Fuzzy C-Means Clustering

Md. Habibur Rahman

Department of Statistics and Data Science
Jahangirnagar University
Savar, Dhaka-1342
Bangladesh
Email: habib.drj@juniv.edu

October 21, 2024

Contents

List of Figures

List of Tables

Chapter 1

Fuzzy C-means Clustering

1.1 Fuzzy C-means Clustering Algorithm

Fuzzy C-means (FCM) clustering is an iterative algorithm used to partition a dataset into k clusters, where each data point can belong to more than one cluster with a certain degree of membership.

To cluster data, specify an array of data points, x_i, with n rows. The number of columns for each data point is equal to the data dimensionality.

$$x_i = [x_{i1}, x_{i2}, ..., x_{ik}]^T \quad \text{where} \quad 1 \leq i \leq n$$

The FCM algorithm computes the cluster centers, C_i. This array contains one row for each cluster center and the number of columns matches the number of columns in $x_{i,}$.

$$C_j = [c_{j1}, c_{j2}, ..., c_{jk}]^T \quad \text{where} \quad 1 \leq j \leq c$$

The FCM algorithm minimizes the following objective function.

$$J_m = \sum_{j=1}^{c} \sum_{i=1}^{n} \mu_{ij}^m D_{ij}^2$$

where,

$$D_{ij} = d_{ij} = \|x_i - C_j\|$$

is the Euclidean distance between the data point and the cluster centroid. The membership degree is calculated as:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

$m(=2)$ is a fuzziness parameter.

The following steps outline the FCM algorithm:

1. **Initialize Parameters**
   - **Number of Clusters k:** Choose the number of clusters to partition the data into.
   - **Fuzziness Parameter m:** Controls the level of cluster fuzziness, typically $m = 2$. The higher the value, the fuzzier the clusters.
   - **Max Iterations and Tolerance:** Define the maximum number of iterations and a threshold for convergence (e.g., 10^{-5}).
2. **Initialize Cluster Centroids**
   - Randomly initialize the cluster centroids or use some heuristic to place them in the feature space.
3. **Calculate Membership Degrees**
   - For each data point, calculate the membership degree μ_{ij} for each cluster j. This membership degree represents how strongly a data point belongs to a cluster.
   - The membership degree is calculated as:

   $$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

   where:
     - x_i is the i-th data point. where $i = 1, 2, ..., n$.
     - c is the number of clusters.
     - $d_{ij} = \|x_i - C_j\|$ is the Euclidean distance between the data point and the cluster centroid.
     - C_j is the centroid of the j-th cluster.
4. **Update Cluster Centroids**
   - Update the cluster centroids using the current membership degrees:

   $$C_j = \frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot x_i}{\sum_{i=1}^{n} (\mu_{ij})^m}$$

   where:
     - n is the number of data points.
     - m is the fuzziness parameter.
     - x_i is the i-th data point.
5. **Check for Convergence**

- Calculate the change in the centroids or the change in the membership degrees.
- If the change is less than a pre-defined tolerance threshold or if the maximum number of iterations is reached, the algorithm converges, and the process stops.

6. **Assign Data Points to Clusters**
   - After convergence, each data point is assigned to the cluster with the highest membership degree. However, since FCM is a fuzzy clustering algorithm, each data point can belong to multiple clusters with varying degrees of membership.

7. **Output the Results**
   - The final cluster centroids.
   - The membership matrix, showing the degree to which each data point belongs to each cluster.
   - The cluster assignments based on the highest membership degree.

Example Workflow

1. Start with an initial guess for cluster centroids.
2. Compute membership degrees for each data point relative to each centroid.
3. Update centroids based on the computed membership degrees.
4. Repeat until the centroids stabilize or the change is minimal.
5. Interpret the resulting clusters using the final centroids and membership degrees.

Key Points

- **Fuzziness:** Controls how ”fuzzy” the boundary between clusters is.
- **Membership:** Each data point has a degree of belonging to each cluster, unlike hard clustering, where each data point belongs strictly to one cluster.
- **Convergence:** The algorithm iterates until the centroids and membership degrees do not change significantly.

Fuzzy C-means clustering is particularly useful when the data points have overlapping boundaries, and strict clustering (as in k-means) might not capture the underlying structure well.

