End-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} Work Done BEFORE MID-TERM
- {03} Before Midterm RESULTS
- {04} Major Midterm COMMENTS
- {05} Work done AFTER END-TERM
- {06} Proposed NEW - IDEAS
- {07} Possible FUTURE WORK
- {08} CONCLUSIONS

</ Problem DESCRIPTION />

The paper frames outlier detection as a robust PCA problem

MOTIVATION OF PROBLEM

- **Challenging Scenarios**: The outliers are close to each other or they are close to the span of the inliers.
- **iSearch Performance**: iSearch is shown to outperform most of the existing methods in these challenging scenarios.

FOCUS

- **Primary Focus**: The paper primarily focuses on the column-wise model.
- **Outliers**: In this model, outliers are considered as a subset of columns in the dataset

</ Work done BEFORE MID-TERM />

ALGORITHM USED

- Data pre-processing
- 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

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

- Finding innovation value

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

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

- Building basis

</ Work done BEFORE MID-TERM />

UNDERSTANDING AND IMPLEMENTATION

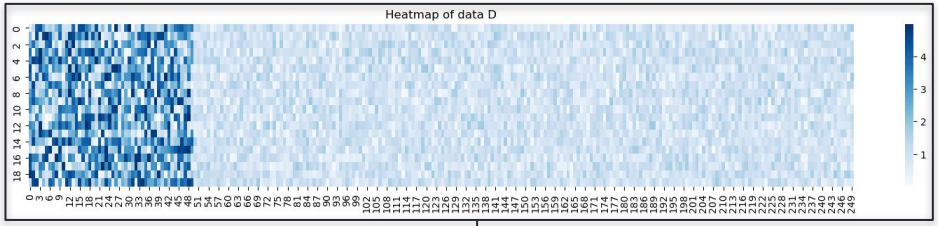
- Implementation of the proposed algorithm.
- Generation of a synthetic dataset.

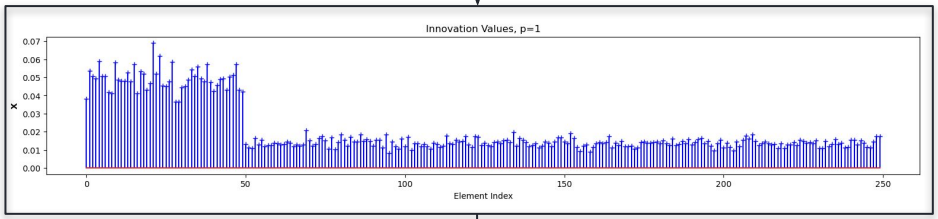
PROBLEM SOLVING AND RESULT

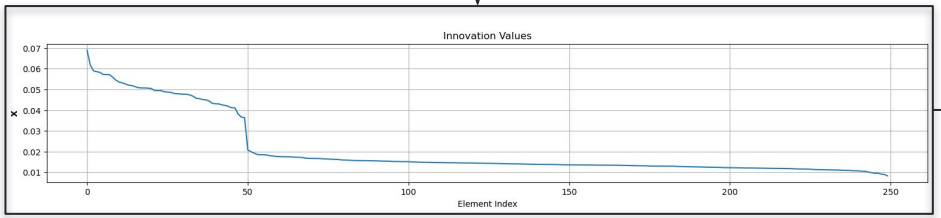
- Issue faced with the time taken by the initial solver.
- Discovery and implementation of the ECOS solver.
- Results obtained from the synthetic dataset with 50 outliers and 200 inliers.

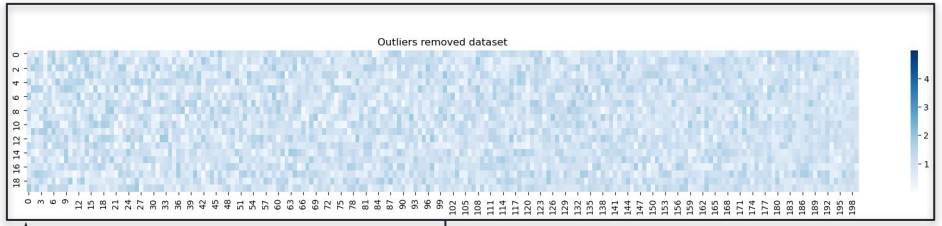
</ Before Midterm RESULTS />

iSearch Algorithm for Data 1 :





