Applications of Mathematics

Douglas Bunker; Lixing Han; Shu Hua Zhang

A proximal ANLS algorithm for nonnegative tensor factorization with a periodic enhanced line search

Applications of Mathematics, Vol. 58 (2013), No. 5, 493–509

Persistent URL: http://dml.cz/dmlcz/143429

58 (2013) APPLICATIONS OF MATHEMATICS No. 5, 493–509

A PROXIMAL ANLS ALGORITHM FOR NONNEGATIVE TENSOR FACTORIZATION WITH A PERIODIC ENHANCED LINE SEARCH

Douglas Bunker, Lixing Han, Flint, Shuhua Zhang[1], Tianjin

(Received June 27, 2011)

Abstract. The Alternating Nonnegative Least Squares (ANLS) method is commonly used for solving nonnegative tensor factorization problems. In this paper, we focus on algorithmic improvement of this method. We present a Proximal ANLS (PANLS) algorithm to enforce convergence. To speed up the PANLS method, we propose to combine it with a periodic enhanced line search strategy. The resulting algorithm, PANLS/PELS, converges to a critical point of the nonnegative tensor factorization problem under mild conditions. We also provide some numerical results comparing the ANLS and PANLS/PELS methods.

Keywords: nonnegative tensor factorization, proximal method, alternating least squares, enhanced line search, global convergence

MSC 2010: 15A69, 65K05, 65F99

1. Introduction

The PARAFAC model [15], also called CANDECOMP [5], decomposes a tensor as the sum of outer products of vectors. It has proved to be a very useful tool to analyze multidimensional data arising in a variety of applications [2], [17], [30], [31]. In some cases, the data tensor is nonnegative and the factors are required to be nonnegative [9], [17]. This leads to the so-called Nonnegative Tensor Factorization (NTF) model, which can be stated as:

Let $\mathcal{A} \in \mathbb{R}_+^{I_1 \times I_2 \times \ldots \times I_N}$ be an order-N ($N \geqslant 3$) nonnegative tensor and K a positive integer. The NTF finds nonnegative matrix factors $X_n \in \mathbb{R}_+^{I_n \times K}$ ($n = 1, 2, \ldots, N$) such that

$$\mathcal{A} \approx [[X_1, X_2, \ldots, X_N]] \tag{1.1}$$

[1] Corresponding author

where $[[X_1, X_2, \ldots, X_N]]$ is the Kruskal operator [9], [17]

$$[[X_1, X_2, \ldots, X_N]] = \sum_{i=1}^{K} \mathbf{x}_1^i \circ \mathbf{x}_2^i \circ \ldots \circ \mathbf{x}_N^i, \tag{1.2}$$

in which $\mathbf{x}_n^i$ is the ith ($i = 1, 2, \ldots, K$) column vector of the matrix X_n and the symbol "$\circ$" denotes the vector outer product.

The NTF model can also be regarded as an extension of the Nonnegative Matrix Factorization (NMF) [18], [24] from order-2 to higher order arrays. This model has recently found a number of important applications involving multidimensional data. For a comprehensive introduction of NTF and NMF, we refer the reader to the recently published book [9] and the references therein.

For a given nonnegative tensor $\mathcal{A}$, currently there is no direct method to find nonnegative matrix factors $X_1, X_2, \ldots, X_N$ for $N \geqslant 3$. It is typical to reformulate (1.1) as an optimization problem which minimizes some type of distance between $\mathcal{A}$ and $[[X_1, X_2, \ldots, X_N]]$ and the most widely used distance is the Euclidean distance. From now on, we call the following nonnegatively constrained minimization problem

$$\min_{X_1 \geqslant 0, \ldots, X_N \geqslant 0} f(X_1, X_2, \ldots, X_N) = \frac{1}{2}\|\mathcal{A} - [[X_1, X_2, \ldots, X_N]]\|^2 \tag{1.3}$$

the NTF problem, where the tensor norm is defined as

$$\|\mathcal{B}\|^2 = \sum_{i_1=1, i_2=1, \ldots, i_N=1}^{I_1, I_2, \ldots, I_N} b^2_{i_1, i_2, \ldots, i_N} \tag{1.4}$$

for a tensor $\mathcal{B} = (b_{i_1, i_2, \ldots, i_N}) \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$. In [19], Lim and Comon proved that the problem (1.3) always has a global minimizer. It is worth mentioning that the approximation problem without nonnegativity constraints

$$\min \frac{1}{2}\|\mathcal{A} - [[X_1, X_2, \ldots, X_N]]\|^2$$

is ill-posed when $N \geqslant 3$ [19]. This gives another motivation to study NTF and its solutions.

Several algorithms have been proposed to solve (1.3) (see, for example, [3], [6], [7], [8], [11], [12], [16], [21], [23], [28], [29]). An attractive approach is the Alternating Nonnegative Least Squares (ANLS) method. This method takes the advantage of the decoupled structure of the objective function and often works well in practice [3], [9], [11], [16]. However, it lacks a satisfactory convergence theory. In fact, Grippo and Sciandrone [13] gave a counter-example showing that a block nonlinear Gauss-Seidel method does not converge to a critical point if the number of blocks is greater or equal to three. The ANLS method for NTF is a special case of the Gauss-Seidel method considered in [13]. Therefore, this method may not converge to a critical point of (1.3) when $N \geqslant 3$.

It has been found that the Alternating Least Squares (ALS) algorithm for tensor decomposition can be slow in some cases, for example, when two factors are almost collinear [3], [15], [20]. Several strategies have been proposed to speed up the ALS method (see for example, [3], [25], [26]). In particular, Rajih, Comon, and Harshman [25] proposed an enhanced line search strategy which can often significantly improve the performance of the ALS method. As an extention of the ALS method to the NTF model, the ANLS method is expected to be slow sometimes. Therefore, an acceleration scheme is desirable for speeding up this method.

