BOSTON UNIVERSITY GRADUATE SCHOOL

COLLEGE OF ENGINEERING

EC503 Fuzzy C-means Clustering Algorithm

Xiang Liu U46406505 **Lingyan Jiang U10514676**

1. Introduction

In this project, we are learning the fuzzy C-means algorithm(FCM) as a method to cluster the different data sets and comparing FCM with the K-means algorithm. We use the number of iterations and the purity function to evaluate the difference between the two algorithms. In order to show the FCM algorithm more specifically, we will implement image segmentation on a picture via FCM. In order to compare FCM with K-means, we implement FCM and K-means in three different data sets to find out what kinds of data performed well on FCM.Meanwhile, we use MNIST dataset to show the purity and max iteration times with different k-clusters on FCM and K-means.

datasets we used:

ds9 dataset: http://cs.joensuu.fi/sipu/datasets/

pathbased2 dataset: http://cs.joensuu.fi/sipu/datasets/

MNIST dataset: http://cs.joensuu.fi/sipu/datasets/

regular_data dataset: generate by ourselves (generate_dataset.m)

2. Fuzzy C-means Clustering Algorithm [1]

Fuzzy C-means Clustering method is a clustering method, like K-means. FCM was developed by Dunn in 1973. In hard clustering methods, like K-means, each data belongs to only one cluster. But in FCM, each data point can belong to more than one cluster. For example, in Fig1. We can easily see there are two clusters and an outlier. We're not sure which cluster this outlier belongs to. If we use K-means, we will get a label for this outlier. The result is not accurate. But if we use FCM, we may get a vector [0.4504,0.5496], which means that this point belongs to cluster 1 with the 0.4504 and cluster 2 with 0.5496. The vector is called the degree of membership.

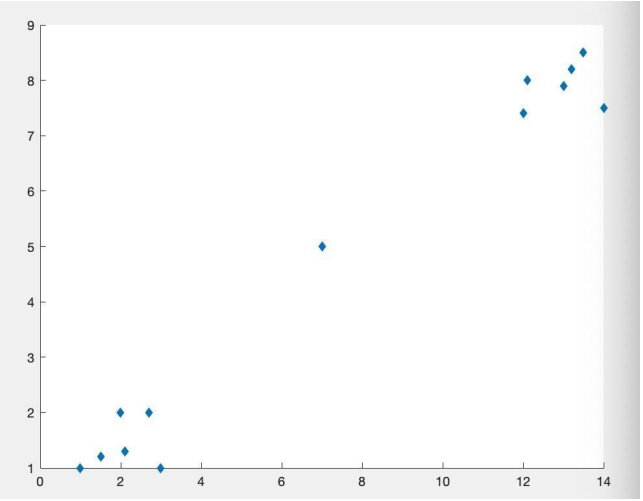

Fig.1 dataset example with outlier

Next, we will show how to calculate the degree of membership.

The goal of the method is to minimize the objective function:

$$J_m = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} \| x_j - c_i \|^2 \qquad 1 \le m < \infty \qquad (1)$$

where *m* is any real number greater than 1, u_{ij} is the degree of membership of x_j in the cluster *j*, x_j is the *j* th of d-dimensional measured data, c_i is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center.

This optimization problem has a constraint: $\sum_{i=1}^{c} u_{ij} = 1, \quad j = 1, 2, \ldots, n \qquad (2)$

In order to solve this optimization problem, we should get its Lagrangian Function and set the first derivative for u_{ij} equal to zero.

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} \| x_j - c_i \|^2 + \lambda_1 \left(\sum_{i=1}^{C} u_{i1} - 1 \right) + \lambda_2 \left(\sum_{i=1}^{C} u_{i2} - 1 \right) + \ldots + \lambda_n \left(\sum_{i=1}^{C} u_{in} - 1 \right) \qquad (3)$$

$$\frac{\partial J}{\partial u_{ij}} = m \| x_j - c_i \|^2 u_{ij}^{m-1} + \lambda_j = 0 \qquad (4)$$

Now we can get:

$$u_{ij}^{m-1} = \frac{-\lambda_j}{m \| x_j - c_i \|^2} \qquad (5)$$

We need to replace λ_j , we use (2) again. Then, we can get the matrix U.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\| x_j - c_i \|}{\| x_j - c_k \|} \right)^{\left(\frac{2}{m-1} \right)}} \qquad (6)$$

We calculate the first derivative for c_i and set it to zero, we can get c_i:

$$c_i = \frac{\sum_{j=1}^{n} \left(x_j u_{ij}^m \right)}{\sum_{j=1}^{n} u_{ij}^m} \quad (7)$$

Note: Because of lack of experience and knowledge, people do not know how to choose m. m is commonly set to two. We will use 2 in our algorithm.

