CS5011: Programming Assignment #3

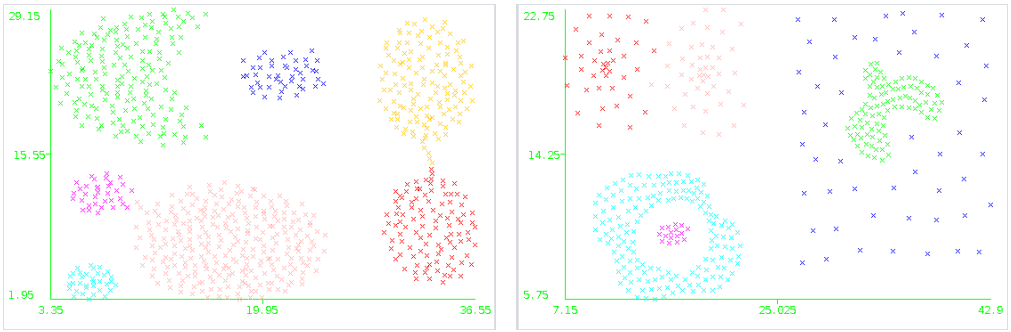
Girish Raguvir J, CS14B059, IIT Madras

November 2016

1 Converting to ARFF format

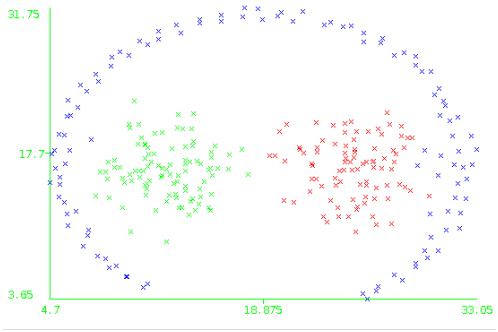
I wrote a python script to automate this conversion for all datasets. The python script *data_processing.py* can be found inside the Code folder. The converted datasets can be found inside the Dataset folder.

2 Analysis of Datasets

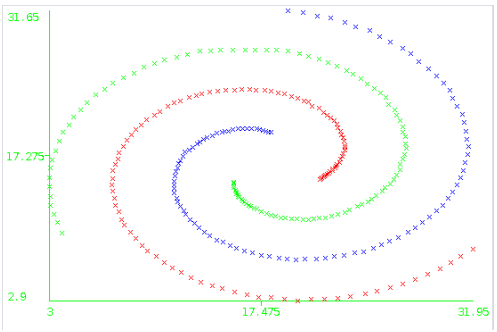


(a) Aggregation (b) Compound

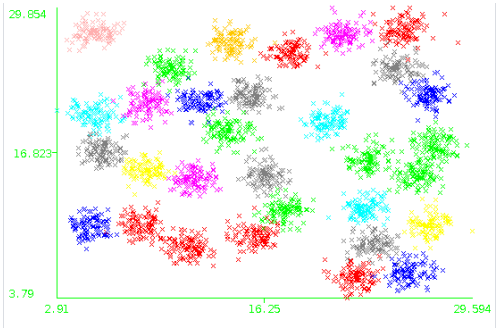
Figure 1: Visualisation of the given datasets



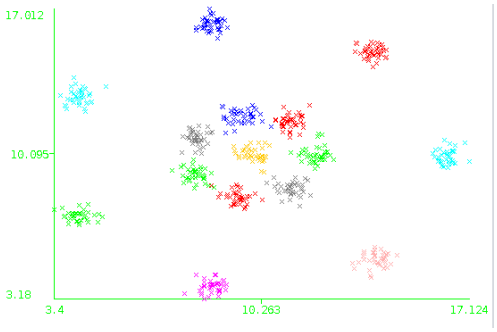
(a) Path-based



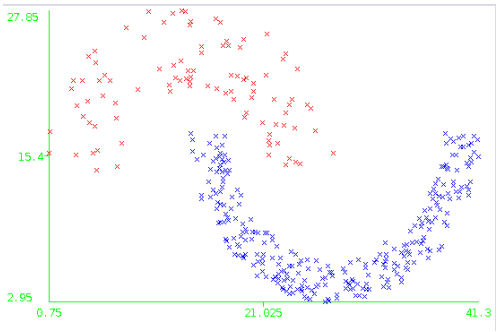
(b) Spiral



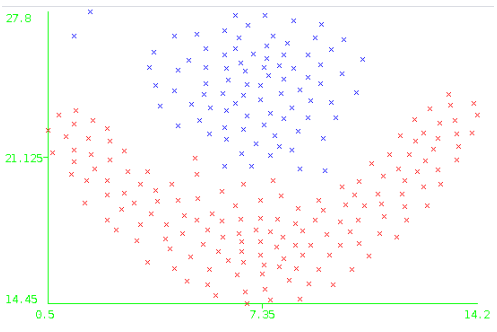
(c) D31



(d) R15



(e) Jain



(f) Flames

Figure 2: Visualisation of the given datasets

2.1 Aggregation Dataset

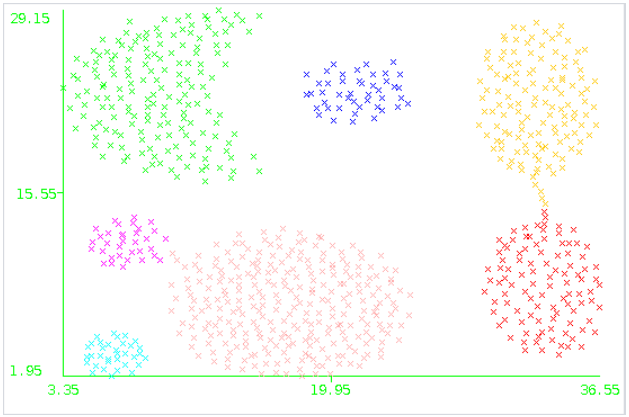


Figure 3: Aggregation.arff

2.1.1 K-Means

As we can see from the visualisation of above, some clusters are well separated while others are quite close with a thin link of points connecting some of them. However we can clearly see that this data has 7 clusters. In the standard K-means algorithm the initial centroids are randomly initialised (in most cases). Thus if the initial randomly chosen centroids are coincidentally closer to the centroids of the actual clusters, we may get back the same clustering but however if the intial centroids are chosen such that they start somewhere in between the closely placed clusters then the clusters obtained may not be the best. In that scenario we will end up breaking the actual clusters to form a new cluster with some points from the both the involved clusters. In the given scenario, the green and blue clusters are quite well separated from the others and hence , more often then not, they will be identified as individual clusters. However there might be some splitting and reshuffling between the closely placed (red, yellow) and (pink, rose, cyan) clusters depending the initialisation of the centroids.

