

# Optimizing of Fuzzy C-Means Clustering Algorithm Using GA

Mohanad Alata, Mohammad Molhim, and Abdullah Ramini

**Abstract**—Fuzzy C-means Clustering algorithm (FCM) is a method that is frequently used in pattern recognition. It has the advantage of giving good modeling results in many cases, although, it is not capable of specifying the number of clusters by itself. In FCM algorithm most researchers fix weighting exponent ( $m$ ) to a conventional value of 2 which might not be the appropriate for all applications. Consequently, the main objective of this paper is to use the subtractive clustering algorithm to provide the optimal number of clusters needed by FCM algorithm by optimizing the parameters of the subtractive clustering algorithm by an iterative search approach and then to find an optimal weighting exponent ( $m$ ) for the FCM algorithm. In order to get an optimal number of clusters, the iterative search approach is used to find the optimal single-output Sugeno-type Fuzzy Inference System (FIS) model by optimizing the parameters of the subtractive clustering algorithm that give minimum least square error between the actual data and the Sugeno fuzzy model. Once the number of clusters is optimized, then two approaches are proposed to optimize the weighting exponent ( $m$ ) in the FCM algorithm, namely, the iterative search approach and the genetic algorithms. The above mentioned approach is tested on the generated data from the original function and optimal fuzzy models are obtained with minimum error between the real data and the obtained fuzzy models.

**Keywords**—Fuzzy clustering, Fuzzy C-Means, Genetic Algorithm, Sugeno fuzzy systems.

## I. INTRODUCTION

**P**ATTERN recognition is a field concerned with machine recognition of meaningful regularities in noisy or complex environments. In simpler words, pattern recognition is the search for structures in data. In pattern recognition, group of data is called a cluster [1]. In practice, the data are usually not well distributed; therefore the "regularities" or "structures" may not be precisely defined. That is, pattern recognition, by its very nature, an inexact science. To deal with the ambiguity, it is helpful to introduce some "fuzziness" into the formulation of the problem. For example, the boundary between clusters could be fuzzy rather than crisp; that is, a data point could belong to two or more clusters with different degrees of membership. In this way, the formulation is closer to the real-

world problem and therefore better performance may be expected. This is the first reason for using fuzzy models for pattern recognition: the problem by its very nature requires fuzzy modeling (in fact, fuzzy modeling means more flexible modeling-by extending the zero-one membership to the membership in the interval  $[0,1]$ , more flexibility is introduced).

The second reason for using fuzzy models is that the formulated problem may be easier to solve computationally. This is due to the fact that a non-fuzzy model often results in an exhaustive search in a huge space (because some key variables can only take values 0 and 1), whereas in a fuzzy model all the variables are continuous, so that derivatives can be computed to find the right direction for the search. A key problem is to find clusters from a set data points.

Fuzzy C-Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method was developed by Dunn [2] in 1973 and improved by Bezdek [3] in 1981 and is frequently used in pattern recognition.

Thus what we want from the optimization is to improve the performance toward some optimal point or points [4]. Luus [5] identifies three main types of search methods: calculus-based, enumerative and random.

Hall, L. O., Ozyurt, I. B. and Bezdek, J. C. [6] describe a genetically guided approach for optimizing the hard ( $J_f$ ) and fuzzy ( $J_m$ ) c-means functional used in cluster analysis. Our experiments show that a genetic algorithm ameliorates the difficulty of choosing an initialization for the c-means clustering algorithms. Experiments use six data sets, including the Iris data, magnetic resonance and color images. The genetic algorithm approach is generally able to find the lowest known  $J_m$  value or a  $J_m$  associated with a partition very similar to that associated with the lowest  $J_m$  value. On data sets with several local extreme, the GA approach always avoids the less desirable solutions. Deteriorate partitions are always avoided by the GA approach, which provides an effective method for optimizing clustering models whose objective function can be represented in terms of cluster centers. The time cost of genetic guided clustering is shown to make series of random initializations of fuzzy/hard c-means, where the partition associated with the lowest  $J_m$  value is chosen, and an effective competitor for many clustering domains.

