PROJECT#1 Write-up

Name: Srinivas Ganesan
Class: MA 493
Professor: Dr. Mansoor Haider
Date: 03/09/2023

Part I): Comparing k++ Initialization and Random Initialization.

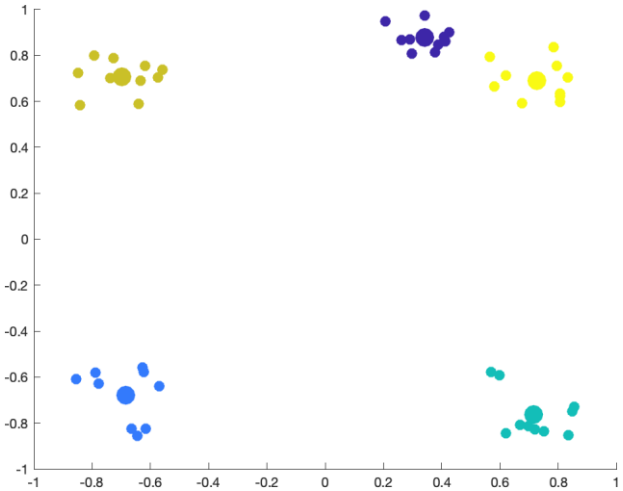
Description: In this part, we are a required to write a script for performing k-means clustering on a given data set(accessed using the file Q1data.mat) using two different types of cluster initializations – k++ initialization, Random initialization. We are required to run 10 realizations of the clustering for each initialization above with k = 5 and compare the performance of the two initialization schemes. Then, we must determine which initialization yields better performance and explain why we think this occurs.

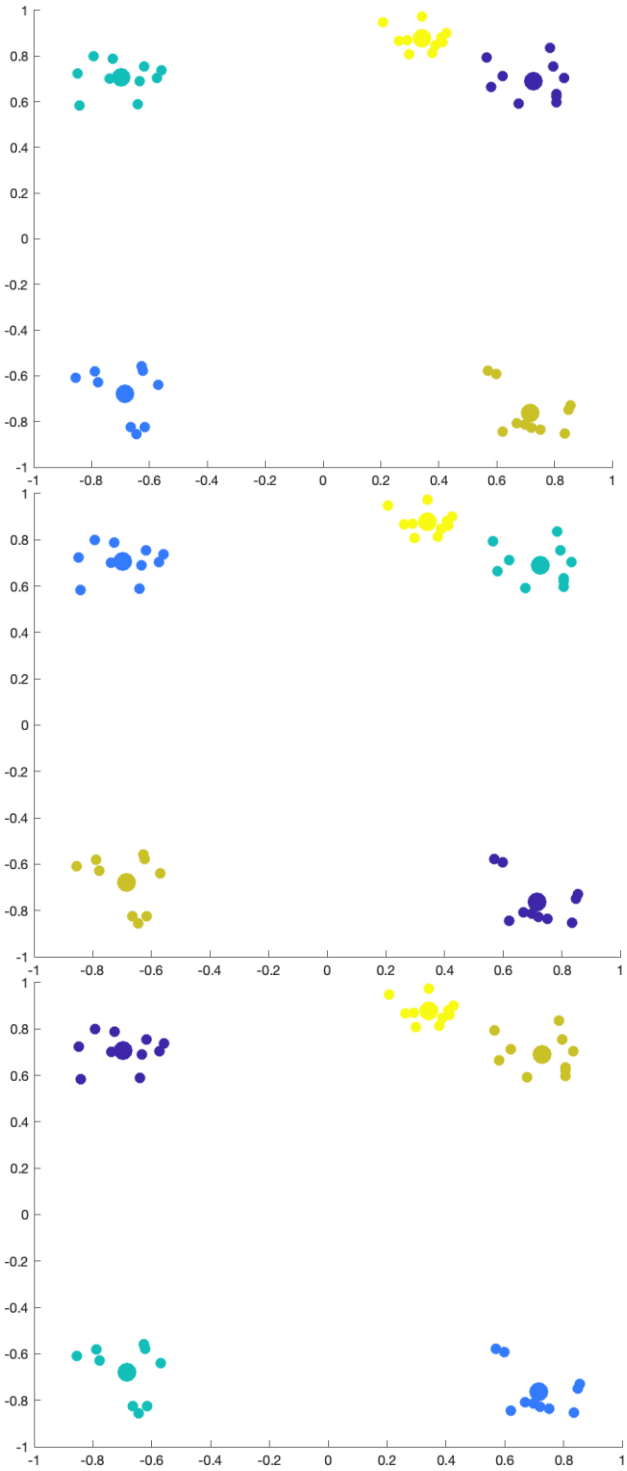
Scripts written by me:

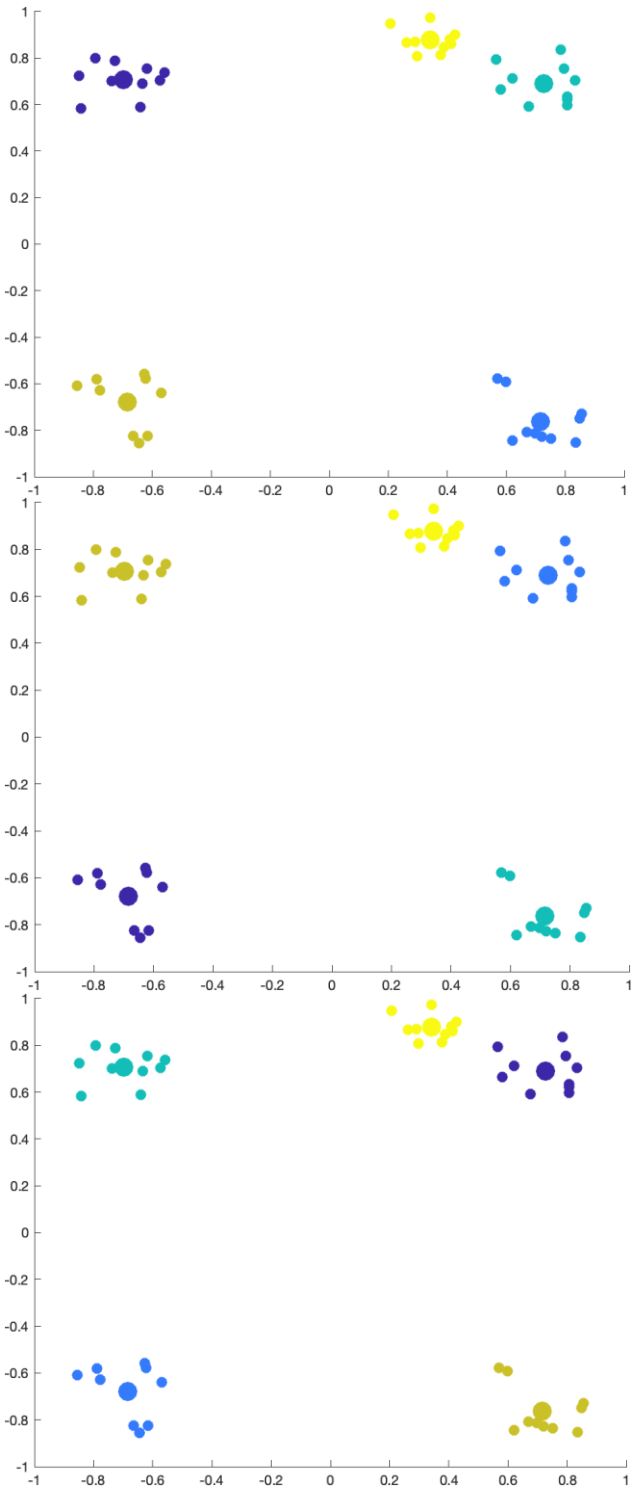
- ma493_proj1_part1.m(Main script): Performs kmeans clustering with both initialization schemes.
- Part1_a_random_init.m : Contains the Random Initialization function that is used in the Main script.
- Part1_b_kplusplus_init.m : Contains the k++ Initialization function that is used in the Main script.

