Fuzzy Cluster Methods for Imputing Incomplete Data

Ahmed M. Elshewey
Computer Science Department,
Faculty of Computers and Information, Suez University, Egypt.
elshewy86@gmail.com

Amira. A. Elsonbaty
Communication & Electronics Department,
Higher institute of engineering and technology,
New Damietta, Egypt, 34517
amira.elsonbaty@ndeti.edu.eg

Abstract Most of the clustering approaches are unsupervised methods that can be used to organize information into groups based on the similarities between individual data objects. The majority of clustering algorithms do not rely on theories common to traditional statistical approaches, such as the underlying distribution of statistical data, and are therefore useful in situations where there is little prior knowledge. The ability of clustering algorithms to expose the underlying data structures can be used in a broad variety of ways. Classification, image processing, recognition of patterns, modelling and identification are included. An overview of fuzzy clustering algorithms based on the functional and neural-fuzzy models of c-means is provided in this paper. Three approaches are used to measure the missing details. The findings showed that among the three strategies, the first approach provides the best outcome.

Keywords: Fuzzy C-Means (FCM), Backpropagation, Neuro-Fuzzy, Missing Data, Clustering.

1. INTRODUCTION

Fuzzy sets make it possible that elements are partly in a set. A degree of membership in a set is given to any element. This membership value can range from 0 (not the set element) to 1 (the set member). A membership function is a relationship between an element's value and its membership degree in a set. Figure 1 gives an overview of membership features. The sets (or classes) in this example are numbers which are negative large negative medium, negative small, near-zero, positive numbers small, positive medium and positive large. The value μ is the amount of membership in the set.

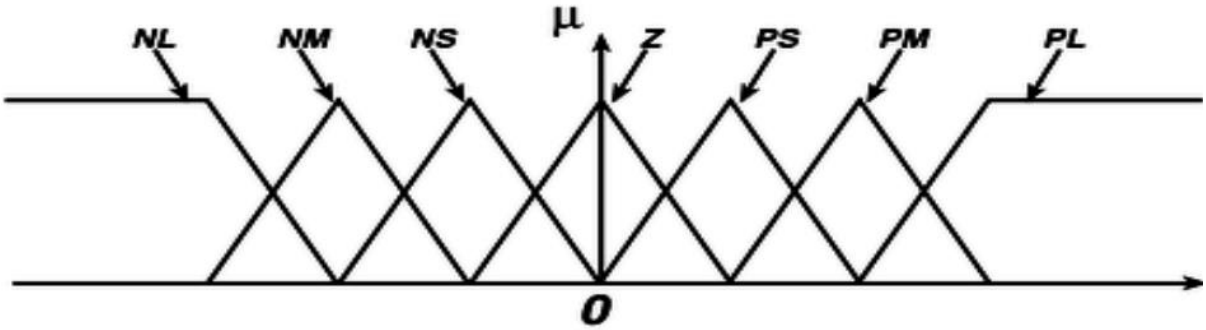

Figure 1. Membership functions of the set of all numbers (N=Negative, P=Positive, L=Large, M=Medium, S=Small).

The process of separating data elements into classes or clusters is data clustering, so that items in the same class are as similar as possible and items in different classes are as different as possible. Different similarity measures can be used to position objects in groups depending on the nature of the data and the reason for which clustering is being used, where the similarity measure controls how the clusters are created. Width, connectivity, and strength are some examples of measures that can be used as in clusters. The problem of clustering a real s-dimensional data set $X = \{x_1, x_2, \ldots\ldots x_n\} \subset R^s$ is considered. Usually, each observation (or datum) consists of numerical values for all s features (such as height, length, etc.), but sometimes data sets can contain vectors that are missing one or more of the feature values. For example, a particular datum x_k might be incomplete, having the form $x_k = (254.3, ?, 333.2, 47.44, ?)^T$, where the second and fifth feature values are missing. The fuzzy c-means (FCM) algorithm [1] is a useful tool for clustering real s-dimensional data, but it is not directly applicable to the

case of incomplete data. We are interested in clustering a set of objects O = { $o_1, \ldots., o_n$} represented by a numerical object data set X = {$x_1, x_2, \ldots\ldots x_n$} $\subset R^S$ into c clusters, 1 < c < n. The numerical data describe the objects by specifying values for s particular features. For example, if the jth component of each datum corresponds to the feature of weight, then the jth component of x_k, denoted by x_k, gives the weight of object o . In some situations, feature vectors in X can have missing components [2], [3]. Any datum with some (but not all) missing feature values is referred to as an incomplete datum. An example of an incomplete datum is $x_5 = (1.23, 3.14, ?, 4.66, ?)^T$, where x_{53} and x_{55} are missing. A dataset with at least one incomplete datum is referred to as an incomplete data set; otherwise, it is called complete. An important empirically-oriented study was done in 1979 [1], [2] and those results are summarized in [2]. It may be the case that predicted values of the missing data are desired [4], [5], [6] (or even predicted values of new data, as in [7]), but often the goal is to simply use the available data to perform some analysis [8], [9]. The goal of the new clustering approaches introduced later is to partition the data set into fuzzy clusters and provide estimates of their cluster centers.