The main differences between this work and the one by Bezdek et al [6] are:

Manuscript received March 18, 2008.

M. Alata and M. Molhim, Assistant professor are with Mechanical Engineering Department, Jordan University of Science and Technology (e-mail: alata@just.edu.jo).

Abdullah Ramini, Graduate student, is with Mechanical Engineering Department, Jordan University of Science and Technology.

- This work used the least square error as an objective function for the genetics algorithm but Bezdek [6] used  $J_m$  as an objective function.
- This work optimized the weighting exponent  $m$  without changing the distance function but Bezdek [6] keeps the weighting exponent  $m = 2.00$  and uses two different distance functions to find an optimal value.

## II. THE SUBTRACTIVE CLUSTERING

The subtractive clustering method assumes each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define the cluster center, based on the density of surrounding data points. The algorithm:

- Selects the data point with the highest potential to be the first cluster center
- Removes all data points in the vicinity of the first cluster center (as determined by radii), in order to determine the next data cluster and its center location
- Iterates on this process until all of the data is within radii of a cluster center
- 

The subtractive clustering method [7] is an extension of the mountain clustering method proposed by R. Yager [8].

The subtractive clustering is used to determine the number of clusters of the data being proposed, and then generates a fuzzy model. However, the iterative search is used to optimize the least square error from the model being generated and the test model. After that, the number of clusters is taken to the Fuzzy C-Means Algorithm.

## III. THE FUZZY C-MEANS CLUSTERING ALGORITHM

Fuzzy C-Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad (1)$$

Where  $m$  is any real number greater than 1, it was set to 2.00 by Bezdek.

$u_{ij}$  is the degree of membership of  $x_i$  in the cluster  $j$ ;  $x_i$  is the  $i^{\text{th}}$  of  $d$ -dimensional measured data;  $c_j$  is the  $d$ -dimension center of the cluster and  $\|\cdot\|$  is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership  $u_{ij}$  and the  $c_j$  cluster centers by:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (2)$$

$$c = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (3)$$

This iteration will stop when

$$\max_{ij} \left\{ u_{ij}^{(k+1)} - u_{ij}^{(k)} \right\} < \varepsilon \quad (4)$$

Where  $\varepsilon$  is a termination criterion between 0 and 1 and  $k$  are the iteration steps.

This procedure converges to a local minimum or a saddle point of  $J_m$ .

The algorithm is composed of the following steps:

1. Initialize  $U = [u_{ij}]$  matrix,  $U(0)$
2. At  $k$ -step: calculate the centers vectors  $C(k)=[c_j]$  with  $U(k)$

$$c = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (5)$$

3. Update  $U(k)$ ,  $U(k+1)$

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (6)$$

If  $\|U(k+1) - U(k)\| < \varepsilon$  then STOP; otherwise return to step 2.

## IV. THE GENETICS ALGORITHM

The GA is a stochastic global search method that mimics the metaphor of natural biological evolution. GAs operates on a population of potential solutions applying the principle of survival of the fittest to produce (hopefully) better and better approximations to a solution [9, 10]. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators

borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

V. RESULTS AND DISCUSSION

A complete program using MATLAB programming language was developed to find the optimal value of the weighting exponent. It starts by performing subtractive clustering for input-output data, build the fuzzy model using subtractive clustering and optimize the parameters by optimizing the least square error between the output of the fuzzy model and the output from the original function by entering a tested data. The optimizing is carried out by iteration. After that, the genetics algorithms optimized the weighting exponent of FCM. The same way, build the fuzzy model using FCM then optimize the weighting exponent *m* by optimizing the least square error between the output of the fuzzy model and the output from the original function by entering the same tested data. Fig. 1 at Appendix A shows the flow chart of the program.

The best way to introduce results is through presenting four examples of modeling of four highly nonlinear functions. Each example is discussed, plotted. Then compared with the best error of original FCM with weighting exponent (*m* =2.00).

