Error Preserving Correction: A Method for CP Decomposition at a Target Error Bound

Anh-Huy Phan, *Member, IEEE*, Petr Tichavský, *Senior Member, IEEE*, and Andrzej Cichocki, *Fellow, IEEE*

***Abstract*—In CANDECOMP/PARAFAC tensor decomposition, degeneracy often occurs in some difficult scenarios, especially, when the rank exceeds the tensor dimension, or when the loading components are highly collinear in several or all modes, or when CPD does not have an optimal solution. In such cases, norms of some rank-1 tensors become significantly large and cancel each other. This makes algorithms getting stuck in local minima while running a huge number of iterations does not improve the decomposition. In this paper, we propose an error preservation correction method to deal with such problem. Our aim is to seek an alternative tensor, which preserves the approximation error, but norms of rank-1 tensor components of the new tensor are minimized. Alternating and all-at-once correction algorithms have been developed for the problem. In addition, we propose a novel CPD with a bound constraint on the norm of the rank-one tensors. The method can be useful for decomposing tensors that cannot be performed by traditional algorithms. Finally, we demonstrate an application of the proposed method in image denoising and decomposition of the weight tensors in convolutional neural networks.**

***Index Terms*—Canonical polyadic decomposition (CPD), bounded CPD, degeneracy, PARAFAC, tensor decomposition for ill conditioned problems.**

I. Introduction

In THIS paper, we consider the CANDECOMP/PARAFAC tensor decomposition, which approximates a tensor $\mathcal{Y}$ by a sum of rank-1 tensors in the form of

$$\mathcal{Y} \approx \hat{\mathcal{Y}} = \sum_{r=1}^{R} \eta_r \, \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}, \tag{1}$$

where $\mathbf{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_R^{(n)}]$ are factor matrices of size $I_n \times R$. The tensor $\mathcal{Y}$ is of size $I_1 \times I_2 \times \cdots \times I_N$, and its approximated tensor, $\hat{\mathcal{Y}}$, is of rank-R. This decomposition has found numerous applications in identification of independent components, signals retrieval in CDMA telecommunications, extraction of hidden components from neural data, image completion and various tracking scenarios [1], [2].

When the loading components $\boldsymbol{u}_r^{(n)}$ are assumed to be unit-length vectors, the weight, η_r, represents the Frobenius norm of the r-th rank-one tensor

$$\begin{aligned}\|\eta_r \, \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}\|_F^2 &= \eta_r^2 \, \|\boldsymbol{u}_r^{(1)}\|^2 \|\boldsymbol{u}_r^{(2)}\|^2 \cdots \|\boldsymbol{u}_r^{(N)}\|^2 \\ &= \eta_r^2.\end{aligned}$$

In some difficult decomposition scenarios, the norms of some rank-1 terms become significantly large and cancel each other. This is often observed when the rank exceeds the tensor dimension, or when the loading components are highly collinear in several or all modes (swamps) [3]. Moreover, it may happen that the CP decomposition does not have an optimal solution [4], [5], because the tensor can be arbitrarily well approximated by tensors of lower ranks. Recently, the CPD of the parameters in the convolutional and fully connected layers has shown a novel application in acceleration of the inference process of convolutional neural networks [6], [7]. The tensors in practice do not have balance dimensions, but often need the approximation with relatively high rank, e.g., 200. The dimensions corresponding to the filter sizes are often small, while the other dimensions, representing the number of inputs and outputs of the layers are often high. Our observation is that decomposition of such tensors using the alternating or nonlinear least squares algorithm often encounters degeneracy. Many rank-1 tensor components have very high Frobenius norms, while some other ones are with low intensity. Such degeneracy causes difficulty with fine-tuning the network, selection of a good set of parameters and stability in the entire network.

The degeneracy phenomenon is widely reported in the literature, e.g., in [1], [3], [8]–[18]. Some efforts have been made to improve stability and convergence for such cases [11], [19]. For example, additional constraints can be imposed on the factor loadings, e.g., column-wise orthogonality [4], [17]–[19], positivity or nonnegativity [10], [20]. An alternative approach is to improve convergence of CP decomposition techniques, e.g., the Alternating Least Squares (ALS), through a regularization. For example, a Tikhonov regularization alters the optimization problem to

$$\min \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 + \frac{\mu}{2} \sum_n \|\mathbf{U}^{(n)}\|_F^2. \tag{2}$$

Manuscript received July 24, 2018; revised November 12, 2018; accepted December 4, 2018. Date of publication December 17, 2018; date of current version January 14, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cedric Fevotte. The work of A.-H. Phan and A. Cichocki was supported in part by the Ministry of Education and Science of the Russian Federation under Grant 14.756.31.0001. The work of P. Tichavský was supported by the Czech Science Foundation through project No. 17-00902S. *(Corresponding author: Anh-Huy Phan.)*

A.-H. Phan is with the Skolkovo Institute of Science and Technology (Skoltech), Moscow 121205, Russia, and also with the Tokyo University of Agriculture and Technology, Tokyo 183-8538, Japan (e-mail: a.phan@skoltech.ru).

P. Tichavský is with the Czech Academy of Sciences, Institute of Information Theory and Automation, Prague CZ-182 08, Czech Republic (e-mail: tichavsk@utia.cas.cz).

A. Cichocki is with the Skolkovo Institute of Science and Technology (Skoltech), Moscow 121205, Russia, with the Systems Research Institute, Polish Academy of Science, Warsav 01-447, Poland, with the Hangzhou Dianzu University (HDU), Hangzhou 310005, China, and also with the Tokyo University of Agriculture and Technology, Tokyo 183-8538, Japan (e-mail: a.cichocki@skoltech.ru).

Digital Object Identifier 10.1109/TSP.2018.2887192

In (2), columns of the factor matrices $\mathbf{U}^{(n)}$ are not normalized, but the scaling η_r are equally distributed in them. The penalty parameter μ is adaptively adjusted and should converge to zero. Another possibility is to employ an iterated Tikhonov regularization [21] represented in a sequence optimization

$$\min \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}_{[i]}\|_F^2 + \frac{\mu}{2}\sum_n \|\mathbf{U}_{[i]}^{(n)} - \mathbf{U}_{[i-1]}^{(n)}\|_F^2, \qquad (3)$$

where $\mathbf{U}_{[i]}^{(n)}$ is an estimate of $\mathbf{U}^{(n)}$ in iteration i and $\hat{\mathcal{Y}}_{[i]}$ is a tensor represented by $\{\mathbf{U}_{[i]}^{(n)}\}$. The parameter μ controls the convergence speed. In practice, an update with a slower convergence is often more stable.

In some applications, we seek an exact CP representation for a data. Decomposition of tensors corresponding to the matrix multiplication is an example. The task is to find a minimum number of scalar multiplications to compute a product of two matrices of given sizes. This is one of the main challenging problems of the theory of complexity. In [22], we minimise the first term in (2) and constrain the second term to a constant.

$$\min \quad \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \quad \text{s.t.} \quad \sum_n \|\mathbf{U}^{(n)}\|_F^2 \le c\,.$$

In this way, it is possible to find an exact decomposition of the matrix multiplication tensors for certain matrix sizes and avoid convergence to singular "solutions" where the norm of some factor matrices converges to infinity.

Although the above-mentioned methods may help in certain situations, we propose a more direct approach to the problem of the diverging components. Different from the existing algorithms for this kind of tensor decomposition, our aim is to correct the rank-1 tensors if their norm is observed to be relatively high during the tensor approximation process. More specifically, we seek a new tensor, $\hat{\mathcal{Y}}$, whose norms of rank-1 tensor components are minimal or relatively small, while it is still possible to explain $\mathcal{Y}$ at the current level of approximation error. Continuing the decomposition with a new tensor which has a lower norm prevents CP algorithms from degeneracy and thereby improves their convergence. This can be achieved by solving the following constrained CP tensor approximation

$$\begin{aligned} \min \quad & f(\boldsymbol{\theta}) = \|\boldsymbol{\eta}\|_2^2 = \sum_{r=1}^{R} \eta_r^2 \\ \text{s.t.} \quad & c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}(\boldsymbol{\theta})\|_F^2 \le \delta^2, \end{aligned} \qquad (4)$$

where $\boldsymbol{\theta}$ represents a vector of all model parameters and $\hat{\mathcal{Y}}(\boldsymbol{\theta})$ or $\hat{\mathcal{Y}}$ represents the estimated tensor of $\mathcal{Y}$ constructed from θ. We call this the Error Preserving Correction (EPC) method.

In Section II, we derive algorithms for the above constrained nonlinear optimization problem: an alternating EPC algorithm and another one based on the Sequential Quadratic Programming (SQP) method to update all the parameters at a time. In the alternating algorithm, we reformulate the optimization in (4) as linear regression sub-problems with a bound constraint for the factor matrices, which in turn can be solved in closed-form through the Spherical Constrained Quadratic Programming (SCQP). For the SQP algorithm, we derive a fast inverse of the Hessian matrix.

TABLE I
NOTATION AND ABBREVIATIONS USED IN THE PAPER

$\mathbf{Y}_{(n)}$	mode-n matricization of $\mathcal{Y}$
$\mathbf{U}^{(n)}$	factor matrices
$\odot$, $\otimes$, $\circ$	Khatri-Rao, Kronecker and outer products
$\circledast$, $\oslash$	Hadamard product and division
$\bigodot_{k\neq n}\mathbf{U}^{(k)}$	Khatri-Rao product of all but one factor matrices
$\boldsymbol{\Gamma}_{-n} = \circledast_{k\neq n}(\mathbf{U}^{(k)T}\mathbf{U}^{(k)})$	Hadamard product of all but one matrices
$\text{vec}(\boldsymbol{\Gamma})$	vectorization
$\text{dvec}(\boldsymbol{\Gamma}) = \text{diag}(\text{vec}(\boldsymbol{\Gamma}))$	
$\mathbf{P}_{R,R}$	a permutation matrix maps $\text{vec}(\mathbf{X}_{R\times R}) = \mathbf{P}_{R,R}\,\text{vec}\left(\mathbf{X}_{R\times R}^T\right)$
SCQP	Spherical Constrained Quadratic Programming
EPC	Error Preserving Correction
SQP	Sequential Quadratic Programming

Other optimization methods can be applied to solve the above optimization problem. For large-scale data, we can apply the sketching or randomized sampling method [23]–[25]. A random selection of the sketches (parts of data used in optimization) is reported to have a positive influence on the convergence.

In the second part of the paper, together with the EPC for CPD, we propose a novel CP decomposition with a constraint on the norm of rank-1 tensors

$$\min \quad \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{\eta}\|_2^2 \le \epsilon^2\,. \qquad (5)$$

Note that the optimization problem (5) is dual to that in (4) and vice versa. This method is similar but not identical to the one in [22] with a bound on the sum of squared Frobenius norm of the factor matrices. A novel ALS algorithm and an SQP algorithm are then derived for the bounded norm CPD.

We also present relations between the alternating EPC and the ordinary ALS algorithm, between the new ALS for CPD with a bound constraint and the ALS with the Tikhonov regularization given in (2).

In Section IV, we present examples of utilization of the proposed algorithms and methods in decomposing artificially constructed tensors, tensors corresponding to the matrix multiplication, a tensor of real-world TV rating data [8], a fourth-order tensor for time-frequency representation of Event-related EEG [26], a weight tensor in the Alexnet convolutional neural network, and an example for image denoising.

II. Error Preserving Correction Algorithms

For convenience, we summarize notations and abbreviations in Table I. We note that the constraint function in the optimization (4) is nonlinear in all the factor matrices, but linear in parameters in a single factor matrix, or in non-overlapping partitions of different factor matrices [27], [28]. A simple approach to handle this kind of constrained nonlinear optimization is to rewrite the objective function and especially the constraint function in a linear form. This can be achieved using the alternating update scheme or the Sequential Quadratic Programming method [29], [30].

A. The Alternating Correction Method

In this section, we present an application of the linear regression in (39) in a tensor decomposition.

At each iteration, we seek a new estimate of the factor matrix $\mathbf{U}^{(n)}$ which reduces the objective function, while preserving the approximation error. Note that by absorbing $\boldsymbol{\eta}$ into the factor matrix $\mathbf{U}^{(n)}$ to give $\mathbf{U}_\eta^{(n)} = \mathbf{U}^{(n)}\,\mathrm{diag}(\boldsymbol{\eta})$, while keeping the other factor matrices $\mathbf{U}^{(k)}$ fixed, $k \neq n$, the objective function becomes

$$\|\boldsymbol{\eta}\|^2 = \|\mathbf{U}^{(n)}\,\mathrm{diag}(\boldsymbol{\eta})\|_F^2 = \|\mathbf{U}_\eta^{(n)}\|_F^2\,.$$

The constraint is rewritten for the factor matrix $\mathbf{U}_\eta^{(n)}$ as

$$\begin{aligned}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 &= \|\mathbf{Y}_{(n)} - \mathbf{U}_\eta^{(n)}\mathbf{T}_n^T\|_F^2 \\ &= \|\mathcal{Y}\|_F^2 + \mathrm{tr}(\mathbf{U}_\eta^{(n)}\mathbf{T}_n^T\mathbf{T}_n\mathbf{U}_\eta^{(n)T}) - 2\,\mathrm{tr}(\mathbf{Y}_{(n)}\,\mathbf{T}_n\,\mathbf{U}_\eta^{(n)T}) \\ &= \mathrm{tr}(\mathbf{U}_\eta^{(n)}\,\boldsymbol{\Gamma}_{-n}\,\mathbf{U}_\eta^{(n)T}) - 2\,\mathrm{tr}(\mathbf{G}_n\,\mathbf{U}_\eta^{(n)T}) + \|\mathcal{Y}\|_F^2 \\ &= \|\mathbf{G}_n\mathbf{V}_n\boldsymbol{\Sigma}^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\boldsymbol{\Sigma}^{\frac{1}{2}}\|_F^2 + \|\mathcal{Y}\|_F^2 - \|\mathbf{G}_n\mathbf{V}_n\boldsymbol{\Sigma}^{\frac{-1}{2}}\|_F^2, \end{aligned} \tag{6}$$

where $\mathbf{T}_n = \odot_{k\neq n}\,\mathbf{U}^{(k)}$, $\mathbf{G}_n = \mathbf{Y}_{(n)}\,\mathbf{T}_n$ is of size $I_n \times R$, and $\boldsymbol{\Gamma}_{-n} = \mathbf{T}_n^T\mathbf{T}_n = \circledast_{k\neq n}(\mathbf{U}^{(k)T}\mathbf{U}^{(k)})$ is of size $R \times R$.

We assume that the matrix $\boldsymbol{\Gamma}_{-n}$ is positive definite, and denote its Eigen-Value Decomposition (EVD) by $\boldsymbol{\Gamma}_{-n} = \mathbf{V}_n\boldsymbol{\Sigma}\mathbf{V}_n^T$, where $\mathbf{V}_n$ is an orthonormal matrix of eigenvectors, and $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1 \geq \cdots \geq \sigma_R > 0)$ is a diagonal matrix of positive eigenvalues. Note that the matrix $\mathbf{V}_n$ comprises right singular vectors associated with the singular values $\sigma_r^{\frac{1}{2}}$ of $\mathbf{T}_n$.

