Project 2

Clustering Algorithms

Team

Group 15

Prasanna Pai – 50132731

Jesal Janani – 50132391

Chandana Satya Prakash – 50134196

K- Means algorithm

Algorithm:

Genes[][] contains a set of data points.

k is the number of clusters to be formed.

1. The first step is to pick k initial clusters centroids. Therefore k data points are randomly selected to be the initial cluster centroids.
2. The minimum distance is taken to be infinity for the first iteration as the data points have not been assigned to a cluster. For the remaining iterations, the distance between the data point and the cluster it belongs to is computed.
3. For every data point, the distance from the data point to all other clusters is computed. These distances are compared with the minimum distance computed in the previous step and the new minimum is determined. The data point is reassigned to the cluster with the minimum distance.
4. After all the data points have been assigned/reassigned to clusters, the new cluster centroids are computed.
5. Step 2-4 is repeated until the algorithm converges that is when no data point is reassigned or the number of iterations mentioned are completed.

K-means clustering is a fairly efficient clustering algorithm. It is also very robust and easy to understand. Because of the above mentioned advantages, it is a commonly used algorithm for clustering. K-means algorithm is also fat. It gives the best results when the clusters are distinct and away from each other.

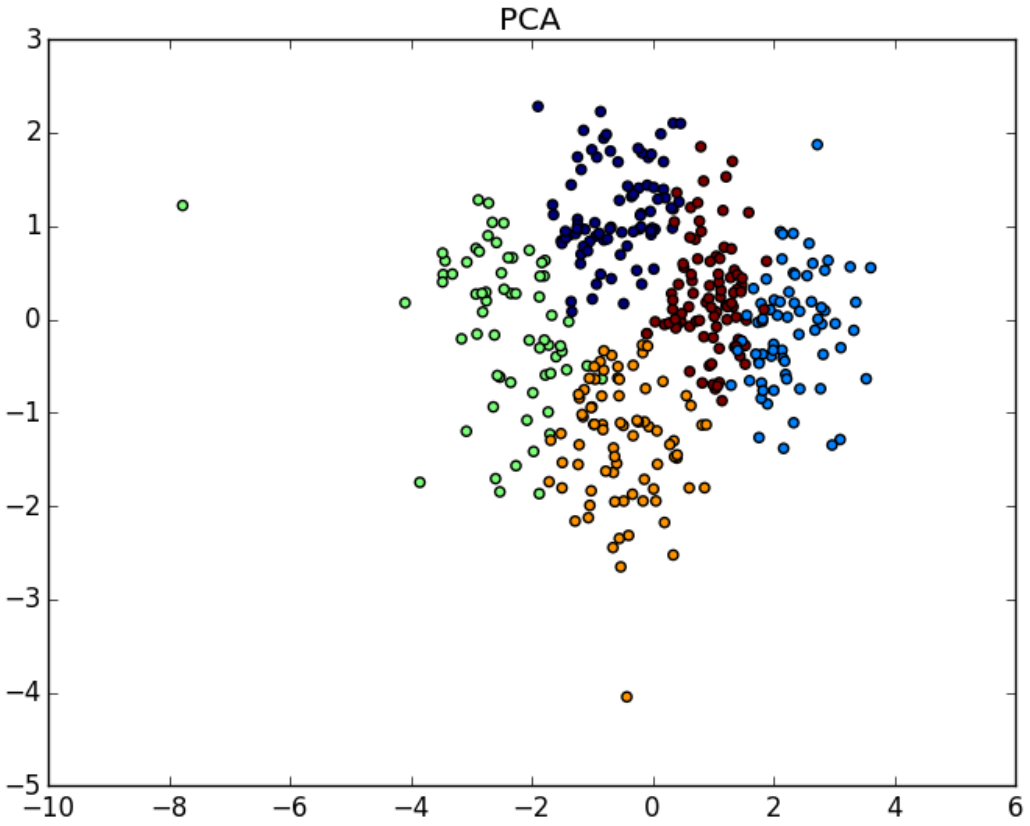
Despite the many advantages it has, there are some disadvantages which makes k- means not the best algorithm there is. K-means requires the knowledge the number of clusters k, apriori. It also does not work well with overlapping clusters, non-linear data set and outliers. Also, randomly choosing the initial cluster centroid may not always result in the best possible results.