1.2 Fuzzy c-means Clustering Example: Two-Dimensional Data with Three Clusters

Dataset:

$$\text{Data points: } (1,2),(2,3),(3,1),(5,4),(6,5),(8,7)$$

Number of clusters: k = 3

Initialization:

- Choose the number of clusters $k = 3$.
- Initialize the cluster centroids:

$$C_1 = (1,2), \quad C_2 = (5,4), \quad C_3 = (8,7)$$

Step 1: Calculate Membership Degrees

For each data point (x_i, y_i), calculate the Euclidean distance to each centroid:

$$d_{ij} = \sqrt{(x_i - C_{jx})^2 + (y_i - C_{jy})^2}$$

where C_{jx} and C_{jy} are the coordinates of centroid j.

For example, the distances for the first data point $(1,2)$ are:

$$d_{11} = \sqrt{(1-1)^2 + (2-2)^2} = 0$$

$$d_{12} = \sqrt{(1-5)^2 + (2-4)^2} = \sqrt{16+4} = \sqrt{20} \approx 4.47$$

$$d_{13} = \sqrt{(1-8)^2 + (2-7)^2} = \sqrt{49+25} = \sqrt{74} \approx 8.60$$

The membership degree μ_{ij} of data point i to cluster j is calculated as:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{3} \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}}$$

where m is the fuzziness parameter, typically $m = 2$.

Step 2: Update Cluster Centroids

The new centroid C_j for each cluster j is calculated as:

$$C_j = \left(\frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot x_i}{\sum_{i=1}^{n} (\mu_{ij})^m}, \frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot y_i}{\sum_{i=1}^{n} (\mu_{ij})^m} \right)$$

Example Iteration:

Let's assume after one iteration, the updated centroids are:

$$C_1 = (1.5, 2.5), \quad C_2 = (4.5, 4.5), \quad C_3 = (7.5, 6.5)$$

Repeat the Process:
Continue to repeat the calculation of membership degrees and update the centroids until the centroids stabilize and the membership degrees converge.

1.3 Example

Example Given that, $A(1,2)$, $B(1,3)$, $C(4,3)$, $D(5,4)$, $E(8,6)$, and $F(8,7)$ be the six points and the initial cluster centers are $C_1 = (1.5,2.5)$, $C_2 = (4.5,4.5)$, and $C_3 = (8.5,6.5)$. Find the cluster solution using the Fuzzy c-means clustering algorithm.

Solution
Let, $x_i = (X_i, Y_i)$ be the data points where $i = 1,2,...,n$ and here, $n = 6$

Here, $C_1 = (1.5,2.5), C_2 = (4.5,4.5), C_3 = (8.5,6.5)$

$d_{ij} = \sqrt{(X_i - C_{jx})^2 + (Y_i - C_{jy})^2}$, here, $i = 1,2,...,6$ and $j = 1,2,3$

$\mu_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}}\right)^{\frac{2}{m-1}}}$ where $m = 2$

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}	$\mu_{i1}^2 X_i$	$\mu_{i1}^2 Y_i$	$\mu_{i2}^2 X_i$	$\mu_{i2}^2 Y_i$	$\mu_{i3}^2 X_i$	$\mu_{i3}^2 Y_i$
A	1	2	0.71	4.3	8.75	0.97	0.03	0.01	0.94	1.88	0	0	0	0
B	1	3	0.71	3.81	8.28	0.96	0.03	0.01	0.92	2.76	0	0	0	0
C	4	3	2.55	1.58	5.7	0.26	0.68	0.05	0.27	0.2	1.85	1.39	0.01	0.01
D	5	4	3.81	0.71	4.3	0.03	0.94	0.03	0	0	4.42	3.53	0	0
E	8	6	7.38	3.81	0.71	0.01	0.03	0.96	0	0	0.01	0.01	7.37	5.53
F	8	7	7.91	4.3	0.71	0.01	0.03	0.97	0	0	0.01	0.01	7.53	6.59
Total									2.13	4.84	6.29	4.94	14.91	12.13

