Nicholas Nhat Tran 1002027150

CSE 4309 Assignment 6

Task 1 (70 pts)

k_means("toy_data/set1a.txt", 2, "round_robin")

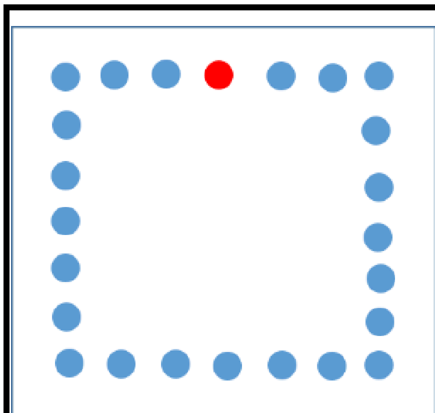
```
   7.0000 --> cluster 2
  29.0000 --> cluster 1
  11.0000 --> cluster 2
   2.0000 --> cluster 2
  16.0000 --> cluster 1
   4.0000 --> cluster 2
  37.0000 --> cluster 1
  22.0000 --> cluster 1
```

k_means("toy_data/set2a.txt", 2, "round_robin")

```
(   61.1056,   516.9514) --> cluster 1
(   86.1015,   467.5088) --> cluster 1
(   86.1056,   354.9514) --> cluster 1
(  120.1015,   151.5088) --> cluster 2
(  123.1056,   516.9514) --> cluster 1
(  129.5719,   209.0000) --> cluster 1
(  142.1015,   398.5088) --> cluster 1
(  148.1015,   467.5088) --> cluster 1
(  171.1015,    98.5088) --> cluster 2
(  198.1056,   441.9514) --> cluster 1
(  219.1056,   204.9514) --> cluster 2
(  229.5771,   102.5427) --> cluster 2
(  247.1015,   500.5088) --> cluster 1
```

(373.1056, 266.9514) --> cluster 2
(450.1056, 165.9514) --> cluster 2
(466.1056, 79.9514) --> cluster 2
(502.5719, 230.0000) --> cluster 2
(510.5719, 40.0000) --> cluster 2
(523.1056, 116.9514) --> cluster 2
(533.1056, 189.9514) --> cluster 2
(569.1056, 120.9514) --> cluster 2

Task 2 (10 pts)



Consider the set of points above. Each dot corresponds to a point. Consider a clustering consisting of two clusters, where the first cluster is the set of all the blue dots, and the second cluster has the red dot as its only element. Can this clustering be the final result of the k-means algorithm? Justify your answer.

Your answer should be based on your visual estimations of Euclidean distances between dots. For this particular question, no greater precision is needed.

This clustering **cannot** be the final result of the k-means algorithm because there are blue dots that are significantly closer to the red centroid than the blue centroid, therefore switching clusters for some blue dots.

Task 3 (10 pts)

Part a

Part a: Will the EM algorithm always give the same results when applied to the same dataset with the same K? To phrase the question in an alternative way, is there any dataset where the algorithm can produce different results if run multiple times, with the value of K kept the same? If your answer is that EM will always give the same results, justify why. If your answer is that EM can produce different results if run multiple times, provide an example.

No, the EM algorithm will not always give the same results when applied to the same dataset with the same K because the EM algorithm is an iterative algorithm that is very sensitive to its starting conditions. The EM algorithm always randomly initializes its K cluster starting parameters, meaning the means, covariances, and other parameters will be different on each separate run of the algorithm.

Part b

Part b: Same question as in part a, but for agglomerative clustering with the d_{min} distance. Here, as "result" we consider all intermediate clusterings, between the first step (with each object being its own cluster) and the last step (where all objects belong to a single cluster). Will this agglomerative algorithm always give the same results when applied to the same dataset? If your answer is "yes", justify why. If your answer is "no", provide an example.

Yes, agglomerative clustering will always give the same results when applied to the same dataset because the type of clustering is deterministic. It always begins in the same state where all data points have their own cluster and generates pairs of clusters with the smallest distances from each other with no possibility of ties.

Task 4 (10 pts)

Consider a dataset consisting of these eight points: 2, 4, 7, 11, 16, 22, 29, 37.

Part a: Show the results (all intermediate clusterings) obtained by applying agglomerative clustering to this dataset, using the d_{min} distance.

Part b: Show the results (all intermediate clusterings) obtained by applying agglomerative clustering to this dataset, using the d_{max} distance.

Part a: Single Linkage

Step 0: {2}, {4}, {7}, {11}, {16}, {22}, {29}, {37}
Step 1: {2, 4}, {7}, {11}, {16}, {22}, {29}, {37}
Step 2: {2, 4, 7}, {11}, {16}, {22}, {29}, {37}
Step 3: {2, 4, 7, 11}, {16}, {22}, {29}, {37}
Step 4: {2, 4, 7, 11, 16}, {22}, {29}, {37}
Step 5: {2, 4, 7, 11, 16, 22}, {29}, {37}
Step 6: {2, 4, 7, 11, 16, 22, 29}, {37}
Step 7: {2, 4, 7, 11, 16, 22, 29, 37}

Part b: Complete Linkage

Step 0: {2}, {4}, {7}, {11}, {16}, {22}, {29}, {37}
Step 1: {2, 4}, {7}, {11}, {16}, {22}, {29}, {37}
Step 2: {2, 4}, {7, 11}, {16}, {22}, {29}, {37}
Step 3: {2, 4}, {7, 11}, {16, 22}, {29}, {37}
Step 4: {2, 4}, {7, 11}, {16, 22}, {29, 37}
Step 5: {2, 4, 7, 11}, {16, 22}, {29, 37}
Step 6: {2, 4, 7, 11, 16, 22}, {29, 37}
Step 7: {2, 4, 7, 11, 16, 22, 29, 37}