Experiment values:

cho.txt for K Means algorithm

Value of K	External Index (Jaccard Coefficient)	Internal Index (Correlation)	Explained Variance Ratio

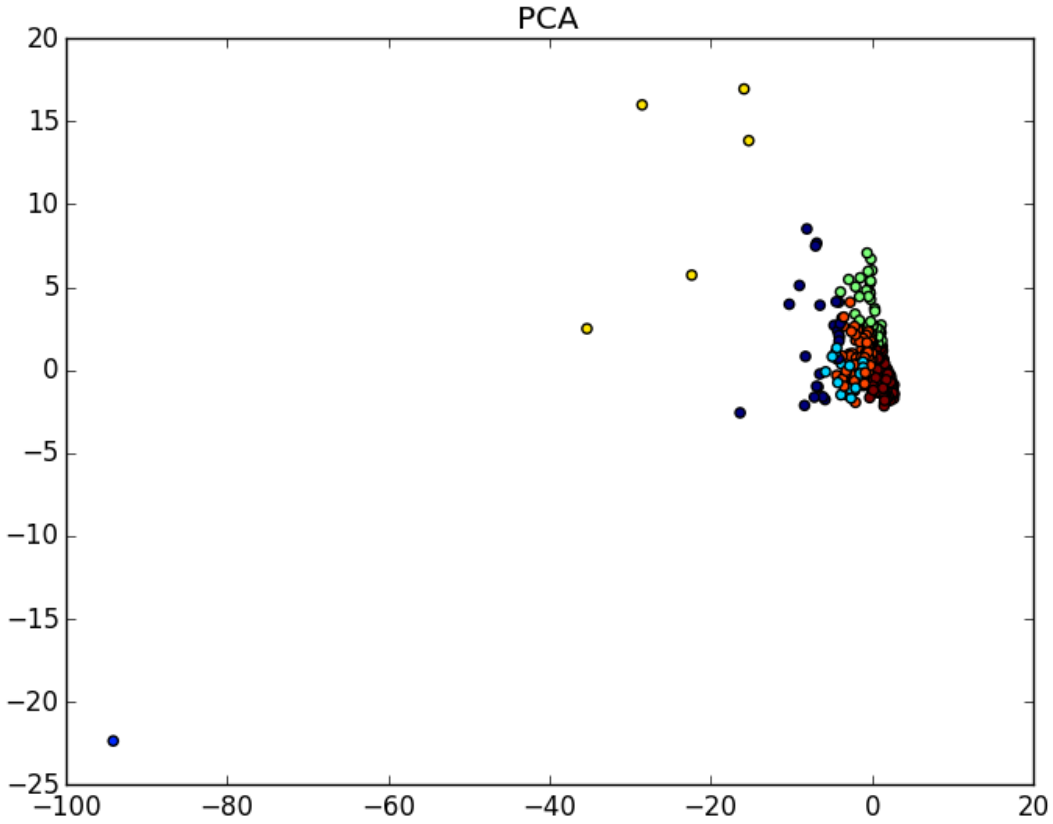
5	0.3409333661983403	-0.43508182465769757	[0.48084471 0.18533784]
7	0.30746609313962164	-0.4360433619668424	[0.48084471 0.18533784]
10	0.2202162625637117	-0.36744826744105324	[0.48084471 0.18533784]
15	0.1916131007040098	-0.33052339169425315	[0.48084471 0.18533784]



iyer.txt for K Means algorithm

Value of K	External Index (Jaccard Coefficient)	Internal Index (Correlation)	Explained Variance Ratio
5	0.2795860407714154	-0.38701347847810025	[0.72268258 0.13481278]
7	0.2947157382213948	-0.396020995011462	[0.72268258 0.13481278]
10	0.31165596919127087	-0.255401469784445	[0.72268258 0.13481278]

15	0.3178341706934622	-0.20710315906491272	[0.72268258 0.13481278]



Hierarchical Agglomerative Clustering with Single Link:

Single linkage clustering is based on grouping clusters in bottom-up fashion where each step involves combining two clusters containing the closest pair of elements not yet belonging to the same cluster. The distance between the two clusters is determined by a single element pair which are closest to each other. These two element pairs are then fused together into a single cluster in the form of nearest neighbor clustering. The resultant of this clustering has been visualized in the form of dendogram which shows the sequence of the cluster formation.