Let $\mathbf{F}_n = \mathbf{G}_n\mathbf{V}_n$. The optimization problem (4) becomes the linear regression with the bounded error constraint

$$\begin{aligned}\min \quad & \|\mathbf{U}_\eta^{(n)}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\boldsymbol{\Sigma}^{\frac{1}{2}}\|_F^2 \leq \delta_n^2, \end{aligned} \tag{7}$$

where $\delta_n^2 = \delta^2 + \|\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-1}{2}}\|_F^2 - \|\mathcal{Y}\|_F^2$. According to Lemma 2 in Appendix A, we can replace the inequality constraint by an equality constraint, and solve the equivalent problem in closed form after replacing $\mathbf{U}_\eta^{(n)}$ by its vectorization and formulating it as a Spherical Constrained QP (SCQP) in (43) for I_nR parameters. An alternative method is that we apply the conversion for matrix variate in Appendix C, and formulate an SCQP for only R parameters. To this end, we perform a reparameterization

$$\mathbf{Z}_n = \frac{1}{\delta_n}(\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\boldsymbol{\Sigma}^{\frac{1}{2}}), \tag{8}$$

$$\mathbf{U}_\eta^{(n)} = (\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-1}{2}} - \delta_n\,\mathbf{Z}_n)\,\boldsymbol{\Sigma}^{\frac{-1}{2}}\,\mathbf{V}_n^T, \tag{9}$$

and represent the Frobenius norm of $\mathbf{U}_\eta^{(n)}$ as

$$\begin{aligned}\|\mathbf{U}_\eta^{(n)}\|_F^2 = \|\mathbf{F}_n\boldsymbol{\Sigma}^{-1}\|_F^2 + \delta_n^2\,\mathrm{tr}(\mathbf{Z}_n\,\boldsymbol{\Sigma}^{-1}\,\mathbf{Z}_n^T) \\ - 2\delta_n\,\mathrm{tr}(\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-3}{2}}\mathbf{Z}_n^T).\end{aligned}$$

The matrix $\mathbf{Z}_n$ of the size $I_n \times R$ is a minimizer of an SCQP for matrix-variate

$$\begin{aligned}\min \quad & \delta_n\,\mathrm{tr}(\mathbf{Z}_n\,\boldsymbol{\Sigma}^{-1}\,\mathbf{Z}_n^T) - 2\,\mathrm{tr}(\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-3}{2}}\mathbf{Z}_n^T) \\ \text{s.t.} \quad & \|\mathbf{Z}_n\|_F^2 = 1, \end{aligned} \tag{10}$$

According to Lemma 3 in Appendix B and the SCQP for matrix variate in Appendix C, the minimizer $\mathbf{Z}_n^\star$ can be derived from the minimizer $\boldsymbol{z}^\star = [z_1^\star, \ldots, z_R^\star]^T$ which can be found in a closed-form of an SCQP of a smaller scale

$$\min \quad \delta_n\boldsymbol{z}^T\,\boldsymbol{\Sigma}^{-1}\,\boldsymbol{z}^T - 2\boldsymbol{c}^T\boldsymbol{z} \quad \text{s.t. } \boldsymbol{z}^T\boldsymbol{z} = 1, \tag{11}$$

where $\boldsymbol{c} = [\ldots, \sigma_r^{\frac{-3}{2}}\,\|\boldsymbol{f}_r^{(n)}\|, \ldots]$.

For a non zero c_r, the r-th column of $\mathbf{Z}_n^\star$ is the r-th column of $\mathbf{F}_n$ scaled by a factor of $\frac{z_r^\star}{\|\boldsymbol{f}_r^{(n)}\|}$

$$\boldsymbol{z}_r^{(n)\star} = \frac{z_r^\star}{\|\boldsymbol{f}_r^{(n)}\|}\,\boldsymbol{f}_r^{(n)}\,.$$

Otherwise, for a zero $c_r = 0$, $\boldsymbol{z}_r^{(n)\star}$ can be any vector of length $\|\boldsymbol{z}_r^{(n)\star}\|^2 = (z_r^\star)^2$. It can also be shown that if $c_r = 0$ for $r > 1$, then $z_r^\star = 0$ [31], hence $\boldsymbol{z}_r^{(n)\star}$ is a zero vector. Replacing $\mathbf{Z}_n$ in (9) by $\mathbf{Z}_n^\star$ yields a new update of $\mathbf{U}_\eta^{(n)}$.

At each iteration, we update $\mathbf{U}_\eta^{(n)}$ by a new matrix having a smaller Frobenius norm, while preserving the approximation error $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \delta^2$. The new estimates of η_r and $\boldsymbol{u}_r^{(n)}$ are respectively the ℓ_2-norm of the vector $\boldsymbol{u}_{\eta,r}^{(n)}$ and its ℓ_2-normalised version

$$\eta_r = \|\boldsymbol{u}_{\eta,r}^{(n)}\|\,, \boldsymbol{u}_r^{(n)} = \frac{\boldsymbol{u}_{\eta,r}^{(n)}}{\eta_r}\,.$$

Similarly, in the next iteration, we update $\mathbf{U}_\eta^{(n+1)}$, then normalise it to obtain new estimates of $\mathbf{U}^{(n+1)}$ and $\boldsymbol{\eta}$. The algorithm sequentially updates all $\mathbf{U}^{(n)}$ and stops when there is no significant improvement in $\boldsymbol{\eta}$. The Alternating Correction for Error Preservation (ACEP) is summarized in Algorithm 1. Similar to the ordinary ALS algorithm, the most expensive step in ACEP is the computation of N matrices $\mathbf{G}_n$. However, we need not compute these terms explicitly as in Step 2, but through a progressive computation of CP gradients [32]. In the case

Algorithm 1: Alternating Error Preserving Correction for CPD (ACEP).

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, a rank R and an error bound δ
Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ of rank R such that $\min \|\boldsymbol{\eta}\|_2^2$ s.t. $\|\mathcal{Y} - \mathcal{X}\|_F^2 \leq \delta^2$

begin
1 Initialize $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ such that $\|\mathcal{Y} - \mathcal{X}\|_F^2 \leq \delta^2$
repeat
for $n = 1, 2, \ldots, N$ **do**
2 Compute $\mathbf{G}_n = \mathbf{Y}_{(n)}\left(\bigodot_{k\neq n}\mathbf{U}^{(k)}\right)$
3 Compute EVD of $\boldsymbol{\Gamma}_{-n} = \circledast_{k\neq n}(\mathbf{U}^{(k)T}\mathbf{U}^{(k)}) = \mathbf{V}_n\boldsymbol{\Sigma}_n\mathbf{V}_n^T$ and $\mathbf{F}_n = \mathbf{G}_n\mathbf{V}_n$
4 Find $\boldsymbol{z}$ in the SCQP (11)
5 $\mathbf{Z}_n = [\ldots, \frac{z_r}{\|\boldsymbol{f}_r^{(n)}\|}\boldsymbol{f}_r^{(n)}, \ldots]$
6 $\mathbf{U}_\eta^{(n)} = (\mathbf{F}_n\boldsymbol{\Sigma}^{\frac{-1}{2}} - \delta_n\,\mathbf{Z}_n)\,\boldsymbol{\Sigma}^{\frac{-1}{2}}\,\mathbf{V}_n^T$ /* see an alternative form in (14) */
7 Update $\boldsymbol{\eta}$ and $\mathbf{U}^{(n)}$: $\eta_r = \|\boldsymbol{u}_{\eta,r}^{(n)}\|$, $\boldsymbol{u}_r^{(n)} = \frac{\boldsymbol{u}_{\eta,r}^{(n)}}{\eta_r}$
until *a stopping criterion is met*

of decomposition of a tensor of the size $I \times I \times \cdots \times I$, the computation of the gradient has a cost of $\mathcal{O}(RI^N)$.

B. Relation Between ACEP and ALS

Consider the case when the first column of $\mathbf{F}_n$ is non-zero, i.e., $c_1 \neq 0$, hence, $z_1^\star \neq 0$, and the matrix $\mathbf{Z}_n^\star$ can be expressed as

$$\mathbf{Z}_n^\star = \bar{\mathbf{F}}_n \ \mathrm{diag}([\ldots, z_r^\star, \ldots]), \tag{12}$$

where columns of $\bar{\mathbf{F}}_n$ are $\frac{\boldsymbol{f}_r^{(n)}}{\|\boldsymbol{f}_r^{(n)}\|}$ for non zero columns $\boldsymbol{f}_r^{(n)}$, and zero vectors elsewhere.

Since $\boldsymbol{f}_1^{(n)}$ is non-zero, the minimizer $\boldsymbol{z}^\star$ of the SCQP in (11) is given in closed-form as

$$z_r^\star = \frac{c_r}{\|\boldsymbol{c}\|(s_r - \lambda)} = \frac{\|\boldsymbol{f}_r(n)\|\sigma_r^{\frac{-3}{2}}}{\|\boldsymbol{c}\|(s_r - \lambda)}, \tag{13}$$

where $s_r = 1 + \frac{\delta_n}{\|\boldsymbol{c}\|}(\sigma_r^{-1} - \sigma_1^{-1})$, and λ is a unique root in $[0, 1)$ of a secular function $\sum_r (z_r^\star)^2 = 1$ [31], [33].

From (9), (12), (13) and the definition of $\boldsymbol{c}$, the new update of $\mathbf{U}_\eta^{(n)}$ can be expressed in a compact form as

$$\begin{aligned}\mathbf{U}_\eta^{(n)} &= (\mathbf{F}_n \boldsymbol{\Sigma}^{\frac{-1}{2}} - \delta_n \bar{\mathbf{F}}_n \,\mathrm{diag}([\ldots, z_r^\star, \ldots]))\, \boldsymbol{\Sigma}^{\frac{-1}{2}}\, \mathbf{V}_n^T \\ &= \mathbf{F}_n \,\mathrm{diag}\left(1 - \frac{\delta_n}{\|\boldsymbol{c}\|\sqrt{\sigma_r}(s_r - \lambda)}\right) \boldsymbol{\Sigma}^{-1} \mathbf{V}_n^T. \end{aligned} \tag{14}$$

Observe that only when $\delta_n = 0$, the above update (14) boils down to the ordinary ALS update for CPD

$$\mathbf{U}_\eta^{(n)} = \mathbf{Y}_{(n)} \mathbf{T}_n \boldsymbol{\Gamma}_{-n}^{-1}.$$

C. Sequential Quadratic Programming for EPC

1) An "all-at-once" algorithm: Similar to the ALS algorithm for CPD, the ACEP algorithm updates one factor matrix per iteration and may require many iterations to converge. Hence, it might be useful to consider an "all-at-once" algorithm for the EPC, which would be analog to the nonlinear algorithms for CPD. The algorithm follows the idea of the sequential quadratic programming [29], [30]. The objective function which represents the sum of Frobenius norms of rank-1 tensors is rewritten as

$$\begin{aligned} f(\boldsymbol{\theta}) &= \sum_{r=1}^{R} \|\boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}\|_F^2 \\ &= \sum_{r=1}^{R} \prod_{n=1}^{N} (\boldsymbol{u}_r^{(n)T}\, \boldsymbol{u}_r^{(n)}), \end{aligned} \tag{15}$$

and the optimization problem in (4) is stated as

$$\min \quad f(\boldsymbol{\theta}) \quad \text{s.t. } c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}(\boldsymbol{\theta})\|_F^2 \leq \delta^2\,, \tag{16}$$

where $\boldsymbol{\theta} = [\mathrm{vec}\left(\mathbf{U}^{(1)}\right)^T, \mathrm{vec}\left(\mathbf{U}^{(2)}\right)^T, \ldots, \mathrm{vec}\left(\mathbf{U}^{(N)}\right)^T]^T$.

As the derivation of the ACEP algorithm, we solve a minisation problem with an equality constraint

$$\min \quad f(\boldsymbol{\theta}) \quad \text{s.t. } c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}(\boldsymbol{\theta})\|_2^2 = \delta^2\,. \tag{17}$$

In order to perform this optimization, we first construct the Lagrangian function

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = f(\boldsymbol{\theta}) + \lambda\,(c(\boldsymbol{\theta}) - \delta^2), \tag{18}$$

then approximate $\mathcal{L}(\boldsymbol{\theta}^{(k)} + \boldsymbol{d}_\theta, \lambda^{(k)} + d_\lambda)$ by a second order Taylor expansion around $(\boldsymbol{\theta}^{(k)}, \lambda^{(k)})$

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\theta}^{(k)} + \boldsymbol{d}_\theta, \lambda^{(k)} + d_\lambda) \approx \\ &\mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)}) + (\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)}))^T \boldsymbol{d} \\ &+ \frac{1}{2}\, \boldsymbol{d}^T\, [\nabla^2 \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)})]\, \boldsymbol{d}, \end{aligned}$$

where $\boldsymbol{d} = \left[\boldsymbol{d}_\theta^T, d_\lambda\right]^T$ represents the vector of increment. This gives an approximation to the gradient

$$\begin{aligned} &\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)} + \boldsymbol{d}_\theta, \lambda^{(k)} + d_\lambda) \\ &\approx \nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)}) + [\nabla^2 \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)})]\, \boldsymbol{d}. \end{aligned}$$

By setting the gradient to zero, we obtain the Newton iteration update as the solution to the following linear equation

$$[\nabla^2 \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)})]\, \boldsymbol{d} = -\nabla \mathcal{L}(\boldsymbol{\theta}^{(k)}, \lambda^{(k)})$$

or more explicitly

$$\begin{aligned} &\begin{bmatrix} \mathbf{H}_{\lambda^{(k)}}(\boldsymbol{\theta}^{(k)}) & \boldsymbol{g}_c(\boldsymbol{\theta}^{(k)}) \\ \boldsymbol{g}_c^T(\boldsymbol{\theta}^{(k)}) & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_\theta \\ d_\lambda \end{bmatrix} \\ &= -\begin{bmatrix} \boldsymbol{g}_f(\boldsymbol{\theta}^{(k)}) + \lambda^{(k)} \boldsymbol{g}_c(\boldsymbol{\theta}^{(k)}) \\ c(\boldsymbol{\theta}^{(k)}) - \delta^2 \end{bmatrix}, \end{aligned} \tag{19}$$

where $\boldsymbol{g}_f(\boldsymbol{\theta})$ and $\boldsymbol{g}_c(\boldsymbol{\theta})$ are gradients of the objective and constraint functions with respect to $\boldsymbol{\theta}$. The Hessian $\mathbf{H}_{\lambda^{(k)}}(\boldsymbol{\theta}^{(k)})$ is computed as

$$\mathbf{H}_{\lambda^{(k)}}(\boldsymbol{\theta}^{(k)}) = \nabla^2 f(\boldsymbol{\theta}^{(k)}) + \lambda^{(k)}\, \nabla^2 c(\boldsymbol{\theta}^{(k)}). \tag{20}$$

The solution in (19) is also a minimizer of the following QP subproblem

$$\begin{aligned} \min \quad & \frac{1}{2} \boldsymbol{d}_\theta^T\, \mathbf{H}_{\lambda^{(k)}}(\boldsymbol{\theta}^{(k)})\, \boldsymbol{d}_\theta + \boldsymbol{g}_f^T(\boldsymbol{\theta}^{(k)})\, \boldsymbol{d}_\theta \\ \text{s.t.} \quad & c(\boldsymbol{\theta}^{(k)}) + \boldsymbol{g}_c^T(\boldsymbol{\theta}^{(k)})\, \boldsymbol{d}_\theta = \delta^2. \end{aligned} \tag{21}$$

Solving either (19) or the QP in (21) gives us the new search direction, $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{\delta}_\theta$, and the Lagrange multiplier, $\lambda^{(k+1)} = \lambda^{(k)} + \delta_\lambda$. However, the Hessian is of size $R(\sum_n I_n) \times R(\sum_n I_n)$, and its inverse is computationally demanding if the tensor dimensions are large. This makes the SQP algorithm impractical. Further details regarding the SQP method can be found in [29], [30].

