Computing and Informatics, Vol. 30, 2011, 701–720

A COMPARATIVE STUDY OF FUZZY C-MEANS ALGORITHM AND ENTROPY-BASED FUZZY CLUSTERING ALGORITHMS

Subhagata Chattopadhyay

School of Computer Studies
Department of Computer Science and Engineering
National Institute of Science and Technology
Berhampur 761008
Orissa, India
e-mail: subhagatachatterjee@yahoo.com

Dilip Kumar Pratihar

Department of Mechanical Engineering
Indian Institute of Technology
Kharagpur 721302
West Bengal, India
e-mail: dkpra@mech.iitkgp.ernet.in

Sanjib Chandra De Sarkar

School of Electrical Sciences
Indian Institute of Technology Bhubaneswar
Bhubaneswar 751013
Orissa, India
e-mail: scdesarkar@yahoo.co.in

Communicated by János Fodor

Abstract. Fuzzy clustering is useful to mine complex and multi-dimensional data sets, where the members have partial or fuzzy relations. Among the various deve-

loped techniques, fuzzy-C-means (FCM) algorithm is the most popular one, where a piece of data has partial membership with each of the pre-defined cluster centers. Moreover, in FCM, the cluster centers are virtual, that is, they are chosen at random and thus might be out of the data set. The cluster centers and membership values of the data points with them are updated through some iterations. On the other hand, entropy-based fuzzy clustering (EFC) algorithm works based on a similarity-threshold value. Contrary to FCM, in EFC, the cluster centers are real, that is, they are chosen from the data points. In the present paper, the performances of these algorithms have been compared on four data sets, such as IRIS, WINES, OLITOS and psychosis (collected with the help of forty doctors), in terms of the quality of the clusters (that is, discrepancy factor, compactness, distinctness) obtained and their computational time. Moreover, the best set of clusters has been mapped into 2-D for visualization using a self-organizing map (SOM).

Keywords: Fuzzy clustering, fuzzy c-means algorithm, entropy-based algorithms, self-organizing maps

1 INTRODUCTION

A cluster is usually represented as either grouping of similar data points around a center (called centroid) or a prototype data instance nearest to the centroid. In other way, a cluster can be represented either with or without a well-defined boundary. Clusters with well-defined boundaries are called crisp clusters, while those without such feature are called fuzzy clusters. The present paper deals with fuzzy clustering only. Clustering is an unsupervised learning of unlabeled data, and such property has separated it from classification, where the class-prediction is done on unlabeled data after a supervised learning on pre-labeled data. As the training is unsupervised in clustering algorithms, these can be safely used on a data set without much knowledge of it. Two most important benefits of clustering are as follows:

1. easy tackling of noisy data and outliers,
2. ability to deal with the data having various types of variables, such as continuous variable that requires standardized data, binary variable, nominal variable (a more generalized representation of binary variable), ordinal variable (where order of data is the most important criterion) and mixed variables, (that is, amalgamation of all above) [19].

Several fuzzy clustering algorithms had been proposed by various researchers. Those algorithms include fuzzy ISODATA, fuzzy C-means, fuzzy K-nearest neighborhood algorithm, potential-based clustering, and others [21]. Recently, some more fuzzy clustering algorithms have been proposed. For example, Fu and Medico [12] developed a clustering algorithm to capture dataset-specific structures at the beginning of DNA microarray analysis process, which is known as *Fuzzy clustering by*

Local Approximation of Membership (FLAME). It worked by defining the neighborhood of each object and identifying cluster supporting objects. Fuzzy membership vector for each object was assigned by approximating the memberships of its neighboring objects through an iterative converging process. Its performance was found to be better than that of fuzzy C-means, fuzzy K-means algorithms and fuzzy self-organizing maps (SOM). Ma and Chan [16] proposed an *Incremental Fuzzy Mining (IFM)* technique to tackle complexities due to higher dimensional noisy data, as encountered in genetic engineering. The basic philosophy of that technique was to mine gene functions by transforming quantitative gene expression values into linguistic terms, and using fuzzy measure to explore any interesting patterns existing between the linguistic gene expression levels. Those patterns could make accurate gene function predictions, such that each gene could be allowed to belong to more than one functional class with different degrees of membership.

Fuzzy C-means (FCM) algorithm, one of the most popular fuzzy clustering techniques, was originally proposed by Dunn [8] and had been modified by Bezdek [4]. FCM is able to determine, and in turn, iteratively update the membership values of a data point with the pre-defined number of clusters. Thus, a data point can be the member of all clusters with the corresponding membership values. The philosophy of FCM has been extensively used in different fields of research [20, 1, 24, 18]. A large number of variants of FCM algorithm had been proposed. Some of these recent algorithms are discussed here. Sikka et al. [22] developed a modified FCM known as MFCM to estimate the tissue and tumor areas in a brain MRI scan. Krinidis and Chatzis [14] proposed a *Fuzzy Local Information C-Means (FLICM)* algorithm, which could remove the inherent hindrances of FCM algorithm. The main features of that algorithm were the (i) use of a fuzzy local similarity measure, (ii) shielding of the algorithm from noise-realted hypersensitivities. Moreover, its performance was not dependent on the empirically adjusted parameters of the conventional FCM algorithm. A novel modified FCM algorithm was developed by Belhassen and Zaidi [3] to overcome the problems faced by conventional FCM algorithm with noisy and low resolution oncological PET data. The former was found to be less error-prone compared to the latter.

On the other hand, entropy-based fuzzy clustering (EFC) is also a very popular technique, in which clusters are formed by means of a similarity-threshold value [23]. By changing this value, the number and quality of clusters can be manipulated. As an EFC is a newer clustering method, its number of reported applications is comparatively less [17, 15, 7]. An attempt was made earlier by the authors [5], to test the performances of EFC algorithm and their different extensions, while carrying out clustering on three standard data sets, namely IRIS [10], WINES [11] and OLITOS [2]. Both IRIS as well as WINES data sets are ideally clustered into three groups. Iris data set contains 150 data items, whose first 50 elements are known as *iris setosa*, the next 50 elements are called *iris versicolor* and the last 50 data belong to *iris virginica*. On the other hand, in WINES data set, there are 178 elements, which are ideally clustered into three groups – the first group contains 59 elements, the second group consists of 71 elements and there are 48 elements in

the third group. OLITOS data set consists of 120 data items, which are distributed among four groups containing 50, 25, 34 and 11 elements.

