CS 573: Assignment 5

Tiantu Xu

1. **Exploration**

   In Question 1, data exploration is run by the command line below:

   ```
   $ python exploration.py
   ```

   1. The randomly selected digits from digits-raw.csv are visualized as below:

Figure 1: Randomly Sampled Digits

   2. The 1000 randomly selected examples in 2d from the digits-embedding.csv is colored as below:

Figure 2: 1000 randomly selected examples

2. K-means Clustering

2.1 Code

My code is run as the command below:

```
$ python kmeans.py digits-embedding.csv 10
```

The output from my code is

```
WC_SSD: 1433531.47
SC: 0.71
NMI: 0.36
```

2.2 Analysis

1. My code is run as the command below.

   ```
   $ python kmeans-analysis_2_12.py
   ```

   By clustering the data from 3 datasets, the plot of the WC_SSD and SC curve is shown below, and each column represents one dataset.

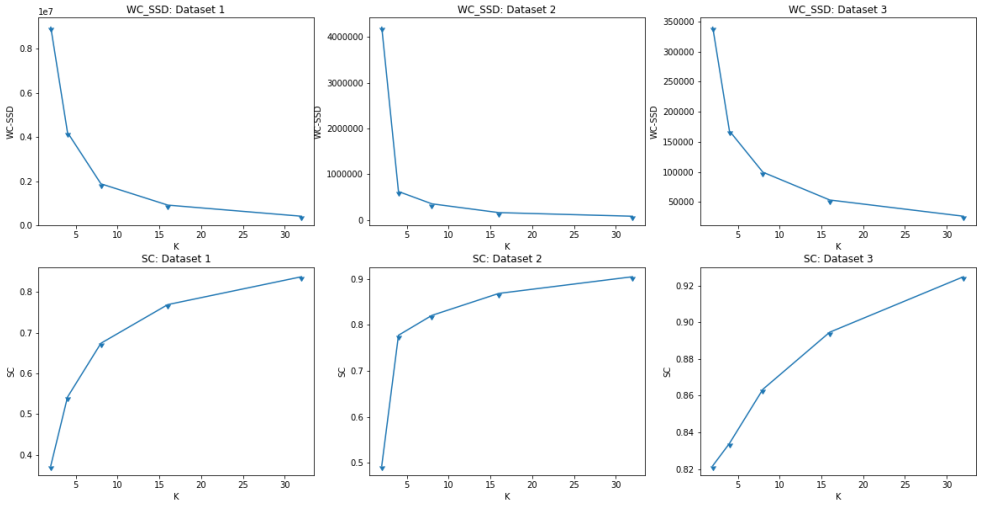

Figure 3: WC_SSD and SC for 3 datasets

2. Using the result from Step 1, I look for the "elbow point" to choose the correct K for each dataset. For dataset 1, I choose K = 8 because WC_SSD does not have significant reduction after K = 8, and SC tends to be stable after K = 8. For dataset 2, I choose K = 4, because both WC_SSD and SC tends to converges after K = 4. For dataset 3, I choose K = 2, because SC is already very close to 1 and does not significantly go up after K = 2.
   In three datasets, SC_SSD monotonically increases as K increases, and SC monotonically decreases as K decreases. For the comparison across the datasets, WC_SSD tends to be smaller from data set 1 to dataset 3, and SC tends to be closer to 1 from dataset 1 to dataset 3.

3. By repeating Step 1 for 10 times by using random seeds from $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, I got the following average and standard deviation of WC_SSD and SC for each K. My code is run as the command below:

   ```
   $ python kmeans-analysis_2_3.py
   ```

   Each column represents one dataset. k-means is more sensitive when the initial centroid number is small (K is small), because in the initial stage, fewer initial centroid tends to be more random in the large 2D space, thus will have a larger standard deviation.

