

Tensor decomposition and forecast for multidimensional time series

Kirill Semkin^{1*} and Vadim Strijov^{2*}

¹*Forecsys® Moscow Russia.

²*Forecsys® Moscow Russia.

*Corresponding author(s). E-mail(s): kirill.semkin32@mail.ru;
strijov@ccas.ru;

Keywords: time series, decomposition, forecast, SSA, CPD

Processing of multidimensional time series is associated with an additional task of determining dependencies between signals. Its inclusion in models boosts the quality of forecasts. On the other hand, taking this dependency into account makes models more complex and less interpretable. The paper proposes a non-parametric method based on tensor data representation and Singular Spectrum Analysis (SSA). It derives a time series decomposition technique and an explicit forecast model. Finally, it applies an elaborated theory to electricity consumption and inertial measurement unit datasets. It compares the obtained quality of forecast with mSSA, VAR, and RNN models.

1 Introduction

The main object of the paper is multivariate time series — a set of m of them $\{x_i(t)\}_{i=1}^m$ observed in discrete time $t \in 1, \dots, N$. We state two problems concerning them. First, make the *forecast* — propose future values $\{x_i(T)\}_{i=1}^m$ of the series at time $T > N$. Second, build the additive *decomposition* — represent each signal in the set as a sum of several components: $x_i(t) = c_1(t) + \dots + c_s(t)$, $\forall i \in 1, \dots, m$.

As for the single series, there are methods to resolve stated problems. Papers [1–3] introduce seasonal-trend-cycle techniques for the decomposition. In [5] authors propose autoregressive methods as well as works [1, 4] suggest exponential smoothing, regression models, neural networks and more for the forecasting. However, these methods can not be directly transferred to the multivariate signals if they are interconnected. Paper [7] models the size of prey and predator populations as a coupled system of differential equations. The dynamics of one population depends directly on the other's and vice-versa. Therefore they can not be analyzed separately.

Some methods take into account series connections and make the forecast simultaneously. First, consider *recurrent neural network* (RNN) model [8]. It connects series and their past values through several layers of nonlinear transformations. This information is encapsulated in the hidden state vector at each time step. With that, one is able to forecast future values [9].

Second, *vector autoregression* (VAR) [10, 11] is a linear stochastic model for multivariate series. Denoting vector $\mathbf{x}_t = (x_1(t) \dots x_m(t))^\top$ as a series realizations at time t , their further dynamics is the following:

$$\mathbf{x}_t = \boldsymbol{\mu} + \sum_{i=1}^p A_i \mathbf{x}_{t-i} + \mathbf{u}_t$$

Here $\boldsymbol{\mu}$ — some constant vector, A_i are matrices $m \times m$, \mathbf{u}_t is a random vector (e.g. white noise $WN(t)$). Signal's connection is defined by A_i matrices, so each series is dependent on each other and their past linearly.

The mentioned methods are able to make the forecast, but, firstly, have a large number of learnable and structural parameters. That means their underlying models have an extensive complexity and their architecture needs to be picked over and tuned. Secondly, the methods' structure does not contain an explicit way of building series decomposition.

To rectify the listed disadvantages, we have developed a new method called *tensor SSA* (tSSA) which has only two adjustable parameters. It extends the SSA method [12] for multivariate time series. That implies using dynamical systems theory as a generative model of the observed series. Namely, finding shared *signal subspace* [13, 14] for all given series is the main task of our paper. This is a low-dimensional linear subspace where all the system's trajectories can be mapped into isomorphically. Finding it enables us to obtain phase representation of observed signals. This idea is illustrated on the fig. 1. Having one series, we use it to build phase space and the specific trajectory of the dynamical system that generated a given series. The forecast is found upon the fact that the continuation of the trajectory lies in the very same phase space. Clustering basis vectors of this space into several groups builds us the decomposition of the series. Named conceptions are developed further in the manuscript.

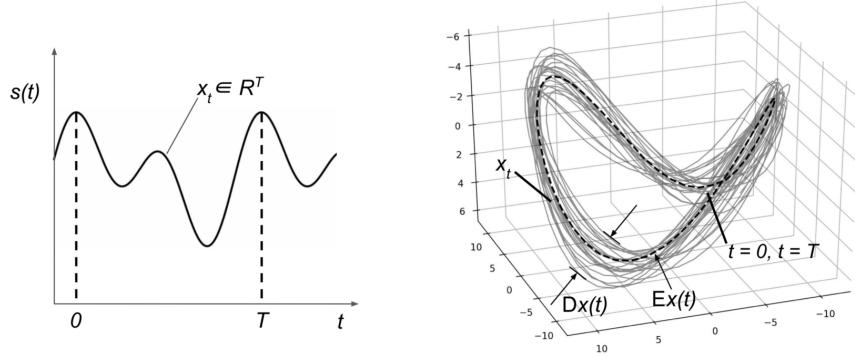


Figure 1 Time series's phase trajectory visualization

Another feature of our method is tensor data representation and utilizing *canonical polyadic decomposition* (CPD) to build signal subspace. In the work [15] authors extend SSA as well but use matrix representation and decomposition instead. Their method is called matrix SSA (mSSA) and it presents different way of making the forecast and the decomposition for multivariate time series. The tSSA is compared with this method too in application to the real data in a later section.

The rest of the paper covers the theory and application of our method to real data. Firstly, the dynamical systems model is introduced and the problem of basis search in the signal subspace is stated and resolved. These results enable us to propose a way of decomposing the signals and making the forecast. Simultaneously, features of obtained solutions are examined and two associated theorems are formulated. Finally, tSSA and mentioned methods are applied to two datasets: electricity consumption and accelerometer/gyroscope observations. We obtain their predictions and additive decompositions. For the latter, a special quality metric is introduced. All results are backed with figures and discussion.