2.1.2 DBScan

DBScan is a density based clustering algorithm. In the given dataset, we have distinct high dense regions with a very low density link or empty space

between the clusters. Due to this clear density gap between clusters and a somewhat unifrom density of points within each cluster, we can find an appropriate epsilon and minpoints value to demarcate the density gaps and hence precisely identify the clusters. There maybe many such values of epsilon and minpoints but one which I found to be hitting near perfection is e = 0.08 & minpoints = 24.

2.1.3 Hierarchical Clustering

Single Link: When single link is used for hierarchical clustering we pay attention solely to the area where the two clusters come closest to each other. Other more distant parts of the cluster and the clusters' overall structure are not taken into account. So in the given scenario, the clusters which are connected by a thin link of points will get merged into one cluster. So we will end up merging the red & yellowand pink & light pink & cyan clusters early on the dendogram. Since all other clusters are well seperated, all the others will form individual clusters. So ideally the hierarchical clustering would give us 5 big clusters. However if we want more clusters, the less dense points slowly start separating out and we will have more and more clusters (upper bounded by number of points).

Complete Link: In complete-linkage clustering, the link between two clusters contains all element pairs, and the distance between clusters equals the distance between those two elements (one in each cluster) that are farthest away from each other. Due to this reason, once again the the closely placed clusters are prone to reshuffling. But the green and the blue would most probably be once again rightly identified as individual clusters.

2.1.4 Inference

For this dataset, DBScan would be the most suitable option.

2.2 Compound Dataset

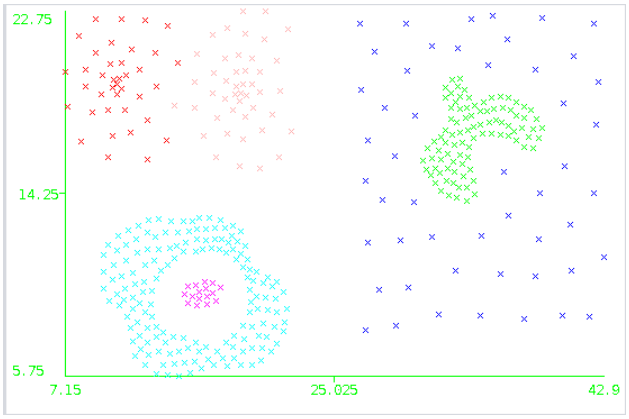


Figure 4: Compound.arff

2.2.1 K-Means

K-Means would perform very badly for this dataset. K-Means is inherently inefficient when it comes to identifying clusters like these. K-means would most probably end up putting the pink and cyan clusters together as one and segment the green & blue clusters into 2 halves horizontally. The red & light pink clusters may or may not be identified correctly depending on the centroid initialisations.

2.2.2 DBScan

The density of the clusters vary widely in the data set. Thus choosing an optimum e and minpoints to identify all the clusters correctly may not be possible. Keeping a low density threshold may combine dissimilar clusters while keeping a high threshold would categorise a lot of data points as noise. This should especially be noted for separating the high dense green region from the very low dense blue region. Separating the red and light pink clusters too would pose a problem as there doesn't seem to a very low density patch between the two which can used by DBScan to identifying them as different clusters. The cyan and pink clusters can be effectively be identified by using DBScan due to each of them being dense clusters with empty space separating them. The cyan and pink cluster are also well separated from the other clusters.

2.2.3 Hierarchical Clustering

Both the single link and complete link would end up putting the red & light pink in one cluster and cyan & pink in one cluster due to their close proximity to each other than to the other clusters. The difference comes in how the green and blue points are clustered. In case of the single link the all the green points would be put together but the blue points will be heavily segmented. On the other hand, the complete link will cause the blue points as well as the green points to split. For k=3, all the points of green and blue would be put in one cluster.So on an overall the clustering obtained when HC is done with single and complete link would be pretty bad due to highly varied nature of the clusters.

2.3 Path-based Dataset

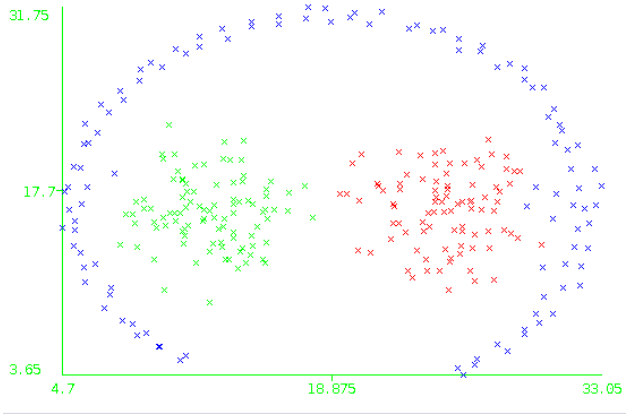


Figure 5: pathbased.arff

2.3.1 K-Means

K-Means would perform badly for this dataset. K-Means algo is once again inherently inefficient when it comes to identifying clusters like these. In this scenario K-Means would most probably end up identifying the green and red clusters correctly but the blue points would be split into 3 classes with points on the left combining with the green cluster, the points on the right combining with the red cluster and the points on the top forming their own cluster and thus giving us 3 clusters quite different from reality.

