Pycon ID 2025

Clustering of High-Achieving Students Using Fuzzy C-Means

Nawindah
Universitas Budi Luhur
E-mail : nawindah@budiluhur.ac.id

Dec 13th-14th,2025
Trilogi University,Jakarta

Basic Consepts Fuzzy C Means

- Data is located in more than one cluster.
- Each data in a cluster has a weight.
- The concept used are fuzzy logic and fuzzy set.
- Fuzzy logic was first developed by Lotfi Zadeh through his writing in 1965 entitled “Fuzzy Set”.
- A fuzzy set is a mathematical model derived from vague qualitative or quantitative data, often generated through natural language. Its membership value is between 0 and 1.

Fuzzy C Means Algorithm

- Enter the data to be grouped *X*, in the form of an *nxm* matrix
- Determine: number of cluster, bobot, maximum iteration, expected smallest error, initial objective function, initial iteration.
- Generate random numbers with and are the elements of the initial partition matrix .

$$\mu_{ik}=\frac{\mu_{ik}}{Q_i}$$

- Calculate the k cluster center:

$$V_{kj}=\frac{\sum_{i=1}^{n}((\mu_{ik})^{w}*x_{ij})}{\sum_{i=1}^{n}(\mu_{ik})^{w}}$$

- Calculate the objective function in the iteration

$$P_t=\sum_{i=1}^{n}\sum_{k=1}^{c}([\sum_{j=1}^{m}(x_{ij}-V_{kj})^2](\mu_{ik}^{w}))$$

- Calculate the change of partition matrix

$$\mu_{ik}=\frac{[\sum_{j=1}^{m}(X_{ij}-V_{kj})^2]^{\frac{-1}{(w-1)}}}{\sum_{k=1}^{k}[\sum_{j=1}^{m}(X_{ij}-V_{kj})^2]^{\frac{-1}{(w-1)}}}$$

Fuzzy C Means Algorithm

1) Select an initial fuzzy pseudo-partition, e.e., assign values to all *Wij*
2) Repeat
3) compute the centroid of each cluster using the fuzzy partition
4) Update the fuzzy partition i.e. the *Wij*
5) Until the centroid don't change

   There's alternative stopping criteria. Ex." change in the error is bellow a specified threshold", or "absolute change in any *Wij* is below a given threshold"

Data

	A	B	C	D
1	Presensi	Tugas	Rapor	Motivasi
2	2	60	73.1	Sedang
3	10	90	75	Tinggi
4	5	87	65.5	Rendah
5	5	89	65.6	Rendah
6	6	77	63.8	Rendah
7	10	89	76.5	Tinggi
8	9	90	77.4	Tinggi
9	6	86	71.6	Tinggi
10	6	85	65.1	Rendah
11	10	90	75.7	Tinggi
12	8	87	75.1	Tinggi
13	8	84	65.8	Rendah
14	4	88	66.5	Sedang
15	10	85	64.8	Tinggi
16	7	86	66.7	Sedang

	A	B	C	D	E
1	Siswa	Presensi	Tugas	Rapor	Motivasi
2	1	0	0.5	0.5	0.5
3	2	1	1	1	1
4	3	0.5	1	0	0
5	4	0.5	1	0	0
6	5	1	1	0	0
7	6	1	1	1	1
8	7	1	1	1	1
9	8	1	1	0.5	1
10	9	1	1	0	0
11	10	1	1	1	1
12	11	1	1	1	1
13	12	1	1	0	0
14	13	0.5	1	0	0
15	14	1	1	0	1
16	15	1	1	0	0.5

Convert numbers to a set of fuzzy numbers by determining the upper and lower limits