We next derive the Hessian and gradient of the Lagrangian, and then apply a similar method for fast inversion of the Hessian used in [34], [35] to solve the linear system in (19).

2) Fast Inversion of the Hessian: Following (48) in Appendix D and (49) in Appendix E, the Hessian $\mathbf{H}_\lambda$ can be expressed as a rank-R^2 adjustment form as

$$\begin{aligned} \mathbf{H}_\lambda &= \mathbf{H}_f + \lambda\, \mathbf{H}_c \\ &= \tilde{\mathbf{G}} + \tilde{\mathbf{Z}}\, \boldsymbol{\Psi}_\lambda\, \tilde{\mathbf{Z}}^T, \end{aligned}$$

where $\boldsymbol{\Psi}_\lambda = \mathbf{P}_{R,R}\,\mathrm{dvec}(\boldsymbol{\Gamma}_\lambda)$, $\tilde{\mathbf{G}}$ is a block diagonal matrix of square matrices, $\tilde{\mathbf{G}}_n$, of size $I_n R \times I_n R$

$$\tilde{\mathbf{G}}_n \;=\; \boldsymbol{\Gamma}_\lambda^{(-n)} \otimes \mathbf{I}_{I_n} - \tilde{\mathbf{Z}}_n\,\boldsymbol{\Psi}_\lambda\,\tilde{\mathbf{Z}}_n^T\,,$$

and $\boldsymbol{\Gamma}_\lambda$ and $\boldsymbol{\Gamma}_\lambda^{(-n)}$ are square matrices of size $R \times R$ respectively adjusted from the matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}_{-n}$ as

$$\boldsymbol{\Gamma}_\lambda^{(-n)}(r,s) = \begin{cases} \lambda\,\boldsymbol{\Gamma}_{-n}(r,s), & r \neq s, \\ (\lambda+1)\,\boldsymbol{\Gamma}_{-n}(r,r), & r = s\,, \end{cases} \tag{22}$$

$$\boldsymbol{\Gamma}_\lambda(r,s) = \begin{cases} \lambda\,\boldsymbol{\Gamma}(r,s), & r \neq s, \\ (\lambda+2)\,\boldsymbol{\Gamma}(r,r), & r = s. \end{cases} \tag{23}$$

With the condition $R < \sum_n I_n$, the matrix $\tilde{\mathbf{Z}}$ is a tall matrix of size $R(\sum_n I_n) \times R^2$. $\tilde{\mathbf{G}}$ is a block diagonal matrix, its inverse is efficiently computed through the inverses of its block matrices $\tilde{\mathbf{G}}_n$. However, since the Hessian matrix is rank-deficient, we suggest increasing its diagonal by a sufficiently large μ to make the smallest eigenvalue positive. This is similar to adding the penalty term $\mu\|\boldsymbol{\theta}\|_F^2$ into the objective function $f(\boldsymbol{\theta})$

$$\min \quad f(\boldsymbol{\theta}) + \mu\|\boldsymbol{\theta}\|_2^2 \quad \text{s.t.} \quad c(\boldsymbol{\theta}) \leq \delta^2,$$

or equivalently minimising the problem (16) with an additional constraint $\|\boldsymbol{\theta}\|_2^2 \leq \alpha^2$

$$\min \quad f(\boldsymbol{\theta}) \quad \text{s.t.} \quad c(\boldsymbol{\theta}) \leq \delta^2, \|\boldsymbol{\theta}\|_2^2 \leq \alpha^2\;.$$

Since shifting eigenvalues does not change the low-rank adjustment structure of the Hessian, following [34], [35], we can invert the damped Hessian $\mathbf{H}_{\lambda,\mu} = \mathbf{H}_\lambda + \mu\mathbf{I}$ as follows

$$\begin{aligned} \mathbf{H}_{\lambda,\mu}^{-1} &= (\tilde{\mathbf{G}}_\mu + \tilde{\mathbf{Z}}\,\boldsymbol{\Psi}_\lambda\,\tilde{\mathbf{Z}}^T)^{-1} \\ &= \tilde{\mathbf{G}}_\mu^{-1} - \tilde{\mathbf{G}}_\mu^{-1}\tilde{\mathbf{Z}}\,(\boldsymbol{\Psi}_\lambda^{-1} + \tilde{\mathbf{Z}}^T\tilde{\mathbf{G}}_\mu^{-1}\tilde{\mathbf{Z}})^{-1}\,\tilde{\mathbf{Z}}^T\tilde{\mathbf{G}}_\mu^{-1}, \end{aligned} \tag{24}$$

where the block diagonal matrix $\tilde{\mathbf{G}}_\mu = \tilde{\mathbf{G}} + \mu\mathbf{I}$ and

$$\tilde{\mathbf{G}}_\mu^{-1} = \mathrm{blkdiag}(\ldots, (\tilde{\mathbf{G}}_n + \mu\mathbf{I})^{-1}, \ldots),$$

$$(\tilde{\mathbf{G}}_n + \mu\mathbf{I})^{-1} = ((\boldsymbol{\Gamma}_\lambda^{(-n)} + \mu\mathbf{I}) \otimes \mathbf{I}_{I_n} - \tilde{\mathbf{Z}}_n\,\boldsymbol{\Psi}_\lambda\,\tilde{\mathbf{Z}}_n^T)^{-1}. \tag{25}$$

If $R < I_n$, the matrices $\tilde{\mathbf{Z}}_n$ are tall and of size $RI_n \times R^2$, the inversion $(\tilde{\mathbf{G}}_n + \mu\mathbf{I})^{-1}$ can be performed even more efficiently as

$$\begin{aligned} &(\tilde{\mathbf{G}}_n + \mu\mathbf{I})^{-1} \\ &= \boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n} - (\boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n})\,\tilde{\mathbf{Z}}_n(\boldsymbol{\Psi}_\lambda^{-1} + \tilde{\mathbf{Z}}_n^T\,(\boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n})\,\tilde{\mathbf{Z}}_n)^{-1} \\ &\quad \tilde{\mathbf{Z}}_n^T\,(\boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n}) \\ &= \boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n} - (\boldsymbol{\Phi}_n \otimes \mathbf{U}^{(n)})\,\mathrm{dvec}(1 \oslash \boldsymbol{\Gamma}_n) \\ &\quad (\boldsymbol{\Psi}_\lambda^{-1} + \mathrm{dvec}(1 \oslash \boldsymbol{\Gamma}_n)(\boldsymbol{\Phi}_n \otimes \boldsymbol{\Gamma}_n)\;\mathrm{dvec}(1 \oslash \boldsymbol{\Gamma}_n))^{-1} \\ &\quad \mathrm{dvec}(1 \oslash \boldsymbol{\Gamma}_n)\,(\boldsymbol{\Phi}_n \otimes \mathbf{U}^{(n)T}) \\ &= \boldsymbol{\Phi}_n \otimes \mathbf{I}_{I_n} - (\boldsymbol{\Phi}_n \otimes \mathbf{U}^{(n)}) \\ &\quad (\mathbf{P}_{R,R}\,\mathrm{dvec}(\boldsymbol{\Gamma}_n^2 \oslash \boldsymbol{\Gamma}_\lambda) + \boldsymbol{\Phi}_n \otimes \boldsymbol{\Gamma}_n)^{-1}(\boldsymbol{\Phi}_n \otimes \mathbf{U}^{(n)T}), \end{aligned} \tag{26}$$

Algorithm 2: `CPD with EPC`.

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, a rank R, and a bound ϵ
Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ of rank R
begin
1 Initialize $\mathcal{X}_{[0]} = [\![\boldsymbol{\eta}_0; \mathbf{U}_0^{(1)}, \mathbf{U}_0^{(2)}, \ldots, \mathbf{U}_0^{(N)}]\!]$
repeat
2 **if** $\|\boldsymbol{\eta}_k\|_2^2 \geq \epsilon^2$ **then**
3 Apply EPC to $\mathcal{X}_{[k]}$ to find a new tensor $\mathcal{X}_{[k+1]}$ with a minimum norm $\|\boldsymbol{\eta}_{k+1}\|_2^2$ s.t $\|\mathcal{Y} - \mathcal{X}_{[k+1]}\|_F \leq \|\mathcal{Y} - \mathcal{X}_{[k]}\|_F$
4 **else**
5 Continue the CPD of $\mathcal{Y}$ to find $\mathcal{X}_{[k+1]}$ by $\mathcal{X}_{[k]}$
until *a stopping criterion is met*

where $\boldsymbol{\Phi}_n = (\boldsymbol{\Gamma}_\lambda^{(-n)} + \mu\mathbf{I})^{-1}$. The last expression is obtained using the following identity

$$\mathbf{P}_{R,R}\;\mathrm{dvec}(\boldsymbol{\Gamma}_n) = \mathrm{dvec}(\boldsymbol{\Gamma}_n)\,\mathbf{P}_{R,R}\,.$$

Now, by exploiting the rank-1 expansion and replacing $\mathbf{H}_\lambda$ by the damped $\mathbf{H}_{\lambda,\mu}$, the inverse of the Hessian $\nabla^2\mathcal{L}$ in (19) can be expressed as

$$\begin{aligned} \begin{bmatrix} \mathbf{H}_{\lambda,\mu} & \boldsymbol{g}_c \\ \boldsymbol{g}_c & 0 \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{H}_{\lambda,\mu}^{-1} \\ 0 \end{bmatrix} \\ &\quad - \frac{1}{\boldsymbol{g}_c^T\mathbf{H}_{\lambda,\mu}^{-1}\boldsymbol{g}_c} \begin{bmatrix} \mathbf{H}_{\lambda,\mu}^{-1}\boldsymbol{g}_c \\ -1 \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_c^T\mathbf{H}_{\lambda,\mu}^{-1} & -1 \end{bmatrix} \end{aligned}$$

then we apply the inversion in (24) with (25) or with (26) to compute the inverse of the Hessian.

From (19), new estimates of the Lagrange multiplier $\lambda^{(k+1)}$ and search direction $\boldsymbol{d}_\theta$ are given by

$$\lambda^{(k+1)} = \frac{c(\boldsymbol{\theta}^{(k)}) - \delta^2 - \boldsymbol{g}_c^T(\boldsymbol{\theta}^{(k)})\,\mathbf{H}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)})\boldsymbol{g}_f(\boldsymbol{\theta}^{(k)})}{\boldsymbol{g}_c^T(\boldsymbol{\theta}^{(k)})\,\mathbf{H}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)})\,\boldsymbol{g}_c(\boldsymbol{\theta}^{(k)})},$$

$$\boldsymbol{d}_\theta = -\mathbf{H}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)})(\boldsymbol{g}_f(\boldsymbol{\theta}^{(k)}) + \lambda^{(k+1)}\,\boldsymbol{g}_c(\boldsymbol{\theta}^{(k)}))\,.$$

Hence, a new estimate of the parameters would be $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{d}_\theta$. In addition, the loading components of rank-1 tensors are then normalised to have an equal norm, i.e.,

$$\boldsymbol{u}_r^{(n)} \leftarrow \boldsymbol{u}_r^{(n)}\,\frac{\eta_r^{1/N}}{\|\boldsymbol{u}_r^{(n)}\|},$$

where $\eta_r = \prod_n \|\boldsymbol{u}_r^{(n)}\|$.

D. Implementation

Algorithm 2 shows a practical implementation of CPD with EPC. A requirement for EPC is that the initial point is feasible, i.e., obeys the constraint $\|\mathcal{Y} - \mathcal{X}_{[k]}\|_F \leq \delta$. In practice, we first fit the tensor by a standard CP model, then verify the norm of rank-1 tensor components. If it exceeds a bound, we apply the correction method as in Step 3, otherwise, continue the CP decomposition in Step 5. Following this, the estimated tensor is a feasible point and the bound $\delta = \|\mathcal{Y} - \mathcal{X}_{[k-1]}\|_F$ is the current approximation error. Since the ACEP algorithm solves

sub-problems in closed-form, the new update points are always in a feasible region, i.e., $c(\boldsymbol{\theta}) \leq \delta^2$, while the objective function decreases sequentially or at least is kept to not increase.

Unlike the ACEP, the SQP algorithm solves the approximate problems. Even if an initial feasible point is provided, the update points may not reside in the feasible region in some first iterations. In practice, we can first execute the ACEP for a small number of iterations, then switch to the SQP or Interior Point (ITP) algorithm.

In addition, one can gradually increase the bound, δ, e.g., by a factor of 1.1, until the norm of rank-1 tensors attains the desired value.

E. EPC – A Tool for Decomposition at a Target Error Bound and Denoising

When the data tensor $\mathcal{Y}$ is corrupted by e.g., additive Gaussian noise with a zero mean and variance σ^2, the constraint error bound, δ, in the decomposition in (4), plays the role of the Frobenius norm of the noise tensor $\mathcal{E}$ with i.i.d. $N(0, \sigma)$ entries, i.e., $\delta^2 = \|\mathcal{E}\|_F^2$. In practice, the noise level σ might be unknown, but it can be determined by inspecting the coefficients of the data (images, signals) in the high frequency bands.

Assume now that $\hat{\mathcal{Y}}^\star$ is a tensor with the smallest rank $\boldsymbol{R}^\star$ such that $\|\mathcal{Y} - \hat{\mathcal{Y}}^\star\|_F^2 = \delta^2$. Decomposition of a noisy data tensor following the CP, Tucker or Tensor-train model, tends to explain the noise when the estimated rank exceeds the true trank $\boldsymbol{R}^\star$. The higher the estimated tensor rank, the more noise the tensor $\hat{\mathcal{Y}}$ will explain from the data $\mathcal{Y}$. Consequently, the decomposition deteriorates the overall noise reduction result.

When the noise is relatively high and dominant to the data, the tensor $\hat{\mathcal{Y}}^\star$ which best approximates the noisy tensor $\mathcal{Y}$ at the noise level, i.e., $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \approx \|\mathcal{E}\|_F^2$, usually has a lower rank than the true rank of the original tensor. Hence a tensor which approximates $\mathcal{Y}$ with its true rank often yields an overestimate, and therefore, the conventional low-rank tensor approximation methods are not well-suited to the noise removal.

Different from the existing tensor decompositions, the constrained decomposition in (4) can maintain the approximation at a target error bound, i.e., $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \delta^2$, with an estimated rank $\boldsymbol{R} \geq \boldsymbol{R}^\star$. It is then straightforward to determine a tensor $\hat{\mathcal{Y}}$ with a minimal rank R for which the constraint is satisfied.

