Sketching

1 JESPRIT Sketching for Mixed Poisson Distribution

In this section, we describe the application of the JESPRIT algorithm to recover the latent parameters of the Mixed Poisson utilizing the Probability Generating Function (PGF).

1.1 Problem Formulation

We consider a dataset $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N_s)}\}$ consisting of N_s independent samples drawn from a Mixed Poisson distribution. The distribution is characterized by:

- A weight matrix $\mathbf{A} \in \mathbb{R}^{d \times r}$ (containing the Poisson rates).
- A probability vector $\boldsymbol{\pi} = [\pi_1, \dots, \pi_r]^T$ (representing the mixture weights of the latent components).

Our goal is to estimate the unknown Matrix $\mathbf{A}$ and probabilities $\boldsymbol{\pi}$ solely from the observed count data $\mathcal{X}$.

1.2 Sampling Strategy: The Empirical PGF

The core of our approach lies in the Probability Generating Function (PGF). The theoretical PGF of a Mixed Poisson distribution takes the form of a weighted sum of exponentials:

$$G_{\mathbf{X}}(\mathbf{t}) = \sum_{k=1}^{r} \pi_k e^{\langle \boldsymbol{\lambda}_k, \mathbf{t}-\mathbf{1} \rangle} \tag{1}$$

where $\boldsymbol{\lambda}_k$ is the k-th column of $\mathbf{A}$.

Since we do not have specific access to the true distribution, we approximate it using the Empirical PGF computed from our dataset. For a specific query point $\mathbf{t} \in \mathbb{C}^d$, the empirical PGF is given by the average:

$$\hat{G}_{\mathbf{X}}(\mathbf{t}) = \frac{1}{N_s} \sum_{j=1}^{N_s} e^{\langle \mathbf{x}^{(j)}, \ln(\mathbf{t}) \rangle} \tag{2}$$

By the Law of Large Numbers, $\hat{G}_{\mathbf{X}}(\mathbf{t}) \to G_{\mathbf{X}}(\mathbf{t})$ as $N_s \to \infty$. This allows us to treat the computed values $\hat{G}_{\mathbf{X}}(\mathbf{t})$ as noisy measurements of the theoretical function.

1.3 Mapping to JESPRIT

We need to map this PGF estimation problem to the multidimensional harmonic retrieval signal model required for the JESPRIT algorithm:

$$y[\mathbf{n}] = \sum_{k=1}^{r} a_k e^{j\langle \boldsymbol{\omega}_k, \mathbf{n} \rangle}$$

We achieve this by choosing specific sampling points $\mathbf{t}$. Unlike the single-point case, ESPRIT requires a sequence of samples along each direction to estimate the shift invariance. For each direction vector $\mathbf{u} \in \mathbb{C}^d$, we sample at points indexed by an integer n:

$$\mathbf{t}(\mathbf{u}, n) = \mathbf{1} + j\Delta n \mathbf{u}, \quad n = 0, 1, \dots, 2N-1$$

Substituting this into the theoretical PGF equation (1):

$$\begin{aligned} G_{\mathbf{X}}(\mathbf{t}(\mathbf{u}, n)) &= \sum_{k=1}^{r} \pi_k e^{\langle \boldsymbol{\lambda}_k, (\mathbf{1}+j\Delta n\mathbf{u})-\mathbf{1} \rangle} \\ &= \sum_{k=1}^{r} \pi_k e^{jn\langle \Delta\boldsymbol{\lambda}_k, \mathbf{u} \rangle} \\ &= \sum_{k=1}^{r} \pi_k (e^{j\langle \Delta\boldsymbol{\lambda}_k, \mathbf{u} \rangle})^n \end{aligned}$$

This equation exactly matches the JESPRIT signal model components:

- **Amplitudes:** The mixture probabilities π_k correspond to the amplitudes a_k.
- **Frequencies:** The scaled rate projections $\langle \Delta\boldsymbol{\lambda}_k, \mathbf{u} \rangle$ correspond to the projected frequencies for that direction.
- **Measurements:** The values computed via the Empirical PGF at these points, denoted $y_{\mathbf{u}}[n]$, serve as the signal samples.

1.4 Parameter Recovery Algorithm

With the mapping established, we proceed to recover the parameters. A crucial aspect of JESPRIT is obtaining a set of matrices that share a common eigenvector basis. To ensure this coherence across different sampling directions, we employ a Global SVD approach rather than processing each direction independently.

