Fuzzy C-means Algorithm Implementation

Yacine FLICI M2 VMI FA

1 Introduction

The goal of this TP was to implement the Fuzzy C-means (FCM) algorithm and apply it to image segmentation, specifically to partition an image into clusters based on pixel values. The FCM algorithm is an unsupervised clustering method that allows each data point (or pixel, in the case of images) to belong to multiple clusters with varying degrees of membership. **GitHub Repository**

2 Fuzzy C-means Algorithm Overview

The Fuzzy C-means algorithm assigns each point a membership value indicating how strongly it belongs to each cluster. Unlike hard clustering algorithms like K-means, which assign each point to exactly one cluster, FCM allows for soft clustering.

3 Algorithm Steps

3.1 Initialization

We randomly initialize the membership matrix U such that each entry u_{ij} represents the membership degree of data point x_i in cluster j.

$$\sum_{j=1}^{C} u_{ij} = 1$$

```
init_centroids() #refer to function
```

3.2 Cluster Centroid Calculation

After the initialization of membership values, the next step is to calculate the cluster centroids using the formula:

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

- m is the fuzziness degree, C_j is the cluster index number j, u_{ij} is the membership degree of the pixel i to the cluster j, x_i is the pixel value index i.

```
update_centroids() #refer to function
```

3.3 Update Membership Values

After computing the centroids, the membership values are updated based on the relative distances between each data point and the centroids of the clusters:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{||x_i - c_j||}{||x_i - c_k||} \right)^{\frac{2}{m-1}}}$$

```
update() #refer to function
```

3.4 Convergence Check

The algorithm checks for convergence by comparing the change in the iterations U between iterations. If below a certain ϵ the algorithm stops.

$$||U^{(k+1)} - U^{(k)}|| < \epsilon$$

```
is_converged() #refer to function
```

3.5 Result

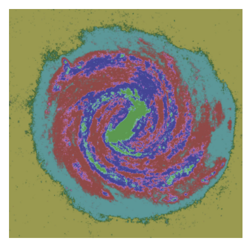
Once the algorithm converges, the final membership values can be used to assign each data point to a cluster. In the case of image segmentation, each pixel is assigned to the cluster for which it has the highest membership value.

4 Application to Image Segmentation

I implemented Fuzzy C-means in Python that take either RGB or Gray. The image was resized to 100x100 pixels to reduce computation time.

- The FCM algorithm was applied to an RGB image to segment it into 5 clusters (i.e., $K = 5$). Each cluster corresponds to a distinct region in the image, based on the pixel intensity and color values.
- It successfully identified a notable feature: the tornado in the middle of the image was clearly segmented into its own cluster.
- Although the clustering algorithm was trained on a version of the image with 10x lower quality, the clustering results on the high-quality image still performed well.

(a) Original Image



(b) Clustered RGB Image with $K = 5$

Figure 1: Comparison of Original Image and Clustered Image

- A heatmap was generated for each cluster to visualize the membership values of each pixel. The heatmap clearly illustrates the distribution of pixels across clusters, with distinct colors representing varying degrees of membership. An example of the generated heatmap is shown in Figure 2.

5 Conclusion

The Fuzzy C-means algorithm allows for soft clustering, where each pixel belongs to multiple clusters with varying degrees of membership. This provides more flexibility in image segmentation tasks where boundaries between regions are not sharp.

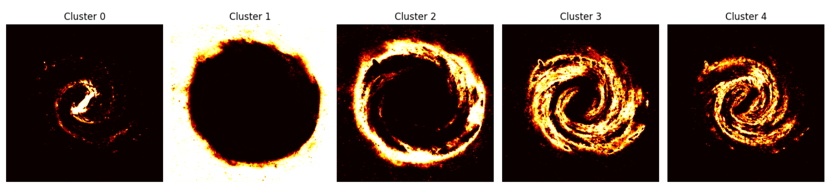


Figure 2: Heatmap visualization of pixel membership for each cluster