2. THE GEOMETRY OF DATA MISSING

We start with a discussion of visualizing incomplete sets of data. In the special case that X $\subset R^s$, it is possible to accurately represent both complete and incomplete data sets via scatter diagrams. To do this as a vertical line with a horizontal portion, we represent an incomplete date. x_k=($x_{k1}, ?)^T$ as a vertical line with the horizontal component x_{k1} and x_k=$(? \, x_{k2})^T$as a horizontal line with a vertical component x_{k2}. Of course, it is not possible to usefully represent the null datum (??) in such a graph. In higher dimensions, depending on the number of missing feature values, an incomplete date could correspond to a line, plane, etc. In figure 2a, the left scatter diagram corresponds to a full data set. An incomplete version of the data is shown in the right plot in figure 2b, where 25 percent of the 20 function values are missing. Three values for the first feature and two values for the second feature are missing in this case. Note the increased difficulty of visually identifying the presence (and location) of the two clusters when there is a lack of a large percentage of feature values.

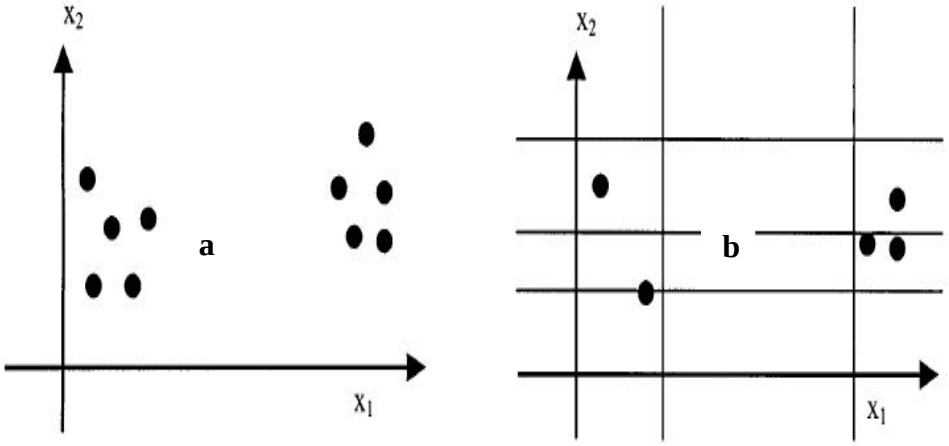

Figure 2. Complete data on left. Incomplete data with 25 of feature values missing on the right.

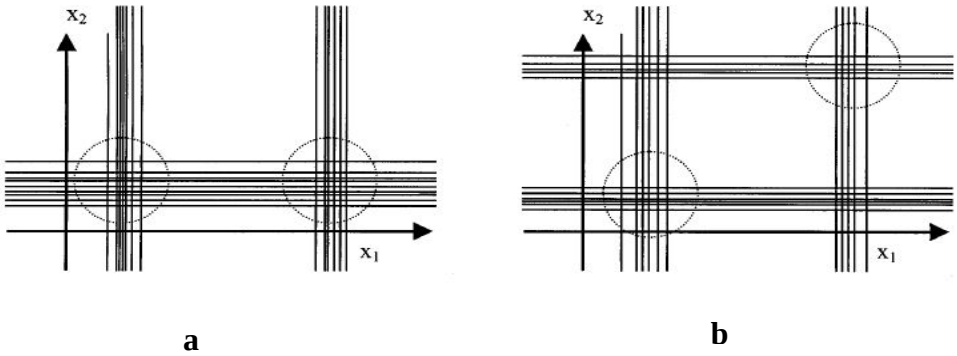

Figure 3. Two maximally incomplete dimensional (2-D) data sets.

There is an important extreme case of the visual difficulty shown in figure 2 that contains what we are going to call "virtual clusters." We consider the most extreme case of incompleteness to make this phenomenon understandable. If each data in X has exactly one known feature value, we can refer to a data set X as maximally incomplete. In the case of a maximally incomplete data set X $\subset R^s$, exactly 50% of the feature values are missing. In this case, there will be no points in a plot such as the right side of figure 2; instead, it will consist entirely of a lattice of n horizontal/vertical lines as shown in figure 3. More generally, a maximally incomplete data set X $\subset R^s$ is missing exactly 100(s-1) /s % of its feature values. It is generally impossible to find good estimates for the cluster centers if X is maximally incomplete; we demonstrate this using the examples in figure 3. Each of the two plots in figure 3 indicates a two-dimensional (2-D) maximum incomplete data collection, which would have two visual clusters if completed. These maximally incomplete data sets were obtained by beginning with complete data whose general locations are indicated by the circles; and then designating one of its two functional values as "missing" for each datum. If we know that c = 2, then from the maximally incomplete data set, is it possible to recover fair estimates of the original (complete data cluster centers? In an incomplete data collection, this is one purpose of cluster analysis. The response is yes for the data underlying the left view in Figure 3, but no for the right view, suggesting two additional "virtual clusters." In figure 3, each of the two plots is compatible with some other complete data sets that have more than two clusters. It is difficult to cluster, and this example highlights an additional challenge inherent in incomplete data clusters.

3. FCM CLUSTERING METHODS

For clustering (whole) object data sets X, the FCM clustering algorithm is commonly used and is used in a pure or modified form. FCM attempts to build a description of the (fuzzy cluster substructure of X along with examples (or prototypes) for each of the clusters using an iterative solving scheme at the same time. The FCM method has been commonly used, adapted and [10-13] generalized. Let's follow the notation that will be used throughout:

x_k=k^{th} s-dimensional data vector, for 1 ≤ k ≤ n (1)

x_{kj} = j^{th} feature value of the k^{th} data vector for 1 ≤ j ≤ s, 1 ≤ k ≤ n (2)

X = $\{x_1, \dots \dots . x_n\}$ (3)

X_W = $\{x_k \in X \mid x_k$ is a complete datum} (the whole – data subset of X) (4)

X_P = $\{x_{kj}$ for 1 ≤ j ≤ s, 1 ≤ k ≤ n | the value of x_{kj} is present in X} (5)

X_M = $\{x_{kj}$=? for 1 ≤ j ≤ s, 1 ≤ k ≤ n | the value of x_{kj} is missing from X} (6)

For example, lets = 3, n = 4, and

X = { [2 ? ?] [3 2 4] [2 1?] [1 1 2] }

Then

X_W = { [3 2 4] [1 1 2] }

X_M = $\{x_{12}, x_{13}, x_{33}\}$ and X_P = $\{x_{11} = 2,\ x_{21} = 3,\ x_{22} = 2,\ x_{23} = 4,\ x_{31} = 2,\ x_{32} = 1,\ x_{41} = 1,\ x_{42} = 1,\ x_{43}$=2}.

