This paper has been accepted in Hybrid Intelligence Systems 2004 (to appear in IEEE Computer Society)

A Fuzzy Clustering Algorithm using Cellular Learning Automata based Evolutionary Algorithm

R. Rastegar A. R. Arasteh A. Hariri M. R. Meybodi

Computer Engineering Department
Amirkabir University of Technology
Tehran, Iran

Abstract

In this paper, a new fuzzy clustering algorithm that uses cellular learning automata based evolutionary computing (CLA-EC) is proposed. The CLA-EC is a model obtained by combining the concepts of cellular learning automata and evolutionary algorithms. The CLA-EC is used to search for cluster centers in such a way that minimizes the clustering criterion. The simulation results indicate that the proposed algorithm produces clusters with acceptable quality with respect to clustering criterion and provides a performance that is superior to that of the C-means algorithm.

1. Introduction

Clustering is an important unsupervised classification method used in identifying some inherent structure presenting in a set of data. The purpose of clustering is to group data into subsets that have useful meaning in the context of a particular problem [1]. Various clustering methods have been developed which may be classified into the following categories: hierarchical clustering, learning network clustering, mixture model clustering, objective-function-based clustering, and partition clustering, etc [2][3]. In a clustering problem, a data set, in N-dimensional Euclidean space $S=\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ where $\mathbf{x}_i \in R^N$ and M is the number of data is given and the problem is to cluster the data group into K clusters. Fuzzy clustering of the objects can be represented by a fuzzy membership matrix called a fuzzy partition. The set of all $K\times M$ non-degenerate constrained fuzzy partition matrices is denoted by Y_{fKM} and is defined as

$$Y_{fKM} = \{U \in R^{K\times M} \mid \sum_{i=1}^{K} U_{ij} = 1, 0 \le U_{ij} \le 1\}$$

Given a criterion for performing fuzzy clustering, the problem is to find the corresponding best partition matrix in Y_{fKM} . The clustering criterion considered here is the function,

$$J_r(U,\lambda) = \sum_{i=1}^{K}\sum_{j=1}^{M} (U_{ij})^{\theta} D_{ij}^2(\lambda_i, \mathbf{x}_k)$$

where, $U \in Y_{hKM}$ is a fuzzy partition matrix; $\theta \in [1,\infty)$ is the weighting exponent on each fuzzy membership; $\lambda=[\lambda_1,\ldots,\lambda_K]$ is a matrix of prototype N-dimensional parameters (cluster centers); and $D_{ik}(\lambda_i,\mathbf{x}_k)$ is a measure of the distance from $\mathbf{x}_k$ to the i^{th} cluster prototype. The Euclidean distance metric is used for all results reported here. A search is undertaken for a good representation of the cluster structure of U based on a (U,λ) minimizer of the above equation.

The Cellular Learning Automata (CLA), which was introduced for the first time in [4], is a mathematical model for modeling dynamical complex systems that consists of a large number of simple components. The simple components of CLA which have learning capabilities act together to solve a particular problem. This model has been applied to several problems such as image processing [5], channel assignment in a cellular mobile system [6], function optimization [7], modeling of rumor diffusion [8], VLSI Placement [9], and modeling commerce networks [10]. In [7], CLA and evolutionary computing are combined to obtain a new model called cellular learning automata based evolutionary computing (CLA-EC) for optimization problems. This model is capable of performing search in complex, large and multimodal landscapes. In this paper, a new clustering algorithm based on CLA-EC is proposed. The CLA-EC is used to search for cluster centers in such a way that minimizes the clustering criterion. Due to parallel nature of CLA-EC model, the proposed algorithm is appropriate for clustering large data sets. In order to demonstrate the effectiveness of the proposed algorithm, four different two-dimensional data sets and IRIS data sets are considered. Our experimental results of clustering indicate that the CLA-EC based clustering algorithm provides a performance that is superior to that of the C-means algorithm.

The rest of the paper is organized as follows. Section 2 gives a brief review of learning automata and CLA-EC model. The proposed clustering algorithm is described in section 3. Section 4 presents the simulation results for different data sets and the last section is the conclusion.

2. Cellular Learning Automata Based Evolutionary Computing

Learning Automata are adaptive decision-making devices operating on unknown random environments. The Learning Automaton has a finite set of actions and each action has a certain probability (unknown for the automaton) of getting rewarded by the environment of the automaton. The aim is to learn choosing the optimal action (i.e. the action with the highest probability of being rewarded) through repeated interaction on the system. If the learning algorithm is chosen properly, then the iterative process of interacting on the environment can be made to result in the selection of the optimal action. Figure 1 illustrates how a stochastic automaton works in feedback connection with a random environment. Learning Automata can be classified into two main families: fixed structure learning automata and variable structure learning automata (VSLA). In the following the variable structure learning automata is described.

