Mid-Term Course Project Presentation

OUTLIER DETECTION AND ROBUST PCA USING A CONVEX MEASURE OF INNOVATION

Authors : Mostafa Rahmani, Ping Li

Published in : 33rd Conference on NeurIPS 2019, Vancouver, Canada

COURSE DETAILS

Course Title : **IE 506 : Machine Learning: Principles and Techniques, Spring 2023**
Instructor : **Prof. P Balamurugan**

THIS WORK IS DONE AS PART OF IE 506 COURSE PROJECT

TEAM DETAILS

Team : **MLTorch**
Member : **Ashish Kumar Uchadiya 22M1521**
Member : **Akansh Verma 22M1515**
TA Incharge : **Krushna Salunke & Vivek Seth**

</ Presentation Outline />

- {01} Problem **DESCRIPTION**
- {02} Past **WORK DONE** & Methods **COMPETING**
- {03} Detail of **PROBLEM CONSIDERED**
- {04} Proposed **SOLUTION APPROACH**
- {05} Experiments **WITH DATASET**
- {06} Computational **FRAMEWORK & HARDWARE**
- {07} Work Done **STATUS**
- {08} Future **PLANS**

</ Problem DESCRIPTION />

The paper frames outlier detection as a robust PCA problem

MOTIVATION OF PROBLEM

In the challenging scenarios in which the outliers are close to each other or they are close to the span of the inliers, iSearch is shown to outperform most of the existing methods.

FOCUS

The paper primarily focuses on the column-wise model, where outliers are a subset of columns in the dataset.

</ Past **WORK DONE** & Methods **COMPETING** />

COHERENCE PURSUIT[3] :

- Inlier is likely to have strong mutual coherence (correlation) with a large number of data points.
- By contrast, an outlier is unlikely to bear strong resemblance to a large number of data points
- Computes Coherence Values for all data points.
- Inner product between the column and the rest of the data points to measure resemblance.
- Ranks data columns points based on coherence values.

</ Detail of **PROBLEM CONSIDERED** />

- Detecting outliers in high-dimensional datasets with 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 form clusters within the data, making it essential to accurately capture the underlying structure of each cluster.

</ Proposed SOLUTION APPROACH />

Subspace Recovery Using iSearch :

The algorithm consist of 4 steps:

1. **Data preprocessing**
2. **Direction search**
3. **Computing the innovation value**
4. **Building basis**

</ Proposed **SOLUTION APPROACH** />

I. Data Preprocessing

1. The input is data matrix $\mathbf{D} \in \mathbb{R}^{M_1 \times M_2}$
   1.1. Define $\mathbf{Q} \in \mathbb{R}^{M_1 \times r_d}$ as the matrix of first r_d left singular vectors of D where r_d is the number of non zero singular values. So D = $Q^T D$
   1.2. Normalize the L2 - norm of columns of D, i.e. set $d_i = d_i / \|d_i\|_2$ for all $1 \leq i \leq M_2$

II. Direction Search

$C^T D$ is the objective function we aim to minimize during the direction search step. By finding the direction vector 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.

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

$$\min_{c} \left\| \mathbf{c}^\top \mathbf{d} \right\|_1 \quad \text{subject to} \quad \mathbf{c}^\top \mathbf{d}_i = 1$$

Or define $\mathbf{C}^* \in \mathbb{R}^{r_d \times M_2}$ as a optimal point

$$\min_{c} \left\| (\mathbf{C}^\top \mathbf{D})^\top \right\|_1 \quad \text{Subject to} \quad \operatorname{diag}\left(\mathbf{C}^\top \mathbf{D}\right) = 1 \qquad [1]$$

</ Proposed **SOLUTION APPROACH** />

III. Computing the Innovation Values

Define vector $\mathbf{x} \in \mathbb{R}^{M_2 \times 1}$ such that $\mathbf{x}(i) = \dfrac{1}{||\mathbf{D}^\top \mathbf{c}_i^*||_1}$

$D^T c^*_i$: This projection captures how well each data point aligns with the optimal direction vector c^*_i.

IV. Building Basis

Construct matrix Y from the columns of corresponding to the smallest elements of x such that they span an r-dimensional subspace.

