PROJECT -2

CLUSTERING ALGORITHMS

Charan Reddy Bodennagari - 50338186

Varsha Ravichandiran - 50315099

Sri Charan Chintapenta - 50313858

CSE 601 - Data Mining and Bioinformatics

Project 2 – Clustering Algorithms

Goal: The goal of this project is to implement five clustering algorithms such as K-means, Hierarchical Agglomerative clustering with Min approach, density-based, mixture model, and spectral clustering and evaluate their results using external index. The results are visualized as a scatter plot after PCA.

Dataset: gene datasets (cho.txt, iyer.txt)

Language: Python

Libraries: Numpy, Pandas, Matplotlib, Scipy

Cluster Evaluation :

The evaluation of the clusters formed are based on the External Indices namely Jaccard Indices and Rand Indices. These are considered as the most reliable evaluation techniques for clustering as they are better than other evaluations such as sensitivity. The calculation of the above mentioned indices consists of four important terms which are as follows

True Positives : It means the given points belongs to the same cluster in the ground truths as well as in the results of our clustering.

True Negatives : It means the given points do not belong to the same cluster in the ground truths as well as in our clustering results.

False Positives : It means the given points do not belong to the same cluster according to the ground truths but in our model prediction they belong to the same cluster.

False Negatives : It means the given points belong to the same cluster according to the ground truths but in our model we predict that they do not belong to the same cluster.

Jaccard Coefficient :

The Jaccard Similarity Index also known as the Jaccard Coefficient is a measure of similarity of two sample sets. It is defined as the ratio of the intersection of two sample sets to the union of those two sample sets. The sample sets in our case are ground truths and results set.

Hence Jaccard Index is given as

$$\text{Jaccard} = \frac{TP}{FP + TP + FN}$$

Where

TP is True Positive, where both the sample sets have same result

FP is False Positive, where the ground truths does not have given points in the same cluster but the results set has those points in the same cluster

FN is False Negative, where the ground truths has given points in the same cluster but the results set does not have those points in the same cluster

Rand Coefficient :

Rand Coefficient also known as the Rand Index is a metric for evaluation of the quality of clustering model. The Rand Index also considers the True Negatives along with the True Positives in the ratio. In other words it is the ratio of all the correctly predicted results to the union of all the results.

Hence the Rand Index is given as

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

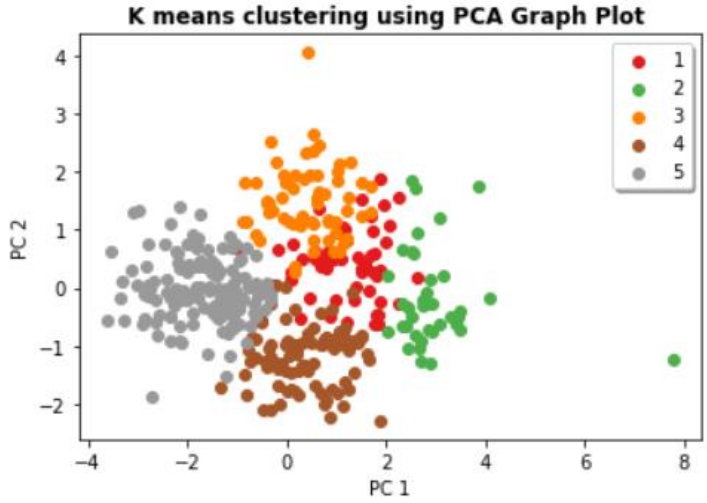
1. K Means Clustering :

It is the simplest and one of the most efficient clustering techniques in which we group all the similar data points into clusters. The number of clusters to be formed should be given in this clustering method as the number of centroids is set accordingly. It is a method in which initially the centroids are chosen randomly or manually and then it iteratively groups all the data points to similar clusters based on the Euclidean distance measure and the centroid is updated as the mean of all the data points in that cluster after every iteration. This process is repeated until there is not further change in the updated centroids or when the specified number of iterations are completed.

