CHAPTER 5

METHODOLOGY

This chapter describes the methodology of this study. The Hard and Fuzzy Based Algorithms, Bio-inspired Algorithms and Hybrid Algorithms as applied to data clustering problems are discussed.

5.1 Hard and Fuzzy Based Data Clustering Algorithms

Partitional methods are used to partition the large data sets into different clusters. Given n data objects, the methods partition \( k \leq n \) data clusters by using iterative procedure. Each cluster has at least one data object and each data object belongs to only one cluster. In these methods, the number of clusters must be given in advance. K-means and Fuzzy c-means are popular partitional methods. In hard or crisp data clustering method (K-means), a record in a data set belongs to only one cluster. K-means algorithm was proposed by MacQueen in 1967 [Mac67]. It is a centre based clustering method. In fuzzy data clustering method (FCM), a record in a data set belongs to more than one cluster with membership grades [Dun74] [Bez81].

5.1.1 K-means and Fuzzy c means (FCM) Algorithms using Different Distance Measures and Cluster Validity Indices

Euclidean distance metric is generally used in many traditional clustering algorithms. In this study, a comparative study of K-means and FCM algorithms is made.
using different distance measures such as Chebyshev and Chi-square. FCM algorithm based on $\sigma$-distance measure has also been developed. The new algorithms are tested on benchmark UCI repository data sets such as contraceptive method choice (CMC), diabetes, liver disorders and statlog (heart). The clustering results are also evaluated through the important cluster validity indices, partition coefficient (PC) and partition entropy (PE).

The Euclidean distance measure is defined in Chapter 4, Section 4.7 of equation (4.7). The distance between the data point $x_i$ and cluster centre $z_j$ is defined as

$$d(x_i, z_j) = \sqrt{\sum_{k=1}^{d} (x_{i,k} - z_{j,k})^2}, 1 \leq i \leq n; 1 \leq j \leq c$$

(5.1)

K-means algorithm based on Chebyshev distance measure

The aim of data clustering is to partition the data set $X = \{x_1, x_2, ..., x_n\}$ into a set of clusters satisfying the following conditions:

i. $z_j \neq \emptyset$, $j = 1, 2, ..., c$

ii. $z_i \cap z_j = \emptyset$, $i, j = 1, 2, ..., c, i \neq j$

iii. $\bigcup_{j=1}^{c} z_j = X$

Given a set of n data points and the number of clusters c, the objective is to select c cluster centres so as to minimize the mean squared distance. It generates the fast solution.
The Chebyshev distance measure is defined in Chapter 4, Section 4.7 of equation 4.7. The distance between the data point \( x_i \) and cluster centre \( z_j \) is defined as

\[
d(x_i, z_j) = \max_{i=1,2,...,n} |x_{i,k} - z_{j,k}|, j = 1, 2, ..., c, k = 1, 2, ..., d
\]

The K-means data clustering algorithm using Chebyshev distance measures is described in Figure 5.1.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set ( X = {x_1, x_2, ..., x_n} ), a set of data points; Select the number of clusters ( c ) (( 2 \leq c \leq n ));</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>( Cluster) centre ( z = {z_1, z_2, ..., z_c} ); ( Objective) function value</td>
</tr>
<tr>
<td></td>
<td>1. Initialize the randomly selected cluster centres from the data set</td>
</tr>
<tr>
<td></td>
<td>2. for i = 1 to max_it do</td>
</tr>
<tr>
<td></td>
<td>3. Calculate ( Chebyshev) distance measure using the equation (5.2)</td>
</tr>
<tr>
<td></td>
<td>4. Select the data point for a cluster with the minimal distances, they belong to that cluster</td>
</tr>
<tr>
<td></td>
<td>5. Compute the ( objective) function value using the equation</td>
</tr>
</tbody>
</table>
|       | \[
|       | \sum_{j=1}^{c} \sum_{x \in C_j} d^2(x, z_j) \] |
|       | 6. Calculate the \( cluster centroid\) using the equation |
|       | \[
|       | z_j = \frac{1}{n_j} \sum_{x \in C_j} x_p \] |
|       | \( where \ n_j \) is the number of data points in cluster \( j \) and \( C_j \) is the subset of data point that forms cluster \( j \) |

**Figure 5.1 K-means Data Clustering Algorithm**
**K-means algorithm based on Chi-square distance measure**

The Chi-square distance measure is defined in Chapter 4, Section 4.7 of equation (4.10). The distance between the data point $x_i$ and cluster centre $z_j$ is defined as

$$d(x_i, z_j) = \sqrt{\sum_{k=1}^{d} \frac{(x_{i,k} - z_{j,k})^2}{x_{i,k} + z_{j,k}}}$$  \hspace{1cm} (5.5)

The K-means algorithm based on Chi-square distance measure is also the same as given in Figure 5.1 except that the computation of distance formula in step 2 is replaced by Chi-square distance equation (5.5) instead of Chebyshev distance equation (5.2).

**K-means algorithm based on Euclidean distance measure**

The K-means algorithm based on Euclidean distance measure is also the same as given in Figure 5.1 except that the computation of distance formula in step 2 is replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).

**Fuzzy c-means algorithm using Chebyshev distance measure**

FCM algorithm is one of the most important fuzzy clustering algorithms. It permits one piece of data to belong to two or more clusters. Given a data set $X = \{x_1, x_2, \ldots, x_n\}$, the FCM algorithm partitions a data set into $c$ fuzzy clusters ($2 \leq c \leq n$) with $z = \{z_1, z_2, \ldots, z_c\}$ cluster centroids by minimizing the objective function value.

