Low-rank nonnegative tensor approximation via alternating projections and sketching

Azamat Sultonov[1], Sergey Matveev[1,2], and Stanislav Budzinskiy[2]

[1]Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow
[2]Marchuk Institute of Numerical Mathematics RAS, Moscow

Abstract

We show how to construct nonnegative low-rank approximations of nonnegative tensors in Tucker and tensor train formats. We use alternating projections between the nonnegative orthant and the set of low-rank tensors, using STHOSVD and TTSVD algorithms, respectively, and further accelerate the alternating projections using randomized sketching. The numerical experiments on both synthetic data and hyperspectral images show the decay of the negative elements and that the error of the resulting approximation is close to the initial error obtained with STHOSVD and TTSVD. The proposed method for the Tucker case is superior to the previous ones in terms of computational complexity and decay of negative elements. The tensor train case, to the best of our knowledge, has not been studied before.

Contents

1 Introduction

Low-rank matrices and tensors have become ubiquitous in a multitude of tasks related to computational science and machine learning [1–3]. These highly structured representations lead to significant reduction in storage and, simultaneously, to elegant and efficient numerical algorithms, which prove useful in solving high-dimensional ODEs and PDEs [4–6], recovering signals from scarce measurements [7,8], global optimization [9], classification [10], etc.

In certain applications, the data are naturally nonnegative, and it is important to retain this property in the approximate representation. The data in question can be probability distributions [11–13], joint multidimensional concentrations of chemical compounds [14, 15], multispectral images [16], audio [17], and others. The issue with the standard approaches to low-rank approximation, both for matrices and tensors, is that they give no guarantee of nonnegativity.

An existing remedy is to employ the framework of nonnegative matrix factorization (NMF) [18] and nonnegative tensor factorization (NTF) [16]. This is a collection of techniques, all of which are based on the idea that every individual factor of the low-rank decomposition should be nonnegative, thereby leading to a nonnegative approximation. The *nonnegative* rank, however, can be larger than the usual rank of the matrix/tensor, so that the resulting representation becomes less compact (its further processing becomes less efficient too).

An alternative view on the problem is less strict as it allows the factors of the low-rank decomposition to have negative entries [19]. The low-rank nonnegative matrix approximation (LRNMA) problem for $X \in \mathbb{R}_+^{m \times n}$ can be formulated as minimization of the Frobenius norm:

$$\|X - Y\|_F \to \min \quad \text{s.t.} \quad Y \in \mathbb{R}_+^{m \times n}, \quad \text{rank}(Y) \le r. \tag{1}$$

Some optimality properties of this best-approximation formulation were studied in [20], viewed as a more general low-rank optimization problem with convex constraints [21, 22]. Stepping away from the search for the *best* approximation, a different approach based on alternating projections was proposed in [23] that provably locally converges to a *good* approximation. Further developments in this direction concerned the computational efficiency of the alternating projections [24, 25] and an augmented Lagrangian method [26].

The most important difference between the algorithms for the NMF and LRNMA formulations is that in NMF, the low-rank iterates are guaranteed to be nonnegative; in LRNMA, however, the intermediate low-rank matrices *do contain* negative elements, but they *converge* to a low-rank nonnegative matrix as the number of negative elements and their magnitude decrease with iterations. This allows one to perceive the alternating projections approach to LRNMA as a filtering procedure that can reduce the magnitude of the negative entries of a low-rank matrix below a certain threshold, acceptable for a given application.

In this paper, we study how alternating projections can be used for multidimensional low-rank nonnegative tensor approximation (LRNTA) and accelerated with randomized low-rank projections, as was done for matrices in [25]. We focus on two popular tensor formats: the Tucker [27] and tensor train (TT) [28] decompositions. Tucker-LRNTA was the subject of [29], where it was treated as consensus optimization between multiple LRNMA problems. To the best of our knowledge, TT-LRNTA is approached for the first time in our work.

2 Preliminaries

2.1 Tensors

We understand tensors as multidimensional arrays, i.e. elements of $\mathbb{R}^{n_1 \times \dots \times n_d}$; see [30] for a gentle introduction. Matrices will be denoted by uppercase letters $(X, Y, \dots)$ and tensors by

bold uppercase letters $(\boldsymbol{X}, \boldsymbol{Y}, \dots)$. The Frobenius norm of a tensor is the Euclidean norm induced by the standard inner product

$$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_F = \sum_{i_1, \dots, i_d} \boldsymbol{X}(i_1, \dots, i_d) \boldsymbol{Y}(i_1, \dots, i_d), \quad \|\boldsymbol{X}\|_F = \sqrt{\langle \boldsymbol{X}, \boldsymbol{X} \rangle_F},$$

and the Chebyshev norm is defined as the entry of maximum magnitude

$$\|\boldsymbol{X}\|_C = \max_{i_1, \dots, i_d} |\boldsymbol{X}(i_1, \dots, i_d)|.$$

The mode-k unfolding of a d-dimensional tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ for $1 \leq k \leq d$ is a matrix $X_{(k)} \in \mathbb{R}^{n_k \times \prod_{j \neq k} n_j}$ defined as

$$\boldsymbol{Y} = \text{PERMUTE}\Big(\boldsymbol{X}, \Big[k, 1, 2, \dots, k-1, k+1, \dots, d\Big]\Big),$$
$$X_{(k)} = \text{RESHAPE}\Big(\boldsymbol{Y}, \Big[n_k, \prod_{j \neq k} n_j\Big]\Big),$$

with the help of Matlab-style PERMUTE and RESHAPE operations. The k-th matricization of $\boldsymbol{X}$ for $1 \leq k \leq d-1$ is a matrix $X_{<k>} \in \mathbb{R}^{n_1 \dots n_k \times n_{k+1} \dots n_d}$ defined as

$$X_{<k>} = \text{RESHAPE}\Big(\boldsymbol{X}, \Big[\prod_{i=1}^{k} n_i, \prod_{j=k+1}^{d} n_j\Big]\Big).$$

The mode-k product of a tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and a matrix $U \in \mathbb{R}^{m_k \times n_k}$ is a tensor

$$\boldsymbol{Y} = \boldsymbol{X} \times_k U \in \mathbb{R}^{n_1 \times \dots \times n_{k-1} \times m_k \times n_{k+1} \times \dots \times n_d}, \quad Y_{(k)} = U X_{(k)}.$$

The Tucker, or multilinear, rank of a d-dimensional tensor $\boldsymbol{X}$ is an ordered d-tuple

$$\text{rank}_{tucker}(\boldsymbol{X}) = \big(\text{rank}(X_{(1)}), \dots, \text{rank}(X_{(d)})\big).$$

If $\text{rank}_{tucker}(\boldsymbol{X}) = (r_1, \dots, r_d)$ then it is possible to represent $\boldsymbol{X}$ as a series of mode-k products

$$\boldsymbol{X} = \boldsymbol{G} \times_1 U_1 \times_2 \dots \times_d U_d$$

between a Tucker core $\boldsymbol{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}$ and Tucker factors $U_k \in \mathbb{R}^{n_k \times r_k}$. This representation is known as the Tucker decomposition of $\boldsymbol{X}$; it is not unique and none of the r_k can be reduced without breaking the exact equality. For more details, see [30, 31].

