Electronic Transactions on Numerical Analysis.
Volume 62, pp. 58–71, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol62s58

ETNA
Kent State University and
Johann Radon Institute (RICAM)

RELAXATION OF THE RANK-1 TENSOR APPROXIMATION USING DIFFERENT NORMS*

HASSAN BOZORGMANESH†

Abstract. The best rank-1 approximation of a real mth-order tensor is equal to solving m 2-norm optimization problems that each corresponds to a factor of the best rank-1 approximation. In this paper, these problems are relaxed by using the Frobenius and L_1-norms instead of the 2-norm. It is shown that the solution for the Frobenius relaxation of optimization problems is the leading eigenvector of a positive semi-definite matrix which is closely related to higher-order singular value decomposition and the solution of the L_1-relaxation can be obtained efficiently by summing over all modes of the associated tensor but one. The numerical examples show that these relaxations can be used to initialize the alternating least-squares (ALS) method and they are reasonably close to the solutions obtained by the ALS method.

Key words. rank-1 approximation, relaxation, tensors, maximum Z-eigenvalue

AMS subject classifications. 15A18, 15A69

1. Introduction.

When dealing with high-dimensional data, using matrices and vectors to represent the data in a model can cause the loss of meaningful information and patterns, since we need to flatten the data. Tensors, as the extension of vectors and matrices, provide the natural tool for representing high-dimensional data [14, 21]. Similar to the matrix case, in order to extract the important patterns and to discard the redundant information, there is a need for dimensionality reduction techniques. Decomposing tensors and using the low-rank forms is one of these approaches. Using low-rank tensors is usual in applications like signal processing [7, 29, 45, 49] and optimal control [16, 44]. Of particular interest is the use of the canonical polyadic (CP) rank-1 approximation due to its lower complexity and because, unlike general CP decompositions, it always exists [35]. Its use in linear and nonlinear signal processing has generated some success [39, 43].

The study of different tensor decompositions and especially the best rank-1 approximation of a tensor has attracted the attention of many researchers in recent decades [3, 5, 11, 24, 25, 26, 30].

As an example, consider Volterra series which are widely used in signal processing for representing nonlinear systems [37, 39] and defined as follows:

$$y(i) = y_0 + \sum_{k=1}^{\infty} y_k(i) \tag{1.1}$$

with

$$y_k(i) = \sum_{i_1,\dots,i_k=0}^{M-1} \mathcal{H}_k(i_1,,\dots,i_k)x(i-i_1)\cdots x(i-i_k),$$

where $y(i)$ and $x(i)$, respectively, are the output and input signals, y_0 is a constant, M is the memory length, and $\mathcal{H}_k$ is the kth-order Volterra kernel. In practice, only a truncated expansion is used, that is, instead of infinity, a large enough integer is used in equation (1.1). Calculating a Volterra kernel can be significantly costly: one approach that is used in the literature in order to reduce the cost is approximating the Volterra kernel with a rank-1 tensor [13, 39].

*Received September 18, 2023. Accepted June 18, 2024. Published online on August 26, 2024. Recommended by D. Kressner.
†Department of Mathematics and Mathematical Statistics, Umeå University, 901 87, Umeå, Sweden (hassan.bozorgmanesh@umu.se).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

It is known that the problem of finding the best (CP) rank-1 approximation is NP-hard [23]. In this paper, the best (CP) rank-1 approximation of a real tensor is considered, that is, for an mth-order tensor $\mathcal{A}$ in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, we consider the following problem:

$$\min_{c,\, \|\mathbf{x}^{(1)}\|_2 = \|\mathbf{x}^{(2)}\|_2 = \cdots = \|\mathbf{x}^{(m)}\|_2 = 1} \|\mathcal{A} - c\mathbf{x}^{(1)} \circ \cdots \circ \mathbf{x}^{(m)}\|_F, \tag{1.2}$$

where c is a real scalar, $\mathbf{x}^{(i)} \in \mathbb{R}^{n_i}$ $(1 \leq i \leq m)$ are real vectors, and $\circ$ denotes the vector outer product. If the scalar μ and vectors $\mathbf{a}^{(i)} \in \mathbb{R}^{n_i}$ with $\|\mathbf{a}^{(i)}\|_2 = 1$ $(1 \leq i \leq m)$ are solutions of this problem, then $\mu \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(m)}$ is called the best rank-1 approximation of $\mathcal{A}$.

Here, we use two relaxations for the best rank-1 approximation of a real tensor in order to create cost-effective methods. The purpose of a relaxation of an optimization problem is to approximate it by weakening the constraints or the objective function in order to have an easier-to-solve problem that approximates the optimal solution of the original problem [34, 42]. The optimal solution of the relaxed maximization (minimization) problem is a lower bound (upper bound) for the optimal solution of the original problem.

In this paper, using relaxations leads to having simpler algorithms to calculate the best rank-1 approximation of a tensor. Also, the output of one of the relaxations (L_1) can be used to initialize alternating least-squares (ALS) and create a faster and more accurate version of ALS than with random initialization.

In the next section, we show that every best rank-1 approximation problem is equal to m 2-norm optimization subproblems. In Section 3, we relax these subproblems using Frobenius and L_1-norms, and provide how their solutions can be obtained. In Section 4, numerical results for tensors with random and real data are presented and, lastly, in Section 5 concluding remarks are given.

2. Preliminaries.

We first need to define the tensor–vector product and 2-norm for tensors.

DEFINITION 2.1. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$ is an mth-order tensor and $\mathbf{y} \in \mathbb{R}^{n_k}$ is a vector, then $\mathcal{A} \times_k \mathbf{y}$ denotes the mode-k product of $\mathcal{A}$ with $\mathbf{y}$. This product is of size $n_1 \times \cdots \times n_{k-1} \times n_{k+1} \times \cdots \times n_m$, and each element of this product is defined as follows:*

$$(\mathcal{A} \times_k \mathbf{y})_{i_1 \ldots i_{k-1} i_{k+1} \ldots i_m} = \sum_{i_k=1}^{n_k} \mathcal{A}(i_1, i_2, \ldots, i_m) y(i_k).$$

DEFINITION 2.2. *For an mth-order tensor like $\mathcal{A}$ in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, the spectral norm or 2-norm is defined as*

$$\|\mathcal{A}\|_2 = \max_{\|\mathbf{x}^{(1)}\|_2 = \|\mathbf{x}^{(2)}\|_2 = \cdots = \|\mathbf{x}^{(m)}\|_2 = 1} |\mathcal{A} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)} \cdots \times_m \mathbf{x}^{(m)}|.$$

It is clear that if $\mathcal{A}$ is a matrix, then the previous norm reduces to the 2-norm of a matrix and it gives the largest singular value, since $|\mathcal{A} \times_1 \mathbf{x}^{(1)} \times_2 \mathbf{x}^{(2)}| = |(\mathbf{x}^{(1)})^t \mathcal{A} \mathbf{x}^{(2)}| = \|\mathbf{x}^{(1)}\|_2 \|\mathcal{A}\mathbf{x}^{(2)}\|_2 |\cos\theta|$, where θ is the angle between the vectors $\mathbf{x}^{(1)}$ and $\mathcal{A}\mathbf{x}^{(2)}$.

