RECITATION PROBLEMS

Q4. Given K equally sized clusters, the probability that a randomly chosen initial centroid will come from any given cluster is lf K, but the probability that each cluster will have exactly one initial centroid is much lower. (It should be clear that having one initial centroid in each cluster is a good starting situation for K-means.) In general, if there are K clusters and each cluster has n points, then the probability, p, of selecting in a sample of size K one initial centroid from each cluster is given by Equation 8.20. (This assumes sampling with replacement.)

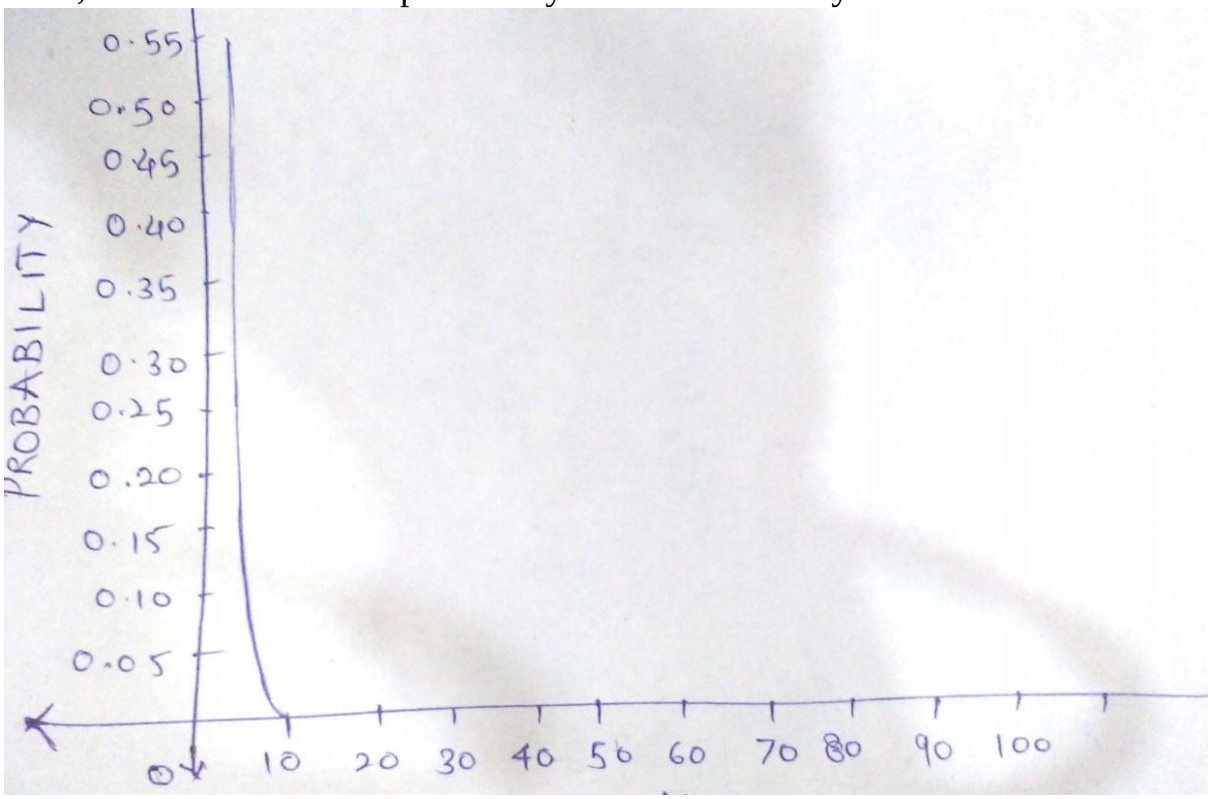
From this formula we can calculate, for example, that the chance of having one initial centroid from each of four clusters is $4!/4^4 = 0.0938$.

P=no. of ways to select one centroid from each cluster/no. of ways to select k centroid $=K!/K^K$

(a) Plot the probability of obtaining one point from each cluster in a sample of size K for values of K between 2 and 100.

Answer:

Here, we can observe that probability value reaches 0 by the time K is 10.



(b) For K clusters, K : 10,100, and 1000, find the probability that a sample of size 2K(contains at least one point from each cluster. You can use either mathematical methods or statistical simulation to determine the answer.

Answer:

By simulation, the probabilities are 0.21, < 10−6, and < 10−6.

Analytically,

Probability of a point does not come from a cluster ➔ $(1-1K)$

and thus,
Probability that all $2K$ points don't come from a cluster ➔$(1-1K)^{2K}$.
Hence,
the probability that at least one of the 200 points comes from a cluster ➔ $(1-(1-1K)^{2K})$.
So, if we assume independence, then an upper bound for the probability that all clusters are represented in the final sample ➔ $(1-(1-1K)^{2K})^{K}$.
The values given by this bound are 0.27, 5.7e-07, and 8.2e-64.

