Robust Versions of the Complex SSA Method

Senov Mikhail

St Petersburg University
Department of Statistical Modelling

Science and Progress

St Petersburg, November 9, 2021

Introduction: Problem

$\mathsf{X}_N = (x_1, \ldots, x_N)$ time series, N series length.

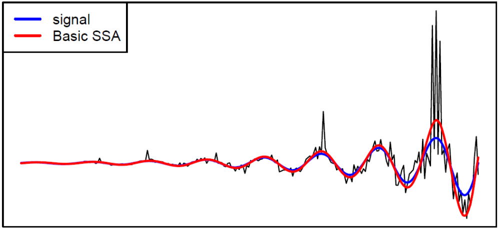
Model: $\mathsf{X}_N = \mathsf{S}_N + \mathsf{R}_N$, S_N signal, R_N noise.

Problem: Signal extraction $\tilde{\mathsf{S}} = F(\mathsf{X}_N)$, F is used method.

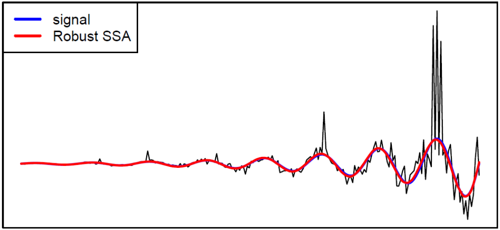
Method: SSA (Singular Spectrum Analysis) [Analysis of Time Series Structure: SSA and Related Techniques, Golyandina N., Nekrutkin V., Zhigljavsky A., 2001].

Introduction: Time series with outliers and SSA

Basic SSA: reacts to outliers.



Robust SSA: does not reacts to outliers.



Introduction: Aim of the talk

Robust SSA is developed for real-valued time series.

Aim: Generalize Robust SSA to the complex-valued time series case.

Common example of complex time series is engineering tasks where data are produced by Fourier transform.

$$\mathsf{X}_N^{(t)} = (x_1^{(t)}, \ldots, x_N^{(t)}),\ t = 1 \ldots M,$$

$$\mathsf{DFT}(\mathsf{X}_N^{(t)}) = f_k^{(t)} = \sum_{n=1}^{N} x_n^{(t)} e^{-2\pi i k n/N},\ k = 1 \ldots N,$$

$$\mathsf{F}_k^{(M)} = (f_k^{(1)}, \ldots, f_k^{(M)}).$$

Methods: Notations

Consider the time series $\mathsf{X}_N = (x_1, \ldots, x_N)$.

$\mathcal{M}_{\mathcal{H}}$ — space of hankel matrix $L \times K$,
$\mathcal{M}_r$ — space of matrix of rank at most r, $L \times K$.

Embedding operator $\mathcal{T} : \mathbb{R}^N \to \mathcal{M}_{\mathcal{H}} : \mathcal{T}(\mathsf{X}_\mathsf{N}) = \mathbf{X}$,

$\Pi_r : \mathcal{M} \to \mathcal{M}_r$ — projector onto the space of matrices of rank at most r in some norm in the matrix space,

$\Pi_{\mathcal{H}} : \mathcal{M} \to \mathcal{M}_{\mathcal{H}}$ — projector onto the space of hankel matrices in some norm in the matrix space.

Methods: Signal extraction

Time series $\mathsf{X}_N = (x_1, \ldots, x_N)$.

Algorithm for signal extraction

$$\tilde{\mathsf{S}} = \mathcal{T}^{-1}\Pi_{\mathcal{H}}\Pi_r\mathcal{T}(\mathsf{X_N}).$$

Norms for both Π_r and $\Pi_{\mathcal{H}}$:

- $\mathbb{L}_2$(Frobenius). Basic SSA signal extraction.
- $\mathbb{L}_1$. Robust SSA version.
- weighted $\mathbb{L}_2$. Robust SSA version.