2 Problem statement

Let us have a set of time series $\{x_i(t)\}_{i=1}^m$ generated by *dynamical system* f . By that we mean a law of coordinates $\mathbf{y} \in X$ evolution in discrete time:

$$\mathbf{y}(t+1) = f(\mathbf{y}(t)), \quad t \in \mathbb{N}$$

$$\mathbf{y}(0) = \mathbf{y}_0$$

Generally, X is a high dimensional smooth manifold. Then, each trajectory of the system spawns our signals through unknown mapping $\phi : X \rightarrow \mathbb{R}^m$:

$$\phi(\mathbf{y}(t)) = \mathbf{x}_t \Leftrightarrow \begin{cases} \phi_1(\mathbf{y}(t)) = x_1(t) \\ \dots \\ \phi_m(\mathbf{y}(t)) = x_m(t) \end{cases}$$

After that, it is hypothesized that trajectories $\mathbf{y}(t)$ actually belong to a smaller dimension manifold $M \subset X$. So now we should find an embedding of M into \mathbb{R}^L for some L and discover a basis in the image of the embedding. Having done that, we will obtain characterization of the initial system in terms of the standard linear space. The same will be equally applied to all $x_i(t)$.

2.1 One signal case

Our future approach roots from SSA method and Takens's theorem [16]. Let's describe it in short. The theorem provides a solution for the stated problem in single series case: we associate any point $\mathbf{y}(t) \in M$ with the following vector:

$$(\phi \circ f^{t-L+1}(\mathbf{y}(t)), \dots, \phi \circ f(\mathbf{y}(t)), \phi \circ \mathbf{y}(t))^T = (x(t-L+1), \dots, x(t-1), x(t))^T$$

It is called *delay vector* at time t and is denoted as $\overleftarrow{\mathbf{x}}_t$. Their dimension L should satisfy $L > 2 \cdot \dim(M)$. Function $\phi(\cdot)$ should also meet some regularity conditions which are considered to be fulfilled.

So, with signal $x(t)$ of length N we build $N - L + 1$ delay vectors. Now we can get what we wanted. Corresponding embedding space is $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\})$ (so called *signal subspace*) and must be low dimensional, meaning $\text{Lin}(\{\overleftarrow{\mathbf{x}}_t\}) \subset \mathbb{R}^L$. Orthonormed basis here is a SVD's U -component of *trajectory matrix* \mathbf{H}_x which is build from delay vectors:

$$\mathbf{H}_x = [\overleftarrow{\mathbf{x}}_1 \dots \overleftarrow{\mathbf{x}}_{N-L+1}]$$

2.2 Series interconnection and tSSA method

Now let's generalize previous results for the multiple series.

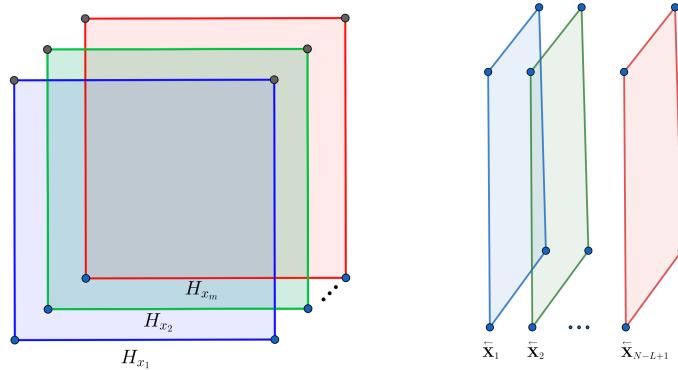


Figure 2 Two views on trajectory tensor. The left is in terms of signals' trajectory matrices $\{x_i(t)\}_{i=1}^m$. The right is in terms of delay matrices.

In this case $\phi(\cdot)$ is multidimensional. So instead of the single delay vector we should consider their set for all m signals at time t . That means the embedding image now contains so called *delay matrices* $(\tilde{\mathbf{x}}_{1_t} \dots \tilde{\mathbf{x}}_{m_t}) := \tilde{\mathbf{X}}_t$. Next, instead of trajectory matrix we should introduce *trajectory tensor* \mathbf{T} . It is constructed by putting delay matrices along the second dimension of the tensor, fig. 2 left. This procedure is equivalent to the one for \mathbf{H}_x from the previous subsection. But the tensor has an important property: \mathbf{T} can equally be built as an alignment of signals trajectory matrices \mathbf{H}_{x_i} along the third dimension, fig. 2 right.

Then, reasoning analogues to the one signal case we obtain *signals' set subspace* as a linear span of delay matrices. But we were supposed to built the subspace for each signal individually.

Before doing so, it's time to give a definition for signals interrelation in our model's scope.

Definition. We call a set of time series *interconnected* if they all share the common subspace with the same basis.

Now we are ready to find this shared elements for the signals. Actually, we just need to decompose each signal's trajectory matrix through the same set of factors. Let's apply CPD to our trajectory tensor and have a look at its slices along the third dimension.

$$\mathbf{T} = \sum_{i=1}^r \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \Leftrightarrow \begin{cases} \mathbf{H}_{x_1} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_1}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \mathbf{H}_{x_2} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_2}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \dots \\ \mathbf{H}_{x_m} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_m}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \end{cases} \quad (1)$$

CPD is defined by tensor rank r and a set of vectors which we can pack in matrices: $A = [\mathbf{a}_1 \dots \mathbf{a}_r]; B = [\mathbf{b}_1 \dots \mathbf{b}_r]; C = [\mathbf{c}_1 \dots \mathbf{c}_r]$. Row k of matrix C is denoted as $\boldsymbol{\sigma}_{x_k}$. It has the sense of singular values but may contain negative numbers. After, from CPD definition as a (1)-kind of decomposition with minimal r we can derive that A, B, C are full-rank matrices.

Most importantly, we have obtained signals' shared subspace $\text{Lin}(\{\mathbf{a}_i\})$ with unified basis $\{\mathbf{a}_i\}_{i=1}^r$. Now it is not orthonormed in general. Then, back in the problem statement we assumed dimensionality of the embedding to be rather small. Therefore relation $r \ll L$ should be true.

Lastly, let's note that *rows* of \mathbf{H}_{x_k} are also delay vectors but have another dimension $N - L + 1$. So the shared subspace for them would be $\text{Lin}(\{\mathbf{b}_i\})$. This result proves our approach to be consistent with choice of delay vectors. As a remark, mSSA lacks such feature.