The tensor train (TT) rank of a d-dimensional tensor $\boldsymbol{X}$ is an ordered $(d-1)$-tuple

$$\text{rank}_{tt}(\boldsymbol{X}) = \big(\text{rank}(X_{<1>}), \dots, \text{rank}(X_{<d-1>})\big).$$

Let $\text{rank}_{tt}(\boldsymbol{X}) = (r_1, \dots, r_{d-1})$; then there exist two matrices $G_1 \in \mathbb{R}^{n_1 \times r_1}$, $G_d \in \mathbb{R}^{r_{d-1} \times n_d}$ and $d-2$ three-dimensional tensors $\boldsymbol{G}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ for $2 \leq k \leq d-2$ such that every entry of $\boldsymbol{X}$ can be calculated as

$$\boldsymbol{X}(i_1, \dots, i_d) = \sum_{\alpha_1, \dots, \alpha_{d-1}} G_1(i_1, \alpha_1) \boldsymbol{G}_2(\alpha_1, i_2, \alpha_2) \dots \boldsymbol{G}_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, i_d),$$

where each α_k ranges from 1 to r_k. This is called the TT decomposition of $\boldsymbol{X}$ and G_1, $\{\boldsymbol{G}_k\}$, G_d are called TT cores. Like Tucker decomposition, it is not unique and exactness cannot be achieved with smaller values of r_k. Find more in [28].

2.2 Sketching

To compute the singular value decomposition (SVD) of an $m \times n$ matrix X ($m \geq n$) is a classic problem in numerical linear algebra. The state-of-the-art algorithms (see [32] for a thorough review) require $O(mn^2)$ flops, whether you need all n singular vectors or only $r \ll n$ dominant ones. This comes out as a serious bottleneck for computing low-rank approximations of large matrices; notably, matrices of size $n \times n^{d-1}$ with $d \geq 3$ appear in the HOSVD [31] and TTSVD [33] algorithms for tensor approximation in Tucker and TT formats, respectively.

Randomized sketching is a powerful modern technique [34], which allows one to find *good* (rather than *best*) rank-r approximations directly, using $O(mnk)$ flops with $k \geq r$, bypassing the full SVD. The main principle is to obtain a *sketch* of the original matrix (by multiplying it with an $n \times k$ random matrix Ψ) that is smaller in size, yet its column space approximates the dominant singular subspace.

A better subspace can be obtained with the *randomized subspace iteration* algorithm [35, Alg. 4.4], which computes the range of $(XX^T)^p X\Psi$ with $p \geq 0$. Practical implementation is presented in Alg. 1. The projection onto the corresponding orthonormal basis followed by truncated SVD (we denote it by SVD_r, where r is the truncation rank) gives the randomized SVD algorithm [35, Alg. 5.1] of the desired complexity (see Alg. 2).

Algorithm 1: Randomized subspace iteration [35, Alg. 4.4]

---

input: Data matrix $X \in \mathbb{R}^{m \times n}$, range sketch size $k \geq r$, number of iterations p, random matrix generator TestMatrix

1   $\Psi \leftarrow \text{TestMatrix}(n, k) \in \mathbb{R}^{n \times k}$
2   $Z_1 \leftarrow X\Psi \in \mathbb{R}^{m \times k}$
3   $[Q, R] \leftarrow \text{QR}(Z_1)$
4   **for** $j = 1, \ldots, p$ **do**
5      $Z_2 \leftarrow Q^T X \in \mathbb{R}^{k \times n}$
6      $[Q, R] \leftarrow \text{QR}(Z_2^T)$
7      $Z_1 \leftarrow XQ \in \mathbb{R}^{m \times k}$
8      $[Q, R] \leftarrow \text{QR}(Z_1)$
9   **return** Q

Algorithm 2: Randomized truncated SVD [35, Alg. 5.1] (HMT)

---

input: Data matrix $X \in \mathbb{R}^{m \times n}$, target rank r, estimated orthonormal basis $Q \in \mathbb{R}^{m \times k}$
1   $Z \leftarrow Q^T X \in \mathbb{R}^{k \times n}$
2   $[U_r, \Sigma_r, V_r] \leftarrow \text{SVD}_\text{r}(Z)$
3   $U_r \leftarrow QU_r$
4   **return** U_r, Σ_r, V_r^T

Another approach to randomized SVD [36] relies on two random matrices $\Psi \in \mathbb{R}^{n \times k}$ and $\Phi \in \mathbb{R}^{l \times m}$. It achieves the same asymptotic complexity as Alg. 2 but with a potentially smaller constant [25]. The first matrix Ψ is used to obtain the orthonormal basis Q. Then, instead of projecting onto Q directly, the second matrix Φ is employed through the Moore-Penrose

pseudoinverse $(\Phi Q)^\dagger \Phi X \in \mathbb{R}^{k \times n}$ (see Alg. 3).

Algorithm 3: Randomized truncated SVD [36] (Tropp)

---

input: Data matrix $X \in \mathbb{R}^{m \times n}$, target rank r, range sketch size $k \geq r$, co-range sketch size $l \geq k$, random matrix generator TestMatrix

1 $\Psi \leftarrow \text{TestMatrix}(n, k) \in \mathbb{R}^{n \times k}$

2 $\Phi \leftarrow \text{TestMatrix}(l, m) \in \mathbb{R}^{l \times m}$

3 $Z \leftarrow X\Psi \in \mathbb{R}^{m \times k}$

4 $[Q, R] \leftarrow \text{QR}(Z)$

5 $W \leftarrow \Phi Q \in \mathbb{R}^{l \times k}$

6 $[P, T] \leftarrow \text{QR}(W)$

7 $G \leftarrow T^{-1} P^T \Phi X \in \mathbb{R}^{k \times n}$

8 $[U_r, \Sigma_r, V_r] \leftarrow \text{SVD}_\text{r}(G)$

9 $U_r \leftarrow Q U_r$

10 return U_r, Σ_r, V_r^T

Before either of the algorithms is applied, the distribution of Ψ (and Φ) must be specified. In this paper, we will use matrices $\Psi \in \mathbb{R}^{n \times k}$ with iid Rademacher entries:

$$\psi_{ij} \sim \text{Rad}, \quad \psi_{ij} = \begin{cases} 1, & \text{with probability } 1/2 \\ -1, & \text{with probability } 1/2 \end{cases}$$

For brevity, we will write $\text{HMT}(p, k)$ for the combination of Algs. 1-2 and $\text{Tropp}(k, l)$ for Alg. 3.

3 Low-rank nonnegative tensor approximation

3.1 Matrix case

An alternating-projections-based approach to solve the LRNMA problem was proposed in [23]. Consider two sets: the nonnegative orthant $\mathbb{R}_+^{m \times n}$ and the set of low-rank matrices

$$\mathcal{M}_{\leq r} = \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) \leq r\}.$$

Both of them are closed, so for every matrix X there are best (with respect to the Frobenius norm) nonnegative $\Pi_{\mathbb{R}_+^{m \times n}}(X) \in \mathbb{R}_+^{m \times n}$ and low-rank $\Pi_{\mathcal{M}_{\leq r}}(X) \subset \mathcal{M}_{\leq r}$ approximations, respectively. The former is unique due to convexity and is given by

$$\Pi_{\mathbb{R}_+^{m \times n}}(X) = \max(X, 0).$$

The projection onto $\mathcal{M}_{\leq r}$, however, is not unique in general, but every matrix in $\Pi_{\mathcal{M}_{\leq r}}(X)$ is obtained as a truncated SVD of X [37]. Hence, with a slight abuse of notation, we will write

$$\Pi_{\mathcal{M}_{\leq r}}(X) = \text{SVD}_\text{r}(X).$$

Given a nonnegative matrix $X \in \mathbb{R}_+^{m \times n}$, the algorithm from [23] then iterates between the two sets as follows:

$$X^{(0)} \leftarrow X, \quad X^{(2k+1)} \leftarrow \Pi_{\mathcal{M}_{\leq r}}(X^{(2k)}), \quad X^{(2k)} \leftarrow \Pi_{\mathbb{R}_+^{m \times n}}(X^{(2k-1)}).$$

Clearly, one can also start with a low-rank matrix and change the order of the projections. Further papers considered more efficient alternating projections that use approximate rank truncation based on tangent spaces [24] and randomized sketching [25].

