CSE 601: DATA MINING AND BIOINFORMATICS

FALL 2018

PROJECT 2: CLUSTERING ALGORITHMS

REPORT

SIDDHARTH SELVARAJ #50247317

ALLEN DANIEL YESA #50246827

ANUSH RAVINDRA SHETTY #50247204

1. K MEANS ALGORITHM

K means is a simple and easiest clustering algorithm which is a type of unsupervised learning.

k-means clustering aims to classify n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

IMPLEMENTATION:

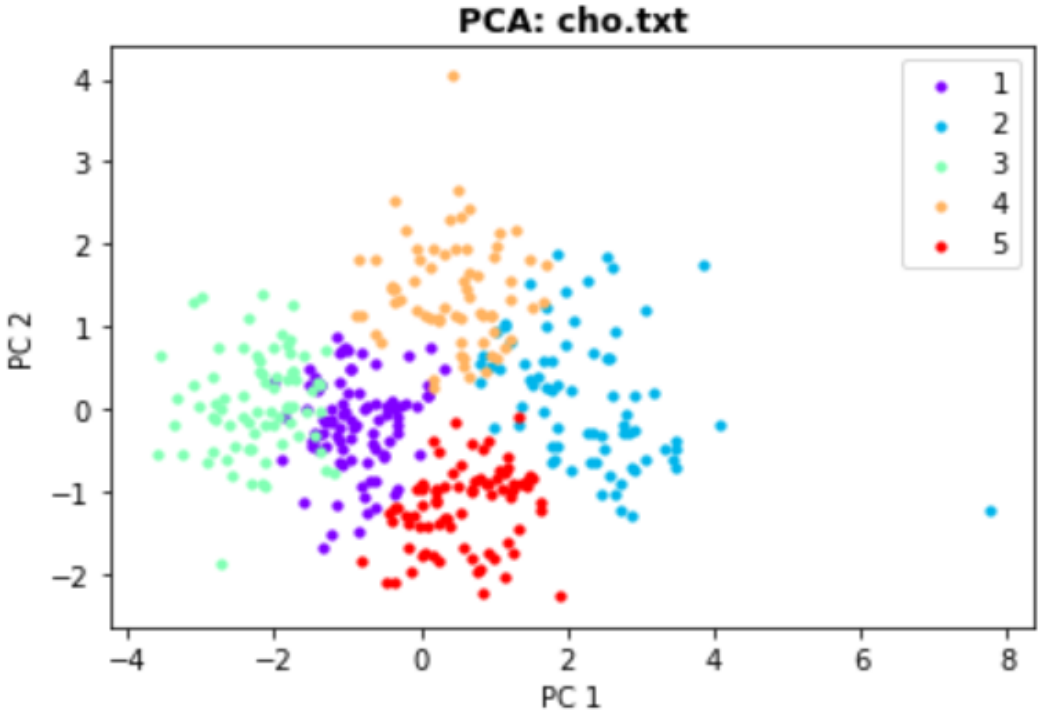
1. We get the input file name from the user and read the file
2. We get the gene id, ground truth and attributes from the file
3. We get the k value i.e. the number of clusters from the user.
4. We assign the centroids randomly form the k value or the get the centroids from the user.
5. We calculate the Euclidian distance and assign the point to the centroid which has minimum distance
6. We compute the new cluster centroid by taking the mean of all the points in the current cluster
7. We compare the new centroids with the old one and we repeat this process again and again until the centroids remain same or the number of iterations are over.
8. We then finally calculate the Jaccard and Rand Index and plot the clusters using PCA to analyze them visually.

VISUALIZATION:

We have used the inbuilt function for plotting the clusters. The high dimensional values are reduced to two dimensions using PCA and are mapped to the respective clusters

a) **cho.txt**

K=5

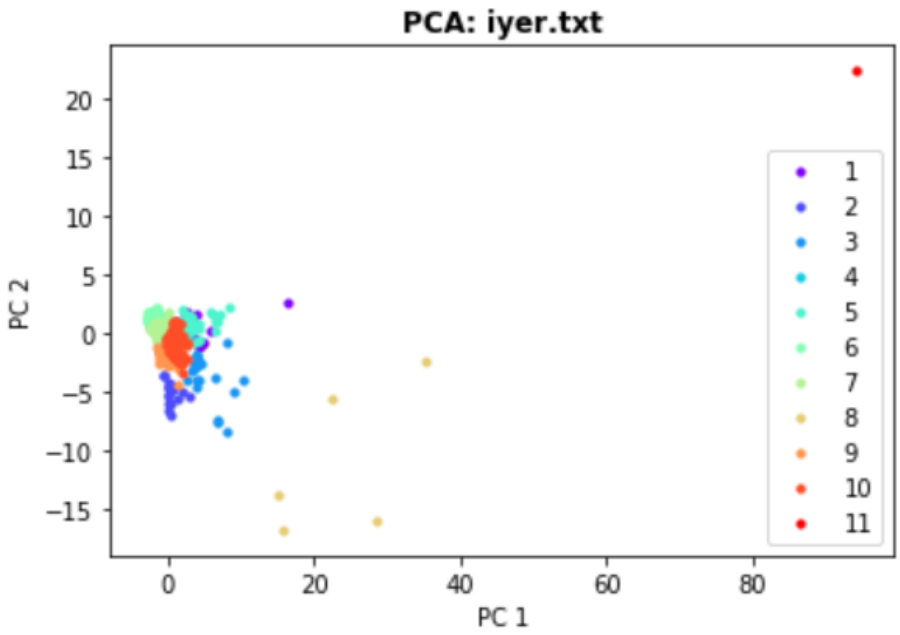


Jaccard Coefficient: 0.3393823358028802

Rand Index: 0.7869473005986738

b) **iyer.txt**

K=11



Jaccard Coefficient: 0.2961646443590269

Rand Index: 0.7814650060421491

RESULT EVALUATION:

When the centroids are chosen at random using the Forgy method, we find that the clustering is effective and the jaccard coefficient and rand index values are in the range of 0.3 and 0.78 respectively.

ADVANTAGES:

- It is simple and easy to implement
- Efficient: O(tkn), where n is # objects, k is # clusters, and t is # iterations. Normally, k, t << n

DISADVANTAGES:

- Need to specify K, the number of clusters in the beginning
- If there is no proper initialization of clusters, we may have local minimum
- We might end up having empty cluster sometimes
- Problem in selecting the centroids. If the centroids are from different clusters clustering becomes easy

VALIDATION OF CLUSTERING RESULTS:

- **External Index:**

We have used the Rand Index and Jaccard Coefficient to compare the clustering results

The similarities were computed using the following quantities:

- M01 = the number of attributes where p was 0 and q was 1
- M10 = the number of attributes where p was 1 and q was 0
- M00 = the number of attributes where p was 0 and q was 0
- M11 = the number of attributes where p was 1 and q was 1

- **Rand Index**

$$Rand = \frac{|Agree|}{|Agree|+|Disagree|} = \frac{|M_{11}|+|M_{00}|}{|M_{11}|+|M_{00}|+|M_{10}|+|M_{01}|}$$

May be dominated by M_{00} so we go for Jaccarrd Coefficient

- **Jaccard Coefficient**

$$Jaccard \quad coefficien \quad t = \frac{|M_{11}|}{|M_{11}|+|M_{10}|+|M_{01}|}$$

2. HIERARCHICAL AGGLOMERATIVE CLUSTERING

Hierarchical Agglomerative Clustering is a method of Clustering analysis which constructs a dendrogram using bottom-up approach, i.e. each attribute is considered as an individual cluster initially and pairs are made based on the sub method which we use to form clusters further.

In this approach we are using the Single Linkage Hierarchical Agglomerative Clustering which uses Euclidean Distance and then uses minimum distance (closest pair) between two data objects to form a cluster. Apart from Single Linkage, Complete Linkage and Average Linkage can be used for clustering which uses distances to compute the clusters

ALGORITHM:

1. For N objects and attributes within the input dataset, calculate distance matrix using Euclidean distance formula.
2. Further on every iteration we find the minimum value (closest pair) within the distance matrix. For this minimum value we take the object ids and merge the first object id with the other. In later iterations when we have several lists as objects label, we use the same technique and merge first list with another and delete the second list/object
3. Compute distances or similarities between objects and data points
4. Repeating the above steps we get the number of clusters required.