Algorithm:

1. Closest pair of clusters (i, j) is the one with the smallest distance value
2. Defined an array minDist for keeping the track of the cluster labels of the points. The cluster formed is replaced in the array by the min of row i and j.
3. The existing row j and column j is eliminated from the plot table.
4. The distance of cluster formed is updated to its minimum distance.
5. Iterate until all the points are clustered.

Pros and Cons of the Algorithm:

Pros:

1. No apriori information about the number of clusters required.
2. Easy to implement and gives best results in some cases

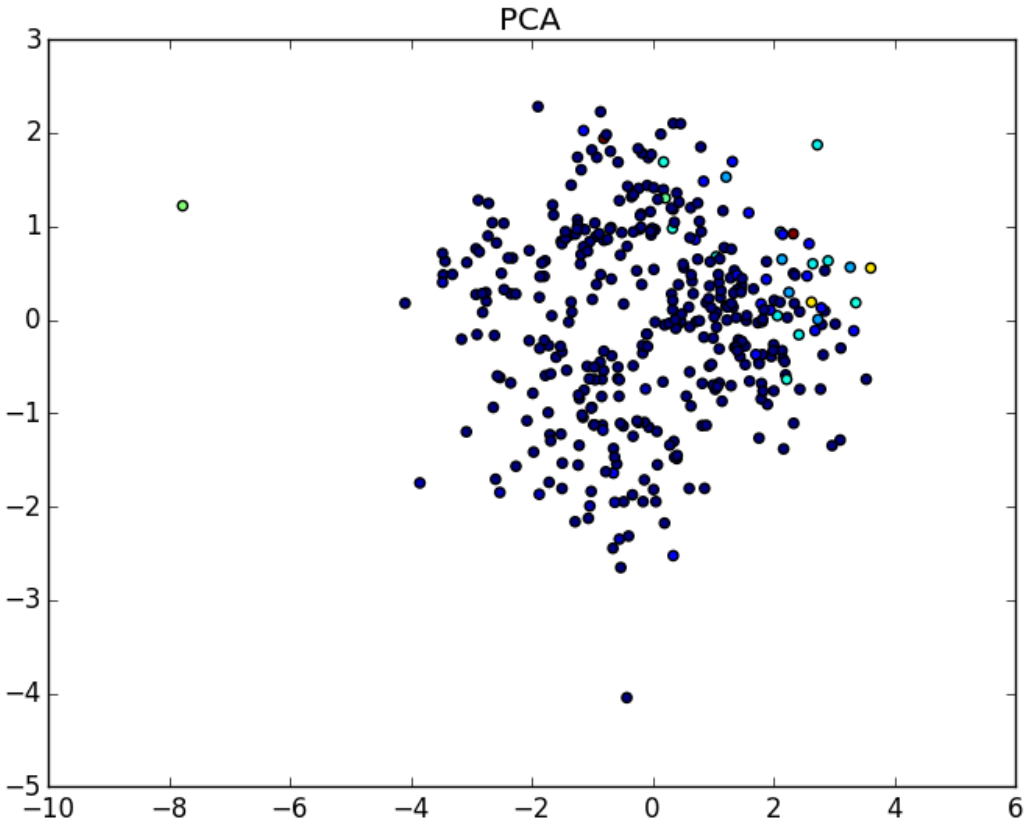
Cons:

1. Time Complexity of O(n^2 logn) where n is the number of data points
2. Sensitivity to noise and outliers
3. Difficulty in handling different size clusters and outliers
4. No objective function is directly minimized.
5. Difficult to identify the correct number of clusters in the denser dendogram

Experiment Values:

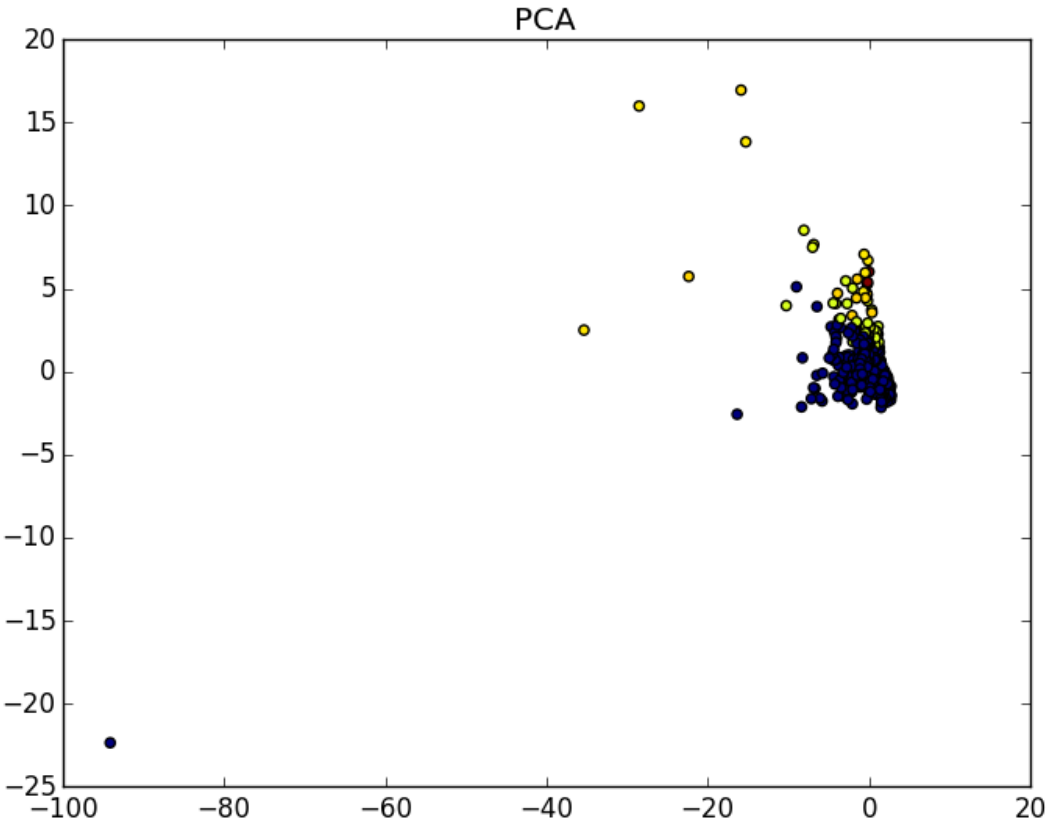
cho.txt for hierarchical algorithm

Value of K	External Index (Jaccard Coefficient)	Internal Index (Correlation)	Explained Variance Ratio
5	0.2303547188438387	-0.35357980410205475	[0.48084471 , 0.18533784]
7	0.23092704192845237	-0.4331469762008344	[0.48084471, 0.18533784]
10	0.2084031426387104	-0.5242359457775692	[0.48084471 , 0.18533784]
15	0.20496355147841755	-0.5128553374732759	[0.48084471, 0.18533784]

