Ignorance-based fuzzy clustering algorithm

Aranzazu Jurio, Miguel Pagola, Daniel Paternain, Edurne Barrenechea, Jose Antonio Sanz and Humberto Bustince
Dpt. de Automática y Computación, Public University of Navarra
Campus Arrosadía s/n, P.O. Box 31006 Pamplona, Spain
{aranzazu.jurio,miguel.pagola,daniel.paternain,edurne.barrenechea,joseantonio.sanz,bustince}@unavarra.es

***Abstract*—In this work an ignorance-based fuzzy clustering algorithm is presented. The algorithm is based on the Entropy-based clustering algorithm proposed by Yao et al. [1]. In our proposal, we calculate the total ignorance instead of using the entropy at each data point to select the data point as the first cluster center. The experimental results show that the ignorance-based clustering improves the data classification made by the EFC in image segmentation.**

***Index Terms*—Clustering, Ignorance functions, Restricted equivalence functions, Image segmentation.**

I. Introduction

Cluster analysis or clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.

Among fuzzy clustering methods, the fuzzy c-means (FCM) method [2], [3] is one of the most popular methods. One important issue in fuzzy clustering is identifying the number and initial locations of cluster centers. In classical FCM algorithm, these initial values are specified manually.

In [4], [5] some methods are proposed that automatically determine the number of clusters and the location of cluster centers by the potential of each data point. Yao et al. in [1] proposed a clustering method based on the entropy measure in place of the potential measure.

A problem of this algorithm is to choose the correct threshold in order to bound the maximun distance of the elements belonging to a cluster. In [6] the data spread is considered to determine the adaptive threshold within parameters optimized by a genetic algorithm. The algorithm [7] eliminates threshold constraint to detect possible cluster members. Cluster centers are formed with minimum entropy, instead of using a fixed-threshold, a decision region is formed with the use of maximum mutual information.

The EFC algorithm has some problems in some practical applications. For example, for a dataset with two classes that are symmetrically sparse, the EFC algorithm does not provide a correct data partitioning due to the first cluster center being located in the middle of all data, so clusters obtained with this method are not correct. Therefore we propose to use other measures that satisfy other properties, instead of distances and the fuzzy entropy, to solve this problem of the symmetric data. In this work we are going to replace the distance between elements by restricted equivalence functions and the entropy by ignorance functions.

This paper is organized in the following way: In Section II the Entropy-based Fuzzy Clustering algorithm is explained. In Section III we explain our proposed algorithm. In Section IV we show some experimental results. Finally, some conclusions are exposed.

II. Entropy-based Fuzzy Clustering

The basis of EFC is to find the elements which, if they are supposed to be the center of the cluster, then the entropy of the total set of elements is the lowest. This entropy is calculated for each element taking into account the similarity of said element with all the elements left (S_{ij}), with the following expression:

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

Such a way the algorithm first selects the element with lowest entropy as the center of the first cluster. Once it is selected, it is deleted from the center candidates list. Also, the elements with similarity with the center bigger than a given threshold (β) are deleted. This similarity threshold between the elements of a cluster must be valued between 0 and 1. Experimentally, the authors have determined that a good and robust value of this threshold is 0.7. Once those elements are deleted from the candidates list, the element with lowest entropy is taken as the center of the second cluster. The process is repeated until the candidates list is empty.

Given a set T with N data, the algorithm is outlined as follows:

1. Calculate the entropy of each $x_i \in T$, for $i = 1, \ldots, N$.
2. Choose $x_{i_{Min}}$ achieving the lowest entropy.
3. Delete from T, $x_{i_{Min}}$ and all the data whose distance to it is smaller than β.
4. If T is not empty, go to step 2.

We must notice that it is not possible to choose a priori the number of clusters in which the algorithm must split the data. The user must modify the value of threshold β to obtain the number of desired clusters.

