EMRAH SARIBOZ

HW-7 Report

Cluster purity and run-time analysis

Algorithm	Dataset	Run-time	Cluster Purity
K-means	iris	0.054 seconds	0.8933
K-means	Mnist	100.16 seconds	0.5
Scipy-Agg	iris	6.16 seconds	0.906
Scipy-Agg	Mnist	25.15 seconds	0.9967
Sklearn-Agg	iris	0.004 seconds	0.893
Sklearn-Agg	Mnist	24.51 seconds	0.9048
DBSCAN	iris	0.014 seconds	0.886
DBSCAN	Mnist	47.29 seconds	1.0

Table 1

Dataset Usage and problem definition

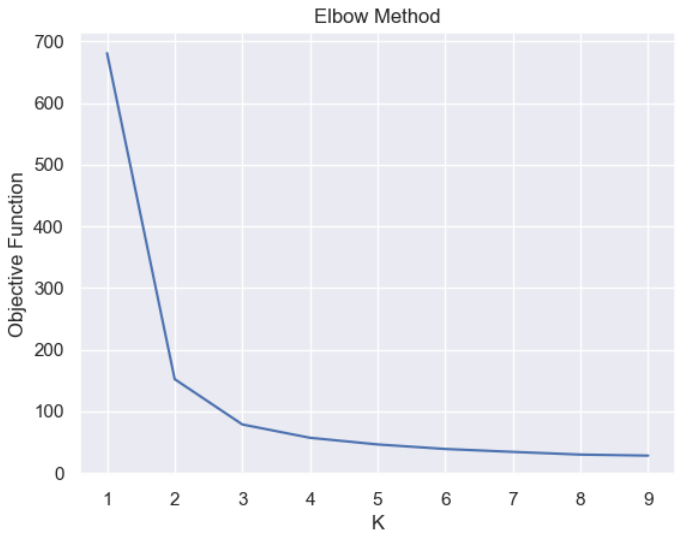
1) I used iris dataset which is known as a classification dataset to create disjoint set (clustering). The goal is to use features of flowers to create clusters that has similarities.
2) For the MNIST dataset, the goal is the same. Using the features of digits dataset, I try to cluster those instances that has similarities. Again, as in iris dataset, we do not use the class labels. We will create class labels.

K-means algorithm

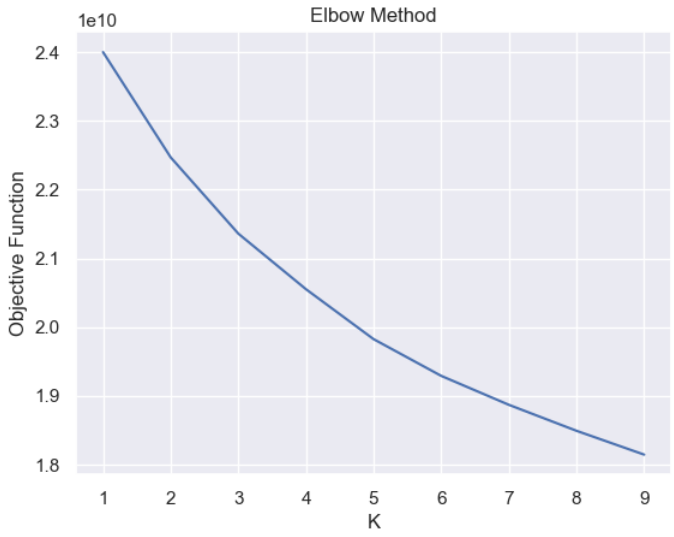
- **n_cluster**
  - Default value for this variable is 8.
  - For the iris dataset, I set this variable to three which is the total different classes.
  - This gave me 0.89 cluster purity. However, when I increase this number to k, cluster purity is dramatically decreased (0.60).
  - It is expected to decrease because it tries to create clusters from a cluster.
  - However, for the mnist dataset, reducing the number of n_cluster from 10 to 3 gave me increased results.
  - **This happens because we do not use all instances of the mnist dataset. We only use 0.1 percent of it.**

ELBOW METHOD FOR DECIDING OPTIMAL K

- Elbow method is used determine the optimal k for the given dataset.
- The y axis of the elbow plot is n_cluster while x axis is the objective function
- In k-means, our goal is to decrease the objective function.



- From the above graph, we can see that for the iris dataset, the optimal number is 3 where we see the elbow with greatest decrease in objective function.



- For the elbow graph above, it is hard to find the optimal k for the algorithm because dataset is not complete.

AGGLOMARATIVE CLUSTERING SCIPY

- By looking at the table above, both Scipy and Sklearn implementation of the agglomerative clustering did good job on iris dataset. Scipy implementation only 5 seconds above compare to the Sklearn implementation.
- However, on the mnist dataset, Scipy implementation did great job by resulting 0.99 cluster purity.