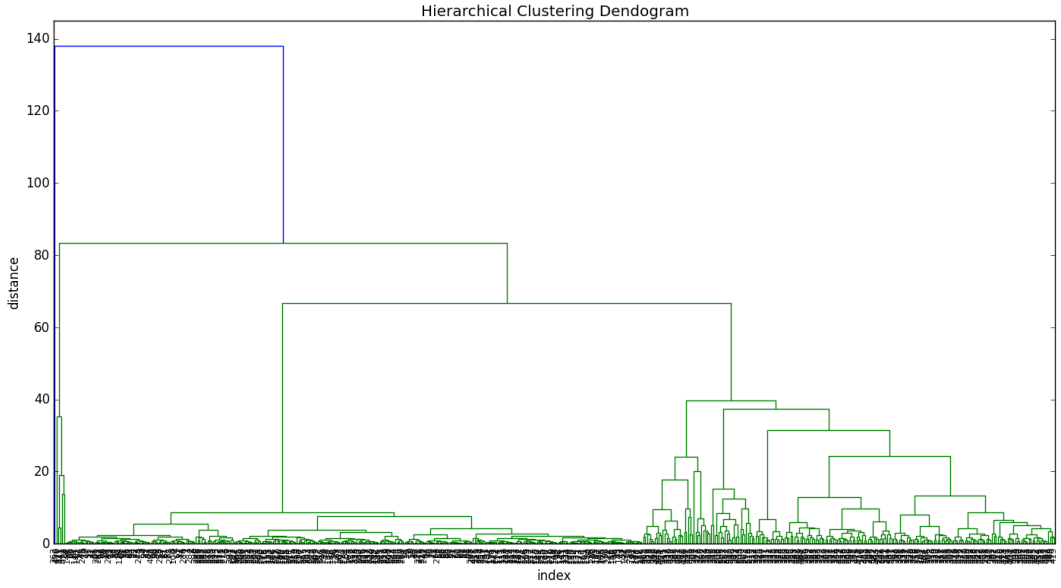


iyer.txt for hierarchical algorithm

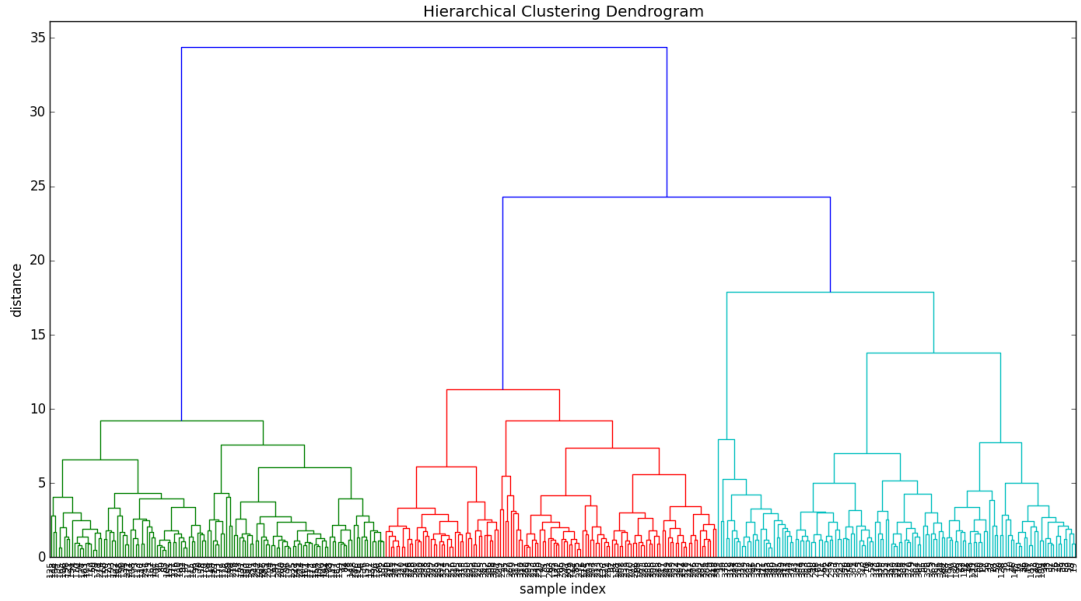
Value of K	External Index (Jaccard Coefficient)	Internal Index (Correlation)	Explained Variance Ratio
5	0.16184796677598406	-0.7285499088655997	[0.72268258 0.13481278]
7	0.1621419120343063	-0.7515205657012287	[0.72268258 0.13481278]
10	0.1635585720896625	-0.776557759047836	[0.72268258 0.13481278]
15	0.17648304694550343	-0.7801409760263412	[0.72268258 0.13481278]



