Project 2 – Technical Review Report

Clustering Methods

CZ4032 Data Analytics & Mining

Name	Matric No	Contribution
Dandapath Soham	U1822646A	25%
Gupta Jay	U1822549K	25%
Kanodia Ritwik	U1822238H	25%
Mundhra Divyesh	U1822168E	25%

School of Computer Science & Engineering

Nanyang Technological University, Singapore

Abstract

Clustering plays a critical role in data analytics and mining with the emergence of big data. Identifying the underlying structures and patterns to form groups of data in an unsupervised setting is important to derive useful insights. In this paper, we present a comprehensive review of different clustering methods on multiple datasets to evaluate their performance, strengths, and caveats in different scenarios. The objective is to study what type methods are useful in varying situations with the help of thorough discussion and analysis.

Introduction

For a technical review of clustering methods, we have analysed four types of unsupervised clustering methods:

1. K-Means/K-Means++ Clustering
2. Agglomerative Hierarchical Clustering
3. DBSCAN Clustering
4. Mean Shift Clustering

For analysis, two datasets are used 'Iris' and 'Spiral'. The Iris dataset is taken from the UCI Machine Learning repository and the Spiral Dataset is synthetically generated.

The rationale behind the two chosen datasets is that the Iris dataset is based on real-life properties of three types of species of flowers. Therefore, the three species are naturally separable from each other into different clusters. On the other hand, the Spiral dataset is synthetically generated and its spiral nature makes the data extremely hard for clustering algorithms to separate into different clusters. Spiral is deliberately chosen to illustrate the weaknesses and caveats of some of the clustering methods.

Dataset	Classes	Samples
Iris	3	150
Spiral	2	194

Table 1: Dataset Description

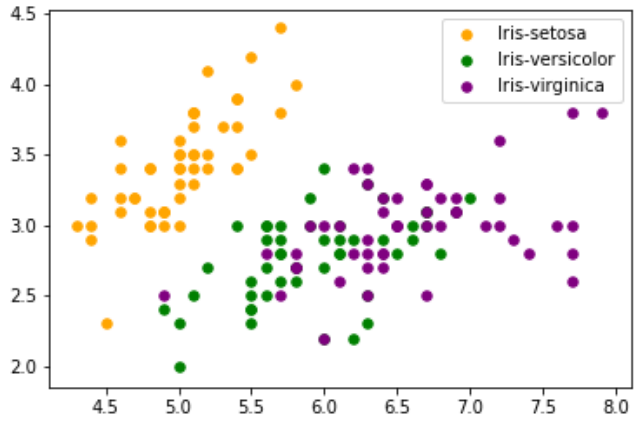

Figure 1: Ground Truth for the Iris Dataset

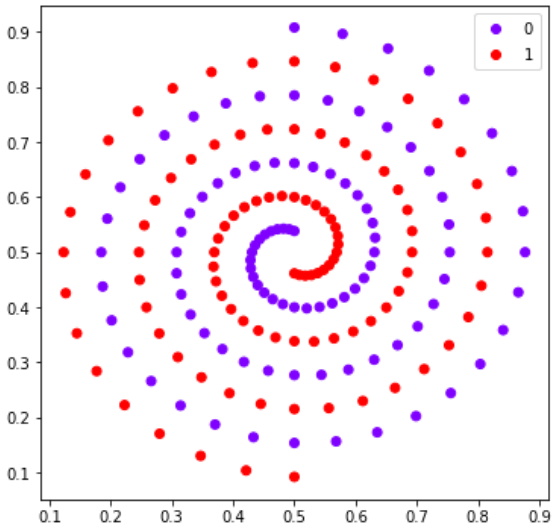

Figure 2: Ground Truth for the Spiral Dataset

It should be taken note that all the clustering algorithms used in this report are unsupervised, therefore, they do not require labels for training. However, the ground truth variable is used to calculate some additional metrics such as accuracy for the purpose of cluster comparisons.

For the scores we have used the following metrics :

1. **Accuracy** – Number of points that have been correctly labelled based on the ground truth.

2. **Silhouette Score** – It is calculated by using the mean intra-cluster distance, a and the mean nearest-cluster distance, b.

$$\text{Silhouette score} = \frac{(b-a)}{max(a,b)}$$

   The best value is 1 and the worst value – 1. A value of 0 indicates that there are overlapping clusters.

3. **Davies Bouldin Score** – The score is defined by measuring the similarity of average similarity of each of the cluster with the other most similar cluster. The minimum score is 0 and the lower the score, the better the result.

   For cluster i,

$$S_i = \left(\frac{1}{T_i} \sum_{j=1}^{T_i} \left| |X_j - A_i| \right|_p^q \right)^{\frac{1}{q}}$$

$$M_{i,j} = \left| |A_i - A_j| \right|_p$$

   where,

   - A_k : Is the centroid of the cluster k

$$R_{i,j} = \frac{S_i + s_j}{M_{i,j}}$$

$$D_i = max_{j \neq i} R_{i,j}$$

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i$$

4. **Calinski Harabasz Score** – It is defined as the ratio between the within-cluster dispersion and between cluster dispersion.

$$CH(k) = \frac{B(k)}{W(k)} * \frac{(n-k)}{k-1}$$

Where,

- B(k) : between cluster variation
- W(k) : within cluster variation
- N : number of data points
- K : number of clusters

The higher the score, the better the clustering results.

5. **Homogeneity Score** - A clustering result satisfies homogeneity if all its cluster contain only data points that are member of a single class.

$$h = 1 - \frac{H(Y_{true}|Y_{pred})}{H(Y_{true})}$$

6. **Completeness Score** – A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same class.

$$c = 1 - \frac{H(Y_{pred}|Y_{true})}{H(Y_{pred})}$$

7. **V measure score** – The v-measure score is the harmonic mean between homogeneity and completeness

$$v = \frac{(1 + beta) * homogenity * completeness}{beta * homogenity * completeness}$$

8. **Adjusted random score** – The random score measures the similarity between the two clusters by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clustering. Adjusted random score accounts for the chance . A lower score indicates a poorer result.

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

9. **Adjusted mutual info score** - Adjusted mutual info score is an adjustment of the mutual information (MI) score to account for chance. It accounts that MI is higher for two clustering with higher clusters.

$$AMI(u,v) = \frac{MI(u,v) - E\big(MI(u,v)\big)}{avg\big(H(u), H(v)\big) - E\big(MI(u,v)\big)}$$

Related Work

Previous approaches for comparing the performance of clustering algorithms can be divided according to the nature of used datasets. While some studies use either real-world or artificial data, others employ both types of datasets to compare the performance of several clustering methods.

A comparative analysis using real world dataset is presented in several works [3, 4, 5, 6, 7, 8]. Clustering algorithms use a combination of clustering measurements, including external and internal validity indexes to compare their performances. The results in the above mentioned works, show us that there is no single algorithm which achieves the best performance on all the measurement for any dataset. Hence, it is necessary to use multiple metrics to measure the performance of a clustering algorithm.

In [4], experiments were performed to compare five different types of clustering algorithms: CLICK, self-organized mapping-based method (SOM), k-means, hierarchical and dynamical clustering. Gene expression time series dataset was used for the above task. It was found that Dynamical Clustering, SOM and k-means gave a high accuracy for multiple experiments while hierarchical clustering did not perform well on larger datasets, yielding lower accuracy than expected.

