Chapter 2

Artificial Neural Network and Classification Techniques
2.1 INTRODUCTION

Cluster analysis is used for grouping of objects that have similar properties. **K-means algorithm** is a well known partitioning technique used for obtaining clusters. K-means algorithm uses mostly euclidean distance measure for grouping of objects. The algorithm is not that efficient when the sample observations belong to a statistical population. This problem can be sorted out by defining a statistical distance measure such as Mahalanobis distance, Fisher-Behrens distance etcetera. The statistical distance measures used by K-means algorithm do improve the results to some extent. In case of overlapping populations also the traditional clustering techniques do not produce good results. These problems have been taken into consideration for the research work and an attempt is made to improve the existing clustering techniques using the principles of learning as suggested by ANN.

Artificial Neural Network learning algorithms are also being used for clustering purposes. ANNs do not make any assumptions regarding the distribution of the population from which the sample units are drawn. In the present study, an ANN based statistical clustering technique is proposed. The proposed technique works similar to K-means algorithm with some modifications. The neighborhood concept of ANN is incorporated in the algorithm. Statistical distance measures are used instead of euclidean distance measure depending on the situation. The initial centers (weights) have to be chosen for initiating the algorithm which may be chosen in different ways. Generally, initial weights are chosen randomly. The proposed algorithm
is an iterative algorithm. A brief review of ANN based classification techniques is given in the next section.

2.2 REVIEW OF ANN BASED CLASSIFICATION TECHNIQUES

Recently, a large number of ANN's learning algorithms have been proposed for classification purposes. Pal, Bezdek and Tsao (1993) have proposed a generalization of Learning Vector Quantization (LVQ) for clustering. Desai and Chidananda Gowda (1993) have described two efficient training methods for self-organizing maps and a new procedure for clustering. Jain and Mao (1992), Mao and Jain (1993, 1995) and Kraajveld, Mao and Jain (1995) have done an elaborate work in the projection and extraction of multivariate data. They have proposed a number of networks and learning algorithms, which provide new tools for feature extraction and data projection. These networks include a network for Sammon's nonlinear projection (SAMANN), linear discriminant analysis (LDA) network, a nonlinear discriminant analysis (NDA) network and a network for nonlinear projection (NP-SOM) based on Kohonen's self-organizing map.

Gallinari, Thiria and Soulie (1988) and Gallinari, Thiria, Badran and Soulie (1991) have studied the relations between discriminant analysis and multilayer perceptrons used for classification purposes. Osman and Fahmy (1994) have expanded the available theoretical framework that establishes a link between discriminant analysis and adaptive feed-forward layered linear output networks used as mean-square classifiers. A description of ANN in relation to cluster analysis is discussed below.
2.3 **NOTATIONS AND TERMINOLOGIES**

The notations and terminologies used in the discussion are given below.

1. \( N \) = Number of sample Units.

2. \( m \) = Number of clusters (Nodes).

3. \( p \) = Number of Variables.

4. \( P_i \) = \( i^{th} \) multivariate normal population with mean vector \( \mu_i \) and variance-covariance matrix \( \Sigma_i \) for \( i=1,2,...,m \).

5. \( C_j \) = \( j^{th} \) cluster associated with the \( j^{th} \) node in the output layer.

6. \( X_i \) = \( i^{th} \) input sample vector where \( X_i = (x_{i1}, x_{i2}, ..., x_{ip})^T \).

7. \( W_j(t) \) = weight vector associated with \( j^{th} \) output node at \( t^{th} \) iteration.

8. \( D_{ij} \) = distance between \( i^{th} \) input sample unit and \( j^{th} \) output node.

9. \( \Sigma \) = common variance-covariance matrix.

10. \( \Sigma_j \) = variance-covariance matrix of the \( j^{th} \) population.

11. \( A = (\Sigma_1+\Sigma_2+...+\Sigma_m)/m \); \( \Sigma_j, j=1,2,...,m \) may be equal or unequal.

