

RESEARCH

Open Access



# EA-ADMM: noisy tensor PARAFAC decomposition based on element-wise average ADMM

Gang Yue and Zhuo Sun\*

\*Correspondence:  
zhuosun@bupt.edu.cn

Laboratory of Wireless Signal  
Processing and Network,  
Beijing University of Posts  
and Telecommunications,  
Beijing 100876, China

## Abstract

Tensor decomposition is widely used to exploit the internal correlation in multi-way data analysis and process for communications and radar systems. As one of the main tensor decomposition methods, CANDECOMP/PARAFAC decomposition has advantages of uniqueness and interpretation properties which are significant in practical applications. However, traditional decomposition method is sensitive to both predefined rank and noise that results in inaccurate tensor decomposition. In this paper, we propose a improved algorithm called the Element-wise Average Alternating Direction Method of Multipliers by minimizing the sum of all factors' trace norm and the noise variance. Our algorithm could overcome the dependence on predefined rank in traditional decomposition algorithms and alleviate the impact of noise. Moreover, this algorithm can be transferred to solve the problem of tensor completion conveniently. The simulation results show that our proposed algorithm could decompose the noisy tensor to the factors with above 90% similarity in various SNR and also interpolate the incomplete tensor with higher similar coefficient and lower relative reconstruction error when the missing rate is less than 0.5.

**Keywords:** Tensor decomposition, Noise interference, Element-wise average, Tensor recovery, Tensor completion

## 1 Introduction

Traditional matrix model of modern communications and radar systems with large-scale terminals has limitations to process the massive volume of signals and has been shifted towards more versatile data analysis tool, namely tensor [1, 2]. Tensor provide a natural and compact representation for such multi-ways data, and can be used to express more complicated intrinsic structures in higher-order data [3]. For further analysis of data components, tensor decomposition has great flexibility in the choice of data constraints and could decompose more interpretable latent components than matrix-based methods [4, 5].

CANDECOMP/PARAFAC (CP) decomposition [6] has been widely used in many applications [7–10] due to its attractive property of uniqueness which is essential for solving practical problems. Moreover, it is analogous to singular value decomposition

(SVD) of matrix in expression format which decomposes a tensor into the sum of several rank-one tensors. The basic CP decomposition approach named alternating least squares (ALS) could decompose a tensor accurately if the actual rank is already known. However, one main obstacle of ALS in practical is that the tensor actual rank is usually unknown which results in the inappropriate decomposed factors. Another obstacle is that its performance is sensitive to the noise due to the least-square formulations. The measured tensor data is common to be interfered by noise in the process of signal acquisition which would degenerate the decomposition performance by ALS. So accurate tensor recovery from noisy measurements is the primary problem.

For accurately fitting the CP model, existing approaches include the Difference in FIT (DIFFIT) [11], Numerical Convex Hull (NumConvHull) [12], CORE CONsistency DIAgnostic (CORCONDIA) [13], Automatic Relevance Determination (ARD) [14], and reconstruction-err-based CP rank selector [15]. All of them enumerate the possible ranks and calculate the factors based on the ALS algorithm that naturally involves multiple batches to converge and thus time-consuming. To avoid the influence of tensor unknown rank, the regular approach is to constrain the total ranks of all factors and reconstructed error, then optimize this objective function by alternating direction method of multipliers (ADMM) [16] or block coordinate descent (BCD) [17] on the assumption that the tensor is noiseless. However, the noise interference is inevitable and affects the tensor actual rank which results in inaccurate decomposed factors. Therefore, both the unknown CP rank and the noise should be considered simultaneously for tensor recovery by ADMM.

In this paper, we propose an element-wise average ADMM algorithm (EA-ADMM) to decompose a tensor interfered by independently identical distribution noise into correct factors and then verify the effectiveness in various signal-noise-ratio (SNR). Besides, we testify the tensor completion capability of this algorithm. All experiments show that our algorithm not only recover the actual components of a noisy tensor effectively but also complete the missing elements in the tensor accurately.

The rest of this paper is organized as follows. The preliminary of CP decomposition is introduced in Sect. 2 and then our element-wise average ADMM algorithm is detailed in Sect. 3. The experiments about tensor recovery and completion are implemented in Sect. 4. In the last Sect. 5, we conclude the results and figure out the future works.

## 2 Preliminary

Before reviewing CP decomposition, we firstly introduce the basic tensor notions and related operations. Scalars are denoted by lower-case letters such as  $i, j, k$ , vectors are denoted by bold lower-case letters such as  $\mathbf{a}, \mathbf{b}, \mathbf{c}$  and matrices are denoted by upper-case letters, e.g.,  $X$ . As for tensor, an  $N$ th-order tensor is denoted by a calligraphic letter, e.g.,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , and its elements are denoted by  $x_{i_1, \dots, i_n}$ . The order  $N$  of a tensor is the number of dimensions, also known as ways or modes. A tensor  $\mathcal{X}$  which is unfolded along mode- $n$  also named as matricization can be denoted by  $X_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ . Tensor matricization rearranges the elements of  $\mathcal{X}$  into the matrix  $X_{(n)}$  in lexicographical order and makes rank calculation convenient.

The inner product of two tensors with same size  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is the sum of the element-wise product.

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum_{i_1, \dots, i_n} a_{i_1, \dots, i_n} b_{i_1, \dots, i_n} \tag{1}$$

The Frobenius norm of an  $N$ th-order  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is defined as

$$\|\mathcal{X}\|_F := \sqrt{\sum_{i_1=1}^{I_1} \dots \sum_{i_n=1}^{I_n} x_{i_1, \dots, i_n}^2} \tag{2}$$

Assume  $A$  and  $B$  as two matrices with the size  $m \times n$  and  $p \times q$  respectively. The Kronecker product of these two matrices denoted by  $A \otimes B$  is an  $mp \times nq$  matrix given by

$$A \otimes B := [a_{ij}B]_{mp \times nq} \tag{3}$$

If  $A = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_r]$  and  $B = [\mathbf{b}_1 \mathbf{b}_2 \dots \mathbf{b}_r]$  are two matrices with same column number, the Khatri-Rao product of these two matrices is defined as the column-wise Kronecker product and represented by the operator  $\odot$

$$A \odot B := [a_1 \otimes b_1 \dots a_r \otimes b_r] \tag{4}$$

Tensor decomposition contains two main approaches namely Tucker decomposition [18] and CP decomposition. Tucker decomposition outputs a core tensor and related factors by calculating singular value decomposition of each  $n$ -mode matrices and then multiplying the tensor with each left singular matrix. Its results lack uniqueness due to the matrix SVD operation. However, CP decomposition as the specific case of Tucker decomposition constrains the core tensor as super-diagonal identity tensor and could decomposes a tensor into the sum of several rank-one tensors. Each rank-one tensor is generated by the outer product of the same column in all factors. CP decomposition only outputs each mode's factor matrix and has the property of uniqueness, so it is convenient to analyze the definite components of one tensor.