As we noted in the Introduction and as is seen directly from the algorithm, the intermediate matrices are not simultaneously low-rank and nonnegative: the even iterates are only nonnegative and the odd iterates are only low-rank. However, both of the subsequences converge to a low-rank nonnegative matrix. It was proved in [23] that if the initial matrix X is sufficiently close to the intersection $\mathcal{M}_{\leq r} \cap \mathbb{R}_+^{m \times n}$, the iterates converge to a quasioptimal solution of the LRNMA problem (1). We aim to draw from these ideas to present multidimensional extensions of the alternating projections approach to Tucker-LRNTA and TT-LRNTA.

3.2 Tucker case

As we discussed in Subsec. 2.1, every tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ admits an exact Tucker decomposition

$$\boldsymbol{X} = \boldsymbol{G} \times_1 U_1 \times_2 \dots \times_d U_d, \quad \boldsymbol{G} \in \mathbb{R}^{r_1 \times \dots \times r_d}, \quad U_k \in \mathbb{R}^{n_k \times r_k},$$

with $\text{rank}_{tucker}\left(\boldsymbol{X}\right) = (r_1, \dots, r_d)$. In practice, one seeks an *approximate* Tucker decomposition

$$\boldsymbol{X} \approx \boldsymbol{Y} = \boldsymbol{G} \times_1 U_1 \times_2 \dots \times_d U_d$$

of given rank $\boldsymbol{r} = (r_1, \dots, r_d)$ or such that the approximation is accurate up to a given threshold

$$\|\boldsymbol{X} - \boldsymbol{Y}\|_F < \varepsilon \|\boldsymbol{X}\|_F.$$

Both the exact and approximate Tucker decompositions can be constructed with the *higher-order* SVD (HOSVD) algorithm [31]: it computes the SVDs of the mode-k unfoldings $X_{(k)}$, chooses the Tucker factors U_k as the left singular factors, and computes the Tucker core by orthogonal projection

$$\boldsymbol{G} = \boldsymbol{X} \times_1 U_1^T \times_2 \dots \times_d U_d^T.$$

Unlike the truncated SVD for matrices, HOSVD does not lead to the *optimal* Tucker approximation; however, it is guaranteed to construct a quasioptimal one:

$$\|\boldsymbol{X} - \text{HOSVD}_{\boldsymbol{r}}\left(\boldsymbol{X}\right)\|_F \leq \sqrt{d} \min_{\boldsymbol{Y}} \|\boldsymbol{X} - \boldsymbol{Y}\|_F, \quad \text{rank}_{tucker}\left(\boldsymbol{Y}\right) \preceq \boldsymbol{r}.$$

The *sequentially truncated* HOSVD (STHOSVD) is a more computationally efficient procedure with similar approximation properties [38]. We provide its pseudocode in Alg. 4. Randomization can be employed to accelerate STHOSVD even further (cf. [39, 40]). By substituting HMT or Tropp in place of SVD$_r$, we get a family of algorithms with different rank-truncation strategies defined by svdr $\in \mathcal{F} = \{\text{SVD}_r, \text{HMT}, \text{Tropp}\}$.

Algorithm 4: Sequentially truncated higher-order SVD [38] (STHOSVD)

input: Data tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, target Tucker rank $\boldsymbol{r} = (r_1, \dots, r_d)$

1   $\boldsymbol{G} \leftarrow \boldsymbol{X}$
2   **for** $k = 1, 2, \dots, d$ **do**
3     $[U_{r_k}, \Sigma_{r_k}, V_{r_k}^T] \leftarrow \text{SVD}_r(G_{(k)}, r_k)$
4     $U_k \leftarrow U_{r_k} \in \mathbb{R}^{n_k \times r_k}$
5     $G_{(k)} \leftarrow \Sigma_{r_k} V_{r_k}^T \in \mathbb{R}^{r_k \times r_1 \dots r_{k-1} n_{k+1} \dots n_d}$
6   **return** $\boldsymbol{G}, U_1, U_2, \dots, U_d$

We will use STHOSVD and its randomized variants as approximate projections onto the closed set of tensors with low Tucker rank

$$\mathcal{M}_{\preceq \boldsymbol{r}}^{tucker} = \{\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d} : \text{rank}_{tucker}\left(\boldsymbol{X}\right) \preceq \boldsymbol{r}\}.$$

Since the projection onto the nonnegative orthant $\mathbb{R}_+^{n_1 \times \dots \times n_d}$ is the same as for matrices,

$$\Pi_{\mathbb{R}_+^{n_1 \times \dots \times n_d}}(\boldsymbol{X}) = \max(\boldsymbol{X}, 0),$$

we can formulate an alternating projection algorithm NSTHOSVD for the LRNTA problem in Tucker format; see Alg. 5.

Algorithm 5: STHOSVD-based alternating projections (NSTHOSVD)

input: Data tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, target Tucker rank $\boldsymbol{r} = (r_1, \dots, r_d)$, number of iterations s, rank-truncation strategy svdr $\in \mathcal{F}$

1   $\boldsymbol{X}^{(0)} \leftarrow \boldsymbol{X}$
2   **for** $i = 1, 2, \dots, s$ **do**
3      $\boldsymbol{X}^{(i)} \leftarrow \max(\boldsymbol{X}^{(i-1)}, 0)$
4      $[\boldsymbol{G}, U_1, \dots, U_d] \leftarrow \mathrm{STHOSVD}(\boldsymbol{X}^{(i)}, \boldsymbol{r}, \mathrm{svdr})$
5      $\boldsymbol{X}^{(i)} \leftarrow \boldsymbol{G} \times_1 U_1 \times_2 \dots \times_d U_d \in \mathbb{R}^{n_1 \times \dots \times n_d}$
6   **return** $\boldsymbol{G}, U_1, U_2, \dots, U_d$

Note that a different alternating projections algorithm NLRT was proposed for Tucker-LRNTA in [29] and its convergence was proved. While we suggest to use approximate projections onto the actual sets of interest $\mathbb{R}_+^{n_1 \times \dots \times n_d}$ and $\mathcal{M}_{\preceq \boldsymbol{r}}^{tucker}$, NLRT performs exact projections onto modified sets

$$\Omega_1 = \left\{ (\boldsymbol{X}_1, \dots, \boldsymbol{X}_d) : \boldsymbol{X}_1 = \dots = \boldsymbol{X}_d \in \mathbb{R}_+^{n_1 \times \dots \times n_d} \right\},$$
$$\Omega_2 = \left\{ (\boldsymbol{X}_1, \dots, \boldsymbol{X}_d) : \left(\boldsymbol{X}_k\right)_{(k)} \in \mathcal{M}_{\leq r_k}, \ 1 \leq k \leq d \right\},$$

which are subsets of the Cartesian product $\left(\mathbb{R}^{n_1 \times \dots \times n_d}\right) \times \dots \times \left(\mathbb{R}^{n_1 \times \dots \times n_d}\right)$. This is essentially a consensus optimization problem, where LRNMA is computed for every mode-k unfolding individually.