The parameters of FCM algorithm are summarized in Table 5.1.
Table 5.1 Parameters of FCM algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Data set</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of data points</td>
</tr>
<tr>
<td>$c$</td>
<td>Number of clusters</td>
</tr>
<tr>
<td>$z$</td>
<td>Cluster centroids</td>
</tr>
<tr>
<td>$u_{ij}$</td>
<td>Fuzzy membership values or degree of association of the $i$-th data point and $j$-th cluster or the degree to which data point $x_i$ belongs to cluster $z_j$</td>
</tr>
<tr>
<td>$m$</td>
<td>Fuzziness index or weighting exponent and $m \in [1, \infty)$</td>
</tr>
<tr>
<td>$d_{ij}$</td>
<td>Distance from the data point $x_i$ to the cluster centroid $z_j$</td>
</tr>
</tbody>
</table>

The fuzzy partition matrix $U = [u_{ij}]$ satisfies the following conditions:

- $u_{ij} \in [0,1], \forall i = 1,2,\ldots,n ; \forall j = 1,2,\ldots,c$ (5.6)
- $\sum_{j=1}^{c} u_{ij} = 1, \forall i = 1,2,\ldots,n$ (5.7)
- $0 < \sum_{i=1}^{n} u_{ij} < n, \forall j = 1,2,\ldots,c; 1 < n < \infty$ (5.8)

The cluster centres of FCM algorithm is computed by

$$z_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}, \ j = 1,2,\ldots,c$$ (5.9)

The objective function value (OFV) of FCM is calculated by
The fuzzy partition membership values are calculated by

\[ u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}^2}{d_{ik}^2} \right)^{\frac{1}{m-1}}} , 1 \leq i \leq n; 1 \leq j \leq c \] (5.11)

### Cluster Validity Indices

The cluster validity indices are applied to evaluate the data clustering results and test the quality of fuzzy partition [Bez74]. In this study, the popular measures such as partition coefficient (PC) and partition entropy (PE) are used to verify the cluster performance. These measures use fuzzy partition membership values.

#### Partition Coefficient (PC)

The partition coefficient (PC) is defined as

\[ PC = \frac{1}{n} \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^2 \] (5.12)

The best partition is obtained when the value of PC has a maximum value.

#### Partition Entropy (PE)

The partition entropy is defined as
The cluster structure is optimal when PE takes its minimum value.

The Fuzzy c-means data clustering algorithm using Chebyshev distance measure is summarized in Figure 5.2.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set $X = {x_1, x_2, \ldots, x_n}$, a set of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Cluster centres $z = {z_1, z_2, \ldots, z_c}$; OFV; PC and PE</td>
</tr>
<tr>
<td>1.</td>
<td>Select the number of clusters $c$ ($2 \leq c \leq n$); Choose fuzziness index $m$ ($m&gt;1$); Initialize fuzzy partition membership values $U^{(0)}$; Iteration error $\varepsilon \leftarrow 0.00001$; Fix the maximum number of iterations $\text{max}_it$; Set the iteration counter $t \leftarrow 0$.</td>
</tr>
<tr>
<td>2.</td>
<td>Calculate the cluster centres using the equation (5.9)</td>
</tr>
<tr>
<td>3.</td>
<td>Calculate the Chebyshev distance using the equation (5.2)</td>
</tr>
<tr>
<td>4.</td>
<td>Compute the OFV using the equation (5.10)</td>
</tr>
<tr>
<td>5.</td>
<td>Update the fuzzy partition membership values using the equation (5.11)</td>
</tr>
<tr>
<td>6.</td>
<td>Calculate PC and PE using the equations (5.12) and (5.13) respectively</td>
</tr>
<tr>
<td>7.</td>
<td>If $|U^{(t+1)} - U^{(t)}| &lt; \varepsilon$ or $t \leftarrow \text{max}_it$ then stop; otherwise set $t \leftarrow t+1$ and go to step (2).</td>
</tr>
</tbody>
</table>

**Figure 5.2 Fuzzy c-means Data Clustering Algorithm**

**Fuzzy c-means algorithm using Chi-square distance measure**

The FCM algorithm using Chi-square distance measure is also the same as given in Figure 5.2 except that the computation of distance formula in step 3 is replaced by Chi-square distance equation (5.5) instead of Chebyshev distance equation (5.2).
Fuzzy c-means algorithm using $\sigma$ - distance measure

The $\sigma$ - distance measure is defined in Chapter 4, Section 4.7 of equation (4.11). The distance between the data point $x_i$ and cluster centre $z_j$ is defined as

$$d(x_i, z_j) = \left\| x_i - z_j \right\| = \sqrt{\frac{\sum_{k=1}^{d} (x_{i,k} - z_{j,k})^2}{\sigma_j}}, \ j = 1, 2, \ldots, c$$

(5.14)

where $\sigma_j$ is the weighted mean distance in cluster $j$ and is given by

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} u_{ij}^m \left\| x_i - z_j \right\|^2}{\sum_{i=1}^{n} u_{ij}^m}}, 1 \leq j \leq c$$

(5.15)

The FCM algorithm using $\sigma$ - distance measure is also the same as given in Figure 5.2 except that the computation of distance formula in step 3 is replaced by $\sigma$ - distance equation (5.14) and (5.15) instead of Chebyshev distance equation (5.2).

Fuzzy c-means algorithm using Euclidean distance measure

The FCM algorithm using Euclidean distance measure is also the same as given in Figure 5.2 except that the computation of distance formula in step 3 is replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).
5.1.2 Possibilistic c-means (PCM) and Fuzzy Possibilistic c-means (FPCM) 

Algorithms for Data Clustering

Fuzzy c-means (FCM) algorithm is sensitive to noises and is easily struck at local minima. The possibilistic c-means (PCM) algorithm solves the problem of FCM algorithm. However, the performance of PCM algorithm depends heavily on the initialization and often deteriorates due to the coincident clustering problem. Fuzzy possibilistic c-means (FPCM) algorithm solves the problem of both FCM and PCM algorithms. Hence, in this section PCM and FPCM algorithms are studied. Five real world data sets are used to test these algorithms.

Possibilistic c-means (PCM) Data Clustering Algorithm

Krishnapuram and Keller [Kri93] proposed the PCM model, which relaxes the constraint (5.7) in FCM model. The fuzzy typicality matrix $T = [t_{ij}]$ satisfies the following conditions:

- $t_{ij} \in [0,1], \forall i = 1,2,\ldots,n; \forall j = 1,2,\ldots,c$ \hspace{1cm} (5.16)
- $0 < \sum_{i=1}^{n} t_{ij} < n, \forall j = 1,2,\ldots,c; 1 < n < \infty$ \hspace{1cm} (5.17)

The cluster centres of PCM algorithm is computed by

$$z_j = \frac{\sum_{i=1}^{n} t_{ij}^m x_i}{\sum_{i=1}^{n} t_{ij}^m}, \quad j = 1,2,\ldots,c$$ \hspace{1cm} (5.18)

The objective function value (OFV) of PCM is computed by
\[ J_{PCM} = \sum_{j=1}^{c} \sum_{i=1}^{n} t_{ij}^m \|d_{ij}\|^2 + \gamma_j \sum_{i=1}^{n} (1 - t_{ij})^m \]  

(5.19)

The fuzzy typicality values are calculated by

\[ t_{ij} = \frac{1}{1 + \left( \frac{d_{ij}}{\gamma_j} \right)^m} \]