In [7], the authors have used a financial dataset to compare between fuzzy based clustering algorithms and in [8], the authors have used a Network traffic dataset to perform clustering using K-Means and DBSCAN clustering algorithms.

Methods

K-Means Clustering

K-means clustering is a centroid-based algorithm, where we calculate distances of the data points to various cluster centroids and assign the point to the cluster with minimum distance to its centroid. Each cluster is associated with a centroid. The 5 major steps in K-means clustering are:

1. *Choose the number of clusters k*
2. *Select k random points from the data as centroids*
3. *Repeat until convergence:*
    a. *Assign all the remaining data points to the closest cluster centroid*
    b. *Recompute the centroids of the newly formed clusters*

Strengths of K-Means:

1. Guarantees convergence
2. Easily interpretable
3. Relatively simple to implement
4. Generalizes to clusters of different shapes and sizes

Weaknesses of K-Means:

1. Choose k manually
2. Dependent on centroid initialization
3. Difficult to classify data of varying sizes and density
4. Affected by outliers
5. Difficult to cluster data with higher dimensions

Experiments with Parameter Settings

Choosing the Number of Clusters

To choose the optimal number of clusters for the dataset, we plotted the Within Cluster Sum of Squared Distances (WCSSD) of the data points to their closest cluster centre. In order to choose the best number of clusters we see if the plot looks like an arm and then choose the elbow on the arm as the optimal number of clusters. We observe that with increase of k, the WCSSD decreases gradually but we would like to select the point where the drop in the graph is substantial at a low cluster number, so that the model can be generalized to unseen data and also avoid the overhead of having many clusters.

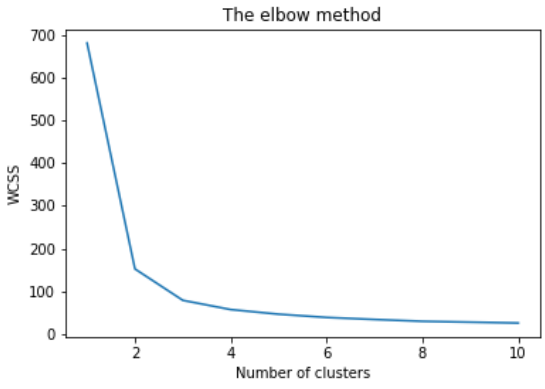

Figure 3: Elbow curve for Iris Dataset

From the above curve, we can see that the elbow point for iris dataset is at 3 clusters, which seems valid as we know that iris dataset has three classes. Below is the result after classifying the dataset into 3 clusters:

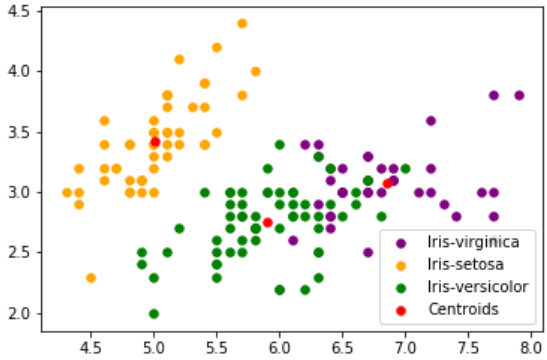

Figure 4: Clustering results for Iris Dataset

Metric	Score
Accuracy	0.893
Silhouette Score	0.735
Davies Bouldin Score	0.662
Calinski Harabasz Score	560.4
Homogeneity Score	0.751
Completeness Score	0.765
V Measure Score	0.758
Adjusted R and Score	0.73
Adjusted Mutual Info Score	0.755

Table 2: K-Means Clustering score on Iris Dataset

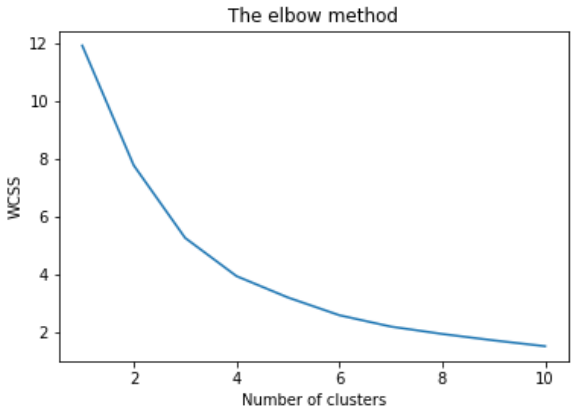

Figure 5: Elbow Curve for the Spiral Dataset

From the above curve, we can see that the elbow point for spiral dataset is at 4 clusters. Below is the result after classifying the spiral dataset into 4 clusters:

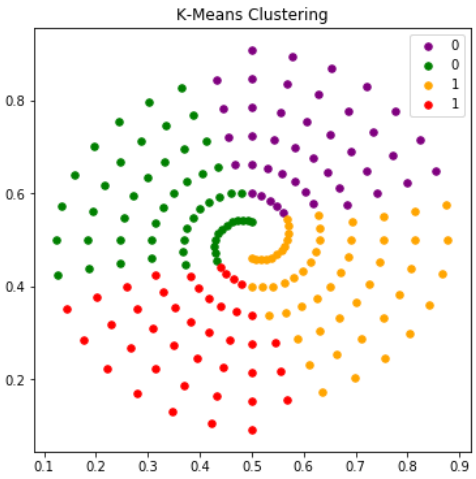

Figure 6: Clustering results for Spiral Dataset

Metric	Score
Accuracy	0.515
Silhouette Score	0.435
Davies Bouldin Score	0.915
Calinski Harabasz Score	127.526
Homogeneity Score	0.001
Completeness Score	0.001
V Measure Score	0.001
Adjusted R and Score	-0.004
Adjusted Mutual Info Score	-0.003

Table 3: K-Means Clustering score on Spiral Dataset

Out of the 4 clusters, 2 clusters belong to class 0 and 2 belong to class 1. The results for classification are not good for the spiral dataset using k-means because of the way k-means algorithm works. We know that k-means finds it difficult to classify datasets with complex spread and varying densities. The Spiral dataset is a perfect example of these kind of datasets and hence we can see that the results are not as good as for the Iris dataset where the points were clustered spatially.

The number of iterations of the k-means algorithm

The *sklearn* library gives us an option to specify the *n_init* parameter for a run of *kmeans*. This parameter signifies the number of times the *kmeans* algorithm will be run with different centroid initializations. The final result will then be the best output of the *n_init* consecutive runs in terms of the WCSSD calculated by the algorithm internally.

Metric	Scores	
	n_init = 10	n_init = 1
Accuracy	0.893	0.813
Silhouette Score	0.735	0.698
Davies Bouldin Score	0.662	0.649
Calinski Harabasz Score	560.4	483.337
Homogeneity Score	0.751	0.67
Completeness Score	0.765	0.737
V Measure Score	0.758	0.702
Adjusted R and Score	0.73	0.605
Adjusted Mutual Info Score	0.755	0.698

Table 4: Different number of iterations on Iris Dataset

Metric	Scores	
	n_init = 10	n_init = 1
Accuracy	0.515	0.541
Silhouette Score	0.435	0.41
Davies Bouldin Score	0.915	0.915
Calinski Harabasz Score	127.526	111.932
Homogeneity Score	0.001	0.005
Completeness Score	0.001	0.005
V Measure Score	0.001	0.005
Adjusted R and Score	-0.004	0.002
Adjusted Mutual Info Score	-0.003	0.001

Table 5: Different number of iterations on Spiral Dataset

Different algorithms

Elkan: This algorithm accelerates the discovery of clusters in the K-Means algorithm while giving exactly the same result as the standard algorithm. It uses triangle inequality to avoid unnecessary distance calculations by keeping track of the lower and upper bounds for distances between points and centres. This algorithm is much more effective with higher number of clusters and with higher dimensional data.

