# Real-coded Genetic Optimization of Fuzzy Clustering

Konstantinos Blekas and Andreas Stafylopatis
National Technical University of Athens
Department of Electrical and Computer Engineering
Phone: +301 7722508
Fax: +301 7722459
email: {kblekas,andreas}@softlab.ece.ntua.gr

ABSTRACT: A genetic approach is developed, which is suitable for the optimization of fuzzy $c$-means clustering. The approach is based on real encoding of the prototype variables (cluster centers) and uses appropriate genetic operators and techniques to optimize the clustering criterion. Experimental results concerning difficult clustering problems show that the proposed approach is very successful in generating fuzzy partitions and prototypes and outperforms the fuzzy $c$-means algorithm in terms of the correct placement of patterns into partitions.

## 1 INTRODUCTION

The task of pattern classification and recognition typically constitutes a major component of an intelligent diagnostic system. Pattern classification can be viewed as including two steps: first, a phase of clustering given samples, and second, classification of new samples based on the knowledge of clusters. Pattern clustering considers a set of unlabeled data objects and seeks to find natural groupings amongst the exemplars. The clusters are formed according to some predefined criterion, which is usually based on a distance measure. This and the number of clusters (which, in general, is assumed to be known a priori) constitute fundamental aspects of the clustering problem.

A broad spectrum of clustering algorithms attempt to generate a partition of the sample data through the (iterative) minimization of an objective function based on the clustering criterion. The kinds of partitions generated and the geometric structure of the clusters are closely related to the distance measure chosen and the objective function being optimized [7, 8, 12, 14]. The partitions are either hard, that is each sample point is unequivocally assigned to a cluster and is considered to bear no similarity to members of other clusters, or fuzzy, in which case a membership function expresses the degree of similarity between the sample and each cluster.

The fuzzy $c$-means (FCM) clustering approach [2] belongs to the general class of $c$-means partitioning models [1, 3] and has been extensively used in various types of pattern and image processing/analysis applications [10]. As an alternative to the typical iterative numerical process used by the algorithm to search for an optimal partitioning, genetic algorithms have recently been applied to this and related problems [4, 5, 6, 15]. Genetic algorithms [9, 13] are an increasingly important approach to finding good suboptimal solutions to large-scale optimization problems and have been shown to be capable of adaptive and robust search over a wide range of search space topologies.

As genetic algorithms are considered more natural for discrete optimization problems, most approaches to clustering have been based on a binary encoding of the parameter space [4, 5, 6]. An increasing interest, however, has focused on the use of real coding for problems with variables in continuous domains. This representation is very close to the natural formulation of many problems and allows the development of effective operators and powerful exploration techniques [11, 13]. In the present paper we develop a real-coded genetic approach to the FCM clustering problem and evaluate its effectiveness on well-known data sets allowing a measure of relative performance.

The next section provides a brief overview of FCM models, while Section 3 describes the proposed real-coded genetic approach to fuzzy clustering. Experimental results are presented and discussed in Section 4. Finally, Section 5 gives the summary and conclusions.

# 2 FUZZY $c$-MEANS CLUSTERING

The fuzzy $c$-means algorithm is one of the best known and best performing fuzzy clustering algorithms. Most of the results, extensions and analysis of the algorithm are due to Bezdek [1, 2, 3, 4, 5]. The FCM algorithm assumes that the number of clusters $c$ is known in advance and minimizes an objective function to find the best set of clusters.
Let $X = \{x_1, \ldots, x_n\}$ denote a set of unlabeled data points (vectors) in $\mathcal{R}^p$, and let $c$ be an integer, $1 < c < n$. Each $x_j$ is the numerical representation of $p$ features associated with a corresponding physical object. Given $X$, a *fuzzy c-partition* of $X$ is represented by a $c \times n$ fuzzy partition matrix $U = [u_{ij}]$ satisfying the conditions: $0 \leq u_{ij} \leq 1$ $(1 \leq i \leq c,\ 1 \leq j \leq n)$, $\sum_{i=1}^{c} u_{ij} = 1$ $(1 \leq j \leq n)$ and $\sum_{j=1}^{n} u_{ij} > 0$ $(1 \leq i \leq c)$, where each value $u_{ij}$ represents the membership of the $j$-th data point to the $i$-th cluster.
The clustering criterion used by the FCM algorithm is associated with the generalized least-squared errors functional