(5.20)

where \( \gamma_j \) is the possibilistic typicality value of data point \( x_i \) belonging to the cluster \( z_j \) and is given by

\[ \gamma_j = \frac{\sum_{i=1}^{n} t_{ij}^m d_{ij}^2}{\sum_{i=1}^{n} t_{ij}^m} \]

(5.21)

The Possibilistic c-means data clustering algorithm is described in Figure 5.3.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set ( X = {x_1, x_2, \ldots, x_n} ), a set of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>( Cluster \ centres \ z = {z_1, z_2, \ldots, z_c} ); ( OFV )</td>
</tr>
<tr>
<td>1.</td>
<td>Select the number of clusters ( c ) ((2 \leq c \leq n)); Choose fuzziness index ( m ) ((m&gt;1)); ( \gamma &gt; 1 ); Initialize fuzzy typicality values ( T^{(0)} ); Iteration error ( \varepsilon ) ( \leftarrow 0.00001 ); Fix the maximum number of iterations ( max_it ); Set the iteration counter ( p \leftarrow 0 ).</td>
</tr>
<tr>
<td>2.</td>
<td>Calculate the ( cluster \ centres ) using the equation (5.18)</td>
</tr>
<tr>
<td>3.</td>
<td>Calculate the ( Euclidean \ distance ) using the equation (5.1)</td>
</tr>
<tr>
<td>4.</td>
<td>Compute the ( OFV ) using the equation (5.19)</td>
</tr>
<tr>
<td>5.</td>
<td>Update the ( fuzzy \ typicality \ values ) using the equation (5.20) and (5.21)</td>
</tr>
<tr>
<td>6.</td>
<td>If (</td>
</tr>
</tbody>
</table>

**Figure 5.3 Possibilistic c-means Data Clustering Algorithm**

57
Fuzzy Possibilistic c-means (FPCM) Data Clustering Algorithm

Fuzzy Possibilistic c-means algorithm proposed by Pal et al. [Pal97] combines the characteristics of both fuzzy and possibilistic c-means algorithms. Membership and typicality values are significant for the accurate characteristics of data structure in data clustering problem. FPCM algorithm normalizes the possibilistic values. The sum of possibilities of all the data objects in a cluster is 1. This algorithm eliminates the problems of FCM and PCM algorithms.

The constraints of FPCM model are given below:

- \[0 \leq u_{ij}, t_{ij} \leq 1, 1 \leq i \leq n; 1 \leq j \leq c\] (5.22)
- \[\sum_{j=1}^{c} u_{ij} = 1, \forall i = 1, 2, \ldots,n\] (5.23)
- \[\sum_{i=1}^{n} t_{ij} = 1, \forall j = 1, 2, \ldots,c\] (5.24)

The cluster centres of FPCM algorithm is computed by

\[z_j = \frac{\sum_{i=1}^{n} (u_{ij}^m + t_{ij}^n)x_i}{\sum_{i=1}^{n} (u_{ij}^m + t_{ij}^n)}, 1 \leq j \leq c\] (5.25)

The objective function value (OFV) of FPCM algorithm depends on both membership and typicality values. It is computed by

\[J_{FPCM} = \sum_{j=1}^{c} \sum_{i=1}^{n} (u_{ij}^m + t_{ij}^n) d_{ij}^2\] (5.26)
The fuzzy partition membership values are calculated by

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}^2}{d_{ik}^2} \right)^{m-1}}, 1 \leq i \leq n; 1 \leq j \leq c$$

(5.27)

The fuzzy typicality values are calculated by

$$t_{ij} = \frac{1}{\sum_{k=1}^{n} \left( \frac{d_{ij}^2}{d_{ik}^2} \right)^{\eta-1}}, 1 \leq i \leq n; 1 \leq j \leq c$$

(5.28)

The Fuzzy possibilistic c-means data clustering algorithm is described in Figure 5.4.

**Input**
Data set $X = \{x_1, x_2, \ldots, x_n\}$, a set of data points

**Output**
Cluster centres $z = \{z_1, z_2, \ldots, z_c\}$; OFV

1. Select the number of clusters $c$ ($2 \leq c \leq n$); Choose fuzziness index $m$ ($m>1$); scale parameter $\eta > 1$; Initialize fuzzy partition membership values $U^{(0)}$; Initialize fuzzy typicality values $T^{(0)}$; Iteration error $\varepsilon \leftarrow 0.00001$; Fix the maximum number of iterations $max_it$; Set the iteration counter $p \leftarrow 0$.

2. Calculate the cluster centres using the equation (5.25)

3. Calculate the Euclidean distance using the equation (5.1)

4. Compute the OFV using the equation (5.26)

5. Update the fuzzy partition membership values and fuzzy typicality values using the equation (5.27) and (5.28) respectively.

6. If $\|U^{(p+1)} - U^{(p)}\| < \varepsilon$ or $p \leftarrow max_it$ then stop; otherwise set $p \leftarrow p+1$ and go to step (2).

**Figure 5.4** Fuzzy Possibilistic c-means Data Clustering Algorithm
5.2 Bio-inspired Algorithm to Data Clustering Using Different Distance Metrics

The popular data clustering algorithms, K-means and FCM have some limitations such as easily struck at local minima and random selection of initial centre values. Recently bio-inspired algorithm namely particle swarm optimization (PSO) is successfully applied to solve data clustering problems. It is a population based global optimization technique. Most of the research contributions are based on Euclidean distance metric. In this study, a distance based approach to PSO algorithm is developed for data clustering problem. Manhattan and Chebyshev distance metrics are applied in PSO-based data clustering algorithm for four real-life benchmark medical data sets and an artificially generated data set.

PSO-Based Data Clustering Algorithm Using Chebyshev Distance Metric

A single particle represents the \( N_c \) cluster centroids. Each particle \( x_i \) in PSO system is constructed as \( x_i = (m_{i1}, m_{i2}, ..., m_{ij}, ..., m_{iN_c}) \) where \( m_{ij} \) denotes the j-th cluster centre vector of the i-th particle and \( N_c \) represents the number of clusters. A swarm is a collection of particles. Hence, a swarm represents a number of candidate clustering solutions for a data set.