What to the signals' linkage, it is regulated by $\boldsymbol{\sigma}_{x_k}$. For instance, if all of them have zero elements in non-overlapping positions then our built subspace breaks down into a direct sum of individual signal's subspaces. It is a total linkage absence case. The exact opposite situation is when $\boldsymbol{\sigma}_{x_k}$ are non-zero for all signals.

2.3 Time series decomposition

We can break series into several components using following conception: *decomposition of trajectory matrix \mathbf{H}_{x_k} defines decomposition of related signal*. Its first part implies representation of \mathbf{H}_{x_k} as a sum of factors (1). This technique comes from SSA. Because all matrices are similarly decomposed, we can focus on the task for a single series.

Before diving into, we should point out the main feature of trajectory matrices — *hankelness*. That simply means equal elements along each anti-diagonal. So it is trivial that every series of length N bijectively corresponds to hankel matrix of size $L \times (N - L + 1)$.

Now, decomposition procedure is the following: we assume factors of \mathbf{H}_{x_k} to be able to pack in s groups in a special way. That means if we sum all factors within each group the result matrices C_1, \dots, C_s will be hankel. This situation would perfectly correspond to representation of $x_k(t)$ as a sum of s signals. Unfortunately, it is almost infeasible even with trivial series [12], so every C_i should be additionally *hankelized*. This operation averages each matrix's anti-diagonals so it becomes hankel. Let's denote it as $\text{Hankel}(\cdot)$ operator.

The whole algorithm can be written as a chain of identical transformations:

$$\begin{aligned} \underline{\mathbf{H}}_{x_k} &\stackrel{1}{=} \sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \stackrel{2}{=} \sum_{i \in \mathbb{I}_1} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top + \dots + \sum_{i \in \mathbb{I}_s} \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \stackrel{3}{=} \\ &\stackrel{3}{=} C_1 + \dots + C_s \stackrel{4}{=} \underline{\text{Hankel}(C_1) + \dots + \text{Hankel}(C_s)} \Leftrightarrow x_k(t) = c_1(t) + \dots + c_s(t) \quad (2) \end{aligned}$$

Here $\mathbb{I}_1 \sqcup \dots \sqcup \mathbb{I}_s = \{1, \dots, r\}$ — chosen groups of indices. Actually, the forth equality needs some extra validation. To make it, first consider hankel operator property:

Lemma 1. *Hankel(\cdot) operator is linear.*

Proof. Having matrices A and B with the same size, let's look at their elements from some anti-diagonal: a_1, \dots, a_n and b_1, \dots, b_n . Then this anti-diagonal in $\text{Hankel}(A+B)$ is exactly $\frac{1}{n} \sum_{i=1}^n (a_i + b_i) = \frac{1}{n} \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n b_i$, resulting in a sum of $\text{Hankel}(A)$ and $\text{Hankel}(B)$ anti-diagonals.

Secondly, having a scalar α it is also trivial that $\frac{1}{n} \sum_{i=1}^n \alpha \cdot a_i = \alpha \frac{1}{n} \sum_{i=1}^n a_i$. Then we derive $\text{Hankel}(\alpha A) = \alpha \text{Hankel}(A)$. \square

Returning to the forth equality in (2), we can combine this lemma and the fact that \mathbf{H}_{x_k} is itself hankel to get a complete justification. This outputs in the correctness of series decomposition procedure for any chosen grouping of factors.

2.4 Optimal decomposition and its computational complexity

Even though decomposition algorithm from the previous subsection is correct for any factors' grouping we might want to find the best one. This would mean every C_i to be as close to hankel matrix as possible. Having that, the formulated decomposition concept would be fully accomplished and signal's components could be naturally extracted. Let's express it as some optimization task.

Consider the first equality from (2) and itself with hankel operator applied. Also recall that \mathbf{H}_{x_k} is hankel. We obtain:

$$\begin{cases} \mathbf{H}_{x_k} = \sum_{i=1}^r \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top \\ \mathbf{H}_{x_k} = \sum_{i=1}^r \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top) \end{cases}$$

Subtract the second equation from the first and denote $H_i = \boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top - \text{Hankel}(\boldsymbol{\sigma}_{x_k}(i) \cdot \mathbf{a}_i \mathbf{b}_i^\top)$. This can be sensed as a residual between initial factor and its hankelized version. Now we get a general relation between them:

$$H_1 + \dots + H_r = 0 \Leftrightarrow H_r = - \sum_{j=1}^{r-1} H_j \quad (3)$$

With that, we can reformulated condition on C_i to be hankel in terms of residual matrices summing into zero inside groups: $\sum_{k \in \mathbb{I}_i} H_k = 0, \forall i \in 1, \dots, s$.

Now, let's simplify our case by decreasing the number of desired components from s to two. If we worked out the easier problem we could consecutively dissect components into two more and have as much of them as we want.

Next, (3) shows all residual matrices already total in zero. Therefore we can discard, for example, H_r as group for it would be defined automatically if one had groups for the rest factors. Let's also introduce indicator-variables $\beta_j \in \{0, 1\}$ showing in what of two groups factor j lies.

Finally, we assume all H_i to be non-zero; otherwise we already have needed dissection. This last condition implies each group to have at least two residual matrices.

We are ready for the optimal decomposition problem statement:

$$\begin{cases} \sum_{j=1}^{r-1} \beta_j H_j = 0 \\ \beta_j \in \{0, 1\}, \forall j \in 1, \dots, r \\ \sum_{i=1}^{r-1} \beta_j \geq 2 \end{cases} \quad (4)$$

To rewrite it in a more familiar way let's vectorize each H_i and put resulting vectors in a matrix Λ . Furthermore, let's introduce $\beta = (\beta_1 \dots \beta_{r-1})$. Then, recall from the previous subsection that it is highly unlikely for residual matrices to perfectly total in zero within all groups. The most we are capable of is to make them as close to zero as possible. So restated problem:

$$\begin{cases} \|\Lambda\beta\| \rightarrow \min_{\beta} \\ \beta_j \in \{0, 1\}, \forall j \in 1, \dots, r \\ \sum_{i=1}^{r-1} \beta_j \geq 2 \end{cases} \quad (5)$$

Let's declare solution of this task as an *optimal series decomposition*. It is represented as Integer Least Squares problem (ILS) which is proved to be NP-hard [17]. In short, subsection's output is:

Theorem 1. *Optimal series decomposition as a special grouping of trajectory matrix factors is NP-hard.*

In spite of the result, methods and heuristics exist to reduce ILS to computationally effective objectives [18].

