IE506: Machine Learning - Principles and Techniques **Jan-April 2024**

End-term Project Report

Outlier Detection and Robust PCA Using a Convex Measure of Innovation

Team Name: MLTorch *Team Members: 23m1515 Akansh Verma*
23m1521 Ashish Kumar Uchadiya

Contents

Abstract

This project report consist of the details on our project which frames outlier detection as a robust PCA problem.This algorithm is called as iSearch. We modelled our objective as a convex optimization problem whose optimal value gives the innovation corresponding to data points. Outliers carry large innovation compared to inliers.iSearch ranks the data points based on their values of innovation

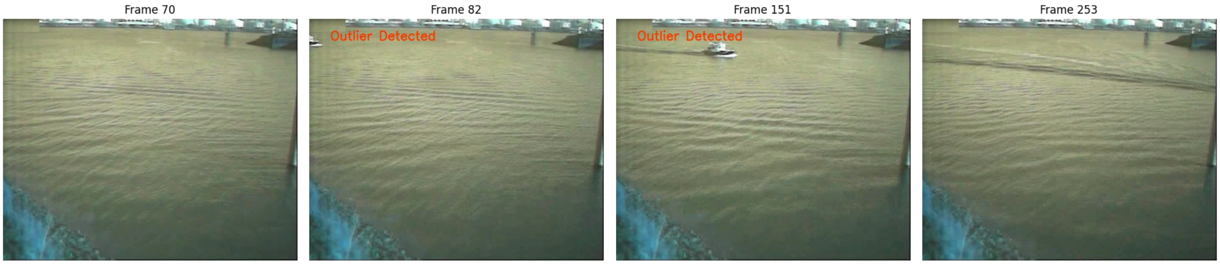

Figure 1: Frames of output video generated after applying algorithm on it. Algorithm has detected and labeled the frames with boat (outlier) as outliers detected.

1 Introduction

The project focuses on robust Principal Component Analysis (PCA) [8] [9] [10] [1] [3] [4] [12] [2] [11] and outlier detection using a strong algorithm called Innovation Search (iSearch).Our goal is to identify outliers, which are data points that do not belongs to the low-dimensional structure formed by the majority of the data.iSearch ranks data points based on their values of innovation, which measures the extent to which a data point deviates from the others.The project addresses different scenarios, including randomly distributed outliers, clustered outliers, and linearly dependent outliers

2 Literature Survey

- **Author's Work on Coherence Values:**
  - Computes Coherence Values for all data points to rank them.
  - Uses inner product between the column and the rest of the data points to measure resemblance.
  - Coherence value for each data column measures resemblance between the column and the rest of the data.
  - Focuses on ranking data points based on coherence values.
- **Work by Authors on iPursuit:**
  - Presents a subspace clustering method.
  - The optimization problem used finds a direction in the span of the data such that it is orthogonal to the maximum number of data points.
  - Introduces two frameworks:
    - First framework: an iterative method that finds the subspaces consecutively by solving a series of simple linear optimization problems.
    - Second framework: integrates iPursuit with spectral clustering to yield a new variant of spectral-clustering-based algorithms.
- **Work by Authors on Direction Search Based Subspace Clustering (DSC):**
  - Presents a new spectral-clustering-based approach called Direction search based Subspace Clustering (DSC) for subspace clustering.
  - Utilizes a convex program for optimal direction search, finding an optimal direction for each data point that has minimum projection on other data points and non-vanishing projection on itself.

3 Methods and Approaches

- This paper frames outlier detection as a robust Principal Component Analysis (PCA) problem.
- It primarily focuses on the column-wise model, where outliers are a subset of columns in the dataset.
- The aim is to detect outliers in high-dimensional datasets where traditional methods struggle.
- Traditional methods face challenges due to various factors such as:
  - Structured outlier patterns,
  - Clustering of inliers, and
  - Linear dependencies among outliers.
- Outliers may exhibit low-dimensional patterns different from the majority of the data.
- Inliers may form clusters within the data, making it essential to accurately capture the underlying structure of each cluster.
- In such cases, the proposed method, named iSearch, outperforms most of the existing methods.

3.1 Algorithm Overview

The algorithm used by iSearch consists of four main steps:

1. **Data Preprocessing**
2. **Direction Search**
3. **Computing the Innovation Values**
4. **Building Basis**

Let's delve into each step:

3.1.1 Data Preprocessing

1. Define $\mathbf{D} \in \mathbb{R}^{M_1 \times M_2}$ as the matrix of first r_d left singular vectors of $\mathbf{D}$ where r_d is the number of non-zero singular values. So $\mathbf{D} = \mathbf{Q}^T\mathbf{D}$.