IMPLEMENTATION:

1. Once the input file is provided, it is parsed separating the attributes and ground truth value. The attributes are stored in a numpy array which is further used for calculation of Euclidean distance.
2. Existing library within sklearn.decomposition is leveraged and utilized for Euclidean distance calculation. We maintain an array of label which maintains the object indexes as specified.
3. Further we iterate over the gene data(attributes) and implement single linkage where we cluster objects using minimum value of attributes within the gene data
4. During the later iterations the data objects are combined as a list and single linkage applies to the combined output. During this process, we fill the subsequent data points with infinity as they are redundant data.
5. The clusters are labelled in the list and are combined as we run the implementation
6. We then plot the data points using PCA method and we calculate Jaccard's coefficient and Rand Index by computing M00, M01, M10 and M11.

VISUALIZATION:

a) **iyer.txt**

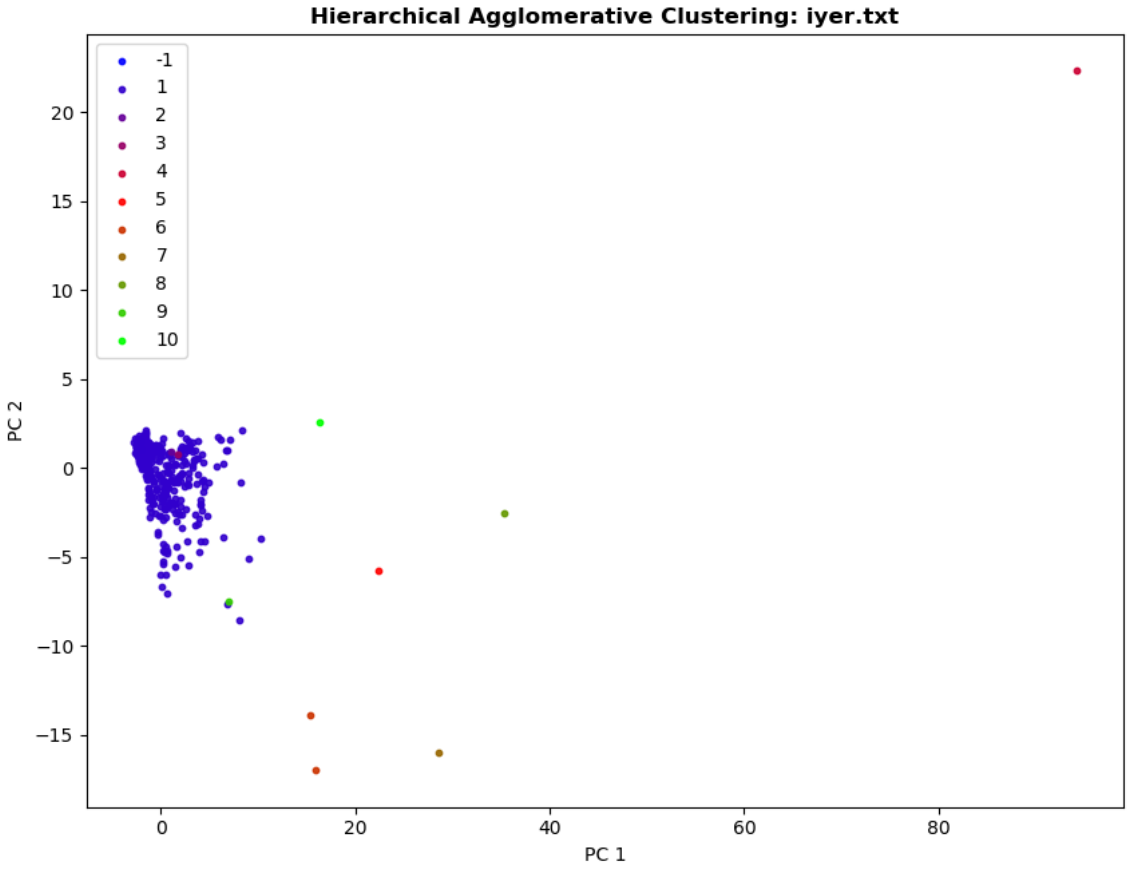


Fig. Hierarchical clustering with data(iyer.txt) and number of clusters = 10

Jaccard Coefficient: 0.1577184221520297

Rand index: 0.19291104385141178

b) cho.txt

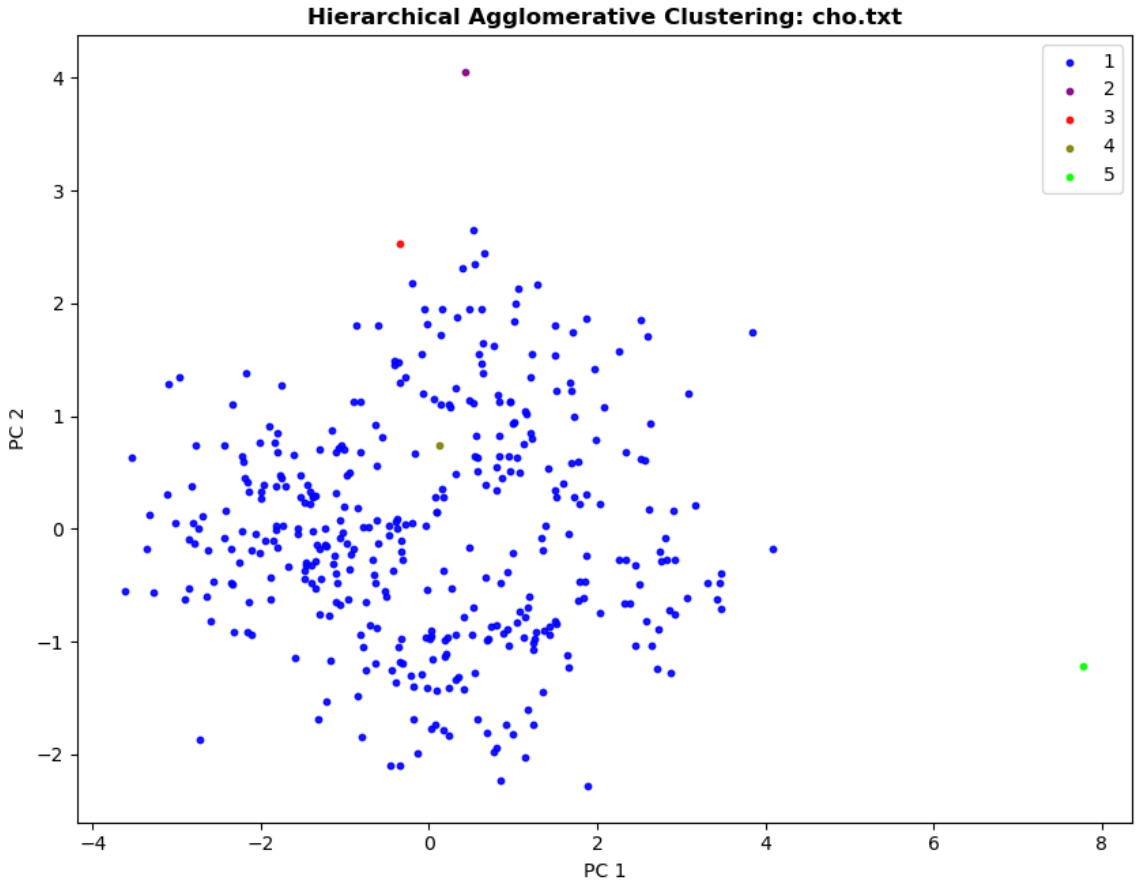


Fig. Hierarchical clustering with data(cho.txt) and number of clusters = 5

Jaccard Coefficient: 0.22839497757358454

Rand Index: 0.24027490670890495

RESULT EVALUATION:

1. In Hierarchical Agglomerative Clustering with Single Linkage, the distance between two clusters is represented by the distance of the closest pair of data objects belonging to different clusters.
2. Since there is no consideration of eps value or centroid, the clusters are determined by one pair of points, i.e., by one link in the proximity graph.
3. There is no guarantee to measure outlier and the clusters might be skewed thereby generating dendrogram which visualizes a lot of data objects together.