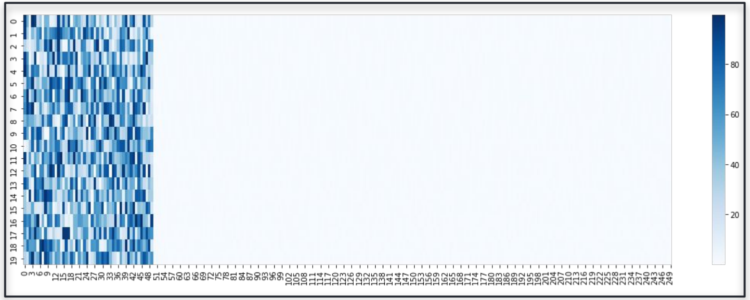
</ Experiments **WITH DATASET** />

DATA 1: Single cluster outliers

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

$$\mathbf{D} = [\mathbf{B}(\mathbf{A} + \mathbf{N})]$$

Where $\mathbf{A} \in \mathbb{R}^{m\times n_i}$, $\mathbf{B} \in \mathbb{R}^{m\times n_o}$



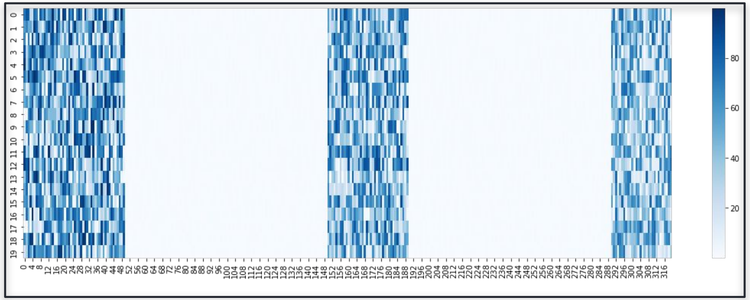
Heatmap of Dataset 1, Outliers (B) cluster in blue, Inliers (A) cluster in white.

DATA 2: Three cluster outliers

$$\mathbf{D} = [\mathbf{B}_1 + \mathbf{A}_1 + \mathbf{B}_2 + \mathbf{A}_2 + \mathbf{b}_3]$$

Where $\mathbf{A}_1 \in \mathbb{R}^{20\times 100}$, $\mathbf{A}_2 \in \mathbb{R}^{20\times 100}$

$\mathbf{B}_1 \in \mathbb{R}^{20\times 50}$, $\mathbf{B}_2 \in \mathbb{R}^{20\times 40}$ & $\mathbf{B}_3 \in \mathbb{R}^{20\times 30}$



Heatmap of Dataset 2, Outliers (B_1, B_2 and B_3) clusters in blue, Inliers (A_1 and A_2) clusters in white.

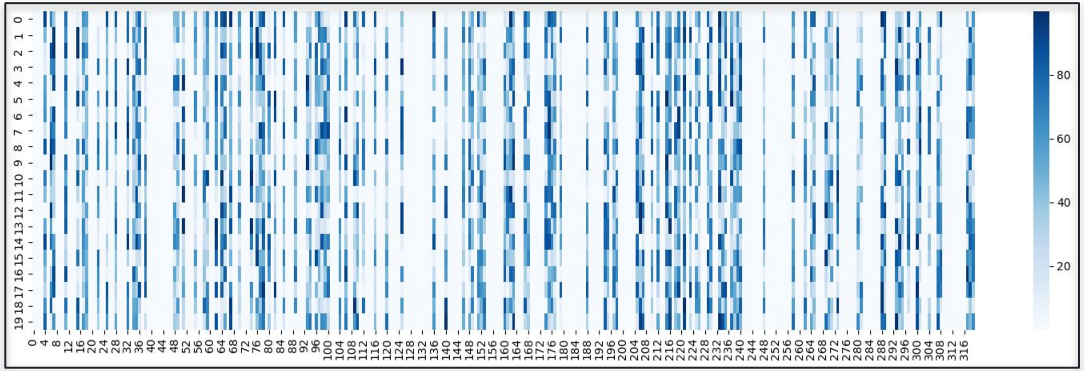
</ Experiments **WITH DATASET** />

DATA 3: Shuffled outliers

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

$$\mathbf{D} = [\mathbf{B}(\mathbf{A} + \mathbf{N})]$$

Where $\mathbf{A} \in \mathbb{R}^{m\times n_i}$, $\mathbf{B} \in \mathbb{R}^{m\times n_o}$



Heatmap of Dataset 3, Randomly disturbed features.