Since Elkan algorithm gives the same result as the default run (with "full" as the algorithm parameter) of K-Means, the only difference we observed was for the time. The Elkan algorithm took ***0.00072s*** to train the k-means on the Iris Dataset with 3 clusters, while the default run took about ***0.00094s*** to train the k-means. Similarly the Elkan algorithm took ***0.00102s*** to train on the Spiral Dataset with 4 clusters, while the default run

took ***0.00138s***. Both the runs gave the exact same result and scores while testing.

K-Means++: This algorithm is a variation of K-Means where we initialize the centroids in a smarter way so that it can give better results compared to random centroid initializations. In this method, we pick the first centroid point randomly and then select a point which is farthest from the picked up point. We make this new point as the new centroid and then repeat this step until we find k centroids. This way all the points selected have maximum probability proportional to distances between them.

Below are the results for the cases when we use k-means++ and random initializations of centroids

Metric	Scores	
	K-Means ++	**Random**
Accuracy	0.893	0.84
Silhouette Score	0.735	0.664
Davies Bouldin Score	0.662	0.634
Calinski Harabasz Score	560.4	484.899
Homogeneity Score	0.751	0.7
Completeness Score	0.765	0.745
V Measure Score	0.758	0.722
Adjusted R and Score	0.73	0.642
Adjusted Mutual Info Score	0.755	0.718

Table 6: Different Algorithms on Iris Dataset

Metric	Scores	
	K-Means ++	**Random**
Accuracy	0.515	0.546
Silhouette Score	0.435	0.399
Davies Bouldin Score	0.915	1.077
Calinski Harabasz Score	127.526	101.857
Homogeneity Score	0.001	0.006
Completeness Score	0.001	0.007
V Measure Score	0.001	0.007
Adjusted R and Score	-0.004	0.004
Adjusted Mutual Info Score	-0.003	0.003

Table 7: Different Algorithms on Spiral Dataset

Mean Shift Clustering

Mean shift clustering or also known as the Mode seeking algorithm is an unsupervised learning algorithm that assigns the data points to clusters iteratively by shifting points toward the mode, i.e., region with the highest density of data points. The number of clusters is not provided as an input and is automatically determined by the algorithm based on the data and the bandwidth value.

1. *Define a window (bandwidth of the kernel) and place the window on a data point*
2. *Calculate the mean for all the points in the window*
3. *Move the centre of the window to the location of the mean*
4. *Repeat steps 2 and 3 until convergence*

Strengths:

1. It is very simple and intuitive.

2. The number of clusters does not need to be specified from beforehand.

3. It is very easy to run in multiprocessor setting with parallel computing.

Weakness:

1. It's very slow when the dataset size increases. The time complexity is O(n^2).

2. It is not very scalable when data set is huge.

Experiments with Parameter Settings

One of the biggest advantage of mean shift clustering is that the number of clusters need not be specified as a parameter. It is automatically calculated based on the quantile parameter. The bandwidth parameter is also calculated from the quantile using the *estimate_bandwidth()* function from *sklearn* using the data samples. Varying the *max_iteration* parameter also does not result in significant performance changes a until it is set very high or very low. Empirically we found out that quantile is the main parameter that needs to be tuned for mean shift clustering to give the best results. Tunning quantile while taking the default value for other parameters we get:

Quantile	0.15	0.2	0.25
Accuracy	0.907	0.793	0.667
Silhouette Score	0.694	0.635	0.809
Davies Bouldin Score	0.868	0.762	0.487
Calinski Harabasz Score	313.28	289.53	353.36
Homogeneity Score	0.796	0.673	0.579
Completeness Score	0.816	0.77	1.0
V Measure Score	0.806	0.718	0.734
Adjusted R and Score	0.759	0.593	0.568
Adjusted Mutual Indo Score	0.793	0.669	0.577

Table 8: Different quantiles on Iris dataset

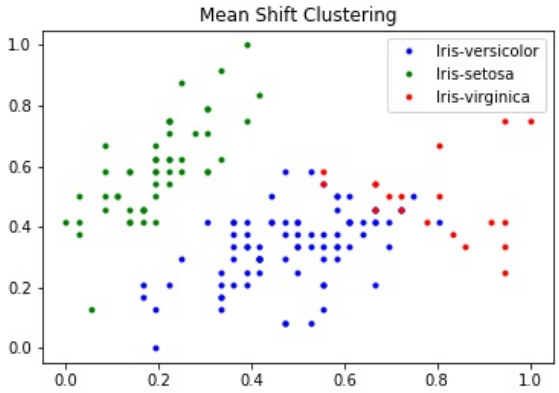

Figure 7 : Clustering results on Iris Dataset

Quantile	0.1	0.105	0.110
Accuracy	0.557	0.546	0.526
Silhouette Score	0.411	0.433	0.403
Davies Bouldin Score	1.37	1.32	1.16
Calinski Harabasz Score	83.44	86.16	74.6
Homogeneity Score	0.0097	0.0066	0.003
Completeness Score	0.01	0.0069	0.003
V Measure Score	0.0098	0.0068	0.003
Adjusted R and Score	0.0079	0.0038	-0.001
Adjusted Mutual Indo Score	0.0059	0.0029	-0.001

Table 9: Different quantiles on Spiral dataset

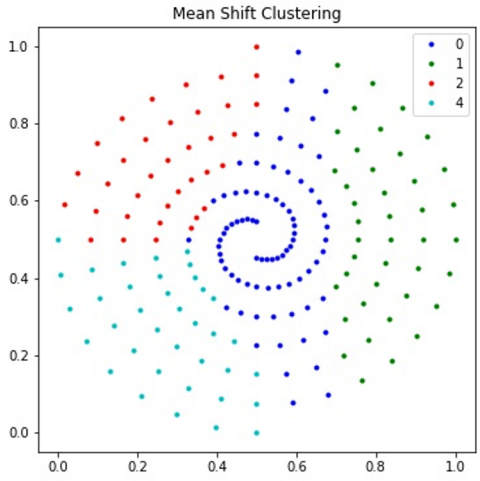

Figure 8 : Clustering results on Spiral Dataset

Agglomerative Hierarchical Clustering

Hierarchical clustering is a popular technique of building a hierarchy of the dataset using two strategies: Agglomerative (bottom-up approach) and Divisive (top-down approach). In this review, we focus on the Agglomerative approach where every data point is assumed to be its own cluster, and merged into bigger clusters as we move upwards in the hierarchy. For this technical review, we focus on agglomerative hierarchical clustering. The three major steps in the algorithm are:

1. *Initialization:*
    a. *Every point is its own cluster*
2. *Repeat:*
    a. *Find "most similar" pair of clusters*
    b. *Merge into a parent cluster*
3. *Until:*
    a. *The desired number of clusters have been reached*
    b. *There is only one cluster*

Strengths:

1. It is not required to set the number of clusters.
2. There is no assumption on the shape of the cluster.
3. The algorithm is non-probabilistic and simple to implement.

