# Tensors for signal and frequency estimation in subspace-based methods: when they are useful?


Nikita Khromov, Nina Golyandina

St. Petersburg State University
Department of Statistical Modeling


25.09.2025, CDAM'2025

# Introduction to Singular Spectrum Analysis (SSA)

SSA - family of methods for time series analysis

Problems that can be solved by SSA-related methods:

- Signal extraction
- Frequency estimation
- Smoothing and Noise reduction
- Signal decomposition (Trend and Periodicity extraction)
- Forecasting
- Missing data imputation
- Change in structure detection
- Many others...

# SSA References

Books:

- J.Elsner and A.Tsonis. Singular Spectrum Analysis: A New Tool in Time Series Analysis, Plenum, 1996.
- N.Golyandina, V.Nekrutrin and A.Zhigljavsky. Analysis of Time Series Structure: SSA and Related Techniques, CRC Press, 2001.
- S.Sanei and H.Hassani. Singular Spectrum Analysis for Biomedical Signals, CRC Press, 2016.
- N.Golyandina, A.Korobeynikov and A.Zhigljavsky. Singular spectrum analysis with R, Springer, 2018.
- N.Golyandina and A.Zhigljavsky. Singular Spectrum Analysis for Time Series, Springer, 2013, 2020 (2nd Edition).

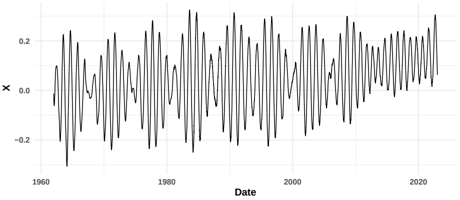
Implementations:

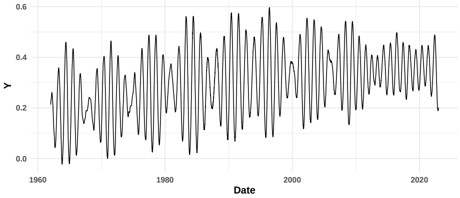
- R Package: Rssa
  https://CRAN.R-project.org/package=Rssa
- Python Package: PyRssa (Python wrapper over Rssa)
  https://pypi.org/project/pyrssa/

# Decomposition and Estimation Example

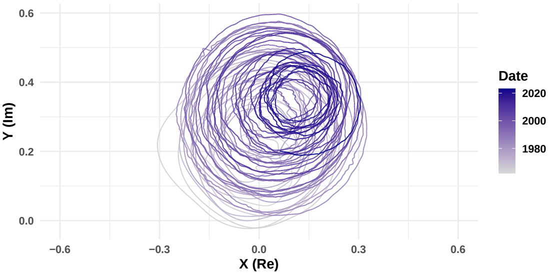
Data: Coordinates of Earth pole motion [IERS EOP 14 C04]

Raw data:




As complex time series:

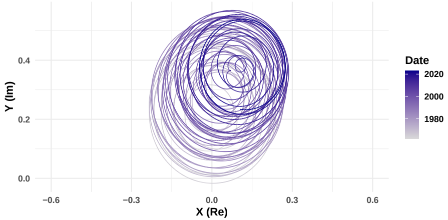


# Decomposition and Estimation Example

Decomposition of time series:

- Low-frequency component + high-frequency component
- Signal + noise
- Trend + Seasonality + Noise

Extracted signal:




Signal parameters estimates:

| Period (Days) | Change rate |
|---|---|
| $365.41$ | $-5.5 \cdot 10^{-6}$ |
| $433.10$ | $-2.2 \cdot 10^{-5}$ |
| $\to \infty$ | $2.7 \cdot 10^{-5}$ |

# SSA Algorithm: Embedding

**Input:** time series $\mathsf{X} = (x_1, x_2, \ldots, x_N)$, window length $L$, signal rank $r$.

1. **Embedding**. Constructing the $L$-*Trajectory* Hankel matrix $\mathbf{X} \in \mathbb{C}^{L \times K}$ from the series $\mathsf{X}$, where $K = N - L + 1$:

$$\mathbf{X} = \mathcal{T}^{(L)}(\mathsf{X}) = \begin{pmatrix} x_1 & x_2 & x_3 & \ldots & x_K \\ x_2 & x_3 & x_4 & \ldots & x_{K+1} \\ x_3 & x_4 & x_5 & \ldots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \ldots & x_N \end{pmatrix}$$

Why not embed into a higher order array (tensor)?

