

# FACT: A new Fuzzy Adaptive Clustering Technique

Faezeh Ensan, Mohammad Hossien Yaghmaee, Ebrahim Bagheri  
Department of Computing, Faculty of engineering  
Ferdowsi University of Mashhad, Mashhad, Iran  
Fa\_En93@stu-mail., hyaghmae@, Eb\_ba63@stu-mail.{um.ac.ir}

## Abstract

Clustering belongs to the set of mathematical problems which aim at classification of data or objects into related sets or classes. Many different pattern clustering approaches based on the pattern membership model could be used to classify objects within various classes. Different models of Crisp, Hierarchical, Overlapping and Fuzzy clustering algorithms have been developed which serve different purposes. The main deficiency that most of the algorithms face is that the number of clusters for reaching the optimal arrangement is not automatically calculated and needs user intervention. In this paper we propose a fuzzy clustering technique (FACT) which determines the number of appropriate clusters based on the pattern essence. Different experiments for algorithm evaluation were performed which show a much better performance compared with the typical widely used K-means clustering algorithm

## 1. Introduction

Clustering belongs to the set of mathematical problems which aim at classification and assignment of data or objects to related sets or classes. The act of classification could well be applied through supervised or unsupervised learning methods [1]. In the Supervised model, Patterns are learnt using some familiar, previously classified data. Multi-layered Perceptron - MLP, Support Vector Machine - SVM, and Decision Trees are illustrious examples of such learning algorithms. This type of learning may be called the learning by example methodology. On the other hand, and in the unsupervised method, mainly named clustering, entities are classified in homogeneous classes so that neighboring patterns are assembled in similar collections. In this approach object-class association is not previously known and clusters are formed based on some object similarity criteria.

As unsupervised learning models, such as clustering, can semi-consciously detect well separated classes amongst available data based on their intrinsic features, they have been extensively used in different scientific fields. Their use can significantly vary from Medicine and its application to disease detection, to Intrusion Detection Systems (IDS) for network activity division into two typical types of intrusive and non-intrusive. New applications of clustering have been found in data (web) mining and adaptive systems where user characteristics modeling, session detection and etc can be achieved through modified clustering algorithms. Pattern recognition can also be an important field of clustering application.

The main point in clustering is to classify a set of patterns like  $P = \{p_1, p_2, \dots, p_N\}$  where  $p_i$  can be defined with  $r$  attributes in  $R^r$  as  $p_i = (p_{i1}, \dots, p_{ir})$  into  $n$  clusters ( $C_j$ ;  $j = 1 \dots n$ ) to minimize the sum of squared distance of each pattern  $p_l$  from the parent cluster's center (e.g. the center of  $m^{\text{th}}$  cluster ( $cc_m$ )). The optimization problem can be formulated as (Eq.1) [2].

$$\text{Min} \sum_{i=1}^n \sum_{p_l \in C_i} \|p_l - cc_i\|^2 ; cc_i = \frac{1}{|C_i|} \sum_{l: p_l \in C_i} p_l \quad (1)$$

Many heuristics have been devised to provide a suitable solution for the aforementioned optimization problem. Between the many proposed algorithms, K-means is the best known and most widely used [3]. Although K-means faces some deficiencies such as dependence on the initial state which is because it aims at minimizing a mean squared error function (Eq. 2) (objective function) for algorithm improvement by which it can converge to local solutions, but it can finely scale up for clustering high volumes of patterns.

$$J = \sum_{j=1}^n \sum_{i=1}^N \|p_i^{(j)} - cc_j\|^2 \quad (2)$$

The other notable shortcoming of the K-means heuristic is that it can not automatically determine the number of existing pattern classes in the provided pattern set [4]. In this clustering technique, the center of each cluster is the mean vector of the cluster members. The K-means algorithm can be performed in four steps. In the first step, k cluster centers are chosen randomly in the solution space. The k selected points comprise the centroids of the clusters. In the next step each pattern is assigned to a cluster based on its Euclidean distance to the cluster center. The positions of the K centroids are then recalculated and new clusters are formed. The procedures of the second and third steps are iteratively repeated until the cluster centroids no longer move significantly or in other words the objective function has reached local minima.

Most clustering techniques assume a well defined distinction between the clusters so that each pattern can only belong to one cluster at a time. This supposition can neglect the natural ability of objects existing in multiple clusters. For this reason and with the aid of fuzzy logic, fuzzy clustering can be employed to overcome this weakness. The membership of a pattern in a given cluster can vary between 0 and 1. In this model one single pattern can have different degrees of membership in various clusters. A pattern belongs to the cluster where it has the highest membership value.