Parts of the proof of the next theorem have been repeated several times in the literature (for example see [15, 40]).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

THEOREM 2.3. *If $\mathcal{A}$ is an mth-order tensor in $\mathbb{R}^{n_1\times n_2\times\cdots\times n_m}$, then to find the best rank-1 approximation of $\mathcal{A}$ (problem (1.2) above), we have*

$$\mathbf{a}^{(i)} = \arg\max_{\|\mathbf{x}^{(i)}\|_2=1} \|\mathcal{A}\times_i \mathbf{x}^{(i)}\|_2$$

for $i = 1, 2, \ldots, m$ and $\mu = \mathcal{A}\times_m \mathbf{a}^{(m)}\cdots\times_2 \mathbf{a}^{(2)}\times_1 \mathbf{a}^{(1)}$.

Proof. Suppose $\mu\mathbf{a}^{(1)}\circ\mathbf{a}^{(2)}\circ\cdots\circ\mathbf{a}^{(m)}$ is the best rank-1 approximation of $\mathcal{A}$. By expanding the minimization problem (1.2), we can rewrite it as

$$\max_{\|\mathbf{x}^{(1)}\|_2=\|\mathbf{x}^{(2)}\|_2=\cdots=\|\mathbf{x}^{(m)}\|_2=1} (\mathcal{A}\times_m \mathbf{x}^{(m)}\cdots\times_2 \mathbf{x}^{(2)}\times_1 \mathbf{x}^{(1)})^2. \tag{2.1}$$

Using the definition of the spectral norm, the result follows. □

The next corollary is the basis of the alternating least-squares (ALS) method [15].

COROLLARY 2.4. *For the best rank-1 approximation (1.2) of $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_m}$, we have*

$$\mathbf{a}^{(i)} = \frac{(\mathcal{A}\times_m \mathbf{a}^{(m)}\cdots\times_{i-1}\mathbf{a}^{(i-1)}\times_{i+1}\mathbf{a}^{(i+1)}\cdots\times_1\mathbf{a}^{(1)})}{\|\mathcal{A}\times_m \mathbf{a}^{(m)}\cdots\times_{i-1}\mathbf{a}^{(i-1)}\times_{i+1}\mathbf{a}^{(i+1)}\cdots\times_1\mathbf{a}^{(1)}\|_2}, \quad i=\{1,\ldots,m\}. \tag{2.2}$$

Proof. The result can be obtained by writing the Karush–Kuhn–Tucker (KKT) condition for every $\mathbf{a}^{(i)}$ in (2.1). □

3. Relaxation of the best rank-1 approximation of tensors.

3.1. Relaxation using the Frobenius norm. As stated in Theorem 2.3, the best rank-1 tensor approximation can be found by solving

$$\max_{\|\mathbf{x}^{(i)}\|_2=1} \|\mathcal{A}\times_i \mathbf{x}^{(i)}\|_2 \tag{3.1}$$

for $i = 1, 2, \ldots, m$. As a relaxation, we use different norms for the objective function of (3.1). Fix an $i\in\{1,2,\ldots,m\}$ and consider the maximization problem

$$\max_{\|\mathbf{x}^{(i)}\|_2=1} \|\mathcal{A}\times_i \mathbf{x}^{(i)}\|_F. \tag{3.2}$$

Since, for any tensor $\mathcal{B}$, we have $\|\mathcal{B}\|_2 \le \|\mathcal{B}\|_F$ [32] and the feasible domains of (3.1) and (3.2) are equal, then problem (3.2) is a relaxation of (3.1).

PROPOSITION 3.1. *Suppose $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times\cdots\times n_m}$. For an $i\in\{1,2,\ldots,m\}$, the optimal solution of optimization problem (3.2) is the eigenvector associated with the largest eigenvalue of the following symmetric positive semi-definite matrix in $\mathbb{R}^{n_i\times n_i}$:*

$$B_i(j,r) = \sum_{\substack{t=1,\\ t\neq i}}^{m}\sum_{k_t=1}^{n_t} \mathcal{A}(k_1,\ldots,k_{i-1},j,k_{i+1},\ldots,k_m)\mathcal{A}(k_1,\ldots,k_{i-1},r,k_{i+1},\ldots,k_m), \tag{3.3}$$

where $j, r\in\{1,\ldots,n_i\}$.

Proof. Considering the matricized form of $\mathcal{A}\times_i \mathbf{x}^{(i)}$, we can get the result. □

ETNA
Kent State University and
Johann Radon Institute (RICAM)

REMARK 3.2. Calculating the matrix B is equal to computing the singular vector associated with the leading singular value of the mode-i matricization of a tensor $\mathcal{A}$, which is closely related to the higher-order singular value decomposition (HOSVD) of a tensor. Therefore, it can be said that the truncated HOSVD is the relaxation of the best rank-1 CP approximation of a tensor. This sheds more light on the fact that left singular vectors have been used for initializing ALS [27] or other methods [6, 8, 26] for many years. SVD-based methods have been popular for calculating low-rank approximation of tensors. They have been modified to be used alternatively in a similar manner to ALS in order to calculate the best rank-1 approximation [15], albeit they update two factors in each iteration. Also, a variant based on different strategies for permutation of factor vectors for the symmetric best rank-1 approximation of tensors was proposed [20] with a convergence result. Another version of this method was used for non-symmetric tensors [19]. In addition, the SVD of matrix flattening of a tensor has been used to find a rank-1 decomposition that approximates its nuclear norm [31].