- I got the above results using the **average** linkage. Now, lets see the results when we use different linkage methods.

- Using "**SINGLE**" linkage:

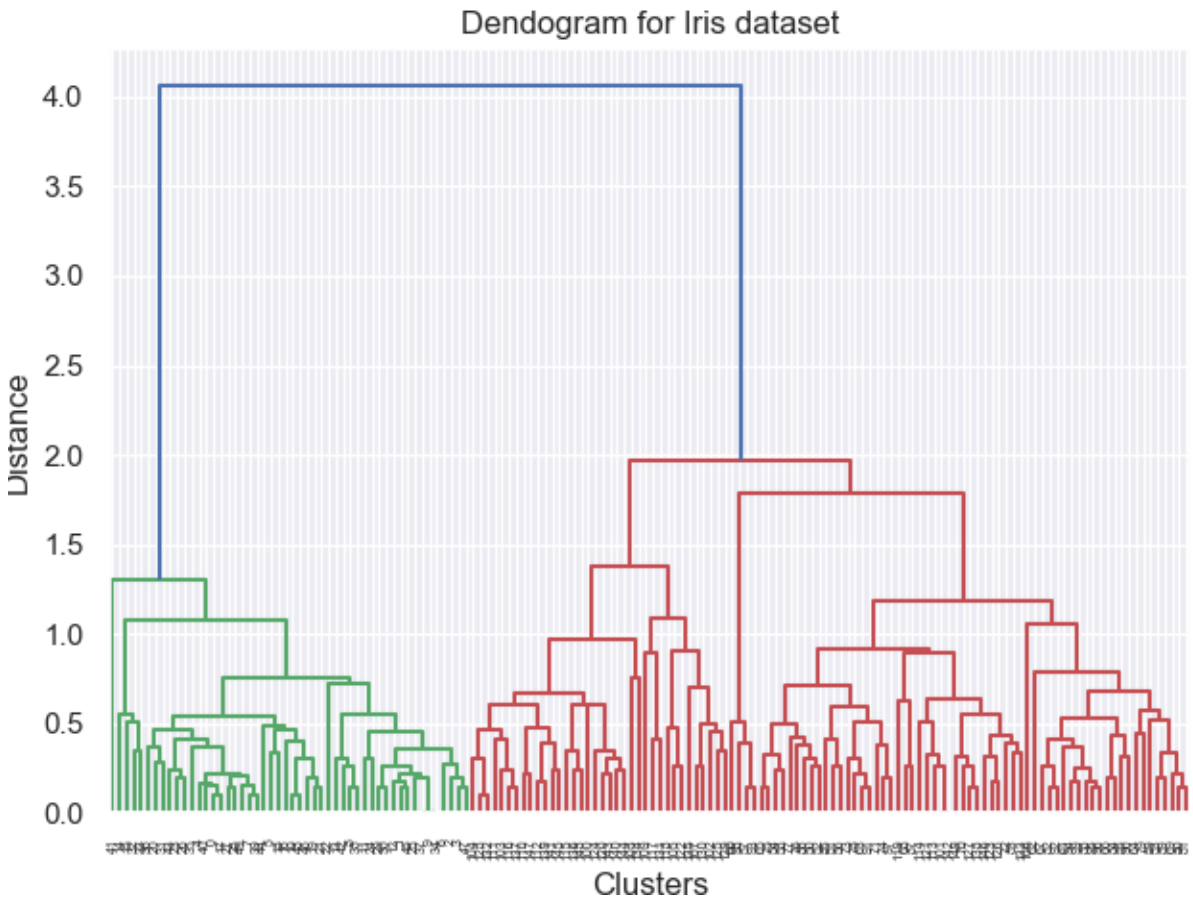
    - I increased the cluster purity from 0.90 to 0.98 on iris dataset!
    - For the mnist dataset, using this linkage method kept the purity score same, 0.99!