A. Example 1 - Modeling a Two Input Nonlinear Function

In this example, a nonlinear function was proposed:

$$z = \frac{\sin(x)}{x} * \frac{\sin(y)}{y} \tag{7}$$

The range  $X \in [-10.5, 10.5]$  and  $Y \in [-10.5, 10.5]$  is the input space of the above equation, 200 data pairs are obtained randomly (Fig. 2).

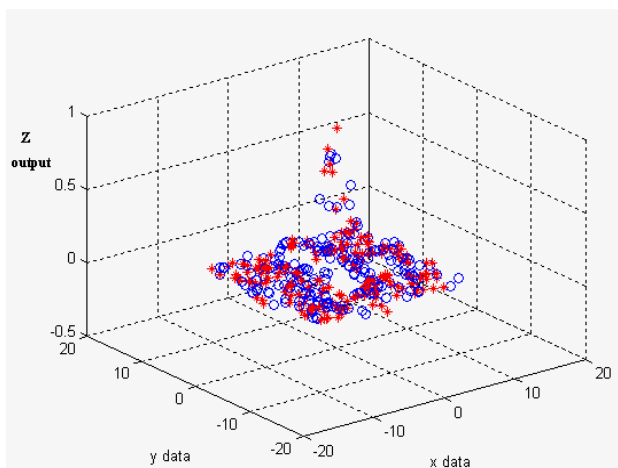


Fig. 2 Random data points of equation (7); blue circles for the data to be clustered and the red stars for the testing data

First, the best least square error was obtained for the FCM of weighting exponent (*m* =2.00) which is (0.0126 with 53 clusters). Next, the optimized least square error of the subtractive clustering is obtained by iteration that is (0.0115 with 52 clusters). We could see here that the error improves by (10%). Then, the clusters number is taken to the FCM algorithm, the error is optimized to (0.004 with 52 clusters) that means the error improves by (310%) and the weighting exponent (*m*) is (1.4149). Results are better shown in Table I at Appendix A.

B. Example 2 - Modeling a One Input Nonlinear Function

In this example, a nonlinear function was proposed also but with one variable *x*:

$$y = \frac{\sin(x)}{x} \tag{8}$$

The range  $X \in [-20.5, 20.5]$  is the input space of the above equation, 200 data pairs were obtained randomly and shown in Fig. 3.

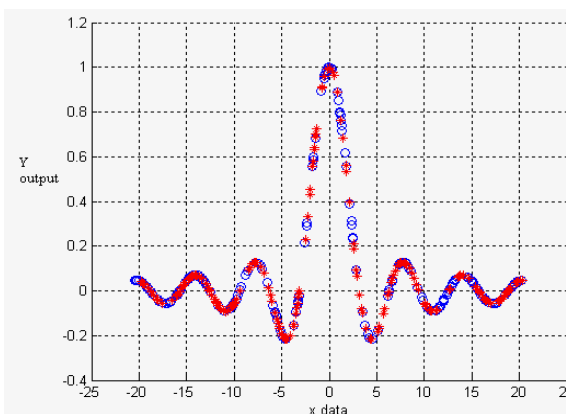


Fig. 3 Random data points of equation (8); blue circles for the data to be clustered and the red stars for the testing data

First, the best least square error is obtained for the FCM of weighting exponent (*m* =2.00) which is (5.1898\*e-7 with 178 clusters). Next, the least square error of the subtractive clustering is obtained by iteration which was (1\*e-10 with 24) clusters since this error pre-defined if the error is less than (1\*e-10). Then, the clusters number is taken to the FCM algorithm, the error is (1.2775\*e-12) with 24 clusters and the weighting exponent (*m*) is (1.7075). The improvement is (4\*e7) % and the number of clusters improved by 741%. Results are better shown in Table 2 at the Appendix.

C. Example 3 - Modeling a One Input Nonlinear Function

In this example, a nonlinear function was proposed:

$$y = (x - 3)^3 - 10 \tag{9}$$

The range  $X \in [1, 50]$  is the input space of the above equation, 200 data pairs were obtained randomly and are shown in Fig. 4.

