Nick Wurzer
V00958569
March 23, 2023
Seng 474, Spring 2023

Assignment 3

Introduction

Two datasets were given in assignment 3, so I first plotted the two datasets to visualize them. Dataset 1, provided in dataset1.csv can be seen in figure 1 below. Similarly, Dataset 2, provided in dataset2.csv can be seen in figure 2 below.

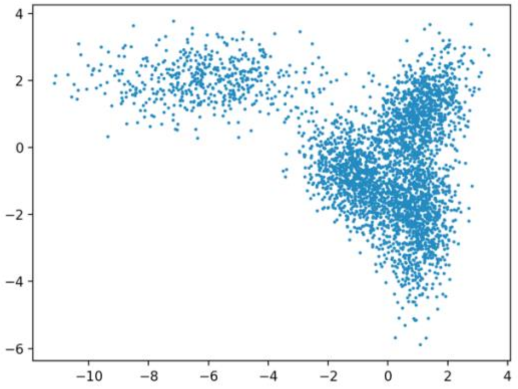

Figure 1 – Dataset 1

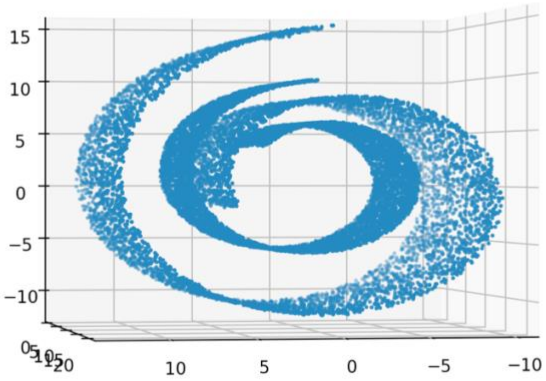

Figure 2 – Dataset 2

Once I had seen these datasets, I mentally made a few predictions about how successful I expected experiments to be on the datasets. In dataset 1, I initially saw two distinct clusters (arguably three, or maybe even four, where the clusters on the right are quite close together). The clusters look approximately normally distributed, so I expected that Lloyd's algorithm and Hierarchical Agglomerative Clustering(HAC) with average linkage to be successful in clustering this dataset. I believed HAC with single linkage would be unsuccessful because there is a bit of a "bridge" between the two clusters. In dataset 2, I believed there to be two distinct clusters. These clusters reminded me of the two moons dataset from class which was successful with HAC with single linkage, but unsuccessful with k-means and HAC with average linkage, so these were my predictions for dataset 2.

Lloyd's Algorithm

Choosing the Number of Clusters

The first step to running Lloyd's algorithm is choosing the number of clusters, or means, to use. This is done by choosing the smallest number of clusters which yields good results. A good result is one that has a small cost. Here the cost is defined as the squared Euclidian distance from the point to the mean of the cluster.

Dataset 1

The cost as a function of the number of clusters for dataset 1 can be seen in figure 3 and figure 4 below, where figure 3 has random initialization and figure 4 has the K-means++ initialization. In both figures the "elbow" of the curve, the point which the curve flattens the most and where further increasing the clusters does not yield a much lower cost, is when the number of clusters is four. This somewhat goes against my prediction, however it is not totally surprising because I wondered if there might be multiple clusters close together on the right side of scatter plot in figure 1. Of course, having knowledge of the data is the best way of knowing how many clusters to use for Lloyd's algorithm. Since I have limited knowledge of the data, I will choose to run Lloyd's algorithm with four clusters for both random initialization and K-means++ initialization for dataset 1.

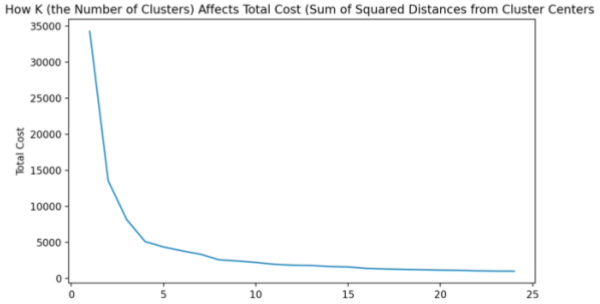

Figure 3 – Cost as a Function of Number of Clusters for Dataset 1
Using K-means with Random Initialization

