# CSE 601 – Data Mining & Bio Informatics

## PROJECT II - CLUSTERING ALGORITHMS

LAKSHMI VENNELA VEERISETTY

SWAPNIKA PEMMASANI

ASISH KAKUMA

## 1. K-Means Clustering Algorithm

### Introduction:

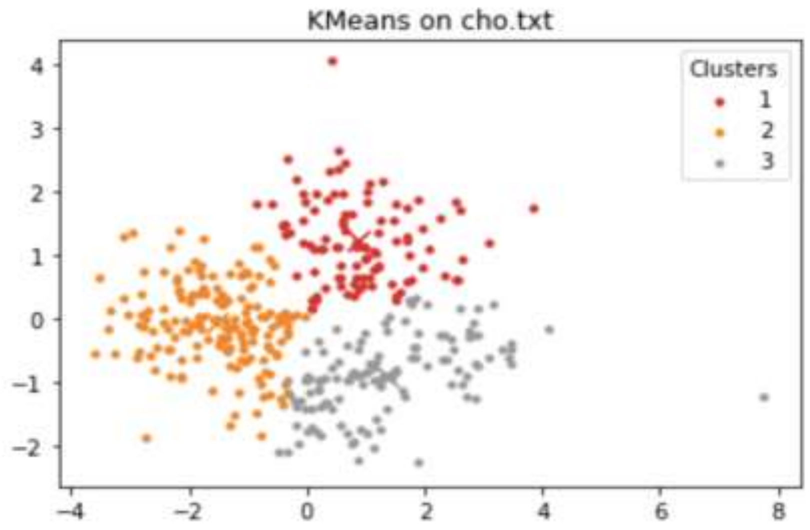
K-Means is an unsupervised learning algorithm which makes inferences from the dataset using the input vectors without referring to labels or outcomes. The objective of the algorithm is to group similar data points together and discover underlying patterns.

1. To achieve this, we initially choose a random **K** value viz., the number of clusters we need the total data points to be segregated.
2. We now initialize random centroids and calculate the distance between all the data points and the centroids.
3. We assign each data point to the nearest centroid, which forms k number of clusters.
4. Now, we calculate mean of all the datapoints within a cluster and change the centroid of each cluster for each iteration
5. We halt the algorithm only when either of two conditions are met viz.,
    1. When we reach convergence i.e. objective values don't change anymore.
    2. When it reaches the maximum no. of. Iterations