1.4.1 Step 1: PGF Sampling & Global Subspace Estimation

We collect the measurements from all M sampling directions. For each direction l, we compute the sequence of PGF samples $y_l[n]$ for $n = 0, \dots, 2N-1$. From this sequence, we construct a square Hankel matrix $\mathbf{Z}_l \in \mathbb{C}^{N \times N}$:

$$\mathbf{Z}_l = \begin{pmatrix} y_l[0] & y_l[1] & \cdots & y_l[N-1] \\ y_l[1] & y_l[2] & \cdots & y_l[N] \\ \vdots & \vdots & \ddots & \vdots \\ y_l[N-1] & y_l[N] & \cdots & y_l[2N-2] \end{pmatrix}$$

We stack these matrices vertically to form a Global Data Matrix $\mathbf{X}_{\text{glob}}$:

$$\mathbf{X}_{\text{glob}} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_M \end{bmatrix}$$

We then compute the Singular Value Decomposition (SVD) of this global matrix:

$$\mathbf{X}_{\text{glob}} \approx \mathbf{U}_{\text{glob}} \boldsymbol{\Sigma} \mathbf{V}^H$$

The r dominant left singular vectors, denoted $\mathbf{U}_{\text{s}}$, span the common signal subspace for the entire dataset. This "locking" of the subspace is essential for the subsequent steps.

1.4.2 Step 2: Rotational Invariance Matrices (RIMs)

We partition the global signal subspace $\mathbf{U}_{\text{s}}$ back into M blocks corresponding to each direction, denoted $\hat{\mathbf{U}}_l$ (each of size $N \times r$). For each block, we exploit the rotational invariance structure. We form $\hat{\mathbf{U}}_{l,\uparrow}$ (first $N-1$ rows) and $\hat{\mathbf{U}}_{l,\downarrow}$ (last $N-1$ rows) and solve the overdetermined system:

$$\hat{\mathbf{U}}_{l,\uparrow} \boldsymbol{\Psi}_l \approx \hat{\mathbf{U}}_{l,\downarrow}$$

This yields M matrices $\{\boldsymbol{\Psi}_1, \dots, \boldsymbol{\Psi}_M\}$. Because they were derived from a common global subspace, they share the same eigenvectors (the columns of the mixing matrix inverse).

1.4.3 Step 3: Joint Diagonalization

We seek the single transformation matrix $\hat{\mathbf{T}}$ that simultaneously diagonalizes all M matrices $\boldsymbol{\Psi}_l$. We solve the optimization problem:

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T}} \sum_{l=1}^{M} \left\| \text{offdiag}(\mathbf{T} \boldsymbol{\Psi}_l \mathbf{T}^{-1}) \right\|_F^2$$

This is solved using Jacobi-like iteration methods. The resulting diagonal matrices $\hat{\boldsymbol{\Phi}}_l = \hat{\mathbf{T}} \boldsymbol{\Psi}_l \hat{\mathbf{T}}^{-1}$ contain the estimated eigenvalues on their diagonals.

1.4.4 Step 4: Rate and Probability Estimation

The diagonal elements of $\hat{\boldsymbol{\Phi}}_l$ provide the estimated frequencies. For the k-th latent component in direction l, the frequency estimate is:

$$\hat{\omega}_{k,l} = \arg((\hat{\boldsymbol{\Phi}}_l)_{kk})$$

Using the relation $\omega_{k,l} = \Delta \langle \boldsymbol{\lambda}_k, \mathbf{u}_l \rangle$, we recover the rate projection. By collecting these projections across all M directions, we solve for the full d-dimensional rate vector $\hat{\boldsymbol{\lambda}}_k$. The complete weight matrix $\hat{\mathbf{A}}$ is then constructed by stacking these estimated vectors as columns:

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{\boldsymbol{\lambda}}_1 & \hat{\boldsymbol{\lambda}}_2 & \dots & \hat{\boldsymbol{\lambda}}_r \end{bmatrix}$$

Finally, to recover the mixture probabilities $\boldsymbol{\pi}$, we use the estimated rates $\hat{\boldsymbol{\lambda}}_k$ to construct a system of linear equations. The observed PGF samples $\mathbf{y}$ (from the data collection step) are modeled as linear combinations of the unknown probabilities:

$$y_l \approx \sum_{k=1}^{r} \pi_k e^{j\Delta\langle\hat{\boldsymbol{\lambda}}_k, \mathbf{u}_l\rangle}, \quad l = 1, \ldots, M$$

This can be written in matrix form as $\mathbf{y} \approx \mathbf{E}\boldsymbol{\pi}$, where the matrix $\mathbf{E}$ has entries $E_{l,k} = e^{j\Delta\langle\hat{\boldsymbol{\lambda}}_k, \mathbf{u}_l\rangle}$. We solve this system via least-squares estimation:

$$\hat{\boldsymbol{\pi}} = (\mathbf{E}^H\mathbf{E})^{-1}\mathbf{E}^H\mathbf{y}$$

subject to the constraints $\pi_k \geq 0$ and $\sum \pi_k = 1$ if desired.