</ Computational FRAMEWORK & HARDWARE USED />

SOLVERS

1. **SCS** : SCS (Splitting Conic Solver) solver from CVXPY (Convex Optimization in Python) which uses ADMM (Alternating Direction Method of Multipliers) to solve our constrained optimization (minimization) problem.
2. **ECOS** : ECOS (Embedded Conic Solver) solver from CVXPY for large size dataset.
3. **LBFGS** : Limited-memory BFGS is a popular optimization algorithm particularly well-suited for problems with large numbers of parameters.
4. **CUPY** : It is a GPU (CUDA) variant of NumPy, for faster matrices computations.
5. **PyTorch** : Used for using LBFGS optimizer and GPU acceleration.

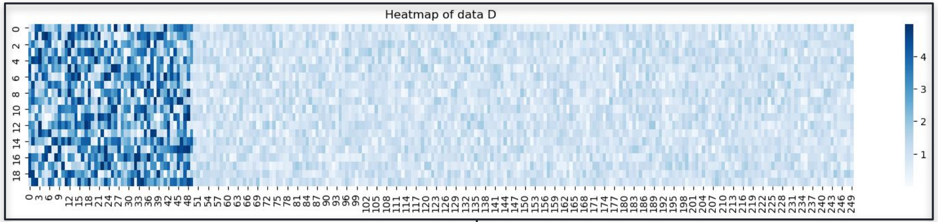
LANGUAGES & LIBRARIES: Python, NumPy, CuPy, OpenCV.

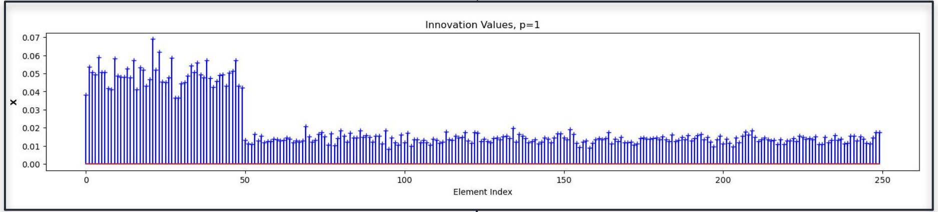
HARDWARE

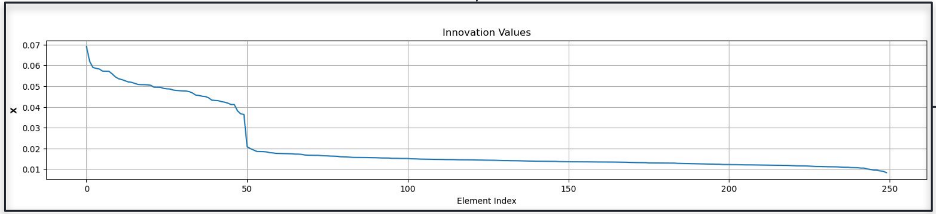
1. **CPU** : For processing small dataset (Data 1 and Data 2) in NumPy and CuPy.
2. **GPU** : For processing large dataset, image and video (Data 3) in PyTorch.

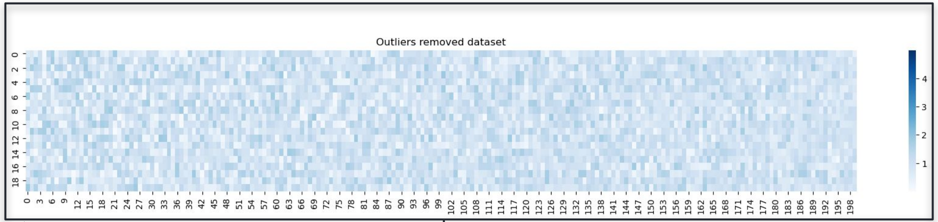
</ Work Done STATUS />

iSearch Algorithm for Data 1 :