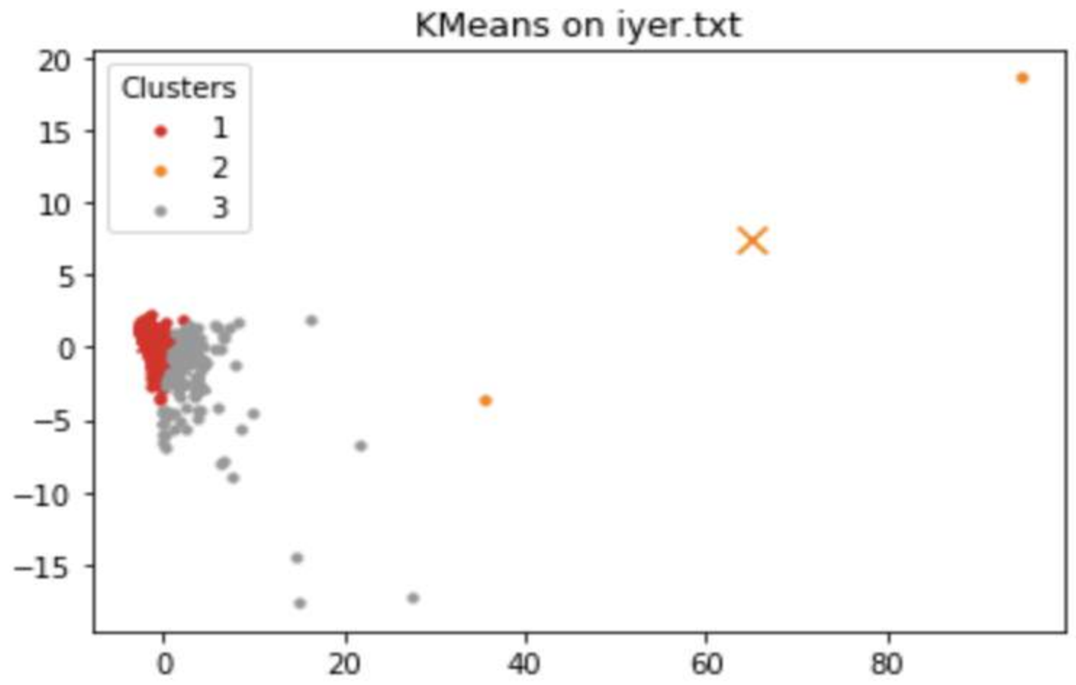
### Implementation

1. Remove the first two columns of the dataset as they are gene ID and ground truth respectively.
2. We choose the number of centroids, **K** value or choose indices as centroids and make these row indices as our initial centroids.
3. Additionally, we choose the max number of iterations.
4. Euclidian distance is calculated from the initial centroids to all the data points.
5. Now, the centroids are updated by taking the mean of all the data points within each cluster respectively.
6. We halt the algorithm until it met the mentioned conditions above.

## Visualization



**Dataset**: Cho.txt
**No. of. Clusters (K):** 3
**No. of. Iterations (I)**: 30
Converged after **11** iterations



**Dataset**: iyer.txt
**No. of. Clusters (K):** 3
**No. of. Iterations (I)**: 50
Converged after **26** iterations

## Validation

We used both Jaccard Co-efficient and Rand Index to validate the results of K-Means Algorithm and cross check using the ground truth using the below formulae.

**Jaccard Coefficient**

$$\frac{|M_{11}|}{|M_{11}|+|M_{10}|+|M_{01}|}$$

**Rand Index**

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

**Validation for Cho.txt**
Jaccard Coefficient: 0.4088574886337618
Rand Index: 0.7574028832988805

**Validation for iyer.txt**
Jaccard Coefficient: 0.2199408701534227
Rand Index: 0.4995304707638549

## Advantages

1. K-Means is a clustering algorithm with the complexity of O(ikn) where i is the number of iterations, k is the number of clusters and n is the number of objects in the dataset.
2. K-Means is computationally faster than hierarchical clustering.

## Disadvantages

1. We need to specify number of clusters beforehand.
2. Unable to handle noisy data and outliers

## 2.Hierarchical Agglomerative Clustering with Min Approach

### Introduction:

Agglomerative clustering works in bottom-up manner. Each object is initially considered as an individual cluster.

Computes the similarity between each of clusters and the two most similar clusters are merged together. The similarity is distance between clusters; minimum distance means more similarity.

In single linkage hierarchical clustering, the distance between two clusters is defined as the shortest distance between two points in each cluster and it is used as similarity measure.

### Implementation of Algorithm:

1. Consider the given input or file and prepare the dataset by removing the un-necessary columns.
2. Initialize the value of K, where K determines the number of clusters to be formed and also tells when algorithm should terminate.
3. Calculate distance matrix of size N*N, where N represents the number of objects or items in the dataset. Each index i, j in the matrix represents distance between $i^{th}$ and $j^{th}$ record.
4. Select two records with minimum distance and merge those 2 records. It reduces the matrix size by 1.
5. The distance matrix is recomputed for remaining points using minimum distance between two of the merged points. For example, if points 2 and 5 are merged in this step. Distance of all other points will be calculated w.r.t to cluster (2,5) using, d((2,5), P) = min(d(2,P),d(5,P)) .
6. Repeat steps 4,5 until the size of distance matrix equals K.

