On Multivariate Singular Spectrum Analysis and Its Variants

On Multivariate Singular Spectrum Analysis and its Variants*

Anish Agarwal
anish90@mit.edu
Massachusetts Institute of Technology
Cambridge, MA, USA

Abdullah Alomar
aalomar@mit.edu
Massachusetts Institute of Technology
Cambridge, MA, USA

Devavrat Shah
devavrat@mit.edu
Massachusetts Institute of Technology
Cambridge, MA, USA

ABSTRACT

We introduce and analyze a simpler, practically useful variant of multivariate singular spectrum analysis (mSSA), a known time series method to impute (or de-noise) and forecast a multivariate time series. Towards this, we introduce a spatio-temporal factor model to analyze mSSA. This model includes the usual components used to model dynamics in time series analysis such as trends (low order polynomials), seasonality (finite sum of harmonics) and linear time-invariant systems. We establish that given N time series and T observations per time series, the in-sample prediction error for both imputation and forecasting under mSSA scales as $1/\sqrt{\min(N,T)T}$. This is an improvement over: (i) the $1/\sqrt{T}$ error scaling of SSA, which is the restriction of mSSA to univariate time series; (ii) the $1/\min(N,T)$ error scaling for Temporal Regularized Matrix Factorized (TRMF), a matrix factorization based method for time series prediction. That is, mSSA exploits both the 'temporal' and 'spatial' structure in a multivariate time series. Our experimental results using various benchmark datasets confirm the characteristics of the spatio-temporal factor model and our theoretical findings—our variant of mSSA empirically performs as well or better compared to neural network based time series methods, LSTM and DeepAR.

ACM Reference Format:
Anish Agarwal, Abdullah Alomar, and Devavrat Shah. 2022. On Multivariate Singular Spectrum Analysis and its Variants. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/ PERFORMANCE '22 Abstracts), June 6–10, 2022, Mumbai, India.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3489048.3526952

1 INTRODUCTION

Multivariate time series data is of great interest across many application areas, including cyber-physical systems, finance, retail, healthcare to name a few. The goal across these domains can be summarized as accurate imputation and forecasting of a multivariate time series in the presence of noisy and/or missing data along with providing meaningful uncertainty estimates.

Setup. We consider a discrete time setting with time indexed as $t \in \mathbb{Z}$. For $N \in \mathbb{N}$, let $f_n : \mathbb{Z} \to \mathbb{R}$, $n \in [N] := \{1, \ldots, N\}$ be the latent time series of interest. For $t \in [T]$ and $n \in [N]$, we observe $X_n(t)$ where for $\rho \in (0, 1]$,

$$X_n(t) = \begin{cases} f_n(t) + \eta_n(t) & \text{with probability } \rho \\ \star & \text{with probability } 1 - \rho. \end{cases}$$

Here $\star$ represents a missing observation and $\eta_n(t)$ represents the per-step noise, which we assume to be an independent (across t, n) mean-zero random variable. Though $\eta_n(t)$ is independent, it is worth noting though the underlying time series, $f_n(\cdot)$, is of course strongly dependent across t, n.

Goal. The objective is two-folds, for $n \in [N]$: (i) imputation – estimating $f_n(t)$ for all $t \in [T]$; (ii) forecasting – learning a model to forecast $f_n(t)$ for $t > T$.

2 ALGORITHM

mSSA is a known method to impute and forecast multivariate time series. The main objective of this work is to introduce and theoretically analyze a simpler, practically useful variant of mSSA. Given the simplicity of the algorithm, we start by describing it below.

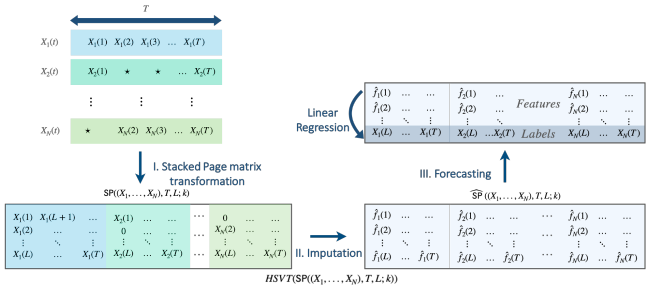

Figure 1: Key steps of proposed variant of mSSA.