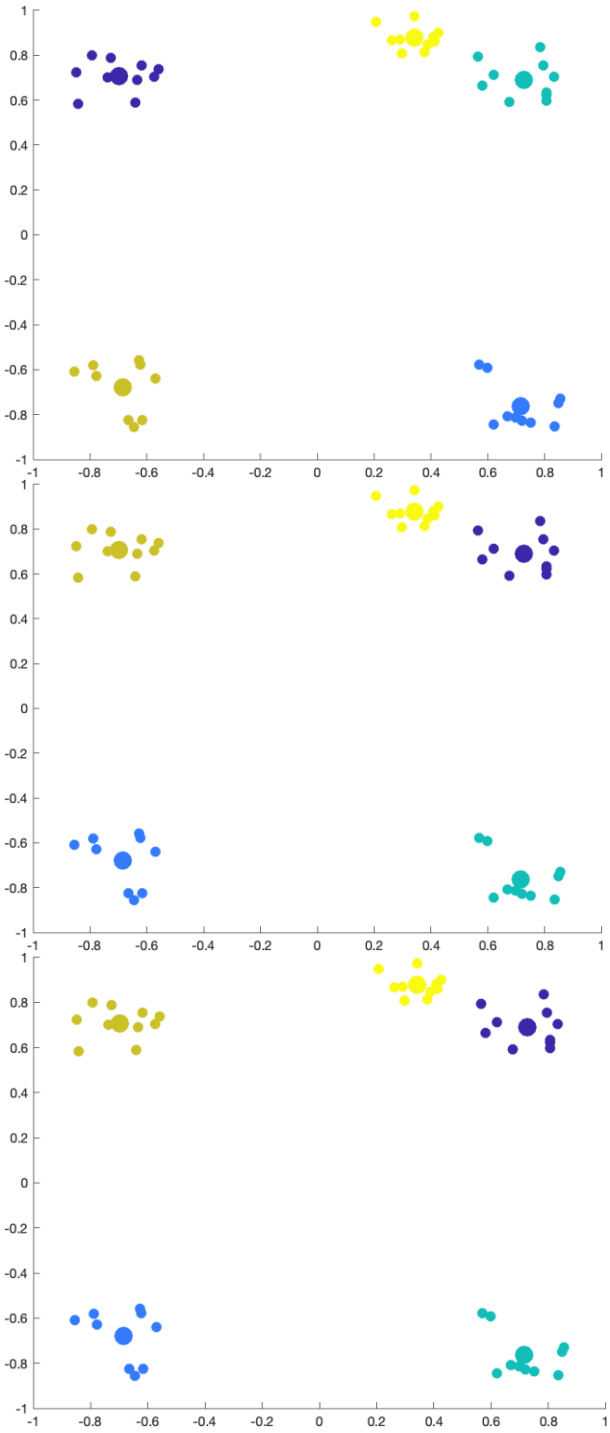
Plots and their Analysis:

1. Plots of 10 realizations of the k++ initialization for k=5:

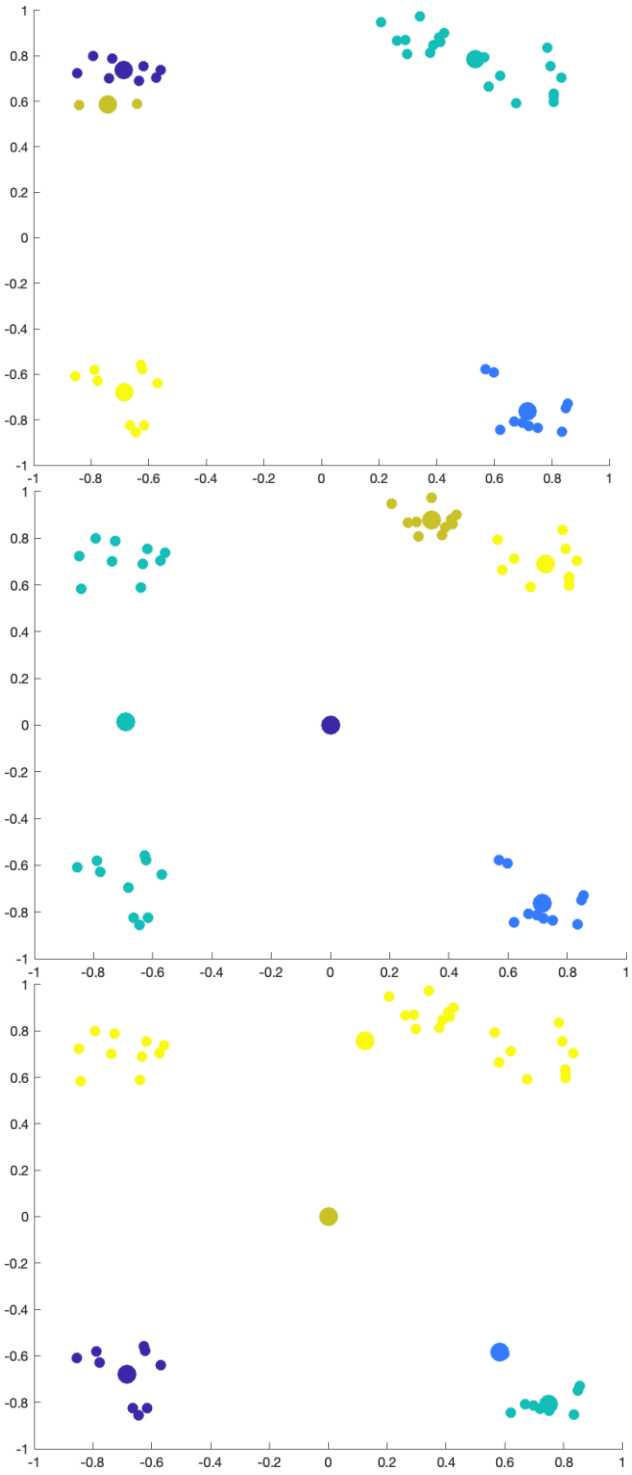


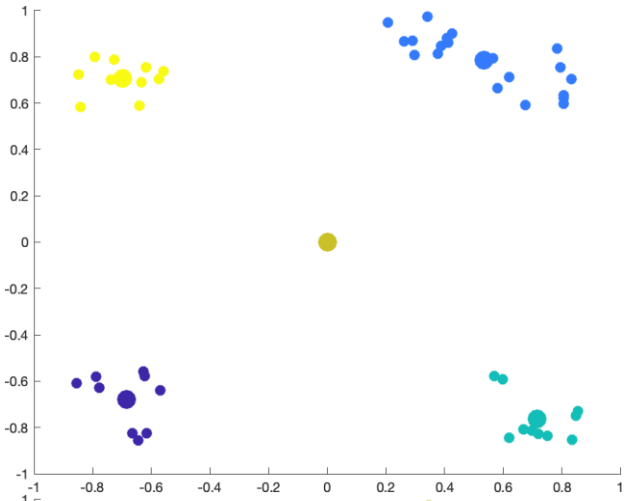

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

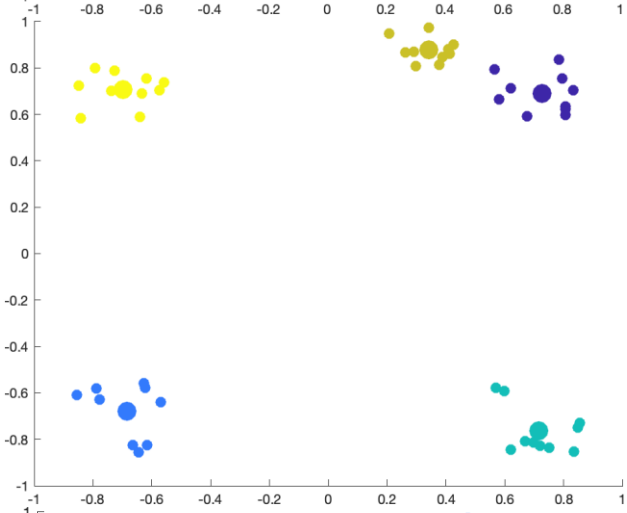

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1

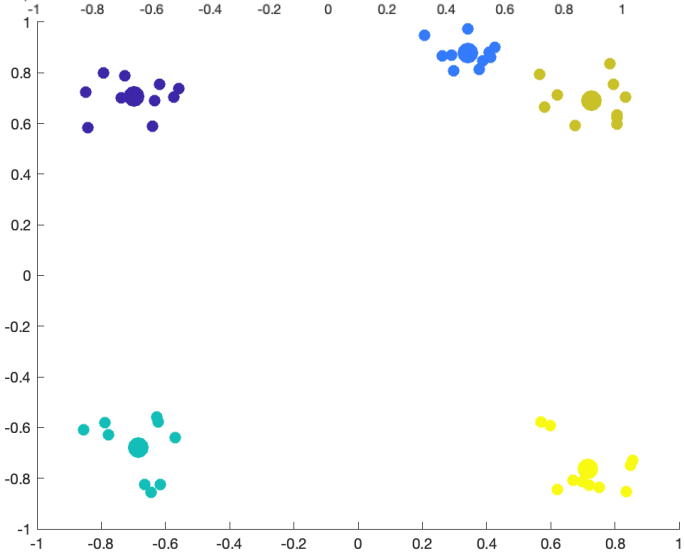


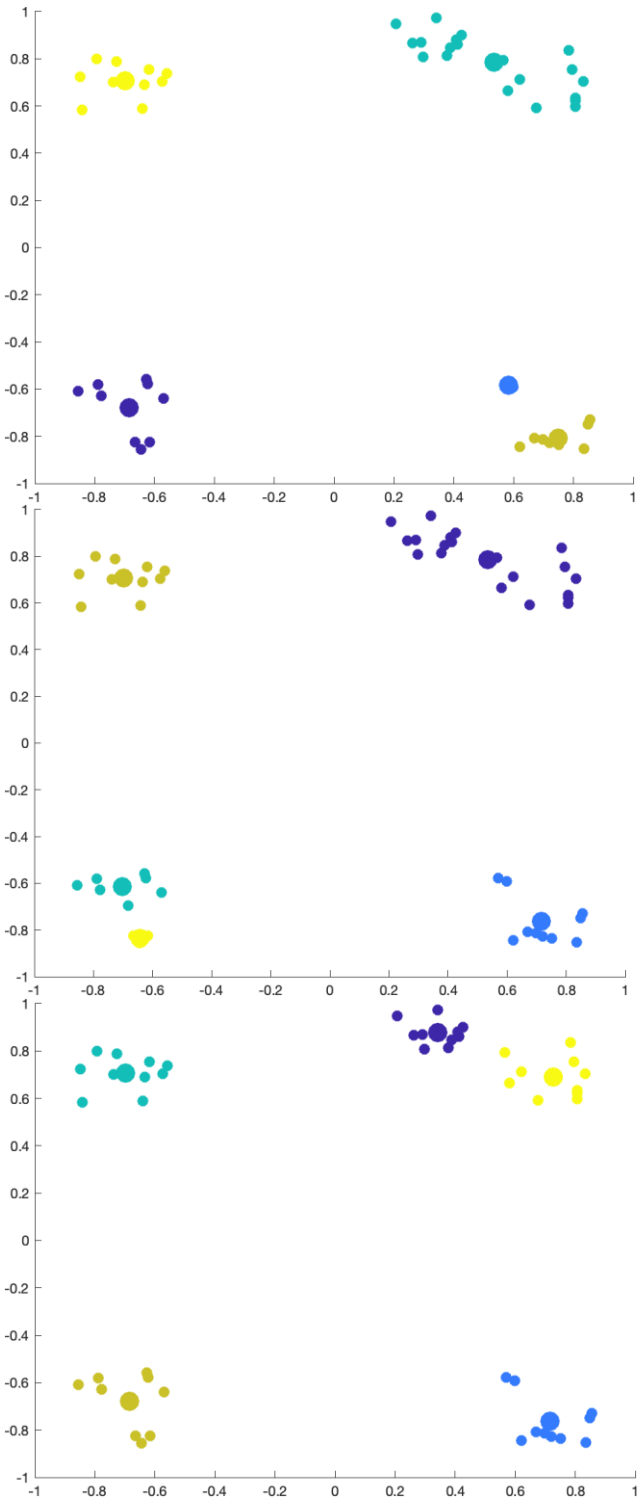
2. Plots of 10 realizations of the Random initialization for k=5:

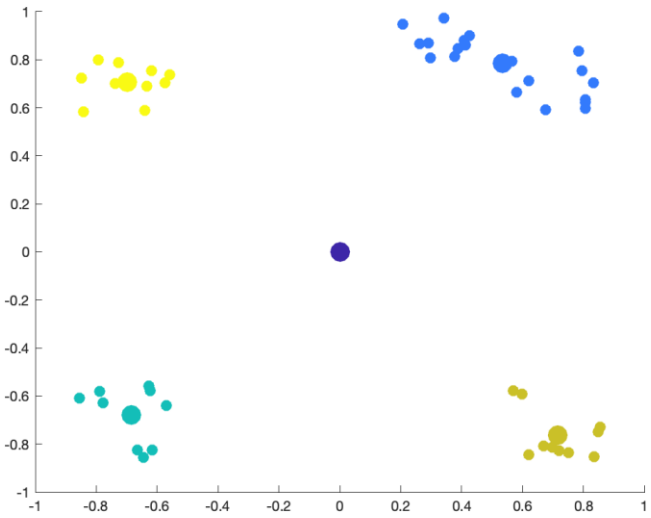

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



3. Analysis of the above plots:

We know that the performance of each initialization scheme can be assessed using the Overall Coherence value(which is obtained from the cluster coherence value) in each clustering plot.

- Cluster coherence value = $q_l = \sum_{j \in I_l} \| x_j - c_l \|^2$, $l = 1,2 \ldots, k$ where I_l denotes the index set for each cluster l.
- Overall Coherence = Q = $\sum_{i=1}^{k} q_i$

Comparing the clustering plots after performing k++ initialization and Random initialization, we observe that:

In Random initialization,

- 8 out of 10 clustering plots are very different from each other. This implies that Random initialization method is very sensitive to the initialization of its first cluster representative.
- Clusters are not far apart from each other.
- The cluster coherence value is very high among some clusters in certain clustering plots.