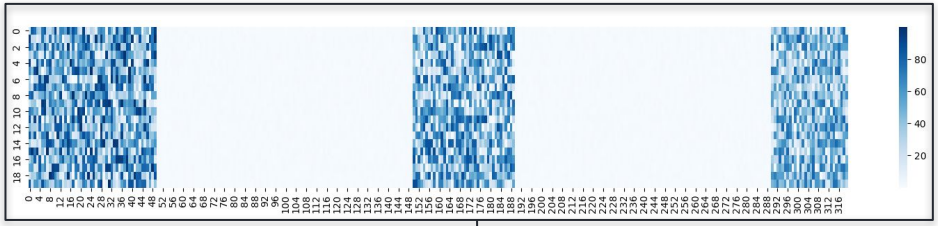


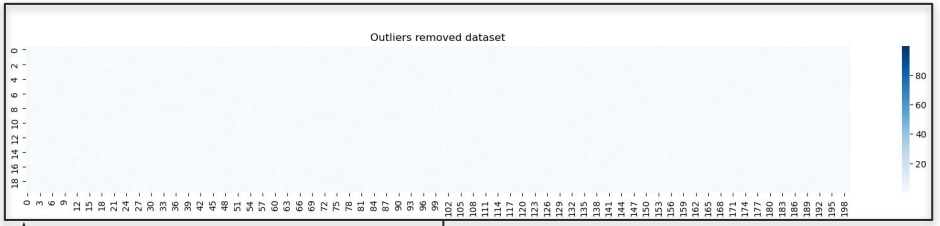


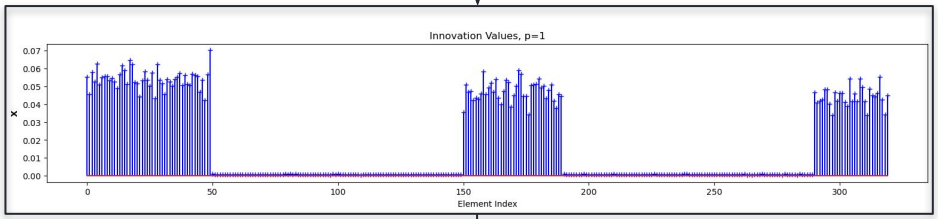
The cluster feature which we had taken initially as outliers (1st 50 columns) are completely identified and removed successfully.

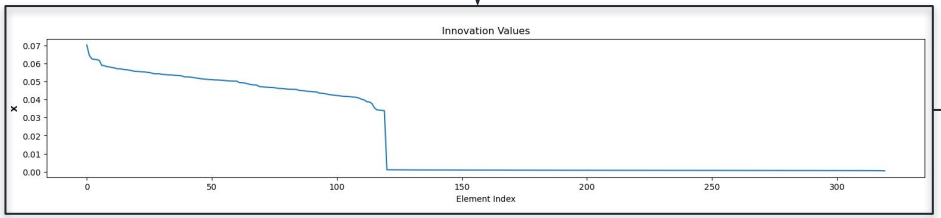
</ Before Midterm RESULTS />

iSearch Algorithm for Data 2 :





