Project2 Clustering Algorithms

Jingsong Li
Department of CSE
University at Buffalo
50322345
jingsong@buffalo.edu

Junyang Li
Department of CSE
University at Buffalo
50320301
junyangl@buffalo.edu

Kaige Gao
Department of CSE
University at Buffalo
50320915
kaigegao@buffalo.edu

Abstract

This project is to implement five clustering algorithms to find clusters of genes that exhibit similar expression profiles.

1 Introduction

The task of this project is to implement five clustering algorithms to find clusters of genes that exhibit similar expression profiles: K-means, Hierarchical Agglomerative clustering with Min approach, density-based, mixture model, and spectral clustering. Compare these five methods and discuss their pros and cons.

For each of the above tasks, the validation of clustering results will use the following methods:

- Using external index (Rand Index and Jaccard Coefficient) and compare the clustering results from different clustering algorithms. The ground truth clusters are provided in the datasets.
- Visualize data sets and clustering results by Principal Component Analysis (PCA).

2 Dataset

Two gene datasets (cho and iyer) will be used as the example data in this roject.

3 Preprocess

3.1 Read data

With the given text file, the program will read each line of it and store as a two-dimension list.

3.2 Data cleaning

The first column of the dataset we get is the index of the sample, which is meaningless, so the program will drop the first column and separate the following data as ground truth(second column) and attributes(following columns).

3.2 Normalization

Normalization is optional in this project because the range of attributes are not that huge to affect the weight of each attributes. In this projects, there is no need for normalization to avoid overflow or underflow, so it’s not implemented.

4 Implementations

4.1 k-means

The first step is initialization, determine the cluster centers randomly or set the center point to the specified value.

The second step is assign each object to the cluster which has the closet distance from the centroid to the object.

The third step is compute cluster centroid as the center of the points in the cluster.

Repeat step 2 and 3 above until the SSE value of clusters tend to stabilize.

4.2 Hierarchical Agglomerative clustering with Min approach

The first step is to build classes which each sample represent a single class. It is the initial stage, and with the cluster goes, the number of classes will decrease.
The second step is to build a distance matrix with Euclidean distance. Normally, it will take N^2 time, but if the for loop only goes half of it and copy the value to the other half, half of the time can be saved, it can help save seconds with large samples.
The third step is to mark outliers. With the threshold manually set, program can find samples having the distance to all other points with the distance larger than threshold and mark them as outliers. These points will be removed from the matrix and add to the first class. The reason to put them into first class is as cluster goes, the size of classes will change, but the first position won't change.
The next step is to reduce the matrix size until it reaches the size we want. For each iteration, the program will find two points with the closest distance, merge them together and set the distance to other points as the smaller distance between these two points.
With the result generated, the program will generate a confuse matrix and calculate rand index with (TP+TN)/(TP+TN+FP+FN) and jaccard coefficient with TP/(TP+FP+FN).
At last, the program will decrease the dimension of input sample to 2dimension with PCA package and plot each class with different color.

4.3 density-based

Instead of using list, the program use class to store information of each sample. So the first step is just to create an empty classes.
The second step is to build a distance matrix with Euclidean distance. Normally, it will take N^2 time, but if the for loop only goes half of it and copy the value to the other half, half of the time can be saved, it can help save seconds with large samples.
The third step is to find out every point that in the certain distance to every point and record their index to an attribute of the class.
After that the program start to traverse every unvisited point to do the clustering. If there is not enough number of points besides, it will be mark as noise. The point that has certain number of points around will be marked as core point, and those points will own their class, if two points are in the certain distance, they will share the same class. More detailly, it's keep find the core point and merge their points around to the cluster.
With the result generated, the program will generate a confuse matrix and calculate rand index with (TP+TN)/(TP+TN+FP+FN) and jaccard coefficient with TP/(TP+FP+FN).
At last, the program will decrease the dimension of input sample to 2dimension with PCA package and plot each class with different color.

4.4 mixture model

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters comprising the means and covariances of the components and the mixing coefficients).
So the first step is E step. Evaluate the responsibilities using the current parameter values. Then M step. Re-estimate the parameters using the current responsibilities. Then evaluate log likelihood if there is no convergence, return to the beginning.