1.5 Results

1.5.1 Phase Unwrapping Impact

In the theoretical derivation of ESPRIT-based methods, phase unwrapping is often cited as a necessary step to resolve the ambiguity of the frequency estimates when the phase arguments exceed the $(-\pi, \pi]$ range. However, we found that applying phase unwrapping in the JESPRIT context actually hurts performance.

Figure 1 compares the parameter estimation error with and without phase unwrapping enabled. For this experiment, the mixing matrix A was fixed to a 3×3 matrix. It can be observed that the unwrapped version consistently yields higher error rates and is robust over a smaller range of grid scale values (Δ).

1.5.2 Parameter Sensitivity Analysis

We evaluate the sensitivity of the JESPRIT algorithm (without phase unwrapping) to its key hyperparameters: the number of directions M, the number of snapshots S, the number of sample points per line N, and the grid scale Δ.

As shown in the subplots of Figure 1a:

- **Directions (M) and Snapshots (S):** The estimation error remains stable and low as M and S increase beyond the sufficient lower bounds (related to d and r). This suggests that the algorithm is robust to over-sampling in these dimensions, and performance does not degrade with larger values, mainly for M.
- **Samples per Line (N):** Unlike M and S, increasing N excessively can lead to performance degradation. While a certain minimum number of points is required for accuracy, very large N effectively extends the sampling range into regions where the phase arguments may exceed the principal range, causing wrapping issues when Δ is fixed.
- **Grid Scale (Δ):** This parameter exhibits a distinct "sweet spot." As discussed previously, the error is minimized when $\Delta \approx 1/\max(A)$. Deviating significantly from this value increases the estimation error due to numerical overflow in the sampling of the PGF.

These results highlight that while M and S can be chosen generously, N and particularly Δ require careful tuning to match the signal characteristics.

1.5.3 Sample Complexity and Rate Range

Finally, we analyze the sample complexity of JESPRIT, specifically how many samples are required to successfully recover the latent factors as the problem dimensions grow. We also investigate the impact of the dynamic range of the poisson rates in $\mathbf{A} \in \mathbb{R}^{d\times r}$.

To quantify this, we performed a grid search over varying ambient dimensions $d \in [1, 10]$ and ranks $r \in [1, 10]$. For each (d, r) pair, we conducted 7 independent random trials. A trial was considered successful if the average Mean Relative Error for both poisson rates in $\mathbf{A}$ and probabilities $\boldsymbol{\pi}$ was less than or equal to 10%. We tested sample sizes of $N_s \in \{1k, 10k, 50k, 100k\}$.

