Multilinear singular value decomposition of a tensor with fibers observed along one mode*

Stijn Hendrikx
Dept. Electrical Engineering (ESAT)
KU Leuven
Leuven, Belgium
ORCID: 0000-0001-9122-1754

Mikael Sørensen
Dept. Electrical Engineering (ESAT)
KU Leuven
Kortrijk/Leuven, Belgium
mr.mikael.sorensen@gmail.com

Lieven De Lathauwer
Dept. Electrical Engineering (ESAT)
KU Leuven
Kortrijk/Leuven, Belgium
lieven.delathauwer@kuleuven.be

***Abstract*—The multilinear singular value decomposition is a fundamental tool for higher-order data analysis. It is often used for orthogonal compression prior to the computation of other tensor decompositions, reducing the computational cost. We introduce an algorithm that uses only standard linear algebra operations for computing the multilinear singular value decomposition of an incomplete tensor with fibers observed along a single mode. This setting is very relevant for applications. For example, in an application where the tensor has a "time" mode, obtaining a fiber along this mode may be considerably easier than doing so along other modes. In the noise-free case, the algorithm is guaranteed to retrieve the exact solution, if the observed fibers satisfy certain deterministic conditions. As such, the approach reveals an interesting feature of the tensor setting that is not present at the matrix level. In the presence of noise, a solution obtained with this algorithm serves as a good initial point for further optimization. We illustrate, both on synthetic and real-life data, that this initialization strategy is fast and significantly reduces the number of iterations needed by an optimization algorithm. One possible use of the approach is as a linear algebra-based orthogonal compression of an incomplete tensor, after which the low multilinear rank approximation can be used as a "complete" proxy of the data for further analysis.**

***Index Terms*—tensor, decomposition, completion**

I. Introduction

The multilinear singular value decomposition (MLSVD) is a basic tool for working with tensors [1]–[3]. The MLSVD is useful, e.g., for a higher-order extension of principal component analysis, for subspace learning or the compression of higher-order data. It has been applied in fields such as signal processing, machine learning and scientific computing [4]–[8]. Additionally, it can be used as preprocessing in the computation of other tensor decompositions such as the canonical polyadic decomposition (CPD), which is a standard tool for blind signal separation. In this setting, the computation of the CPD is sped up by first orthogonally compressing the tensor using the MLSVD and then computing the CPD of the core [9].

*Research supported by: (1) Flemish Government: This research received funding from the Flemish Government (AI Research Program). Lieven De Lathauwer and Stijn Hendrikx are affiliated to Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium. This work was supported by the Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project no. G0F6718N (SeLMA). (2) KU Leuven Internal Funds: iBOF/23/064, C14/22/096, C16/15/059, IDN/19/014.

Datasets are often incomplete. Values might be intrinsically missing, or, in large-scale settings, only a subset of the data is used to reduce the computational cost. In tensor completion, the goal is to estimate the missing values by fitting an (approximately) low (multilinear) rank model to the available entries. There are two main approaches: (i) computing a tensor of fixed rank such that the approximation error on the available entries is minimal and (ii) minimizing the rank of a tensor such that the approximation error is smaller than some tolerance. Since minimizing the rank is a difficult non-convex problem, convex relaxations are often considered in practice [10]–[12]. In this work, we consider approaches of the former type, which are usually optimization-based [13]–[15].

We develop the key insight that the tensor setting offers intriguing new possibilities over the matrix setting, not only for compression and signal separation but also for completion. Obviously, matrix completion is not possible when entire columns are missing. On the other hand, tensor completion is possible when fibers are observed along a single mode. Moreover, fiber-wise (as opposed to entry-wise) observation makes that the completion can be done using only linear algebra. We have already shown this for CPD-based completion [16], [17]. In this work, we extend the study to MLSVD-based completion, which is numerically of greater interest as the unknowns are orthogonal. Our approach differs from pseudoskeleton algorithms, in which tensor decompositions are estimated using a subset of the fibers along *each* mode of the tensor [18]–[20]. The setting in which fibers are observed along a single mode is important in applications. By way of example, in a chemical setting where the tensor has "time" and "concentration" modes, obtaining additional samples in time may be much easier and cheaper than varying the concentrations (which perhaps requires a new experiment and intervention of a laboratory assistant).

