Deep Augmented Robust Hankel Matrix Recovery

Erick Rodriguez
University of Central Florida
Orlando, Florida
erick.rodriguez@ucf.edu

Advisor: Dr. HanQin Cai
University of Central Florida
Orlando, Florida
hqcai@ucf.edu

Abstract

Hankel matrix recovery algorithms play a vital role in time-sensitive signal processing tasks, such as seismic data analysis for earthquake prediction, where signals must be reconstructed from incomplete and corrupted observations. While machine learning techniques have shown promise in general low-rank matrix recovery problems and robust principal component analysis, existing methods for Hankel matrix recovery remain computationally intensive and scale poorly. We propose DeepHankel, a novel deep unfolding-based model that learns a set of parameters for an inference procedure tailored for Hankel matrix reconstruction. Unlike related deep unfolding approaches that require expensive singular value decompositions (SVDs) at each iteration, DeepHankel performs SVD only once during initialization, significantly reducing computational costs. Tests on synthetic data show that DeepHankel recovers Hankel-structured signals up to 3x faster than the state-of-the-art HSNLD algorithm and only requiring half as many iterations.

1. Introduction

The low-rank Hankel matrix recovery problem is one that commonly appears in signal processing problems, including seismic analysis [3], nuclear magnetic resonance (NMR) spectroscopy [7], and time series analysis [6], among others. Given a vector of samples from an underlying signal $y = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}$, the corresponding Hankel matrix is defined as follows:

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

The structure of Y allows for certain matrix decompositions to reveal underlying information about the original signal. There are three problems that exist in practice: (a) *missing data*, (b) *sparse outliers*, and (c) *ill-conditioned matrices*.

1.1. Missing data

In some real-world problems, it is common for a signal to be only partially observed. For instance, in a dataset of Uber movement speed data [4], only 15 percent of values are observed. A meaningful analysis of the full dataset will require a sufficiently accurate recovery of the missing 85 percent of data. This is typically made possible by constraining the problem to recovering a low-rank matrix, allowing models to learn recurring patterns in the data.

1.2. Sparse outliers

Within the observed portion of the signal, some percentage of values are corrupted by external noise $\mathbf{S}$. In most applications, these outliers must be isolated from the underlying signal to perform any further analysis. In some cases, these outliers could be the desired portion of the data. In seismic analysis, for instance, these outliers may represent unusual tectonic activity that occurs moments before an earthquake strikes. In these applications, it is critical for the outliers to be fully isolated from the ground truth signal.

If these outliers were not Hankel structured, simply enforcing the Hankel constraint (where each entry of an antidiagonal has the same value) would result in a successful separation from the outlier and ground truth signal. For the purposes of this paper, we analyze the more difficult and applicable case where the sparse outliers are Hankel structured, corrupting entire antidiagonals at a time.

1.3. Ill-conditioned matrices

Consider the Direction of Arrival (DOA) estimation problem in Fig. 1, where a radar signal emitted by multiple sources is picked up by a series of sensors, producing an observed signal. Note that the signals are reaching the sensors at very similar angles; this results in increased interference between the sources' signals. After the data is

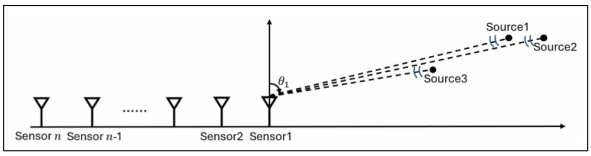


Figure 1. Example of the DOA estimation problem, where multiple sources at similar angles are emitting signals that are received by a series of sensors.

collected, depending on the similarity of the angles, isolating the ground truth of each source becomes a harder task.

Mathematically, the degree of difficulty can be represented by the condition number, which itself is the ratio between the first and r-th singular values, in descending order of magnitude, of the corresponding rank-r Hankel matrix. Matrices with a high condition number are referred to as ill-conditioned, and are harder for models to recover.

1.4. Problem statement

With these constraints in mind, models in low-rank Hankel matrix recovery aim to recover both the complete ground truth Hankel matrix $\mathbf{Z}$ and the sparse outlier matrix $\mathbf{S}$ from an observed matrix $\mathbf{Y}$ such that $\mathbf{Y} = \mathbf{Z} + \mathbf{S}$.

2. Notation

In this paper, we utilize lowercase letters for scalars, bold lowercase letters for vectors (e.g. $\mathbf{z}$), bold capital letters for matrices (e.g. $\mathbf{Z}$) and calligraphic letters for operators (e.g. $\mathcal{S}$). $\|\cdot\|$ is used to represent the absolute value, whereas $\|\cdot\|_F$ is used for the Frobenius norm. The operator $\mathcal{G}$ is used to represent the Hankel operator, which Hankelizes a vector $\mathbf{v}$ to a Hankel matrix $\mathbf{V}$; likewise, the adjoint Hankel operator $\mathcal{G}^*$ turns a matrix $\mathbf{V}$ into a rescaled Hankel vector representation $\mathbf{v}$.

