CIS6930 Fall 2017: Project II-Clustering

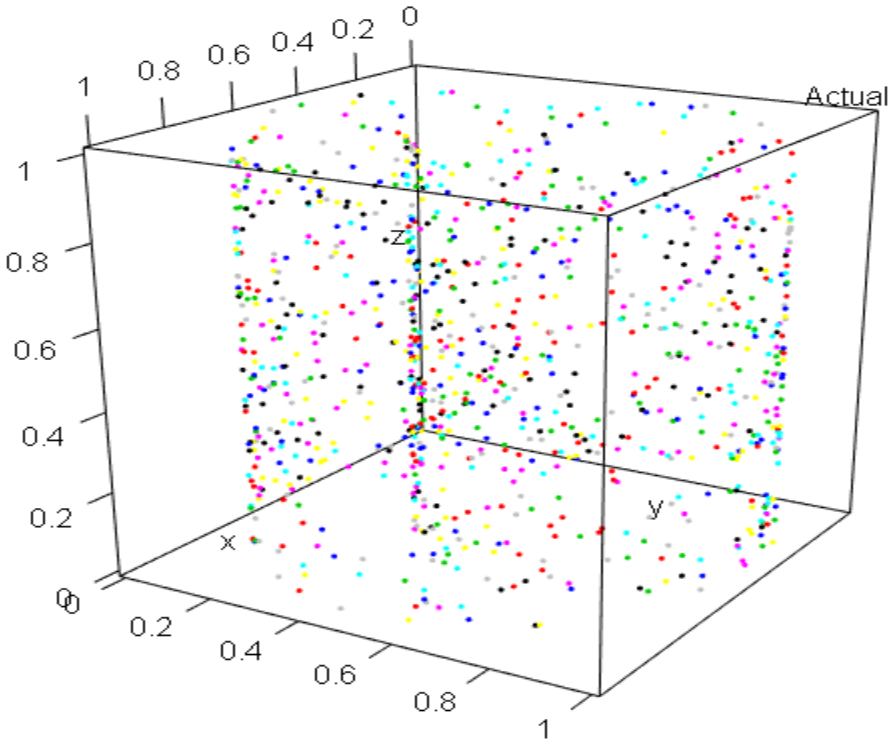
Vagisha Tyagi, UFID:- 0428-9808

Dataset-1

The aim is to apply 4 types of clustering techniques namely _Hierarchical Clustering, K-means Clustering, Density-based Clustering and Graph-based Clustering on Dataset1 which contains 1000 data points of 8 clusters in 3D space.

Initial data observation:

To analyse the original dataset , I used plot3D method of package rgl in R which gave following result:



As we can see that data doesn't prominently form groups and are not clearly separated, hence we can expect some low accuracies for the clustering techniques that will be applied.