Table 1.1: Iteration 1

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.71	4.3	8.75	0.97	0.03	0.01
B	1	3	0.71	3.81	8.28	0.96	0.03	0.01
C	4	3	2.55	1.58	5.7	0.26	0.68	0.05
D	5	4	3.81	0.71	4.3	0.03	0.94	0.03
E	8	6	7.38	3.81	0.71	0.01	0.03	0.96
F	8	7	7.91	4.3	0.71	0.01	0.03	0.97

$$C_j = \left(\frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n} (\mu_{ij})^m}, \frac{\sum_{i=1}^{n} (\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n} (\mu_{ij})^m} \right)$$

$$C_1 = \left(\frac{2.13}{1.93}, \frac{4.84}{1.93} \right) = (1.10, 2.51)$$

$$C_2 = \left(\frac{6.29}{1.35}, \frac{4.94}{1.35}\right) = (4.66, 3.66)$$

$$C_3 = \left(\frac{14.91}{1.87}, \frac{12.13}{1.87}\right) = (8.27, 6.73)$$

Table 1.2: Iteration 2

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.52	4.02	8.67	0.98	0.02	0.00
B	1	3	0.5	3.72	8.17	0.98	0.02	0.00
C	4	3	2.94	0.93	5.67	0.09	0.89	0.02
D	5	4	4.17	0.48	4.26	0.01	0.97	0.01
E	8	6	7.73	4.08	0.78	0.01	0.03	0.96
F	8	7	8.23	4.72	0.38	0.00	0.01	0.99

$$C_j = \left(\frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n}(\mu_{ij})^m}\right)$$

$$C_1 = \left(\frac{1.95}{1.93}, \frac{4.82}{1.93}\right) = (1.01, 2.50)$$

$$C_2 = \left(\frac{7.88}{1.73}, \frac{6.15}{1.73}\right) = (4.54, 3.55)$$

$$C_3 = \left(\frac{15.21}{1.90}, \frac{12.39}{1.90}\right) = (8.25, 6.72)$$

Table 1.3: Iteration 3

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.5	3.86	8.65	0.98	0.02	0.00
B	1	3	0.5	3.58	8.15	0.98	0.02	0.00
C	4	3	3.03	0.77	5.65	0.06	0.92	0.02
D	5	4	4.26	0.64	4.24	0.02	0.96	0.02
E	8	6	7.82	4.24	0.76	0.01	0.03	0.96
F	8	7	8.31	4.89	0.38	0.00	0.01	0.99

$$C_j = \left(\frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n}(\mu_{ij})^m}\right)$$

$$C_1 = \left(\frac{1.93}{1.92}, \frac{4.81}{1.92}\right) = (1.00, 2.50)$$

$$C_2 = \left(\frac{8.01}{1.77}, \frac{6.24}{1.77}\right) = (4.53, 3.53)$$

$$C_3 = \left(\frac{15.21}{1.90}, \frac{12.39}{1.90}\right) = (8.25, 6.72)$$

Table 1.4: Iteration 4

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.5	3.85	8.65	0.98	0.02	0.00
B	1	3	0.5	3.57	8.15	0.98	0.02	0.00
C	4	3	3.04	0.75	5.65	0.06	0.93	0.02
D	5	4	4.27	0.66	4.24	0.02	0.95	0.02
E	8	6	7.83	4.26	0.76	0.01	0.03	0.96
F	8	7	8.32	4.91	0.38	0.00	0.01	0.99

$$C_j = \left(\frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n}(\mu_{ij})^m}\right)$$

$$C_1 = \left(\frac{1.93}{1.92}, \frac{4.81}{1.92}\right) = (1.00, 2.50)$$

$$C_2 = \left(\frac{7.98}{1.77}, \frac{6.21}{1.77}\right) = (4.51, 3.51)$$

$$C_3 = \left(\frac{15.21}{1.90}, \frac{12.39}{1.90}\right) = (8.25, 6.72)$$

Table 1.5: Iteration 5

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.5	3.82	8.65	0.98	0.02	0.00
B	1	3	0.5	3.55	8.15	0.98	0.02	0.00
C	4	3	3.04	0.72	5.65	0.05	0.93	0.02
D	5	4	4.27	0.69	4.24	0.02	0.95	0.03
E	8	6	7.83	4.29	0.76	0.01	0.03	0.96
F	8	7	8.32	4.94	0.38	0.00	0.01	0.99

$$C_j = \left(\frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n}(\mu_{ij})^m}\right)$$