The vectors in the same mode were assembled to constitute each factor matrix of CP composition, such as  $U^{(i)} = [u_1^{(i)} u_2^{(i)} \dots u_R^{(i)}]$ , where  $1 < i < N$  and  $R$  is defined as the CP-rank which represents the minimum number of rank-one components. So the CP decomposition can be expressed as Eq. 5 which is similar with matrix SVD in format.

$$\mathcal{X} = \sum_{r=1}^R u_r^{(1)} \circ u_r^{(2)} \dots \circ u_r^{(N)} = \llbracket [U^{(1)}, U^{(2)}, \dots, U^{(N)}] \rrbracket \tag{5}$$

where  $\circ$  denotes the outer product and  $\llbracket \cdot \rrbracket$  denotes the Kruskal operator [19] as the sum of outer products by columns in a set of matrices.

The basic algorithm used for CP decomposition is ALS to establish the projection from tensor matricization of each mode to factors as Eq. 6.

$$X_{(n)} = U^{(n)} \left( U^{(N)} \odot \dots \odot U^{(n+1)} \odot U^{(n-1)} \dots \odot U^{(1)} \right)^T \tag{6}$$

where  $n = 1, 2, \dots, N$ . The core idea of ALS is to solve each factor matrix one by one with the other factors fixed until all factors converge to stability. Each factor matrix could be calculated by least-square as Eq. 7

$$U^{(n)} = X_{(n)} \left( \left( U^{(N)} \circledast \dots \circledast U^{(n+1)} \circledast U^{(n-1)} \dots \circledast U^{(1)} \right)^T \right)^\dagger \tag{7}$$

where  $\dagger$  denotes pseudo inverse of one matrix.

The ALS algorithm needs a predefined rank  $R$  and randomly initializes all factors in the first. However, it is difficult to choose a proper rank in advance generally. If the predefined rank  $R$  is less than actual rank  $R^*$ , the difference between reconstructed tensor and original tensor is unacceptable due to under fitting. If the predefined rank  $R$  is larger than the actual rank  $R^*$ , the reconstructed tensor over fits the noise which results in a large difference between the decomposed factors and the actual ones. Therefore, a noiseless tensor and corresponding proper rank are necessary for CP decomposition.

### 3 Methods

#### 3.1 Problem formulation

Accurate tensor decomposition is essential to analyze and interpret each component of data. Traditional CP decomposition by ALS assumes that the tensor is noiseless with the already known rank. However, both assumptions are usually dissatisfied in the practical application. In general, the measured tensor is defined as Eq. 8 in which the noisy tensor  $\mathcal{M}$  is the sum of original tensor  $\mathcal{X}$  and the independently identical distribution noise  $\mathcal{N}$  where  $\mathcal{X}$  can be decomposed by CP strictly and  $\mathcal{N}$  has the same size with  $\mathcal{X}$ .

$$\begin{aligned} \mathcal{M} &= \mathcal{X} + \mathcal{N} \\ \text{s.t. } \mathcal{X} &= \left[ \left[ U^{(1)}, U^{(2)}, \dots, U^{(N)} \right] \right] \\ n_{i_1, i_2, \dots, i_n} &\sim N(\hat{\mu}, \hat{\delta}^2) \end{aligned} \tag{8}$$

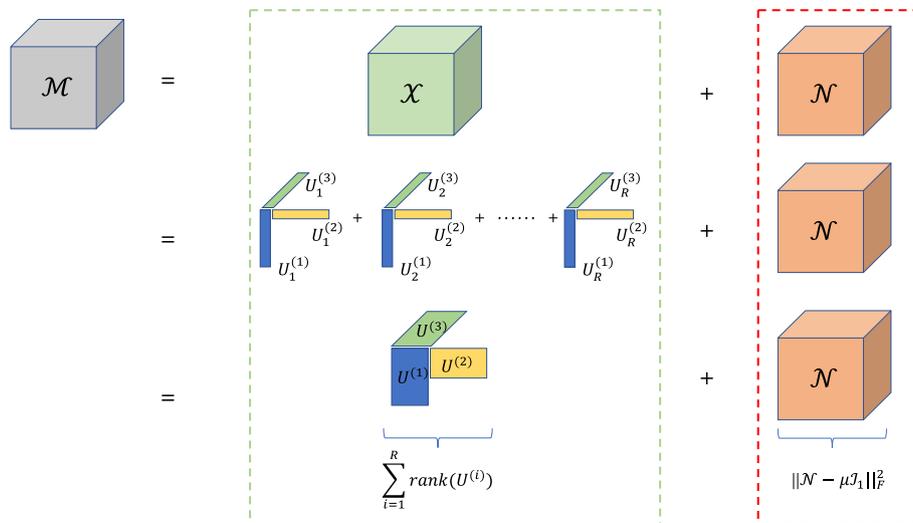
Noise and improper rank degenerate the decomposition accuracy and result in faulty diagnosis. If we could obtain the accurate factors  $U^{(n)}, n = 1, 2, \dots, N$  from the the noisy tensor  $\mathcal{M}$  as Fig. 1 illustrated, the original tensor can be reconstructed by the decomposed factors.

Therefore, our aim is to recover the original tensor from the noisy measurements without the prior knowledge about actual rank. If the original tensor has a low rank, this problem can be formulated to calculate the lowest-rank factors to minimize the noise variance. So the optimization objective function is to minimize the sum of factors' rank and the noise variance as Eq. 9.

$$\begin{aligned} \min_{U^{(i)}, \mu} & \sum_{i=1}^N \alpha_n \text{rank}(U^{(i)}) + \frac{\lambda}{2} \|\mathcal{N} - \mu \mathcal{I}_1\|_F^2 \\ \text{s.t. } \mathcal{M} &= \mathcal{X} + \mathcal{N}, \mathcal{X} = U^{(1)} \circledast U^{(2)} \circledast \dots \circledast U^{(N)} \end{aligned} \tag{9}$$

where both  $\alpha_n$  and  $\lambda$  in Eq. 9 denote the adaptive coefficients to balance the factors' rank and noise variance in the same order of magnitude and avoid any one term predominant in the optimization process.