The present paper is a novel attempt to compare the performances of FCM algorithm and EFC algorithm along with its proposed extensions on four data sets, such as IRIS, WINES, OLITOS and psychosis (collected with the help of forty doctors) in terms of quality of the clusters made and their computational time. It is to be mentioned that the quality of the clusters has been decided based on discrepancy factor, compactness and distinctness.

The rest of the paper is organized as follows: Section 2 explains the different tools and techniques used in the present work. The results of these techniques are stated and discussed in Section 3. Some concluding remarks are made in Section 4 and the scope for future work is indicated in Section 5.

2 TOOLS AND TECHNIQUES USED

The working principles of different tools and techniques used in the present work, such as fuzzy C-means algorithm, entropy-based fuzzy clustering (EFC) algorithms and self-organizing map (SOM), have been explained briefly in this section.

2.1 Fuzzy C-Means (FCM) Algorithm

FCM is one of the most popular fuzzy clustering techniques, which was proposed by Dunn [8] in 1973 and eventually modified by Bezdek [4] in 1981. It is an approach, where the data points have their membership values with the cluster centers, which will be updated iteratively. The FCM algorithm consists of the following steps:

- Step 1: Let us suppose that M-dimensional N data points represented by x_i $(i = 1, 2, \ldots, N)$, are to be clustered.
- Step 2: Assume the number of clusters to be made, that is, C, where $2 \leq C \leq N$.
- Step 3: Choose an appropriate level of cluster fuzziness $f > 1$.
- Step 4: Initialize the $N \times C \times M$ sized membership matrix U, at random, such that $U_{ijm} \in [0, 1]$ and $\sum_{j=1}^{C} U_{ijm} = 1.0$, for each i and a fixed value of m.
- Step 5: Determine the cluster centers CC_{jm}, for j^{th} cluster and its m^{th} dimension by using the expression given below:

$$CC_{jm} = \frac{\sum_{i=1}^{N} U_{ijm}^{f} x_{im}}{\sum_{i=1}^{N} U_{ijm}^{f}}. \tag{1}$$

- Step 6: Calculate the Euclidean distance between i^{th} data point and j^{th} cluster center with respect to, say m^{th} dimension like the following:

$$D_{ijm} = \|(x_{im} - CC_{jm})\|. \tag{2}$$

- Step 7: Update fuzzy membership matrix U according to D_{ijm}. If $D_{ijm} > 0$, then
$$U_{ijm} = \frac{1}{\sum_{c=1}^{C} (\frac{D_{ijm}}{D_{icm}})^{\frac{2}{f-1}}}. \tag{3}$$
If $D_{ijm}= 0$, then the data point coincides with the corresponding data point of j^{th} cluster center CC_{jm} and it has the full membership value, that is, $U_{ijm} = 1.0$.
- Step 8: Repeat from Step 5 to Step 7 until the changes in $U \leq \epsilon$, where ϵ is a pre-specified termination criterion.

2.1.1 Entropy-Based Fuzzy Clustering (EFC) Algorithm

In this algorithm, entropy values of the data points are calculated first and then the data point having the minimum entropy value is selected as the cluster center [23]. It is also an iterative approach, where the data points are clustered based on a threshold value of similarity. The data points, which are not being selected inside any of the clusters are termed as outliers. The principle of EFC algorithm is explained below. Let us assume that there are N data points in M-dimensional [T] hyperspace, where each data point X_i $(i = 1, 2, 3, \ldots, N)$ is represented by a set of M values (i.e., $X_{i1}, X_{i2}, X_{i3}, \ldots, X_{iM}$). Thus, the data set can be represented by an $N \times M$ matrix. The Euclidean distance between any two data points (e.g., i and j) is determined as follows:
$$D_{ij} = \sqrt{\sum_{k=1}^{M} (X_{ik} - X_{jk})^2}. \tag{4}$$

There is a maximum of N^2 distance values among N data points and out of which ${}^{N}C_2$ distances belong to D_{ij} (where $i < j$) and D_{ji} each. Moreover, there are N diagonal values, each of which is equal to zero (where $i = j$). Now, similarity between any two points (i.e., i and j) can be calculated as follows:
$$S_{ij} = e^{-\alpha\ D_{ij}}, \tag{5}$$

where α is a numerical constant. It is to be noted that the similarity value between any two points lies in the range of 0.0 to 1.0. The value of α is calculated based on the assumption that the similarity value S_{ij} is set equal to 0.5, when the distance between two data points (i.e., D_{ij}) becomes equal to the mean distance $\overline{D}$, which is represented as follows:
$$\overline{D} = \frac{1}{{}^{N}C_2} \sum_{i=1}^{N} \sum_{j>i}^{N} D_{ij}. \tag{6}$$

From Equation (5), α can be determined as follows:
$$\alpha = -\frac{\ln 0.5}{\overline{D}}. \tag{7}$$

Next, entropy (E_i) of each data point with respect to the other data points is calculated as follows:

$$E_i = -\sum_{j \in X}^{j \neq i} (S_{ij} \log_2 S_{ij}) + (1 - S_{ij}) \log_2 (1 - S_{ij})). \tag{8}$$

During clustering, the data point having the minimum entropy value is selected as the cluster center. The clustering algorithm is explained below.

Clustering Algorithm: It consists of the following steps:

1. Calculate E_i $(i = 1, 2, 3, \ldots, N)$ for each X_i lying in $[T]$ hyperspace.
2. Determine minimum E_i and select $X_{i,Min}$ as the cluster center.
3. Put $X_{i,Min}$ and the data points having similarity with $X_{i,Min}$ greater than β (threshold value for similarity) in a cluster and remove them from $[T]$.
4. Check whether $[T]$ hyperspace is empty. If yes, terminate the program, else go to Step 2.