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^m D_{ij} \tag{1}$$

where $m > 1$ is a weighting exponent (degree of fuzzification), $V = [v_1, \ldots, v_c]$ $(v_i \in \mathcal{R}^p)$ is the vector of geometric centers (cluster prototypes), and $D_{ij}$ is some similarity (distance) metric between $x_j$ and $v_i$, which is taken equal to the squared distance

$$D_{ij} = \|x_j - v_i\|_A^2 = (x_j - v_i)^\top A (x_j - v_i) \tag{2}$$

the matrix $A$ being a positive definite $n \times n$ weight matrix. Following the choice of $A$, different norms can be identified implying different geometric and statistical properties of the generated partition. If $A$ is taken as the identity matrix $I$, the resulting Euclidean norm implies hyperspherical clusters.
Optimal fuzzy partitions of $X$ are defined as pairs $(U, V)$ that locally minimize $J_m$. For $m > 1$ and assuming that cluster prototypes are distinct from sample points, the following conditions are necessary (but not sufficient) for local optimality [1]:

$$v_i = \left( \sum_{j=1}^{n} (u_{ij})^m x_j \right) / \sum_{j=1}^{n} (u_{ij})^m \qquad 1 \leq i \leq c \tag{3}$$

$$u_{ij} = \left( \sum_{k=1}^{c} \left( \frac{D_{ij}}{D_{kj}} \right)^{1/(m-1)} \right)^{-1} \qquad 1 \leq i \leq c,\ 1 \leq j \leq n \tag{4}$$

Given an initial vector of centers, an alternate application of the above equations can be used for iterative optimization of $J_m$. This computation, which constitutes the basic FCM algorithm, exhibits good convergence properties. To check the quality of the clusterings obtained through local minima of $J_m$ extra validity criteria are usually applied. Generalizations of $J_m$ can be considered that imply various geometric properties for the partition. The FCM model can also be formulated in a connectionist manner [8], allowing for an efficient neuro-fuzzy approach to data clustering.

# 3 REAL-CODED GENETIC OPTIMIZATION

Genetic algorithms seem to offer a promising alternative in the direction of optimizing the FCM functional $J_m$. Representation and exploration of the problem parameter space can be based on encoding and evolving either both $U$ and $V$, or only one of the variables $U$ and $V$. In the latter case, one variable is eliminated and optimization of $J_m$ is performed over the other. Elimination of $U$ or $V$ can be obtained through appropriate substitution from (4) or (3) respectively into (1). In various related approaches binary coding has been generally employed for representing data partitioning [4, 5, 6, 15].
The traditional binary coding has serious drawbacks when applied to multidimensional problems of high numerical precision, since binary representation generates prohibitively large search spaces. For problems with variables over continuous domains, it seems natural to represent genes directly as floating-point numbers (problem variables) and chromosomes as vectors of real numbers, thus enabling the exploration of large domains without sacrificing precision or memory. Also, due to the ability of real-variable functions to follow slight changes in the variables, it is possible to perform fine local tuning of the obtained solutions and to operate in the presence of constraints.

