CLUSTERING ALGORITHM

Data Mining and Bioinformatics

ABHINAV KUMAR	akumar39
SACHIN KUMAR KUPPAYYA	skuppayy
VINOOTH RAO KULKARNI	vinoothr

K –means Algorithm

K-means is a type of unsupervised learning, which is used when we have unlabeled data. The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features provided. Data points are clustered based on feature similarity.

Algorithm:

1. Split the data and assign to feature matrix, gene, ground_truth
2. Select the initial centroid. For this we are randomly select K points using randint function of numpy
3. Based on the initial centroids we assign the every data to particular cluster.
4. Now we apply the Kmeans algorithm in loop
5. Based on the points in the given cluster we take the mean of the data points and update this as new centroid.
6. For this new centroids we assign the points based on the closest distance to the centroid
7. We repeat this till there are no new centroid created or no new clusters are created
8. We compute the Jaccard coefficient using below formula

$$Jaccard\ coefficient = \frac{|M_{11}|}{|M_{11}| + |M_{10}| + |M_{01}|}$$

9. We compute Rand index using the below formula

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

Where, M_{11} : (agree, same cluster) M_{00} : (agree, different clusters)

M_{10} : (disagree, different clusters) M_{01}: (disagree, different clusters)

10. Using PCA reduce the dimension to 2D and plot the principal components using seaborn

For the given case, since we already knew the ground truth clusters we used that to define our k value. However, in general case we will not be having ground truth clusters. So we can use the elbow method to define K values

Results:

Input:
Data = cho.txt
K = 5
Initial centroid (randomly chosen using randint) = [11, 63, 326, 337, 351]