2.5 Forecasting

In our model prediction making comes down to the restoration of delay vectors for every signal in a set. Because we have the shared basis $\{\mathbf{a}_i\}_{i=1}^r$ among the series we can process them separately.

If $\{\mathbf{a}_i\}_{i=1}^r$ was orthonormal we could inherit SSA prediction technique [12]. But we don't have this warranty so we are going to adopt for the current case.

Having input series $x(t)$ we want to forecast it for one step further. Delay vector $\overleftarrow{\mathbf{x}}_{N+1} = (x(N-L) \dots x(N+1))^T$, with the last component unknown, belongs to the $\text{Lin}(\{\mathbf{a}_i\})$. In terms of full-ranked matrix $A \in \mathbb{R}^{L \times r}$ (see section 2.2) it can be written as:

$$\begin{aligned} \overleftarrow{\mathbf{x}}_{N+1} = A\boldsymbol{\lambda} &\Leftrightarrow \begin{cases} \mathbf{x}_{kn} = A_{kn}\boldsymbol{\lambda} \\ x(N+1) = \mathbf{a}_{pr}^T\boldsymbol{\lambda} \end{cases}, \text{ where} \\ A &= \begin{pmatrix} A_{kn} \\ \mathbf{a}_{pr}^T \end{pmatrix} \\ \overleftarrow{\mathbf{x}}_{N+1} &= (\mathbf{x}_{kn} \ x(N+1))^T \end{aligned} \tag{6}$$

Here variable $\lambda \in \mathbb{R}^r$. Delay vector and matrix are broken into two blocks associated with the *known* and *unknown* part of the series. To make prediction we need to find λ so we focus on the first obtained equation.

The matrix of this linear system $A_{kn} \in \mathbb{R}^{(L-1) \times r}$ has a rank at least $r - 1$ by its construction. But we assume elimination of one row in A will actually keep the rank full. As experiments show this is not burdensome. After that, we note that the linear system itself is overdetermined due to the model's hypothesis $r \ll L$. Therefore it is reasonable to get solution in a least squares sense: $\lambda = (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn}$. After substituting in the second equation, forecast is given by the formula:

$$x(N+1) = \mathbf{a}_{pr}^T (A_{kn}^T A_{kn})^{-1} A_{kn}^T \mathbf{x}_{kn} \quad (7)$$

Here $\mathbf{a}_{pr}^T (A_{kn}^T A_{kn})^{-1} A_{kn}^T = \mathbf{d}^T$ — row vector, and once computed can be reused to make prediction for arbitrary horizon ahead.

Now let's rewrite (7) in terms of $x(t)$:

$$x(t) = \sum_{i=1}^{L-1} d_i \cdot x(t-i)$$

This expression leads to the final result about the forecast behavior:

Theorem 2. *The model of tSSA's forecast is autoregressive AR($L - 1$). It can be fully characterized by coefficients d_i found in (7).*

3 Computational experiment

3.1 Goals and setting

In the current section we will demonstrate application of our method for decomposition and forecast of multidimensional time series. We will also compare it with the other models: mSSA, VAR, RNN. The latter two are only able for prediction. The mSSA has a lot in common with our approach and is valid for both tasks.

To measure quality of the decomposition we need special metrics. They are motivated by the corresponding analysis of our method (see section 2.3) :

Definition. *Absolute hankel error* of matrix M is

$$\text{AHE} = \|M - \text{Hankel}(M)\|_2$$

Definition. *Relative hankel error* of matrix M is

$$\text{RHE} = \frac{\text{AHE}}{\|M\|_2}$$

AHE can be imagined as a total standard deviation of a matrix anti-diagonals. RHE is its more convenient modification which will be used later. Now, recall that the series decomposition on additive components is equivalent to trajectory matrix decomposition, viz. obtaining matrices C_i (see section 2.3). So the RHE will be calculated for them as well as mean RHE for every series in a set — $\overline{\text{RHE}}_{ts}$, and the mean among all series — $\overline{\text{RHE}}$.

To measure forecast quality for the single series we will use standard metrics: *mean squared error* MSE_{ts} and *mean absolute percentage error* MAPE_{ts} . Their averaging for all series are denoted as $\overline{\text{MSE}}$ and $\overline{\text{MAPE}}$.

Two sets of data will be involved in experiment. First, electricity consumption paired with its generation price (fig. 3). Second, six signals from the motion sensors — accelerometer and gyroscope measurements for walking movements [19] (fig. 4). We assume signals inter-dependency based on economical/physical reasons as it is important premise for our approach (see section 2.2). Individual series has a length $\approx 3 \cdot 10^3$ and $\approx 1.5 \cdot 10^3$ where last 20% are deferred test samples for prediction evaluation.

We should also describe peculiarities of tSSA application. Firstly, tensor rank computation is NP-hard [20], so its number will be explicitly indicated whenever needed. Secondly, the used algorithm for CPD is ALS [21] and it has an approximation error.