In the approach presented here, we have considered floating-point encoding of the cluster centers $v_i = [v_{ij}]$ $(1 \le i \le c,\ 1 \le j \le p)$ as the appropriate parameter-space representation for the FCM algorithm. Thus, a chromosome is a vector (string) of total length $cp$, composed of $c$ subvectors of length $p$ corresponding to cluster centers. Given a vector $V$, the membership values $u_{ij}$ $(1 \le i \le c,\ 1 \le j \le n)$ are eliminated via substitution from (4) into (1) and the computed value of $J_m$ provides the fitness of the string. (Actually, the fitness is obtained via appropriate normalization of $J_m$ yielding a form suitable for maximization.) This approach is justified by the fact that, for fixed $V$, minimization of $J_m$ over $U$ can be done by applying (4) [1].
The main features of the proposed real-coded genetic algorithm (RCGA) are described next. Given an initial population of $K$ randomly created vectors, a reproduction procedure takes place repeatedly, during which members of the population are recombined and a new generation of vectors is created, until a maximum number of generations is attained.
At each generation step, for each current member $k$ we decide with probability $p_c$ (which generally assumes high values) whether crossover will be applied or not, and, in the positive case, another member $l$ is randomly selected and crossover is performed between the vectors corresponding to the two parents. A recombination operator that we have considered is a variant of single-point crossover. If we denote by $V^k$ and $V^l$ the vectors corresponding to members $k$ and $l$ respectively, a new vector $V$ is created as follows. A split point $s$ $(1 \le s \le cp-1)$ is randomly selected and two new vectors are created by swapping all elements between positions $s$ and $cp$. From the two generated vectors the one which is closer to $V^k$ (in the sense that its major part comes from $V^k$) is taken as the new vector and replaces $V^k$. In this way, the characteristics of individual population members are preserved to some extent, retarding the decrease of population diversity and the advent of premature convergence. Another crossover scheme that has been considered in our approach is the following. A position $s$ is selected at random within the range $1 \le s \le p-1$, where $p$ is the size of subvectors, and a crossover point is considered at position $s$ of each subvector of $V^k$ and $V^l$. Then crossover is performed between each pair of corresponding subvectors in a manner analogous to that described above at the whole vector level.
It must be noted that the above reproduction schemes are *synchronous*, in the sense that all $K$ children can be created simultaneously and independently based on the precedent generation, thus achieving a high degree of parallelism. In implementing the crossover schemes we may use either uniform selection of the mate vector $V^l$ or selection based on fitness value. In the latter case, members with high fitness value have more chances to be selected and recombined in the next generation. In any case, due to the synchronous scheme of reproduction, even the 'bad' members of the population will take part in the selection operation and, thus, population diversity is maintained.
Population diversity is further enhanced by applying a mutation operator. As is the case with crossover, several mutation operators for real coding have appeared in the literature [11, 13]. In our case, we have considered an adaptation of the *non-uniform mutation* described in [13]. The non-uniform mutation operator can be applied to each individual real-valued component of each subvector of a vector with probability $p_m$, which generally assumes very small values. Another approach would be to randomly select a subvector and an element within the selected subvector and apply the operator to that element. Yet another mechanism would be to apply the non-uniform mutation operator according to a given small probability to a whole vector (all elements) rather than to selected individual elements. In our implementation we have taken the first of the above options.
Let us assume that (regardless of the mechanism adopted) the domain of the element $v_{ij}$ $(1 \le i \le c, 1 \le j \le p)$ which will undergo mutation is $[a_{ij}, b_{ij}]$. If the operator is applied at generation step $t$ and $T$ is the maximum number of generations then the new value of the element will be

$$v'_{ij} = \begin{cases} v_{ij} + \Delta(t, b_{ij} - v_{ij}) & \text{if } \tau = 0 \\ v_{ij} - \Delta(t, v_{ij} - a_{ij}) & \text{if } \tau = 1 \end{cases} \tag{5}$$

with $\tau$ being a random digit that may take a value of 0 or 1, and

$$\Delta(t, y) = y\left(1 - r^{(1-\frac{t}{T})^b}\right) \tag{6}$$

where $r$ is a random number from the interval $[0, 1]$ and $b$ is a system parameter, which determines the degree of dependency on the number of generations. The function $\Delta$ returns a value in the range $[0, y]$ such that the probability of returning a number close to 0 increases as $t$ increases. (Other functions with the above property can be used as well.) Thus, the size of the interval for the element value becomes lower with the passing of generations. This causes the mutation operator to make a uniform search of the space initially, and a very local search at later stages. It is this property that provides the system with fine local tuning capabilities.