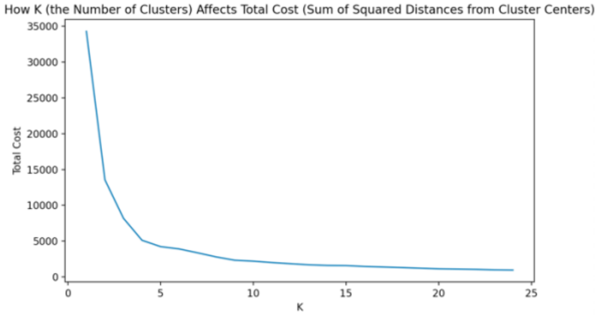

Figure 4 - Cost as a Function of Number of Clusters for Dataset 1
Using K-means with K-means++ initialization

Dataset 2

For dataset 2, picking the number of clusters is much harder because there is no clear elbow in the graph. This is most likely because the dataset as seen in figure 2, is not normally distributed. I had predicted that K-means would not be an effective way to cluster this data, and the cost as a function of the number of clusters as seen in figures 5 and 6 below are further evidence that this method will not be effective for clustering dataset 2. I will be clustering dataset 2 using Lloyd's algorithm with random initialization and k-means++ initialization anyways, so for both methods I will choose to use six clusters. Six clusters is my best guess at an "elbow" for both functions.

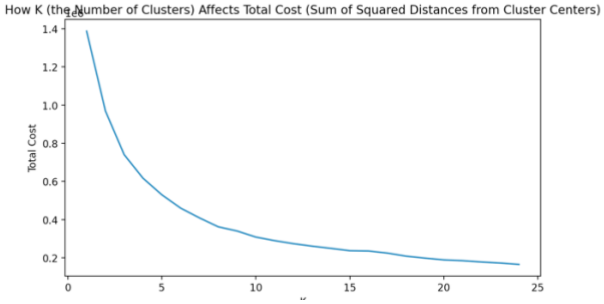

Figure 5 – Cost as a Function of Number of Clusters for Dataset 2
Using K-means with Random Initialization

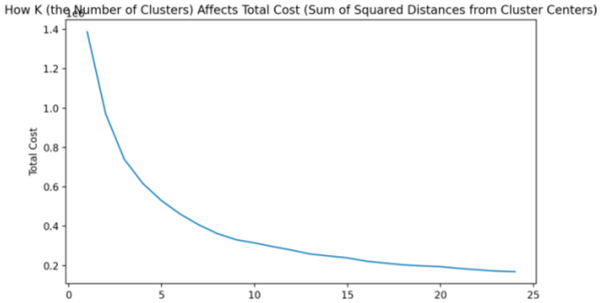

Figure 6 - Cost as a Function of Number of Clusters for Dataset 2
Using K-means with K-means++ initialization

Results

Dataset 1

Running k-means with random initialization and k-means++ initialization with four cluster centers on dataset 1 is shown below in figure 7 and figure 8 respectively. The clustering was successful in both cases. It's good to see the left-most distribution mostly separate from the multiple blobs on the right. This is in line with my initial predictions.

The two plots also appear to be the exact same, although I have not double checked each point. This makes sense for a smaller number of clusters since they might have a high chance at approaching the same local minimum cost, but I would expect as more clusters were added, especially clusters close together, there might be two initializations which would approach different local minimum costs and therefore the clusters would be different.