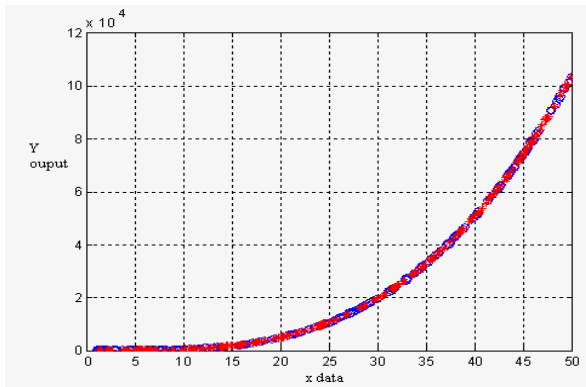


Fig. 4 Random data points of equation (9); blue circles for the data to be clustered and the red stars for the testing data

First, the best least square error is obtained for the FCM of weighting exponent ( $m = 2.00$ ) which is ( $3.3583 \times 10^{-17}$  with 188 clusters). Next, the least square error of the subtractive clustering is obtained by iteration which is ( $1.6988 \times 10^{-17}$  with 103 clusters) since the least error can be taken from the iteration. Then, the clusters number is taken to the FCM algorithm, the error was ( $2.2819 \times 10^{-18}$  with 103 clusters) and the weighting exponent ( $m$ ) is 100.8656. Here we could see that the number of clusters is reduced from 188 to 103 clusters that mean the number of rules is reduced and the error is improved by 14 times. Results are better shown in Table III at Appendix A.

The whole results are better shown in Table IV at Appendix A.

## VI. CONCLUSION

In this work, the subtractive clustering parameters, which are the radius, squash factor, accept ratio, and the reject ratio are optimized using the GA.

The original FCM proposed by Bezdek is optimized using GA and another values of the weighting exponent rather than ( $m = 2$ ) are giving less approximation error. Therefore, the least square error is enhanced in most of the cases handled in this work. Also, the number of clusters is reduced.

The time needed to reach an optimum through GA is less than the time needed by the iterative approach. Also GA provides higher resolution capability compared to the iterative search due to the fact that the precision depends on the step value in the "for loop function" which is max equal to 0.001 for the radius parameter in the subtractive clustering algorithm, but for GA, it depends on the length of the individual and the range of the parameter which is 0.00003 for the radius parameter also. So GA gives better performance and has less approximation error with less time.

Also it can be concluded that the time needed for the GA to optimize an objective function depends on the number and the length of the individual in the population and the number of parameter to be optimized.

