

Computational Intelligence Lab Report - Lab 12 - Mr. Amini

Seyed Mohammadsaleh Mirzatabatabaei (9623105 - smsmt@aut.ac.ir)

Amirkabir University of Technology (Tehran polytechnic)

Table of Contents

C-mean fuzzy theory and formulas:	1
Fuzzy C-mean algorithm in command line:	2
Clear recent data	2
Import data from excel file	2
Fuzzy C-mean algorithm	3
Cluster centers	3
Membership function table	3
Practices:	4
Practice 1	4
Practice 2	5
Final comments	6

C-mean fuzzy theory and formulas:

data : $\{x_1, x_2, x_3, \dots, x_n\}$

n : Number of samples

center : $\{v_1, v_2, v_3, \dots, v_c\}$

c : Number of clusters

cost function : $J = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \|x_{ik} - v_i\|, m \in [1, \infty], \sum_{i=1}^c \mu_{ik} = 1$

μ_{ik} : membership value of sample x_k to a cluster with center v_i – in fuzzy between 0 and 1

1. Suppose there are n data points, $X = \{x_1, \dots, x_n\}$. Fix $c, 2 \leq c < n$, and initialize $U^{(0)} \in M_c$.

2. At iteration $l, l = 0, 1, 2, \dots$ compute the c-mean vectors

$$v_i^{(l)} = \frac{\sum_{k=1}^n \mu_{ik}^{(l)} x_k}{\sum_{k=1}^n \mu_{ik}^{(l)}}$$

where $[\mu_{ik}^{(l)}] = U^{(l)}$, and $i = 1, 2, \dots, c$.

3. Update $U^{(l)}$ to $U^{(l+1)}$ using

$$\mu_{ik}^{(l+1)} = \begin{cases} 1 & \|x_k - v_i^{(l)}\| = \min_{1 \leq j \leq c} (\|x_k - v_j^{(l)}\|) \\ 0 & \text{otherwise} \end{cases}$$

4. Compare $U^{(l)}$ with $U^{(l+1)}$: if $\|U^{(l+1)} - U^{(l)}\| < \epsilon$ for a small constant ϵ stop; otherwise, set $l = l + 1$ and go to Step 2.

Fuzzy C-mean algorithm in command line:

The `fcm` function performs *FCM clustering*. It starts with a random initial guess for the cluster centers; that is the mean location of each cluster. Next, `fcm` assigns every data point a random membership grade for each cluster. By iteratively updating the cluster centers and the membership grades for each data point, `fcm` moves the cluster centers to the correct location within a data set and, for each data point, finds the degree of membership in each cluster. This iteration minimizes an objective function that represents the distance from any given data point to a cluster center weighted by the membership of that data point in the cluster.

Clear recent data

```
clear; close all; clc;
```

Import data from excel file

Importing data from the following text file: Social_Network_Ads.csv

```
% Setup the Import Options and import the data
opts = delimitedTextImportOptions("NumVariables", 5);

% Specify range and delimiter
opts.DataLines = [2, 400];
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["Var1", "Gender", "Age", "EstimatedSalary", "Purchased"];
opts.SelectedVariableNames = ["Gender", "Age", "EstimatedSalary", "Purchased"];
opts.VariableTypes = ["string", "categorical", "double", "double", "double"];

% Specify file level properties
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, "Var1", "WhitespaceRule", "preserve");
opts = setvaropts(opts, ["Var1", "Gender"], "EmptyFieldRule", "auto");

% Import the data
SocialNetworkAds = readtable("C:\Users\smsmt\Desktop\Social_Network_Ads.csv", opts);

% Clear temporary variables
clear opts;

% Social_Network_Ads table
SocialNetworkAds(1:10, :)
```

ans = 10×4 table

	Gender	Age	EstimatedSalary	Purchased
1	Male	19	19000	0
2	Male	35	20000	0
3	Female	26	43000	0

	Gender	Age	EstimatedSalary	Purchased
4	Female	27	57000	0
5	Male	19	76000	0
6	Male	27	58000	0
7	Female	27	84000	0
8	Female	32	150000	1
9	Male	25	33000	0
10	Female	35	65000	0

Fuzzy C-mean algorithm

```
% Convert categorical gender to number
SocialNetworkAds.Gender = double(SocialNetworkAds.Gender);
data = table2array(SocialNetworkAds);

% Calculate mean and standard deviation of data each column
std = std(data);
mean = mean(data);

% Normalize each column data with mean=0 and std=1
norm_data = normalize(data);

% Run C-mean fuzzy clustering
numberOfClusters = 7;
options = [4, 1000, 1e-5, false]; % Exponent:4, Max. Iterations:100, Min. Improvement: 1e-05
[center, U] = fcm(norm_data, numberOfClusters, options);
```