$$C_1 = \left(\frac{1.93}{1.92}, \frac{4.81}{1.92}\right) = (1.00, 2.50)$$

$$C_2 = \left(\frac{7.98}{1.77}, \frac{6.21}{1.77}\right) = (4.51, 3.51)$$

$$C_3 = \left(\frac{15.21}{1.90}, \frac{12.39}{1.90}\right) = (8.25, 6.72)$$

Table 1.6: Iteration 6

Item	X_i	Y_i	d_{i1}	d_{i2}	d_{i3}	μ_{i1}	μ_{i2}	μ_{i3}
A	1	2	0.5	3.82	8.65	0.98	0.02	0.00
B	1	3	0.5	3.55	8.15	0.98	0.02	0.00
C	4	3	3.04	0.72	5.65	0.05	0.93	0.02
D	5	4	4.27	0.69	4.24	0.02	0.95	0.03
E	8	6	7.83	4.29	0.76	0.01	0.03	0.96
F	8	7	8.32	4.94	0.38	0.00	0.01	0.99

$$C_j = \left(\frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot X_i}{\sum_{i=1}^{n}(\mu_{ij})^m}, \frac{\sum_{i=1}^{n}(\mu_{ij})^m \cdot Y_i}{\sum_{i=1}^{n}(\mu_{ij})^m}\right)$$

$$C_1 = \left(\frac{1.93}{1.92}, \frac{4.81}{1.92}\right) = (1.00, 2.50)$$

$$C_2 = \left(\frac{7.98}{1.77}, \frac{6.21}{1.77}\right) = (4.51, 3.51)$$

$$C_3 = \left(\frac{15.21}{1.90}, \frac{12.39}{1.90}\right) = (8.25, 6.72)$$

1.4 Theoretical Explanation of Fuzzy Partition Coefficient

The fuzzy partition coefficient (FPC) is a measure of the amount of ”fuzziness” in a fuzzy c-means clustering result. It was introduced by Bezdek in 1973 and is defined as (Bezdek; 1973):

$$FPC = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^2$$

where:

- n is the number of data points
- c is the number of clusters
- u_{ij} is the membership degree of data point i in cluster j

(Reference: Bezdek, J. C. (1973). Cluster validity with fuzzy sets, Journal of Cybernetics 3(3): 58–73.)
The FPC ranges from $\frac{1}{c}$ to 1, with 1 indicating the least amount of fuzziness (i.e. the clustering is most crisp). A higher FPC value generally indicates better clustering results.
The FPC can be interpreted as the average of the sum of the squares of the membership grades. As the membership degrees become more evenly distributed across clusters, the FPC decreases, indicating more fuzziness in the partition.

1.5 Numerical Example for Fuzzy Partition Coefficient

Let's calculate the FPC for a simple example with 5 data points and 3 clusters:

$$U = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0.7 & 0.2 & 0.1 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.1 & 0.7 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$$

The membership matrix U shows the degree to which each data point belongs to each cluster.
To calculate the FPC:
1. Sum the squares of each row (data point) in U:

$$\begin{aligned} \text{Row 1:} \quad & 0.1^2 + 0.8^2 + 0.1^2 = 0.66 \\ \text{Row 2:} \quad & 0.7^2 + 0.2^2 + 0.1^2 = 0.54 \\ \text{Row 3:} \quad & 0.3^2 + 0.6^2 + 0.1^2 = 0.46 \\ \text{Row 4:} \quad & 0.2^2 + 0.1^2 + 0.7^2 = 0.54 \\ \text{Row 5:} \quad & 0.6^2 + 0.2^2 + 0.2^2 = 0.44 \end{aligned}$$

2. Sum the row sums: $0.66 + 0.54 + 0.46 + 0.54 + 0.44 = 2.64$
3. Divide by the number of data points ($n = 5$): $\frac{2.64}{5} = 0.528$

Therefore, the fuzzy partition coefficient for this example is 0.528, indicating a moderate amount of fuzziness in the clustering result.

