CS 4140/6140: Data Mining HW 4

Novella Alvina - u1413401

February 2023

Part I

Hierarchical Clustering

There are many variants of hierarchical clustering; here we explore only 2: single link (measuring shortest link) and complete link (measuring longest link)

1 Question 1.A

Run the two hierarchical clustering variants on dataset C1.txt until there are k = 3 clusters, and report the results as sets. For example, you could report a table with columns for each cluster and two rows for two linkage criteria or a figure made by using scatter in matplotlib.

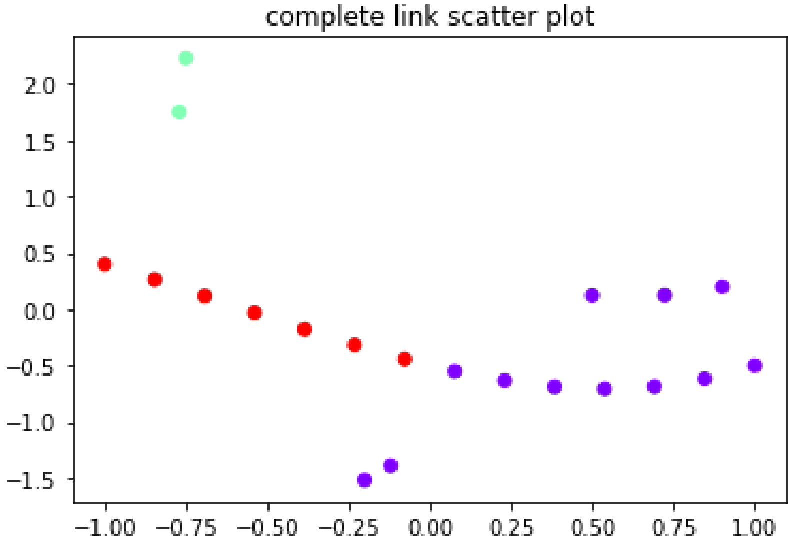


Figure 1: complete link scatter plot

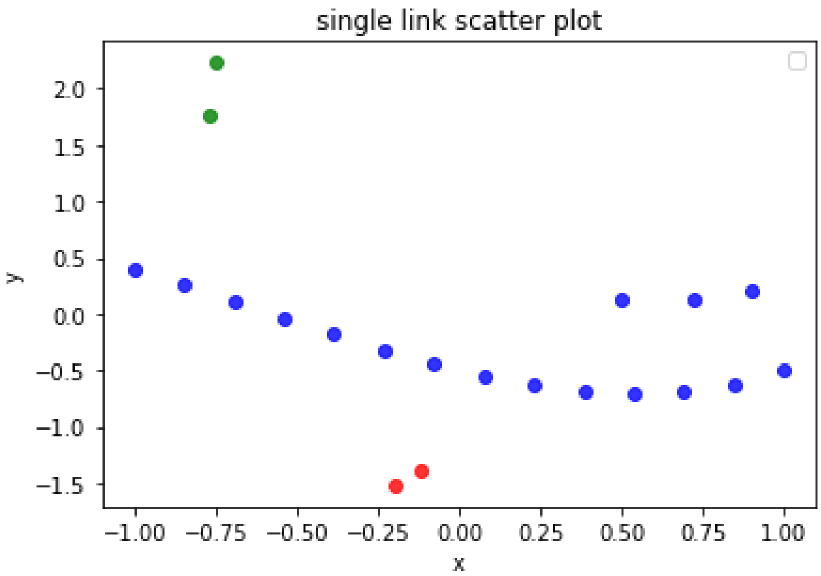
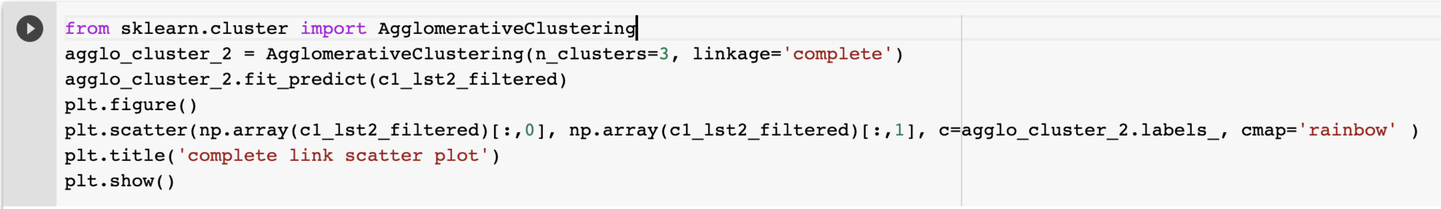


