Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 18, No. 2, pp. 469–495, 2022

HYBRID ALTERNATING EXTRA-GRADIENT AND NEWTON'S METHOD FOR TENSOR DECOMPOSITION*

Junwei Zhang and Yuning Yang†

Abstract: This paper considers modified versions of the alternating least-squares (ALS) and the regularized alternating least-squares (RALS) algorithms for tensor decomposition. We propose two hybrid alternating methods by combining the extra-gradient method with Newton's method, where at eacah subproblem, the correction step of the extra-gradient is replaced by a Newton step. Theoretically, the step-size of the correction step can be possibly chosen in a wide range. Under certain assumptions, we analyze the global convergence of our algorithm. Preliminary numerical experiments show the effectiveness of the proposed methods, compared to the standard ALS and RALS algorithms.

Key words: *alternating least-squares, extra-gradient, Newton's method, Łojasiewicz inequality, tensor decomposition*

Mathematics Subject Classification: *15A23, 15A69, 90C26, 90C30*

1 Introduction

In this paper, we introduce modified versions of the ALS [3, 6] and RALS [10] for tensor canonical polyadic decomposition (CPD) problem, i.e., finding the best approximation of a given tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with R rank-one components, which is written as:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\left\|\mathscr{A} - \sum\nolimits_{s=1}^{R} \boldsymbol{x}_{1,s} \circ \cdots \circ \boldsymbol{x}_{d,s}\right\|_F^2 \\ &\text{subject to} \quad \boldsymbol{x}_{1,s} \in \mathbb{R}^{n_1}, \ldots, \boldsymbol{x}_{d,s} \in \mathbb{R}^{n_d}, \quad s = 1, \ldots, R, \end{aligned} \tag{1.1}$$

where $\|\cdot\|_F$ stands for the Frobenius norm and $\boldsymbol{x}_{1,s} \circ \cdots \circ \boldsymbol{x}_{d,s}$ is a rank-one tensor generated by taking the outer products of d vectors.

This class of tensor-based problems is rich enough to encompass many optimization problems considered in practice. Applications can be found in various fields throughout science and engineering, including signal processing, machine learning, statistics, as well as numerical linear algebra [5, 8, 15, 16]. Developing algorithms to detect local minimizers or critical points of the objective functional is necessary for both theoretical research and

*This work was supported by the National Natural Science Foundation of China (No. 11801100), the Fok Ying Tong Education Foundation (No. 171094), and the Innovation Fund Designated for Graduate Students of Guangxi Province (YCSW2020055).

†Corresponding author

practical application of tensor computations [8]. Successful block coordinate descent type algorithms for tensor CPD problems include such as the ALS [3, 6] and RALS [12].

The ALS was initially introduced by Carol and Chang [3], and Harshman [6]. Taking the detachable structure of the original nonconvex least squares problem, the ALS solves the problem in the Gauss-Seidel fashion. Among various methods for solving CPD, the ALS remains a "workhorse" algorithm [8], and has been regarded as one of the most efficient methods for solving CPD [17]. Despite the advantage of the ALS, it has some drawbacks [4, 14]. The Hessian of the least-squares subproblem may not be positive definite since the subproblem might not be strongly convex, which results in the non-uniqueness of the solution within inner iterations, making it challenging to analyze the convergence of the ALS. In fact, [16] pointed out that understanding the convergence of the ALS is nontrivial. Moreover, the ALS may require a high number of iterations to converge when swamp occurs [10], where the swamp phenomenon means that the algorithm gets trapped in a small region for a large amount of iterations. The RALS algorithm was proposed [10] to avoid the non-uniqueness by introducing a proximal term into every subproblem and gives better convergence behavior with its global convergence given in [19]. While in the case that the swamp does not occur, the original ALS usually performs better, compared to the RALS with a fixed regularization parameter [19].

Due to the limitations of the above techniques, accelerating these algorithms has become the focus of several works in recent years. An acceleration version of the RALS algorithm has been proposed [19] using Aitken-Stefensen formula while no convergence result was provided for that algorithm. A trust-region algorithm based the ALS was proposed [7] to alleviate the occurrence of the swamp; however, the global convergence needs an assumption that the generated sequence is a Cauchy sequence. A self-adaptive RALS was proposed [11] to accelerate the origin RALS algorithm by using self-adaptive step-size. Another class of studies recognizes (1.1) as a nonlinear least-squares problem, and develops second-order algorithms to solve this problem, such as Gauss-Newton, Levenberg-Marquardt and inexact NLS algorithm; see [16] for an overview of the methods; Whereas, these algorithms either have no theoretical convergence guarantee, or require additional assumptions for the convergence.

The aforementioned deficiency or limitations motivate us to explore modified versions of the ALS and RALS. Extra-gradient method (EGM) was initially proposed by Korpelevich [9] for solving convex variational inequality problems. Different from gradient descent, EGM adds an extra correction step, which admits a prediction-correction fashion. Recently, EGM was extended [13] to tackle nonsmooth nonconvex optimization problems within a suitable step-size range. In this paper, by combining EGM with Newton's method, we develope two hybrid alternating extra-gradient and Newton's methods. Specifically, at each subproblem, unlike EGM, our algorithms utilize the Newton step in the correction step of the extra-gradient method. Furthermore, the step-size is self-adaptive to a certain extent. If the step-size of the prediction step is chosen close to its upper bound, then the step-size of the correction step can be chosen in a large range. This might give flexibility to the proposed algorithms. Considering that the Hessian of each subproblem may be singular, we use the Moore-Penrose pseudoinverse [2] or add a proximal term to make it feasible to access the Newton step. Under certain assumptions, the global convergence of the proposed algorithms are established. Actually, the ALS can be regarded as a block Newton method, see Sect. 3 for details of the scheme. Taking advantage of the extra-gradient method and Newton's method, and choosing the suitable step-size, we can achieve the acceleration of the standard ALS and RALS algorithms. Preliminary numerical examples illustrate the effectiveness of the proposed algorithms.

This paper is organized as follows. In Sect. 2, we introduce some notations and termi-

nologies for tensor approximation. In Sect. 3 we proposed our hybrid alternating method. We discuss two descent inequalities of our algorithms in Sect. 4, which contribute to the global convergence analysis in Sect. 5. Numerical experiments are presented in Sect. 6. Finally, Sect. 7 summarizes conclusions.

2 Preliminary

We use boldface lowercase letters $\boldsymbol{a}, \boldsymbol{b}, \ldots$ to denote vectors, italic capital $A, B, \ldots$ for matrices, bold capital letters $\boldsymbol{X}, \boldsymbol{Y}, \ldots$ for matrix sets defined below (2.2), and bold calligraphic letters $\mathscr{X}, \mathscr{Y}, \ldots$ for tensors. I denotes an identity matrix, whose size is clear from the context. The $(i_1, \ldots, i_d)$-th component of a d-way tensor $\mathscr{X}$ is denoted as $x_{i_1,\ldots,i_d}$. For $\mathscr{X}, \mathscr{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, their inner product is defined as $\langle \mathscr{X}, \mathscr{Y} \rangle = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} x_{i_1 \cdots i_d} \cdot y_{i_1 \cdots i_d}$. The Frobenius norm of $\mathscr{X}$ is defined as $\|\mathscr{X}\|_F = \sqrt{\langle \mathscr{X}, \mathscr{X} \rangle}$. For a real symmetric matrix A, $A \succ 0$ means that A is a positive definite. $(\cdot)^\top$ indicates the matrix transposition and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the max (min) eigenvalue of a matrix, respectively.

Kronecker product. The kronecker product of matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ is a $mp \times nq$ matrix, given by

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

Khatri-Rao product. The Khatri-Rao product of two matrices $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$ and $B = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n] \in \mathbb{R}^{p \times n}$ is a $mp \times n$ matrix given by

$$A \odot B := [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1, \ldots, \boldsymbol{a}_n \otimes \boldsymbol{b}_n].$$

Outer product. The outer product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is the rank-one matrix given by $\boldsymbol{a} \circ \boldsymbol{b} := \boldsymbol{a}\boldsymbol{b}^\top$. Similarly, the outer product $\boldsymbol{x}_1 \circ \cdots \circ \boldsymbol{x}_d$ of d vectors $\boldsymbol{x}_i \in \mathbb{R}^{n_i}, i = 1, \ldots, d$ is a rank-one tensor $\mathscr{X}$.

CP decomposition. Given a tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, CPD decomposes $\mathscr{A}$ into the sum of several rank-one tensors [8]

$$\mathscr{A} = \sum\nolimits_{s=1}^{R} \boldsymbol{x}_{1,s} \circ \cdots \circ \boldsymbol{x}_{d,s} := [\![X_1, \ldots, X_d]\!], \tag{2.1}$$

where $\boldsymbol{x}_{i,s} \in \mathbb{R}^{n_i}$, and $X_i := (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,R}) \in \mathbb{R}^{n_i \times R}, 1 \le i \le d$ are called the factor matrices.

Unfolding. The mode-i unfolding of a tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a matrix $A_{(i)} \in \mathbb{R}^{n_i \times \prod_{j \ne i}^{d} n_j}$, where the entries are arranged in a certain order. For example, CPD of $\mathscr{A}$ in (2.1) can be represented as

$$A_{(i)} = X_i \left(X_d \odot \cdots \odot X_{i+1} \odot X_{i-1} \odot \cdots \odot X_1\right)^\top.$$

Model description. (2.1) is rarely exact. Thus the optimization problem (1.1) of tensor CPD can be rewritten as (1.1):

$$\min_{\boldsymbol{X}} F(\boldsymbol{X}) = \frac{1}{2}\|\mathscr{A} - \sum\nolimits_{s=1}^{R} \boldsymbol{x}_{1,s} \circ \cdots \circ \boldsymbol{x}_{d,s}\|_F^2 = \frac{1}{2}\|\mathscr{A} - [\![X_1, \ldots, X_d]\!]\|_F^2, \tag{2.2}$$

where we denote the tuple of the factor matrices $(X_1, \ldots, X_d)$ as $\boldsymbol{X}$, i.e.,

$$\boldsymbol{X} := (X_1, \ldots, X_d).$$

To simplify the notations, in the sequel, we also denote $\boldsymbol{X}^{(k)} := (X_1^{(k)}, \ldots, X_d^{(k)})$, and

$$(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) := (X_1^{(k+1)}, \ldots, X_{i-1}^{(k+1)}, X_i^{(k)}, \ldots, X_d^{(k)}) \qquad (1 \leq i \leq d)$$

as the intermediate iterates.

Let $\boldsymbol{X}^* = (X_1^*, \ldots, X_d^*)$ be a critical point of (2.2) and (2.3). Then first order optimal condition yields that

$$\nabla F(\boldsymbol{X}^*) = 0 \Leftrightarrow X_i^* M_{-i}^{*\top} M_{-i}^* = A_{(i)} M_{-i}^*, \quad 1 \leq i \leq d.$$

Combining Khatri-Rao product with unfolding, (2.2) with respect to each X_i but with other factors held fixed reduces to a least squares subproblem, which can be written as:

$$\begin{aligned} \min_{X_i} F(\boldsymbol{X}) &= \frac{1}{2} \| A_{(i)} - X_i \left(X_d \odot \cdots \odot X_{i+1} \odot X_{i-1} \odot \cdots \odot X_1 \right)^\top \|_F^2 \\ &:= \frac{1}{2} \| A_{(i)} - X_i M_{-i}^\top \|_F^2, \quad 1 \leq i \leq d, \end{aligned} \tag{2.3}$$

where we denote $M_{-i} := X_d \odot \cdots \odot X_{i+1} \odot X_{i-1} \odot \cdots \odot X_1$.

3 The Proposed Methods

We first recall the ALS and extra-gradient method (EGM).