Output**:**
The Jaccard value: 0.4254
The Rand Index value: 0. 8076

Visualization

The PCA below is K-Means algorithm on cho.txt with cluster size = 5

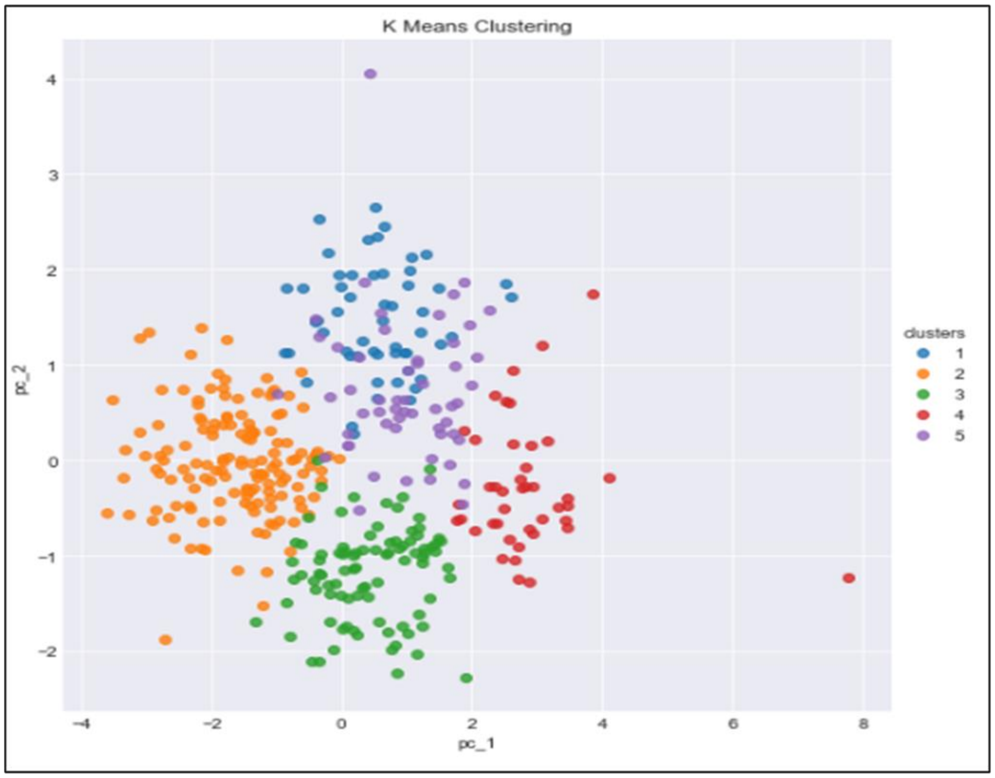


Fig 1:Kmeans Cho.txt

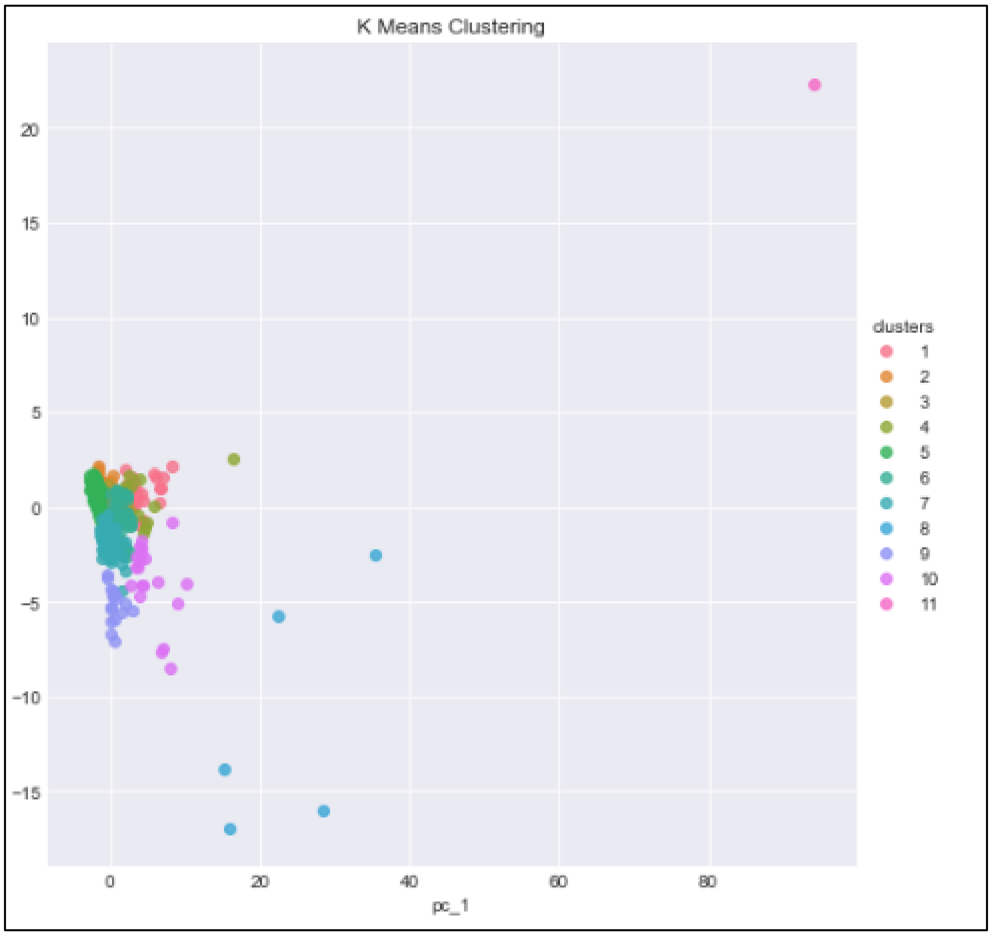
Results:

<u>Input:</u>
Data = iyer.txt
K = 11
Initial centroid = [5, 40, 49, 75, 110, 136, 316, 337, 393, 477, 506]

<u>Output:</u>
The Jaccard value: 0.3545
The Rand Index value : 0. 7982

Visualization

The PCA below is K-Means algorithm on iyer.txt with cluster size = 11



<u>Fig 2: Kmeans iyer.txt</u>

Advantages of K-means:

1. K-means runs is efficient for low values of k. Run time is O(tkn), where n is the number of objects, k is number of clusters and t is number of iterations
2. K-means simple and efficient to implement.
3. K-means is suitable for large data sets

Disadvantages of K-means:

1. K-means clustering requires prior knowledge of K. Choosing the right K values sometimes becomes difficult
2. In addition, we observed that it depends a lot on the initial centroids that we choose. Different random points generate different cluster.
3. Not suitable for all types of data as mostly restricted to data in which there is a notion of a center
4. Empty clusters may appear

Future work:

1. To solve initial centroid problem we can use have multiple runs and choose the one with smaller SSE
2. We can use other methods like hierarchical clustering to determine initial centroids
3. To resolve empty clusters we can choose the one that contributes to most SSE
4. Normalizing and eliminating outliers in preprocessing can help in improving the accuracy

Hierarchical Agglomerative Clustering with Single Link

Hierarchical Agglomerative Clustering builds the hierarchy starting from individual points in the dataset. Each iteration two closest elements are merged based on the chosen distance and updated to the remaining points. Iterations are repeated until only one cluster is left.