Therefore, to find the best rank-1 approximation of a tensor, we have the following relaxation scheme, which we call Frobenius relaxation.

Algorithm 1 Frobenius relaxation for best rank-1 approximation of a tensor.

Input: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_m}$.
Output: $\mu \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \ldots \circ \mathbf{a}^{(m)}$.

1: Choose one $j \in \{1, ..., m\}$ such that $j = \underset{i \in \{1,\ldots,m\}}{\arg\max} \{n_i\}$.
2: **for** $i = 1, ..., m, i \neq j$ **do**
3: Calculate $\mathbf{a}^{(i)}$ as the eigenvector associated with the largest eigenvalue of B_i, defined in (3.3).
4: **end for**
5: Calculate $\mathbf{a}^{(j)}$ from (2.2).
6: Let $\mu = \mathcal{A} \times_m \mathbf{a}^{(m)} \ldots \times_2 \mathbf{a}^{(2)} \times_1 \mathbf{a}^{(1)}$.

REMARK 3.3. Since every $\mathbf{a}^{(i)}$ is calculated separately, for a non-cubic tensor like $\mathcal{A} \in \mathbb{R}^{100 \times 200 \times 50}$ we choose $j = 2$ in Step 1, because it is faster to first obtain $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(3)}$ by Step 3 and then using (2.2) to obtain $\mathbf{a}^{(2)}$. The purpose of this is to solve eigenvalue problems with smaller sizes. Using Step 5 can make the Frobenius relaxation in one factor vector different than the corresponding leading left singular vector. In addition, the numerator of (2.2) can be used for calculating μ in Step 6. Therefore, it can be stated that Algorithm 1 is a variant of the truncated HOSVD [10] except for Step 5.

3.2. Relaxation using L_1-norm. Another relaxation can be defined for problem (3.1) by using the L_1-norm. The L_1-norm [31] of the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$ is defined as

$$\|\mathcal{A}\|_{L_1} = \sum_{t=1}^{m} \sum_{i_t=1}^{n_t} |\mathcal{A}(i_1, i_2, \ldots, i_m)|.$$

Therefore, the L_1-relaxation of problem (3.1) is as follows for every $i \in \{1, 2, \ldots, m\}$:

$$\max_{\|\mathbf{x}^{(i)}\|_2 = 1} \|\mathcal{A} \times_i \mathbf{x}^{(i)}\|_{L_1}. \tag{3.4}$$

Since for non-negative scalars $\{h_1, \ldots, h_k\}$, we have

$$\sqrt{\sum_{i=1}^{k} h_i^2} \leq \sum_{i=1}^{k} h_i,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

then for every tensor $\mathcal{B}$, it follows that $\|\mathcal{B}\|_2 \leq \|\mathcal{B}\|_F \leq \|\mathcal{B}\|_{L_1}$ and therefore (3.4) is a relaxation of (3.1).

THEOREM 3.4. *If $\mathcal{A}$ is a non-negative tensor in $\mathbb{R}^{n_1\times n_2\times\cdots\times n_m}$ and for an $i \in \{1,2,\ldots,m\}$, $\mathbf{a}^{(i)}$ is the optimal solution of optimization (3.4), then*

$$\mathbf{a}^{(i)} = \frac{\displaystyle\sum_{\substack{t=1,\\ t\neq i}}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(k_1,\ldots,k_{i-1},:,k_{i+1},\ldots,k_m)}{\left\|\displaystyle\sum_{\substack{t=1,\\ t\neq i}}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(k_1,\ldots,k_{i-1},:,k_{i+1},\ldots,k_m)\right\|_2}, \tag{3.5}$$

where $\mathcal{A}(k_1,\ldots,k_{i-1},:,k_{i+1},\ldots,k_m)$ is a vector obtained by fixing all indices of tensor $\mathcal{A}$ but the kth one.

Proof. We only consider the case $i = 1$. Like before, since $\max_{\|\mathbf{x}\|_2=1}\|(\mathcal{A}\times_1\mathbf{x})\|_{L_1}^2 = \max_{\|\mathbf{x}\|_2\leq 1}\|(\mathcal{A}\times_1\mathbf{x})\|_{L_1}^2$, the KKT condition is necessary and sufficient.

Since $\mathcal{A}$ is a non-negative tensor, we need only to consider non-negative solutions, and therefore

$$\frac{\partial}{\partial x_j}\|(\mathcal{A}\times_1\mathbf{x})\|_{L_1} = \frac{\partial}{\partial x_j}\left(\sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\sum_{r=1}^{n_1}\mathcal{A}(r,k_2,\ldots,k_m)x_r\right) = \sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(j,k_2,\ldots,k_m).$$

Hence, the KKT condition is as follows:

$$\sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(:,k_2,\ldots,k_m) - 2\lambda\mathbf{a}^{(i)} = 0$$

$$\Longrightarrow \quad \mathbf{a}^{(i)} = \frac{\displaystyle\sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(:,k_2,\ldots,k_m)}{2\lambda} = \frac{\displaystyle\sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(:,k_2,\ldots,k_m)}{\left\|\displaystyle\sum_{t=2}^{m}\sum_{k_t=1}^{n_t}\mathcal{A}(:,k_2,\ldots,k_m)\right\|}.$$

Thus, the proof is completed. ∎

Unlike the Frobenius relaxation, the L_1-relaxation is defined for non-negative (or non-positive) tensors, but, as we are going to see in the numerical experiments, the L_1-relaxation is much faster and it also can be used for tensors containing negative and positive elements with loss of some accuracy.

The L_1-relaxation is formally formulated as follows:

Algorithm 2 L_1-relaxation for best rank-1 approximation of a tensor.

Input: $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\ldots\times n_m}$.
Output: $\mu\mathbf{a}^{(1)}\circ\mathbf{a}^{(2)}\circ\ldots\circ\mathbf{a}^{(m)}$.

1: **for** $i = 1,\ldots,m$ **do**
2: Calculate $\mathbf{a}^{(i)}$ by (3.5).
3: **end for**
4: Let $\mu = \mathcal{A}\times_m\mathbf{a}^{(m)}\ldots\times_2\mathbf{a}^{(2)}\times_1\mathbf{a}^{(1)}$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

REMARK 3.5. It should be noted that, unlike most previous algorithms for calculating the best rank-1 approximation of a tensor, the Frobenius and L_1-relaxations calculate every $\mathbf{a}^{(i)}$ separately. This can be really helpful when we know a tensor is symmetric; in this case we need only to obtain one $\mathbf{a}^{(i)}$ and put it equal to the rest. Therefore, in the case of mth-order symmetric tensors, the calculation of the best rank-1 approximation by the Frobenius and L_1-relaxations can become m times faster.