The ALS solves (2.2) in the following manner:

$$\begin{aligned} X_i^{(k+1)} &\in \arg\min_{X_i} F(X_1^{(k+1)}, \ldots, X_{i-1}^{(k+1)}, X_i, X_{i+1}^{(k)}, \ldots, X_d^{(k)}) \\ &= \arg\min_{X_i} F(X_{<i}^{(k+1)}, X_i, X_{>i}^{(k)}), \quad 1 \leq i \leq d. \end{aligned}$$

Using our notations (2.3), each subproblem of the ALS can be written as (for $k = 1, 2, \ldots$)

ALS

$$\text{For } i = 1, \ldots, d, \quad X_i^{(k+1)} \in \arg\min_X \frac{1}{2} \| A_{(i)} - X M_{-i}^{(k+1)\top} \|_F^2, \tag{3.1}$$

where

$$M_{-i}^{(k+1)} := X_d^{(k)} \odot \cdots \odot X_{i+1}^{(k)} \odot X_{i-1}^{(k+1)} \odot \cdots \odot X_1^{(k+1)}. \tag{3.2}$$

Throughout this paper, we denote

$$\begin{aligned} \nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) &:= X_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} - A_{(i)} M_{-i}^{(k+1)}, \\ \nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) &:= M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} = \nabla_{ii}^2 F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}), \end{aligned} \tag{3.3}$$

as the partial gradient and the partial Hessian of $F(\cdot)$ with respect to X_i, respectively.

Indeed, (3.1) can be regarded as a Newton step with unit step-size if the Hessian is invertible: By (3.3), we can reformulate the ALS as follows:

$$\begin{aligned} X_i^{(k+1)} &= A_{(i)}M_{-i}^{(k+1)}(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{-1} \\ &= X_i^{(k)} - (X_i^{(k)}M_{-i}^{(k+1)\top}M_{-i}^{(k+1)} - A_{(i)}M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{-1} \\ &= X_i^{(k)} - \nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) \left(\nabla_{ii}^2 F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})\right)^{-1}. \end{aligned}$$

EGM has been extended to tackle a nonsmooth nonconvex optimization problem $\min_{\boldsymbol{x}\in\mathbf{R}^n}\{f(\boldsymbol{x}) + g(\boldsymbol{x})\}$ [13], where $f(\boldsymbol{x})$ is a smooth nonconvex function and $g(\boldsymbol{x})$ a simple nonsmooth convex function. For $k \geq 1$, EGM can be written as following prediction-correction recursion (for $k = 1, 2, \ldots$)

EGM

$$\begin{aligned} \boldsymbol{y}_k &= \operatorname{prox}_{s_k g}(\boldsymbol{x}_k - s_k \nabla f(\boldsymbol{x}_k)), \\ \boldsymbol{x}_{k+1} &= \operatorname{prox}_{\alpha_k g}(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{y}_k)), \end{aligned}$$

where $\operatorname{prox}_{tg}(\boldsymbol{x}) := \arg\min_{\boldsymbol{y}\in\mathbb{R}^n} g(\boldsymbol{y}) + \frac{1}{2t}\|\boldsymbol{y} - \boldsymbol{x}\|^2$, and $0 < s_k \leq \alpha_k \leq \frac{1}{L}$ and L is the Lipschitz constant of $\nabla f(\boldsymbol{x})$.

In what follows, we denote

$$\nabla_i F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) := \bar{X}_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} - A_{(i)} M_{-i}^{(k+1)}.$$

Motivated by the ALS and EGM, we proposed our hybrid alternating extra-gradient and Newton's method (HAEN) by replacing the correction step of EGM with the Newton step, which is given as follows (for $k = 1, 2, \ldots$)

HAEN

For $i = 1, \ldots, d$,

$$\begin{aligned} \bar{X}_i^{(k)} &= X_i^{(k)} - \beta_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}), \\ X_i^{(k+1)} &= X_i^{(k)} - \omega_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) \left(\nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})\right)^{\dagger}, \end{aligned} \tag{3.4}$$

To see more clear HAEN, consider $\min_{\boldsymbol{x}} f(\boldsymbol{x})$. Then one step of HAEN is

$$\bar{\boldsymbol{x}} = \boldsymbol{x} - \beta \nabla f(\boldsymbol{x}),\ \boldsymbol{x}^+ = \boldsymbol{x} - \omega \nabla f(\bar{\boldsymbol{x}}) \left(\nabla^2 f(\bar{\boldsymbol{x}})\right)^{\dagger}.$$

In HAEN, $\beta_i^{(k)} > 0$ is the extra-gradient step-size and $\omega_i^{(k)} > 0$ is the Newton step-size. $\beta_i^{(k)}$ depends on $M_{-i}^{(k+1)}$ and $\omega_i^{(k)}$ depends on $M_{-i}^{(k+1)}$ and $\beta_i^{(k)}$. Clearly when $\beta_i^{(k)} = 0$ and $\omega_i^{(k)} = 1$, (3.4) reduces exactly to ALS. In particular, unlike the step-sizes of EGM that have to be constrained in $(0, \frac{1}{L}]$, $\omega_i^{(k)}$ here can be possibly larger. The choice of $\beta_i^{(k)}$ and $\omega_i^{(k)}$ in (3.4) will be specified in the next section.

On the other hand, to overcome the possible singularity of the partial Hessian of each subproblem, we use the Moore-Penrose pseudoinverse $\nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})^{\dagger}$ instead of the inverse of $\nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$ in the Newton step. As pointed our in [8], such

a pseudoinverse has a special form as the partial Hessian is constructed by the Khatri-Rao product of certain matrices. In particular, note that the size of the partial Hessian is $R \times R$, which is independent of the dimension of the problem. Therefore, computing the pseudoinverse of the partial Hessian is practical when R is small; this is the case in practice usually. If R is large, one can solve a system of linear equations instead to obtain $X_i^{(k+1)}$.

We remark that very recently, [20] proposed a stochastic extra-step quasi-Newton method for nonsmooth nonconvex problems, where for the non-stochastic and smooth setting, it reduces to

$$\bar{\boldsymbol{x}} = \boldsymbol{x} - \beta W \nabla f(\boldsymbol{x}),\ \boldsymbol{x}^+ = \boldsymbol{x} - \alpha W \nabla f(\boldsymbol{x}) - \lambda \nabla f(\bar{\boldsymbol{x}}),$$

where W is a matrix to capture the higher-order information. Clearly, even if $W = \nabla^2 f(\boldsymbol{x})^\dagger$, our scheme is different from this algorithm. On the other hand, HAEN is executed in an alternating fashion, and the range of the step-sizes and the convergence results, which will be presented later, are also different.

The RALS was proposed to improve the convergence property of the ALS by introducing a proximal term into every subproblem, which is defined as [10] (for $k = 1, 2, \ldots$)

RALS

$$\text{For } i = 1, \ldots, d, \quad X_i^{(k+1)} \in \arg\min_X \frac{1}{2}\|A_{(i)} - X M_{-i}^{(k+1)\top}\|_F^2 + \frac{\alpha}{2}\|X - X_i^{(k)}\|_F^2,$$

where $\alpha > 0$. By combining the RALS with HAEN, we propose the hybrid regularized alternating extra-gradient and Newton's method (HRAEN). Likewise, we first remake the notations referring to HRAEN as follows:

$$\begin{aligned}
\hat{F}(X_{<i}^{(k+1)}, X, X_{\geq i}^{(k)}) &:= \frac{1}{2}\|A_{(i)} - X M_{-i}^{(k+1)\top}\|_F^2 + \frac{\alpha}{2}\|X - X_i^{(k)}\|_F^2,\\
\nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) &:= X_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} - A_{(i)} M_{-i}^{(k+1)},\\
\nabla_i \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) &:= \bar{X}_i^{(k)} (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I) - A_{(i)} M_{-i}^{(k+1)} - \alpha X_i^{(k)},\\
\nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) &:= M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I.
\end{aligned}$$

Then, HRAEN is given by the following recursion (for $k = 1, 2, \ldots$)

HRAEN

For $i = 1, \ldots, d$,

$$\begin{aligned}
\bar{X}_i^{(k)} &= X_i^{(k)} - \beta_i^{(k)} \nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}),\\
X_i^{(k+1)} &= X_i^{(k)} - \omega_i^{(k)} \nabla_i \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) \left(\nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})\right)^{-1}.
\end{aligned} \tag{3.5}$$

Similarly, the parameters of (3.5) resemble (3.4) close and will be specified in the next section.

4 One-Step Descent Property

We introduce in this section two technical properties of our algorithms, which study the one-step sufficient decrease of HAEN and HRAEN. For simplicity of presentation, we drop

the subscripts and the superscripts and remark that (for $i = 1, \ldots, d$ and $k = 1, 2, \ldots$)

$$\begin{aligned} &A = A_{(i)} \in \mathbb{R}^{n_i \times \prod_{j \neq i}^{d} n_j},\ M^\top = M_{-i}^{(k+1)\top} \in \mathbb{R}^{R \times \prod_{j \neq i}^{d} n_j},\ X = X_i^{(k)} \in \mathbb{R}^{n_i \times R}, \\ &X^+ = X_i^{(k+1)} \in \mathbb{R}^{n_i \times R}, \bar{X} = \bar{X}_i^{(k)} \in \mathbb{R}^{n_i \times R},\ \beta = \beta_i^{(k)},\ \omega = \omega_i^{(k)}. \end{aligned} \tag{4.1}$$

In addition, we give some notations about the compact SVD of the following matrices used in the sequel:

$$\begin{aligned} &H = M^\top M = U \Lambda U^\top,\ H^\dagger = U \Lambda^{-1} U^\top, \\ &M = V \Lambda^{\frac{1}{2}} U^\top, \\ &\hat{H} = M^\top M + \alpha I = \hat{U} \hat{\Lambda} \hat{U}^\top \succ 0, \end{aligned} \tag{4.2}$$

where $\alpha > 0$, $H, \hat{H} \in \mathbb{R}^{R \times R}$ are symmetric, $U \in \mathbb{R}^{R \times r}, V \in \mathbb{R}^{\prod_{j \neq i}^{d} n_j \times r}$ ($r \leq R$ and r is the rank of H), $\hat{U} \in \mathbb{R}^{R \times R}$ are column-orthogonal matrices, and $\Lambda \in \mathbb{R}^{r \times r}$, $\hat{\Lambda} \in \mathbb{R}^{R \times R}$ are two diagonal matrices. Note that Λ and $\hat{\Lambda}$ are invertible.

We begin with a technical lemma.

Lemma 4.1. *Let $f(Y) := \frac{1}{2}\|A - YM^\top\|_F^2$ where A and M are defined in* (4.1)*. Then for any Y, there holds*

$$\nabla f(Y) U U^\top = \nabla f(Y). \tag{4.3}$$

Proof. With the remark claimed above, (4.3) can be expanded as

$$\begin{aligned} \nabla f(Y) U U^\top &= (Y M^\top M - A M) U U^\top \\ &= Y M^\top M U U^\top - A M U U^\top \\ \text{(by (4.2))} \qquad &= Y U \Lambda U^\top U U^\top - A V \Lambda^{\frac{1}{2}} U^\top U U^\top \\ &= Y U \Lambda U^\top - A V \Lambda^{\frac{1}{2}} U^\top \\ &= Y M^\top M - A M \\ &= \nabla f(Y). \end{aligned}$$

Hence (4.3) follows. □

Proposition 4.2. *Consider the least squares problem $\min_Y f(Y) = \frac{1}{2}\|A - YM^\top\|_F^2$, if $\beta \in (0, \frac{1}{\lambda_{\max}(H)})$, $\omega \in (0, \frac{2}{1 - \beta \lambda_{\max}(H)})$, where H is defined in* (4.2)*, then*

$$f(X) - f(X^+) \geq \lambda_{\min}(L) \|X - X^+\|_F^2,$$

where $X^+ := X - \omega \nabla f(\bar{X}) \left(\nabla^2 f(\bar{X})\right)^\dagger, \bar{X} := X - \beta \nabla f(X)$, and $L := \frac{H((I - \beta H)^{-1} - \omega I)}{\omega} + \frac{H}{2}$.