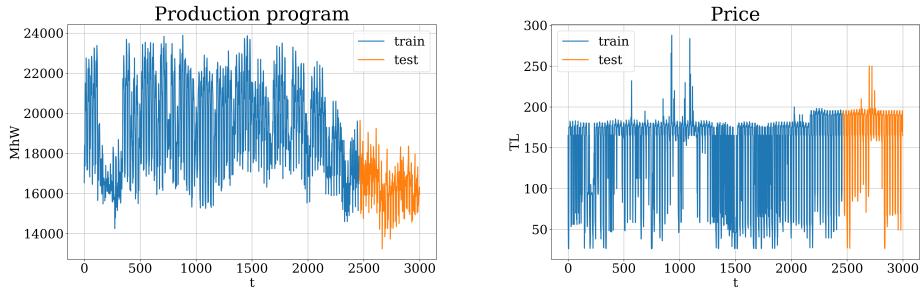


Figure 3 Time series for electricity consumption and its price

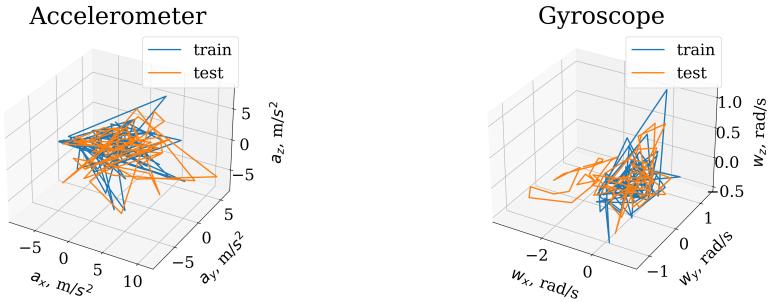


Figure 4 Time series for linear and angular accelerations

Thanks for that decomposition (1) is not precise. So this relative errors will be specified below. Thirdly, ILS problem (5) must be solved to find series decomposition. But matrix involved in it has a large row dimension. To boost computation we will cut it to several hundreds which is still much bigger then the size of sought parameters vector.

Finally, let's fix value L unified for all models. It is generally responsible for the size of series' history on which methods can be conditioned. For tSSA and mSSA it was described in the beginning of the section 2. For RNN it is the length of encoder. For VAR it is the maximum lag number. We pin it to be $L = 500$ for electricity and $L = 1000$ for motion.

3.2 Data availability

Electricity consumption dataset is available at <https://sourceforge.net/p/mvr/code/HEAD/tree/data/TurkElectricityConsumption.csv>. Accelerometry data is available at <https://data.mendeley.com/datasets/45f952y38r/5>.

3.3 Results and discussion

Let's begin with the forecast results. First, relation between prediction metrics for tSSA and trajectory tensor's rank is depicted on the fig. 5 and 6. The common pattern for MSE and MAPE is that forecast quality decreases with greater ranks. Furthermore, overtraining effect for MAPE can be seen in both cases as metrics increase after reaching minimum. Nonetheless, CPD approximation error deceases monotonously, see fig. 7.

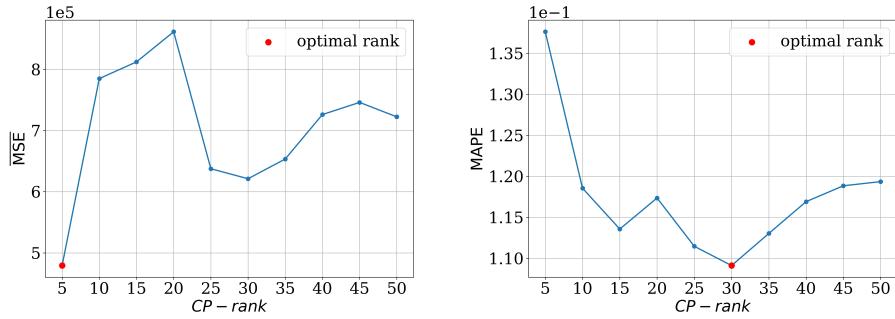


Figure 5 MSE and MAPE metrics for tSSA forecast depending on CPD rank. Optimal point is marked with red. Electricity data.

The best tSSA forecast is shown on fig. 8 and 9. For electricity, CPD rank is rather high and so the order of the forecast model (see section 2.5). It allows for a detailed prediction. On the other hand, high rank for the motion data could actually led to unstable and unbound trajectories. But with the smaller rank model has fewer degrees of freedom. Consequently, we can obtain only averaged values for the future.

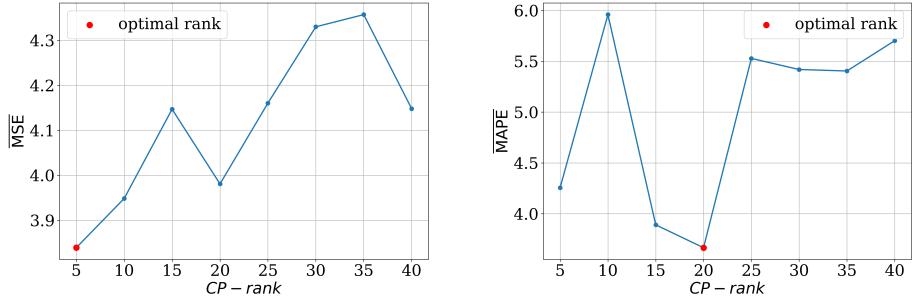


Figure 6 MSE and $\overline{\text{MAPE}}$ metrics for tSSA forecast depending on CPD rank. Optimal point is marked with red. Motion data.

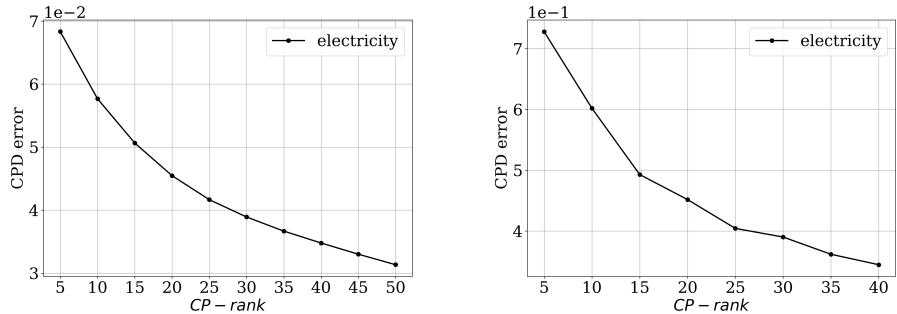


Figure 7 Relative CPD approximation errors depending on CPD rank. Left is for electricity, right is for motion.