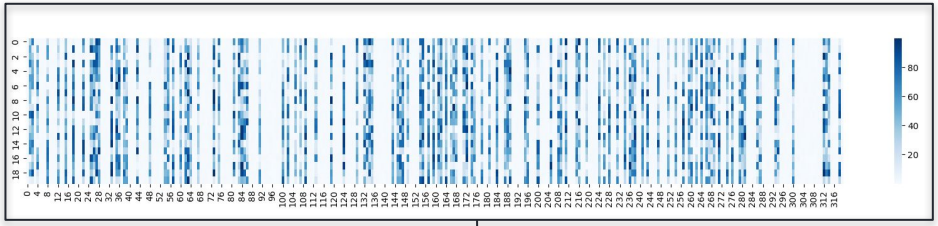


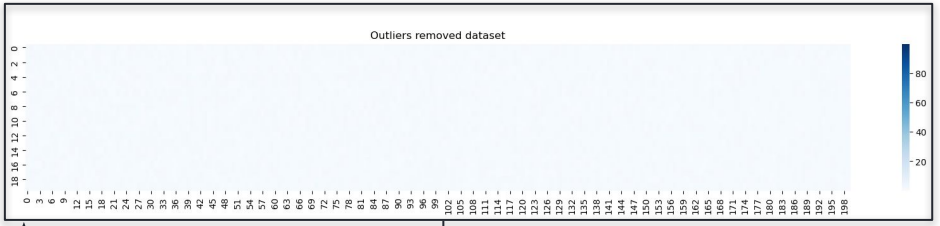


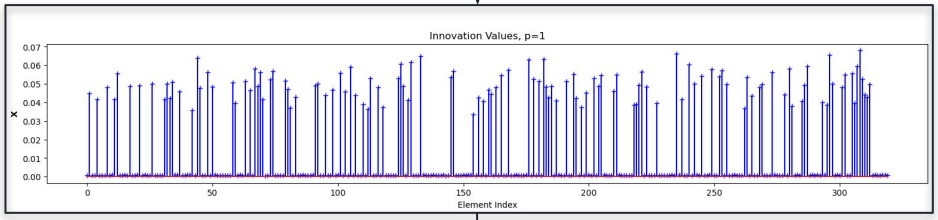
3 clusters of feature which we had taken initially as outliers are completely identified and removed successfully.

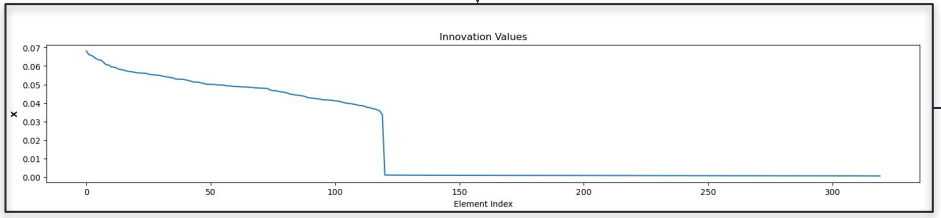
</ Before Midterm RESULTS />

iSearch Algorithm for Data 3 :









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

</ Major Midterm COMMENTS />

BY INSTRUCTOR

- **End Term Results:** The results on video data should be showcased by the end of the term.
- **Video Datasets:** The team must try working on some video datasets.

BY TAs

- **Presentation Format:** Crisp bullet points should be used instead of paragraphs in presentation slides.
- **Conclusion Slide:** A conclusion slide has not been added yet.
- **Final Review Expectations:** For the final review, experiments on video data are expected.

</ Work done AFTER END-TERM />

DATASET COLLECTION

- HOPKINS 155
- SELF MADE
- SOME OTHERS FROM INTERNET

PRE-PROCESSING

- Converted videos into grayscale (1-channel)
- Method of extracting frames from the videos

PCA APPLICATION

- Reduced the dimension upto 90%.
- Helped to run the algorithm faster .

ALGORITHM INPUT

- It take input a matrix (reduced in dimension).
- Applied the optimization algorithm
- Return the innovation value.

</ Work done AFTER END-TERM />

STANDARDIZATION

- Of innovation value of a frame.

$$Z = \frac{x - \mu}{\sigma}$$

Z - SCORE

- Taking maximum Z value of each frame.
- One value for each frame.

Z - SCORE SORTING

- Sorting the Z-scores of all frames.
- For detecting outliers frames.

KNEE FINDING

- Applied kneed algorithm to find knee points.
- Used knee point to decide the threshold.

</ Work done AFTER END-TERM />

OUTLIER FRAME DETECTION

- Process of identifying the outlier frames based on the threshold.
- Compare the threshold with max z-score of each frame
- Frame whose max z-score greater than threshold, declared as an outlier.

VIDEO GENERATION

- Frames were taken in sequence from the video
- Marked as outlier based on the threshold.
- Fed to videowriter.

</ Proposed NEW - IDEAS />

INTRODUCTION OF Z-SCORE

- For comparing frames in a video with each other.
- To automate the selection of the index corresponding to outlier.

USE OF KNEED ALGORITHM

- To find the optimal point till where the outlier exist in the sorted z-score array.

</ POSSIBLE FUTURE WORK/>

- There is a possibility of searching for some solver that take even more less time.
- Some changes in the proposed algorithm could be find which can increase the speed of computation.

</ 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.

</ Computational FRAMEWORK & HARDWARE USED />

SOLVERS

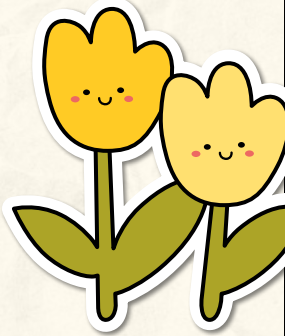
1. **SCS** : SCS (Splitting Conic Solver) solver from CVXPY (Convex Optimization in Python) which uses ADMM (Alternating Direction Method of Multipliers) to solve our constrained optimization (minimization) problem.
2. **ECOS** : ECOS (Embedded Conic Solver) solver from CVXPY for large size dataset.
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.

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-YQ0QlqdHCrl