2.4 **ANN AND CLUSTER ANALYSIS**

The input layer consists of \( p \) input nodes through which the sample input vectors are presented. A second set of \( m \) nodes associated with \( m \) clusters is termed as output layer. The nodes in the input layer and the output layer are connected. Each connection has a weight, denoted by \( w_{ij} \) which is the strength of the connection from input node \( i \) to output node \( j \). This setup is termed as a single layer network in ANN (Figure 1.3).
In cluster analysis, the objective is to find the weights \( w_j \) so that the sample units are assigned to the nodes to which they already belong, in successive cycles or iterations based on some criterion. Generally, the initial weights are chosen at random or by some other method. Sample units are presented one after the other and are assigned to output nodes on the basis of a specified criterion. This is called a cycle. The weights are updated and again the cycle is repeated until either the weights converge or a specified number of cycles have been completed. Since each node \( j \) is associated with a weight vector \( W_j \) similar to the discriminant coefficient, they are used for classifying an input vector to a node which corresponds to a group or a cluster.

2.5 ANN BASED STATISTICAL CLUSTER ANALYSIS

A clustering algorithm which works perfectly on one type of data may completely fail on other types of data. In spite of the numerous research efforts, data clustering, as a general principle that will work on all situation remains a difficult and essentially an unsolved problem. If the clusters in a set are compact and isolated in the sense that the between-cluster variation is much larger than within-cluster variation, then any clustering method will be able to detect the clusters, irrespective of the cluster shape. Otherwise, choice of the distance measures can make substantial difference in the clustering results. It is found that most clusters in real data sets are not well isolated. There may be overlapping of clusters. Partitioning techniques with euclidean distance have an undesirable property of splitting large clusters under some circumstances. Sometimes it produces unusually large or small clusters. Euclidean distance based clustering algorithm favours clusters of equal size.
Otherwise, it gives drastic results. In such a situation, normalization of the data also cannot solve the problem. It is easy to find examples of data sets in real life applications which do not have well separated clusters. The well known Fisher-Iris data consisting of 150, four dimensional patterns from 3 classes (Iris setosa, Iris versicolor, Iris virginica) do not have well separated classes. It is found that the first species is well separated, while the other two species have overlapping features. So it may sometimes be split into only two well separated clusters using existing clustering techniques. The traditional K-means partitioning technique may also fail to produce well defined clusters in case of overlapping populations. In such cases an alternative approach is to be taken into account.

One way of solving this problem is to introduce statistical distance measure such as Mahalanobis distance in the clustering criterion. The statistical measure has variance-covariance term, which takes care of the variations within the cluster due to the variables considered. But sometimes variance-covariance matrix may become singular if the cluster size is less than the dimension of the data set. Mao and Jain (1996) have suggested regularized Mahalanobis distance to solve the singularity problem by adding a small value along the diagonal of variance-covariance matrix. K-means algorithm with statistical distance also does not give satisfactory results. Subsequently, one can go in for ANN techniques. ANN learning algorithms can be used as a tool for clustering objects.

The unsupervised learning of ANN have dealt with classification problems. Kohonen's self-organizing map (SOM) is often used to cluster the
input data set. This is explained in Chapter 1. In SOM, neighboring cells in the network topology compete in their activities by means of mutual lateral interactions and develop adaptively into specific detectors of different input values. The SOM has the special property of effectively creating spatially organized "internal representations of various features of input values and their abstractions". SOM is a two layered network that can organize a topological map from a random starting point. The resulting map shows the natural relationships among the patterns that are given to the network. The network combines an input layer with a competitive layer of processing units and is trained by unsupervised learning. SOM uses the topological neighborhood. Each node in the output layer has a set of nodes surrounding it, which is taken as neighbors. The nodes in the input layer are connected to the nodes in the output layer with a two dimensional weight vector. The weight values correspond to physical locations within the space occupied by the input sample units. Thus, the neighboring units in the physical space may occupy unrelated locations in weight space. During training, a two-dimensional point is selected at random to be the input vector. The weight vector of all units within the neighborhood change slightly toward the value of the input. As the training continues with different input points, the size of the neighborhood decreases gradually until it encompasses only a single unit. At the completion of the training, the weight vector for each unit will approximately be equal to the physical coordinates of the unit. In SOM, neighborhood is selected from a set of nodes surrounding the assigned node. This results in a large number of nodes in the output layer. The drawbacks of K-means algorithm and the self-organizing feature map algorithm have been
kept in mind and an attempt is made to resolve them by proposing ANN based clustering technique, by combining features of both the algorithms [Chandrasekaran and Kiruthika, 1997; Kiruthika and Chandrasekaran, 1997].