Q7. Suppose that for a data set

- there are m points and K clusters,
- half the points and clusters are in "more dense" regions,
- half the points and clusters are in "less dense" regions, and
- the two regions are well-separated from each other.

For the given data set, which of the following should occur in order to minimize the squared error when finding if clusters:
(a) Centroids should be equally distributed between more dense and less dense regions.
(b) More centroids should be allocated to the less dense region.
(c) More centroids should be allocated to the denser region.
Note: Do not get distracted by special cases or bring in factors other than density. However, if you feel the true answer is different from any given above, justify your response.
Answer:
(c). More centroids should be allocated to the denser region because the less dense regions need more centroids if the squared error is to be minimized.

Q11. Total SSE is the sum of the SSE for each separate attribute. What does it mean if the SSE for one variable is low for all clusters? Low for just one cluster? High for all clusters? High for just one cluster? How could you use the per variable SSE information to improve your clustering?

Answer:
If the SSE of one attribute is low for all clusters ➔ The variable basically must be a constant and cannot be use to separate data into groups
If the SSE of one attribute is relatively low for just one cluster ➔ The attribute helps in identifying that cluster.
If the SSE of an attribute is relatively high for all clusters ➔ It could mean that the attribute is noise.
If the SSE of an attribute is relatively high for one cluster ➔ There is a likelihood that the information provided by the attributes with low SSE differs from that which defines the cluster. It could merely be the case that the clusters

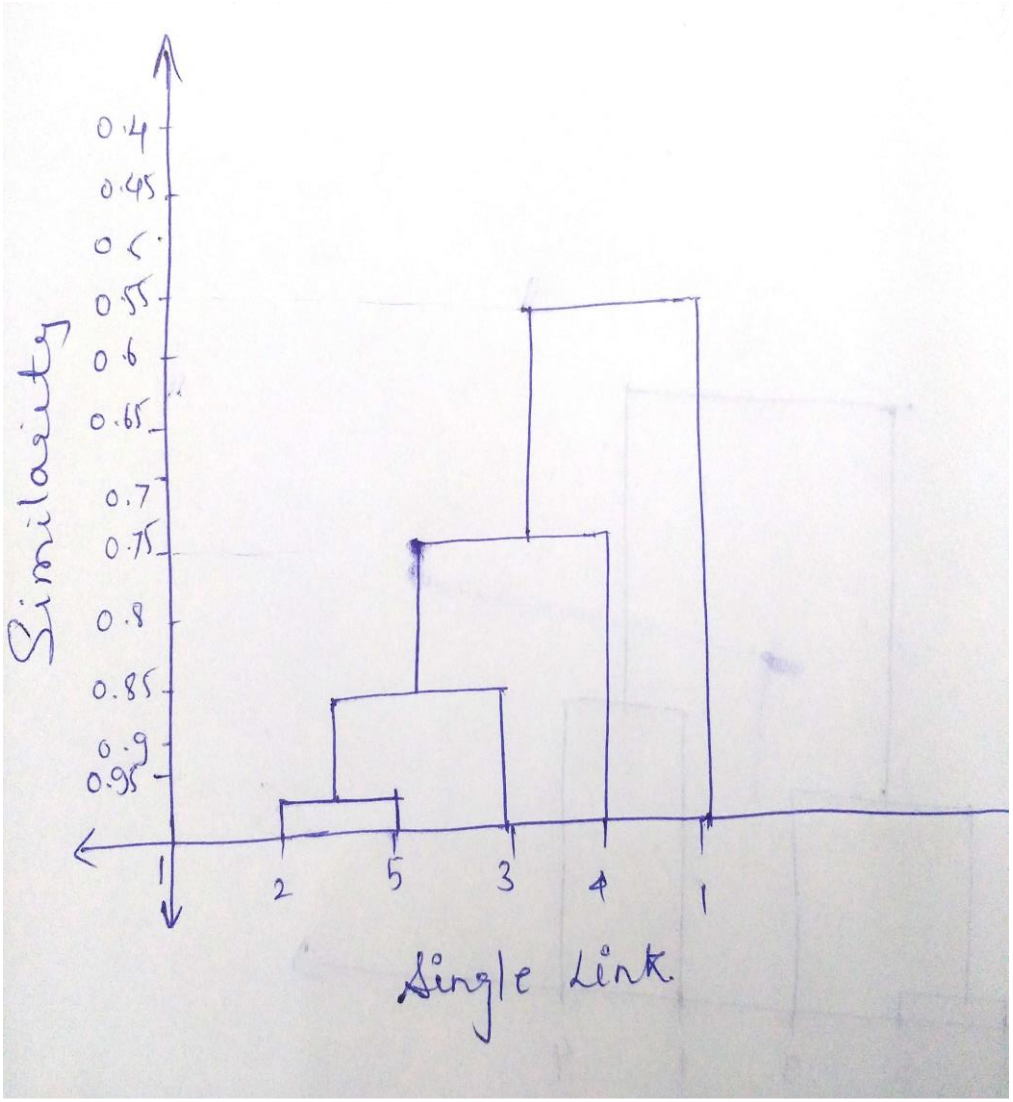
defined by this attribute are different from those defined by the other attributes; but in any case, it means that this attribute does not help define the cluster.