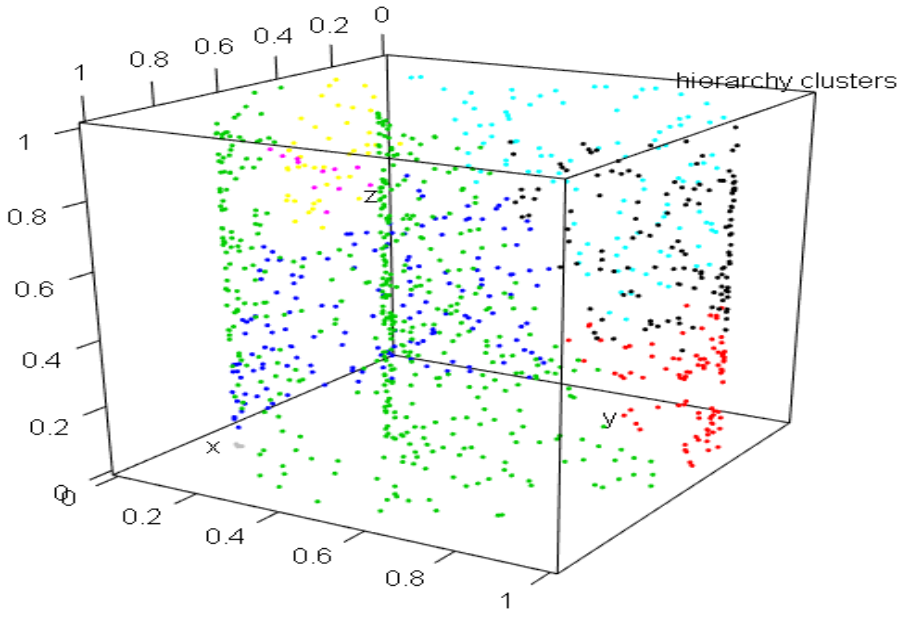
Data Preparation

It is common to normalize all variables before clustering so that putting all variables into the same range, the variables can be weighed equally-although it is not compulsory. Any missing data is checked and removed.

Clustering Techniques:

Hierarchical Clustering:

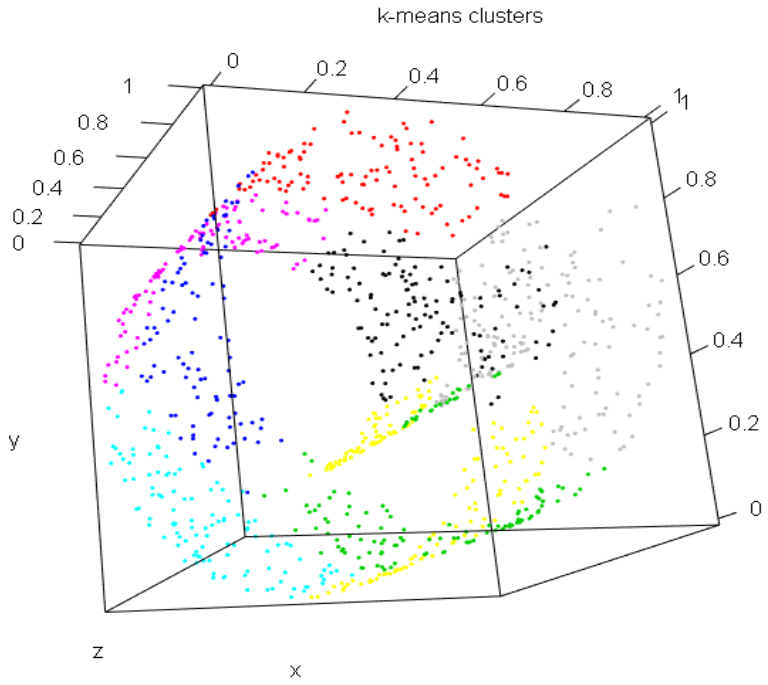
- Hierarchical clustering works bottom up where it puts each data point in its own cluster, identifies the closest two clusters and combine them into one cluster repeating all the way till all the data points are in a single cluster.
- The result of hierarchical clustering is a tree-based representation of the objects, which is also known as dendrogram. Observations can be subdivided into groups by cutting the dendrogram at a desired similarity level.
- There can be many ways to determine closeness of 2 clusters, here centroid linkage clustering is chosen where it finds the centroid of each cluster and calculates the distance between centroids of two clusters.
- To interpret results, 3D graph shows the hierarchical clustering results using plot3D method of package rgl in R. As we can see below, to some extent, grouping of data has been done into 8 clusters.



- Applying this technique on the data set and comparing the clustered result labels with the ground truth labels, an **accuracy of 0.129** is obtained.

K-means Clustering:

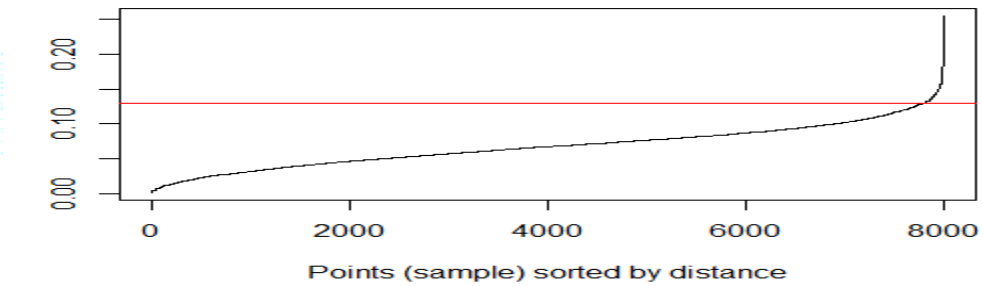
- K-means clustering requires us to specify the number of clusters to extract and is computationally faster.
- Since the initial cluster assignments are random, I set the seed to ensure reproducibility.
- This clustering technique involves with randomly selecting the the initial centroids, you need to repeat the clustering experiments for multiple times and compute the average value of the accuracies.
- To interpret results, 3D graph shows the k-means clustering results using plot3D method of package rgl in R. As we can see below, to some extent, grouping of data has been done into 8 clusters.