**Jaccard Coefficient**

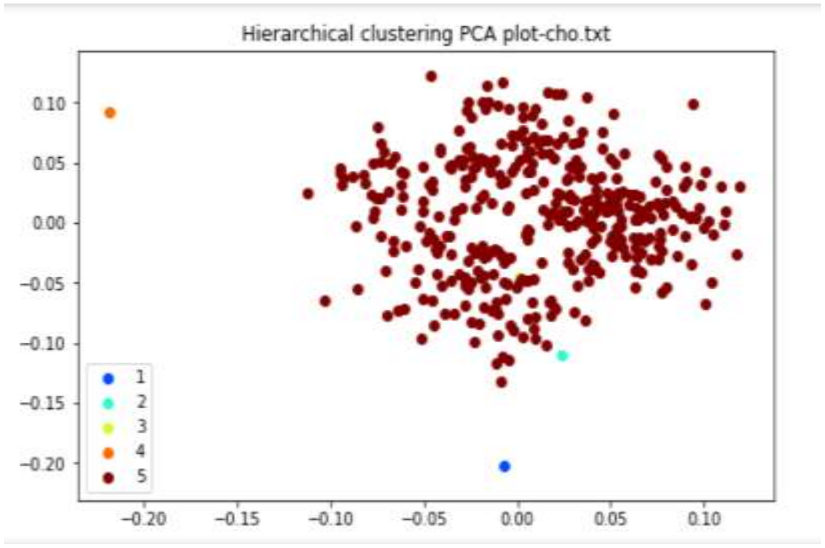
$$\frac{|M_{11}|}{|M_{11}|+|M_{10}|+|M_{01}|}$$

**Rand Index**

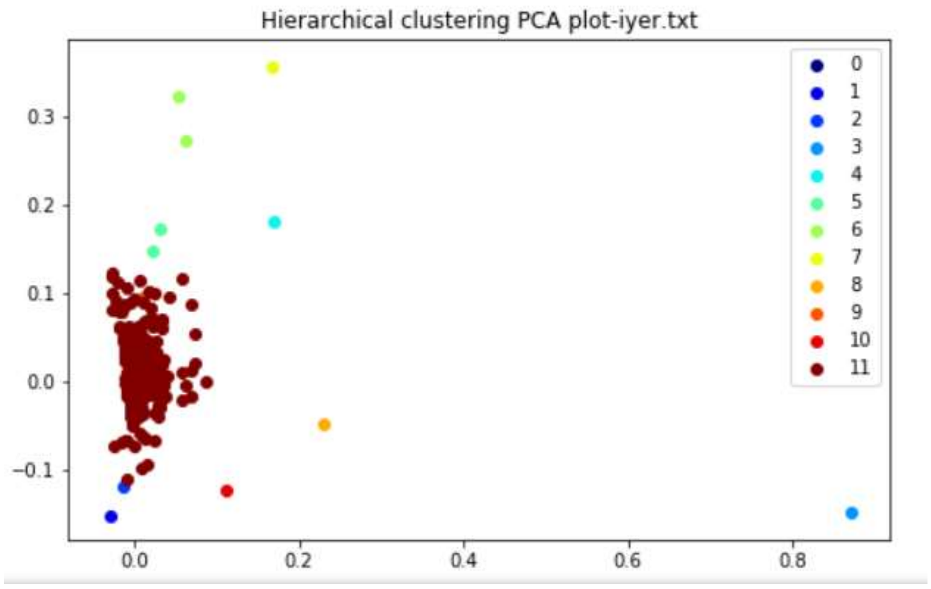
$$Rand = \frac{|Agree|}{|Agree|+|Disagree|} = \frac{|M_{11}|+|M_{00}|}{|M_{11}|+|M_{00}|+|M_{10}|+|M_{01}|}$$

## Visualizations:

1. HCA Plot for **"cho.txt"**
   For K=3,
   **Jaccard Coefficient**: 0.2283,
   **Rand Index**:0.2402



2. HCA Plot for "iyer.txt"
   For K=3**,**
   **Jaccard Coefficient**: 0.1584**, Rand Index**:0.1941



### Advantages of Hierarchical Clustering:

1. Any number of clusters can be produced by cutting the dendrogram at particular level.
2. Hierarchical clustering corresponds to meaningful taxonomies.
3. It is shown to capture concentric clusters.
4. It provides hierarchical relations between clusters.

