp. 872-878, 1993.
[37] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization,* 11:341–359, 1997.

Hassan M. Elragal received the B.Sc., M.Sc. in Electrical Engineering form Alexandria University, Alexandria, Egypt, in 1991 and 1995 respectively. He received his Ph.D. degree in Electrical Engineering at Southern Methodist University, Dallas, Texas in July 2001. Dr. Hassan is currently an Associate Professor in the Department of Electrical Engineering, University of Bahrain. His research interest is in the area of Artificial Intelligent, Knowledge Base System, Expert Systems, Fuzzy logic, Neural networks, Genetic algorithm, Particle Swarm Optimization and their applications