Notice that the available feature values in X_P and X are the same, and that only a subset of these feature values is available in X_W. In this sense X_P and X, contain more "information" than X_W. The objective function is given by equation (7).

$min_{U,v}\{J_m(U, v) = \sum_{i=1}^{c}\sum_{k=1}^{n} U_{ik}^m \|x_k - v_i\|_A^2\}$ (7)

Where m >1 is the fuzzification parameter, $\|x_2\|_A^2 = x^T Ax$ is the vector A-norm

$U \in M_{fcn}$ = $\begin{Bmatrix} U \in R^{c\times n}: U_{ik} \in [0,1]\ \forall\ i,k;\ \sum_{i=1}^{c} U_{ik}\ 1\ \forall k; \\ \sum_{k=1}^{n} U_{ik} > 0\ \forall\ i \end{Bmatrix}$ (8)

and v = $\{v_1, v_2, \dots \dots \dots . v_c\} \subset R^s$. (9)

When A =$I_{s\times s}$, $\|x\|_A$ =$\|x\|_2$, the Euclidean norm. The elements of the membership matrix U represent the (relative) degrees of membership of each datum in each cluster; i.e., U_{ik} = the degree to which x_k belongs to the i th cluster. Minimizing U values of equation (7) are less fuzzy for values of m near 1 and fuzzier for large values of m. The choice of m=2 is widely accepted as a good choice of fuzzification parameter. The set M_{fcn} in equation (8) is referred to as the set of non-degenerates (meaning each cluster is nonempty) fuzzy $c \times n$ partition matrices. The vector v_i is the prototype or Centre of the ith cluster. The FCM algorithm for solving equation (7) alternates optimizations of J_m over the U and v variables. This method of alternating optimization has been well studied [14-15] and typically provides iterate sequences that converge q -linearly to local (and sometimes global) minimizers. The details of one version of FCM are now given. The fuzzy c-means algorithm is satisfied with the algorithm (1).

Algorithm 1: Fuzzy c-means algorithm

1- Fix m, c, and ε satisfying: m >1, 1< c < n, and ε > 0. Pick $v^0 \subset R^s$, an $||\cdot||_A$ norm for J_m, and a termination norm $||\cdot||$ for FCM. Then at step r, r =0 ,1, 2,.....

2- Calculate

$$U^{(r+1)} = arg \min_{U \in M_{fcn}} \{J_m(U, v^r)\} \quad (10)$$

using the following for $1 \le j \le c$ and $1 \le k \le n$:

$$U^{(r+1)} = \left(D_{ik}^{1/(1-m)}\right) / \left(\sum_{j=1}^{c} D_{jk}^{1/(1-m)}\right) \quad (11)$$

where $D_{ik} = \left\|x_k - v_i^{(r)}\right\|_A^2$ (12)

3- Calculate

$$v^{(r+1)} = arg \min_{v} \left\{J_m(U^{(r+1)}, v)\right\} \quad (13)$$

using the following for $1 \le i \le c$ and $1 \le j \le s$:

$$v_{ij}^{(r+1)} = \left(\sum_{k=1}^{n} (U_{ik}^{(r+1)})^m \; x_{kj}\right) / \left(\sum_{k=1}^{n} (U_{ik}^{(r+1)})^m\right) \quad (14)$$

4- Using J (y) < ε , where y = 0,1,2,..... . If true, then stop. Otherwise, set and return to FCM-2.

3.1. Whole Data Strategy (WDS)

If the proportion of incomplete data is small, then simply removing all incomplete data and applying FCM to the remaining complete data can be useful. This is what we will refer to as the whole data strategy (WDS). Although we have not carefully studied this problem, setting a fairly strict limit on the number of vectors that are cast out of X because they have one or more missing values seems appropriate. We suppose that the WDS should be used only if $\frac{n_p}{n_s} \ge 0.75$, where n_p = | X_P | and n_s = | X | · s. In the notation introduced at the beginning of the section, this simply amounts to applying FCM to the data subset X_W. The WDS method may provide good prototype estimates v={v 1,v 2,.....,v c}, but it will not explicitly provide feature vector cluster membership information, as the data in this set are not represented by columns in the corresponding reduced version. However, cluster membership can be calculated for incomplete data using the nearest prototype classification scheme based on the corresponding reduced version of the prototype.

3.2. Partial Distance Strategy (PDS)

For the case that X M is so high that the WDS cannot be justified, the top strategy suggested by Dixon[1] consists of calculating partial (squared Euclidean) distances using all usable (i.e. non-missing) feature values, and then scaling this sum by the reciprocal proportion of the components used. We will call this the partial distance strategy (PDS) and use the illustration to explain the PDS form of the D-ik calculation in equation (12) as:

$D_{ik} = \|x_k - v_i^{(r)}\|_2^2$

$= \|(1\ ?\ ?\ 4\ ?)^T - (5\ 6\ 7\ 8\ 9)^T\|_2^2$

$= \frac{5}{(5-3)}((1-5)^2 + (4-8)^2)$.