3.3   Tensor train case

In the same vein, exact and approximate TT decompositions of a tensor $\boldsymbol{X}$ can be computed with TTSVD [28], whose pseudocode we show in Alg. 6. The resulting approximation of given TT rank $\boldsymbol{r} = (r_1, \dots, r_{d-1})$ is quasioptimal

$$\|\boldsymbol{X} - \mathrm{TTSVD}_{\boldsymbol{r}}(\boldsymbol{X})\|_F \leq \sqrt{d-1} \min_{\boldsymbol{Y}} \|\boldsymbol{X} - \boldsymbol{Y}\|_F, \quad \mathrm{rank}_{tt}(\boldsymbol{Y}) \preceq \boldsymbol{r},$$

and is typically used as an approximate projection onto the set of tensors with low TT rank

$$\mathcal{M}_{\preceq \boldsymbol{r}}^{tt} = \{\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d} : \mathrm{rank}_{tt}(\boldsymbol{X}) \preceq \boldsymbol{r}\}.$$

Algorithm 6: TTSVD [28]

input: Data tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, target TT rank $\boldsymbol{r} = (r_1, \dots, r_{d-1})$

1   $G \leftarrow \mathrm{RESHAPE}(\boldsymbol{X}, [n_1, n_2 \dots n_d)]) \in \mathbb{R}^{n_1 \times n_2 \dots n_d}$
2   $[U_{r_1}, \Sigma_{r_1}, V_{r_1}^T] \leftarrow \mathrm{SVD}_r(G, \ r_1)$
3   $G_1 \leftarrow U_{r_1}$
4   **for** $k = 2, \dots, d-1$ **do**
5      $G \leftarrow \mathrm{RESHAPE}(\Sigma_{r_{k-1}} V_{r_{k-1}}^T, \ [r_{k-1} n_k, n_{k+1} \dots n_d]) \in \mathbb{R}^{r_{k-1} n_k \times n_{k+1} \dots n_d}$
6      $[U_{r_k}, \Sigma_{r_k}, V_{r_k}^T] \leftarrow \mathrm{SVD}_r(G, \ r_k)$
7      $\boldsymbol{G}_k \leftarrow \mathrm{RESHAPE}(U_{r_k}, [r_{k-1}, n_k, r_k]) \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$
8   $G_d \leftarrow \Sigma_{r_{d-1}} V_{r_{d-1}}^T$
9   **return** $G_1, \boldsymbol{G}_2, \dots, \boldsymbol{G}_{d-1}, G_d$

Swapping SVD$_r$ for a randomized SVD algorithm such as HMT or Tropp is a straightforward way to reduce the computational complexity (cf. [39, 41, 42]). We use these variants of TTSVD to solve the TT-LRNTA problem with alternating projections. Our algorithm NTTSVD is presented in Alg. 7.

Algorithm 7: TTSVD-based alternating projections (NTTSVD)

> **input:** Data tensor $\boldsymbol{X} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$, target TT rank $\boldsymbol{r} = (r_1, \ldots, r_{d-1})$, number of
>   iterations s, rank-truncation strategy svdr $\in \mathcal{F}$
> **1** $\boldsymbol{X}^{(0)} \leftarrow \boldsymbol{X}$
> **2** for $i = 1, 2, \ldots, s$ do
> **3** $\boldsymbol{X}^{(i)} \leftarrow \max(\boldsymbol{X}^{(i-1)}, 0)$
> **4** $[G_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_{d-1}, G_d] \leftarrow \text{TTSVD}(\boldsymbol{X}^{(i)}, \boldsymbol{r}, \text{svdr})$
> **5** $\boldsymbol{X}^{(i)} \leftarrow \sum\limits_{\alpha_1, \ldots, \alpha_{d-1}} G_1(j_1, \alpha_1) \boldsymbol{G}_2(\alpha_1, j_2, \alpha_2) \ldots \boldsymbol{G}_{d-1}(\alpha_{d-2}, j_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, j_d)$
> **6** return $G_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_{d-1}, G_d$

3.4 Computational complexities

In this section, we will estimate the one-iteration complexity of NSTHOSVD and NTTSVD. The most computationally expensive part of both algorithms is rank truncation, for which we have 3 options: SVD, HMT(p, k), and Tropp(k, l). For an $m \times n$ matrix with $m \geq n$ and truncation rank r, the standard truncated SVD costs $O(mn^2)$ floating point operations [32, 37]. As for the randomized approaches, HMT(p, k) requires $O(mn(pk + r))$ operations and Tropp(k, l) runs in $O(mn(r + k + l))$. While the asymptotic complexities are independent of the random matrix distribution, the constants can be reduced if structured random matrices are used (see [25]).

Turning to STHOSVD and TTSVD, let us denote by r the maximum truncation rank of the tensor (Tucker or TT), let $n = \max(n_1, \ldots, n_d)$, and assume that r is small compared to n. Every step of STHOSVD (Alg. 4) is dominated by rank truncation of unfolding matrices, whose sizes decrease progressively from $n \times n^{d-1}$ down to $n \times r^{d-1}$. The whole procedure then costs $O(n^{d+1})$ for SVD, $O(n^d(pk + r))$ for HMT(p, k), and $O(n^d(r + k + l))$ for Tropp(k, l). The analysis of TTSVD (Alg. 6) is absolutely analogous: the matricizations become smaller as $n \times n^{d-1}$, $rn \times n^{d-2}$, $\ldots$, $rn \times n$, so that the total cost is also $O(n^{d+1})$, $O(n^d(pk + r))$, and $O(n^d(r + k + l))$, respectively.

To compute a single element of a tensor given in the Tucker and TT formats, $O(dr^2)$ and $O(r^d)$ operations are needed, respectively. However, given the rich multilinear structure, it is possible to build the full tensor directly, using $O(n^d r)$ in both cases, which is faster than computing n^d elements individually. The nonnegative projections then require $O(n^d r)$ operations. This tells us that the overall complexity of a single iteration of NSTHOSVD and NTTSVD is defined by rank truncation, and we list it in Tab. 1. With randomization, we reach balance in complexity of low-rank and nonnegative projections: both of them scale linearly with n^d, the number of elements of the tensor.

	SVD	HMT(p, k)	Tropp(k, l)
NSTHOSVD	$O(n^{d+1})$	$O(n^d(pk + r))$	$O(n^d(r + k + l))$
NTTSVD	$O(n^{d+1})$	$O(n^d(pk + r))$	$O(n^d(r + k + l))$

Table 1: The asymptotic computational complexity of a single iteration of NSTHOSVD and NTTSVD algorithms with different rank-truncation methods.

3.5 Related work: existing theoretical analysis of alternating projections

Originally, the method of alternating projections was developed to compute the orthogonal projection of a given point onto the intersection of two closed subspaces in a Hilbert space [43]. Provided that the sum of these subspaces is closed too, the iterates converge linearly in norm for any starting point with a rate defined by the Friedrichs angle between the two subspaces [44,45].

One way to generalize closed subspaces is to consider closed convex sets (such as the nonnegative orthant $\mathbb{R}_+^{n_1 \times \dots \times n_d}$). For a pair of two closed convex sets, the method of alternating projections no longer finds the best approximation of the starting point, but solves the feasibility problem: converges to an arbitrary point in the intersection [46]. The iterates converge for every starting point, but do so only weakly. To prove strong convergence or even linear convergence, it is required that the pair of closed convex sets satisfies certain regularity assumptions [47].

The setting of closed convex sets is the most natural for the method of alternating projections. Indeed, owing to the Bunt-Motzkin theorem, every point of a finite-dimensional Euclidean space has a unique best approximation by a set if and only if this set is closed and convex [48]. Meanwhile, alternating projections have been extensively used for nonconvex sets as well, even though the projections are not uniquely defined for them (which complicates the global convergence analysis). Many sets that appear in practice are, however, *prox-regular*: the projections onto them are locally unique, i.e. the best approximation is uniquely defined for every point that is close enough to the set [49]. This property is shared, for example, by closed convex sets (obviously) and smooth manifolds [50]. When at least one of the two sets is prox-regular and their intersection satisfies certain regularity properties, the method of alternating projections locally linearly converges to a point in the intersection [51]. Notably, if the two sets are smooth manifolds, the regularity assumption translates to their intersection being transversal [50] and can be further relaxed to nontangential intersections [52]. Moreover, it was shown in [52] that alternating projections on smooth manifolds converge to quasioptimal approximations.

The sets of rank-r matrices, tensors of Tucker rank $\boldsymbol{r}$, and tensors of TT rank $\boldsymbol{r}$ are smooth manifolds [53] and the set of low-rank matrices $\mathcal{M}_{\leq r}$ is prox-regular at every rank-r matrix [54]. This suggests good behavior of the alternating projections for the LRNMA and LRNTA problems. The matrix case, where truncated SVD produces the optimal low-rank approximation, was theoretically addressed in [23]. In the tensor case, STHOSVD and TTSVD lead to quasioptimal projections, which will make the convergence analysis for NSTHOSVD and NTTSVD more delicate.