Proof. We begin this proof by performing the Taylor expansion of $f(X)$ at X^+

$$\begin{aligned} f(X) &= f(X^+) + \langle \nabla f(X^+), X - X^+ \rangle + \frac{1}{2} \langle (X - X^+) \nabla^2 f(X^+), X - X^+ \rangle \\ &= f(X^+) + \langle \nabla f(X^+), X - X^+ \rangle + \frac{1}{2} \langle (X - X^+) M^\top M, X - X^+ \rangle \\ \text{(by (4.2))} \qquad &= f(X^+) + \langle \nabla f(X^+), X - X^+ \rangle + \frac{1}{2} \langle (X - X^+) H, X - X^+ \rangle \\ &= f(X^+) + \langle \nabla f(X^+), X - X^+ \rangle + \frac{1}{2} \|X - X^+\|_H^2. \end{aligned} \tag{4.4}$$

Proceeding as the proof of Lemma 4.1, $\nabla f(X^+)$ can be written as follows:

$$\begin{aligned}
\nabla f(X^+) &= X^+M^\top M - AM \\
\text{(by (4.2))} \qquad &= X^+H - AM \\
&= (X - \omega\nabla f(\bar{X})(\nabla^2 f(\bar{X}))^\dagger)H - AM \\
(H = \nabla^2 f(\bar{X})) \qquad &= (X - \omega\nabla f(\bar{X})H^\dagger)H - AM \\
&= XH - AM - \omega\nabla f(\bar{X})H^\dagger H \\
&= \nabla f(X) - \omega\nabla f(\bar{X})U\Lambda^{-1}U^\top U\Lambda U^\top \\
&= \nabla f(X) - \omega\nabla f(\bar{X})U\Lambda^{-1}\Lambda U^\top \\
&= \nabla f(X) - \omega\nabla f(\bar{X})UU^\top \\
\text{(by (4.3))} \qquad &= \nabla f(X) - \omega\nabla f(\bar{X}).
\end{aligned} \tag{4.5}$$

Note that $\bar{X} := X - \beta\nabla f(X)$, $\nabla f(\bar{X})$ can be reformulated as

$$\begin{aligned}
\nabla f(\bar{X}) &= \bar{X}H - AM \\
&= (X - \beta\nabla f(X))H - AM \\
&= XH - AM - \beta\nabla f(X)H \\
&= \nabla f(X) - \beta\nabla f(X)H \\
&= \nabla f(X)(I - \beta H).
\end{aligned} \tag{4.6}$$

It follows from $\beta \in (0, \frac{1}{\lambda_{\max}(H)})$ that $I - \beta H$ is invertible, and so

$$\nabla f(X) = \nabla f(\bar{X})(I - \beta H)^{-1}. \tag{4.7}$$

Substituting (4.7) into (4.5), $\nabla f(X^+)$ can be rewritten as

$$\nabla f(X^+) = \nabla f(\bar{X})(I - \beta H)^{-1} - \omega\nabla f(\bar{X}). \tag{4.8}$$

Taking two successive operations of $X^+ := X - \omega\nabla f(\bar{X})H^\dagger$ gives

$$\begin{aligned}
\omega^{-1}(X - X^+)H &= \nabla f(\bar{X})H^\dagger H \\
&= \nabla f(\bar{X})U\Lambda^{-1}U^\top U\Lambda U^\top \\
&= \nabla f(\bar{X})UU^\top \\
\text{(by (4.3))} \qquad &= \nabla f(\bar{X}).
\end{aligned} \tag{4.9}$$

(4.8) in connection with (4.9) yields

$$\begin{aligned}
\nabla f(X^+) &= \omega^{-1}(X - X^+)H(I - \beta H)^{-1} - \omega\omega^{-1}(X - X^+)H \\
&= \omega^{-1}(X - X^+)H(I - \beta H)^{-1} - (X - X^+)H \\
&= (X - X^+)H\left(\omega^{-1}(I - \beta H)^{-1} - I\right) \\
&= (X - X^+)H\omega^{-1}\left((I - \beta H)^{-1} - \omega I\right).
\end{aligned} \tag{4.10}$$

Combining (4.10) with (4.4) gives

$$\begin{aligned}
f(X) - f(X^+) &= \langle (X - X^+)H\omega^{-1}\left((I - \beta H)^{-1} - \omega I\right), X - X^+\rangle \\
&\quad + \frac{1}{2}\langle (X - X^+)H, X - X^+\rangle \\
&= \|X - X^+\|_L^2,
\end{aligned} \tag{4.11}$$

where

$$L = \frac{H\left((I-\beta H)^{-1}-\omega I\right)}{\omega} + \frac{H}{2}.$$

Noticing the compact SVD of H, L can be written as

$$\begin{aligned} L &= U\left(\frac{\Lambda\left((I-\beta\Lambda)^{-1}-\omega I\right)}{\omega}+\frac{\Lambda}{2}\right)U^\top \\ &= U\left(\frac{2\Lambda(I-\beta\Lambda)^{-1}-\omega\Lambda}{2\omega}\right)U^\top \\ &= U\left(\frac{\Lambda\left(2(I-\beta\Lambda)^{-1}-\omega I\right)}{2\omega}\right)U^\top \\ &= U\cdot\operatorname{diag}\left(\ldots,\lambda_j\left(\frac{2-(1-\beta\lambda_j)\omega}{2\omega(1-\beta\lambda_j)}\right),\ldots\right)\cdot U^\top, \end{aligned}$$

where $\lambda_j > 0$ $(1 \le j \le r)$ is the j-th diagonal entry of Λ.

It is easily seen that if $\beta \in (0, \frac{1}{\lambda_{\max}(H)})$, $\omega \in (0, \frac{2}{1-\beta\lambda_{\max}(H)})$, $L \succ 0$ and (4.11) yields

$$f(X) - f(X^+) \ge \lambda_{\min}(L)\|X - X^+\|_F^2.$$

The proof has been completed. □

Similarly we have:

Proposition 4.3. *Consider* $\min_Y \hat{f}(Y) = f(Y) + \frac{\alpha}{2}\|Y - X\|_F^2$, *where* $\alpha > 0$ *and* $f(Y) = \frac{1}{2}\|A - YM^\top\|_F^2$. *Let* $Y^+ := Y - \omega\nabla\hat{f}(\bar{Y})(\nabla^2\hat{f}(\bar{Y}))^{-1}$ *and* $\bar{Y} := Y - \beta\nabla\hat{f}(Y)$, $\forall Y$. *Then for* X, X^+ *defined as those in Proposition 4.2, if* $\beta \in (0, \frac{1}{\lambda_{\max}(\hat{H})})$ *and* $\omega \in (0, \frac{2}{1-\beta\lambda_{\max}(\hat{H})})$, *then*

$$f(X) - f(X^+) \ge (\lambda_{\min}(\hat{L}) + \frac{\alpha}{2})\|X - X^+\|_F^2,$$

where $\hat{L} = \frac{\hat{H}((I-\beta\hat{H})^{-1}-\omega I)}{\omega} + \frac{\hat{H}}{2}$.

Proof. Similar to those in the proof of Proposition 4.2, Taylor expansion of $\hat{f}(Y)$ at Y^+ yields

$$\begin{aligned} \hat{f}(Y) &= \hat{f}(Y^+) + \langle\nabla\hat{f}(Y^+), Y - Y^+\rangle + \frac{1}{2}\langle(Y-Y^+)\nabla^2\hat{f}(Y^+), Y - Y^+\rangle \\ &= \hat{f}(Y^+) + \langle\nabla\hat{f}(Y^+), Y - Y^+\rangle + \frac{1}{2}\langle(Y-Y^+)(M^\top M + \alpha I), Y - Y^+\rangle \\ \text{(by (4.2))} \quad &= \hat{f}(Y^+) + \langle\nabla\hat{f}(Y^+), Y - Y^+\rangle + \frac{1}{2}\langle(Y-Y^+)\hat{H}, Y - Y^+\rangle \\ &= \hat{f}(Y^+) + \langle\nabla\hat{f}(Y^+), Y - Y^+\rangle + \frac{1}{2}\|Y - Y^+\|_{\hat{H}}^2. \end{aligned} \tag{4.12}$$

It is clear that $\hat{H} \succ 0$ as $\alpha > 0$. The remainder of the argument is analogous to that in Proposition 4.2. Similar to (4.5) we have

$$\begin{aligned} \nabla\hat{f}(Y^+) &= Y^+(M^\top M + \alpha I) - AM - \alpha X \\ \text{(by (4.2))} \quad &= (Y - \omega\nabla\hat{f}(\bar{Y})(\nabla^2\hat{f}(\bar{Y}))^{-1})\hat{H} - AM - \alpha X \\ (\hat{H} = \nabla^2\hat{f}(\bar{Y})) \quad &= Y\hat{H} - AM - \alpha X - \omega\nabla\hat{f}(\bar{Y})\hat{H}^{-1}\hat{H} \\ &= \nabla\hat{f}(Y) - \omega\nabla\hat{f}(\bar{Y}). \end{aligned} \tag{4.13}$$

Combining $\bar{Y} := Y - \beta\nabla\hat{f}(Y)$ with $\nabla\hat{f}(\bar{Y})$ gives

$$\begin{aligned}\nabla\hat{f}(\bar{Y}) &= \bar{Y}\hat{H} - AM - \alpha X \\ &= (Y - \beta\nabla\hat{f}(Y))\hat{H} - AM - \alpha X \\ &= Y\hat{H} - AM - \alpha X - \beta\nabla\hat{f}(Y)\hat{H} \\ &= \nabla\hat{f}(Y) - \beta\nabla\hat{f}(Y)\hat{H} \\ &= \nabla\hat{f}(Y)(I - \beta\hat{H}),\end{aligned} \tag{4.14}$$

where $(I - \beta\hat{H})^{-1}$ exists if $\beta \in (0, \frac{1}{\lambda_{\max}(\hat{H})})$. Reformulating (4.14) gives

$$\nabla\hat{f}(Y) = \nabla\hat{f}(\bar{Y})(I - \beta\hat{H})^{-1}. \tag{4.15}$$

Inserting (4.15) into (4.13) yields

$$\nabla\hat{f}(Y^+) = \nabla\hat{f}(\bar{Y})(I - \beta\hat{H})^{-1} - \omega\nabla\hat{f}(\bar{Y}). \tag{4.16}$$

Rearranging $Y^+ := Y - \omega\nabla\hat{f}(\bar{Y})(\nabla^2\hat{f}(\bar{Y}))^{-1}$ leads to

$$\nabla\hat{f}(\bar{Y}) = \omega^{-1}(Y - Y^+)\hat{H}. \tag{4.17}$$

Substituting (4.17) into (4.16), we have

$$\nabla\hat{f}(Y^+) = (Y - Y^+)\hat{H}\omega^{-1}\left((I - \beta\hat{H})^{-1} - \omega I\right). \tag{4.18}$$

Plugging (4.18) back into (4.12) yields

$$\begin{aligned}\hat{f}(Y) - \hat{f}(Y^+) &= \langle (Y - Y^+)\hat{H}\omega^{-1}\left((I - \beta\hat{H})^{-1} - \omega I\right), Y - Y^+\rangle \\ &\quad + \frac{1}{2}\langle (Y - Y^+)\hat{H}, Y - Y^+\rangle \\ &= \|Y - Y^+\|_{\hat{L}}^2,\end{aligned}$$

where

$$\hat{L} = \frac{\hat{H}\left((I - \beta\hat{H})^{-1} - \omega I\right)}{\omega} + \frac{\hat{H}}{2}.$$

Employing the compact SVD of $\hat{H}$ in (4.2), we have

$$\begin{aligned}\hat{L} &= \hat{U}\left(\frac{\hat{\Lambda}\cdot\left((I - \beta\hat{\Lambda})^{-1} - \omega I\right)}{\omega} + \frac{\hat{\Lambda}}{2}\right)\hat{U}^\top \\ &= \hat{U}\cdot\operatorname{diag}\left(\ldots, \hat{\lambda}_j\left(\frac{2 - (1 - \beta\hat{\lambda}_j)\omega}{2\omega(1 - \beta\hat{\lambda}_j)}\right), \ldots\right)\cdot\hat{U}^\top,\end{aligned}$$

where $\hat{\lambda}_j$ $(1 \le j \le R)$ is the j-th diagonal entry of $\hat{\Lambda}$.