The general formula for the partial distance calculation of D_{ik} is given by:

$$D_{ik} = \frac{s}{I_k} \sum_{j=1}^{s} (x_{kj} - v_{ij})^2 \, I_{kj} \quad (15)$$

Where

$$I_{kj} = \begin{cases} 0, & \text{if } x_{kj} \in X_M \\ 1, & \text{if } x_{kj} \in X_P \end{cases} \quad \text{for } 1 \le j \le s \text{ and } 1 \le k \le n$$

and $I_k = \sum_{j=1}^{s} I_{kj}$

The PDS version of the FCM algorithm, referred to here as PDSFCM, is obtained by making two modification of the FCM algorithm for the special case of $\|x\|_A = \|x\|_2$. These modifications are:

1- Calculate D_{ik} as in equation (15) for incomplete data.
2- Replace the calculation of $v_{ij}^{(r+1)}$ in equation (14) with equation (16).

$$v_{ij}^{(r+1)} = \left(\sum_{k=1}^{n}(U_{ik}^{(r+1)})^m \; x_{kj} \; I_{kj}\right) / \left(\sum_{k=1}^{n}(U_{ik}^{(r+1)})^m \; I_{kj}\right) \quad (16)$$

3.3. Fuzzy C-Means for Imputing Missing Data

The fuzzy c-means method is used for imputing the missing data in the data set.
We estimate the missing data from the following equation:

$$l = U_{ik} \; v_{ij} \quad (17)$$

After calculating equation (17), we estimate each missing value in the data set to the corresponding value in equation (17).

4. NEURO-FUZZY MODEL

Neuro-fuzzy refers to artificial neural network combinations and fuzzy logic[16, 17]. By integrating the human-like reasoning style of fuzzy systems with the learning and interaction structure of neural networks, neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques. There are several aspects in common with both neural networks and fuzzy systems. They can be used to solve a problem (e.g. classification of patterns, regression or density estimation) if no mathematical model of the given problem exists. They only have some drawbacks and benefits that vanish almost entirely when integrating both definitions. We used fuzzy c-means with a backpropagation algorithm in this section. To classify our data set as follows, we used three methods:

4.1. First Method

In this method, after importing the data set using equation, we trained our data set through the backpropagation algorithm (17), the error for the neurons in the output layer is calculated according to this equation:

$$\text{Error} = \text{OUT}\,(1 - \text{OUT})\,(U_{ik} - \text{OUT}) \quad (18)$$

4.2. Second Method

We trained our data set in this process without imputing missing values. The output, according to this equation, of each neuron in the hidden layer:

$$Z = F(net1) = 1/(1+e^{-net1}) \quad (19)$$

Where

$$net1 = \sum_i w_{ij} x_i$$

s.t, *i* is the number of input nodes, j is the number of hidden nodes. Equation (19) is used without missing values when the data is set. We measure net1 in accordance with the following equation when missing values are found in the dataset:

$$net1 = \sum_i w_{ij} x_i \; I_i \quad (20)$$

where

$$I_i = \begin{cases} 0, & \text{if } x_i \in X_M \\ 1, & \text{if } x_i \in X_P \end{cases}$$

s.t, X_M is the pattern with missing values, X_P is the complete pattern.
The error for the neurons in the output layer is calculated according to equation (18).

4.3. Third Method

In this technique, the data set is split into two parts, the first part contains the full patterns, and the second part contains the missing values patterns. First, we train the complete patterns, and in training the patterns that have missing values, the modified weights produced from the complete training patterns are used. Second, the modified weights created from the training of the patterns used in the testing stage with missing values. The error is determined according to equation (18) for the neurons in the output layer.

5. APPLICATIONS

For classification on incomplete data sets, such as an iris data set, we applied the above approaches.

5.1. Iris Data set

A multivariate data set is the iris flower data set, or Fisher's iris data set. On the iris data set containing missing values, we apply the fuzzy c-means algorithm (1), take the fuzzification parameter = 2, ε = 2 and the cumulative iteration for the objective function was 20 iterations. We get the objective function after 20 iterations after training the algorithm as:

obj_fcn(1)=5.0388×10^{008}
obj_fcn(2) = -152.5520
obj_fcn(3) = -152.5391
obj_fcn(4) = -152.0550
obj_fcn(5) = -149.1206
obj_fcn(6) = -86.2745
obj_fcn(7) = -57.8511
obj_fcn(8) = 2.0644
obj_fcn(9) = 1.1895×10^{003}
obj_fcn(10) = -8.7147×10^{003}
obj_fcn(11) = -143.0042
obj_fcn(12) = 29.6309
obj_fcn(13) = -391.6248
obj_fcn(14) = -151.5536
obj_fcn(15) = 29.5731
obj_fcn(16) = 94.3892
obj_fcn(17) = -166.8372
obj_fcn(18) = -42.3832
obj_fcn(19) =$-3.1269e\times10^{004}$
obj_fcn(20) = -75.3997