4.5 spectral clustering

The first step is pre-processing. Compute Laplacian matrix L based on Similarity matrix W and Degree matrix D. Then Normalized Laplacian matrix L.

The second step is Decomposition. Find the eigenvalues and eigenvectors of L. Build embedded space from the eigenvectors corresponding to the k smallest eigenvalues.
The last step is Clustering. Apply k-means to the reduced n x k space to produce k clusters.

5 Results

5.1 k-means

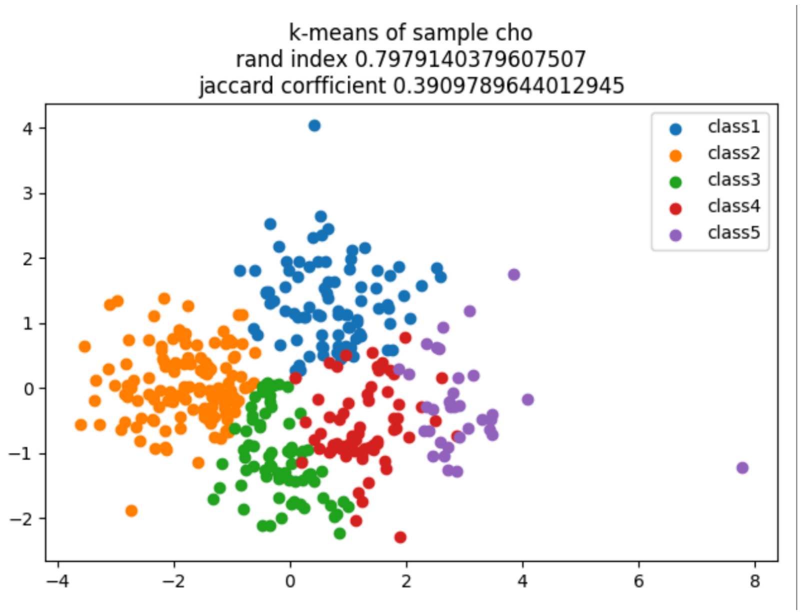