2. Normalize the L_2-norm of columns of $\mathbf{D}$, i.e., set $\|\mathbf{d}_i\|_2 = 1$ for all $1 \leq i \leq M_2$.

3.1.2 Direction Search

Define $\mathbf{C}^* \in \mathbb{R}^{r_d \times M_2}$ such that $\mathbf{c}_i^* \in \mathbb{R}^{r_d \times 1}$ is the optimal point of

$$\min_{\mathbf{c}} \|\mathbf{c}^T\mathbf{D}\|_1 \quad \text{subject to} \quad \mathbf{c}^T\mathbf{d}_i = 1$$

or define $\mathbf{C}^* \in \mathbb{R}^{r_d \times M_2}$ as the optimal point of

$$\min_{\mathbf{C}} \| \left(\mathbf{C}^T\mathbf{D}\right)^T \|_1 \quad \text{subject to} \quad \operatorname{diag}\left(\mathbf{C}^T\mathbf{D}\right) = \mathbf{1}$$

$\mathbf{c}^T\mathbf{D}$ is the objective function we aim to minimize during the direction search step. By finding the direction vector $\mathbf{c}$ that minimizes this projection, we are identifying a direction that captures the most essential information in the data while minimizing the impact of outliers.

3.1.3 Computing the Innovation Values

Define vector $\mathbf{x} \in \mathbb{R}^{M_2 \times 1}$ such that $\mathbf{x}(i) = 1/\left\|\mathbf{D}^T\mathbf{c}_i^*\right\|_1$.
$\mathbf{D}^\top\mathbf{c}_i^*$: Visually, this projection captures how well each data point aligns with the optimal direction vector $\mathbf{c}_i^*$.

3.1.4 Building Basis

Construct matrix $\mathbf{Y}$ from the columns of $\mathbf{D}$ corresponding to the smallest elements of $\mathbf{x}$ such that they span an r-dimensional subspace.
Output: The column-space of $\mathbf{Y}$ is the identified subspace.

3.2 Work Done

The proposed approach is illustrated using a synthetic numerical example. Let's suppose $\mathbf{D} \in \mathbb{R}^{20 \times 250}, n_i = 200, n_o = 50$, and $r = 3$. Assume that $\mathbf{D}$ follows Assumption 1.

Assumption 1. The columns of $\mathbf{A}$ are drawn uniformly at random from $\mathcal{U} \cap \mathbb{S}^{M_1-1}$. The columns of $\mathbf{B}$ are drawn uniformly at random from $\mathbb{S}^{M_1-1}$. To simplify the exposition and notation, it is assumed without loss of generality that $\mathbf{T}$ in Data Model 1 is the identity matrix, i.e, $\mathbf{D} = \left[\begin{array}{cc} \mathbf{B} & \mathbf{A} \end{array}\right]$.
Data looks like this: fig.[2]

Assumption 2. The columns of $\mathbf{A}$ are drawn uniformly at random from $\mathcal{U} \cap \mathbb{S}^{M_1-1}$. The columns of $\mathbf{B}$ are drawn uniformly at random from $\mathbb{S}^{M_1-1}$. To simplify the exposition and notation, it is assumed without loss of generality that $\mathbf{T}$ in Data Model 1 is the identity matrix, i.e, $\mathbf{D} = \left[\begin{array}{c} \mathbf{B_1}\ \mathbf{A_1}\ \mathbf{B_2}\ \mathbf{A_2}\mathbf{B_3} \end{array}\right]$, where $\mathbf{A_1} \in \mathbb{R}^{20 \times 100}$, $\mathbf{A_2} \in \mathbb{R}^{20 \times 100}, \mathbf{B_1} \in \mathbb{R}^{20 \times 50}, \mathbf{B_2} \in \mathbb{R}^{20 \times 40}, \mathbf{B_3} \in \mathbb{R}^{20 \times 30}$
Data looks like this: fig.[3]

Assumption 3. After suffling the dataset from assumption 2. Data looks like this: fig.[3]

4 Data set Details

4.1 Synthetic data

4.1.1 With one cluster fig.[2]

$\mathbf{D} \in \mathbb{R}^{20 \times 250}, n_i = 200, n_o = 50$, and $r = 3$.