# SSA Algorithm: Decomposition, Grouping, Reconstruction

2. **Decomposition**. Constructing the singular value decomposition (SVD) of the matrix $\mathbf{X}$: $\mathbf{X} = \sum_{j=1}^{\operatorname{rank}\mathbf{X}} \sqrt{\lambda_j} U_j V_j^{\mathrm{H}} = \sum_{j=1}^{\operatorname{rank}\mathbf{X}} \widehat{\mathbf{X}}_j$ where $\mathrm{H}$ denotes Hermitian conjugation, $U_j$ and $V_j$ are left and right singular vectors of $\mathbf{X}$, $\sqrt{\lambda_j}$ — its singular values in descending order.

3. **Grouping**. Grouping the terms $\widehat{\mathbf{X}}_j$ from the decomposition related to the signal: $\mathbf{S} = \sum_{j=1}^{r} \widehat{\mathbf{X}}_j = \Pi_r \mathbf{X}$, where $\Pi_r$ is the projector onto the space of matrices with rank not greater than $r$.

4. **Reconstruction**. Applying projection onto the space of Hankel matrices: $\widetilde{\mathbf{S}} = \Pi_{\mathcal{H}} \widehat{\mathbf{S}}$, and return to the series form: $\widetilde{\mathsf{S}} = \left(\mathcal{T}^{(L)}\right)^{-1} (\widetilde{\mathbf{S}})$

# SSA-Rank of a Series

**Definition**

Series X has rank $d < N/2$ in terms of SSA, if the rank of its $L$-trajectory matrix equals $d$ for any $L$ such that $d \leqslant \min(L, N - L + 1)$.
If such $d$ exists, then X is called a series of finite rank.

If the signal S is a series of finite rank, then it is generally recommended to use $\mathrm{rank}(\mathsf{S})$ as parameter $r$ in the SSA method

Series rank examples

- rank of S with $s_n = A\exp(\alpha n)$, $\alpha \in \mathbb{C}$, equals 1
- rank of S with $s_n = A\sin(2\pi\omega n + \varphi)$, $0 < \omega < 1/2$, equals 2

# Signal Model

What we consider a signal $\mathsf{S} = (s_1, s_2, \ldots, s_N)$:

- The trajectory matrix $\mathbf{S} = \mathcal{T}^{(L)}(\mathsf{S})$ is rank-deficient
  ( $\Longrightarrow$ the time series is of some finite rank: $\mathrm{rank}(\mathsf{S}) = r$)
- Any signal $\mathsf{S}$ can be represented in the form of a finite sum:

$$s_n = \sum_j p_j(n) \exp(\alpha_j n + \mathrm{i} 2\pi\omega_j n),$$

  where $p_j(n)$ is a polynomial in $n$
- Real case:

$$s_n = \sum_j p_j(n) \exp(\alpha_j n) \sin(2\pi\omega_j n + \varphi_j),$$

ESPRIT method estimates damping factors $\alpha_j$ and frequencies $\omega_j$

# ESPRIT Algorithm: General Idea

Consider a signal S with elements $s_n$:

$$s_n = \sum_{j=1}^{2} \exp(\alpha_j n + \mathrm{i}(2\pi\omega_j n + \varphi_j)) = A_1 z_1^n + A_2 z_2^n$$

where $A_j = \exp(\mathrm{i}\varphi_j)$, $z_j = \exp(\alpha_j + \mathrm{i}2\pi\omega_j)$

Signal subspace basis is given by

$$\mathbf{M} = \begin{pmatrix} z_1 & z_2 \\ z_1^2 & z_2^2 \\ \vdots & \vdots \\ z_1^L & z_2^L \end{pmatrix} \Rightarrow \overline{\mathbf{M}} = \underline{\mathbf{M}} \begin{pmatrix} z_1 & \\ & z_2 \end{pmatrix} \Rightarrow \underline{\mathbf{M}}^{-}\overline{\mathbf{M}} = \begin{pmatrix} z_1 & \\ & z_2 \end{pmatrix}$$

where $\overline{\mathbf{M}}$ denotes $\mathbf{M}$ without the first row, $\underline{\mathbf{M}}$ — without the last
$\underline{\mathbf{M}}^{-}$ denotes the pseudoinverse of $\underline{\mathbf{M}}$