In this paper, we focus on algorithmic improvement of the ANLS method. To enforce convergence, we propose a framework that incorporates a proximal point scheme into the ANLS method. This leads to the so-called PANLS method. We also propose to use an enhanced line search periodically to accelerate the PANLS method. We prove that the resulting PANLS/PELS method converges to a critical point of the nonnegative tensor factorization problem under mild conditions. We provide some numerical results to compare the ANLS and PANLS/PELS algorithms.

The paper is organized as follows. We give necessary conditions for a local minimizer of (1.3) in Section 2, which can be used to terminate an algorithm for NTF. In Section 3, we present the proximal ANLS (PANLS) framework for NTF and provide its convergence properties. Then in Section 4, we present Algorithm PANLS/PELS which combines the PANLS method with a periodic enhanced line search scheme. We also provide a convergence result for this algorithm. We describe an implementation of PANLS/PELS and provide some numerical results in Section 5. Some final remarks are given in Section 6. Our notations are similar to those used in [9], [17].

2. Necessary conditions for optimality

Let $A_{(n)} \in \mathbb{R}^{I_n \times I_1 \dots I_{n-1} I_{n+1} \dots I_N}$ denote the mode-n unfolding matrix of tensor $\mathcal{A}$ and $X_{\odot -n}$ the Khatri-Rao product

$$X_{\odot -n} = X_N \odot \dots \odot X_{n+1} \odot X_{n-1} \odot \dots \odot X_1, \tag{2.1}$$

for $n = 1, 2, \dots, N$. Then the objective function in (1.3) can be rewritten in matrix form

$$f(X_1, X_2, \dots, X_N) = \frac{1}{2} \| A_{(n)} - X_n (X_{\odot -n})^T \|_F^2, \quad n = 1, 2, \dots, N, \tag{2.2}$$

where the matrix norm $\|\cdot\|_F$ is the Frobenius norm. The partial gradient of f with respect to X_n $(n = 1, 2, \ldots, N)$ is

$$\nabla_{X_n} f(X_1, \ldots, X_N) = -A_{(n)} X_{\odot-n} + X_n[(X_{\odot-n})^T X_{\odot-n}] \in \mathbb{R}^{I_n \times K}. \tag{2.3}$$

For a point $(X_1, \ldots, X_N) \in \mathbb{R}^{I_1 \times K} \times \mathbb{R}^{I_2 \times K} \times \ldots \times \mathbb{R}_+^{I_n \times K} \times \ldots \times \mathbb{R}^{I_N \times K}$, we denote the projected partial gradient with respect to $\mathbb{R}_+^{I_n \times K}$ by $\mathcal{P}_{(n)} \nabla_{X_n} f$, which is defined as

$$[\mathcal{P}_{(n)} \nabla_{X_n} f(X_1, \ldots, X_N)]_{ij} = \begin{cases} [\nabla_{X_n} f]_{ij} & \text{if } [X_n]_{ij} > 0, \\ \min\{0, [\nabla_{X_n} f]_{ij}\} & \text{if } [X_n]_{ij} = 0, \end{cases} \tag{2.4}$$

for $n = 1, 2, \ldots, N$.

Problem (1.3) is an optimization problem with nonnegativity constraints. The following lemma gives necessary conditions for its local minimizers, which can be used to terminate an algorithm for NTF. These conditions are equivalent to the Karush-Kuhn-Tucker conditions.

Lemma 1. *If* $(X_1, X_2, \ldots, X_N) \in \mathbb{R}_+^{I_1 \times K} \times \mathbb{R}_+^{I_2 \times K} \times \ldots \times \mathbb{R}_+^{I_N \times K}$ *is a local minimizer of the problem* (1.3), *then*

$$\mathcal{P}_{(n)} \nabla_{X_n} f(X_1, \ldots, X_N) = 0, \tag{2.5}$$

for $n = 1, 2, \ldots, N$.

We call a point $(X_1, X_2, \ldots, X_N)$ satisfying (2.5) a critical point of the problem (1.3).

3. A proximal ANLS framework for NTF

We start with a description of the ANLS method for NTF. This method starts from initial nonnegative matrix factors $X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)}$. At the kth iteration, assume that $(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})$ is obtained. The method generates the next iteration by alternately solving the subproblems

$$\min_{X_n \geqslant 0} \Phi_n^{(k)}(X_n) = \frac{1}{2}\|A_{(n)} - X_n(X_{\odot-n}^{(k)})^T\|_F^2, \tag{3.1}$$

where

$$X_{\odot-n}^{(k)} = X_N^{(k)} \odot \ldots \odot X_{n+1}^{(k)} \odot X_{n-1}^{(k+1)} \odot \ldots \odot X_1^{(k+1)}, \tag{3.2}$$

for $n = 1, 2, \ldots, N$. Every subproblem (3.1) is a convex optimization problem and therefore, each of its minimizers is a global minimizer. Moreover, $X_n \in \mathbb{R}_+^{I_n \times K}$ is a minimizer of (3.1) if and only if

$$\|\mathcal{P}\nabla\Phi_n^{(k)}(X_n)\|_F = 0, \tag{3.3}$$

where

$$\nabla\Phi_n^{(k)}(X_n) = -A_{(n)}X_{\odot-n}^{(k)} + X_n(X_{\odot-n}^{(k)})^T X_{\odot-n}^{(k)} \tag{3.4}$$

and

$$[\mathcal{P}\nabla\Phi_n^{(k)}(X_n)]_{ij} = \begin{cases} [\nabla\Phi_n^{(k)}(X_n)]_{ij} & \text{if } [X_n]_{ij} > 0, \\ \min\{0, [\nabla\Phi_n^{(k)}(X_n)]_{ij}\} & \text{if } [X_n]_{ij} = 0. \end{cases} \tag{3.5}$$

Algorithm 1 (ANLS)

Step 0. *Initialization.* Choose initial nonnegative matrices: $X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)}$. Set $k = 0$.