Figure: k-means result for sample cho with number of cluster 5

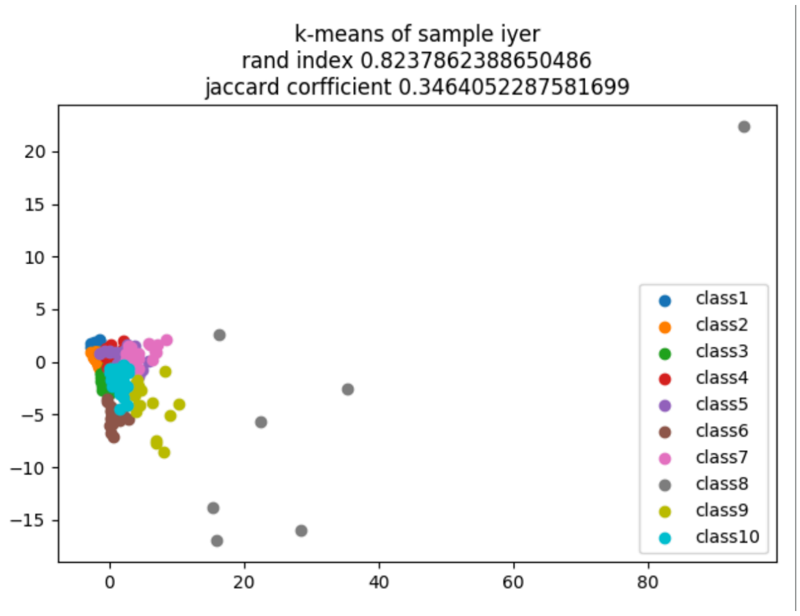

Figure: k-means result for sample iyer with number of cluster 10

5.2 Hierarchical Agglomerative clustering with Min approach

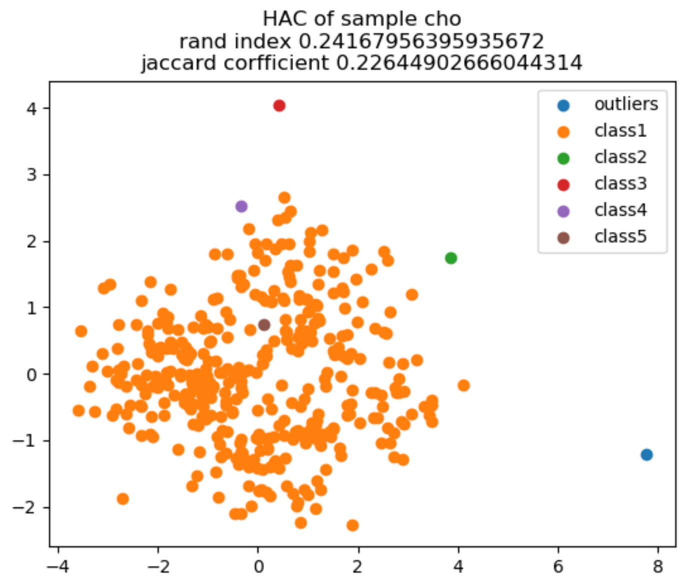

Figure: HAC-min result for sample cho with number of cluster 5 and threshold distance 4

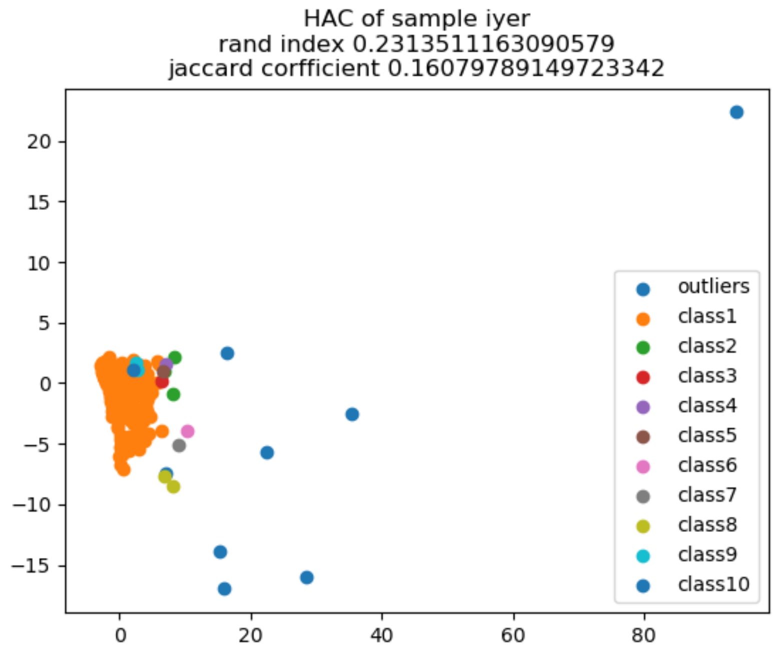

Figure: HAC-min result for sample iyer with number of cluster 10 and threshold distance 4