2.3.2 DBScan

In the given data set, the red and green clusters seem to have more or less the same density of points and are decently seperated. But however the blue cluster is more elongated and hence has lower density. Hence once again we face this conundrum of choosing e and minpoints rightly such that the density threshold is not too high that all the blues are classified as noise and neither too low that the blue becomes a part of the red and green clusters.Identifying such values of e and minpoints is not very apparent and hence have to done in a systematic manner In the given dataset it may not even be possible. However the green and red clusters can be identified effectively due their and high density and low density separation.

See section 5.1.1 for a more detailed analysis.

2.3.3 Hierarchical Clustering

Single Link: Due to the close proximity of points belonging to different clusters at either ends, using single link would cause most of the points to be pooled into one cluster resulting in very poor performance.

Complete Link: For this data set, compared to single link, complete link would perform much better. Complete link is likely to find most the red and green cluster rightly. But however the blue points would be segmented with some going to the red & green clusters and others forming their own clusters.

2.4 Spiral Dataset

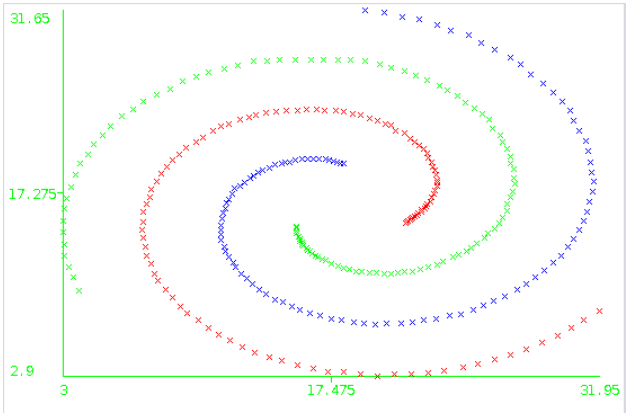


Figure 6: spiral.arff

2.4.1 K-Means

K-Means would perform badly for this dataset. K-Means algo is once again inherently inefficient when it comes to identifying clusters like these. In this scenario K-Means would most probably end up splitting the data points into 3 clusters radially which is definitely not the nature of the actual clusters.

2.4.2 DBScan

DBScan would work really well for this data set due to presence of high density clusters with zero density regions separating between them. See Table 5 for the ideal choice of values for e and minpoints. Each of the spirals would be identified as separate cluster which is indeed the actual nature of the data.

See section 5.2.1 for a more detailed analysis.

2.4.3 Hierarchical Clustering

Single Link: Single link hierarchical clustering is the most suited for this data set. We can clearly see this from the visualisation. Single link would pool in all all points connected through the spiral and hence the entire spiral in one cluster. Thus each of the spirals would be identified as a separate cluster which is indeed the actual nature of the data.

Complete Link: For this dataset, using a complete link would end up giving a clustering very similar to K-Means. It would result in clusters radially separated with separating boundaries being along the radius. This is definitely not what we observe from looking at the data.

2.4.4 Inference

DBScan with appropriate value for e and minpoints or Hierarchical Clustering with single link would be most suitable for this data set.

2.5 D31 Dataset

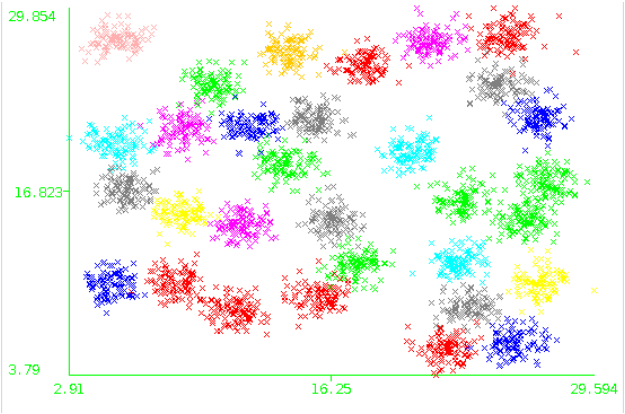


Figure 7: D31.arff

2.5.1 K-Means

K-means would not perform very well for this data set. The clusters are too closely placed and there is also overlap between a few clusters. Due to this, the performance K-Means heavily relies on the choice of the initial centroids chosen. With 31 clusters present, getting all the 31 centroids right is difficult and due to this, exact retrieval of the clusters may not be possible.

See Section 6.1 for a more detailed analysis.

2.5.2 DBScan

DBScan would performs poorly for this dataset. This is due to not-so-well separated set of clusters, with respect to the boundary points, that are present in the dataset. Density based clustering methods work best only when the clusters under consideration are highly dense themselves and are also separated by sufficiently less dense regions from nearby clusters. This is not true with the current data set and hence the performance wouldn't be optimal.

See Section 6.2 for a more detailed analysis.

2.5.3 Hierarchical Clustering

In the given data set the centroids of most clusters are sufficiently seperated from nearby clusters. The points of cluster are also quite dense aroud their respective centers. This setup is very suitable for HC with the Ward's link or complete link for instance. Thus HC with appropriate linkage would likely be able to successfully retrieve most of the clusters.

2.6 R15 Dataset

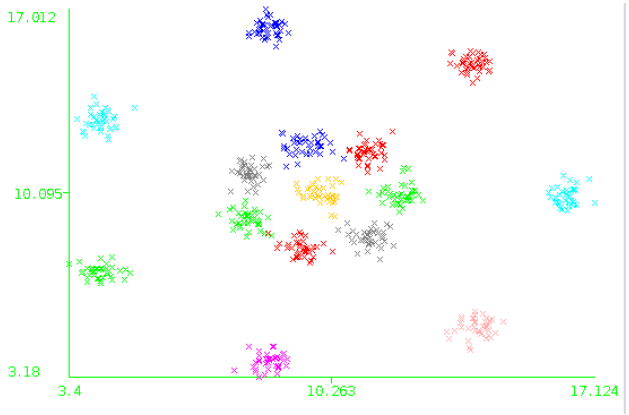


Figure 8: R15.arff

2.6.1 K-Means

The outer clusters are well separated and inner clusters are also decently dense around their centroids. This setup is quite ideal for K-means. But however, the performance largely relies on the initialisation of the centroids. If initial centroids are assigned quite close to the actual centroid most points would be clustered correctly. However if the initial centroids are placed somewhere in the empty space in the middle, some reshuffling would occur causing some outer clusters to split and some inner clusters to merge. Initialising using farthest first would work really well here.