The first term in Eq. 9 represents the sum of weighted rank and the second term is more generic about noise variance formulation. Assuming that the actual noise mean  $\hat{\mu}$  and variance  $\hat{\delta}^2$  are unknown and  $\mu$  is a variable representing noise's means distribution, the variance of  $\mathcal{N} - \mu \mathcal{I}_1$  can be expressed as  $\text{var}(\mathcal{N} - \mu \mathcal{I}_1) = \text{var}(\mathcal{N}) + \text{var}(\mu \mathcal{I}_1)$



**Fig. 1** The structure of the noisy tensor decomposition. The green box denotes the tensor that needs to be obtained and the red box denotes the noise interference

where  $\mathcal{I}_1$  denotes the tensor whose all elements is one. At the same time, the variance of a tensor can be calculated by Frobenius norm  $\|\cdot\|_F^2$ . If  $\mu \neq \hat{\mu}$ , the unbiased variance will be larger than  $\hat{\delta}^2$  as the simple proof below where  $N = I_1 \times I_2 \cdots I_N - 1$  denotes the element number of tensor:

$$\begin{aligned}
 \|\mathcal{N} - \mu \mathcal{I}_1\|_F^2 &= \|\mathcal{N} - \hat{\mu} \mathcal{I}_1 + \hat{\mu} \mathcal{I}_1 - \mu \mathcal{I}_1\|_F^2 \\
 &= \|\mathcal{N} - \hat{\mu} \mathcal{I}_1\|_F^2 + \|\hat{\mu} \mathcal{I}_1 - \mu \mathcal{I}_1\|_F^2 \\
 &\geq \|\mathcal{N} - \hat{\mu} \mathcal{I}_1\|_F^2 \\
 &= N \times \hat{\delta}^2
 \end{aligned} \tag{10}$$

### 3.2 Element-wise average ADMM

Because matrix rank minimization is a NP-hard problem, Eq. 9 can not be solved directly. Fortunately, the trace norm is proved to be a tightest convex relaxation form of rank minimization [20] and it is easy to be optimized by gradient descent. Therefore, the optimization objective function can be reformulated as follows:

$$\begin{aligned}
 \min_{U^{(i)}, \mu} \sum_{i=1}^N \alpha_n \|U^{(i)}\|_* + \frac{\lambda}{2} \|\mathcal{M} - U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(N)} - \mu \mathcal{I}_1\|_F^2 \\
 \text{s.t. } \mathcal{M} = \mathcal{X} + \mathcal{N}
 \end{aligned} \tag{11}$$

where  $U^{(n)} \in \mathbb{R}^{I_n \times R}$  for  $n = 1, 2, \dots, N$  and  $R$  is a positive integer which denotes the rank upper boundary of a tensor.

ADMM is widely used to solve the optimization programming no matter the problem is convex or nonconvex [21, 22]. In this paper, we also use ADMM algorithm to solve our optimization objective function. Due to the interaction between the trace norm and the

noise variance, some auxiliary variables  $A^{(i)}$  need to be employed in our model and we rewrite the Eq. 11 as follow:

$$\begin{aligned} \min_{U^{(i)}, \mu} \sum_{i=1}^N \alpha_n \|A^{(i)}\|_* + \frac{\lambda}{2} \|\mathcal{M} - U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(N)} - \mu \mathcal{I}_1\|_F^2 \\ \text{s.t. } \mathcal{M} = \mathcal{X} + \mathcal{N}, A^{(i)} = U^{(i)}, i = 1, 2, \dots, N \end{aligned} \tag{12}$$

In order to eliminate the constraints, the augmented Lagrangian function is used to transfer Eq. 12 as follows:

$$\begin{aligned} \mathcal{L}_\beta(A^{(1)}, \dots, A^{(N)}, U^{(1)}, \dots, U^{(N)}, \mu, Y^{(1)}, \dots, Y^{(N)}) \\ = \min_{U^{(i)}, \mu} \sum_{i=1}^N \alpha_n \|A^{(i)}\|_* + \frac{\lambda}{2} \|\mathcal{M} - U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(N)} - \mu \mathcal{I}_1\|_F^2 \\ + \sum_{i=1}^N \left( \langle Y^{(i)}, A^{(i)} - U^{(i)} \rangle + \frac{\beta}{2} \|A^{(i)} - U^{(i)}\|_F^2 \right) \end{aligned} \tag{13}$$

where  $Y^{(i)}$  is the Lagrange multipliers and  $\beta > 0$  is a penalty parameter. Then we use ADMM to minimize  $\mathcal{L}_\beta$  iteratively in the following sequence:

Firstly, for updating  $\{U^{(1)}, \dots, U^{(N)}\}$ , it is needed to unfold the tensor  $(\mathcal{M} - \mu \mathcal{I}_1)$  along each mode and then update related factor matrix one by one with the others fixed. So, the optimization is rewritten as follows:

$$\min_{U^{(n)}} \frac{\lambda}{2} \|U^{(n)} B^{(n)} - (\mathcal{M} - \mu_k \mathcal{I}_1)_{(n)}\|_F^2 + \frac{\beta}{2} \|U^{(n)} - A_k^{(n)} - Y_k^{(n)} / \beta\|_F^2 \tag{14}$$

where  $B^{(n)} = (U_k^{(N)} \dots \circ U_k^{(n+1)} \circ U_{k+1}^{(n-1)} \dots \circ U_{k+1}^{(1)})^T$  and  $U^{(n)}$  could be solved by Eq. 15:

$$U_{k+1}^{(n)} = \left( \lambda (\mathcal{M} - \mu_k \mathcal{I}_1)_{(n)} (B^{(n)})^T + \beta A_k^{(n)} + Y_k^{(n)} \right) \left( \lambda B^{(n)} (B^{(n)})^T + \beta I \right)^{-1} \tag{15}$$

Secondly, for updating  $\{A^{(1)}, \dots, A^{(N)}\}$ , the object is to minimize the sum of a matrix trace norm and its Frobenius norm as follows:

$$\min_{A^{(n)}} \alpha_n \|A^{(n)}\|_* + \frac{\beta}{2} \|A^{(n)} - U_{k+1}^{(n)} + Y_k^{(n)} / \beta\|_F^2 \tag{16}$$

and such format has a closed-form solution [23] which can be calculated as follows:

$$A_{k+1}^{(n)} = \text{SVT}_{\alpha_n / \beta} \left( U_{k+1}^{(n)} - Y_k^{(n)} / \beta \right) \tag{17}$$

where  $\text{SVT}_\delta(A) = U \text{diag}(\{(\sigma - \delta)_+\}) V^T$  is the singular value threshold (SVT) operator,  $U$  and  $V$  are from the matrix SVD results as equation  $A = U \text{diag}(\{\delta_i\}_{1 \leq i \leq r}) V^T$ , and the operator  $t_+$  denotes  $t_+ = \max(0, t)$ .