We use single linkage so we consider the pair of point which are closest to each other.

Algorithm:

1. Read the data using pandas
2. Initialize each element to a cluster
3. Calculate the distance matrix, which is the distance between each pair of points
4. Find the index of the two closest elements
5. Merge the two closest points and update the distance
6. If we already have a cluster then update the distance based on the minimum distance
7. Delete the clusters, which was already merged from the matrix
8. Repeat the merge until all items are merged
9. We compute the Jaccard coefficient using below formula

$$Jaccard\ coefficient = \frac{|M_{11}|}{|M_{11}| + |M_{10}| + |M_{01}|}$$

10. We compute Rand index using the below formula

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

Where, M_{11} : (agree, same cluster) M_{00} : (agree, different clusters)

M_{10} : (disagree, different clusters) M_{01}: (disagree, different clusters)

11. Using PCA reduce the dimension to 2D and plot the principal components using seaborn

Results:

Input:
Data = cho.txt
K = 5

Output:
The Jaccard value: 0.2284
The Rand Index value: 0. 2403

Visualization

The PCA below is HAC algorithm on cho.txt with cluster size = 5

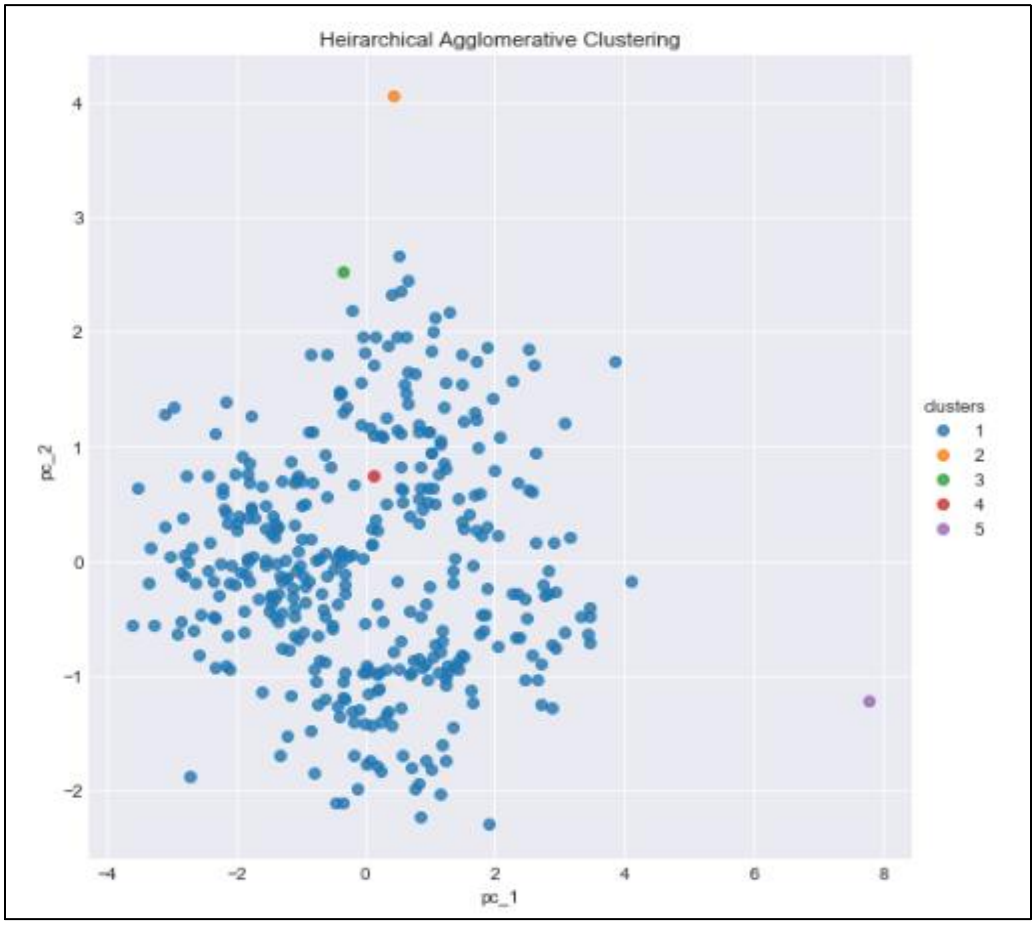


Fig 3: HAC cho.txt

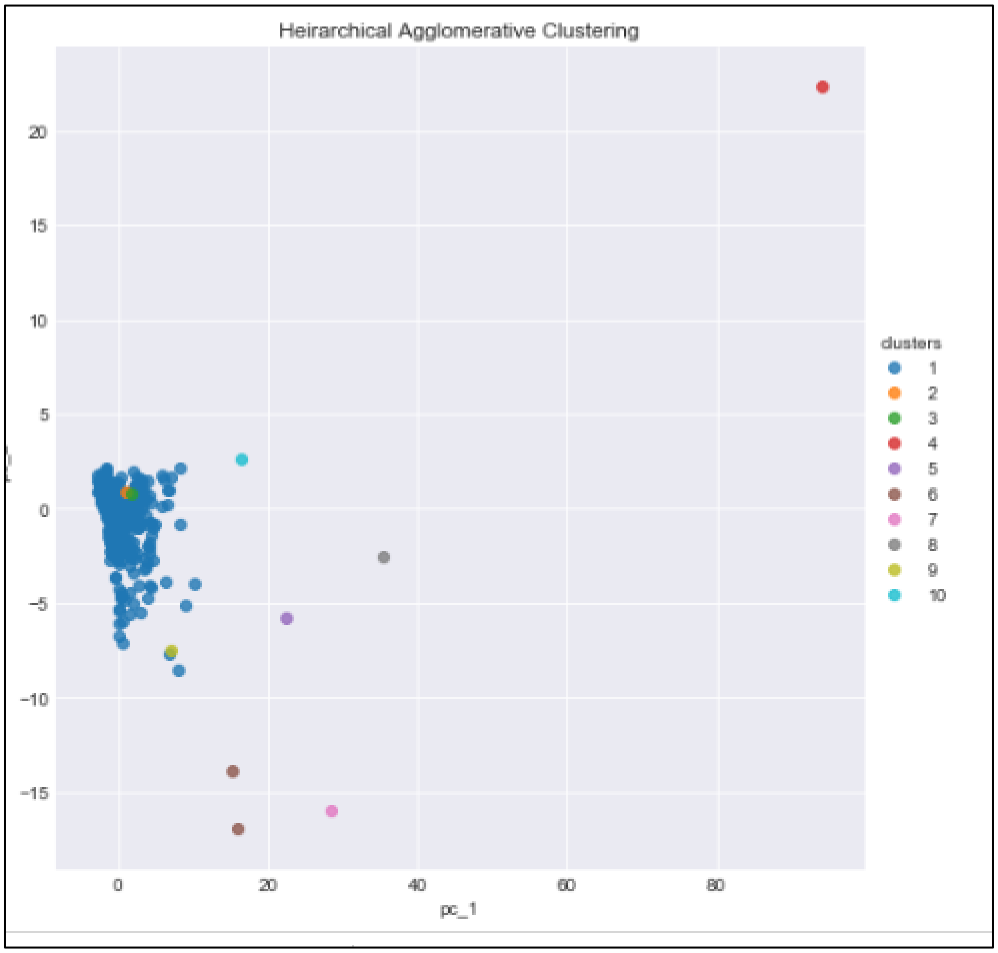
Results:

<u>Input:</u>
Data = iyer.txt
K = 10

<u>Output:</u>
The Jaccard value: 0.1582
The Rand Index value: 0. 1883

Visualization

The PCA below is HAC algorithm on iyer.txt with cluster size = 10



<u>Fig 4: HAC iyer.txt</u>

Advantages of HAC:

1. Hierarchical clustering do not have to assume any particular number of clusters. Any desired number of clusters can be obtained by cutting the dendogram at the proper level.
2. Hierarchical clustering outputs a hierarchy structure of points. The data 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 dendogram.
3. Meaningful taxonomies can be produced.

Disadvantages of HAC:

1. Once a decision is made to combine two clusters it cannot be undone
2. No objective function is directly minimized
3. Different schemes have problems with one of the following: sensitivity to outliers, breaking large clusters, difficulty handling different sized clusters and irregular shapes
4. Runtime is O (N^3), hence not suitable for large datasets
5. Sensitive to outliers

Density Based DBSCAN

DBSCAN is a density based clustering algorithm. Here clusters are dense regions in the data space separated by regions of lower object density. We use 2 parameters epsilon which is the distance between and Minpts which is the minimum number of points that has be present in the epsilon radius There are three kind if points

1. Core points: A point is core point if it has more than a specified number of points within eps
2. Border point: A border point has fewer than Minpts within eps but is in the neighborhood of a core point
3. A noise is any point that is not a core point nor border point