Implementation of the Algorithm :

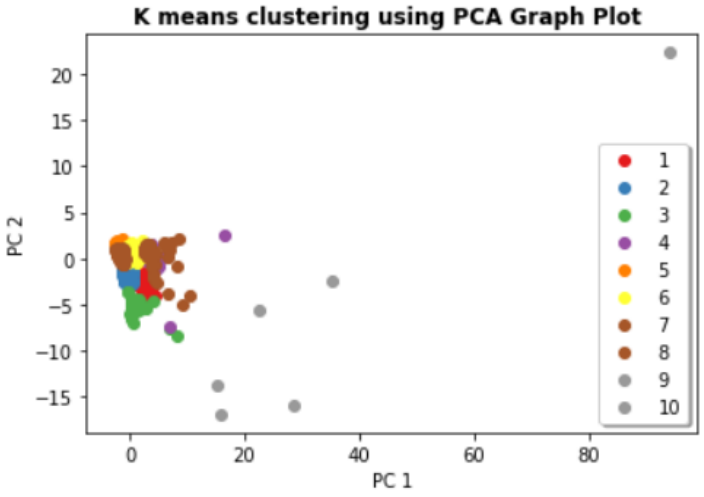
1. Initially the centroids are set either randomly or manually based on the number of clusters required (k) which is also given manually.
2. The Euclidean distance of each data point from all the centroids is calculated based on their attributes and the data point is assigned to the cluster having the least Euclidean distance from the data point.
3. Once all the data points are assigned to the clusters, the centroids are updated based on the mean of all the data points in that cluster.
4. The above steps are repeated until the cluster centroids are stabilized or the number of iterations is completed.

Results Visualization :




```
Jaccard Coefficient =  0.4097058706149978
Rand Index =  0.8014577572552283
```

Fig : Visualizations of cho.txt




```
Jaccard Coefficient =  0.375696956799319
Rand Index =  0.8353692071129003
```

Fig : Visualizations of iyer.txt

Results Evaluation :

The K Means Model converges faster, and in most cases has a time complexity efficient than other clustering techniques.

Cho.txt has all the data points spread across with even distance which means the average distance between the data points is healthy which resulted in five evenly grouped clusters.

Iyer.txt has majority of the data points densely populated in a specific region which means the average distance between two data points is imbalanced causing majority of the points to be grouped into the same clusters.

The Rand and Jaccard Coefficient values for cho.txt and iyer.txt are similar and overall K means has done an efficient job in predicting the clusters.

Advantages of K Means Clustering:

Easy to implement and time complexity is better than other clustering techniques.

The Scalability of K Means to large data sets is most reliable.

There will be a guaranteed convergence where the centroids will be stabilized.

Disadvantages of K Means Clustering:

Have to manually provide the number of clusters to be formed as it is difficult for the K Means model to predict the number of clusters to be formed.

It does not perform effectively against non-spherical shaped data set.

The results might be slightly different each time as the initial centroids are not the same every time. Hence it is heavily dependent on the initial centroids.

2. Hierarchical Agglomerative clustering:

Single-linkage clustering is one of the several methods of hierarchical clustering. It is based on grouping clusters in bottom-up fashion (agglomerative clustering), at each step combining two clusters that contain the closest pair of elements not yet belonging to the same cluster as each other. In the beginning of the agglomerative clustering process, each element is in a cluster of its own. The clusters are then sequentially combined into larger clusters, until all elements end up being in the same cluster. At each step, the two clusters separated by the shortest distance are combined. The function used to determine the distance between two clusters, known as the **linkage function**, is what differentiates the agglomerative clustering methods.

In single-linkage clustering, the distance between two clusters is determined by a single pair of elements: those two elements (one in each cluster) that are closest to each other. The shortest of these pairwise distances that remain at any step causes the two clusters whose elements are involved to be merged. The method is also known as nearest neighbor clustering. The result of the clustering can be visualized as a dendrogram, which shows the sequence in which clusters were merged and the distance at which each merge took place.

Mathematically, the linkage function – the distance D(X,Y) between clusters *X* and *Y* – is described by the expression

$$D(X,Y) = \min_{x \in X, y \in Y} d(x,y),$$

where *X* and *Y* are any two sets of elements considered as clusters, and *d*(*x*,*y*) denotes the distance between the two elements *x* and *y*.