1.6 FCM Algorithm

The implementation of FCM is

i) Initialise $U^{(0)} = \{u_{kn}^{(0)} | k = 1, \dots, K; n = 1, \dots, N\}$.

ii) At the step t, with the partition matrix $U(t)$, calculate the centroids $c_k | k = 1, \dots, K$ using $c_k = \frac{\sum_{n=1}^{N} u_{kn}^{m} x_n}{\sum_{n=1}^{N} u_{kn}^{m}}$.

iii) Update the partition matrix U_{t+1} using $u_{kn} = \frac{1}{\sum_{l=1}^{K} \left(\frac{D(x_n, c_k)^2}{D(x_n, c_l)^2} \right)^{\frac{1}{m-1}}}$ which implies a constraint that the probabilistic memberships satisfy $\sum_{k=1}^{K} u_{nk} = 1$.

iv) Repeat Steps 2 and 3 until $\max_{kn}\{|u_{kn}^{(t+1)} - u_{kn}^{(t)}|\} < \varepsilon$.

This iteration will stop when $\max_{kn}\{|u_{kn}^{(t+1)} - u_{kn}^{(t)}|\} < \varepsilon$, where ε is a small positive real number and t is the iteration index. There are many implementations of FCM in different platforms: in Fuzzy Logic Toolbox of MATLAB, function fcm applies the FCM method; in the package e1071 in R, function cmeans also implements the FCM method.

Chapter 2

Python Code

2.1 Fuzzy C-Means clustering for Data

Example Apply Fuzzy C-means clustering to cluster the objects, each object is a two-dimensional data point. Also, obtain the optimal number of clusters.

Table 2.1: Data for Fuzzy C-means Clustering

ID	X	Y
0	1.9	2.0
1	1.5	2.0
2	2.0	2.5
3	1.5	2.5
4	3.0	6.0
5	3.5	6.5
6	3.0	7.0
7	5.5	3.5
8	5.0	4.0
9	4.5	3.5
10	8.0	7.0
11	9.0	7.0
12	8.5	7.0
13	8.5	7.5

Solution The two-dimensional scatter plot of the data.

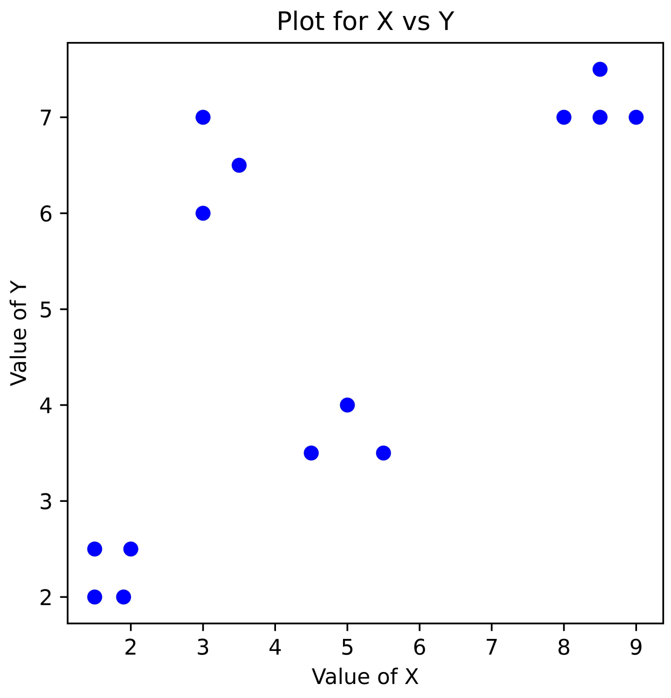

Figure 2.1: Schematic scatter plot of the Data.