In k++ initialization,

- The clustering plots of k++ initialization are very similar to each other. Therefore, sensitivity to the initialization of the first cluster representative is low.
- Clusters are widely spread out.
- All the clusters in each clustering plot have low cluster coherence value.

From the above points we can reach the conclusion that,

- All the clustering plots of k++ initialization are highly coherent(tightly knit) clusters, whereas most of the clustering plots of Random initialization have clusters that are much lesser coherent. This implies that the overall coherence of the clustering is higher with k++ initialization than with Random initialization, which leads us to conclude that k++ initialization is better than Random initialization.

Part II): Significance of the Elbow Method.

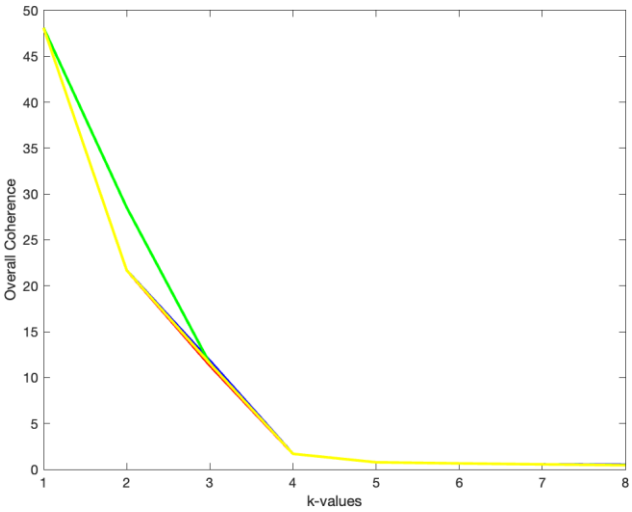
Description: In this part, we are a required to extend the main script from Part I). The aim is to create two elbow method graphs, one for each type of initialization scheme for k values ranging from 1 to 8, with each graph containing 5 curves(1 for each realization). Then, we need to evaluate the graphs and determine the "best" choice of k for each type of initialization. We also need to discuss how certain this choice is and how it relates to the number of clusters in the data set based on visual inspection of the data set.

Scripts modified by me:

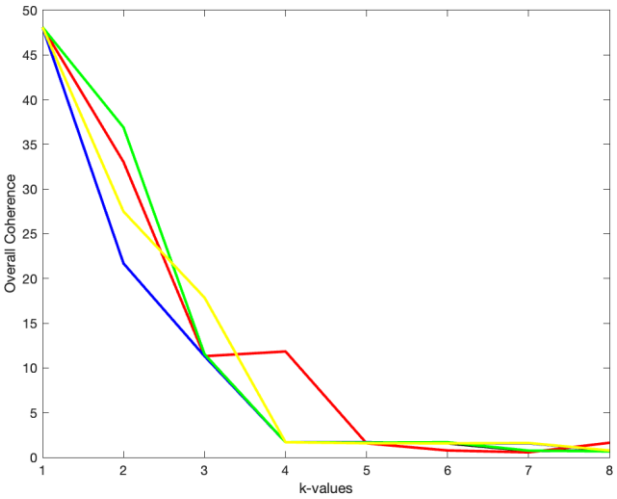
- ma493_proj1_part2.m(Main script): Performs kmeans clustering with both initialization schemes and provided the elbow plot for each initialization scheme.
- Part1_a_random_init.m : Contains the Random Initialization function that is used in the Main script.
- Part1_b_kplusplus_init.m : Contains the k++ Initialization function that is used in the Main script.

Plots and their Analysis:

1. Elbow plot for clustering with k++ initialization:



2. Elbow plot for clustering with Random initialization:



3. Analysis of the plots:

By observing the elbow plots of both initialization schemes we can conclude that,

For k++ initialization,
The elbow plot suggests k=4 as the “best” k value(k=4 is where the slope of the graph changes drastically). This is consistent with the clustering plots of k++ initialization shown in Part I). In the Upper-right hand corner of each clustering plot, we observe two clusters that are very close to each other. If we reduce the number of cluster representatives in this region from 2 to 1, the overall coherence doesn’t increase much(as observed from the elbow plot). Therefore, k=4 is the ideal choice for the k value when using k++ initialization.