ADVANTAGES:

1. The data set provided can have non-elliptical shaped structure, i.e. the values can have different multivariate normal distribution of data.
2. It provides with ordering of the objects, which is useful for data visualization.
3. No prior information of clusters is required.
4. Dendrograms generated can be visualized for the clusters present in the data.

DISADVANTAGES:

1. Time complexity of this algorithm is high i.e. $O(n^2)$ as we have to compute distance matrix and then label the data objects.
2. A skewed image of clusters are seen due to chaining effect. This is particularly bad for datasets with high variance.
3. This algorithm is sensitive to outliers and noises. Hence a better approach would be to disregard all the data objects with high deviation.

3. DENSITY BASED SPATIAL CLUSTERING AND APPLICATION WITH NOISE

DBSCAN is a density based clustering algorithm. Based on the eps and the minimum points, it classifies the points as core points, border points and noise(outliers).

- Core points - A point which has more neighbors than the minimum points within the eps range.
- Border points - A point which has less neighbors than the minimum points within the eps range.
- Noise - A point that is neither a core nor a border point is classified as noise.

INPUTTING THE DATA:

The data files are inputted using the input () function. From understanding the input data, the first column is the gene_id, the second column is the ground truth clusters and the rest columns contain the gene's expression values (attributes). The eps value and the minimum points are inputted from the user.

DBSCAN IMPLEMENTATION:

The following steps were followed in implementing the density based spatial clustering and application with noise clustering:

1. For each gene that is unvisited, find its neighbors using the getNeighbors() function which returns points within the eps distance.
2. If the number of neighbors is greater or equal to the minimum points, it is classified to the corresponding cluster (using cluster_count).
3. If the number of neighbors is less than the minimum points, it is classified as noise.
4. Then this point is passed to the ExpandCluster() function where for every unvisited point in the neighborhood, its new neighbors are determined using the getNeighbors() function.
5. If the number of the new neighbors is greater or equal to the minimum points, the new neighbor points are merged with the neighbor points and the cluster is expanded.
6. After every point(gene) has been visited, the cluster list gives the cluster id of all the points.

VISUALIZATION:

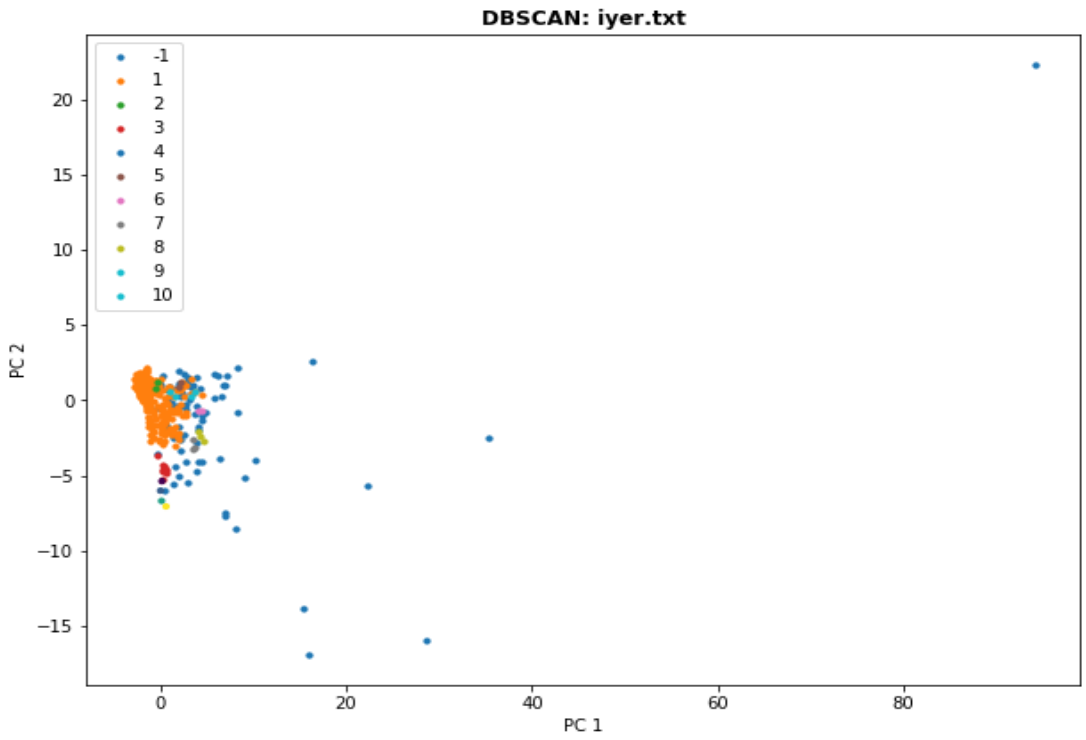
The visualization is obtained by the principal component analysis of the attributes. The high dimensional attributes are reduced to 2 dimensions using PCA which are then mapped to the respective clusters.

