SENG 474 A02: Assignment 3

Binary Classification Experiments: Lloyd’s algorithm (k-means),

Hierarchical agglomerative clustering

Chenuri Gunaratne

March 23rd, 2023

1. The Data

The two datasets we will be applying the clustering methods are dataset1.csv, which contains 3500 two-dimensional examples generated by a Gaussian mixture model, and dataset2.csv, which consists of 14,801 three-dimensional examples. When examining the structure of the points in dataset2.csv using a 3D scatterplot, we observed a cylindrical spiral structure (Figure 1). This suggests the presence of clusters that may be arranged in a spiral-like pattern, which could have implications for both K-means and hierarchical clustering.

For K-means clustering, the placement of the initial cluster centers can significantly affect the resulting clusters. Poor initial placement can lead to a suboptimal solution. One way to address this is to use a more advanced initialization method, such as K-means++.

In hierarchical clustering, the spiral structure of the clusters in dataset2.csv can make it difficult to determine the appropriate level to cut the dendrogram. The arrangement of the clusters in a spiral pattern may result in hierarchical clustering producing less clear and distinct clusters that are harder to interpret. This could be a challenge in cases where the hierarchical clustering results need to be easily interpretable.

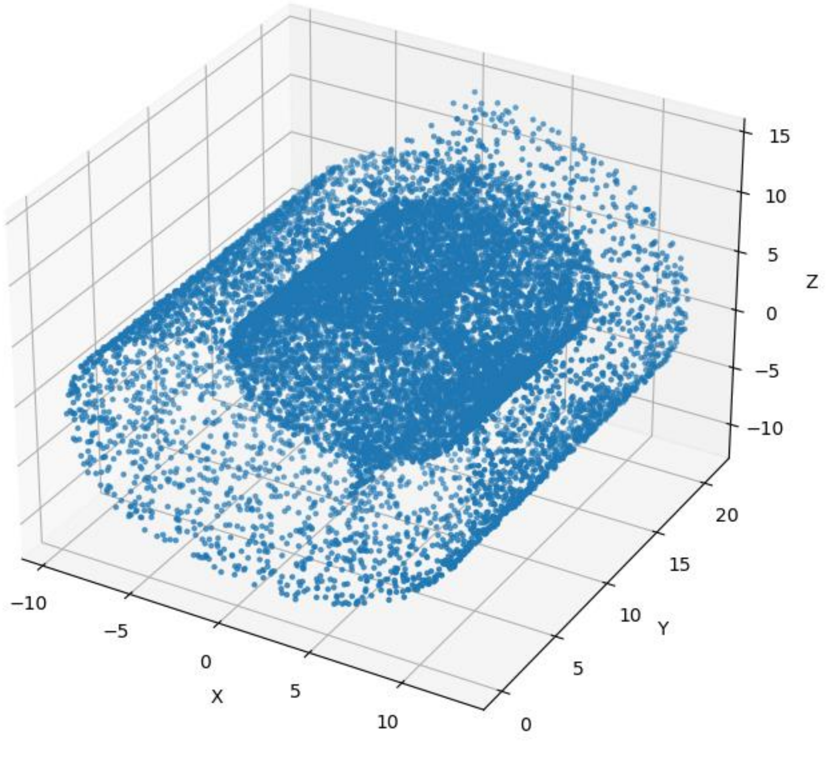

Figure 1: 3D plot of Dataset2.csv

2. Lloyd’s Algorithm (K-means)

Clustering is the process of grouping together similar data points based on some similarity measure. The K-means algorithm works by iteratively partitioning the data into K clusters, where K is a predetermined number of clusters specified by the user.

The implemented k-means function which you can find in the notebook "A3.ipynb" takes in three arguments: X, which is the data matrix consisting of n rows (examples) and d columns (features); k, which is the number of clusters desired; and initialization, which is the initialization method used to initialize the cluster centers. There are two initialization methods to choose from: K-means++ initialization and Uniform random initialization.