Step 1. *Check Termination.* If the termination criterion is met, stop.

Step 2. *Main Iteration.*

for $n = 1 : N$

Solve

$$X_n^{(k+1)} = \underset{X_n \geqslant 0}{\operatorname{argmin}}\ \Phi_n^{(k)}(X_n) \tag{3.6}$$

end

Step 3. Set $k = k + 1$ and go to Step 1.

The ANLS method has been implemented in several papers (see, for example, [3], [9], [11], [16]). The differences among these implementations are how the subproblems (3.6) are solved. The ANLS method often works well in practice. However, it may not converge to a critical point of (1.3) when $N \geqslant 3$ [13].

The proximal point method was proposed by Rockafellar [27]. It has been widely used in optimization to enforce convergence. In [13], Grippo and Sciandrone showed that a proximal point modification of their block Gauss-Seidel method has a nice global convergence property. Motivated by the work of Grippo and Sciandrone, we present a Proximal ANLS (PANLS) framework for NTF.

The PANLS method is similar to the ANLS method in structure. It alternately solves the subproblems

$$\min_{X_n \geqslant 0} \Psi_n^{(k)}(X_n), \tag{3.7}$$

for $n = 1, 2, \ldots, N$, where $\beta_n^{(k)} \geqslant 0$ is a scalar, and

$$\Psi_n^{(k)}(X_n) = \frac{1}{2}\|A_{(n)} - X_n(X_{\odot-n}^{(k)})^T\|_F^2 + \frac{\beta_n^{(k)}}{2}\|X_n - X_n^{(k)}\|_F^2. \tag{3.8}$$

We note that for $\beta_n^{(k)} \geqslant 0$, the subproblem (3.8) is a convex problem. If $\beta_n^{(k)} > 0$, then the objective function $\Psi_n^{(k)}$ is strictly convex and therefore (3.8) has a unique minimizer. The factor $X_n \in \mathbb{R}_+^{I_n \times K}$ is the minimizer of (3.8) if and only if

$$\|\mathcal{P}\nabla\Psi_n^{(k)}(X_n)\|_F = 0, \tag{3.9}$$

where

$$\nabla\Psi_n^{(k)}(X_n) = -A_{(n)}X_{\odot-n}^{(k)} + X_n(X_{\odot-n}^{(k)})^T X_{\odot-n}^{(k)} + \beta_n^{(k)}(X_n - X_n^{(k)}) \tag{3.10}$$

and $\mathcal{P}\nabla\Psi_n^{(k)}(X_n)$ is defined in a similar manner to (3.5).

Algorithm 2 (PANLS)

Step 0. *Initialization.* Choose initial nonnegative matrices: $X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)}$. Set $k = 0$.

Step 1. *Check Termination.* If the termination criterion is met, stop.

Step 2. *Main Iteration.* for $n = 1 : N$

Choose nonnegative $\beta_n^{(k)}$ and solve

$$X_n^{(k+1)} = \underset{X_n \geqslant 0}{\operatorname{argmin}}\ \Psi_n^{(k)}(X_n), \tag{3.11}$$

end

Step 3. Set $k = k + 1$ and go to Step 1.

Similar to Proposition 7 in [13], we have the following convergence result for the PANLS method.

Theorem 1 (Convergence of PANLS)**.** *Suppose that there exist constants M and m such that*

$$0 < m \leqslant \beta_n^{(k)} \leqslant M,$$

for $n = 1, 2, \ldots, N$ and for all $k \geqslant 0$. If the sequence $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}$ generated by PANLS *has limit points, then every limit point $(X_1^*, X_2^*, \ldots, X_N^*)$ is a critical point of* (1.3).

4. Accelerating PANLS by Periodic Enhanced Line Search

To speed up the ALS method for PARAFAC, Rajih, Comon, and Harshman [25] proposed an enhanced line search (ELS) technique: At the kth iteration, it introduces $\Delta_n^{(k)} = X_n^{(k)} - X_n^{(k-1)}$ for $n = 1, 2, \ldots, N$, and does a line search along the direction $(\Delta_1^{(k)}, \Delta_2^{(k)}, \ldots, \Delta_N^{(k)})$ by solving

$$\alpha^{(k)} = \operatorname*{argmin}_{\alpha \in \mathbb{R}} \; g(\alpha), \tag{4.1}$$

where

$$g(\alpha) = \|A_{(1)} - (X_1^{(k-1)} + \alpha \Delta_1^{(k)})((X_N^{(k-1)} + \alpha \Delta_N^{(k)}) \odot \ldots \odot (X_2^{(k-1)} + \alpha \Delta_2^{(k)}))^T\|_F^2.$$

The ELS strategy then defines $\widetilde{X}_n^{(k)} = X_n^{(k-1)} + \alpha^{(k)} \Delta_n^{(k)}$ and uses the ALS method to update the matrix factors based on $(\widetilde{X}_1^{(k)}, \ldots, \widetilde{X}_N^{(k)})$. Numerical experiments in [25] show that using ELS can improve the performance of the ALS method. Recently Nion and De Lathauwer [22] have applied the ELS strategy to solve complex-valued tensor decomposition problems. Comon et al. [10], [28] have indicated that the ELS strategy is applicable to any iterative optimization, and a collection of Matlab codes is freely available [33].