Figure 2: single link scatter plot

```
[array([ 1.      , -0.75    ,  2.23129]), array([ 2.        , -0.77      ,  1.7523932])]
[array([ 3.      , -0.2     , -1.51712]), array([ 4.      , -0.12    , -1.38923])]
[array([5.    , 0.5   , 0.123]), array([6.    , 0.7234, 0.125 ]), array([7. , 0.9, 0.2]),
array([ 8. , -1. ,  0.4]), array([ 9.        , -0.8461538,  0.2630405]),
array([10.        , -0.6923077,  0.1173418]), array([11.        , -0.5384615, -0.031634 ])
array([12.        , -0.3846154, -0.1784251]), array([13.        , -0.2307692, -0.3175694])
array([15.        ,  0.0769231, -0.5510696]), array([16.        ,  0.2307692, -0.6345016])
array([17.        ,  0.3846154, -0.6884388]), array([18.        ,  0.5384615, -0.7074192])
array([19.        ,  0.6923077, -0.6859809]), array([20.        ,  0.8461538, -0.6186618])
array([21. ,  1. , -0.5])]
```

2 Question 1.B



```
from sklearn.cluster import AgglomerativeClustering
agglo_cluster_2 = AgglomerativeClustering(n_clusters=3, linkage='complete')
agglo_cluster_2.fit_predict(c1_lst2_filtered)
plt.figure()
plt.scatter(np.array(c1_lst2_filtered)[:,0], np.array(c1_lst2_filtered)[:,1], c=agglo_cluster_2.labels_, cmap='rainbow' )
plt.title('complete link scatter plot')
plt.show()
```

Figure 3: agglomerative clustering using sklearn

Part II

Assignment-Based Clustering

Two good initialization methods for this type of clustering are the Gonzalez (Algorithm 8.2.1 in M4D book) and k-means++ (Algorithm 8.3.2) algorithms.

3 Question 2.A

Run Gonzalez and k-means++ on data set C2.txt for k = 4. To avoid too much variation in the results, choose c1 as the point with index 1.

For Gonzalez, report the centroids and clusters (make a figure using scatter in matplotlib) and 4-center cost and 4-means cost

```
centroids = {1, 1001, 1029, 343}
4-center cost: 3.481123664686921
4-mean cost: 1.600536590805209
clusters: [array([-2.7694973,  2.6778586]), array([-2., 14.]), array([-0.4032861, -5.447
```

4 Question 2.B

Since k-means++ is a randomized algorithm, you will need to report the variation.

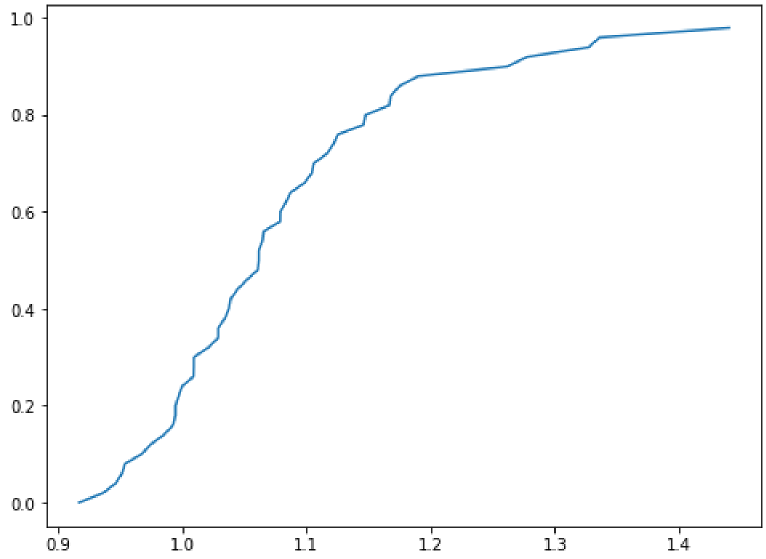


Figure 4: cumulative density function of the 4-means cost of k-means++

```
The standard deviation of center cost: 2.725887784911488
The standard deviation of mean cost: 0.10766825889070293
The fraction of subsets similar to Gonzalez: 0.14
```

5 Question 2.C

Recall that Lloyd's algorithm for k-means clustering starts with a set of k centers C and runs as described in Algorithm 8.3.1 (in M4D).