The function first initializes K cluster centers either randomly or using the K-means++ initialization method. Then, it iteratively assigns each data point to the nearest cluster center and updates the cluster centers based on the mean of the data points in each cluster. This process is repeated until the cost function (which is the sum of squared distances between data points and their assigned cluster centers) no longer improves.

To ensure that the function finds a good clustering solution, the initialization and assignment process is repeated for multiple initializations (n_init=10) and the solution with the lowest cost function is returned.

2.1 Dataset 1

2.1.1Uniform Random Initialization

The k-means clustering algorithm was applied with uniform random initialization to dataset1.csv over a range of number of clusters, resulting in the graph shown in figure 2. The quality of the clustering was measured using the cost, which represents how well the data points are assigned to their respective clusters. Specifically, the cost is calculated as the sum of the squared distances between each data point and its assigned cluster center.

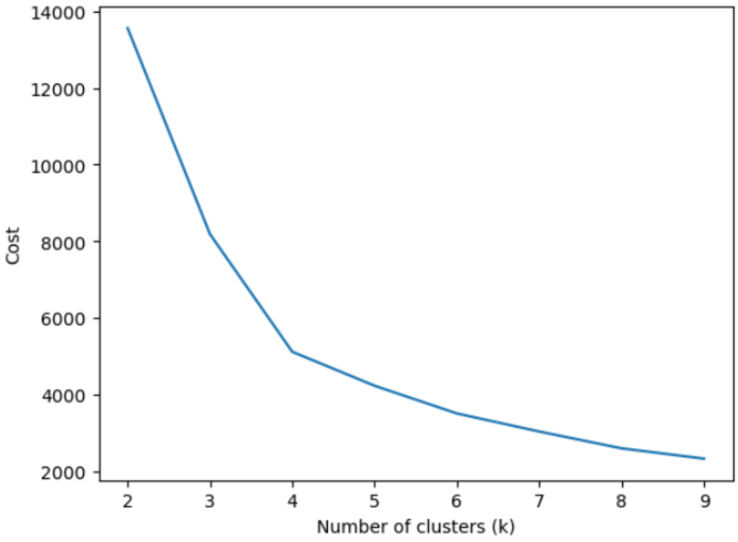

Figure 2: Cost vs Clusters, Dataset 1: Uniform Random Initialization

The graph shows that as we increase the number of clusters from 2 to 9, the cost decreases rapidly at first. This is because having more cluster centers allows for a better fit of the data points, which results in a better assignment of the points to their respective clusters. However, as we continue to increase the number of clusters, the cost reduction slows down, eventually reaching a point where adding more clusters does not provide a significant improvement in the clustering quality.

In this particular case, the graph shows that the cost reduction slows down significantly after 4 clusters. Therefore, using uniform random initialization for kmeans, 4 clusters would be a good choice for this dataset. Increasing the number of clusters beyond 4 would not provide a significant improvement in the clustering quality.

2.1.2 K-means++ Initialization

The k-means++ initialization method is often used to improve the quality of clustering. This method starts by randomly selecting one example from the dataset as the first cluster center, and then iteratively selects the next center from the remaining examples with a probability proportional to the squared distance to the closest existing center. By doing so, the initial cluster centers are well-separated, leading to better clustering and a lower cost.

Interestingly on this dataset, even with the k-means++ initialization method, the rate of decrease in the cost still slows down after k=4 as seen in figure 3, similar to when k-means is used with uniform random initialization (Figure 2). This indicates that further increasing the number of clusters beyond this point does not significantly improve the quality of clustering.