Libraries used

- import numpy as np, numpy.random

  from scipy.spatial import distance
- import numpy as np
- import matplotlib.pyplot as plt
- import pandas as pd

  from fcmeans import FCM

Implementation

```
[10]: import numpy as np, numpy.random
      from scipy.spatial import distance
      k = 2
      p = 4
```

```
[11]: X = np.array([
                [0,0.5,0.5,0.5],
                [1,1,1,1],
                [0.5,1,0,0],
                [0.5,1,0,0],
                [1,1,0,0],
                [1,1,1,1],
                [1,1,1,1],
                [1,1,0.5,1],
                [1,1,0,0],
                [1,1,1,1],
                [1,1,1,1],
                [1,1,0,0],
                [0.5,1,0,0],
                [1,1,0,1],
                [1,1,0,0.5]])
```

```
# Print the number of data and dimension
n = len(X)
d = len(X[0])
addZeros = np.zeros((n, 1))
X = np.append(X, addZeros, axis=1)
print("The FCM algorithm: \n")
print("The training data: \n", X)
print("\nTotal number of data: ",n)
print("Total number of features: ",d)
print("Total number of Clusters: ",k)
```

Implementation

```
The FCM algorithm:

The training data:
 [[0.  0.5 0.5 0.5 0. ]
 [1.  1.  1.  1.  0. ]
 [0.5 1.  0.  0.  0. ]
 [0.5 1.  0.  0.  0. ]
 [1.  1.  0.  0.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  0.5 1.  0. ]
 [1.  1.  0.  0.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  0.  0.  0. ]
 [0.5 1.  0.  0.  0. ]
 [1.  1.  0.  1.  0. ]
 [1.  1.  0.  0.5 0. ]]

Total number of data:  15
Total number of features:  4
Total number of Clusters:  2
```

```
#Randomly initialize the weight matrix¶
weight = np.random.dirichlet(np.ones(k),size=n)
print("\nThe initial weight: \n", np.round(weight,2))
```

```
The initial weight:
 [[0.15 0.85]
 [0.45 0.55]
 [0.02 0.98]
 [0.18 0.82]
 [0.41 0.59]
 [0.34 0.66]
 [0.94 0.06]
 [0.14 0.86]
 [0.9  0.1 ]
 [0.64 0.36]
 [0.91 0.09]
 [0.   1.  ]
 [0.41 0.59]
 [0.76 0.24]
 [0.58 0.42]]
```

Implementation

```
for it in range(3): # Total number of iterations

    # Compute centroid
    for j in range(k):
        denoSum = sum(np.power(weight[:,j],2))

        sumMM =0
        for i in range(n):
            mm = np.multiply(np.power(weight[i,j],p),X[i,:])
            sumMM +=mm
        cc = sumMM/denoSum
        C[j] = np.reshape(cc,d+1)
```

```
print("\nUpdating the fuzzy pseudo partition")
```

```
Updating the fuzzy pseudo partition
```

```
for i in range(n):
    denoSumNext = 0
    for j in range(k):
        denoSumNext += np.power(1/distance.euclidean(C[j,0:d], X[i,0:d]),1/(p-1))
        for j in range(k):
            w = np.power((1/distance.euclidean(C[j,0:d], X[i,0:d])),1/(p-1))/denoSumNext
            weight[i,j] = w
```

Implementation

```
print("\nSSE: ",np.round(SSE,4))
```

```
SSE:  1.6009
```

```
!pip install fuzzy-c-means
```

```
Requirement already satisfied: fuzzy-c-means in ./venv/lib/python3.12/site-packages (1.7.2)
Requirement already satisfied: joblib<2.0.0,>=1.2.0 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (1.5.2)
Requirement already satisfied: numpy<2.0.0,>=1.21.1 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (1.26.4)
Requirement already satisfied: pydantic<3.0.0,>=2.6.4 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (2.12.5)
Requirement already satisfied: tabulate<0.9.0,>=0.8.9 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (0.8.10)
Requirement already satisfied: tqdm<5.0.0,>=4.64.1 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (4.67.1)
Requirement already satisfied: typer<0.10.0,>=0.9.0 in ./venv/lib/python3.12/site-packages (from fuzzy-c-means) (0.9.4)
Requirement already satisfied: annotated-types>=0.6.0 in ./venv/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.6.4->fuzzy-c-means)
(0.7.0)
Requirement already satisfied: pydantic-core==2.41.5 in ./venv/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.6.4->fuzzy-c-means)
(2.41.5)
Requirement already satisfied: typing-extensions>=4.14.1 in ./venv/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.6.4->fuzzy-c-means)
(4.15.0)
Requirement already satisfied: typing-inspection>=0.4.2 in ./venv/lib/python3.12/site-packages (from pydantic<3.0.0,>=2.6.4->fuzzy-c-means)
(0.4.2)
Requirement already satisfied: click<9.0.0,>=7.1.1 in ./venv/lib/python3.12/site-packages (from typer<0.10.0,>=0.9.0->fuzzy-c-means) (8.3.1)
```

