Deep Augmented Robust Hankel Matrix Recovery

Erick Rodriguez
Advisor: Dr. HanQin Cai
Center for Research in Computer Vision REU

July 30, 2025

Preliminary - Hankel Matrices

Given a signal $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$

Preliminary - Hankel Matrices

Given a signal $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$

We can construct the Hankel matrix:

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

Motivation

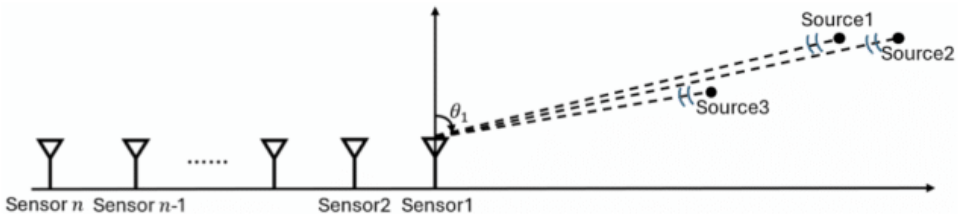


Figure: Direction of Arrival (DOA) radar problem

Motivation

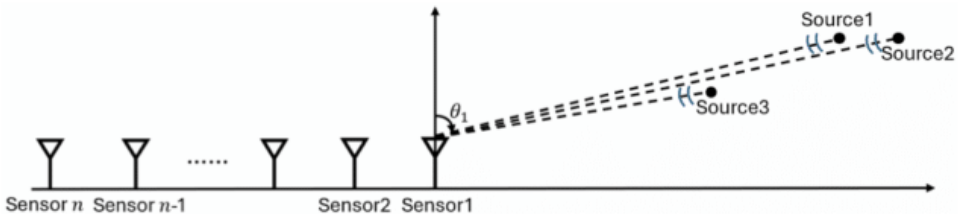


Figure: Direction of Arrival (DOA) radar problem

- Missing data

Motivation

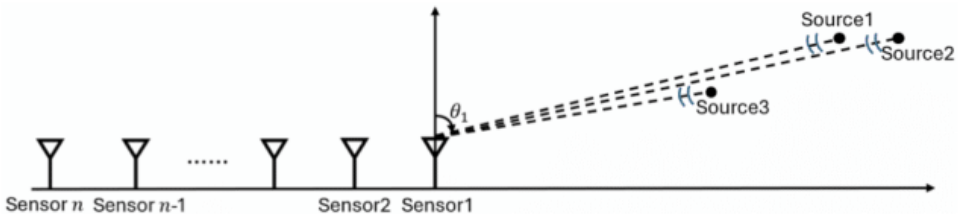


Figure: Direction of Arrival (DOA) radar problem

- Missing data
- Sparse outliers

Motivation

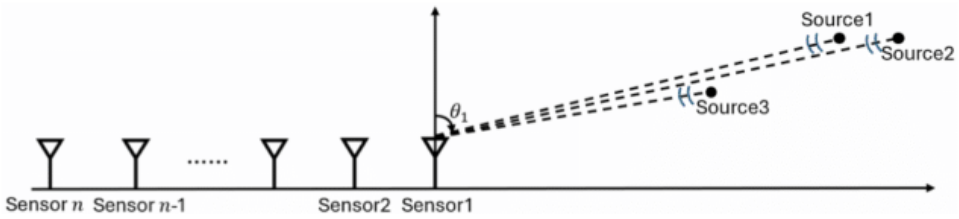


Figure: Direction of Arrival (DOA) radar problem

- Missing data
- Sparse outliers
- Ill-conditioned problem

Motivation

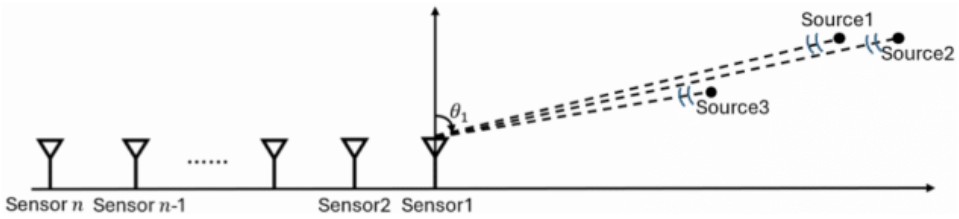


Figure: Direction of Arrival (DOA) radar problem

- Missing data
- Sparse outliers
- Ill-conditioned problem

We propose an algorithm that can recover a Hankel matrix in these circumstances, in a way that is **learnable** and **scalable**

DeepHankel - Deep Unfolding

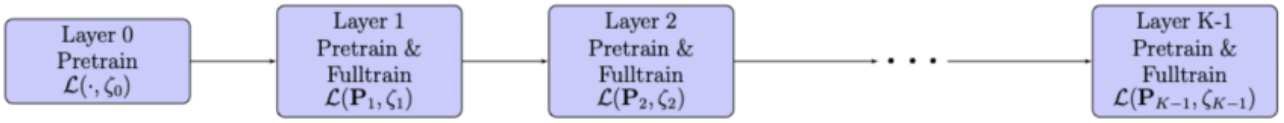


Figure: Deep unfolding process for DeepHankel, learning ζ_k and $\mathbf{P}_k$ at each layer k

DeepHankel - Inference

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

Figure: Inference algorithm for DeepHankel, after learning ζ_k and $\mathbf{P}_k$ for all layers

Preliminary Numerical Results

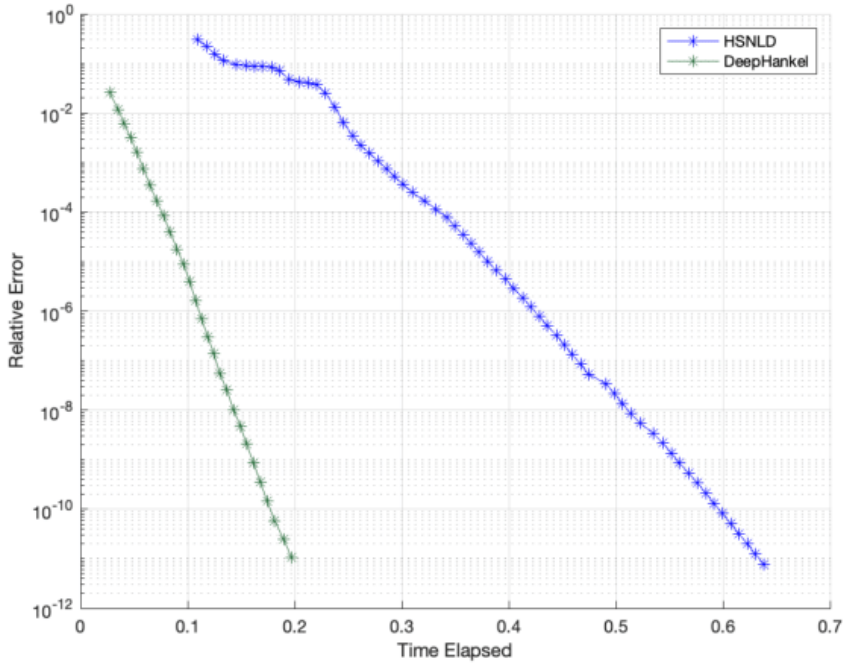


Figure: Runtime comparison between HSNLD and DeepHankel for n = 8191 observations, r = 10 rank, and 20% outlier rate. DeepHankel achieves a roughly 3x speedup compared to HSNLD.

Thank You!