An estimate obtained with our algorithm may serve as a good starting point for an optimization algorithm because it can be computed relatively quickly with standard linear algebra routines. We extend the common practice of using an algebraic initialization strategy for optimization-based algorithms to tensors with fibers observed along one mode.

In the remainder of this section, we introduce the necessary tensor-related notation. The standard MLSVD algorithm for

fully observed tensors is summarized in Section II. In Section III, this algorithm is generalized to tensors with fibers observed along a single mode. In Section IV, we illustrate the use of our algorithm by applying it to synthetic data and real fluorescence data.

Notation. A scalar, vector, matrix and tensor are denoted by $t, \mathbf{t}, \mathbf{T}$ and $\mathcal{T}$ respectively. The dimensions of a tensor are denoted as $I_1 \times I_2 \times I_3$. A selected part of a tensor is denoted as in Matlab using indices and colons, in which a colon indicates an index that has not been fixed. For example, the mode-1 slices and mode-3 vectors of a third-order tensor $\mathcal{T}$ are respectively denoted by $\mathbf{T}_{i::}$ and $\mathbf{t}_{ij:}$ for $i = 1, \ldots, I_1$ and $j = 1, \ldots, I_2$. A permutation and reshaping of a tensor $\mathcal{T}$ is denoted by, for example, $\mathbf{T}_{[1;2,3]} \in \mathbb{R}^{I_1 \times I_2 I_3}$, which holds tensor entry $t_{i_1 i_2 i_3}$ at position $(i_1, (i_2 - 1)I_2 + i_3)$. Vectorization is denoted by $\text{vec}\,(\mathcal{T})$ and corresponds to $\mathbf{t}_{[3,2,1]}$. The Kronecker, mode-n, outer and element-wise products are respectively denoted by $\otimes, \cdot_n, \otimes$ and $*$. Here, an entry of $\mathcal{X} = \mathcal{T} \cdot_n \mathbf{A}$ equals $x_{i_1 \ldots i_{n-1} r i_{n+1} \ldots i_N} = \sum_{j=1}^{I_n} t_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} a_{rj}$ with $\mathbf{A} \in \mathbb{C}^{R \times I_n}$. A row selection matrix that selects the rows that correspond to the ones in the binary vector $\mathbf{w} = \begin{bmatrix} w_1 \cdots w_I \end{bmatrix}^{\mathrm{T}} \in \{0,1\}^I$ is denoted by $\mathbf{S}^{(\mathbf{w})} \in \{0,1\}^{F \times I}$ with $F = \sum_{i=1}^{I} w_i$. A vector of length I containing only ones is denoted by $\mathbf{1}_I$.

II. MLSVD of a complete tensor

The Tucker decomposition factorizes a tensor $\mathcal{T}$ as

$$\mathcal{T} = \mathcal{G} \cdot_1 \mathbf{U}^{(1)} \cdot_2 \mathbf{U}^{(2)} \cdot_3 \mathbf{U}^{(3)}, \tag{1}$$

with core $\mathcal{G} \in \mathbb{C}^{R_1 \times R_2 \times R_3}$, and factor matrices $\mathbf{U}^{(1)} \in \mathbb{C}^{I_1 \times R_1}$, $\mathbf{U}^{(2)} \in \mathbb{C}^{I_2 \times R_2}$ and $\mathbf{U}^{(3)} \in \mathbb{C}^{I_3 \times R_3}$ [2]. The minimal tuple $R_{\text{ML}} = (R_1, R_2, R_3)$ is the multilinear rank of the tensor. If Eq. (1) only holds approximately, then the right-hand side is a low multilinear rank approximation of $\mathcal{T}$.