5.1 Question 2.C.1

Run Lloyds Algorithm with C initially with points indexed 1,2,3,4. Report the final clusters and the 4-means cost.

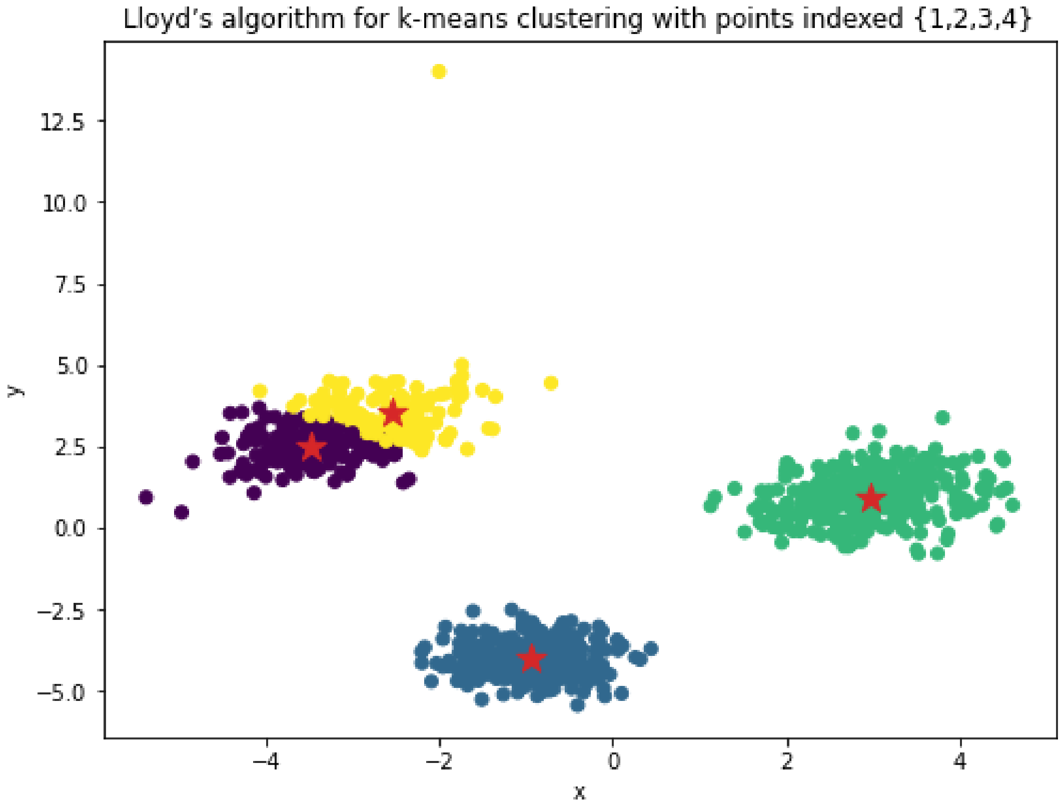


Figure 5: Lloyd's algorithm for k-means clustering with points indexed 1,2,3,4

5.2

Question 2.C.2 Run Lloyds Algorithm with C initially as the output of Gonzalez above. Report the final clusters and the 4-means cost.