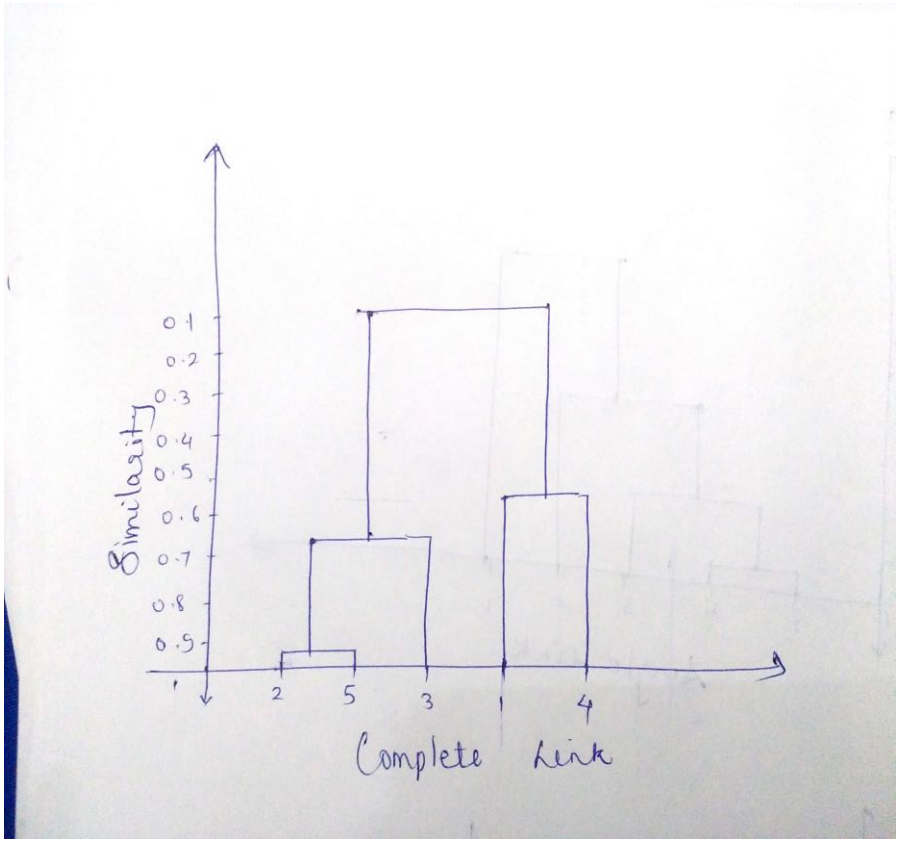
Here, the idea is to eliminate attributes that have poor distinguishing power between clusters for all clusters. To improve our clustering the attributes with high SSE for all clusters which a have relatively high SSE with respect to other attributes should be eliminated.

Q17. Hierarchical clustering is sometimes used to generate K clusters, K > I by taking the clusters at the Kth level of the dendrogram. (Root is at level 1.) By looking at the clusters produced in this way, we can evaluate the behaviour of hierarchical clustering on different types of data and clusters, and compare hierarchical approaches to K-means.
The following is a set of one-dimensional points: {6,12,18,24,30,42,48}.

Answer:
Similarity Matrix :

	P1	P2	P3	P4	P5
P1	1.0	0.10	0.41	0.55	0.35
P2	0.10	1.0	0.64	0.47	0.98
P3	0.41	0.64	1.0	0.44	0.85
P4	0.55	0.47	0.44	1.0	0.76
P5	0.35	0.98	0.85	0.76	1.0


Similarity
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1
2
5
3
4
1
Single Link



(a) For each of the following sets of initial centroids, create two clusters by assigning each point to the nearest centroid, and then calculate the total squared error for each set of two clusters. Show both the clusters and the total squared error for each set of centroids.

1. {18,45}

Answer:

First cluster is 6, 12, 18, 24, 30.
Error = 360.
Second cluster is 42, 48.
Error = 18.
Total Error = 378

2. {15,40}

Answer:

First cluster is 6, 12, 18, 24.
Error = 180.
Second cluster is 30, 42, 48.
Error = 168.
Total Error = 348.