The algorithm is composed of the following steps:

FCM Algorithm

Step 1: Initialize U

Step 2: for i = 1:k

Calculate c_i: $$c_i = \frac{\sum_{j=1}^{n} \left(x_j u_{ij}^m \right)}{\sum_{j=1}^{n} u_{ij}^m}$$

Step 3: Update U^k:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\| x_j - c_i \|}{\| x_j - c_k \|} \right)^{\left(\frac{2}{m-1} \right)}}$$

Step 4: If $\| U^k - U^{k-1} \|^2 < \varepsilon$, then STOP; otherwise go to Step 2.

3. Implement: Image Segmentation

Like K-means, we can use FCM to do image segmentation. We input an image and use K clusters to segment this image. Below are the original image and the images produced by FCM in Fig.2:

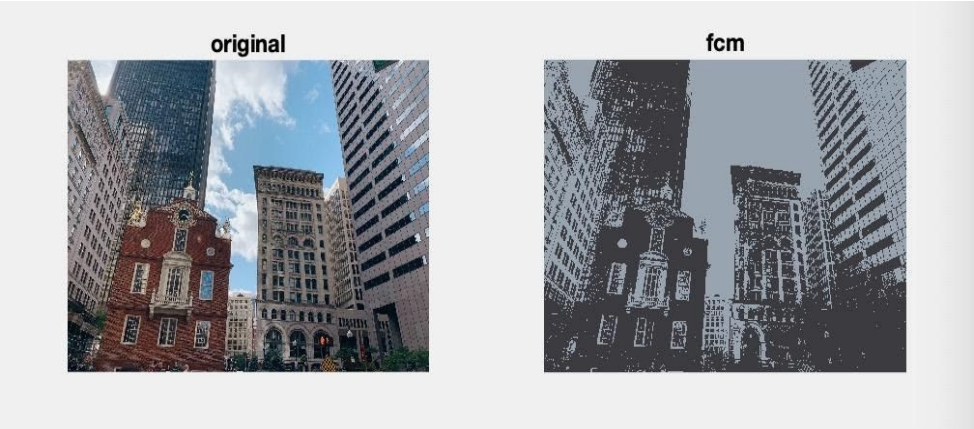

Fig.2(a) image segmentation result of Boston downtown with the number of clusters equals to 2

Fig.2(b) image segmentation result of Boston downtown with the number of clusters equals to 4

Fig.2(c) image segmentation result of Boston downtown with the number of clusters equals to 8

We can see from Fig.2(a)-(c), as the number of clusters increases, the image becomes more colorful. If we only use 2 clusters, in Fig.2(a), the image becomes a black and white picture.

4. Compared Fuzzy C-means Clustering with K-means

K-means computes cluster centroids for each distance measure in order to minimize the sum with respect to the specified measure. So K-means algorithm aims at minimizing an objective function known as squared error function. We have learned the K-means algorithm in the class. We want to focus on the difference between FCM and K-means in the shape of the dataset iteration time and purity. We will use different datasets to cluster and see the results.

a) Compared FCM with K-means on irregular shaped clustered dataset

We compare FCM with K-means in regular shaped clustered data. This dataset is generated by ourselves in generate_dataset.m function. As we can see in Fig.3, FCM is doing well in this clustering while K-means is not. In the right image of Fig.3, the data in the middle of the bottom is misclassified and the same situation in the top right corner dataset. FCM is suitable for the regular pattern data. Shape of clusters is an important factor in choosing an appropriate clustering algorithm. FCM algorithm was suitable for the regular pattern, particularly is good for circular and rectangular clustered data .

