Lloyd's Algorithm & Hierarchical Agglomerative Clustering

SENG 474 Assignment 3
Austin Bassett
V00905244

March 2021

1 Introduction

The nature of this report is to explore two clustering methods, Lloyd's algorithm and hierarchical agglomerative clustering. Two variants for each clustering method were used. Lloyd's algorithm variants were uniform random and k-means++ initialization, and hierarchical agglomerative's variants were single and average linkage. These four variants used two datasets for testing, one two-dimensional (dataset one) and one three-dimensional (dataset two). The results were graphed on 2D and 3D scatter plots for analysis.

2 Lloyd's Algorithm

Lloyd's algorithm alternates between updating clusters based on the centroids and recalculating the centroids until no change has occurred. The clusters are updated with the points with minimum distance to each centroid. The centroids are recalculated based on the means of all points in a cluster. Optimal cluster numbers for each dataset and method were first found using elbow graphs, where the y-axis was cost and the x-axis was the number of clusters. I decided the optimal cluster number was the point where the graph started to increase. The following sections will explore uniform random initialization and k-means++ initialization for obtaining the initial centroids.

2.1 Uniform Random Initialization

The uniform random initialization method chooses random centers from a uniform distribution with all points having the same probability of being chosen.

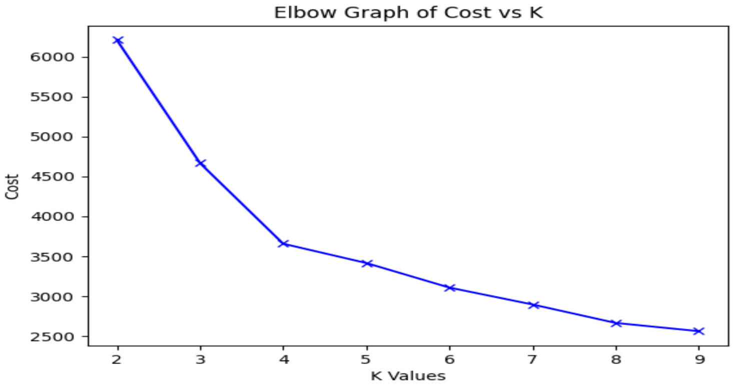
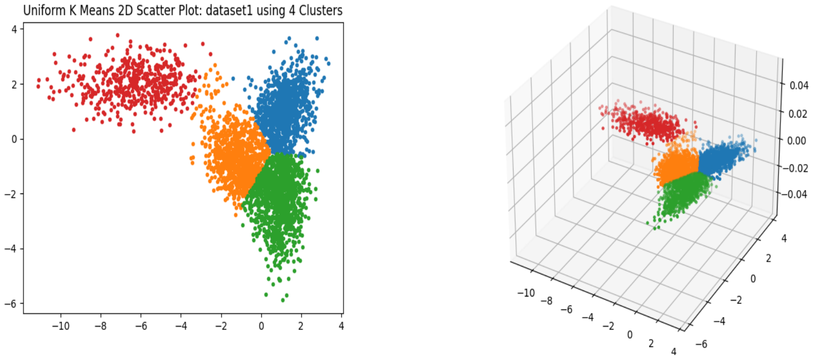


Figure 1: Uniform Random Elbow Graph for Dataset One.

The optimal number of clusters for dataset one was four. This was the point where the slope of the graph started increasing (see Figure 1). I felt four was the optimal number of clusters because dataset one contains 3500 data points. This would also ensure the clusters were distributed somewhat evenly, meaning there wouldn't be clusters with one data point or a cluster with 90 percent of the data points.



(a) 2D Scatter Plot

(b) 3D Scatter Plot

Figure 2: Uniform Random Scatter Plots for Dataset One.

Looking at the scatter plots (see Figure 2) we see the data points were almost uniformly distributed between the clusters. The data points within the green, blue and orange clusters are densely packed into the center regions of their cluster. Compared to the red cluster where the data points are more spread out. The uniform initialization method was effective at clustering a two-dimensional dataset.