Methods: Illustrative example

Example

$(x_1, \dots, x_N)$, norm $p(x)$, $\operatorname*{argmin}_c p((x_1 - c, \dots, x_N - c)) = ?$

Norms:

- $\mathbb{L}_2$. $\overline{x} = \operatorname*{argmin}_c \sqrt{\sum_{i=1}^N |x_i - c|^2}$.
  If $x_i \to \infty$, then $\overline{x} \to \infty$. Non-robust.
- $\mathbb{L}_1$. $\mathrm{med} = \operatorname*{argmin}_c \sum_{i=1}^N |x_i - c|$.
  If $x_i \to \infty$, then $\mathrm{med} \not\to \infty$. Robust.
- weighted $\mathbb{L}_2$. $\overline{x}_w = \operatorname*{argmin}_c \sqrt{\sum_{i=1}^N w_i |x_i - c|^2}$.
  If $x_i \to \infty$, then $w_i = 0$ and $\overline{x}_w \not\to \infty$. Robust.

$\mathbb{L}_2$ CSSA (Basic)

Trajectory matrix $\mathbf{Y} \in \mathbb{C}^{L\times K}$ as input.

$\mathbf{X}^{\mathrm{H}} = \overline{\mathbf{X}^{\mathrm{T}}}$.

Problem

$$\|\mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}}\|_F^2 \longrightarrow \min_{\mathbf{M},\mathbf{V}}, \ \mathbf{M} \in \mathbb{C}^{L\times r}, \mathbf{V} \in \mathbb{C}^{K\times r}$$

$$\mathsf{SVD}(\mathbf{Y}) = \sum_{i=1}^{r} \sqrt{\lambda_i} U_i V_i^{\mathrm{H}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathrm{H}},$$

U_i eigenvector of $\mathbf{Y}\mathbf{Y}^{\mathrm{H}}$, λ_i i-th eigenvalue of $\mathbf{Y}\mathbf{Y}^{\mathrm{H}}$, V_i eigenvector of $\mathbf{Y}^{\mathrm{H}}\mathbf{Y}$.

There is closed-from solution $\mathbf{M}\mathbf{V}^{\mathrm{H}}$, $\mathbf{M} = \mathbf{U}\mathbf{\Lambda}$.

$\mathbb{L}_1$ CSSA

Trajectory matrix $\mathbf{Y} \in \mathbb{C}^{L \times K}$ as input.

Problem

$$\|\mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}}\|_1 \longrightarrow \min_{\mathbf{M},\mathbf{V}}, \mathbf{M} \in \mathbb{C}^{L \times r}, \mathbf{V} \in \mathbb{C}^{K \times r}$$

There is no closed-from solution. Solve it iteratively

$$\mathbf{M}(\mathbf{t}) = \operatorname*{argmin}_{\mathbf{M} \in \mathbb{C}^{L \times r}} \|\mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}}(t-1)\|_1$$

$$\mathbf{V}(\mathbf{t}) = \operatorname*{argmin}_{\mathbf{V} \in \mathbb{C}^{K \times r}} \|\mathbf{Y} - \mathbf{M}(t)\mathbf{V}^{\mathrm{H}}\|_1.$$

Can be reduced to the linear case and solved by evaluating weighted-median.

Weighted $\mathbb{L}_2$ CSSA: Algorithm

Trajectory matrix $\mathbf{Y} \in \mathbb{C}^{L \times K}$ as input.

Iteratively compute matrix of weights $\mathbf{W} \in \mathbb{R}^{L \times K}$.

Problem

$$\|\mathbf{W}^{1/2} \odot (\mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}})\|_F^2 \longrightarrow \min_{\mathbf{M},\mathbf{V}}, \mathbf{U} \in \mathbb{C}^{L \times r}, \mathbf{V} \in \mathbb{C}^{K \times r}$$

There is not closed-from solution. Solve it iteratively

$$\mathbf{M}(\mathbf{t}) = \underset{\mathbf{M} \in \mathbb{C}^{L \times r}}{\operatorname{argmin}} \|\mathbf{W}^{1/2} \odot (\mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}}(t-1))\|_F^2$$

$$\mathbf{V}(\mathbf{t}) = \underset{\mathbf{V} \in \mathbb{C}^{K \times r}}{\operatorname{argmin}} \|\mathbf{W}^{1/2} \odot (\mathbf{Y} - \mathbf{M}(t)\mathbf{V}^{\mathrm{H}})\|_F^2$$

Can be reduced to the IRLS.