5.3 density-based

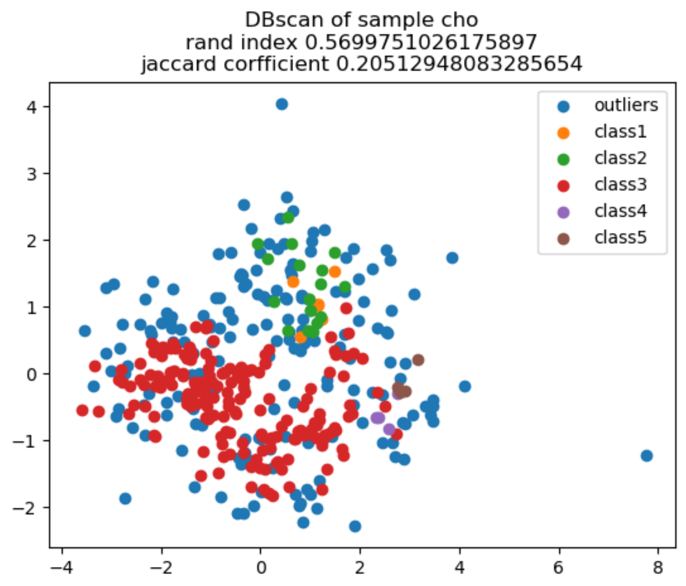

Figure: DBscan result for sample iyer with ε =1.08 and MinPts =4

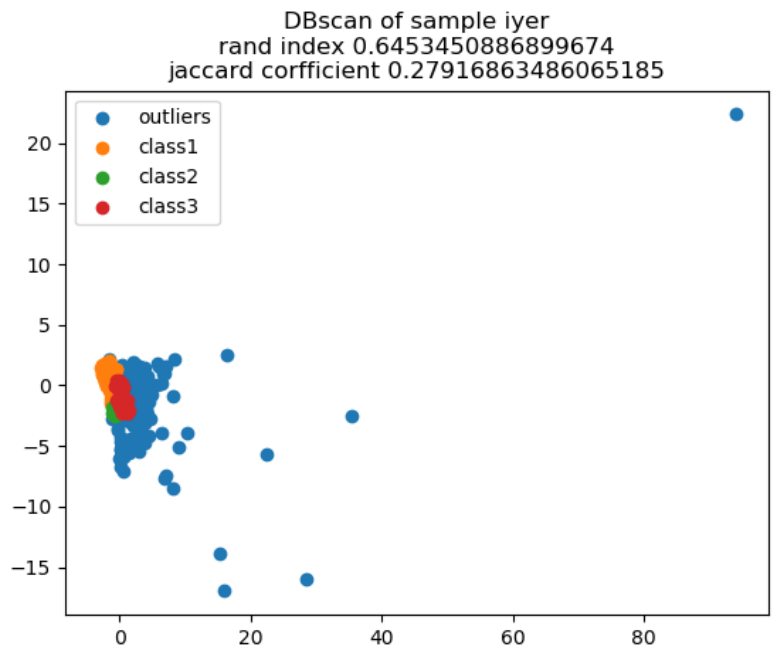

Figure: DBscan result for sample iyer with ε =1 and MinPts =5

5.4 mixture model

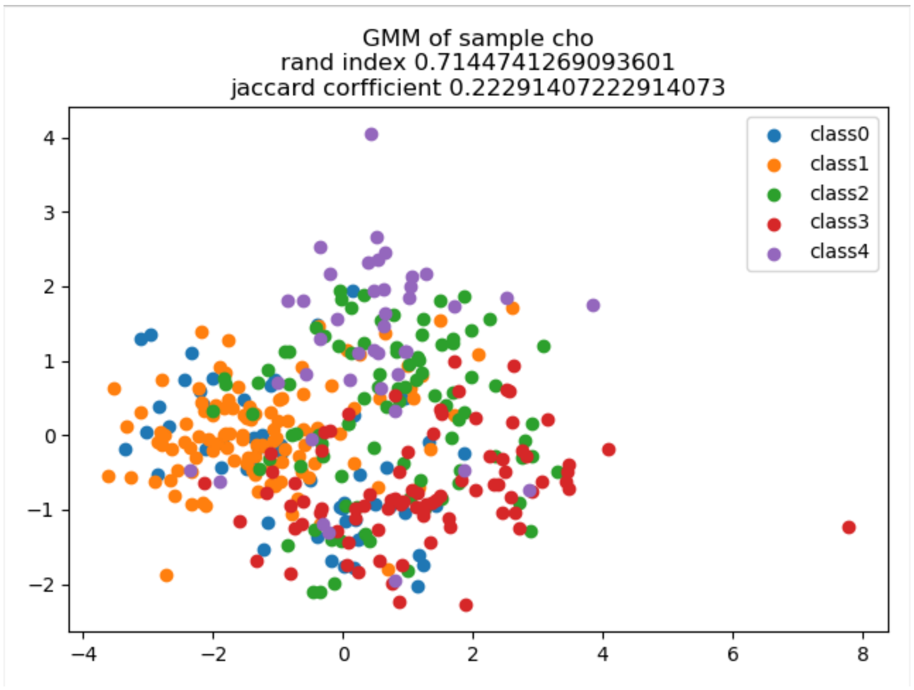

Figure: GMM result for sample cho

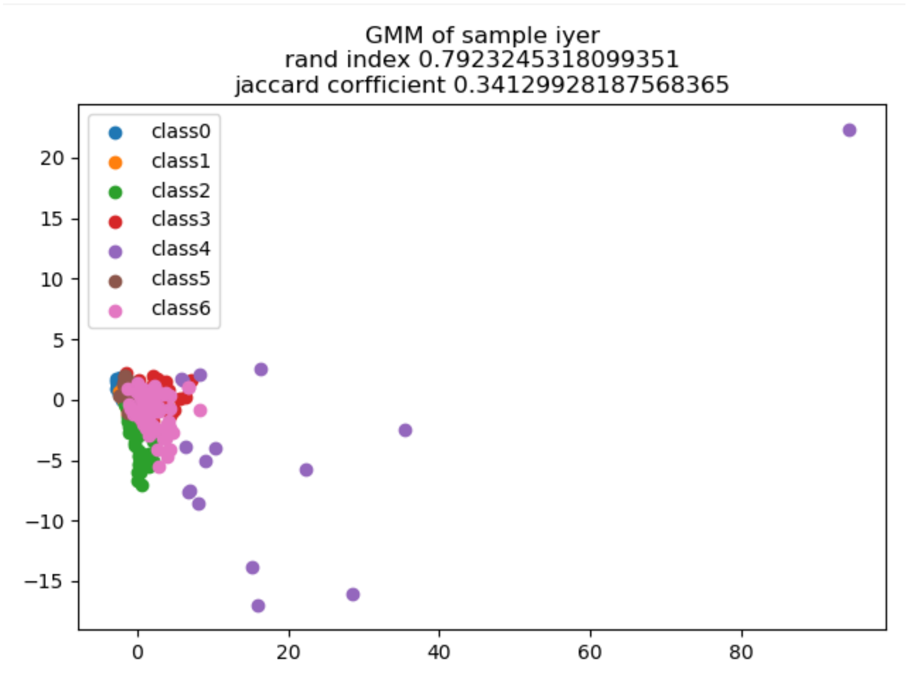

Figure: GMM result for sample iyer

5.5 spectral clustering

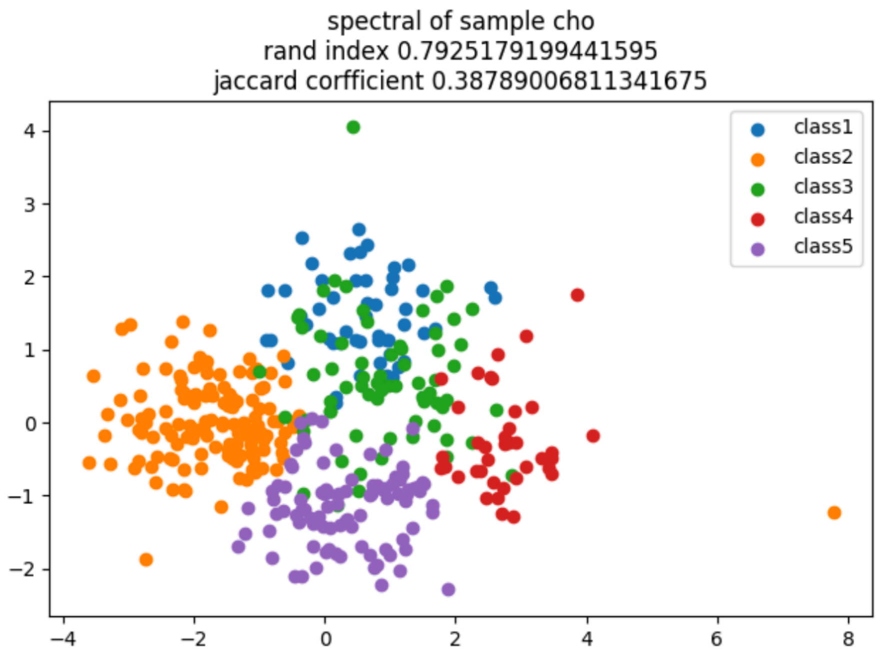

Figure: spectral result for sample cho with number of cluster 5 and sigma 2

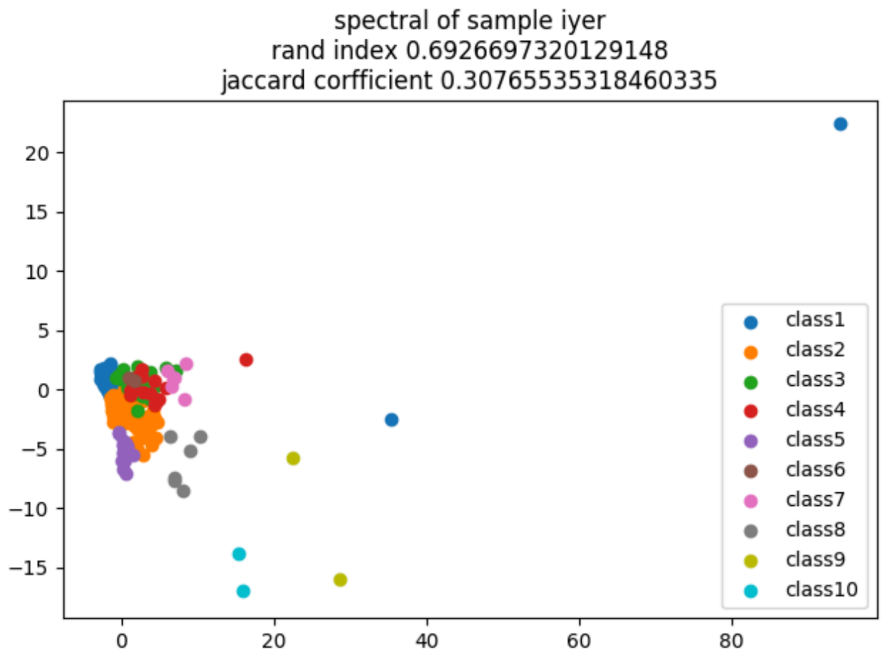

Figure: spectral result for sample iyer with number of cluster 10 and sigma 7

6 Evaluation

6.1 k-means

Advantages

1. It is relatively simple, the implementation is also very easy, and the convergence speed is fast.
2. The result of clustering performs well.
3. The interpretability of the algorithm is relatively strong.
4. The main parameter to be adjusted is only the cluster number k.

Disadvantages

1. Results depend on the initialization of the classification center.
2. Sensitive to outliers (easy to cause center point deviation).
3. The result is a local optimal solution instead of a global optimal solution..