2.6.2 DBScan

The outer clusters are dense and well separated from the other clusters. Hence the outer clusters can be accurately identified using DBScan. However the inner clusters are too close and there is some overlap among some inner cluster as well. Every cluster has some boundary points too close to another cluster. So DBScan would end up putting all the inner clusters into one cluster. So most likely we would end up with one huge cluster in the middle with 7 distinct outer clusters.

2.6.3 Hierarchical Clustering

In the given data set the centroids of most clusters are sufficiently seperated from nearby clusters. The points of cluster are also quite dense aroud their respective centers. This setup is very suitable for HC with the Ward's link or complete link for instance. Thus HC with appropriate linkage would likely be able to successfully retrieve most of the clusters.

2.7 Jain Dataset

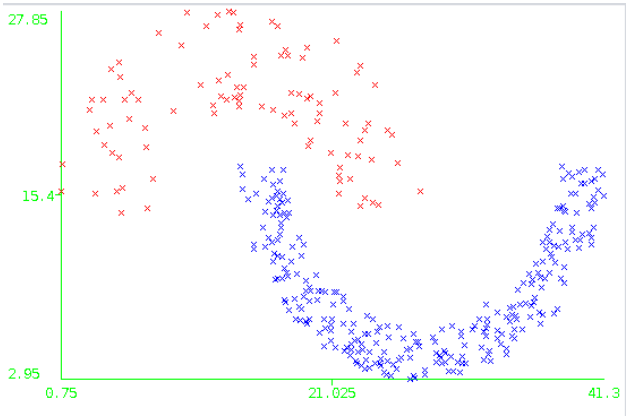


Figure 9: jain.arff

2.7.1 K-Means

For this dataset, K-Means would rightly identify the cluster at either ends. But in the region of close proxity between the red and blue clusters it is possible tha some points of one cluster would be put in as part of the other cluster depending on the initialisation of the centroids. But it will be small fraction and hence the overall purity would still be note worthy.

2.7.2 DBScan

In this dataset, there is a clear difference in density of the two clusters. The red cluster is less dense than the blue cluster. Hence once again we face this conundrum of choosing e and minpoints rightly such that the density threshold is not too high that all the reds are classified as noise and neither too low that the blue and red are merged to form one cluster. If we decrease the density threshold to put all the red points in one cluster, this would in turn cause a merging of the red and blue into one big undesirable cluster. Identifying such values of e and minpoints is not very apparent and hence have to done in a systematic manner In the given dataset it may not even be possible as the density of points in the region separating the two clusters and the density in parts of the red cluster are about the same.

See Section 4 for a more detailed analysis.

2.7.3 Hierarchical Clustering

Single Link: For the given data, single link would end up putting most of the points in one cluster due to the close proximity of the red and blue points in the left end of the blue cluster.Thus single link would not be an ideal choice for this data set.

Complete Link: For this dataset, complete link would perform much better than single link and the clusters would be rightly identified at either ends. But in the region of close proxity between the red and blue clusters some points of one cluster would be put in as part of the other cluster. But it will be small fraction and hence the overall purity would still be note worthy.

2.7.4 Inference

Though not the best, K-Means and Hierarchical Clustering with complete link would perform best compared to others.

2.8 Flame Dataset

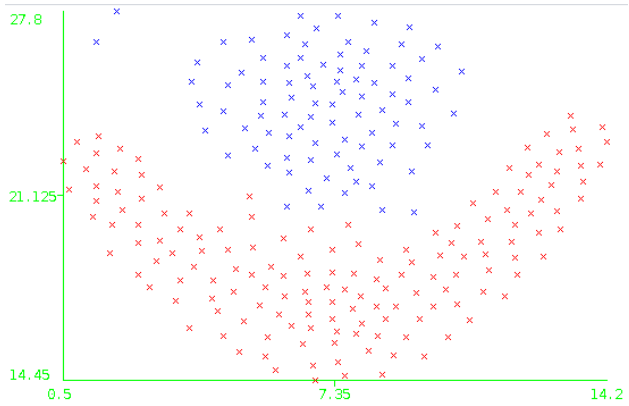


Figure 10: flame.arff

2.8.1 K-Means

K-Means would not perform very well in this data set. Due to large extent of red and either ends of the red cluster being closer to blue cluster, the points at the either end would be pushed into the blue cluster instead. The

fraction of the points which will be moved to other cluster will depend on the initialisation of the centroids.

2.8.2 DBScan

As we can see from the visualisation of the dataset, though the density of the two clusters are more or less the same, the density of points in the region separating the two clusters is about the same as well. This poses a serious problem when we want to identify the individual clusters using a density based clustering approach. Whatever threshold we set, we either end up classifying all the points as noise (when the threshold is greater than the existing density) or putting all the points in one cluster (when the threshold is smaller than the existing density).

See section 5.3.1 for a more detailed analysis.

2.8.3 Hierarchical Clustering

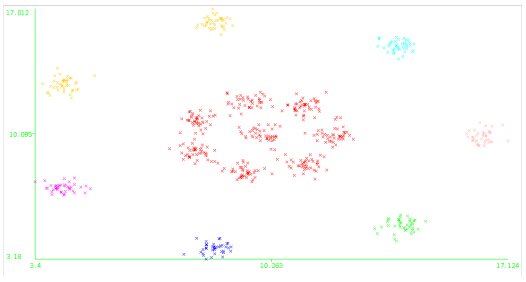
Single Link: As we can see form the visualisation, the density in the region connecting the red and blue clusters is not very different from the density of the individual clusters. Thus using single link would result in all the points being put in one cluster which is definitely not the nature of the data.