When $\beta \in (0, \frac{1}{\lambda_{\max}(\hat{H})})$, $\omega \in (0, \frac{2}{1 - \beta\lambda_{\max}(\hat{H})})$, $\hat{L} \succ 0$ and then

$$\hat{f}(Y) - \hat{f}(Y^+) \ge \lambda_{\min}(\hat{L})\|Y - Y^+\|_F^2. \tag{4.19}$$

By the definition of $\hat{f}(\cdot)$ and (4.19), let $X = Y$, $X^+ = Y^+$, and so

$$f(X) - f(X^+) - \frac{\alpha}{2}\|X^+ - X\|_F^2 = \hat{f}(X) - \hat{f}(X^+) \geq \lambda_{\min}(\hat{L})\|X - X^+\|_F^2,$$

which yields

$$f(X) - f(X^+) \geq (\lambda_{\min}(\hat{L}) + \frac{\alpha}{2})\|X - X^+\|_F^2.$$

The proof is completed. □

Remark 4.4. In the above two propositions, we can observe that if β is close to $\lambda_{\max}(H)$, then β may take a large range. In any case, we see that the choice of ω can be larger than 2. This gives more flexibility to the algorithms.

5 Convergence Analysis

In this section, we adopt the KL-inequality techinique to analysis the global convergence of the sequence $\{\boldsymbol{X}^{(k)}\}$ generated by HAEN (3.4) and HRAEN (3.5) under some assumptions. We divide the proof of our convergence result in several steps. First, we need to state the Łojasiewicz gradient inequality in Lemma 5.1. Next, we establish the sufficiently decreasing inequality in Lemma 5.2 and the relative error inequality in Lemma 5.4, Lemma 5.5 and Lemma 5.6. With the above lemmas, we show that any accumulation point of the sequence generated by our algorithms is a stationary point of (2.3) in Theorem 5.7. Finally, to summarize what we have proved, we achieve the global convergence of our algorithms in Theorem 5.8.

The Łojasiewicz gradient inequality is stated as follows:

Lemma 5.1 (Lemma 2.1 of [1])**.** *Let φ be a real analytic function on a neighborhood of $\boldsymbol{x}^* \in \mathbb{R}^n$. Then there exist constant $c > 0$ and $\mu \in [0, 1)$ such that*

$$\|\nabla\varphi(\boldsymbol{x})\| \geq c|\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{x}^*)|^\mu$$

in some neighborhood U of $\boldsymbol{x}^*$.

Then, the following lemma is exploited to show the sufficiently decreasing inequality by the one-step descent property studied in the preceding Section.

Lemma 5.2. *Let $\{\boldsymbol{X}^{(k)}\}$ be a sequence generated by the HAEN (3.4). If $\beta_i^{(k)} \in \big[\epsilon_1, \frac{1}{\lambda_{\max}(H_i^{(k)})} - \epsilon_1\big]$, $\omega_i^{(k)} \in \big[\epsilon_1, \frac{2}{1-\beta_i^{(k)}\lambda_{\max}(H_i^{(k)})} - \epsilon_1\big]$, where $H_i^{(k)} := \nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$, and $\epsilon_1 > 0$ is a small enough but fixed constant, then there exists a universal constant $\epsilon_0 > 0$, such that*

$$F(\boldsymbol{X}^{(k)}) - F(\boldsymbol{X}^{(k+1)}) \geq \epsilon_0\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F^2. \quad \textit{and} \quad \lim_{k\to\infty} F(\boldsymbol{X}^{(k)}) = F^* \geq 0.$$

Remark 5.3. Similar result holds for the sequence $\{\boldsymbol{X}^{(k)}\}$ generated by HRAEN (3.5), and we do not restate it again.

Proof. From Proposition 4.2 and noticing the range of $\beta_i^{(k)}$ and $\omega_i^{(k)}$, there is a universal constant $\epsilon_0 > 0$ such that

$$F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) - F(X_{\leq i}^{(k+1)}, X_{>i}^{(k)}) \geq \epsilon_0\|X_i^{(k+1)} - X_i^{(k)}\|_F^2,\ 1 \leq i \leq d,$$

where we recall that $(X_{\le i}^{(k+1)}, X_{>i}^{(k)}) := (X_1^{(k+1)}, \ldots, X_i^{(k+1)}, X_{i+1}^{(k)}, \ldots, X_d^{(k)})$, Summing up which yields

$$F(\boldsymbol{X}^{(k)}) - F(\boldsymbol{X}^{(k+1)}) = \sum_{i=1}^{d} F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)}) - F(X_{\le i}^{(k+1)}, X_{>i}^{(k)}) \ge \epsilon_0 \|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F^2.$$

Since $F(\cdot)$ is lower bounded and $F(\boldsymbol{X}^{(k)})$ is decreasing, $F(\boldsymbol{X}^{(k)})$ has a limit and $\lim_{k\to\infty} F(\boldsymbol{X}^{(k)}) = F^* \ge 0$. The proof is completed. □

After that, we turn to show our relative error inequality below.

Lemma 5.4. *Let $\{\boldsymbol{X}^{(k)}\}$ be a sequence generated by HAEN (3.4) and assume that it is bounded. If $\beta_i^{(k)} \in (0, \frac{1}{\lambda_{\max}(H_i^{(k)})})$ and $\omega_i^{(k)} \in (0, \frac{2}{1-\beta_i^{(k)}\lambda_{\max}(H_i^{(k)})})$ $(1 \le i \le d)$, where $H_i^{(k)} := \nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$, then there exists a constant $b_1 > 0$, such that*

$$\|\nabla F(\boldsymbol{X}^{(k)})\|_F \le b_1 \|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F. \tag{5.1}$$

Proof. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are bounded, then for any $1 \le i \le d$, there exists a constant $\hat{b} > 0$, such that

$$\|\nabla_i F(\boldsymbol{X}) - \nabla_i F(\boldsymbol{Y})\|_F \le \|\nabla F(\boldsymbol{X}) - \nabla F(\boldsymbol{Y})\|_F \le \hat{b}\|\boldsymbol{X} - \boldsymbol{Y}\|_F. \tag{5.2}$$

Since $\{\boldsymbol{X}^{(k)}\}$ is bounded, it is clear that $M_{-i}^{(k+1)}$ (recall its definition in (3.2)), $A_{(i)}$, $\nabla_i F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)})$ and $\nabla_{ii}^2 F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)})$ are bounded, respectively. Recalling (4.7) gives

$$\nabla_i F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) = \nabla_i F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)})(I - \beta_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)}). \tag{5.3}$$

Then substituting (5.3) into the Newton step in (3.4) yields

$$\begin{aligned}
X_i^{(k+1)} &= X_i^{(k)} - \omega_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) \left(\nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})\right)^\dagger \\
&= X_i^{(k)} - \omega_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)})(I - \beta_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)}) \left(\nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})\right)^\dagger \\
&= X_i^{(k)} - \omega_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)})(I - \beta_i^{(k)} M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})^\dagger \\
&= X_i^{(k)} - \omega_i^{(k)} \nabla_i F(X_{<i}^{(k+1)}, X_{\ge i}^{(k)}) \Big((M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})^\dagger \\
&\quad - \beta_i^{(k)} (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})^\dagger\Big).
\end{aligned}$$

Adopting the compact SVD representation of H, we have

$$\begin{aligned}
& (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})^\dagger - \beta_i^{(k)} (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top} M_{-i}^{(k+1)})^\dagger \\
\text{(by (4.2))} \qquad & = H^\dagger - \beta_i^{(k)} H H^\dagger \\
& = U\Lambda^{-1}U^\top - \beta_i^{(k)} U\Lambda U^\top U \Lambda^{-1} U^\top \\
& = U\Lambda^{-1}U^\top - U\beta_i^{(k)} I U^\top \\
& = U(\Lambda^{-1} - \beta_i^{(k)} I)U^\top \\
& = U \cdot \mathrm{diag}(\ldots, \lambda_j^{-1} - \beta_i^{(k)}, \ldots) \cdot U^\top,
\end{aligned}$$

where H, U, Λ and λ_j were defined in Proposition 4.2.

Since $\beta_i^{(k)} \in (0, \frac{1}{\lambda_{\max}(H)})$ and $\omega_i^{(k)} \in (0, \frac{2}{1-\beta_i^{(k)}\lambda_{\max}(H)})$, it is straightforward to show that

$$(\omega_i^{(k)})^{-1}\left((M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{\dagger} - \beta_i^{(k)}(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{\dagger}\right)^{-1}$$

exists and so

$$(\omega_i^{(k)})^{-1}(X_i^{(k)} - X_i^{(k+1)}) \left((M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{\dagger} - \beta_i^{(k)}(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})(M_{-i}^{(k+1)\top}M_{-i}^{(k+1)})^{\dagger}\right)^{-1} = \nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}).$$

Consequently, we infer that there exists a constant $\bar{b} > 0$, such that

$$\|\nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})\|_F \leq \bar{b}\|X_i^{(k)} - X_i^{(k+1)}\|_F \leq \bar{b}\|(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) - (X_{\leq i}^{(k+1)}, X_{>i}^{(k)})\|_F. \quad (5.4)$$

Taking the above argument into consideration, we now turn to prove (5.1).

$$\begin{aligned}
\|\nabla F(\boldsymbol{X}^{(k)})\|_F &= (\sum_{i=1}^{d}\|\nabla_i F(\boldsymbol{X}^{(k)})\|_F^2)^{\frac{1}{2}} \\
&\leq \sum_{i=1}^{d}\|\nabla_i F(\boldsymbol{X}^{(k)})\|_F \\
&\leq \sum_{i=1}^{d}\|\nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})\|_F + \sum_{i=1}^{d}\|\nabla_i F(\boldsymbol{X}^{(k)}) - \nabla_i F(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})\|_F \\
&\leq \bar{b}\sum_{i=1}^{d}\|(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) - (X_{\leq i}^{(k+1)}, X_{>i}^{(k)})\|_F + \hat{b}\sum_{i=1}^{d}\|\boldsymbol{X}^{(k)} - (X_{<i}^{(k+1)}, X_{\geq i}^{(k)})\|_F \\
&= \bar{b}\sum_{i=1}^{d}\|(X_1^{(k+1)} - X_1^{(k+1)}, \ldots, X_{i-1}^{(k+1)} - X_{i-1}^{(k+1)}, X_i^{(k)} - X_i^{(k+1)}, X_{i+1}^{(k)} \\
&\quad - X_{i+1}^{(k)}, \ldots, X_d^{(k)} - X_d^{(k)})\|_F + \hat{b}\sum_{i=1}^{d}\|(X_1^{(k)} - X_1^{(k+1)}, \ldots, X_{i-1}^{(k)} \\
&\quad - X_{i-1}^{(k+1)}, X_i^{(k)} - X_i^{(k)}, \ldots, X_d^{(k)} - X_d^{(k)})\|_F \\
&= \bar{b}\sum_{i=1}^{d}\|(0, \ldots, 0, X_i^{(k)} - X_i^{(k+1)}, 0, \ldots, 0)\|_F + \hat{b}\sum_{i=1}^{d}\|(X_1^{(k)} - X_1^{(k+1)}, \ldots, \\
&\quad X_{i-1}^{(k)} - X_{i-1}^{(k+1)}, 0, \ldots, 0)\|_F \\
&\leq \bar{b}d\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F + \hat{b}d\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F \\
&= (\bar{b} + \hat{b})d\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F,
\end{aligned}$$

where the third inequality holds by (5.2) and (5.4).