REMARK 3.6. Considering that the Frobenius relaxation is based on calculating the eigenvalues of some matrices and L_1-relaxation is based on summing over all indices of tensors but one, both relaxations are of Nick's Class [52, p. 29]. Especially, for the Frobenius relaxation, the cost is $O(k^{2.376})$ while for L_1-relaxation it is $O(k)$, where k here is the largest mode of the tensor.

4. Numerical experiments. In this section, the two proposed relaxation methods are compared with the method given by He, Li, and Zhang (HLZ) [22], which is a relaxation; since this method is not iterative, it is possible to carry out comparisons. The ALS with initialization using random and relaxation methods are also compared here. By initialization using a relaxation method, we mean using the solution given by a relaxation method to initialize the ALS method. For running the ALS method, the Tensor Toolbox 3.6 is used [2, 28]. The stopping criterion for ALS is when the improvement is less than 10^{-4}. In the following, ALSr is used to denote ALS with random initialization.

The computations were made in MATLAB 2024a with an Intel® Core™ i7-13700H CPU 2.40 GHz and 16 GB RAM.

Suppose we have an algorithm named K for calculating the best rank-1 approximation. In the following, $T(K)$ denotes the timings of the algorithm K. Also, if $\mathcal{R}_K(\mathcal{A})$ is the computed best rank-1 approximations of a tensor $\mathcal{A}$ by the algorithm K, then

$$\frac{\|\mathcal{A}-\mathcal{R}_K(\mathcal{A})\|_F-\|\mathcal{A}-\mathcal{R}_{ALSr}(\mathcal{A})\|_F}{\|\mathcal{A}\|_F} \tag{4.1}$$

is used as a measure of the error for the algorithm K, denoted then by $E(K)$. We note that if $E(K)$ is negative, that means the algorithm is more accurate than the ALSr method.

EXAMPLE 4.1. For this example, random tensors are used. For tensors of a given size, the timings and errors are averages over 100 runs of the algorithms. First, we consider tensors whose every element is a uniformly distributed random number in $[0, 1]$. The results for third-order and fourth-order tensors are given in Table 4.1(A) and Table 4.2(A), respectively. As we know, the best rank-1 approximation of a non-negative tensor is almost always unique [41]. All algorithms are close to the same rank-1 tensor. It can be seen that the relative accuracy of both relaxations and HLZ increase as the size of the tensors increases. However, if we do not divide the fraction in (4.1), that is, if we only consider the absolute error, then the errors of random tensors are going to be around the values given by the second row of Table 4.1(A) for non-negative random tensors and Table 4.1(B) for tensors containing both negative and positive elements.

The Frobenius relaxation is fastest for tensors with small sizes; the L_1-relaxation is more accurate than the Frobenius relaxation and HLZ, and it is also the fastest method except for small-sized tensors. For large tensors, the L_1-relaxation has a clear advantage over the Frobenius relaxation and HLZ in timing due to the cost of building matrices and calculating eigenvalues. HLZ is a bit faster than Frobenius relaxation but less accurate.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.1
Comparing Frobenius relaxation, L_1-relaxation, and HLZ for third-order random tensors. All algorithms are run 100 times for each row. All times T are in seconds.

A) Each member of the tensors is a uniformly distributed random number in $[0, 1]$.

n_1	n_2	n_3	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)	T(ALSr)
10	10	10	7e−04	9e−04	8e−04	1.11e−04	4.07e−05	0.0046	0.0036
20	20	20	0.0021	0.0012	0.0015	1.68e−05	2.87e−06	0.0012	0.0046
50	50	50	0.0028	0.0019	0.0027	1.23e−06	7.62e−08	1.90e−04	0.0065
100	100	100	0.0067	0.0024	0.0073	1.59e−07	4.60e−09	4.93e−05	0.0089
150	150	150	0.0149	0.0059	0.0142	4.82e−08	9.18e−10	2.18e−05	0.0155
200	200	200	0.0412	0.0108	0.0323	2.04e−08	2.96e−10	1.25e−05	0.0296
500	500	500	0.7694	0.1038	0.5479	1.31e−09	7.54e−12	2.01e−06	0.3345
50	75	100	0.0042	0.0019	0.0041	4.18e−07	1.77e−08	9.79e−05	0.0068
10	50	200	0.0016	0.0013	0.0016	1.44e−06	2.02e−07	2.49e−04	0.0048
100	200	500	0.0363	0.0117	0.0358	1.64e−08	2.73e−10	1.01e−05	0.0339
150	300	600	0.1064	0.0311	0.0877	6.04e−09	6.94e−11	5.52e−06	0.0804
100	300	700	0.0791	0.0250	0.0707	7.78e−09	1.16e−10	7.14e−06	0.0640
200	400	800	0.3094	0.0641	0.2871	2.56e−09	2.21e−11	3.08e−06	0.1791
100	200	1000	0.0748	0.0254	0.0740	8.10e−09	1.30e−10	4.89e−06	0.0653

B) Each member of the tensors is a uniformly distributed random number in $[-1, 1]$.

n_1	n_2	n_3	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)	T(ALSr)
10	10	10	8e−04	9e−04	8e−04	0.0133	0.0288	0.0138	0.0081
20	20	20	0.0018	0.0011	0.0017	0.0046	0.0078	0.0034	0.0109
50	50	50	0.0029	0.0019	0.0032	7.38e−04	0.0011	3.71e−04	0.0094
100	100	100	0.0061	0.0024	0.0069	1.48e−04	2.24e−04	3.01e−05	0.0090
150	150	150	0.0188	0.0056	0.0156	6.89e−05	9.96e−05	1.37e−05	0.0167
200	200	200	0.0384	0.0106	0.0303	3.90e−05	5.58e−05	7.36e−06	0.0290
500	500	500	0.7661	0.1121	0.5397	6.64e−06	9.00e−06	1.12e−06	0.3453
50	75	100	0.0043	0.0020	0.0043	3.93e−04	5.11e−04	1.44e−04	0.0078
10	50	200	0.0039	0.0017	0.0021	0.0018	0.0021	6.87e−04	0.0081
100	200	500	0.0403	0.0132	0.0333	4.90e−05	5.69e−05	5.47e−06	0.0350
150	300	600	0.1069	0.0326	0.0915	2.42e−05	2.80e−05	3.49e−06	0.0822
100	300	700	0.0878	0.0282	0.0773	3.35e−05	3.67e−05	5.87e−06	0.0702
200	400	800	0.3449	0.0726	0.3276	1.37e−05	1.59e−05	1.94e−06	0.1985
100	200	1000	0.0774	0.0246	0.0755	3.87e−05	4.26e−05	1.25e−07	0.0659