# ESPRIT Algorithm

**Input**: same as in SSA: X, $L$, $r$

1. **Embedding**. $\mathbf{X} = \mathcal{T}^{(L)}(\mathsf{X})$
2. **Decomposition**. $\mathbf{X} = \sum_{j=1}^{\operatorname{rank}\mathbf{X}} \sqrt{\lambda_j} U_j V_j^{\mathrm{H}}$, $\mathbf{U}_r = [U_1 : U_2 : \ldots : U_r]$
3. **Estimation**. Finding eigenvalues $z_j$ of matrix $\underline{\mathbf{U}}_r^{-} \overline{\mathbf{U}}_r$

   From $z_j = \exp(\alpha_j + \mathrm{i}2\pi\omega_j)$ parameters $\alpha_j$ and $\omega_j$ can be found

# Multi-Channel Time Series, MSSA

$\mathsf{X} = \left(\mathsf{X}^{(1)}, \mathsf{X}^{(2)}, \ldots, \mathsf{X}^{(P)}\right), \qquad \mathsf{X}^{(p)} = \left(x_1^{(p)}, x_2^{(p)}, \ldots, x_N^{(p)}\right)$ – channels

The only change in the algorithms — Embedding step:

$$\mathbf{X} = \mathcal{T}_{\mathrm{MSSA}}^{(L)}(\mathsf{X}) = \left[\mathbf{X}^{(1)} : \mathbf{X}^{(2)} : \ldots : \mathbf{X}^{(P)}\right],$$

$$\mathbf{X}^{(p)} = \mathcal{T}^{(L)}(\mathsf{X}^{(p)})$$

When to choose MSSA over SSA for each channel:

- All channels have “similar” structure
- “Supporting” channels with lower noise level

# MSSA Trajectory Matrix

$$\mathsf{X} = \left(\mathsf{X}^{(1)}, \mathsf{X}^{(2)}, \ldots, \mathsf{X}^{(P)}\right), \qquad \mathsf{X}^{(p)} = \left(x_1^{(p)}, x_2^{(p)}, \ldots, x_N^{(p)}\right)$$

$$\mathbf{X} = \mathcal{T}_{\mathrm{MSSA}}^{(L)}(\mathsf{X}) =$$

$$\left(\begin{array}{ccccc|ccccc|c}
x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \ldots & x_K^{(1)} & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \ldots & x_K^{(2)} & \ldots \\
x_2^{(1)} & x_3^{(1)} & x_4^{(1)} & \ldots & x_{K+1}^{(1)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} & \ldots & x_{K+1}^{(2)} & \ldots \\
x_3^{(1)} & x_4^{(1)} & x_5^{(1)} & \ldots & x_{K+2}^{(1)} & x_3^{(2)} & x_4^{(2)} & x_5^{(2)} & \ldots & x_{K+2}^{(2)} & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ldots \\
x_L^{(1)} & x_{L+1}^{(1)} & x_{L+2}^{(1)} & \ldots & x_N^{(1)} & x_L^{(2)} & x_{L+1}^{(2)} & x_{L+2}^{(2)} & \ldots & x_N^{(2)} & \ldots
\end{array}\right)$$

$$\underbrace{\phantom{x_1 x_2 x_3 \ldots x_K}}_{\mathbf{X}^{(1)}} \qquad \underbrace{\phantom{x_1 x_2 x_3 \ldots x_K}}_{\mathbf{X}^{(2)}}$$

But why limit yourself to matrices?
Matrix is just a 2D tensor

# Introducing Tensors to the Algorithm

$$\begin{array}{rlclcl} \text{Basic SSA:} & \text{Time series X} & \mapsto & \text{Matrix } \mathbf{X} & \mapsto & \mathrm{SVD}(\mathbf{X}) \\ \text{Tensor SSA:} & \text{Time series X} & \mapsto & \text{Tensor } \mathcal{X} & \mapsto & \underbrace{\mathrm{TD}}_{\text{Some Tensor Decomposition}}(\mathcal{X}) \end{array}$$

Tensor SVD Extensions:

- Higher-Order SVD (HOSVD)
- Canonical Polyadic Decomposition (CPD)
- T-SVD
- $(L_r, L_r, 1)$-Decomposition