The MLSVD is a special case of the Tucker decomposition, in which the factor matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ are column-wise orthonormal and the core $\mathcal{G}$ is "all-orthogonal" and "ordered". "All-orthogonal" means that all slices along the same mode are mutually orthogonal and "ordered" means that the Frobenius norms of the slices along the same mode are monotonically non-increasing. Each factor matrix holds a basis for the vectors of $\mathcal{T}$ along its corresponding mode. Therefore, the factor matrices can be obtained from the first R_n left singular vectors for $n = 1, 2, 3$ of $\mathbf{T}_{[1;3,2]}$, $\mathbf{T}_{[2;3,1]}$ and $\mathbf{T}_{[3;2,1]}$, respectively. Since the factor matrices are column-wise orthonormal, the corresponding core can be computed as

$$\mathcal{G} = \mathcal{T} \cdot_1 \mathbf{U}^{(1)^{\mathrm{T}}} \cdot_2 \mathbf{U}^{(2)^{\mathrm{T}}} \cdot_3 \mathbf{U}^{(3)^{\mathrm{T}}}.$$

Computing the MLSVD as described above allows one to compute the factor matrices in parallel. In the sequential truncation variant of this algorithm, the tensor is compressed by a factor matrix as soon as it has been computed [21]. This rules out a parallel implementation, but it does lower the required computational effort as the size of the tensor decreases with each step.

III. MLSVD of a fiber-wise observed tensor

A. Algorithm

Assume that $\mathcal{T}$ is a third-order tensor of low multilinear rank R_{ML} of which some mode-3 fibers have been observed. The ones in $\mathbf{W} \in \{0,1\}^{I_1 \times I_2}$ indicate which mode-3 fibers of $\mathcal{T}$ are available, with $\sum_i \sum_j w_{ij} = F$.

The mode-3 unfolding of $\mathcal{T}$ consists of columns that are either completely available or missing since its columns correspond to the mode-3 fibers of $\mathcal{T}$. Factor matrix $\mathbf{U}^{(3)}$ can thus be obtained by computing the first R_3 left singular vectors of

$$\tilde{\mathbf{T}}_{[3;2,1]} \in \mathbb{R}^{I_3 \times F} = \mathbf{T}_{[3;2,1]} \mathbf{S}^{(\text{vec}(\mathbf{W}))^{\mathrm{T}}},$$

which contains only the completely available columns of $\mathbf{T}_{[3;2,1]}$. However, the mode-1 and 2 unfoldings of the tensor do not exhibit this pattern, meaning that $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ must be computed using another approach. The mode-2 unfolding

$$\mathbf{T}_{[2;1,3]} \in \mathbb{R}^{I_2 \times I_1 I_3} = \begin{bmatrix} \mathbf{T}_{1::} & \ldots & \mathbf{T}_{I_1::} \end{bmatrix}$$

consists of the mode-1 slices $\mathbf{T}_{i::} \in \mathbb{R}^{I_2 \times I_3}$ for $i = 1, \ldots, I_1$ of $\mathcal{T}$. The rows of these slices correspond to the mode-3 fibers of $\mathcal{T}$, meaning that they are either completely available or missing. Due to the low multilinear rank of $\mathcal{T}$, the submatrices containing the available rows of each slice can be expressed as

$$\tilde{\mathbf{T}}_{i::} = \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)} \mathbf{F}^{(i)} \quad \text{with } \mathbf{F}^{(i)} = \left(\sum_{r=1}^{R_1} \mathbf{G}_{r::} u_{ir}^{(1)} \right) \mathbf{U}^{(3)^{\mathrm{T}}}. \tag{2}$$

Under the condition that $\text{rank}\left(\mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)}\right) = \text{rank}\left(\mathbf{F}^{(i)}\right) = R_2$, there exist transformation matrices $\mathbf{H}^{(i)}$ such that

$$\tilde{\mathbf{T}}_{i::} \mathbf{H}^{(i)} = \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)} \quad \text{for } i = 1, \ldots, I_1,$$

which constitutes a set of linear equations in $\mathbf{U}^{(2)}$ and $\left\{\mathbf{H}^{(i)}\right\}$. If the equations are not exact, for example in the presence of noise, they can be solved in the least squares sense by solving

$$\min_{\mathbf{H}^{(1)}, \ldots, \mathbf{H}^{(I)}, \mathbf{U}^{(2)}} \sum_i \left\| \tilde{\mathbf{T}}_{i::} \mathbf{H}^{(i)} - \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)} \right\|_F^2.$$