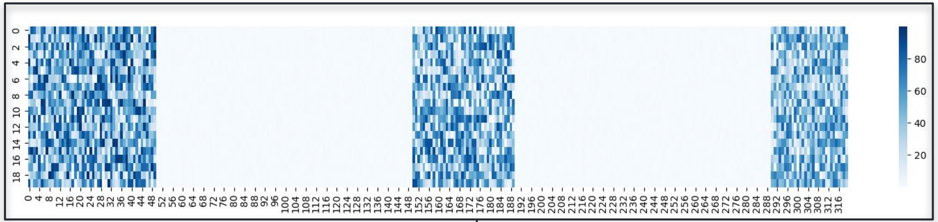


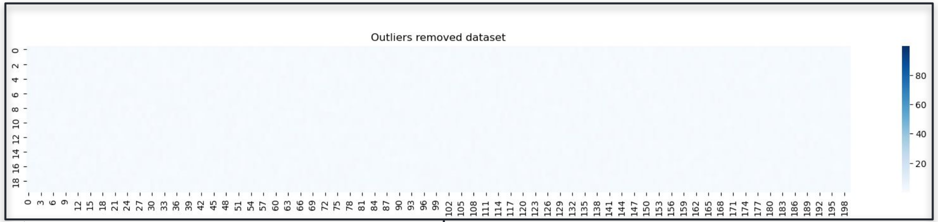


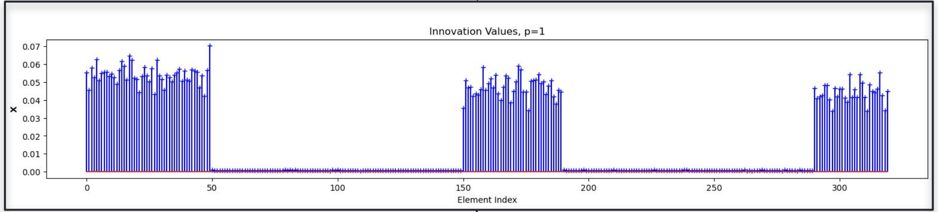
The cluster feature which we had taken initially as outliers (1st 50 columns) are completely identified and removed successfully.

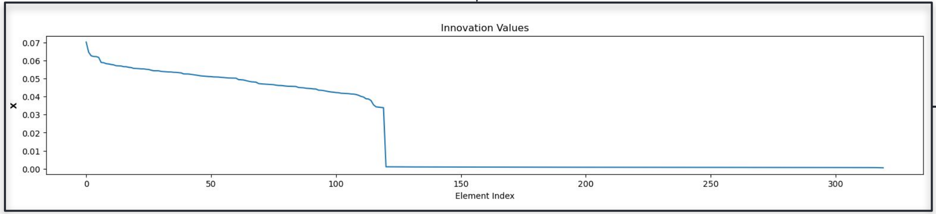
</ Work Done STATUS />

iSearch Algorithm for Data 2 :





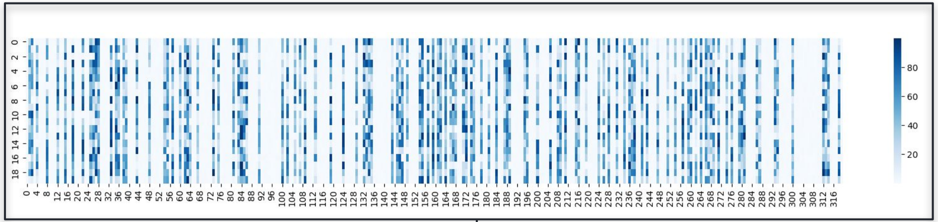


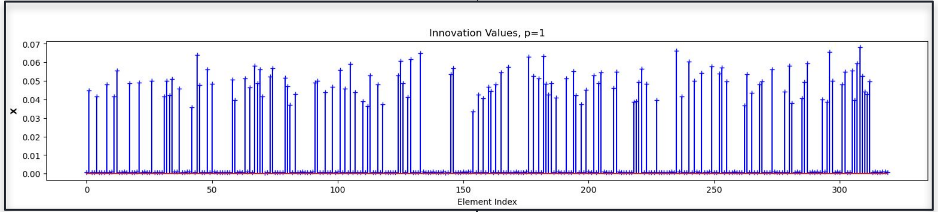


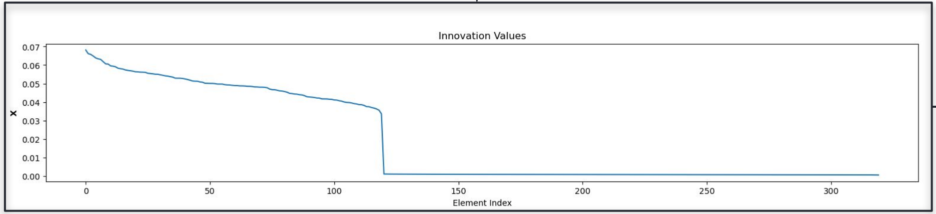
3 clusters of feature which we had taken initially as outliers are completely identified and removed successfully.