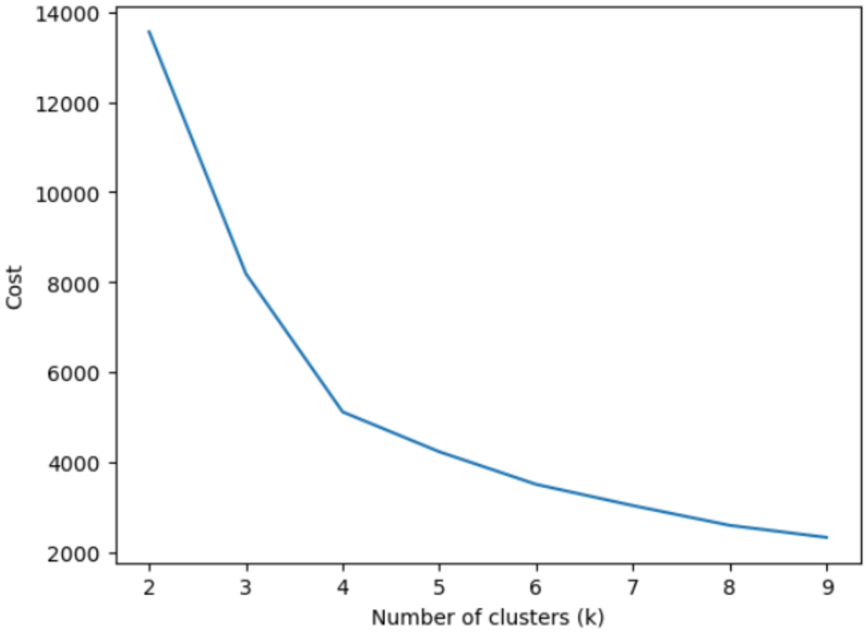

Figure 3: Cost vs Clusters, Dataset 1: K-Means++ Initialization

2.2 Dataset 2 and K-Means Clustering

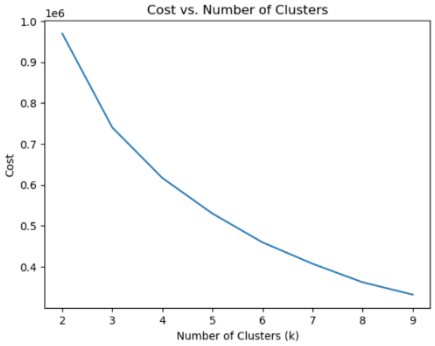

Figure 4: Cost vs Clusters, Dataset 2: Uniform Random Initialization

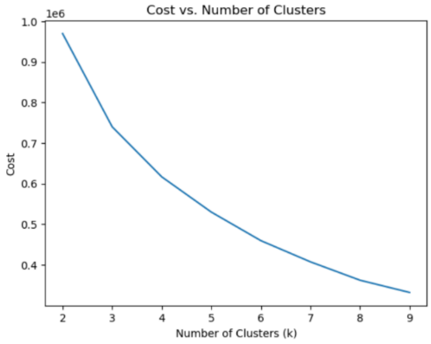

Figure 5: Cost vs Clusters, Dataset 2: K-means++ Initialization

Figure 4 and 5, which correspond to dataset 2 with uniform random initialization and k-means ++ initialization, respectively, we observe a gradual decrease in cost as the number of clusters increases, without any distinct elbow point which makes it challenging to determine the optimal number of clusters.

Fortunately, there are alternative methods that can be used to determine the optimal number of clusters, particularly when dealing with non-spherical datasets. For example, spectral clustering is a powerful alternative to k-means clustering that can work well on datasets with complex shapes. Additionally, hierarchical clustering and density-based clustering are other popular approaches that can be effective in identifying the optimal number of clusters for datasets with varying shapes and densities.