Python Code for Fuzzy C-Means Clustering for the Data.

```
'''Fuzzy C-means Clustering'''
"""Created on Mon Sep  9 18:50:39 2024 @author: RM Habib"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#==Install Pacage===#
#!pip install scikit-fuzzy
import skfuzzy as fuzz

# Create a matrix with the specified values
data = np.array([[1.9, 2],[1.5,2],
                 [2, 2.5],[1.5, 2.5],
                 [3,6],[3.5,6.5],[3,7],
                 [5.5, 3.5],
                 [5, 4],[4.5,3.5],
                 [8, 7],[9,7],[8.5,7],
                 [8.5, 7.5]])

import string
# Get the first 10 uppercase letters
letters = string.ascii_uppercase[:data.shape[0]]

# Create a DataFrame from the matrix
D = pd.DataFrame(data, columns=['X', 'Y'])
# Display the DataFrame
print(D)

plt.figure(figsize=(5, 5))
plt.plot(D['X'],D['Y'],'o',color='b')
plt.xlabel("Value of X")
plt.ylabel("Value of Y")
plt.title('Plot for X vs Y')
plt.show()

np.random.seed(104729)
FPC = []
K=[]
for k in range(2, 11):
    # Perform fuzzy c-means clustering
    cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
        D.T, k, m=2, error=0.0005, maxiter=1000, init=None)
    FPC.append(fpc)
    K.append(k)

K
FPC
k=pd.DataFrame(FPC).idxmax()+2
k[0]

plt.figure(figsize = (6, 5))
plt.plot(range(2, 11), np.round(FPC,3), ls='-',
         linewidth=2, color='red',marker='o',
         markerfacecolor='blue', markersize=7)
plt.axvline(x=k[0], color='b',ls='--',linewidth=1.5)
```

```
#plt.axhline(y=78, color='b',ls='--',linewidth=2)
plt.title('Optimal number of cluster')
plt.xlabel('Number of clusters for Fuzzy C-means')
plt.ylabel('Fuzzy partition coefficient (FPC)')
#plt.savefig('D:/Elbow k-Means.jpg',dpi=700)
plt.show()

# Perform fuzzy c-means clustering
np.random.seed(199999)
cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
    D.T, k[0], m=2, error=0.0005, maxiter=1000, init=None)

# Print results
print("Cluster centers:\n", cntr)
print("\nWithin-cluster sum of squares:\n", jm)
print("\nFinal partition matrix:\n", np.round(u,3))
print("\nNumber of iterations:\n", p)
print("\nThe fuzzy partition coefficient (FPC):\n",fpc)

# Predict cluster membership for each data point
cluster_membership = np.argmax(u, axis=0)
cluster_membership

# Create a list of colors for each cluster
colors = ['r', 'g', 'b', 'm','c',  'y', 'k']

# Create a figure with specified size
plt.figure(figsize=(5, 5))
# Plot the data points with their respective cluster colors
for i in range(len(D)):
    plt.plot(D.iloc[i, 0], D.iloc[i, 1], 'o',
             color=colors[cluster_membership[i]])

# Set the x-axis and y-axis labels
plt.xlabel("Value of X")
plt.ylabel("Value of Y")
plt.title('Plot for X vs Y')
plt.show()
```