The quality of PSO algorithm is measured according to the following criteria:

- **Quantization error (QE) or Fitness value:** It can be used to express the quality measure of PSO-based clustering algorithm. It is defined as

\[
J_e = \frac{\sum_{j=1}^{N_c} \left[ \sum_{y_{zp} \in C_j} d(z_{zp}, m_j) / n_j \right]}{N_c}
\]

(5.29)
where $z_p$ is the p-th data vector, $m_j$ is the j-th centre vector, $d(z_p, m_j)$ is the distance between $z_p$ and $m_j$, $C_j$ is the j-th cluster, $n_j$ is the number of data vectors in $C_j$.

- **Maximum average distance (MAD):** It is defined as

$$
\max \left\{ \frac{\sum_{z \in C_j} d(z_p, m_j)}{n_j} \right\}
$$

(5.30)

The PSO-Based data clustering algorithm is described in Figure 5.5.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set $X = {x_1, x_2, ..., x_n}$, a set of data vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td><em>Cluster centres</em> $z = {z_1, z_2, ..., z_c}$; <em>Fitness value</em>; <em>MAD</em></td>
</tr>
<tr>
<td>1. Select the number of clusters $c$ ($2 \leq c \leq n$); Number of particles; Fix the maximum number of iterations $max_it$; Initialize each particle to contain $N_c$ randomly selected cluster centroids</td>
<td></td>
</tr>
<tr>
<td>2. for $t = 1$ to $max_it$</td>
<td></td>
</tr>
<tr>
<td>a) for each particle $i$</td>
<td></td>
</tr>
<tr>
<td>i) for each data vector $z_p$</td>
<td></td>
</tr>
<tr>
<td>- Calculate the <em>Chebyshev distance</em> $d(z_p, m_j)$ using the equation (5.2).</td>
<td></td>
</tr>
<tr>
<td>- Assign data vector $z_p$ to cluster $C_j$ such that $d(z_p, m_j) = \min_{j=1,2,...,N_c} {d(z_p, m_j)}$.</td>
<td></td>
</tr>
<tr>
<td>ii) Calculate the <em>fitness value</em> and <em>MAD</em> using the equation (5.29) and (5.30).</td>
<td></td>
</tr>
<tr>
<td>b) Determine the <em>personal best position</em> for each particle and <em>global best solution</em> for the swarm using the equations (2.1) and (2.2) respectively.</td>
<td></td>
</tr>
<tr>
<td>3. Update the cluster centroids using the equations (2.3) and (2.4).</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.5 PSO-Based Data Clustering Algorithm**

61
PSO-Based Data Clustering Algorithm Using Manhattan Distance Metric

The Manhattan distance measure is defined in Chapter 4, Section 4.7 of equation (4.8).

The distance between the data point $x_i$ and cluster centre $z_j$ is defined as

$$d(x_i, z_j) = \sum_{k=1}^{d} |x_{i,k} - z_{j,k}|, 1 \leq i \leq n; 1 \leq j \leq c$$  \hspace{1cm} (5.32)

The PSO-based data clustering algorithm using Manhattan distance metric is also the same as given in Figure 5.5 except that the computation of distance formula in step 2 is replaced by Manhattan distance equation (5.32) instead of Chebyshev distance equation (5.2).

PSO-Based Data Clustering Algorithm Using Euclidean Distance Metric

The PSO-based data clustering algorithm using Euclidean distance metric is also same as given in Figure 5.5 except that the computation of distance formula in step 2 is replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).

5.3 Hybrid Data Clustering Algorithms

Generally individual algorithms may have some drawbacks. Two or more algorithms are integrated to form hybrid algorithm. Recently, hybrid algorithms are found to be more popular to solve many real world problems.
5.3.1 Hybridization of K-means and Penalized Fuzzy c-means (K-PFCM) Algorithm Using Different Distance Metrics and Cluster Validity Measures

In K-means algorithm, a data vector belongs to only one cluster. In fuzzy c-means algorithm, a data vector belongs to more than one cluster with varying degrees of membership values. It is suitable for many real world data sets. There are many derivatives of FCM algorithm. Penalized fuzzy c-means (PFCM) is one of the derivatives of FCM algorithm. It is an effective and robust data clustering method. It is affected by random selection of initial centroid values. Euclidean distance measure is also usually applied in many clustering algorithms. The problems can be resolved by developing hard-fuzzy data clustering algorithm, namely Hybrid K-PFCM with different distance measures. The final centroid values of K-means algorithm is given as input to the PFCM algorithm. The hybrid algorithm performs better than K-means and FCM algorithms. The data clustering results are also evaluated through standard cluster validity measures such as intra-cluster distance, partition coefficient (PC), partition entropy (PE), Maxdis and Mindis. The hybrid algorithm is effective under these criteria. Five benchmark data sets from UCI repository are taken for the experimentation.

Hybrid K-PFCM Data Clustering Algorithm Using Chessboard Distance Measure

Yang [Yan93] proposed the idea of adding penalty term to the FCM objective function. An extended version of FCM algorithm is called Penalized FCM (PFCM) which combines classification maximum likelihood (CML) procedure and penalty idea. The aim is to develop K-means based penalized fuzzy c-means algorithm called, hybrid K-PFCM.
The proposed hybrid algorithm gives more improved result than K-means and FCM algorithms.

The important components of PFCM algorithm are described below:

The proportion $\alpha$ is calculated by

$$\alpha_j = \frac{\sum_{i=1}^{n} u_{ij}^m}{c \sum_{j=1}^{n} \sum_{i=1}^{m} u_{ij}^m} \tag{5.33}$$

The cluster centres of PFCM algorithm is computed by

$$a_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}, 1 \leq j \leq c \tag{5.34}$$

The OFV is defined as

$$J_{PFCM} = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m d_{ij}^2 - \omega \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m \ln \alpha_j \tag{5.35}$$

where, $\omega \geq 0, \alpha_j \geq 0, \forall j$ and $\sum_{j=1}^{c} \alpha_j = 1 \tag{5.36}$

The fuzzy c-partition is given by

$$u_{ij} = \left( \frac{\sum_{k=1}^{c} (d_{ij}^2 - \omega \ln \alpha_j)^{(m-1)/2}}{\sum_{k=1}^{c} (d_{ik}^2 - \omega \ln \alpha_k)^{(m-1)/2}} \right)^{-1}, 1 \leq i \leq n; 1 \leq j \leq c \tag{5.37}$$
Cluster Validity Measures

The data clustering results are evaluated through the following validity measures:

- **Intra-cluster distance (ICD):** It is defined as the average of the sum of all distances between the data points within a cluster and the centroid of the cluster. It is defined as

  \[
  ICD = \frac{1}{n} \sum_{j=1}^{K} \sum_{x \in C_j} \| x - a_j \|^2
  \]  
  \[ (5.38) \]

  where \( a_j \) denotes the centroid of \( j \)-th cluster

- **PC and PE:** These measures are defined in equations (5.12) and (5.13)

- **Maxdis and Mindis:** The Maxdis represents the maximum distance between the cluster centres and Mindis means the minimum distance between the cluster centres.

  \[
  Maxdis = \max_{i \neq j} \{ d(a_i, a_j) \}, 1 \leq i, j \leq c
  \]  
  \[ (5.39) \]

  \[
  Mindis = \min_{i \neq j} \{ d(a_i, a_j) \}, 1 \leq i, j \leq c
  \]  
  \[ (5.40) \]

The Hybrid K-PFCM data clustering algorithm is described in Figure 5.6.