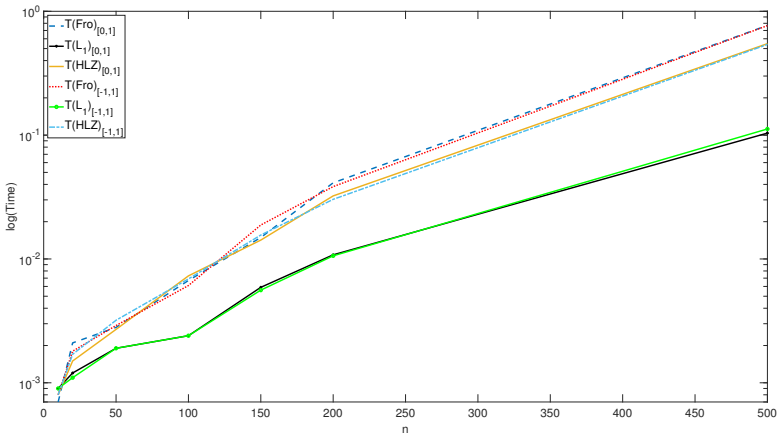

FIG. 4.1. *Log(Time) versus n for cubic random third-order tensors in $\mathbb{R}^{n\times n\times n}$.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.2
Comparing Frobenius relaxation, L_1-relaxation, and HLZ for fourth-order random tensors. All algorithms are run 100 times for each row. All times T are in seconds.

A) Each member of the tensors is a uniformly distributed random number in $[0, 1]$.

n_1	n_2	n_3	n_4	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)	T(ALSr)
10	10	10	10	0.0012	9e−04	0.0018	1.95e−05	8.04e−07	4.59e−04	0.0055
20	20	20	20	0.0036	0.0023	0.0047	1.43e−06	1.25e−08	5.85e−05	0.0092
40	40	40	40	0.0106	0.0035	0.0128	9.11e−08	2.06e−10	7.79e−06	0.0162
100	100	100	100	0.3260	0.0904	0.3092	2.45e−09	8.59e−13	4.94e−07	0.4206
10	20	30	40	0.0041	0.0018	0.0053	8.97e−07	7.77e−09	6.03e−05	0.0084
70	80	90	100	0.1514	0.0425	0.1602	4.89e−09	2.44e−12	8.90e−07	0.2062
10	20	30	200	0.0079	0.0034	0.0139	1.83e−07	1.10e−09	1.24e−05	0.0137
10	50	100	500	0.1067	0.0416	0.1138	9.20e−09	1.59e−11	1.99e−06	0.1608

B) Each member of the tensors is a uniformly distributed random number in $[-1, 1]$.

n_1	n_2	n_3	n_4	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)	T(ALSr)
10	10	10	10	0.0013	9e−04	0.0019	0.0031	0.0039	0.0014	0.0104
20	20	20	20	0.0034	0.0016	0.0048	3.42e−04	4.15e−04	1.03e−04	0.0095
40	40	40	40	0.0123	0.0045	0.0142	3.82e−05	4.72e−05	8.90e−06	0.0186
100	100	100	100	0.2961	0.0817	0.2888	2.54e−06	3.11e−06	7.58e−07	0.4008
10	20	30	40	0.0049	0.0022	0.0068	2.25e−04	3.28e−04	1.31e−04	0.0121
70	80	90	100	0.1463	0.0406	0.1534	4.10e−06	5.10e−06	1.55e−06	0.2130
10	20	30	200	0.0057	0.0026	0.0112	4.63e−05	1.39e−04	6.17e−05	0.0108
10	50	100	500	0.0999	0.0378	0.1087	6.61e−06	1.72e−05	1.01e−05	0.1576

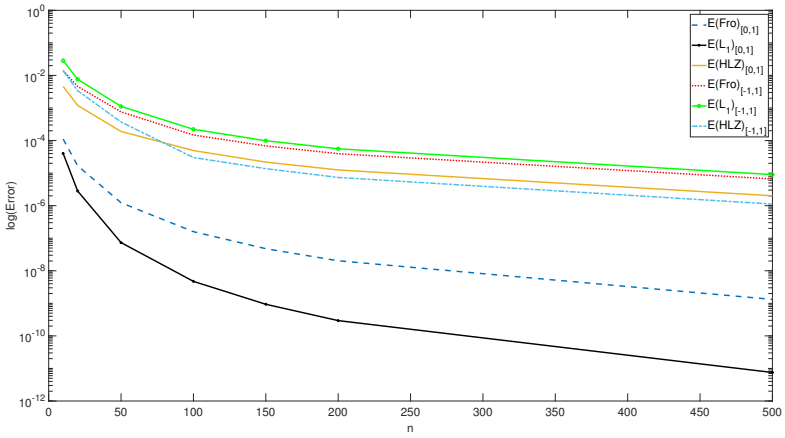

FIG. 4.2. *Log(Error) versus n for cubic random third-order tensors in $\mathbb{R}^{n\times n\times n}$.*

Secondly, we consider random tensors whose elements are chosen uniformly distributed in $[-1, 1]$. The results are presented in Table 4.1(B) and Table 4.2(B). In this case, since we do not have a uniqueness result, every method points to a different rank-1 tensor. It can be seen that the Frobenius relaxation is more accurate than the L_1-relaxation, but the timings are similar to the case of non-negative tensors. HLZ for this type of tensor is more accurate

ETNA
Kent State University and
Johann Radon Institute (RICAM)

than the other two methods. The ALSr consumes more time to calculate rank-1 tensors for small-sized tensors in $[-1, 1]$ in comparison to the case of non-negative tensors.