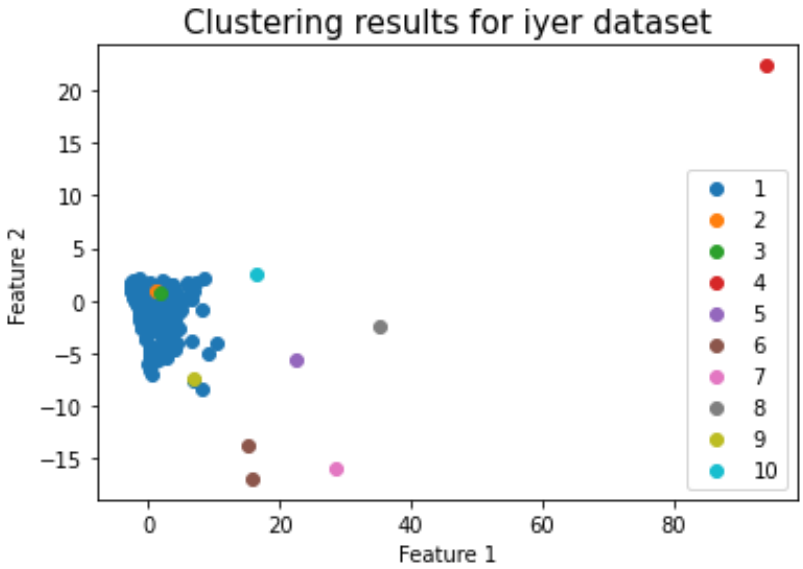
Implementation:

- Euclidean distances between the data points are calculated and stored in a matrix
- Minimum distance from the matrix is taken and their corresponding indices are merged into single cluster.
- The distance matrix is updated by popping the merged indices and recalculating the distances with the new merged indices.
- The steps 2 and 3 gets repeated until the corresponding number of clusters are obtained
- Finally, the clusters obtained for all the datapoints are plotted as a scatter plot along with the datapoints that are converted to 2D points using PCA.
- The clustering results are evaluated using external index like Jaccard Coefficient and Rand Index.

Results :

Jaccard Coefficient = 0.2297

Rand Index = 0.2424



Jaccard Coefficient = 0.1582

Rand Index = 0.1883

Results Evaluation :

The points which are closest to each other formed a cluster earlier than those far away from each other.

The clusters are not evenly distributed. It looks like a one dominant cluster. The points which are farther apart falls in separate cluster.

The Hierarchical Model is much more sensitive to outliers and noise which impacts the Jaccard Coefficient sometimes.

Advantages:

- Hierarchical clustering outputs a hierarchy, a structure that is more informative than the unstructured set of flat clusters returned by k-means. Therefore, it is easier to decide on the number of clusters by looking at the dendrogram.
- It is easy to implement.

Disadvantages:

- It is not possible to undo the previous step: once the instances have been assigned to a cluster, they can no longer be moved around.
- It is not suitable for large datasets due to its time complexity.
- It is very sensitive to outliers.
- Use of different distance metrics for measuring distances between clusters may generate different results.

3. Density Based:

Density based clustering algorithm is a spatial non-parametric algorithm. Given a set of points it groups together points that are closely packed and marking as outliers points that lie alone in a low-density region. The number of clusters need not be given as a parameter in the case of DBSCAN. It partitions the data into different clusters based on the structure of the data and the parameters given. It takes two parameters to cluster the data such as

Eps: The radius inside which the points are said to be in the neighborhood

Minpts: The minimum number of points that needs to be present in the neighborhood to construct a cluster to consider a point a core cluster point. Based on the two points the data points are classified into three categories as below

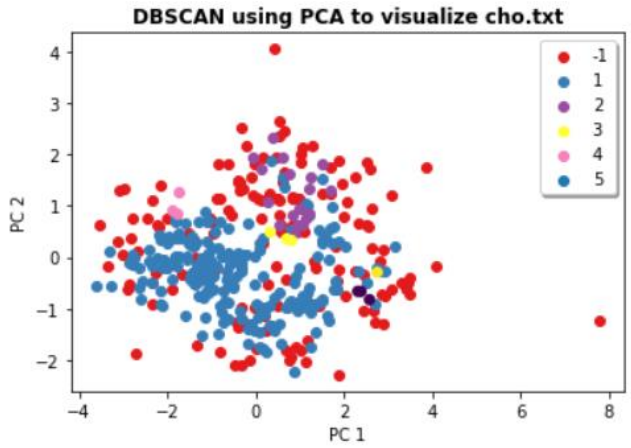
Core point: If a point has more than Minpts in its epsilon neighbor hood it is called a core point.