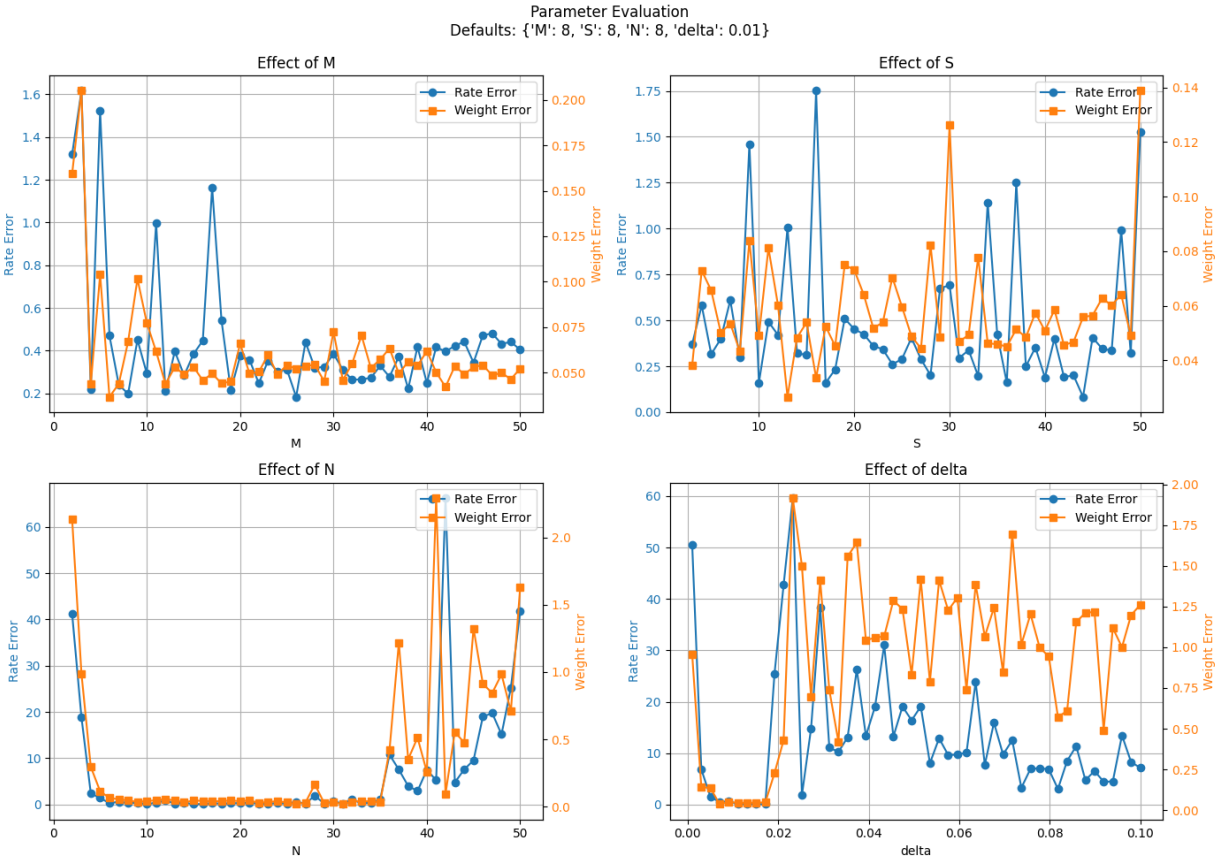
Simulations were conducted only while the success rate remained above 70%. If the success rate dropped below this threshold, further simulations for higher ranks r were halted, as failure was deemed certain. This explains the absence of data points for higher r values in the heatmaps.

We compared the sample complexity for two scenarios:

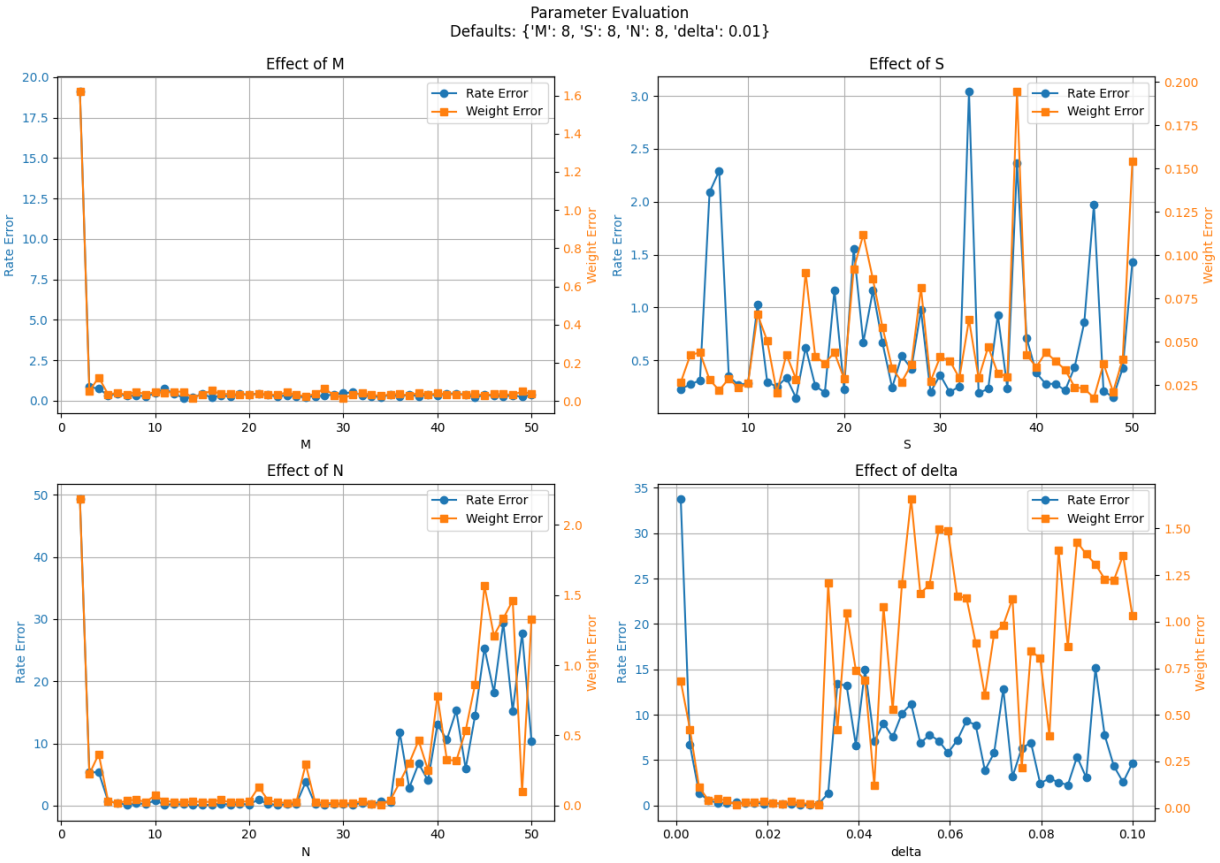
1. **Small Range**: Rates $\mathbf{A}$ randomly drawn from $[0, 100]$.
2. **Large Range**: Rates $\mathbf{A}$ randomly drawn from $[0, 10000]$.