- Choosing nstart=25, meaning 25 starting points as initial centres are chosen, iter.max = 100, algorithm="MacQueen" parameters in k-means method, **accuracy of 0.123** was observed.

Density Based Clustering:

- Performed density based clustering with DBSCAN method which estimates the density around each data point by counting the number of points in a user-specified eps-neighborhood and applies a used-specified minPts thresholds to identify core, border and noise points.
- In a second step, core points are joined into a cluster if they are density-reachable. Finally, border points are assigned to clusters. The algorithm only needs parameters eps and minPts.
- The knee in kNNdistplot can be used to find suitable values for eps. Knee in knndistplot was observed at 0.13 and hence eps was chosen as 0.13.



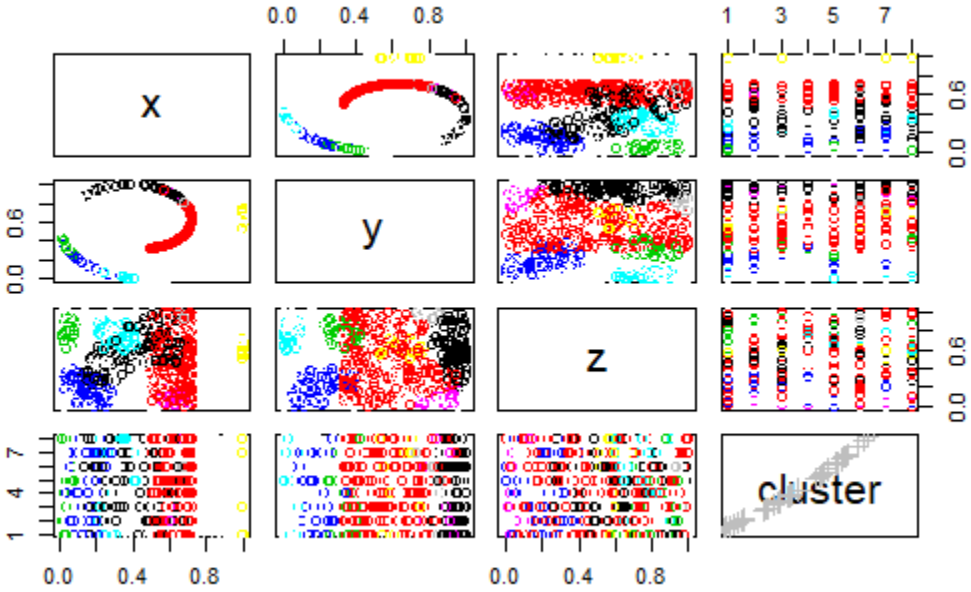
- DBSCAN divided the data into 8 clusters showing as below:

DBSCAN clustering for 1000 objects.
Parameters: eps = 0.13, minPts = 20
The clustering contains 8 cluster(s) and 445 noise points.