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.194

III. Ignorance-based Fuzzy Clustering

The EFC algorithm does not obtain the correct partition of the data depending on the dataset. For example, if we have a dataset with two different classes in which the data is sparse symmetrically (see figure 2 - Data 1), the EFC does not split the data correctly. In this case the element with the lowest entropy is located in the center of the picture, grouping in the same class almost all of the elements. This is one of the problems of the EFC that we want to improve.

We propose to replace two concepts of the EFC algorithm. First one, we are going to replace the distance between elements by restricted equivalence functions to calculate the similarity between elements. In addition we are going to use ignorance functions instead entropy functions so, for us, the center of the cluster is the element which causes that the partition of the data has the lowest ignorance.

A. Equivalence between two data points

To calculate how similar are two elements, we are going to use restricted equivalence functions.

Definition 1: [8] A function $REF : [0,1]^2 \to [0,1]$ is called a restricted equivalence function, if it satisfies the following conditions:

(1) $REF(x,y) = REF(y,x)$ for all $x, y \in [0,1]$;
(2) $REF(x,y) = 1$ if and only if $x = y$;
(3) $REF(x,y) = 0$ if and only if $x = 1$ and $y = 0$ or $x = 0$ and $y = 1$;
(4) $REF(x,y) = REF(n(x), n(y))$ for all $x, y \in [0,1]$, n being a strong negation;
(5) For all $x, y, z \in [0,1]$, if $x \leq y \leq z$, then $REF(x,y) \geq REF(x,z)$ and $REF(y,z) \geq REF(x,z)$.

An example of REF that we will use within the algorithm is the following:

$$REF(x,y) = (1 - |x^3 - y^3|)^3$$

In figure 1 we can see the plot of previous REF.

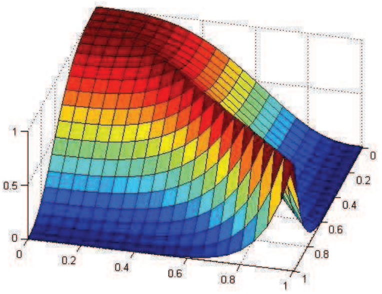


Fig. 1. Function REF

Each element is defined by some attributes (characteristics). In this way, to calculate the equivalence between two elements we aggregate the values of REF between every attribute. Such a way the equivalence between two elements with n attributes is the following:

$$\begin{aligned} &Eq(x,y) = \\ &\quad M(REF(x_1,y_1), REF(x_2,y_2), \ldots, REF(x_n,y_n)) \end{aligned} \tag{1}$$

where M is defined as follows:

Definition 2: [9]An n-ary aggregation function is a function

$$M : [0,1]^n \to [0,1]$$

such that

(i) $M(x_1, \ldots, x_n) \leq M(y_1, \ldots, y_n)$ whenever $x_i \leq y_i$ for all $i \in 1, \ldots, n$.
(ii) $M(0, \ldots, 0) = 0$ and $M(1, \ldots, 1) = 1$.

An example of aggregation function that satisfies these properties is the arithmetic mean.

With equation (1) we calculate how much similar are two elements, instead of calculate the distance as is done in [1].

B. Ignorance functions

We first introduced Ignorance functions in [10] applied to image thresholding. When choosing membership functions that represent the image in the process of thresholding, evidently, there are pixels of the image for which the expert is absolutely sure that the representation chosen is the correct one. Nevertheless, there are also pixels for which the expert does not know if the representation taken is the best.If the membership degree to the object of a pixel is 1, then the expert has total knowledge (total sureness) that the pixel belongs to the object (background).Also if the membership degree of a pixel to the object is 0.5 and to the background is 0.5, we say that the expert is totally ignorant, total doubt,of whether the pixel belongs to the object (background).We proposed in [10] to represent the expert's lack of knowledge by means of what we called *Ignorance functions*. The considerations above and others led us to present the following definition:

Definition 3: A function

$$G_u : [0,1]^2 \to [0,1]$$

is called an ignorance function if it satisfies the following conditions:

(G_u1) $G_u(x,y) = G_u(y,x)$ for all $x, y \in [0,1]$;
(G_u2) $G_u(x,y) = 0$ if and only if $x = 1$ or $y = 1$;
(G $_u3$) If $x = 0.5$ and $y = 0.5$, then $G_u(x,y) = 1$;
(G_u4) G_u is decreasing;
(G_u5) G_u is continuous.