**Hybrid K-PFCM Data Clustering Algorithm Using City Block Distance Measure**

The hybrid K-PFCM data clustering algorithm using City block distance measure is also the same as given in Figure 5.6 except that the computation of distance formula in
step 1 (ii), step 2 and step 6 are replaced by City block distance equation (5.32) instead of Chebyshev distance equation (5.2).

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set $X = {x_1, x_2, \ldots, x_n}$, a set of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>$Cluster centres \ z = {z_1, z_2, \ldots, z_c}$; $OFV$; $Intra-cluster$; $PC$; $PE$; $Maxdis$ and $Mindis$</td>
</tr>
</tbody>
</table>

1. Select the number of clusters $c$ ($2 \leq c \leq n$); Initialize the randomly selected cluster centres from the data set; Fix the maximum number of iterations $max_it$;
   i) $for$ $i = 1$ to $max_it$ $do$
   ii) Calculate $Chebyshev distance measure$ using the equation (5.2)
   iii) Select the data point for a cluster with the minimal distances, they belong to that cluster
   iv) Compute the $objective function value$ using the equation
   $$\sum_{j=1}^{c} \sum_{x \in C_j} d^2(x, a_j)$$ (5.41)
   v) Calculate the $cluster centroid$ using the equation
   $$a_j = \frac{1}{n_j} \sum_{x \in C_j} x$$ (5.42)
   where $n_j$ is the number of data points in cluster $j$ and $C_j$ is the subset of data point that form cluster $j$
   vi) Record the final cluster centres

2. Calculate $Chebyshev distance measure$ using final centres from step (1) and obtain the membership values using the equation
   $$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}}{d_{ik}} \right)^2}, 1 \leq i \leq n; 1 \leq j \leq c$$ (5.43)

contd.
Figure 5.6 Hybrid K-PFCM Data Clustering Algorithm

Hybrid K-PFCM Data Clustering Algorithm Using Euclidean Distance Measure

The hybrid K-PFCM data clustering algorithm using Euclidean distance measure is also the same as given in Figure 5.6 except that the computation of distance formula in step 1 (ii), step 2 and step 6 are replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).

5.3.2 Distance Based Hybrid Approach for Cluster Analysis Using Variants of K-Means and Evolutionary Algorithm

Distance based hybrid approach for cluster analysis using variants of K-Means algorithm such as K-Medoids and K-Means++ with PSO technique is developed. The algorithms are tested on four popular benchmark data sets from UCI machine learning repository and an artificial data set using Euclidean, City block and Chebyshev distance
metrics. The data clustering results are evaluated through the fitness function value. A comparative study of hybrid K++_PSO algorithm with other algorithms is also performed.

K-Medoids algorithm [Han01] [Kau90], one of the variants of K-Means algorithm, uses the most representative data points called medoids instead of centroids. Both K-means and K-Medoids are widely used partitional data clustering algorithms. However, they are easily struck at local optimal solution and are sensitive to random selection of initial centres. The number of clusters also must be known in advance. K-means++ [Art07], another variant of K-Means algorithm, uses a new technique of selecting initial centroids by random initial centers with specific probabilities. The new seeding method has better performance and convergence rate than other algorithms. PSO technique is global evolutionary technique. In order to improve the data clustering results, K-Means algorithm is integrated with PSO technique, called hybrid K_PSO algorithm. Euclidean distance metric is traditionally used in all these algorithms. In this study, a new hybrid algorithm, hybrid K++_PSO algorithm is developed using Chebyshev and City block distance metrics. The data clustering results are evaluated through the following fitness function value or intra-cluster distance. The proposed algorithm reports better result than other algorithms, K-means, K-Medoids, K-Means++, PSO, K_PSO and K-Med_PSO.

**Fitness Function Value**

The distance between each data vector and within a cluster and the cluster centre of that cluster is computed and added up. It is computed by using
\[
\sum_{j=1}^{N_j} \sum_{\forall p \in C_j} d(z_p, m_j)
\]  \hspace{1cm} (5.44)

where \( d(z_p, m_j) \) is the distance between the data object \( z_p \) and the centroid \( m_j \).

**K-Medoids Data Clustering Algorithm Using Chebyshev Distance Metric**

In this algorithm, the centres are located among the data points themselves. A medoid is defined as the data point of a cluster, whose mean dissimilarity to all the data points in the cluster is minimum.

The K-Medoids data clustering algorithm is given in Figure 5.7

**K-Medoids Data Clustering Algorithm Using City Block Distance Metric**

The K-Medoids data clustering algorithm using City block distance measure is also the same as given in Figure 5.7 except that the computation of distance formula in step 3 is replaced by City block distance equation (5.32) instead of Chebyshev distance equation (5.2).

**K-Medoids Data Clustering Algorithm Using Euclidean Distance Metric**

The K-Medoids data clustering algorithm using City block distance measure is also the same as given in Figure 5.7 except that the computation of distance formula in step 3 is replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).
Input
Data set \( X = \{x_1, x_2, \ldots, x_n\} \), a set of data points; Select the number of clusters \( c \) \((2 \leq c \leq n)\);

Output
Cluster centres \( z = \{z_1, z_2, \ldots, z_c\} \); Fitness function value

1. Initialize the randomly selected cluster centres from the data set

2. Assign each data object to the cluster associated with the closest cluster centres.

3. Recalculate the positions: finding the data object \( i \) within the cluster that minimizes

\[
\sum_{x_i \in C_j} d(x_i, z_j)
\]  \hspace{1cm} (5.45)

where \( C_j \) is the cluster containing the data object \( x_i \) and \( d(x_i, z_j) \) is the distance between the data object \( x_i \) and the centroid \( z_j \).