In the present work, we do not concentrate on the theoretical side of why NSTHOSVD and NTTSVD converge to low-rank nonnegative tensors. Instead, we rely on strong numerical evidence showing that they do in a number of different experiments. The successful outcome, in turn, motivates us to prove rigorous convergence guarantees in the future papers.

4 Numerical experiments

In this section, we evaluate and compare the performance of deterministic and randomized variants of the NSTHOSVD and the NTTSVD algorithms, which were introduced in Sec. 3. The examples we consider are

- the Hilbert tensor (Subsec. 4.1),

- a mixture of multidimensional Gaussians (Subsec. 4.2),

- a hyperspectral image (Subsec. 4.3).

For every Tucker-LRNTA experiment, we provide the respective results obtained with the NLRT algorithm [29] to compare its performance with NSTHOSVD. Note that comparing NTTSVD with NSTHOSVD and NLRT is not particularly meaningful since the difference in running times are mostly dictated by how well a given dataset is approximated in Tucker and TT tensor formats.

All the experiments were carried out in Python (3.9.12) with Intel(R) Core(TM) i3-8130U CPU@2.20GHz and 8GB of RAM.'

4.1 Hilbert tensor

Our first example of an approximately low-rank nonnegative tensor is the Hilbert tensor

$$\boldsymbol{X}(i_1, \ldots, i_d) = \frac{1}{i_1 + \ldots + i_d - d + 1},$$

which is a multidimensional extension of the well-known Hilbert matrix. In Table 2, we present the results obtained with NSTHOSVD, NTTSVD, and NLRT when applied to a 3-dimensional Hilbert tensor of size $128 \times 128 \times 128$ with Tucker ranks $\boldsymbol{r} = (3, 2, 4)$ and TT-ranks $\boldsymbol{r} = (3, 2)$.

First of all, note that the low-rank approximations obtained with simple STHOSVD and TTSVD contain negative elements whose total Frobenius norm is about 10^{-2}. Using 250 iterations of NSTHOSVD and NTTSVD (deterministic or randomized), we can reduce their Frobenius norm down to 5 double-precision machine epsilons. The total number of negative elements also decreases, and sometimes we manage to remove them completely. Remarkably, the relative approximation errors grow by only about 3% (Frobenius) and 7.5% (Chebyshev) so that the resulting low-rank tensors are still *good* approximations. The running time is an important aspect too: using randomized sketching, we achieved speed-up factors of about 7-11, compared to NSTHOSVD/NTTSVD with deterministic low-rank projections. All the variants show identical linear decay of the Frobenius norm of the negative elements (see Fig. 1).

Before comparing NSTHOSVD with NLRT, we would like to point out that NLRT never actually forms a tensor in the Tucker format. At every iteration, it computes low-rank approximations $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_d) \in \Omega_2$ of every unfolding (like HOSVD) but proceeds to the nonnegative projection without computing the Tucker core (unlike HOSVD). While NLRT iterations (all its d components) do converge to a single nonnegative tensor with low Tucker ranks, it is not formed explicitly. In Table 2, the NLRT-related results on relative errors and negative elements are computed based on the following low-rank STHOSVD approximation,

$$\hat{\boldsymbol{X}} = \text{STHOSVD}_{\boldsymbol{r}} \left(\frac{1}{d} \sum_{k=1}^{d} \Pi_{\mathbb{R}_+^{n_1 \times \ldots \times n_d}} (\boldsymbol{X}_k) \right), \tag{2}$$

and the running time is shown for the original NLRT (i.e. excluding the additional STHOSVD).

The results in Table 2 show that NSTHOSVD is superior to NLRT in terms of speed and mitigating negative elements, and they achieve similar relative errors. Deterministic NSTHOSVD is about 3 times faster than NLRT (just as STHOSVD is asymptotically d times faster than HOSVD) and its randomized versions are 23-39 times faster. The Frobenius norm of the negative elements of $\hat{\boldsymbol{X}}$ is of order 10^{-9}, which is an improvement over the initial 10^{-2}, but is much larger than what NSTHOSVD achieves. We also compare the properties of $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, and $\hat{\boldsymbol{X}}$ after 250 iterations of NLRT (see Table 3).

Method	Running time (Second)	Relative error (Frobenius)	Relative error (Chebyshev)	Negative elements (Frobenius)	Negative elements (%)
TTSVD	0.6	$7.72 \cdot 10^{-2}$	$3.67 \cdot 10^{-1}$	$9.7 \cdot 10^{-3}$	$6.3 \cdot 10^{-3}$
NTTSVD, SVD_r	121	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	$9.3 \cdot 10^{-16}$	$1.9 \cdot 10^{-4}$
NTTSVD, HMT(1, 12)	16	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	0.0	0.0
NTTSVD, HMT(0, 15)	13	$7.88 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	$1.2 \cdot 10^{-16}$	$9.5 \cdot 10^{-5}$
NTTSVD, Tropp(4, 30)	10	$8.13 \cdot 10^{-2}$	$3.82 \cdot 10^{-1}$	0.0	0.0
STHOSVD	0.5	$7.72 \cdot 10^{-2}$	$3.67 \cdot 10^{-1}$	$9.7 \cdot 10^{-3}$	$6.3 \cdot 10^{-3}$
NSTHOSVD, SVD_r	118	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	$5.9 \cdot 10^{-16}$	$1.4 \cdot 10^{-4}$
NSTHOSVD, HMT(1, 11)	17	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	0.0	0.0
NSTHOSVD, HMT(0, 15)	14	$7.89 \cdot 10^{-2}$	$3.95 \cdot 10^{-1}$	0.0	0.0
NSTHOSVD, Tropp(6, 35)	10	$7.89 \cdot 10^{-2}$	$3.94 \cdot 10^{-1}$	$2.5 \cdot 10^{-16}$	$4.8 \cdot 10^{-5}$
NLRT	390	$7.88 \cdot 10^{-2}$	$3.99 \cdot 10^{-1}$	$8.6 \cdot 10^{-10}$	$3.3 \cdot 10^{-4}$

Table 2: Comparison of NTTSVD, NSTHOSVD, and NLRT for low-rank nonnegative tensor approximation of a $128 \times 128 \times 128$ Hilbert tensor with Tucker ranks $(3, 2, 4)$ and TT-ranks $(3, 2)$: running times, relative errors, and negative elements after 250 iterations.

Tensor	Relative error (Frobenius)	Relative error (Chebyshev)	Negative elements (Frobenius)	Negative elements (Chebyshev)	Negative elements (%)
$\boldsymbol{X}_1$	$7.88 \cdot 10^{-2}$	$3.99 \cdot 10^{-1}$	$1.9 \cdot 10^{-10}$	$1.1 \cdot 10^{-10}$	$3.3 \cdot 10^{-4}$
$\boldsymbol{X}_2$	$7.88 \cdot 10^{-2}$	$3.99 \cdot 10^{-1}$	$8.5 \cdot 10^{-10}$	$5.1 \cdot 10^{-10}$	$3.3 \cdot 10^{-4}$
$\boldsymbol{X}_3$	$7.88 \cdot 10^{-2}$	$3.99 \cdot 10^{-1}$	$4.6 \cdot 10^{-11}$	$2.7 \cdot 10^{-11}$	$3.3 \cdot 10^{-4}$
$\hat{\boldsymbol{X}}$	$7.88 \cdot 10^{-2}$	$3.99 \cdot 10^{-1}$	$8.6 \cdot 10^{-10}$	$5.2 \cdot 10^{-10}$	$3.3 \cdot 10^{-4}$

Table 3: Comparison of the NLRT components $\{\boldsymbol{X}_k\}_{k=1}^3$ and the auxiliary tensor $\hat{\boldsymbol{X}}$ in Tucker format for low-rank nonnegative tensor approximation of a $128 \times 128 \times 128$ Hilbert tensor with Tucker ranks $(3, 2, 4)$: relative errors and negative elements after 250 iterations.