In [10] we presented a construction method of G_u functions form t-norms. An example of ignorance function is:

$$G_u(x,y) = \begin{cases} 4(1-x)(1-y) & \text{if } (1-x)\cdot(1-y) \leq 0.25 \\ \frac{1}{4((1-x)(1-y))} & \text{otherwise} \end{cases} \tag{2}$$

The Ignorance functions estimate the uncertainty that exists when there are two membership functions. However, in this case we want to calculate the total ignorance of a set of elements by means of their membership degree to a cluster. If we are completely sure that an element is the center of the cluster, then we have no ignorance. In the case of the membership of the element to the cluster is 0.5 the we said that we have total ignorance. Therefore we can use following expression to calculate the ignorance associated to each element by means of ignorance functions:

$$Ig(x) = G_u(x, 1-x)$$

For equation (2) the Ignorance function for a single element results:

$$Ig(x) = \begin{cases} 4(1-x)x & \text{if } x \geq 0.5 \\ \frac{1}{4(1-x)x} & \text{otherwise} \end{cases} \quad (3)$$

This way we can obtain expressions that satisfy the properties that are demanded to fuzzy entropies. It is left as future work to prove in which cases the ignorance functions are entropies.

C. Algorithm

```
1. Calculate the ignorance of each $x_i \in T$, for $i = 1, \ldots, N$.
   1.1. Calculate the restricted equivalence between each pair of data.
        $Eq(x_i, x_j) =$
            $M(REF(x_{i1}, y_{x1}), REF(x_{i2}, y_{x2}), \ldots, REF(x_{in}, x_{jn}))$
        for all $j = 1..N$ where $j \neq i$                                  (4)
   1.2. Calculate the ignorance of each pair of data:
        $Ig(Eq(x_i, x_j)) = (1 - Eq(x_i, x_j)) * -Eq(x_i, x_j)$
        .
   1.3. Calculate the ignorance of each datum.
        $I_T(x_i) = \frac{\sum_{j=1}^{N} Ig(Eq(x_i, x_j))}{N}$
        if we are working with a set of $N$ data.
2. Choose $x_{i_{Min}}$ achieving the lower ignorance.
3. Delete from $T$, $x_{i_{Min}}$ and all the data whose distance to it is smaller than $\beta$.
4. If $T$ is not empty, go to step 2.
```

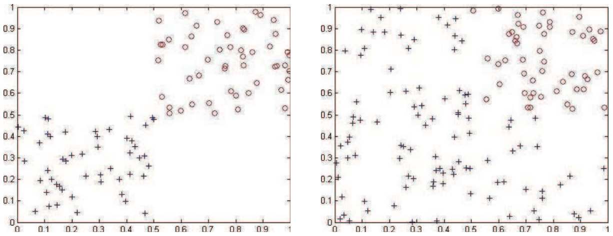
IV. Experimental Results

In this section we are going to compare the results obtained by our method, previously explained, and the ones obtained by the Entropy-based Fuzzy Cluster method (EFC), on which our proposal is based.

We make two kind of experiments. We first prove synthetic data, and then we prove with real images.

A. Synthetic data

In this experiment we prove the algorithm in two cases: on one side we prove it with two perfectly linearly separable and symmetric datasets, and on the other one we prove it with two linearly separable datasets with two straight lines. These two data distributions can be seen in Figure 2.



Data 1 Data 2

Fig. 2. Original data distribution

To make the comparison, we show the results obtained by our proposed method and the ones obtained by the EFC, studied in Section II. As we have already said, to apply any of these two algorithms it is necessary to previously choose the threshold that fixes the maximun distance among the data which belong to the same cluster. In this experiment, as we know that the original data are divided into two classes, we use the suitable threshold in each case to obtain a solution with two clusters. To prove which of the solutions is more likely to the original data distribution, we use the accuracy rate, it means, the number of well classified data respect to the whole data.