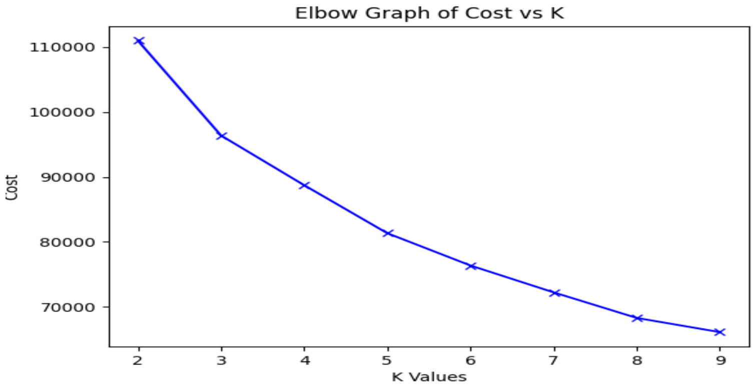
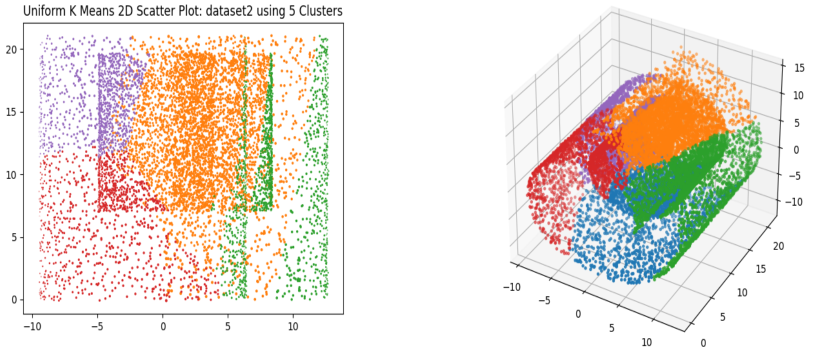


Figure 3: Uniform Random Elbow Graph for Dataset Two.

For dataset two the optimal number of clusters to use was five. Looking at the elbow graph we can see the slope was decreasing over the entire range, meaning I could choose any number as the optimal number (see Figure 3). Instead of choosing a number at random I decided to choose based off the total number of data points in dataset two, which was almost 15,000. Knowing this I decided on five clusters because there was a potential for each cluster to contain approximately 3,000 data points.



(a) 2D Scatter Plot

(b) 3D Scatter Plot

Figure 4: Uniform Random Scatter Plots for Dataset Two.

Looking at the scatter plots (see Figure 4) we can see the data points were almost uniformly distributed. Unlike with dataset one, we see more of a spread of data points within the clusters instead of dense spots. While it does appear in the 3D scatter plot that orange contains a dense region in the middle. However, this is not the case because in this area both the inner and outer shell are coloured orange, giving the appearance of a dense region of data points. The uniform initialization was also effective at clustering a three-dimensional dataset.

2.2 K-Means++ Initialization

The k-means++ initialization method randomly selects a point as its first centroid. Then the distance of each point from the nearest previously chosen centroid is calculated, once all the calculations are completed the next centroid is decided. The next centroid is selected with probability proportional to its distance from the nearest previously picked centroid. Finally, these steps are repeated until k centroids have been determined.

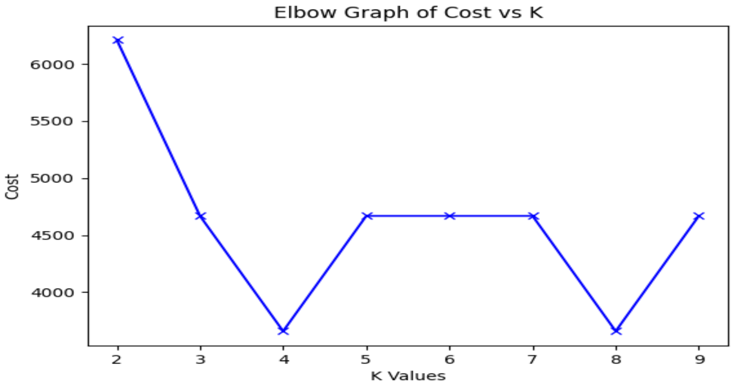
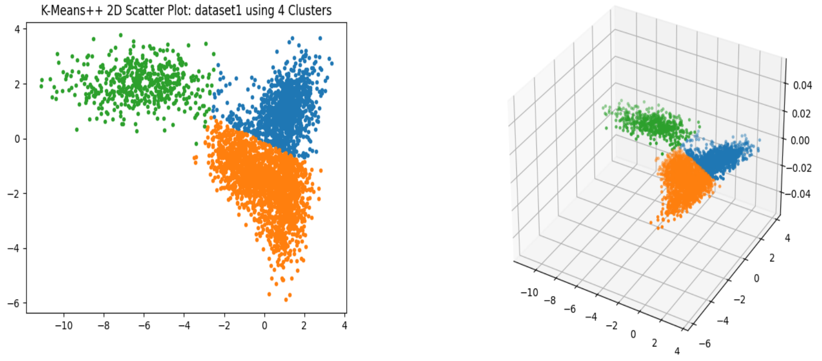


Figure 5: K-Means++ Elbow Graph for Dataset One.