When the approximation error in EPC does not reach the bound, the approximated tensor $\hat{\mathcal{Y}}$ was set up with a relatively low rank. We should try another decomposition with a higher rank. When EPC attains the error bound, it is possible that the rank of $\hat{\mathcal{Y}}$ is not minimal. One can start another decomposition with a smaller estimated rank to reduce the number of parameters or to obtain the optimal low-rank tensor approximation. However, it might not be necessary for the denoising application.

Fig. 2 illustrates the approximation errors for tensors which are of rank-10 but corrupted by Gaussian noise. EPC maintains the approximation error at the target bounds when the rank of the decomposition exceeds the true tensor rank. Another important observation is that the overall norm of rank-1 tensors increases with the increasing of the rank R, then decreases

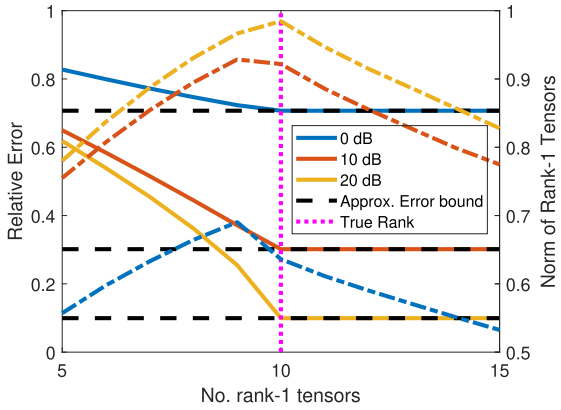


Fig. 1. EPC for order-3 tensors of the dimension $I = 50$ and rank $R = 10$ corrupted by Gaussian noise at SNRs $= 0$, 10 and 20 dB. Solid and dashed lines, respectively, show the relative errors and norms of rank-1 tensors for various ranks of the approximated tensors.

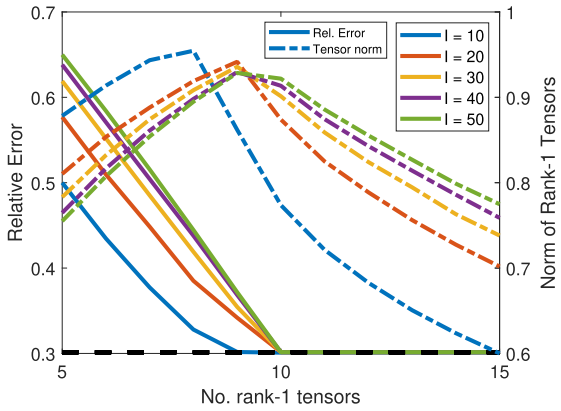


Fig. 2. EPC for order-3 tensors of various dimensions $I = 10, 20, \ldots, 50$ corrupted by Gaussian noise at SNR $= 10$dB. Solid and dashed lines respectively show the relative errors and norms of rank-1 tensors for various ranks of the approximated tensors.

when the estimated rank exceeds the true tensor rank. This is quite straightforward since the decomposition yields several highly collinear rank-1 tensors, which finally tend to be nearly identical after the norm miminization. These results are further confirmed in Fig. 2 for decomposition of tensors of various sizes $I = 10, 20, \ldots, 50$.

III. Canonical Polyadic Tensor Decomposition With Bound on Norm of Rank-1 Tensors

In contrast to the tensor approximation with a minimal norm of rank-1 tensors, in this section, we consider a constrained CPD, in which the norm of rank-1 tensors is bounded

$$\min \quad \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{\eta}\|_2^2 \leq \epsilon^2. \tag{27}$$

A. Alternating Update Algorithm

Similar to the previous section, we can absorb $\boldsymbol{\eta}$ into a factor matrix $\mathbf{U}^{(n)}$ and rewrite the above optimization problem as

$$\min \quad \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{U}_\eta^{(n)}\|_F^2 \leq \epsilon^2 \ , \tag{28}$$

or equivalently as a Quadratic programming with a bounded norm for $\mathbf{U}_\eta^{(n)}$

$$\begin{aligned} \min \quad & \operatorname{tr}(\mathbf{U}_\eta^{(n)} \boldsymbol{\Gamma}_{-n} \mathbf{U}_\eta^{(n)T}) - 2\operatorname{tr}(\mathbf{G}_n \mathbf{U}_\eta^{(n)T}) \\ \text{s.t.} \quad & \|\mathbf{U}_\eta^{(n)}\|_F^2 \le \epsilon^2 . \end{aligned} \tag{29}$$

The above expression follows the expansion of the Frobenius norm in (6). Next, we convert the above matrix-variate QP to the one for a vector of length $R \times 1$.

Let $\boldsymbol{\Gamma}_{-n} = \mathbf{V}_n \boldsymbol{\Sigma} \mathbf{V}_n^T$ be the eigenvalue decomposition of $\boldsymbol{\Gamma}_{-n}$. Then denote $\mathbf{F}_n = \mathbf{G}_n \mathbf{V}_n$ and $\mathbf{Z}_n = \mathbf{U}_\eta^{(n)} \mathbf{V}_n$. The optimization in (29) is transformed into

$$\begin{aligned} \min \quad & \operatorname{tr}(\mathbf{Z}_n \boldsymbol{\Sigma} \mathbf{Z}_n^T) - 2\operatorname{tr}(\mathbf{F}_n \mathbf{Z}_n^T) \\ \text{s.t.} \quad & \|\mathbf{Z}_n\|_F^2 \le \epsilon^2 . \end{aligned} \tag{30}$$

Similar to (10) and according to Lemma 3 in Appendix B, the minimizer $\mathbf{Z}_n^\star$ to the matrix-variate QP in (30) can be derived from the minimizer $\boldsymbol{z}^\star$ of the following constrained QP

$$\min \quad \boldsymbol{z}^T \boldsymbol{\Sigma} \boldsymbol{z} - 2\boldsymbol{c}^T \boldsymbol{z} \quad \text{s.t. } \boldsymbol{z}^T \boldsymbol{z} \le \epsilon^2 , \tag{31}$$

where the vector $\boldsymbol{c} = [\ldots, \|\boldsymbol{f}_r^{(n)}\|_2, \ldots]^T$ comprises the norm of columns of $\mathbf{F}_n$. We note that the above QP problem with an inequality constraint can be solved in closed-form.

If $\sum_{r=1}^R \frac{c_r^2}{\sigma_r^2} \le \epsilon^2$, the minimizer of (31) can be simply expressed as

$$\boldsymbol{z}^\star = \boldsymbol{c} \oslash \boldsymbol{\sigma} . \tag{32}$$

This case often occurs when the current parameter point is in a feasible set, and the bound is set to a relatively high value.

Otherwise, $\boldsymbol{z}^\star$ is a minimizer of the QP over a sphere which again can also be solved in closed-form [31]

$$\min \quad \boldsymbol{z}^T \boldsymbol{\Sigma} \boldsymbol{z} - 2\boldsymbol{c}^T \boldsymbol{z} \ \text{ s.t. } \boldsymbol{z}^T \boldsymbol{z} = \epsilon^2 . \tag{33}$$

Finally, we obtain $\mathbf{Z}_n = \mathbf{F}_n \operatorname{diag}(\boldsymbol{z} \oslash \boldsymbol{c})$ and the new update of $\mathbf{U}_\eta^{(n)}$

$$\mathbf{U}_\eta^{(n)} = \mathbf{Z}_n \mathbf{V}_n^T = \mathbf{G}_n \mathbf{V}_n \operatorname{diag}(\boldsymbol{z} \oslash \boldsymbol{c}) \mathbf{V}_n^T . \tag{34}$$

The proposed algorithm works in the same manner as the ordinary ALS algorithm. We call this the BALS algorithm and summarize it in Algorithm 3.

Algorithm 3: ALS for Bounded CPD (BALS).

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, a rank R, and a bound ϵ
Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ of rank R such that $\min \|\mathcal{Y} - \mathcal{X}\|_F^2$ s.t. $\|\boldsymbol{\eta}\|_2^2 \le \epsilon^2$

begin
1. Initialize $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$
   repeat
      for $n = 1, 2, \ldots, N$ **do**
2. Compute $\mathbf{G}_n = \mathbf{Y}_{(n)} \left(\bigodot_{k \ne n} \mathbf{U}^{(k)} \right)$
3. Compute EVD of $\boldsymbol{\Gamma}_{-n} = \mathbf{V}_n \boldsymbol{\Sigma} \mathbf{V}_n^T$
4. Find $\boldsymbol{z}$ in the SCQP (31)
5. $\mathbf{U}_\eta^{(n)} = \mathbf{G}_n \mathbf{V}_n \operatorname{diag}(\boldsymbol{z} \oslash \boldsymbol{c}) \mathbf{V}_n^T$ /* an alternative form in (35) */
6. Update $\boldsymbol{\eta}$ and $\mathbf{U}^{(n)}$: $\eta_r = \|\boldsymbol{u}_{\eta,r}^{(n)}\|$, $\boldsymbol{u}_r^{(n)} = \frac{\boldsymbol{u}_{\eta,r}^{(n)}}{\eta_r}$
   until *a stopping criterion is met*

B. Relation Between BALS and ALS with Smoothness Constraint

We consider the case when the last column of $\mathbf{F}_n$ is non zero, i.e., $c_R \ne 0$. Assuming that the eigenvalues of $\boldsymbol{\Gamma}_{-n}$ are ordered in the descending order, i.e., $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_R > 0$, the SCQP in (33) has a minimizer given in form

$$\boldsymbol{z}^\star = \left[\ldots, \frac{c_r}{\sigma_r - \widetilde{\lambda}}, \ldots \right],$$

where $\widetilde{\lambda}$ is a unique solution of the following secular equation in $[\sigma_R - \|\boldsymbol{c}\|, \sigma_R - \|\boldsymbol{c}\|(1 - 1/\epsilon)]$

$$\boldsymbol{z}^{\star T} \boldsymbol{z}^\star = \sum_r \frac{c_r^2}{(\sigma_r - \widetilde{\lambda})^2} = \epsilon^2 .$$

This equation can be solved in closed-form [31], [33]. The minimizer in (32) is a particular case of the above when $\widetilde{\lambda} = 0$. Hence, from the conversion of the QP for matrix-variate in Appendix C and Lemma 3, we can write $\mathbf{Z}_n^\star$ as

$$\mathbf{Z}_n^\star = \mathbf{F}_n \operatorname{diag}(\ldots, (\sigma_r - \widetilde{\lambda})^{-1}, \ldots) .$$

Replacing this into $\mathbf{U}_\eta^{(n)}$, we obtain a simple update rule

$$\begin{aligned} \mathbf{U}_\eta^{(n)} &= \mathbf{G}_n \mathbf{V}_n \operatorname{diag}(\ldots, (\sigma_r - \widetilde{\lambda})^{-1}, \ldots) \mathbf{V}_n^T \\ &= \mathbf{G}_n (\boldsymbol{\Gamma}_{-n} - \widetilde{\lambda} \mathbf{I}_R)^{-1} . \end{aligned} \tag{35}$$

The update rule (35) is indeed similar to the ALS update rule derived for the objective function in (2) with $\mu = -\widetilde{\lambda}$. Here, we show a relation between the regularization parameter μ and the bound ϵ. In the decomposition in (2), the regularisation or damping parameter μ can be fixed or adaptively adjusted to keep the cost function non-increasing. In our algorithm, the parameter $\widetilde{\lambda}$ is a root of the secular equation and is updated in each iteration.

C. Sequential Quadratic Programming Method

Similar to the SQP algorithm for the optimization problem in (16), we relax the unit-length constraints of the loading components and develop an SQP algorithm for the CPD with a bounded rank-1 tensor norm

$$\min \quad c(\boldsymbol{\theta}) \quad \text{s.t.} \quad f(\boldsymbol{\theta}) \le \epsilon^2 , \tag{36}$$

where the functions $f(\boldsymbol{\theta})$ and $c(\boldsymbol{\theta})$ exchange their roles in the optimization problem (16). The Lagrangian to the above constrained optimization problem is given by

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = c(\boldsymbol{\theta}) + \lambda \left(f(\boldsymbol{\theta}) - \epsilon^2 \right) .$$

Similar to the Lagrangian in (18), the new search direction is a minimizer of the QP subproblem

$$\begin{aligned} \min \quad & \frac{1}{2} \boldsymbol{d}_\theta^T \widetilde{\mathbf{H}}_{\lambda^{(k)}}(\boldsymbol{\theta}^{(k)}) \boldsymbol{d}_\theta + \boldsymbol{g}_{\lambda^{(k)}}^T(\boldsymbol{\theta}^{(k)}) \boldsymbol{d}_\theta \\ \text{s.t.} \quad & f(\boldsymbol{\theta}^{(k)}) + \boldsymbol{g}_f^T(\boldsymbol{\theta}^{(k)}) \boldsymbol{d}_\theta \le \epsilon^2, \end{aligned} \tag{37}$$

TABLE II
PERFORMANCE COMPARISON FOR ALGORITHMS IN EXAMPLE 1

	ALS +line search	NLS	ALS+EPC +ALS	ALS+EPC +BALS
Relative Error	0.3387	0.3372	0.3254	0.3240
Running time (s)	254.2	1087.7	283.5	1080.8
Norm of Rank-one tensors	2.08×10^8	4.75×10^8	3.14×10^4	4.43×10^3

where

$$\widetilde{\mathbf{H}}_\lambda(\boldsymbol{\theta}) = \nabla^2 c(\boldsymbol{\theta}) + \lambda \nabla^2 f(\boldsymbol{\theta}),$$
$$\boldsymbol{g}_\lambda(\boldsymbol{\theta}) = \boldsymbol{g}_c(\boldsymbol{\theta}) + \lambda \boldsymbol{g}_f(\boldsymbol{\theta}).$$

If the non-constrained solution, i.e., $-\widetilde{\mathbf{H}}_\lambda^{-1}\, \boldsymbol{g}_\lambda$, is in the feasible set, then it is the minimizer and $d_\lambda = 0$. Otherwise, we need to solve the QP with an equality constraint. Similar to (21) it leads to finding the solution to a system of linear equations

$$\begin{bmatrix} \widetilde{\mathbf{H}}_\lambda & \boldsymbol{g}_f \\ \boldsymbol{g}_f^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_\theta \\ d_\lambda \end{bmatrix} = - \begin{bmatrix} \boldsymbol{g}_c + \lambda \boldsymbol{g}_f \\ f - \epsilon^2 \end{bmatrix}, \tag{38}$$

where the Hessian $\widetilde{\mathbf{H}}_\lambda = \lambda\, \mathbf{H}_{1/\lambda}$ shares the low-rank adjustment structure of $\mathbf{H}_\lambda$. This finally leads to the compact update rules for the Lagrange multiplier and the search direction

$$\lambda^{(k+1)} = \frac{f(\boldsymbol{\theta}^{(k)}) - \epsilon^2 - \boldsymbol{g}_f^T(\boldsymbol{\theta}^{(k)})\, \widetilde{\mathbf{H}}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)}) \boldsymbol{g}_c(\boldsymbol{\theta}^{(k)})}{\boldsymbol{g}_f^T(\boldsymbol{\theta}^{(k)})\, \widetilde{\mathbf{H}}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)})\, \boldsymbol{g}_f(\boldsymbol{\theta}^{(k)})},$$

$$\boldsymbol{d}_\theta = -\widetilde{\mathbf{H}}_{\lambda,\mu}^{-1}(\boldsymbol{\theta}^{(k)})(\boldsymbol{g}_c(\boldsymbol{\theta}^{(k)}) + \lambda^{(k+1)}\, \boldsymbol{g}_f(\boldsymbol{\theta}^{(k)})).$$

IV. NUMERICAL RESULTS

In this section, we illustrate the efficiency of the proposed EPC method and the bounded norm CPD in some hard scenarios for decomposition of synthesized tensors and tensors of real-world data.

