ELSEVIER

Fuzzy Sets and Systems 113 (2000) 381–388

FUZZY sets and systems

www.elsevier.com/locate/fss

Entropy-based fuzzy clustering and fuzzy modeling

J. Yao*, M. Dash, S.T. Tan, H. Liu

Department of Information Systems and Computer Science, National University of Singapore, 10 Kent Ridge, Crescent, Singapore 119260, Singapore

Received May 1997; received in revised form January 1998

Abstract

Fuzzy clustering is capable of finding vague boundaries that crisp clustering fails to obtain. But time complexity of fuzzy clustering is usually high, and the need to specify complicated parameters hinders its use. In this paper, an entropy-based fuzzy clustering method is proposed. It automatically identifies the number and initial locations of cluster centers. It calculates the entropy at each data point and selects the data point with minimum entropy as the first cluster center. Next it removes all data points having similarity larger than a threshold with the chosen cluster center. This process is repeated till all data points are removed. Unlike previous methods of its kind, it does not need to revise entropy value for each data point after a cluster center is determined. This saves a lot of time. Also it requires just two parameters that are easy to specify. It is able to find the natural clusters in the data. The clustering method is also extended to construct a rule-based fuzzy model. A new way of estimating initial membership functions for fuzzy sets is presented. The experimental results show that the fuzzy model is good in predicting output variable values. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy sets; Cluster analysis; Entropy

1. Introduction

Cluster analysis has been a fundamental research area in data analysis and pattern recognition. Clustering helps find natural boundaries in the data. Fuzzy clustering is suitable for handling the problem of vague boundaries of clusters and provides a basis for constructing rule-based fuzzy model that has simple representation and good performance for non-linear problems. In fuzzy clustering, the requirement of a crisp partition of the data is replaced by a weaker requirement of fuzzy partition [10], where the association among data is represented by fuzzy relations. Among fuzzy clustering methods, the fuzzy c-mean (FCM) method [1] is one of the most popular methods. A number of methods [14,11,3,8,5] have been proposed to improve the performance of original FCM algorithm and to decrease its computational complexity. One important issue in fuzzy clustering is identifying the number and initial locations of cluster centers. In original FCM algorithm, these initial values are specified manually.

Yager and Filev [13] and Chiu [4] proposed methods that automatically determine the number of clusters and the locations of cluster centers. Chiu's method is a modification of Yager and Filev's mountain method in which the *potential* of each data point is determined based on its distance from other data points. A data point having many data points nearby

* Corresponding author.
E-mail address: yaojun@iiscs.nus.edu.sg (J. Yao)

PII: S0165-0114(98)00038-4

has a high *potential* and the data point having the highest *potential* is chosen as the first cluster center. Next the *potentials* of all other data points are revised (reduced) according to their distance from the chosen cluster center. This procedure is repeated till no data point has its *potential* above a threshold. This method requires values for three parameters: (1) the radius beyond which data points have little influence on the calculation of *potential*, (2) the amount of *potential* to be subtracted from each data point as a revision after a cluster center is determined, and (3) the threshold that *potential* uses to stop selecting cluster centers. Although these methods are simple and effective, they are computationally expensive as after each determination of a cluster center the *potential* values of all other data points are revised. With a larger number of cluster centers, the problem of recalculating the potential values aggravates. Besides, values of the three parameters vary a lot from one data set to another.

We propose to use an entropy measure in place of the *potential* measure. Our method does not require any revision after finding a cluster center unlike previous methods. Entropy at each data point is calculated based on a similarity measure. Data points in the middle of the clusters will have lower entropy than other data points; in other words they have better chance of being selected as cluster center. The data point having minimum entropy is chosen as the first cluster center. Data points having similarity with this cluster center less than a threshold are removed from being considered as cluster centers in the rest of the iterations. The rationale here is that the data points having high similarity with the chosen cluster center should belong to the same cluster with a high probability, and are not likely to be centers of any other clusters. This is repeated until there is no data point left. An advantage of this method compared to the other methods is its lower computational complexity as the entropy values are calculated only once. An additional advantage is the fewer number of parameters required and the parameters take values in a narrow range.