Just like with uniform random initialization the optimal number of clusters for dataset one is four. Looking at the elbow graph we see that the two lowest points are at four and eight (see Figure 5). The decision for using four clusters instead of eight was because it would allow comparison with the uniform random results.

Looking at the scatter plots (see Figure 6) we see the orange and green clusters make up the majority of the plot. Which is completely different from the uniform random method. However, there are only three clusters even though four clusters were specified. This could be due to either my implementation of k-means++ or not enough data points for this method and cluster number. Because as we will see with dataset two my k-means++ implementation was able to cluster the data into the specified five clusters.



(a) 2D Scatter Plot

(b) 3D Scatter Plot

Figure 6: K-Means++ Scatter Plots for Dataset One.

Just like with uniform random initialization the optimal number of clusters for dataset two was five. Looking at the elbow graph we can see there is a lower point located at eight clusters, but by using five clusters I could compare the results with the uniform random results (see Figure 7).

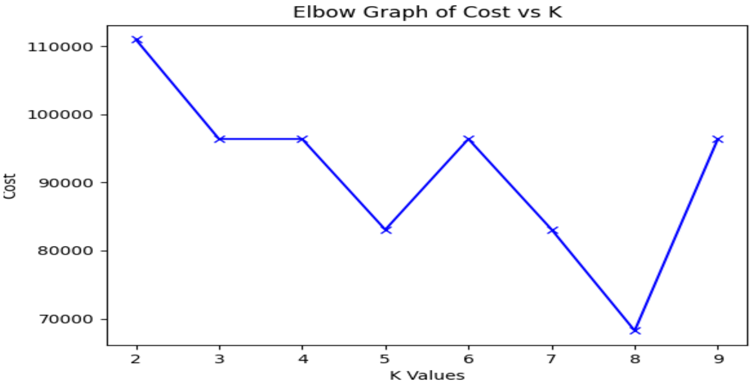
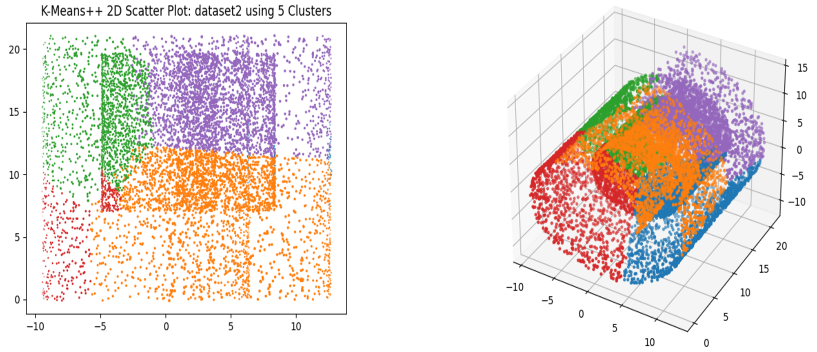


Figure 7: K-Means++ Elbow Graph for Dataset Two.

Looking at the scatter plots (see Figure 8) we find something quite interesting, the 2D scatter plot appears to only contain four clusters even though five were specified, but upon closer inspection we see blue dots scatter throughout. These blue dots become more prominent with the 3D plot. Unlike with the uniform random initialization, we see less evenly distributed data between the clusters, with the orange and purple clusters containing the majority of data points.