(b) Do both sets of centroids represent stable solutions; i.e., if the K-means algorithm was run on this set of points using the given centroids as the starting centroids, would there be any change in the clusters generated?

Answer: Yes, both centroids are stable solutions.

(c) What are the two clusters produced by single link?
Answer: The two clusters are {6, 12, 18, 24, 30} and {42, 48}.

(d) Which technique, K-means or single link, seems to produce the "most natural" clustering in this situation? (For K-means, take the clustering with the lowest squared error.)
Answer: The most natural clustering is produced by Min technique.

(e) What definition(s) of clustering does this natural clustering correspond to? (Well-separated, centre-based, contiguous, or density.)
Answer: MIN produces contiguous clusters. Also, density can be considered. Even centre-based can be thought of as correct.

(f) What well-known characteristic of the K-means algorithm explains the previous behavior?
Answer:
K-Means works well if the data set has values which creates well separated clusters. It works poor otherwise for finding different shapes. The idea of minimizing squared error creates small clusters. Thus, in this problem, the low error clustering solution is unnatural.

Q21. Compute the entropy and purity for the confusion matrix in Table 8.14

Table 8.14. Confusion matrix for Exercise 21.

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Total
#1	1	1	0	11	4	676	693
#2	27	89	333	827	253	33	1562
#3	326	465	8	105	16	29	949
Total	354	555	341	943	273	738	3204

Answer:
Entropy -
Cluster 1 - 0.20
Cluster 2 - 1.84
Cluster 3 – 1.70
Total – 1.44
Purity –

Cluster 1 - 0.98
Cluster 2 – 0.53
Cluster 3 – 0.49
Total – 0.61

Q22. You are given two sets of 100 points that fall within the unit square. One set of points is arranged so that the points are uniformly spaced. The other set of points is generated from a uniform distribution over the unit square.

(a) Is there a difference between the two sets of points?
Answer:
Yes. The random points will have regions of lesser or greater density, while the uniformly distributed points will, of course, have uniform density throughout the unit square.

(b) If so, which set of points will typically have a smaller SSE for K:10 clusters?
Answer:
For K:10, the random set of points will have a lower SSE.

(c) What will be the behavior of DBSCAN on the uniform data set? The random data set?
Answer:
For the uniform dataset, it will be treated like a single cluster and all the members will be classified into it accordingly. However, this might vary per the threshold and can also be classified as noise.
In random data set, DBSCAN will find new shapes or patterns and classify them into clusters as density varies.

Q23. Using the data in Exercise 24, compute the silhouette coefficient for each point, each of the two clusters, and the overall clustering.

Answer:

	P1	**P2**	**P3**	**P4**
P1	0	0.10	0.65	0.55
P2	0.10	0	0.70	0.60
P3	0.65	0.70	0	0.30
P4	0.55	0.60	0.30	0

Let *a* indicate the average distance of a point to other points in its cluster.
Let *b* indicate the minimum of the average distance of a point to points in another cluster.
Point P1: SC = 1- a/b = 1 - 0.1/((0.65+0.55)/2)= 5/6 = 0.833
Point P2: SC = 1- a/b = 1 - 0.1/((0.7+0.6)/2) = 0.846
Point P2: SC = 1- a/b = 1 - 0.3/((0.65+0.7)/2) = 0.556
Point P2: SC = 1- a/b = 1 - 0.3/((0.55+0.6)/2) = 0.478
Cluster 1 Average SC = (0.833+0.846)/2 = 0.84
Cluster 2 Average SC = (0.556+0.478)/2 = 0.52
Overall Average SC = (0.840+0.517)/2 = 0.68

Q24. Given the set of cluster labels and similarity matrix shown in Tables 8.15 and 8.16, respectively, compute the correlation between the similarity matrix and the ideal similarity matrix, i.e., the matrix whose ijth entry is 1 if two objects belongs to the same cluster, and 0 otherwise.

Answer:
Here, for Point P1, cluster label is 1, P2 is 1, P3 is 2, P4 is 2.
Similarity matrix can be given as,

Point	P1	P2	P3	P4
P1	1	0.8	0.65	0.55
P2	0.8	1	0.7	0.6
P3	0.65	0.7	1	0.9
P4	0.55	0.6	0.9	1