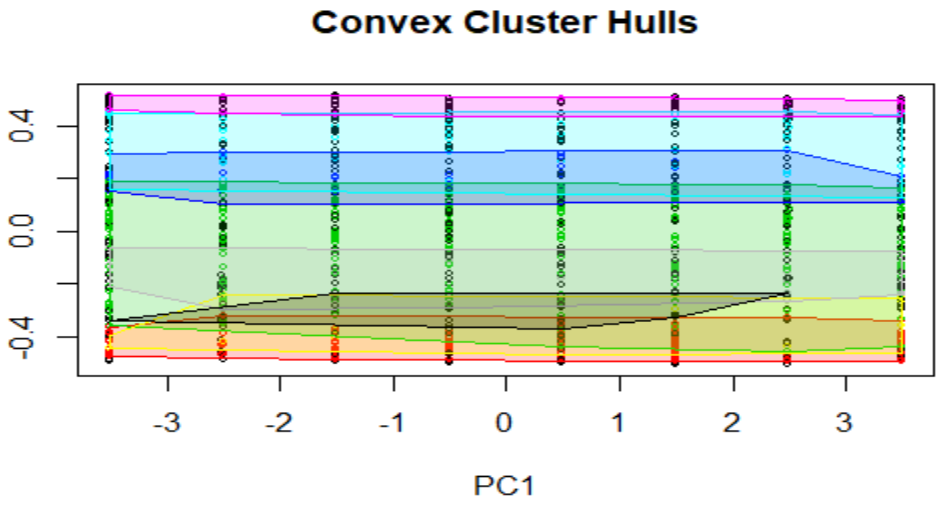
```
  0   1   2   3   4   5   6   7   8
445 117 265  33  59  34  18  20   9
```

Available fields: cluster, eps, minPts

- To interpret results, plot clusters and add noise (cluster 0) as crosses shown as below. Could not use 3D graph for this due to presence of noise as number 0 which cannot be taken as colors



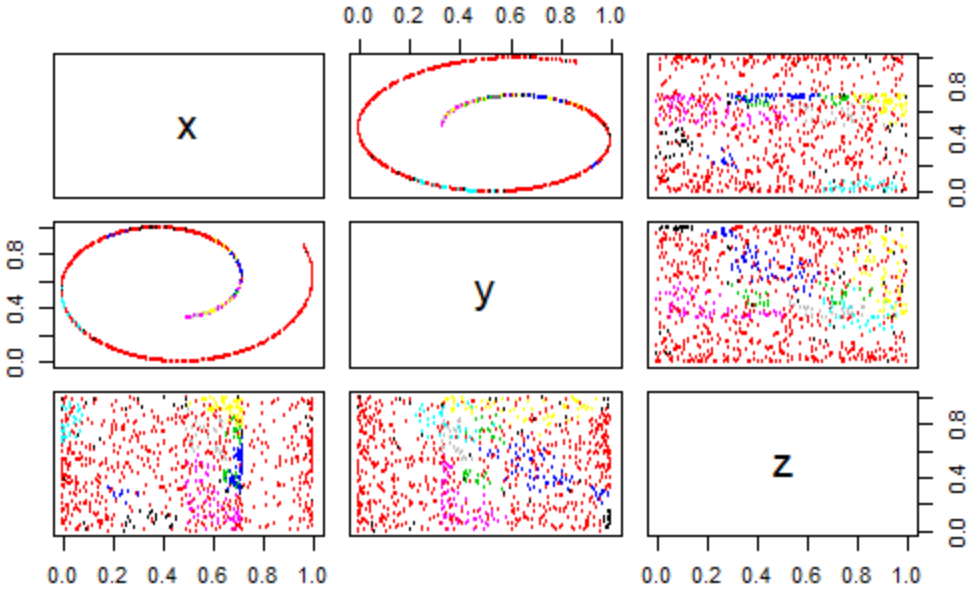
- And convex cluster hulls are shown as below:



- Applying this technique on the data set and comparing the clustered result labels with the ground truth labels, an **accuracy of 0.114** is obtained.

Graph Based Clustering:

- Here I used shared nearest neighbor clustering which constructs a **shared nearest neighbor graph for a given k.** The edge weights are the number of shared k nearest neighbors
- Then, we find the number of points which have a similarity of eps or greater and also find the core points, i.e., all points that have an SNN density greater than MinPts.
- Form clusters from the core points and assign border points
- eps is used on a similarity (the number of shared neighbors)
- To interpret results, Out of k = 8 NN 1 (eps) have to be shared to create a link in the sNN graph. A point needs a least 8 (minPts) links in the sNN graph to be a core point.
- Noise points have cluster id 0 and are shown in black. Could not use 3D graph for this due to presence of noise as number 0 which cannot be taken as colors. Results are shown below:



- Applying this technique on the data set and comparing the clustered result labels with the ground truth labels, an **accuracy of 0.119** is obtained.