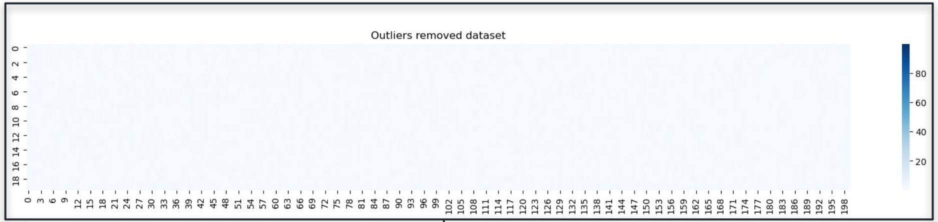
</ Work Done **STATUS** />

iSearch Algorithm for Data 3 :









This this the features are spread randomly are completely identified and removed successfully.

</ Future PLANS />

We Will try to apply iSearch algorithm for image sequence data sample.

THANK YOU

</ REFERENCES />

Paper References

[1] PAPER : **Outlier Detection and Robust PCA Using a Convex Measure of Innovation, NeurIPS 2019**
| Authors : Mostafa Rahmani, Ping Li | **Link :** **http://papers.nips.cc/paper/9568-outlier-detection-and-robust-pca-using-a-convex-measure-of-innovation.pdf**

[2] PAPER : **Innovation Pursuit: A New Approach to the Subspace Clustering Problem, ICML 2017**
| Authors : Mostafa Rahmani, George Atia | **Link :** **http://proceedings.mlr.press/v70/rahmani17b/rahmani17b.pdf**

[3] PAPER : **Coherence Pursuit: Fast, Simple, and Robust Subspace Recovery, ICML 2017**
| Authors : Mostafa Rahmani, George Atia | **Link :** **http://proceedings.mlr.press/v70/rahmani17a/rahmani17a.pdf**

[4] PAPER : **Outlier Detection and Data Clustering via Innovation Search, 30 Dec 2019**
| Authors : Mostafa Rahmani, Ping Li | **Link :** **https://arxiv.org/pdf/1912.12988v1.pdf**

[5] PAPER : **Outlier Detection and Data Clustering via Innovation Search, 30 Dec 2019**
| Authors : Mostafa Rahmani, George Atia | **Link :** **https://arxiv.org/pdf/1912.12988v1.pdf**

Article References

[1] ARTICLE : **Eigen decomposition of a covariance matrix**
| Editor : Vincent Spruyt | **Link :** **https://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/#Eigendecomposition_of_a_covariance_matrix**

[2] ARTICLE : **PCA and image compression with numpy**
| Editor : The Glowing Python | **Link :** **https://glowingpython.blogspot.com/2011/07/pca-and-image-compression-with-numpy.html**

[3] ARTICLE : **Anomaly detection using PCA reconstruction error**
| Editor : StackExchange | **Link :** **https://stats.stackexchange.com/questions/259806/anomaly-detection-using-pca-reconstruction-error**

[4] ARTICLE : **DatA414 Introduction to machine learning**
| Editor : Herman Kamper | **Link :** **https://www.kamperh.com/data414/**

</ REFERENCES />

Video Reference

[1] VIDEO : **Principal Component Analysis (PCA) _ Part 1 _ Geometric Intuition**
| Creator : Nitish Singh | **Link :** https://youtu.be/iRbsBi5W0-c?si=HMlw7VAcwwptB27l

[2] VIDEO : **Principal Component Analysis (PCA) | Part 2 | Problem Formulation and Step by Step Solution**
| Creator : Nitish Singh | **Link :** https://www.youtube.com/watch?v=tXXnxjj2wM4

[3] VIDEO : **Principal Component Analysis (PCA) | Part 3 | Code Example and Visualization**
| Creator : Nitish Singh | **Link :** https://www.youtube.com/watch?v=tofVCUDrg4M

[4] VIDEO : **Robust Principal Component Analysis (RPCA)**
| Creator : Steve Brunton | **Link :** https://www.youtube.com/watch?v=yDpz0PqULXQ&t=21s

[5] VIDEO : **PCA 1 - Introduction**
| Creator : Herman Kamper | **Link :** https://www.youtube.com/playlist?list=PLmZlBIcArwhMfNuMBg4XR-YQ0QIqdHCrl