Email : LE000288@umn.edu
ID : 5674954

Question 1

(a)

Partitional, fuzzy, complete

(b)

Partitional, exclusive, partial

(c)

Hierarchical, overlapping, complete

Question 2

(a)

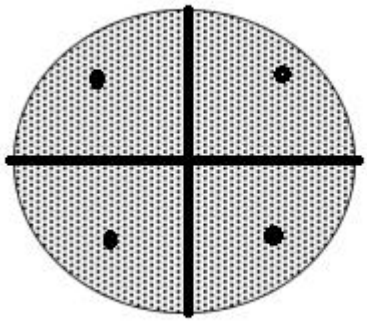

Figure 1.

Since the shaded regions in each of the figures have uniform density of points, the centroids in their respective quarter sector would locate on the 45-degree line and be closer to the circle periphery.

There are infinite ways to partition this dataset, because it is a circle. Each solution is a global minimum.

(b)

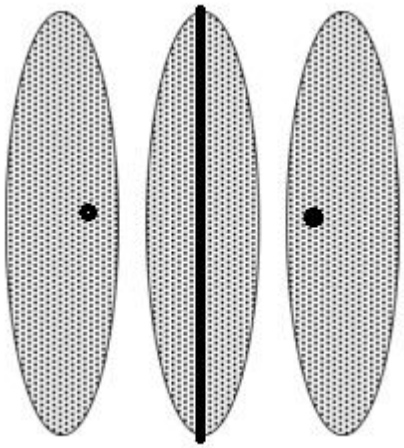

Figure 2(a).

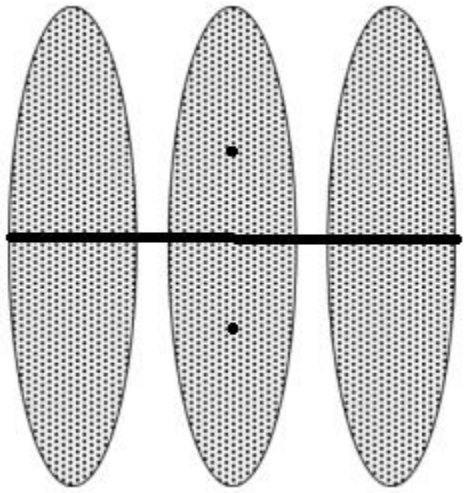

Figure 2(b).

There are two solutions. For the first one, the k-means separate the dataset vertically. The centroids would locate in the right side of the first ellipse and the left side of the third ellipse, respectively. First one is global minimum. For the second one, the k-means separate the dataset horizontally. The centroids would locate in the top of the second ellipse and the bottom of the second ellipse. Second one is local minimum.

(c)

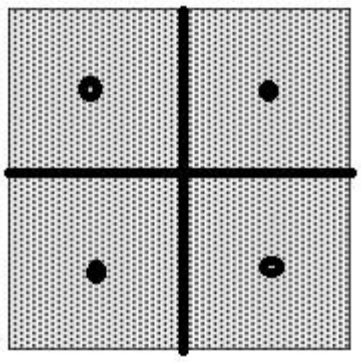

Figure 3(a)

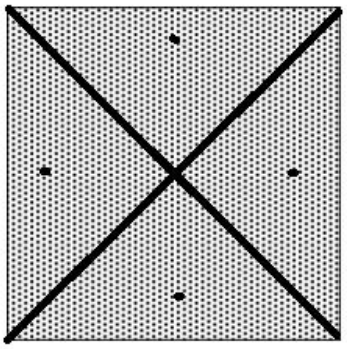

Figure 3(b)

There are two solutions. For the first one, the centroids in their respective quarter square would locate on the 45-degree line and locate in the middle of the quarter square. This is a global minimum. For the second one, the centroids in their respective quarter triangle would locate on the 45-degree line and be closer to the circle periphery. This is a local minimum.

Question 3

(a)

	1	2	3	4	5
1	0	3	5	2.24	9.22
2	3	0	4	4.47	7.62
3	5	4	0	4.47	4.24
4	2.24	4.47	4.47	0	8.6
5	9.22	7.62	4.24	8.6	0

(b)