Weakness:

1. It is hard to make the algorithm work on big data.
2. There are many approaches available to calculate the similarity between the clusters, and each has their limitations.
3. In the bottom-up agglomerative approach, once a decision is made to combine two clusters, it is permanent and cannot be reversed.

4. The algorithm is sensitive to noisy data or data with outliers.
5. The model performance drops when we deal with clusters of varying sizes.

Experiments with Parameter Settings

Choosing the number of Clusters:

There is no standardized method to choose the number of clusters in agglomerative hierarchical clustering. One method is to use prior domain knowledge to choose the number of clusters or specify an explicit stopping criteria such as diameter or density thresholds. For our analysis, we have used Silhouette, Davies Bouldin, Homogeneity, and Calinski Harabasz Scores to determine the optimal number of clusters.

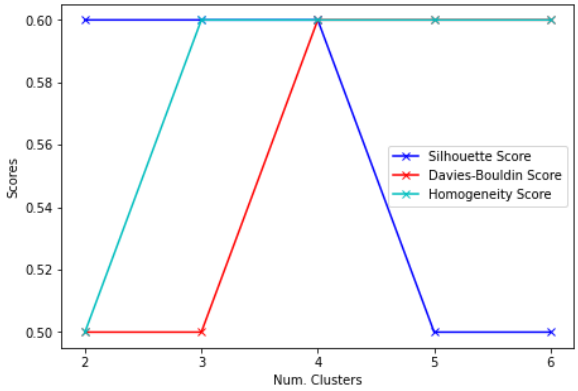

Figure 9: Metrics vs. Number of Clusters for the Iris Dataset

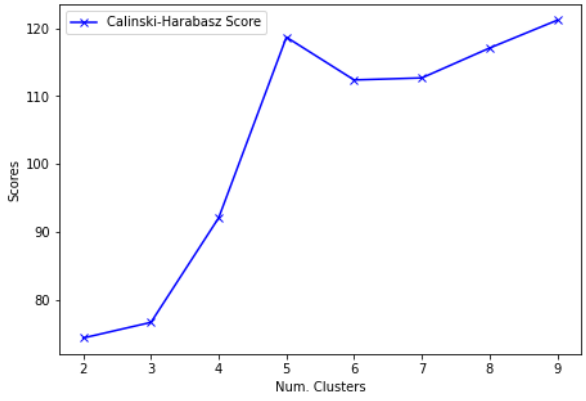

Figure 10: Metrics vs. Number of Clusters for the Spiral Dataset

For the Iris dataset, the number of clusters is set to 3 as the average of the three scores indicate the best results and is confirmed by our domain knowledge since the Iris dataset contains 3 types of flower species.

For the Spiral dataset, the number of clusters is set to 5 as per the Calinski Harabasz score.

Linkage Types:

There are various methods to calculate the distance between the clusters and every method has its own limitations. We have experimented with 4 types of linkage types:

1. **Ward Linkage** – Combining the clusters where increase in within cluster variance is to the smallest degree.

$$\Delta(A,B) = \sum_{i \in A \cup B} ||\vec{x_i} - \vec{m}_{A \cup B}||^2 - \sum_{i \in A} ||\vec{x_i} - \vec{m}_A||^2 - \sum_{i \in B} ||\vec{x_i} - \vec{m}_B||^2$$

2. **Complete Linkage** – Maximum distance between elements in clusters.

$$\max \{ d(a,b) : a \in A, b \in B \}.$$

3. **Average Linkage** – Average of the distances of all pairs.

$$\frac{1}{|A| \cdot |B|} \sum_{a \in A} \sum_{b \in B} d(a,b).$$

4. **Single Linkage** – Minimum distance or distance between closest elements in clusters.

$$\min \{ d(a,b) : a \in A, b \in B \}.$$

Linkage Type	ward	complete	average	single
Accuracy	**0.8**	**0.8**	**0.8**	0.7
Silhouette Score	**0.6**	**0.6**	**0.6**	0.5
Calinski Harabasz Score	380.2	**459.3**	**459.3**	154.3
Homogeneity Score	0.6	**0.7**	**0.7**	0.6
Completeness Score	**0.7**	**0.7**	**0.7**	1.0

Table 10: Different Linkage Types on the Iris Dataset

Linkage Type	ward	complete	average	single
Silhouette Score	**0.3**	**0.3**	**0.3**	-0.1
Davies Bouldin Score	0.9	1.0	0.9	**0.7**
Calinski Harabasz Score	118.7	95.6	**121.2**	4.6
Completeness Score	0.0	0.0	0.0	**1.0**

Table 11: Different Linkage Types on the Spiral Dataset

From **Table 10** and **Table 11**, 'complete' and 'average' linkage perform better in terms of accuracy and other metrics we have chosen for clustering evaluation. The 'single' linkage type performs worst. This is reasonable since 'single linkage' type tends to produce long thin clusters in which nearby elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other than two elements of other clusters [1].

Affinity Types:

There are various methods to calculate the linkage distance. Some popular distance metrics include Euclidean and Manhattan distances. We have experimented with 3 types of affinity types:

1. **Euclidian (L2 Norm)** – Length of a line segment between the two points.
$$d(p, q) = \sqrt{(p-q)^2}.$$
2. **Manhattan (L1 Norm)** – Sum of the absolute differences between the two vectors
$$\|a - b\|_1 = \sum_i |a_i - b_i|$$
3. **Cosine** – Average of the distances of all pairs, where S(C) is the cosine similarity.
$$\text{cosine distance} = D_C(A, B) := 1 - S_C(A, B).$$

Affinity Type	euclidean	manhattan	cosine
Davies Bouldin Score	**0.5**	**0.5**	0.6
Calinski Harabasz Score	**459.3**	380.2	452.9
Homogeneity Score	**0.7**	0.6	**0.7**

Table 12: Different Affinity Types on the Iris Dataset

Affinity Type	euclidian	manhattan	cosine
Accuracy	0.5	0.6	**0.6**
Silhouette Score	**0.3**	**0.3**	0.1
Davies Bouldin Score	1.0	**0.9**	2.0
Calinski Harabasz Score	95.6	110.0	40.8

Table 13: Different Affinity Types on the Spiral Dataset

From **Table 12** and **Table 13**, it is hard to determine which affinity type performs better on both the datasets. It is unclear because the different metrics indicate that different types of affinities are better for every case.

Tree Computation Types:

It is a feature in the Scikit Learn library to save computation time. We may stop early the construction of the tree at a specified number of clusters. This is useful to decrease computation time if the number of clusters is not small compared to the number of samples. This option is useful only when specifying a connectivity matrix [2]. Since the number of clusters are fixed in Step 1 of our analysis, early stopping of the tree does not make a difference in any of our results.

DBSCAN Clustering

Density based spatial clustering technique of application with noise (DBSCAN), is a density based clustering technique. Given a set of points, it tries to cluster together points that are densely packed in the spatial region. It also quantifies low densely packed data points as outliers. It does not require the number of clusters to be pre-defined and finds out the appropriate number during the clustering.

For this clustering technique, the algorithm requires epsilon, ϵ and min samples as the parameter. ϵ is defined as the neighbourhood region which the algorithm considers quantifying the density of the region. If the density is above the threshold, then all the points in the neighbourhood forms a cluster. Min sample point is the minimum number of sample points required in a region for a cluster to be formed. A core point are those points which have at least min points within the distance ϵ of the cluster.