In this algorithm, E_i is calculated in such a way that a data point, which is far away from the rest of data points, may also be selected as a cluster center. To prevent such a situation, another parameter called γ (in %) has been introduced, which is nothing but a threshold used to declare a cluster to be a valid one. If the number of data points present in a cluster becomes greater than or equal to $\frac{\gamma N}{100}$, we declare it as a valid cluster. Otherwise, these data points will be treated as the outliers. The above EFC algorithm has been extended as follows.

2.1.2 Proposed Extensions of EFC Algorithm

The above EFC algorithm developed by Yao et al. [23], in which entropy values of the data points are calculated only once, is called *Method 1*. Moreover, the constant α (refer to Equation (7)) is determined only once. Thus, in their approach, they considered the fixed values of α (used to calculate similarity) and entropy, and we call it *Approach 1*. To extend their work, two other approaches (namely *Approaches 2* and *3*) have been developed. In *Approach 2*, entropy values are calculated only once but the similarity values are updated iteratively for the remaining data points in $[T]$, after some clusters are formed. In *Approach 3*, both the entropy as well as similarity values are updated iteratively from the leftover points in $[T]$, after a particular cluster is determined. Besides these two approaches, two other methods (for example, *Method 2* and *Method 3*) have been developed by the authors earlier [5], to determine the cluster centers, which are discussed below.

Method 2: Determination of cluster centers based on total similarity of the data points

In this method, a data point having the maximum total similarity with other data points has been selected as a cluster center. Total similarity of a data point is calculated as follows:

$$S_i = \sum_{j \in X}^{j \neq i} S_{ij}. \tag{9}$$

All the three approaches have been developed as discussed above. As a simpler expression (refer to Equation (9)) is used in this method, compared to that in *Method 1* (that is, Equation (8)), it is expected to be faster than *Method 1*.

Method 3: Determination of cluster centers based on dissimilarity-to-similarity ratio (DSR)

In this method, a ratio of dissimilarity to similarity (DSR) is calculated for each data point by considering its similarity with all other points by using the following expression:

$$DSR_i = \sum_{j \in X}^{j \neq i} \frac{(1 - S_{ij})}{S_{ij}}. \tag{10}$$

The point having the minimum DSR is selected as the cluster center. As it is a simple expression, it avoids a lot of computation involved in Equation (8). Thus, this method is also expected to be computationally faster than *Method 1*. All the above three approaches have also been developed in this method.

2.2 Cluster Visualization Tool: Self-Organizing Map

Self-organizing map (SOM) is a popular tool used to map the higher dimensional data into a lower dimension (say, 2-D or 3-D) by keeping their topological information intact, for visualization [13]. It works based on the principle of unsupervised and competitive learning. It consists of two layers, namely *input layer* and *competition layer*. In the *input layer*, there are N multivariate data points, which are to be mapped to 2-D, for the ease of visualization. *Competition layer* carries out three basic operations, such as *competition*, *cooperation* and *updating*. The winner neuron or the best match for an input is determined through competition. The neighborhood surrounding the winner neuron is then identified and they cooperate with each other. The winner neuron along with its neighbors are then updated. Interested readers may refer to [9], for a detailed description of the algorithm.

3 RESULTS AND DISCUSSION

The quality of the obtained clusters has been tested in terms of discrepancy factor (DF), compactness and distinctness. The DF relates the quality of the obtained

clusters with that of the already known and ideal clusters. It is calculated as follows:

$$DF = \frac{1}{2}\left[\sum_{i=1}^{C}(A_i + B_i) + OL\right], \tag{11}$$

where C represents the number of valid clusters, A_i indicates the number of wrong elements included in the i^{th} cluster, B_i denotes the number of right elements missed by the i^{th} cluster and OL represents the number of outliers. Compactness of a cluster is determined by calculating the average Euclidean distance of the members of a cluster with respect to its center. Similarly, distinctness of a set of clusters is decided by calculating the average of the inter-cluster Euclidean distances. Moreover, the above clustering algorithms have been compared in terms of their computational time (seconds). The user time values of the algorithm have been recorded for 50 times (as a slight variation is observed in some of the user time values) on a P-IV PC and the average of the above values (that is, t_{av} in seconds) is calculated.

3.1 Clustering of IRIS data

Clustering is done on IRIS data by using both the FCM as well as EFC algorithms. In FCM algorithm, the value of fuzziness f and termination criterion ϵ are set equal to 2.0 and 0.001, respectively, after a careful study. The set of clusters, thus obtained, are shown in Table 1.

Cluster 1	**Cluster 2**	**Cluster 3**
50	0	0
0	47	3
0	13	37

Table 1. FCM-yielded set of clusters on IRIS data

It is to be noted that there is no outlier and DF is coming out to be equal to 16. Figure 1 a) shows 2-D plot of the above set of clusters as obtained by using the SOM, in which three clusters have been distinctly identified. The average computational time t_{av} of the algorithm is found to be equal to 0.0252 seconds.

Entropy-based fuzzy clustering (EFC) methods and their different approaches are found to be sensitive to the threshold value of similarity, that is, β. As β increases, the number of clusters is seen to increase, and then it reaches the maximum value, corresponding to a value of β. Moreover, the number of clusters is found to decrease with the further increase in value of β. On the other hand, the number of outliers is seen to be either equal to zero or near to zero for small values of β, but it suddenly increases with the increase in β value.

Table 2 shows the best set of clusters obtained by using different method-approach combinations of the EFC algorithm on IRIS data set.