As our interest is in clustering of statistical observations, we propose an ANN based clustering algorithm that makes use of statistical distance measures, such as Mahalanobis distance and Fisher-Behrens distance, for calculating distances between input and output nodes such that variance-covariance matrix also participates in assignment of sample units to the nodes. The winning node is the node that has minimum distance with an input vector \( \mathbf{X} \). A neighborhood of a point \( \mathbf{X} \) is a set of nodes that are at a distance less than or equal to a prespecified value.

In the present discussion, neighborhood of a point \( \mathbf{X} \) in \( p \)-dimension is defined in terms of the maximum distance (MD) of the input vector \( \mathbf{X} \) with all output nodes. The neighborhood of \( \mathbf{X} \) is the set of nodes which fall inside the sphere whose radius declines with increasing number of iterations or cycles. The radius of neighborhood can be \( \text{MD}/2^t \) where \( t \) is the iteration, \( t = 1,2,\ldots \).

Neighborhood can also be defined in many other ways depending on the requirement. The logic behind using the above definition of neighborhood is as follows. All the sample units are fixed in the \( p \)-dimensional sample space. In order to cluster these points, certain number of seed points to represent clusters are initially assumed amongst the sample units. Since the sample points are fixed, it is possible only to move these reference points in the sample space. These reference points are the nodes which have associated with the weight vectors. In the conventional algorithms for clustering, only the
reference point which is closest to the input vector $X$ is updated or recalculated. But we propose to include even the other nodes which are in the neighborhood of $X$ to get updated but with a smaller learning coefficient $\beta$. A higher learning parameter $\alpha$ is used with the winning node, that is, the node that is closest to $X$ in the sample space. By using this concept, more than one reference point in the space get adjusted but with different accuracy, thereby giving opportunity for other reference points to get trained.

The proposed algorithm depends on initial weights for the nodes, the two gain terms $\alpha$ and $\beta$ values. A detailed discussion of the algorithm is given in the next section.

2.6 ALGORITHM FOR GROUPING STATISTICAL OBSERVATIONS

2.6.1 ALGORITHM

An algorithm for grouping statistical observations is described below. A single layer network is constructed for the proposed algorithm (Figure 1.3). The input sample units are presented one after the other through the input layer and the final clusters are obtained in the output layer.

**Step 1:** Choose the initial weights randomly.

**Step 2:** Present the input vector $X_i$ through the input layer.

**Step 3:** Compute the distances between the input vector and each of the $m$ output nodes using a specified metric depending on the nature of the observation.
Step 4: Find the winning node in the output layer with minimum distance. Suppose \( j^{th} \) output node is the winning node with minimum distance then \( X_i \) is assigned to the \( j^{th} \) node. Define a neighborhood around the \( X_i \). The neighborhood is defined as a sphere with reducing radius as the iteration continues. In other words, the classification criterion is

\[
X_i \in C_j \text{ if } D_{ij} < D_{ik} ; j, k = 1, 2, ..., m
\]

Step 5: Update the weights of node \( j \) and its neighbors using

\[
W_j(t+1) = W_j(t) + \alpha(t)*(X_i - W_j(t)) \quad \text{for the winning node } j
\]

and

\[
W_k(t+1) = W_k(t) + \beta(t)*(X_i - W_k(t)) \quad \text{for the node } k \text{ in the neighborhood of } X_i \text{ with } \alpha(t+1) = f(\alpha(t)), \beta(t+1) = f(\beta(t)) \quad \text{for successive iteration where } \alpha(t) \text{ and } \beta(t) \text{ are gain terms, that are non-increasing as the iteration continues. The range of values of } \alpha(t) \text{ and } \beta(t) \text{ are given by, } 0 < \alpha(t) < 1, 0 < \beta(t) < 1.
\]

Step 6: Repeat Steps 3 to 5 until the weight matrices corresponding to two successive iterations are identical with respect to a predefined accuracy.