For Random initialization,

Although each realization produces slightly different elbow plots, the elbow graph suggests k=4 as the "best" k value. This k value is consistent with most of the clustering plots of Random initialization shown in Part I). We can observe that a few clustering plots even have a cluster in the middle of the plot with no cluster elements, and four other filled clusters. This implies that k=4 is the ideal choice for the k value when using Random initialization.

Part III): Using kmeans clustering with k++ initialization on the MNIST dataset(Handwritten Digits).

Description: In this part, we are a required to use the main script from Part I) to perform kmeans clustering on the MNIST dataset. The aim is to cluster the first 100 images from the MNIST data set with k = 3, 4, 5, 6, 7, 8, 9, 10. We also need to create an elbow plot to determine the "best" k value for clustering. In addition, we need to determine the Success Score S (out of 100) for the clustering by comparing information in the Labels for each image in the MNIST dataset to the cluster representative for each of the 100 images (patterns).

Scripts modified by me:

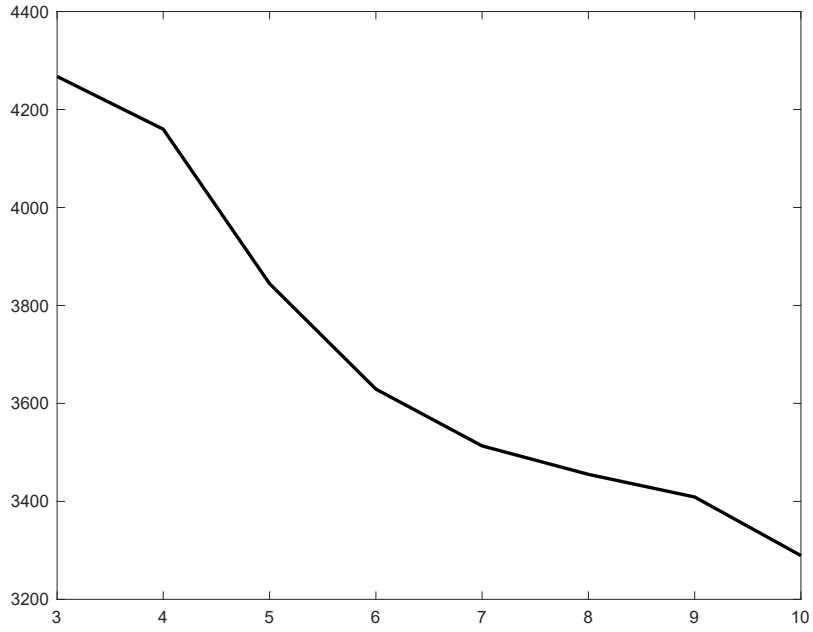
- DigitsImagesPrimer.m(Main script): Converts images in the MNIST dataset to vectors that can be used for clustering, performs kmeans clustering with k++ initialization, and provides the elbow plot along with the Success Score S for clustering.

Scripts modified by me:

- readMNIST.m: Read digits and labels from raw MNIST data files.

1. Outputs and their significance:

    - Elbow plot:

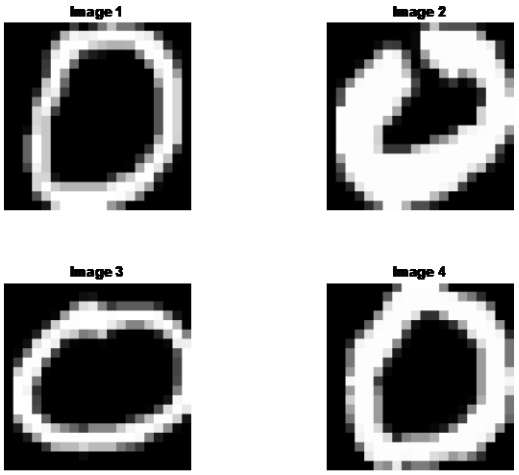


Significance: The above plot is used to determine the "best" k value for clustering. From the above plot, we can observe that the slope of the line joining k=6 and k=7 is very different from to the slope of the line joining k=4 and k=5. Therefore, we can conclude that k=6(or even k=7) is the "best" k value for clustering the images in the MNIST dataset.