4.1.2 With three cluster fig.[3]

$\mathbf{D} = \left[\begin{array}{c} \mathbf{B_1}\ \mathbf{A_1}\ \mathbf{B_2}\ \mathbf{A_2}\mathbf{B_3} \end{array}\right]$, where $\mathbf{A_1} \in \mathbb{R}^{20 \times 100}$, $\mathbf{A_2} \in \mathbb{R}^{20 \times 100}, \mathbf{B_1} \in \mathbb{R}^{20 \times 50}, \mathbf{B_2} \in \mathbb{R}^{20 \times 40}, \mathbf{B_3} \in \mathbb{R}^{20 \times 30}$

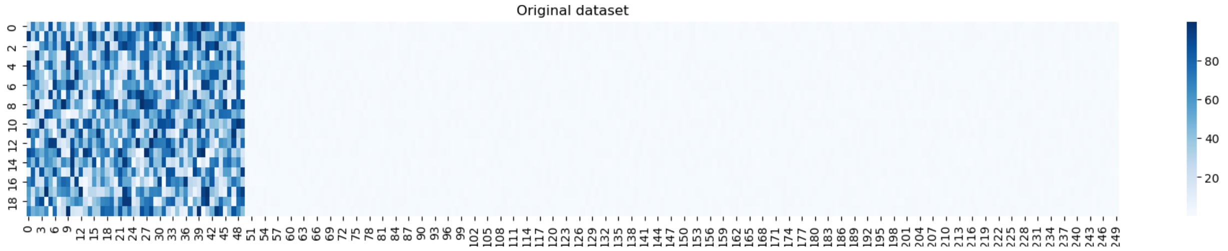

Figure 2: Heatmap of Dataset 1 following assumption 1. Dataset with one cluster of outliers.

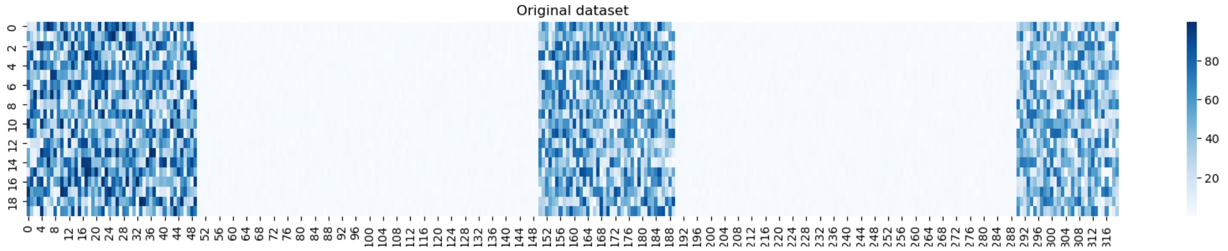

Figure 3: Heatmap of Dataset 2 following assumption 3. Dataset with 3 cluster of outliers.

4.1.3 With suffled features

Data looks like this: fig.[4]

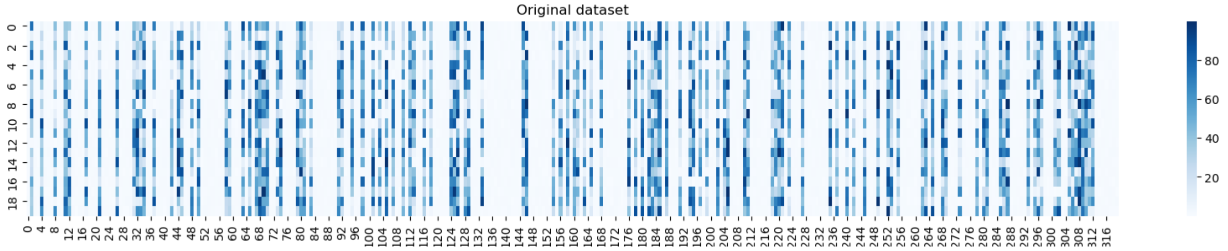

Figure 4: Heatmap of Dataset 3 following assumption 3. Dataset with 3 shuffled cluster of outliers.

4.2 Pre-processing Technique

1. Using SVD
2. By normalizing ℓ_2-norm of the columns of $\mathbf{D}$, i.e., set $\mathbf{d}_i$ equal to $\mathbf{d}_i/\left\|\mathbf{d}_i\right\|_2$ for all $1 \leq i \leq M_2$.
3. For video data set we did SVD then apply normalization and then resized the frame.