4. Repeat step 2 and step 3 until converges.

---

**Figure 5.7 K-Medoids Data Clustering Algorithm**

**Hybrid K++_PSO Data Clustering Algorithm Using Chebyshev Distance Metric**

The K-Means++ algorithm is combined with PSO technique, called hybrid K++_PSO. The description of this algorithm is summarized in Figure 5.8.
| **Input** | Data set $X = \{x_1, x_2, \ldots, x_n\}$, a set of data points; Select the number of clusters $c$ ($2 \leq c \leq n$); Fix the maximum number of iterations $\text{max\_it}$; Number of particles. |
| **Output** | Cluster centres $z = \{z_1, z_2, \ldots, z_c\}$; Fitness function value |

1. Select an initial centre uniformly at random from the data set $X$
2. while $|z| < c$ do
   - Choose the next centre randomly from $X$, where every $x \in X$ has a probability of
     \[
     d(x, z) = \frac{\sum_{x \in X} \min_{j=1,2,\ldots,c} ||x_j - z_j||}{\sum_{j \in \mathbb{Z}} \min_{j=1,2,\ldots,c} ||x_j - z_j||}
     \] of being selected
   end while
3. i) for $i = 1$ to $\text{max\_it}$ do
   ii) Calculate Chebyshev distance measure using the equation (5.2) of each object in the data set from each of the cluster centroids of step 2
   iii) Select the data point for a cluster with the minimal distances to which they belong.
   iv) Calculate the new cluster centre using
     \[
     z_j = \frac{\sum_{k=1}^{n_j} x_k}{n_j}
     \] (5.47)
     where $n_j$ represents the number of data points in the $i$-th cluster
   v) Interchange the new cluster centres to old cluster centres.
   \[contd.\]
4. The final cluster centres of step 3 is to be taken as the initial cluster centre for particle 1 and \( N_c \) randomly selected cluster centroids for the remaining particles.

5. \( for \ t = 1 \ to \ max_it \)
   a) \( for \ each \ particle \ i \)
      i) \( for \ each \ data \ vector \ z_p \)
         - Calculate the Chebyshev distance \( d(z_p, m_q) \) using the equation (5.2).
         - Assign data vector \( z_p \) to cluster \( C_q \) such that
           \[
           d(z_p, m_q) = \min_{j=1,2,...,N_c} \{d(z_p, m_j)\}.
           \]
      ii) Calculate the fitness value using the equation (5.44).
   b) Determine the personal best position for each particle and global best solution for the swarm using the equations (2.1) and (2.2) respectively.

6. Update the cluster centroids using the equations (2.3) and 2.4).

---

**Figure 5.8 Hybrid K++_PSO Data Clustering Algorithm**

**Hybrid K++_PSO Data Clustering Algorithm Using City Block Distance Metric**

The Hybrid K++_PSO data clustering algorithm using City block distance measure is also the same as given in Figure 5.8 except that the computation of distance formula is replaced by City block distance equation (5.32) instead of Chebyshev distance equation (5.2).
Hybrid K++_PSO Data Clustering Algorithm Using Euclidean Distance Metric

The Hybrid K++_PSO data clustering algorithm using City block distance measure is also the same as given in Figure 5.8 except that the computation of distance formula is replaced by Euclidean distance equation (5.1) instead of Chebyshev distance equation (5.2).

5.3.3 Hybrid Fuzzy Based Nature-inspired Clustering Algorithms with Validity Measures

Fuzzy c-means (FCM) algorithm is one of the most widely used data clustering methods. However, it suffers from some limitations like easily struck at local minima and sensitive to noise and outlier. Fuzzy possibilistic c-means (FPCM) algorithm is one of the good methods for noisy environment. Partition index maximization (PIM) is one of the extensions of FCM algorithm by adding partition coefficient (PC) into FCM objective function. Nature-inspired algorithm such as particle swarm optimization (PSO) is a global optimization technique. It overcomes the problem of local optima. The performance of PSO algorithm can be further improved with the help of fuzzy clustering algorithms. Most of the traditional clustering algorithms are based on Euclidean distance measure. In this study, two new hybrid algorithms namely fuzzy possibilistic c-means based particle swarm optimization algorithm (FPCM-PSO) and fuzzy c-means based particle swarm optimization algorithm with PIM (FUZZY-PSO-PIM) are developed using different measures including Euclidean, Manhattan and Chessboard distance. The hybrid FUZZY-PSO-PIM algorithm is efficient and report more encouraging results than other clustering techniques for all distance measures for biomedical data sets and an artificial data set.
The clustering results are evaluated with respect to many cluster validity measures. Hybrid algorithm based on Chessboard distance measure produces better results than the other distance measures.

Cluster Validity Measures

The performance of data clustering algorithms is evaluated through some of the popular validity measures [Bez74] [Ram98]. A summary of selected cluster validity measures is given in Table 5.2.

<table>
<thead>
<tr>
<th>Validity Measure</th>
<th>Functional Description</th>
<th>Optimal Partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra Distance</td>
<td>$\sum_{j=1}^{K} \sum_{x \in C_j}</td>
<td></td>
</tr>
<tr>
<td>Inter Distance</td>
<td>$\min_{i \neq j}</td>
<td></td>
</tr>
<tr>
<td>Xie-Beni Index</td>
<td>$XB = \frac{\sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^2 d_{ij}^2}{n \times (\min_{i \neq j}</td>
<td></td>
</tr>
<tr>
<td>Partition Coefficient</td>
<td>$PC = \frac{1}{n} \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^2$</td>
<td>Maximum</td>
</tr>
<tr>
<td>Partition Entropy</td>
<td>$PE = -\frac{1}{n} \left{ \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij} \log u_{ij} \right}$</td>
<td>Minimum</td>
</tr>
<tr>
<td>Fukuyama-Sugeno Index</td>
<td>$FS_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m (</td>
<td></td>
</tr>
</tbody>
</table>
Hybrid FPCM-PSO Data Clustering Algorithm Using Chessboard Distance Measure

Fuzzy possibilistic c-means (FPCM) algorithm is combined with particle swarm optimization (PSO), called Hybrid FPCM-PSO. The Hybrid FPCM-PSO data clustering algorithm using Chessboard distance measure is described in Figure 5.9.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set $X = {x_1, x_2, \ldots, x_n}$, a set of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Cluster centres $z = {z_1, z_2, \ldots, z_c}$; fitness value; Value of Cluster Validity</td>
</tr>
</tbody>
</table>