The objective function g in (4.1) is a polynomial of α with a degree of $2N$ and a positive leading coefficient. Solving the one-dimensional minimization problem (4.1) for $\alpha^{(k)}$ is a relatively easy task compared to the updating of matrix factors. However, evaluating g can be time consuming for large scale problems. In this case, using an enhanced line search at every iteration can take significant time. Moreover, it is unnecessary to use an ELS when the updates (3.11) make a satisfactory progress. Therefore, it seems beneficial to use the enhanced line search less frequently. In this section, we extend the enhanced line search idea to the PANLS method for NTF. In particular, the ELS is used after every T iterations for some positive integer $T \geqslant 2$. We remark that the periodic execution of ELS in the standard ALS method has been proposed in the literature, see for example, [10]. To avoid an excessively large step size $\alpha^{(k)}$, we impose some upper and lower bounds on α: $\alpha_1 \leqslant \alpha \leqslant \alpha_2$.

Algorithm 3 (PANLS/PELS)

Step 0. *Initialization.* Choose integer $T \geqslant 2$, $\alpha_1 < 0$, $\alpha_2 > 0$, and initial nonnegative matrices: $X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)}$. Set $k = 0$.

Step 1. *Check Termination.* If the termination criterion is met, stop.

Step 2. *Periodic Enhanced Line Search.* If $k \geqslant T$ and $\mathrm{mod}(k, T) = 0$, compute $\alpha^{(k)}$ by solving

$$\alpha^{(k)} = \operatorname*{argmin}_{\alpha \in [\alpha_1, \alpha_2]} g(\alpha) \tag{4.2}$$

and set $\widetilde{X}_n^{(k)} = X_n^{(k-1)} + \alpha^{(k)}\Delta_n^{(k)}$; otherwise, set $\widetilde{X}_n^{(k)} = X_n^{(k)}$ for $n = 1, 2, \ldots, N$.
Step 3. *Main Iteration.*

for $n = 1:\ N$

Compute

$$\widetilde{X}_{\odot-n}^{(k)} = \widetilde{X}_N^{(k)} \odot \ldots \odot \widetilde{X}_{n+1}^{(k)} \odot X_{n-1}^{(k+1)} \odot \ldots \odot X_1^{(k+1)}.$$

Choose nonnegative $\beta_n^{(k)}$ and find $X_n^{(k+1)}$ by solving

$$\min_{X_n \geqslant 0} \Theta_n^{(k)}(X_n) = \frac{1}{2}\|A_{(n)} - X_n(\widetilde{X}_{\odot-n}^{(k)})^T\|_F^2 + \frac{\beta_n^{(k)}}{2}\|X_n - \widetilde{X}_n^{(k)}\|_F^2, \tag{4.3}$$

end

Step 4. Set $k = k + 1$ and go to Step 1.

We note that if $\beta_n^{(k)} \geqslant 0$, then subproblem (4.3) is a convex optimization problem. If $\beta_n^{(k)} > 0$, the objective function $\Theta_n^{(k)}$ is strictly convex and therefore (4.3) has a unique minimizer. Moreover, $X_n \in \mathbb{R}_+^{I_n \times K}$ is the minimizer of (4.3) if and only if

$$\|\mathcal{P}\nabla\Theta_n^{(k)}(X_n)\|_F = 0, \tag{4.4}$$

where

$$\nabla\Theta_n^{(k)}(X_n) = -A_{(n)}\widetilde{X}_{\odot-n}^{(k)} + X_n(\widetilde{X}_{\odot-n}^{(k)})^T\widetilde{X}_{\odot-n}^{(k)} + \beta_n^{(k)}(X_n - \widetilde{X}_n^{(k)}) \tag{4.5}$$

and $\mathcal{P}\nabla\Theta_n^{(k)}(X_n)$ is similarly defined as in (3.5).

We now present a convergence result for the PANLS/PELS algorithm.

Theorem 2 (Convergence of PANLS/PELS)**.** *Suppose that there exist constants M and m such that*

$$0 < m \leqslant \beta_n^{(k)} \leqslant M,$$

for $n = 1, 2, \ldots, N$ and for all $k \geqslant 0$. Assume the PANLS/PELS *method generates an infinite sequence of approximations $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}$. If the subsequence $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}_{\mathrm{mod}(k,T)\neq 0}$ has limit points, then each of these limit points is a critical point of* (1.3).

Remark 1. When $\mathrm{mod}(k, T) = 0$, $(\widetilde{X}_1^{(k)}, \ldots, \widetilde{X}_N^{(k)})$ may be an infeasible point of (1.3) due to the enhanced line search (4.2). Our convergence result focuses on the subsequence $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}_{\mathrm{mod}(k,T)\neq 0}$. In this case, $\widetilde{X}_n^{(k)} = X_n^{(k)}$ for $n = 1, 2, \ldots, N$. Taking this into account, Theorem 2 is proved by modifying the proof of Proposition 7 in [13].

P r o o f. Let $J_0 = \{k \mid \mathrm{mod}(k,T) \neq 0\}$. Then for all $k \in J_0$ and for $n = 1, 2, \ldots, N$, we have

$$\mathcal{P}\nabla\Theta_n^{(k)}(X_n^{(k+1)}) = 0, \tag{4.6}$$

where

$$\nabla\Theta_n^{(k)}(X_n^{(k+1)}) = -A_{(n)}X_{\odot-n}^{(k)} + X_n^{(k+1)}(X_{\odot-n}^{(k)})^T X_{\odot-n}^{(k)} + \beta_n^{(k)}(X_n^{(k+1)} - X_n^{(k)}) \tag{4.7}$$

and

$$[\mathcal{P}\nabla\Theta_n^{(k)}(X_n^{(k+1)})]_{ij} = \begin{cases} [\nabla\Theta_n^{(k)}(X_n^{(k+1)})]_{ij} & \text{if } [X_n^{(k+1)}]_{ij} > 0, \\ \min\{0, [\nabla\Theta_n^{(k)}(X_n^{(k+1)})]_{ij}\} & \text{if } [X_n^{(k+1)}]_{ij} = 0, \end{cases} \tag{4.8}$$