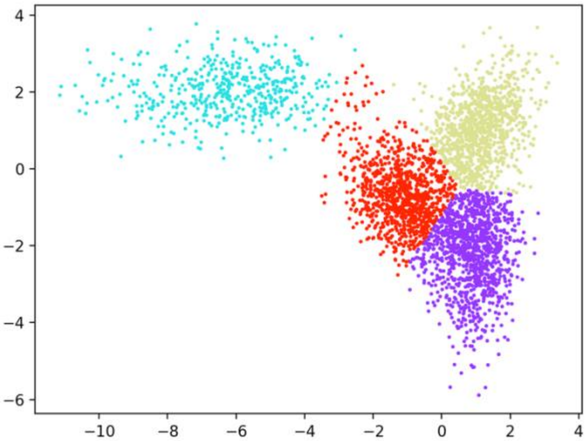

Figure 7 – Clustering Dataset 1 Using K-means with Random Initialization

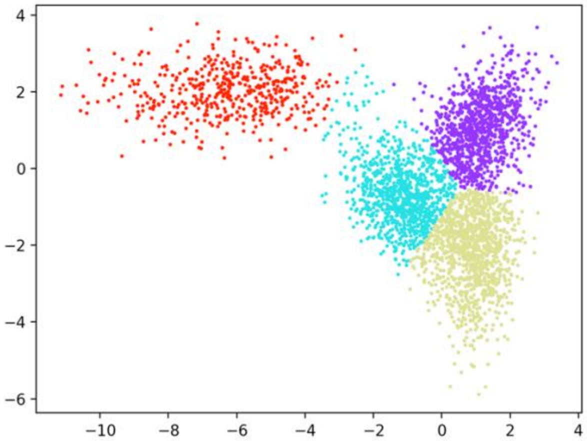

Figure 8 – Clustering Dataset 1 Using K-means with K-means++ Initialization

Dataset 2

The plots for K-means with random initialization and k-means++ initialization with six cluster centers for dataset 2 are shown in figures 9 and 10 respectively. These clusterings were unsuccessful. From the scatter plots, we know there are two distinct clusters, but k-means not only picked the wrong number of clusters, but the algorithm mixed the two true clusters together in each of its chosen clusters. This is in line with my initial predictions.

These plots actually do have different clusters, confirming that different initializations can yield different clusters. The random initialization has part of the green cluster on the highest portion of the plot, whereas the k-means++ initialization does not have a small portion of a cluster on top. Other differences can be seen in the plots as well.

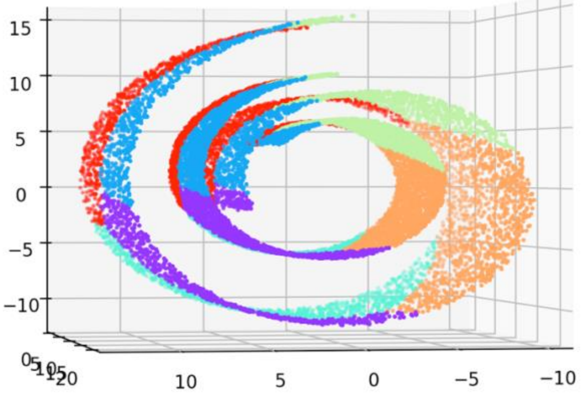

Figure 9 – Clustering Dataset 2 Using K-means with Random Initialization

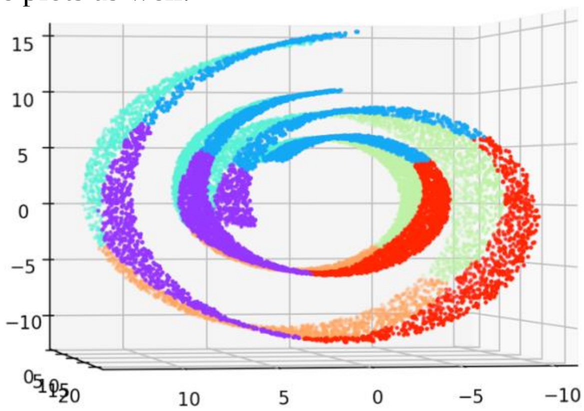

Figure 10 – Clustering Dataset 2 Using K-means with K-means++ Initialization

Hierarchical Agglomerative Clustering

Choosing The Number of Clusters

Again, the first step to clustering using Hierarchical agglomerative clustering is to choose the number of clusters. This time we will use a dendrogram to select the number of clusters. In the dendrogram, a long distance between a parent and child signifies that a large amount of dissimilarity has been reduced by splitting that cluster into two clusters. So we want to choose a minimum number of clusters that has a large distance from the top of the dendrogram to the highest child of the last cluster.

Dataset 1

For dataset 1, the dendrograms using agglomerative clustering for single linkage and average linkage are shown in figure 11 and figure 12 below. The number of clusters to choose is less obvious for figure 11 than figure 12, since figure 11 shows mostly small changes in dissimilarity, whereas in figure 12 we have several very long branches indicating a large decrease in dissimilarity for a small number of clusters. In figure 11, I'm going to choose 10 clusters (the last blue branch) because this is where I perceive the longest branches to end. For figure 12, I'm going to choose two clusters because I think only the blue section of the dendrogram has long enough branches to be worth clustering together. These dendrograms align with my initial hypotheses since the dendrogram for the single linkage on dataset 1 is less clear than that dendrogram for average linkage on dataset 2, indicating that HAC by average linkage may be the better option for clustering dataset 1.