Step 7: Repeat Step 2 to 6 till all the sample inputs are considered.

Step 8: Steps 2 to 7 are repeated until the same clusters are obtained in successive cycles or a specified number of cycles are completed.
2.6.2 CLASSIFICATION

Let us assume that there are \( m \) clusters \( C_1, C_2, \ldots, C_m \). An input vector \( X_i \) is classified as follows:

\[ X_i \in C_j \text{ if } D_{ij} < D_{ik}, \quad i=1,2,\ldots,N; \quad j,k = 1,2,\ldots,m; \quad j \neq k \]

where \( D_{ij} \) is as defined earlier.

The classification criterion suitable for different situations are given by

**Case 1:** All types of observations

\[ X_i \in C_j \text{ if } X_i^T (W_j(t) - W_k(t)) > (1/2) * (W_j(t)^T W_j(t) - W_k(t)^T W_k(t)) \]

**Case 2:** Populations with equal variance-covariance matrix

\[ X_i \in C_j \text{ if } X_i^T \Sigma^{-1} * (W_j(t) - W_k(t)) > (1/2) * (W_j(t) + W_k(t)) * \Sigma^{-1} * (W_j(t) - W_k(t)) \]

**Case 3:** Populations with unequal variance-covariance matrices

\[ X_i \in C_j \text{ if } X_i^T A^{-1} * (W_j(t) - W_k(t)) > (1/2) * (W_j(t) + W_k(t)) * A^{-1} * (W_j(t) - W_k(t)) \]

for \( j,k = 1,2,\ldots,m; \quad j \neq k \).

If \( \Sigma \) or \( A \) is known apriori, then they are used directly, otherwise their estimates are used. Without loss of generality, \( \Sigma \) (\( A \)) is taken to be the variance-covariance matrix of the given data set in the first cycle. \( \Sigma \) (\( A \)) is re-calculated for every cycle.
2.6.3 IMPLEMENTATION OF THE ALGORITHM

In the present study, the observations are assumed to have come from statistical population. The initial weights for the output nodes can be chosen in different ways, as follows.

(a) Generally, the weights are chosen randomly.

(b) In case of two groups, the weights associated with one node may be taken as the maximum value of each variable whereas for the other node as the vector of minimum values.

(c) The weights based on the mean and standard deviation of each variable in the sample input vectors can be taken for two population case.

(d) Weights are generated randomly between the maximum and minimum of each of the variables.

The initial weights play a major role in the formation of the clusters. In the next section, we have illustrated with examples the group formation with different initial weights. In case of two populations, the proposed algorithm is similar to K-means algorithm but not exactly the same, as the neighbors cannot be defined for the winning node. Different possibilities of obtaining clusters with different initial weights have been studied.
It is observed that while classifying statistical observations, statistical distance measures perform better than euclidean measure. As euclidean distance measure does not account for the co-variations within the group of variates, the distance measures are chosen according to the situation. The different measures are discussed here.

Case 1: Euclidean distance can be used for all types of observations. Euclidean distance between input vector $X_i$ and the output node $j$ is given by

$$D_{ij} = (X_i - W_j(t))^T (X_i - W_j(t))$$

Case 2: Mahalanobis distance is the most commonly used statistical distance measure. It is used in case of equal variance-covariance matrices for the populations under consideration. Mahalanobis distance between input vector $X_i$ and the output node $j$ is given by

$$D_{ij} = (X_i - W_j(t))^T \Sigma^{-1} (X_i - W_j(t))$$

$\Sigma$ is replaced by its estimate if not known.

Case 3: Fisher-Beherens distance is used for populations with unequal variance-covariance matrices. The distance between input vector $X_i$ and the output node $j$ is given by Fisher-Beherens distance

$$D_{ij} = (X_i - W_j(t))^T A^{-1} (X_i - W_j(t))$$

for $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, m$.