There are two parameters: $L \geq 1$ and $k \geq 1$. For simplicity and without loss of generality, assume that T is an integer multiple of L. The algorithm consists of the following steps.

Step 1: Transform the time series into a matrix. we start by transforming each time series $X_n(t)$, $t \in [T]$ $n \in [N]$, into an $L \times T/L$ matrix where the entry of the matrix in row $i \in [L]$ and column $j \in [T/L]$ is $X_n(i + (j-1) \times L)$. This matrix induced by the time series is called the Page matrix. We then create a stacked Page matrix from these N time series by performing a column wise concatenation of the N matrices. We denote this matrix as $\text{SP}((X_1, \ldots, X_N), T, L)$, and note that it has L rows and $N \times T/L$ columns.

Step 2: Singular value thresholding (imputation). We replace missing values (i.e. $\star$s) in $\text{SP}((X_1, \ldots, X_N), T, L)$ by 0, then, we compute its singular value decomposition $\text{SP}((X_1, \ldots, X_N), T, L) = \sum_{\ell=1}^{\min(L, N \times T/L)} s_\ell u_\ell v_\ell^T$, where $s_1 \geq s_2 \cdots \geq s_{\min(L, N \times T/L)} \geq 0$ denote its ordered singular values, and $u_\ell \in \mathbb{R}^L, v_\ell \in \mathbb{R}^{N \times T/L}$ denote its left and right singular vectors, respectively,

*The full version of this paper is accessible through: https://arxiv.org/abs/2006.13448

SIGMETRICS/PERFORMANCE '22 Abstracts, June 6–10, 2022, Mumbai, India

ACM ISBN 978-1-4503-9141-2/22/06.
https://doi.org/10.1145/3489048.3526952

Table 1: Comparison of finite-sample results with relevant algorithms in the literature.

Method	Functionality	Mean Estimation	
	Multivariate time series	Imputation	Forecasting
This Work	Yes	$1/\sqrt{\min(N,T)T}$	$1/\sqrt{\min(N,T)T}$
mSSA - Literature	Yes	–	–
SSA [1, 3]	No	$T^{-1/4}$	–
Neural Network [5]	Yes	–	–
TRMF [4, 6]	Yes	$(\min(N,T))^{-1}$	–

for $\ell \in [\min(L, N \times T/L)]$. Let $\widehat{\rho}$ be the fraction of observed entries across all time series. Let the normalized, truncated version of $\mathrm{SP}((X_1, \ldots, X_N), T, L$ be

$$\widehat{\mathrm{SP}}((X_1, \ldots, X_N), T, L; k) = \frac{1}{\widehat{\rho}} \sum_{\ell=1}^{k} s_\ell u_\ell v_\ell^T,$$

From $\widehat{\mathrm{SP}}((X_1, \ldots, X_N), T, L; k)$, we can *read off* $\hat{f}_n(t)$ for $n \in [N]$, $t \in [T]$, the *de-noised and imputed estimate* of the N time series over T time steps.

Step 3: Lean a linear model (forecating). To forecast, we learn a linear model $\hat{\beta}((X_1, \ldots, X_N), T, L; k) \in \mathbb{R}^{L-1}$, which is the solution to

$$\text{minimize} \quad \sum_{m=1}^{N \times T/L} (y_m - \beta^T x_m)^2 \quad \text{over} \quad \beta \in \mathbb{R}^{L-1},$$

where y_m is the mth component of $(1/\widehat{\rho})[X_1(L), \quad X_1(2 \times L), \ldots, X_1(T), \quad X_2(L), \ldots, X_2(T), \ldots, X_N(T)] \in \mathbb{R}^{N \times T/L}$, and $x_m \in \mathbb{R}^{L-1}$ corresponds to the vector formed by the entries of the first $L-1$ rows in the mth column of $\widehat{\mathrm{SP}}((X_1, \ldots, X_N), T, L; k)$ for $m \in [N \times T/L]$. Note, to define y_m, we impute missing values in $X_1, \ldots, X_n$ by 0. Then the *forecasted estimate* at time $t = L \times m'$ for $m' \in [T/L]$ for time series $n \in [N]$ is $\bar{f}_n(L \times m') = \hat{\beta}((X_1, \ldots, X_N), T, L; k)^T x_m$ where $m = m' + (n-1) \times T/L$. Figure 1 has a visual depiction of the key steps above.