The unknowns $\mathbf{H}^{(i)}$ can be eliminated through the substitution $\mathbf{H}^{(i)} = \tilde{\mathbf{T}}_{i::}^{\dagger} \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)}$:

$$\min_{\mathbf{U}^{(2)}} \sum_i \left\| \left(\tilde{\mathbf{T}}_{i::} \tilde{\mathbf{T}}_{i::}^{\dagger} - \mathbf{I} \right) \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)} \right\|_F^2.$$

The matrix $\tilde{\mathbf{T}}_{i::} \tilde{\mathbf{T}}_{i::}^{\dagger} - \mathbf{I}$ is an orthogonal projector onto the orthogonal complement of the column space of $\tilde{\mathbf{T}}_{i::}$. Let $\mathbf{V}^{(i)}$ be a basis of the orthogonal complement of the column space of $\tilde{\mathbf{T}}_{i::}$, which can be obtained from the SVD of $\tilde{\mathbf{T}}_{i::}$, then the optimization problem can be formulated as

$$\begin{aligned} &\min_{\mathbf{U}^{(2)}} \sum_i \left\| \mathbf{V}^{(i)^{\mathrm{T}}} \mathbf{S}^{(\mathbf{w}_{i:})} \mathbf{U}^{(2)} \right\|_F^2 \\ &= \min_{\mathbf{U}^{(2)}} \left\| \mathbf{U}^{(2)^{\mathrm{T}}} \left[\mathbf{S}^{(\mathbf{w}_{1:})^{\mathrm{T}}} \mathbf{V}^{(1)}, \ldots, \mathbf{S}^{(\mathbf{w}_{I_1:})^{\mathrm{T}}} \mathbf{V}^{(I_1)} \right] \right\|_F^2 \end{aligned}$$

This shows that the columns of the optimal $\mathbf{U}^{(2)}$ must lie in the left nullspace of $\left[\mathbf{S}^{(\mathbf{w}_{1:})}\mathbf{V}^{(1)}, \ldots, \mathbf{S}^{(\mathbf{w}_{I_1:})}\mathbf{V}^{(I_1)}\right]$. Therefore, $\mathbf{U}^{(2)}$ can be retrieved from the SVD of this matrix.

Finally, if needed the core can be obtained by solving the following linear system of equations

$$\tilde{\mathbf{T}}_{[2,1;3]}\mathbf{U}^{(3)} = \mathbf{S}^{(\text{vec}(\mathbf{W}))}\left(\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}\right)\mathbf{G}_{[2,1;3]}. \quad (3)$$

Pseudocode of the algorithm is given below. It can easily be generalized to tensors of order $N > 3$, in which case the tensor contains $\prod_{i \neq 2,N} I_i$ matrix slices of dimensions $I_2 \times I_N$ instead of only I_1 slices as in Eq.(2).

procedure `mlsvd_fs`$(\mathcal{T}, \mathbf{W}, R)$
  % Computation of third factor matrix
  Compute SVD $\mathbf{T}_{[3;2,1]}\mathbf{S}^{(\text{vec}(\mathbf{W}))^{\text{T}}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\text{T}}$
  $\mathbf{U}^{(3)} = \mathbf{U}(:, 1 : R_3)$
  Orthogonal compression $\mathcal{T} = \mathcal{T} \cdot_3 \mathbf{U}^{(3)^{\text{T}}}$
  % Computation of remaining factor matrices
  for $n = 1, 2$ **do**
    $\mathcal{X} = \mathcal{T}_{[m;n;3]}$, with $m = \{1, 2\} \setminus n$
    $\mathbf{Y} = \mathbf{W}_{[m;n]}$
    $\Omega = \{i : \sum_{j=1}^{I_n} y_{ij} \geq R_n\}$ for $i \in \{1, \ldots, I_n\}$
    for $i \in \Omega$ **do**
      Compute SVD $\mathbf{S}^{(\mathbf{y}_{i:})}\mathbf{X}_{i::} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\text{T}}$
      $\mathbf{V}^{(i)} = \mathbf{S}^{(\mathbf{y}_{i:})^{\text{T}}}\mathbf{U}(:, I_n - R_n + 1 : I_n)$
    Compute SVD $[\mathbf{V}^{(\Omega(1))}, \ldots, \mathbf{V}^{(\Omega(K))}] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\text{T}}$
    $\mathbf{U}^{(n)} = \mathbf{U}(:, I_n - R_n + 1 : I_n)$
  % Computation of core
  $\mathbf{A} = \mathbf{S}^{(\text{vec}(\mathbf{W}))}\left(\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}\right)$
  Solve $\tilde{\mathbf{T}}_{[2,1;3]}\mathbf{U}^{(3)} = \mathbf{A}\mathbf{G}_{[2,1;3]}$ for $\mathbf{G}_{[2,1;3]}$
  return $\mathcal{G}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}$