Conclusion for Dataset-1 results :-

Comparison of above clustering methods:

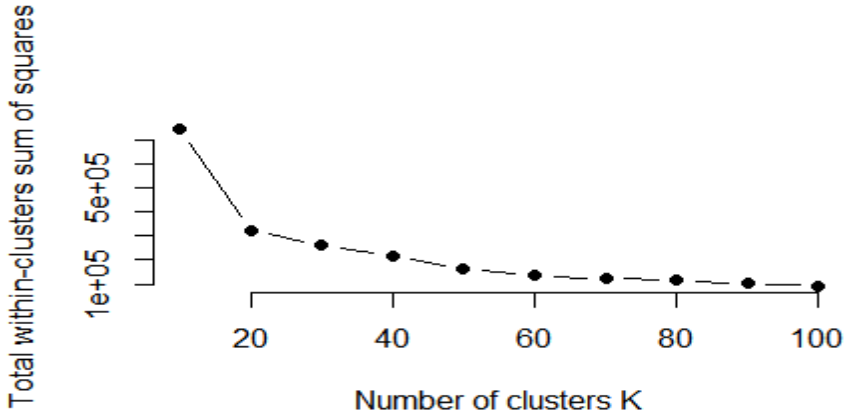
- As we observed, different clustering methods produced different accuracies and in our case, hierarchical clustering gave the best accuracy as 0.129.
- However, there is no general consensus to point which one produces better clustering as we already saw the original data was itself intermixed and showed no groupings.
- Practicing above clustering methods, we can confirm that hierarchical clustering can actually be slow as it has to make several merge /split decisions.

- Other thing that can be observed about k-means is that bad initialization can lead to poor convergence speed and bad overall clustering too. It's sensitive to outliers too.
- Compared to centroid-based clustering like K-Means, density-based clustering works by identifying "dense" clusters of points, allowing it to learn clusters of arbitrary shape
- DBSCAN can find arbitrarily shaped clusters and it can also identify outliers in the data as we observed from our plots.
- DBSCAN as in density based clustering cannot cluster data sets well with large differences in densities, since the minPts-\varepsilon combination cannot then be chosen appropriately for all clusters.
- Graph based clustering can provide more detailed information about inner structure of dataset in form of cliques, clusters, centrality and outliers. The common characteristic of graph-based clustering methods developed in recent years is that they build a graph on the set of data and then use.