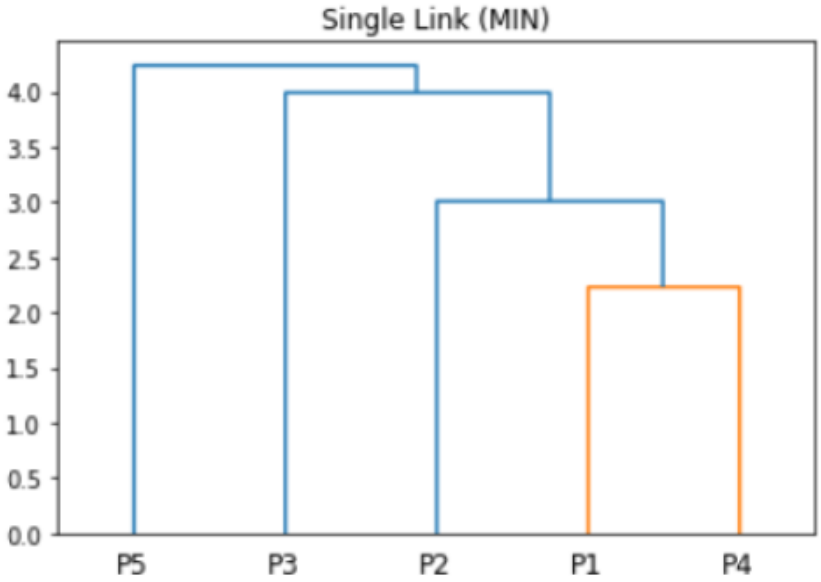

Figure 4

1. The smallest distance: Distance(P1, P4) = 0. Combine P1 with P4.
2. The smallest distance: Distance(P1,P2) = 3. Combine {P1, P4} with P2.
3. The smallest distance: Distance(P2,P3) = 4. Combine {{P1,P4},P2} with P3.
4. Combine {{{P1,P4},P2},P3} with P5.

(c)

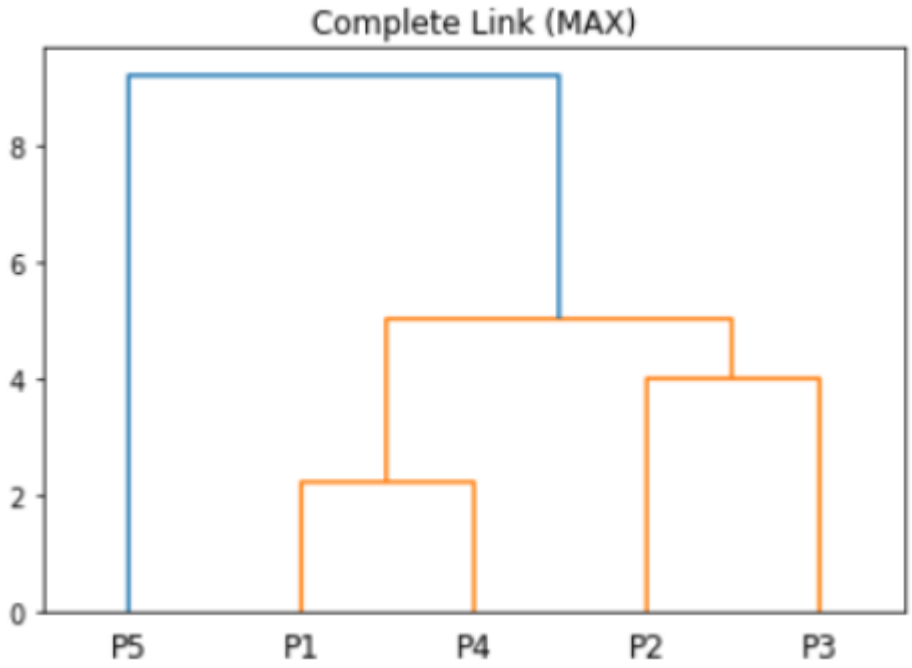

Figure 5

1. The smallest distance: Distance(P1, P4) = 2.24. Combine P1 with P4.
2. The smallest distance: Distance(P2,P3) = 4. Combine P2 with P3.
3. Distance({P1,P4},{P2,P3}) = 5. Distance({P1,P4},{P5}) = 9.22. Distance({P2,P3}{P5}) = 7.62. Combine {P1,P4} with {P2,P3} and combine {{P1,P4},{P2,P3}} with P5.

Question 4

(i) Fig.(a)

Corresponding dataset: (e)
Explanation: In the distance matrices, the $(i,j)^{th}$ entry in the matrix corresponds to the distance between point i and point j. All distances between points in the Fig.(a) are very low, thus, we know that the four points gather.

(ii) Fig.(b)

Corresponding dataset: (f)
Explanation: From the first row and the last row, we know that there is a increasing trend in the differences of distance for point a and decreasing trend in the differences of distance for point d. From the second row, we know that the distance between point 1 and point 2 equals to the distance between point 2 and point 3. From the third row, we know that the distance between point 2 and point 3 equals to the distance between point 3 and point 4.
So we can determine that the corresponding dataset would be (f).

(iii) Fig.(c)

Corresponding dataset: (d)
Explanation: From the distance matrix, we know that every point has same different distances to the other two points and very high distance to the rest point. So, the Fig.(d) corresponds to the Fig.(c).