Then for updating  $\mu^{k+1}$ , the optimization object is rewritten as follows:

$$\min_{\mu} \left\| \mathcal{M} - U_{k+1}^{(1)} \odot \cdots \odot U_{k+1}^{(N)} - \mu \mathcal{I}_1 \right\|_F^2 \tag{18}$$

Since each element in the noise tensor is independent and identically distributed, the solution is the mean value of the element-wise difference between  $\mathcal{M}$  and  $U_{k+1}^{(1)} \odot \cdots \odot U_{k+1}^{(N)}$  as below, that is why our algorithm is called element-wise average ADMM:

$$\mu_{k+1} = \frac{1}{N+1} \sum_{i_1, \dots, i_n} \left( m_{i_1, \dots, i_n} - \left( U_{k+1}^{(1)} \circ \cdots \circ U_{k+1}^{(N)} \right)_{i_1, \dots, i_n} \right) \tag{19}$$

In the last,  $Y^{(n)}$  is updated only dependent on  $A_{k+1}^{(n)}$  and  $U_{k+1}^{(n)}$  as follows:

$$Y_{k+1}^{(n)} = Y_k^{(n)} + \beta \left( A_{k+1}^{(n)} - U_{k+1}^{(n)} \right) \tag{20}$$

The iteration calculation stops until any one of the convergence criteria meets. Here we set the convergence criteria as a tradeoff between the factor residual and the iterations. As for the factor residual, the primal residual measures the difference between the factor and the auxiliary factor in the same iteration with the Frobenius norm expression as  $\|U_k^{(n)} - A_k^{(n)}\|_F$ , and another residual measures the difference between the same auxiliary factor in two successive iterations with the expression as  $\|A_k^{(n)} - A_{k-1}^{(n)}\|_F$ . As for the iteration, we set the maximum number of iterations to 1000. In summary, the element-wise average ADMM is outlined in Algorithm 1.

---

**Algorithm 1** Element-wise Average ADMM

---

**Input:**  $\mathcal{M}, R, \lambda$  and  $\alpha_n$   
**Output:**  $U^{(i)} \ i = 1, 2 \cdots, N$   
**Initialize:**  $Y_0^{(n)} = A_0^{(n)} = 0, \mu = 0, U_0^{(n)} = \text{rand}(I_n, R), \text{iter}_{max} = 1000, \beta = 10^{-4}$  and  $\text{tol} = 10^{-5}$   
**while** not converged **do**  
    **for**  $n = 1: N$  **do**  
        update  $U_{k+1}^{(n)}$  by Eq. 15  
        update  $A_{k+1}^{(n)}$  by Eq. 17  
    **end for**  
    update  $\mu$  by Eq. 19  
    **for**  $n = 1: N$  **do**  
        update  $Y_{k+1}^{(n)}$  by Eq. 20  
    **end for**  
    **check if**  $\|U_k^{(n)} - A_k^{(n)}\|_F < \text{tol}$  **and**  $\|A_k^{(n)} - A_{k-1}^{(n)}\|_F < \text{tol}$  **or**  $k > \text{iter}_{max}$   
**end while**  
**Output:**  $U^{(i)} \ i = 1, 2 \cdots, N$

---

The Algorithm 1 also needs to initialize all variables in the first and the operator  $\text{rand}(\cdot, \cdot)$  is to randomly initialize the n-mode factors matrix with size  $I_n \times R$ .

**3.3 Extension to tensor completion**

Partial elements of the measured tensor may be missed due to the problem in the signal acquisition process, terminal failure or other malicious attack. Tensor completion is to infer the unobserved elements according to the partially observed elements. Multiple advanced tensor completion algorithms are summarized in [24] still without considering

the non-zero mean noise condition. We can extend our generic noise expression to tensor completion application and validate its effectiveness. Although only a part of elements are valid, these observed data still keep the character of low rank and the noise still fits independently identical distribution. Therefore, the optimization objective function remains unchanged except for adding one constraint as follows:

$$\min_{U^{(i)}, \mu} \sum_{i=1}^N \alpha_n \left\| (U^{(i)}) \right\|_* + \frac{\lambda}{2} \left\| \mathcal{M} - U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(N)} - \mu \mathcal{I}_1 \right\|_F^2 \tag{21}$$

s.t.  $\mathcal{M}_\Omega = (\mathcal{X} + \mathcal{N})_\Omega$

where  $\Omega$  denotes the set of observed elements index while the remains are missing.

All procedures for tensor completion are similar to EA-ADMM except for the step to update the noise mean  $\mu$  which could be calculated by the elements in  $\Omega$  as follows:

$$\mu_{k+1} = \frac{1}{|\Omega|} \sum_{i_1, \dots, i_n} (P_\Omega(m_{i_1, \dots, i_n} - (U_{k+1}^{(1)} \circ \dots \circ U_{k+1}^{(N)})_{i_1, \dots, i_n})) \tag{22}$$

where  $|\Omega|$  denotes the number of observed elements and  $P_\Omega(\cdot)$  keeps the elements in  $\Omega$  and zeros out the set. Finally the unobserved elements can be calculated by the outer production of factors expressed as  $P_{\Omega^c}(U^{(1)} \circ \dots \circ U^{(N)})$  where the set  $\Omega^c$  is complementary set of  $\Omega$

### 3.4 Computational complexity estimation

The update process of each variable has definite expression as Eqs. 15, 17 and 19 in a iteration loop, which mainly consists of simple matrix operations. Therefore, the computational complexity estimation can be straightforward derived. We calculated the  $i$ -th factor  $U_k^{(i)}$ , auxiliary variable  $A_k^{(i)}$  and noise mean  $\mu_k$  respectively in the  $k$ -th iteration as an example and the other factors estimations  $U_k^{(j)}, j \neq i$  are the analogous.

- $U_k^{(i)}$  update The factor’s computational complexity involves the term  $(\lambda(\mathcal{M} - \mu_k \mathcal{I}_1)_{(n)}(B^{(n)})^T + \beta A_k^{(n)} + Y_k^{(n)})$ , term  $(\lambda B^{(n)}(B^{(n)})^T + \beta I)^{-1}$  and these two terms multiplication. The first term’s computational complexity is  $O(I_i R(1 + \prod_{j \neq i} I_j^2))$ . The second term needs to be divided in the sub-term  $(\lambda B^{(n)}(B^{(n)})^T + \beta I)$  whose complexity is  $O(\sum_{j \neq i} I_j R^2)$  and the inversion operator whose complexity is  $O(R^3)$ . The final matrix multiplication’s complexity is  $O(I_i R^2)$
- $A_k^{(i)}$  update The auxiliary variable needs to be decomposed by SVD, therefore this computational complexity can be represented by  $O(I_i R^2)$  where the matrix size is  $I_n \times R$  and  $R$  is much less than  $I_n$ .
- $\mu_k$  update The noise mean variable only involves the element average operation in the tensor. The maximum complexity can be represented by  $O(\prod_{n=1}^N I_n)$

All the computational complexity calculation assumes that the tensor is dense and each arithmetic with individual elements has complexity  $O(1)$ . If the tensor is sparse, the complexity decreases by the exploiting sparsity and the structure of Khatri-Rao production [25].