and moreover,

$$\begin{aligned} f(X_1^{(k+1)}, X_2^{(k+1)}, \ldots, X_N^{(k+1)}) & \\ \leqslant f(X_1^{(k+1)}, \ldots, X_n^{(k+1)}, X_{n+1}^{(k)}, \ldots, X_N^{(k)}) & \\ \leqslant f(X_1^{(k+1)}, \ldots, X_{n-1}^{(k+1)}, X_n^{(k)}, \ldots, X_N^{(k)}) - \frac{\beta_n^{(k)}}{2}\|X_n^{(k+1)} - X_n^{(k)}\|_F^2 & \\ \leqslant f(X_1^{(k+1)}, \ldots, X_{n-1}^{(k+1)}, X_n^{(k)}, \ldots, X_N^{(k)}) - \frac{m}{2}\|X_n^{(k+1)} - X_n^{(k)}\|_F^2 & \\ \leqslant f(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}). & \end{aligned} \tag{4.9}$$

Assume that $(X_1^*, X_2^*, \ldots, X_N^*)$ is a limit point of the subsequence $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}_{k \in J_0}$. Then there exists an infinite set of indices $J_1 \subset J_0$ such that

$$\lim_{k\to\infty, k\in J_1} (X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}) = (X_1^*, X_2^*, \ldots, X_N^*). \tag{4.10}$$

Clearly, $(X_1^*, X_2^*, \ldots, X_N^*) \in \mathbb{R}_+^{I_1 \times K} \times \mathbb{R}_+^{I_2 \times K} \times \ldots \times \mathbb{R}_+^{I_N \times K}$. Expressions (4.10) and (4.9) imply that

$$\lim_{k\to\infty, k\in J_0} f(X_1^{(k+1)}, X_2^{(k+1)}, \ldots, X_N^{(k+1)}) - f(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)}) = 0.$$

Using (4.9) again gives

$$\lim_{k\to\infty, k\in J_1} \|X_n^{(k+1)} - X_n^{(k)}\|_F^2 = 0, \quad n = 1, 2, \ldots, N.$$

This and (4.10) imply

$$\lim_{k\to\infty, k\in J_1}(X_1^{(k+1)},\dots,X_n^{(k+1)},X_{n+1}^{(k)},\dots,X_N^{(k)}) = (X_1^*,X_2^*,\dots,X_N^*), \tag{4.11}$$

for $n = 1,2,\dots,N$. Now letting $k\to\infty$, $k\in J_1$ and taking the limit in (4.6), we obtain

$$\mathcal{P}_{(n)}\nabla_{X_n} f(X_1^*,X_2^*,\dots,X_N^*) = 0, \quad n = 1,2,\dots,N.$$

Thus $(X_1^*,X_2^*,\dots,X_N^*)$ is a critical point of NTF problem (1.3). □

5. Numerical tests

5.1. Implementation.

We have presented the ANLS, PANLS, and PANLS/PELS methods in a general framework which can be implemented in various ways, using different termination criteria and different algorithms to solve their subproblems.

We have implemented the PANLS/PELS method in Matlab [32] in conjunction with the Tensor Toolbox of Bader and Kolda [1]. For comparison purpose, we have also implemented the ANLS and PANLS methods.

Our termination criteria are based on Lemma 1. Specifically, we terminate PANLS/PELS, PANLS, and ANLS if

$$\frac{\mathrm{PGN}^{(k)}}{\mathrm{PGN}^{(0)}} \leqslant \mathrm{Tol}, \tag{5.1}$$

for some given tolerance Tol, where $\mathrm{PGN}^{(k)}$ denotes the norm of the projected gradient at point $(X_1^{(k)},X_2^{(k)},\dots,X_N^{(k)})$:

$$\mathrm{PGN}^{(k)} = \sqrt{\sum_{n=1}^{N}\|\mathcal{P}_{(n)}\nabla_{X_n} f(X_1^{(k)},X_2^{(k)},\dots,X_N^{(k)})\|_F^2}. \tag{5.2}$$

A key step in PANLS/PELS, PANLS, or ANLS is to solve subproblems (4.3), (3.11), or (3.6). Since $\widetilde{X}_n^{(k)}$ and $\widetilde{X}_{\odot-n}^{(k)}$ in (4.3) may no longer be nonnegative, a nonnegative least squares solver that can handle infeasible starting points is needed. We have used a slightly modified PBBNLS2, a nonmonotone projected Barzilai-Borwein method for nonnegative least squares problem proposed in [14] to solve subproblems (4.3), (3.11), and (3.6). The PBBNLS2 algorithm is terminated when

$$\|\mathcal{P}\nabla\Theta_n^{(k)}(X_n)\|_F,\ \|\mathcal{P}\nabla\Psi_n^{(k)}(X_n)\|_F,\ \text{or}\ \|\mathcal{P}\nabla\Phi_n^{(k)}(X_n)\|_F \leqslant \max\left\{\frac{1}{5^k}, 10^{-8}\right\}. \tag{5.3}$$

With this choice of tolerance, the subproblems are not solved very accurately in the first few iterations. After 12 iterations, however, the tolerance becomes 10^{-8} which leads to fairly good approximate solutions to the subproblems. We have found that this strategy is cost effective through our numerical tests.

In PANLS/PELS and PANLS, we set the value of $\beta_n^{(k)}$ to be

$$\beta_n^{(k)} = \max\Big\{\frac{1}{2^k}, 10^{-3}\Big\}, \quad n = 1, 2, \ldots, N,$$

for $k = 1, 2, \ldots$ The parameter T is set to be $T = 5$ in PANLS/PELS. We have found through numerical experiments that this choice of T is generally suitable. We use the Matlab built-in function `fminbnd` to carry out the line search (4.2) in PANLS/PELS. The `fminbnd` finds a local minimum of a function of one variable within a given interval $[\alpha_1, \alpha_2]$ ([32]). In our numerical tests, we used the values $\alpha_1 = -10^4$, $\alpha_2 = 10^4$.