4.3 Data procurement

1. Generated inliers random samples form uniform distribution in range (0,1)
2. Generated outliers random samples form uniform distribution in range (0,1) and then scaled by a scaler to give the sense of outliers.

5 Experiments

We performed the experiment on the following datasets:

- With one cluster.
- With three clusters.

- With shuffled feeatures.

After solving for $\mathbf{C}^*$ for datset D_1 and projection the D_1 on $\mathbf{C}^*$ we get the plot [5]

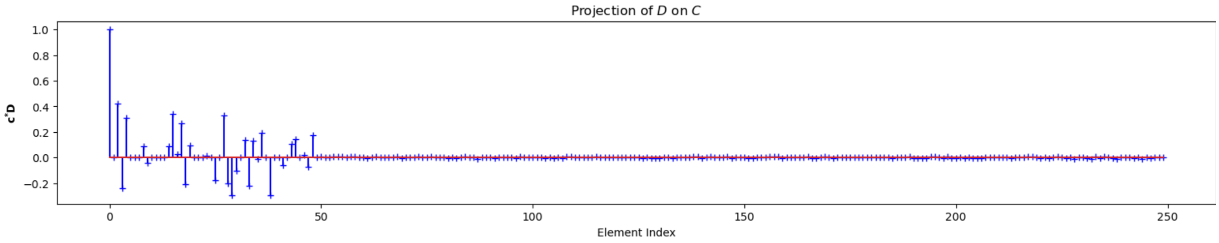

Figure 5: Projection of dataset $\mathbf{D_1}$ on $\mathbf{C}^*$ obtained from solver after solving from data 1.

After solving for $\mathbf{C}^*$ for dataset D_2 and projection the D_2 on $\mathbf{C}^*$ we get the plot [6]

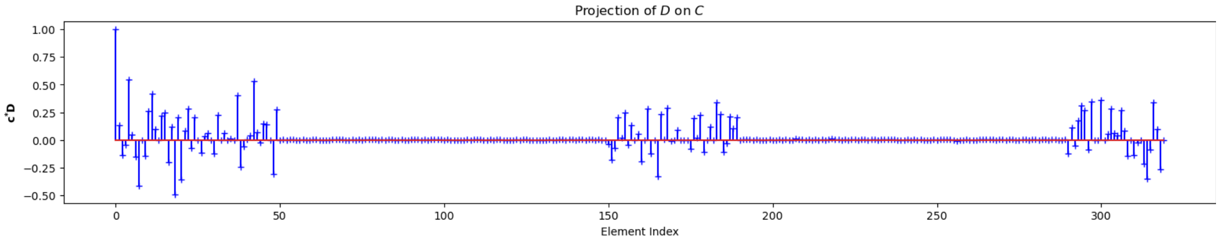

Figure 6: Projection of dataset $\mathbf{D_2}$ on $\mathbf{C}^*$ obtained from solver after solving from data 2.

After solving for $\mathbf{C}^*$ for dataset D_3 and projection the D_3 on $\mathbf{C}^*$ we get the plot [7]

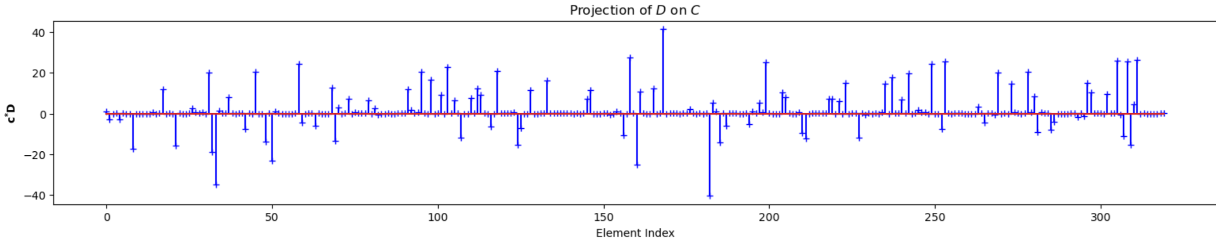

Figure 7: Projection of dataset $\mathbf{D_3}$ on $\mathbf{C}^*$ obtained from solver after solving from data 3.