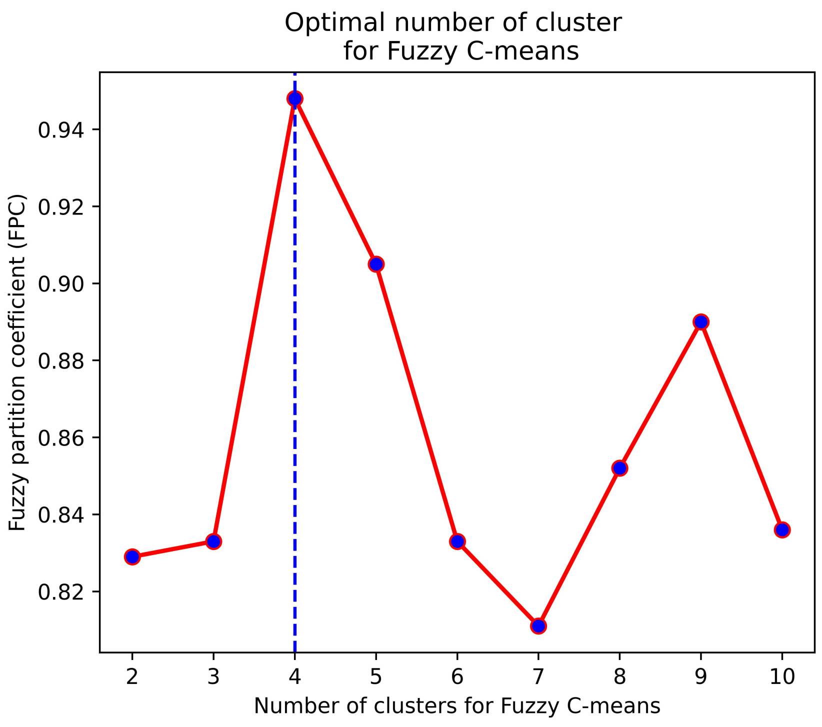

Figure 2.2: Optimal number of Cluster for Fuzzy C-Means Clustering

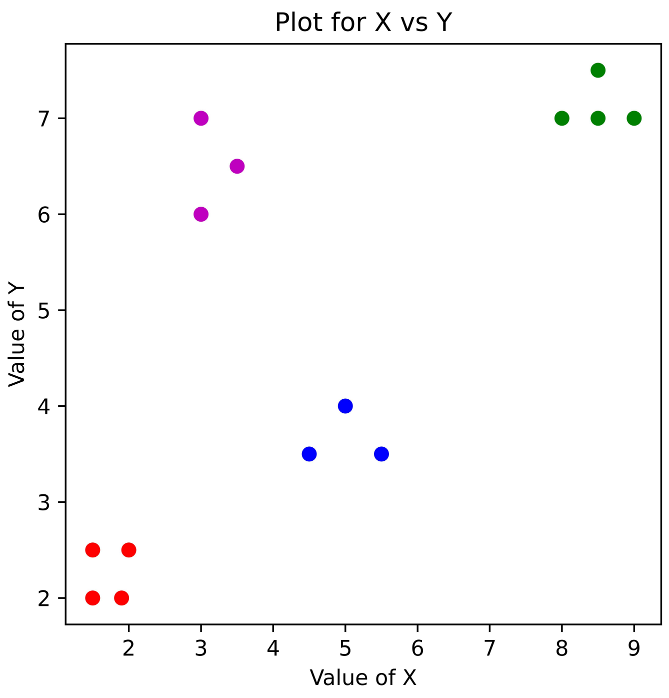

Figure 2.3: Schematic scatter plot for data based on Fuzzy C-means Cluster solution.

2.2 Fuzzy C-Means clustering for Iris Data

```
'''Fuzzy C-means Clustering for Iris Data'''
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#==Install Pacage===#
#!pip install scikit-fuzzy
import skfuzzy as fuzz
from sklearn.datasets import load_iris

# Load the Iris dataset
iris = load_iris()
X = iris.data.T

np.random.seed(104729)
FPC = []
for k in range(2, 11):
    # Perform fuzzy c-means clustering
    cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
        X, k, m=2, error=0.0005, maxiter=1000, init=None)
    FPC.append(fpc)

k=pd.DataFrame(FPC).idxmax()+2
k[0]

plt.figure(figsize = (6, 5))
plt.plot(range(2, 11), np.round(FPC,3), ls='--',
         linewidth=2, color='red',marker='o',
         markerfacecolor='blue', markersize=7)
plt.axvline(x=k[0], color='b',ls='--',linewidth=2)
plt.text(2.5,0.55,r'$k=2$')
plt.title('Optimal number of cluster')
plt.xlabel('Number of clusters for Fuzzy C-means')
plt.ylabel('Fuzzy partition coefficient (FPC)')
plt.show()

'''#error=0.005: the termination criterion (stop when the maximum
absolute difference between iteration coefficients is less than this value)
```

```
#init=None: the initial partition matrix (U matrix) is randomly generated'''

# Perform fuzzy c-means clustering
np.random.seed(199999)
cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(
    X, k[0], m=2, error=0.0005, maxiter=1000, init=None)

# Print results
print("Cluster centers:\n", cntr)
print("\nWithin-cluster sum of squares:\n", jm)
print("\nFinal partition matrix:\n", np.round(u,3))
print("\nNumber of iterations:\n", p)
print("\nThe fuzzy partition coefficient (FPC):\n",fpc)

# Predict cluster membership for each data point
cluster_membership = np.argmax(u, axis=0)
cluster_membership

DT=pd.DataFrame(iris.data)
# Create a list of colors for each cluster
colors = ['r', 'g', 'b', 'm','c',  'y', 'k']

# Create a figure with specified size
plt.figure(figsize=(5, 5))
# Plot the data points with their respective cluster colors
for i in range(len(DT)):
    plt.plot(DT.iloc[i, 0], DT.iloc[i, 1], 'o',
             color=colors[cluster_membership[i]])

# Set the x-axis and y-axis labels
plt.xlabel("Value of Sepal length")
plt.ylabel("Value of Sepal width")
plt.title('Plot for Iris data')
plt.show()
```

Bibliography

Bezdek, J. C. (1973). Cluster validity with fuzzy sets, *Journal of Cybernetics* **3**(3): 58–73.