Therefore, the total complexity in one iteration is the sum of all variables' complexity and the predominant complexity is  $O((N + 1)R \prod_{n=1}^N I_n)$  if the tensor is low rank. For comparison, the computational complexity of ALS [26] is represented as Eq. 23. It is obvious that these two algorithms have similar predominant complexity term (the second term) for dense tensor due to low rank property and the same matrix operators, such as Khatri-Rao production, SVD and matrix inversion.

$$\sum_{i=1}^N \left( \prod_{j \neq i} I_j \right) (7R^2 + R) + 3R \prod_{n=1}^N I_n + \sum_{i=1}^N I_i (R^2 + R) + 11R^3 \tag{23}$$

#### 4 Results and discussion

In this section, we conducted the comparative experiments to evaluate the effectiveness of EA-ADMM involving in tensor recovery, tensor completion from the noisy tensor. The performances between our algorithm and traditional CP decomposition by ALS are compared by the synthesized data. A noiseless tensor data with size  $20 \times 40 \times 30$  is composed by the uniform distribution  $U(0, 1)$  factors with a definite rank, then we add the independent and identical distribution noise to the noiseless tensor.

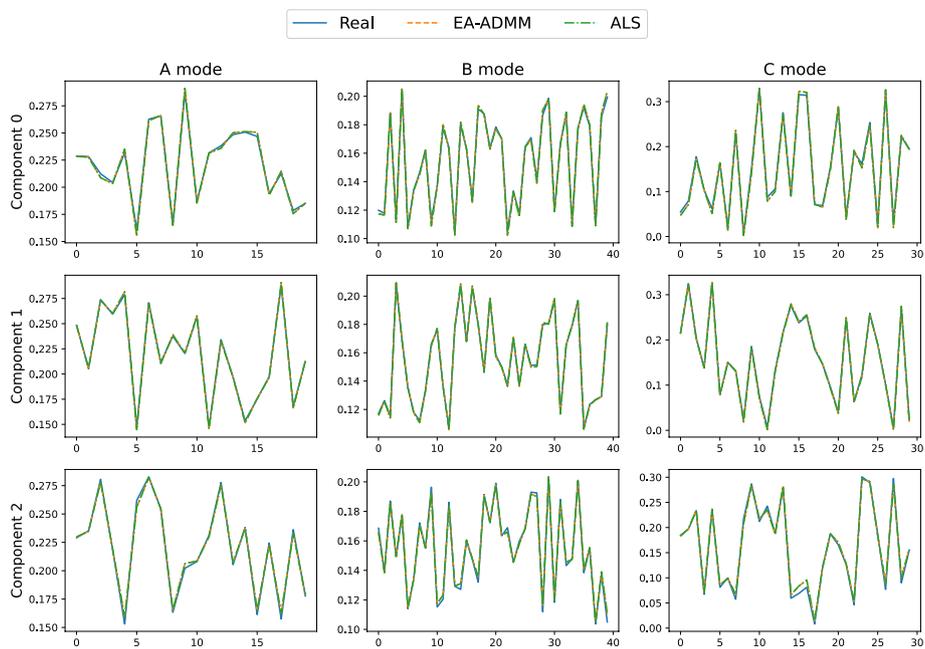
Here the metric we consider to evaluate the performance for tensor recovery and completion is average tucker congruence coefficient [27] (hereinafter referred to as similar coefficient) which measures the similarity of each component between decomposed factors with the actual factors and it is calculated by averaging the unique maximum combinations of components value. Since the tensor can be generated by the factors as Eq. 5 and the tensor decomposition is mainly to analyze the internal correlation in data, the factor similarities are effective metrics to measure the decomposition accuracy. Furthermore, the relative root mean square error (rRMSE) of the reconstructed tensor  $\hat{\mathcal{X}}$  as another metric is used to compare the accuracy in tensor completion as follows:

$$\text{rRMSE} := \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|_F}{\|\mathcal{X}\|_F} \tag{24}$$

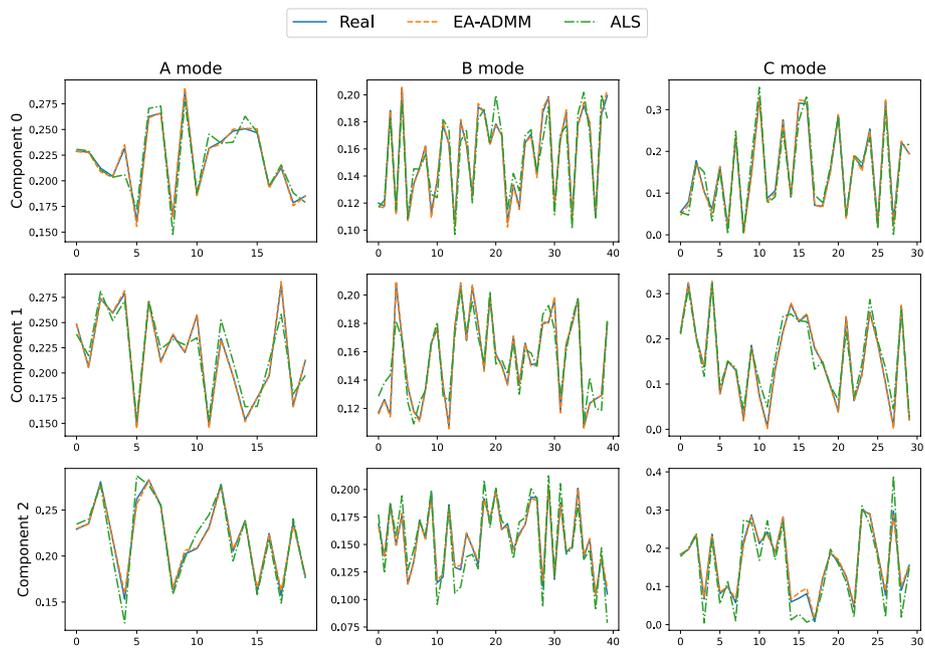
##### 4.1 Tensor recovery

For tensor recovery, we compare the traditional ALS algorithm and the proposed EA-ADMM algorithm to decompose the tensors with different rank in various SNR. If the rank of tensor is given, the predefined rank of both two algorithm are set to the actual rank, the SNR is 20dB, the parameters  $\alpha_n$  in Eq. 12 are equal to  $1/N$  and  $\lambda$  is set to 5, moreover the iterative number is 1000 and tolerance threshold is the same  $10^{-5}$  as our algorithm. A rank-3 tensor's decomposed factors by these two algorithms are illustrated in Fig. 2. As we can see, all components in each tensor mode fit the actual value perfectly and the similar coefficient of both algorithms are above 99.9%.