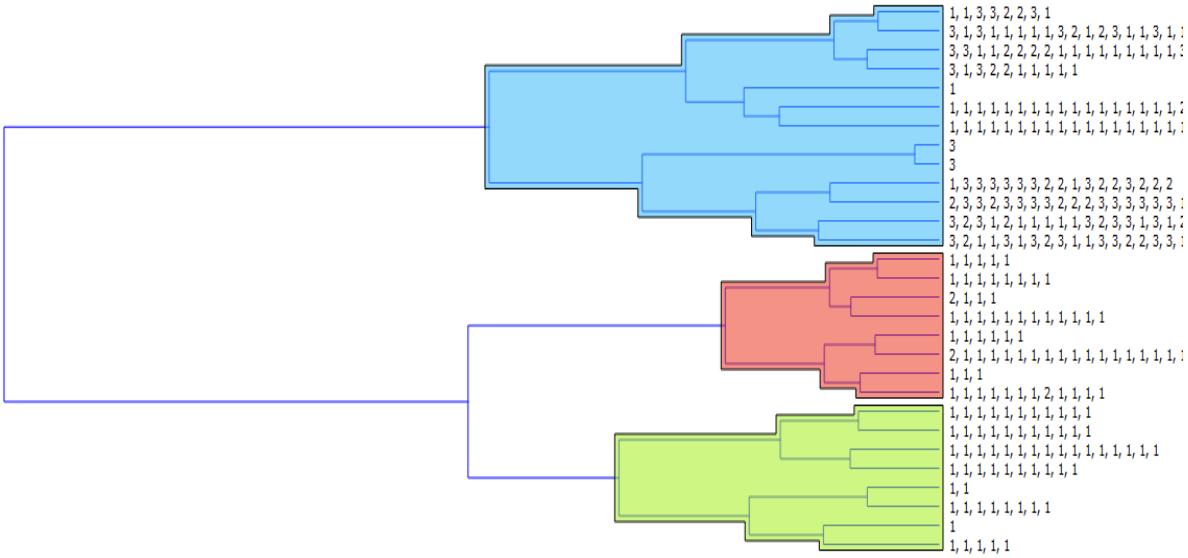
We need to compute the correlation between the off-diagonal elements of the distance matrix and the ideal similarity matrix.
We get:
Standard deviation of the vector **x** : $\sigma x = 0.5164$
Standard deviation of the vector **y** : $\sigma y = 0.1703$
Covariance of **x** and **y**: $\text{cov}(\mathbf{x}, \mathbf{y}) = -0.200$
Therefore, $\text{corr}(\mathbf{x}, \mathbf{y}) = \text{cov}(\mathbf{x}, \mathbf{y})/\sigma x \sigma y = -0.227$

PRACTICUM PROBLEMS

2.1 Problem 1

Load the auto-mpg sample dataset into the Orange application - ensure that origin is set as a target attribute type, as it will be used as a class label. Perform a Hierarchical Clustering using Linkage set to Average, after calculating Distances, with Pruning set to a Max Depth of 5. Also, set Selection to Top N with a value of 3. This will result in a shallow tree of depth 5, and a final cut resulting in 3 clusters. Examine the resulting clusters (C1,C2,C3) via Distributions analysis - is there a clear relationship between the cluster assignment and class label (1,2,3)? What are the probabilities calculated for each value of origin for each cluster? Does changing the Max Depth affect the results in any way?

Answer:



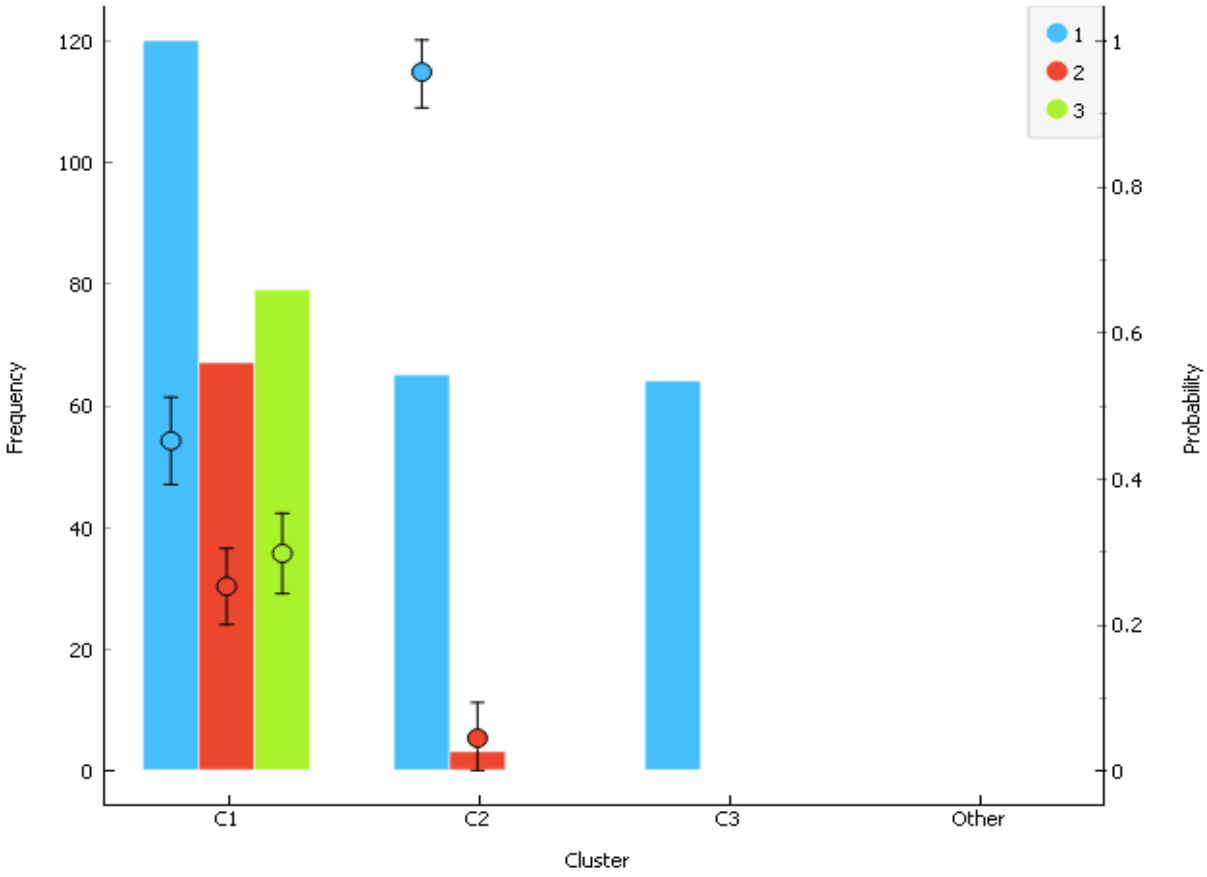
From distribution analysis, we can see that there is no clear relationship between cluster assignments and class labels.