Implementation

```
print("\nThe final weights: \n", np.round(weight,2))
```

```
The final weights:
 [[0.5  0.5 ]
 [0.53 0.47]
 [0.46 0.54]
 [0.46 0.54]
 [0.49 0.51]
 [0.53 0.47]
 [0.53 0.47]
 [0.53 0.47]
 [0.49 0.51]
 [0.53 0.47]
 [0.53 0.47]
 [0.49 0.51]
 [0.46 0.54]
 [0.52 0.48]
 [0.51 0.49]]
```

```
for i in range(n):
    cNumber = np.where(weight[i] == np.amax(weight[i]))
    X[i,d] = cNumber[0][0]
```

```
print("\nThe data with cluster number: \n", X)
```

```
The data with cluster number:
 [[0.  0.5 0.5 0.5 0. ]
 [1.  1.  1.  1.  0. ]
 [0.5 1.  0.  0.  1. ]
 [0.5 1.  0.  0.  1. ]
 [1.  1.  0.  0.  1. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  0.5 1.  0. ]
 [1.  1.  0.  0.  1. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  1.  1.  0. ]
 [1.  1.  0.  0.  1. ]
 [0.5 1.  0.  0.  1. ]
 [1.  1.  0.  1.  0. ]
 [1.  1.  0.  0.5 0. ]]
```

```
SSE = 0
for j in range(k):
    for i in range(n):
        SSE += np.power(weight[i,j],p)*distance.euclidean(C[j,0:d], X[i,0:d])
```