# 4 EXPERIMENTAL RESULTS

In our experiments we have considered a variety of classification problems. The selected databases include feature vectors that are defined on high-dimensional spaces and some of them are noisy containing hard examples.
The first data set that was used to test our real-coded genetic approach was the Fisher's Iris database. Iris data consists of 150 feature vectors described by 4 continuous-valued attributes with 3 classes. Another database was the B. German's Glass Identification database which is composed of 214 9-dimensional feature vectors and presents a classification of types of glass to one of two general classes (window glass or non-window glass). Finally, we have used the James Cook University Thyroid gland database which is a collection of 215 instances whose 5 attributes are all continuous-valued. The vectors belong to one of three decision classes that define a prediction of a patient's thyroid to the class of euthyroidism, hypothyroidism or hyperthyroidism. Among the three databases, the last one is considered the hardest example of clustering problem.
Experiments have been conducted with specific numbers of clusters for each of the three data collections. In all experiments the output of the algorithm was the best solution (in terms of the value of the generalized least-squared errors functional $J_m$) found during the search (5000 generation steps). After that, we were able to evaluate the classification performance of the system, by considering each generated cluster as a separate class and labeling each pattern with the cluster of maximum membership value (equation (4)). Then, classification rate can be obtained as the percentage of patterns which are correctly placed into clusters (in the sense that they are assigned the same label as the majority of patterns assigned to the cluster).
For all databases and for each number of clusters considered, a series of 10 experiments were performed using different seed values for the random number generator. The population size of the genetic algorithm was nearly double the respective size of chromosomes. The parameter values were $m = 2.0$ and $b = 4.0$, while the values of the crossover and mutation probabilities were respectively $p_c = 1.0$ and $p_m = 0.015$. Best results were found using the first formulation of crossover as described in Section 3 and the selection scheme based on fitness value.
To evaluate the effectiveness of our approach the same experiments were carried out using the standard fuzzy $c$-means (FCM) algorithm for the three databases. The parameter value of $m$ was the same used in the real-coded genetic algorithm (RCGA). In implementing both methods we have considered Euclidean norm for the clustering criterion.
Table 1 reports the results for the three respective data collections and for the two approaches considered. The best $J_m$ function value found and the classification rate are both displayed. Moreover, the average number of steps required to find the best value is also given in the case of the genetic algorithm. As can be observed from the table, the value of the objective function generated by the genetic formulation is in most cases lower than the corresponding value from fuzzy $c$-means. This means that a better set of clusters is produced leading to a higher classification rate. It must be noted that for small numbers of clusters the results are very close for both clustering techniques. As the number of clusters increases, however, the genetic-based technique is superior. Moreover, the superiority of the genetic approach is more apparent in the case of the Thyroid database which represents the hardest problem among the three benchmarks considered.

Table 1: Comparative results

| | RCGA | | | FCM | |
|---|---|---|---|---|---|
| *Nb. of clusters* | $J_m(U,V)$ | *Class. rate (%)* | *Avg. nb. steps* | $J_m(U,V)$ | *Class. rate (%)* |
| IRIS DATABASE | | | | | |
| 3 | 60.57 | 89.33 | 848 | 60.57 | 89.33 |
| 6 | 24.58 | 90.67 | 2315 | 24.67 | 90.67 |
| 9 | 15.53 | 98.00 | 2144 | 15.62 | 97.33 |
| 12 | 10.89 | 98.67 | 3226 | 10.95 | 98.00 |
| GLASS DATABASE | | | | | |
| 2 | 298.87 | 93.92 | 2589 | 298.93 | 93.92 |
| 4 | 133.89 | 95.79 | 2234 | 134.25 | 95.32 |
| 6 | 83.48 | 96.26 | 1790 | 84.61 | 94.39 |
| 8 | 58.74 | 99.53 | 945 | 61.52 | 98.13 |
| THYROID DATABASE | | | | | |
| 3 | 17757.25 | 79.06 | 2756 | 17757.88 | 79.06 |
| 6 | 6715.80 | 87.44 | 2620 | 6716.67 | 87.44 |
| 9 | 3998.42 | 88.37 | 1638 | 4014.10 | 87.44 |
| 12 | 2663.87 | 91.62 | 1816 | 2740.62 | 88.37 |