# Reminder: Trajectory Matrix

- Single-Channel:

$$\mathcal{T}^{(L)}(\mathsf{X}) = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{pmatrix}$$

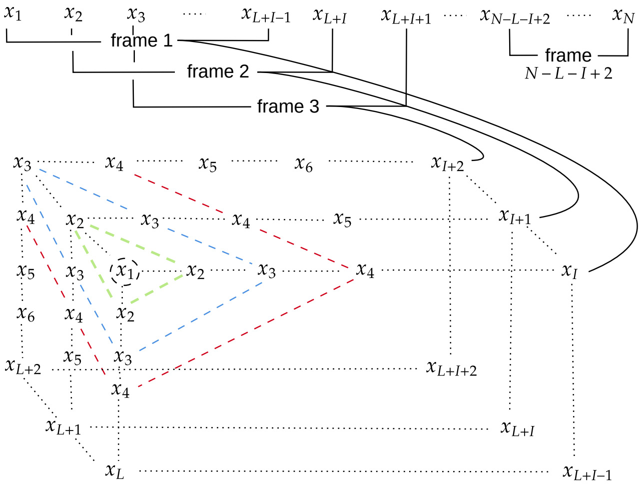
- Multi-Channel:

$$\mathcal{T}^{(L)}_{\text{MSSA}}(\mathsf{X}) = \left(\begin{array}{cccc|cccc|c} x_1^{(1)} & x_2^{(1)} & \dots & x_K^{(1)} & x_1^{(2)} & x_2^{(2)} & \dots & x_K^{(2)} & \dots \\ x_2^{(1)} & x_3^{(1)} & \dots & x_{K+1}^{(1)} & x_2^{(2)} & x_3^{(2)} & \dots & x_{K+1}^{(2)} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots \\ x_L^{(1)} & x_{L+1}^{(1)} & \dots & x_N^{(1)} & x_L^{(2)} & x_{L+1}^{(2)} & \dots & x_N^{(2)} & \dots \end{array}\right)$$

# Mapping Single-Channel Time Series to a Tensor

$$\mathsf{X} = (x_1,\, x_2,\, \dots,\, x_N)$$

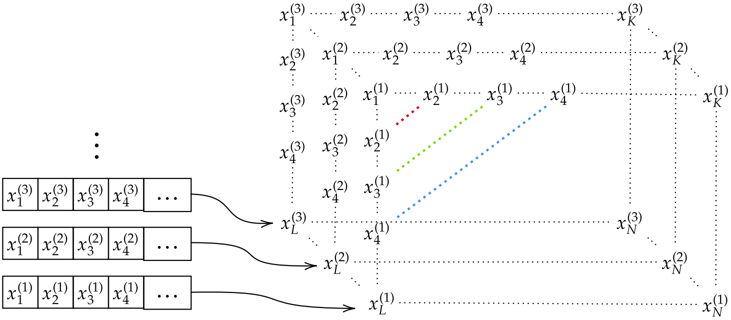
$$\mathsf{X} \mapsto \mathcal{T}^{(I,L)}_{\mathrm{T-SSA}}(\mathsf{X}) = \mathcal{X} \in \mathbb{C}^{I\times L\times K},\ K = N - I - L + 2$$



# Mapping Multi-Channel Time Series to a Tensor

$$\mathsf{X} = \left(\mathsf{X}^{(1)}, \mathsf{X}^{(2)}, \ldots, \mathsf{X}^{(P)}\right), \mathsf{X}^{(p)} = \left(x_1^{(p)}, x_2^{(p)}, \ldots, x_N^{(p)}\right)$$

$$\mathsf{X} \mapsto \mathcal{T}^{(L)}_{\mathrm{T-MSSA}}(\mathsf{X}) = \mathcal{X} \in \mathbb{C}^{P \times L \times K}, \; K = N - L + 1$$



# Some Tensor Decompositions

Unlike in matrix case, there exist several definitions of tensor ranks and SVD generalizations based on them.

**Definition (Tensor rank)**

Tensor $\mathcal{A}$ has rank 1, if there exist vectors $B$, $C$ and $D$ such that $\mathcal{A} = B \circ C \circ D$, where $\circ$ denotes an outer product.