Complete Link: Complete link too would face a problem similar to the one faced by single link. But due to the couple of blue points in the top left corner, some of the points on the left end of the red cluster may end forming a new cluster with them.All the other points will form the other cluster. This again is not same as the natural clusters shown in the visualisation.

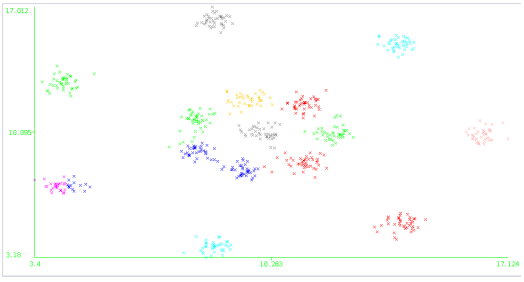
2.8.4 Inference

Though not the best, K-Means would perform best for this data set when compared to others.

3 K-Means on R15



(a) Clusters for K=8



(b) Clusters for K=15

K	1	2	3	4	5	6	7	8	9	10
CP	0.067	0.133	0.200	0.267	0.333	0.400	0.467	0.533	0.600	0.667
K	11	12	13	14	15	16	17	18	19	20
CP	0.733	0.800	0.862	0.862	0.928	0.928	0.997	0.997	0.997	0.997

Table 1: K & CP (Cluster Purity)

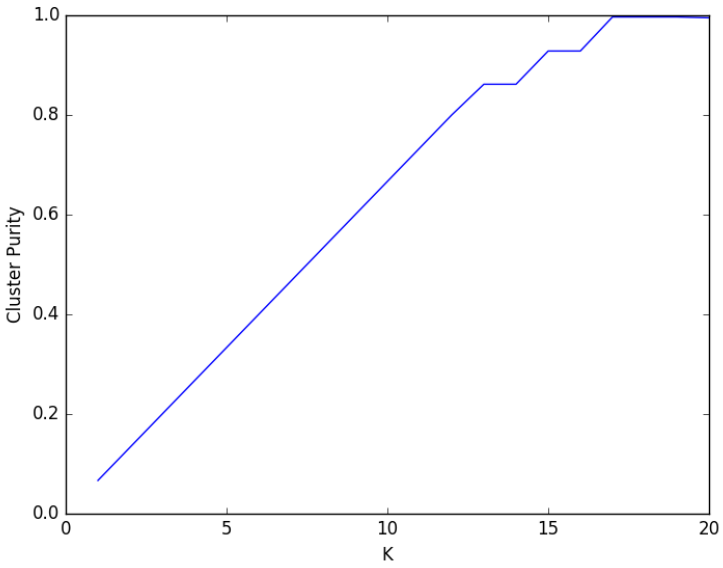


Figure 11: Cluster Purity vs K

The cluster purity for various values of K ranging from 1 to 20 can be seen in the above table.As we can see from the table, the cluster purity for K=8 is 0.533.The cluster purity increases with increase in K but the rate at which the cluster purity increases decreases.The cluster purity stagnates after K=15 which gives an indication to the actual existence of 15 clusters in the given dataset.

Please refer *p3.py* under the Code folder for details of the code written.

4 DBScan on Jain Dataset

e X m	16	17	18	19	20	21	22	23	24
0.070	0.740	0.737	0.737	0.737	0.737	0.727	0.727	0.727	0.654
0.080	0.740	0.740	0.740	0.740	0.740	0.740	0.740	0.740	0.740
0.090	0.812	0.812	0.812	0.740	0.740	0.740	0.740	0.740	0.740
0.100	0.740	0.858	0.839	0.815	0.740	0.740	0.740	0.740	0.740
0.110	0.740	0.740	0.740	0.740	0.836	0.836	0.818	0.818	0.804

Table 2: Cluster Purity for different e (epsilon) and m (minpoints) values

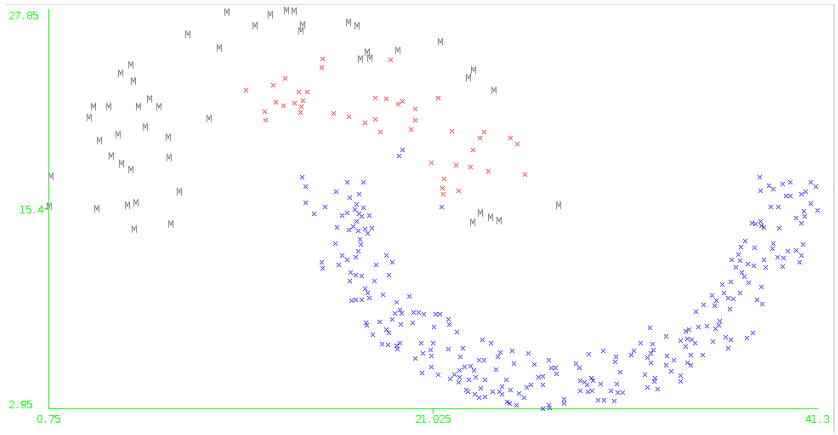


Figure 12: Best DBScan (e: 0.1, minpoints: 17, Cluster Purity: 0.858)

In this dataset, there is clear difference in density of the two clusters. The red cluster is less dense than the blue cluster. Hence once again we face this conundrum of choosing e and minpoints rightly such that the density threshold is not too high that all the reds are classified as noise and neither too low that the blue and red are merged to form one cluster. If we decrease the density threshold to put all the red points in one cluster, this would in turn cause a merging of the red and blue into one big undesirable cluster. Identifying such values of e and minpoints is not very apparent and hence have to done in a systematic manner In the given dataset it may not even be possible as the density of points in the region separating the two clusters and the density in parts of the red cluster are about the same.