Also, we should mention that for non-cubic tensors, Step 1 of Algorithm 1 is vital for the Frobenius relaxation. It can be seen that j should be selected as 3 and 4 in Step 1 of Algorithm 1 for third-order and fourth-order non-cubic tensors in Table 4.1 and Table 4.2, respectively. The plots of the logarithms of calculation times and errors versus the size of cubic tensors of this example are presented in Figure 4.1 and Figure 4.2, respectively.

EXAMPLE 4.2. In this example, we use six real chemometric data tensors[1] and two other tensors, Traffic Speed [33] and Brain Network [48]. The results are given in Table 4.3. The excitation-emission matrices (EEM), flow injection analysis [36], lipoproteins [12], and fluorescence [4] databases consist of negative and positive elements, the wine gas chromatography–mass spectrometry (GC-MS) data [46] and Traffic Speed are non-negative, the porphyrin data [51] is positive, and the Brain Network tensor is binary. Also, only 10.5% of the wine GC-MS data is non-zero. The outputs of Frobenius relaxation are close to the ALSr, but for the L_1-relaxation, unlike random tensors, the error is significantly larger. The Frobenius relaxation, by using Step 1 of Algorithm 1, can become significantly faster. HLZ performs poorly for most of the third-order tensors, but it works much better for the fourth-order fluorescence tensor. Interestingly L_1-relaxation also performs better for this tensor.

The L_1-relaxation is faster than the other two methods, but it is less accurate. Since the L_1-relaxation is much faster than the ALSr, we can use it as an initialization for the ALS, that is, the solution of L_1-relaxation is used for initializing the ALS; we call this combination of these two methods, ALS-L_1. The regular ALS uses the random initialization. We also used Frobenius relaxation (ALSFro) and HLZ (ALSHLZ) for initializing the ALS method; these four methods are compared in Table 4.4. As can be seen from the table, for some tensors, ALSFro and ALS-L_1 managed to get to the solution faster, while the ALSHLZ is always slower than ALSr. In addition, ALSFro and ALSHLZ could get a more accurate solution for five out of seven tensors while this number for ALS-L_1 is six.

The computed time of ALS with a relaxation initialization is the sum of the consumed time for that relaxation plus the running time of ALS with the initialization using the output of the relaxation.

EXAMPLE 4.3. As the last example, five large sparse tensors are used from FROSTT [47]; three fourth-order tensors, NIPS [17] (count of words, 1.8e$-$04% non-zero elements), Chicago Crime[2] (number of crimes, 1.5% non-zero elements), and Flickr [18] (user has tagged an image, binary); and also, two fifth-order tensors, LBNL-Network [38] (packet length sent in a timestep, positive, 4e$-$12% non-zero elements), and VAST2015 MC1 [50] (attendance, binary, 7.7e$-$05% non-zero elements).

The results of computations using relaxations are given in Table 4.5. Here, there is a clear distinction between the computation time of L_1-relaxations and the other ones, L_1-relaxation is much faster and has a similar accuracy to other methods. From an accuracy point of view, HLZ performs better for this set of tensors than the previous example.

Again, the different initializations of ALS method are compared for this example. Table 4.6 shows that the computational cost of Frobenius and HLZ relaxations makes the corresponding ALS methods unreasonably costly. The only relaxation that could compete with ALSr computationally and reduce the cost for some tensors is ALS-L_1. Accuracy-wise, ALS-L_1 also performs better and gives a better accuracy in four out of five tensors, while this number for ALSFro and ALSHLZ is only one.

[1] Available at: https://ucphchemometrics.com/datasets/.
[2] City of Chicago data portal.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.3
Comparing Frobenius relaxation, L_1-relaxation, and HLZ for real data tensors. All times T are in seconds.

Database	n_1	n_2	n_3	n_4	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)
EEM (Y)	28	13324	8	–	0.0097	0.0083	0.0113	4.52e−04	0.0586	0.2418
Porphyrin	170	274	35	–	0.0139	0.0041	0.0118	7.56e−04	0.0596	0.1400
Wine GC-MS	44	2700	200	–	0.0609	0.0409	0.1418	1.03e−04	0.1893	0.4515
Lipoproteins	25	24	1600	–	0.0052	0.0027	0.0065	3.82e−06	0.0344	0.1174
Flow injection	12	100	89	–	0.0031	0.0017	0.0041	5.61e−04	0.0107	0.1539
Fluorescence	405	136	19	5	0.0253	0.0110	0.0365	2.77e−05	7.39e−05	4.2208e−05
Traffic Speed	216	61	144	–	0.0105	0.0032	0.0080	2.05e−05	5.12e−04	0.0035
Brain Network	68	68	212	–	0.0076	0.0034	0.0063	4.72e−06	0.0019	0.0405

TABLE 4.4
Comparing ALS with initializations using random (ALSr), L_1-relaxation (ALS-L_1), Frobenius relaxation (ALSFro), and HLZ (ALSHLZ). All times T are in seconds.

Database	n_1	n_2	n_3	n_4	T(ALSr)	T(ALSFro)	T(ALSL_1)	T(ALSHLZ)	E(ALSFro)	E(ALSL_1)	E(ALSHLZ)
EEM (Y)	28	13324	8	–	0.0491	0.0550	0.0370	0.0600	−2.18e−07	−4.09e−07	+8.04e−08
Porphyrin	170	274	35	–	0.0190	0.0227	0.0204	0.0246	−1.36e−07	−1.26e−06	−3.32e−07
Wine GC-MS	44	2700	200	–	0.2548	0.2323	0.1728	0.3339	−5.58e−06	−4.54e−07	−1.83e−06
Lipoproteins	25	24	1600	–	0.0113	0.0156	0.0169	0.0206	−2.40e−08	−1.70e−08	−1.49e−08
Flow injection	12	100	89	–	0.0089	0.0129	0.0132	0.0143	+7.98e−07	-3.60e−07	-2.95e−07
Fluorescence	405	136	19	5	0.0693	0.0702	0.0801	0.0945	+1.94e−07	−3.00e−08	−1.30e−07
Traffic Speed	216	61	144	–	0.0182	0.0191	0.0280	0.0331	−1.50e−10	−3.82e−10	−3.94e−10
Brain Network	68	68	212	–	0.0148	0.0195	0.0140	0.0183	−1.13e−06	+6.44e−07	+4.90e−07

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 4.5
Comparing the HLZ, Frobenius relaxation, and L_1-relaxation for real data tensors. All times T are in seconds.