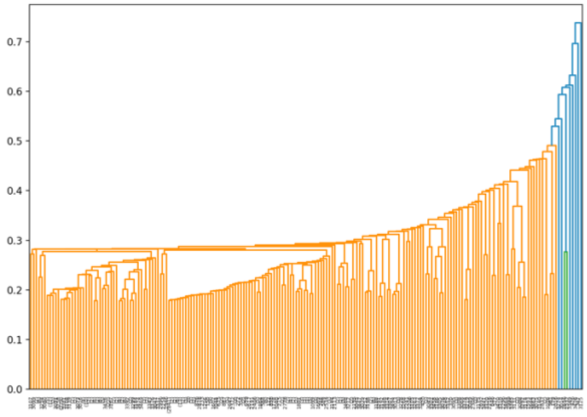

Figure 11 – Dendrogram of Using HAC with Single Linkage on Dataset 1

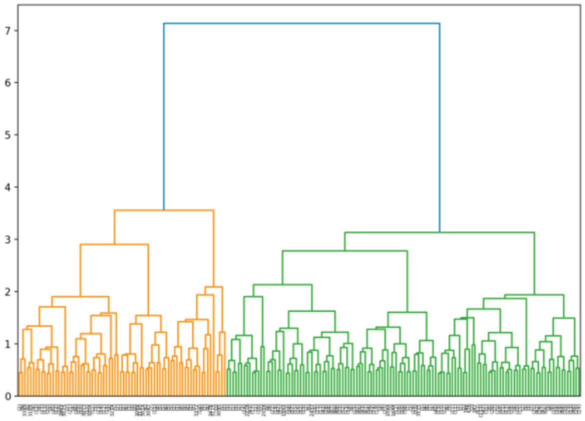

Figure 12 – Dendrogram of Using HAC with Average Linkage on Dataset 1

Dataset 2

In dataset 2, the success of HAC with each type of linkage is reversed. Here we see that single linkage in figure 13 yields a dendrogram with a few longer branches and then a bunch of short ones, while the dendrogram for average linkage in figure 14 does not have any distinctly long branches. For HAC with single linkage on dataset 2, I'm choosing to have five clusters (again at the last blue branch) and only two clusters for HAC with average linkage on dataset 2. Again, this is in line with initial predictions since dataset 2 is not a normal distribution and we would expect average linkage to work poorly for that dataset.

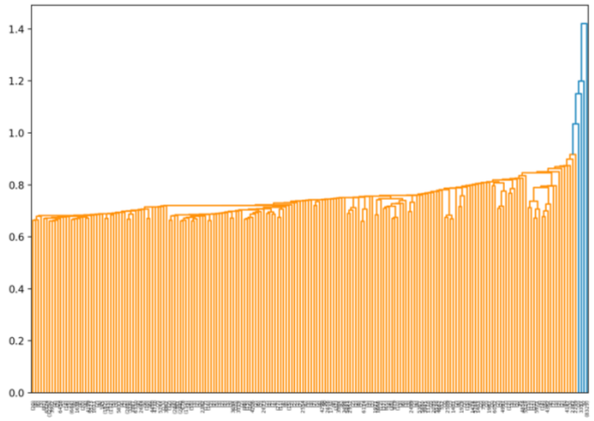

Figure 13 – Dendrogram of Using HAC with Single Linkage on Dataset 2

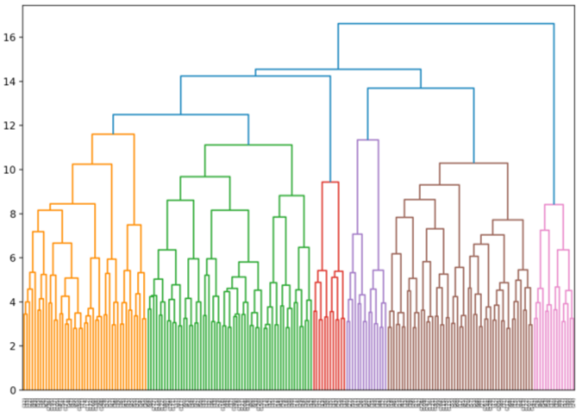

Figure 14 – Dendrogram of Using HAC with Average Linkage on Dataset 2

Results

Dataset 1

As expected, HAC with single linkage poorly clustered dataset 1. The algorithm classified datapoints on the outermost portion of the normal distribution as their own clusters, while clustering everything else together. Intuitively, this is not what we want. HAC with average linkage on the other hand worked well. It successfully clustered the two large distributions. These results are in line with initial predictions.