### Disadvantages of Hierarchical Clustering:

1. Computationally expensive and sensitive to outliers.
2. Decision once made cannot be undone.
3. No objective function is directly minimized.
4. Difficulty handling different sized clusters and irregular shapes.

## 3.DENSITY - BASED SPATIAL CLUSTERING (DBSCAN)

### Introduction:

DBSCAN is a density-based algorithm. It locates regions of high density that are separated from other regions of low density, where density = number of points within a specified radius.
Points in dataset can be classified into three types:
**Core points**: Points having more than minimum number of points in its neighborhood. These points are present at the interior of a cluster. Any core points that are close enough – within a distance of given radius – are present in the same cluster.
**Border points**: Points having less than minimum number of points in its neighborhood, but still present in neighborhood of other core points. Any border point that is close enough to a core point is present in the same cluster as the core point.
**Noise**: Any point that is not a core or border point and also having less than minimum number of points in its neighborhood. Noise are not present in any cluster.
• **Density-reachable**: An object q is directly density-reachable from object p if q is within the ε neighborhood of p and p is a core object.
• **Density-connectivity**: An object p is density-connected to object q with respect to ε and minimum number of points if there is an object o such that both p and q are density-reachable from o with respect to ε and minimum number of points.
**DBSCAN Algorithm and Implementation**:
DBSCAN algorithm has 2 parameters
**ɛ-Neighborhood**: object within ɛ from an object.
**Minpts**: minimum number of points to be present in the ɛ-Neighborhood.

### Algorithm Steps:

1.Prepare the given dataset by removing un-necessary columns.
2.Initialize the ε-Neighborhood and minpts which are passed as parameters to algorithm.
3.The initial clusters of every point is assigned to 0.

4.For every point in dataset if the point is not visited then calculate the number of points using ‘regionQuery’ in the Neighborhood and if count of points is less than minpts then mark the point has Noise.
5.If the count of points in the neighborhood is greater than or equal to minpts then classify the label of that point to the clusterId by incrementing the clustered and these point is passed to ‘addToCluster’ function which marks all the neighborhood points of this cluster with this clusterId and marks them visited.
6.Repeat this process till all the points are visited and labelled with cluster id’s.

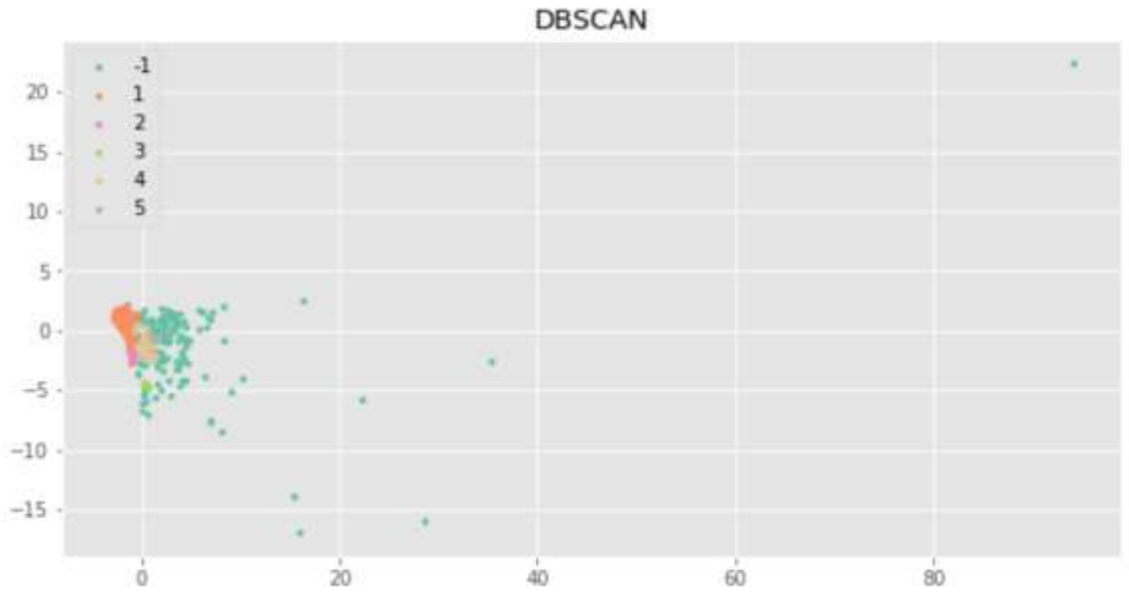
## Visualizations:

**Jaccard Coefficient**

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

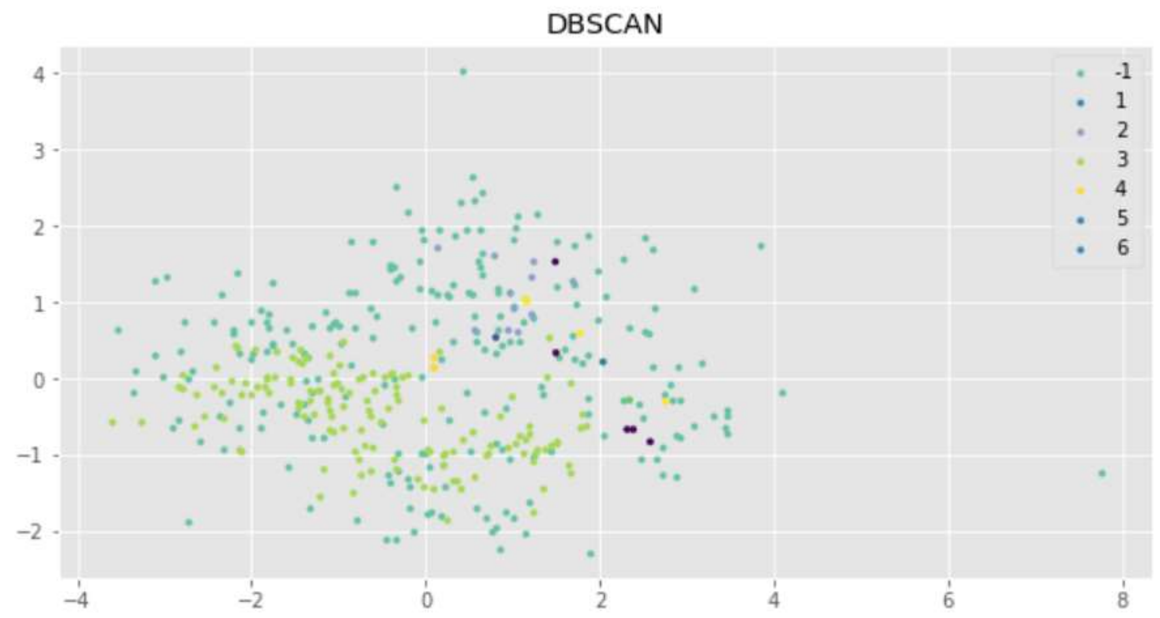
1.For “iyer.txt” file with ε = 1, Minpts = 3.

```
Jaccard: 0.2835567491646023
Rand: 0.6526755683922646
```



2.For“cho.txt” with ε = 1, Minpts = 3.

```
Jaccard: 0.20445756872671025
Rand: 0.5443501838975543
```



## PROS of DBSCAN:

1. It doesn’t require to mention number of clusters.
2. DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster.
3. This algorithm is resistant to noise.
4. DBSCAN can handle clusters of different shapes and sizes.

## CONS of DBSCAN:

1. It is not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data is processed.
2. The quality depends on the distance measure used in the function regionQuery, used to identify the neighbors.
3. If the data and scale are not well understood, choosing a meaningful distance threshold ε can be difficult.