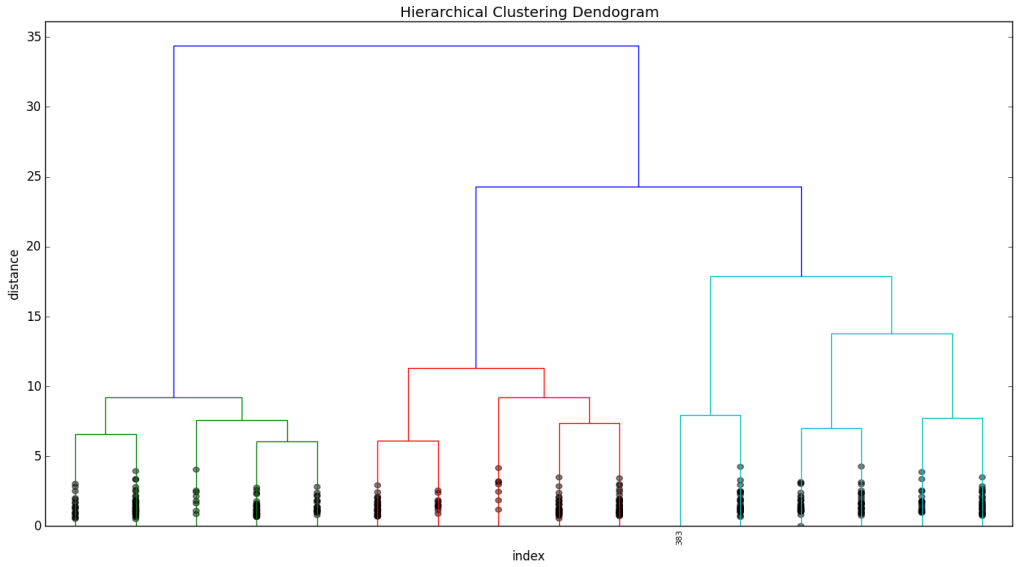
Dendogram for iyer.txt



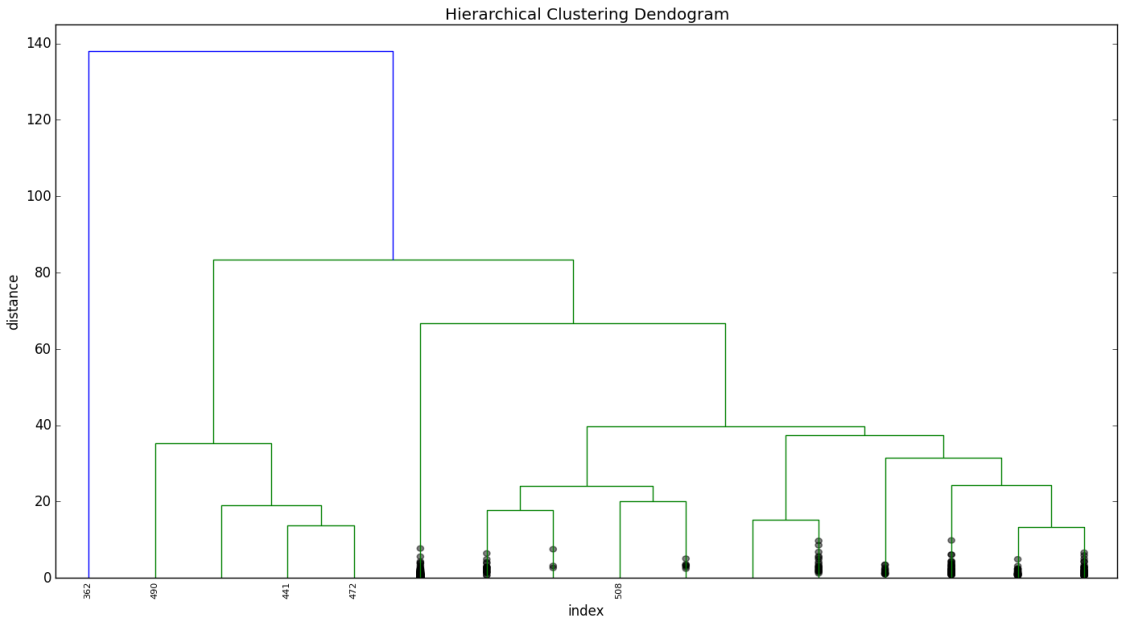
Dendogram for cho.txt



Dendogram for cho.txt (truncated to 16 leaf counts)



Dendogram for iyer.txt (truncated to 16 leaf counts)



Density Based SCAN (DBSCAN)

DBScan Description and Implementation:

The basic idea behind the DBScan algorithm is that clusters are regions of high object density separated by regions of low object density. The main parameters for the algorithm are ε and minPts:

- ε: radius for the neighborhood of a point
- minPts: minimum number of points in the given ε-neighborhood

Together, these 2 parameters define the density threshold that determine the cluster formations. Data points are classified based on these parameters as follows:

- core points: points that have more than minPts points in their ε-neighborhood
- border points: points that have fewer than minPts points in their ε-neighborhood but lie in the ε-neighborhood of a core point
- noise points: points that are neither core nor border points

A point y is directly density-reachable from a point x if x is a core point and y is in x's ε-neighborhood. A point z is indirectly density-reachable from a point x if z is directly density-reachable from a point y which in turn, is directly density-reachable from point x, and so on. The DBScan algorithm implemented in our project is as follows:

- All the data points are initially marked as not visited
- For each unvisited data point p in dataset D:
    - p is marked as visited
    - if p has fewer than minPts points in its ε-neighborhood, then it is marked as a noise point
    - else, p is a core point and a new cluster is initialized
    - all the density-reachable points from p are collected into this new cluster
    - this is done by adding the neighbor points into the cluster and then looking for other density reachable points from any core points present in the neighbor points
    - this process is continued until there are no more core points left to process for that cluster

DBScan Pros and Cons:

Since DBScan identifies clusters based on the point densities and not shapes or distances, it can handle clusters of varying shapes and sizes. It is resistant to noise points so that they do not affect the identification of clusters in the way that k-Means does.