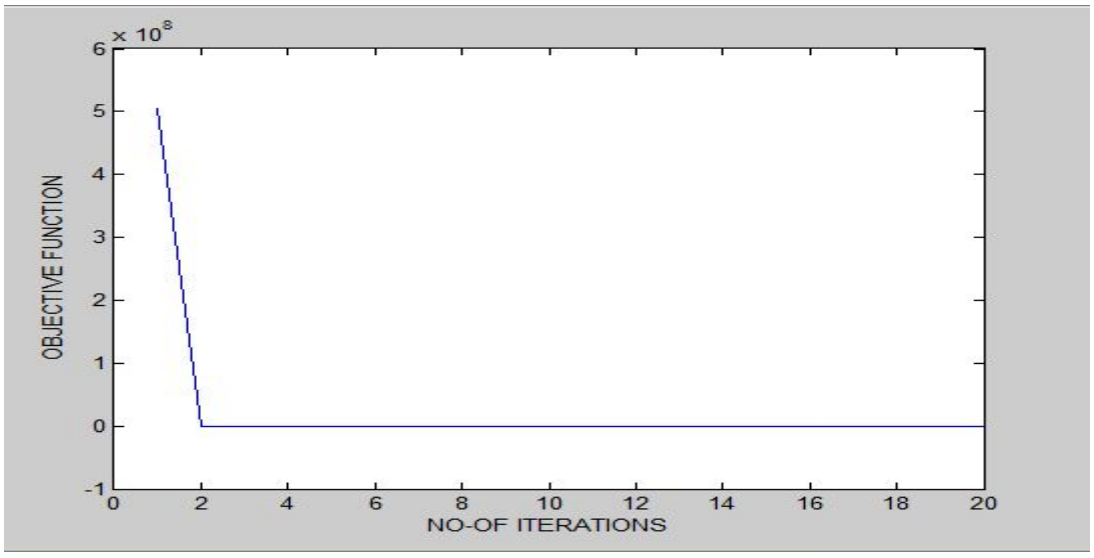

Figure 4. Relation between the number of iterations and the function of the goal.

The mean square error is 0.4 when we apply the fuzzy c-means imputation on the iris data set containing missing values. Using three methods in section (4), we applied the neuro-fuzzy model. For the classification of the data collection, a feed-forward neural network is used. We applied 90 data set patterns for training and 60 patterns for testing in the first process. The feed-forward neural network consists of three layers; four neurons are the input layer, four neurons are one hidden layer, and three neurons are the output layer. We used 0.1 for the learning rate. The misclassified error was 6% of the report. Figure (5) represents the mean square error for the first performance during training.

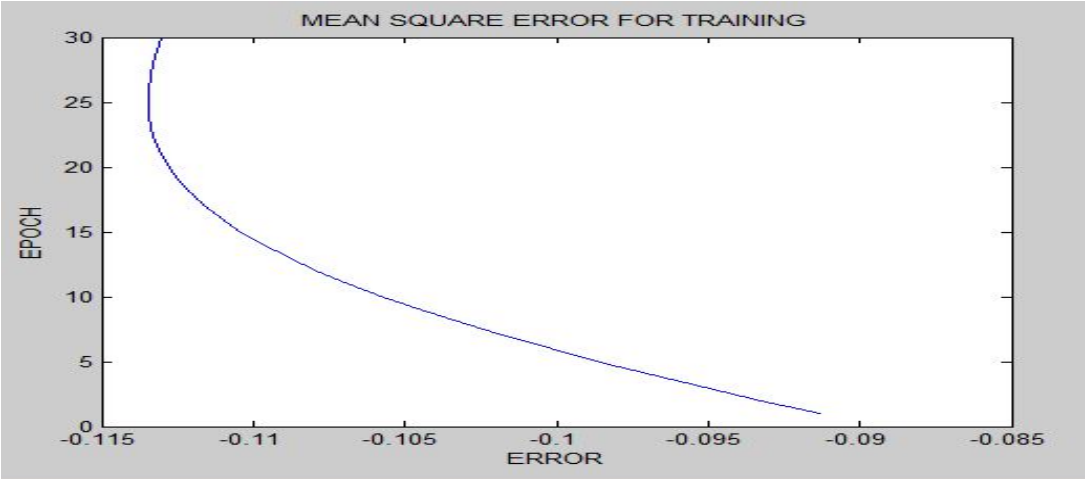

Figure 5. Mean square error during training for the first output.

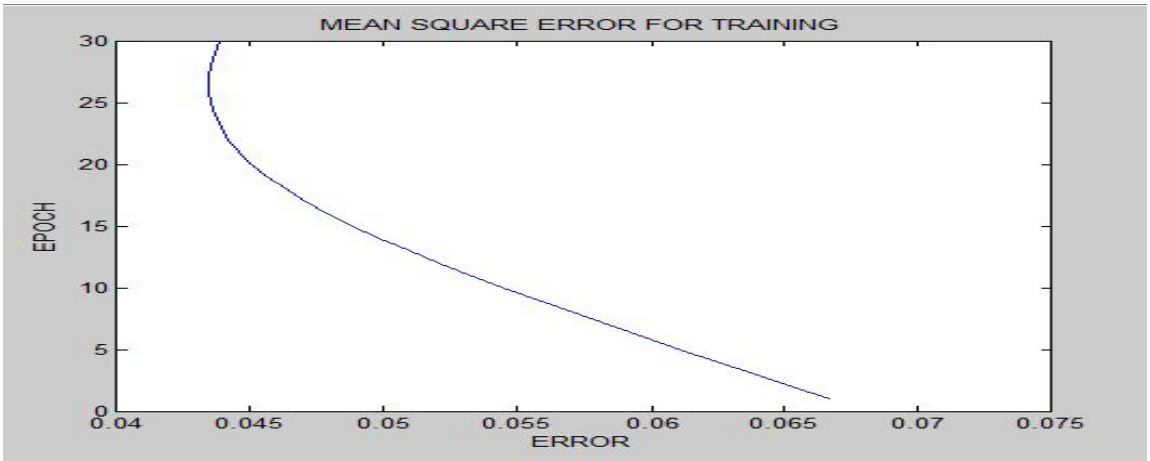

Figure 6. Mean square error during training for the second output.