In the next section, we introduce an entropy measure for fuzzy clustering. In Section 3 we describe the algorithm using the measure and discuss some of its practical aspects. In Section 4 we outline how a fuzzy model can be constructed based on the fuzzy clustering method. Experimental results of a number of data sets for both fuzzy clustering and fuzzy modeling are shown in Section 5. The paper concludes in Section 6 with discussions of future work.

2. Entropy measure for cluster estimation

Consider a set of N data points in an M-dimensional hyper-space, where each data point x_i, $i=1,\ldots,N$, is represented by a vector of M values (i.e., $x_{i1},x_{i2},\ldots,x_{iM}$). Values of each dimension are normalized in the range [0.0–1.0]. Let us assume that there are a number of clusters in the data. For a data point to be a cluster center the ideal situation is when it is close to the data points in the same cluster and is away from the data points in other clusters. This situation prohibits the data points in the border of the cluster from becoming cluster centers.

2.1. The measure

We say the data has orderly configurations if it has distinct clusters, and has disorderly or chaotic configurations otherwise. From entropy theory [6], we know that entropy (or probability) is less for orderly configurations, and more for disorderly configurations. If we try to visualize the complete data set from individual data points then an orderly configuration means that for most individual data points there are some data points close to it (i.e., they probably belong to the same cluster), and others away from it. In a similar reasoning, a disorderly configuration means that most of the data points are scattered randomly. So, if we evaluate entropy at each data point then the data point with minimum entropy is a good candidate for cluster center. This may not be valid if the data has outliers in which case they should be removed first before determining the cluster centers. More about this is discussed in the next section.

An entropy value between two data points is in the range [0.0–1.0]. It is very low (close to 0.0) for very close or very distant pairs of data points, and very high (close to 1.0) for those data points separated by the distance close to the mean distance of all pairs of data points. We use a similarity measure (S) that is based on distance, and assumes a very small value (close to 0.0) for very close pairs of data points that probably fall to the same cluster, and a very large value (close to 1.0) for very distant pairs of data points that probably fall to different clusters. Entropy at one data point with respect to another data point is given by: $E = -S\log_2 S - (1-S)\log_2(1-S)$. E assumes the maximum value of 1.0 when S is 0.5, and the minimum value of 0.0 when S is 0.0 or 1.0 [9]. The total entropy value at a data point x_i with respect to all other data points is calculated as

$$E_i = -\sum_{j\in X}^{j\neq i} (S_{ij}\log_2 S_{ij} + (1-S_{ij})\log_2(1-S_{ij})), \tag{1}$$

where S_{ij} is the similarity between x_i and x_j normalized to [0.0–1.0]. The similarity between two data points, S_{ij} is given by

$$S_{ij} = \mathrm{e}^{-\alpha D_{ij}}, \tag{2}$$

where D_{ij} is the distance between the data points x_i and x_j. If we plot similarity against distance, then the curve will have a larger curvature for larger α. The experiments with various values for α suggest that it should be robust for all kinds of data sets, and not just for certain data sets. In this work, α is calculated automatically by assigning similarity of 0.5 in formula (2) when the distance of two data points have the mean distance of all pairs of data points. This produces good results as shown by the experimental results in Section 5. Mathematically, it is given as $\alpha = -\ln 0.5/\bar{D}$, where $\bar{D}$ is the mean distance among the pairs of data points in a hyper-space. Hence, α is determined by the data and can be calculated automatically.

3. Fuzzy clustering method using entropy measure

In this section we describe Entropy-based Fuzzy Clustering (EFC) algorithm and then discuss some of its practical aspects.

We evaluate entropy at each data point and select the data point with the least entropy value as the first cluster center. Then we remove this cluster center and all the data points that have similarity with this center greater than a threshold β from being considered for cluster centers in the rest of the iterations. Next the second cluster center is selected that has the least entropy value among the remaining data points, and again this cluster center and the data points having similarity greater than β are removed. This process is repeated until no data point is left. The parameter β can be viewed as a threshold of similarity or association value among the data points in the same cluster. It takes a value in the range [0.0–1.0], and a value of 0.7 is quite robust as shown by our experiments.

In the algorithm T is the input data with N data points each of which has M dimensions.