In this paper we aim to propose a fuzzy clustering technique which is capable of detecting the most appropriate number of clusters based on a density factor. This algorithm is completely insensitive to the initial number of employed clusters; however the initial value should always be lower than the optimal cluster number. Although a very low number of initial clusters will increase the computation time and CPU usage but it will prevent the algorithm from choosing the incorrect number of clusters. The method discovers the number of clusters by intelligently splitting capable clusters and creating new cluster centers through outlier detection.

The paper is organized in the following sections: In section 2 the fuzzy clustering algorithms are minutely studied. Section 3 will introduce the proposed fuzzy clustering heuristic. Sections 4 presents experimental results obtained from the algorithm implementation and the following section will conclude and provide related works in the field.

## 2. Fuzzy Clustering

Many different pattern clustering approaches based on the pattern membership model could be used to classify objects within various classes. In the first

model a pattern can only be contained in one cluster. This model is usually called the crisp clustering model. Hierarchical clusters on the other hand are devised so that one cluster can consist of one or more clusters [5]. In this model sibling clusters can not overlap. Overlapping clustering allows two or more clusters to have some degree of intersection [6]. In contrast, the fuzzy clustering algorithms allow each pattern to have some degree of membership to any or each of the available clusters. The membership degree could be seen as the probability of a pattern belonging to a specific cluster. Let n and c be the pattern set size and number of clusters, respectively. Matrix  $U = [u_{ij}]_{c \times n}$ , denotes the membership value of pattern i with respect to cluster j, where  $0 \leq u_{ij} \leq 1$  and  $\sum_{i=1}^c u_{ij} = 1$ .

Among the different fuzzy clustering models that have been proposed to satisfy the given definition, Fuzzy C-means devised by Bezdek employs an iterative approach to optimize the objective function introduced in (Eq. 3) [7].

$$j(U, C_1, C_2, \dots, C_n) = \sum_{i=1}^c \sum_{j=1}^N u_{ij}^m d_{ij}^2 \quad (3)$$

A regulating factor  $m \geq 1$  is used in the objective function to set the fuzziness of the algorithm. Setting  $m = 1$  will make the clustering method crisp. The usual value for m is 2.  $C_i$  is specified as the centroids of the  $i^{\text{th}}$  cluster in (Eq. 3) and  $d_{ij}$  is the Euclidian distance between the  $i^{\text{th}}$  centroid ( $c_i$ ) and the  $j^{\text{th}}$  data point.

The membership degree of each pattern is initially randomly selected so that  $\sum_{i=1}^c u_{ij} = 1$ . In the next phases of the C-means clustering algorithm the membership values and the centroid of all clusters are recalculate using (Eq. 4) and (Eq. 5).

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \quad (4)$$

$$C_i = \frac{\sum_{j=1}^n u_{ij}^m p_j}{\sum_{j=1}^n u_{ij}^m} \quad (5)$$

The fuzzy C-means algorithm can be undertaken in 3 steps. The steps are briefly summarized:

**Step 1 – Parameter Setting**

The initial values of the membership value matrix are randomly selected according to the previously stated criterion. Values for tuning parameters  $\epsilon$  and  $m$  are also assigned in this step.

**Step 2 – Formula Recalculation**

Having the cluster centroids from the previous iteration, fresh centroid values are calculated based on (Eq. 5). As the cluster centroids change, the membership values of all patterns regarding each cluster will change accordingly. The adjustment is done by the application of (Eq. 4).

**Step 3 – Dissimilarity Computation**

The dissimilarity between the cluster centroids and the patterns are computed using (Eq.3).

As the fuzzy C-means algorithm follows an iterative approach, steps 2 and 3 are continuously revisited until the differentiation reaches below the predefined threshold ( $\epsilon$ ). It is important to note that in the fuzzy C-means algorithm, different initializations cause different evolutions of the algorithm. It can actually converge to the same result but probably with a different number of iteration steps.