Finally, in PANLS/PELS, if $\operatorname{mod}(k, T) = 0$ and the modified PBBNLS2 method does not obtain a nonnegative $X_n^{(k+1)}$ for some n within the allocated computational budget (which is set to be 1000 inner iterations), we reset $\widetilde{X}_n^{(k)} = X_n^{(k)}$ for $n = 1, 2, \ldots, N$ and resolve (4.3) using the new $\widetilde{X}_n^{(k)}$'s.

Remark 2. We wish to point out that other termination criteria for ANLS have been used in the literature. For example, one such criterion is to terminate ANLS when the difference between objective function values of two consecutive iterations is within a given tolerance. Some other methods for solving the ANLS subproblems (3.6) have also been used (see, for example, [3], [9], [11], [16]). These methods can be used to solve the PANLS and PANLS/PELS subproblems. We remark that if the same stopping criteria for ANLS, PANLS, and PANLS/PELS and their subproblems are used, the three algorithms implemented using different subproblem solvers should exhibit similar relative performance, e.g., in terms of the number of iterations used.

In our implementation of the ANLS, PANLS, and PANLS/PELS algorithms, the gradient of the objective function (1.3) and the gradients of the subproblem functions (3.1), (3.8), and (4.3) are computed at every iteration in both the inner and the outer loops to test termination conditions. This is not most cost effective in terms of CPU time. However, this implementation serves our purpose in this paper well, that is, to compare the relative performance of the three algorithms.

5.2. Numerical results.

We carried out our numerical tests on a Dell Optiplex 755 computer with 4 GB of RAM and a 3 GHz Intel Core Due CPU E8400 running Windows Vista.

To assess the performance of the ANLS, PANLS, and PANLS/PELS methods, we first tested them on some randomly generated order-3 and order-4 NTF problems. In our tests, the tensor $\mathcal{A}$ was generated by

$$\mathcal{A} = [[Y_1, Y_2, \ldots, Y_N]], \tag{5.4}$$

where

$$Y_n = \text{rand}(I_n, M), \quad n = 1, 2, \ldots, N,$$

and $N = 3$ or 4. If $M \leqslant K$, then $\mathcal{A}$ is exactly factorizable, i.e., the objective value of the problem (1.3) at a global minimizer is 0. When $M > K$, this objective value is generally nonzero. The initial matrix factors were chosen as

$$X_n^{(0)} = \text{rand}(I_n, K), \quad n = 1, 2, \ldots, N. \tag{5.5}$$

For each tensor $\mathcal{A}$, we ran each of PANLS/PELS, PANLS, and ANLS five times. In each run, the three algorithms used the same randomly generated $(X_1^{(0)}, X_2^{(0)}, \ldots, X_N^{(0)})$ as in (5.5). The three algorithms were terminated when the stopping condition (5.1) was satisfied. We tried both $\text{Tol} = 10^{-6}$ and $\text{Tol} = 10^{-7}$ as the tolerance value. At the termination of an algorithm, we recorded the means of their corresponding numbers of iterations used, the CPU times used, values of the final $\text{PGN}^{(k)}$ as defined in (5.2), and residual norms computed by

$$\|\mathcal{A} - [[X_1, X_2, \ldots, X_N]]\|$$

in `NIT`, `Time`, `PGN`, and `RN`, respectively.

We observed that the final `RN` values obtained by using tolerances $\text{Tol} = 10^{-6}$ and $\text{Tol} = 10^{-7}$ for problems with $M > K$ are very close—the difference between the `RN` values obtained using the two tolerances is smaller than 10^{-6} for all problems, which is neglectable since `RN` > 48. This indicates that $\text{Tol} = 10^{-6}$ is a suitable tolerance for problems with $M > K$. For exactly factorizable problems with $M \leqslant K$, the value of `RN` is 0 at a global minimizer. The final `RN` values obtained by using tolerances $\text{Tol} = 10^{-6}$ and $\text{Tol} = 10^{-7}$ are more distinguishable.

We found that the values of `NIT`, `Time`, `PGN`, and `RN` are very close for PANLS and ANLS. Therefore, we only report the numerical results of ANLS and PANLS/PELS here. For the PANLS/PELS method, we also report `LStime`, the mean values of times used by the enhanced line search.

We summarize the numerical results in Tables 1, 2, and 3 with $M = 5$, $K = 5$, $\text{Tol} = 10^{-7}$, $M = 5$, $K = 6$, $\text{Tol} = 10^{-7}$, and $M = 10$, $K = 5$, $\text{Tol} = 10^{-6}$ respectively. From these tables, we observe that on this set of problems, the PANLS/PELS method significantly outperforms the ANLS method in terms of the number of iterations used and the CPU time used.