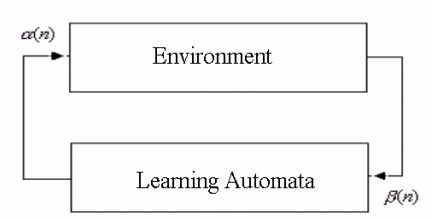


Fig.1. The interaction between learning automata and environment

A VSLA is a quintuple < *α, β, p, T(α,β,p)* >, where *α is* an action set with *s* actions, *β is* an environment response set and the probability set *p* containing *s* probabilities, each being the probability of performing every action in the current internal automaton state. Function *T* is the reinforcement algorithm, which modifies the action probability vector *p* with respect to the performed action and the received response. Let a VSLA operate in an environment with *β={0,1}*. Let $n \in N$ be the set of nonnegative integers that represent instances of iterations. A general linear schema for updating action probabilities can be represented as follows. Let action *i* be performed at instance *n* then

If *β(n)=0 (reward)* ,

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \qquad j \neq i$$

If *β(n)=1(penalty)* ,

$$p_i(n+1) = (1-b)p_i(n)$$
$$p_j(n+1) = (b/s-1) + (1-b)p_j(n) \quad j \neq i$$

Where *a* and *b* are reward and penalty parameters. When *a=b*, the automaton is called L_{RP}. If *b=0* the automaton is called L_{RI} and if $0<b<<a<1$, the automaton is called $L_{R\varepsilon P}$. For more Information about learning automata the reader may refer to [11] [12].

The CLA-EC model [7] is obtained by combining cellular learning automata [13] and evolutionary computing. This model is capable of performing search in complex, large and multimodal landscapes. In CLA-EC, similar to other evolutionary algorithms, the parameters of the search space are encoded in the form of genomes. Each genome has two components, model genome and string genome. Model genome is a set of learning automata. The set of actions selected by this set of learning automata determines the second component of genome (string genome). Based on a local rule, a reinforcement signal vector is generated and given to the set of learning automata. According to the learning algorithm, each learning automaton in the set of learning automata updates its internal state according to a learning algorithm. Then each learning automata in a cell chooses one of its actions using its probability vector. The set of actions chosen by the set of automata residing in a cell determines a candidate string genome that may replace the current string genome. The fitness of this string genome is then compared to the fitness of the string genome residing in that cell. If the fitness of the generated genome is better than the quality of the string genome of the cell, the generated string genome becomes the string genome of that cell. The process of generating string genome by the cells of the CLA-EC is repeated until a termination condition is satisfied. In order to have an effective algorithm, the designer of the algorithm must be careful about determining a suitable genome representation, fitness function for the problem at hand, the parameters of CLA such as the number of cells (population size), the topology, and the type of the learning automata for each cell. Assume $f: \{0,1\}^m \rightarrow \Re$ be a real function that is to be minimized. In order to use CLA-EC for optimization of function *f*, first a set of learning automata will be assigned to each cell of CLA-EC. The number of learning automata assigned to a cell of CLA-EC is the number of bits in the string genome representing points of the search space of *f*. Each automaton has two actions: 0 and 1. The CLA-EC iterates the following steps until the termination condition is met.

Step1: every automaton in cell *i* chooses one of its actions using its action probability vector.

Step 2: cell *i* generates a new string genome, η^i, by combining the actions chosen by the set of learning automata of cell *i*. The newly generated string genome is obtained by concatenating the actions of the automata (0 or 1) assigned to that cell.

Step 3: Every cell *i* computes the fitness value of string genome η^i; if the fitness of this string genome is better than the one in the cell, then the new string genome η^i becomes the string genome of that cell. That is

$$\xi_{n+1}^{i} = \begin{cases} \xi_n^i & f(\xi_n^i) \le f(\eta_{n+1}^i) \\ \eta_{n+1}^i & f(\xi_n^i) > f(\eta_{n+1}^i) \end{cases}$$

where ξ_n^i and η_n^i present the string genome and the new string genome of cell *i* at instance *n*.