In practical, the actual rank is usually unknown, so it is common to set the predefined rank to the upper boundary which is described in [28] and the other parameters keep same as before. In such more general situation, the decomposed results are illustrated



**Fig. 2** EA-ADMM tensor recovery performance compared with ALS where SNR is 20dB and pre-define rank is equal to actual rank



**Fig. 3** EA-ADMM tensor recovery performance compared with ALS where SNR is 20dB and pre-define rank is larger than actual rank

in Fig. 3 which indicates different performances of these two algorithms. The similar coefficient of the EA-ADMM is still above 99.4%, but the ALS decreases to 90.1%. If the predefined rank is larger than actual rank, the ALS over fits the noise and could not decompose the accurate factors, however due to the constraint on the trace norm and

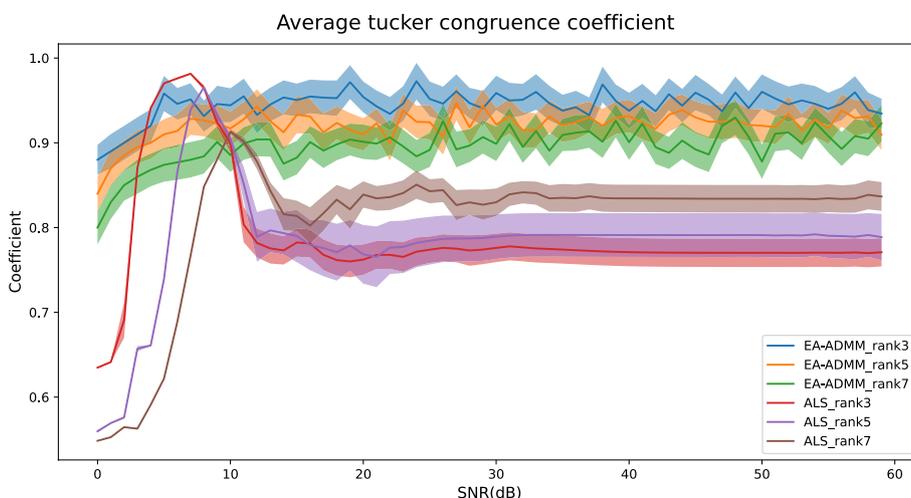
noise’s variance, the EA-ADMM still has the capability to fit the actual components perfectly which means that it could recover the original tensor from the decomposed factors in the noisy situation and thus the tensor’s actual rank can be inferred by the factors.

For further verifying the noise-tolerant capability, we vary the SNR from 0dB to 60dB and compare these two algorithms with different rank tensors. Firstly, three set of different SNR noisy tensors with rank 3, 5 and 7 are generated respectively, then the predefined ranks are selected randomly in the range larger than the actual rank. We repeated the experiments 50 times and the mean and variance of the similar coefficient are illustrated in Fig. 4. It shows that no matter whatever the actual rank is, the EA-ADMM performance keeps more stable with the average coefficient of above 90.5% if the predefined rank is larger than rank, while the performance of ALS fluctuate seriously in different SNR. However, there exists an interesting phenomenon that the ALS in some SNR ranges has high average similarity coefficient with small variance which shows more accurate and stable decomposition performance than the other SNR ranges. Such interesting range seems to be dependent on the actual rank and it is important for ALS where it is less sensitive to the predefined rank.

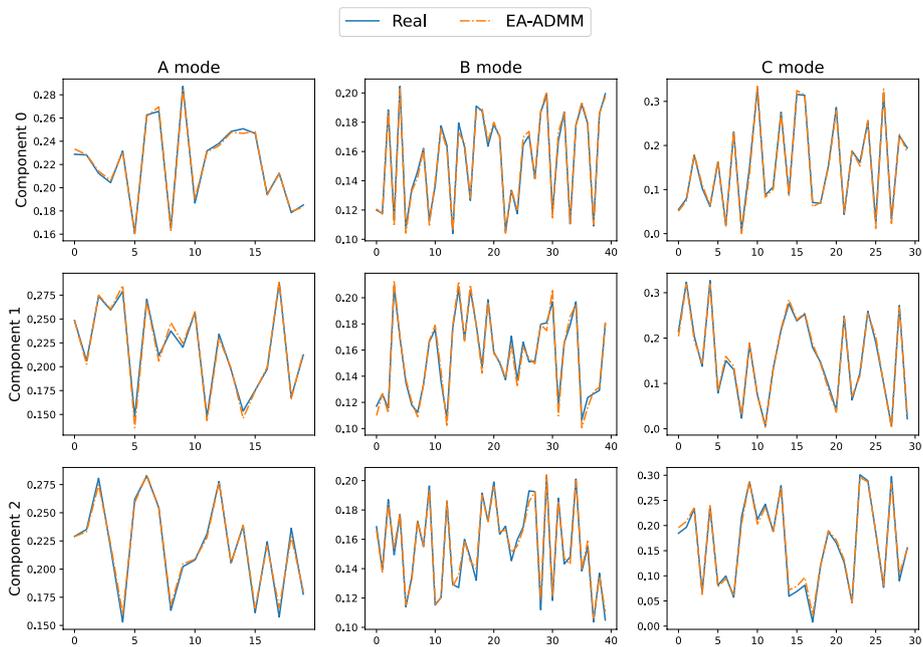
### 4.2 Tensor completion

Besides noise interference, there are often cases of tensor completion due to the data loss caused by equipment failure in the process of data acquisition. Therefore, we verify whether the EA-ADMM algorithm has the capability of tensor completion in the first. The elements in the tensor are randomly selected and set to 0 as the missing data, and both two metrics above mentioned are used to evaluate the performance of tensor completion. As seen in Fig. 5, 10% elements of the tensor were masked, the mean of noise is zero and SNR is set to 20dB, then the similar coefficient is 99.6% and RSE is 0.016, so our algorithm can complete the missing data with noise interference.

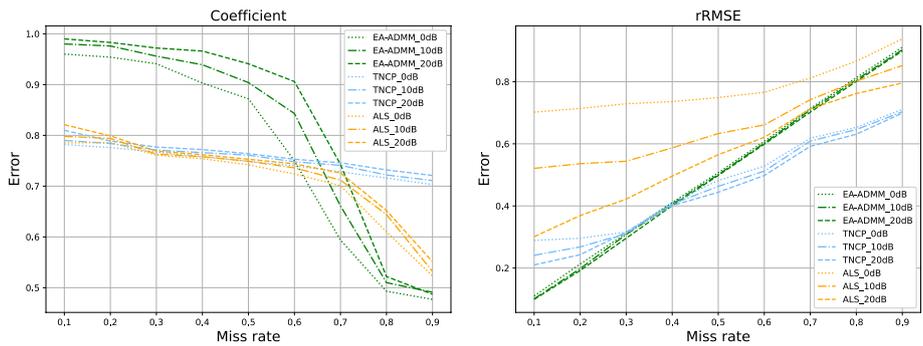
We further conducted comparative experiments to evaluate the performance of tensor completion when the missing rate increases from 0.1 to 0.9 in the 20dB, 10dB and 0dB SNR respectively. The predefined rank is larger than actual rank and the optimal