Border point: A point is considered border point if it is not a core point and it is in the neighborhood of a core point.
Outlier: A point is called an outlier if it is neither a core point nor a border point.

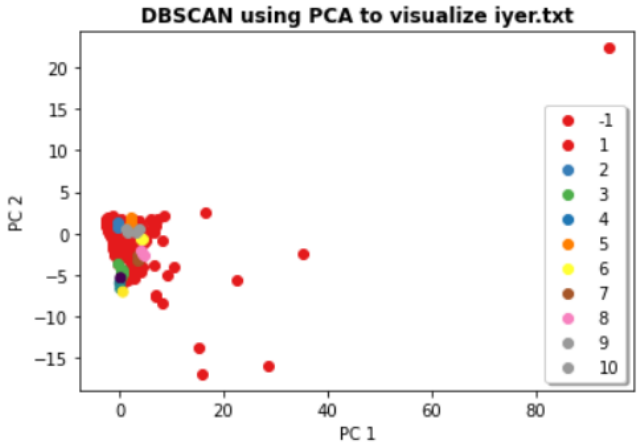
Implementation :

- Initially we take the input from the user to set the Epsilon and Min points parameters for our model.
- Then we calculate the distance matrix from each point to all other points.
- For every point in the dataset, if the data point is not already visited check for the neighbor points for that point and if the number of neighbors points is greater than the minpts consider this point as a core point.
- If it is a core point expand the cluster by finding out the indirectly reachable neighbors by calling the region query function.
- If it is not a core point mark it as an outlier

Results :




```
Jaccard Coefficient =  0.21188257110517064
Rand Index =  0.560726462455368
```




```
Jaccard Coefficient =  0.20348510711644638
Rand Index =  0.46404827733277465
```

Results Evaluation :

The Algorithm is highly sensitive to parameters as the clusters formed differed significantly for even a slight change in the parameter values of Epsilon and Min points.

For iyer.txt it has predicted most of the points as outliers as the parameter values did not meet the criteria and also the points are closely distanced.

For cho.txt we have seen better results as the points are evenly distributed across the data space and the outliers are handled better than other clustering techniques.

Advantages:

It can detect clusters of arbitrary shape.

The number of clusters need not be decided in advance, this algorithm divides the data into clusters based on the parameters.

It is not much affected by outliers; it is highly resistant to outliers.

Disadvantages:

It is quiet sensitive to the parameters.

This is not good with datasets of varying density.

Not ideal for datasets with sparse data

4. Gaussian Mixture Models:

Gaussian mixture models are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general do not require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.

A Gaussian mixture model is parameterized by two types of values, the mixture component weights, and the component means and variances/covariances. For a Gaussian mixture model with K components, the k^{th} component has a mean of μk and covariance matrix of Σk for the multivariate case. The mixture component weights are defined as ϕk for component Ck, with the constraint that $\sum_{i=1}^{K} \phi_i = 1$ so that the total probability distribution normalizes to 1. If the component weights are learned, they are the a-posteriori estimates of the component probabilities given the data.

If the number of components K is known, expectation maximization is the technique most commonly used to estimate the mixture model's parameters. In frequentist probability theory, models are typically learned by using maximum likelihood estimation techniques, which seek to maximize the probability, or likelihood, of the observed data given the model parameters.

Expectation maximization (EM) is a numerical technique for maximum likelihood estimation and is usually used when closed form expressions for updating the model parameters can be calculated. Expectation maximization is an iterative algorithm and has the convenient property that the maximum likelihood of the data strictly increases with each subsequent iteration, meaning it is guaranteed to approach a local maximum or convergence threshold.

Implementation:

1. Initialize the mean μk, covariance Σk and mixing coefficients πk by some random values are by given values.

2. Estimation step/E-step is performed by calculating the latent variables ϕk/Zk using the below formula

$$
\begin{aligned}
E(z_{ik}) &= p(z_{ik}=1|x_i,\pi,\mu,\Sigma) \\
&= \frac{p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)}{\sum_{k=1}^{K} p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)} \\
&= \frac{\pi_k \mathcal{N}(x_i|u_k,\Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|u_k,\Sigma_k)}
\end{aligned}
$$

3. Maximization step/M-step is performed to update the value of the parameters (μk, Σk, πk) calculated using ML method. These parameters are updated using the below formula

$$\pi_k = \frac{\sum_i r_{ik}}{n} \qquad \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$$

$$\Sigma_k = \frac{\sum_i r_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_i r_{ik}}$$