In the epsilon vs minpoints grid shown above, when we consider each row (ie. fixed epsilon), the purity increases as m increases, reaches a maxima and then either remains or goes down from there. We observe a similar trend in the column by column analysis as well. So the best values of e=0.1 & minpoints=17, with cluster purity of 0.858, can be considered as a local maxima of sorts. This behaviour is observed as we have considered a spectrum of density thresholds, starting from a very low density threshold to a high density threshold, which contains the close-to-optimum threshold for the given data as well. The performance of DBScan for this dataset is limited due to the difference in densities of the clusters.

Please refer *p4.py* under the Code folder for details of the code written.

5 DBScan vs Hierarchical Clustering

Please refer *p5.py* under the Code folder for details of the code written.

5.1 Path-Based Dataset

5.1.1 DBScan

e X m	1	2	4	8	16	32
0.100	0.367	0.367	0.367	0.400	0.623	0.537
0.200	0.367	0.367	0.367	0.367	0.367	0.323
0.400	0.367	0.367	0.367	0.367	0.367	0.367
0.800	0.367	0.367	0.367	0.367	0.367	0.367
1.600	0.367	0.367	0.367	0.367	0.367	0.367
3.200	0.367	0.367	0.367	0.367	0.367	0.367

Table 3: Cluster Purity for different e (epsilon) and m (min_points)

In the given data set, the red and green clusters seem to have more or less the same density of points and are decently seperated. But however the blue cluster is more elongated and hence has lower density. Hence once again we face this conundrum of choosing e and minpoints rightly such that the density threshold is not too high that all the blues are classified as noise and neither

too low that the blue becomes a part of the red and green clusters. In the table above the CP values of 0.367 correspond to the scenario where all the points are merged into one cluster and the CP values of 0.623 corresponds to the scenario where the most of the red and green clusters are identified correctly.

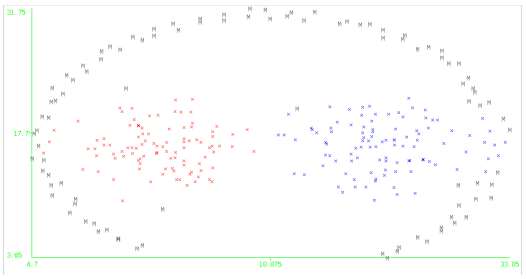
5.1.2 Hierarchical Clustering

In case of Hierarchical Clustering we would tend to get better CP with greater k as the scenario with all points being their own clusters, though not the best, has a CP of 1.0. So keeping that in mind, though the best clustering we have got with respect to the CP is k=6 and link=Mean, we must actually look at it's performance in k=3. For k=3, the Ward's linkage gives the best CP of 0.753 (see figures below).

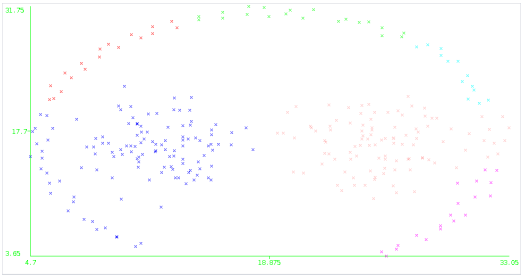
k X l	SI	CO	AV	ME	CE	WA	AD	NE
1	0.367	0.367	0.367	0.367	0.367	0.367	0.367	0.367
2	0.370	0.433	0.423	0.633	0.633	0.633	0.627	0.367
3	0.373	0.707	0.730	0.700	0.733	0.753	0.640	0.367
4	0.377	0.707	0.767	0.760	0.770	0.753	0.653	0.367
5	0.383	0.707	0.767	0.823	0.770	0.800	0.673	0.367
6	0.520	0.707	0.810	0.867	0.813	0.810	0.687	0.367

Table 4: Cluster Purity for different k and l (links)

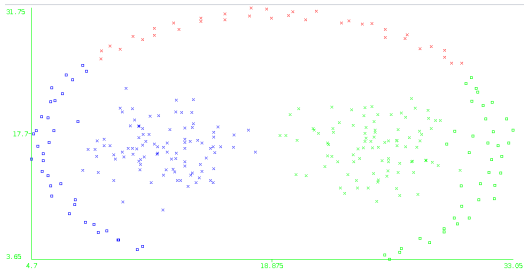
5.1.3 Inference



(a) Best DBScan (e: 0.1, m: 16, CP: 0.623)



(b) Best HC (Link: Mean, K: 6, CP: 0.867)



(c) HC (Link: Ward, K: 3, CP: 0.753)

Hierarchical Clustering with appropriate parameter values works better than DBScan for this dataset. But both are not the most suitable clustering methods for this dataset. The performance of DBScan is limited due the not-so-ideal cluster densities while HC doesnot perform optimally due to the closeness of the data points of the different clusters at either ends.

5.2 Spiral Dataset

5.2.1 DBScan

e X m	1	2	4	8	16	32
0.100	1.000	1.000	1.000	0.724	0.301	0.000
0.200	0.340	0.340	0.340	0.340	0.340	0.340
0.400	0.340	0.340	0.340	0.340	0.340	0.340
0.800	0.340	0.340	0.340	0.340	0.340	0.340
1.600	0.340	0.340	0.340	0.340	0.340	0.340
3.200	0.340	0.340	0.340	0.340	0.340	0.340

Table 5: Cluster Purity for different e (epsilon) and m (min_points)

DBScan, with appropriate values of e and minpoints, works really well for this data set due to presence of high density clusters with zero density regions separating between them. As we can see from the table, e=0.1 and m=1 or 2 or 4 gives us a 100% cluster purity. Each of the spirals are clearly identified as separate clusters.