Algorithm.

```
EFC ($T$)
1. Calculate entropy for each $x_i$ in $T$ for $i = 1,\dots,N$.
2. Choose $x_{i_{\mathrm{Min}}}$ with least entropy.
3. Remove $x_{i_{\mathrm{Min}}}$ and the data points having similarity
   with $x_{i_{\mathrm{Min}}}$ greater than $\beta$ from $T$.
4. If $T$ is not empty go to Step 2.
```

If the data has outliers that are very distant from the rest of the data then EFC may prefer these points for the cluster centers as the entropy for these will be less. To tackle this problem we introduce a parameter γ that acts as a threshold between potential cluster centers and the outliers. Before selecting a data point as cluster center we count the number of data points that have similarity with this data point greater than β. If this number is less than γ, then the data point is unfit to be a cluster center and should be rejected. In our experiments we choose 5% of the total number of data points as the threshold value for γ. This also helps prevent over-fitting of the data.

4. Identification of fuzzy model

In the last two sections we designed a fuzzy clustering method, EFC. In this section we will show that EFC can be applied to construct a fuzzy model for predicting values of output variables. We will follow a modeling procedure commonly seen in other fuzzy methods (e.g., [12,4,13]), and primarily discuss the design issues of the fuzzy model based on EFC.

Sugeno and Takagi [12] suggested the representation of fuzzy model in the form of fuzzy rules. A fuzzy rule is based on a fuzzy partition of the input space. In each fuzzy subspace, an input–output relation is formed. For a data point with output variable value unknown, the input variable values of the data point are applied to all rules and each rule gives a value by fuzzy reasoning, the predicting output value is obtained by aggregation of all the values given by rules. Consider a set of c cluster centers $(x_1^*, x_2^*, \ldots, x_c^*)$ in an M-dimensional hyper-space. Say, the last L dimensions are output dimensions, and the first $M - L$ dimensions are input dimensions. Then each data point x_k^* can be decomposed into two vectors: y_k^* in $(M - L)$-dimensional input space and z_k^* in L-dimensional output space. Then a fuzzy model is a collection of c rules in the form:

If U is close to y_k^* Then V is close to z_k^*,

where U is the input vector and V is the output vector of a data point. The membership function representing the degree to which rule k is satisfied is given as

$$\mu_k = \mathrm{e}^{-\sigma_k \|u - y_k^*\|^2}, \tag{3}$$

where u is the input vector, $U = u$, and σ_k is automatically calculated from the data (more on this later in this section). $\|\cdot\|$ denotes Euclidean distance. The output vector, $V = v$, is calculated as

$$v = \frac{\sum_{k=1}^{c} \mu_k z_k^*}{\sum_{k=1}^{c} \mu_k}. \tag{4}$$

We can write a fuzzy rule in a more specific form:

If u_1 is A_{k1} AND u_2 is A_{k2} AND $\cdots$ AND
u_{M-L} is $A_{k(m-l)}$ Then V is v, for $k = 1, \ldots, c$

where u_j is the jth input variable. A_{kj} is given by

$$A_{kj} = \mathrm{e}^{-\sigma_k (u_j - y_{kj}^*)^2}, \tag{5}$$

where y_{kj}^* is jth element of kth cluster center y_k^*. The AND operator is implemented by multiplication.

The parameter σ_k is crucial for the fuzzy model to perform well. The initial value of this parameter is generally provided by users or is an arbitrary value. The initial value can be optimized in order to improve the performance of a model. The conventional way is using the gradient descent technique and a back-propagation algorithm where the parameters are optimized by an iterative process. The results of optimization and efficiency of this process depend on the initial values.

We propose a simple and automatic way of estimating the initial value of σ_k: for each cluster center, we find the closest cluster center to it and calculate the distance $D_{\min}$ between these two cluster centers. The initial value of σ_k is automatically obtained as

$$\sigma_k = \frac{-\ln 0.5}{D_{\min}/2}. \tag{6}$$

This formula implies that in the fuzzy set around a cluster center, if there is a data point mid-way between the cluster center and its closest neighboring cluster center then the membership value of this data point belonging to the fuzzy set should be 0.5. The experimental results in next section show the effectiveness of this estimation.

5. Experimental study

We want to show that:

1. EFC can find natural clusters in the data (Section 5.1), and
2. a fuzzy model based on EFC is good in predicting output variable values (Section 5.2).