$\Sigma (A)$ is replaced by its estimate if not known, as discussed in Section 2.6.2.
The gain terms $\alpha(t)$ ranges from 0.3 to 0.9 and $\beta(t)$ from 0.01 to 0.2 in the process of updating weights during the training phase.

2.7 DATA SETS

The performance of the algorithm for cluster analysis has been evaluated using four data sets. The four data sets are briefly described below.

Data set A: The first data set relates to the Financial ratios of Bankrupt and Solvent companies [Altman, 1968; Morrison, 1990]. It consists of 66 companies with 5 types of financial ratios namely,

$X_1 = (working\ capital)/(total\ assets)$
$X_2 = (retained\ earnings)/(total\ assets)$
$X_3 = (earnings\ before\ interest\ and\ taxes)/(total\ assets)$
$X_4 = (market\ value\ equity)/(book\ value\ of\ total\ liabilities)$
$X_6 = (sales)/(total\ assets)$

Data set B: The second data set consists of Iris data [Anderson, 1935; Fisher, 1936 and Morrison, 1990]. The well known data set consists of 150, 4 dimensional patterns from three classes. The three classes are Iris setosa, Iris versicolor and Iris virginica. It contains four measurements on 50 flowers from each of the three species.
Data set C: This data set is generated from two bivariate normal population. Two hundred sample units are simulated. The mean vectors and the common variance-covariance matrix of the population are given by,

\[ \mu_1 = [10, 15]^T, \mu_2 = [15, 20]^T \text{ and} \]

\[ \Sigma = \begin{bmatrix} 150 & 90 \\ 90 & 200 \end{bmatrix} \]

Data set D: Hundred and fifty sample units are generated from three different four-variate normal population. The mean vectors are taken to be

\[ \mu_1 = [5, 6, 7, 9]^T \]

\[ \mu_2 = [13, 14, 15, 18]^T \]

\[ \mu_3 = [26, 27, 28, 26]^T \]

and the common variance-covariance matrix of the population is given by

\[ \Sigma = \begin{bmatrix} 16 & 14 & 12 & 11 \\ 14 & 28 & 13 & 15 \\ 12 & 13 & 27 & 16 \\ 11 & 15 & 16 & 19 \end{bmatrix} \]

These four data sets differ from one another in one or more characteristics, such as the data source (artificial/real data), dimensionality of the sample vector (p), number of sample units (N) and the number of classes (m) given in Table 2.1.
2.8 RESULTS AND DISCUSSION

The four data sets, described in the previous section, are presented as input data into the proposed algorithm. The distance measures are chosen according to the nature of the data set. The algorithm is implemented using PASCAL programming language.

To illustrate the working of the algorithm, we consider Data set A, namely, Finance Data set. However, the final results are summarised for all the four data sets.

Data set A consists of two populations (groups). The cluster formation of Data set A using three different initial weights are discussed. Various stages of the algorithm and the results are summarised below.

1. First we consider the case where the algorithm assumes random initial weights for the two output nodes associated with two clusters. The random initial weights assumed by the algorithm are

\[
\begin{bmatrix}
0.211 & 0.472 \\
0.129 & 0.893 \\
0.400 & 0.775 \\
0.196 & 0.452 \\
0.511 & 0.077
\end{bmatrix}
\]
The final clusters are obtained in twenty iterations.

The converged weight matrix is given by

\[
\begin{bmatrix}
-8.128 & 39.572 \\
-64.861 & 31.012 \\
-30.794 & 12.258 \\
35.811 & 242.500 \\
1.312 & 2.011
\end{bmatrix}
\]

It is observed that 5 sample units are misclassified. The confusion matrix is given below.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Actual Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>29</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>4</td>
</tr>
</tbody>
</table>

2. Next, the algorithm calculates the initial weights for the two clusters as the maximum and minimum of each variable. The initial weight matrix based on maximum and minimum of five variables is given by

\[
\begin{bmatrix}
-185.1 & 72.4 \\
-308.9 & 68.6 \\
-280.0 & 34.1 \\
0.7 & 771.7 \\
0.1 & 6.7
\end{bmatrix}
\]
The convergence occurred in two iterations. The converged weights are obtained as

\[
\begin{pmatrix}
-3.498 & 48.468 \\
-53.287 & 44.403 \\
-28.368 & 22.356 \\
31.595 & 308.045 \\
1.408 & 2.189
\end{pmatrix}
\]

and the confusion matrix is given by

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Actual Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>32</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>1</td>
</tr>
</tbody>
</table>

In this case, 9 sample units are misclassified.