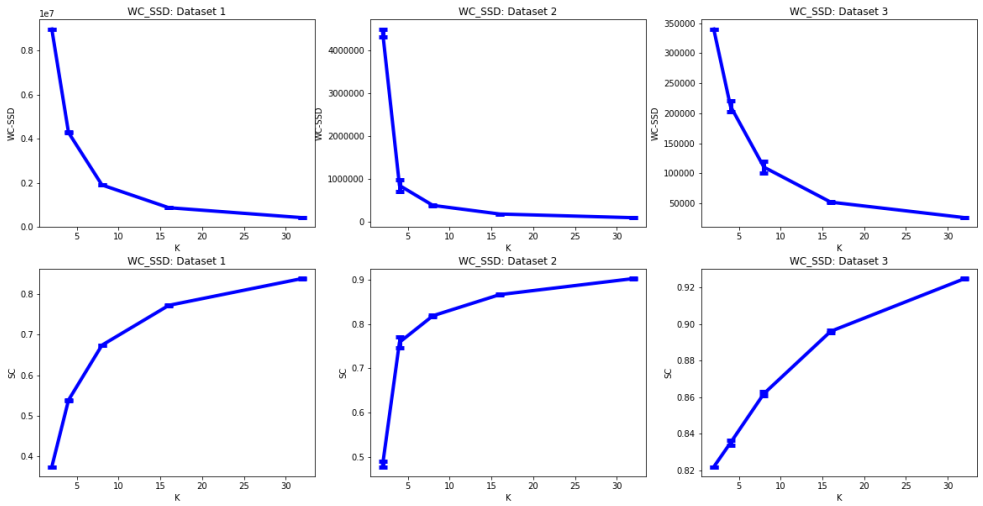

Figure 4: Average and standard deviation of WC_SSD and SC for 3 datasets

4. By choosing K as 8, 4, 2 for each dataset, respectively, we get the following NMI for each dataset, and the clustering result is shown in the figure below. My code is run as the command below:

   ```
   $ python kmeans-analysis_2_4.py
   ```

Figure 5: Clustering results by choosing K as 8, 4, 2 for three datasets

From the clustering results shown in the figure, the clusters have a larger boundary from data set 1 to dataset 3, when the choices of K is smaller.
The output NMI from my code is shown below.

```
Dataset 1: K = 8
NMI: 0.35
```

```
Dataset 2: K = 4
NMI: 0.45
Dataset 3: K = 2
NMI: 0.49
```

By comparing the NMI result, the conclusion is that (dataset 3, K = 2) provides the better clustering result than (dataset 2, K = 4), and is better than (dataset 1, K = 8).

3. **Hierarchical Clustering**
   In question 3, the code is run as below.

```
$ python clustering.py
```

1. The dendrogram below shows the clustering result using the scipy agglomerative single linkage. The model performance on 3 models (DT, BT, and RF) is shown below.

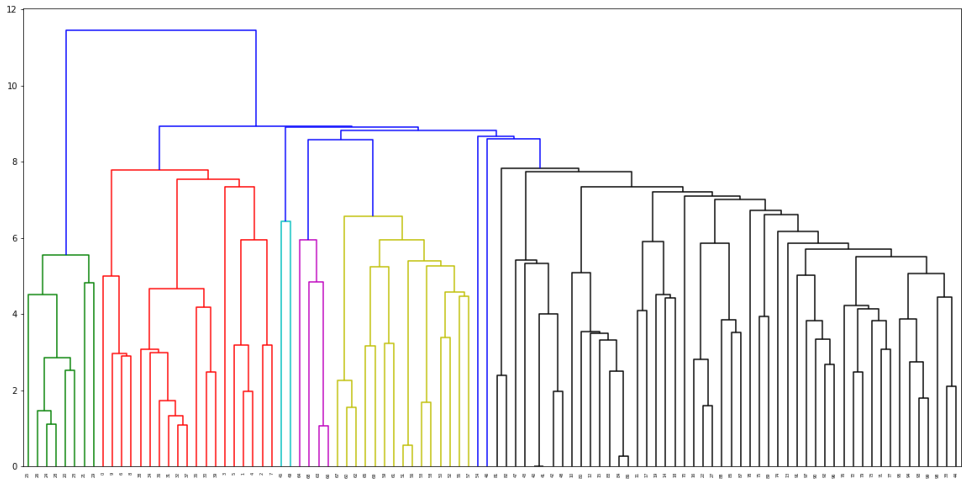

Figure 6: Dendrogram using single linkage

2. The dendrogram below shows the clustering result using the scipy agglomerative complete linkage.

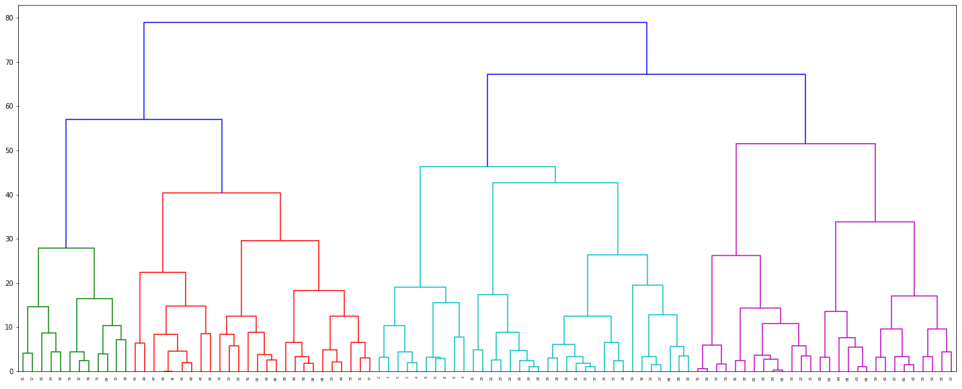

Figure 7: Dendrogram using complete linkage

The dendrogram below shows the clustering result using the scipy agglomerative average linkage.