4. The clusters might have different densities.
5. The clusters might have different sizes.
6. The clusters might have irregular shapes.

6.2 Hierarchical Agglomerative clustering with Min approach

Advantages

1. Easy to implement.
2. Do not have to assume any particular number of clusters. Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level.
3. Well suited for problems involving point linkages, e.g. taxonomy trees.

Disadvantages

1. Inability to make corrections once the splitting/merging decision is made.
2. No objective function is directly minimized.
3. Sensitivity to noise and outliers.
4. Prohibitively expensive for high dimensional and massive datasets.
5. Difficulty handling different sized clusters and irregular shapes.
6. Breaking large clusters.
7. Severe effectiveness degradation in high dimensional spaces due to the curse of dimensionality phenomenon.

6.3 density-based

Advantages

1. Discovery of arbitrary-shaped clusters with varying size.
2. Resistance to noise and outliers

Disadvantages

1. Sensitive to parameters—hard to determine the correct set of parameters.
2. Unsuitable for high-dimensional datasets because of the curse of dimensionality phenomenon, this means that is hard to find a appropriate value for ε.
3. Cannot handle varying densities, since the (MinPts, ε) combination cannot be chosen appropriately for all clusters then.

6.4 mixture model

Advantages

1.Give probabilistic cluster assignments

2.Have probabilistic interpretation
3.Can handle clusters with varying sizes, variance etc.
Disadvantages
1.Initialization matters
2.Choose appropriate distributions
3.Overfitting issues

6.5 spectral clustering

Advantages
1. Spectral clustering only needs similarity matrix between data, so it is very effective for dealing with sparse data clustering.
2. Due to the use of dimensionality reduction, the complexity of processing high-dimensional data clustering is better than that of traditional clustering algorithms.
Disadvantages
1. If the dimension of the final clustering is very high, the running speed of spectral clustering and the final clustering effect are not good due to insufficient dimension reduction.
2. The clustering effect depends on the similarity matrix, and the final clustering effect obtained by different similarity matrices may be very different.