Database	n_1	n_2	n_3	n_4	n_5	T(Fro)	$T(L_1)$	T(HLZ)	E(Fro)	$E(L_1)$	E(HLZ)
NIPS	2482	2862	14036	17	–	4.1266	0.1769	3.1679	1.16e−04	7.88e−05	1.93e−04
Chicago Crime	6186	24	77	32	–	9.1623	0.2631	15.6486	0.0165	0.0016	0.0928
Flickr	10000	100000	200000	731	–	1.5244	0.0304	0.7096	0.0044	0.0022	2.9250e−04
LBNL-Network	1605	4198	1631	4209	868131	2.2166	0.1211	0.8291	0.0650	0.0419	0.0123
VAST2015 MC1	165427	11374	2	100	89	66.9776	1.7924	24.5954	5.07e−06	3.83e−07	5.05e−06

TABLE 4.6
Comparing ALS with initializations using random (ALSr), L_1-relaxation ($ALSL_1$), Frobenius relaxation (ALSFro), and HLZ (ALSHLZ). All times T are in seconds.

Database	n_1	n_2	n_3	n_4	n_5	T(ALSr)	T(ALSFro)	$T(\text{ALS}L_1)$	T(ALSHLZ)	E(ALSFro)	$E(\text{ALS}L_1)$	E(ALSHLZ)
NIPS	2482	2862	14036	17	–	0.9433	5.2441	0.8362	5.0001	−8.22e−04	−2.63e−04	−1.67e−04
Chicago Crime	6186	24	77	32	–	1.5897	11.3448	1.3663	16.6996	+5.21e−05	+4.02e−06	0.0070
Flickr	10000	100000	200000	731	–	0.3529	1.6128	0.3226	0.8487	0.0020	−0.0016	0.0020
LBNL-Network	1605	4198	1631	4209	868131	6.9415	10.7160	7.5660	4.7040	+1.21e−05	−0.0067	0.0375
VAST2015 MC1	165427	11374	2	100	89	14.5325	80.2956	15.9605	37.7069	+9.05e−07	−4.61e−08	+4.95e−06

ETNA
Kent State University and
Johann Radon Institute (RICAM)

5. Conclusion. In this paper, relaxing the NP-hard tensor best rank-1 approximation resulted in having simpler algorithms. Also, the initialization of ALS by the L_1-relaxation led to a faster and more accurate version of ALS than with random inputs.

In addition, the proposed algorithms can be used to approximate the maximum Z-eigenvalue of symmetric tensors and solutions of multivariate quadratic systems [9], since both have close ties to the problem of best rank-1 approximation of tensors.

Acknowledgments. During this work, the author was supported by the Kempe Foundations under grant JCK-2022.2. The author is grateful to the editor and two anonymous reviewers for their helpful comments which improved the quality of this paper.

REFERENCES

[1] E. ACAR, E. E. PAPALEXAKIS, G. GÜRDENIZ, M. A. RASMUSSEN, A. J. LAWAETZ, M. NILSSON, AND R. BRO, *Structure-revealing data fusion*, BMC Bioinformatics, 15 (2014), pp. 1–17.

[2] B. W. BADER AND T. G. KOLDA, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653.

[3] H. BOZORGMANESH AND M. HAJARIAN, *Triangular decomposition of CP factors of a third-order tensor with application to solving nonlinear systems of equations*, J. Sci. Comput., 90 (2022), Paper No. 74, 25 pages.

[4] R. BRO, Å. RINNAN, AND N. K. M. FABER, *Standard error of prediction for multilinear PLS: 2. Practical implementation in fluorescence spectroscopy*, Chemometrics Intell. Lab. Syst., 75 (2005), pp. 69–76.

[5] M. CHE, A. CICHOCKI, AND Y. WEI, *Neural networks for computing best rank-one approximations of tensors and its applications*, Neurocomputing, 267 (2017), pp. 114–133.

[6] J. CHEN AND Y. SAAD, *On the tensor SVD and the optimal low rank orthogonal approximation of tensors*, SIAM J. Matrix Anal. Appl., 30 (2008/09), pp. 1709–1734.

[7] A. CICHOCKI, D. MANDIC, L. DE LATHAUWER, G. ZHOU, Q. ZHAO, C. CAIAFA, AND H. A. PHAN, *Tensor decompositions for signal processing applications: From two-way to multiway component analysis*, IEEE Signal Process. Mag., 32 (2015), pp. 145–163.

[8] A. P. DA SILVA, P. COMON, AND A. L. DE ALMEIDA, *An iterative deflation algorithm for exact CP tensor decomposition*, in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE Conference Proceedings, Los Alamitos, 2015, pp. 3961–3965.

[9] ———, *On the reduction of multivariate quadratic systems to best rank-1 approximation of three-way tensors*, Appl. Math. Lett., 62 (2016), pp. 9–15.

[10] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

[11] ———, *On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[12] M. DYRBY, M. PETERSEN, A. K. WHITTAKER, L. LAMBERT, L. NØRGAARD, R. BRO, AND S. B. ENGELSEN, *Analysis of lipoproteins using 2D diffusion-edited NMR spectroscopy and multi-way chemometrics*, Analytica Chimica Acta, 531 (2005), pp. 209–216.

[13] G. FAVIER, A. Y. KIBANGOU, AND T. BOUILLOC, *Nonlinear system modeling and identification using Volterra-PARAFAC models*, Internat. J. Adapt. Control Signal Process., 26 (2012), pp. 30–53.

[14] S. FERNANDES, H. FANAEE-T, AND J. GAMA, *NORMO: A new method for estimating the number of components in CP tensor decomposition*, Eng. Appl. Art. Intell., 96 (2020), Paper No. 103926, 9 pages.

[15] S. FRIEDLAND, V. MEHRMANN, R. PAJAROLA, AND S. K. SUTER, *On best rank one approximation of tensors*, Numer. Linear Algebra Appl., 20 (2013), pp. 942–955.

[16] S. GARREIS AND M. ULBRICH, *Constrained optimization with low-rank tensors and applications to parametric problems with PDEs*, SIAM J. Sci. Comput., 39 (2017), pp. A25–A54.

[17] A. GLOBERSON, G. CHECHIK, F. PEREIRA, AND N. TISHBY, *Euclidean embedding of co-occurrence data*, J. Mach. Learn. Res., 8 (2007), pp. 2265–2295.