Problem	Algorithm	NIT	Time (LStime)	PGN	RN
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	75.6	1.96 (0.49)	0.49×10^{-3}	0.76×10^{-4}
$I_3 = 50$	ANLS	240.2	4.53	0.50×10^{-3}	0.82×10^{-4}
$I_1 = 100,\ I_2 = 100,$	PANLS/PELS	77.8	16.89 (4.39)	0.23×10^{-2}	0.21×10^{-3}
$I_3 = 100$	ANLS	231.8	36.62	0.25×10^{-2}	0.24×10^{-3}
$I_1 = 100,\ I_2 = 150,$	PANLS/PELS	74.2	47.71 (12.62)	0.56×10^{-2}	0.28×10^{-3}
$I_3 = 200$	ANLS	215.4	102.36	0.63×10^{-2}	0.32×10^{-3}
$I_1 = 25,\ I_2 = 25,$	PANLS/PELS	40.80	4.17 (0.84)	0.67×10^{-3}	0.56×10^{-4}
$I_3 = 25,\ I_4 = 25$	ANLS	102.4	8.37	0.79×10^{-3}	0.69×10^{-4}
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	42.4	81.52 (14.34)	0.62×10^{-2}	0.19×10^{-3}
$I_3 = 50,\ I_4 = 50$	ANLS	115.2	185.04	0.68×10^{-2}	0.21×10^{-3}
$I_1 = 20,\ I_2 = 40,$	PANLS/PELS	38.8	48.56 (9.71)	0.51×10^{-2}	0.13×10^{-3}
$I_3 = 60,\ I_4 = 80$	ANLS	96.6	97.64	0.54×10^{-2}	0.14×10^{-3}

Table 1. Numerical results for problems with $M = 5,\ K = 5$

Problem	Algorithm	NIT	Time (LStime)	PGN	RN
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	93.2	10.07 (0.85)	0.65×10^{-3}	0.34×10^{-3}
$I_3 = 50$	ANLS	378.2	35.48	0.69×10^{-3}	0.25×10^{-3}
$I_1 = 100,\ I_2 = 100,$	PANLS/PELS	144	47.73 (11.16)	0.31×10^{-2}	0.12×10^{-2}
$I_3 = 100$	ANLS	514	138.70	0.35×10^{-2}	0.13×10^{-3}
$I_1 = 100,\ I_2 = 150,$	PANLS/PELS	88.6	86.30 (25.19)	0.73×10^{-2}	0.60×10^{-3}
$I_3 = 200$	ANLS	272.4	186.30	0.10×10^{-1}	0.17×10^{-2}
$I_1 = 25,\ I_2 = 25,$	PANLS/PELS	101.4	23.63 (3.70)	0.11×10^{-2}	0.67×10^{-3}
$I_3 = 25,\ I_4 = 25$	ANLS	236	46.88	0.12×10^{-2}	0.57×10^{-3}
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	93.2	231.35 (59.02)	0.10×10^{-1}	0.22×10^{-2}
$I_3 = 50,\ I_4 = 50$	ANLS	209.4	386.52	0.11×10^{-1}	0.16×10^{-2}
$I_1 = 20,\ I_2 = 40,$	PANLS/PELS	123.6	211.80 (60.33)	0.54×10^{-2}	0.14×10^{-2}
$I_3 = 60,\ I_4 = 80$	ANLS	330.8	407.97	0.64×10^{-2}	0.13×10^{-2}

Table 2. Numerical results for problems with $M = 5,\ K = 6$

Our second numerical experiment involves the order-3 nonnegative tensor arising from Sugar Production using Fluorescence Spectroscopy introduced in [4], where $\mathcal{A} \in \mathbb{R}^{268\times 571\times 7}$. We tested and compared the ANLS and PANLS/PELS methods on this tensor with $K = 4$ and $K = 6$, using $\mathrm{Tol} = 10^{-6}$ and the same randomly

Problem	Algorithm	NIT	Time (LStime)	PGN	RN
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	137.6	4.75 (1.23)	0.99×10^{-2}	48.7965
$I_3 = 50$	ANLS	569.6	8.85	0.10×10^{-1}	48.7965
$I_1 = 100,\ I_2 = 100,$	PANLS/PELS	150	36.30 (12.89)	0.55×10^{-1}	148.1935
$I_3 = 100$	ANLS	566.4	88.63	0.56×10^{-1}	148.1935
$I_1 = 100,\ I_2 = 150,$	PANLS/PELS	162.6	118.66 (42.39)	0.15×10^{0}	261.8624
$I_3 = 200$	ANLS	567	267.57	0.15×10^{0}	261.8624
$I_1 = 25,\ I_2 = 25,$	PANLS/PELS	78.4	9.08 (2.46)	0.87×10^{-2}	59.1337
$I_3 = 25,\ I_4 = 25$	ANLS	240	19.99	0.10×10^{-2}	59.1337
$I_1 = 50,\ I_2 = 50,$	PANLS/PELS	79.4	168.36 (41.55)	0.12×10^{0}	273.2424
$I_3 = 50,\ I_4 = 50$	ANLS	243.4	392.14	0.13×10^{0}	273.2424
$I_1 = 20,\ I_2 = 40,$	PANLS/PELS	77.6	111.54 (30.89)	0.77×10^{-1}	219.6184
$I_3 = 60,\ I_4 = 80$	ANLS	244.8	256.23	0.82×10^{-1}	219.6184

Table 3. Numerical results for problems with $M = 10$, $K = 5$

generated initial nonnegative $(X_1^{(0)}, X_2^{(0)}, X_3^{(0)})$. The two methods are compared based on their ability to reduce the ratio

$$\varrho_k = \frac{\|\mathcal{A} - [[X_1^{(k)}, X_2^{(k)}, X_3^{(k)}]]\| - \|\mathcal{A} - [[X_1^*, X_2^*, X_3^*]]\|}{|\mathcal{A} - [[X_1^{(0)}, X_2^{(0)}, X_3^{(0)}]]\| - \|\mathcal{A} - [[X_1^*, X_2^*, X_3^*]]\|},$$

where k denotes the iteration index, and the optimal (X_1^*, X_2^*, X_3^*) is obtained by running the corresponding algorithm using a smaller tolerance Tol $= 10^{-8}$. We report the numerical results in Figures 1 and 2. As can be seen, PANLS/PELS is faster than ANLS.