The steps are defined as follows :

---

a. *Select ϵ and min points*
b. *Find the points in the ϵ of the neighbourhood of each point*
c. *Identify the core points which are more than the min point and are below ϵ distance from the core point.*
d. *Find the connected component of the core points in the neighbourhood.*
e. *Assign each non-core component to a neighbouring cluster if the cluster is an ϵ neighbour, else assign it to noise*

Strengths:

1. It does not require the number of clusters to be pre-defined.
2. DBSCAN has a notion of noise and is robust to outlier.
3. DBSCAN can find arbitrarily shaped clusters. By controlling the min points parameter, the single link effect is reduced.

Weakness:

1. DBSCAN is not entirely deterministic. Points that are on the border of two clusters can be categorised to either of the cluster depending on the order in which they are processed.
2. DBSCAN cannot cluster data with large differences in the densities.
3. If data and the distance measure used is not well understood, the threshold ϵ can become difficult to choose.

Experiments with Parameter Settings

Changing the value of ϵ

Figure 11 and 12 shows the result of varying epsilon on the iris and the spiral dataset.

In the Iris dataset, from inspection, ϵ as 1 gives the best result. The values 0.01and .05 gave the exact same clustering. Hence in the score reported below only 0.05 is reposted.

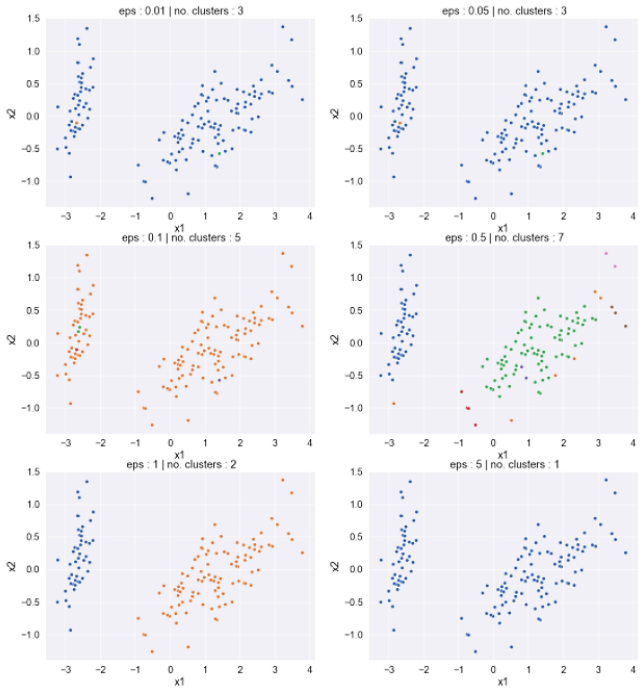

Figure 11 Clustering output for varying epsilon on Iris

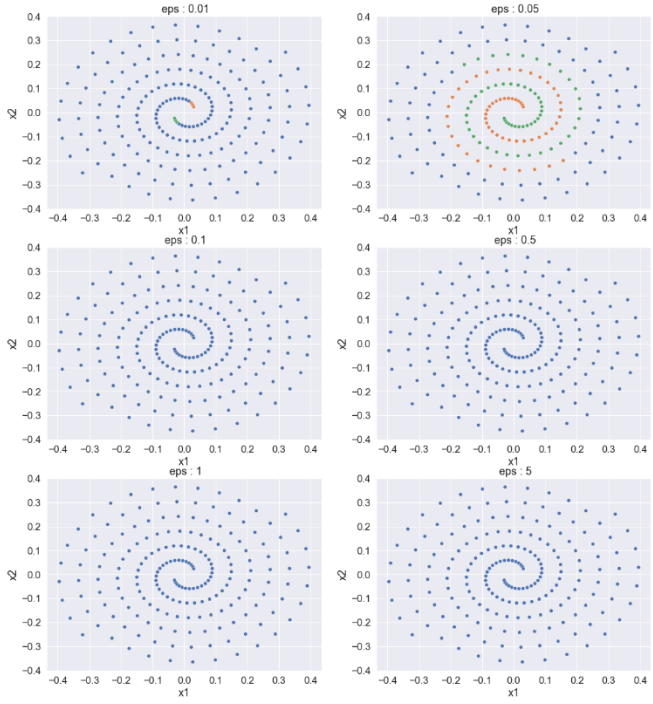

Figure 12 Clustering output on Spiral by varying epsilon

epsilon	0.05	0.1	0.5	1	5
# clusters	3	5	7	2	1
Accuracy	0.37	0.36	0.65	0.66	0.33
Silhouette Score	-0.53	-0.51	0.19	0.68	-
Davies Bouldin Score	1.07	0.93	5.84	0.38	-
Calinski Harabasz Score	2.99	3.23	110.2	501.92	-
Homogeneity Score	0.03	0.05	0.63	0.57	-
Completeness Score	0.22	0.21	0.62	0.99	-

Table 14 : Different epsilon on Iris dataset

In this case of the Spiral dataset, the values such 0.1, 0.5, 1 and 5 all had one cluster hance there was no metric scores calculated apart from accuracy. For the table below, only one of them is displayed.

epsilon	0.05	0.1	0.5
# clusters	3	3	1
Accuracy	0.5	0.5	-
Silhouette Score	-0.35	-0.08	-
Davies Bouldin Score	5.69	19.89	-

Calinski Harabasz Score	0.08	0.53	-
Homogeneity Score	0.03	0.58	-
Completeness Score	0.13	0.37	-
V measure score	0.050	0.45	-
Adjusted Random Score	0.0003	0.34	-
Adjusted Mutual Info Score	0.03	0.45	-

Table 15 : Different epsilon on Spiral dataset

As is evident from visual inspection, a ϵ of 0.1 was the most suited for the clustering parameter.

Changing the value of min points

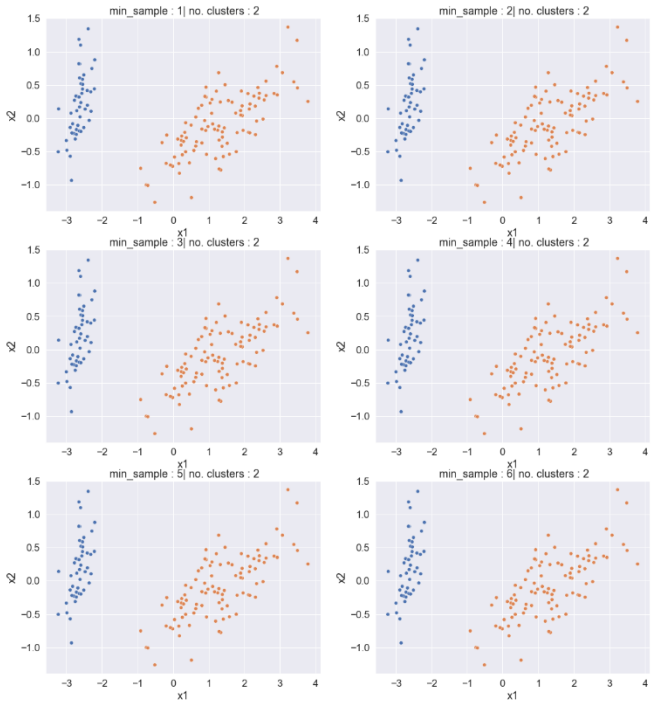

Figure 13: Clustering Output on Iris by varying min points

Changing the minimum points produced the same result for all the values. Hence only one of the values, 0.01 is displayed for the table below.