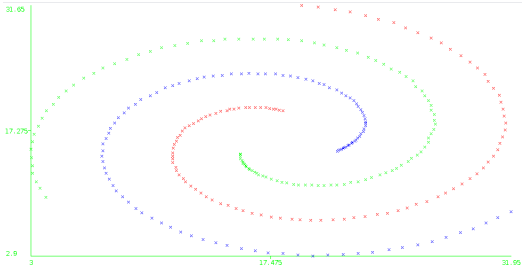
5.2.2 Hierarchical Clustering

k X l	SI	CO	AV	ME	CE	WA	AD	NE
1	0.340	0.340	0.340	0.340	0.340	0.340	0.340	0.340
2	0.663	0.349	0.365	0.369	0.388	0.372	0.346	0.340
3	1.000	0.381	0.365	0.391	0.404	0.423	0.356	0.340
4	1.000	0.397	0.391	0.391	0.404	0.433	0.365	0.340
5	1.000	0.423	0.397	0.391	0.404	0.452	0.375	0.340
6	1.000	0.455	0.404	0.426	0.413	0.490	0.381	0.340

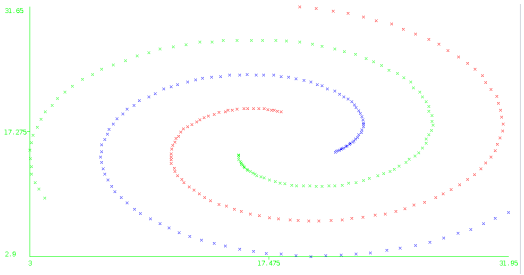
Table 6: Cluster Purity for different k and l (links)

Single link hierarchical clustering, as we can see, is the most suited for this data set. We can also clearly see this from the visualisation. Single link would pool in all all points connected through the spiral and hence the entire spiral in one cluster. Thus each of the spirals would be identified as a separate cluster which is indeed the actual nature of the data.

5.2.3 Inference



(d) Best DBScan (e: 0.1, m: 4, CP: 1.0)



(e) Best HC (Link: Single, K: 3, CP: 1.0)

As shown above, for appropriate parameter values both DBScan and Hierarchical clustering are able to successfully identify the clusters.

5.3 Flame Dataset

5.3.1 DBScan

e X m	1	2	4	8	16	32
0.100	0.646	0.646	0.637	0.637	0.396	0.000
0.200	0.637	0.637	0.637	0.637	0.637	0.633
0.400	0.637	0.637	0.637	0.637	0.637	0.637
0.800	0.637	0.637	0.637	0.637	0.637	0.637
1.600	0.637	0.637	0.637	0.637	0.637	0.637
3.200	0.637	0.637	0.637	0.637	0.637	0.637

Table 7: Cluster Purity for different e (epsilon) and m (min_points)

As we can see from the visualisation of the dataset, though the density of the two clusters are more or less the same, the density of points in the region separating the two clusters is about the same as well. This poses a serious problem when we want to identify the individual clusters using a density based clustering approach. Whatever threshold we set, we either end up classifying all the points as noise (when the threshold is greater than the existing density) or putting all the points in one cluster (when the threshold is smaller than the existing density). The constant CP of 0.67 that we observe in the table above corresponds to this scenario where all the points are put in one cluster.

5.3.2 Hierarchical Clustering

k X l	SI	CO	AV	ME	CE	WA	AD	NE
1	0.637	0.637	0.637	0.637	0.637	0.637	0.637	0.637
2	0.646	0.637	0.833	0.921	0.646	1.000	0.642	0.637
3	0.646	0.825	1.000	0.921	0.871	1.000	0.646	0.637
4	0.650	0.992	1.000	0.988	0.992	1.000	0.646	0.637
5	0.650	1.000	1.000	0.988	0.992	1.000	0.646	0.637

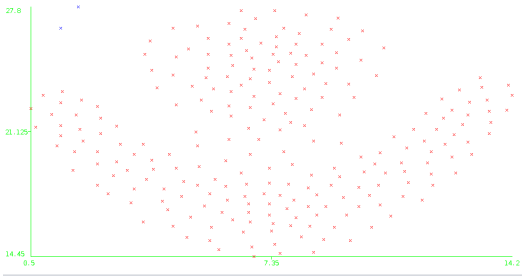
Table 8: Cluster Purity for different k and l (links)

As we can see from the table and the figure (f) below, Hierarchical Clustering with Ward linkage and k=2 successfully identifies both the clusters.

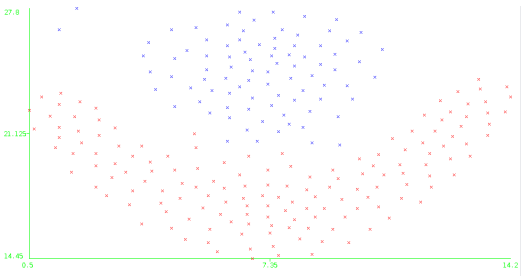
With Ward's linkage method, the distance between two clusters is the sum of squared deviations from points to centroids. The goal of Ward's linkage method is to minimize the within-cluster sum of squares.

When you consider the visualisation of this dataset, we can clearly see 2 clusters with their centroids well seperated. The points of each cluster is also quite dense around their respective centroids. These kind of clusters are exactly what Ward's linkage identifies. Thus Ward's linkage works perfectly for the given dataset.

5.3.3 Inference



(f) Best DBScan (e: 0.1, m: 2, CP: 0.646)



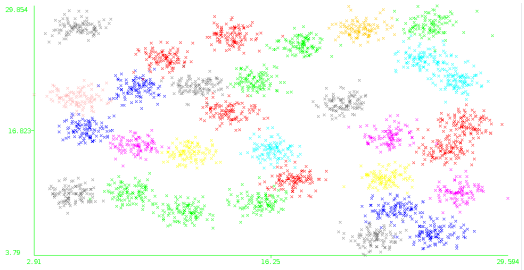
(g) Best HC (Link: Ward, K: 2, CP: 1.0)

Hierarchical Clustering with appropriate parameter values works much better than DBScan for this dataset due to the nature of the clusters.

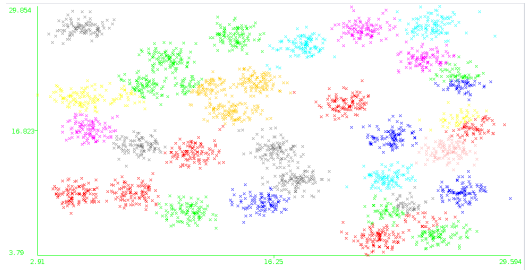
6 D31 Dataset Analysis

Please refer *p6.py* under the Code folder for details of the code written.