(a) 2D Scatter Plot

(b) 3D Scatter Plot

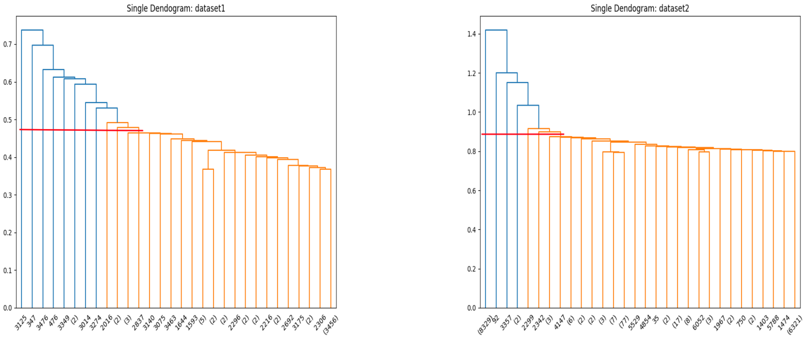
Figure 8: K-Means++ Scatter Plots for Dataset Two.

3 Hierarchical Agglomerative Clustering

The hierarchical agglomerative clustering method uses the bottom-up approach to build a hierarchy of clusters. The bottom-up method initializes each observation in their own clusters, then merges pairs as one moves up the hierarchy. Determining which clusters to merge is done with a metric called linkage. Linkage takes elements from each cluster and calculates the distance between them. The following sections will be exploring the results of single linkage and average linkage.

3.1 Single Linkage

The single linkage method merges two clusters with the minimum euclidean distance. Before running hierarchical agglomerative clustering I wanted to find the most optimal number of clusters. Taking a horizontal cut through the largest rectangles of the dendrograms, I used the total number of vertical lines that were cut as the optimal number. The optimal number of clusters were eleven and seven for dataset one and two respectively (see Figure 9).

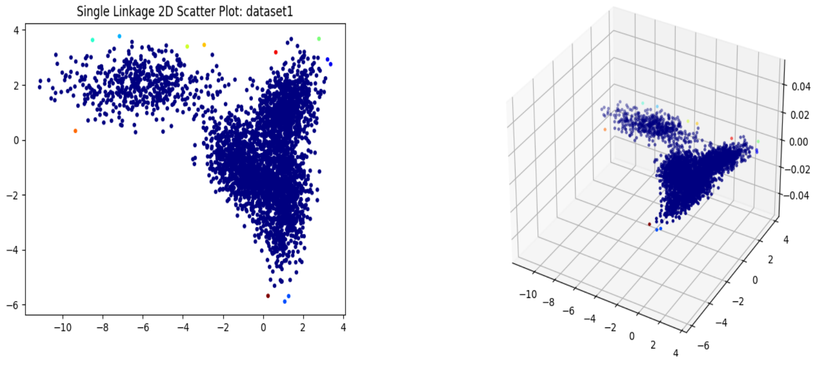


(a) Dataset One

(b) Dataset Two

Figure 9: Single Linkage Dendrograms for Both Datasets.

Looking at both scatter plots (see Figure 10) for dataset one we see a cluster with over 90 percent of the data points (dark blue), a cluster with four data points (blue), and the other nine clusters only containing a single data point. This indicates single linkage is not optimal for clustering the data within dataset one, as there isn't enough distance between data points.

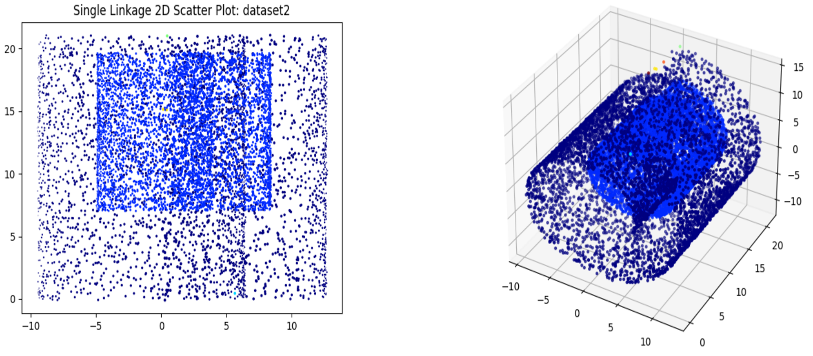


(a) 2D Scatter Plot

(b) 3D Scatter Plot

Figure 10: Single Linkage Scatter Plots for Dataset One.

Looking at both scatter plots (see Figure 11) for dataset two we see two clusters that contain a majority of the data points (dark blue and blue). The next largest cluster contains two data points (yellow) and the last four clusters only containing a single data point each. While single linkage clustered the data better than with dataset one, there is still not enough distance between points for an effective clustering.