Weighted $\mathbb{L}_2$ CSSA: Weights computation

Outliers are determined by their absolute value.

Algorithm

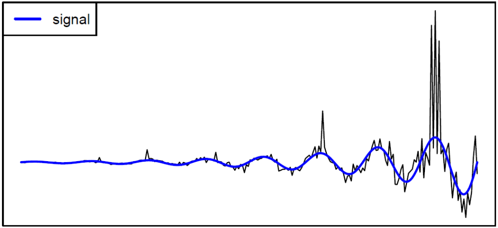
1. Residuals matrix computing $\mathbf{R} = \{r_{ij}\}_{i,j=1}^{n,p} = \mathbf{Y} - \mathbf{M}\mathbf{V}^{\mathrm{H}}$
2. Updating matrix $\mathbf{\Sigma} = \{\sigma_{ij}\}_{i,j=1}^{L,K}$
3. Computing weights $\mathbf{W} = \{w_{ij}\}_{i,j=1}^{L,K} = \{w(\frac{r_{ij}}{\sigma_{ij}})\}_{i,j=1}^{L,K}$, using

$$w(x) = \begin{cases} (1 - (\frac{|x|}{\alpha})^2)^2 & |x| \leq \alpha \\ 0 & |x| > \alpha \end{cases}$$

Algorithm similar to loess method.

Weighted $\mathbb{L}_2$ CSSA: Heteroscedastic noise specifics

Consider time series with heteroscedastic noise.



Right-sided non-outliers is bigger than left-sided outliers.

Basic: Normalizing coefficient σ_{ij} is the same for each series element. Outliers determination is incorrect.

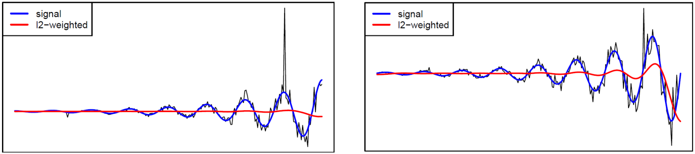
Modification: Compute σ_{ij} as trend of residual series.

Basic CSSA: R-package Rssa (Korobeynikov et al, 2020)

L1 CSSA: Was implemented in R language.

Weighted L2 CSSA: Was implemented in R language using real-valued implementation (Tretyakova, 2020).

A typo that was essential for the weighted $\mathbb{L}_2$ implementation was found in article (Chen et al, 2015) and corrected.



(a) Re (b) Im

Figure: weighted $\mathbb{L}_2$ version with typo

Example: Time series

Motivation:

- Demonstrate that Robust versions better than basic CSSA for some series with outliers.
- Demonstrate that modified weighted $\mathbb{L}_2$ version is better than the unmodified one for the case of heteroscedastic noise.

Example: Time series of length $N = 240$ with heteroscedastic noise,

$$f_n = e^{4n/N} e^{2n\pi/30i} + \frac{1}{2} e^{4n/N} \varepsilon_n, \ \varepsilon_n \sim CN(0,1),$$

with 0% outliers and 5% outliers with outlier value $5f_i$.

Example: RMSE comparison

Table: RMSE of different methods

Outliers	L2	L1	w-L2	mod. w-L2
0%	**1.28**	1.52	1.90	1.36
5%	6.14	1.78	1.66	**1.48**

All comparisons with the **best** meaningful for $\alpha = 0.05$.

Example conclusions:

- Basic CSSA demonstrates the best result without outliers,
- Robust versions demonstrates significantly better result than basic CSSA with outliers,
- Modified method shows better result than unmodified for heteroscedastic noise.

Conclusions

- Robust SSA versions were generalized to the complex-valued case.
- Complex Robust SSA versions were implemented in R language.
- Example demonstrating the effectiveness of the methods was considered.