Outlier Detection using Spectral Clustering

Anirudh Srinivasan CS20BTECH11059
Haritha R AI20BTECH11010
Nelakuditi Rahul Naga AI20BTECH11029
Rahul V AI20BTECH11030
Saketh Vaddamani CS20BTECH11054

1. Problem Statement

The goal of spectral clustering-based outlier detection is to locate data points in a collection that considerably depart from the mean or are regarded as outliers. The objective is to isolate these outlier points from the other data points and cluster them separately based on how closely or distantly they are related to other data points in the dataset.

Anomaly identification in network traffic, product defect detection in manufacturing processes, and many other real-world situations can all involve the occurrence of outliers. In these situations, spotting outliers is crucial because they may indicate important dangers or opportunities that need to be addressed or taken advantage of.

As it can capture intricate relationships between the data points, even when the data is high-dimensional and non-linear, spectral clustering is a potent technique for outlier detection. In order to cluster the data points, it is necessary to describe the data points as a similarity matrix, compute the Laplacian matrix, and then use its eigenvectors and eigenvalues. The data points that do not fit into any of the clusters are subsequently known as the outliers.

Overall, detecting outliers in a dataset via spectral clustering is a valuable and effective method that can aid in making more informed decisions across a variety of fields.

2. Description of the dataset

The first 7 columns (cov1 to cov7) represent different coverage measures for some unknown genomic regions. The next three columns (sal_pur_rat, igst_itc_tot_itc_rat, and lib_igst_itc_rat) represent some measures of interest for the samples. Specifically, sal_pur_rat represents the ratio of sales to purchases, igst_itc_tot_itc_rat represents the ratio of integrated goods and services tax (IGST) and input tax credit (ITC), and lib_igst_itc_rat represents the ratio of the library IGST and ITC.

The format of the dataset we used are shown in table 1. Details about basic statistics of the data are given in the table 2. Other details about the dataset have been provided in the code that we have submitted.

3. Algorithms used

Before implementing spectral clustering algorithm, let us go through the basics of EigenValues and EigenVectors.

3.1. EigenValues and EigenVectors

If there exists a vector x which isn't all zeros and a scalar λ for a matrix A, such that

$$Ax = \lambda x \tag{1}$$

Then, we call x to be the eigenvector of A for the eigenvalue λ.

3.2. Adjacency Matrix and Degree Matrix

A graph is a set of nodes with a corresponding set of edges which connects the nodes. Graphs are generally used to represent many types of data. A graph can be represented as an adjacency matrix, where the row and column indices are used to represent nodes. The value located at those indices will indicate whether there is an edge between two nodes or not. So, for example if in row 1 and column 2, there is a 1 present in adjacency matrix. Then, we can say that there is an edge between 1 and 2. A degree matrix of a graph is a diagonal matrix, where the degree of node present is at index (i,i) in the matrix.

3.3. Graph Laplacian

A Graph Laplacian can be considered to be another matrix representation of a graph. It has several good properties, which can be very helpful when implementing Spectral Clustering. One of the fascinating property of Laplacian is, we can find the degree of the nodes of a graph from its diagonal and the off-diagonal gives the

cov1	cov2	cov3	cov4	cov5	cov6	cov7	sal_pur_rat	igst_itc_tot_itc_rat	lib_igst_itc_rat
0.997796872	0.999887763	0.215933502	0.196712828	0	0.955616028	0.998810174	-0.032580695	1.761759359	-0.054328994
0.994003688	0.979901847	-0.337135366	-0.248633998	0	0.640811704	0.553917779	-0.03202564	-0.629311312	-0.053516379
0.947603047	0.455666574	0.001742827	0.128609757	-0.004054398	-0.162069022	0.960601374	-0.030208666	1.535696531	-0.054214797
0.396577269	0.919933109	0.49645071	0.576823525	-0.340717816	0.802363209	0.673710394	-0.032058052	0.449159691	-0.054126384

