Deep Augmented Robust Hankel Matrix Recovery

Erick Rodriguez, Faculty Advisor: Dr. HanQin Cai
University of Central Florida, Orlando, FL

Center for Research in Computer Vision
UCF UNIVERSITY OF CENTRAL FLORIDA

Overview

Deep unfolding approaches to learn parameters in algorithms have been employed with great success in several fields, including principal component analysis and robust matrix recovery. In the realm of Hankel matrix recovery problems, however, there is a lack of deep unfolding approaches that scale for time sensitive situations.

To address this, we propose DeepHankel, a novel algorithm that employs deep unfolding to address robust Hankel matrix recovery while being more scalable than other deep unfolding algorithms and computationally efficient than traditional approaches.

Introduction

In time sensitive signal processing applications, the problem of low-rank Hankel matrix recovery frequently appears. Given a vector of samples from an underlying signal **y** = [y_1, y_2, … , y_n], we have a corresponding Hankel matrix of the form:

$$\mathcal{G}(\mathbf{y}) = \mathbf{Y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n_2} \\ y_2 & y_3 & \cdots & y_{n_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n_1} & y_{n_1+1} & \cdots & y_n \end{bmatrix} \in \mathbb{C}^{n_1 \times n_2}$$

The structure of ***Y*** allows for certain matrix decompositions to reveal underlying information about the original signal. In practice, however, the original vector is only partially observed, corrupted by noise, and often ill-conditioned. With this in mind, we want to recover the underlying matrix ***Z*** and sparse outlier matrix ***S*** such that ***Y*** = ***Z*** + ***S***.

In related problems, deep unfolding methods such as CORONA [1] have shown the potential of creating a learnable algorithm for matrix recovery problems. However, these models typically compute a costly SVT at every step, hurting scalability. Our proposed algorithm, DeepHankel, is a Hankel matrix recovery algorithm that is both **learnable** and **scalable**.

Method

DeepHankel is based on a traditional algorithm for Hankel matrix recovery, titled Hankel Structured Newton-Like Descent (HSNLD) [2]. HSNLD updates ***S*** in a memory-efficient vector representation **s** through the following step, where **f** is the observed vector and Γ is the sparsification operator that keeps the largest $\gamma_k \alpha \rho$ magnitude entries to be outliers:

$$\mathbf{s}_{k+1} = \Gamma_{\gamma_k \alpha \rho} \left(\Pi_\Omega \left(\mathbf{f} - \mathbf{z}_{k+1} \right) \right) \tag{1}$$

HSNLD also stores matrices ***L, R*** $\in \mathbb{C}^{n \times r}$ such that ***Z*** = ***LR****, iteratively updating both through the Newton-like descent step:

$$\begin{bmatrix} \mathbf{L}_{k+1} \\ \mathbf{R}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_k - \eta \nabla_{\mathbf{L}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) (\mathbf{R}_k^* \mathbf{R}_k)^{-1} \\ \mathbf{R}_k - \eta \nabla_{\mathbf{R}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) (\mathbf{L}_k^* \mathbf{L}_k)^{-1} \end{bmatrix} \tag{2}$$

DeepHankel replaces the sparse vector update in (1) with the following, where T is a learnable outlier detection operator and ζ_k is a learned zeta parameter at iteration k:

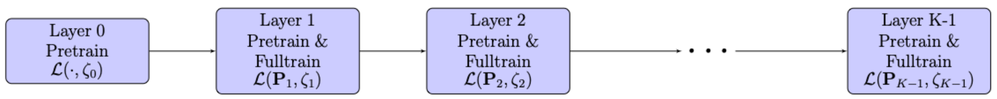
$$\mathbf{s}_{k+1} = \mathcal{T}_{\zeta_{k+1}} (\Pi_\Omega (\mathbf{f} - \mathbf{z}_{k+1})) \tag{3}$$

By [Theorem 1, 3], there exists preconditioners $\mathbf{P}_k$ that can speed up the convergence of a first-order algorithm. Thus, we choose to replace (2) with the following:

$$\begin{bmatrix} \mathbf{L}_{k+1} \\ \mathbf{R}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_k - \nabla_{\mathbf{L}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) \cdot \mathbf{P}_k \\ \mathbf{R}_k - \nabla_{\mathbf{R}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) \cdot \mathbf{P}_k \end{bmatrix} \tag{4}$$

We choose $\mathbf{P}_k \in \mathbb{R}^{r \times r}$ to be diagonal, where r is the rank of the matrix and the diagonal entries [p_1, … , p_r] are learnable parameters.