Approach 1 of Method 3 is found to yield the best set of clusters for this data (corresponding to which, the DF is coming out to be the lowest). It is important to

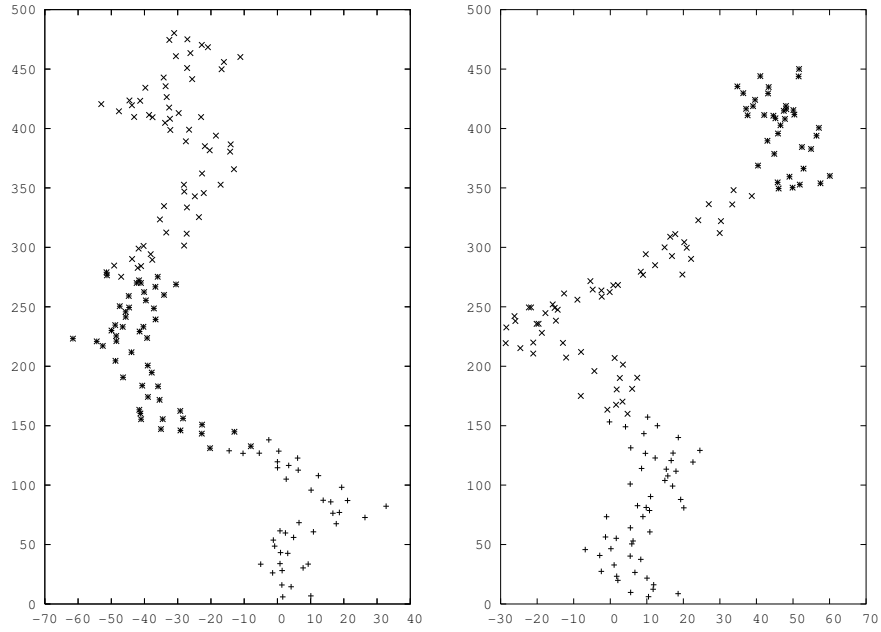


Fig. 1. The best set of clusters on IRIS data obtained by a) FCM algorithm, b) Approach 1 of Method 3 of EFC algorithm

	Approach 1	**Approach 2**	**Appro ach 3**
Method 1	$\beta = 0.695$ 50 0 0 0 42 0 0 18 32 OL = 8, DF = 26	$\beta = 0.5575$ 50 0 0 0 11 32 0 41 3 OL = 13, DF = 27	$\beta = 0.476$ 50 0 0 0 11 32 0 12 34 OL = 4, DF = 23
Method 2	$\beta = 0.675$ 0 0 50 46 0 0 22 28 0 OL = 4, DF = 26	$\beta = 0.635$ 0 50 0 49 0 0 29 0 9 OL = 13, DF = 42	$\beta = 0.67$ 0 0 50 46 0 0 22 19 0 OL = 13, DF = 35
Method 3	$\beta =$ **0.675** 0 0 50 50 0 0 9 32 0 OL = 9, DF = 18	$\beta = 0.68$ 0 0 44 49 0 0 9 34 0 OL = 14, DF = 23	$\beta = 0.5$ 775 0 0 50 49 0 0 22 19 0 OL = 9, DF = 32

Table 2. The best set of clusters obtained by different method-approach combinations of EFC algorithm on IRIS data

note that it has occurred corresponding to a value of β equals to 0.675. For visualization, the best set of clusters (consisting of multi-dimensional data) obtained above have been mapped to 2-D by utilizing a SOM (refer to Figure 1 b)). The clusters are found to be distinct but at the same time compact in nature. Computational time of the best method-approach combination of the EFC algorithm is found to be equal to 0.0242 seconds.

3.2 Clustering of WINES Data

Both the FCM as well as EFC algorithms have been used to do the clustering of WINES data. Their performances have been tested on the said data, as explained below.

The parameters: f and ϵ of the FCM algorithm have been set equal to 2.0 and 0.001, respectively.

Cluster 1	**Cluster 2**	**Cluster 3**
44	0	15
0	52	21
0	20	27

Table 3. FCM-yielded set of clusters on WINES data

Table 3 shows the obtained clusters and their data points as yielded by the above algorithm. To check the quality of the set of clusters obtained above, DF has been calculated and found to be equal to 55.5. The above set of clusters involving multi-dimensional data have been mapped into 2-D by using the SOM, for visualization, which are shown in Figure 2 a).

It is observed that the clusters are not overlapping to each other and the arrangement of the data points inside a cluster is compact in nature. The computational time (that is, t_{av}) of this algorithm is coming out to be equal to 0.0388 seconds.

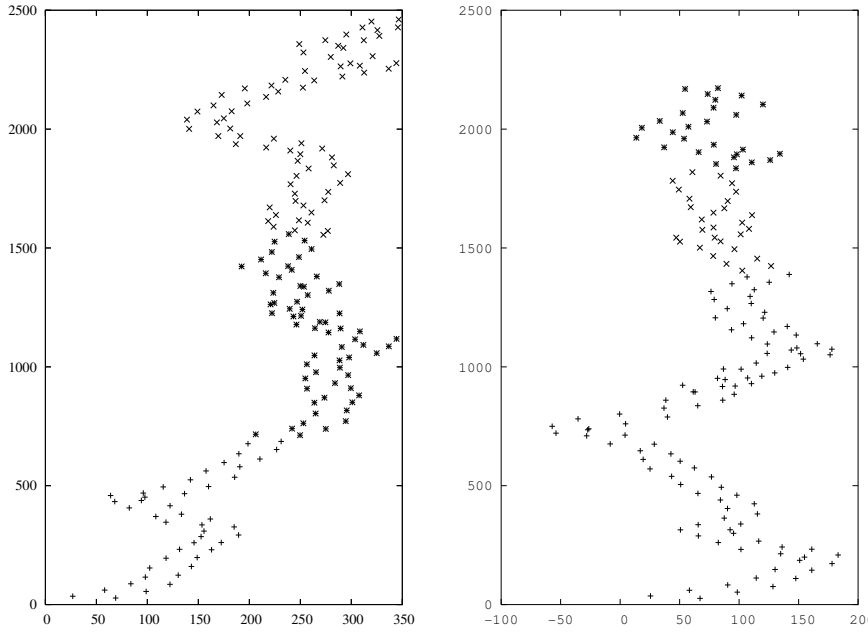
Three methods and their approaches of EFC algorithm have been tried to cluster the WINES data. The obtained clusters are shown in Table 4.

It is to be noted from the above table that Approach 3 of Method 1 has yielded the best set of clusters. Figure 2 b) shows the above best set of clusters, after reducing their dimensions to two. The clusters are found to be compact as well as distinct too. The average user time t_{av} of Approach 3 of Method 1 is found to be equal to 0.0450 seconds.

3.3 Clustering of OLITOS Data

OLITOS data set has been clustered using both the FCM as well as EFC algorithms and the results are explained below.