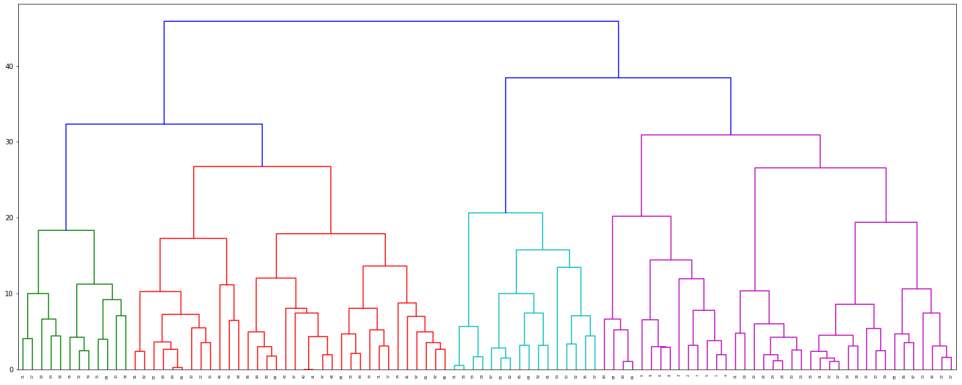

Figure 8: Dendrogram using average linkage

3. By using the K in Section 2, the WC_SSD and the SC using three different linkage is plotted as below.

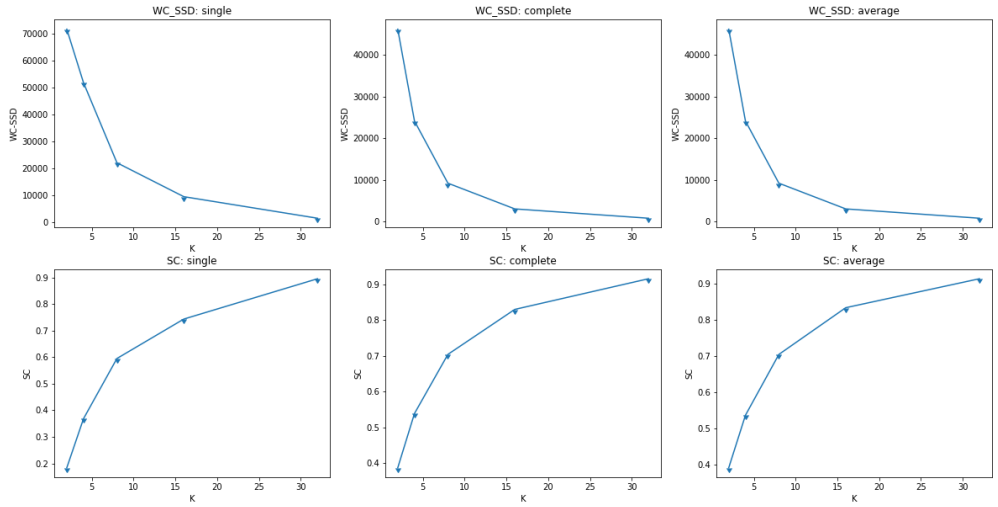

Figure 9: Dendrogram using average linkage

4. I look for the "elbow point" to choose the correct K for each linkages. For single linkage, the WC_SSD and the SC tends to converge after K = 8; For complete and average, the WC_SSD does not have significant drop after K = 8, so that we choose K = 8. In conclusion, I should K as 8, 8, 8 for single linkage, complete linkage, and average linkage, respectively. This choice is the same as the K I chose in Dataset 1 in Section 2.

5. The NMI can be found in the output of my code:

```
single K: 8
NMI: 0.32
complete K: 8
NMI: 0.36
average K: 8
NMI: 0.34
```

The results tends to be similar across 3 distance measures. The NMI on dataset 1 in Section 2 is 0.35, which is very close to the result using hierarchical clustering in Section 3.