6.1 K-Means



(h) Actual Clusters



(i) Clusters obtained with K=32

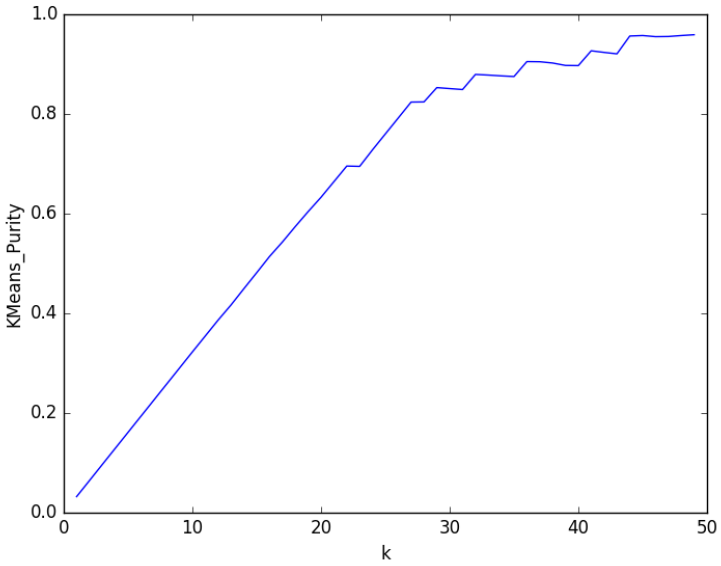


Figure 13: Cluster Purity vs K

As we can see from the figure above as well as the graph of Cluster Purity vs K, we are not able to recover all the 31 clusters by setting k=32 or any other value till 50 for that matter. However the cluster purity does increase for higher values of K and plateaus after a value of 40.

6.2 DBScan

e X m	120	130	140	150	160	170	180	190	200
0.080	0.225	0.258	0.187	0.096	0.000	0.000	0.000	0.000	0.000
0.100	0.065	0.129	0.161	0.216	0.222	0.246	0.159	0.096	0.095
0.200	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.032	0.032

Table 9: Cluster Purity for different e (epsilon) and m (min_points)

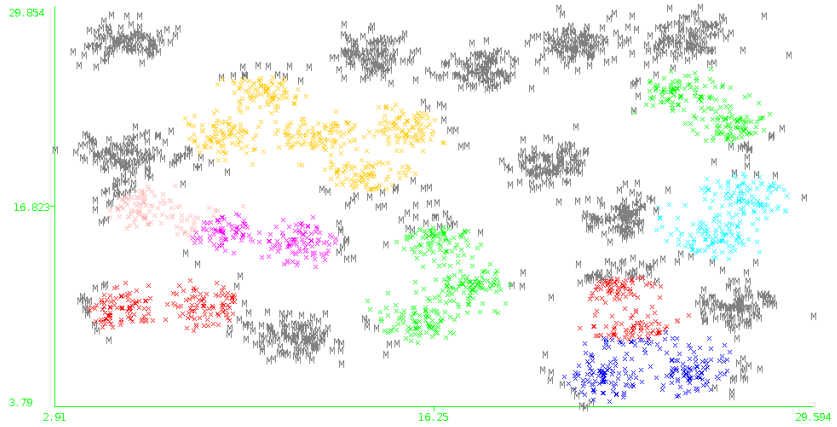


Figure 14: Best DBScan (e: 0.08, minpoints: 130, Cluster Purity: 0.258)

As we can see from the table as well the figure above, DBScan performs poorly for this dataset. This is due to not-so-well separated set of clusters, with respect to the boundary points, that are present in the original dataset. Density based clustering methods work best only when the clusters under consideration are highly dense themselves and are also separated by sufficiently less dense regions from nearby clusters. This is not true with the current data set and hence the poor performance.

6.3 Hierarchical Analysis

l X k	28	29	30	31	32	33	34	35
WARD	0.882	0.910	0.939	0.963	0.963	0.963	0.963	0.963

Table 10: Cluster Purity for WARD link for different k

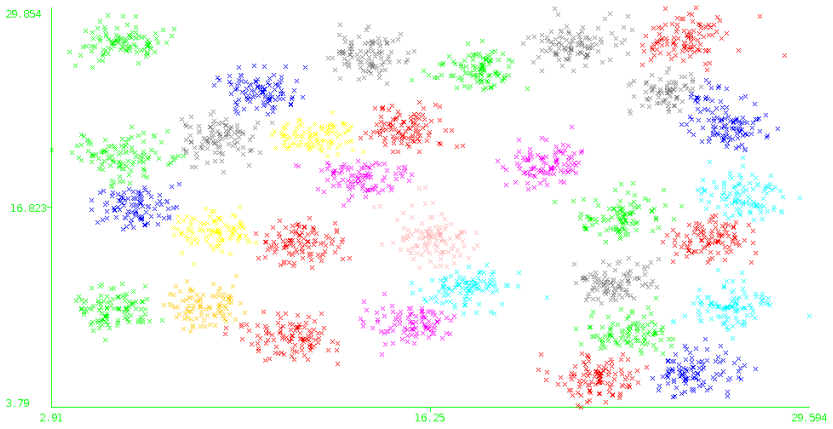


Figure 15: Best HC (Link: WARD, K: 31, Cluster Purity: 0.963)

With Ward's linkage method, the distance between two clusters is the sum of squared deviations from points to centroids. The goal of Ward's linkage method is to minimize the within-cluster sum of squares. In the given data set the centroids of most clusters are sufficiently seperated from nearby clusters. The points of cluster are also quite dense aroud their respective centers. This setup is very suitable for the Ward's linkage. And as we can see, Hierarchical Clustering with Ward's linkage does perform really well giving us a cluster purity of 0.963 for k=31.