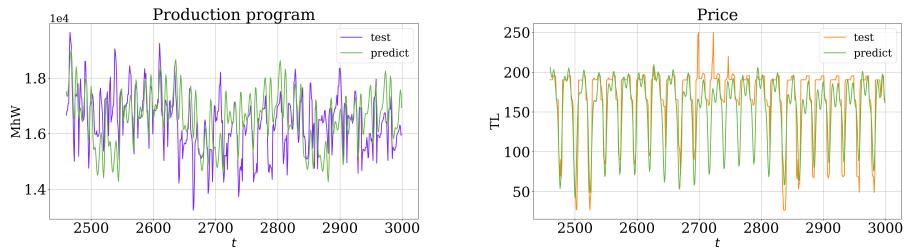


Figure 8 tSSA forecast for electricity data. CPD rank = 30

Final forecast metrics for all models are in tab. 1 and 2. In most cases our method showed the best output. Close values were obtained with mSSA. The VAR model appeared unstable for the chosen prediction horizons. RNN was able to learn only the constant function from the data for electricity, but large window size for accelerometry enabled it to make a better prediction.

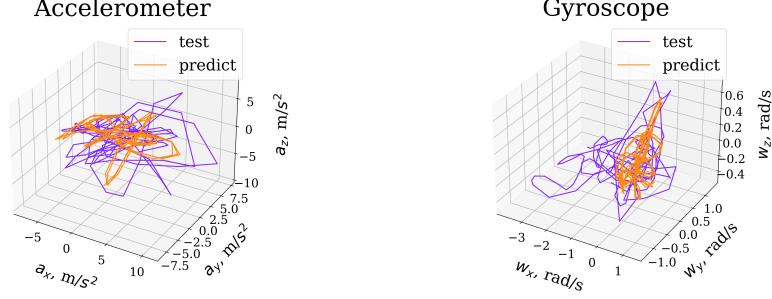


Figure 9 tSSA forecast for motion data. CPD rank = 20

Table 1 Prediction quality of models for electricity data

	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Production}}, 10^6$	1.24	1.51	7.81	2.70
$\overline{\text{MSE}}_{\text{Price}}, 10^3$	0.88	1.03	4.85	30.0
$\overline{\text{MSE}}, 10^6$	0.62	0.75	3.91	135.00
$\overline{\text{MAPE}}_{\text{Production}}$	0.054	0.060	0.137	0.999
$\overline{\text{MAPE}}_{\text{Price}}$	0.164	0.170	0.360	1.004
MAPE	0.109	0.115	0.249	1.002

Table 2 Prediction quality of models for motion data

	<i>tSSA</i>	<i>mSSA</i>	<i>VAR</i>	<i>RNN</i>
$\overline{\text{MSE}}_{\text{Accel}}$	7.351	6.980	8.108	6.604
$\overline{\text{MSE}}_{\text{Gyro}}$	0.610	0.636	0.631	0.639
$\overline{\text{MSE}}$	3.981	3.808	4.370	3.622
$\overline{\text{MAPE}}_{\text{Accel}}$	3.558	3.516	3.370	1.747
$\overline{\text{MAPE}}_{\text{Gyro}}$	3.773	3.943	10.427	5.641
MAPE	3.666	3.730	6.899	3.694

Now we move to the decomposition. Fig. 10 illustrates relation between quality metrics and CPD rank. In both cases $\overline{\text{RHE}}$ rapidly reaches a minimum and then does not change or increase. Recalling the same result for the forecast we can conclude that tSSA has the greatest generalization capacity for the small tensor ranks. At the same time, the rank is proportional to the signal subspace dimension (see section 2). Combining, we have an evidence that the major premise of our method — small real dimension of the data — holds.

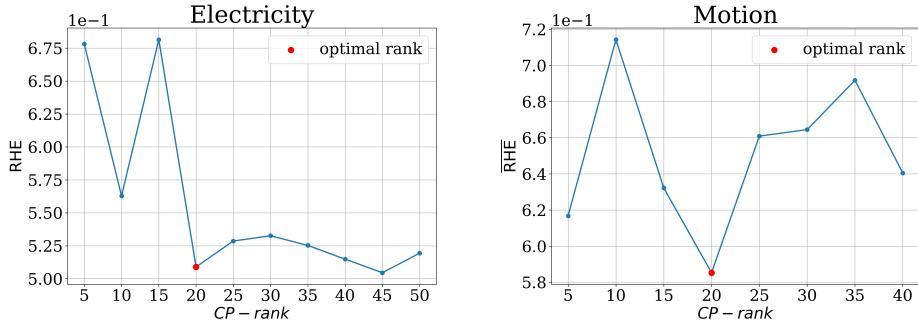


Figure 10 Relation between $\overline{\text{RHE}}$ metrics and CPD rank. The left is for electricity data, the right is for motion data. Optimal point is marked with red

One the fig. 11, 12, 13 one can see the best tSSA decomposition on two components. Due to rather long computation of ILS we do not make more. But it is actually not necessary to catch the notion of a shared basis for observed signals. For example, decomposition of electricity generation and its price are almost identical up to the bias and scale. The same situation can be observed for the second components of the linear and angular accelerations, but the first has its own shape. Now, let's interpret obtained results. On the fig. 11 we have two high-frequency harmonics with various amplitudes and biases. On the fig. 12, 13 we can see rather tangled periodical trajectories. The first components have a much smaller scale then the second so they may represent some subtle part of walking motions. So, all this implies complex dynamical system behind the scenes.