Example 1 (Decomposition of the convolutional tensor in the Alexnet convolutional neural network): In this example, we show the efficiency of the proposed algorithm over the alternating least squares (ALS) with a simple line search and nonlinear least squares (NLS) algorithms for low-rank approximation of the convolutional kernel at the 10th layer in the pretrained Alexnet convolutional neural network [36]. This decomposition aims in acceleration of the inference process of convolutional neural networks [6], [7]. However, this is out of scope of the paper, and we compare only the approximation errors of the considered algorithms. The tensor had a size of $256 \times 384 \times 3 \times 3$, but was reshaped to the dimension of $256 \times 384 \times 9$, then approximated by a CPD with a rank $R = 200$. Decomposition of such tensors using the ALS[37] or the NLS algorithm [38] often encountered degeneracy. The ratio between the largest and smallest norms even exceeded 600, and the total squared norms of the estimated rank-1 tensors on average were 2.08×10^8 and 4.75×10^8 after 5000 iterations using ALS and NLS, respectively. Such degeneracy caused the entire network instability in fine-tuning [7]. The results are compared in Table II.

We applied EPC after running the ALS update 1000 iterations, then executed the ALS again or the BALS algorithm.

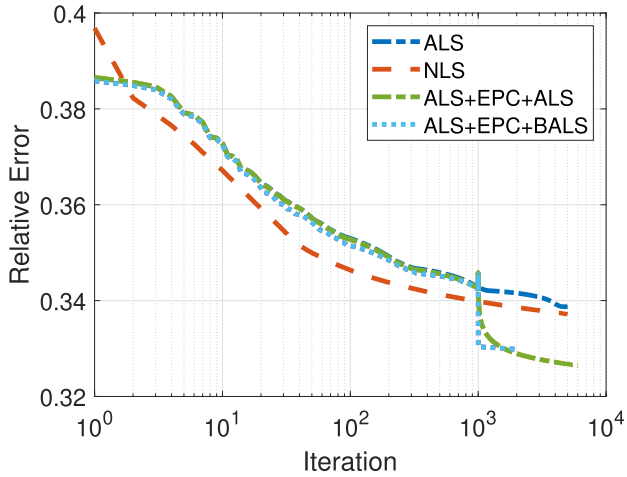


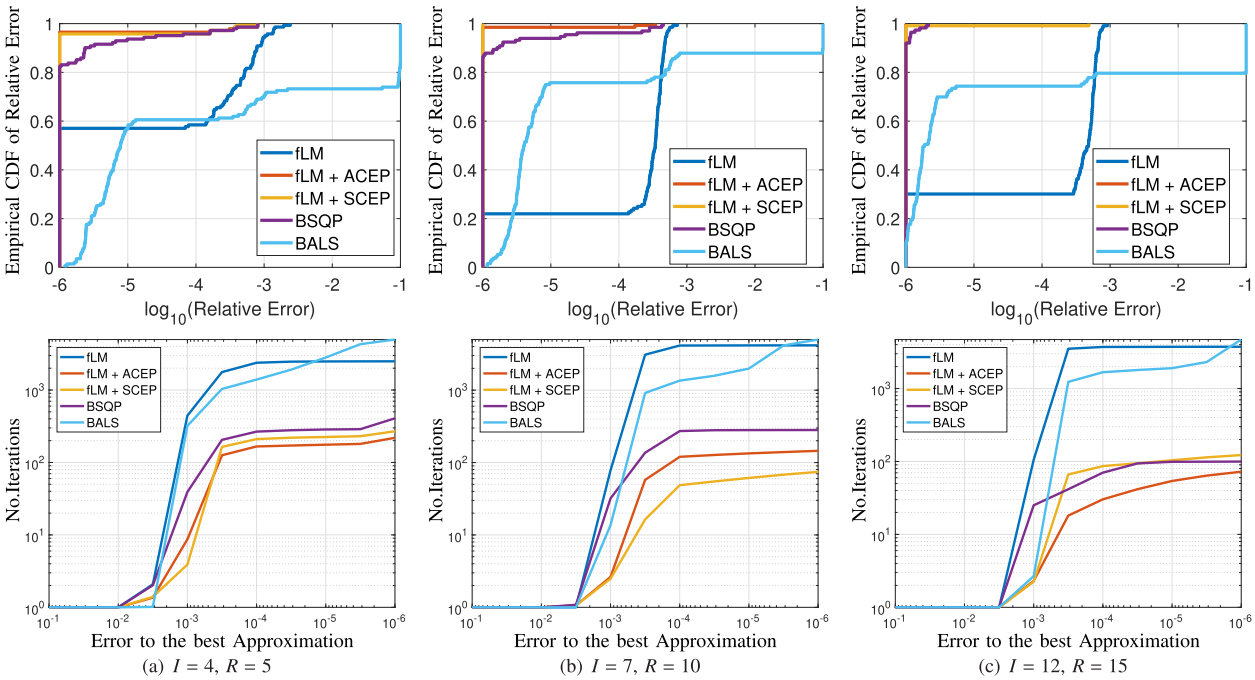
Fig. 3. Both ALS and NLS in Example II converged slowly because of the high norm of rank-1 tensors. After the EPC, ALS and BALS quickly attained significantly lower approximation error.

Fig. 3 illustrates the mean relative errors of the four algorithms over the iterations. The curves were computed from 100 independent runs. It is clear that the ALS quickly attained the lower approximation error after the correction, whereas the BALS even achieved better performances than ALS+EPC. In general the NLS algorithm achieved lower approximation errors than ALS, but these two algorithms converged slowly to suboptimal results because of degeneracy. For completeness, we provide the running time of the considered algorithms. ALS without line search ran 1000 iterations within 20.5 seconds, while EPC completed the rank-1 tensor correction in 188.3 seconds. After the correction, ALS convereged quickly in 74.6 seconds.

Example 2 (The case when factor matrices have highly collinear columns): In this example, we decomposed cubic tensors with size and rank, respectively, given by $I_n = 4$ and $R = 5$, $I_n = 7$ and $R = 10$, $I_n = 12$ and $R = 15$. We generated the factor matrices such that their first I_n loading components were highly correlated using the subroutine "`gen_matrix`" in the TENSORBOX [37]. The results consistently confirmed for 150 independent runs, and for each run, the parameteres were initialalized randomly.

Results for the noise-free cases are compared in Fig. 4. *Success ratio at a specific error, e.g., 10^{-6}, is the percentage of independent runs in which approximation error achieved by an algorithm differs from the best error by less than 10^{-6}.* For these hard decomposition scenarios, the fLM algorithm could explain the tensors with a relative error of 10^{-6} in about 57% of independent runs for the tensors of size $5 \times 5 \times 5$, but in less than 30% of the runs for the tensors of bigger sizes $7 \times 7 \times 7$ and $12 \times 12 \times 12$. In most of the tests, the fLM algorithm got stuck in local minima with a relative error of around 10^{-3}. However, when using either with ACEP or with the SQP method for ECP (SCEP), the success ratios were dramatically improved and exceeded 96% for the relative error of 10^{-6}.

For the same tensors, we applied algorithms for the bounded CPD. The bound of the norm of rank-1 tensors was adjusted during the iteration process. The BALS achieved higher success



(a) $I = 4$, $R = 5$ (b) $I = 7$, $R = 10$ (c) $I = 12$, $R = 15$

Fig. 4. Decomposition of the noise-free tensors in Example 2. (top) Success ratios of the considered algorithms. A relative error of 10^{-6} is considered perfect for decomposition of a noise-free tensor. (bottom) Numbers of iterations of algorithms needed to attain the best relative error.

ratios than fLM for tensors with dimension $I_n = 7$ and $I_n = 12$. It could explain the tensors with a relative error of 10^{-5} in 60–70% of the runs. The BSPQ achieved a much higher success ratio than the BALS.

In another assessment, we compared the number of iterations of algorithms needed to achieve the best relative error. For example, in order to achieve a performance near the best relative error with an error of 10^{-6}, the fLM algorithm might need a thousand of iterations, while with ACEP or SCEP, this algorithm needed respectively on average only 72 and 122 iterations. BSQP required 400 iterations as shown in Fig. 4(a) for decomposition of tensors of size $4 \times 4 \times 4$. This is because the algorithm iterated to adjust the bound of the norm of rank-1 tensors.

As seen in Fig. 4, when the algorithms reached the approximation error of 10^{-4}, they quickly attained the approximation error of 10^{-8}. In total, the number of iterations of the three algorithms, fLM+ACEP, fLM+SCEP and BSQP, were on the same order and comparable. In summary, the BSQP, Interior Point method for bounded norm constrained CPD (BITP) and fLM with EPC explained the noise-free tensors with a nearly perfect accuracy in less than 300 iterations.

For the test cases with noisy tensors, we added some small perturbation to the noise-free tensors. Fig. 5 illustrates the success ratio and the number of iterations to achieve the best relative error. The fLM algorithm attained a relative error which differs from the best by less than 10^{-6} in 47% and 18% of independent runs for the tensors with dimension of $I_n = 4$ and $I_n = 7$, respectively, while BSQP met the same accuracy level in 67% and 20% of the runs. The results confirm that the EPC method, either ACEP or SCEP, gained the success ratio of the fLM up to 79% and 42%, respectively, while this algorithm demanded a lower number of iterations than fLM.

Example 3 (Decomposition of block tensors): This example was inspired by the block-term decomposition [39] of the tensors which had rank exceeding the dimensions, and highly collinear loading components. We constructed the tensors from two blocks of size $6 \times 6 \times 6$, each of rank 6, and collinearity degrees among the loading components were within a range of $[0.95, 0.999]$

$$\mathcal{Y} = \mathcal{I} \times_1 \mathbf{U}_{1,1} \times_2 \mathbf{U}_{1,2} \times_3 \mathbf{U}_{1,3} + \mathcal{I} \times_1 \mathbf{U}_{2,1} \times_2 \mathbf{U}_{2,2} \times_3 \mathbf{U}_{2,3},$$

where $\mathcal{I}$ represents the diagonal tensor. Our experience is that such tensors are challenging for most existing CP techniques. We ran the fastALS algorithm in 10 iterations to generate initial values.

The fLM did not complete the decomposition for the noise-free tensors within the error range of 10^{-6} even after 3000 iterations as seen in Fig. 6 (bottom). The reason is that the norm of estimated rank-1 tensors increased to relatively large values, on average around 3994.3. Using the EPC methods, i.e., ACEP or SCEP, we reduced the norm to 11.8. By this way, the fLM algorithm converged in a few hundreds of iterations as illustrated in Fig. 6 (bottom) for one run of the decomposition. Similar to the fLM algorithm, RALS [21] and NLS [38] algorithms failed for this data even with 10000 iterations. Due to this, we did not consider these two algorithms for further analysis.

In Fig. 6(top), "fLM+SCEP+fLM" represents the process of three stages: running fLM until it stopped, then applying SCEP to correct the rank-1 tensors, and finally running the fLM again.

The results confirm that the proposed correction method worked efficiently. When using EPC, the fLM could complete

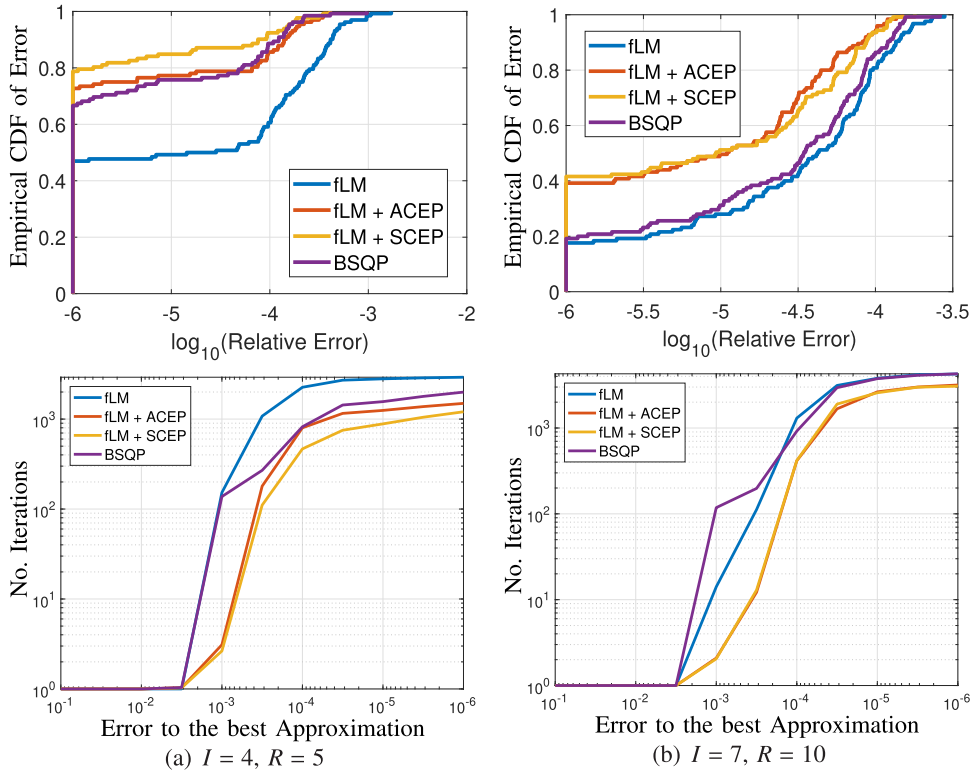


Fig. 5. Performance of CPD of the noisy tensors in Example 2. (top) Success ratios of the considered algorithms. (bottom) Numbers of iterations of algorithms to attain the best relative error.

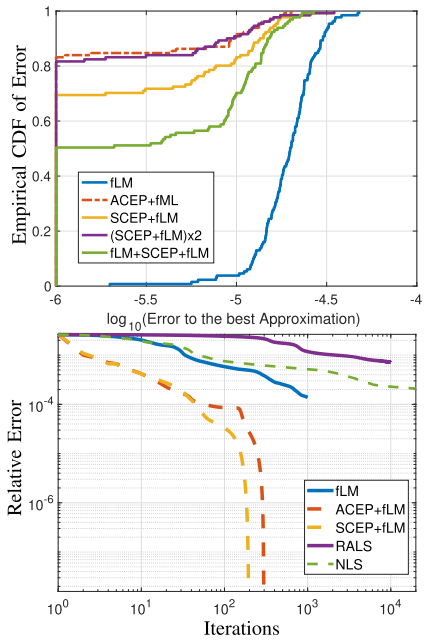


Fig. 6. (top) Success ratios of the fLM algorithm with and without EPC in Example 3. (bottom) Relative errors in one run of the decomposition.