The clusters obtained by the FCM algorithm after setting the values of f and ϵ to 2.0 and 0.001, respectively, on the OLITOS data, are shown in Table 5. For



a) b)

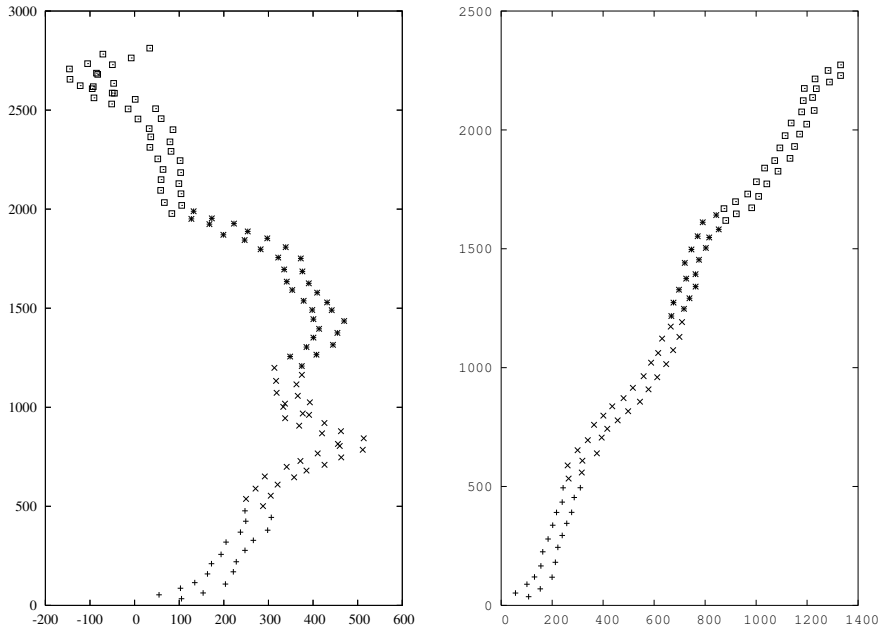
Fig. 2. The best set of clusters on WINES data obtained by a) FCM algorithm, b) Approach 3 of Method 1 of EFC algorithm

	Approach 1	**Approach 2**	**Approach 3**
Method 1	$\beta = 0.5255$	$\beta = 0.5$	$\beta =$ **0.575**
	59 0 0	59 0 0	**57 0 0**
	48 4 15	60 1 1	**34 2 23**
	0 38 1	0 28 19	**0 44 0**
	OL = 13, DF = 66	OL = 10, DF = 90	**OL** = 18**, DF** = 54.5
Method 2	$\beta = 0.565$	$\beta = 0.5675$	$\beta = 0.59$
	51 0 0	50 0 0	45 0 15
	53 9 0	53 0 1	47 4 1
	0 20 26	0 23 25	0 45 0
	OL = 19, DF = 92	OL = 17, DF = 85.5	OL = 21, DF = 72
Method 3	$\beta = 0.5685$	$\beta = 0.569$	$\beta = 0.59$
	50 0 0	50 0 0	44 0 15
	53 9 0	53 9 1	48 4 1
	0 20 25	0 23 25	0 45 0
	OL = 21, DF = 94	OL = 17, DF = 104	OL = 21, DF = 70.5

Table 4. The best set of clusters obtained by different method-approach combinations of EFC algorithm on WINES data

Cluster 1	Cluster 2	Cluster 3	Cluster 4
18	6	15	9
4	8	6	7
6	2	11	15
2	2	1	9

Table 5. FCM-yielded set of clusters on OLITOS data



a) b)

Fig. 3. The best set of clusters on OLITOS data obtained by a) FCM algorithm, b) Approach 1 of Method 1 of EFC algorithm

the above set of clusters, the value of DF is coming out to be equal to 74.5. The above multi-dimensional clusters are mapped into 2-D using the SOM, as shown in Figure 3 a). The distinctness of the above set of clusters has been proved through the above figure. The average user time value, that is, t_{av} of this algorithm is found to be equal to 0.0321 seconds.

The performances of EFC-based methods and their approaches are tested on the OLITOS data set. Approach 1 of Method 1 (corresponding to $\beta = 0.3875$) is found to yield the best set of clusters (refer to Table 6), as the DF corresponding to this method-approach combination is found to be the least.

The clusters are found to be well-distinguished from each other, through the calculation of the inter-cluster Euclidean distances. The similar observations have been

	Approach 1	**Approach 2**	**Approa ch 3**
Method 1	$\beta = 0.3875$	$\beta = 0.355$	$\beta = 0.3575$
	10 3 5 32	16 8 12 14	16 7 9 16
	1 2 4 18	7 10 3 5	5 9 4 7
	4 20 7 3	16 16 2 0	14 18 2 0
	5 3 1 2	6 3 2 0	6 3 2 0
	OL $= 0$**, DF** $= 59$	OL = 0, DF = 85.5	OL = 2, DF = 76
Method 2	$\beta = 0.7$	$\beta = 0.611$	$\beta = 0.6$
	3 0 4 6	25 1 12 9	30 7 2 9
	4 0 0 1	19 1 0 1	21 0 1 1
	0 10 2 1	14 12 1 0	15 6 9 0
	0 0 1 0	2 0 2 0	2 4 0 0
	OL = 88, DF = 107.5	OL = 21, DF = 67	OL = 13, DF = 72.5
Method 3	$\beta = 0.7$	$\beta = 0.611$	$\beta = 0.61$
	3 0 5	25 1 12 9	25 10 2 9
	4 0 1	19 1 0 1	19 0 1 1
	0 10 1	14 12 1 0	14 4 11 0
	0 0 0	2 0 2 0	2 2 0 1
	OL = 96, DF = 95.5	OL = 21, DF = 67	OL = 19, DF = 79

Table 6. The best set of clusters obtained by different method-approach combinations of EFC algorithm on OLITOS data

made from Figure 3 b) also, which is obtained after mapping the multi-dimensional data into 2-D. Computational time of the above method-approach combination of EFC algorithm is found to be equal to 0.0277 seconds.