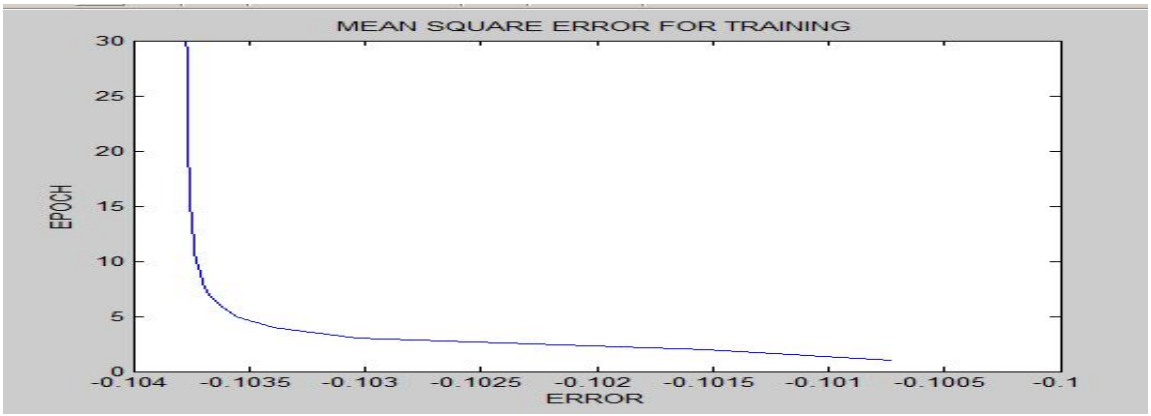

Figure 7. Mean square error during training for the third output.

In the second approach, we used 90 data set patterns for training and 60 patterns for research. The feed-forward neural network consists of three layers; four neurons are the input layer, four neurons are one hidden layer, and three neurons are the output layer. We used 0.1 for the learning rate. The misclassified mistake was 8% of the evaluation.

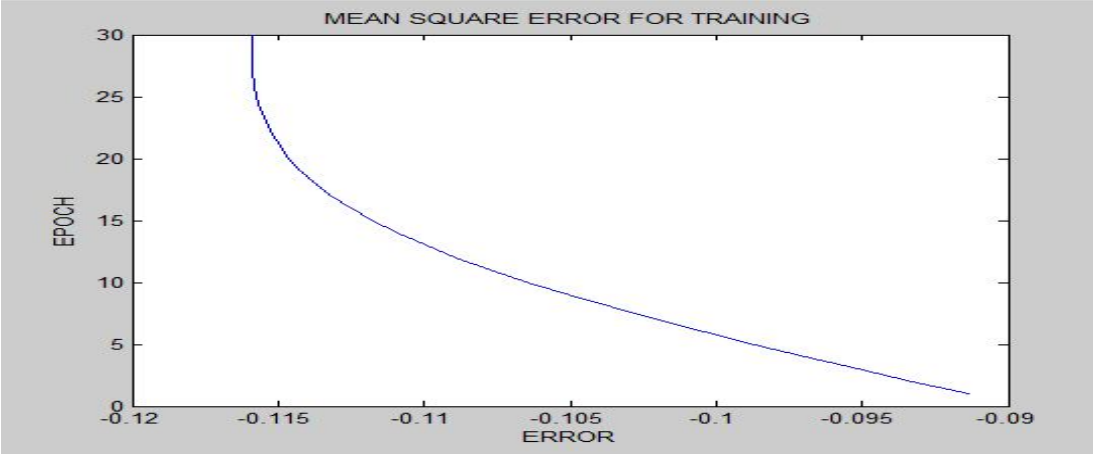

Figure 8. Mean square error during training for the first output.