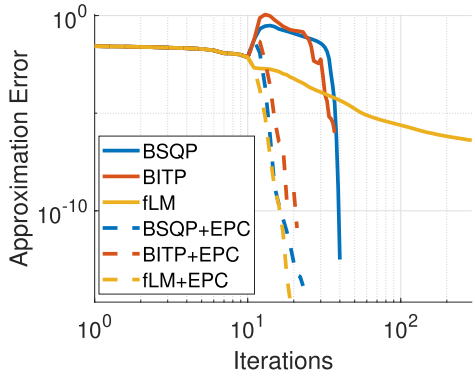


Fig. 7. Comparison of the approximation errors for various CPD algorithms for the multiplicative tensors of size $9 \times 9 \times 9$ which has rank of $R = 23$.

the decomposition in more than 80% of the runs. The results were reported over 130 independent runs.

Example 4 (Decomposition of tensor for multiplication of two matrices of size 3×3): In this example, we compare the performance of algorithms for CPD with and without EPC and algorithms for CPD with a bounded rank-1 tensor norm. The tensor considered in this example is the multiplication tensor in the case of two matrices of size 3×3. This tensor is of size $9 \times 9 \times 9$, contains only zeros and ones, and obeys

$$\text{vec}\,(\mathbf{AB}) = \mathcal{Y} \times_1 \text{vec}\left(\mathbf{A}^T\right)^T \times_2 \text{vec}\left(\mathbf{B}^T\right)^T ,$$

where $\mathbf{A}$ and $\mathbf{B}$ are of size 3×3. The tensor is considered of rank-$R = 23$. In [22], we developed an LM algorithm to update the vector of parameters which is assumed to be on a ball with a prescribed diameter.

Decomposition of such tensor using ALS or LM often gets stuck in false local minima or requires a huge number of

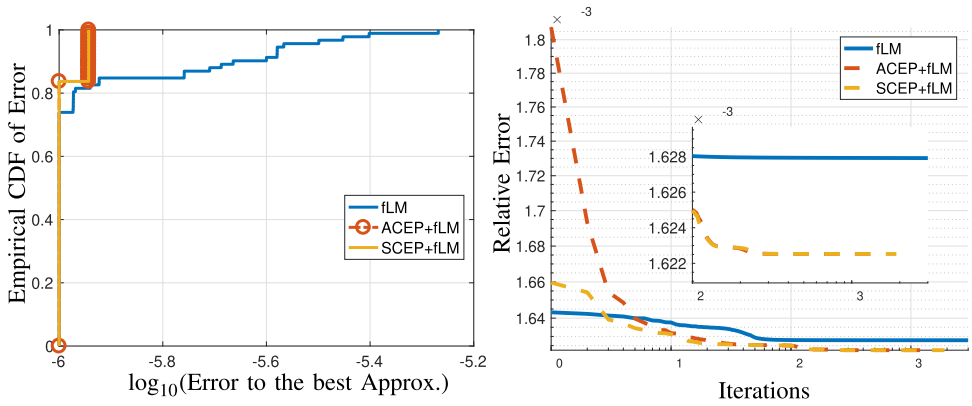


Fig. 8. (a) Comparison of the success ratios of the fLM algorithm with and without EPC in the decomposition of the TV-ratings data. (b) Illustration of the changes of the relative errors in one run of the estimation. The fLM got stuck in a false local minimum after at most 100 iterations.

iterations. This is because the norm of estimated rank-1 tensors is significantly large.

For this case, we first ran the standard ALS/fLM algorithm in 10 iterations, and used the outcome to initialize the BSQP and BITP algorithm for the bounded CPD. The bound of the rank-1 tensor norm was set to $\epsilon = 15$. The results show that the two algorithms converged after a few dozens of iterations. This is much faster than running fLM without EPC.

In another comparison, we applied EPC to the tensors estimated using the fLM algorithm. The corrected tensor was then used to initalize the BSQP, BITP, and fLM algorithms [34]. The results are compared in Fig. 7. For this later test, the three algorithms converged after only 10 iterations.

Example 5 (Decomposition of the TV-ratings data [8]): We decomposed the TV-ratings data [8] which comprises 16 rating scales $\times$ 15 American TV shows $\times$ 30 subjects. This data is well known to illustrate the degeneracy in CPD for example with the rank $R = 2$, 3 or 4 [13], [15]. Here, we compared the fLM algorithm with and without the EPC for the decomposition of rank-$R = 10$. We first ran the ALS algorithm in 100 iterations to generate the initial parameters, then executed the fLM algorithm. For the EPC method, the bound of the approximation error was set to 1.01 times the approximation error of the initial point. The success ratios of the considered algorithms are plotted in Fig. 8(a). In 74.6% of the runs, the relative errors obtained by the fLM were very close to the best results, with a difference of less than 10^{-6}. The success ratio of fLM was considerably improved after executing the EPC either with ACEP or SCEP (see Fig. 8(a)). In Fig. 8(b), we illustrate the relative errors of algorithms as a function of the number of iterations in one run. The fLM started with a lower error but got stuck in a false local minimum after 100 iterations. Since the error bound was set to higher than the approximation error, the fLM with EPC started with a higher relative error, but in the final, this algorithm achieved a lower approximation error as seen in Fig. 8(b).

Example 6 (Factorization of time-frequency representation of Event-related EEG): In this example, we decomposed a fourth-order tensor consisting of 28 inter-trial phase coherence

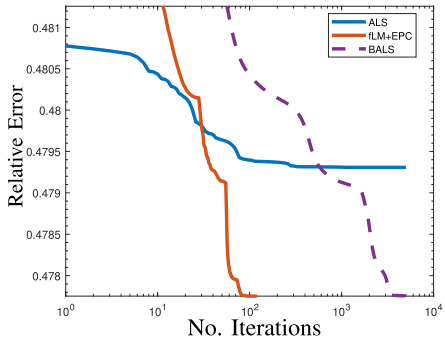
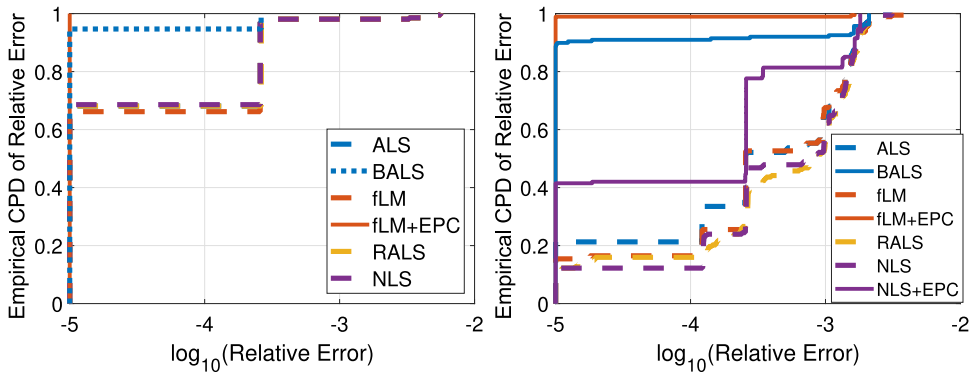


Fig. 9. Relative errors of three algorithms in one run of the decomposition of the IPTC tensor with a rank $R = 15$.

(ITPC) measurements of EEG signals for 14 subjects during a proprioceptive pull of the left and right hands [26]. The ITPC dataset was represented as a 4-way tensor of 28 measurements $\times$ 61 frequency bins $\times$ 64 channels $\times$ 72 time frames.

We approximated the ITPC tensor by low-rank tensors of ranks $R = 10$ and 15. Interpretation of the results can be found in [26]. Algorithms used the same initial values and stopped when differences of successive relative errors $\varepsilon = \dfrac{\|\mathcal{Y} - \widehat{\mathcal{Y}}\|_F}{\|\mathcal{Y}\|_F}$ were lower than 10^{-8}, or until the maximum number of iterations (5000) was achieved. For each run, we initialized the ALS algorithm by 10 random tensors, and ran it in 10 iterations. The estimated tensor yielding the smallest approximation error was then used to initialize BALS and fLM+EPC. Similar to the previous example, we set the approximation error bound slightly higher than the initial approximation error. This makes BALS and fLM+EPC starting with higher approximation errors than ALS, but these algorithms could find feasible points in a few iterations.

Fig. 9 compares convergence behaviour of the considered algorithms. The ALS got stuck in a false local minimum after around 400 iterations, and failed to improve the approximation error, while norms of estimated rank-1 tensors increased significantly to a value of 10^8. Both BALS and fLM+EPC converged



(a) $R = 10$. (b) $R = 15$.

Fig. 10. Success ratios of the developed algorithm and other three algorithms in the decomposition of the fourth order ITPC tensor in Example 6.

to tensors with lower approximation errors. Similar results were observed in more than 99% of the runs for the fLM+EPC. Significantly higher success ratios of the developed algorithms were confirmed over 200 independent runs as shown in Fig. 10. For the decomposition with the rank of $R = 10$, the ALS algorithm succeeded in 69.2% of the runs, i.e., the difference between relative approximation errors achieved by this algorithm and the best lower than 10^{-6}. With controlling the bound of rank-1 tensors, the BALS algorithm had a higher success ratio of 94%, while the fLM with EPC achieved a nearly perfect success ratio of 99.3%.

The performance of the ALS was even worse when the decomposition was with a higher rank ($R = 15$). Fig. 10(b) shows that the algorithm succeeded in only 19.2% of the runs. In contrast, our BALS achieved a much better success ratio of 83%, and the fLM+EPC still succeeded with a very high ratio of 98.74%. As in Examples 1 and 3, both RALS [21] and NLS[38] achieved relatively low success ratios, comparable with those of ALS and fLM. The performances were obtained after running RALS 5000 iterations, and NLS 2000 iterations. By running EPC after NLS, we improved the performance of NLS.

Example 7 [Image denoising.]: This example demonstrates an application of the EPC method in image denoising. Given that the intensities of pixels in a small window are highly correlated, our method was able to learn hidden structures which represent relations between small patches of pixels. For a color image $\mathcal{T}$ degraded by additive Gaussian noise, we constructed 5-th order tensors at location-(r, c), $\mathcal{Y}_{r,c}$, of a size $w \times w \times 3 \times (2d+1) \times (2d+1)$, comprising $(2d+1)^2$ blocks, $\mathcal{Y}_{r,c}(:,:,:,d+1+i,d+1+j) = \mathcal{T}_{r+i,c+j}$, around the patch of size $w \times w \times 3$ $\mathcal{T}_{r,c} = \mathcal{T}(r : r+w-1, c : c+w-1, :)$, where $i, j = -d, \ldots, 0, \ldots, d$, and d represents the neighbour width. Each tensor $\mathcal{Y}_{r,c}$ was then approximated by the decomposition in (4) using the EPC method, where $\delta^2 = 3\sigma^2 w^2 (2d+1)^2$, and σ the noise level. Finally, we used the approximated tensors, $\hat{\mathcal{Y}}_{r,c}$, to reconstruct the patches and the entire image.

Fig. 11 shows six benchmark color images of size $256 \times 256 \times 3$ used in our simulations. We corrupted them by additive white Gaussian noise at SNR $= 10$ dB. Block tensors were of sizes $8 \times 8 \times 3$ ($w = 8$) and the search area of width $d = 3$. We applied the DCT spatial filtering as a preprocessing before the tensorization. We started the decomposition with the tensor rank of 8, and adjusted it to attain the error bound.

Fig. 11. Six images of the size 256×256 are used in Example 7.

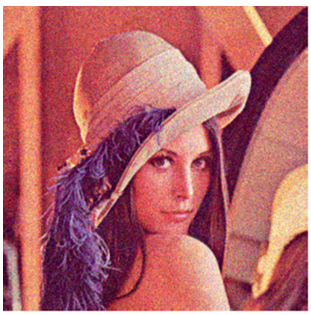
For the constrained approximation problem of $\mathcal{Y}_{r,c}$, we applied several tensor decompositions, including the Tensor-Train (TT-SVD) [40], the Tucker approximation (TKA) with a predefined approximation error [41], and the Bayesian Robust tensor factorisation (BRTF) for low-rank CP decomposition [42]. In addition, we recovered the image with sparsity constraints using a dictionary of 256 atoms learnt by K-SVD [43]. For this method, color image layers were flattened into an array of size 256×768. The dictionary was learnt for patches of size 8×8.

Multilinear ranks in TKA are determined based on inspecting the singular values of projection matrices to keep the approximation error within an error bound defined by the noise level [41]. The TT-SVD also determined the ranks of the core tensors based on the relevant singular values and their explained variance. The BRTF is a CPD method which can prune out irrelevant rank-1 tensors. We note that although these tensor decomposition methods can determine appropriate models using their own criteria, none of them can preserve the approximation error at a prescribed error, which in the denoising application is the noise level. Their approximated tensors can attain lower approximation errors than the required level. This in practice implies an overestimation, and thereby the reconstructed results might be deteriorated.

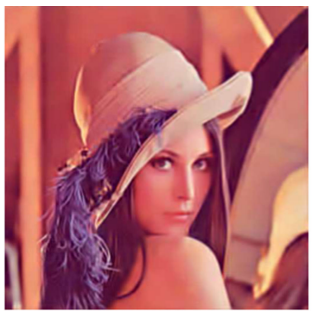
Table III compares performances of the tensor-based decomposition methods and K-SVD. In all the simulations, EPC outperformed the noise removal methods based on the other tensor decompositions and the dictionary learning method K-SVD. The reconstructed images of some images are illustrated in Figs. 12 and 13. The closer inspection in Fig. 12 shows that the EPC-based denoising method suppressed the noise but preserved the complex structures in the reconstructed images.

TABLE III
THE PERFORMANCE COMPARISON OF ALGORITHMS CONSIDERED IN EXAMPLE 7 IN TERMS OF PSNR (DB) AND SSIM FOR IMAGE DENOISING WHEN SNR = 10 dB

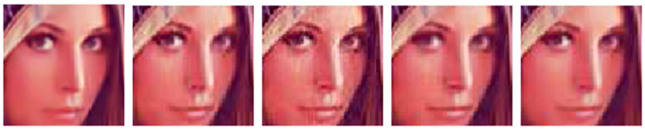
Algorithms	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
	Lena		Tiffany		Barbara	
EPC	**33.34**	**0.924**	**36.01**	**0.932**	**33.73**	**0.927**
TT-SVD	32.68	0.892	35.47	0.913	33.04	0.901
Tucker	32.74	0.919	35.47	0.926	32.92	0.919
BRTF	32.07	0.840	35.07	0.888	33.10	0.899
K-SVD	32.72	0.908	35.61	0.928	32.64	0.908
	Pepper		Pens		House	
EPC	**33.15**	**0.926**	**32.20**	**0.896**	**35.41**	**0.895**
TT-SVD	32.07	0.861	31.61	0.884	34.38	0.877
Tucker	32.23	0.917	31.27	0.884	34.40	0.885
BRTF	31.42	0.825	31.82	0.877	33.67	0.823
K-SVD	32.60	0.918	31.14	0.862	34.67	0.881



(a) Noisy image at SNR = 10 dB

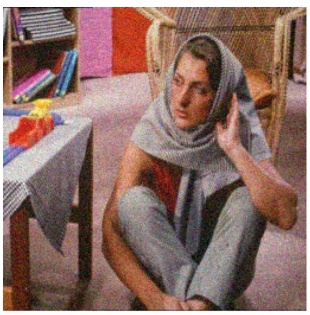


(b) EPC, SSIM = 0.924



(c) From left to right, EPC(SSIM = 0.924), Tucker(0.919), TT-SVD(0.892), BRTF(0.840) and K-SVD(0.908)

Fig. 12. The Lena image corrupted by noise at 10 dB SNR, and a closer inspection of the reconstructed images in Example 7.