Step 4: *Se* cells of the neighboring cells of the cell *i* are selected. This selection is based on the fitness value of the neighboring cells according to the truncation strategy [14].

```
While not done do
  For each cell i in CLA do in parallel
      Generate a new string genome;
      Evaluate the new string genome;
      If f(new string genome) < f(old string genome) then
          Accept the new string genome
      End if
      Select Se cells from neighbors of cell i ;
      Generate the reinforcement signal vector;
      Update internal state LAs of cell i
  End parallel for
End while
```

Fig. 2. Pseudocode of CLA-EC

Step 5: Based on the selected neighboring cells a reinforcement vector is generated. This vector becomes the input to the set of learning automata associated to the cell. Let *Ne(i)* be set selected neighbors of cell *i*. Define,

$$N_{i,j}(k) = \sum_{l \in Ne(i)} \delta(\xi_n^{l,j} = k),$$

where,

$$\delta(\exp) = \begin{cases} 1 & \text{if exp is true} \\ 0 & otherwise \end{cases}$$

$\beta^{i,j}$, the reinforcement signal given to the learning automaton *j* of cell *i* is computed as follows,

$$\beta_n^{i,j} = \begin{cases} u(N_{i,j}(1) - N_{i,j}(0)) & if\ \xi_t^{i,j} = 0 \\ u(N_{i,j}(0) - N_{i,j}(1)) & if\ \xi_t^{i,j} = 1 \end{cases}$$

where *u*(.) is a step function. The overall operation of CLA-EC is summarized in the algorithm of figure 2.

3. A CLA-EC based clustering algorithm

We propose to use the CLA-EC model to determine the *K* cluster centers of the data set in R^N; thereby clustering the set of *M* points of $S=\{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$. The aim is to search the cluster centers in such a way that clustering criterion be minimized. The proposed algorithm consists of three phases: preprocessing phase, the CLA-EC phase and the clustering phase.

3.1. Preprocessing phase

The purpose of preprocessing phase is to reduce the size of the search space in which CLA-EC will operate. To reduce the size of the search space, first the largest and the smallest values of each dimension of data set is found as follows:

$$\min_j = \min_{1 \le i \le M} \{\mathbf{x}_{i,j}\}$$

$$\max_j = \max_{1 \le i \le M} \{\mathbf{x}_{i,j}\}$$

$$\Delta_j = \max_j - \min_j$$

where $\mathbf{x}_{i,j}$ is the *j*th components of $\mathbf{x}_i$. Second, a new search space where we denote it by R' which is $R' = [0,\Delta_1] \times \ldots \times [0,\Delta_N]$ is defined. Where × is Cartesian product sign.

3.2. The CLA-EC phase

In the CLA-EC phase, clusters are optimized with respect to the clustering criterion. The characteristics of the applied CLA-EC are as follows.

String genome representation: Each string genome is represented by a binary string consisting of $M \times N$ parts where each part is a representation of an encoded real number. Let $\lambda'_{i,j}$ be $(i \times N + j)^{\text{th}}$ part of the string genome where *j* is the dimension of the center of cluster *i* in R'. If the binary representation of $\lambda'_{i,j}$ has w_{ij} bits then in a *N*-dimensional space with *K* clusters, the length of a string genome will be $m=\Sigma\Sigma w_{ij}$.

Fitness function: To compute the fitness value of ξ, first, we compute $\lambda'_{i,j}$ by decoding ξ, and set $\lambda_{i,j}$ to be $(\lambda'_{i,j}+min_j)$. The fitness value of genome is computed as follows:

$$f(\xi) = J_\theta(U, \boldsymbol{\lambda}),$$

where,

$$U_{ij} = 1/((D(\lambda_i, \mathbf{x}_j)^{\frac{2}{\theta-1}}) \times \sum_k (D(\lambda_k, \mathbf{x}_j)^{\frac{2}{1-\theta}}))$$

Parameters of CLA: A one-dimensional CLA with wrap around connection and with the neighborhood shown in figure 3a is used. The neighbors of cell *i* are cell *i-1* and cell *i+1*. The architecture of each cell is shown in figure 3b. Each cell is equipped with *m* learning automata. The string genome determiner compares the new string genome with the string genome residing in the cell. The string with the higher quality replaces the string genome of the cell. Depending on the neighboring string genomes and the string genome of the cell, a reinforcement signal will be generated by the signal generator.

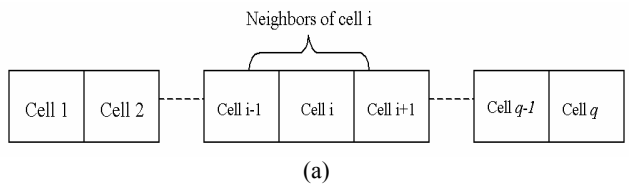
Termination Criteria: CLA-EC stops after a pre-specified number of iterations. The best string genome found in the last iteration is the solution to the clustering problem. For the experimentations that follow the maximum number of iteration is set to 500.