3.4 Clustering of Psychosis Data

Psychoses data [6] have been clustered using the above two techniques, as explained below.

3.4.1 Clustering Based on FCM Algorithm

One thousand psychosis data have been clustered by pre-setting the number of clusters to 7, as there are seven identifiable psychotic diseases. After a careful study, the parameters: cluster-fuzziness f and termination criterion ϵ are set equal to 2.0 and 0.001, respectively. The clusters are formed by putting the individual data point into a cluster, with which it has the highest membership value. The resultant seven clusters are found to contain 162, 141, 148, 108, 162, 103 and 176 data-points, respectively. These clusters are mapped into 2-D for visualization using the SOM (refer to Figure 4 a)) and their qualities are assessed.

3.4.2 Clustering Based on EFC and Its Proposed Extensions

As the performance of an EFC algorithm depends on the threshold value of similarity (β), experiments are carried out for all the above method-approach combinations by varying β. Trials are made to identify the best set of clusters for each of the method-approach combinations. Approach 3 of Method 1 is found to yield the best set of clusters (seven in number containing 231, 143, 207, 159, 60, 77 and 76 datapoints) with only 4.7% outliers, corresponding to a β equal to 0.455 (refer to Figure 4 b)).

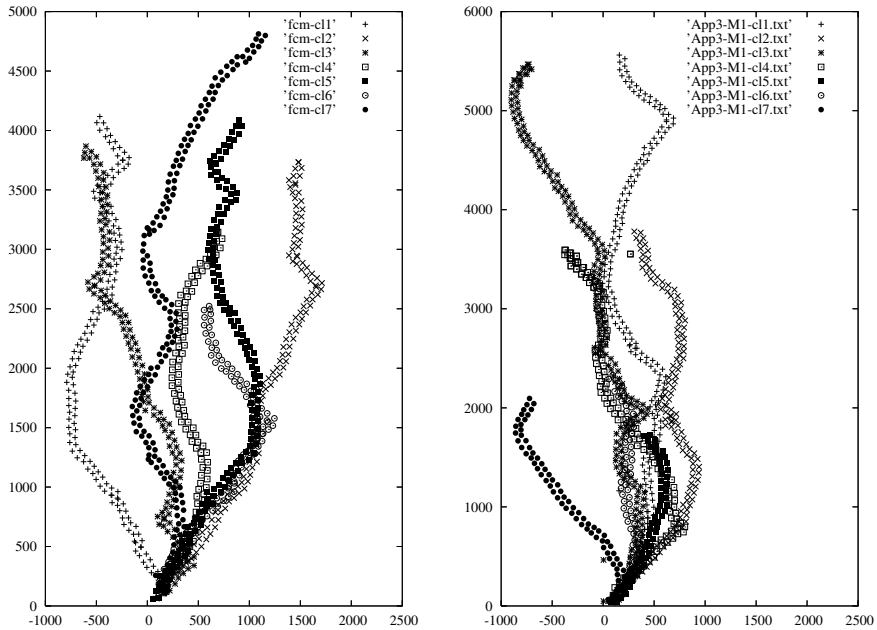


Fig. 4. The best set of clusters obtained using a) Fuzzy C-Means algorithm, b) Approach-3 of Method 1 of EFC algorithm on psychosis data

3.5 Comparisons

The performances of FCM algorithm have been compared to those of EFC algorithms, in terms of the quality of clusters obtained and their computational time values, which are discussed below.

3.5.1 Quality of the Developed Clusters

The quality of a cluster may be expressed with the help of the following measures: discrepancy factor (DF), compactness and distinctness as discussed above. Table 7

compares the best set of clusters obtained by using two algorithms (that is, FCM algorithm and EFC algorithms) on three standard data sets, such as IRIS, WINES and OLITOS.

Clustering technique	**IRIS**	**WINES**	**OLITOS**
EFC	App 1 of Method 3 $\beta = 0.675$ 00 00 50 50 00 00 09 32 00 OL = 09, DF = 18	App 3 of Method 1 $\beta = 0.575$ 57 00 00 34 02 23 00 44 00 OL = 18, DF = 55	App 1 of Method 1 $\beta = 0.3875$ 10 03 05 32 01 02 04 18 04 20 07 03 05 03 01 02 OL = 00, DF = 59
FCM	50 00 00 00 47 03 00 13 37 DF = 16	44 15 00 00 21 52 00 27 20 DF = 56	18 06 15 09 04 08 06 07 06 02 11 15 02 02 01 09 DF = 75

Table 7. The best set of clusters obtained by FCM and EFC algorithms for IRIS, WINES and OLITOS data sets

In case of IRIS data, out of different method-approach combinations of EFC algorithm, Approach 1 of Method 3 has yielded the best set of clusters, which are found to be slightly worse, in terms of DF, compared to those obtained by the FCM algorithm. Moreover, a close watch of Figures 1 a) and 1 b) reveals that both the approaches are able to yield the distinct clusters but the clusters determined by the EFC algorithm are found to be more compact compared to those provided by the FCM algorithm.

In case of WINES data, Approach 3 of Method 1 of EFC algorithm is able to identify and make better clusters (in terms of DF) compared to those of FCM algorithm. Both algorithms are able to provide with the distinct clusters but the clusters obtained by EFC algorithm are found to be more compact compared to those achieved by the FCM algorithm (refer to Figures 2 a) and 2 b)).

For OLITOS data, Approach 1 of Method 1 of EFC algorithm has recorded better performance compared to that of the FCM algorithm, in terms of DF. It has been observed from Figures 3 a) and 3 b) that EFC algorithm is able to yield more compact and distinct clusters compared to those obtained by the FCM algorithm.

In case of psychosis data, the term DF cannot be determined, as the ideal clusters are not known beforehand. Both the FCM and EFC algorithms are able to give distinct clusters but the clusters obtained by the EFC algorithm are found to be more compact compared to those yielded by the FCM algorithm.

Thus, in terms of DF, the EFC algorithm is found to perform better than the FCM algorithm in case of WINES and OLITOS data sets, whereas the former has

been defeated by the latter in case of IRIS data. It is interesting to note that EFC algorithm is able to generate more compact clusters for all the data sets. Moreover, the performances of the clustering algorithms are found to be data-dependent.