**3. FACT Heuristic**

Although Fuzzy C-means algorithm shows strengths in many areas but it lacks the ability to determine the appropriate number of clusters for pattern classification and requires the user to define the correct number of clusters. Many applications of clustering like pattern recognition or intrusive data classification require the clustering algorithm to decide on the proper number of clusters, as the correct number of classes is not a priori known. In the proposed heuristic we devise an algorithm which exploits a modified version of Fuzzy C-means in which U (Membership Degree Matrix) is not randomly initialized. The other two strengths of this heuristic is that it is based on a fuzzy split-outlier detector and a Cluster Density Criterion (CDC). The fuzzy split algorithm was to some extent inspired by [8]. Some fuzzy clustering algorithms such as [9] are based on the minimization of the objective function value as their ultimate goal. This criterion serves as a great factor for the algorithms with a predefined number of clusters; however in heuristics which have an adaptive approach to cluster number assessment, this factor cannot be used. This is because the objective function will decrease with the increase of the number of clusters and hence causes further cluster splitting which results in an incorrect number of clusters (the number of

clusters will most likely end up being identical to the number of available patterns). For this reason using the objective functions as the basis for successful split assessment is unreasonable. We define and apply CDC for split success comparison. FACT is comprised of 3 main steps which are further explained in the following paragraphs:

**Step 1 – Initialization**

The existing version of the Fuzzy C-means is applied to the set of available patterns by setting the initial cluster number and  $m$  to 2. The outputs of this step are the preliminary values for U and CDC.

**Step 2 – Outlier Detection**

a) *Cluster Member Assignment*

Every pattern in the fuzzy clustering algorithm has a membership degree in all available clusters. The process of pattern to cluster assignment is done through allocating the pattern to the cluster in which it has the highest membership degree. Matrix M,  $[m_{ij}]_{c \times n}$  is defined as follows:

$$M_{ij} = \begin{cases} U_{ij}, & \text{if } \text{Max}_{i=1}^c U_{ij} = U_{ij} \\ 0, & \text{else} \end{cases} \quad (6)$$

b) *Local Outlier Detection*

In this sub step the candidates in each cluster to be the final outliers over all of the patterns are selected. This process selects the pattern with the lowest non-zero membership value in vector  $M_i$  where  $i$  shows the current cluster (7).

$$\text{Candidate}_i = \text{Min}_{j=1}^n (M_{ij}) \quad \text{where } M_{ij} \neq 0 \quad (7)$$

c) *Final Outlier Selection and Splitting*

The pattern with the lowest value in the Candidate vector (OP) is selected as the ultimate outlier. The coordinates of OP are used as the basis for the center of a new cluster. Let  $OP = \{op_1, op_2 \dots op_r\}$  be the outlier point, the new cluster center will be calculated using (Eq.8):

$$\text{Center } (c+1) = OP + \lambda \quad (8)$$

Where  $\lambda = (\lambda_1, \lambda_2 \dots \lambda_r) \sim 0$ .

Having calculated the value of the new cluster center, the previous composition of pattern classifications can be altered and rearranged based on  $c+1$  cluster. Matrix U is updated using (Eq. 4) where the upper bound of  $k$  is  $c+1$ . The modified version of

fuzzy C-means is now tuned using the calculated U and c+1 number of clusters and is used to create the new cluster composition. After having split the cluster formation into a new arrangement, the CDC will be updated (Eq. 9). The value obtained from the division of the new CDC to the former CDC is multiplied by a coefficient,  $\alpha$ , which is between 0 and 1. To show that splitting has improved the clustering,  $\theta_{t+1}$  should be larger than  $\theta_t$  and thus the splitting procedure is confirmed and stabilized. The value for  $\alpha$  is usually set to 0.2. The  $\theta$  is named the Feedback Control Parameter (FCP) which controls the system behavior using a feedback from the prior iteration.

$$CDC_{t+1} = \sum_{i=1}^{c+1} \sum_{j=1}^n M_{ij} \quad (9)$$

$$\theta_{t+1} = \alpha \left( \frac{CDC_{t+1}}{CDC_t} \right) + (1 - \alpha) \theta_t \quad (10)$$

If the splitting has been unsuccessful 2.c is repeated with the next pattern in the Candidate vector.

### Step 3 – Test

If none of the patterns available in the Candidate vector can serve as a successful splitting point for improving the current cluster arrangement, the algorithm will terminate with the current composition on hand else it will increase the number of clusters by one unit and resume algorithm execution from 2.a.