Tensor $\mathcal{A}$ has rank $R$, if it can be represented as a sum of $R$ rank-1 tensors: $\mathcal{A} = \sum_{i=1}^{R} \mathcal{B}_i$, $\mathrm{rank}(\mathcal{B}_i) = 1$, and such $R$ is minimal.

The representation of a tensor as a sum of $R$ rank-1 tensors is called Canonical Polyadic Decomposition (CPD).

- Considered for signal extraction in [Kouchaki, Sanei (2013)]
- Does not provide any form of orthogonality of components
- Requires to know the tensor rank in advance
- No connection between signal SSA-rank and the rank of a trajectory tensor

# Some tensor decompositions

## Definition

$n$-mode vectors of a tensor $\mathcal{A}$ are vectors obtained from $\mathcal{A}$ by varying the index of the $n$-th direction and keeping the other indices fixed (analog of rows and columns of a matrix).

$n$-rank of a tensor $\mathcal{A}$, denoted by $R_n = \mathrm{rank}_n(\mathcal{A})$, is the dimension of the linear space spanned by the $n$-mode vectors.

HOSVD: $\mathcal{A} = \sum_{i=1}^{R_1} \sum_{l=1}^{R_2} \sum_{k=1}^{R_3} z_{ilk} U_i^{(1)} \circ U_l^{(2)} \circ U_k^{(3)}$

- Considered for signal parameter estimation in [Papy et al. (2005)], [Papy et al. (2009)]
- Provides orthogonality of components
- Does not require any prior knowledge about tensor $n$-ranks
- There is a proven connection between number of components and signal SSA-rank

# Higher-Order SVD. Higher-Order Orthogonal Iterations

$$\mathrm{SVD}(\mathbf{X}) = \sum_{j=1}^{\mathrm{rank}(\mathbf{X})} \sqrt{\lambda_j} U_j V_j^{\mathrm{H}}$$

$$\mathrm{HOSVD}(\mathcal{X}) = \sum_{i=1}^{\mathrm{rank}_{1}(\mathcal{X})} \sum_{l=1}^{\mathrm{rank}_{2}(\mathcal{X})} \sum_{k=1}^{\mathrm{rank}_{3}(\mathcal{X})} z_{ilk} U_i^{(1)} \circ U_l^{(2)} \circ U_k^{(3)}$$

- $\widetilde{\mathbf{X}} = \sum_{j=1}^{R} \ldots \Rightarrow \left\| \mathbf{X} - \widetilde{\mathbf{X}} \right\|_F = \min_{\mathrm{rank}(\widehat{\mathbf{X}}) \leqslant R} \left\| \mathbf{X} - \widehat{\mathbf{X}} \right\|_F$
- $\widetilde{\mathcal{X}} = \sum_{i=1}^{R_1} \sum_{l=1}^{R_2} \sum_{k=1}^{R_3} \ldots \Rightarrow \left\| \mathcal{X} - \widetilde{\mathcal{X}} \right\|_F \geqslant \min_{\mathrm{rank}_m(\widehat{\mathcal{X}}) \leqslant R_m} \left\| \mathcal{X} - \widehat{\mathcal{X}} \right\|_F$

Truncation of SVD is optimal, but truncation of HOSVD is not

Iterative algorithm for finding optimal approximation – HOOI

# T-SSA, T-MSSA and T-ESPRIT with HOSVD

**Input:** time series X, window length: $(I, L)$ for single-channel or $L$ for multi-channel, signal ranks $(r_1, r_2, r_3)$, $d$ — estimation dimension for HO-ESPRIT.

1. **Embedding.**

| | |
|---|---|
| Single-channel | $\mathsf{X} \mapsto \mathcal{T}^{(I,L)}_{\mathrm{T-SSA}}(\mathsf{X}) = \mathcal{X}$ |
| Multi-channel | $\mathsf{X} \mapsto \mathcal{T}^{(L)}_{\mathrm{T-MSSA}}(\mathsf{X}) = \mathcal{X}$ |

2. **Decomposition & Approximation.** Using $(r_1, r_2, r_3)$ $\mathcal{X} \mapsto \mathrm{Trunc}(\mathrm{HOSVD}(\mathcal{X})) = \widetilde{\mathcal{S}}$ or $\mathcal{X} \mapsto \mathrm{HOOI}(\mathcal{X}) = \widetilde{\mathcal{S}}$