Table 1. Basic Format of the Data

	cov1	cov2	cov3	cov4	cov5	cov6	cov7	sal_pur_rat	igst_itc_tot_itc_rat	lib_igst_itc_rat
count	1199.000000	1199.000000	1199.000000	1199.000000	1199.000000	1199.000000	1199.000000	1.199000e+03	1.199000e+03	1.199000e+03
mean	0.956896	0.855770	0.214263	0.147359	0.036329	0.599809	0.527768	-1.251042e-11	-5.004165e-12	1.199000e+03
std	0.135031	0.244927	0.408193	0.388080	0.177615	0.334306	0.385322	1.000000e+00	1.000000e+00	1.000000e+00
min	-0.312219	-0.531958	-0.818128	-0.839158	-0.719622	-0.682734	-0.859529	-3.531330e-02	-1.066436e+00	-5.444774e-02
25%	0.982505	0.840675	-0.095193	-0.143054	0.000000	0.382479	0.245701	-3.284146e-02	-8.884636e-01	-5.424427e-02
50%	0.999235	0.969806	0.175910	0.097584	0.000000	0.691423	0.595623	-3.254101e-02	-3.457085e-01	-5.382146e-02
75%	0.999993	0.996604	0.563061	0.457633	0.000000	0.873218	0.869592	-3.194269e-02	7.059485e-01	-5.191380e-02
max	1.000000	1.000000	1.000000	0.979015	0.999196	0.999999	1.000000	3.436719e+01	2.177948e+00	3.318828e+01

Table 2. Basic Statistics of the Data

negative weights. We see that when a graph is completely disconnected, then all the eigenvalues of a Laplacian are 0. But as we keep adding edges, the eigenvalue increases and we will find that the number of 0 eigenvalues in a Laplacian corresponds to number of connected components in a graph.

The first non-zero eigenvalue of a Laplacian is called the spectral gap, which gives us a notion of how densely a graph is connected. The second eigenvalue, also called the Fiedler value, has the corresponding vector, the Fiedler vector. The Fiedler value gives an approximation of the minimum graph cut required to separate the graph into two connected components.

3.4. Spectral Clustering

Before performing spectral clustering on the given dataset, the dataset is scaled and normalized appropriately using **StandardScaler** and **normalize** functions from the sklearn library.

Spectral clustering essentially requires us to use a standard clustering method such as K-means on relevant eigenvectors of a Laplacian (or) Normalized Laplacian matrix (the ones that correspond to smallest several eigenvalues of the Laplacian (or) Normalized Laplacian matrix except for the smallest eigenvalue which will have a value of 0) constructed using similarity matrix A of a similarity graph.

For a given set of datapoints, the similarity matrix can be defined as a symmetric matrix A, where the entry A_{ij} represents the measure of the similarity between data points with indices i and j. We will construct the similarity matrix A using a **Mutual-kNN** similarity graph. The similarity matrix A can be constructed using a Mutual-kNN similarity graph as follows for the given dataset :

1. Compute the distance matrix that stores the distance between every two datapoints. Note that a row in the given dataset is a single datapoint. We use the **Cosine Similarity** distance metric to compute the distance between two datapoints. The Cosine Similarity distance between two vectors $\vec{a}$ and $\vec{b}$ is given by :

$$d = 1 - \frac{\vec{a}.\vec{b}}{||\vec{a}|| \times ||\vec{b}||} \tag{2}$$

   where $\vec{a}.\vec{b}$ denotes the dot product between vectors $\vec{a}$ and $\vec{b}$. Also, $||a||$ and $||b||$ denote the L_2-norm of vectors $\vec{a}$ and $\vec{b}$ respectively.

2. Using the distance matrix computed above, we now construct the similarity matrix A. Let us consider two datapoints x_i and x_j in the given dataset. There is an edge between these two datapoints in the Mutual-kNN similarity graph if and only if x_i is in the k-nearest neighbours of x_j **and** x_j is in the k-nearest neighbours of x_i. If there is an edge between datapoints x_i and x_j, we set $A_{ij} = 1$. Otherwise, we have $A_{ij} = 0$.