## 4. Experimental Results

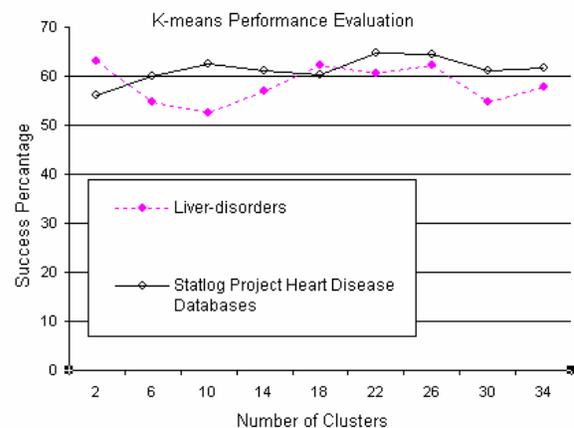
Four main pattern sets were used in the first set of experiments. Wisconsin Breast Cancer Databases containing 699 patterns were cleaned to be used in the comparison procedure. The patterns were 9 dimensional data with 2 main classes (malignant and benign). Pima Indians Diabetes Database was the second pattern set used which included 768 patterns with 8 attributes for each pattern. The training was done to test positive or negative diabetes tests. The patterns were initially obtained from the National Institute of Diabetes and Digestive and Kidney Diseases. The third pattern set was the Liver-disorders Database from the BUPA Medical Research Ltd. This pattern set consisted of 345 patterns each with 7 numeric-valued attributes. The Statlog Project Heart Disease Database is made up of 270 patterns which are used to classify normal and abnormal patients using 13 different traits. The four pattern sets were taken from the UCI Machine Learning Repository at [10]. The

pattern sets were divided into two parts for train and test purposes. The exact division is shown in table 1.

**Table 1. The number of train and test patterns used in each pattern set. 1 shows “Pattern numbers”.**

Pattern Set	PN <sup>1</sup>	Train Patterns			Test patterns		
		Sum	class 1	class 2	Sum	class 1	class 2
Wisconsin Breast Cancer	683	500	303	197	183	141	42
Pima Indians Diabetes	768	500	318	182	268	182	86
Liver-disorders	345	250	110	140	95	35	60
Statlog project Heart Disease Database	270	220	124	96	50	26	24

For the sake of clarity and to show K-means’ dependency to the number of clusters, different variations of cluster numbers were created for k-means performance evaluation ranging from 2 to 34 clusters. Figure 1 depicts K-means performance under different circumstances. As it can be inferred from the diagram, different number of clusters employed, can significantly affect the final outcome and hence be evaluated as a negative impact on the overall algorithm performance.



**Figure 1. K-means performance evaluation**

The proposed heuristic was examined under several different criteria and compared with the K-means algorithm. As the K-means algorithm clusters data using a predefined number of classes and this is different from what the proposed heuristic does, the appropriate comparison model should have been

devised. In our analysis, FACT was first applied to the pattern sets for clustering. This step detected the available classes ( $c$ ) in the pattern sets. K-means was then executed with three different initial cluster numbers of  $c-1$ ,  $c$  and  $c+1$ . Although the evaluations were initially done based upon the number of FACT detected classes but the optimal number of classes known from the omniscient were also applied to the K-means algorithm to compare the optimal success ratio of both heuristics. As K-means provides different clustering compositions each time it is run due to its sensitivity to the initial state; it was executed 100 runs on every pattern set with the specified cluster number and the average results were used. Table 2 compares the performance of both heuristics based on the percentage of correct pattern classification. As it can be clearly seen, the FACT algorithm outperforms the K-means algorithm in 3 of the pattern bases and reaches optimality in the Liver-disorders pattern set. The important point is that due to the differences in the essence of the algorithms the number of appropriate clusters for the FACT might differ from the optimal number of clusters for K-means, but even with the selection of the best number of clusters in K-means, FACT still shows better performance.

The next experiment was done on pattern sets which were statistically created to form well-defined class boundaries. Each pattern set had colonies of patterns consisting of 200 objects. 4, 5 and 7 colonies had been integrated into the pattern sets forming pattern sets with 800 (PS1), 1000 (PS2) and 1400 (PS3) patterns. To compare the performance of each clustering model, the Quadratic Error (QE) factor introduced in [11] was used. Let  $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$  be the members of cluster  $i$  and  $cc_i$  be the centroid of the  $i^{\text{th}}$  cluster, QE is defined as the average of the mean squared distances of each pattern to the cluster centroids as shown in (Eq.11).

$$QE = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{p_j \in C_i} \|p_j - cc_i\|^2}{|C_i|} \quad (11)$$