B. Conditions for uniqueness

The third factor matrix can be retrieved if and only if rank $\left(\tilde{\mathbf{T}}_{[3;2,1]}\right)$ equals R_3. The core can be retrieved if the linear system in Eq. (3) has a unique solution, which is satisfied if and only if rank $\left(\mathbf{S}^{(\text{vec}(\mathbf{W}))}\left(\mathbf{U}^{(2)} \otimes \mathbf{U}^{(1)}\right)\right) = R_1R_2$. In the generic case, that is, under the assumption that the entries of the factor matrices and the core can be thought of as drawn from a continuous probability distribution, these conditions reduce to $F > R_3$ and $F > R_1R_2$. The latter condition implies the former, since $R_3 \leq R_1R_2$ in any tensor of multilinear rank (R_1, R_2, R_3).

For the retrieval of the second factor matrix, it is required that rank $\left(\mathbf{S}^{(\mathbf{w}_{i:})}\mathbf{U}^{(2)}\right) = R_2$ and rank $\left(\mathbf{F}^{(i)}\right) = R_2$ hold for some mode-1 slices $\tilde{\mathbf{T}}_{i::}$. The latter condition can be reduced to $R_3 \geq R_2$, since it follows from Eq. (2) that rank $\left(\mathbf{F}^{(i)}\right) = \min(R_2, R_3)$. The former condition reduces to $\sum_j w_{ij} \geq R_2$ in the generic case, which states that at least R_2 rows need to be observed in a slice. The set Ω, with cardinality K, consists of the indices i for which the slices $\tilde{\mathbf{T}}_{i::}$ satisfy both conditions. Additionally, for each index $j \in \{1, \ldots, I_2\}$, the jth row should be available in at least one slice $\tilde{\mathbf{T}}_{i::}$ for $i \in \Omega$:

$$\forall j \in \{1, \ldots, I_2\}, \exists i \in \Omega : w_{ij} = 1.$$

Finally, the available rows in a slice $\tilde{\mathbf{T}}_{i::}$ for $i \in \Omega$ should overlap with at least R_2 available rows of the other slices $\tilde{\mathbf{T}}_{j::}$ for $j \in \{\Omega \setminus i\}$ and form a fully connected graph. This condition is a generalization of the uniqueness condition for the rank-1 matrix completion approach in [16], [22] and will be discussed in more detail in a journal version of this paper. Due to the symmetry of the problem, similar conditions hold for the retrieval of the first factor matrix.

IV. Experiments

In this section, `mlsvd_fs` is compared to relevant algorithms from literature on synthetic and real data. Algorithm `geomCG` performs Riemannian optimization on the manifold of tensors of fixed multilinear rank [14]. Algorithm `tucker_opt` updates the factor matrices and the core in an alternating fashion using the nonlinear conjugate gradient method [23]. Both are initialized by sampling the entries of the core and factor matrices from the standard normal distribution. All experiments are performed on a laptop with an AMD Ryzen 7 PRO 3700U processor and 32GB RAM.