The implementation is based on the pseudo code provided in the clustering slides

Algorithm:

```
DBSCAN(D, eps, MinPts)
        C = 0
        for each unvisited point P in dataset D
        mark P as visited
        NeighborPts = regionQuery(P, eps)
        if sizeof(NeighborPts) < MinPts
        mark P as NOISE
        else
        C = next cluster

expandCluster(P, NeighborPts, C, eps, MinPts)
        expandCluster(P, NeighborPts, C, eps, MinPts)
        add P to cluster C
        for each point P' in NeighborPts
        if P' is not visited
        mark P' as visited
        NeighborPts' = regionQuery(P', eps)
        if sizeof(NeighborPts') >= MinPts
        NeighborPts = NeighborPts joined with NeighborPts'
I       f P' is not yet member of any cluster
        add P' to cluster C

regionQuery(P, eps)
return all points within P's eps-neighborhood (including P)
```

Results:

Input:
Data = cho.txt
epsilon = 1.1
Minpts = 3

Output**:**
The Jaccard value: 0.2037
The Rand Index value: 0. 5665

Visualization

The PCA below is DBSCAN algorithm on cho.txt

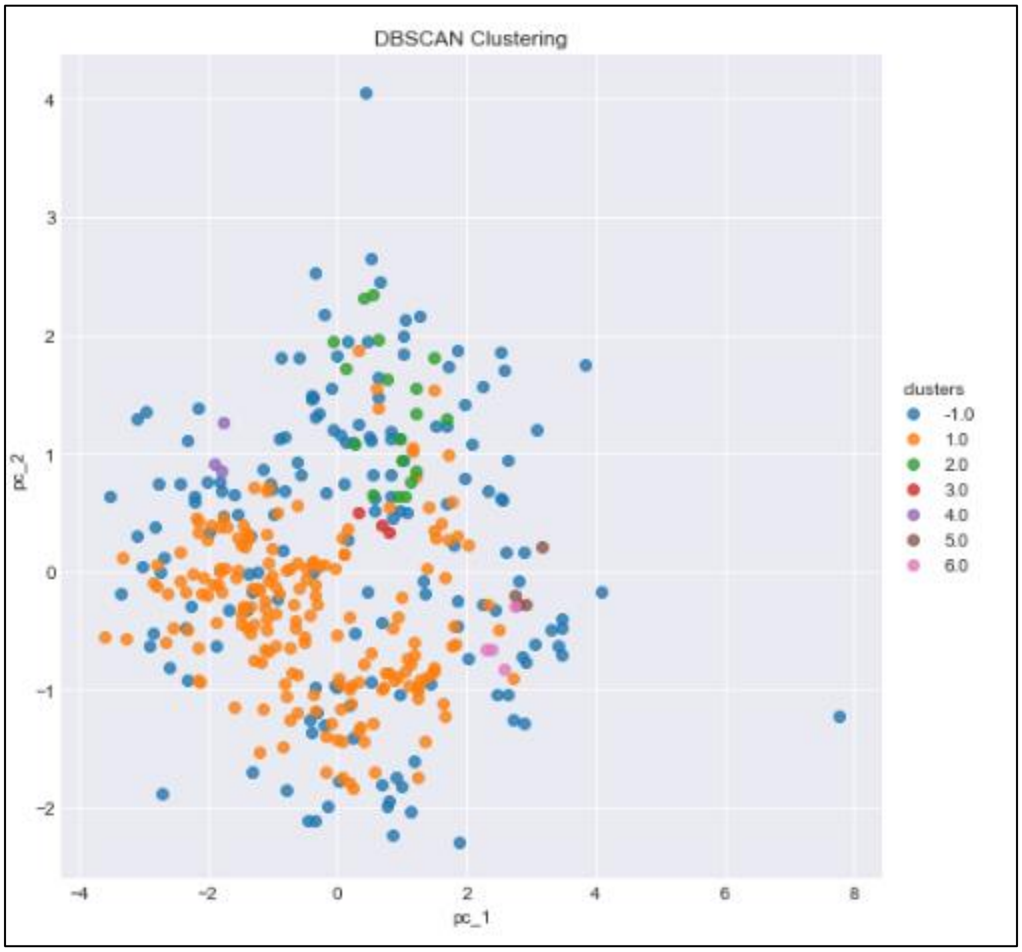


Fig 4: DBSCAN cho.txt

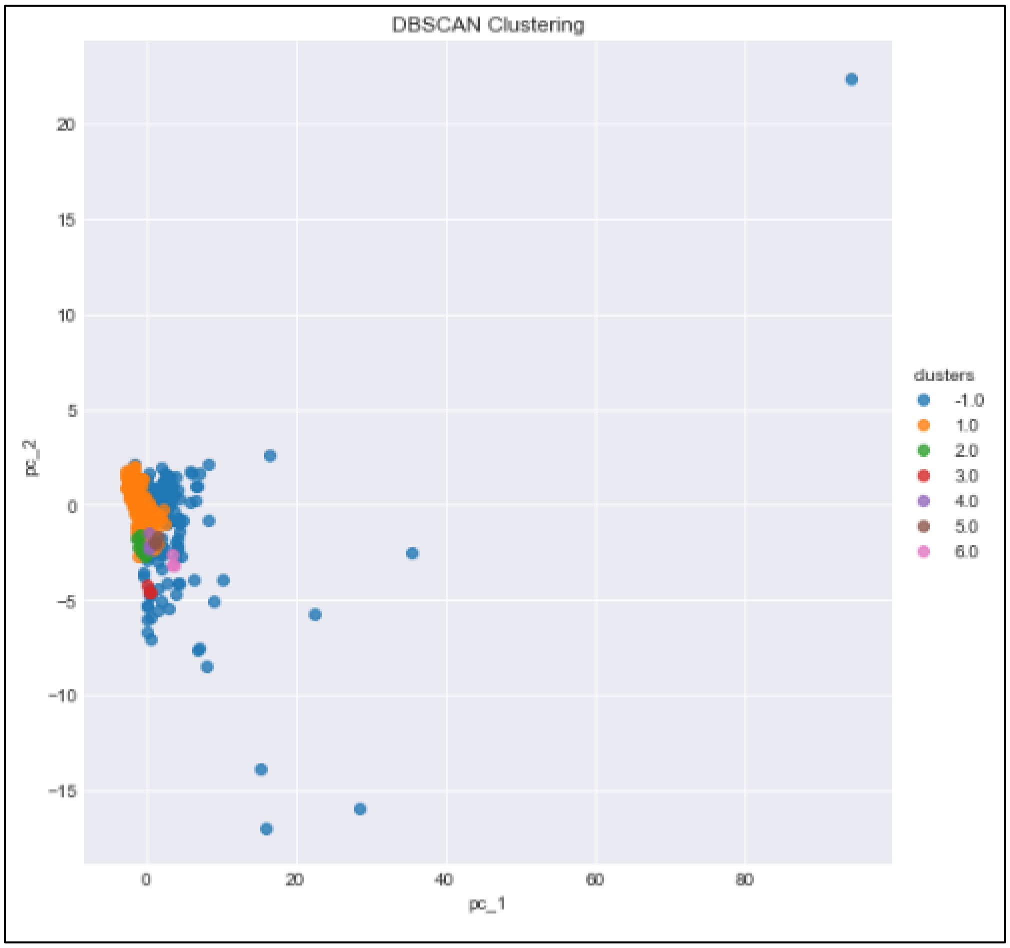
Results:

<u>Input:</u>
Data = iyer.txt
epsilon = 1.1
Minpts = 3

<u>Output**:**</u>
The Jaccard value: 0.2385
The Rand Index value: 0.5625

Visualization

The PCA below is DBSCAN algorithm on iyer.txt



Advantages of DBSCAN:

1. DBSCAN can handle clusters of different shapes and sizes
2. DBSCAN is resistant to noise
3. DBSCAN does not require one to specify the number of clusters in the data
4. DBSCAN time complexity is $O(n^2)$ and space complexity is $O(n)$
5. DBSCAN can find arbitrarily shaped clusters

Disadvantages of DBSCAN:

1. DBSCAN cannot handle varying densities
2. It is sensitive to parameters, hard to determine the correct set of parameters
3. It is highly sensitive to parameters
4. The quality of the DBSCAN based on the Euclidean distance however, for high dimension data, this is rendered useless due to curse of dimensionality

MAP reduce K means clustering

Kmeansbash :

1. Select initial centroids randomly using randint
2. Create input and output folder
3. Run in a while loop and run the Hadoop mapper and reducer using the dataset
4. Now we get a new centroid from the Reducer.
5. We compare the new centroid with the old centroid. If they are equal then we break because there are no new clusters obtained
6. If they are not equal then we copy the new centroid to centroid and then run the mapper and reducer again
7. Now when they are equal we break.
8. After break the output is wrote on the output file
9. Here we preprocess the output file and then map the clusters to the points
10. We use this to find the Jaccard similarity and rand index
11. Then we use PCA to plot the principal components