4.2 Multidimensional Gaussian mixture

In the second experiment, we test our approach on synthetic data that are an example of the low-rank density approximation problem: a multidimensional mixture of Gaussians

$$f(x) = \sum_{j=1}^{m} \alpha_j \exp\left((x - \mu_j)^\top A_j^{-1} (x - \mu_j) \right), \quad x \in \mathbb{R}^d,$$

with weights $\alpha_j \in \mathbb{R}$, means $\mu_j \in \mathbb{R}^d$, and covariance matrices $A_j \in \mathbb{R}^{d \times d}$. Every individual Gaussian has approximately low rank (it is a rank-1 tensor if A_j is diagonal) so the mixture can be approximated as well. We consider the mixture $f(x)$ in a hypercube $[-a, a]^d$ and discretize the domain on an equidistant tensor-product grid with step $2a/(n-1)$, which gives a d-dimensional $n \times \ldots \times n$ tensor $\boldsymbol{X}$.

We choose the 4-dimensional scenario with a balanced mixture of 2 Gaussians ($\alpha_1 = \alpha_2$) with the following means,

$$\mu_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top, \quad \mu_1 = \begin{bmatrix} 0.5 & -0.5 & 0.5 & -0.5 \end{bmatrix}^\top,$$

Figure 1: Comparison of deterministic and randomized variants of NSTHOSVD (a) and NTTSVD (b) for low-rank nonnegative tensor approximation of a $128 \times 128 \times 128$ Hilbert tensor with Tucker ranks $(3, 2, 4)$ and TT-ranks $(3, 2)$: the Frobenius norm of the negative part over 250 iterations.

and covariance matrices,

$$A_1 = \begin{bmatrix} 0.403 & 0.236 & 0.159 & 0.188 \\ 0.236 & 0.422 & 0.193 & 0.313 \\ 0.159 & 0.193 & 0.124 & 0.164 \\ 0.188 & 0.313 & 0.164 & 0.288 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.173 & 0.229 & 0.200 & 0.191 \\ 0.229 & 0.347 & 0.254 & 0.201 \\ 0.200 & 0.254 & 0.348 & 0.252 \\ 0.191 & 0.201 & 0.252 & 0.360 \end{bmatrix}.$$

We take $a = 1$ as the size of the domain and $n = 64$ as the size of the tensor.

We carried out 200 iterations of NTTSVD, NSTHOSVD, and NLRT with Tucker ranks $(14, 14, 14, 14)$ and TT-ranks $(10, 20, 10)$; see Table 4. Simple STHOSVD and TTSVD produce tensors with many, about 40%, negative elements. The application of NTTSVD and NSTHOSVD decreases their number to 1% and 2%, and their norms almost 400 and 1000 times, respectively (we show the convergence curves in Fig. 2). Compared to NSTHOSVD, NLRT leaves 1.5 times more negative elements with a 3 times higher norm. Sketching accelerates NTTSVD and NSTHOSVD 2-3 times, while deterministic NSTHOSVD itself is 3 times faster than NLRT. The increase of the relative error, due to sketching, is within 5% for most of the methods. We also compare the 4 NLRT components with the auxiliary low-rank tensor (2) in Table 5.

12

Method	Runnung time (Second)	Relative error (Frobenius)	Relative error (Chebyshev)	Negative elements (Frobenius)	Negative elements (%)
TTSVD	5.9	$7.4 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	5.3	41.0
NTTSVD, SVD_r	963	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	1.1
NTTSVD, HMT(1, 40)	550	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	1.1
NTTSVD, HMT(0, 45)	452	$8.7 \cdot 10^{-2}$	$1.8 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	0.89
NTTSVD, Tropp(38, 100)	371	$9.1 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	0.62
NTTSVD, Tropp(35, 100)	325	$9.4 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$	$1.4 \cdot 10^{-2}$	0.49
STHOSVD	3.4	$2.2 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$	1.8	38.0
NSTHOSVD, SVD_r	670	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	1.8
NSTHOSVD, HMT(1, 24)	447	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-3}$	1.8
NSTHOSVD, HMT(0, 24)	346	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.7 \cdot 10^{-3}$	1.3
NSTHOSVD, Tropp(22, 80)	230	$2.7 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$1.7 \cdot 10^{-3}$	0.99
NSTHOSVD, Tropp(18, 80)	205	$3.5 \cdot 10^{-2}$	$8.8 \cdot 10^{-2}$	$1.8 \cdot 10^{-3}$	0.75
NLRT	2096	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$5.1 \cdot 10^{-3}$	3.1

Table 4: Comparison of NTTSVD, NSTHOSVD, and NLRT for low-rank nonnegative tensor approximation of a $64 \times 64 \times 64 \times 64$ Gaussian mixture with Tucker ranks $(14, 14, 14, 14)$ and TT-ranks $(10, 20, 10)$: running times, relative errors, and negative elements after 200 iterations.

Tensor	Relative error (Frobenius)	Relative error (Chebyshev)	Negative elements (Frobenius)	Negative elements (Chebyshev)	Negative elements (%)
$\boldsymbol{X}_1$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$3.1 \cdot 10^{-3}$	$2.4 \cdot 10^{-4}$	2.0
$\boldsymbol{X}_2$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$3.2 \cdot 10^{-3}$	$2.2 \cdot 10^{-4}$	1.9
$\boldsymbol{X}_3$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$2.5 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$	2.7
$\boldsymbol{X}_4$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$2.8 \cdot 10^{-3}$	$2.3 \cdot 10^{-4}$	2.2
$\hat{\boldsymbol{X}}$	$2.6 \cdot 10^{-2}$	$1.0 \cdot 10^{-1}$	$5.1 \cdot 10^{-3}$	$3.5 \cdot 10^{-4}$	3.1

Table 5: Comparison of the NLRT components $\{\boldsymbol{X}_k\}_{k=1}^4$ and the auxiliary tensor $\hat{\boldsymbol{X}}$ in Tucker format for low-rank nonnegative tensor approximation of a $64 \times 64 \times 64 \times 64$ Gaussian mixture with Tucker ranks $(14, 14, 14, 14)$: relative errors and negative elements after 200 iterations.

Figure 2: Comparison of deterministic and randomized variants of NSTHOSVD (a) and NTTSVD (b) for low-rank nonnegative tensor approximation of a $64 \times 64 \times 64 \times 64$ Gaussian mixture with Tucker ranks $(14, 14, 14, 14)$ and TT-ranks $(10, 20, 10)$: the Frobenius and Chebyshev norms of the negative part and the density of negative elements over 200 iterations.

14

4.3 Hyperspectral image

Our final example is an openly available hyperspectral image of the Washington DC National Mall[1] of size $307 \times 307 \times 191$, the last dimension being the spectral bands (we also linearly scale the elements to $[0,1]$). With Tucker ranks $(40, 40, 33)$ and TT-ranks $(33, 33)$, the image can be compressed 215 and 51 times, respectively. To measure the quality of low-rank approximations, we use 3 values: the relative error in the Frobenius norm, the band-wise mean of the structural similarity index measure (SSIM, [55]), and the statistical R^2 coefficient of determination,

$$R^2 = 1 - \frac{\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2}{\|\boldsymbol{X} - \alpha\|_F^2}, \quad \alpha = \frac{1}{\prod_{k=1}^d n_k} \sum_{(i_1,\ldots,i_d)} \boldsymbol{X}(i_1,\ldots,i_d) \in \mathbb{R},$$

where $\boldsymbol{Y}$ is a low-rank approximant.