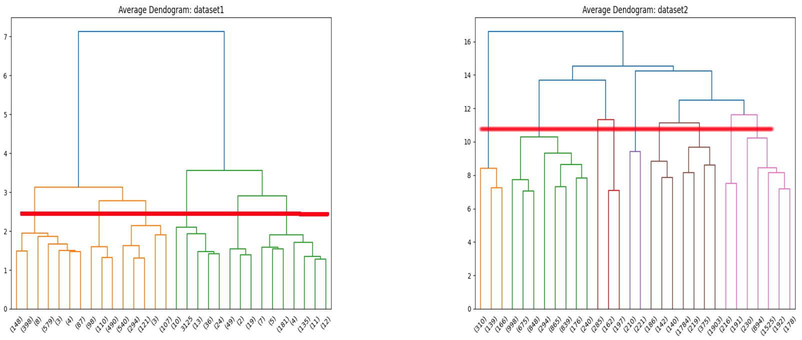
(a) 2D Scatter Plot

(b) 3D Scatter Plot

Figure 11: Single Linkage Scatter Plots for Dataset Two.

3.2 Average Linkage

The average linkage method merges two clusters with the minimum mean euclidean distance. Just like with single linkage, I found the optimal number of clusters using horizontal cuts through the largest rectangles of the dendrograms (see Figure 12). From these cuts I found the optimal number of clusters to be six and nine for dataset one and two respectively.

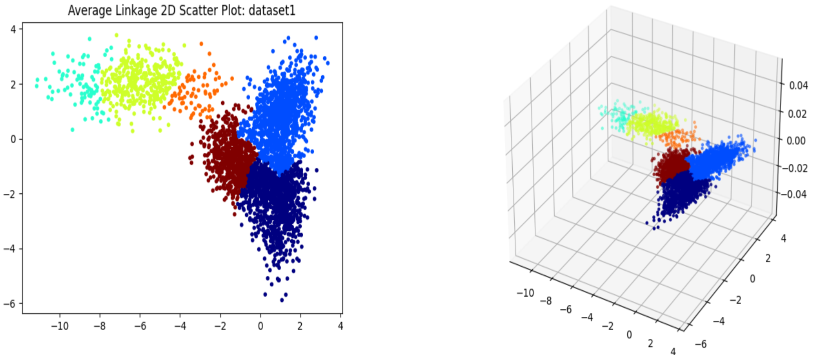


(a) Dataset One

(b) Dataset Two

Figure 12: Average Linkage Dendrograms for Both Datasets.

Looking at both scatter plots (see Figure 13) for dataset one we see a nice distribution of data points in all six clusters. With the blue cluster being the largest and orange being the smallest. This indicates that average linkage is an effective way to cluster the data points from dataset one.

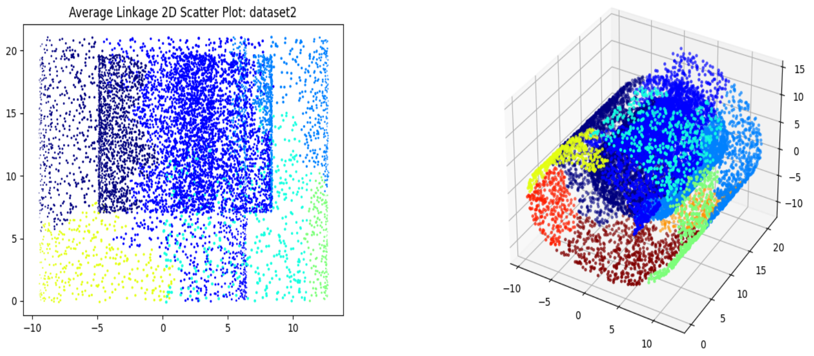


(a) Dataset One

(b) Dataset Two

Figure 13: Average Linkage Scatter Plots for Dataset One.

Looking at both scatter plots (see Figure 14) for dataset two we also see a nice distribution of data points among the nine clusters. However, this is only noticeable in the 3D plot because the dataset is of dimension three, so the 2D plot can't accurately represent all the clusters. This indicates average linkage was effective at clustering the data points from dataset two.



(a) Dataset One

(b) Dataset Two

Figure 14: Average Linkage Scatter Plots for Dataset Two.

4 Conclusion

From the results we see that the uniform random initialization and the average linkage variants were the most effective at clustering. This is because these variants spread the data points into clusters that would be beneficial for analysis. Whereas, k-means++ and single linkage seemed to glob the data points into one giant cluster, and filling the other k-1 clusters with at most two data points.