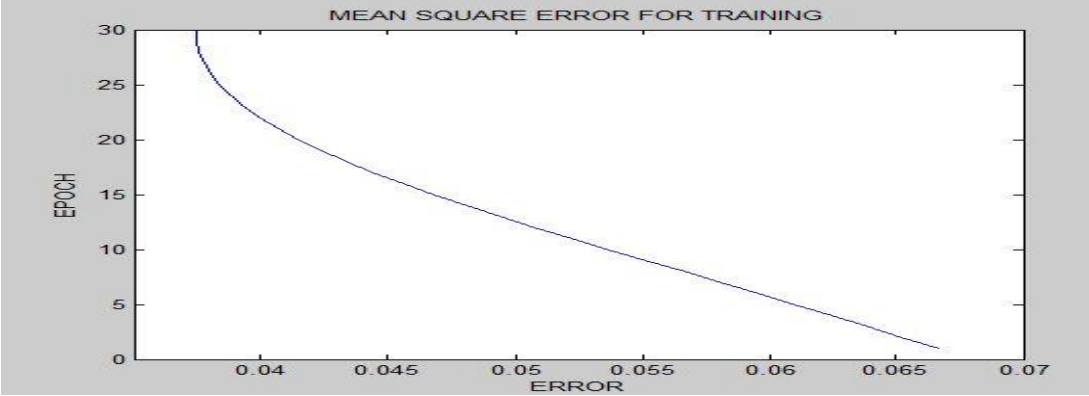

Figure 9. Mean square error during training for the second output

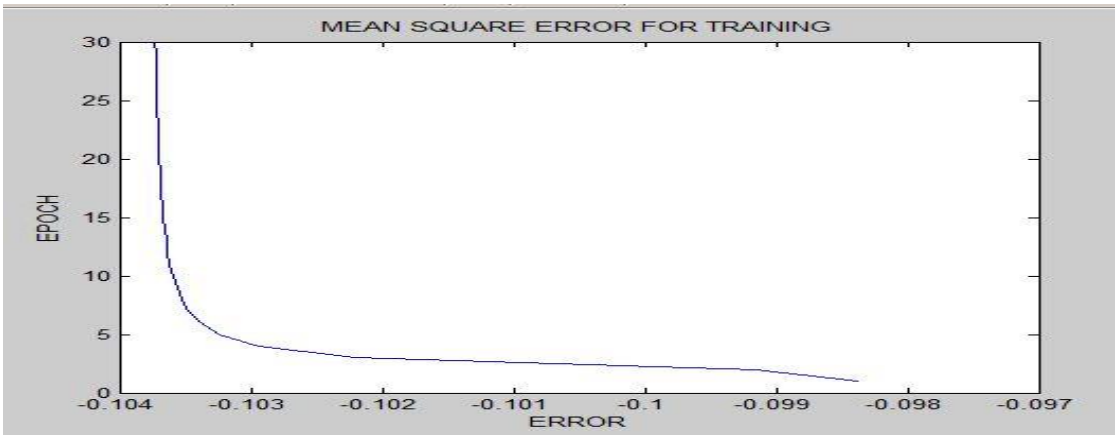

Figure 10. Mean square error during training for the third output

We used 90 data set patterns for training and 60 patterns for study in the second method. Three layers consist of the feed-forward neural network; the input layer is four neurons, one hidden layer is four neurons, and the output layer is three neurons. For the learning rate, we used 0.1. The misclassified error was 8 percent of the assessment. For the first output during training, Figure 8 reflects the mean square error.

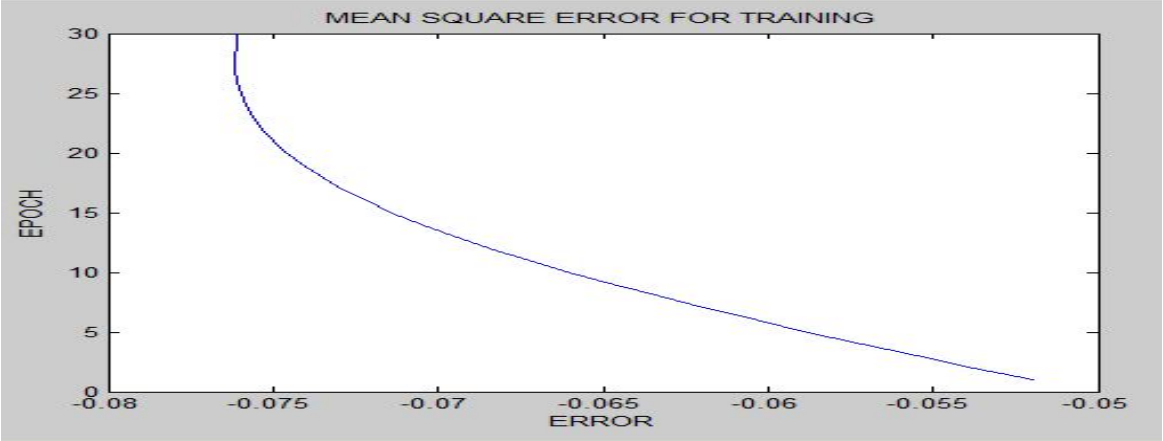

Figure 11. Mean square error during training for the first output

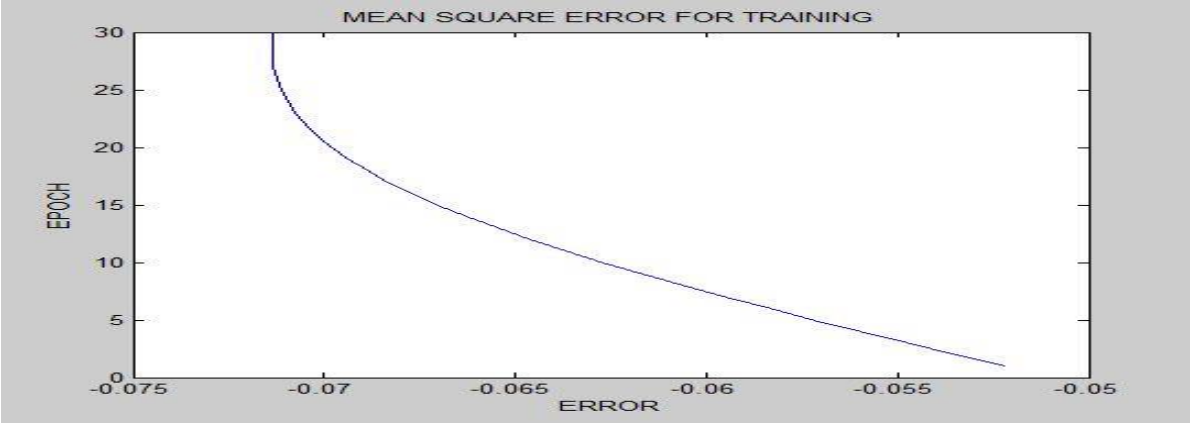

Figure 12. Mean square error during training for the second output.