a) Iyer.txt:

Different values were set for the eps and minimum points and it was found that when the Eps is set to 1.42 and the minimum points as 2 ,the algorithm produces 10 clusters as in the given data. The outliers are labeled as -1 and the other points are labelled according to their cluster ids.

Eps= 1.42

Minimum points=2



Jaccard Coefficient = 0.20501094324625382

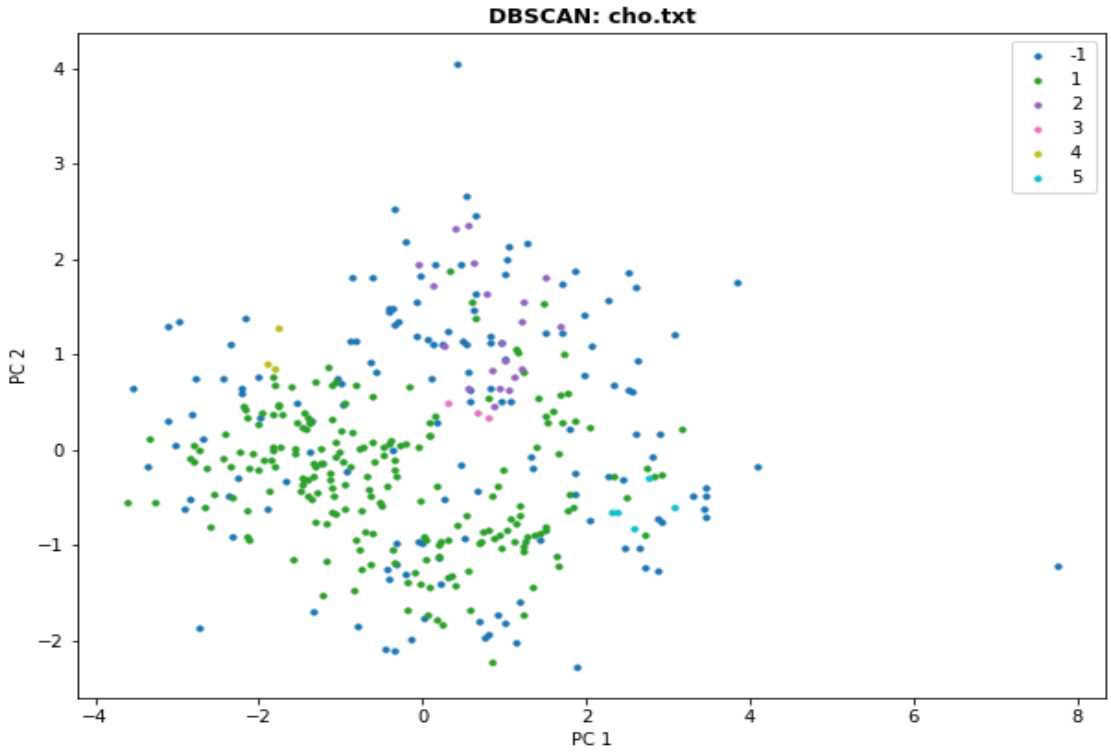
Rand Index = 0.46865003797387844

b) Cho.txt:

Different values were set for the eps and minimum points and it was found that when the Eps is set to 1.13 and the minimum points as 3 ,the algorithm produces 5 clusters as in the given data. The outliers are labeled as -1 and the other points are labelled according to their cluster ids.

Eps=1.13

Minimum points=3



Jaccard Coefficient = 0.21148292848649566

Rand Index = 0.5638272168380359

RESULT EVALUATION:

DBSCAN algorithm classifies points based on the number of neighbors it has within the eps radius. The time complexity of the algorithm is $O(n^2)$. It handles outliers and the clustering depends on the parameters used.