It can be seen that the projection of outlier are more then inliers on all the three dataset $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$. Once we get the values of $\mathbb{C}^*$ we solve for innovation values of features for each dataset, the plot of innovation values for datasets are in Figure [8], [9] and [10] respectively.

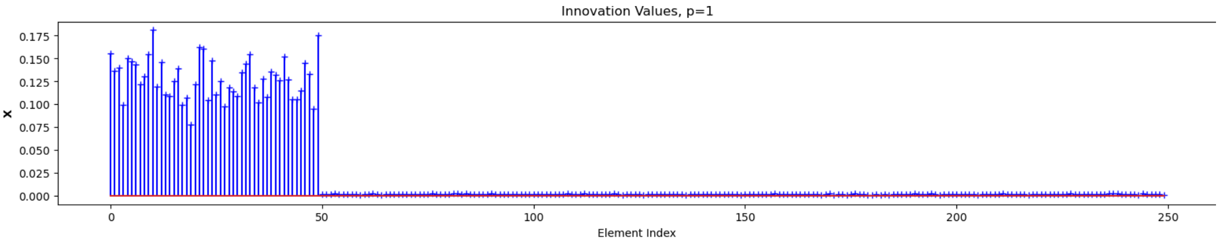

Figure 8: Innovation values of each feature (column) for $\mathbf{D_1}$ (single outlier cluster).

The sudden decreasing trend of innovation values can be seen for $\mathbf{D_1}$ and $\mathbf{D_2}$ corresponding to the outliers clusters, and once the data is scuffled the innovations get higher at the indices corresponding to outliers.

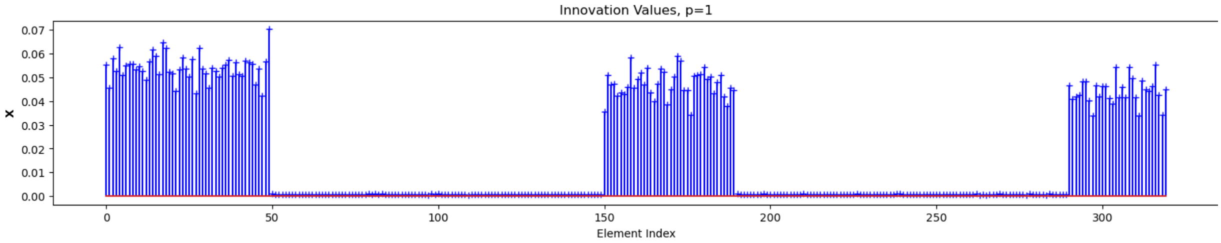

Figure 9: Innovation values of each feature (column) for $\mathbf{D_2}$ (3 outlier cluster).

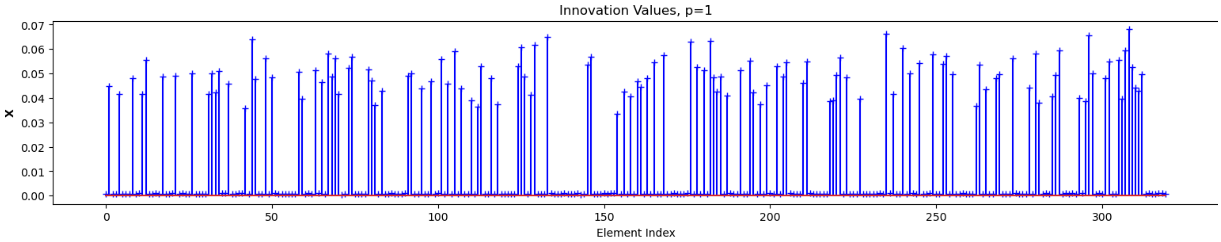

Figure 10: Innovation values of each feature (column) for $\mathbf{D_3}$ (shuffled outliers).

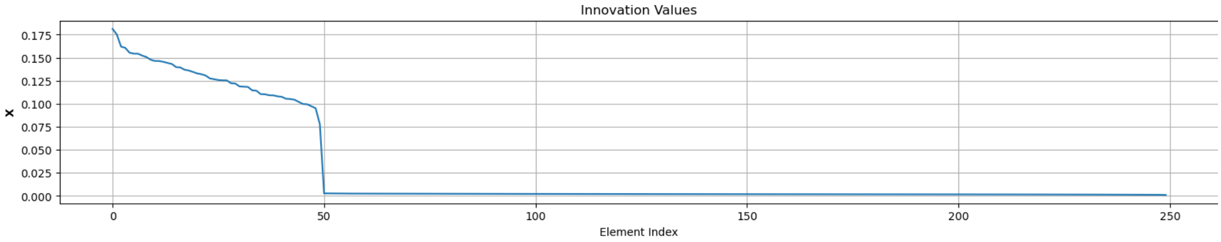

Figure 11: Sorted Innovation values of each feature (column) for $\mathbf{D_1}$ (single outlier cluster).