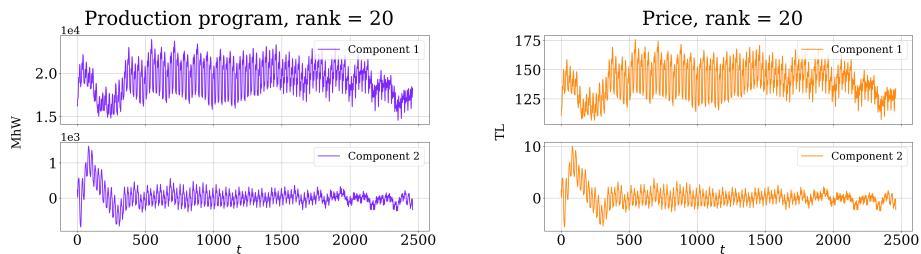


Figure 11 tSSA decomposition for electricity data. CPD rank = 20

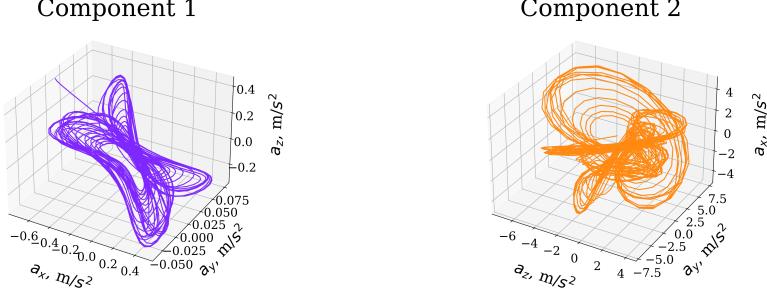


Figure 12 tSSA decomposition for accelerometer data. CPD rank = 10

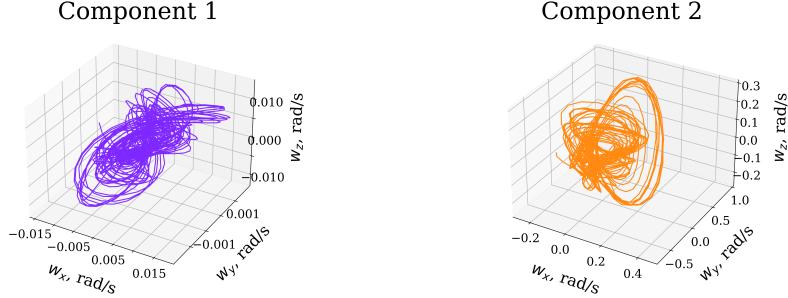


Figure 13 tSSA decomposition for gyroscope data. CPD rank = 10

In comparison, we present mSSA's decomposition with four components on the fig. 14 and three components on the fig. 15, 16. There is no need to solve heavy ILS problem or pick over CPD rank, but it is a manual procedure based on singular values analysis of some matrix [12]. In that case, it is simpler to gain more items and therefore obtain more detailed decomposition. Nonetheless, automation here is hindered. Moreover, in motion data case components are quite trivial and uninformative as method's approximation was too rude.

Lastly, tSSA slightly concedes in decomposition quality of the electricity and motion data, see tab. 3, 4. To conclude, the complexity of tSSA decomposition is the major stumbling block of our method which requires development of its own solutions and heuristics. For the record, mSSA has already had those [12].

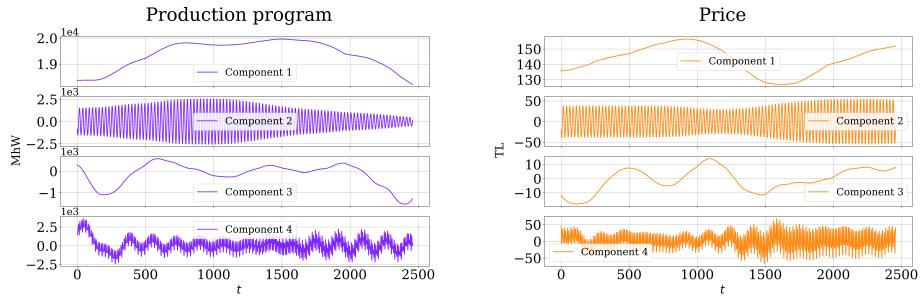


Figure 14 mSSA decomposition for electricity data

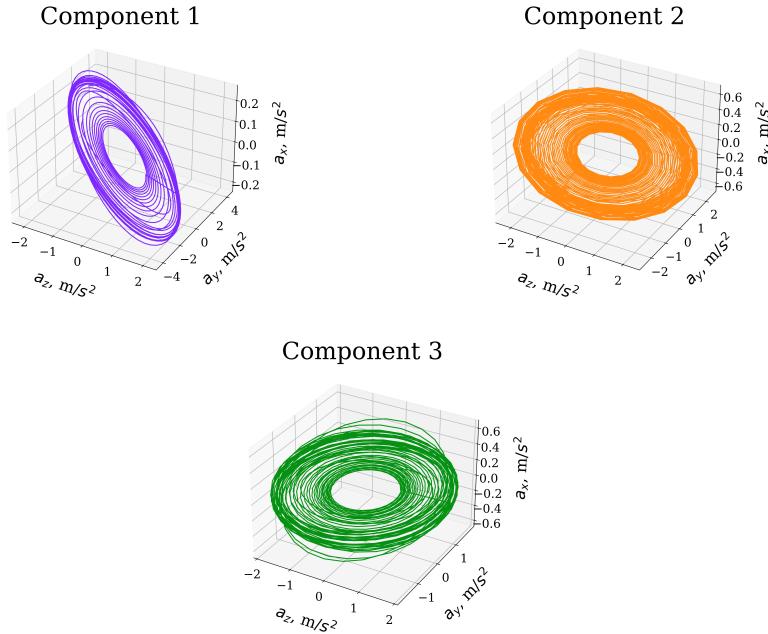


Figure 15 mSSA decomposition for accelerometer data

Table 3 Decomposition quality of models for electricity data

	tSSA	mSSA
$\bar{RHE}_{\text{Production}}$	0.507	0.308
\bar{RHE}_{Price}	0.511	0.31
\bar{RHE}	0.509	0.309