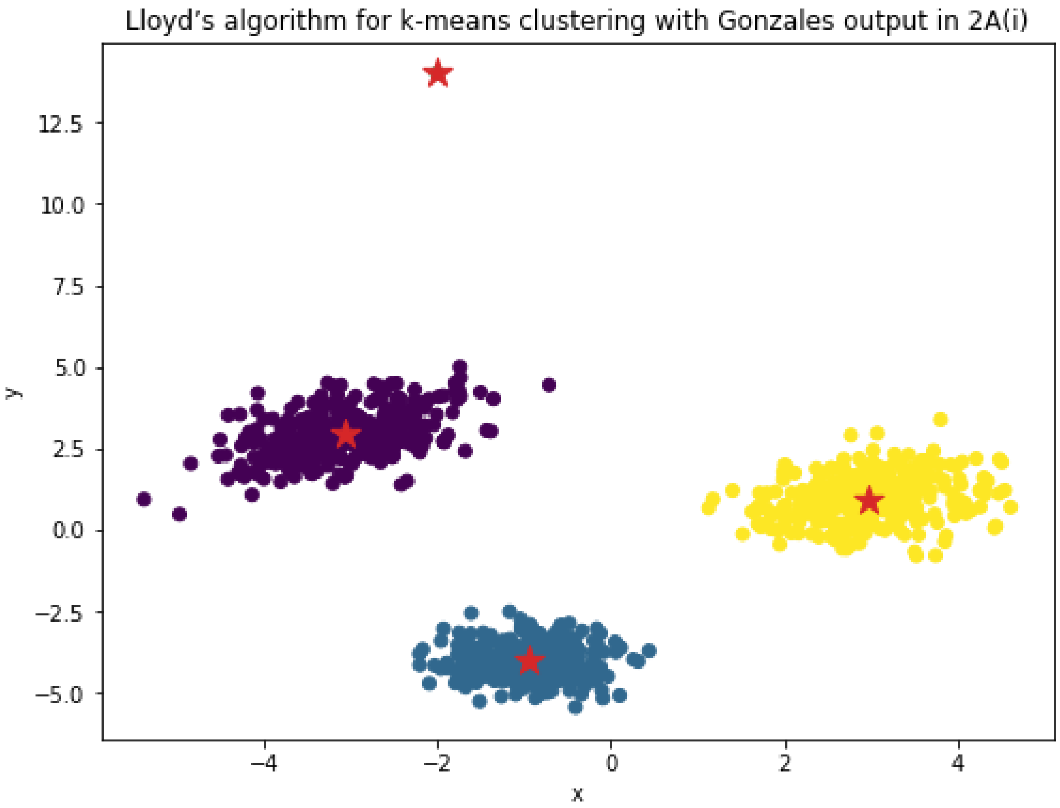


Figure 6: Lloyd’s algorithm for k-means clustering with output from 2A(i)

5.3 Question 2.C.3

Run Lloyds Algorithm with C initially as the output of each run of k-means++ above. Plot a cumula- tive density function of the 4-means cost. Also report the fraction of the trials that the subsets are the same as the input (where the input is the result of k-means++).

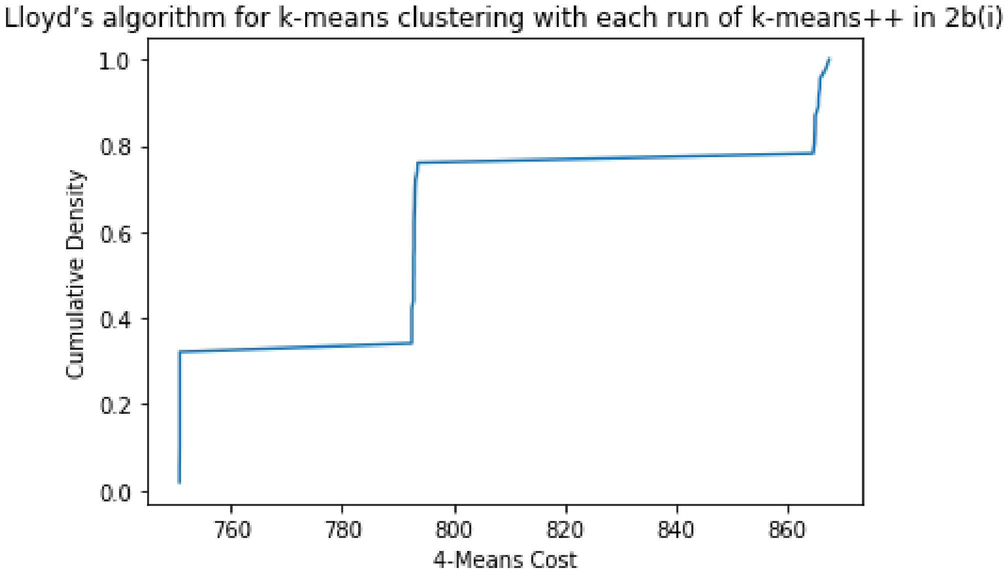


Figure 7: Lloyd’s algorithm for k-means clustering with each run of k-means++ in 2B(i)

```
The fraction of subsets similar to result of k-means++: 0.0
```