The incorporation of equations (3) and (4) allow for the implementation of a deep unfolding process, where each layer is pre-trained to minimize the loss at that iteration with respect to the parameters ζ_k and $\mathbf{P}_k$. All layers besides Layer 0 are then fully trained before moving on to the next layer, where, for some layer k, all parameters from 0 to k are trained on the loss with respect to ζ_k and $\mathbf{P}_k$, as illustrated in the below figure:



Note:

$$\mathscr{L} = \mathbb{E}_{(Z, \mathbf{X}_\star) \sim \mathscr{D}_{train}} \left\| \mathbf{L}_k(Z, \Theta) \left(\mathbf{R}_k(Z, \Theta) \right)^\top - \mathbf{X}_\star \right\|_F^2$$

where Θ *is the set of* ζ_k *and* $\mathbf{P}_k$ *at layer k*

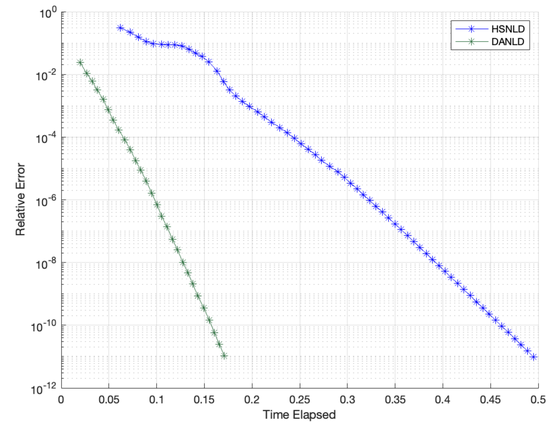
Algorithm

Algorithm 1 DeepHankel

Input: $\Pi_\Omega \mathbf{f}$: Partially observed Hankel matrix in reweighted vector form; r: rank of the underlying Hankel matrix; p: observation rate; α: outlier density; K: number of layers; $\{\zeta_k\}$: **learned** thresholding values; $\{P_k\}$: **learned** matrix preconditioners
Initialization: Let $\mathbf{L}_0$, $\mathbf{R}_0$ be produced by spectral initialization
for $k = 0, 1, \ldots, K-1$ **do**
  $\mathbf{z}_{k+1} = \mathcal{G}^*(\mathbf{L}_k \mathbf{R}_k^*)$
  $\mathbf{s}_{k+1} = \mathcal{T}_{\zeta_{k+1}}(\Pi_\Omega(\mathbf{f} - \mathbf{z}_{k+1})$
  $\mathbf{L}_{k+1} = \mathbf{L}_k - \nabla_{\mathbf{L}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) \cdot \mathbf{P}_k$
  $\mathbf{R}_{k+1} = \mathbf{R}_k - \nabla_{\mathbf{R}} \ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1}) \cdot \mathbf{P}_k$
end for

Inference algorithm of DeepHankel after the deep unfolding learning phase

Preliminary Numerical Results



Runtime comparison between HSNLD and DeepHankel for n = 8191 observations, r = 10 rank, and 20% outlier rate. DeepHankel achieves a ~3x speedup compared to HSNLD.

References

[1] O. Solomon, R. Cohen, Y. Zhang, Y. Yang, Q. He, J. Luo, R. van Sloun, Y. Eldar. "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Transactions on Medical Imaging, 2020.*

[2] HQ Cai, L. Huang, X. Lu, J. You. "Accelerating Ill-conditioned Hankel Matrix Recovery via Structured Newton-like Descent," *Inverse Problems, 2025.*

[3] J. Liu, X. Chen, Z. Wang, W. Yin, HQ Cai. "Towards Constituting Mathematical Structures for Learning to Optimize. *International Conference on Machine Learning, 2023.*