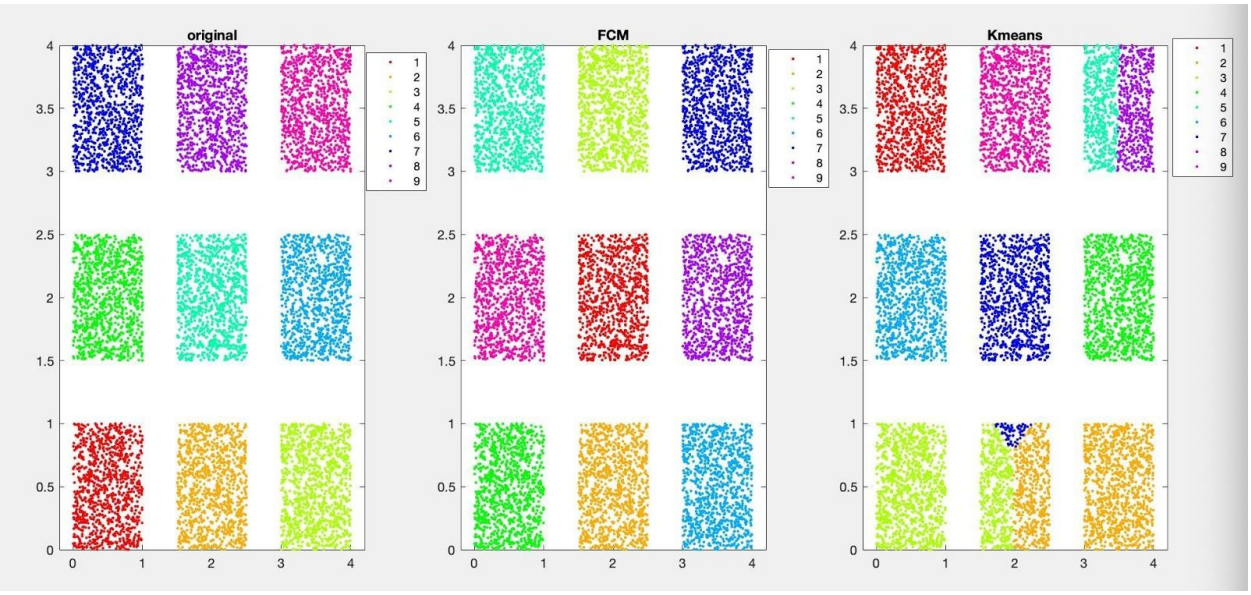

Fig.3 K-means and FCM result of regular shaped clustered data

b) Compared FCM with K-means on irregular shaped clustered dataset

For irregular shaped clustered dataset, we use the dataset from http://cs.joensuu.fi/sipu/datasets/ and name it as ds9.mat . We input this dataset into FCM and K-means. We plot the original dataset and the output of FCM and K-means. In Fig.4(a), the first figure is the original dataset. The second figure is the cluster via FCM. The third figure is the cluster via K-means. It shows clearly that both FCM and k-means are not doing so well on this

irregular dataset. The bottom clustered dataset is mixed with blue and red labels, although both purity is over 0.84(see in Fig.4(b)) in this case and every similarity to each other.

This shows clearer in Fig.5 that both FCM and K-means is not very suitable for irregular shaped data. As we can see in Fig.5(a) that the circle plotted dataset has three color labels in one class. And in this case, the purity is very different. For the purity of FCM is at about 0.3 and K-means is way higher than that.

Although the purities have large differences, the classification is not as good as what we have seen. Both clusterings of FCM and K-means are not clear classification. Consequently, we understand that K-means and FCM algorithms are not good clustering options to partition datasets containing nested clusters scattering with irregular patterns as seen in Fig.4 and Fig.5.