Cluster centers

convert centers to SocialNetworkAds data scale without normalize

```
center = center .* std + mean
```

```
center = 7x4
    1.5033    36.606    68173    0.31628
    1.465     36.3    67649    0.29746
    1.3986    44.023    86530    0.78473
    1.5689    35.839    66909    0.27553
    1.4264    43.821    85137    0.77361
    1.9354    35.705    59776    0.053191
    1.0814    34.853    62657    0.066237
```

Membership function table

μ_{ik} : value of sample x_k to a cluster with center v_i — in fuzzy between 0 and 1

```
U
```

```
U = 7x399
    0.149    0.14788    0.14788    0.1461    0.15093    0.14773 ...
    0.14927    0.14708    0.15274    0.15182    0.15072    0.14679
```

0.11501	0.10865	0.10727	0.10366	0.11418	0.1005
0.15489	0.15527	0.14822	0.14561	0.15826	0.15993
0.11632	0.11036	0.10797	0.10426	0.11544	0.10205
0.17357	0.19981	0.12842	0.12112	0.17414	0.21895
0.14193	0.13096	0.2075	0.22743	0.13633	0.12406

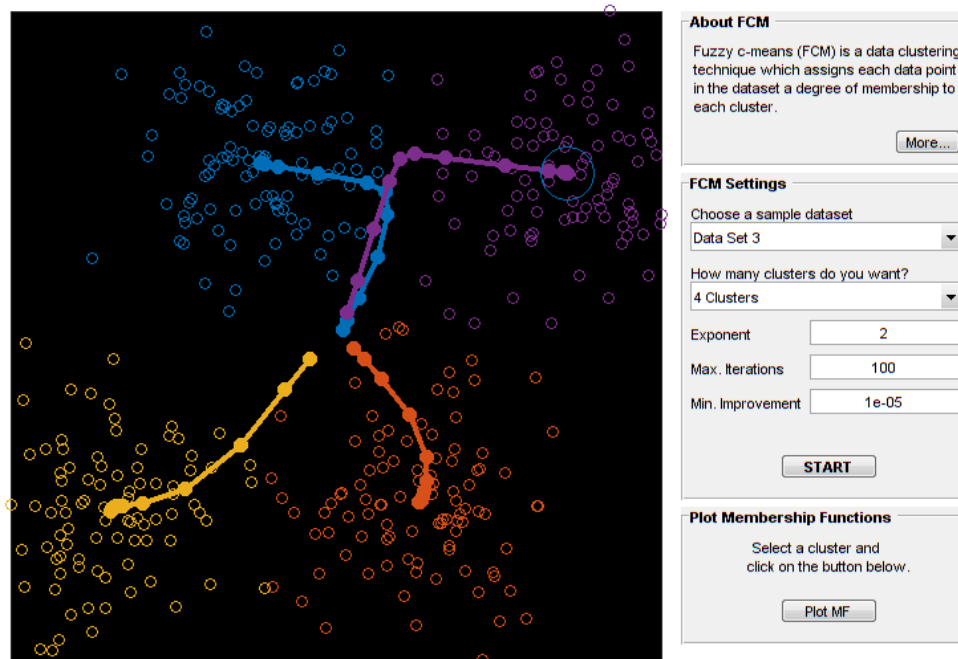
```
sum(U, 1)
```

```
ans = 1×399
```

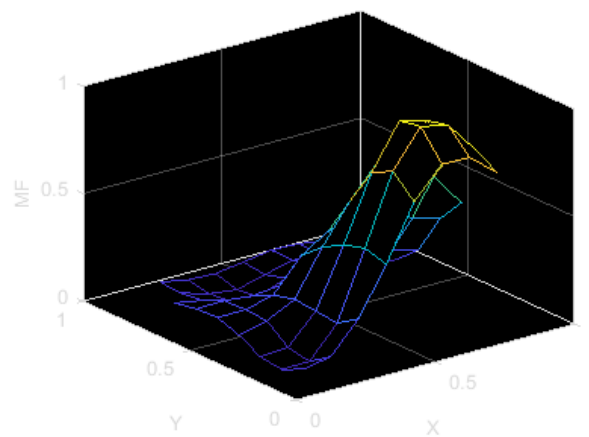
```
1      1      1      1      1      1 ...
```

Practices:

Practice 1



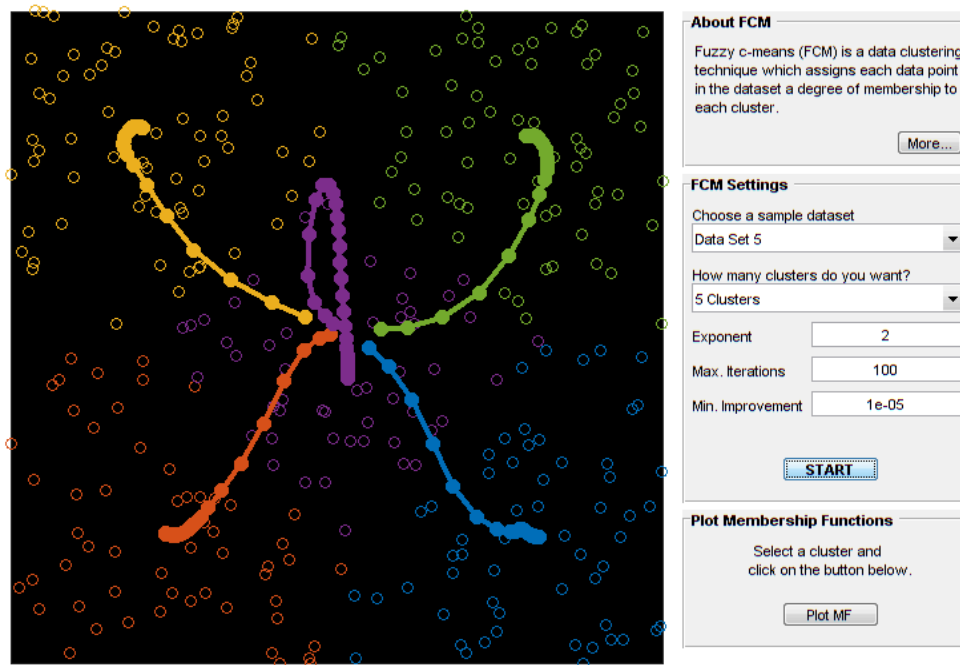
Matlab fcm toolbox for dataset 3 with 4 cluster



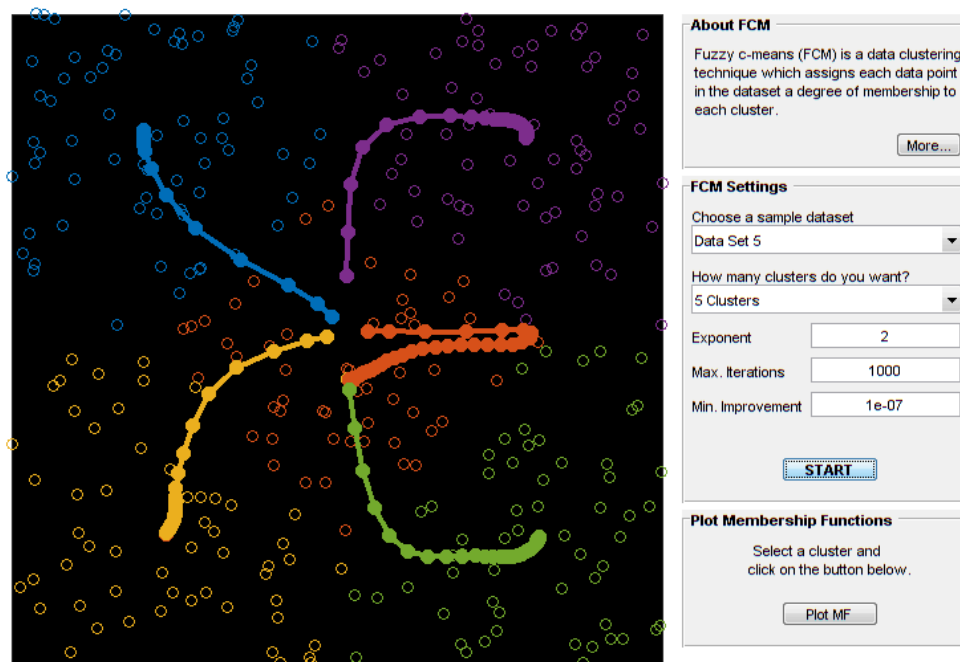
MF plot for cluster 2 (orange)