Min points	1
# clusters	2
Accuracy	0.66
Silhouette Score	0.68
Davies Bouldin Score	0.38
Calinski Harabasz Score	501.92
Homogeneity Score	0.58
Completeness Score	0.99
V measure score	0.73
Adjusted Random Score	0.56
Adjusted Mutual Info Score	0.73

Table 16: Different min points on Iris dataset

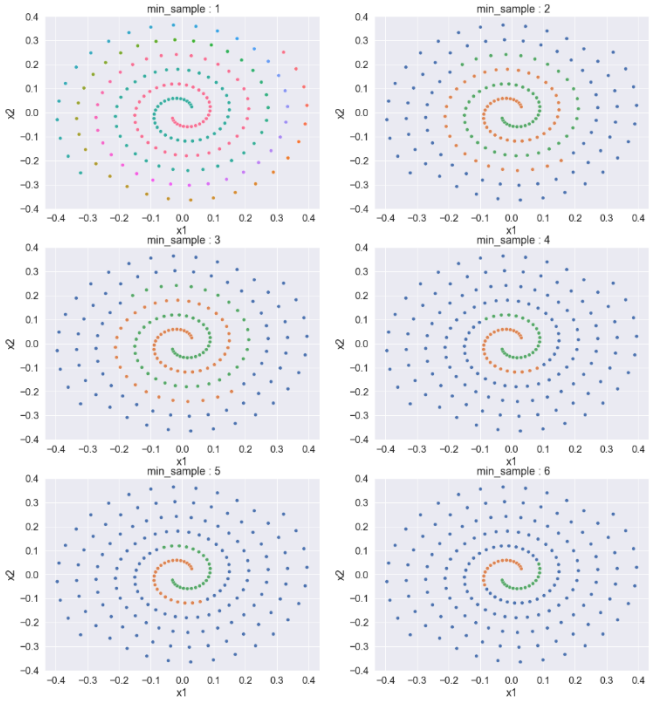

Figure 14: Clustering output on Spiral by varying min points

For the spiral dataset, setting the min points 2 and 3 gave the same result.

Min points	1	2	4	5	6
# clusters	82	3	3	3	3
Accuracy	0.01	0.5	0.5	0.5	0.5
Silhouette Score	0.19	-0.08	-0.19	-0.2	-0.23
Davies Bouldin Score	0.60	19.8	11.8	10.9	6.81
Calinski Harabasz Score	4.44	0.53	0.35	0.38	0.51
Homogeneity Score	1	0.58	0.25	0.24	0.16
Completeness Score	0.23	0.37	0.23	0.23	0.20
V measure score	0.38	0.45	0.24	0.24	0.18
Adjusted Random Score	0.45	0.34	0.06	0.05	0.024
Adjusted Mutual Info Score	0.26	0.45	0.24	0.23	0.17

Table 17: Different min points on Spiral dataset

Using K-distance graph

For the iris dataset, from the experiments above, the value of min points and thus k is set as 3. The plot for the k distance graph is shown below.

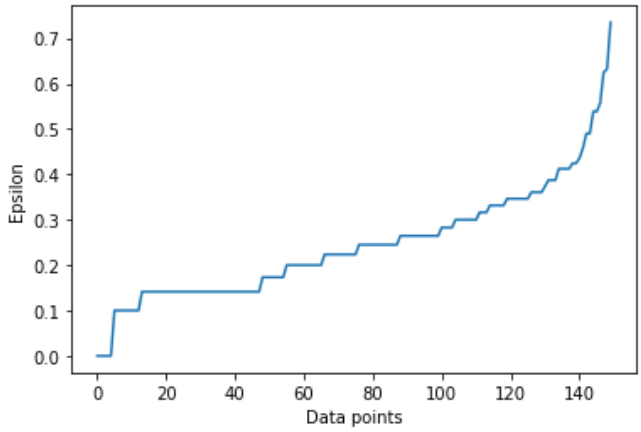

Figure 15 K-distance graph for Iris dataset

From graph, the value of ϵ was selected as 0.4 where an elbow is shown. The result is seen below for the clustering.

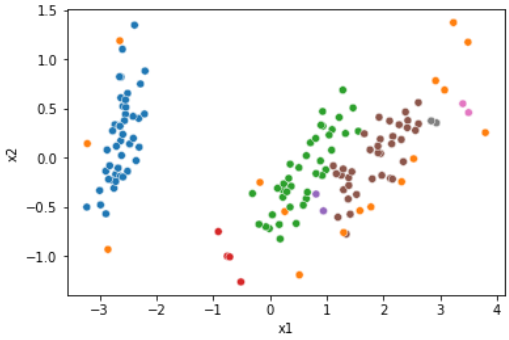

Figure 16 Clustering based on the K-distance graph on DBSCAN parameters

The number of clusters have increased(~8) due to the smaller ϵ. However, the difference between the prominent clusters or classes in the actual dataset is more visible.

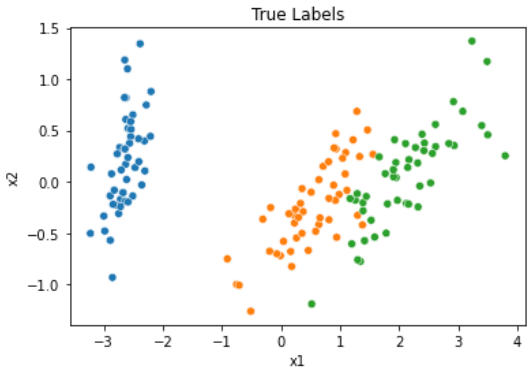

Figure 17 True Iris classes using PCA decomposition

For the spiral dataset using the experiments above, the min points were set to 3 since it produced the best clustering result.

K-distance graph was plotted for min point 3.

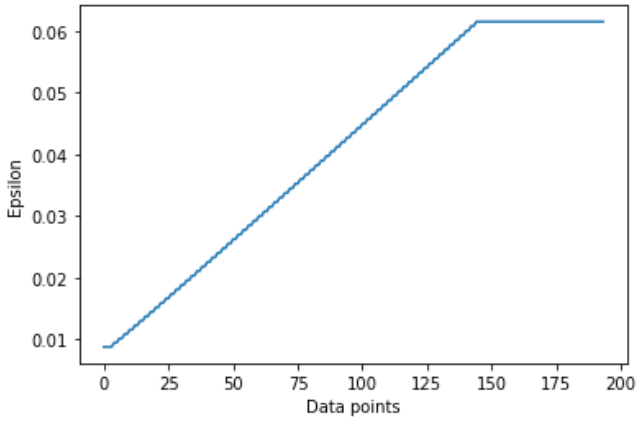

Figure 18 K-distance graph on Spiral Dataset

From the graph above, there was no elbow as such that could be identified. Hence, no conclusive value of epsilon could be derived from the k-distance graph. However, the initial value of 0.5 as ϵ that was selected during the experiments previously does fall in the rise. Hence setting 0.5 as the ϵ does seem to be valid.

Conclusion

To conclude, the four clustering algorithms differ in the way they form clusters as illustrated in the paper. Also, while K-means and hierarchical clustering expect the number of clusters to be an input, the mean shift and DBSCAN clustering calculates it automatically based on other parameters. The Iris dataset is a benchmark dataset for clustering algorithms. All four approaches described in the paper are able to perform equally well on the dataset.