However, the performance of DBScan is highly sensitive to the ε and minPts parameters. Changing the parameters has a significant impact on the outcome of the DBScan algorithm. DBScan is also not resistant to datasets with varying density regions. The parameters set at the start determine the density threshold and potential clusters with varying density regions are not taken into account for the purposes of this algorithm. Since the performance of DBScan depends so much on the parameters, it is difficult to determine the ideal/optimum parameters for a given dataset that can give the desirable cluster output.

Experiment Values

Epsilon	MinPts	# of Clusters	External Index Jaccard Coefficient)	Rand Coefficient
1.0	7	3	0.20310643965300026	0.4056618969636769
1.0	23	0	0.2300732905581358	0.2300732905581358
1.0	39	0	0.2300732905581358	0.2300732905581358
1.0	55	0	0.2300732905581358	0.2300732905581358
1.2	23	1	0.21624319774497905	0.46254933018336064
1.2	39	1	0.2038928310895948	0.3554457837794303
1.2	55	0	0.2300732905581358	0.2300732905581358
1.4	7	1	0.2234849142044173	0.39923219415286315
1.4	23	1	0.2590132611514228	0.5627265161480846
1.4	39	1	0.24103303086934005	0.5475314773550968
1.4	55	1	0.2255125284738041	0.4934092190394373
1.6	7	1	0.22443424838634418	0.33065317189723215
1.6	23	1	0.22748157207549666	0.37257376036940587
1.6	39	1	0.2513122064801294	0.490697736851996
1.6	55	1	0.27874274348359046	0.5813981583398212

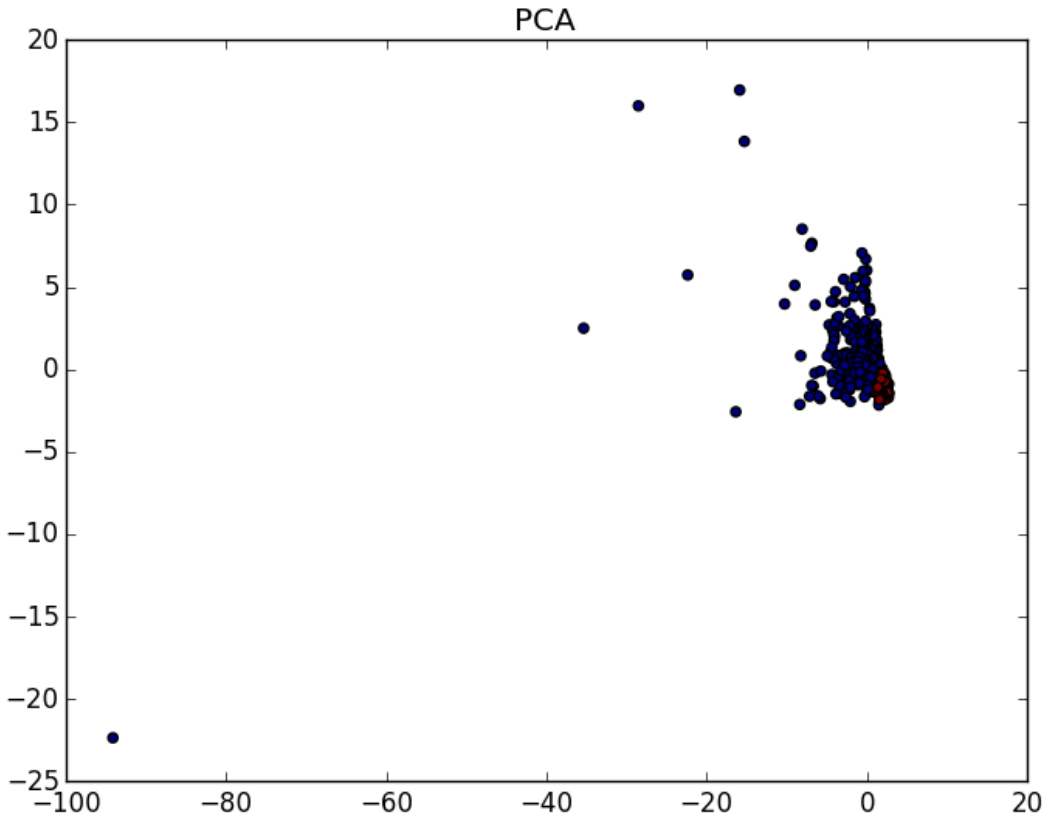
1.8	7	1	0.22504661191841846	0.2691213186931193
1.8	23	1	0.22459753822121295	0.28039678917554833
1.8	39	1	0.2274245939675174	0.3295524712072807
1.8	55	1	0.24228541337546078	0.42056162581545814
2.0	7	1	0.22505469687345614	0.2630540417192408
2.0	23	1	0.22504661191841846	0.2691213186931193
2.0	39	1	0.22546978854132274	0.2757792155494107
2.0	55	1	0.22520437164874446	0.2901017476979248