In order to accomplish task 1 we choose four data sets with class labels. The idea is to check whether EFC is able to find the clusters in which data with the same label gather together. For task 2, we choose the data which has continuous output variable to show that the fuzzy model built using EFC is able to predict output variable values.

5.1. Cluster analysis on several data sets

To show that EFC is able to find the natural clusters we follow this procedure: (a) four data sets having class labels are chosen (summarized in Table 1), (b) EFC is applied to these data sets after removing the class labels, and (c) results of EFC are compared with the given classes, and discrepancies arising from mismatch between the given classes and the achieved

Table 1
Summary of data sets

Data	No. of data points	No. of classes
Iris	150	3
Wine	178	3
Thyroid	244	3
BC	699	2

Table 2
Clusters of Iris data

		No. of cluster $\beta = 0.70$			No. of cluster $\beta = 0.75$				No. of cluster $\beta = 0.50$	
		1	2	3	1	2	3	4	1	2
Actual	1	50			50				50	
class	2		3	47		2	28	20	4	46
label	3		48	2		39	9	2		50

Table 3
Clusters of Wine data

		No. of cluster $\beta = 0.7$						No. of cluster $\beta = 0.75$		No. of cluster $\beta = 0.5$		
		1	2	3	4	5	6	1	2	1	2	3
Actual	1	41	18					24	25	26		33
class	2	10		1	2	30	28	70	1		2	69
label	3			6	42			27	21		45	3

Table 4
Clusters of Thyroid data

		No. of cluster $\beta = 0.75$			No. of cluster $\beta = 0.70$	
		1	2	3	1	2
Actual	1	76	44	30	115	35
class	2	30	5		9	26
label	3		13	16	28	1

Table 5
Clusters of BC data

		No. of cluster $\beta = 0.6$		No. of cluster $\beta = 0.50$			
		1	2	3	4	1	2
Actual	1	450	8	451	1	2	4
label	2	26	215	35	43	106	57

clusters are reported. For (c) we obtain crisp clusters by assigning each data point to the cluster with which it has highest similarity.

EFC needs values for two parameters: the β and the threshold γ for discarding the outliers, if any, in the data. The first parameter β may take different values for these data sets, but our experiments show a value around 0.7 is robust. The second parameter γ is required only when the data has outliers; so if one knows that certain data set has no outlier then there is no need for specifying the value of this parameter. For data sets with outliers we fixed the value of γ to 5%, otherwise it takes the default value of 0%. The results are shown in Tables 2–5.

For Iris data, with β value 0.7, the discrepancies between the actual clusters and the achieved clusters is very few (a total of 5 in 150 data points). When β value decreases to 0.5, only two clusters are obtained. In this case, class 2 and class 3 are almost merged to one cluster as data points in class 1 are well separated from the data points in these two classes. When β is 0.75, 4 clusters are obtained. Notice that the class of 2 and 3 are partitioned to 3 clusters and the discrepancies increase to 13.

In case of Wine data, the results are good when β is 0.7. Although EFC obtained 6 clusters as compared to the given 3 classes, the number of discrepancies is only 13 in 178 data points. For other values of β (0.75 and 0.5), discrepancies are much higher. When β is 0.75, only two clusters are formed and the number of discrepancies is 73. This is because the number of data points having similarity greater than 0.75 may not be more than the number required by the