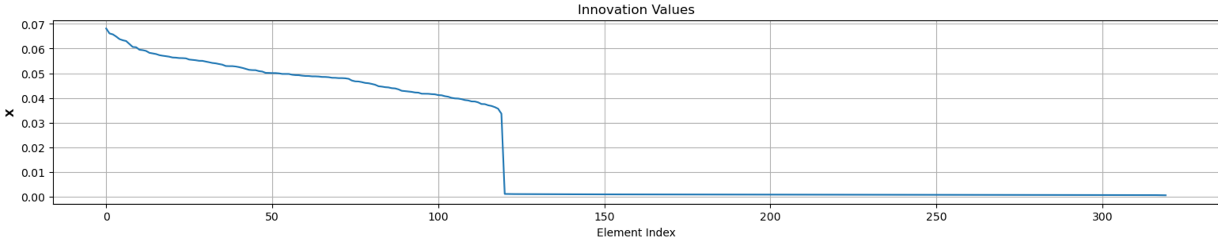

Figure 12: Sorted Innovation values of each feature (column) for $\mathbf{D_2}$ (3 outlier cluster).

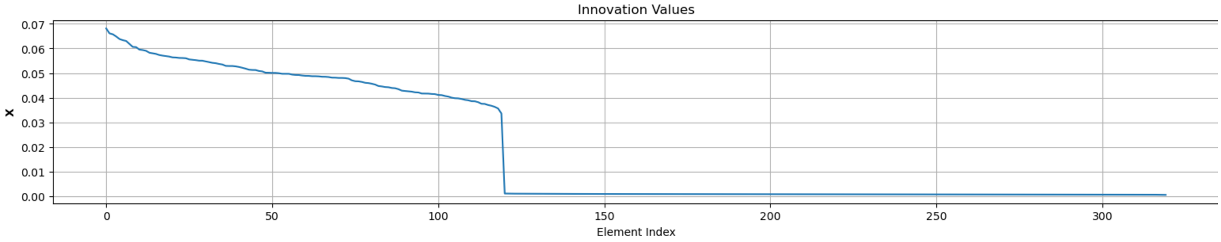

Figure 13: Sorted Innovation values of each feature (column) for $\mathbf{D_3}$ (shuffled outliers).

If we plot the decreasing trend of innovation values for datasets are in Figure [11], [11] and [11] respectively.
From the sorted innovation values plots we can separated the outliers features from our original dataset.

The columns having high innovation values can be removed by using the indices of high innovation values.

5.1 Algorithm Used

Experiment on Synthetic Dataset with One Cluster

- We generated the dataset of the form $D = [B(A+N)]^T$, following assumption 1, assumption 2, and assumption 3:
- The heatmap for datsets are shown in Figure [2], [3] and [4] respectively.
- Data pre-processing was performed using SVD if required:
- After preprocessing, we used an ADMM solver to solve our optimization equation to find C^*.
- Finally, we computed the innovation value.Figure 5.

6 Results

We have applied our algorithm to several datasets that are described above, and the results obtained are shown below:
Once the outliers are removed our dataset will look like Figure [14], [15] and [16] respectively.

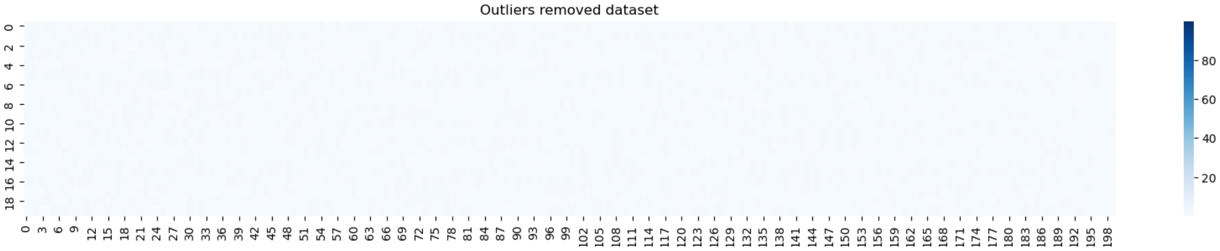

Figure 14: Heatmap of Dataset 1 after removal of outliers features.