## 4.SPECTRAL CLUSTERING ALGORITHM

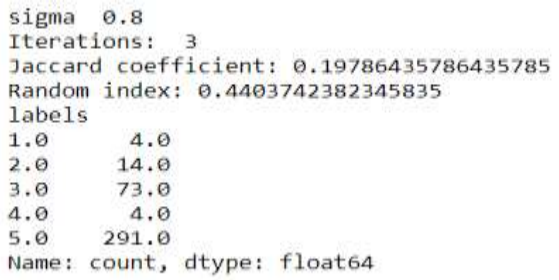
In Spectral Clustering, any connected points or the points that are immediate to each other are put in the same cluster. These points are treated as the nodes of a graph. So this clustering is treated as a graph partitioning problem.
A Similarity graph is computed and the data is projected on to a low dimensional space and the clusters are created.
There are no assumptions about the shape or form of cluster.

### Implementation of Algorithm:

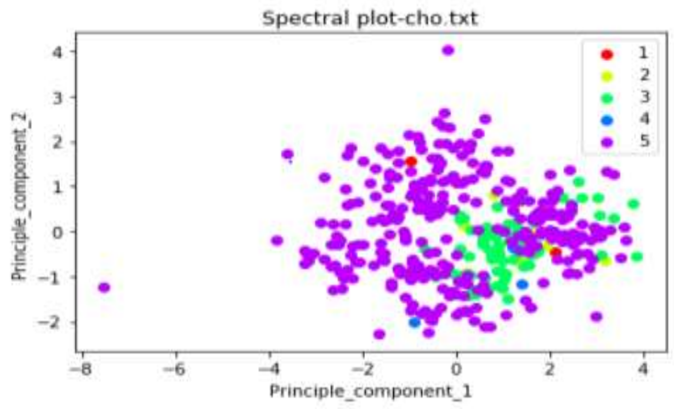
1.The data matrix is constructed by using the input file given and removing the unnecessary columns.
2.Build the similarity matrix by calculating the weights using the Euclidean distance.
3.Compute the degree matrix by summing all the values in a row and place in the respective diagonal position.
4.Now, Laplacian matrix is computed by taking the difference between similarity and degree matrix.
5.Eigen vectors and Eigen values are calculated for the laplacian matrix obtained.
6.Build embedded space from the eigenvectors corresponding to the *k* smallest eigenvalues
7. Apply *k*-means to the reduced *n* x *k* space to produce *k* clusters, where k is the number of input clusters given.

### Visualization:

1.For "cho.txt"

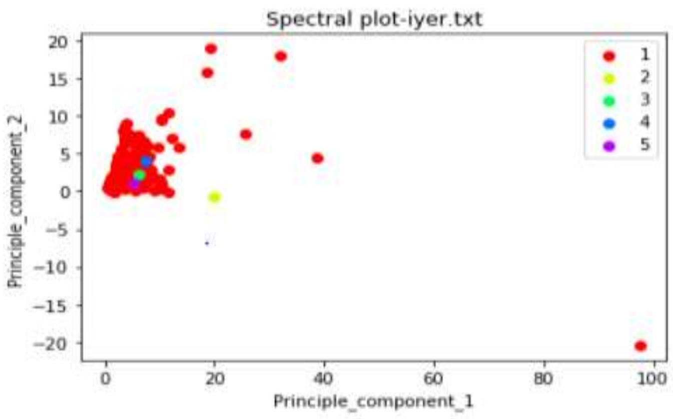


```
sigma  0.8
Iterations:  3
Jaccard coefficient: 0.19786435786435785
Random index: 0.4403742382345835
labels
1.0      4.0
2.0     14.0
3.0     73.0
4.0      4.0
5.0    291.0
Name: count, dtype: float64
```



2.For “iyer.txt”

```
sigma  0.8
Iterations:  2
Jaccard coefficient: 0.1563480947555009
Random index: 0.16844314580846947
labels
1.0    513.0
2.0      1.0
3.0      1.0
4.0      1.0
5.0      1.0
Name: count, dtype: float64
```



### PROS of Spectral Clustering:

1.Performs well with wide variety of data. Clusters not assumed to be any certain shape or distribution.
2. Fast for sparse data sets for several elements.
3.Do not necessarily need the actual data set, just similarity/distance matrix, or even just Laplacian. Because of this, we can cluster one dimensional data.

### CONS of Spectral Clustering

1.Need to choose the number of clusters k, although there is a heuristic to help choose.
2.Can be costly to compute, because to calculate eigen values and eigen vectors.