Clusters are not necessarily formed per the class labels.

Only cluster C3 is pure having only class 1.

Whereas, C2 and C1 are having all the classes into it.

C3 has more entropy as compared to Cluster C2.



The probabilities calculated for each value of origin of each clusters are as below:

For Cluster C1:

1 ➔ 0.451 + or - 0.060

2 ➔ 0.252 + or - 0.052

3 ➔ 0.297 + or - 0.055

For Cluster C2:

1 ➔ 0.956 + or - 0.049

2 ➔ 0.044 + or - 0.049

For Cluster C3:

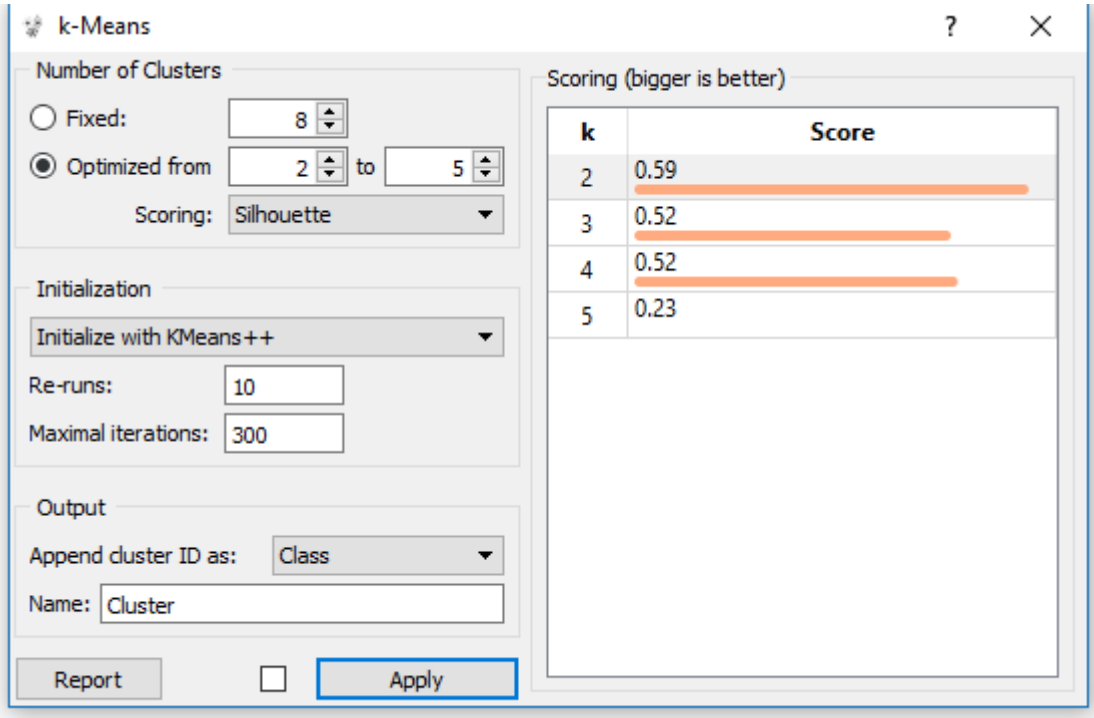
1 ➔ 1

Changing the max depth does not change the result in anyway.