Mapper:

1.Now the input data is split and assigned to different mappers
2. The initial centroids created is present in the as centroid file which is being read by the mapper
3. Now for every data it checks which the closest centroid using scipy Euclidean distance
4. For every data its emits cluster id, gene id and gene attributes

Reducer:

1. Now the reduced gets the output from the mapper, which is sorted by cluster
2. We add the features until the cluster Id does not change
3. This added features is used later to find the mean of the cluster points which is the new centroid
4. We do the same for all the clusters and find the new centroids
5. Finally we write the new centroid to file which is later used to compare

Results:

Input:
Data = cho.txt
K = 5
Initial centroid (randomly chosen using randint) = [11, 63, 326, 337, 351]

Output**:**
The Jaccard value: 0.4254
The Rand Index value: 0. 8076

Visualization

The PCA below is K-Means algorithm on cho.txt with cluster size = 5

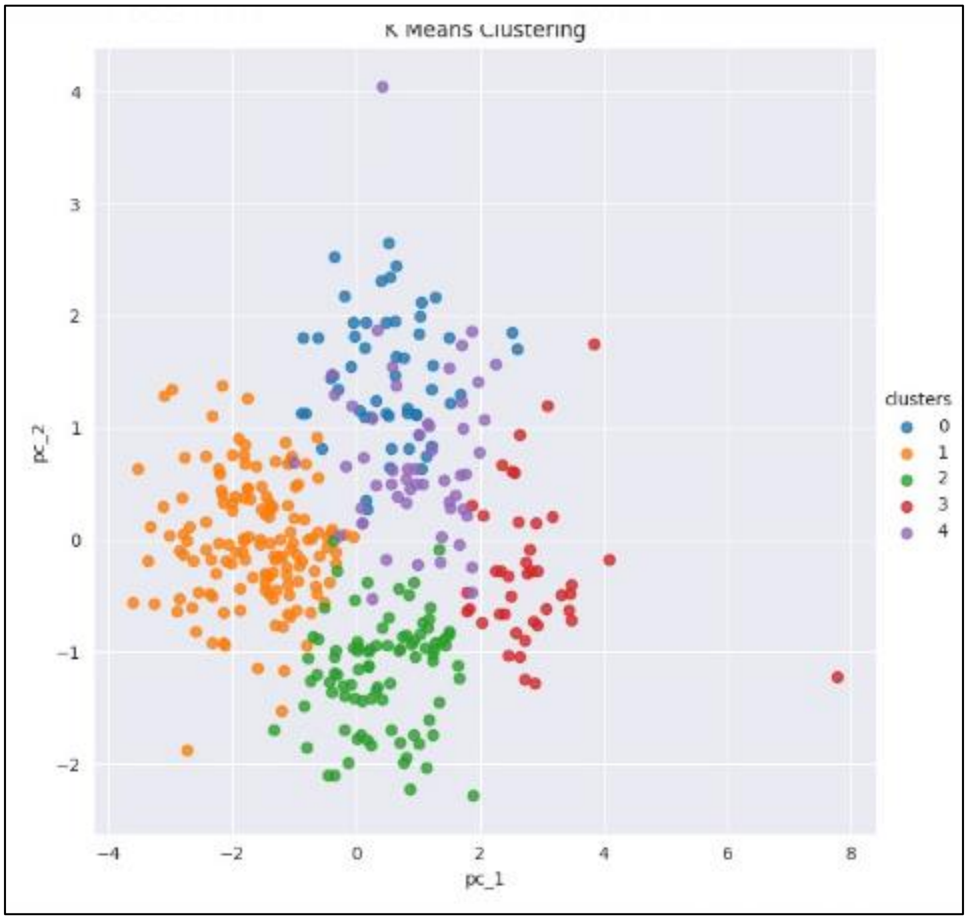


Fig 7: Map reduce Kmeans Cho.txt

Results:

Input:
Data = iyer.txt
K = 11
Initial centroid = [5, 40, 49, 75, 110, 136, 316, 337, 393, 477, 506]