3 MODEL AND RESULTS

This variant of the mSSA algorithm is arguably quite simple, with its major steps consisting of singular value thresholding and ordinary least squares. Despite its simplicity, we find it to perform competitively or outperforms popular baselines such as LSTM and DeepAR. However, aside from the recent theoretical analysis of SSA in [1, 2], theoretical explanation of the success of mSSA has been absent. This begs the question: *When and why does mSSA work?* As our primary contribution, we provide an answer to this question under a spatio-temporal factor model which we introduce next.

Spatio-temporal factor model. Note that the collection of latent multivariate time series $f_n(t)$, for $n \in [N]$, $t \in [T]$ can be collectively viewed as a $N \times T$ matrix. To capture the spatial structure, i.e. the relationship across rows, we model this matrix to be low-rank – there exists a low-dimensional latent factor associated with each of N time series; analogously, there exists a low-dimensional latent factor associated with each of the T time steps. To capture the temporal structure, we further assume that each component of the latent temporal factor has a *(approximately) low-rank Hankel matrix*. This proposed model captures a variety of traditional models in time series analysis, which posits that a time series is a mixture of a trend (low-order polynomial), a seasonal (finite sum of harmonics), and a stationary (linear time-invariant function) component. We show that each of these components indeed has a low-rank Hankel (or Page) matrix representation. Further, we establish that the set of time series that have a (approximately) low-rank Hankel representation is closed under component-wise addition and multiplication. Such a model *calculus* helps characterize the representational strength of the proposed model.

Finite sample analysis of mSSA. Under the spatio-temporal factor model, we establish that the imputation and (in-sample) forecasting prediction error scale as $1/\sqrt{\min(N,T)T}$.For $N = 1$, it implies that the univariate SSA algorithm has imputation and forecasting error scaling as $1/\sqrt{T}$. That is, mSSA improves performance by a $\sqrt{N}$ factor over SSA by utilizing information across the N time series. This also improves upon the prior work of [1] which established the weaker result that SSA has imputation error scaling as $1/T^{\frac{1}{4}}$—also, [1] does not establish a result for the forecasting error of SSA. Further, existing matrix estimation or completion based methods applied to the $N \times T$ matrix of time series observations (i.e, without first performing the Page matrix transformation as done in mSSA) establish that the imputation prediction error scales as $1/\min(N,T)$. See Table 1 for a summary of our theoretical results.

Extensions: variance and tensor SSA (tSSA). First, we extend mSSA to estimate the latent time-varying variance, i.e. $\mathbb{E}[\eta_n^2(t)]$, $n \in [N]$, $t \in [T]$. We establish the efficacy of such an extension when the time-varying variance is also modeled through a spatio-temporal factor model. This enables meaningful uncertainty quantification for the estimation of the mean produced by mSSA. Second, we propose a tensor variant of SSA, termed tSSA, which exploits recent developments in tensor estimation. In tSSA, rather than doing a column-wise stacking of the Page Matrices induced by each of the N time series to form a larger matrix, we instead view each Page matrix as a slice of a $L \times T/L \times N$ order-three tensor. We characterize the relative performance of tSSA, mSSA, and "vanilla" matrix estimation (ME) with respect to imputation error.

REFERENCES

[1] Anish Agarwal, Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. 2018. Model Agnostic Time Series Analysis via Matrix Estimation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 3 (2018), 40.

[2] Anish Agarwal, Devavrat Shah, Dennis Shen, and Dogyoon Song. 2021. On Robustness of Principal Component Regression. *Accepted to appear in Journal of the American Statistical Association* (2021).

[3] Nina Golyandina, Vladimir Nekrutkin, and Anatoly A Zhigljavsky. 2001. *Analysis of time series structure: SSA and related techniques.* Chapman and Hall/CRC.

[4] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. 2015. Collaborative Filtering with Graph Information: Consistency and Scalable Methods. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2107–2115.

[5] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* (2019).

[6] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in neural information processing systems*. 847–855.