3. The initial weights associated with the two output nodes are calculated based on the mean and standard deviation of each variable calculated as (mean - 2 * standard deviation, mean + 2 * standard deviation). The initial weights based on mean and standard deviation is

\[
\begin{pmatrix}
-23.155 & 58.493 \\
-84.251 & 56.990 \\
-51.712 & 35.248 \\
-36.128 & 330.847 \\
0.661 & 2.780
\end{pmatrix}
\]
and the converged weights obtained in ten iterations are

\[
\begin{bmatrix}
-7.107 & 41.588 \\
-62.467 & 34.127 \\
-31.948 & 15.509 \\
34.231 & 256.289 \\
1.359 & 2.032
\end{bmatrix}
\]

The confusion matrix is given by

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Actual Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>31</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>2</td>
</tr>
</tbody>
</table>

Three sample units are misclassified in this case.

The Finance data is also subjected to K-means algorithm independently and the following confusion matrix is obtained.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Actual Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Cluster 1</td>
<td>33</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>0</td>
</tr>
</tbody>
</table>

The above results show that the proposed technique performs better than K-means algorithm. The results also show that initial weights have some influence in the cluster formation.
The data sets are presented as input in the network. The gain terms \( u(t) \) is taken as 0.4 and \( P(t) \) is given the value 0.1. The initial weights are chosen randomly. The clustering process is carried out several times. The consistent results have been presented.

Table 2.2 to Table 2.5 give the confusion matrices for the best and worst cases for given data sets. It can be seen from Table 2.2 that for the best case the misclassification (in percentage) is 4.55, 8.67, 16 and 5.33 for Data sets A, B, C and D respectively. The misclassification percentage for worst cases are above 20 for the given data sets. The variation is due to choosing of different initial weights and the gain parameter \( \alpha \) and \( \beta \). Table 2.6 describes the average percentage of misclassification obtained by applying proposed algorithms and K-means algorithm to the four data sets.

It is also found that the two groups obtained by the cluster analysis in some cases start swapping after particular point of time thereby getting into an indefinite loop. To resolve this loop, it is mandatory to specify the maximum limit for the number of iterations or cycles.

Instead of working with one set of output nodes it is proposed to use two sets of nodes A and B with A as empty and B containing all the m output nodes [Chandrasekaran and Kiruthika, 1995]. As the algorithm starts working, if a sample input is assigned to a node in set B, then that node is moved to set A and a new node is added to set B. Finally, set A will consist of nodes assigned to the sample inputs and set B consists of nodes that have not been assigned to any input presented so far. B is taken to be a buffer set
of k nodes. The inputs are assigned to nodes according to a clustering criterion. The number of nodes in set A gives the number of clusters.

In the next chapter, the clustered groups are subjected to repetitive discriminant analysis. The discriminant analysis is carried out till same groups are obtained in two successive cycles.

2.9 SUMMARY

Artificial Neural Network based clustering technique is proposed in this chapter. The proposed technique makes use of self-organizing principle of ANN and K-means algorithm. Statistical distance measures are used in the algorithm. The concept of neighborhood of an input vector X is introduced and used in the training phase. In ANN, neighborhood is defined for the nodes. In the present study, the neighborhood of a point X is defined in terms of maximum distance of input vector X with the output nodes. Generally, in traditional clustering algorithms, the reference point (node) alone, that is closest to the input vector X, is recalculated. By using the neighborhood concept, more than one reference point in the space that fall within the neighborhood of X, get adjusted, thereby giving other reference points also to get trained. The initial weights can be chosen in different ways. The formation of clusters are dependent on initial weights and the two gain terms \( \alpha \) and \( \beta \). The performance of the algorithm was tested using real and simulated data sets. The results were compared with conventional clustering methods. It was found that the results obtained using proposed technique were good and encouraging.