Output:
The Jaccard value: 0.3366
The Rand Index value : 0.8227

Visualization

The PCA below is K-Means algorithm on iyer.txt with cluster size = 1

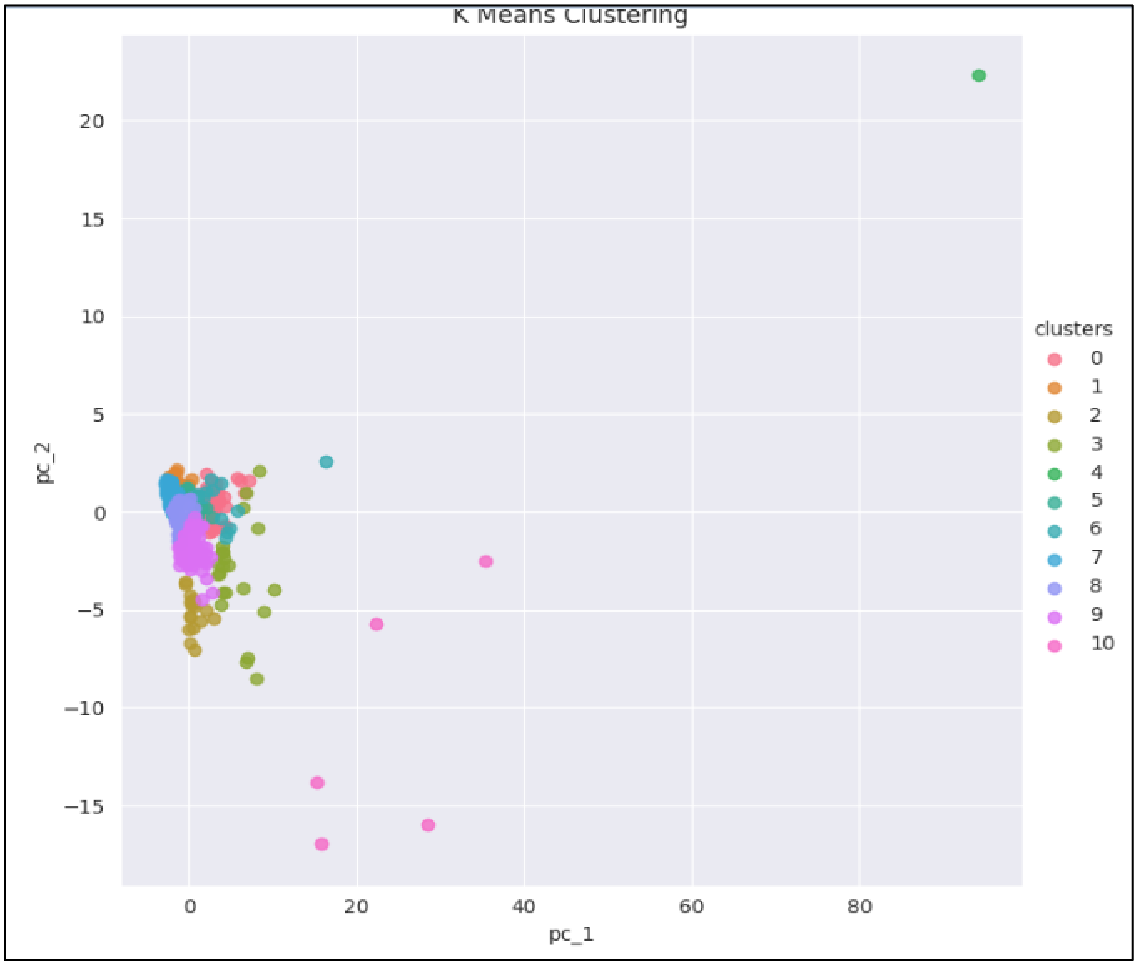


Fig 8: Map reduce Kmeans iyer.txt

Advantages of Map reduce Kmeans

1. Map reduce implementation of Kmeans can be used for large datasets giving us time efficient results
2. The time complexity of Mapreduce Kmeans is O(log k^2) compared to O(tkn) of Kmeans
3. Parallel implementation of Kmeans provides all advantages provided by standard Kmeans implementation

Disadvantages of Map reduce Kmeans

1. For large dataset computing K is difficult
2. For smaller datasets mapreduce takes more time than the standard Kmeans execution
3. It has all the disadvantages of the standard Kmeans implementation

Conculsion

1. Kmeans did the best job for clustering the data for both the datasets as it has the highest jaccard coefficient value and also the PCA plot also shows the same
2. Since we don't have to specify the K value hierarchical clustering is good to provide initial summary of the datasets
3. DBSCAN can help in identifying noise and outliers in the data
4. K means and HAC cannot identify outliers in the data
5. Map reduce Kmeans is faster due to parallel execution and calculation of centroids
6. K means was the easiest to implement
7. For HAC we need to maintain a distance matrix which takes some space and we need to update it in every iteration
8. Parallel and standard Kmeans provide same results