ADVANTAGES:

- DBSCAN does not require a pre-set number of clusters
- DBSCAN can handle clusters of different sizes and arbitrary shapes.
- It can handle noises.

DISADVANTAGES:

- It is sensitive to parameters.
- It is not capable of handling varying densities.

REFERENCES:

https://en.wikipedia.org/wiki/DBSCAN#Algorithm

4.MAP REDUCE K MEANS CLUSTERING ALGORITHM

IMPLEMENTATION:

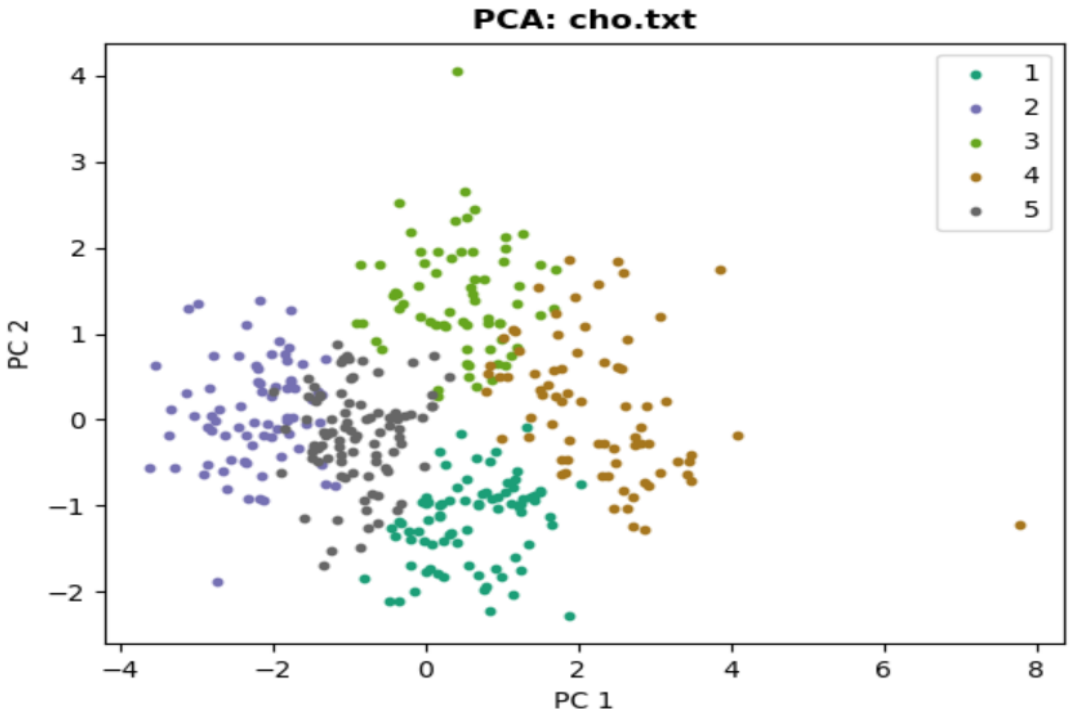
We have 4 python files:

- Main or driver file
  In this file we get the input from the user and call the mapper and reducer recursively until the centroids remain same or for the given number of iterations and we also calculate the external index.
- Mapper file
  In the mapper function we get the input file from the user and the centroids and we just assign the points to the centroids based on the distance and we pass the assigned centroid to the reducer.
- Reducer file
  In the reducer we get the input from the mapper and we recalculate the new centroids by taking the mean of the points that are assigned to a particular cluster. And write it into a file which will be used by the mapper again
- File to plot
  Finally, after the convergence or when the given number of iterations are done we plot the cluster using this file using PCA so that we get a visual representation of the clusters

1. We get the input file name from the user and read the file
2. We get the gene id, ground truth and attributes from the file
3. We get the k value i.e. the number of clusters from the user.
4. We assign the centroids randomly form the k value or the get the centroids from the user
5. We get the number of iterations from the user
6. We run the mapper and reducer until the centroids are same for 2 consecutive runs or the until end of the number of iterations specified
7. In the mapper we assign the attributes to the cluster and pass it as input to the reducer
8. In the reducer we get the cluster assignment and compute the new centroids by taking mean of the assigned values
9. Finally, in the main we assign the attributes to the respective centroids and calculate the external index
10. We use the plotting file to plot the plot using PCA