(5.1) follows by setting $b_1 := (\bar{b} + \hat{b})d$. □

Lemma 5.5. *Under the boundedness assumption of* $\{\boldsymbol{X}^{(k)}\}$, *let* $\{\boldsymbol{X}^{(k)}\}$ *be a sequence generated by HRAEN* (3.5). *If* $\beta_i^{(k)} \in (0, \frac{1}{\lambda_{\max}(\hat{H}_i^{(k)})})$ *and* $\omega_i^{(k)} \in (0, \frac{2}{1-\beta_i^{(k)}\lambda_{\max}(\hat{H}_i^{(k)})})$ $(1 \leq i \leq d)$,

where $\hat{H}_i^{(k)} := \nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$*, then there exists a constant* $b_2 > 0$*, such that*

$$\|\nabla F(\boldsymbol{X}^{(k)})\|_F \leq b_2 \|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F. \tag{5.5}$$

Proof. In the case of HRAEN, the proofs are almost identical. Similar to the previous lemma, we can obtain the boundedness of $\nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})$ and $\nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)})$. Combining the Newton step in (3.5) with (4.14), we have

$$\begin{aligned}
X_i^{(k+1)} &= X_i^{(k)} - \omega_i^{(k)} \nabla_i \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)}) \left(\nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})\right)^{-1} \\
&= X_i^{(k)} - \omega_i^{(k)} \nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) \left(I - \beta_i^{(k)} (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I)\right) \\
&\quad \cdot (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I)^{-1} \\
&= X_i^{(k)} - \omega_i^{(k)} \nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) \left((M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I)^{-1} - \beta_i^{(k)} I\right).
\end{aligned}$$

Using the compact SVD representation of $\hat{H}$, we have

$$\begin{aligned}
& (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I)^{-1} - \beta_i^{(k)} I \\
(\text{by (4.2)}) \quad & = \hat{H}^{-1} - \beta_i^{(k)} I \\
& = \hat{U}(\hat{\Lambda}^{-1} - \beta_i^{(k)} I)\hat{U}^\top \\
& = \hat{U} \cdot \operatorname{diag}(\ldots, \hat{\lambda}_j^{-1} - \beta_i^{(k)}, \ldots) \cdot \hat{U}^\top,
\end{aligned}$$

where $\hat{H}, \hat{U}, \hat{\Lambda}$ and $\hat{\lambda}_j$ were defined in Proposition 4.3. Since $\beta_i^{(k)} \in (0, \frac{1}{\lambda_{\max}(\hat{H})})$ and $\omega_i^{(k)} \in (0, \frac{2}{1-\beta_i^{(k)}\lambda_{\max}(\hat{H})})$, we have

$$\nabla_i \hat{F}(X_{<i}^{(k+1)}, X_{\geq i}^{(k)}) = (\omega_i^{(k)})^{-1} (X_i^{(k)} - X_i^{(k+1)}) \left((M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I)^{-1} - \beta_i^{(k)} I\right)^{-1}.$$

Thus, proceeding in a similar manner as Lemma 5.4, there exists a constant $b_2 > 0$, such that

$$\|\nabla \hat{F}(\boldsymbol{X}^{(k)})\|_F = (\sum_{i=1}^{d} \|\nabla_i \hat{F}(\boldsymbol{X}^{(k)})\|_F^2)^{\frac{1}{2}} \leq b_2 \|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F.$$

Note that

$$\nabla_i \hat{F}(\boldsymbol{X}^{(k)}) = X_i^{(k)} (M_{-i}^{(k+1)\top} M_{-i}^{(k+1)} + \alpha I) - A_{(i)} M_{-i}^{(k+1)} - \alpha X_i^{(k)} = \nabla_i F(\boldsymbol{X}^{(k)}),$$

which implies (5.5). The proof is completed. □

We summarize the previous two lemmas together.

Lemma 5.6. *Under the boundedness assumption of* $\{\boldsymbol{X}^{(k)}\}$*, let* $\{\boldsymbol{X}^{(k)}\}$ *be a sequence generated by HAEN* (3.4) *or HRAEN* (3.5)*. If* $\beta_i^{(k)} \in (0, \frac{1}{\lambda^*})$ *and* $\omega_i^{(k)} \in (0, \frac{2}{1-\beta_i^{(k)}\lambda^{**}})$*, then there exists a constant* $b > 0$*, such that*

$$\|\nabla F(\boldsymbol{X}^{(k)})\|_F \leq b \|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F.$$

where $b = \max\{b_1, b_2\}$*,* $\lambda^* = \max\{\lambda_{\max}(H_i^{(k)}), \lambda_{\max}(\hat{H}_i^{(k)})\}$*,* $\lambda^{**} = \min\{\lambda_{\max}(H_i^{(k)}), \lambda_{\max}(\hat{H}_i^{(k)})\}$*, where* $H_i^{(k)} := \nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$ and $\hat{H}_i^{(k)} := \nabla_{ii}^2 \hat{F}(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$.

Based on Lemma 5.2 and 5.6, we can derive the following results, that any accumulation point of $\{\boldsymbol{X}^{(k)}\}$ is a stationary point of (2.3).

Theorem 5.7. *Let $\{\boldsymbol{X}^{(k)}\}$ be a bounded sequence generated by HAEN* (3.4) *or HRAEN* (3.5). *If $\beta_i^{(k)} \in \left[\epsilon_1, \frac{1}{\lambda^*} - \epsilon_1\right]$, $\omega_i^{(k)} \in \left[\epsilon_1, \frac{2}{1-\beta_i^{(k)}\lambda^{**}} - \epsilon_1\right]$, where λ^*, λ^{**} are defined in Lemma 5.6 and $\epsilon_1 > 0$ is a small enough but fixed constant, then any accumulation point of $\{\boldsymbol{X}^{(k)}\}$ is a stationary point of* (2.3).

Proof. By the boundedness assumption of $\{\boldsymbol{X}^{(k)}\}$, the set of accumulation points of $\boldsymbol{X}^*$ is nonempty. Thus, for any accumulation point $\boldsymbol{X}^*$, it follows that there exists a convergent subsequence $\{\boldsymbol{X}^{(k_l)}\}$ of $\{\boldsymbol{X}^{(k)}\}$ such that $\lim_{l\to\infty} \boldsymbol{X}^{(k_l)} = \boldsymbol{X}^*$. By Lemma 5.2 and 5.6, we have

$$F(\boldsymbol{X}^{(k_l)}) - F(\boldsymbol{X}^{(k_l+1)}) \geq \frac{\epsilon_0}{b^2}\|\nabla F(\boldsymbol{X}^{(k_l)})\|_F^2 \geq \frac{\epsilon_0}{b^2}\|\nabla_i F(\boldsymbol{X}^{(k_l)})\|_F^2, \quad 1 \leq i \leq d.$$

Since $F(\cdot)$ and $\nabla_i F(\cdot)$ are continuous, letting $l \to \infty$ into both sides of the above inequality yields

$$\nabla_i F(\boldsymbol{X}^*) = 0, \qquad 1 \leq i \leq d,$$

i.e., $\boldsymbol{X}^*$ is a stationary point of (2.3). The proof has been completed. □

Finally, with the help of the preceding lemmas, we can now prove the global convergence of HAEN (3.4) and HRAEN (3.5).

Theorem 5.8 (Global convergence of HAEN and HRAEN)**.** *Under the setting of Theorem 5.7, the whole sequence converges to $\boldsymbol{X}^*$ which is a stationary point, i.e.,*

$$\lim_{k\to\infty} \boldsymbol{X}^{(k)} = \boldsymbol{X}^*.$$

Proof. Since the sequence $\{\boldsymbol{X}^{(k)}\}$ is bounded, then there exists a subsequence $\{\boldsymbol{X}^{(k_l)}\}$ of $\{\boldsymbol{X}^{(k)}\}$ converging to $\boldsymbol{X}^*$, i.e., $\lim_{l\to\infty} \boldsymbol{X}^{(k_l)} = \boldsymbol{X}^*$, where $\boldsymbol{X}^*$ is an accumulation point of $\{\boldsymbol{X}^{(k)}\}$. We first establish the assumption that $F(\boldsymbol{X}^{(k)}) > F(\boldsymbol{X}^{(k+1)}) > F(\boldsymbol{X}^*)$ for all k. Note that Lemma 5.2 implies that the sequence $F(\boldsymbol{X}^{(k)})$ is nonincreasing. We assume that $\boldsymbol{X}^{(k)} \neq \boldsymbol{X}^{(k+1)}$ for all k, otherwise the iteration stops, which leads to $F(\boldsymbol{X}^{(k)}) > F(\boldsymbol{X}^{(k+1)})$ for all k. Moreover, since the function $F(\cdot)$ is continuous and Lemma 5.2, it is straightforward to derive that $\lim_{l\to\infty} F(\boldsymbol{X}^{(k_l)}) = F(\boldsymbol{X}^*)$ which implies that $\lim_{k\to\infty} F(\boldsymbol{X}^{(k)}) = F(\boldsymbol{X}^*)$. Next, we can reasonably assume that $F(\boldsymbol{X}^{(k)}) > F(\boldsymbol{X}^*)$ for all k. If not, there would exist integer $k_0 > 0$ such that $F(\boldsymbol{X}^{(k_0)}) = F(\boldsymbol{X}^*)$; by Lemma 5.2

$$\epsilon_0\|\boldsymbol{X}^{(k_0)} - \boldsymbol{X}^{(k_0+1)}\|_F^2 \leq F(\boldsymbol{X}^{(k_0)}) - F(\boldsymbol{X}^{(k_0+1)}) \leq F(\boldsymbol{X}^{(k_0)}) - F(\boldsymbol{X}^*) = 0,$$

and we can get that $\boldsymbol{X}^{(k_0)} = \boldsymbol{X}^{(k_0+1)}$. Using $\boldsymbol{X}^{(k_0+1)}$ in place of $\boldsymbol{X}^{(k_0)}$ and repeating the argument, we get $\boldsymbol{X}^{(k_0)} = \boldsymbol{X}^{(k_0+1)} = \cdots = \boldsymbol{X}^*$, which means iteration terminates. Therefore, in the following we assume that $F(\boldsymbol{X}^{(k)}) > F(\boldsymbol{X}^{(k+1)}) > F(\boldsymbol{X}^*)$ for all k.

Let $\delta > 0$ be such that $B(\boldsymbol{X}^*, \delta) = \{\boldsymbol{X} | \|\boldsymbol{X} - \boldsymbol{X}^*\|_F \leq \delta\} \subset U$, where U is the neighborhood such that the Łojasiewicz inequality (Lemma 5.1) holds around the cluster point $\boldsymbol{X}^*$. Then there must be an integer $k_1 > 0$, such that

$$\boldsymbol{X}^{(k_1)} \in B(\boldsymbol{X}^*, \frac{\delta}{2}) \subset U \quad \textit{and} \quad c_1(F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{1-\mu} \leq \frac{\delta}{2}, \tag{5.6}$$

where $\mu \in (0, 1]$ and $c_1 = \frac{b}{c\epsilon_0(1-\mu)}$. Such k_1 exists due to $\lim_{k\to\infty} F(\boldsymbol{X}^{(k)}) = F(\boldsymbol{X}^*)$.

The key idea of proving the theorem is to establish the following claim: for all $k \geq k_1$, there holds

$$\boldsymbol{X}^{(k)} \in B(\boldsymbol{X}^*, \delta) \tag{5.7}$$

and then

$$\|\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)}\|_F \leq c_1 \left((F(\boldsymbol{X}^{(k)}) - F(\boldsymbol{X}^*))^{1-\mu} - (F(\boldsymbol{X}^{(k+1)}) - F(\boldsymbol{X}^*))^{1-\mu} \right), \tag{5.8}$$

based on which we can prove the limit of $\{\boldsymbol{X}^{(k)}\}$ is $\boldsymbol{X}^*$. We will show this claim by induction method.