3. Hierarchical Agglomerative Clustering

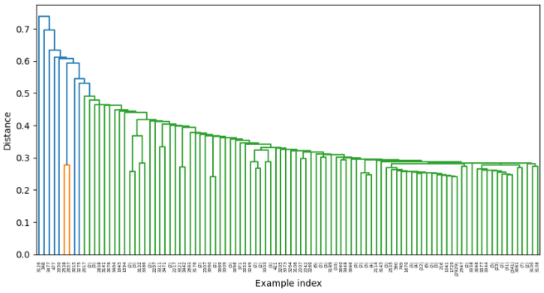

Figure 6: Dendogram, Dataset 1: Single Linkage

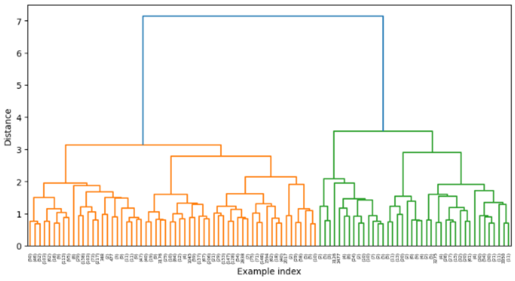

Figure 7: Dendogram, Dataset 1: Average Linkage

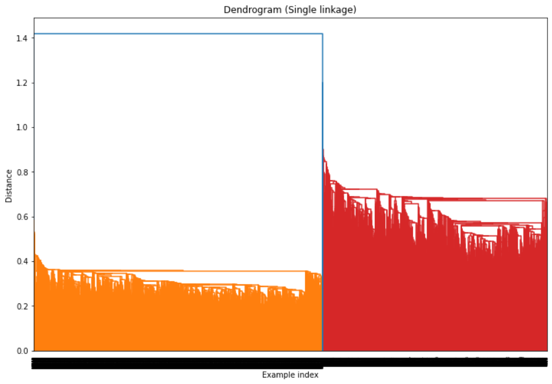

Figure 8: Dendogram, Dataset 2: Single Linkage

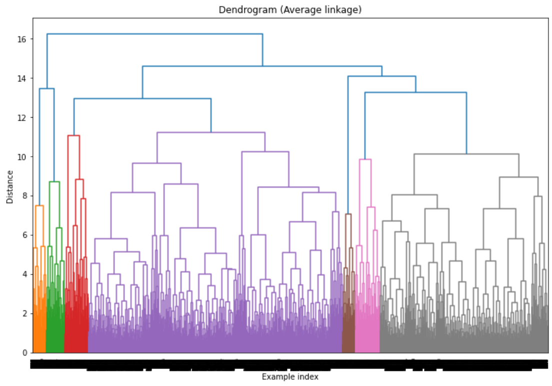

Figure 9: Dendogram, Dataset 2: Average Linkage

Hierarchical clustering is a powerful alternative to k-means clustering that can work well on datasets with complex shapes and structures. This method groups similar data points together in a hierarchical tree-like structure, allowing for the identification of clusters at different levels of granularity. One advantage of hierarchical clustering is that it does not require specifying the number of clusters a priori, as the number of clusters can be determined based on the structure of the dendrogram. However, to determine the optimal number of clusters, we need to find a good threshold value for cutting the dendrogram.

When creating dendrograms to visualize hierarchical clustering, one common approach is to use a threshold value to cut the dendrogram at the tallest vertical line. This threshold value determines how many clusters will be formed. However, it can be difficult to visually determine the tallest vertical line, especially when dealing with a large amount of data.

One alternative method is to loop through different threshold values and observe the resulting number of clusters for each threshold. By doing so, we can identify the range of threshold values for which the number of clusters remains the same. This range of threshold values corresponds to a region where the vertical lines are all relatively tall. To determine the optimal number of clusters, we can simply choose the value that appears most frequently within this range of threshold values. This approach ensures that we are selecting a stable and consistent number of clusters, which can be useful for downstream analyses.

Based on this and the dendrograms shown in figures 6 to 9, we can observe that the optimal number of clusters varies depending on the dataset and linkage method used. For dataset 1, single linkage resulted in 7 clusters while average linkage resulted in 2 clusters. On the other hand, for dataset 2, single linkage resulted in 4 clusters while average linkage resulted in 7 clusters.

In summary, hierarchical clustering can be an effective approach for clustering data, particularly when dealing with complex shapes and structures. While it may be challenging to determine the optimal number of clusters based on the dendrogram alone, using a threshold value to identify a stable range of cluster numbers can provide a reliable method for determining the optimal number of clusters. This approach can help ensure that subsequent analyses are robust and meaningful. Furthermore, it is worth noting that the optimal number of clusters derived using this method can vary depending on the type of linkage used, as illustrated by the different optimal k-values observed for each dataset and linkage method.