Max Jaccard = 0.27874274348359046, eps: 1.6, minPts: 55

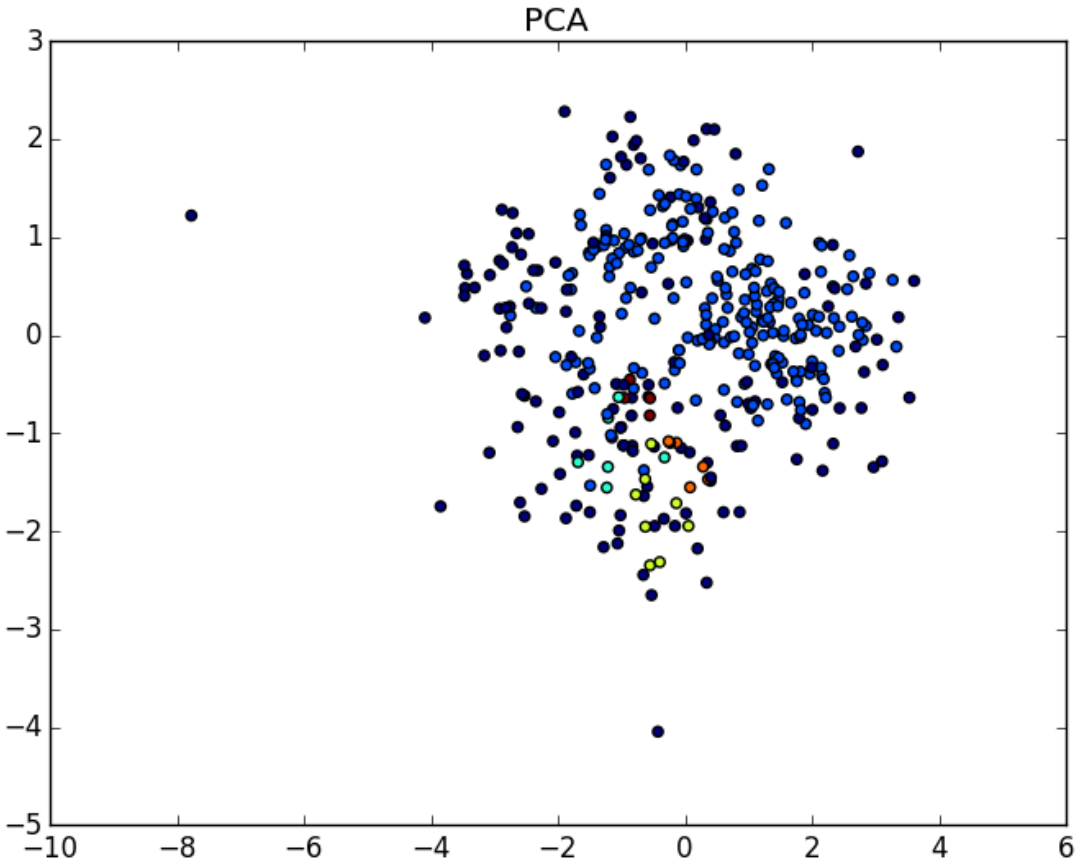
Max Rand = 0.5813981583398212, eps: 1.6, minPts: 55

Max Clusters = 5, eps: 1.2, minPts: 7

PCA for dataset iyer.txt using DBSCAN -



PCA for dataset cho.txt using DBSCAN -



Comparative Analysis :

Usually, to assess cluster stability, we compare several algorithms which basically "share" some similarity (e.g. k-means and hierarchical clustering, because euclidean distance work for both). For assessing the concordance between two cluster solutions, there is a debate about “where to cut a dendrogram?” Hence according to the given datasets, K Means seems to be ta better fit since we are calculating the centroids of the clusters and the distance between the clusters and the data points are recalculated for clustering which gives it an upper edge over Hierarchical and DBSCAN.