When $k = k_1$, by Lemma 5.2, we obtain

$$\|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^{(k_1+1)}\|_F \leq \frac{F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^{(k_1+1)})}{\epsilon_0 \|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^{(k_1+1)}\|_F},$$

which together with Lemma 5.6 yields

$$\|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^{(k_1+1)}\|_F \leq \frac{b(F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^{(k_1+1)}))}{\epsilon_0 \|\nabla F(\boldsymbol{X}^{(k_1)})\|_F}. \tag{5.9}$$

Combining the above inequality (5.9) with Lemma 5.1 gives

$$\|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^{(k_1+1)}\|_F \leq \frac{b}{c\epsilon_0} \frac{F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^{(k_1+1)})}{(F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{\mu}}. \tag{5.10}$$

Consider the concave function $f(t) = t^{1-\mu}$ $(t \geq 0)$, which implies

$$f(t_1) - f(t_2) \geq f'(t_1)(t_1 - t_2), \ \forall t_1, t_2 \geq 0,$$

and so

$$(F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{1-\mu} - (F(\boldsymbol{X}^{(k_1+1)}) - F(\boldsymbol{X}^*))^{1-\mu} \geq (1-\mu) \frac{F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^{(k_1+1)})}{(F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{\mu}}. \tag{5.11}$$

Direct combination of the two above inequalities (5.10) and (5.11) yields

$$\|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^{(k_1+1)}\|_F \leq c_1 \left((F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{1-\mu} - (F(\boldsymbol{X}^{(k_1+1)}) - F(\boldsymbol{X}^*))^{1-\mu} \right), \tag{5.12}$$

where $c_1 = \frac{b}{c\epsilon_0(1-\mu)}$.

Suppose that for $k = k_1 + 1, k_1 + 2, \ldots, k_2$, (5.7) and (5.8) holds. Now we focus on $k = k_2 + 1$. By the method analogous to that used above, for $k = k_2 + 1$, (5.8) holds. Using the triangle inequality and (5.6), we have

$$\begin{aligned}
\|\boldsymbol{X}^{(k_2+1)} - \boldsymbol{X}^*\|_F &\leq \|\boldsymbol{X}^{(k_2+1)} - \boldsymbol{X}^{(k_1)}\|_F + \|\boldsymbol{X}^{(k_1)} - \boldsymbol{X}^*\|_F \\
&\leq \sum_{k=k_1}^{k_2} \|\boldsymbol{X}^{(k+1)} - \boldsymbol{X}^{(k)}\|_F + \frac{\delta}{2} \\
\text{(by (5.12))} \qquad &\leq c_1 \sum_{k=k_1}^{k_2} \left((F(\boldsymbol{X}^{(k)}) - F(\boldsymbol{X}^*))^{1-\mu} - (F(\boldsymbol{X}^{(k+1)}) - F(\boldsymbol{X}^*))^{1-\mu} \right) + \frac{\delta}{2} \\
&\leq c_1 \left((F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{1-\mu} - (F(\boldsymbol{X}^{(k_2+1)}) - F(\boldsymbol{X}^*))^{1-\mu} \right) + \frac{\delta}{2} \\
&\leq c_1 (F(\boldsymbol{X}^{(k_1)}) - F(\boldsymbol{X}^*))^{1-\mu} + \frac{\delta}{2} < \delta,
\end{aligned}$$

i.e., $\boldsymbol{X}^{(k_2+1)} \in B(\boldsymbol{X}^*, \delta)$. As a consequence, the induction method shows that (5.7) holds for all $k \geq k_1$, i.e., given δ sufficiently small, there exists k_1 such that $\|\boldsymbol{X} - \boldsymbol{X}^*\|_F \leq \delta$. Since $\delta > 0$ is arbitrary (subject to $B(\boldsymbol{X}^*, \delta) \subset U$), which means that the whole $\boldsymbol{X}^{(k)}$ converges to $\boldsymbol{X}^*$, i.e., $\lim_{k\to\infty} \boldsymbol{X}^{(k)} = \boldsymbol{X}^*$. Finally, by Theorem 5.7, we conclude that $\boldsymbol{X}^*$ is a stationary point of (2.3). The proof is completed. □

6 Preliminary Numerical Experiments

In this section, we compare the HAEN and HRAEN with standard and regularized version of ALS algorithm in tensor CPD. The experiments were done with Matlab R2020a, Tensorlab [18] and implemented on a desktop computer with i7 CPU 3.0GHz CPU and 16 GB of RAM. In the experiments, we conduct a preliminary comparison of the four algorithms on CPD with different various tensor dimensions and noise level. Consider the input data tensor $\mathscr{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, given as

$$\mathscr{A} := \frac{\mathscr{C}}{\|\mathscr{C}\|_F} + \sigma \frac{\mathscr{N}}{\|\mathscr{N}\|_F},$$

where $\mathscr{C}$ admits a CPD, i.e., $\mathscr{C} = [\![C_1 \ldots, C_d]\!]$, where $C_i \in \mathbb{R}^{n_i \times R}$ $(i = 1, \ldots, d)$ denote the factor matrices and R is the parameter of tensor CP-rank; $\mathscr{N}$ denotes the noise term, and σ controls the noise level. The entries of $C_1, \ldots, C_d$ and $\mathscr{N}$ are drawn from a zero-mean unit-variance Gaussian distribution. All of these algorithms stop either the iterations exceed 1×10^5 or a tolerance relative error of 1×10^{-5} is met the update

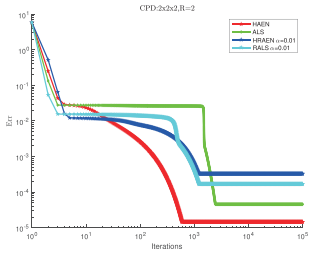
$$\frac{\|\boldsymbol{X}^{(k+1)} - \boldsymbol{X}^{(k)}\|_F}{\|\boldsymbol{X}^{(k+1)}\|_F}$$

between two subsequent iterates. The initial guess $\boldsymbol{X}^{(0)}$ is also drawn form a zero-mean unit-variance Gaussian distribution, and we set the column size of each factor to be R. We use the built-in function `normest` in Matlab to estimate $\lambda_{\max}(\cdot)$ in the experiments. Accounting for the constraint of $\beta_i^{(k)}$ and $\omega_i^{(k)}$ $(1 \leq i \leq d)$ is updated in each iterate, we adopt the fraction structure and introduce the parameters h and o to indirectly control $\beta_i^{(k)}$ and $\omega_i^{(k)}$ respectively. Take HAEN for example, $\beta_i^{(k)} = (\text{normest}(H_i^{(k)})h)^{-1}$, $\omega_i^{(k)} = 2((1 - \beta_i^{(k)}\text{normest}(H_i^{(k)}))o)^{-1}$, where $H_i^{(k)} := \nabla_{ii}^2 F(X_{<i}^{(k+1)}, \bar{X}_i^{(k)}, X_{>i}^{(k)})$. Empirically, in each experiment, we set $h = 14,\ o = 1.5$ for HAEN and $h = 1.8,\ o = 3.5,\ \alpha = 0.01$ for HRAEN. For RALS, the regularization parameter α is fixed to 0.01.

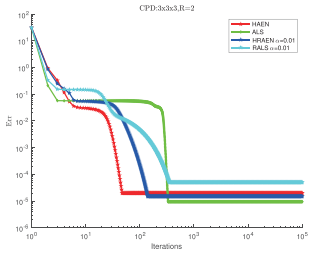
At first, we give six examples of 3-way tensor CPD with no noise in Fig.1 to illustrate the resistance of the swamp phenomenon, the accuracy of the decomposition, and the diminution of iterations of our algorithms compared with classic methods in various scale tensor approximation. The first two examples are given as follows and the details of others are presented in the appendix. In Fig.1, the x-axes denotes the iterations range from 1 to 100000, and the y-axes denotes the Error $\|\mathscr{A} - [\![A^0, B^0, C^0]\!]\|_F$. The figures shows our algorithms take advantage in iterations-cost or accuracy compared with classic methods, and both effectively alleviate the swamp, while it seems that HAEN performs better. In addition, it can be seen from the figures that ALS is affected by the swamp mostly , while HAEN overcomes the swamp better than the other algorithms.

Example 6.1 $(2 \times 2 \times 2)$**.** Let the matrices

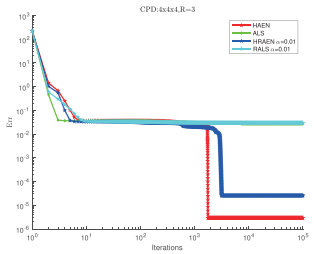
$$A = \begin{bmatrix} -0.1441 & 0.9117 \\ -0.2607 & 2.3062 \end{bmatrix},\ B = \begin{bmatrix} 0.8476 & 0.0047 \\ 0.6427 & 0.1298 \end{bmatrix},\ C = \begin{bmatrix} 0.5814 & -1.4926 \\ -0.553 & 0.4291 \end{bmatrix}$$



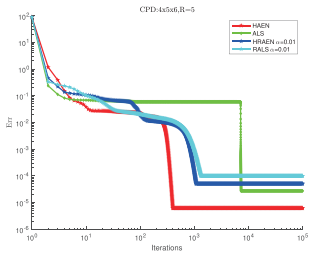
(a) $2 \times 2 \times 2$ $R = 2$



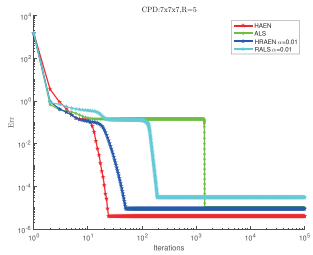
(b) $3 \times 3 \times 3$ $R = 2$



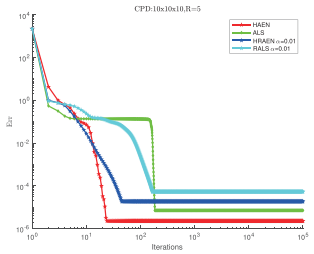
(c) $4 \times 4 \times 4$ $R = 3$



(d) $4 \times 5 \times 6$ $R = 5$



(e) $7 \times 7 \times 7$ $R = 5$



(f) $10 \times 10 \times 10$ $R = 5$

Figure 1: Six examples for CPD with swamp occurring

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{2} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following initial matrices:

$$A^0 = \begin{bmatrix} 0.4905 & 0.2301 \\ 0.1438 & 2.3655 \end{bmatrix}, \ B^0 = \begin{bmatrix} -0.7813 & -0.6737 \\ -0.1589 & 0.714 \end{bmatrix}, \ C^0 = \begin{bmatrix} 0.2552 & 0.7421 \\ -0.3273 & -1.0057 \end{bmatrix}$$

Example 6.2 $(3 \times 3 \times 3)$**.** Let the matrices

$$A = \begin{bmatrix} -1.853 & 1.3504 \\ 0.3656 & 0.0522 \\ -0.9224 & 0.2292 \end{bmatrix}, \ B = \begin{bmatrix} -0.7874 & 0.2162 \\ 0.523 & -0.3203 \\ 0.0649 & 0.6434 \end{bmatrix}, \ C = \begin{bmatrix} -2.4392 & -0.1907 \\ 0.1657 & 0.5781 \\ 1.0482 & -0.5955 \end{bmatrix}$$

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{2} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following initial matrices:

$$A^0 = \begin{bmatrix} 0.4625 & -1.8335 \\ -2.0939 & 0.1853 \\ -0.15 & 0.765 \end{bmatrix}, \ B^0 = \begin{bmatrix} 1.1796 & 0.7754 \\ 0.753 & -0.0567 \\ 0.4841 & -0.0385 \end{bmatrix}, \ C^0 = \begin{bmatrix} 1.0204 & -1.7973 \\ 1.0873 & -2.2953 \\ -0.4708 & -1.6226 \end{bmatrix}$$

Secondly, we report the four algorithms for CPD with various size and noise level in Table 1 and Table 2 (each case is averaged over 50 instances), where Err. $= \|\mathscr{A} - [\![X_1, \ldots, X_d]\!]\|_F$ represents the accuracy, and "Iter." denotes the iterations.

Table 1: HAEN, ALS, HRAEN ($\alpha = 0.01$) and RALS ($\alpha = 0.01$) for solving CPD with different size and noise level σ. Each case is averaged over 50 instances. Err. $= \|\mathscr{A} - [\![X_1, \ldots, X_d]\!]\|_F$ represents the accuracy, and "Iter." denotes the iterations.