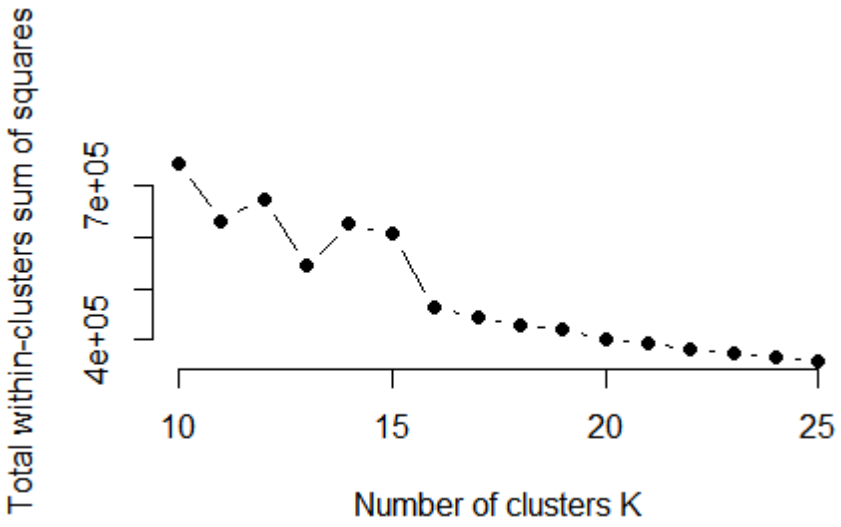
Dataset-2

Dataset 2 contains more than 1 million data points with the coordinates in 4 dimensions. For such large dataset, the clustering method that we want to choose should be fast enough.
Let's analyse the available methods and choose the one for our clustering analysis of dataset 2.
The hierarchy and SNN methods are not feasible because both method need to initialize a matrix based on data so in this case the matrix will be too large (3700GB) for the memory. Comparing kmeans with dbscan, the kmeans runs significantly faster and the parameters can be easily obtained.
If you compare the time complexities of K-Means with other methods: K-Means is O(tkn), where n is the number of objects, k is the number of clusters, and t is how many iterations it takes to converge. K-Medoids is O(k(n−k)^2) for each iteration. Agglomerative hierarchical clustering isO(n^3) (more efficient agglomerative clustering techniques are O(n^2) while exhaustive divisive hierarchical clustering is O(2^n) so we can see why these don't scale well for large datasets.
Therefore, the kmeans is used as clustering method for the task 2.

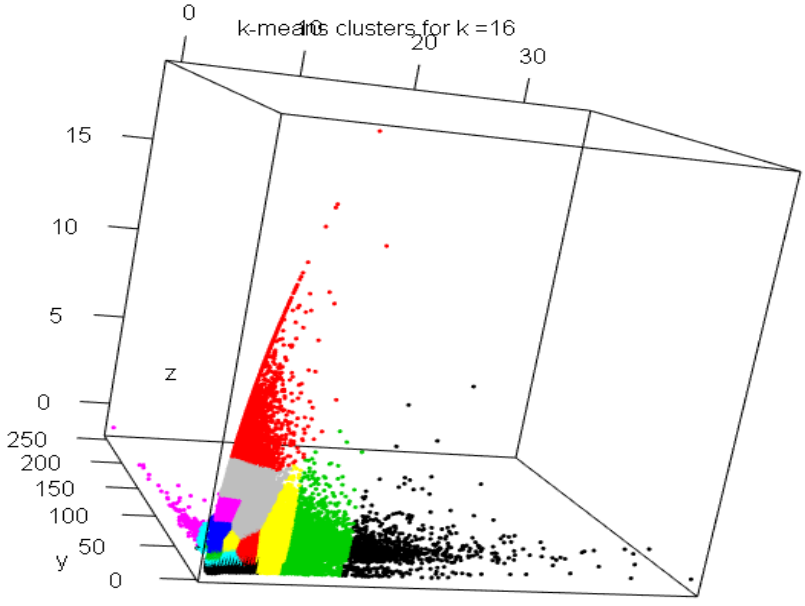
To find out the number of clusters that should be best to seek, the elbow method is used.
The first elbow screen used the input as (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) For each k, calculate the total within-cluster sum of square (wss) and plot as shown



The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters. This illustrates the group numbers should be between 10 and 25.
Then a second elbow experiment ran with input from 10 to 25. From the plot, a conclusion of cluster group number can be determined as two possible number: 15 or 16.



3D plot for k-means on dataset 2 is obtained as:



Running K means method with k=16 on data set, we obtain an evaluation parameters table as

```
Evaluation parameters for k=16: "
    totss tot.withinss betweenss iter
 4024728       590563   3434165  129
```

The totss is the total sum of squares and betweenss is the between groups deviance. The good clustering fit usurally have a bss/tss ratio close to one means most data points are separated into different groups. In the case of k=16, the averaged ratio of bss and tss is 82% which indicate a good fit. And when k=15, the averaged ratio of bss and tss is 81.6%.

RMSD and similarity matrix can only be calculated once we have actual results present in dataset so that actual and predictions can be compared.

Conclusion for Dataset-2 results:

- When we have a large dataset, it is always better to use k-means as it is computationally fast.
- Since, the k-means algorithm is parameterized by the value *k*, which is the number of clusters that you want to create, we need to determine the optimal number of clusters before we begin.
- The optimal number of clusters can be found by direct methods- elbow or silhouette OR Statistical Testing method such as gap statistic method.
- In our case, we chose elbow method to predict no. of clusters as it is simple to use.
- Finally, the k-means method could be applied to the large dataset generating above described results.

References:

[1]https://www.researchgate.net/post/K_modes_clustering_how_to_choose_the_number_of_clusters

[2]https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Density-Based_Clustering#Step_1:_Pre-Processing

[3]https://www.rdocumentation.org/packages/rgl/versions/0.97.0/topics/plot3d