4.The above 2 steps are performed until the maximum log likelihood reaches a convergence threshold value. The maximum log likelihood is calculated using the below formula

$$E[\ln p(x,z|\pi,\mu,\Sigma)] = \sum_{i=1}^{n}\sum_{k=1}^{K} r_{ik}\{\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k,\Sigma_k)\}$$

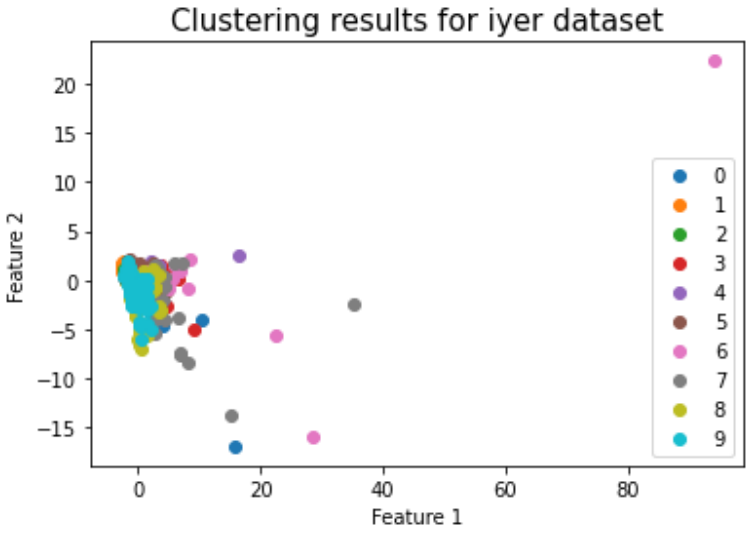
5. Finally, the clusters obtained for all the datapoints are plotted as a scatter plot along with the datapoints that are converted to 2D points using PCA.

6. The clustering results are evaluated using external index like Jaccard Coefficient and Rand Index.

Results :

Jaccard Coefficient = 0.2691

Rand Index = 0.6979



Jaccard Coefficient = 0.3309

Rand Index = 0.7902

Result Evaluation :

The Gaussian Mixture Model is comparatively slower than K means as it involves complex computations. Just as in the k-means expectation–maximization approach, this algorithm can sometimes miss the globally optimal solution, and thus in practice multiple random initializations are used.

Unlike KMeans , In GMM we do not have a binary assignment of each point to a particular cluster . Instead we have the Gaussian distribution determine the likelihood of a point belonging to a particular cluster .

Advantages:

It gives the probabilistic cluster assignments.
It can handle clusters with varying sizes and shapes.

Disadvantages:

Initialization parameters plays an important role in the cluster assignments.
Algorithm is complex in nature.
Computationally expensive if the number of distributions is large, or the data set contains very few observed data points**.**

5. Spectral Clustering :

Spectral Clustering is an unsupervised machine learning algorithm which implements a graph-based approach to determine the information related to local neighbors. In this type of clustering, the affinity determines what points fall under which cluster which means, A similarity graph is built to store the distances between the data points which in this case are the vertices of the graph and the weight of the edge is the similarity. The nodes are then mapped to a lower-dimensional space that can be easily segregated to form clusters. It is useful in clustering the datapoints when data is not spherical in shape and is in a complicated shape. The following are the main concepts used in implementing the Spectral Clustering Model.

Similarity Matrix : It is an nxn matrix, where n represents the total number of data points and each position ij of the matrix consists of the weight between the edges i and j. This weight is calculated using the gaussian kernel (w).