3. Related work

To solve the problem of low-rank Hankel matrix recovery, several algorithms already exist. One example is Accelerated Structured Alternating Projections (ASAP) [1], which provides fast Hankel matrix recovery but cannot account for missing data. Another example is Robust-EMaC [5], which can account for missing data but utilizes semi-definite programming, which is both computationally and memory-wise inefficient.

In related problems, including robust principal component analysis (RPCA) and low-rank matrix recovery for general matrices, deep unfolding has been utilized to achieve faster convergence. For instance, Learned Robust Matrix Completion (LRMC) [8] leverages deep unfolding to recover matrices efficiently, but is applied to general matrices and does not take advantage of any matrix structure. This creates memory inefficiencies for Hankel-structured problems; creating the full Hankel matrix during the algorithm is unnecessary and would not scale well for signals with a large number of observations. Another approach, CORONA [10] applies deep unfolding for RPCA, but its inference model computes a costly singular value thresholding (SVT) at each iteration, hurting scalability.

One promising Hankel recovery algorithm is Hankel Structured Newton-Like Descent (HSNLD) [2], which recovers the ground truth Hankel matrices $\mathbf{Z}$ and $\mathbf{S}$ in their vector forms $\mathbf{z}$ and $\mathbf{s}$ respectively by Algorithm 1, where the underlying Hankel matrix Z is formed by $\mathbf{L}\mathbf{R}^*$, where $\mathbf{L}, \mathbf{R} \in \mathbb{C}^{n\times r}$. Note that the full Hankel matrix $\mathbf{Z}$ and sparse outlier matrix $\mathbf{S}$ are never formed; by using their vector representations $\mathbf{z}$ and $\mathbf{s}$, the algorithm saves a considerable amount of memory. The loss function ℓ is defined by Eq. (1).

Algorithm 1 Hankel Structured Newton-Like Descent (HSNLD)

Input: $\Pi_\Omega \mathbf{f}$: partial observation on the corrupted Hankel matrix in reweighted vector form; r: the rank of underlying Hankel matrix; p: observation rate; α: outlier density; $\{\gamma_k\}$: parameters for sparsification operator; η: step size; K: maximum number of iterations.

1: **Initialization:** Let $\mathbf{L}_0, \mathbf{R}_0$ be produced by spectral initialization.
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: $\quad \mathbf{z}_{k+1} = \mathcal{G}^*(\mathbf{L}_k \mathbf{R}_k^*)$
4: $\quad \mathbf{s}_{k+1} = \Gamma_{\gamma_k \alpha p}(\Pi_\Omega(\mathbf{f} - \mathbf{z}_{k+1}))$
5: $\quad \begin{bmatrix}\mathbf{L}_{k+1}\\ \mathbf{R}_{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{L}_k - \eta\nabla\ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1})(\mathbf{R}_k^*\mathbf{R}_k)^{-1}\\ \mathbf{R}_k - \eta\nabla\ell(\mathbf{L}_k, \mathbf{R}_k; \mathbf{s}_{k+1})(\mathbf{L}_k^*\mathbf{L}_k)^{-1}\end{bmatrix}$
6: **end for**
7: **Output:** $\mathbf{z}_{\text{output}} = \mathcal{G}^*(\mathbf{L}_K \mathbf{R}_K^*)$: recovered Hankel matrix in vector form.

HSNLD currently outperforms other state of the art algorithms for low-rank Hankel matrix recovery problems, shown empirically for both DOA and NMR datasets [2]. However, this approach does not leverage any computational gains from deep unfolding. HSNLD also suffers from requiring a higher iteration count than other algorithms despite being faster overall, taking 72 iterations for DOA estimation compared to a competing algorithm, SAP, which took only 32 iterations for its recovery. Part of the issue is the sparsification operator used, Γ, which keeps the largest $\gamma_k \alpha p$ entries as outliers and vanishes the remaining entries to 0. This approach requires more iterations to successfully separate outliers from the ground truth Hankel matrix.

