CS 519: Applied Machine Learning I

HW6: Compare clustering methods

Submitted By: Md Ishtiaq Ahmed

Aggie ID: 800606216

Dataset Description:

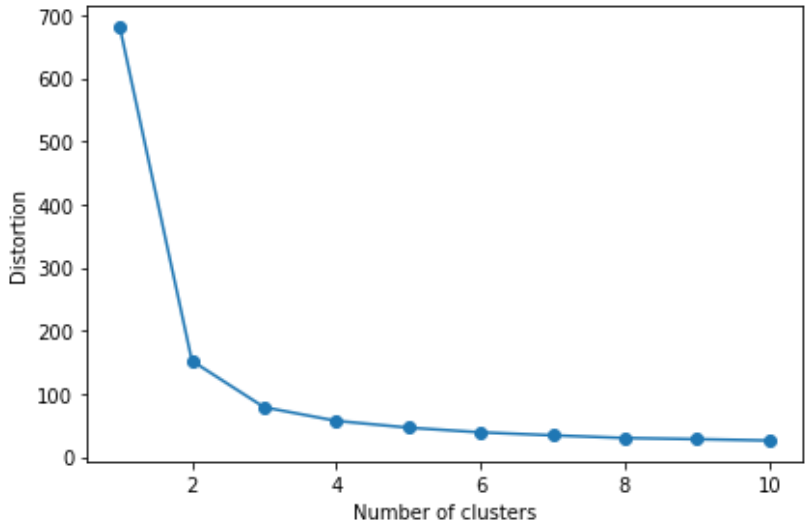
Two datasets are used in the assignment:

1. Iris dataset: The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. There is a total of 4 features. This dataset is available in sklearn library.
2. MNIST Dataset: The MNIST (Modified National Institute of Standards and Technology) dataset consists of 70,000 instances representing images of handwritten digits from 0 to 9. Each image has a size of 28x28 pixels. The dataset has 784 features per image. The dataset is imported using fetch_openml function where the dataset name is "mnist_784"

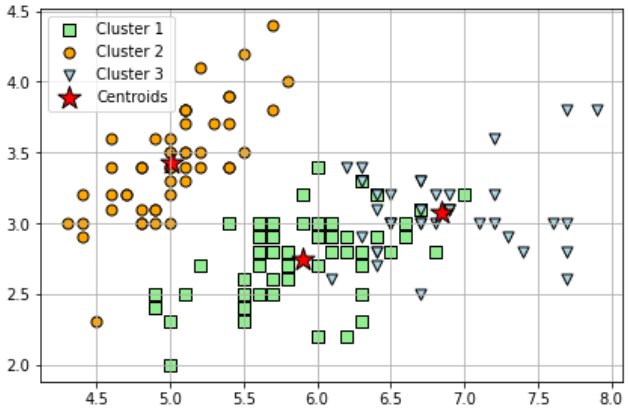
Analysis of the IRIS Dataset

K-means algorithm

First, I used the elbow method to determine the number of clusters (K). From the elbow plot the point where sharp distortion occur is a suitable value to pick K. Below is the elbow diagram of IRIS dataset:



From the plot, it is visible that from cluster 3, sharp distortion started to happen. So, I have selected k=2. I plotted the 3 clusters along with their centroid. I also calculated the Sum Squared Error and Silhouette Score to get an idea of the clustering performance.



K=3

Cluster 2 seems well separated whereas there is some overlapping between Cluster 1 and Cluster 3. Below is the performance parameters value:

```
SSE: 78.85
Silhouette Score: 0.55
Running time to fit the model: 47.976 ms
```

The Sum Squared Error gives an idea about the compactness of the clusters where lower SSE indicates better clustering performance. However, SSE is not always a good measure as with the number of clusters increasing, SSE decreases. In this case, SSE value 78.85 indicates that the clustering performance is not that good.