In Table 6, we see that 100 iterations of deterministic NTTSVD and NSTHOSVD lower the Frobenius norm of the negative elements 350 and 100 times, respectively, compared to TTSVD and STHOSVD. Both the relative error and the R^2 score stay the same, and SSIM undergoes a reduction by $0.03-0.04$. The randomized variants based on HMT with 1 power-method iteration lead to similar results, but achieve them about 1.5 times faster. With the more computationally efficient randomized approaches, SSIM seems to degrade more severely than the 2 other quality measures. Find the convergence curves for NTTSVD and NSTHOSVD in Fig. 3. The approximation quality achieved with NLRT is identical to deterministic NSTHOSVD; however, the norm of the negative elements is 2.4 times higher, and it runs 4 times slower (6 times compared to randomized NSTHOSVD). As Table 7 shows, the negative elements in the low-rank auxiliary tensor (2) have a larger norm than the unfoldings, which NLRT operates on. Finally, in Fig. 4 we present the actual images for visual evaluation.

Method	Running time (Second)	Relative error (Frobenius)	SSIM	R^2	Negative elements (Frobenius)
TTSVD	3.2	$1.8 \cdot 10^{-1}$	0.66	0.94	2.2
NTTSVD, SVD_r	341	$1.8 \cdot 10^{-1}$	0.63	0.94	$6.0 \cdot 10^{-3}$
NTTSVD, HMT(1, 75)	236	$1.8 \cdot 10^{-1}$	0.62	0.94	$6.0 \cdot 10^{-3}$
NTTSVD, HMT(0, 75)	173	$2.1 \cdot 10^{-1}$	0.57	0.92	$1.0 \cdot 10^{-2}$
NTTSVD, Tropp(60, 150)	151	$2.6 \cdot 10^{-1}$	0.46	0.88	$1.8 \cdot 10^{-2}$
NTTSVD, Tropp(50, 150)	102	$2.7 \cdot 10^{-1}$	0.45	0.87	$1.9 \cdot 10^{-2}$
STHOSVD	3.4	$1.8 \cdot 10^{-1}$	0.64	0.94	2.1
NSTHOSVD, SVD_r	440	$1.8 \cdot 10^{-1}$	0.60	0.94	$1.9 \cdot 10^{-2}$
NSTHOSVD, HMT(1, 75)	296	$1.9 \cdot 10^{-1}$	0.60	0.94	$1.9 \cdot 10^{-2}$
NSTHOSVD, HMT(0, 75)	268	$2.1 \cdot 10^{-1}$	0.53	0.92	$2.6 \cdot 10^{-2}$
NSTHOSVD, Tropp(60, 150)	225	$2.8 \cdot 10^{-1}$	0.40	0.86	$5.4 \cdot 10^{-2}$
NSTHOSVD, Tropp(50, 150)	131	$2.9 \cdot 10^{-1}$	0.39	0.85	$5.1 \cdot 10^{-2}$
NLRT	1874	$1.8 \cdot 10^{-1}$	0.60	0.94	$4.6 \cdot 10^{-2}$

Table 6: Comparison of NTTSVD, NSTHOSVD, and NLRT for low-rank nonnegative tensor approximation of a $307 \times 307 \times 191$ hyperspectral image of the Washington DC National Mall with Tucker ranks $(40, 40, 33)$ and TT-ranks $(33, 33)$: running times, relative errors, negative elements, SSIM, and R^2 score after 100 iterations.

---

[1]Data available at https://github.com/JakobSig/HSI2RGB/blob/master/washington_hsi.mat

(a)

(b)

Figure 3: Comparison of deterministic and randomized variants of NSTHOSVD (a) and NTTSVD (b) for low-rank nonnegative tensor approximation of a $307 \times 307 \times 191$ hyperspectral image of the Washington DC National Mall with Tucker ranks $(40, 40, 33)$ and TT-ranks $(33, 33)$: the Frobenius and Chebyshev norms of the negative part and the density of negative elements over 100 iterations.

Tensor	Relative error (Frobenius)	SSIM	R^2	Negative elements (Frobenius)
$\boldsymbol{X}_1$	$1.8 \cdot 10^{-1}$	0.60	0.94	$3.6 \cdot 10^{-2}$
$\boldsymbol{X}_2$	$1.8 \cdot 10^{-1}$	0.60	0.94	$3.6 \cdot 10^{-2}$
$\boldsymbol{X}_3$	$1.8 \cdot 10^{-1}$	0.60	0.94	$1.5 \cdot 10^{-2}$
$\hat{\boldsymbol{X}}$	$1.8 \cdot 10^{-1}$	0.60	0.94	$4.6 \cdot 10^{-2}$

Table 7: Comparison of the NLRT components $\{\boldsymbol{X}_k\}_{k=1}^{3}$ and the auxiliary tensor $\hat{\boldsymbol{X}}$ in Tucker format for low-rank nonnegative tensor approximation of a $307 \times 307 \times 191$ hyperspectral image of the Washington DC National Mall with Tucker ranks $(40, 40, 33)$: negative elements, relative errors, SSIM, and R^2 score after 100 iterations.

Figure 4: Comparison of the approximations of a $307 \times 307 \times 191$ hyperspectral image of the Washington DC National Mall achieved with TTSVD, STHOSVD, and different iterative LRNTA approaches (after 100 iterations). We present the 50th spectral band.

Figure 5: Comparison of randomized NSTHOSVD and NLRT for low-rank nonnegative tensor approximation: the decay of the Frobenius norm of the negative elements for the Hilbert tensor (a), the Gaussian mixture (b), and the hyperspectral image (c).

5 Conclusion

In this work, we looked at a natural multidimensional extension of randomized alternating projections for the LRNMA problem [25] and proposed two algorithms, NSTHOSVD and NTTSVD, for the Tucker and tensor train formats, respectively. The numerical experiments showed that our approach allows to reduce the number (and the absolute value) of the negative elements in the low-rank approximation without significant loss of accuracy. Comparing with the NLRT method [29], which was developed for the Tucker case, we observed that our algorithm NSTHOSVD is superior in terms of computational efficiency per iteration and in how fast it reduces the negative elements (see Fig. 5).

The use of randomization allowed us to obtain algorithms, whose complexity scales linearly with the number of elements of the tensor, thereby achieving balance in complexity of low-rank and nonnegative projections. Moreover, by choosing the configuration parameters of randomized sketching (such as the oversampling, the distribution of the random matrix) one can tune the methods to achieve the desired trade-off between speed and accuracy.

Though the proposed algorithms work in numerical experiments, they still require a proof of convergence. We will study their theoretical properties in future papers.

Acknowledgements

Data Availability

The datasets generated during and/or analysed during the current study are available in the github repository https://github.com/azamat11235/NLRTA.

References

[1] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. Oseledets, M. Sugiyama, D. P. Mandic, *et al.*, "Tensor networks for dimensionality reduction and large-scale optimization: Part

2 applications and future perspectives," *Foundations and Trends® in Machine Learning*, vol. 9, no. 6, pp. 431–673, 2017.

[2] N. D. 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, 2017.

[3] B. N. Khoromskij, "Tensor numerical methods in scientific computing," in *Tensor Numerical Methods in Scientific Computing*, De Gruyter, 2018.

[4] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, "Direct solution of the chemical master equation using quantized tensor trains," *PLoS computational biology*, vol. 10, no. 3, p. e1003359, 2014.

[5] S. A. Matveev, D. A. Zheltkov, E. E. Tyrtyshnikov, and A. P. Smirnov, "Tensor train versus Monte Carlo for the multicomponent Smoluchowski coagulation equation," *Journal of Computational Physics*, vol. 316, pp. 164–179, 2016.

[6] F. Allmann-Rahn, R. Grauer, and K. Kormann, "A parallel low-rank solver for the six-dimensional vlasov-maxwell equations," *arXiv preprint arXiv:2201.03471*, 2022.