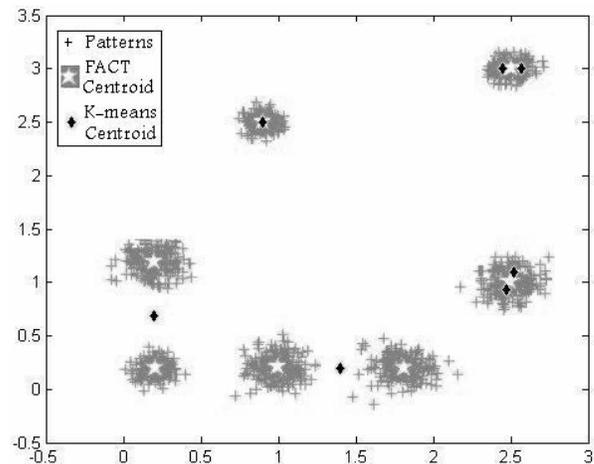
Table3 compares the values for the QE factor obtained from different heuristics. The smaller the value for the Quadratic Error is, the higher the inter object relationship in one cluster would be. FACT related QE values show much better performance for the proposed algorithm. Figure 2 shows one of the devised pattern sets depicting the cluster centers chosen using each algorithm which reveals a better centroid placement strategy in the FACT algorithm.

**Table 2. The comparison of the two heuristics based on the correct classifications rate.1 and 2 show the Number of detected Clusters (NC) using the FACT algorithm and the Success Percentage achieved (SP), respectively. The number of clusters used to reach the best success percentage and the success percentage achieved in K-means are also shown in 3, ONC, and 4, OSP.**

Pattern Set	FACT		K-means			
	$NC^1$	$SP^2$	$NC$	$SP$	$ONC^3$	$OSP^4$
Wisconsin Breast Cancer	8	100.00	7	98.57	2	100.0
			8	98.36		
			9	98.91		
Pima Indians Diabetes	68	71.31	67	67.67	14	69.96
			68	65.10		
			69	66.10		
Liver-disorders	2	63.16	2	63.16	2	63.16
			3	63.16		
			4	53.68		
Statlog Project Databases	9	70.00	8	59.50	18	64.66
			9	60.66		
			10	60.66		

**Table 3. QE-based comparison**

Pattern Set	Number of Clusters	Quadratic Error of FACT	Quadratic Error of K-means
PS1	4	0.005934	0.006044
PS2	5	0.0064	0.0306
PS3	7	0.0075	0.0321



**Figure 2. PS3 and centroid placement strategy**

## 5. Conclusion

In this paper we have proposed a fuzzy adaptive clustering algorithm which modifies the well known

fuzzy C-means. The Fuzzy C-means algorithm is altered so that it is initialized based on the Membership Degree Matrix from the previous iteration. The number of pattern classes used in the clustering process is adaptively calculated. Comparisons done between the typical K-means algorithm and the proposed heuristic demonstrate a better performance concerning the correct clustering percentage and the Quadratic Error factor. It is notable that the outstanding feature of the FACT algorithm is that it detects the correct number of pattern classes adaptively.

## 6. References

- [1] Everitt, B.S., Landau, S., Leese, M., Cluster Analysis, London: New York, Halsted Press, 1993.
- [2] Hansen, P., Mladenovic, N., J-Means: a new local search heuristic for minimum sum-of-squares clustering, Pattern Recognition 34 (2), 2001, pp.405–413.
- [3] MacQueen, J., Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol.2, 1967, pp.281-297.
- [4] Sanjiv, K.B., Adaptive K-Means Clustering, [FLAIRS Conference 2004](#)
- [5] A survey of recent advances in hierarchical clustering algorithms The Computer Journal, Volume 26, Issue 4, 1983, pp. 354-359.
- [6] Barthélemy, J.P., Brucker, F., NP-hard approximation problems in overlapping clustering, Journal of Classification 18, 2001, pp.159-183.
- [7] Bezdek, J.C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [8] Guan, Y., Ghorbani, A., and Belacel, N., K-means+: An autonomous clustering algorithm, in submission.
- [9] Belacel, N., Hansen, P., and Mladenovic, N., Fuzzy J-means: A new heuristic for fuzzy clustering, Pattern Recognition 35, 2002, pp.2193–2200.
- [10] <http://www.ics.uci.edu/~mllearn/MLSummary.html>
- [11] [Bacao, F., Lobo, V., Painho, M., Self-Organizing Maps as efficient initialization procedures and substitutes for k-means clustering](#), International Conference on Computational Science, 2005