# Identifying the outliers using Spectral Clustering

Fraud Analytics CS6890 - Assignment 3

M Anand Krishna
*M.Tech Computer Science*
*IIT Hyderabad*
Hyderabad, India
cs22mtech14003@iith.ac.in

Akash KS
*M.Tech Computer Science*
*IIT Hyderabad*
Hyderabad, India
cs22mtech11012@iith.ac.in

K Manish
*M.Tech Computer Science*
*IIT Hyderabad*
Hyderabad, India
cs22mtech11008@iith.ac.in

A SivaSai
*M.Tech Computer Science*
*IIT Hyderabad*
Hyderabad, India
cs22mtech11013@iith.ac.in

***Abstract*—In this report, we present a spectral clustering-based approach for detecting outliers in a dataset and thereby identifying malicious dealers. Our method involves the following key steps: data preprocessing, similarity matrix computation, Laplacian matrix construction, eigenvector extraction, and k-means clustering. After obtaining the clusters, we compute the cluster centroids and identify outliers within each cluster by comparing their distances to the centroid with a predefined threshold. The spectral clustering method has proven to be effective in identifying underlying patterns and structures in the data, which can be instrumental in detecting malicious dealers and outliers. This report details the implementation and application of our spectral clustering-based method, along with an in-depth discussion of the core concepts and techniques involved.**

***Index Terms*—Laplacian matrix,k-means clustering, Eigen vectors**

## I. Introduction

In today's increasingly connected world, the increase of online transactions has given rise to various forms of fraudulent activities, including malicious dealers. Identifying these malicious entities is crucial for maintaining trust and ensuring the integrity of the marketplace. One effective approach to detecting potentially malicious behavior is by analyzing transaction data and identifying outliers. Outliers may represent dealers that exhibit unusual patterns or characteristics compared to their non-malicious counterparts. In this report, we explore the use of k-way spectral clustering, a machine learning technique, to cluster transaction data and detect outliers. This method enables us to transform the data into a lower-dimensional space, where clustering algorithms can more effectively separate and identify distinct groups. By analyzing the identified clusters, we can detect potential outliers which contributes to the ongoing efforts to maintain a secure and trustworthy online transaction environment.

## II. Problem Statement

The rapid rise of transactions in today's digital landscape has resulted in the emergence of malicious dealers who engage in misleading practises, abuse system weaknesses, and destroy trust in the online marketplace. The key problem is effectively detecting and isolating these fraudulent entities from legitimate transactions in the middle of transaction data that is growing in size and complexity. Traditional solutions have proven ineffective, underscoring the critical need to create advanced methodologies and ways to successfully address this issue. The use of k-way spectral clustering is one such promising solution, which enables the detection and isolation of rogue traders by grouping similar transactions together and identifying patterns of fraudulent behaviour.

## III. Description of the dataset

The given dataset, `data.csv`, is an unlabeled dataset that comprises various continuous features. Each row in the dataset represents a data point with 10 distinct variables: cov1, cov2, cov3, cov4, cov5, cov6, cov7, sal_pur_rat, igst_itc_tot_itc_rat, and lib_igst_itc_rat. The dataset's purpose is to identify potential malicious dealers, which can be achieved by detecting outliers using clustering algorithms like k-way spectral clustering.

Since the dataset is unlabeled, we must rely on unsupervised learning techniques to reveal any underlying structure or relationships between the data points. By applying k-way spectral clustering, we can group the data points into separate clusters, potentially distinguishing between malicious and non-malicious dealers. Additionally, this method allows us to identify outliers within each cluster, which may correspond to dealers with anomalous behavior, possibly indicative of malicious activities.

It is crucial to preprocess the dataset to ensure that the clustering algorithm can effectively discern patterns and identify groups. Preprocessing steps include normalization, which scales the feature values to a standard range, making the data more suitable for clustering algorithms.