The spiral dataset is a more challenging dataset, hence only DBSCAN can perform well in the spiral dataset. This could be because the algorithm considers the density while making the clusters. Hence for the two spirals, the DBSCAN can separate them based on the closely packed points in the clusters present in the centre. Since the two spirals have spacing between them, they are considered as separate clusters. However, as the spirals moves outwards, the density amongst the point within the same cluster increases, thus DBSCAN considers them as separate cluster. This could be the static nature of the epsilon that is required to be selected in the algorithm. The other three algorithms, K-Means, Mean Shift and Agglomerative Clustering algorithms are not able to perform well for the spiral dataset because they consider the distance between points rather than density of the points in a particular neighbourhood which is required for datasets like spiral.

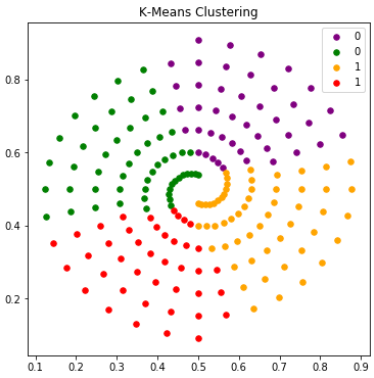

Figure 19: K-Means Clustering

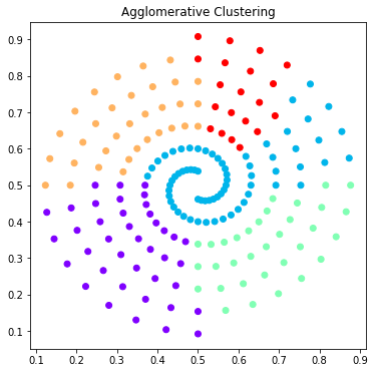

Figure 20: Agglomerative Clustering

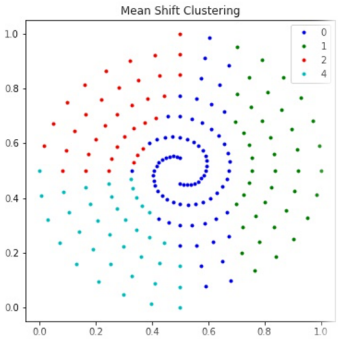

Figure 21: Mean Shift Clustering

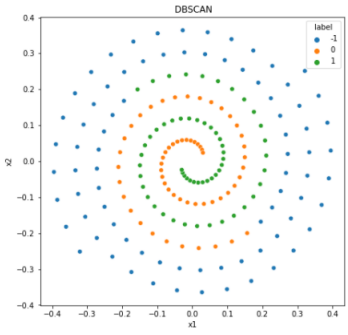

Figure 22: DBSCAN Clustering

Figure 23: Clustering Results on the Spiral Dataset

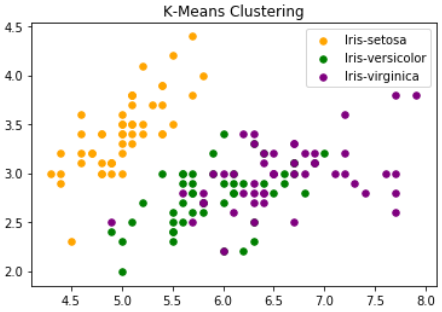

Figure 24: K-Means Clustering

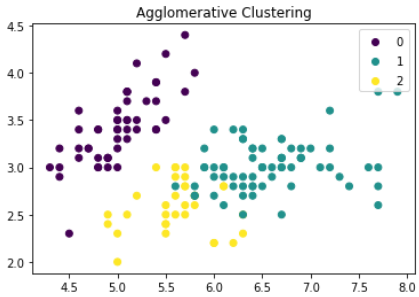

Figure 25: Agglomerative Clustering

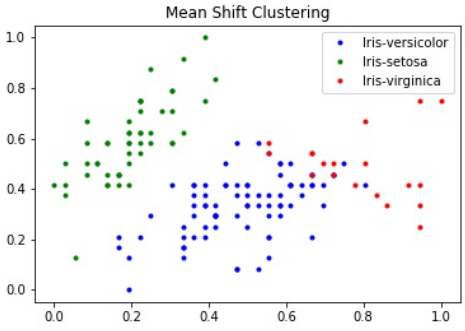

Figure 26: Mean Shift Clustering

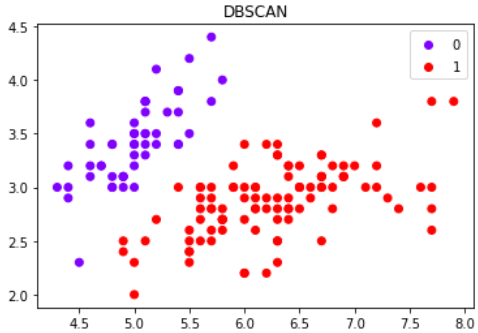

Figure 27: DBSCAN Clustering

Figure 28: Clustering Results on the Iris Dataset

Works Cited

[1] Wikipedia, “Single-linkage clustering,” 29 May 2021. [Online]. Available: https://en.wikipedia.org/wiki/Single-linkage_clustering.

[2] S. Learn, “sklearn.cluster.AgglomerativeClustering,” 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html.

[3] C. F. S. M. Costa IG, “Comparative analysis of clustering methods for gene expression time course data.,” in *Genetics and Molecular Biology*, 2004.

[4] K. M. H. J. Jung YG, Clustering performance comparison using K-means and expectation maximization algorithms. Biotechnology & Biotechnological Equipment, 2014.

[5] S. I. T. M. F. P. Kinnunen T, Comparison of clustering methods: A case study of text-independent speaker modeling. Pattern Recognition Letters, 2011.

[6] C. I. d. A. D. L. T. S. A. de Souto MC, Clustering cancer gene expression data: a comparative study. BMC bioinformatics, 2008.

[7] P. Y. W. G. Kou G, Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Information Sciences, 2014.

[8] A. M. M. A. Erman J, Traffic classification using clustering algorithms. In: Proceedings of the 2006 SIGCOMM workshop on mining network data. ACM, 2006.

Appendix

Agglomerative Clustering Full Results

Linkage Type	ward	complete	average	single
Accuracy	0.8	0.8	0.8	0.7
Silhouette Score	0.6	0.6	0.6	0.5
Davies-Bouldin Score	0.5	0.5	0.5	0.5
Calinski Harabasz Score	380.2	459.3	459.3	154.3
Homogeneity Score	0.6	0.7	0.7	0.6
Completeness Score	0.7	0.7	0.7	1.0
V Measure Score	0.7	0.7	0.7	0.7
Adjusted Rand Score	0.6	0.6	0.6	0.6
Adjusted Mutual Info Score	0.7	0.7	0.7	0.7

Table 18: Linkage Types on the Iris Dataset

Linkage Type	**ward**	**complete**	**average**	**single**
Accuracy	0.5	0.5	0.5	0.5
Silhouette Score	0.3	0.3	0.3	-0.1
Davies Bouldin Score	0.9	1.0	0.9	0.7
Calinski Harabasz Score	118.7	95.6	121.2	4.6
Homogeneity Score	0.0	0.0	0.0	0.0
Completeness Score	0.0	0.0	0.0	1.0
V Measure Score	0.0	0.0	0.0	0.0
Adjusted Rand Score	-0.0	0.0	-0.0	0.0
Adjusted Mutual Info Score	-0.0	0.0	-0.0	0.0