3. **Reconstruction or Estimation.**
   - **Reconstruction.** $\mathsf{S} = \mathcal{T}^{-1}\left(\Pi_{\mathcal{H}_\mathrm{T}}(\widetilde{\mathcal{S}})\right)$, $\Pi_{\mathcal{H}_\mathrm{T}}$ — projector onto the space of Hankel tensors
   - **Estimation.** Finding eigenvalues $z_j$ of matrix $\underline{\mathbf{U}}^{-}\overline{\mathbf{U}}$, where $\mathbf{U} = \mathbf{U}_d = \left[U_1^{(d)} : U_2^{(d)} : \ldots : U_{r_d}^{(d)}\right]$. From $z_j = \exp(\alpha_j + \mathrm{i}2\pi\omega_j)$ damping factors $\alpha_j$ and frequencies $\omega_j$ of the signal can be found

# Single-Channel Series Comparison

$$x_n = e^{\alpha_1 n} e^{2\pi \mathrm{i} \omega_1 n} + e^{\alpha_2 n} e^{2\pi \mathrm{i} \omega_2 n} + \zeta_n$$

$\zeta_n$ — Complex white gaussian noise, $\mathrm{D}(\zeta_n) = 0.04^2$, $\omega_1 = 0.2$, $\omega_2 = 0.22$, $\alpha_1 = \alpha_2 = 0$ (same results for $\alpha_1 = \alpha_2 < 0$ and $\alpha_1 < \alpha_2 < 0$).

RMSE results with respect to parameters choice:

- Parameters estimation

| | Best | Worst | Mean |
|---|---|---|---|
| **T-ESPRIT** | **0.0012** | **0.0033** | **0.0016** |
| **ESPRIT** | 0.0013 | 0.0140 | 0.0031 |

T-method is better

- Signal extraction

| | Best | Worst | Mean |
|---|---|---|---|
| **T-SSA** | 0.019 | **0.024** | **0.020** |
| **SSA** | **0.018** | 0.031 | 0.021 |

T-method is worse with optimal parameter choice but better on average

# Multi-Channel Series Comparison

$$x_n^{(m)} = a_1^{(m)} e^{2\pi \mathrm{i}\omega_1 n} + a_2^{(m)} e^{2\pi \mathrm{i}\omega_2 n} + \zeta_n^{(m)},$$

$\zeta_n^{(m)}$ — Complex white gaussian noise, $\mathrm{D}\left(\zeta_n^{(m)}\right) = 0.2^2$,
$\omega_1 = 0.2$, $\omega_2 = 0.22$

RMSE results with respect to parameters choice:

- Parameters estimation

| | Best | Worst | Mean |
|---|---|---|---|
| **T-M-ESPRIT** | **5.33e-04** | **9.99e-04** | **6.16e-04** |
| **M-ESPRIT** | 5.36e-04 | 1.80e-03 | 7.13e-04 |

T-method is better

- Signal extraction

| | Best | Worst | Mean |
|---|---|---|---|
| **T-MSSA** | **0.065** | **0.068** | **0.066** |
| **MSSA** | 0.066 | 0.152 | 0.086 |

T-method is better

# Dtack Modifications

Possible problem for ESPRIT: components with close frequencies can mix into one in the presence of noise. Solution: using Dstack mapping.

Consider $\mathsf{X} = (x_1, x_2, \ldots, x_N)$, $M = \lfloor N/D \rfloor$, then

$$\mathrm{Dstack}_D(\mathsf{X}) = \mathcal{D}_D(\mathsf{X}) = \left[\begin{array}{c|c|c|c} \underbrace{\begin{matrix} x_1 \\ x_{D+1} \\ x_{2D+1} \\ \vdots \\ x_{(M-1)D+1} \end{matrix}}_{\mathsf{X}_D^{(1)}} & \underbrace{\begin{matrix} x_2 \\ x_{D+2} \\ x_{2D+2} \\ \vdots \\ x_{(M-1)D+2} \end{matrix}}_{\mathsf{X}_D^{(2)}} & \begin{matrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \cdots \end{matrix} & \underbrace{\begin{matrix} x_D \\ x_{2D} \\ x_{3D} \\ \vdots \\ x_{MD} \end{matrix}}_{\mathsf{X}_D^{(D)}} \end{array}\right]$$

| | |
|---|---|
| Dstack-SSA | $\mathsf{X} \mapsto \mathbf{X} = \mathcal{T}_{\mathrm{MSSA}}^{(L)}(\mathcal{D}_D(\mathsf{X}))$ |
| Dstack-T-SSA | $\mathsf{X} \mapsto \mathcal{X} = \mathcal{T}_{\text{T-MSSA}}^{(L)}(\mathcal{D}_D(\mathsf{X}))$ |