1. *Initialize parameters of FPCM and PSO:* Select the number of clusters $c$ ($2 \leq c \leq n$); Choose fuzziness index $m$ ($m > 1$); Number of particles; $c_1, c_2, \omega$; Iteration error $\varepsilon \leftarrow 0.00001$; Fix the maximum number of iterations $\text{max}_\text{it}$
2. Initialize position and velocity values, $pbest$ for each particle and $gbest$ for the swarm.
3. Read the data set, Initialize fuzzy partition membership values $U^{(0)}$ and fuzzy typicality values $T^{(0)}$ for the given number of particles.
4. Calculate the *cluster centres* for each particle using the equation (5.25).
5. Calculate the *Chebyshev distance* for each particle using the equation (5.3).
6. Update the *fuzzy partition membership values* and *fuzzy typicality values* using the equation (5.27) and (5.28) respectively.
7. Compute the *fitness value* of each particle using the equation (5.26).
8. Update the *personal best position* for each particle and *global best solution* for the swarm using the equations (2.1) and (2.2) respectively.
9. Compute the values of *various cluster validity*.
10. Update the *cluster centroids* using the equations (2.3) and 2.4).
11. Check the max_it if not go to step 5.
12. Record the *final fitness value* and *cluster validity values*.

**Figure 5.9 Hybrid FPCM-PSO Data Clustering Algorithm**
Hybrid FPCM-PSO Data Clustering Algorithm Using Manhattan Distance Measure

The Hybrid FPCM-PSO data clustering algorithm using Manhattan distance measure is also the same as given in Figure 5.9 except that the computation of distance formula is replaced by Manhattan distance equation (5.32) instead of Chessboard distance equation (5.2).

Hybrid FPCM-PSO Data Clustering Algorithm Using Euclidean Distance Measure

The Hybrid FPCM-PSO data clustering algorithm using Euclidean distance measure is also the same as given in Figure 5.9 except that the computation of distance formula is replaced by Euclidean distance equation (5.1) instead of Chessboard distance equation (5.2).

Hybrid FUZZY-PSO-PIM Data Clustering Algorithm Using Chessboard Distance Measure

In partition index maximization (PIM) algorithm [Ozd08], partition coefficient (PC) is added into the objective function of FCM algorithm. The objective function value (OFV) of PIM algorithm is

\[
J_{\text{PIM}} = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m d_{ij}^2 - \beta \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m
\]  

(5.48)

where \( \beta \) is defined as

\[
\delta \min \{d^2(z_j, (z_j + z_{j'})/2); j \neq j'; 0 < \delta < 1 \}
\]  

(5.49)

The fuzzy partition membership values are calculated by
\[ u_{ij} = \frac{1}{\sum_{k=1}^{c}(\frac{d_{ij}^2 - \beta}{d_{ik}^2 - \beta})^{m-1}}, 1 \leq i \leq n; 1 \leq j \leq c \]  \hspace{1cm} (5.50)

In this study, fuzzy particle swarm optimization (FUZZY-PSO) algorithm is integrated with partition index maximization (PIM) algorithm, called Hybrid FUZZY-PSO-PIM. The Hybrid FUZZY-PSO-PIM data clustering algorithm using Chessboard distance measure is summarized in Figure 5.10.

<table>
<thead>
<tr>
<th>Input</th>
<th>Data set ( X = {x_1, x_2, \ldots, x_n} ), a set of data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Cluster centres ( z = {z_1, z_2, \ldots, z_c} ); fitness value; Value of Cluster Validity</td>
</tr>
</tbody>
</table>

1. Initialize parameters of FUZZY-PSO: Select the number of clusters \( c \) \((2 \leq c \leq n)\); Choose fuzziness index \( m \) \((m>1)\); Number of particles; \( c_1, c_2, \omega \); Iteration error \( \varepsilon \leftarrow 0.00001 \); Fix the maximum number of iterations \( \text{max}_\text{it} \).
2. Initialize position and velocity values, \( p_{\text{best}} \) for each particle and \( g_{\text{best}} \) for the swarm.
3. Read the data set, Initialize fuzzy partition membership values \( U^{(0)} \) for the given number of particles.
4. Calculate the cluster centres for each particle using the equation (5.9).
5. Calculate the Chebyshev distance for each particle using the equation (5.3).
6. Update the fuzzy partition membership values using the equation (5.11).
7. Compute the fitness value of each particle using the equation (5.10).
8. Update the personal best position for each particle and global best solution for the swarm using the equations (2.1) and (2.2) respectively.
9. Compute the values of various cluster validity.
10. Update the cluster centroids using the equations (2.3) and 2.4).

... contd.
The Hybrid FUZZY-PSO-PIM data clustering algorithm using Manhattan distance measure is also the same as given in Figure 5.10 except that the computation of distance formula is replaced by Manhattan distance equation (5.32) instead of Chessboard distance equation (5.2).
Hybrid FUZZY-PSO-PIM Data Clustering Algorithm Using Euclidean Distance Measure

The Hybrid FUZZY-PSO-PIM data clustering algorithm using Euclidean distance measure is also the same as given in Figure 5.10 except that the computation of distance formula is replaced by Euclidean distance equation (5.1) instead of Chessboard distance equation (5.2).

5.3.4 Data Clustering Based on Hybrid of Fuzzy and Swarm Intelligence Algorithm Using Euclidean and Non-Euclidean Distance Metrics

Constriction factor particle swarm optimization (cfPSO), another variant of PSO technique is also a population based global optimization technique for solving data clustering problem. Euclidean distance is a well known and more commonly used metric in many clustering algorithms. Some drawbacks of this distance metric include sensitive to noise and outliers and inability to account for elliptic decision boundaries. A fuzzy based constriction factor PSO with FCM (FUZZY-cfPSO-FCM) algorithm is developed using Non-Euclidean distance metrics such as Kernel, Mahalanobis and New distance on several benchmark UCI machine learning repository data sets. The proposed hybrid algorithm makes use of the advantages of FCM and cfPSO algorithms. The clustering results are also evaluated through fitness value, accuracy rate and failure rate. The proposed hybrid algorithm achieves better result on various data sets.

In this study, fuzzy based constriction factor PSO (FUZZY-cfPSO) algorithm is integrated with fuzzy c-means (FCM), called Hybrid FUZZY-cfPSO-FCM. The Non-
Euclidean distance metrics such as Kernel, Mahalanobis and New distance are used. The performance of the algorithms is evaluated through fitness value, accuracy rate and error rate.