The graphic results obtained can be viewed in Figures 3 and 4, following in both cases the same distribution: the image (a) represents the result obtained by our method and the image (b) is the one obtained by the EFC. Below each image it is shown the threshold used in that result.

The numeric comparison of the accuracy is shown in Table I.

	Ignorance	EFC
Data1	100%	53%
Data2	72%	35.33%
Mean	86%	44.17%

TABLE I
Result comparison of experiment 1

As it can be checked, the first data distribution is relatively easy. However, the EFC algorithm is not able to correctly split the clusters, because it chooses as the first centroid the datum situated in the center of the graphic, so it is not able to identify the two classes. Nevertheless, the algorithm based on restricted equivalence and ignorance functions identify with a 100% of accuracy each datum.

In the second kind of data distribution, it can be clearly viewed that, despite any of the two studied methods gets the ideal solution, the one obtained by equivalence and ignorance functions is better than the one obtained by similarity and entropy functions. In this case, the EFC creates a class with only three data, which is far away from what it should be, while our proposed method creates two clusters more well-balanced, which is closer to the ideal distribution.

Therefore, for these two data distributions, we can conclude that our proposal improves 41.835% on average the EFC algorithm.

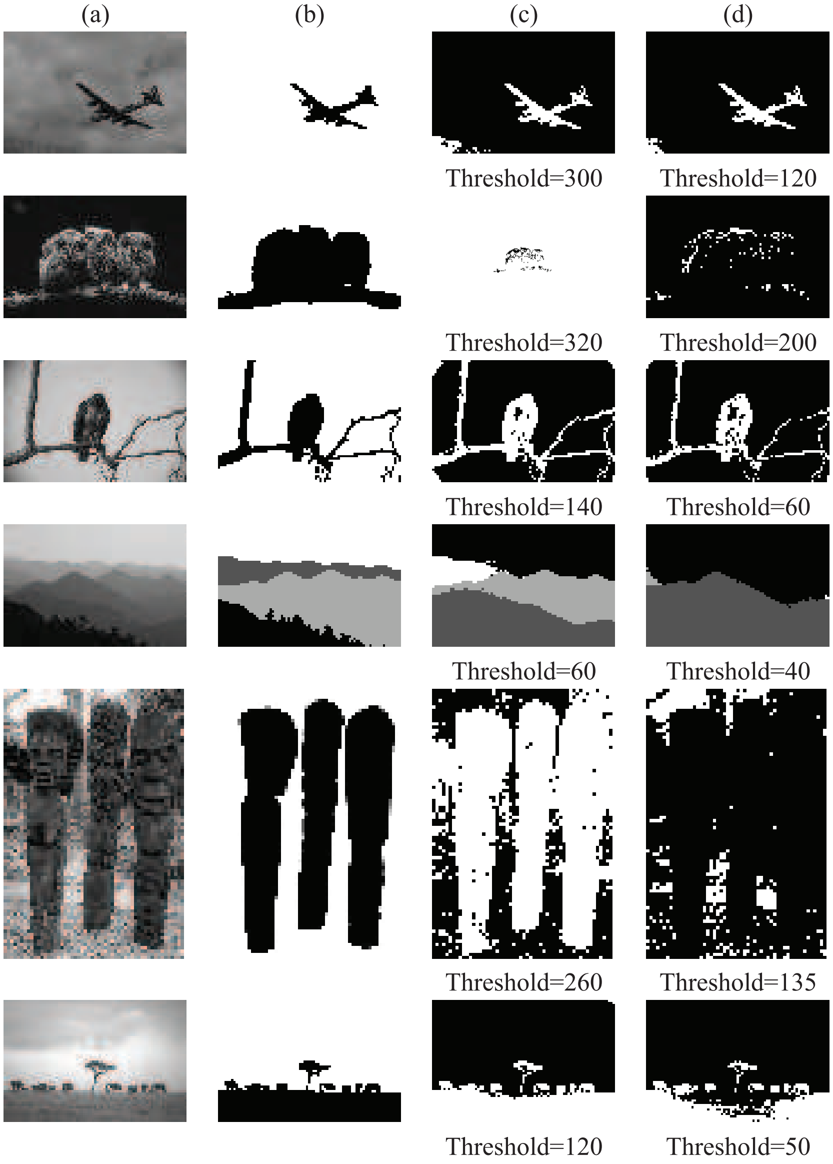


Fig. 5. Image results

B. *Images*