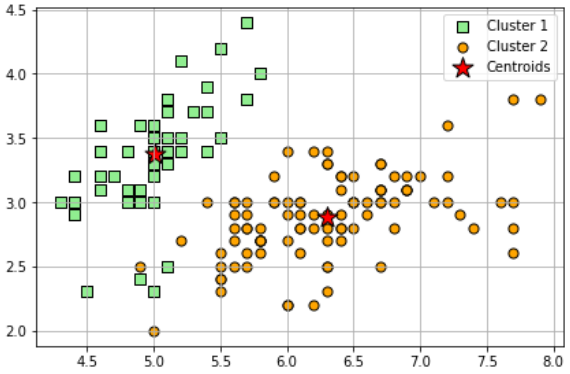
I checked the Silhouette co-efficient to get a better idea about the clustering performance. The Silhouette Score measures the cohesion and separation of the clusters. It ranges from -1 to 1, with higher values indicating better clustering performance. In this case, the Silhouette Score is 0.55, which suggests a moderate level of separation between the clusters.

The model fitting time is also low which is only 47.976 millisecond.

Using k=2, I got the below value:

```
SSE: 152.35
Silhouette Score: 0.68
Running time to fit the model: 30.087 ms
```

Here, Silhouette Score is better as K-means can separate the dataset more clearly in 2clustersr.



K=2

Analysis of class labels as ground truth

To investigate further, I checked the cluster labels against the ground truth (actual label). To do this, I checked the actual labels of all three clusters separately. For each cluster, I found the most frequent class label and considered the cluster for that class label. That means, if cluster 1 has class label 0 the most, then I considered cluster 1 is for label 0. Then I calculated the accuracy of each cluster using below formula:

accuracy_cluster1 = correct_label1/total_label1, here correct_label indicates the the no of most frequent label and the total_label is total no of data point in that cluster. Then I aggregate the accuracy for all of three clusters. Below is the outcome:

```
Accuracy of Cluster 1 against ground truth: 0.77
Accuracy of Cluster 2 against ground truth: 1.00
Accuracy of Cluster 3 against ground truth: 0.95
Overall Accuracy against ground truth: 0.89
```

We can see that, using K-means, good accuracy has been achieved against the ground truth.

Hierarchical approach

I used the hierarchical approach offered by both the scipy and Scikit-learn libraries which gives following result:

Hierarchical Library	Linkage Type	Silhouette Score (K=10)	Model Fitting Time (ms)
scipy	complete	0.51	2
scikit-learn	complete	0.51	1.999
	average	0.55	2.997

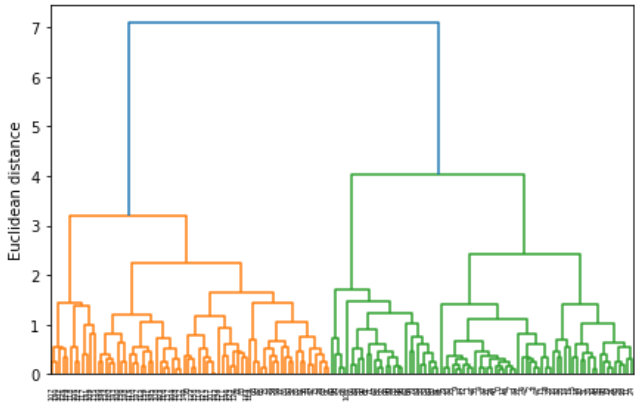


Figure: Dendrogram using scipy library

From the dendrogram using the scipy library, I decided to get the final cluster at level 3.

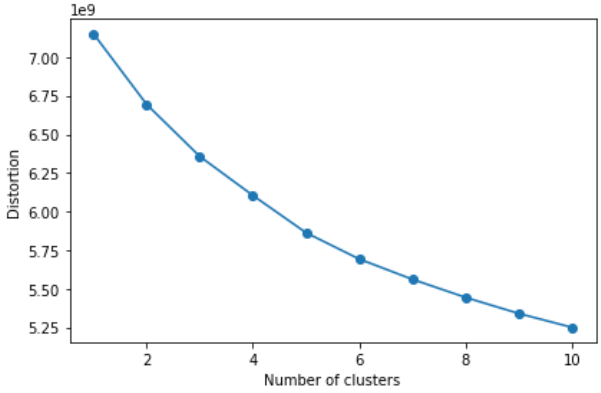
Using 'complete' linkage both scipy and scikit-learn library gives a silhouette score of 0.51. Using 'average' linkage in SKlearn gives a slightly better score which is 0.55. the scores are like the score we got from the K-means algorithm. It indicates that the clustering performance is almost the same using K-means and hierarchical approach for IRIS dataset. The fitting time of the 'average' linkage method is slightly higher it might be because average linkage calculates the average distance between all pairs of data points in two clusters.