Figures 2a and 2b show the success rate (percentage of trials that met the 10% error threshold) for each configuration. The results indicate that a larger range of values in $\mathbf{A}$ improves the recoverability of the latent factors. With a larger dynamic range, the "directions" in the count space are more distinct, essentially providing a higher effective signal-to-noise ratio for the subspace estimation. This allows the algorithm to correctly discover more latent factors (higher r) for a given sample size compared to the small range scenario.

Furthermore, the results suggest that the sample complexity depends primarily on the number of latent factors r, rather than the ambient dimension d. As observed in the heatmaps, increasing d while keeping r constant results in a negligible increase in the sample size required for successful recovery.

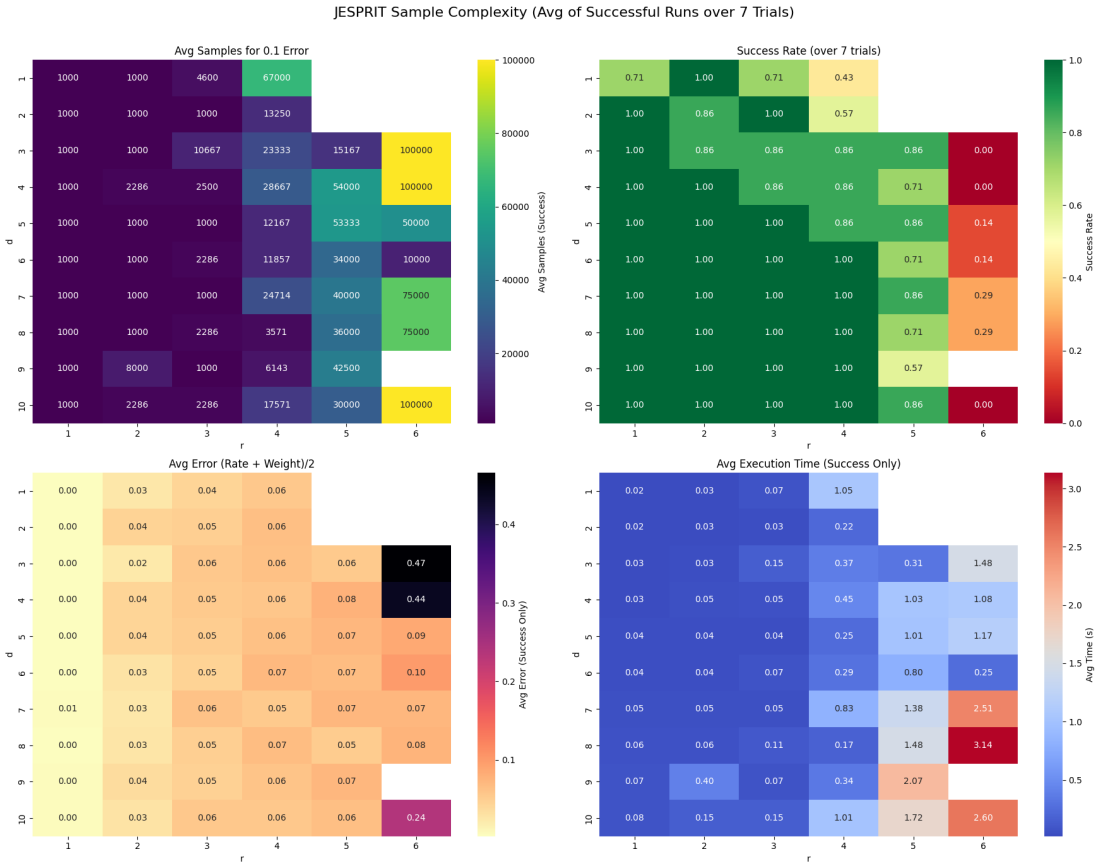


(a) With Phase Unwrapping

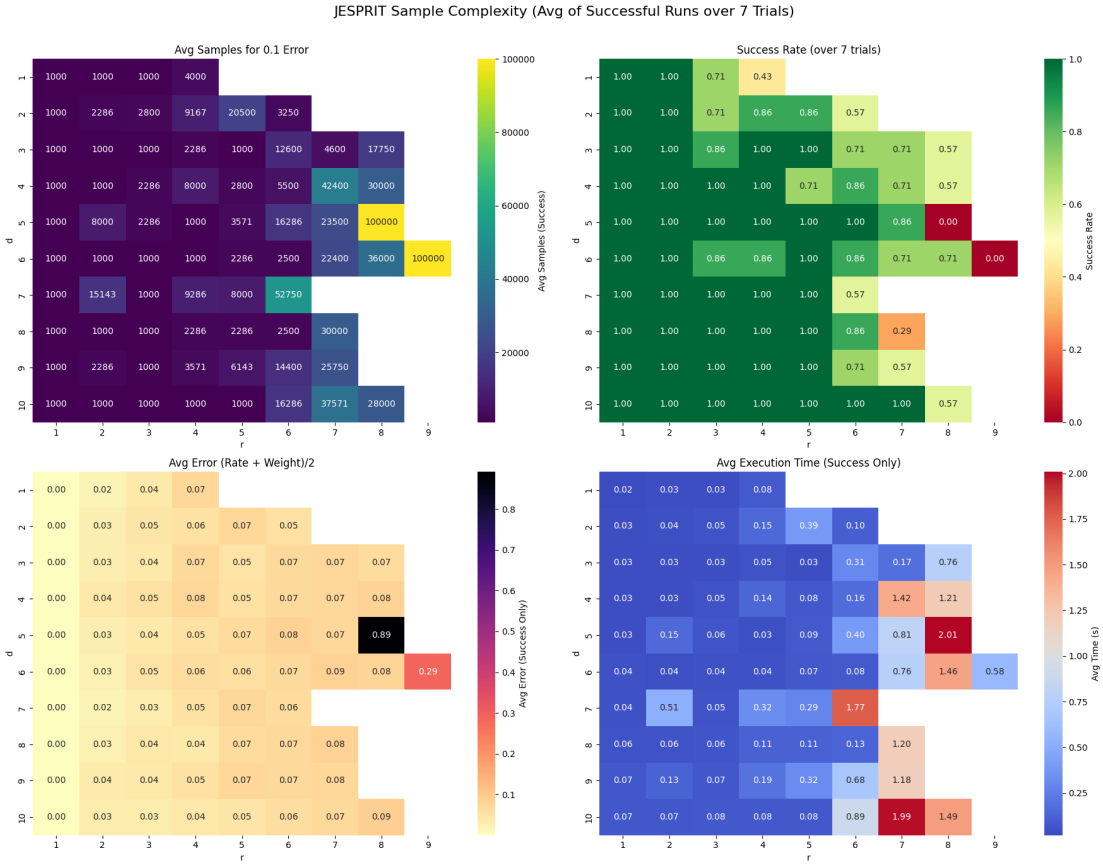


(b) Without Phase Unwrapping

Figure 1: Comparison of parameter estimation error: (a) with phase unwrapping enabled, and (b) without phase unwrapping. The unwrapped version (bottom) shows sensitivity to Δ but robustness to M and S.



(a) Sample Complexity for $A \in [0, 100]$.



(b) Sample Complexity for $A \in [0, 10000]$.

Figure 2: Comparison of Sample Complexity for different rate ranges: (a) Small range $[0, 100]$, and (b) Large range $[0, 10000]$. The larger range allows for successful recovery of higher ranks.