In the second experiment, we work with six images in order to segment them. These images has been got from [11]. For every pixel, we work with three attributes: its gray intensity level, its coordinate x and its coordinate y. We compare the obtained solution with the ideal segmentation, which is calculated manually. As in the previous experiment, the thresholds needed for the algorithms execution are calculated for each image. They are chosen in the way that the obtained result has the same number of clusters that the ideal segmented one. In this sense, all the images must be divided into two clusters, but the image number four, that is divided in four different clusters.

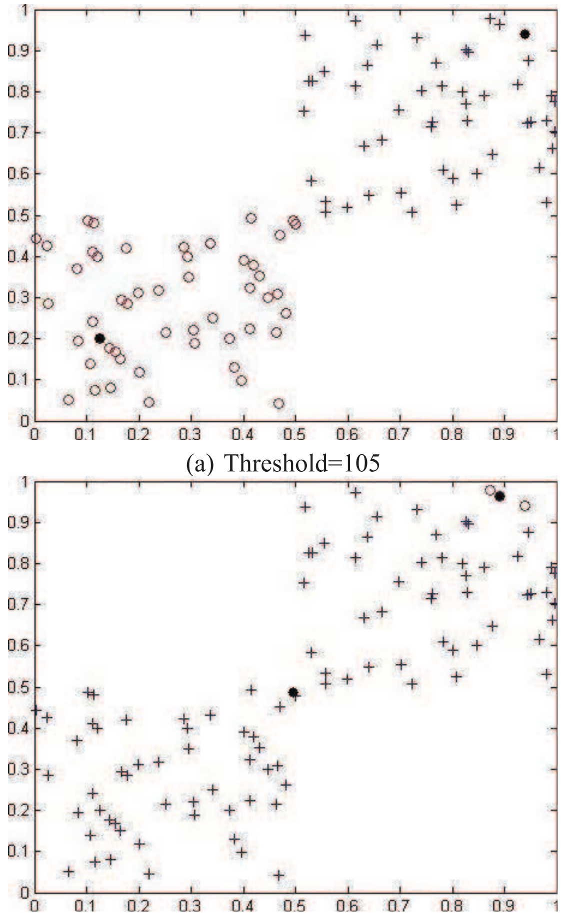
The obtained images are shown in Figure 5, where the column (a) represents the original image, the column (b) represents its ideal segmentation, the column (c) is the result of our algorithm, and the column (d) is the result obtained with the EFC algorithm. Below each image, it is shown the threshold used.

The numeric comparison of the number of pixels well classified by each method is the one shown in Table II.

The algorithm proposed, based on restricted equivalence and ignorance functions, presents an improvement of 10% in average over the EFC.

V. Conclusions and Future Research

In this work we have proposed a modification of the EFC algorithm. We have changed the similarity for restricted equivalence functions, and the entropy functions for ignorance functions. Based on the experimental results, our proposal improves the data classification made by the EFC.