7 Conclusion

cho	k-means	HAC	DBScan	GMM	spectral
Rand Index	0.7979	0.2417	0.5699	0.7145	0.7925
Jaccard Coeficient	0.3910	0.2264	0.2051	0.2229	0.3879

iyer	k-means	HAC	DBScan	GMM	spectral
Rand Index	0.8238	0.2314	0.6453	0.7923	0.6927
Jaccard Coeficient	0.3464	0.1608	0.2791	0.3413	0.3077

Based on above, k-means can get the best clustering result. GMM and spectral are also suitable for these two datasets. Both HAC-min model and DBscan model are not suitable for either sample of cho or iyer. The HAC requires sample be tightly in a group and far away between groups, but sample given don't meet the need, so the result shows a trend of pointing out the noise rather than show the clustering.
DBscan model can't handle sample with varies density, which make it has a extremely bad result at sample iyer.

Appendix

Ground truth of two sample are given below:

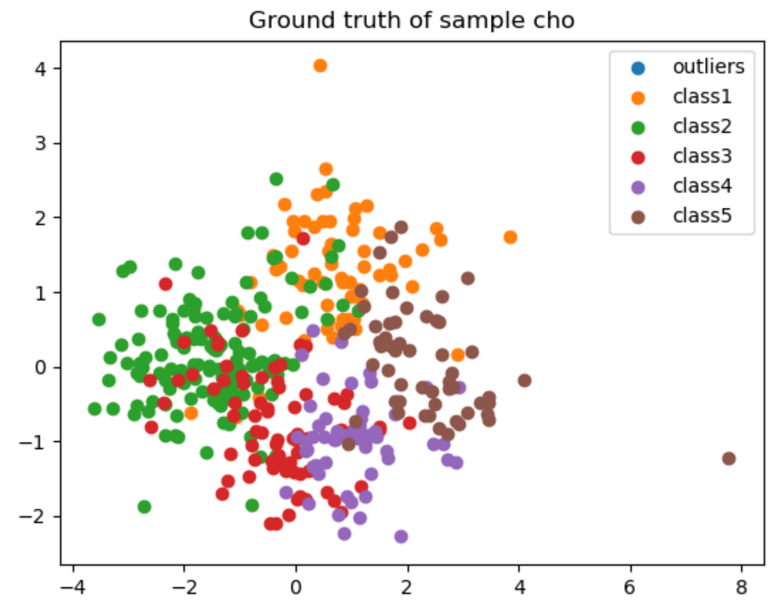

Figure: Ground truth of sample cho

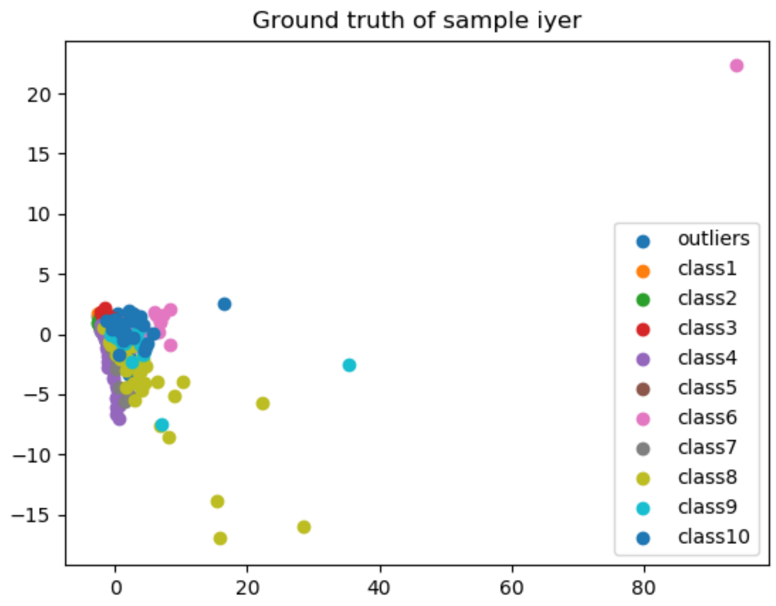

Figure: Ground truth of sample iyer