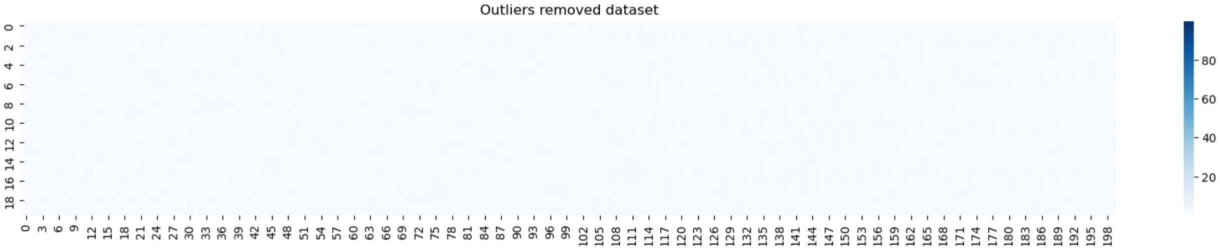

Figure 15: Heatmap of Dataset 2 after removal of outliers features.

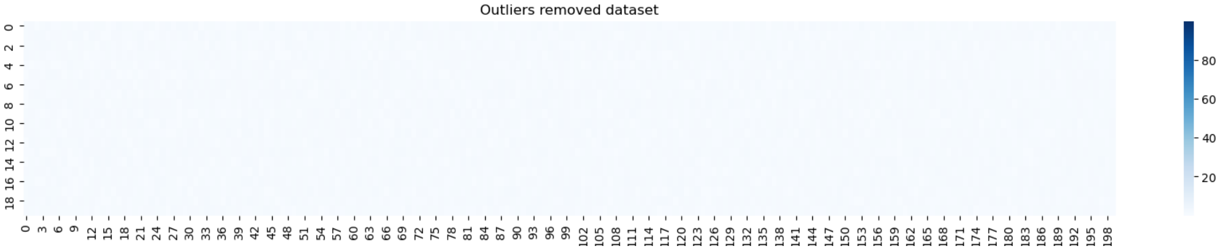

Figure 16: Heatmap of Dataset 3 after removal of outliers features.

I can be seen that the values having higher vales are compalately removed from the data and can be seen in the Figures [14], [15] and [16] respectively.

7 Work Done After Mid Term

Initially we applied the algorithm on synthetic dataset 1 (with 1 outliers cluster), dataset 2 (with 3 outliers clusters) and dataset 3 (with shuffled clusters). Later we applied the algorithm on **video dataset**.

7.1 Dataset Collection

1. Hopkins 115 dataset
2. Several self made videos.
3. Some other videos collected from internet.

7.2 Data Preprocessing

1. Converted frames of videos into gray scale (1-channel).
2. Applied **PCA** to reduce the 90% of dimensions (columns), only used 10% of dimension (columns) for calculations.

7.3 Algorithm Input

1. It takes input frame (reduced in dimension) one at a time preprocessed from above step.
2. Applied the optimization algorithm on this frame.
3. Return the innovation values for all columns in that frame.
4. Repeating above step for all frames and calculating innovations values for each columns.
5. At this point we have (number of frames × number of columns) in each frame innovation values.

7.4 Z-Score Calculation

1. Standardizing the innovation values of each frames $\mathbf{Z_i} = \frac{x_i - \mu}{\sigma}$, where μ and σ are the mean and standard deviation of innovation values of a particular frames.
2. Repeating point 1 for all frames.
3. Taking maximum of standardized z values as a **Z- score** of that particular frame.
4. These Z-scores of all frames can be used to compare one frame with other.