3.3. The Clustering Phase

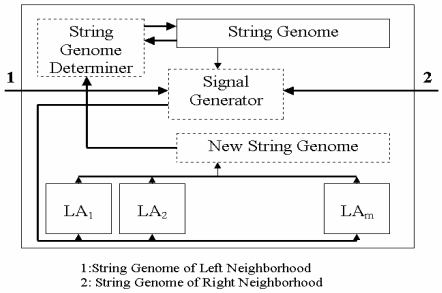
In this phase, the clusters are created using their centers, which are encoded in the best string genome reported by

the pervious phase. This is done by assigning each point $\mathbf{x}_i$, $i=1...M$, to one of the clusters C_k with center $\boldsymbol{\lambda}_k$ where,

$$C_k = \arg\max_{1 \le j \le K} U_{ij}$$



(a)



(b)

Fig. 3. The topology of the CLA-EC used in this paper.

4. Simulation Results

Several simulations are conducted in order to evaluate the effectiveness of the proposed method. The results are then compared with the results obtained for C-means algorithm. Simulations are conducted for five different data sets, which we call them Data 1, Data 2, Data 3, Data 4, and IRIS Data set. The characteristics of these data sets are given below.

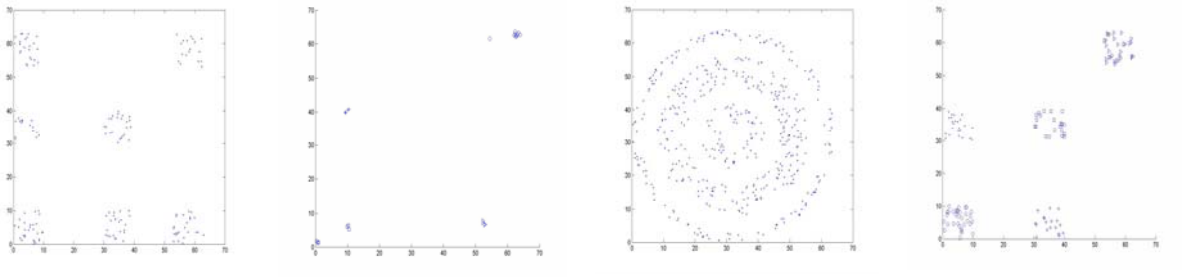
Data 1: a two-dimensional data set with 7 clusters and 160 points as shown in figure 4a.

Data 2: a two-dimensional data set with 4 clusters and 135 points as shown in figure 4b.

Data 3: a two-dimensional data set with 2 clusters and 477 points as shown in figure 4c.

Data 4: a two-dimensional data set with 4 clusters and 128 points as shown in figure 4d.

Iris data: This data set represents different categories of irises having four feature values. The four feature values represent the sepal length, sepal width, petal length and the petal width in centimeter. It has 3 clusters with 150 points.



(a) (b) (c) (d)

Fig. 4. a) Data 1 b) Data 2 c) Data 3 d) Data 4

For the sake of convenience in presentation, we use the notation *CLA-EC* (*automata(a,b), se ,q)* to refer to a CLA-EC algorithm with *q* cells, the number of selected cells is *Se* and learning automata's reward and penalty parameters are *a, b* respectively. For each simulation, the maximum number of iterations for CLA-EC is set to be 500. It is clear that as the mean and the standard deviation over all runs decrease, the quality of the clustering improves. With a careful inspection of the results reported in [15] it is found that as the number of cells increases, the mean and the standard deviation of the result decrease. Also, it has been found that, better results are obtained when each automaton uses L_{RP} or $L_{R\varepsilon P}$ learning algorithm and when *Se* is set to 2. We compare the results of the proposed algorithm with that of C-means algorithm. For this experimentation, q is equal to 5, each automaton uses L_{RP} learning algorithm with *a=b=0.01, Se* is *2* and the maximum number of iterations is set to be 500. For Data 1 it is found that the CLA-EC based algorithm provides the optimal value of 2322.11 in 100% of the runs (50 runs) whereas C-means algorithm attains this value in 74% of the runs and gets trapped at a local minimum for the other runs. For Data 2, CLA-EC based algorithm attains the best value of 201.23 in all of the runs. C-means, on the other hand, attains this value for 70% of the runs, while in other runs

it gets stuck at some of its local minima (such as 932.23). For Data 3, Data 4, and IRIS data set the CLA-EC based algorithm attains the best values of 15752.31, 1345.34, and 67.75 in 100%, 94%, and 82% of the runs, respectively. The best values attained by the C-means algorithm for these data sets are 15752.31, 1345.34, and 67.75 in 36%, 60%, and 78% of runs, respectively. Table 1 shows the summary of results of this experimentation. With a careful inspection of the results it is found that the CLA-EC(L_{RP}(0.01,0.01),2,5) performs better than C-means method for Data 1, Data 2, Data 3, Data 4.