A. Synthetic low multilinear rank tensors

In this experiment, tensors of dimensions $100 \times 100 \times 100$ with multilinear rank $R_{\text{ML}} = (10, 10, 10)$ are generated by sampling the entries of the core and factor matrices from the standard normal distribution. Next, additive noise $\mathcal{N}$ is added to the tensors $\mathcal{T}$. The signal-to-noise ratio (SNR) is given by $20 \log_{10} \frac{||\mathcal{T}||_{\text{F}}}{||\mathcal{N}||_{\text{F}}}$. The mode-3 fibers are randomly selected such that 20% have been observed. The approximation error of the estimated factor matrices $\hat{\mathbf{U}}^{(n)}$ can be quantified as the maximal canonical angle $\angle\left(\mathbf{U}^{(n)}, \hat{\mathbf{U}}^{(n)}\right)$ between the subspaces range $\left(\mathbf{U}^{(n)}\right)$ and range $\left(\hat{\mathbf{U}}^{(n)}\right)$. The approximation error on the full tensor is expressed as a relative error

$$E_{\text{rel}} = \frac{\left|\left|\mathcal{T} - \hat{\mathcal{G}} \cdot_1 \hat{\mathbf{U}}^{(1)} \cdot_2 \hat{\mathbf{U}}^{(2)} \cdot_3 \hat{\mathbf{U}}^{(3)}\right|\right|_{\text{F}}}{||\mathcal{T}||_{\text{F}}},$$

in which $\hat{\mathcal{G}}$ is the estimated core.

Fig. 1 shows the median approximation errors over 10 randomly generated tensors. Note that $\angle\left(\mathbf{U}^{(2)}, \hat{\mathbf{U}}^{(2)}\right)$ is omitted in this figure, as it is similar to $\angle\left(\mathbf{U}^{(1)}, \hat{\mathbf{U}}^{(1)}\right)$. Since mode-3 fibers are available, the results of the `mlsvd_fs` for this mode are equally accurate as those obtained by the optimization algorithms. However, the approximation errors in mode 1 and 2 for `mlsvd_fs` are considerably higher than for the optimization algorithms, since the algebraic problems for mode 1 and 2 are harder than for mode 3. This in turn causes a similar difference in the overall approximation error E_{rel} between `mlsvd_fs` and the optimization algorithms.

While `mlsvd_fs` yields a lower accuracy than the optimization-based algorithms, it requires significantly less computation time. On average, the computation times for `mlsvd_fs`, `geomCG` and `tucker_opt` respectively are $0.07, 14.07$ and 4.13 seconds: the former is two orders of magnitude faster for this problem. The speed of this algorithm

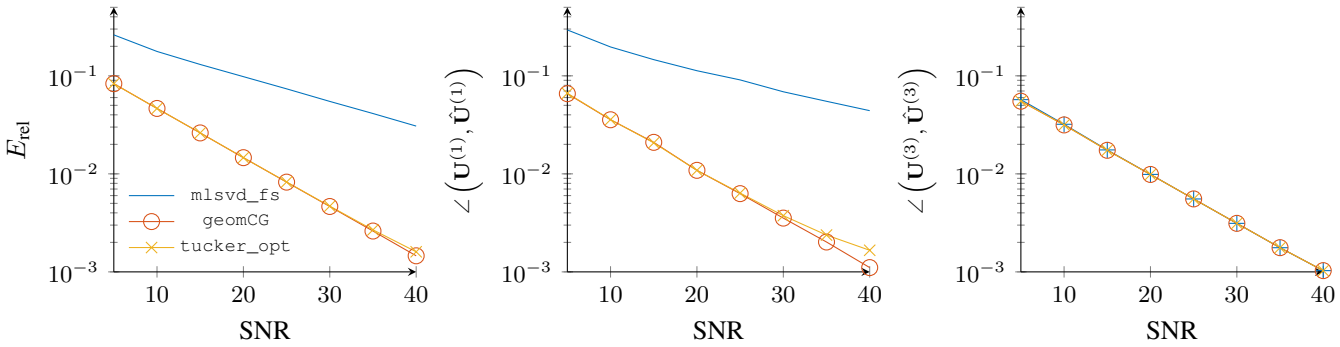


Fig. 1. Due to the less accurate estimation of the factor matrices for mode 1 and 2, the relative error of the results obtained by `mlsvd_fs` is also less accurate compared to the results of the optimization algorithms. However, `mlsvd_fs` takes far less computation time and is algebraic, making it a suitable initialization method for optimization-based algorithms.

combined with its algebraic nature makes it an ideal strategy for *initializing* an optimization algorithm. For example, by initializing `geomCG` in this manner, the algorithm converges to solutions that are as accurate as those in Fig. 1 in fewer iterations, which reduces the average computation time to 5.40 seconds. This reduction is even larger for more difficult problems, such as when the entries of the core and the factor matrices are sampled from the uniform distribution in the interval $[0, 1]$. In this case, `geomCG` with random initialization takes 51.55 seconds on average, while `geomCG` with `mlsvd_fs` initialization takes only 6.46 seconds.