Moreover, the Newton-like descent step employed in HSNLD involves two matrix preconditioners, defined to be $(\mathbf{R}_k^*\mathbf{R}_k)^{-1}$ and $(\mathbf{L}_k^*\mathbf{L}_k)^{-1}$ at each iteration. It has been theoretically proven that there exists an optimal set of ma-

$$\ell(\mathbf{L},\mathbf{R};\mathbf{s}) = \frac{1}{2p}\left\langle \Pi_\Omega\left(\mathcal{G}^*(\mathbf{L}\mathbf{R}^*)+\mathbf{s}-\mathbf{f}\right), \mathcal{G}^*(\mathbf{L}\mathbf{R}^*)+\mathbf{s}-\mathbf{f}\right\rangle + \frac{1}{2}\left\|(\mathcal{I}-\mathcal{G}\mathcal{G}^*)(\mathbf{L}\mathbf{R}^*)\right\|_{\mathrm{F}}^2 \tag{1}$$

Figure 2. The loss function utilized in Algorithms 1 and 2

trix preconditioners $\mathbf{P}_k$ for $k = 1, 2...K$ that speed up the convergence of a first-order algorithm (Theorem 1, [9]), there is no theoretical guarantee that the chosen matrices in HSNLD are optimal, meaning that there are potentially better choices for $\mathbf{P}_k$ that can be learned.

4. Proposed method

To address the problem of low-rank Hankel matrix recovery, our paper proposes DeepHankel, a learnable model that addresses the scalability concerns of other machine learning-based methods while exceeding the speed of traditional Hankel matrix recovery models. DeepHankel employs a deep unfolding approach, where a neural network leverages end-to-end training on a predefined algorithm to learn optimal values for certain parameters at each iteration of the algorithm. Then, DeepHankel performs an inference phase using those optimal values to recover both the low-rank Hankel matrix and corresponding sparse outlier matrix.

4.1. Deep unfolding

A feedforward neural network with K layers, where K is chosen by the user, is trained to learn two sets of parameters:

- $\{\zeta_k\}$ for $k = 0, 1 \ldots, K$, the set of thresholding values that are used on a given thresholding operator to separate the ground truth signal from the outlier values at each iteration.
- $\{\mathbf{P}_k\}$ for $k = 1, 2 \ldots, K$, the set of learned matrix preconditioners that allow for faster convergence of the inference algorithm's descent step. In particular, $\mathbf{P}_k$ is chosen to be an $r \times r$ diagonal matrix, where the diagonal entries $\begin{bmatrix} p_1 & p_2 & \ldots & p_r \end{bmatrix}$ are learnable parameters.

The training is conducted layer-wise for layers $0, 1, \ldots, K-1$. At Layer 0, a pretraining phase is done to learn ζ_0, and for all other layers, there is both a pretraining phase to learn ζ_k and $\mathbf{P}_k$ and a full training phase to learn all ζ and $\mathbf{P}$ from ζ_0, $\mathbf{P}_1$ to ζ_k and $\mathbf{P}_k$.

For all layers, we define the loss function to be

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

where Θ is defined to be the set of ζ and $\mathbf{P}$ activated during the respective training phase. A visualization of the deep unfolding architecture is provided in Fig. 3.

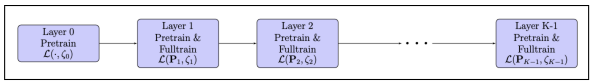


Figure 3. Representation of the deep unfolding architecture used to learn parameters for DeepHankel.

4.2. Inference

After the learning process is complete, DeepHankel can perform an inference phase similar to HSNLD's core algorithm, using the learned thresholding values and matrix preconditioners. The inference phase is outlined in Algorithm 2, where the loss function ℓ is defined by Eq. (1).

Algorithm 2 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

In Algorithm 2, $\mathcal{T}$ represents a thresholding function with respect to the learned ζ value. For testing purposes, the thresholding function chosen was the soft-thresholding function with $\|\cdot\|$ denoting the absolute value, and applied element-wise for every $\mathbf{x}_i$ in a vector $\mathbf{x}$:

$$\mathcal{S}_\zeta(\mathbf{x}_i) = \frac{max(\|\mathbf{x}_i - \zeta\|, 0)}{\|\mathbf{x}_i\|} \cdot \mathbf{x}_i$$

This replaces the sparsification operator Γ, whose computation requires a partial sorting that scales to the order of $\mathcal{O}(n \log n)$, whereas the soft thresholding function scales to the order of $\mathcal{O}(n)$. Other thresholding methods, such as adaptive thresholding or a neural network trained to directly detect outliers, could be used in place of $\mathcal{S}_\zeta(\mathbf{x}_i)$.

Note that, when updating $\mathbf{L}$ and $\mathbf{R}$, there is no learned step size η_k. Since the deep unfolding phase learns the relative magnitudes of each entry $\mathbf{P}_k$, it would be redundant to learn an additional scaling factor.