Question 5

(a)

Number of Centroids in Circle (a): 2
Number of Centroids in Circle (b): 0
Number of Centroids in Circle (c): 0
Brief explanation:
The 100 points from B and C would keep attracting the right centroid. After many iterations, the right centroid would move to the place between B and C.

(b)

Number of Centroids in Circle (a): 1

Number of Centroids in Circle (b): 1
Number of Centroids in Circle (c): 1
Brief explanation:
The first step of the k-means algorithm is assigning all points to the closest centroid. So, 50000 points from A would be assigned to the left centroid and 50 points from C would be assigned to the right centroid. Then, the centroids would be recomputed. After iterations, the centroids would separate to each cluster respectively.

(c)
Number of Centroids in Circle (a): 1
Number of Centroids in Circle (b): 1
Number of Centroids in Circle (c): 1
Brief explanation:
The three centroids would not move. All points from A would be assigned to the left centroid, all points from B would be assigned to the middle centroid, and all points from C would be assigned to the right centroid.

Question 6

For DBSCAN, the running time of DBSCAN is O(n^2). For this complex case, DBSCAN might take long time to determine the Eps and MinPts. Furthermore, varying densities is another problem DBSCAN faced. For complete-link, the noise and outliers would be the challenge. Although complete-line is less susceptible to noise and outliers compare to single-link, the performance of resisting noise of complete is not very good. Furthermore, complete-link tends to break large clusters and biased towards globular clusters.

I prefer DBSCAN for this specific data set, it can handle clusters of different shapes and sizes and handle noise and outliers. Moreover, this data set clearly shows that the upper block is denser than the lower block and the points inside a boundary are denser than the points outside the boundary. The performance of DBSCAN would be really good if we find the suitable Eps and MinPts.

Question 7
(a)

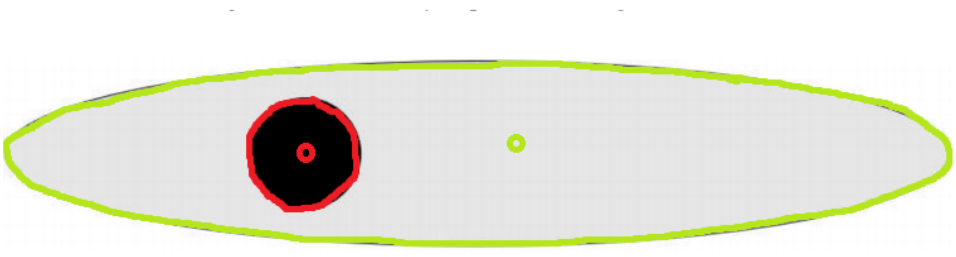

Figure 6

Figure 6 shows how the two-dimensional points split into two clusters by Gaussian mixture model clustering.

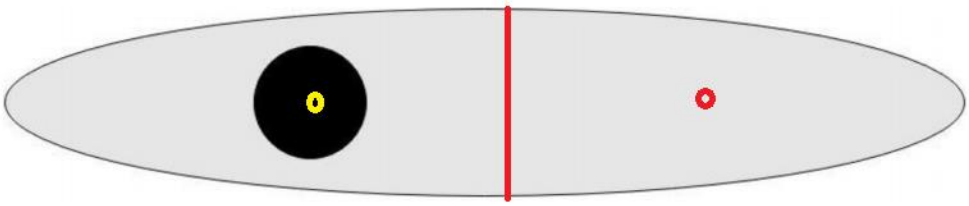

Figure 7

Figure 7 shows how the two-dimensional points split into two clusters by K-means.

(b)

SNN Clustering.

Question 8

(a)

Similarity:

They both specify a fixed number of clusters.

Difference:

SOM imposes a topographic ordering on the centroids and nearby centroids are also updated while the k-means do not have the ordering on the centroids and only update the nearest centroid.

(b)

Similarity

Both algorithms produce hierarchical clustering.

Difference:

Bisecting k-means is divisive clustering while hierarchical clustering using group average is agglomerative clustering.

(c)

Similarity:

Both algorithms based on density.

Difference:

SNN density-based clustering regard the number points that have an SNN

similarity of Eps or greater to each point as the density. For DENCLUE, contribution of each point to the density is given by an influence or kernel function.

(d)

Advantage:

SNN similarity also addresses problems with clusters of varying density. The SNN similarity of a pair of points only depends on the number of nearest neighbors two objects share, not how far these neighbors are from each object. Thus, SNN similarity performs an automatic scaling with respect to the density of the points.

Disadvantage:

The SNN similarity would ignore the distance between data points and this may causes mistakes.