(a) Threshold=105

(b) Threshold=120

Fig. 3. Data 1

	Ignorance	EFC
Image1	97.19%	98.26%
Image2	66%	62.03%
Image3	96.27%	93.96%
Image4	76.63%	48.93%
Image5	86.47%	63.87%
Image6	97.92%	91.40%
Mean	86.75%	76.41%

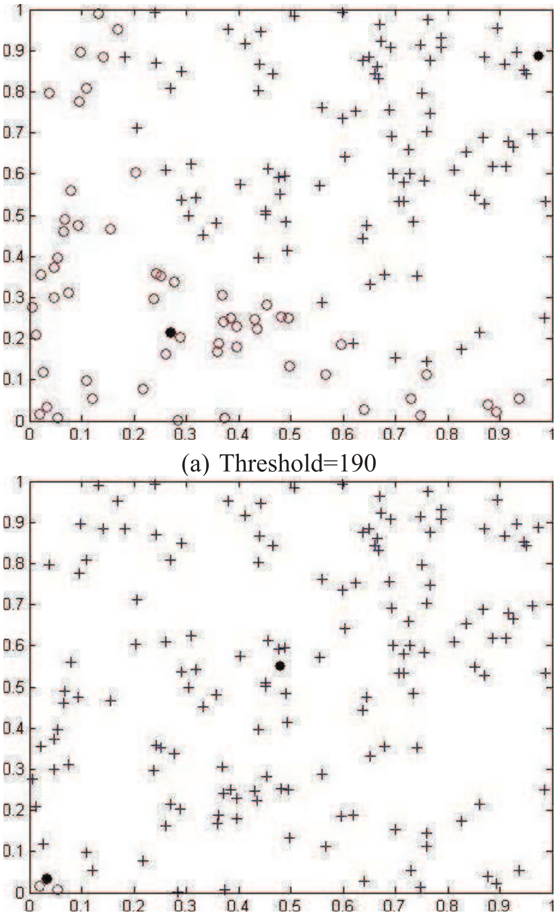
TABLE II
RESULT COMPARISON OF EXPERIMENT 2

As future research lines, the first topic is finding the way of calculate automatically the threshold. Besides, the restricted equivalence and ignorance functions can be applied on different cluster techniques, to improve the obtained results.

REFERENCES

[1] J. Yao, M. Dash, S. T. Tan, and H. Liu, “Entropy-based fuzzy clustering and fuzzy modeling,” *Fuzzy Sets Syst.*, vol. 113, no. 3, pp. 381–388, 2000.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.



(a) Threshold=190

(b) Threshold=140

Fig. 4. Data 2

[3] J. C. Bezdek, M. R. Pal, J. Keller, and R. Krisnapuram, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Norwell, MA, USA: Kluwer Academic Publishers, 1999.

[4] R. Yager and D. Filev, “Generation of fuzzy rules by mountain clustering,” *J. Intell. Fuzzy Syst.*, vol. 2, pp. 209–219, 1994.

[5] S. Chiu, “Fuzzy model identification based on cluster estimation,” *J. Intell. Fuzzy Syst.*, vol. 2, pp. 267–278, 1994.

[6] L.-Y. Wei and C.-H. Cheng, “An entropy clustering analysis based on genetic algorithm,” *J. Intell. Fuzzy Syst.*, vol. 19, no. 4,5, pp. 235–241, 2008.

[7] T. Temel and N. Aydin, “A threshold free clustering algorithm for robust unsupervised classification,” in *BLISS ’07: Proceedings of the 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 119–122.

[8] H. Bustince, E. Barrenechea, and M. Pagola, “Restricted equivalence functions,” *Fuzzy Sets and Systems*, vol. 157, no. 17, pp. 2333 – 2346, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/B6V05-4JRV76F-1/2/b1404a37bf4e0a74786f864b0d1e93ba

[9] T. Calvo, A. Kolesárová, M. Komorníková, and R. Mesiar, “Aggregation operators: properties, classes and construction methods,” pp. 3–104, 2002.

[10] H. Bustince, M. Pagola, E. Barrenechea, J. Fernandez, P. Melo-Pinto, P. Couto, H. Tizhoosh, and J. Montero, “Ignorance functions. an application to the calculation of the threshold in prostate ultrasound images,” *Fuzzy Sets and Systems*, vol. In Press, Corrected Proof, pp. –, 2009. [Online]. Avail-

able: http://www.sciencedirect.com/science/article/B6V05-4W0R0H6-1/2/42b4c060c865ba693e08574ea2582c05

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.