			HAEN			ALS			HRAEN			RALS		
n_1, n_2, n_3	R	σ	Err.	Iter.	Time	Err.	Iter.	Time	Err.	Iter.	Time	Err.	Iter.	Time
10,10,10	5	0.1	0.04	88	0.02	0.04	270	0.06	0.04	63	0.02	0.04	142	0.03
10,10,10	5	0.5	0.19	31	0.01	0.19	74	0.02	0.19	115	0.03	0.19	150	0.03
15,15,15	5	0.1	0.03	338	0.09	0.03	770	0.18	0.03	56	0.02	0.03	198	0.05
15,15,15	5	0.5	0.13	28	0.01	0.13	115	0.03	0.13	74	0.02	0.13	119	0.03
20,20,20	5	0.1	0.02	22	0.01	0.02	495	0.15	0.02	146	0.05	0.02	230	0.07
20,20,20	5	0.5	0.10	26	0.01	0.10	86	0.03	0.10	48	0.02	0.10	121	0.04
30,30,30	10	0.1	0.02	180	0.11	0.02	618	0.31	0.02	388	0.22	0.02	689	0.35
30,30,30	10	0.5	0.09	54	0.03	0.09	317	0.16	0.09	159	0.10	0.09	439	0.23
40,50,60	10	0.1	0.01	265	0.25	0.02	948	0.69	0.01	96	0.09	0.08	812	0.60
40,50,60	10	0.5	0.06	52	0.05	0.06	913	0.67	0.06	98	0.09	0.06	1032	0.77
50,60,70	20	0.1	0.02	197	0.41	0.02	1482	2.35	0.01	175	0.34	0.11	1456	2.29
50,60,70	20	0.5	0.07	214	0.55	0.07	305	0.55	0.07	97	0.23	0.07	3465	6.29
70,80,90	20	0.1	0.02	626	3.64	0.03	1353	6.26	0.02	439	2.36	0.14	1810	7.32
70,80,90	20	0.5	0.06	848	4.58	0.06	2013	8.35	0.05	398	1.97	0.05	3873	15.10

From Table 1, it can be seen that HAEN and HRAEN require less iterations to converge and keep the less time-cost in most cases versus RALS ($\alpha = 0.01$). When compared with ALS, it is easily seen that HAEN requires less or the same level iterations and time-cost in most cases. In the case that the order $d = 4$, HAEN and HRAEN still more efficient versus RALS from Table 2. Notably, HAEN performs better and needs less iterations and time-cost for solving 4-way tensor CPD.

Table 2: HAEN, ALS, HRAEN ($\alpha = 0.01$) and RALS ($\alpha = 0.01$) for solving CPD with different size and noise level σ. Each case is averaged over 50 instances. Err. $= \|\mathscr{A} - [\![X_1, \ldots, X_d]\!]\|_F$ represents the accuracy, and "Iter." denotes the iterations.

			HAEN			ALS			HRAEN			RALS		
n_1, n_2, n_3, n_4	R	σ	Err.	Iter.	Time	Err.	Iter.	Time	Err.	Iter.	Time	Err.	Iter.	Time
10,10,10,10	5	0.1	0.01	28	0.02	0.02	37	0.02	0.01	348	0.19	0.01	1245	0.59
10,10,10,10	5	0.5	0.07	41	0.02	0.08	389	0.19	0.08	120	0.07	0.07	792	0.39
15,15,15,15	5	0.1	0.01	27	0.03	0.01	63	0.05	0.01	840	0.82	0.07	940	0.68
15,15,15,15	5	0.5	0.04	39	0.04	0.04	68	0.05	0.04	325	0.32	0.04	1224	0.90
20,20,20,20	5	0.1	0.00	25	0.05	0.01	446	0.65	0.01	284	0.54	0.18	592	0.86
20,20,20,20	5	0.5	0.02	35	0.07	0.03	41	0.06	0.04	501	0.95	0.03	3369	4.79
30,30,30,30	5	0.1	0.00	24	0.22	0.00	34	0.25	0.02	971	8.44	0.26	1106	7.78
30,30,30,30	5	0.5	0.01	85	0.72	0.01	104	0.72	0.02	806	6.98	0.04	7283	50.33
40,40,40,40	5	0.1	0.00	32	0.86	0.00	40	0.87	0.10	281	7.38	0.40	1313	27.25
40,40,40,40	5	0.5	0.02	36	0.96	0.02	496	10.32	0.01	1231	32.17	0.40	1431	29.78
50,50,50,50	5	0.1	0.00	25	1.67	0.01	510	25.65	0.16	190	12.41	0.48	1819	92.58
50,50,50,50	5	0.5	0.01	42	2.65	0.01	108	5.38	0.02	895	54.99	0.48	1612	80.21

The above experiments show that, compared with the standard and the regularized version of the ALS algorithm respectively, HAEN and HRAEN can not only alleviate the occurrence of swamp better, but also reduce the time-cost and iterations in most cases.

7 Concluding Remarks

We proposed hybrid alternating extra-gradient and Newton's methods, namely HAEN and HRAEN, for tensor decomposition. Specifically, for each subproblem, the correction step of the extra-gradient is replaced by a Newton step. If the partial Hessian is singular, its pseudoinverse is used to replace the inverse. As discussed in Remark 4.4, the choice of the step-size ω related to the correction step can be possibly chosen large, giving more flexibility to the algorithms. Under mild assumptions, HAEN and HRAEN achieve global convergence. As shown in the experiments, our algorithms are more efficient, and more resist to the swamp, compared to the standard and the regularized version of the ALS algorithm. A potential execution that would reduce the efficiency of the proposed algorithms is the computation of the largest eigenvalue of the partial Hessian when choosing the step-sizes. Although we use the Matlab built-in function `normest` to accelerate the computation, if R is large, this might still not be efficient. A possible alternative is to use line search instead.

References

[1] P.A. Absil, R. Mahony and B. Andrews, Convergence of the iterates of descent methods for analytic cost functions, *SIAM J. Optim.* 16 (2005) pp. 531–547.

[2] J.C.A. Barata and M.S. Hussein, The Moore-Penrose pseudoinverse: A tutorial review of the theory, *Braz. J. Phys.* 42 (2012) 146–165.

[3] J.D. Carroll and J.J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition, *Psychometrika* 35 (1970) 283–319.

[4] P. Comon, X. Luciani and A.L. De Almeida, Tensor decompositions, alternating least squares and other tales, *J. Chemometr.* 23 (2009) 393–405.

[5] W. Hackbusch, Tensor spaces and numerical tensor calculus, *Springer Science & Business Media* 42 (2012) 3–20.

[6] R.A. Harshman, Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis, *UCLA working papers in phonetics* 16 (1970) 1–84.

[7] F. Jiang, D. Han and X. Zhang, A trust-region-based alternating least-squares algorithm for tensor decompositions, *J. Comput. Math.* 36 (2018) 351–373.

[8] T.G. Kolda and B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (2009) 455–500.

[9] G.M. Korpelevich, The extragradient method for finding saddle points and other problems, *Matecon* 12 (1976) 747–756.

[10] N. Li, S. Kindermann and C. Navasca, Some convergence results on the regularized alternating least-squares method for tensor decomposition, *Linear Algebra Appl.* 438 (2013) 796–812.

[11] X. Mao, G. Yuan and Y. Yang, A self-adaptive regularized alternating least squares method for tensor decomposition problems, *Anal. Appl.* 18 (2020) 129–147.

[12] C. Navasca, L. De Lathauwer and S. Kindermann, Swamp reducing technique for tensor decomposition, in: *2008 16th European Signal Processing Conference*, IEEE, 2008, pp. 1–5.

[13] T.P. Nguyen, E. Pauwels, E. Richard and B. W. Suter, Do, Extragradient method in optimization: convergence and complexity, *J. Optim. Theory Appl.* 176 (2018) 137–162.

[14] P. Paatero, A weighted non-negative least squares algorithm for three-way 'parafac' factor analysis, *Chemom. Intell. Lab. Syst.* 38 (1997) 223–242.

[15] J.D. Sidiropoulos, R. Bro and G. B. Giannakis, Parallel factor analysis in sensor array processing, *IEEE Trans. Signal Process.* 48 (2000) 2377–2388.

[16] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis and C. Faloutsos, Tensor decomposition for signal processing and machine learning, *IEEE Trans. Signal Process.* 65 (2017) 3551–3582.

[17] G. Tomasi and R. Bro, A comparison of algorithms for fitting the parafac model, *Comput. Stat. Data Anal.* 50 (2006), 1700–1734.

[18] N. Vervliet, O. Debals and L. De Lathauwer, Tensorlab 3.0–numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization, in: *2016 50th Asilomar Conference on Signals, Systems and Computers*, IEEE, 2016, pp. 1733–1738.

[19] X. Wang, C. Navasca and S. Kindermann, On accelerating the regularized alternating least-squares algorithm for tensors, *Electron. Trans. Numer. Anal.* 48 (2018) 1–14.

[20] M. Yang, A. Milzarek, Z. Wen and T. Zhang, A stochastic extra-step quasi-newton method for nonsmooth nonconvex optimization, *Math. Program.* (2021), https://doi.org/10.1007/s10107-021-01629-y.

Manuscript received 27 June 2021
revised 24 August 2021
accepted for publication 8 October 2021

Junwei Zhang
College of Mathematics and Information Science
Guangxi University, Nanning, 530004, China
E-mail address: JW2hang@outlook.com

Yuning Yang
College of Mathematics and Information Science
Guangxi University, Nanning, 530004, China
E-mail address: yyang@gxu.edu.cn

A More Examples

Example A.1 $(4 \times 4 \times 4)$**.** Let the matrices

$$A = \begin{bmatrix} -1.3077 & 2.7694 & -0.0631 \\ -0.4336 & -1.3499 & 0.7147 \\ 0.3426 & 3.0349 & -0.205 \\ 3.5784 & 0.7254 & -0.1241 \end{bmatrix},$$

$$B = \begin{bmatrix} 1.4897 & -1.2075 & 1.0347 \\ 1.409 & 0.7172 & 0.7269 \\ 1.4172 & 1.6302 & -0.3034 \\ 0.6715 & 0.4889 & 0.2939 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.7873 & -0.8095 & -0.7549 \\ 0.8884 & -2.9443 & 1.3703 \\ -1.1471 & 1.4384 & -1.7115 \\ -1.0689 & 0.3252 & -0.1022 \end{bmatrix}$$

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{3} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following initial matrices:

$$A^0 = \begin{bmatrix} -0.6003 & -0.1941 & -1.0722 \\ 0.49 & -2.1384 & 0.961 \\ 0.7394 & -0.8396 & 0.124 \\ 1.7119 & 1.3546 & 1.4367 \end{bmatrix},$$

$$B^0 = \begin{bmatrix} -1.9609 & 0.8252 & -0.2725 \\ -0.1977 & 1.379 & 1.0984 \\ -1.2078 & -1.0582 & -0.2779 \\ 2.908 & -0.4686 & 0.7015 \end{bmatrix},$$

$$C^0 = \begin{bmatrix} -2.0518 & 0.508 & 1.1275 \\ -0.3538 & 0.282 & 0.3502 \\ -0.8236 & 0.0335 & -0.2991 \\ -1.5771 & -1.3337 & 0.0229 \end{bmatrix}.$$

Example A.2 $(4 \times 5 \times 6)$**.** Let the matrices

$$A = \begin{bmatrix} 0.3464 & -0.887 & 0.5039 & -0.6132 & -1.1153 \\ -0.2611 & 0.7414 & -0.8225 & -0.659 & 0.6672 \\ -0.1847 & 1.3922 & 0.201 & -0.8332 & 0.7953 \\ -0.1017 & 2.4739 & -1.0071 & 0.3782 & 1.0375 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.0204 & 1.7725 & -0.5603 & 0.567 & -1.2874 \\ 0.6190 & -2.5088 & -1.2265 & -0.0014 & 0.2181 \\ 1.8031 & -0.4566 & 0.793 & 0.6239 & -1.5666 \\ 0.0530 & 2.4304 & -2.1099 & 0.1264 & 0.7833 \\ -0.1779 & -0.4715 & -0.7994 & 0.6809 & -0.3106 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.6553 & -1.8324 & -1.5868 & -0.5771 & 0.043 \\ -0.5375 & 0.8486 & -1.0191 & -0.8364 & -0.9489 \\ 0.3286 & 0.4052 & -1.3852 & 0.853 & 0.5416 \\ 1.0541 & -0.7025 & 0.9549 & 0.4773 & -0.8211 \\ -1.9797 & 1.499 & -0.6011 & 0.3023 & -1.0719 \\ -1.8674 & 0.1378 & -1.1719 & 0.4158 & -1.0741 \end{bmatrix}$$