VISUALIZATION:

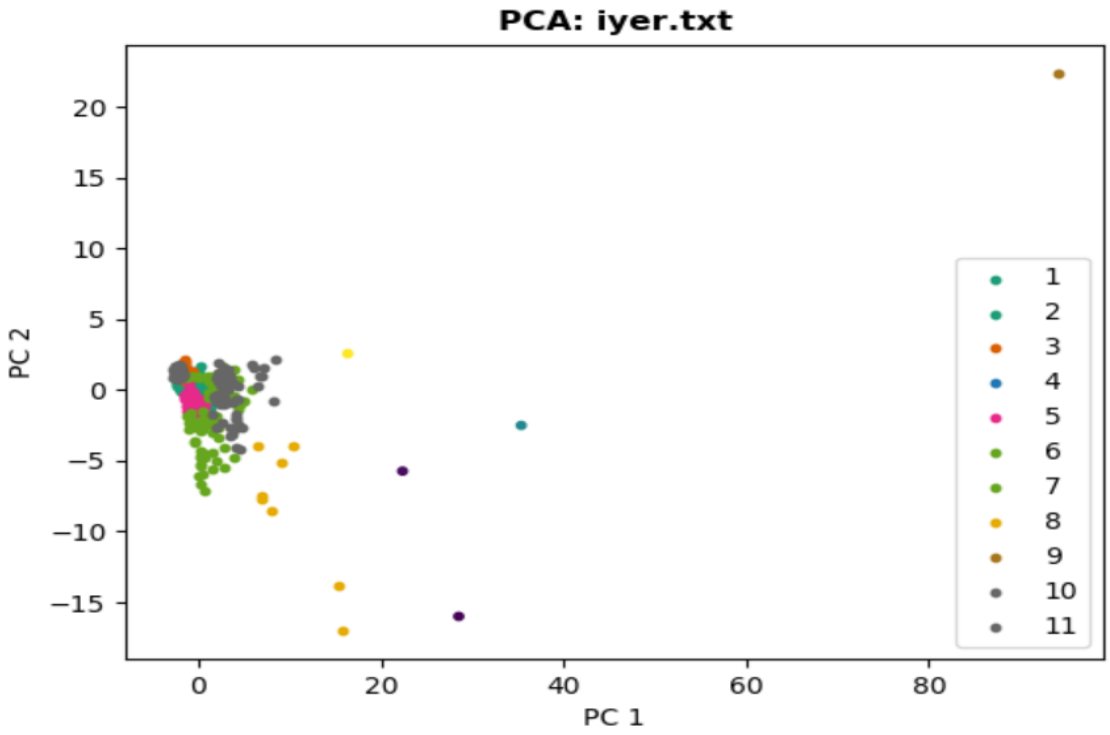
a) **Cho.txt**
K=5



Jaccard Coefficient: 0.34115342945245436

Rand Index: 0.7877661145265644

b) **Iyer**
K=11



Jaccard Coefficient: 0.3163583953879414

Rand Index: 0.8402852343343722

RESULT EVALUATION:

Map Reduce - KMeans highly scalable and it is more efficient in handling large data compared to simple KMeans algorithm. From the visualization we can see that the clustering done using Map Reduce is more effective which is evident from external index.

ADVANTAGES:

- Since it is a parallel implementation, the map reduce k-means is much faster for big datasets than the usual k-means algorithm.
- It is highly scalable for even multi-dimensional attributes.

DISADVANTAGES:

- It requires the number of clusters and iterations to be initialized.
- The time increases as the number of iterations increase.

EFFORTS TO IMPROVE PERFORMANCE

We have split the process to mapper and reducer in which the mapper part assigns the cluster to the points and the reducer part computes centroids iteratively.

We can use a combiner to improve the performance and the processing speed of the map reduce algorithm by decreasing the amount of intermediate read and write as mentioned in the paper: Improved MapReduce k-Means Clustering Algorithm with Combiner (https://ieeexplore.ieee.org/document/7046097)