B. Fluorescence data

In fluorescence spectroscopy, the spectra of the radiation emitted by a sample are measured for varying excitation wavelengths. The dataset consists of the measured spectra for five samples containing different concentrations of three amino acids [24]. The ranges of excitation and emission wavelengths respectively are 250 nm to 300 nm and 250 nm to 450 nm with 1 nm increments, resulting in a tensor $\mathcal{T}$ of dimensions $51 \times 201 \times 5$. Since each amino acid contributes a rank-1 tensor, the spectra of the individual amino acids can be recovered by computing a rank $R = 3$ CPD of $\mathcal{T}$ [25]. However, the recovered spectra can be distorted due to the Rayleigh scatter present in the data where the excitation and emission wavelengths are approximately equal. This can be mitigated by considering the entries of the tensor in this area to be missing, which results in a tensor with missing mode-3 fibers. In this case, only the partially observed tensor is available. Therefore, the approximation error is expressed as the relative error on the available entries

$$E_{\text{rel},\mathcal{V}} = \frac{\left|\left|\mathcal{V} * \left(\mathcal{T} - \left[\!\left[\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \hat{\mathbf{U}}^{(3)}\right]\!\right]\right)\right|\right|_{\text{F}}}{||\mathcal{V} * \mathcal{T}||_{\text{F}}}$$

in which $\mathcal{V} = \mathbf{W} \otimes \mathbf{1}_{I_3}$ indicates the available entries and $\left[\!\left[\hat{\mathbf{U}}^{(1)}, \hat{\mathbf{U}}^{(2)}, \hat{\mathbf{U}}^{(3)}\right]\!\right]$ denotes a CPD with factor matrices $\hat{\mathbf{U}}^{(n)}$ for $n = 1, 2, 3$.

The CPD of a tensor with missing mode-3 fibers can be computed in an algebraic manner as described in [16]. Algorithm `mlsvd_fs` offers an alternative algebraic approach, namely by first computing the MLSVD of this incomplete tensor. The CPD of the MLSVD core can then be computed using any standard CPD algorithm since all its entries are available. Finally, expanding the CPD factor matrices using the MLSVD factor matrices yields the CPD of the original tensor. This is similar to a standard approach for computing the CPD of a fully observed tensor, in which the tensor is first orthogonally compressed by means of the MLSVD [9].

The relative errors $E_{\text{rel},\mathcal{V}}$ obtained by the algebraic CPD and `mlsvd_fs` approaches respectively equal 0.0943 and 0.0886. Additionally, the former algorithm takes about 300 seconds, while the latter takes only about one second. More accurate solutions can be obtained by using an optimization algorithm. In this case, `cpd_nls` from Tensorlab [26], [27] was used, which employs a Gauss–Newton algorithm with dogleg trust region. This algorithm was initialized with factor matrices with entries sampled from the standard normal distribution and obtains a solution with $E_{\text{rel},\mathcal{V}} = 0.0171$. The solution obtained by compressing using `mlsvd_fs`, computing the CPD of the core and expanding the factor matrices provides a good starting point for an optimization algorithm. While `cpd_nls`, initialized with random factor matrices, requires 31 iterations (median over 100 random initializations) to reach convergence, only 16 iterations are needed to obtain an equally accurate solution when initialized using the `mlsvd_fs` approach.

V. Conclusion

We have introduced an algebraic approach for computing the MLSVD of a tensor with fibers observed along one mode. Numerical results were given for experiments with synthetic data and fluorescence data. These experiments highlight that its computational cost is low compared to other related algorithms and that it is a useful initialization strategy for an optimization algorithm. An interesting feature is that the technique can be used to obtain at low cost a "complete" proxy of incomplete fiber-wise observed tensor data. The MLSVD structure of this proxy can be exploited to speed up further analysis, such as the computation of a CPD under constraints [26], [28]. In a future journal version, we will discuss the algorithm and the corresponding uniqueness conditions in more detail.