The fitness value of cfPSO algorithm is calculated by (5.44). The fitness value of fuzzy and hybrid algorithms is obtained by (5.10). The Huang’s accuracy measure [Hua99] is determined by

\[
 r = \frac{\sum_{i=1}^{k} n_i}{n} \quad (5.51)
\]

where \( n_i \) is the number of data occurring in both the i-th cluster and its corresponding true cluster, \( k \) is the number of clusters and \( n \) is the total number of data points in the data set.

The accuracy rate (AR) and error rate (ER) are determined by the equations (5.52) and (5.53) respectively.

\[
 AR = r \times 100 \quad (5.52)
\]

and

\[
 ER = 100 - AR \quad (5.53)
\]

**Hybrid FUZZY-cfPSO-FCM Data Clustering Algorithm Using Kernel Distance Metric**

The Kernel distance metric is defined in Chapter 4, Section 4.7 of equation (4.13) and (4.14). The distance between the data point \( x_i \) and cluster centre \( z_j \) is defined as
\[ d(x_i, z_j) = 2(1 - K(x_i, z_j)), 1 \leq i \leq n; 1 \leq j \leq c; 1 \leq k \leq d \] (5.54)

where \[ K(x_i, z_j) = \exp\left(-\frac{\| x_{i,k} - z_{j,k} \|}{\sigma^2}\right), 1 \leq i \leq n; 1 \leq j \leq c; 1 \leq k \leq d \] (5.55)

\[ \sigma = \sqrt{\frac{\sum_{i=1}^{n} u_{ij}^m \| x_{i,k} - z_{j,k} \|^2}{\sum_{i=1}^{n} u_{ij}^m}}, 1 \leq j \leq c; 1 \leq k \leq d \] (5.56)

The fuzzy partition membership values are calculated by (5.11)

The cluster centres are computed by

\[ z_j = \frac{\sum_{i=1}^{n} u_{ij}^m K(x_i, z_j) x_i}{\sum_{i=1}^{n} u_{ij}^m K(x_i, z_j)}, 1 \leq j \leq c \] (5.57)

In fuzzy c-means (FCM) algorithm, the centre values are calculated using the equation (5.57).

The Hybrid FUZZY-cfPSO-FCM Data Clustering Algorithm Using Kernel Distance Metric is described in Figure 5.11.
### Input
Data set $X = \{x_1, x_2, \ldots, x_n\}$, a set of data points

### Output
Cluster centers $z = \{z_1, z_2, \ldots, z_c\}$; fitness value; Accuracy rate and Error rate

1. **Initialize parameters of FUZZY-cfPSO:** Select the number of clusters $c$ ($2 \leq c \leq n$); Choose fuzziness index $m$ ($m>1$); Number of particles; Acceleration constants $c_1, c_2$; Constriction factor $\chi$; Iteration error $\varepsilon \leftarrow 0.00001$; Fix the maximum number of iterations $\text{max}_it$

2. Initialize position and velocity values, $pbest$ for each particle and $gbest$ for the swarm.

3. Read the data set, Initialize fuzzy partition membership values $U^{(0)}$ for the given number of particles.

4. Calculate the cluster centers for each particle using the equation (5.9).

5. Compute the Kernel distance for each particle using the equations (5.54) to (5.56).

6. Update the fuzzy partition membership values using the equation (5.11).

7. Compute the fitness value of each particle using the equation (5.10).

8. Compare and update the every particle’s fitness value with the previous particle’s best solution using the equation (2.1).

9. Compute the global best fitness value using (2.2).

10. Change the velocity of the particle according to (2.5).

11. Change the position of the particle according to (2.4).

12. Check the $\text{max}_it$, if not go to step 5.

13. Record the final membership values $U^{best}$ and center values $z^{best}$.

14. Initialize the fuzzy membership values $U^{(0)} = U^{best}$ and clustering centers $z^{(0)} = \{z^{(0)}_1, z^{(0)}_2, \ldots, z^{(0)}_c\} = z^{best}$ particle 1 and randomly choose the other particles membership values and calculate the corresponding center values using the equation (5.9).

*Contd.*,
15. Compute the *Kernel distance* for each particle using the equations (5.54) to (5.56).
16. Update the new membership values for each particle using the equation (5.11).
17. Compute the *fitness value* of each particle using the equation (5.10)
18. Determine the best fitness value among various particles and record the corresponding center values.
19. Find the *accuracy rate* and *error rate* according to (5.52) and (5.53)
20. Check the max_it if not go to step 15.
21. Record the clustering results.

**Figure 5.11 Hybrid FUZZY-cfPSO-FCM Data Clustering Algorithm**

**Hybrid FUZZY-cfPSO-FCM Data Clustering Algorithm Using Mahalanobis Distance Metric**

The Mahalanobis distance metric is defined in Chapter 4, Section 4.7 of equation (4.12). It is the distance between the data point $x_i$ and cluster centre $z_j$, normalized by the standard deviation of the cluster in each dimension. It takes into account the membership as well as similarity between the data point and cluster centroid. It is the best suited model for clusters that have elliptical shapes. It is defined as

$$d(x_i, z_j) = \sqrt{(X - Z)^t \sigma^{-1} (X - Z)} = \sqrt{\sum_{k=1}^{d} \left( \frac{x_{i,k} - z_{j,k}}{\sigma_k} \right)^2}, 1 \leq i \leq n; 1 \leq j \leq c$$

(5.58)
The Hybrid FUZZY-cfPSO-FCM data clustering algorithm using Mahalanobis distance metric is also the same as given in Figure 5.11 except that the computation of distance formula is replaced by Mahalanobis distance equation (5.58) instead of Kernel distance metric (5.54) to (5.56).

**Hybrid FUZZY-cfPSO-FCM Data Clustering Algorithm Using New Distance Metric**

The New distance metric is defined in Chapter 4, Section 4.7 of equation (4.13) to (4.15). It is robust to noise and outliers. The distance between the data point \(x_i\) and cluster centre \(z_j\) is defined as

\[
d(x_i, z_j) = 1 - \exp(-\beta \| x_{i,k} - z_{j,k} \|^2), 1 \leq i \leq n; 1 \leq j \leq c; 1 \leq k \leq d
\]  

(5.59)

where \(\beta\) and \(\bar{x}\) are defined by (4.14) and (4.15) respectively.

The Hybrid FUZZY-cfPSO-FCM data clustering algorithm using New distance metric is also the same as given in Figure 5.11 except that the computation of distance formula is replaced by New distance equation (5.59) instead of Kernel distance metric (5.54) to (5.56).