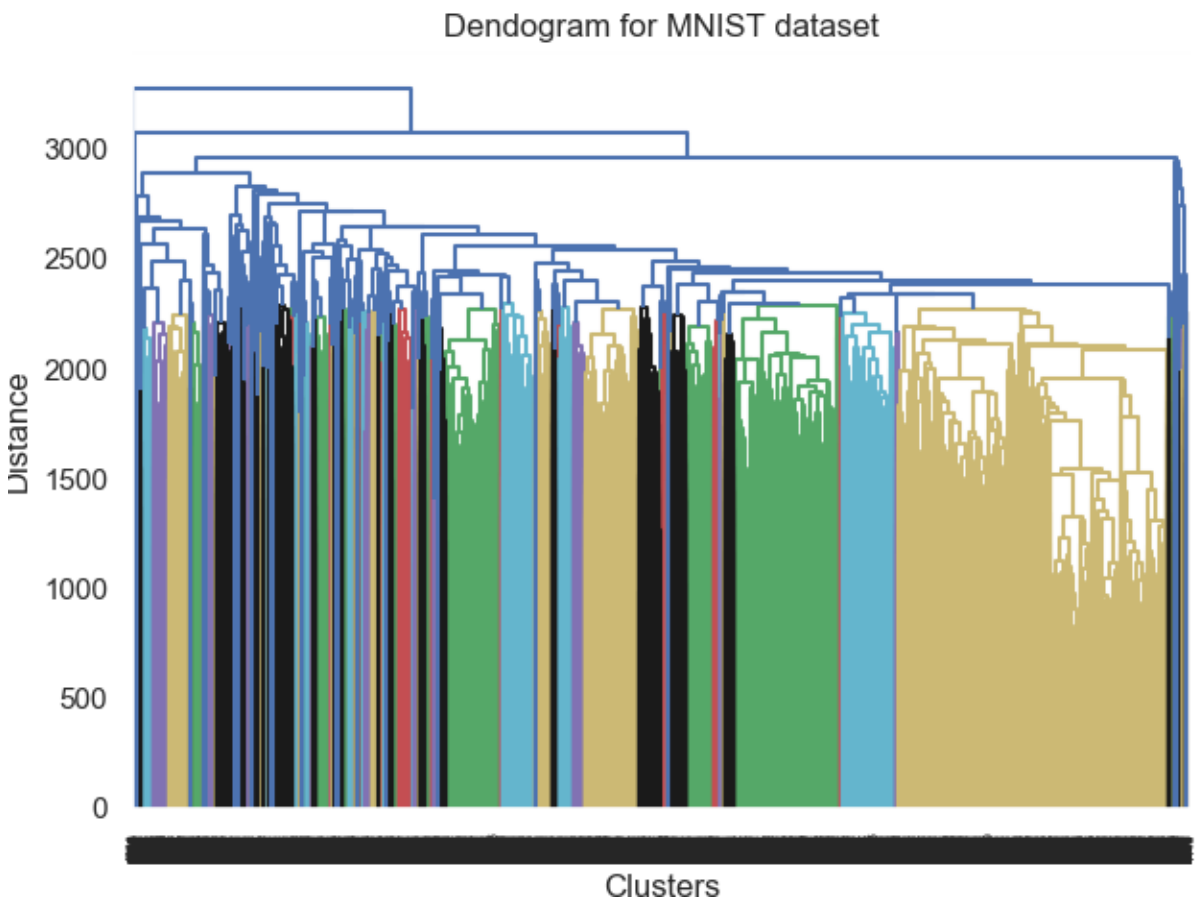
- Using "**COMPLETE**" linkage:

    - I increased the cluster purity from 0.90 to 0.98!
    - For the mnist dataset, using this linkage method lowered purity score from 0.90 to 0.80.
    - For the iris dataset, this linkage method gave me 0.80 purity score.

DENDOGRAM ANALYSIS

- Dendrogram plots the way clustering is made and which instances are clustered together. However, it is only useful when we have small number of instances like iris. As we see from the plot below, it is not useful on mnist dataset.





AGGLOMARATIVE CLUSTERING SKLEARN

- This approach on iris dataset gave me the lowest purity among all clustering algorithms.
- It did same thing for mnist dataset except k means algorithm.
- I got the above results using the default linkage method.

- Using "**SINGLE**" linkage:
    - I increased the cluster purity from 0.89 to 0.98 on iris dataset!
    - For the mnist dataset, using this linkage method kept the purity score same, 0.99!

- Using "**COMPLETE**" linkage:

- For the mnist dataset, using this linkage method lowered purity score from 0.90 to 0.50.
- For the iris dataset, this linkage method gave me 0.49 purity score. **Overall, this is the lowest I have since in this homework for the iris.csv dataset.**

DBSCAN Algorithm

- For this algorithm, I created a function called optimal which returns the optimal epsilon and min_samples for the DBSCAN algorithm.
- Here are the optimal numbers for these hyperparameters

Iris Dataset

- **The optimal epsilon and the min sample value is 0.1 5**
- **Using these values, we get following cluster purity 1.0**

Mnist dataset

- **The optimal epsilon and the min sample value is 0.5 3**
- **Using these values, we get following cluster purity 1.0**

EPSILON and MINSAMPLES

- **Eps:** The distance to consider when we find neighbor.
- **Min_samples:** minimum required number to consider as core point
- By changing these values to the optimum numbers, we see that we can get the purest cluster among all clustering algorithms.

FINAL THOUGHTS

- I think in terms of robustness, k means algorithm did the worst job while DBSCAN algorithm did the best job.
- Hierarchical clustering algorithms also did relatively better job compare to k means algorithm