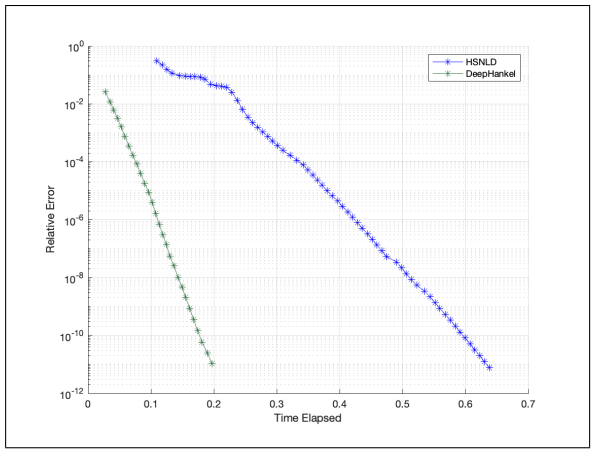


Figure 4. Runtime comparison between HSNLD and DeepHankel for n = 8191 observations, r = 10 rank, and 20% outlier rate. DeepHankel achieves a roughly 3x speedup compared to HSNLD.

5. Preliminary results

To test against the baseline HSNLD model, DeepHankel was trained with $K = 30$ layers and 1000 epochs for each layer's pretrain and full train phase. The training phase was done on Python using PyTorch, taking under half an hour to train. For a fairer comparison, the inference portion was conducted on MATLAB for both models. The results were computed using an Apple M3 chip and 16GB RAM.

A runtime comparison plot is shown in Fig. 4, where the time elapsed is plotted against the loss at a logarithmic scale. Both models successfully recover the ground truth with a precision of around 10^{-11}, but DeepHankel recovers the matrix in under 0.2 seconds, compared to HSNLD's 0.65 second runtime. DeepHankel's 30 layers (and thus 30 iterations on the inference phase) also put it at roughly half the number of iterations required for convergence against HSNLD's 61 iterations.

6. Conclusion

In this paper, we introduce DeepHankel, a novel low-rank Hankel matrix recovery algorithm capable of recovering a partially observed, noisy, and ill-conditioned Hankel matrix using methods that scale well to higher observations. DeepHankel employs deep unfolding techniques to learn thresholding values and matrix preconditioners for each iteration. Empirical results show that the model can recover Hankel matrices 3 times as fast as the baseline HSNLD.

References

[1] HanQin Cai, Jian-Feng Cai, Tianming Wang, and Guojian Yin. Accelerated structured alternating projections for robust spectrally sparse signal recovery. *IEEE Transactions on Signal Processing*, 69:809–821, 2021. 2

[2] HanQin Cai, Longxiu Huang, Xiliang Lu, and Juntao You. Accelerating ill-conditioned hankel matrix recovery via structured newton-like descent. *Inverse Problems*, 41(7):075015, July 2025. 2

[3] Ke Chen and Mauricio D. Sacchi. Robust reduced-rank filtering for erratic seismic noise attenuation. *GEOPHYSICS*, 80(1):V1–V11, Jan. 2015. 1

[4] Xinyu Chen, Chengyuan Zhang, Xi-Le Zhao, Nicolas Saunier, and Lijun Sun. Forecasting sparse movement speed of urban road networks with nonstationary temporal matrix factorization. *Transportation Science*, 2025. 1

[5] Yuxin Chen and Yuejie Chi. Robust spectral compressed sensing via structured matrix completion. *IEEE Transactions on Information Theory*, 60(10):6576–6601, 2014. 2

[6] Jonathan Gillard and Konstantin Usevich. Structured low-rank matrix completion for forecasting in time series analysis. *International Journal of Forecasting*, 34(4):582–597, Oct. 2018. 1

[7] Daniel J. Holland, Mark J. Bostock, Lynn F. Gladden, and Daniel Nietlispach. Fast multidimensional nmr spectroscopy using compressed sensing. *Angewandte Chemie International Edition*, 50(29):6548–6551, June 2011. 1

[8] Chandra Kundu, Abiy Tasissa, and HanQin Cai. A dual basis approach for structured robust euclidean distance geometry, 2025. 2

[9] Jialin Liu, Xiaohan Chen, Zhangyang Wang, Wotao Yin, and Hanqin Cai. Towards constituting mathematical structures for learning to optimize. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21426–21449. PMLR, 23–29 Jul 2023. 3

[10] Ben Luijten, Nishith Chennakeshava, Yonina C. Eldar, Massimo Mischi, and Ruud J.G. van Sloun. Ultrasound signal processing: From models to deep learning. *Ultrasound in Medicine Biology*, 49(3):677–698, 2023. 2