References

[1] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, Apr. 2000.

[2] L. R. Tucker, "The extension of factor analysis to three-dimensional matrices," in *Contributions to mathematical psychology*, H. Gulliksen and N. Frederiksen, Eds., New York: Holt, Rinehart & Winston, 1964, pp. 110–127.

[3] A. Franc, "Etude algébrique des multitableaux: Apports de l'algebre tensorielle," *These de Doctorat, Spécialité Statistiques, Univ. de Montpellier II, Montpellier*, 1992.

[4] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[5] A. Cichocki, D. P. Mandic, L. De Lathauwer, *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[6] V. Khoromskaia and B. N. Khoromskij, *Tensor numerical methods in quantum chemistry*. Walter de Gruyter GmbH & Co KG, 2018.

[7] A. Franc, *Tensor ranks for the pedestrian for dimension reduction and disentangling interactions*, 2022. arXiv: 2201.07473.

[8] P. M. Kroonenberg, *Applied multiway data analysis*. John Wiley & Sons, 2008.

[9] C. A. Andersson and R. Bro, "Improving the speed of multi-way algorithms: Part I: Tucker3," *Chemometrics and Intelligent Laboratory Systems*, vol. 42, no. 1–2, pp. 93–103, 1998.

[10] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, 025010 (19pp), 2011.

[11] M. Signoretto, Q. Tran Dinh, L. De Lathauwer, and J. A. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Machine Learning*, vol. 94, pp. 303–351, 2014.

[12] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[13] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Morup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, Mar. 2011.

[14] D. Kressner, M. Steinlechner, and B. Vandereycken, "Low-rank tensor completion by Riemannian optimization," *BIT Numerical Mathematics*, vol. 54, no. 2, pp. 447–468, Nov. 2013.

[15] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 71–79, Sep. 2014.

[16] M. Sørensen and L. De Lathauwer, "Fiber sampling approach to canonical polyadic decomposition and application to tensor completion," *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 3, pp. 888–917, 2019.

[17] S. Hendrikx, M. Boussé, N. Vervliet, M. Vandecappelle, R. Kenis, and L. De Lathauwer, *Tensorlab*$^+$, Available online, Version of Dec 2022 downloaded from https://www.tensorlabplus.net.

[18] C. F. Caiafa and A. Cichocki, "Generalizing the column–row matrix decomposition to multi-way arrays," *Linear Algebra and its Applications*, vol. 433, no. 3, pp. 557–573, Sep. 2010.

[19] M. Mahoney, M. Maggioni, and P. Drineas, "Tensor-CUR decompositions for tensor-based data," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 957–987, 2008.

[20] I. Oseledets and E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," *Linear Algebra and its Applications*, vol. 432, no. 1, pp. 70–88, Jan. 2010.

[21] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM Journal on Scientific Computing*, vol. 34, no. 2, A1027–A1052, Jan. 2012.

[22] D. Hadwin, K. Harrison, and J. Ward, "Rank-one completions of partial matrices and completely rank-nonincreasing linear functionals," *Proceedings of the American Mathematical Society*, vol. 134, no. 8, pp. 2169–2178, 2006.

[23] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Systems and Signal Processing*, vol. 26, pp. 677–692, 3 Jul. 2015.

[24] R. Bro, "PARAFAC. tutorial and applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, pp. 149–171, 2 Oct. 1997.

[25] G. Tomasi and R. Bro, "PARAFAC and missing values," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 163–180, Feb. 2005.

[26] N. Vervliet and L. De Lathauwer, "Numerical optimization based algorithms for data fusion," in *Data Fusion Methodology and Applications*, M. Cocchi, Ed., vol. 31, Elsevier, 2019, ch. 4, pp. 81–128.

[27] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab 3.0*, Available online at https://www.tensorlab.net, Mar. 2016.

[28] N. Vervliet, O. Debals, and L. De Lathauwer, "Exploiting efficient representations in large-scale tensor decompositions," *SIAM Journal on Scientific Computing*, vol. 41, no. 2, A789–A815, Mar. 2019.