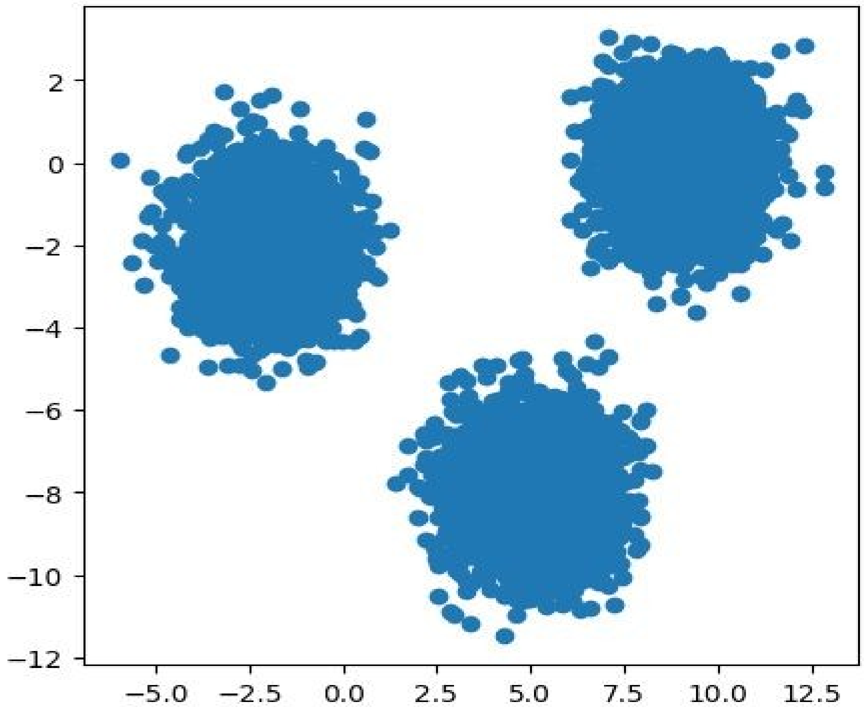
Implementation

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from fcmeans import FCM
```

```
samples=3000
x=np.concatenate((np.random.normal((-2,-2),size=(samples,2)),
                  np.random.normal((9,0),size=(samples,2)),
                   np.random.normal((5,-8),size=(samples,2))
))
```

```
plt.figure(figsize=(5,5))
plt.scatter(x[:,0],x[:,1],alpha=1)
plt.show()
```


```
plt.figure(figsize=(5,5))
plt.scatter(x[:,0],x[:,1],alpha=1)
plt.show()
```

Implementation

```
fcm=FCM(n_clusters=2)
fcm.fit(x)
```

```
fcm_center=fcm.centers
fcm_labels=fcm.predict(x)
```

```
fcm_center
```

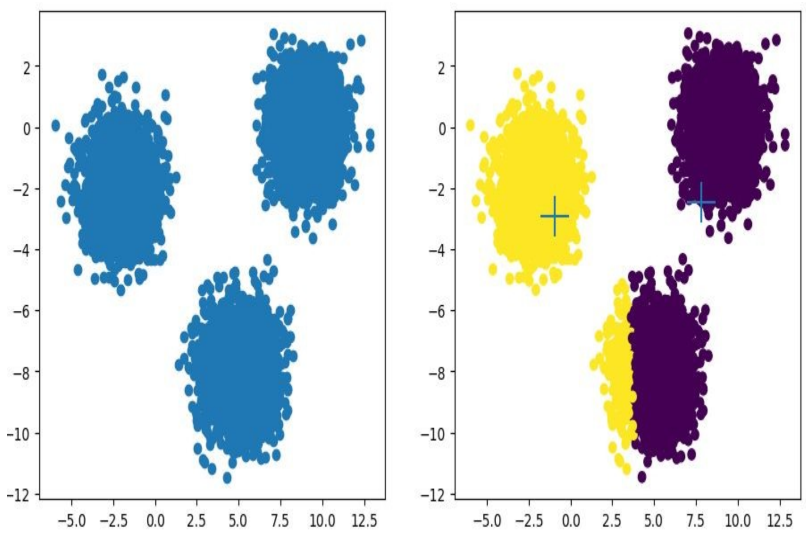
```
array([[ 7.82164474, -2.46697422],
       [-0.94409682, -2.90282964]])
```

```
fcm_labels
```

```
array([1, 1, 1, ..., 0, 0, 0])
```

```
f,axes=plt.subplots(1,2,figsize=(11,5))
axes[0].scatter(x[:,0],x[:,1],alpha=1)
axes[1].scatter(x[:,0],x[:,1],c=fcm_labels,alpha=1)
axes[1].scatter(fcm_center[:,0],fcm_center[:,1],marker="+",s=500)
plt.show()
```

Implementation



Reference

- L. A. Zadeh, "Fuzzy sets," Information and Control 8, pp. 338-353, 1965.
- Timothy, J.Ross, 2006, “Fuzzy Logic With Engineering Application”, John Wiley &. Sons, Chichester, England.
- Kusumadewi, Sri dan Hari, Purnomo. (2010). Aplikasi Logika Fuzzy Untuk Mendukung. Keputusan. Yogyakarta : Graha Ilmu
- Bezdek, J. C.., Robert Ehrlich., & William Full (1981). FCM: The Fuzzy c-Means. Clustering Algorithm. Bieniawski, Z.T., (1989)
- #https://www.kaggle.com/code/engrshabir/fuzzy-c-mean-clustering-algorithm

Thank You

- Link slide :

https://github.com/nawindahx/light_talk_2025