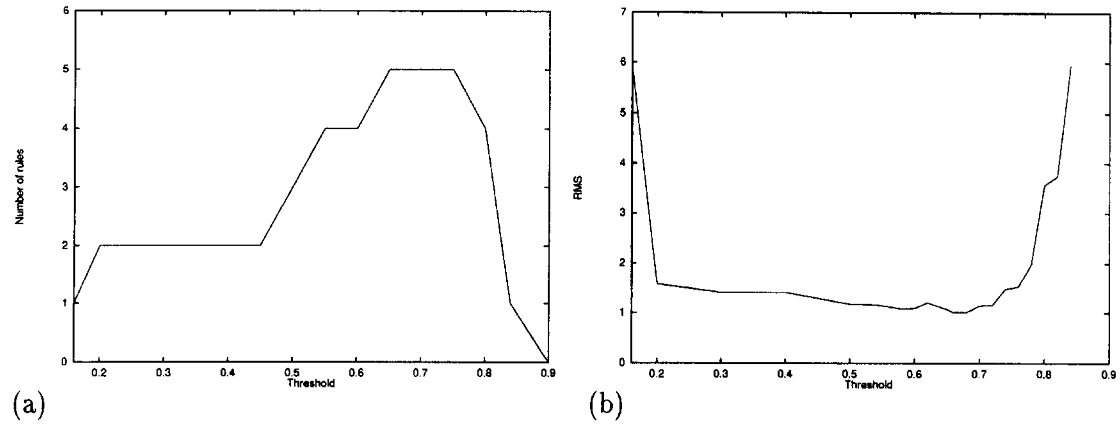


Fig. 1. Model size and prediction error as functions of β.

outlier condition. Here we use 5% of the total number of data points as the outlier condition. When β is 0.5, the number of discrepancies is 38 in 178 data points.

As for Thyroid data, there are many discrepancies. When we plotted the data on the 3D space which is determined by the three most important variables chosen by a feature ranking method, we noticed substantial overlapping among the three given classes in one distinct cluster while the rest of the data points of class 2 and class 3 spread widely along certain dimensions. Therefore, although the clustering result of EFC is consistent with the distribution of data points, the cluster analysis is not appropriate for this kind of data.

The data points of BC (Breast Cancer) data are uniformly distributed in the hyper-space and all variables take integer values in the range [1–10]. With the requirement that the number of the neighboring points with similarity larger than β around a cluster center should be more than 5% of total number of data points, when β is larger than 0.7, no cluster is formed. When β is 0.6, EFC produces two classes with just 34 discrepancies in a total of 699 data points.

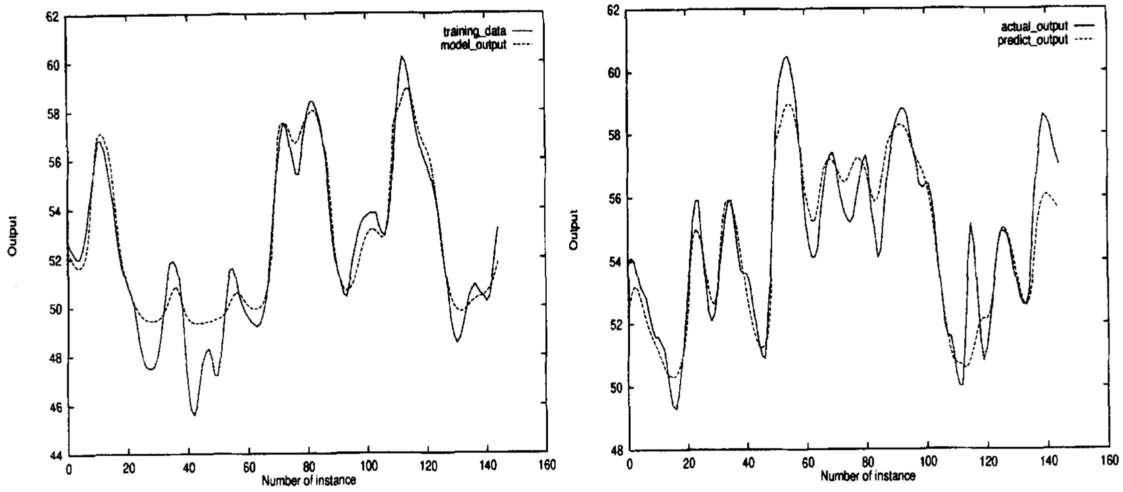
Going by the above clustering results, we recommend the user to specify a β value around 0.70 as a first choice. The time complexity of our clustering algorithm is $O(N^3M)$ which is less than most other cluster methods. Moreover, no need of optimizing parameters saves a lot of time, the number of clusters is automatically determined.

5.2. Construction of fuzzy model for Gas data

In this section we show that EFC can be applied to construct a fuzzy model for predicting output variable values. We consider Gas data taken from [2] that is commonly used by the community. It is about time series process, and has an input vector $u(t)$ and a single output $y(t)$. Ten candidate input variables $(y(t-1),\ldots,y(t-4),u(t-1),\ldots,u(t-6))$ of past time affect the present time output $y(t)$. The data set consists of 290 data points (input–output pairs). We use the first 145 data points to train the fuzzy model. The other 145 data points are used as the testing data for comparing the predicted output values with the actual output values. The results are shown in Figs. 1 and 2.