2.2 Problem 2

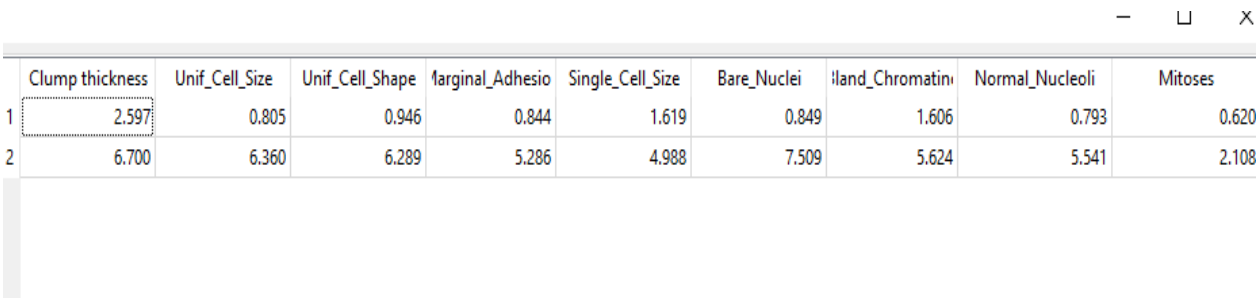
Load the breast-cancer-wisconsin-cont dataset into the Orange application, and run a k-means analysis with the number of clusters Optimized from values for k from 2 to 5. Use Silhouette scoring - what is the score for each value of k? For the best score, what are the coordinates of the centroids? What are the distances between the centroids for the best score?

Answer:



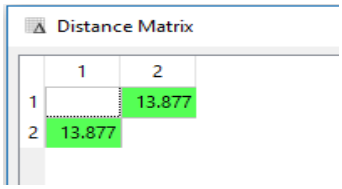
As seen the best silhouette score is obtained for 2 clusters.

The centroid of the clusters for optimized value of k=2 is as seen in the figure below.



	Clump thickness	Unif_Cell_Size	Unif_Cell_Shape	Marginal_Adhesio	Single_Cell_Size	Bare_Nuclei	Bland_Chromatin	Normal_Nucleoli	Mitoses
1	2.597	0.805	0.946	0.844	1.619	0.849	1.606	0.793	0.620
2	6.700	6.360	6.289	5.286	4.988	7.509	5.624	5.541	2.108

The distance between centroid for the best score is 13.877



2.3 Problem 3

Load the Boston dataset (sklearn.datasets.load boston()) into Python using a Pandas dataframe. Perform a K-Means analysis on unscaled data, with the number of clusters ranging from 2 to 6. Provide the Silhouette score to justify which value of k is optimal. What information do the values of Homogeneity/Completeness provide as well? Calculate the mean values for all features in each cluster for the optimal clustering - how do these values differ from the centroid coordinates?

Answer:

```
range_n_clusters = [2,3,4,5,6]

n_clusters = 0
for n_clusters in range_n_clusters:


    clusterer = KMeans(n_clusters=n_clusters,  init='k-means++')
    clusterer.fit(df)
    cluster_labels = clusterer.fit_predict(df)


    silhouette_avg = metrics.silhouette_score(df, cluster_labels)

    print("For n_clusters =", n_clusters,
          "The average silhouette_score is :", silhouette_avg)
```

```
For n_clusters = 2 The average silhouette_score is : 0.691398118833
For n_clusters = 3 The average silhouette_score is : 0.723403034161
For n_clusters = 4 The average silhouette_score is : 0.568219170853
For n_clusters = 5 The average silhouette_score is : 0.570738665513
For n_clusters = 6 The average silhouette_score is : 0.501258930507
```

Silhouette score is high when we optimize to 3 clusters.
The values for homogeneity is 0.187370799835 and completeness is 0.629506604287.
Hence, when the value of completeness is more it states that all the data points that are members of a given class are elements of the same cluster.
Larger values of homogeneity and completeness are desirable.