APPENDIX

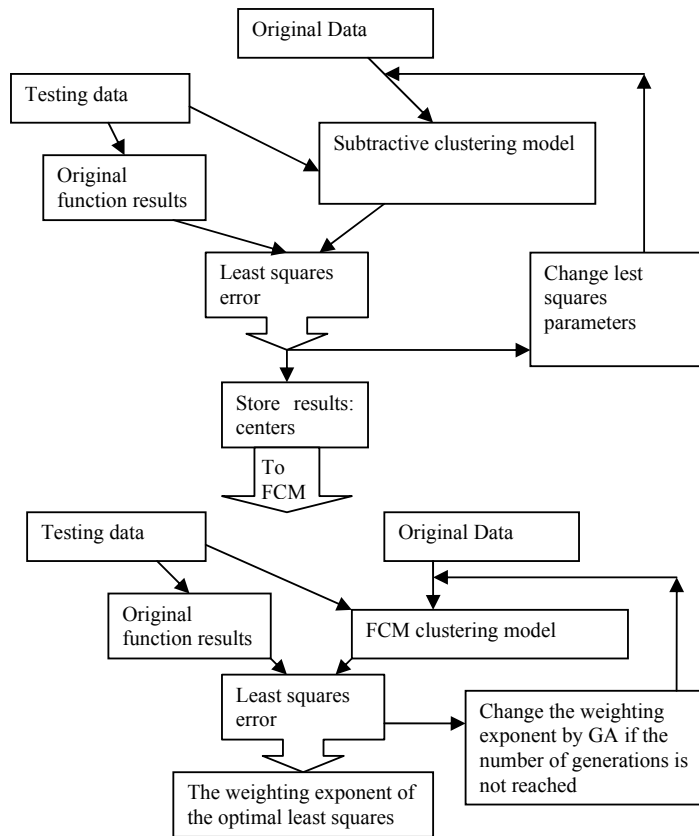


Fig. 1 The flowchart of the software

TABLE I  
THE RESULTS OF EQUATION (7): M IS THE WEIGHTING EXPONENT AND TIME IN SECONDS

Iteration M=2.00		Error		Clusters	
		0.0126		53	
Iteration for the subtractive and GA for FCM	Subtractive clustering		FCM		Time
	Error	Clusters	Error	m	
	0.0115	52	0.004	1.4149	19857.4

TABLE II  
THE RESULTS OF EQUATION (8): M IS THE WEIGHTING EXPONENT AND TIME IN SECONDS

Iteration m=2.00		Error		Clusters	
		5.1898 e-7		178	
Iteration for the subtractive and GA for FCM	Subtractive clustering		FCM		Time
	Error	Clusters	Error	M	
	1 e-10	24	1.2775 e-10	1.7075	3387.2

TABLE III  
THE RESULTS OF EQUATION (9): M IS THE WEIGHTING EXPONENT AND TIME IN SECONDS

Iteration M=2.00		Error		Clusters	
		3.3583 e-17		188	
Iteration for the subtractive and GA for FCM	Subtractive clustering		FCM		Time
	Error	Clusters	Error	m	
	1.6988 e-17	103	2.2819 e-18	100.8656	21440

TABLE IV  
THE FINAL LEAST SQUARE ERRORS AND CLUSTERS NUMBER FOR THE ORIGINAL FCM AND FOR THE FCM WHICH THEIR NUMBERS OF CLUSTERS WERE GOT FROM THE ITERATIVELY OR GENETICALLY OPTIMIZED SUBTRACTIVE CLUSTERING

The function	The original FCM (m=2)		Iteration then genetics		
	Error	Clusters	Error	Clusters	New (m)
<b>Eq (7)</b>	0.0126	53	0.004	52	1.4149
<b>Eq (8)</b>	5.1898 e-7	178	1.2775 e-10	24	1.7075
<b>Eq (9)</b>	3.3583 e-18	188	2.2819 e-18	103	100.8656

## REFERENCES

- [1] Li-Xin Wang, A Course in Fuzzy Systems and Control, (Prentice Hall, Inc.) Upper Saddle River, NJ 07458; 1997: 342-353.
- [2] J. C. Dunn, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics 3; 1973: 32-57.
- [3] Bezdek, J. C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, NY, 1981.
- [4] Beightler, C. S., Phillips, D. J., & Wild, D. J., Foundations of optimization (2<sup>nd</sup> ed.). (Prentice-Hall) Englewood Cliffs, NJ, 1979.
- [5] Luus, R. and Jaakola T. H. I., Optimization by Direct Search and Systematic Reduction of the Size of Search Region, AIChE Journal 1973; 19(4): 760 – 766.
- [6] Hall, L. O., Ozyurt, I. B. and Bezdek, J. C., Clustering with a genetically optimized approach, IEEE Trans. Evolutionary Computation 1999; 3(2), 103-112.
- [7] Chiu S. Fuzzy model identification based on cluster estimation. 1994, Journal of Intelligent and Fuzzy Systems; 2:267–78.
- [8] Yager, R. and D. Filev, Generation of Fuzzy Rules by Mountain Clustering, Journal of Intelligent & Fuzzy Systems, 1994; 2 (3): 209-219.
- [9] Ginat, D., Genetic Algorithm - A Function Optimizer, NSF Report, Department of Computer Science, University of Maryland, College Park, MD20740, 1988.
- [10] Wang, Q. J., Using Genetic Algorithms to Optimize Model Parameters, Journal of Environmental Modeling and Software 1997; 12(1):