3.5.2 Computational Time

The computational time (that is, average user time) values of different method-approach combinations of EFC algorithm and the FCM algorithm have been compared, while carrying out clustering of the above three standard data sets, such as IRIS (150×3), OLITOS (120×4) and WINES (178×3), on a P-IV PC. Figure 5 shows the above comparisons on a graphical plot.

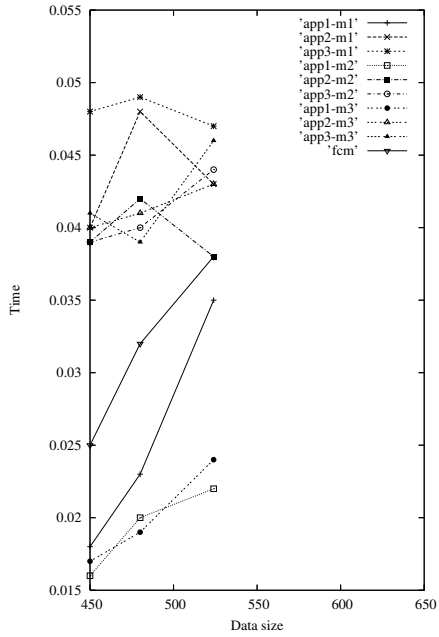


Fig. 5. Average user time values of different clustering algorithms vs. data size

It is interesting to note that Approach 3 of Method 1 of EFC algorithm is found to be the slowest of all. On the other hand, Approach 1 of Method 2 of EFC algorithm is seen to be the fastest of all, except for the OLITOS data. The FCM algorithm is found to be faster than Approach 3 of Method 1 but slower than Approach 1 of Method 2 of EFC algorithm. Except for a few combinations of EFC algorithm, the average user time values are found to increase with the data size, in general; but, the reverse trends have been noticed in Approaches 2 and 3 of Method 1, Approach 2 of Method 2 and Approach 3 of Method 3 and it could be due to the fact that the computational time depends on not only the size but also

the type of the data. It is important to note that similar observations have also been made while clustering the psychosis data using the FCM and EFC algorithms.

From the above observations, we can summarize that the computational time of Method 2 of EFC algorithm is the least, as it involves a lower amount of computation compared to other methods of EFC algorithm. Method 1 of EFC algorithm is found to take the maximum CPU time, as it deals with more arithmetic operations also involving some logarithmic terms. As far as the approaches of EFC algorithm are concerned, the computational time is seen to be the least in Approach 1, as the similarity and entropy values are calculated only once, whereas Approach 3 takes the maximum computational time, as it involves more computations to update the similarity and entropy values iteratively.

4 CONCLUDING REMARKS

From the above study, the following conclusions have been drawn:

- In terms of DF, the FCM algorithm is found to perform better than the EFC algorithm in case of IRIS data, whereas the former has been defeated by the latter in case of WINES and OLITOS data sets. Thus, the performance of the algorithm is data-dependent.
- EFC algorithm is able to yield more distinct and at the same time more compact clusters compared to those obtained by the FCM algorithm.
- Approach 3 of Method 1 and Approach 1 of Method 2 of EFC algorithm are found to be the slowest and fastest of all, respectively.
- FCM algorithm is seen to be faster than Approach 3 of Method 1 of EFC algorithm but slower than Approach 1 of Method 2 of EFC algorithm.
- Method 1 and Method 2 of EFC algorithm are found to be the slowest and fastest of all, respectively.
- Computation time of Approach 1 and Approach 3 of EFC algorithm are seen to be the least and highest of all, respectively.
- SOM algorithm is able to map higher dimensional clustered data into 2-D for visualization, after preserving the topological information intact.

5 SCOPE FOR FUTURE WORK

The present work is an attempt to carry out comparative study of FCM algorithm and EFC algorithms of clustering, in terms of the quality of the clusters made and their computational time. Fuzzy reasoning algorithms will be developed in future by using the best set of clusters, thus obtained. Presently, the authors are working on these issues.

REFERENCES

[1] Albayrak, S.—Armasyali, F.: Fuzzy C-Means Clustering on Medical Diagnostic System. Proc. Int. XII Turkish Symp. on Artif. Intel. NN, 2003.

[2] Armanino, C.—Leardi, R.—Lanteri, S. et al.; Chemom Intell Lab Syst. Vol. 5, 1989, pp. 343–354.

[3] Belhassen, S.—Zaidi, H.: A Novel Fuzzy C-Means Algorithm for Unsupervised Heterogeneous Tumor Quantification in PET. Medical Physics, Vol. 37, 2010, No. 3, pp. 1309–1324.

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

[5] Chattopadhyay, S.—Pratihar, D. K.—De Sarkar, S. C.: Performance Studies of Some Similarity-Based Fuzzy Clustering Algorithm. International Journal of Performability Engineering, Vol. 2, 2006, No. 2, pp. 191–200.

[6] Chattopadhyay, S.: Fuzzy Logic-Based Expert Systems for Screening and Prediction of Adult Psychoses. Ph. D. thesis, IIT Kharagpur, India, 2006.

[7] Cheng, C.—Fu, A. W.—Zhang, Y.: Entropy-Based Subspace Clustering for Mining Numerical Data. Proc. Int. Conf. on Knowledge Discovery and Data Mining, KDD, 1999.

[8] Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. J. Cybernet, Vol. 3, 1973, pp. 32–57.

[9] Dutta, P.—Pratihar, D. K.: Some Studies on Mapping Methods. International Journal of Business Intelligence and Data Mining, Vol. 1, 2006, No. 3, pp. 347–370.

[10] Fisher, A.: Annals of Eugenics. Vol. 7, 1936, pp. 179–188.

[11] Forina, M.—Armanino, C.—Castino, M. et al.: Vitis. Vol. 25, 1986, No. 189.

[12] Fu, L.—Medico, E.: FLAME: A Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data. BMC Bioinformatics. Vol. 8, 2007, No. 3, doi:10.1186/1471-2105-8-3.

[13] Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Heidelberg, Germany, 1995.