Practice 2

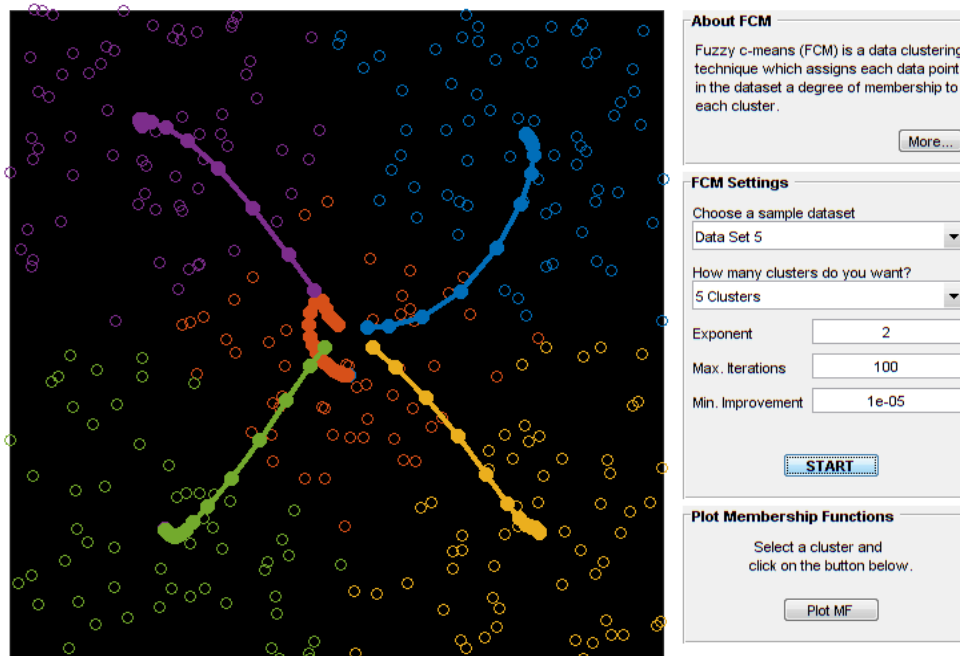
Matlab fcm toolbox for dataset 5 (for 3 different times)



Matlab fcm toolbox for dataset 5 with 5 cluster (number1)



Matlab fcm toolbox for dataset 5 with 5 cluster, 1000 Max. Iterations and 1e-07 Min. Improvement (number2)



Matlab fcm toolbox for dataset 5 with 5 cluster (number3)

Final comments

در ابتدای گزارش موارد مربوط به تئوری الگوریتم آورده شده است تعریف پارامتر ها فرمول ها و الگوریتم مینی مم کردن تابع هزینه نیز در این بخش آمده است

برای پیاده سازی با استفاده از بخش ایمپورت متلب اسکریپت نوشته شده و سپس ستون جنسیت به عدد تبدیل می شود در نهایت میانگین و انحراف از معیار داده های هر ستون محاسبه شده سپس داده های هر ستون با مقادیر یکسان میانگین و انحراف از معیار نرمالایز شده اند، پس از اجرای الگوریتم و پیدا شدن مراکز کلاستر ها و جدول تابع عضویت برای هر نمونه مقدار مراکز دسته ها با ضرب در انحراف معیار و سپس جمع با میانگین به مقادیر اصلی خود بازگردانی شده اند

تمرین ها نیز با استفاده از تولباکس مربوطه پیاده شده اند. در تمرین 2 برای دیتاست شماره 5 با استفاده از 5 کلاستر چندین بار فرمان شروع و اجرای الگوریتم زده شده است ، با اینکه در ابتدا خانه ها از مقادیر متفاوتی شروع به اجرای الگوریتم می کنند اما با این حال در نهایت مراکز دسته ها به مکان یکسانی به صورت دقیق همگرا می شوند که اگر چندین بار نیز ران شود مراکز به همین مقدار همگرا می شوند با عوض کردن تعداد ایتريشن ها و حداقل اختلاف برای توقف الگوریتم نیز صرفا این مقادیر دقیق تر می شوند و گر نه نقطه ی پایانی برای مراکز دسته ها در نهایت یکتا خواهند بود، همچنین داده هایی که بیشترین تعلق را به این دسته ها دارند نیز مشاهده می شود که در هر بار تکرار الگوریتم یکسان می باشند

اگر فرض کنیم الگوریتم برای داده های ما در نهایت همگرا باشد، می توان گفت به ازای فرض اولیه ی رندوم مراکز دسته ها در نهایت آنها به یک نقطه همگرا می شوند چرا که تابع هزینه در یک نقطه به مقدار کمینه خود می رسد و این نقطه کمینه همواره نشان دهنده ی مراکز دسته ها می باشد