- Example of a Good Cluster when k=6(This is the "best k value according to the elbow plot):

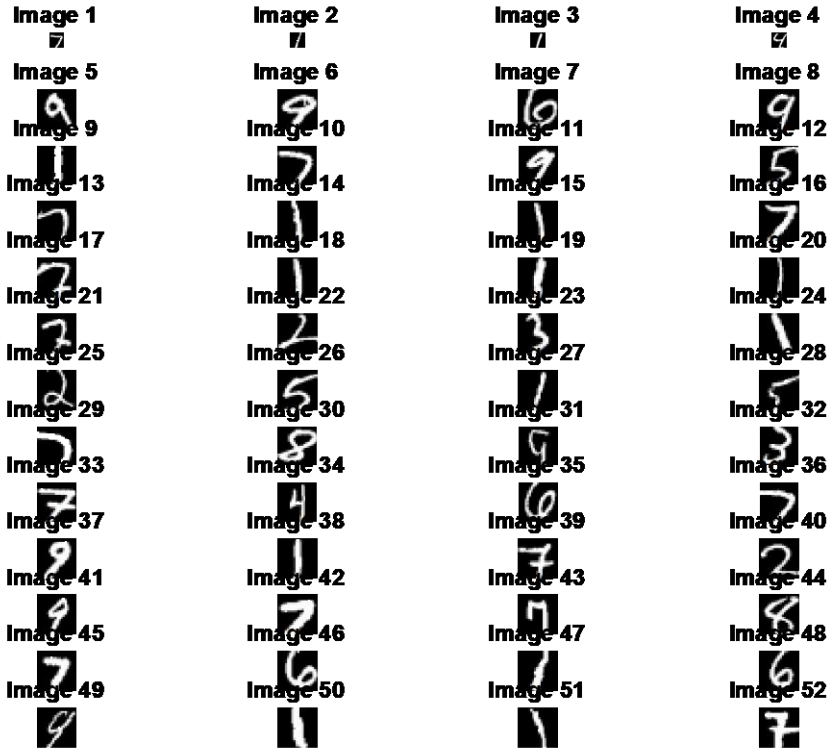
Significance: The above figure contains all the elements of a specific cluster after performing clustering with k++ initialization with k=6 on the MNIST dataset. This cluster has the digit ‘4’ as its cluster representative. Out of the 18 elements in this cluster, 11 of them are images of the digit ‘4’, and the others are digits that have a similar digit structure as ‘4’. For example, the digits ‘9’ and ‘6’ have a straight line connected with a circle in them. This structure is like that of the digit ‘4’. The above cluster is a good example of a correct classification of images based on pattern.

- Example of another Good Cluster when k=6(This is the “best k value according to the elbow plot):



Significance: The above figure contains all the elements of a specific cluster after performing clustering with k++ initialization with k=6 on the MNIST dataset. This cluster has the digit ‘0’ as its cluster representative. Out of the 4 elements in this cluster, all the elements are the images of the digit ‘0’. Since the above cluster doesn’t have any mismatched elements in it, it is an example of a perfect cluster.

- Example of a Bad Cluster when k=6(This is the “best k value according to the elbow plot):



Significance: The above figure contains all the elements of a specific cluster after performing clustering with k++ initialization with k=6 on the MNIST dataset. This cluster has the digit '1' as its cluster representative. This is an example of a bad cluster because it contains several digits that are very dissimilar to the cluster representative. For example, the digits – '8', '6','3','2' have a rounder digit structure compared to '1' which comprises of a straight line.

- Vector containing the most repeated element in each cluster when k=6:

  mostrepeated = [1;0;3;2;4;0]

  Significance: Each element of this vector is assigned as a cluster representative.

- Vector containing the frequencies of the most repeated element in each cluster when k=6:

  freq_mostrepeated = [14;4;9;3;11;4]

  Significance: The elements of this vector are added together to give the Success Score S of the clustering.

- Success Score of the clustering when k=6:

  S = Sum of elements in freq_mostrepeated = 45

  Significance: This number denotes the number of images out of the first 100 images in the MNIST dataset that were identical to the cluster representative of the cluster that they were placed in.

2. Discussion of the strengths and limitations of using the above approach in the context of this application:

   Strength: Using Kmeans clustering with k++ initialization is useful in the above approach because it helps cluster images with similar patterns(digit structure). For example, the cluster shown above with the digit '4' as its cluster representative predominantly contains images of the digits '4','9' and '6'. These digits are very similar with respect to the structure of the digits.

   Limitation: Using Success score to determine the efficiency of clustering was not appropriate for the above application. This is because the Success score method is used find out the total number of images(out of 100) that contained digits identical to the cluster representative of the cluster that they were part of. Since the purpose of clustering the images was to identify digits were similar to each other, and not 'identical' to one another, calculating Success Score was not a correct measure of the efficiency of clustering.