[18] O. GÖRLITZ, S. SIZOV, AND S. STAAB, *PINTS: Peer-to-peer infrastructure for tagging systems.*, in IPTPS'08: Proceedings of the 7th International Conference on Peer-To-Peer Systems, USENIX Association, Tampa Bay, 2008, p. 19.

[19] Y. GUAN, M. T. CHU, AND D. CHU, *Convergence analysis of an SVD-based algorithm for the best rank-1 tensor approximation*, Linear Algebra Appl., 555 (2018), pp. 53–69.

[20] ———, *SVD-based algorithms for the best rank-1 approximation of a symmetric tensor*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1095–1115.

[21] M. HAJARIAN, *Conjugate gradient-like methods for solving general tensor equation with Einstein product*, J. Franklin Inst., 357 (2020), pp. 4272–4285.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[22] S. He, Z. Li, and S. Zhang, *Approximation algorithms for homogeneous polynomial optimization with quadratic constraints*, Math. Program., 125 (2010), pp. 353–383.
[23] C. J. Hillar and L.-H. Lim, *Most tensor problems are NP-hard*, J. ACM, 60 (2013), Paper No. 45, 39 pages.
[24] S. Hu, *An inexact augmented Lagrangian method for computing strongly orthogonal decompositions of tensors*, Comput. Optim. Appl., 75 (2020), pp. 701–737.
[25] S. Hu, D. Sun, and K.-C. Toh, *Best nonnegative rank-one approximations of tensors*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 1527–1554.
[26] E. Kofidis and P. A. Regalia, *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl., 23 (2001/02), pp. 863–884.
[27] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[28] ———, *MATLAB Tensor Toolbox*, Tech. Rep., Sandia National Laboratories, 2021.
[29] S. Kouchaki and S. Sanei, *Tensor factorisation for narrowband single channel source decomposition*, J. Franklin Inst., 354 (2017), pp. 3152–3169.
[30] D. Kressner, M. Steinlechner, and B. Vandereycken, *Low-rank tensor completion by Riemannian optimization*, BIT, 54 (2014), pp. 447–468.
[31] Z. Li, *Bounds on the spectral norm and the nuclear norm of a tensor based on tensor partitions*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1440–1452.
[32] Z. Li, Y. Nakatsukasa, T. Soma, and A. Uschmajew, *On orthogonal tensors and best rank-one approximation ratio*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 400–425.
[33] T. Liao, J. Yang, C. Chen, and Z. Zheng, *A neural tensor decomposition model for high-order sparse data recovery*, Information Sci., 658 (2024), Paper No. 120024, 18 pages.
[34] G. Nemhauser, M. Todd, and A. Kan, *Optimization*, Handbooks in Operations Research and Management Science, Elsevier, North-Holland, Amsterdam, 1989.
[35] J. Nie and L. Wang, *Semidefinite relaxations for best rank-1 tensor approximations*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1155–1179.
[36] L. Nørgaard and C. Ridder, *Rank annihilation factor analysis applied to flow injection analysis with photodiode-array detection*, Chemometrics Intell. Lab. Syst., 23 (1994), pp. 107–114.
[37] T. Ogunfunmi, *Adaptive Nonlinear System Identification*, Springer, New York, 2007.
[38] R. Pang, M. Allman, V. Paxson, and J. Lee, *The devil and packet trace anonymization*, ACM SIGCOMM Comput. Commun. Rev., 36 (2006), pp. 29–38.
[39] F. C. Pinheiro and C. G. Lopes, *A low-complexity nonlinear least mean squares filter based on a decomposable Volterra model*, IEEE Trans. Signal Process., 67 (2019), pp. 5463–5478.
[40] L. Qi, *Rank and eigenvalues of a supersymmetric tensor, the multivariate homogeneous polynomial and the algebraic hypersurface it defines*, J. Symbolic Comput., 41 (2006), pp. 1309–1327.
[41] Y. Qi, P. Comon, and L.-H. Lim, *Uniqueness of nonnegative tensor approximations*, IEEE Trans. Inform. Theory, 62 (2016), pp. 2170–2183.
[42] R. L. Rardin, *Optimization in Operations Research*, Prentice Hall, Upper Saddle River, 1998.
[43] M. Rupp and S. Schwarz, *Gradient-based approaches to learn tensor products*, in 2015 23rd European Signal Processing Conference (EUSIPCO), IEEE Conference Proceedings, Los Alamitos, 2015, pp. 2486–2490.
[44] B. Schmitt, B. N. Khoromskij, V. Khoromskaia, and V. Schulz, *Tensor method for optimal control problems constrained by fractional three-dimensional elliptic operator with variable coefficients*, Numer. Linear Algebra Appl., 29 (2022), Paper No. e2404, 29 pages.
[45] 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 Trans. Signal Process., 65 (2017), pp. 3551–3582.
[46] T. Skov, D. Ballabio, and R. Bro, *Multiblock variance partitioning: a new approach for comparing variation in multiple data blocks*, Analytica Chimica Acta, 615 (2008), pp. 18–29.
[47] S. Smith, J. W. Choi, J. Li, R. Vuduc, J. Park, X. Liu, and G. Karypis, *FROSTT: The formidable repository of open sparse tensors and tools*, 2017. http://frostt.io/
[48] L. Wang, Z. Zhang, and D. Dunson, *Common and individual structure of brain networks*, Ann. Appl. Stat., 13 (2019), pp. 85–112.
[49] Y. Wang, S. Tang, and X. Gu, *Parameter estimation for nonlinear Volterra systems by using the multi-innovation identification theory and tensor decomposition*, J. Franklin Inst., 359 (2022), pp. 1782–1802.
[50] M. Whiting, K. Cook, G. Grinstein, J. Fallon, K. Liggett, D. Staheli, and J. Crouser, *VAST challenge 2015: Mayhem at Dinofun World*, in Visual Analytics Science and Technology (VAST) 2015, IEEE Conference Proceedings, Los Alamitos, 2015, pp. 113–118.
[51] J. Wold, R. Bro, A. Veberg, F. Lundby, A. Nilsen, and J. Moan, *Active photosensitizers in butter detected by fluorescence spectroscopy and multivariate curve resolution*, J. Agricult. Food Chem., 54 (2006), pp. 10197–10204.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[52] Y. YE, *A new complexity result on minimization of a quadratic function with a sphere constraint*, in Recent Advances in Global Optimization, C. A. Floudas and P. M. Pardalos, eds., Princeton University Press, Princeton, 1991, pp. 19–31.