Table 19: Linkage Types on the Spiral Dataset

Affinity Type	**euclidean**	**manhattan**	**cosine**
Accuracy	0.8	0.8	0.8
Silhouette Score	0.6	0.6	0.6
Davies Bouldin Score	0.5	0.5	0.6
Calinski Harabasz Score	459.3	380.2	452.9
Homogeneity Score	0.7	0.6	0.7
Completeness Score	0.7	0.7	0.7
V Measure Score	0.7	0.7	0.7
Adjusted Rand Score	0.6	0.6	0.6
Adjusted Mutual Info Score	0.7	0.7	0.7

Table 20: Affinity Types on the Iris Dataset

Num. Clusters	5	5	5
Affinity Type	euclidian	manhattan	cosine
Accuracy	0.5	0.6	0.6
Silhouette Score	0.3	0.3	0.1
Davies Bouldin Score	1.0	0.9	2.0
Calinski Harabasz Score	95.6	110.0	40.8
Homogeneity Score	0.0	0.0	0
Completeness Score	0.0	0.0	0
V Measure Score	0.0	0.0	0
Adjusted Rand Score	0.0	0.0	0
Adjusted Mutual Info Score	0.0	0.0	0

Table 21: Affinity Types on the Spiral Dataset

Num. Clusters	5.0	5.0	5.0
type_exp	auto	True	False
Accuracy	0.5	0.5	0.5
Silhouette Score	0.3	0.3	0.3
Davies-Bouldin Score	0.9	0.9	0.9
Calinski-Harabasz Score	118.7	118.7	118.7
Homogeneity Score	0.0	0.0	0.0
Completeness Score	0.0	0.0	0.0
V Measure Score	0.0	0.0	0.0
Adjusted Rand Score	-0.0	-0.0	-0.0
Adjusted Mutual Info Score	-0.0	-0.0	-0.0

Table 22: Tree Computation Types on the Spiral Dataset

K-Means Full Results

Metric	Scores	
	n_init = 10	n_init = 1
Accuracy	0.893	0.813
Silhouette Score	0.735	0.698
Davies Bouldin Score	0.662	0.649
Calinski Harabasz Score	560.4	483.337
Homogeneity Score	0.751	0.67
Completeness Score	0.765	0.737
V Measure Score	0.758	0.702
Adjusted R and Score	0.73	0.605
Adjusted Mutual Info Score	0.755	0.698

Table 23: Different number of iterations on Iris Dataset

Metric	Scores	
	n_init = 10	n_init = 1
Accuracy	0.515	0.541
Silhouette Score	0.435	0.41
Davies Bouldin Score	0.915	0.915
Calinski Harabasz Score	127.526	111.932
Homogeneity Score	0.001	0.005
Completeness Score	0.001	0.005
V Measure Score	0.001	0.005
Adjusted R and Score	-0.004	0.002
Adjusted Mutual Info Score	-0.003	0.001

Table 24: Different number of iterations on Iris Dataset

Metric	Scores	
	K-Means ++	Random
Accuracy	0.893	0.84
Silhouette Score	0.735	0.664
Davies Bouldin Score	0.662	0.634
Calinski Harabasz Score	560.4	484.899
Homogeneity Score	0.751	0.7
Completeness Score	0.765	0.745
V Measure Score	0.758	0.722
Adjusted R and Score	0.73	0.642
Adjusted Mutual Info Score	0.755	0.718

Table 25: Different Algorithms on Iris Dataset

Metric	Scores	
	K-Means ++	Random
Accuracy	0.515	0.546
Silhouette Score	0.435	0.399
Davies Bouldin Score	0.915	1.077
Calinski Harabasz Score	127.526	101.857
Homogeneity Score	0.001	0.006
Completeness Score	0.001	0.007
V Measure Score	0.001	0.007
Adjusted R and Score	-0.004	0.004
Adjusted Mutual Info Score	-0.003	0.003

Table 26: Different Algorithms on Spiral Dataset

Mean Shift Full Results

Quantile	0.1	0.105	0.110
Accuracy	0.557	0.546	0.526
Silhouette Score	0.411	0.433	0.403
Davies Bouldin Score	1.37	1.32	1.16
Calinski Harabasz Score	83.44	86.16	74.6
Homogeneity Score	0.0097	0.0066	0.003
Completeness Score	0.01	0.0069	0.003
V Measure Score	0.0098	0.0068	0.003
Adjusted R and Score	0.0079	0.0038	-0.001
Adjusted Mutual Indo Score	0.0059	0.0029	-0.001

Table 27: Different Quantiles on the Spiral dataset

Quantile	0.15	0.2	0.25
Accuracy	0.907	0.793	0.667
Silhouette Score	0.694	0.635	0.809
Davies Bouldin Score	0.868	0.762	0.487
Calinski Harabasz Score	313.28	289.53	353.36
Homogeneity Score	0.796	0.673	0.579
Completeness Score	0.816	0.77	1.0
V Measure Score	0.806	0.718	0.734
Adjusted R and Score	0.759	0.593	0.568
Adjusted Mutual Indo Score	0.793	0.669	0.577

Table 28: Different Quantiles on the Spiral dataset

DBSCAN Full Result

epsilon	0.05	0.1	0.5	1	5
# clusters	3	5	7	2	1
Accuracy	0.37	0.36	0.65	0.66	0.33
Silhouette Score	-0.53	-0.51	0.19	0.68	-
Davies Bouldin Score	1.07	0.93	5.84	0.38	-
Calinski Harabasz Score	2.99	3.23	110.2	501.92	-
Homogeneity Score	0.03	0.05	0.63	0.57	-
Completeness Score	0.22	0.21	0.62	0.99	-

Table 29: Experiment on changing epsilon on Iris

epsilon	0.05	0.1	0.5
# clusters	3	3	1
Accuracy	0.5	0.5	-
Silhouette Score	-0.35	-0.08	-
Davies Bouldin Score	5.69	19.89	-
Calinski Harabasz Score	0.08	0.53	-
Homogeneity Score	0.03	0.58	-
Completeness Score	0.13	0.37	-
V measure score	0.050	0.45	-
Adjusted Random Score	0.0003	0.34	-
Adjusted Mutual Info Score	0.03	0.45	-

Table 30: Experiment on changing epsilon on Spiral

Min points	1
# clusters	2
Accuracy	0.66
Silhouette Score	0.68
Davies Bouldin Score	0.38
Calinski Harabasz Score	501.92
Homogeneity Score	0.58
Completeness Score	0.99
V measure score	0.73
Adjusted Random Score	0.56
Adjusted Mutual Info Score	0.73

Table 31: Experiment on changing min points on Iris

Min points	1	2	4	5	6
# clusters	82	3	3	3	3
Accuracy	0.01	0.5	0.5	0.5	0.5
Silhouette Score	0.19	-0.08	-0.19	-0.2	-0.23
Davies Bouldin Score	0.60	19.8	11.8	10.9	6.81
Calinski Harabasz Score	4.44	0.53	0.35	0.38	0.51
Homogeneity Score	1	0.58	0.25	0.24	0.16
Completeness Score	0.23	0.37	0.23	0.23	0.20
V measure score	0.38	0.45	0.24	0.24	0.18
Adjusted Random Score	0.45	0.34	0.06	0.05	0.024
Adjusted Mutual Info Score	0.26	0.45	0.24	0.23	0.17

Table 32 : Experiment on changing min points on Spiral