Fig. 1(a) displays the model size and Fig. 1(b) displays the root-mean-square (RMS) error of testing data for different β values. The RMS error reaches the minimum when β is 0.68 and the rule number (5) of fuzzy model is acceptable. For lower values of β the fuzzy model becomes simpler and the RMS error increases, while for higher values of β the model over-fits the data and results in high RMS error. From Fig. 1, we can see that we can get a desirable model when β is specified around 0.7.

In Fig. 2 we show the comparison between the predicted values and the actual values. Fig. 2(a) shows the comparison between the actual output and the output of the model on training data when we specify β as 0.68 in the construction of the fuzzy model. Fig. 2(b)



(a) training data

(b) testing data

Fig. 2. Comparison of output of the model and actual values.

shows a comparison between the actual output and the output of the model for testing data. The results show that our fuzzy model is able to predict the output variable values.

6. Discussion and conclusion

In this work, we focus on clustering of continuous (numerical) data. As for nominal data, we contend that it is more suitable to be handled by other kinds of algorithms such as conceptual clustering [7]. Applying our method to mixed data with both nominal and continuous attributes is another interesting issue. It is challenging to find an effective similarity measure for mixed data which is one of our future research directions.

In this paper we present a new method, EFC, for fuzzy clustering. An entropy measure is defined for identifying the number of clusters and their centers. This measure does not need to revise values of all other data points after determining a cluster center unlike other similar methods. It needs fewer number of parameters and they take values in a small range. If the data has no outliers then EFC needs only one parameter, β. It signifies the degree of association among the data points in a cluster. A higher value will produce clusters with more closely associated data points, and a lower value will produce clusters with more loosely associated data points. Hence, it incorporates user's subjective judgement regarding the clusters in a data. We built a fuzzy model using EFC and proposed a way of estimating the initial membership function of a fuzzy set. Experimental results show that EFC is able to find natural clusters in the data and can be applied to construct rule-based fuzzy model.

Acknowledgements

We would like to thank Dr. Stephen L. Chiu in Rockwell Science Center for providing testing data.

References

[1] J.C. Bezdek, Cluster validity with fuzzy sets, J. Cybernet. (1974) 58–71.
[2] G.E.P. Box, G.M. Jenkins, Times Series Analysis, Forecasting and Control, Holden-Day, San Francisco, 1970.
[3] T.W. Cheng, D.B. Goldgof, L.O. Hall, Fast clustering with application to fuzzy rule generation, FUZZY-IEEE/IFES (1995) 2289–2295.
[4] S.L. Chiu, Fuzzy model identification based on cluster estimation, J. Intell. Fuzzy Systems 2 (1994) 267–278.
[5] H. Choe, J.B. Jordan, On the optimal choice of parameters in a fuzzy c-means algorithm, FUZZY-IEEE (1992) 349–354.
[6] J.D. Fast, Entropy: the significance of the concept of entropy and its applications in science and technology, in:

The Statistical Significance of the Entropy Concept, Philips Technical Library, Eindhoven, 1962.

[7] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine Learning (1987) 139–172.

[8] K. Kamei, D.M. Auslander, K. Inoue, A fuzzy clustering method for multidimensional parameter selection in system with uncertain parameters, FUZZY-IEEE (1992) 355–362.

[9] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty and Information, in: Uncertainty and Information, Prentice-Hall International Editions, 1988.

[10] G.J. Klir, B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, in: Pattern Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[11] S. Medasani, J. Kim, R. Krishnapuram, Estimation of membership functions for pattern recognition and computer vision, in: Fuzzy Logic and its applications to engineering, Information Sciences and Intelligent System, Kluwer Academic Publishers, Dordrecht, 1995, pp. 45–54.

[12] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1988) 15–33.

[13] R.R. Yager, D.P. Filev, Generation of fuzzy rules by mountain clustering, J. Intell. Fuzzy Systems 2 (1994) 209–219.

[14] B. Yuan, G.J. Klir, J.F. Swan-Stone, Evolutionary fuzzy c-means clustering algorithm, FUZZY-IEEE (1995) 2221–2226.