(a) Noisy image at SNR = 10 dB

(b) EPC, SSIM = 0.927

(c) From left to right, EPC(SSIM = 0.927), Tucker(0.919), TT-SVD(0.901), BRTF(0.899) and K-SVD(0.908)

Fig. 13. The "Barbara" image corrupted by noise at 10 dB SNR, and a closer inspection of the reconstructed images in Example 7.

V. CONCLUSIONS

For difficult scenarios of the CP tensor decomposition, when large loading components may cancel each other, we propose to seek new decompositions with the same approximation error but with a minimum norm of the rank-one components. In particular, we derive solutions to two constrained optimization problems, one for the error preserving correction method, and another one for the bounded CPD. The factor matrices in the two optimization problems can be updated in closed-form expressions in an alternating update scheme through solving Spherical Constrained Quadratic Programming. In addition, the SQP-based all-at-once algorithms have been developed with a low complexity for the inversion of the Hessian matrices. Moreover, we presented a relation between the new alternating algorithms and the standard ALS algorithm. In simulations, we confirmed the efficiency of the proposed algorithms for the CD decomposition of tensors with rank exceeding the tensor dimensions (multiplication tensors) and on tensors with highly collinear rank-one components. The EPC method is particularly suited for CPD with a target error bound. Demonstration for image denoising shows that the proposed method achieved better performances than the method based on other tensor decompositions and the K-SVD dictionary learning method. Finally, the proposed algorithms are implemented in the Matlab package TENSORBOX which is available online at website of the first author.

APPENDIX A
LINEAR REGRESSION WITH A BOUND CONSTRAINT

In this appendix, we summarize, for the sake of completeness, a few known results on the linear regression with a quadratic constraint. Proofs of the propositions can be found in [31], [44].

The linear regression problem with a constraint on the regression error is stated as

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|^2 \quad \text{s.t.} \quad \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\| \leq \delta, \tag{39}$$

where $\boldsymbol{y}$ is a vector of length I of dependent variables, $\mathbf{A}$ is a regressor matrix of a size $I \times K$, and a nonnegative regression bound δ.

It is obvious that if $\delta \geq \|\boldsymbol{y}\|$, then the zero vector $\boldsymbol{x} = 0$ is a minimizer of (39). Therefore, in order to achieve a meaningful regression, the regression bound δ needs to be in a specific range defined by Lemma 1.

Lemma 1 (Range of the bound δ): The problem (39) has a minimizer of nonzero entries when

$$\|\mathbf{\Pi}_{\mathbf{A}}^{\perp}\, \boldsymbol{y}\| \leq \delta < \|\boldsymbol{y}\|, \tag{40}$$

where $\mathbf{\Pi}_{\mathbf{A}}^{\perp}$ is an orthogonal complement of the column space of $\mathbf{A}$.

For simplicity, we assume that $\mathbf{A}$ is a full rank matrix, otherwise, we solve the problem with a compressed regressor matrix with a smaller bound

$$\min \quad \|\boldsymbol{x}\|^2 \quad \text{s.t.} \quad \|\hat{\boldsymbol{y}} - \hat{\mathbf{A}}\boldsymbol{x}\| \leq \hat{\delta} \tag{41}$$

where $\hat{\boldsymbol{y}} = \mathbf{U}^T \boldsymbol{y}$, $\hat{\mathbf{A}} = \mathbf{U}^T \mathbf{A}$, and $\hat{\delta}^2 = \delta^2 - \|\mathbf{\Pi}_{\mathbf{A}}^{\perp}\, \boldsymbol{y}\|^2$.

We show that the inequality sign in (39) can be replaced by the equality sign.

Lemma 2: The minimizer of (39) is equivalent to the minimizer of the following optimization problem

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|^2 \quad \text{s.t.} \quad \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\| = \delta. \tag{42}$$

We present an algorithm when the matrix of regressors $\mathbf{A}$ is of full column rank, $K \leq I$.

Let $\mathbf{A} = \mathbf{U}\,\mathrm{diag}(\boldsymbol{s})\mathbf{V}^T$ be an SVD of $\mathbf{A}$, where $\mathbf{V}$ is an orthonormal matrix of size $K \times K$, and $\boldsymbol{s} = [s_1, \ldots, s_K] > 0$. Hence $\boldsymbol{\Pi}_{\mathbf{A}}^{\perp} = \mathbf{I} - \mathbf{U}\,\mathbf{U}^T$.

Let $\hat{\boldsymbol{y}} = \mathbf{U}^T\boldsymbol{y}$, $\hat{\delta} = \sqrt{\delta^2 - \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp}\,\boldsymbol{y}\|^2}$ and $\boldsymbol{z} = \frac{1}{\hat{\delta}}(\hat{\boldsymbol{y}} - \mathrm{diag}(\boldsymbol{s})\mathbf{V}^T\boldsymbol{x})$, then

$$\begin{aligned}
\boldsymbol{x} &= \mathbf{V}\,\mathrm{diag}(\boldsymbol{s}^{-1})(\hat{\boldsymbol{y}} - \hat{\delta}\boldsymbol{z}) \\
\|\boldsymbol{x}\|_F^2 &= (\hat{\boldsymbol{y}} - \hat{\delta}\boldsymbol{z})^T\,\mathrm{diag}(\boldsymbol{s}^{-2})(\hat{\boldsymbol{y}} - \hat{\delta}\boldsymbol{z}) \\
\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|^2 &= \|\mathbf{U}^T\boldsymbol{y} - \mathrm{diag}(\boldsymbol{s})\mathbf{V}^T\boldsymbol{x}\|_F^2 + \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp}\,\boldsymbol{y}\|^2 \\
&= \hat{\delta}^2\,\|\boldsymbol{z}\|^2 + \|\boldsymbol{\Pi}_{\mathbf{A}}^{\perp}\,\boldsymbol{y}\|^2\,.
\end{aligned}$$

By this reparameterization, the optimization problem (42) becomes a QP over a sphere which has a closed-form solution, e.g., see [31], [33]

$$\min_{\boldsymbol{z}} \quad \boldsymbol{z}^T\,\mathrm{diag}(\hat{\delta}\boldsymbol{s}^{-2})\boldsymbol{z} - 2\,\hat{\boldsymbol{y}}^T\,\mathrm{diag}(\boldsymbol{s}^{-2})\boldsymbol{z} \;\text{ s.t. }\; \boldsymbol{z}^T\boldsymbol{z} = 1. \tag{43}$$

Appendix B
A Simplification Method For SCQP With Identical Eigenvalues

We consider a QP problem over a sphere

$$\min \;\; \frac{1}{2}\,\tilde{\boldsymbol{x}}^T\,\mathrm{diag}(\boldsymbol{s})\,\tilde{\boldsymbol{x}} + \boldsymbol{c}^T\tilde{\boldsymbol{x}}\,,\;\; \text{s.t.} \quad \tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{x}} = 1, \tag{44}$$

where $\boldsymbol{c}^T\boldsymbol{c} = 1$, and $\boldsymbol{s} = [s_1 = 1 \leq s_2 \leq \cdots \leq s_K]$.

We denote J the number of distinct eigenvalues, $\tilde{\boldsymbol{s}} = [\tilde{s}_1 = 1 < \tilde{s}_2 < \cdots < \tilde{s}_J]$, over a set of K eigenvalues, s_k, in (44), and classify $\boldsymbol{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_J]$ into J sub-vectors, and each $\boldsymbol{c}_j$ consists of entries c_k such that $s_k = \tilde{s}_j$, i.e., $\boldsymbol{c}_j = [c_{k\in\mathcal{I}_j}]$, where $\boldsymbol{I}_j = \{k : s_k = \tilde{s}_j\}$. In addition, we define a vector

$$\tilde{\boldsymbol{c}} = [\|\boldsymbol{c}_1\|, \|\boldsymbol{c}_2\|, \ldots, \|\boldsymbol{c}_J\|]\,. \tag{45}$$

Then the following relation holds.

Lemma 3: The minimizer of (44) can be deduced from the minimizer of the SCQP with distinct eigenvalues

$$\min \;\; \frac{1}{2}\,\boldsymbol{z}^T\,\mathrm{diag}(\tilde{\boldsymbol{s}})\,\boldsymbol{z} + \tilde{\boldsymbol{c}}^T\boldsymbol{z} \quad \text{s.t. } \boldsymbol{z}^T\boldsymbol{z} = 1\,,$$

as follows

- For non zero $\tilde{c}_j$, $\boldsymbol{x}_{\mathcal{I}_j} = \frac{z_j}{\tilde{c}_j}\,\boldsymbol{c}_j$
- If $c_1 = 0$ and $d^2 = \sum_{j=2}^{J} \dfrac{\tilde{c}_j^2}{(\tilde{s}_j - 1)^2} \leq 1$, $\boldsymbol{x}_{\mathcal{I}_1}$ can be arbitrary vectors on the ball $\|\boldsymbol{x}_{\mathcal{I}_1}\|^2 = 1 - d^2$,
- Otherwise for zeros $\tilde{c}_j$, $\boldsymbol{x}_{\mathcal{I}_j}$ all are zeros.

Proof: We consider a simple case when some eigenvalues are identical, e.g., $s_1 = s_2 = \cdots = s_L < s_{L+1} < \cdots < s_K$. If $\boldsymbol{c}_{1:L}$ are all zeros, the objective function is independent of $\tilde{\boldsymbol{x}}_{1:L} = [\tilde{x}_1, \tilde{x}_2 \ldots, \tilde{x}_L]$, hence, $\tilde{\boldsymbol{x}}_{1:L}$ can be any point on the ball $\|\tilde{\boldsymbol{x}}_{1:L}\|^2 = d^2 = 1 - \sum_{k=L+1}^{K}\tilde{x}_k^2$. Otherwise, $\tilde{\boldsymbol{x}}_{1:L}$ is a minimizer to the constrained linear programming while fixing the other parameters $\tilde{x}_{L+1}, \ldots, \tilde{x}_K$

$$\min \quad \boldsymbol{c}_{1:L}^T\,\tilde{\boldsymbol{x}}_{1:L} \quad \text{s.t.} \quad \|\tilde{\boldsymbol{x}}_{1:L}\| = d$$

which yields $\tilde{\boldsymbol{x}}_{1:L} = \frac{-d}{\|\boldsymbol{c}_{1:L}\|}\boldsymbol{c}_{1:L}$. For both cases, we can define

$$\begin{aligned}
\boldsymbol{z} &= [-d, \tilde{x}_{L+1}, \ldots, \tilde{x}_K], \\
\tilde{\boldsymbol{c}} &= [\|\boldsymbol{c}_{1:L}\|, c_{L+1}, \ldots, c_K], \\
\tilde{\boldsymbol{s}} &= [s_1, s_{L+1}, \ldots, s_K],
\end{aligned}$$

and perform a reparameterization to estimate $\boldsymbol{z}$ from a similar constrained QP but with distinct eigenvalues $\tilde{\boldsymbol{s}}$

$$\min \;\; \frac{1}{2}\,\boldsymbol{z}^T\,\mathrm{diag}(\tilde{\boldsymbol{s}})\,\boldsymbol{z} + \tilde{\boldsymbol{c}}^T\boldsymbol{z} \quad \text{s.t. } \boldsymbol{z}^T\boldsymbol{z} = 1\,.$$

Similarly, we can convert (44) to a problem with $\tilde{s}_1 < \tilde{s}_2 < \cdots < \tilde{s}_J$. Now based on the fact that for the zero coefficients $\tilde{c}_j$, the corresponding $z_j^{\star}$ will also be zeros, except for only the case $c_1 = 0$ and $1 \geq d^2$ [31]. This concludes the proof. ■

Appendix C
SCQP With Matrix-Variates

We consider an SCQP for a matrix-variate $\mathbf{X}$ of size $I \times R$ given in the form of

$$\min f(\mathbf{X}) = \frac{1}{2}\;\mathrm{tr}(\mathbf{X}^T\mathbf{Q}\mathbf{X}) + \mathrm{tr}(\mathbf{B}^T\mathbf{X}) \quad \text{s.t. } \|\mathbf{X}\|_F^2 = 1\,, \tag{46}$$

where $\mathbf{Q}$ is a positive semidefinite matrix of size $I \times I$ and $\mathbf{B}$ is of size $I \times R$. The objective function can be rewritten in a similar form to (44) as

$$f(\mathbf{X}) = \;\frac{1}{2}\boldsymbol{x}^T\,(\mathrm{diag}(\boldsymbol{\sigma}) \otimes \mathbf{I}_R)\boldsymbol{x} + \boldsymbol{v}^T\,\boldsymbol{x},$$

where $\boldsymbol{x} = \mathrm{vec}\,(\mathbf{X}^T\mathbf{U})$, $\boldsymbol{v} = \mathrm{vec}\,(\mathbf{B}^T\mathbf{U})$ and $\mathbf{Q} = \mathbf{U}\,\mathrm{diag}(\boldsymbol{\sigma})\mathbf{U}^T$ is an EVD of $\mathbf{Q}$. Due to the Kronecker product, each eigenvalue σ_i, $i = 1, \ldots, I$, is replicated R times. Let $\boldsymbol{z}^{\star}$ of length I be a (unique) minimizer of an SCQP

$$\min \frac{1}{2}\,\boldsymbol{z}^T\,\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{z} + \boldsymbol{c}^T\,\boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{z}^T\boldsymbol{z} = 1\,, \tag{47}$$

where $\boldsymbol{c} = [c_1, \ldots, c_I]$, $c_i = \|\mathbf{B}^T\boldsymbol{u}_i\|$. According to Lemma 3, for a nonzero coefficient c_i, we have $\boldsymbol{x}_i = \frac{z_i}{c_i}\mathbf{B}^T\boldsymbol{u}_i$, otherwise, $\boldsymbol{x}_i$ can be any vector on the ball $\boldsymbol{x}_i^T\,\boldsymbol{x}_i = z_i^2$ for a zero vector $\mathbf{B}^T\boldsymbol{u}_i$.