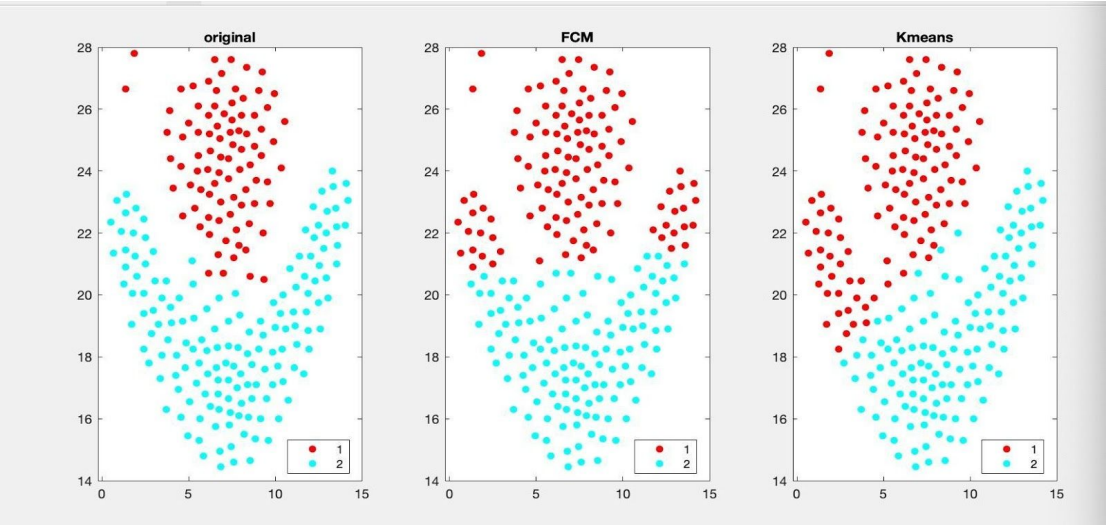

Fig.4(a) K-means and FCM plotting of ds9 dataset

```
fcm_purity =

    0.8500


kmeans_purity =

    0.8417
```

Fig.4(b) K-means and FCM purity of ds9 dataset

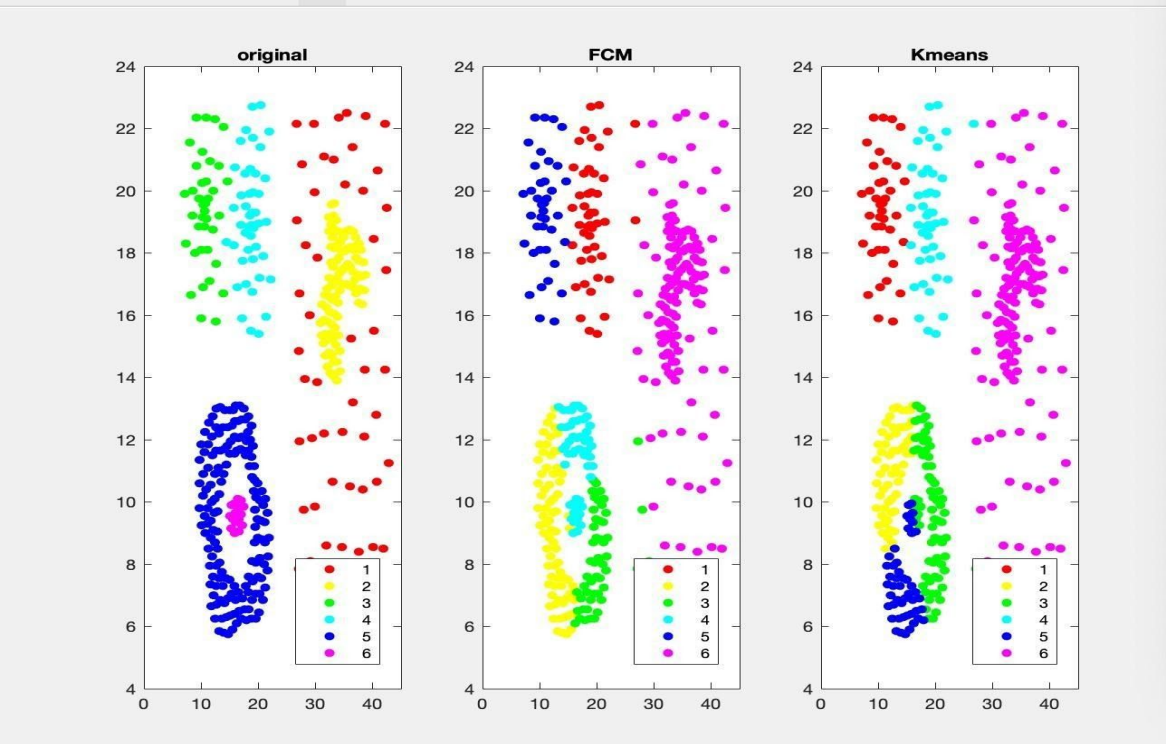

Fig.5(a) K-means and FCM plotting of pathbased2 dataset

```
>> eg_pathbased2_plot

fcm_purity =

    0.2932


kmeans_purity =

    0.3860
```

Fig.5(b) K-means and FCM purity of pathbased2 dataset

c) Compared FCM with K-means on iteration times

We use MNIST dataset in K-means and FCM with k=2, k = 4, k = 6, k = 8, k = 10, k = 12, and k = 14 to compare the iteration times. The Fig.6 is the iteration time for FCM and K-means. For every k, we found that FCM needs more iteration times than K-means and K-means is faster than FCM.

```
iterations =

    12    14    16    11    14     9    10
```