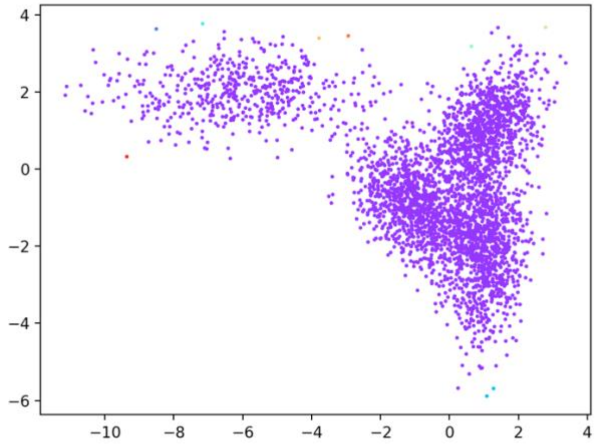

Figure 15 – Clustering Dataset 1 using HAC with Single Linkage

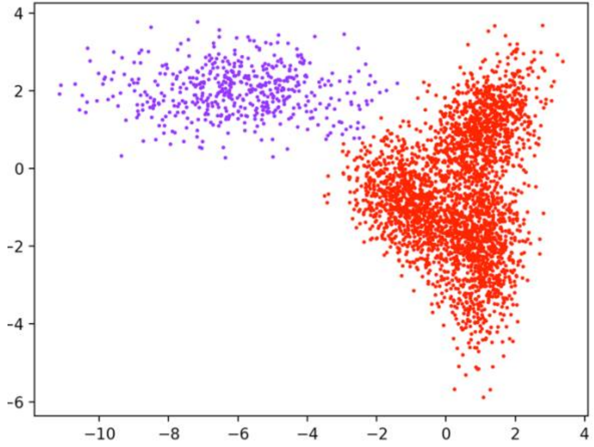

Figure 16 – Clustering Dataset 1 using HAC with Average Linkage

Dataset 2

HAC with single linkage on dataset 2 was largely successful. It clustered the two spiraled distributions correctly. The dendrogram may have led us slightly askew, because two clusters probably would have been better than five. The extra clusters only cluster some outliers which are likely part of the two larger distributions. HAC with average linkage was largely unsuccessful because it clustered a portion of one of the two main clusters, then clustered everything else together. These results are in line with earlier predictions as well since the spiral pattern is expected to work better with single linkage than average linkage when using HAC.

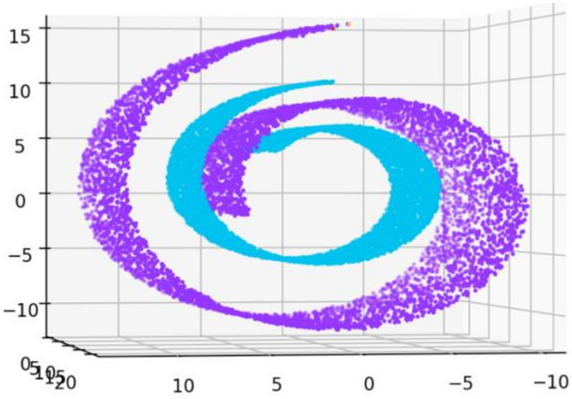

Figure 17 – Clustering Dataset 2 using HAC with Single Linkage

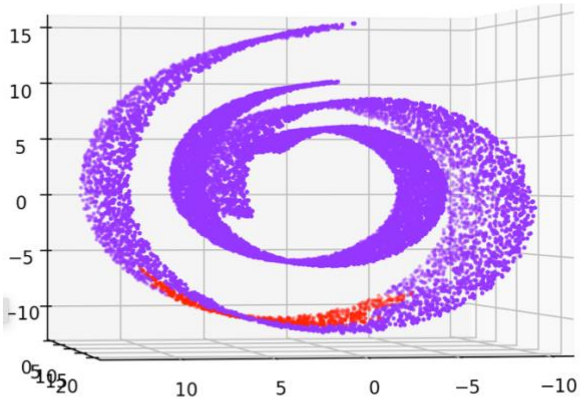

Figure 18 – Clustering Dataset 2 using HAC with Average Linkage

Conclusion

To conclude, my initial predictions were correct, where HAC with single linkage was expected to work well with dataset 2, but the other methods were expected to work well with dataset 1 due to the distributions of the data. Our methods of choosing the number of clusters were largely successful and gave some more intuition as to how effective clustering would be, but knowledge about the datasets would be the best approach for choosing the correct number of clusters. This can be seen for dataset 1, where it is difficult to tell whether the data represents two, three, or four clusters. Furthermore, the cost as a function of the number of clusters for k-means gave us a different number of clusters than HAC with average linkage, even though both methods were successful at clustering the data. Having background knowledge on the data is the best approach to knowing how many clusters to use. Lastly, in these examples we could use a scatter plot to visualize the data, but in higher dimensions this may be more difficult. For higher dimensional data I would guess that having background knowledge on the data would be even more important for choosing the best clustering method and the correct number of clusters to use.