Appendix D
Gradient and Hessian of the Objective Function $f(\boldsymbol{\theta})$ in (16)

Let $\boldsymbol{\beta}_n = [\boldsymbol{u}_1^{(n)T}\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_R^{(n)T}\boldsymbol{u}_R^{(n)}]$ and $\boldsymbol{\beta}_{-n} = \circledast_{k\neq n}\,\boldsymbol{\beta}_{n=1}^N$, $\boldsymbol{\beta} = \circledast_n\,\boldsymbol{\beta}_n$. The gradient $\boldsymbol{g}_f$ and Hessian

$\mathbf{H}_f$ of the objective function w.r.t. to $\mathbf{U}^{(n)}$ are given by

$$\begin{aligned}\boldsymbol{g}_f &= \left[\ldots, \operatorname{vec}\left(\frac{\partial f}{\partial \mathbf{U}^{(n)}}\right)^T, \ldots\right]^T \\ &= \left[\ldots, \operatorname{vec}\left(\mathbf{U}^{(n)}\operatorname{diag}(\boldsymbol{\beta}_{-n})\right)^T, \ldots\right]^T \\ \mathbf{H}_f &= \nabla^2 f = \mathbf{D} + 2\,\mathbf{V}\,\mathbf{F}\,\mathbf{V}^T,\end{aligned}$$

where $\mathbf{F} = [\mathbf{F}_{n,m}]$ is an $N \times N$ partitioned matrix of matrices $\mathbf{F}_{n,m}$ with $\mathbf{F}_{n,n} = 0$ and $\mathbf{F}_{n\neq m} = \operatorname{diag}(\boldsymbol{\beta}_{-(n,m)})$, and

$$\begin{aligned}\mathbf{D} &= \operatorname{diag}([\boldsymbol{\beta}_{-1} \otimes 1_{I_1}, \ldots, \boldsymbol{\beta}_{-N} \otimes 1_{I_N}]), \\ \mathbf{V} &= \operatorname{blkdiag}(\mathbf{V}_1, \ldots, \mathbf{V}_N), \\ \mathbf{V}_n &= \operatorname{blkdiag}(\boldsymbol{u}_1^{(n)}, \ldots, \boldsymbol{u}_R^{(n)}).\end{aligned}$$

The Hessian $\mathbf{H}_f$ can also be represented in an equivalent form of a block diagonal matrix and a rank-R adjustment

$$\begin{aligned}\mathbf{H}_f = \operatorname{blkdiag}(\ldots, \operatorname{diag}(\boldsymbol{\beta}_{-n} \otimes 1_{I_n}) - 2\,\tilde{\mathbf{V}}_n\,\operatorname{diag}(\boldsymbol{\beta})\,\tilde{\mathbf{V}}_n^T, \ldots) \\ + 2\,\tilde{\mathbf{V}}\,\operatorname{diag}(\boldsymbol{\beta})\,\tilde{\mathbf{V}}^T, \qquad (48)\end{aligned}$$

where $\tilde{\mathbf{V}}_n = \mathbf{V}_n \operatorname{diag}(1 \oslash \boldsymbol{\beta}_n)$ and $\tilde{\mathbf{V}} = [\tilde{\mathbf{V}}_1^T, \ldots, \tilde{\mathbf{V}}_N^T]^T$ is of size $R(\sum_n I_n) \times R$.

APPENDIX E
GRADIENT AND HESSIAN OF THE CONSTRAINT FUNCTION $c(\boldsymbol{\theta})$ IN (16)

According to Theorem 2 [34], the gradient and Hessian of the constraint function $c(\boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$ are given by

$$\begin{aligned}\boldsymbol{g}_c &= \left[\ldots, \operatorname{vec}\left(\mathbf{U}^{(n)}\boldsymbol{\Gamma}_{-n} - \mathbf{Y}_{(n)}\left(\bigodot_{k\neq n} \mathbf{U}^{(n)}\right)\right)^T, \ldots\right]^T \\ \mathbf{H}_c &= \mathbf{G} + \mathbf{Z}\mathbf{K}\mathbf{Z}^T,\end{aligned}$$

where

$$\begin{aligned}\mathbf{G} &= \operatorname{blkdiag}(\boldsymbol{\Gamma}_{-n} \otimes \mathbf{I}_{I_n}), \\ \mathbf{Z} &= \operatorname{blkdiag}(\ldots, \mathbf{I}_R \otimes \mathbf{U}^{(n)}, \ldots), \\ \mathbf{K} &= [\mathbf{K}_{n,m}], \mathbf{K}_{n,n} = 0, \mathbf{K}_{n\neq m} = \operatorname{dvec}(\boldsymbol{\Gamma}_{-(n,m)}).\end{aligned}$$

The Hessian $\mathbf{H}_c$ can also be expressed as [35]

$$\mathbf{H}_c = \operatorname{blkdiag}(\boldsymbol{\Gamma}_{-n} \otimes \mathbf{I}_{I_n} - \tilde{\mathbf{Z}}_n\,\boldsymbol{\Psi}\,\tilde{\mathbf{Z}}_n^T) + \tilde{\mathbf{Z}}\,\boldsymbol{\Psi}\,\tilde{\mathbf{Z}}^T \qquad (49)$$

where $\tilde{\mathbf{Z}} = [\tilde{\mathbf{Z}}_n]$, $\tilde{\mathbf{Z}}_n = (\mathbf{I}_R \otimes \mathbf{U}^{(n)})\operatorname{dvec}(1 \oslash \boldsymbol{\Gamma}_n)$, $\boldsymbol{\Psi} = \mathbf{P}_{R,R}\operatorname{dvec}(\boldsymbol{\Gamma})$. Note that $\tilde{\mathbf{V}} = \tilde{\mathbf{Z}}(:, [1, R+1, \ldots, R^2])$ and $\boldsymbol{\beta} = \operatorname{diag}(\boldsymbol{\Gamma})$, $\boldsymbol{\beta}_{-n} = \operatorname{diag}(\boldsymbol{\Gamma}_{-n})$.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the Referees for their time and efforts to review this paper, and for very constructive comments which helped to improve the quality and presentation of the paper.

REFERENCES

[1] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," *Found. Trends Mach. Learn.*, vol. 9, no. 4/5, pp. 249–429, 2016.

[2] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5423–5436, Oct. 2016.

[3] B. C. Mitchell and D. S. Burdick, "Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies," *J. Chemometrics*, vol. 8, pp. 155–168, 1994.

[4] W. P. Krijnen, T. K. Dijkstra, and A. Stegeman, "On the non-existence of optimal solutions and the occurrence of degeneracy in the CANDECOMP/PARAFAC model," *Psychometrika*, vol. 73, pp. 431–439, 2008.

[5] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 1084–1127, 2008.

[6] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *Proc. British Mach. Vision Conf.*, BMVA Press, 2014.

[7] V. Lebedev, Y. Ganin, M. Rakhuba, I. V. Oseledets, and V. S. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned CP-Decomposition," *CoRR*, vol. abs/1412.6553, 2014.

[8] M. E. Lundy, R. E. Harshamn, and J. B. Kruskal, "A two-stage procedure incorporating good features of both trilinear and quadrilinear Models," in *Multiway Data Analysi*. Amsterdam, The Netherlands: North Holland, 1989, pp. 123–130.

[9] R. A. Harshman and M. E. Lundy, *Data Preprocessing and the Extended Parafac Model*. New York, NY, USA: Praeger, 1984.

[10] P. Paatero, "A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis," *Chemometrics Intell. Lab. Syst.*, vol. 38, no. 2, pp. 223–242, 1997.

[11] P. Paatero, "Construction and analysis of degenerate PARAFAC models," *J. Chemometrics*, vol. 14, no. 3, pp. 285–299, 2000.

[12] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometrics*, vol. 23, no. 7–8, pp. 393–405, 2009.

[13] R. A. Harshman, "The problem and nature of degenerate solutions or decompositions of 3-way arrays," presented at Workshop on Tensor decompositions, Palo Alto, CA, USA, 2004.

[14] W. P. Krijnen, T. K. Dijkstra, and A. Stegeman, "On the non-existence of optimal solutions and the occurrence of "degeneracy" in the Candecomp/Parafac model," *Psychometrika*, vol. 73, pp. 431–439, 2008.

[15] A. Stegeman, "CANDECOMP/PARAFAC: From diverging components to a decomposition in block terms.," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 2, pp. 291–316, 2012.

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

[17] P. Giordani and R. Rocci, "Constrained Candecomp/Parafac via the Lasso," *Psychometrika*, vol. 78, no. 4, pp. 669–884, Oct. 2013.

[18] P. Giordani and R. Rocci, "Candecomp/Parafac with ridge regularization," *Chemometrics Intell. Lab. Syst.*, vol. 129, pp. 3–9, 2013.

[19] W. S. Rayens and B. C. Mitchell, "Two-factor degeneracies and a stabilization of PARAFAC," *Chemometrics Intell. Lab. Syst.*, vol. 38, no. 2, pp. 173–181, 1997.

[20] L.-H. Lim and P. Comon, "Nonnegative approximations of nonnegative tensors," *J. Chemometrics*, vol. 23, no. 7/8, pp. 432–441, 2009.

[21] C. Navasca, L. De Lathauwer, and S. Kindermann, "Swamp reducing technique for tensor decomposition," in *Proc. 16th Eur. Signal Process. Conf.*, 2008, pp. 1–5.

[22] P. Tichavský, A.-H. Phan, and A. Cichocki, "Numerical CP decomposition of some difficult tensors," *J. Comput. Appl. Math.*, vol. 317, pp. 362–370, 2017.

[23] Y. Wang, H.-Y. Tung, A. J. Smola, and A. Anandkumar, "Fast and guaranteed tensor decomposition via sketching," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 991–999.

[24] C. Battaglino, G. Ballard, and T. Kolda, "A practical randomized CP tensor decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 39, no. 2, pp. 876–901, 2018.

[25] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 284–295, Mar. 2016.

[26] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, "Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG," *NeuroImage*, vol. 29, no. 3, pp. 938–947, 2006.

[27] P. Tichavský, A.-H. Phan, and A. Cichocki, "Partitioned alternating least squares technique for canonical polyadic tensor decomposition," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 993–997, Jul. 2016.
[28] A.-H. Phan, P. Tichavský, and A. Cichocki, "Partitioned hierarchical alternating least squares algorithm for CP tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 2542–2546.
[29] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–151, 1995, doi: 10.1017/S0962492900002518.
[30] R. Fletcher, *Nonlinear Programming*. Hoboken, NJ, USA: Wiley, 2000, pp. 229–258.
[31] A.-H. Phan, M. Yamagishi, D. Mandic, and A. Cichocki, "Quadratic programming over ellipsoids (with applications to constrained linear regression and tensor decomposition)," arXiv:1711.04401, 2017.
[32] A.-H. Phan, P. Tichavský, and A. Cichocki, "Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4834–4846, Oct. 2013.
[33] W. Gander, G. H. Golub, and U. von Matt, "A constrained eigenvalue problem," *Linear Algebra Appl., Special Issue Dedicated to Alan J. Hoffman*, vol. 114, pp. 815–839, 1989.
[34] A.-H. Phan, P. Tichavský, and A. Cichocki, "Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 126–147, 2013.
[35] P. Tichavský, A.-H. Phan, and A. Cichocki, "A further improvement of a fast damped Gauss-Newton algorithm for CANDECOMP-PARAFAC tensor decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal, Process.*, 2013, pp. 5964–5968.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. 25*, 2012, pp. 1097–1105.
[37] A.-H. Phan, P. Tichavský, and A. Cichocki, TENSORBOX: A Matlab package for tensor decomposition. 2012. [Online]. Available: http://www.bsp.brain.riken.jp/ phan/tensorbox.php.
[38] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, Tensorlab v3.0. 2016. [Online]. Available: http://esat.kuleuven.be/sista/tensorlab/, 2016.
[39] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms – Part I: Lemmas for partitioned matrices," *SIAM J. Matrix Anal. Appl., Special Issue on Tensor Decompositions and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.
[40] I.V. Oseledets, "Tensor-train decomposition," *SIAM J. Scientific Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.
[41] A.-H. Phan, A. Cichocki, A. Uschmajew, P. Tichavský, G. Luta, and D. Mandic, "Tensor networks for latent variable analysis. Part I: Algorithms for tensor train decomposition," arXiv:1609.09230 [math.OC].
[42] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.
[43] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–3322, Nov. 2006.
[44] W. Gander, "Least squares with a quadratic constraint," *Numer. Math.*, vol. 36, no. 3, pp. 291–307, Sep. 1980.

Anh-Huy Phan (M'11) received the master's degree from the Hochiminh University of Technology, Ho Chi Minh, Vietnam, in 2005, and the Ph.D. degree from the Kyushu Institute of Technology, Kitakyushu, Japan, in 2011. From April 2012 to April 2015, he was a Visiting Research Scientist with the TOYOTA Collaboration Center, BSI-RIKEN. Since October 2011 till March 2018, he was a Research Scientist with the Laboratory for Advanced Brain Signal Processing, Brain Science Institute (BSI), RIKEN. Since May 2018, he has been with the SKOLTECH, Center for Computational and Data-Intensive Science and Engineering, Moscow, Russia, as an Assistant Professor. He is also a Visiting Associate Professor with the Tokyo University of Agriculture and Technology (TUAT), Fuchu, Japan and a Visiting Scientist with the Center for Brain Science (CBS), RIKEN, Japan. His research interests include multilinear algebra, tensor computation, tensor networks, fusion network, nonlinear system, blind source separation, and brain-computer interface.

Petr Tichavský (M'98–SM'04) received the Ph.D. degree in theoretical cybernetics from the Czechoslovak Academy of Sciences, Prague, Czech Republic, in 1992, and the Research Professor degree from the same institution in 2017. He is with the Institute of Information Theory and Automation, Czech Academy of Sciences. He is author and co-author of research papers in the area of sinusoidal frequency/ frequency-rate estimation, adaptive filtering and tracking of time-varying signal parameters, algorithm-independent bounds on achievable performance, sensor array processing, independent component analysis, and blind source separation, and tensor decompositions. He served as an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS from 2002 to 2004 and as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2005 to 2009 and from 2011 to 2016. From 2008 to 2011 and from 2016 till now, he has served as a member of the IEEE SPS committee Signal Processing Theory and Methods (SPTM). He has also served as a General Co-Chair of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP 2011 in Prague, Czech Republic.

Andrzej Cichocki (F'13) received the M.Sc. (with honors), Ph.D., and Dr.Sc. (habilitation) degrees, all in electrical engineering from the Warsaw University of Technology, Warsaw, Poland. He spent several years with the University Erlangen-Nurnberg, Germany as an Alexander-von-Humboldt Research Fellow and a Guest Professor. In 1995–2018, he was a Team Leader and the Head of the laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Japan, and he is currently a Professor with the Skolkovo Institute of Science and Technology(SKOLTECH), Moscow, Russia, and a Visiting/Adjunct Professor with the Tokyo University of Agriculture and Technology (TUAT), Fuchu, Japan, Hangzhou Dianzi University (HDU), Hangzhou, China, Nicolaus Copernicus University (UMK), Toruń, Poland, and with the Institute of Systems Research (IBS), Polish Academy of Science, Warsaw, Poland. He is author of more than 500 peer-review papers and 6 monographs in English (two of them translated to Chinese). He serves or served as an Associated Editor of the IEEE TRANSACTIONS ON SIGNALS PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, and the *Journal of Neuroscience Methods*. He was as a founding Editor-in-Chief for the *Journal Computational Intelligence and Neuroscience*. Currently, his research focus on deep learning, tensor decompositions, tensor networks for big data analytics, multiway blind source separation, and Brain Computer Interface and their biomedical applications. His publications currently report more than 37 000 citations according to Google Scholar, with an h-index of 84. He is currently among the three most cited Polish computer scientists.