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{5} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following initial matrices:

$$A^0 = \begin{bmatrix} -0.3904 & -1.3157 & -1.8269 & 0.3176 & 1.0395 \\ -0.6901 & 0.6271 & 0.5359 & 0.1628 & -0.1168 \\ -0.4482 & -1.3313 & 0.6929 & 1.1591 & -0.6483 \\ 0.7144 & -0.1308 & -0.6875 & 0.1284 & -0.038 \end{bmatrix},$$

$$B^0 = \begin{bmatrix} -0.1993 & 1.1886 & -0.3627 & 0.3189 & 1.0571 \\ 0.8817 & -0.4149 & -0.7388 & -1.4565 & -0.5876 \\ -0.0561 & -0.6586 & -1.4798 & 1.3218 & 0.3022 \\ -0.8412 & 0.984 & 0.1534 & 0.4872 & 0.2195 \\ -0.1548 & 0.1012 & -0.317 & -1.5666 & -0.8782 \end{bmatrix},$$

$$C^0 = \begin{bmatrix} -2.8614 & 0.6712 & -0.8926 & -0.2359 & 0.4998 \\ 0.5356 & 0.1509 & -0.8953 & 0.6528 & -0.4841 \\ -0.342 & -0.9913 & 0.3133 & 1.9645 & 0.2384 \\ -0.5979 & 0.8368 & 0.667 & 0.8846 & 0.7782 \\ 0.4189 & 0.4745 & 0.8281 & 0.0845 & 0.9243 \\ -0.6897 & 1.2522 & 0.0065 & -0.5745 & 0.5881 \end{bmatrix}.$$

Example A.3 ($7 \times 7 \times 7$)**.** Let the matrices

$$A = \begin{bmatrix} -0.7929 & -0.2036 & 1.676 & 0.1909 & -1.9683 \\ 0.9308 & 0.0179 & -0.3251 & -0.7348 & -1.6861 \\ -1.3504 & 0.1004 & 0.1011 & 0.7884 & -0.0869 \\ 0.7998 & 0.8764 & -0.5767 & -1.9654 & 0.3074 \\ 0.5996 & 0.7007 & -0.0615 & -1.9555 & 0.3375 \\ 1.1004 & 0.652 & 0.7256 & 1.6243 & 1.067 \\ -1.4197 & 0.1785 & -1.2273 & -0.1494 & 0.3329 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.0735 & -1.4831 & 0.9551 & 0.7039 & -1.0478 \\ 1.0478 & -0.1065 & -0.1308 & -0.7526 & -0.7354 \\ 2.0589 & 0.2453 & 0.6494 & -0.0169 & -1.3566 \\ 0.0267 & 0.1458 & 0.5985 & -0.4417 & 0.7286 \\ -0.2405 & -1.1181 & -0.1925 & 1.7013 & 0.6072 \\ 1.0385 & -0.5969 & -1.9808 & -0.0306 & -0.8283 \\ -0.5072 & -0.6795 & 0.3629 & 0.2039 & 2.8876 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.9191 & -1.0383 & -0.7637 & 0.505 & 0.0866 \\ 0.4058 & 0.5294 & 0.5495 & 0.0405 & 1.5126 \\ 1.9162 & -2.2171 & -0.3746 & -0.2547 & 1.7933 \\ -2.4557 & 0.1074 & -1.6592 & -0.5477 & 1.1026 \\ -0.0911 & 0.4389 & -0.3102 & 1.6766 & -0.3258 \\ -0.0247 & -0.041 & -0.6199 & 1.173 & -0.5828 \\ 0.1999 & 0.3643 & -1.0246 & -0.1891 & 0.7516 \end{bmatrix}$$

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{5} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following initial matrices:

$$A^0 = \begin{bmatrix} 0.8322 & 0.4448 & 0.1098 & 1.8829 & -0.3927 \\ 0.9481 & 1.1409 & -1.6547 & 0.0555 & -0.622 \\ -1.9737 & 0.4477 & 1.1107 & -0.6139 & -1.1905 \\ -0.3919 & 0.3154 & -2.1079 & 0.587 & -1.8785 \\ -0.6767 & 0.9456 & -0.5498 & -1.2067 & -0.424 \\ -0.016 & 0.4287 & 0.0943 & 0.5453 & 0.7772 \\ 0.5152 & -1.3246 & -0.0382 & 0.2509 & -0.7139 \end{bmatrix},$$

$$B^0 = \begin{bmatrix} 1.5846 & -0.7576 & -0.06 & 1.5607 & 1.7421 \\ -0.8883 & 0.7386 & 1.3857 & 1.5862 & -2.0015 \\ 2.1408 & -1.1144 & 1.2178 & 0.8563 & 0.8355 \\ -0.6922 & -1.7059 & -1.4951 & -1.4245 & -0.3428 \\ 0.0993 & 0.6612 & 0.0373 & 0.0397 & -0.478 \\ 1.435 & -1.7296 & 0.8029 & -1.3799 & -0.8891 \\ 1.2334 & -2.1381 & 0.9739 & 1.2331 & 1.2634 \end{bmatrix},$$

$$C^0 = \begin{bmatrix} 0.3832 & 1.5857 & -0.5491 & 0.9986 & 0.0531 \\ -0.1189 & 1.2502 & 0.2837 & -0.7573 & -0.237 \\ 0.4172 & -0.1156 & 0.2128 & 0.5961 & -0.0627 \\ 1.0132 & -1.3318 & -2.2028 & 2.1232 & 1.2711 \\ -0.8695 & -2.3428 & 1.2511 & 1.3117 & 0.2211 \\ -0.7947 & -0.9266 & 2.0247 & -0.6999 & 1.664 \\ 0.6885 & 1.1296 & -0.0389 & -1.0196 & -0.043 \end{bmatrix}.$$

Example A.4 $(10 \times 10 \times 10)$**.** Let the matrices

$$A = \begin{bmatrix} 1.3779 & -0.6862 & -0.2128 & 0.5963 & -0.2002 \\ 1.8512 & -0.8133 & -0.044 & -0.1135 & 0.9404 \\ -1.8977 & 0.7934 & 0.4582 & 0.807 & 0.3492 \\ -1.7787 & -0.3819 & -0.4414 & -0.0898 & 1.8593 \\ -0.9226 & -1.3712 & -1.0549 & -0.0063 & 0.9271 \\ -1.9979 & 0.0103 & -0.1556 & -0.0919 & -1.227 \\ -0.3571 & 0.2041 & 0.1291 & -0.9212 & -0.3272 \\ -0.3364 & -0.411 & 0.5094 & -0.927 & 0.8916 \\ 0.2504 & 0.6636 & -0.0301 & -0.9612 & 0.2882 \\ 0.2862 & 0.2258 & -0.4574 & 1.7848 & 2.2652 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.0479 & -0.1031 & -0.6543 & 0.4805 & 0.1769 \\ -1.5519 & -2.799 & 1.2449 & -0.3868 & 3.4663 \\ 0.4441 & 0.3933 & -1.2923 & 0.4216 & -0.2146 \\ -0.9118 & 0.9902 & -0.6144 & 1.0877 & 0.4863 \\ 0.0494 & -1.2976 & 0.2417 & -2.2493 & 0.3309 \\ 1.0780 & -1.522 & 0.5493 & 1.8045 & 1.2679 \\ 0.3082 & 0.621 & 0.4676 & -0.6321 & 1.0905 \\ 0.2996 & -1.5075 & 0.1915 & 1.3165 & -0.9465 \\ -0.1972 & -1.6794 & -0.2298 & 1.5516 & -0.4385 \\ -0.1464 & 0.789 & -0.5792 & -1.4689 & 0.3432 \end{bmatrix},$$

$$C = \begin{bmatrix} -0.0584 & 0.8960 & 1.2951 & -0.2298 & -1.0618 \\ 2.535 & -1.8135 & 2.7681 & -1.4617 & 0.4505 \\ 0.4386 & 1.5667 & -0.4953 & -2.8823 & -0.2728 \\ 0.4375 & 0.8465 & 0.4688 & -0.0475 & -0.1015 \\ -0.8377 & 0.1102 & -0.6573 & -0.4625 & -1.4291 \\ -1.3075 & -1.1611 & -1.717 & -0.5766 & -0.7644 \\ 0.7941 & -0.3975 & 1.4705 & -0.846 & 0.4101 \\ -0.1973 & 0.2543 & 0.6941 & -1.8172 & -0.7899 \\ 0.6492 & 1.2078 & -0.5107 & -0.5217 & 0.1616 \\ -0.8315 & -1.0335 & 0.1134 & 0.1614 & 1.9779 \end{bmatrix}.$$

be the three factor matrices of aim tensor $\mathscr{A} = \sum_{i=1}^{5} \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$, where the vectors $\boldsymbol{a}_i$, $\boldsymbol{b}_i$ and $\boldsymbol{c}_i$ are the ith columns of A, B and C respectively. CPD is calculated with the following

initial matrices:

$$A^0 = \begin{bmatrix} 0.5661 & -1.0402 & -0.1138 & 0.8761 & 0.5523 \\ 0.3759 & 0.9973 & 1.9371 & -0.8765 & 1.8201 \\ -0.277 & -0.0261 & -0.1708 & -1.2174 & 0.3426 \\ 0.3501 & -0.657 & 0.3012 & -1.6148 & 0.1796 \\ -0.2913 & 0.6777 & 0.326 & 1.8401 & -1.0193 \\ 0.1861 & -0.5108 & 0.9248 & -0.8189 & 0.0376 \\ 0.5766 & 0.446 & 0.2153 & 0.9921 & 0.1371 \\ 0.3397 & 1.5166 & 0.3662 & 0.5338 & -1.5211 \\ -0.6728 & 0.9378 & 0.3222 & -1.5267 & -0.0189 \\ -0.537 & -0.1602 & 2.689 & 2.0229 & 0.1632 \end{bmatrix},$$

$$B^0 = \begin{bmatrix} -0.7212 & 0.6000 & -0.5644 & 2.4516 & 0.0574 \\ 0.4106 & 0.5714 & 1.0433 & -1.4626 & 1.0658 \\ -1.2126 & 0.6851 & 0.846 & -0.6355 & 1.6207 \\ -0.5737 & 1.0094 & -0.4955 & -0.3855 & 0.1219 \\ 0.1054 & 0.9909 & -0.2068 & -0.9423 & -1.238 \\ -0.6051 & 0.0337 & -0.1558 & -0.6738 & 0.2441 \\ 0.4218 & -0.4503 & -0.2754 & -1.9242 & 1.3983 \\ -0.3628 & -0.1107 & -2.4432 & -0.1124 & -0.0955 \\ -0.8741 & 1.2379 & -0.4273 & -0.5185 & 0.3876 \\ 0.9316 & -1.1979 & 0.3091 & 0.535 & -0.9663 \end{bmatrix},$$

$$C^0 = \begin{bmatrix} 1.5092 & 0.0837 & 0.2021 & 1.0382 & -0.5708 \\ 0.4038 & 0.3746 & 0.8763 & 0.3305 & 1.3047 \\ -0.4221 & 2.6532 & 0.8079 & 0.4758 & -0.0426 \\ -1.674 & 0.3327 & -1.6033 & -2.0905 & 0.8955 \\ -0.6876 & 0.1408 & -2.3621 & -0.174 & 2.2849 \\ -1.0272 & 1.5778 & -0.7017 & 0.0192 & 0.0668 \\ -0.4926 & 0.0895 & 1.6519 & -0.86 & 1.4946 \\ 0.3468 & -0.673 & 0.2351 & -0.0229 & -1.0725 \\ 0.8294 & 0.9319 & -0.1518 & -0.6023 & 1.8233 \\ 0.1556 & -0.3579 & -0.1559 & 0.8699 & -1.2084 \end{bmatrix}.$$