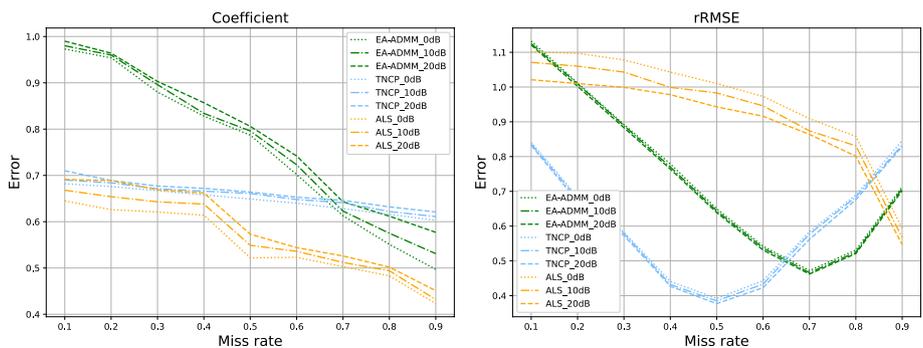
**Fig. 4** EA-ADMM noise-tolerance performance compared with ALS where SNR is from 0 to 60dB



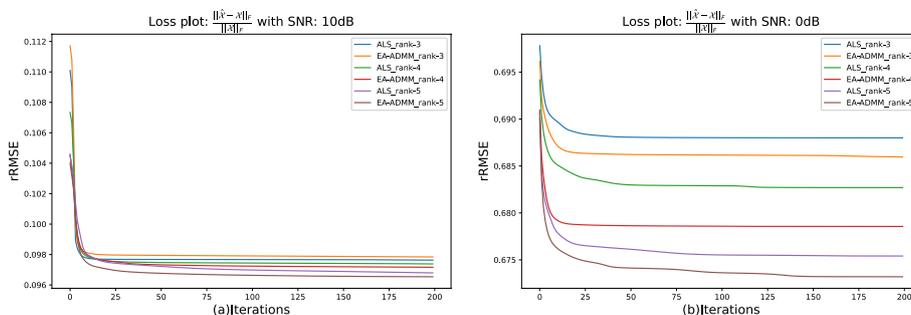
**Fig. 5** Tensor completion performance when the missing rate as 0.1 and  $\hat{\mu} = 0$



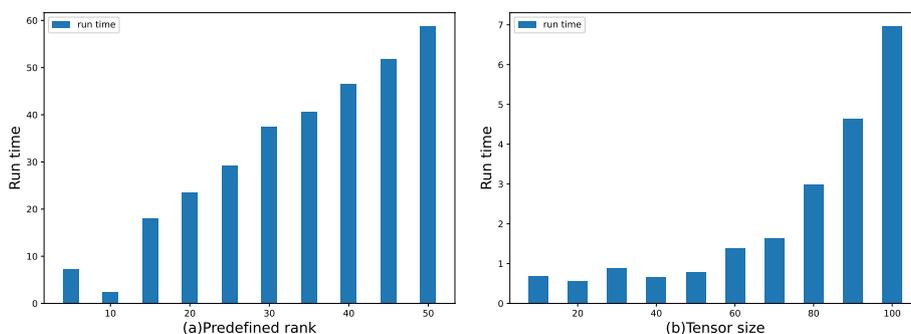
**Fig. 6** Tensor completion performance when the missing rate varies from 0.1 to 0.9 and  $\hat{\mu} = 0$



**Fig. 7** Tensor completion performance when the missing rate varies from 0.1 to 0.9 and  $\hat{\mu} \neq 0$



**Fig. 8** Tensor recovery performance with various SNRs



**Fig. 9** CPU run time with different predefined ranks and tensor sizes

parameters are selected by the Grid Search. The traditional ALS and another algorithm called Trace Norm CP (TNCP) in [16] are contrastive to our EA-ADMM. If the mean of noise is equal to zero as Fig. 6, the EA-ADMM has higher similar coefficient and lower rRMSE value than the other algorithms when the missing rate is below 0.5, and then there is a rapid decline in performance. If the mean of noise is non-zero as Fig. 7, the similar coefficient of EA-ADMM decreases slightly when the missing rate is less than 0.5, and the rRMSE reaches the minimum value when the missing rate is 0.7. On the whole, the non-zero mean noise will degenerate the performance of tensor completion for all algorithms. The TNCP could achieve lower rRMSE, but could not decompose to the similar factors. The ALS also has the capability of tensor completion somehow, but the performance is relatively poor. However, if the the missing rate is less than 0.5, the proposed EA-ADMM algorithm could obtain both more similar factors and lower rRMSE.

### 4.3 Complexity comparison

We compare our algorithm’s computational complexity with traditional algorithm ALS in high and low SNR. The tensor size is  $20 \times 40 \times 30$  and the predefined rank is larger than actual rank. When the SNR is 10dB in Fig. 8a, our algorithm needs a similar iteration number with ALS to achieve a slightly lower relative reconstruction error. This result shows that if the noise is negligible, these two algorithms have the same computational complexity and the over-estimated rank for traditional ALS only affects the factor similarity. When the SNR is 0dB in Fig. 8b, the convergence speed of our algorithm

is higher than ALS with a better recovery performance. In such condition, the noise is non-negligible, so the over-estimated rank for ALS results in a much higher relative reconstruction error due to the noise overfitting. However, our algorithm is less sensitive to the over-estimated rank and the noise.

Besides, we record our algorithm's run time with different tensor sizes with various predefined ranks. The CPU run time as the rough metric could evaluate the computational complexity. When the rank varies, the tensor size is fixed as  $100 \times 100 \times 100$  and the result is illustrated in Fig. 9a. The run time is linear with the rank. When the tensor size varies, the tensor keeps cube and its rank is fixed as 5. The run time increases exponentially with the increase of tensor size as Fig. 9 b. All experimental results are consistent with theoretical analysis in subsect. 3.4.

## 5 Conclusion

Tensor is an effective representation to model the multidimensional structure signals in communications and radar system and then tensor decomposition, especially the CP decomposition which has unique decomposed factors, is a powerful method to analysis the intrinsic linear correlation in the tensor data. Tradition CP decomposition by ALS is based on the assumption that the tensor rank is known without noise interference. However, both of the assumptions are not commonly satisfied in practical applications that degenerates the decomposition performance. In this paper, we proposed an element-wise average ADMM algorithm to recover the original tensor from noisy measurements. Our algorithm is to minimize the noise variance on the constraint of all factors' rank sum. More than tensor recovery, our algorithm can be easily extended to realize tensor completion. The experiments involving in tensor recovery and tensor completion in various SNR indicate that our algorithm is robust to decompose the tensor into accurate factors. However, our algorithm is still time-costing due to high-dimensional matrix operations. In the future, we would combine the tensor decomposition with deep learning to boost the update speed by simultaneously back propagation and the interesting phenomenon mentioned in subsect. 4.1 also deserves to be exploited.