5. Conclusions

In this paper, a new fuzzy clustering algorithm using cellular learning automata based evolutionary computing (CLA-EC) was introduced. The CLA-EC finds the cluster centers, in such a way that the clustering criterion be minimized. In order to demonstrate the effectiveness of proposed algorithm, four different two-dimensional data sets and an IRIS data set were considered. The results of simulations showed that our algorithm has a performance that is superior to that of the C-means algorithm. Due to the parallel nature of CLA-EC, the proposed algorithm is very suitable for clustering large data sets.

Table 1. The results of the CLA-EC(L_{RP}(0.01,0.01),2,5) algorithm (*maximum 500 iterations)* and the C-means algorithm for Data 1,2,3,4, IRIS - Columns 'Mean' and 'Std' show the mean and standard deviation over 50 runs.

Data Set	(CLA-EC) Mean	(CLA-EC) Std	(C-means) Mean	(C-means) Std
1	2322.11	0	3615.186	1965.651
2	201.23	0	319.46	23.57
3	15845.59	128.48	17312.24	420.73
4	1345.34	0	1675.34	239.43
Iris	69.12	4.37	68.80	7.042

6. References

[1] Bandyopadhyay, S., and Maulik, U., "An Evolutionary Technique based on K-means Algorithm for Optimal Clustering in R^N", Information Sciences, No. 146, PP. 221-237, 2002.

[2] Jain, A. K., Duin, R. P. W., and Mao, J., "Statistical Pattern Recognition: A Review", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol 22, PP. 4-37, 2000.

[3] Yang, M. S., and Wu, K. L., "A Similarity-Based Robust Clustering Method", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 26, No. 4, April 2004.

[4] Meybodi, M. R., Beygi, H., and Taherkhani, M., "Cellular Learning Automata", in Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, PP. 153-163, 2001.

[5] Meybodi, M. R., and, Kharazmi, M. R., "Image Restoration Using Cellular Learning Automata", in Proceedings of the Second Iranian Conference on Machine Vision, Image Processing and Applications, Tehran, Iran, PP. 261-270, 2003.

[6] Beigy, H., and Meybodi, M. R., "A Self-Organizing Channel Assignment Algorithm: A Cellular Leaning Automata Approach", Vol. 2690 of Springer-Verlag Lecture Notes in Computer Science, PP. 119-126, Springer-Verlag, 2003.

[7] Rastegar, R., and Meybodi, M. R., "A New Evolutionary Computing Model based on Cellular Learning Automata", to Appear in Proceeding of 2004 IEEE Conference on Cybernetics and Intelligent Systems (IEEECIS2004), Singapore, 2004.

[8] Meybodi, M. R., and Taherkhani, M., "Application of Cellular Learning Automata to Modeling of Rumor Diffusion", in Proceedings of 9th Conference on Electrical Engineering, Power and Water institute of Technology, Tehran, Iran, PP. 102-110, May 2001.

[9] Meybodi, M. R., and Mehdipour, F., "VLSI Placement Using Cellular Learning Automata", in Proceedings of 8th Annual International Computer Society of Iran Computer Conference CSICC2001, Mashhad, Iran, PP. 195-203, 2003.

[10] Meybodi, M. R., and Khojaste, M. R., "Application of Cellular Learning Automata in Modeling of Commerce Networks", in Proceedings of 6th Annual International Computer Society of Iran Computer Conference CSICC2001, Isfahan, Iran, PP. 284-295, 2001.

[11] Narendra, K. S., and Thathachar, M. A. L., *Learning Automata: An Introduction*, Printice-Hall Inc, 1989.

[12] Thathachar, M. A. L., Sastry, P. S., "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.

[13] Beigy, H., and Meybodi, M. R., "A Mathematical Framework for Cellular Learning Automata", Advanced in Complex Systems, to appear.

[14] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.

[15] Rastegar, R., Arasteh, A., Hariri, A., and Meybodi, M. R., "Fuzzy Clustering based on CLA-EC", Technical Reports, Computer Eng Department, Amirkabir University, Tehran, Iran, summer 2004.