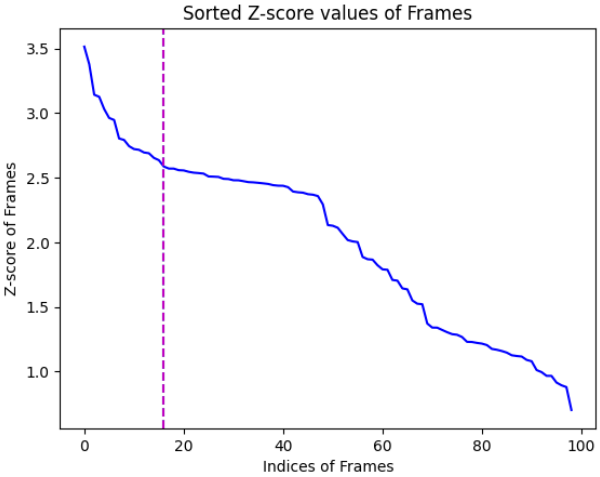

Figure 17: Plotted Z-score values of all frames in descending order and applied kneed algorithm

7.5 Z-Score Sorting and finding elbow of curve

1. The Z-scores obtained for all frames from above step will be sorted in decreasing sense.
2. Applying **kneed Algorithm** to find the elbow of the Z-score curve.
3. The **kneed Algorithm** will return the threshold for detecting outliers frames.
4. The x-axis values before the threshold calculated above are the indices of outliers frames.

7.6 Output Video Generation

1. Frames were taken in sequence from the video.
2. Marked as outlier based on the threshold.
3. Fed to videowriter (class of cv2 library) which give a output video file.

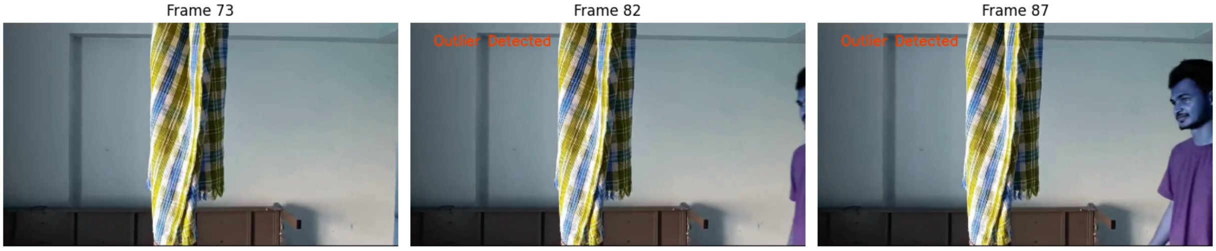

Figure 18: Frames of output video generated after applying algorithm on it. Algorithm has detected and labeled the frames with boy (outlier) as outliers detected.

8 New Ideas Proposed

1. Introduction of **Z-score** to compare the frames of video.
2. Use of **kneed Algorithm** to automate the for Z-score threshold finding procedure of outliers frames.

9 Conclusion

- Successfully detected outlier in synthetic dataset.
- Successfully detected outlier in the video dataset.
- Proposed algorithm is quite robust and can be applied to detect outlier in many cases.
- Proposed Algorithm takes time for large video dataset.

References

[1] Steve Brunton. Robust principal component analysis (rpca). Online Video, 2024. `https://www.youtube.com/watch?v=yDpz0PqULXQ&t=21s`.

[2] Herman Kamper. Data414 introduction to machine learning. *Kamper's Website*.

[3] Herman Kamper. Pca 1 - introduction. Online Video, 2024. `https://www.youtube.com/playlist?list=PLmZlBIcArwhMfNuMBg4XR-YQ0QIqdHCrl`.

[4] The Glowing Python. Pca and image compression with numpy. *Glowing Python Blog*, 2011.

[5] Mostafa Rahmani and George Atia. Innovation pursuit: A new approach to the subspace clustering problem. In *International conference on machine learning*, pages 2874–2882. PMLR, 2017.

[6] Mostafa Rahmani and George K Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. volume 65, pages 6260–6275. IEEE, 2017.

[7] Mostafa Rahmani and Ping Li. Outlier detection and robust pca using a convex measure of innovation. volume 32, 2019.

[8] Nitish Singh. Principal component analysis (pca) - part 1 - geometric intuition. Online Video, 2024. `https://youtu.be/iRbsBi5W0-c?si=HMIw7VAcwwptB27l`.

[9] Nitish Singh. Principal component analysis (pca) - part 2 - problem formulation and step by step solution. Online Video, 2024. `https://www.youtube.com/watch?v=tXXnxjj2wM4`.

[10] Nitish Singh. Principal component analysis (pca) - part 3 - code example and visualization. Online Video, 2024. `https://www.youtube.com/watch?v=tofVCUDrg4M`.

[11] Vincent Spruyt. Eigen decomposition of a covariance matrix. *Vision Dummy*, 2014.

[12] StackExchange. Anomaly detection using pca reconstruction error. *StackExchange*.