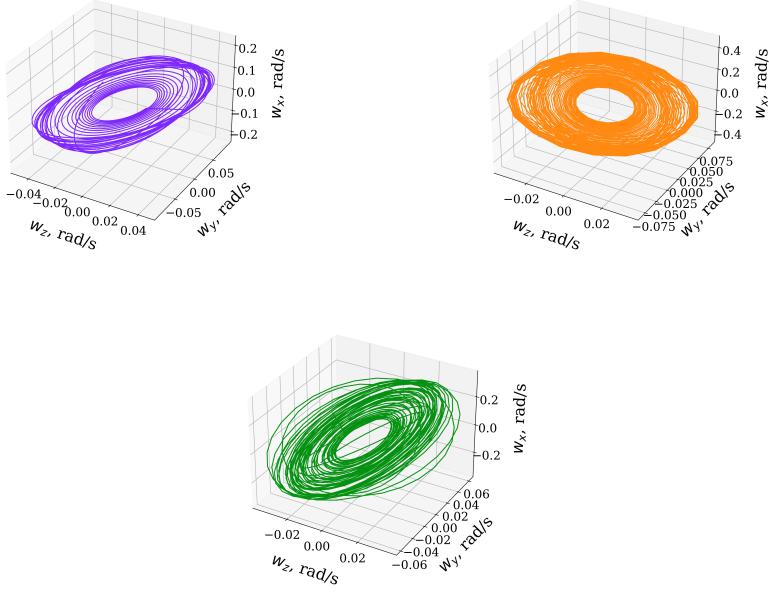


Figure 16 mSSA decomposition for gyroscope data

Table 4 Decomposition quality of models for motion data

	tSSA	mSSA
$\overline{\text{RHE}}_{\text{Accel}}$	0.438	0.394
$\overline{\text{RHE}}_{\text{Gyro}}$	0.732	0.468
$\overline{\text{RHE}}$	0.585	0.431

4 Conclusion

Proposed tensor method is devoted to prediction and decomposition of multidimensional time series with substantial interconnection between them. It is easy to use — there are only two adjustable parameters which does not require heavy learning procedures. Besides, the theory behind is based on the general framework of dynamical systems and demands only a possibility of low-dimensional data representation. When applied to real sets of signals, these assumptions were actually confirmed as well as the neural and statistical models were outperformed by our approach.

Nonetheless, a bottleneck was discovered which is an NP-hardness of optimal series decomposition. Resolving this problem or finding another way to build signal's components are the major ways for further method improvement. Another direction of the future research is an extending of the dynamical systems approach and finding another ways of building shared signals subspace.

References

- [1] Enders, W., Sons, J.W.: Applied Econometric Time Series. Wiley series in probability and statistics. Wiley, New York (2010).
<https://books.google.ru/books?id=vzJ0CgAAQBAJ>
- [2] Dagum, E., Bianconcini, S.: Seasonal Adjustment Methods and Real Time Trend-Cycle Estimation, p. 283. Springer, Switzerland (2016). <https://doi.org/10.1007/978-3-319-31822-6>
- [3] Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: Stl: A seasonal-trend decomposition procedure based on loess (with discussion). Journal of Official Statistics **6**, 3–73 (1990)
- [4] Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, Australia (2021). <https://books.google.ru/books?id=gZB-zgEACAAJ>
- [5] Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day series in time series analysis and digital processing. Holden-Day, Minnesota (1970). <https://books.google.ru/books?id=5BVfnXaq03oC>
- [6] Motrenko, A., Simchuk, E., Khairullin, R., Inyakin, A., Kashirin, D., Strijov, V.: Continuous physical activity recognition for intelligent labour monitoring. Multimedia Tools and Applications **81**(4), 4877–4895 (2022) <https://doi.org/10.1007/s11042-021-11288-y>

- [7] Volterra, V., Brelot, M.: Leçons sur la Théorie Mathématique de la Lutte pour la vie. Cahiers Scientifiques. Fasc Series. Gauthier-Villars et cie., Michigan (1931).
<https://books.google.ru/books?id=12rQAAAAMAAJ>
- [8] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) <https://doi.org/10.1162/neco.1997.9.8.1735>
- [9] Tealab, A.: Time series forecasting using artificial neural networks methodologies: A systematic review. Future Computing and Informatics Journal **3**(2), 334–340 (2018) <https://doi.org/10.1016/j.fcij.2018.10.003>
- [10] Asteriou, D., Hall, S.: Vector Autoregressive (VAR) Models and Causality Tests, pp. 333–346 (2016). https://doi.org/10.1057/978-1-137-41547-9_15
- [11] Zellner, A.: An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. Journal of the American Statistical Association **57**(298), 348–368 (1962) <https://doi.org/10.1080/01621459.1962.10480664>
- [12] Stepanov, D., Golyandina, N.: SSA-based approaches to analysis and forecast of multidimensional time series, pp. 293–298 (2005)
- [13] Krim, H., Viberg, M.: Two decades of array signal processing research: The parametric approach. Signal Processing Magazine, IEEE **13**, 67–94 (1996) <https://doi.org/10.1109/79.526899>
- [14] Ignatov, A.D., Strijov, V.V.: Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer. Multimedia tools and applications **75**, 7257–7270 (2016) <https://doi.org/10.1007/s11042-015-2643-0>
- [15] Hassani, H., Mahmoudvand, R.: Multivariate singular spectrum analysis: A general view and new vector forecasting approach. International Journal of Energy

- and Statistics **01** (2013) <https://doi.org/10.1142/S2335680413500051>
- [16] Takens, F.: Detecting Strange Attractors in Turbulence. In: Rand, D., Young, L.-S. (eds.) *Dynamical Systems and Turbulence*, Warwick 1980. Lecture Notes in Mathematics, vol. 898, pp. 366–381. Springer, Berlin (1981). Chap. 21. <https://doi.org/10.1007/bfb0091924>
- [17] Emde-Boas, P.: Another NP-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Report. Department of Mathematics. University of Amsterdam. Department, Univ., Amsterdam (1981). <https://books.google.ru/books?id=tCQiHQAAACAAJ>
- [18] Graffarend, E., Zwanzig, S., Awange, J.: Integer Least Squares, pp. 579–603. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-94598-5_16
- [19] Nahid, A.-A., Sikder, N., Rafi, I.: KU-HAR: An Open Dataset for Human Activity Recognition. Mendeley Data (2021). <https://doi.org/10.17632/45f952y38r.5>
- [20] Håstad, J.: Tensor rank is np-complete. Journal of Algorithms **11**(4), 644–654 (1990) [https://doi.org/10.1016/0196-6774\(90\)90014-6](https://doi.org/10.1016/0196-6774(90)90014-6)
- [21] Kolda, T., Bader, B.: Tensor decompositions and applications. SIAM Review **51**, 455–500 (2009) <https://doi.org/10.1137/07070111X>