## 5 CONCLUSIONS

We have developed a genetic approach for the optimization of the clustering criterion in the fuzzy $c$-means algorithm. The proposed scheme is based on coding the space of cluster prototypes (centers) by means of floating point values and using appropriate crossover and mutation operators. The method has been tested experimentally on difficult clustering problems and has shown very good performance in terms of the rate of correct classifications during testing.

This work has allowed us to experiment with the use of real-coded genetic algorithms in a problem area that is particularly suited to this type of state-space representation. In this context, we assessed the effectiveness of problem-specific genetic operators in exploring the search space and performing local tuning of the obtained solutions. The results show that real encoding offers a promising approach to the solution of large optimization problems on continuous domains and can be successfully applied to enhance the performance of established techniques in various application areas. The advantage of genetic-based techniques is reinforced by the possibility of obtaining efficient parallel implementations in a straightforward manner, thus ensuring fast and effective solutions to hard problems.

## References


[1] Bezdek J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, 1981.

[2] Bezdek J.C., Ehrlich R., Full W.: FCM: The Fuzzy $c$-Means Clustering Algorithm. Computers and Geosciences, 10, 1984, 191–203.

[3] Bezdek J.C., Pal, S.K.: Fuzzy Models for Pattern Recognition. IEEE Press, 1992

[4] Bezdek J.C, Boggavarapu S., Hall L.O., Bensaid A.: Genetic Algorithm Guided Clustering. In: Proc. First IEEE Conf. on Evolutionary Computation, Orlando, Florida, June 1994, Vol. I, 34–39.

[5] Bezdek J.C., Hathaway R.J.: Optimization of Fuzzy Clustering Criteria Using Genetic Algorithms. In: Proc. First IEEE Conf. on Evolutionary Computation, Orlando, Florida, June 1994, Vol. II, 589–594.

[6] Buckles B.P., Petry F.E., Prabhu D., George R., Srikanth R.: Fuzzy Clustering with Genetic Search. In: Proc. First IEEE Conf. on Evolutionary Computation, Orlando, Florida, June 1994, Vol. I, 46–50.

[7] Dave R.N., Bhaswan K.: Adaptive Fuzzy $c$-Shells Clustering and Detection of Ellipses. IEEE Trans. on Neural Networks, 3, 1992, 643–662.

[8] Davis J., Warms T., Winters W.: A Neural Network Implementation of the Fuzzy $c$-Means Clustering Algorithm. In: Proc. Int. Joint Conf. on Neural Networks, Seattle, 1991.

[9] Goldberg D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[10] Hall L.O., Bensaid A.M., Clarke L.P., Velthuizen R.P., Silbiger M.S., Bezdek J.C.: A Comparison of Neural Network and Fuzzy Clustering Techniques in Segmenting Magnetic Resonance Images of the Brain. IEEE Trans. on Neural Networks, 3, 1992, 672–682.

[11] Herrera F., Lozano M., Verdegay J.L.: Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. Technical Report #DECSAI-95107, Universidad de Granada, Spain, Feb. 1995.

[12] Krishnapuram R., Nasraoui O., Frigui H.: The Fuzzy C Spherical Shells Algorithm: A New Approach. IEEE Trans. on Neural Networks, 3, 1992, 663–671.

[13] Michalewicz Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, 1992.

[14] Simpson P.K.: Fuzzy Min-Max Neural Networks-Part 2: Clustering. IEEE Trans. on Fuzzy Systems, 1, 1993, 32–45.

[15] Van Le T.: Evolutionary Fuzzy Clustering. In: Proc. IEEE Int. Conf. on Evolutionary Computation, Perth, Western Australia, Nov.-Dec. 1995, Vol. 2, pp. 753–758.