Undersampling: $\omega \mapsto \hat{\omega} = D\omega \implies |\omega| \leqslant \dfrac{1}{2D}$ is required

# Single-Channel Series Comparison with Dstack

$$x_n = \cos(2\pi\omega_1 n) + \cos(2\pi\omega_2 n) + \xi_n$$

$\omega_1 = 0.02$, $\omega_2 = 0.0205$, $\xi_n$ — white gaussian noise, $\mathrm{D}(\xi_n) = \sigma^2$

RMSE results with respect to parameters choice:

- Parametes estimation, $\sigma = 0.2$

| | Best | Worst | Mean |
|---|---|---|---|
| **ESPRIT** | **4.98e-05** | **5.66e-05** | **5.32e-05** |
| **Dstack-ESPRIT** | 5.30e-05 | 7.63e-05 | 6.24e-05 |
| **T-Dstack-ESPRIT** | 5.40e-05 | 6.24e-05 | 5.86e-05 |

T-method is worse

- Parametes estimation, $\sigma = 0.6$

| | Best | Worst | Mean |
|---|---|---|---|
| **ESPRIT** | 0.101 | 0.138 | 0.115 |
| **Dstack-ESPRIT** | 0.009 | 0.011 | 0.010 |
| **T-Dstack-ESPRIT** | **0.003** | **0.005** | **0.004** |

T-method is better

# Single-Channel Series Comparison with Dstack

$$x_n = \cos(2\pi\omega_1 n) + \cos(2\pi\omega_2 n) + \xi_n$$

$\omega_1 = 0.02$, $\omega_2 = 0.0205$, $\xi_n$ — white gaussian noise, $\mathrm{D}(\xi_n) = \sigma^2$

RMSE results with respect to parameters choice:

- Signal extraction, $\sigma = 0.2$

| | Best | Worst | Mean |
|---|---|---|---|
| **SSA** | **0.020** | **0.021** | **0.021** |
| **Dstack-SSA** | 0.052 | 0.066 | 0.057 |
| **T-Dstack-SSA** | 0.050 | 0.052 | 0.051 |

T-method is worse

- Signal extraction, $\sigma = 0.6$

| | Best | Worst | Mean |
|---|---|---|---|
| **SSA** | **0.086** | **0.095** | **0.090** |
| **Dstack-SSA** | 0.175 | 0.208 | 0.189 |
| **T-Dstack-SSA** | 0.175 | 0.179 | 0.178 |

T-method is worse

# Results and Fututre Work

Results:

- Tensor-based methods are generally less dependent on the parameters choice, and has lower minimal and maximal RMSEs except for single-channel signal extraction case, where basic SSA has lower minimal RMSE
- Mean and median RMSE are close for both methods in the single-channel case but in the multi-channel case tensor methods have advantage
- Tensor methods with applying Dstack mapping can be useful for estimation of parameters in presence of strong noise

Future Work:

- Trying other tensor decompositions (CPD, T-SVD)
- Implementation of tensor modifications for other SSA-based methods
- ...

# Algorithms Complexities

$\mathcal{X} \in \mathbb{C}^{I \times L \times K}$, $\mathbf{X} \in \mathbb{C}^{\hat{L} \times \hat{K}}$, $I < L < K$, $\hat{L} < \hat{K}$,
$I + L + K = N + 2$, $\hat{L} + \hat{K} = N + 1$

- SVD$(\mathbf{X})$:
  $O(\hat{L}^2 \hat{K})$,
  or $O(r\hat{L}\hat{K})$ if only need $r$-rank approximation,
  or $O(rN\log(N))$ if $\mathbf{X}$ is Hankel
- HOSVD$(\mathcal{X})$:
  $O(ILKN)$,
  or $O(ILK(r_1 + r_2 + r_3))$ if only need $(r_1, r_2, r_3)$-rank approximation,
  or $O((r_1 + r_2 + r_3)I(L + K)\log(L + K))$ if $\mathcal{X}$ is Hankel