Fig.6(a) K-means iteration times on MNIST dataset

```
iterations =

    82    92    65    94    74    89    67
```

Fig.6(b) FCM iteration times on MNIST dataset

d) Compared FCM with K-means on purity

We also use MNIST dataset in both algorithms with k=2, k = 4, k = 6, k = 8, k = 10, k = 12, and k = 14. As we see in Fig.7 below, both have significant increases when k=8. K-means has more obvious fluctuations in the change of purity than FCM, which is easy to observe. But as I said before, we can't take purity as the only judgement. FCM's purity is also related to its data cluster shape. We also can take error rate, impact of density, impact of noise into consideration.

```
average_purity_fcm =

    0.2063    0.2402    0.2450    0.2561    0.3033    0.2486    0.2621
```

Fig.7(b) FCM purity on MNIST dataset

```
average_purity_kmeans =

    0.2096    0.3795    0.4812    0.5137    0.5898    0.6439    0.8322
```

Fig.7(b) FCM purity on MNIST dataset

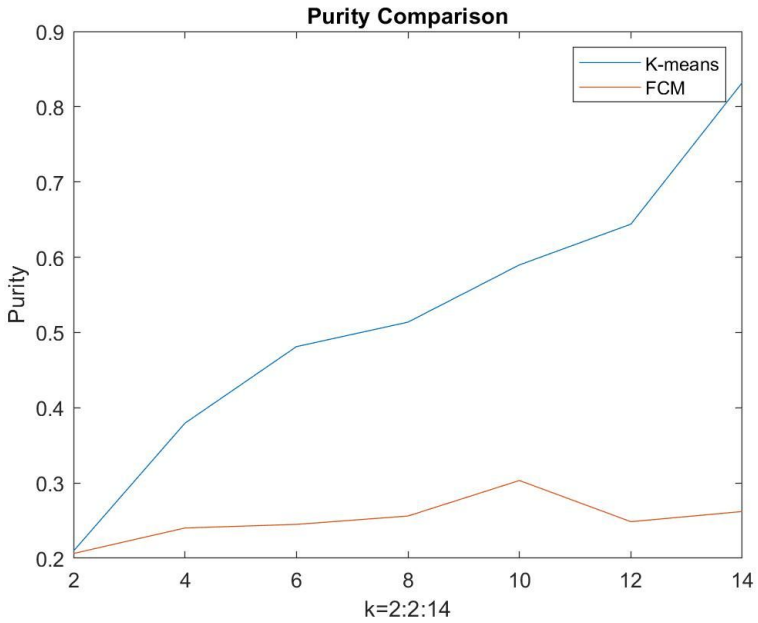

Fig.7(c) FCM and K-means purity comparision

5. Conclusion

In conclusion, the shape of clusters is an important factor in choosing an appropriate clustering algorithm. Both K-means and FCM are not doing so well in an irregular dataset. Neither K-means nor FCM were successful to find the concave and other kinds of arbitrary shaped clusters when they are not well separated. However, their performances were better for regular dataset, especially for the circular and rectangular clusters. What's more, K-means was always extremely faster than FCM in all datasets containing the clusters scattering in regular or irregular patterns. FCM is an algorithm based on more iterative fuzzy calculations and that is the reason that its execution time was found comparatively higher than K-means. Purity is always a good criterion to define whether the clustering algorithm is doing well or not. But in this case, we have seen that we can not take purity as the only judgement. FCM's purity is also related to its data cluster shape. We also need to take error rate, impact of density, impact of noise into consideration.

As reported in many studies, while FCM will give better results for noisy clustered datasets, K-means will be a good choice for large datasets because of its execution speed. Thus, the use of K-means should be a good starting point for large datasets due to its fast execution time. Besides, from what we researched on the article of UCI repository, said that FCM is better than KM in the accuracy of clusters on the diabetes dataset and also on medical and genetic dataset.

As a final conclusion, there is no algorithm which is the best for all cases. There is always a tradeoff. Thus, the datasets should be carefully examined for shapes and scatter of clusters in order to decide for a suitable algorithm.

6. References

[1] A Tutorial on Clustering Algorithms. (n.d.). Retrieved April 17, 2020, from https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html

7. Github

https://github.com/xiangliu9701/EC503-final-project