### Abbreviations

EA-ADMM	Element-wise average alternative direction method of multipliers
SNR	Signal noise ratio
ALS	Alternative least square
SVD	Singular value decomposition
SVT	Singular value threshold
TNCP	Trace norm CANDECOMP/PARAFAC decomposition
RMSE	Root mean square error

### Acknowledgements

Not applicable.

### Author Contributions

Both authors have contributed toward this work as well as in compilation of this manuscript. Both authors read and approved the final manuscript.

### Funding

This paper is supported in part by National Natural Science Foundation of China with Grant 62171063 and in part by the Manufacturing High Quality Development Fund Project in China under Grant TC210H03D.

### Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

## Declarations

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

We agree to the publication of the paper.

### Competing interests

The authors declare that they have no competing interests.

Received: 19 March 2022 Accepted: 28 September 2022

Published online: 09 October 2022

## References

1. L. Wan, R. Liu, L. Sun, H. Nie, X. Wang, Uav swarm based radar signal sorting via multi-source data fusion: A deep transfer learning framework. *Inf. Fusion* **78**, 90–101 (2022). <https://doi.org/10.1016/j.inffus.2021.09.007>
2. Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, R.S. Blum, Low-rank tensor decomposition-aided channel estimation for millimeter wave mimo-ofdm systems. *IEEE J. Sel. Areas Commun.* **35**(7), 1524–1538 (2017). <https://doi.org/10.1109/JSAC.2017.2699338>
3. G. Yue, Z. Sun, J. Fan, Ag-Irtr: An adaptive and generic low-rank tensor-based recovery for iiot network traffic factors denoising. *IEEE Access* **10**, 69839–69850 (2022). <https://doi.org/10.1109/ACCESS.2022.3187112>
4. A.P. Liavas, N.D. Sidiropoulos, Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. *IEEE Trans. Signal Process.* **63**(20), 5450–5463 (2015). <https://doi.org/10.1109/TSP.2015.2454476>
5. M. Roald, C. Schenker, J.E. Cohen, E. Acar, PARAFAC2 AO-ADMM: Constraints in all modes. *arXiv* (2021). <https://doi.org/10.48550/ARXIV.2102.02087>. <https://arxiv.org/abs/2102.02087>
6. R.A. Harshman, Foundations of the parafac procedure : Models and conditions for an “explanatory” multimodal factor analysis. *Ucla Working Papers in Phonetics* **16** (1970)
7. A. Streit, G. H. A. Santos, R. M. M. Leão, E. de Souza e Silva, D. Menasché, D. Towsley, Network anomaly detection based on tensor decomposition. *Comput. Netw.* **200**, 108503 (2021). <https://doi.org/10.1016/j.comnet.2021.108503>
8. N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* **65**(13), 3551–3582 (2017). <https://doi.org/10.1109/TSP.2017.2690524>
9. V. Ranjbar, M. Salehi, P. Jandaghi, M. Jalili, Qanet: Tensor decomposition approach for query-based anomaly detection in heterogeneous information networks. *IEEE Trans. Knowl. Data Eng.* **31**(11), 2178–2189 (2019). <https://doi.org/10.1109/TKDE.2018.2873391>
10. Y. Ouyang, K. Xie, X. Wang, J. Wen, G. Zhang, Lightweight trilinear pooling based tensor completion for network traffic monitoring. in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 2128–2137 (2022). <https://doi.org/10.1109/INFOCOM48880.2022.9796873>
11. M.E. Timmerman, H. Kiers, Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. *Br. J. Math. Stat. Psychol.* **53**(1), 1–16 (2011)
12. Y. Yu, Z. Li, X. Liu, K. Hirota, X. Chen, T. Fernando, H.H.C. Lu, A nested tensor product model transformation. *IEEE Trans. Fuzzy Syst.* **27**(1), 1–15 (2019). <https://doi.org/10.1109/TFUZZ.2018.2851575>
13. G. Tsitsikas, E.E. Papalexakis, The core consistency of a compressed tensor. in *2019 IEEE Data Science Workshop (DSW)* (2019), pp. 1–5. <https://doi.org/10.1109/DSW.2019.8755593>
14. J. Zhao, L. Chen, W. Pedrycz, W. Wang, Variational inference-based automatic relevance determination kernel for embedded feature selection of noisy industrial data. *IEEE Trans. Ind. Electron.* **66**(1), 416–428 (2019). <https://doi.org/10.1109/TIE.2018.2815997>
15. S. Pouryazdian, S. Beheshti, S. Krishnan, Candecomp/parafac model order selection based on reconstruction error in the presence of kronecker structured colored noise. *Digit. Signal Process.* **48**(C), 12–26 (2016)
16. Y. Liu, F. Shang, L. Jiao, J. Cheng, H. Cheng, Trace norm regularized candecomp/parafac decomposition with missing data. *IEEE Trans. Cybern.* **45**(11), 2437–2448 (2015). <https://doi.org/10.1109/TCYB.2014.2374695>
17. Y. Xu, W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to non-negative tensor factorization and completion. *SIAM J. Imaging Ence* **6**(3), 1758–1789 (2015)
18. L. Tucker, Some mathematical notes on three-mode factor analysis. *Psychometrika* **31**(3), 279–311 (1966)
19. J.B. Kruskal, Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.* **18**(2), 95–138 (1977)
20. M. Fazel, matrix rank minimization with applications. PhD thesis, Dissertation Abstracts International, vol. 63-04, Section: B. (Adviser: Stephen P. Boyd, 2002), pp. 1981
21. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2010)
22. L. Wan, K. Liu, W. Zhang, Deep learning-aided off-grid channel estimation for millimeter wave cellular systems. in *IEEE Transactions on Wireless Communications*, vol. 1 (2021). <https://doi.org/10.1109/TWC.2021.3120926>
23. W. Li, J. Hu, C. Chen, On accelerated singular value thresholding algorithm for matrix completion. *Appl. Math.* **5**(21), 3445–3451 (2014)
24. Q. Song, H. Ge, J. Caverlee, X. Hu, Tensor completion algorithms in big data analytics. *ACM Trans. Knowl. Discov. Data.* (2019). <https://doi.org/10.1145/3278607>
25. T.G. Kolda, J. Sun, Scalable tensor decompositions for multi-aspect data mining, in *2008 Eighth IEEE International Conference on Data Mining* (2008), pp. 363–372. <https://doi.org/10.1109/ICDM.2008.89>

26. P. Comon, X. Luciani, A.L.F. de Almeida, Tensor decompositions, alternating least squares and other tales. *J. Chemom.* **23**(7–8), 393–405 (2009) <https://doi.org/10.1002/cem.1236>. [analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1236](https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1236)
27. U. Lorenzo-Seva, J.M. Ten Berge, Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology* **2**(2), 57–64 (2006)
28. B. Alexeev, M.A. Forbes, J. Tsimerman, Tensor rank: Some lower and upper bounds. in *2011 IEEE 26th Annual Conference on Computational Complexity* (2011). <https://doi.org/10.1109/cc.2011.28>

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---