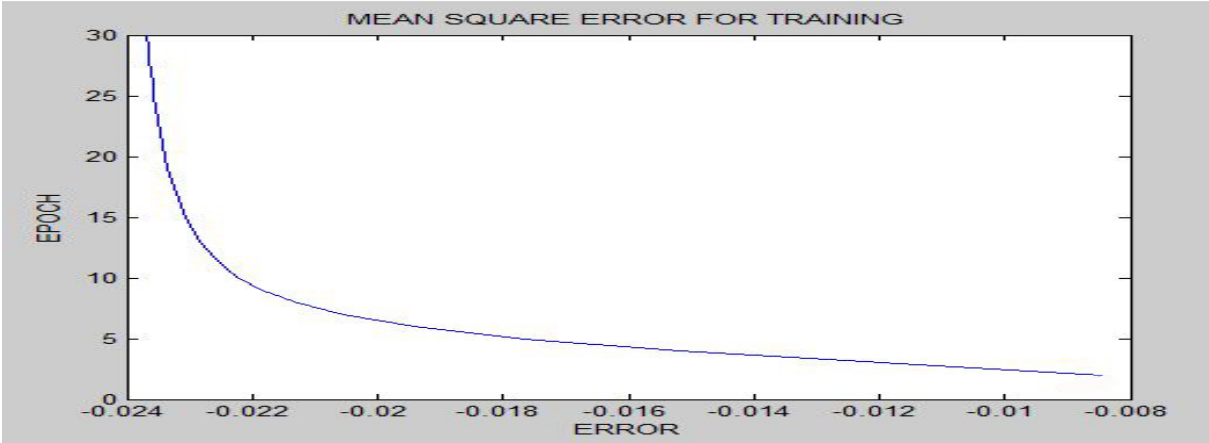

Figure 13. Mean square error during training for the third output.

The misclassified error created from the three neuro-fuzzy model methods used on the iris data set is represented in Table 1.

Method	Method 1	Method 2	Method 3
Misclassified Error	6%	8%	9%

Table 1. The misclassified error produced from the three methods of the neuro-fuzzy model.

6. CONCLUSION

We have addressed fuzzy c-means (FCM) for clustering in this paper. FCM attempts to generate a description of the (fuzzy) cluster along with examples for each of the clusters using an iterative solving scheme at the same time. Neuro-fuzzy refers to artificial neural network combinations and fuzzy logic. With the backpropagation algorithm,

we used fuzzy c-means (FCM). To classify our data collection, we used three methods. The findings showed that the least misclassified error is given by the first method, so the first method is the best method among the three methods.

7. REFERENCES

[1] Li, D., Gu, H. and Zhang, LA "fuzzy c-means clustering algorithm based on nearest-neighbor intervals for incomplete data", Expert Systems with Applications, 37(10), pp.6942-6947, 2010.
[2] Bezdek, J.C., "Pattern recognition with fuzzy objective function algorithms", Springer Science & Business Media, 2013.
[3] Di Nuovo, A.G., "Missing data analysis with fuzzy C-Means: A study of its application in a psychological scenario", Expert Systems with Applications, 38(6), pp.6793-6797, 2011.
[4] Li, D., Gu, H. and Zhang, L., "A hybrid genetic algorithm–fuzzy c-means approach for incomplete data clustering based on nearest-neighbor intervals", Soft Computing, 17(10), pp.1787-1796, 2013.
[5] Goel, S. and Tushir, M., "A new iterative fuzzy clustering approach for incomplete data", Journal of Statistics and Management Systems, 23(1), pp.91-102, 2020.
[6] Li, T., Zhang, L., Lu, W., Hou, H., Liu, X., Pedrycz, W. and Zhong, C., "Interval kernel fuzzy c-means clustering of incomplete data", Neurocomputing, 237, pp.316-331, 2017.
[7] Aydilek, I.B. and Arslan, A., "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm", Information Sciences, 233, pp.25-35, 2013.
[8] Tang, J., Zhang, G., Wang, Y., Wang, H. and Liu, F., "A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation", Transportation Research Part C: Emerging Technologies, 51, pp.29-40, 2015.
[9] Hamidzadeh, J. and Moradi, M., "Enhancing data analysis: uncertainty-resistance method for handling incomplete data", Applied Intelligence, 50(1), pp.74-86, 2020.
[10] Rahman, M.G. and Islam, M.Z., "Missing value imputation using a fuzzy clustering-based EM approach", Knowledge and Information Systems, 46(2), pp.389-422, 2016.
[11] Himmelspach, L. and Conrad, S., "June. Fuzzy clustering of incomplete data based on cluster dispersion", in, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (pp. 59-68). Springer, Berlin, Heidelberg, 2010.
[12] Zhang, L., Lu, W., Liu, X., Pedrycz, W. and Zhong, C., "Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values", Knowledge-Based Systems, 99, pp.51-70, 2016.
[13] Amiri, M. and Jensen, R., "Missing data imputation using fuzzy-rough methods", Neurocomputing, 205, pp.152-164, 2016.
[14] Zhang, L., Bing, Z. and Zhang, L., "A hybrid clustering algorithm based on missing attribute interval estimation for incomplete data", Pattern Analysis and Applications, 18(2), pp.377-384, 2015.
[15] Sefidian, A.M. and Daneshpour, N., "Missing value imputation using a novel grey based fuzzy c-means, mutual information-based feature selection, and regression model", Expert Systems with Applications, 115, pp.68-94, 2019.
[16] Nikfalazar, S., Yeh, C.H., Bedingfield, S. and Khorshidi, H.A., "Missing data imputation using decision trees and fuzzy clustering with iterative learning", Knowledge and Information Systems, 62(6), pp.2419-2437, 2020.
[17] Bu, F., Chen, Z., Zhang, Q. and Yang, L.T., "Incomplete high-dimensional data imputation algorithm using feature selection and clustering analysis on cloud", The Journal of Supercomputing, 72(8), pp.2977-2990, 2016.