## 5. GMM Clustering

Gaussian mixture models are probabilistic model for representing normally distributed sub populations within overall populations. Mixture models in general don’t require knowing which sub-population a data point belongs to allowing the model to learn sub-populations automatically.
It can be used to cluster unlabeled data similar to k-means but there are advantages over k-means.

The main objective is to maximize the log likelihood value and it uses EM algorithm.

## Implementation of Algorithm:

1.Prepare the dataset from the given input file and initialize the GMM parameters like mean, covariance, prior probability, and also set the values for no.of.iterations, convergence threshold and smoothing threshold.

2.Now, we compute Expectation step. Here for given parameter values we can compute the expected values of the latent variables using the formulae:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3.Now we implement the maximization step. Since above computed value is common to the expressions for mean, variance and prior probability we can simply define:

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

And then we can calculate the revised parameters by using:

$$\pi_k^* = \frac{N_k}{N}$$

$$\mu_k^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\Sigma_k^* = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

4.So in the above M-step we maximize the expected complete log-likelihood
5.Iterate E and M steps until log-likelihood of data does not increase anymore.
-converge to local optimal
-need to restart algorithm with initial guess of parameters.

## PROS of GMM Clustering:

1.Give probabilistic cluster assignments
2.Have probabilistic interpretation
3.Can handle clusters with varying sizes, variance etc

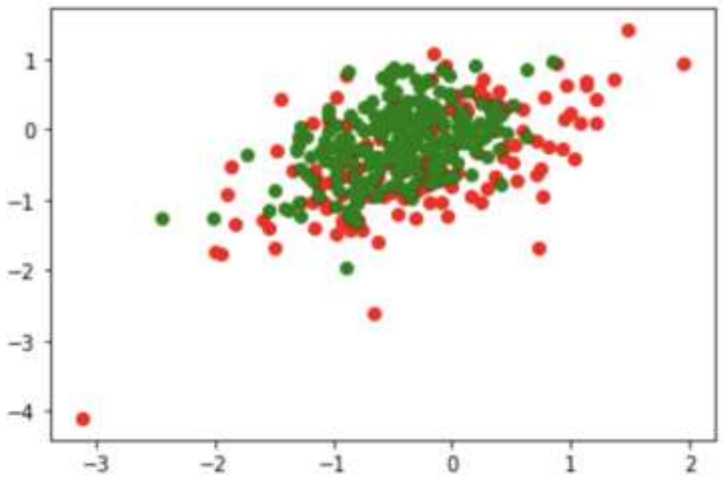
## CONS of GMM Clustering:

1.Initialization matters
2.Choose appropriate distributions
3.Overfitting issues

## Visualization:

1.For “cho.txt”, n=5, iter=100,mean,variance,prob as given

| | |
|---|---|
| Jaccard is:0.3021 | Rand is:0.6842 |



## PERFORMANCE ANALYSIS

Jaccard and Rand for all the clustering algorithms: for cho.txt dataset

| | Jaccard Coefficient | Rand Index |
|---|---|---|
| K-means clustering(k=3) | 0.4088 | 0.7574 |
| Hierarchical Agglomerative clustering(K=3) | 0.2283 | 0.2402 |
| Density-based clustering (ε = 1, Minpts = 3) | 0.2044 | 0.5443 |
| Mixture model clustering | 0.3569 | 0.7545 |
| Spectral clustering(sigma=0.8,iter=3) | 0.1978 | 0.4403 |

Jaccard and Rand for all the clustering algorithms: for iyer.txt dataset

| | Jaccard Coefficient | Rand Index |
|---|---|---|
| K-means clustering(k=3) | 0.2199 | 0.4995 |
| Hierarchical Agglomerative clustering(K=3) | 0.1584 | 0.1941 |
| Density-based clustering (ε = 1, Minpts = 3) | 0.2835 | 0.6526 |
| Mixture model clustering | 0.3021 | 0.6842 |
| Spectral clustering(sigma=0.8,iter=3) | 0.1563 | 0.1684 |