$$w_{ij} = \exp(-\| x_i - x_j \|^2 / \sigma^2)$$

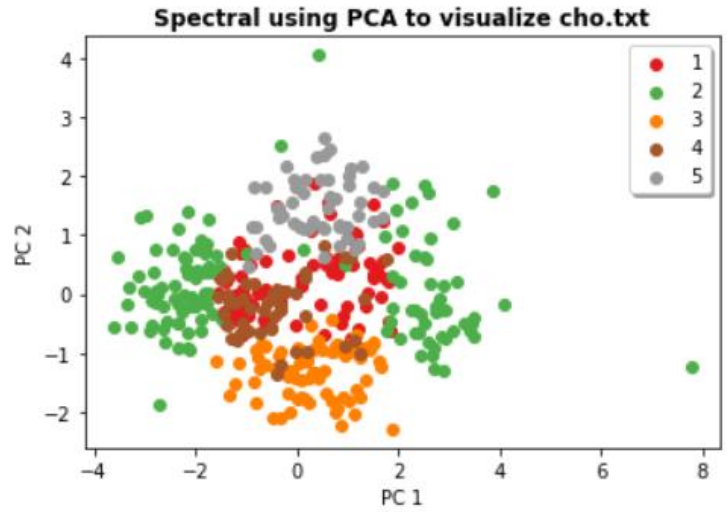
Degree Matrix : It is an nxn diagonal matrix, where each value ij where (i=j) consists of the sum of all the values in ith row.

Laplacian Matrix : It is an nxn symmetrical matrix which the result of subtracting the similarity matrix and the degree matrix.

Implementation of Algorithm :

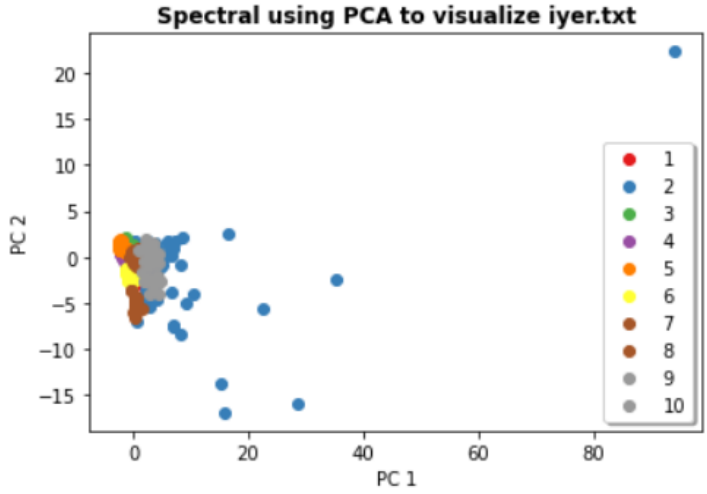
1. Initially we build the similarity matrix from the data points given. similarity to be a metric that determines how close two points are in our space. We use Gaussian Kernel to determine our similarity
2. We then construct a Degree matrix from our similarity matrix which is a diagonal matrix with sum of ith row in position “ii” in the matrix.
3. Next, we calculate the Laplacian matrix which is the difference of degree matrix and similarity matrix. We then calculate the eigenvalues and their corresponding eigenvectors of the Laplacian Matrix.
4. Reduce the embedded data space in such a way that we take the smallest K eigen values and their eigen vectors where K is the value which maximizes the complete expression.
5. Apply KMeans Algorithm to this reduced space and Label the points to their corresponding clusters.

Results Visualisations :




```
Jaccard Coefficient =  0.21447624540408045
Rand Index =  0.7074820800558405
```

Fig : Spectral Clustering for cho.txt




```
Jaccard Coefficient =  0.2746283737236111
Rand Index =  0.8227312010595274
```

Fig : Spectral Clustering for iyer.txt

Result Evaluation :

We can observe that as the sigma value increases we get better results for Rand and Jaccard coefficients as sigma exponentially weighs the proximity of two data points and decays exponential with the distance between two points.

The Gaussian Kernel which is used for determining the edge weights or the similarity between two data points also influenced the spectral clustering process as the similarity matrix is the most important part of the algorithm.

The choice of centroids when we apply KMeans to the reduced feature space also influenced the resulting clusters as the centroids can be chosen either randomly or manually.

The complex process of building similarity graph and then eigen vectors of the Laplacian matrix makes the algorithm slower than other clustering algorithms.

Advantages of Spectral Clustering :

Ability to find clusters for all arbitrary shaped datasets.

Not sensitive to parameters unlike some of the clustering techniques like DBSCAN.

Does not make strong assumptions about the dataset unlike some clustering techniques like Gaussian Mixture Model.

Disadvantages of Spectral Clustering :

Complex to implement as it requires calculation of similarity matrix using gaussian kernel and then reducing the embedded space.

The time taken would higher for large data sets as it is computationally expensive on large datasets.

The accuracy decreases and the time to implement increases significantly on large data sets.