Note the similarity matrix A we constructed above is a **symmetric** matrix. Now, we construct the **Laplacian** matrix using the similarity (or) adjacency matrix A as follows :

$$L = D - A \tag{3}$$

where D is a diagonal matrix defined as follows :

$$D_{ii} = \sum_j A_{ij} \tag{4}$$

Further, we construct the **Normalized Laplacian** matrix $\hat{L}$ as follows :

$$\hat{L} = D^{-1/2} L D^{-1/2} = I_{N \times N} - D^{-1/2} A D^{-1/2} \tag{5}$$

where $I_{N\times N}$ denotes an identity matrix of size $N \times N$, N is the total number of datapoints.

Now, we find the eigenvector matrix (of size $N \times K$) corresponding to the first K eigenvectors i.e; the eigenvectors corresponding to the K smallest eigenvalues of the **Normalized Laplacian** matrix $\hat{L}$. Note that row i in the eigenvector matrix is a K-dimensional representation of the datapoint i.

Finally, we cluster the datapoint representations we obtained in the eigenvector matrix using K-Means clustering. To better visualize the clusters and the outliers in the given dataset, we have projected the K-dimensional data into 2D real space using **TSNE** algorithm.

4. Final Results and Conclusions

The results obtained after clustering the K-dimensional data and projecting them into 2D real space using **TSNE** algorithm for different values of hyperparameters k, K (k - Number of nearest neighbours in Mutual-kNN, K - Number of clusters in K-Means) are as follows :

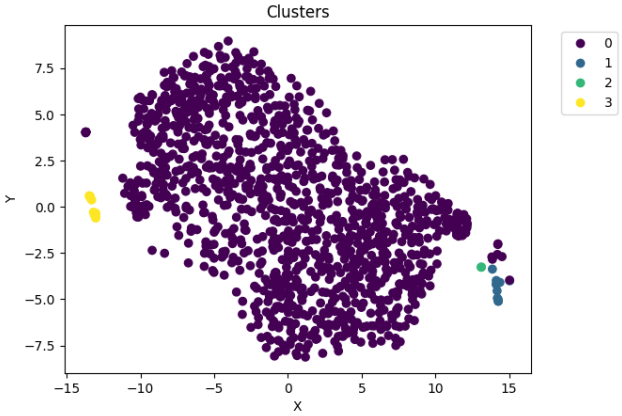

Figure 1. $k = 10, K = 4$

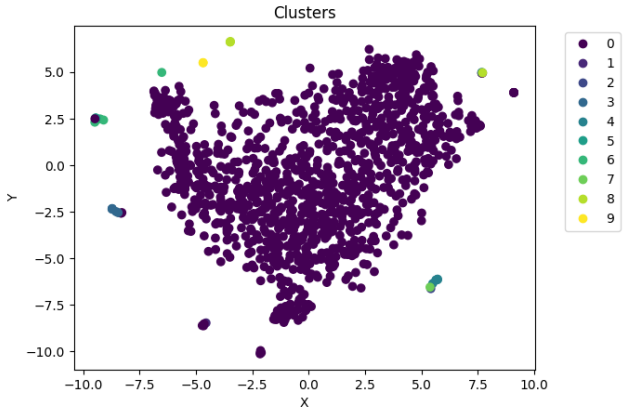

Figure 2. $k = 10, K = 10$

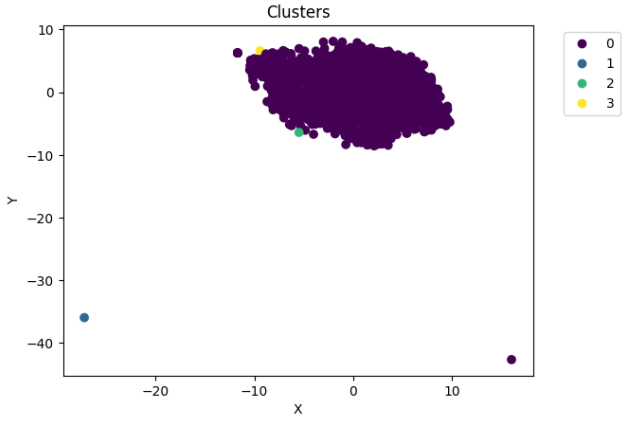

Figure 3. $k = 15, K = 4$

References

[1] Spectral Clustering Wikipedia - Link

[2] Spectral Clustering slides shared through AIMS

[3] Spectral Clustering Blog - Link