Analysis of the MNIST Dataset

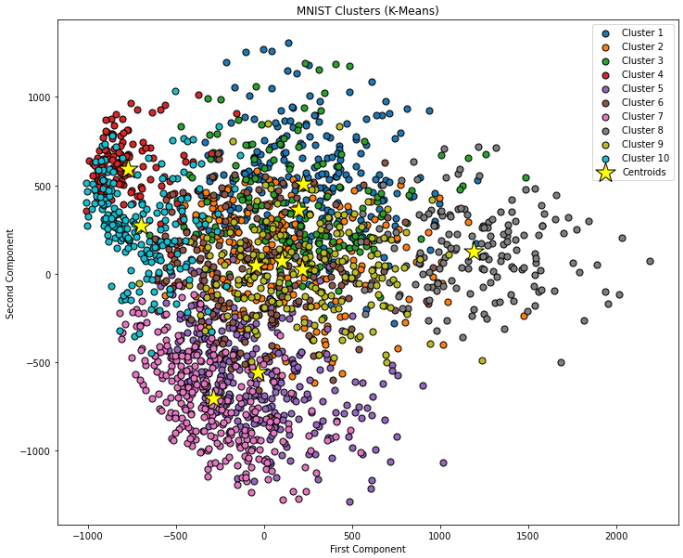
For analyzing the MNIST dataset, I took .03% of the total which is 2100 instances using the train_test_split method, and stratify using the data labels to make sure that the subset has a representation of all class labels.

K-means algorithm

The elbow plot suggests that the distortion is on the rise from the beginning when K=10. So, I selected k=10 to implement K-means algorithm.



I also used PCA to reduce the no of features to two dimensions to get a better view of all the 10 clusters along with the centroids.



K=10

From the above cluster plot, it is clear that the clusters are not well separated which is also confirmed by the SSE and silhouette score.

```
SSE: 5250368043.24
Silhouette Score: 0.06
Running time to fit the model: 3132.188 ms
```

SSE is very high which suggests that the data points are more spread out and the silhouette score is almost 0, which means this algorithm is not suitable for this dataset. Since the number of instances is high compared to IRIS dataset, fitting time is also high, and it is over 3 seconds.

Analysis of class labels as ground truth

For the mnist dataset, I used adjusted random score to evaluate the performance of the algorithms by comparing the similarity between the predicted cluster assignments and the true labels. Here, **Adjusted Rand Index is 0.40** which means the clustering algorithm can capture some of the true labels, but it is not up to the desired level.

Hierarchical approach

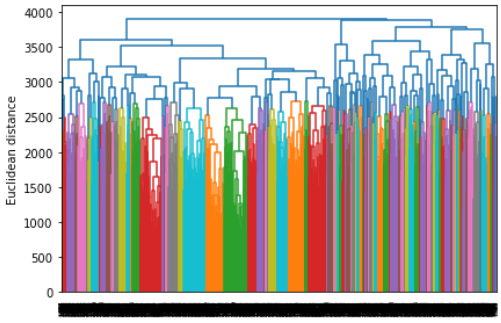


Figure: Dendrogram using scipy library

The dendrogram using scipy library hierarchical approach shows that the clustering is complex for the MINST dataset and its difficult to determine which is the desired cluster level. So, I used 10 clusters like K-means algorithm to get some results using both scipy and scikit-learn library.

Hierarchical Library	Linkage Type	Silhouette Score (K=10)	Model Fitting Time (ms)
scipy	complete	0.03	1691.56
scikit-learn	complete	0.03	1678.619
	average	0.06	1873.473

The above table gives a similar result to K-means algorithm. Here as well, the fitting time for 'average' linkage method is slightly higher and gives a slightly better silhouette score than 'complete' linkage method. However, the silhouette score is still very low and it suggests that this model is not suitable for clustering this subset of the MNIST dataset.