[14] Krinidis, S.—Chatzis, V.: A Robust Fuzzy Local Information C-Means Clustering Algorithm. IEEE Trans. on Image Processing, Vol. 19, 2010, No. 5, pp. 1328–1337.

[15] Lung, C. H.—Nandi, A.—Zaman, M.: Application of Clustering to Early Software Life Cycle Phases. `www.sce.carleton.ca/faculty/lung/`.

[16] Ma, P.—Chan, K.: Incremental Fuzzy Mining of Gene Expression Data for Gene Function Prediction. IEEE Trans. on Biomedical Engineering, 2010.

[17] Manco, G.—Ortale, R.—Sacca, D.: Similarity Based Clustering of Web Transaction. Proc. ACM Symp. of appl. compu., 2003.

[18] Migaly, S.—Abonyi, J.—Szeifert, F.: Fuzzy Self-Organizing Map Based on Regularized Fuzzy C-Means Clustering. Advances in Soft Computing, Engineering Design and Manufacturing, in J. M. Benitez, O. Cordon, F. Hoffmann, et al. (Eds.), Springer Engineering Series, (Revised papers of the 7th On-line World Conference on Soft Computing in Industrial Applications (WSC7)), 2002, pp. 99–108.

[19] Oikonomakou, N.—Vagirgiannis, M.: Web Document Clustering. NEMIS conference, `http://www.db-net.aueb.gr/`, 2003.

[20] Pal, N. R.—Bezdek, J. C.: On Cluster Validity for the Fuzzy C-Means Model. `IEEEFS`, Vol. 3, 1995, No. 3, p. 370.

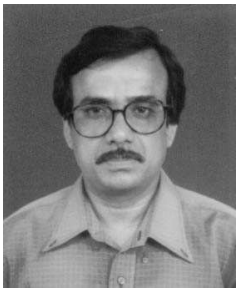
[21] Pratihar, D. K.: Soft Computing. Narosa Publishing House, New-Delhi, India, 2008.

[22] Sikka, K.—Sinha, N.—Singh, P. K.—Mishra, A. K.: A Fully Automated Algorithm Under Modified FCM Framework for Improved Brain MR Image Segmentation. Magnetic Resonance Imaging, Vol. 27, 2009, No. 7, pp. 994–1004.

[23] Yao, J.—Dash, M.—Tan, S. T.—Liu, H.: Entropy-Based Fuzzy Clustering and Fuzzy Modeling. Fuzzy Sets and System. Vol. 113, 2000, pp. 381–388.

[24] Zhang, D. Q.—Chen, S. C.: A Novel Kernelized Fuzzy C-Means Algorithm With Application in Medical Image Segmentation. Artif. Intel. Med, Vol. 32, 2004, pp. 37–50.

Subhagata Chattopadhyay is a healthcare informatics professional with both clinical and technical skills. He received his MBBS and DGO from Calcutta Medical College, Calcutta University, India; Master of Science in bioinformatics from Sikkim Manipal University, India; and a Ph. D. degree from the School of Information Technology, Indian Institute of Technology, Kharagpur, India. He was a postdoctoral fellow at Asia-Pacific u-Health Centre in Australian School of Business, The University of New South Wales (UNSW), Sydney (Australia) and coordinated a large project on assessment of e-Health implementation-evaluations for the developing countries under the leadership of World Health Organization (WHO) and UNSW. Knowledge engineering, knowledge management, expert systems and e-Health implementation-evaluations remain his key research fields. He has published over forty technical papers in various peer-reviewed journals and conferences of international repute. He has also published several book chapters and possesses one Indian copyright for his innovative work on mental health informatics. He is an international board member of journal of e-working and editorial board member of SMST Window. He has chaired several sessions and delivered tutorials in international conferences and symposiums. His biography has been selected as a distinguished medical professional in Marquis Who's Who (Medicine and Healthcare) in 2006–2007. Currently he is a full Professor in the Department of Computer Science and Engineering, National Institute of Science and Technology, India.



Dilip Kumar Pratihar received his Ph. D. from IIT Kanpur, India, in the year 2000. Besides several scholarships and the best paper awards, he received the University Gold Medal for securing the highest marks in the University in 1988, A. M. Das Memorial Medal in 1987, Institution of Engineers' Medal in 2002, and others. He completed his post-doctoral studies in Japan (6 months) and Germany (1 year) under the Alexander von Humboldt Fellowship Programme. He is working, at present, as a Professor, in the Department of Mechanical Engineering, IIT Kharagpur, India. His research areas include robotics, soft computing and manufacturing science. He has published more than 125 papers in different journals and conference proceedings. He has written a textbook on "Soft Computing", which has been published by Narosa Publishing House, New Delhi and Alpha Science International Publisher, UK. Recently, this book has been translated into Chinese language. He has edited a book on "Intelligent and Autonomous Systems", which has been published by Springer-Verlag, Germany, in 2010. Recently, he has co-authored another textbook on "Analytical Engineering Mechanics", which is in press of Narosa Publishing House, New Delhi, India. He has been included as a member of the program committee for several International Conferences. He has been selected as the Editorial Board Member of eight International Journals. He has been elected as a Fellow of the Institution of Engineers (I) and Member of IEEE.

Sanjib Chandra De Sarkar obtained his M. Tech. and Ph. D. degrees from the University of Calcutta, India. In1971, Professor De Sarkar joined the University of Calcutta as Lecturer. In 1977 he moved to Indian Institute of Technology (IIT) Kharagpur, India, where he continued up to February 2006. There he served as Assistant Professor in the Department of Electronics and Electrical Communication Engineering and as Professor in the Department of Computer Science and Engineering and in the School of Information Technology. He also served the Institute as the Head of Computer and Informatics Center, Head of Computer Science and Engineering Department, Founder Head of School of Information Technology, Dean of Academic Affairs and Deputy Director. In February, 2006 he became the Vice Chancellor of KIIT University, Bhubaneswar, India. In March 2010, he joined Indian Institute Technology (IIT) Bhubaneswar, India, where he is serving as Professor and Technical Adviser. Principal research interests of Professor De Sarkar include artificial intelligence and knowledge based systems, algorithms and compiler design.