[7] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2012.

[8] S. Budzinskiy and N. Zamarashkin, "Tensor train completion: local recovery guarantees via Riemannian optimization," *arXiv preprint arXiv:2110.03975*, 2021.

[9] D. Zheltkov and E. Tyrtyshnikov, "Global optimization based on TT-decomposition," *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 35, no. 4, pp. 247–261, 2020.

[10] Y. Yang, D. Krompass, and V. Tresp, "Tensor-train recurrent neural networks for video classification," in *International Conference on Machine Learning*, pp. 3891–3900, PMLR, 2017.

[11] S. Dolgov, K. Anaya-Izquierdo, C. Fox, and R. Scheichl, "Approximation and sampling of multivariate probability distributions in the tensor train decomposition," *Statistics and Computing*, vol. 30, no. 3, pp. 603–625, 2020.

[12] G. S. Novikov, M. E. Panov, and I. V. Oseledets, "Tensor-train density estimation," in *Uncertainty in Artificial Intelligence*, pp. 1321–1331, PMLR, 2021.

[13] Y. Hur, J. G. Hoskins, M. Lindsey, E. M. Stoudenmire, and Y. Khoo, "Generative modeling via tensor train sketching," *arXiv preprint arXiv:2202.11788*, 2022.

[14] E. Shcherbakova and E. Tyrtyshnikov, "Nonnegative tensor train factorizations and some applications," in *International Conference on Large-Scale Scientific Computing*, pp. 156–164, Springer, 2019.

[15] G. Manzini, E. Skau, D. P. Truong, and R. Vangara, "Nonnegative tensor-train low-rank approximations of the Smoluchowski coagulation equation," in *International Conference on Large-Scale Scientific Computing*, pp. 342–350, Springer, 2021.

[16] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[17] V. Leplat, N. Gillis, and A. M. Ang, "Blind audio source separation with minimum-volume beta-divergence NMF," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3400–3410, 2020.

[18] N. Gillis, *Nonnegative matrix factorization*. SIAM, 2020.

[19] B. Vanluyten, J. C. Willems, and B. De Moor, "Nonnegative matrix factorization without nonnegativity constraints on the factors," *Submitted for publication*, 2008.

[20] C. Grussler and A. Rantzer, "On optimal low-rank approximation of non-negative matrices," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 5278–5283, IEEE, 2015.

[21] C. Grussler, A. Rantzer, and P. Giselsson, "Low-rank optimization with convex constraints," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 4000–4007, 2018.

[22] F. Andersson, M. Carlsson, and C. Olsson, "Convex envelopes for fixed rank approximation," *Optimization Letters*, vol. 11, no. 8, pp. 1783–1795, 2017.

[23] G.-J. Song and M. K. Ng, "Nonnegative low rank matrix approximation for nonnegative matrices," *Applied Mathematics Letters*, vol. 105, p. 106300, 2020.

[24] G. Song, M. K. Ng, and T.-X. Jiang, "Tangent space based alternating projections for nonnegative low rank matrix approximation," *arXiv preprint arXiv:2009.03998*, 2020.

[25] S. A. Matveev and S. Budzinskiy, "Sketching for low-rank nonnegative matrix approximation: a numerical study," *arXiv preprint arXiv:2201.11154*, 2022.

[26] H. Zhu, M. K. Ng, and G.-J. Song, "An approximate augmented Lagrangian method for nonnegative low-rank matrix approximation," *Journal of Scientific Computing*, vol. 88, no. 2, pp. 1–22, 2021.

[27] L. R. Tucker *et al.*, "The extension of factor analysis to three-dimensional matrices," *Contributions to mathematical psychology*, vol. 110119, 1964.

[28] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[29] T.-X. Jiang, M. K. Ng, J. Pan, and G. Song, "Nonnegative low rank tensor approximation and its application to multi-dimensional images," *arXiv preprint arXiv:2007.14137*, 2020.

[30] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[31] 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, 2000.

[32] J. Dongarra, M. Gates, A. Haidar, J. Kurzak, P. Luszczek, S. Tomov, and I. Yamazaki, "The singular value decomposition: Anatomy of optimizing an algorithm for extreme scale," *SIAM review*, vol. 60, no. 4, pp. 808–865, 2018.

[33] I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse of dimensionality, or how to use SVD in many dimensions," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3744–3759, 2009.

[34] P.-G. Martinsson and J. A. Tropp, "Randomized numerical linear algebra: Foundations and algorithms," *Acta Numerica*, vol. 29, pp. 403–572, 2020.

[35] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[36] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, "Practical sketching algorithms for low-rank matrix approximation," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 4, pp. 1454–1485, 2017.

[37] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Baltimore: The Johns Hopkins University Press, fourth edition ed., 2013.

[38] 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, pp. A1027–A1052, 2012.

[39] M. Che and Y. Wei, "Randomized algorithms for the approximations of Tucker and the tensor train decompositions," *Advances in Computational Mathematics*, vol. 45, no. 1, pp. 395–428, 2019.

[40] S. Ahmadi-Asl, S. Abukhovich, M. G. Asante-Mensah, A. Cichocki, A. H. Phan, T. Tanaka, and I. Oseledets, "Randomized algorithms for computation of Tucker decomposition and higher order SVD (HOSVD)," *IEEE Access*, vol. 9, pp. 28684–28706, 2021.

[41] B. Huber, R. Schneider, and S. Wolf, "A randomized tensor train singular value decomposition," in *Compressed sensing and its applications*, pp. 261–290, Springer, 2017.

[42] D. Kressner, B. Vandereycken, and R. Voorhaar, "Streaming tensor train approximation," *arXiv preprint arXiv:2208.02600*, 2022.

[43] R. Escalante and M. Raydan, *Alternating projection methods*. SIAM, 2011.

[44] F. Deutsch, "Rate of convergence of the method of alternating projections," in *Parametric optimization and approximation*, pp. 96–107, Springer, 1984.

[45] S. Kayalar and H. L. Weinert, "Error bounds for the method of alternating projections," *Mathematics of Control, Signals and Systems*, vol. 1, no. 1, pp. 43–59, 1988.

[46] H. H. Bauschke and J. M. Borwein, "Dykstra's alternating projection algorithm for two sets," *Journal of Approximation Theory*, vol. 79, no. 3, pp. 418–443, 1994.

[47] H. H. Bauschke and J. M. Borwein, "On the convergence of von neumann's alternating projection algorithm for two sets," *Set-Valued Analysis*, vol. 1, no. 2, pp. 185–212, 1993.

[48] F. Deutsch, *Best approximation in inner product spaces*, vol. 7. Springer, 2001.

[49] R. Poliquin, R. Rockafellar, and L. Thibault, "Local differentiability of distance functions," *Transactions of the American mathematical Society*, vol. 352, no. 11, pp. 5231–5249, 2000.

[50] A. S. Lewis and J. Malick, "Alternating projections on manifolds," *Mathematics of Operations Research*, vol. 33, no. 1, pp. 216–234, 2008.

[51] A. S. Lewis, D. R. Luke, and J. Malick, "Local linear convergence for alternating and averaged nonconvex projections," *Foundations of Computational Mathematics*, vol. 9, no. 4, pp. 485–513, 2009.

[52] F. Andersson and M. Carlsson, "Alternating projections on nontangential manifolds," *Constructive approximation*, vol. 38, no. 3, pp. 489–525, 2013.

[53] A. Uschmajew and B. Vandereycken, "Geometric methods on low-rank matrix and tensor manifolds," in *Handbook of variational methods for nonlinear geometric data*, pp. 261–313, Springer, 2020.

[54] D. R. Luke, "Prox-regularity of rank constraint sets and implications for algorithms," *Journal of Mathematical Imaging and Vision*, vol. 47, no. 3, pp. 231–238, 2013.

[55] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3660–3677, 2012.