Mean Values:
For Cluster 1

```
C0 = df.loc[df['CLUST'] == 0]
```

```
C0.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
count	11.000000	11.0	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000000	11.000(
mean	1.963207	0.0	16.708182	0.090909	0.707727	5.916091	91.818182	2.323691	4.727273	386.909091	17.000000	187.546364	17.212
std	0.912947	0.0	5.457133	0.301511	0.159456	0.312366	7.972555	0.874302	0.467099	41.353245	3.191865	74.268586	6.0352(
min	0.228760	0.0	8.140000	0.000000	0.520000	5.272000	79.200000	1.419100	4.000000	307.000000	14.700000	70.800000	9.8100(
25%	1.500405	0.0	14.070000	0.000000	0.571500	5.733000	84.000000	1.679800	4.500000	393.500000	14.700000	128.950000	13.580(
50%	2.149180	0.0	19.580000	0.000000	0.624000	5.950000	94.000000	2.283400	5.000000	403.000000	14.700000	227.610000	16.140(
75%	2.413010	0.0	19.580000	0.000000	0.871000	6.115500	98.450000	2.570300	5.000000	403.000000	20.950000	244.235000	18.825(
max	3.535010	0.0	21.890000	1.000000	0.871000	6.405000	100.000000	3.990000	5.000000	437.000000	21.200000	262.760000	27.800(

For Cluster 2

```
C1 = df.loc[df['CLUST'] == 1]
```

```
C1.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LS
count	80.000000	80.00000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80.000000	80
mean	0.081582	24.16875	6.226500	0.087500	0.464586	6.577125	49.226250	4.942381	3.550000	225.450000	17.892500	391.370625	8.4
std	0.071202	32.25609	6.345701	0.284349	0.048092	0.660163	24.182559	1.813620	1.330271	21.745886	1.513716	8.875882	5.6
min	0.013110	0.00000	0.460000	0.000000	0.385000	5.399000	2.900000	1.757200	1.000000	187.000000	13.600000	341.600000	1.9
25%	0.034833	0.00000	2.460000	0.000000	0.439000	6.012250	32.175000	3.917500	3.000000	216.000000	17.600000	389.632500	4.5
50%	0.057575	0.00000	5.070000	0.000000	0.449000	6.524500	45.750000	5.033750	3.000000	224.000000	17.900000	394.175000	6.8
75%	0.096653	40.00000	6.910000	0.000000	0.488000	7.004500	62.050000	5.873750	4.000000	243.000000	18.700000	396.900000	9.9
max	0.387350	100.00000	25.650000	1.000000	0.581000	8.034000	97.000000	12.126500	7.000000	265.000000	20.200000	396.900000	30

For Cluster 3

```
C2 = df.loc[df['CLUST'] == 2]
```

```
C2.describe()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
count	102.000000	102.0	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.000000	102.0000
mean	10.910511	0.0	18.572549	0.078431	0.671225	5.982265	89.913725	2.077164	23.019608	668.205882	20.195098	371.8030
std	12.120759	0.0	2.091641	0.270177	0.062720	0.722131	13.275049	0.672498	4.339504	9.763884	0.021698	35.00609
min	0.105740	0.0	18.100000	0.000000	0.532000	3.561000	40.300000	1.129600	4.000000	666.000000	20.100000	240.5200
25%	4.844605	0.0	18.100000	0.000000	0.614000	5.619500	87.675000	1.575675	24.000000	666.000000	20.200000	354.8475
50%	7.795775	0.0	18.100000	0.000000	0.693000	6.113000	95.350000	1.904700	24.000000	666.000000	20.200000	389.3650
75%	12.613775	0.0	18.100000	0.000000	0.713000	6.391250	98.775000	2.508125	24.000000	666.000000	20.200000	396.9000
max	88.976200	0.0	27.740000	1.000000	0.770000	8.780000	100.000000	4.098300	24.000000	711.000000	20.200000	396.9000

Centroid for K=3

```
centers = clust_model1.cluster_centers_
roundc= np.round(centers,1)
print(roundc)
```

```
[[  1.09000000e+01   0.00000000e+00   1.86000000e+01   1.00000000e-01
    7.00000000e-01   6.00000000e+00   8.99000000e+01   2.10000000e+00
    2.30000000e+01   6.68200000e+02   2.02000000e+01   3.71800000e+02
    1.79000000e+01   1.74000000e+01   0.00000000e+00]
 [  4.00000000e-01   1.57000000e+01   8.40000000e+00   1.00000000e-01
    5.00000000e-01   6.40000000e+00   6.04000000e+01   4.50000000e+00
    4.50000000e+00   3.11200000e+02   1.78000000e+01   3.83500000e+02
    1.04000000e+01   2.49000000e+01   1.50000000e+00]
 [  1.50000000e+01  -0.00000000e+00   1.79000000e+01   0.00000000e+00
    7.00000000e-01   6.10000000e+00   8.99000000e+01   2.00000000e+00
    2.25000000e+01   6.44700000e+02   1.99000000e+01   5.78000000e+01
    2.04000000e+01   1.31000000e+01   2.00000000e+00]]
```

Difference between mean values is not much for all features and centroid for each cluster.
Centroid is more precise than mean, it is used for cluster location.