In this project, I implemented two python files, one is "Clustering Algorithms 1" and the other is "Clustering Algorithms 2". Because the Iyer dataset contains a small number of outliers, I use different processing methods to process the outliers in these two different python files. In "Clustering Algorithms 1", I performed k-means clustering and spectral clustering on the Cho dataset and Iyer dataset. When processing the Iyer dataset, I directly treated the outliers as normal points for clustering. That is, the Iyer data is directly divided into ten clusters. In "Clustering Algorithms 2", I once again performed k-means clustering and spectral clustering on the Cho dataset and the Iyer dataset. Unlike the previous one, this time I first removed the outliers from the dataset. After the clustering is completed, the outliers are returned to their original positions to ensure that the amount of data in the Iyer dataset remains consistent.

In addition, I ran each cluster 100 times, calculated the internal index SSE of the cluster each time, and use the cluster with the lowest SSE as the final result. After getting the clustering results, I used the adjusted Rand index as the external index. Finally, I compared the results predicted by the clustering algorithm I implemented with the results predicted by scikit-learn, it can be seen that the adjusted Rand indexes of the predicted results of the clustering algorithm I implemented are higher in most cases.

**Flow of the k-means algorithm:**

In the implementation of the k-means algorithm, I first randomize K center points, calculate the Euclidean distance between each point in the data set and the center points, and assign each point to the nearest center point, so that K clusters are formed. Then update the center point of each cluster and assign each point to its new nearest center point. . . Repeat the assignment and update steps until the cluster does not change. Finally, when the centroid no longer changes, calculate the SSE.

**Flow of the spectral clustering algorithm:**

In the implementation of the spectral clustering algorithm, I performed the following process:

1) Construct a sample similarity matrix S.

2) According to the similarity matrix S, an adjacency matrix A using the k-nearest neighbor method is constructed, and a degree matrix D is constructed. After comparison, the clustering effect is better when the value of k is 40 to 60, so I select 50 as the value of k.

3) Calculate the Laplacian matrix, L = D-A.

4) Construct a standardized Laplacian matrix D^ (-1/2) L D^ (-1/2).

5) Calculate the eigenvector f corresponding to the smallest m eigenvalues of D^ (-1/2) L D^ (-1/2).

6) Combining the corresponding eigenvector f into a feature matrix F.

7) For each row in F as a sample, use the k-means clustering method to perform clustering.

8) Get the m clusters.

**Compare the performance using external and internal index on the two given datasets:**

After introducing the process of the two algorithms, let’s compare the results based on adjusted Rand index (external index) and SSE (internal index). The bounded range of ARI (adjusted Rand index) is [-1, 1]: negative values are bad (independent labels), similar clusters have a positive ARI, 1.0 is the perfect match score.

1. Use **k-means** clustering I implemented for the Cho dataset:

SSE: 60.967744723762145

adjusted Rand index: 0.4475000274739754

1. Use **k-means** clustering of sciki-learn for the Cho dataset:

adjusted Rand index: 0.445531161593973

1. Use **k-means** clustering I implemented for the Iyer dataset (no treatment for outliers):

SSE: 15.208764423889082

adjusted Rand index: 0.2671347011479459

1. Use **k-means** clustering I implemented for the Iyer dataset (delete outliers):

SSE: 14.204476523298052

adjusted Rand index: 0.371055847383926

1. Use **k-means** clustering of sciki-learn for the Iyer dataset:

adjusted Rand index: 0.22383873517794003

1. Use ***spectral*** clustering I implemented for the Cho dataset:

SSE: 0.8169329593742033

adjusted Rand index: 0.347206481819754

1. Use ***spectral*** clustering of sciki-learn for the Cho dataset:

adjusted Rand index: 0.44129189070888103

1. Use ***spectral*** clustering I implemented for the Iyer dataset (no treatment for outliers):

SSE: 2.7579104629803584

adjusted Rand index: 0.243739928799015

1. Use ***spectral*** clustering I implemented for the Iyer dataset (delete outliers):

SSE: 2.63976587764725

adjusted Rand index: 0.29380876631564645

1. Use ***spectral*** clustering of sciki-learn for the Iyer dataset:

adjusted Rand index: 0.22409326018025702

**The pros and cons of each algorithm:**

**K-Means' main advantages:**

1) The principle is simple and easy to implement

2) Strong explainability

**K-Means' main disadvantages:**

2) Local optimal

3) Sensitive to noise and abnormal points

5) The clustering effect depends on the initialization of the cluster center

6) Not ideal for non-convex data sets or data with large differences in category size

**Spectral clustering's main advantages:**

1) Spectral clustering only requires the similarity matrix between data, so it is very effective for clustering with sparse data. This is difficult for traditional clustering algorithms such as K-Means.

2) Because of the use of dimensionality reduction, the complexity in processing high-dimensional data clustering is better than traditional clustering algorithms.

**Spectral clustering's main disadvantages:**

1) If the dimension of the final cluster is very high, the running speed of the spectral cluster and the final clustering effect are not ideal due to the insufficient dimensionality reduction.

2) The clustering effect depends on the similarity matrix. The final clustering effect obtained by different similarity matrices may be very different.