## IV. Algorithm used

**Algorithm 1** Outlier Detection using K-way Spectral Clustering

**Require:** Dataset $data$, number $k$ of clusters to construct
**Ensure:** Outliers detected in each cluster

Load the dataset: $data \leftarrow$ `data.csv`
Normalize the data: $data\_norm \leftarrow$ StandardScaler$(data)$
Compute the similarity matrix: $similarity\_matrix \leftarrow$ pairwise_distances$(data_norm)$
Compute the Laplacian matrix, by subtracting similartiy matrix from 'D', where 'D' represents the degree matrix ie: $L \leftarrow D - similarity\_matrix$
Compute the eigenvectors and eigenvalues: $eigenvalues, eigenvectors \leftarrow$ eig$(L)$
Sort the eigenvectors and eigenvalues in ascending order
Select the first $k$ eigenvectors: $eigenvectors\_selected \leftarrow eigenvectors[:, :k]$
Normalize the selected eigenvectors: $(eigenvectors\_norm \leftarrow$ normalize$(eigenvectors_selected))$
Now, apply K-means clustering: $(labels \leftarrow$ KMeans$(eigenvectors_norm, k))$
Compute cluster centroids: $centroids \leftarrow$ cluster_centers_$(labels)$
Define the threshold multiplier: $threshold\_multiplier \leftarrow 1.5$
**for** each unique label in $labels$ **do**
    Extract the cluster: $cluster \leftarrow data[labels == label]$
    **if** $cluster$ is not empty **then**
        Compute the centroid: $centroid \leftarrow centroids[label]$
        Compute distances: $distances \leftarrow$ norm$(eigenvectors[labels == label] - centroid)$
        Compute mean distance: $mean\_distance \leftarrow$ mean$(distances)$
        Compute threshold: $threshold \leftarrow mean\_distance \times threshold\_multiplier$
        Identify outliers: $outliers \leftarrow distances > threshold$
        **if** $outliers$ not empty **then**
            Print the outliers and their corresponding cluster label
        **end if**
    **end if**
**end for**

## V. Implementation Details

In this section, we provide an overview of the implementation details, steps to run, and hyperparameters used in the K-way Spectral Clustering algorithm for outlier detection.

### A. Implementation Details

The algorithm is implemented using Python programming language. Key libraries used for the implementation are:

- `pandas` for data manipulation and loading.
- `numpy` for numerical computations.
- `scikit-learn` for data preprocessing, clustering, and distance computations.

### B. Steps to Run

The following steps outline the process to run the K-way Spectral Clustering algorithm in a Jupyter Notebook

1) Install the required Python libraries, such as pandas, numpy, and scikit-learn, if not already installed. You can use the command '`pip install pandas numpy scikit-learn`' for installation.
2) Download the input dataset `data.csv` and save it in a suitable directory.
3) Open the provided `Assignment3.ipynb` file.
4) In the Jupyter Notebook, Change the file path in the following code block to match the location of the `data.csv` file on your system.
5) Execute the all code blocks in the Jupyter Notebook sequentially, as they appear in the notebook, to perform the K-way Spectral Clustering algorithm for outlier detection.
6) Observe the output for outlier detection, which is displayed as a list of outliers detected in each cluster.

### C. Hyperparameters

The following hyperparameters are used in the K-way Spectral Clustering algorithm:

- Number of clusters, $k$ = 6: The number of clusters to construct during the K-means clustering step.
- Sigma ($\sigma$ = 1): The bandwidth parameter for the Gaussian kernel used to compute the similarity matrix.
- Threshold multiplier = 1.5 : A multiplier used to compute the threshold for outlier detection based on the mean distance within each cluster.

Adjusting these hyperparameters can affect the clustering results and the identification of outliers.

## VI. Results

The K-way Spectral Clustering algorithm was applied to the dataset for outlier detection, and the following outliers were identified within the clusters:

- **Cluster 0:** {3, 37, 59, 68, 82, 206, 232, 249, 251, 309, 314, 361, 408, 432, 527, 533, 689, 720, 1065}
- **Cluster 1:** {608, 899}

These results indicate that there are 19 outliers in Cluster 0 and 2 outliers in Cluster 1. According to the algorithm and the chosen threshold, the other clusters do not have any data points that are considered outliers.