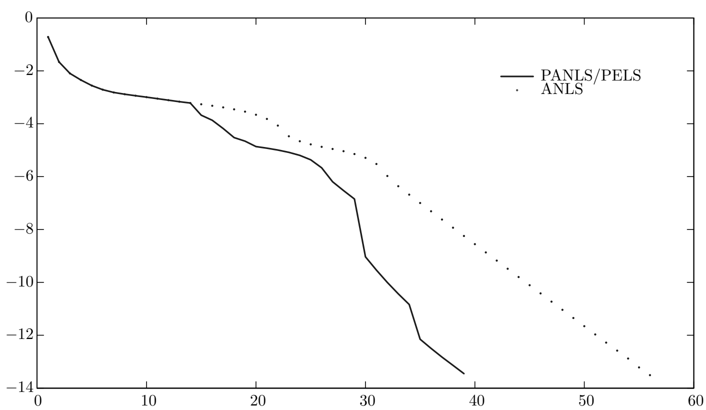

Figure 1. Comparison of ANLS and PANLS/PELS on the Sugar Production tensor when $K = 4$, where the x-axis represents the iteration index k and the y-axis represents the log-scaled ratio ϱ_k, i.e., $\log_{10} \varrho_k$.

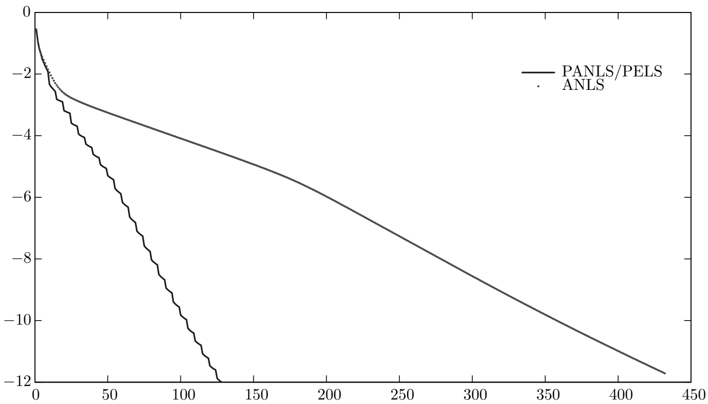

Figure 2. Comparison of ANLS and PANLS/PELS on the Sugar Production tensor when $K = 6$, where the x-axis represents the iteration index k and the y-axis represents the log-scaled ratio ϱ_k, i.e., $\log_{10} \varrho_k$.

6. Final remarks

In order to improve the ANLS method for NTF, we propose to use a proximal technique to enforce its convergence. We also use a periodic enhanced line search to speed up the convergence. The resulting method, PANLS/PELS, is globally convergent in the sense that each limit point of the subsequence $\{(X_1^{(k)}, X_2^{(k)}, \ldots, X_N^{(k)})\}_{\mathrm{mod}(k,T)\neq 0}$ $(T \geqslant 2)$ is a critical point of the nonnegative tensor factorization problem. Our numerical tests show that the PANLS/PELS method outperforms the ANLS method in terms of the number of iterations used and the CPU time used.

We have presented the PANLS/PELS algorithm (i.e., Algorithm 3) as a general framework. It can be implemented in various different ways. In our implementation, we used a slightly modified version of the PBBNLS2 method [14] to solve the subproblem (4.3). This subproblem can be solved using other methods, such as the ones developed in [3], [9], [11], [16]. The enhanced line search (4.2) can also be carried out using other approaches. In [25], [22], the authors explicitly express the objective function $g(\alpha)$ as a polynomial of degree 6 for order-3 tensors. We have observed that this strategy can save computing time for order-3 large scale problems. It remains to be seen if this is the case for tensors of higher orders (i.e., $N \geqslant 4$).

The effectiveness of using the periodic ELS strategy has been demonstrated by numerical experiments in [10], [28] and in this paper. It is desirable to investigate how to make more efficient use of ELS. For example, one possibility is to use ELS only when the updates (3.11) do not make a satisfactory progress. It is also desirable to compare the PANLS/PELS algorithm with the algorithms proposed in [28]. We leave both problems for future research.

Acknowledgment. We wish to thank the reviewer for his/her very valuable comments and suggestions. D. Bunker and L. Han were supported in part by a Research and Creative Activities Grant from UM-Flint. S. Zhang was supported in part by the National Basic Research Program of China (2012CB955804) and the National Natural Science Foundation of China (10771158 and 11171251).

References

[1] *B. W. Bader, T. G. Kolda*: Tensor toolbox for Matlab, version 2.2. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/` (2008).

[2] *R. Bro*: PARAFAC. Tutorial and applications. Chemometrics and Intelligent Laboratory Systems *38* (1997), 149–171.

[3] *R. Bro*: Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications. Ph.D. thesis. University of Amsterdam, Amsterdam, 1998.

[4] *R. Bro*: Exploratory study of sugar production using fluorescence spectroscopy and multi-way analysis. Chemometrics and Intelligent Laboratory Systems *46* (1999), 133–147.

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

[6] *A. Cichocki, R. Zdunek, S. Amari*: Nonnegative matrix and tensor factorization. IEEE Signal Processing Magazine *25* (2008), 142–145.

[7] *A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, S. Amari*: Non-negative tensor factorization using alpha and beta divergences. Proc. of the 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, April 2007.

[8] *A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, S. Amari*: Novel multi-layer non-negative tensor factorization with sparsity constraints. Adaptive and Natural Computing Algorithms. Lecture Notes in Computer Science 4432. Proc. of the 8th International Conference on Adaptive and Natural Computing Algorithms, Warsaw, Poland, April 2007, Springer, Berlin.

[9] *A. Cichocki, R. Zdunek, A. H. Phan, S. Amari*: Nonnegative Matrix and Tensor Factorizations, Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, Chichester, 2009.

[10] *P. Comon, X. Luciani, A. L. F. de Almeida*: Tensor decompositions, alternating least squares and other tales. Journal of Chemometrics *23* (2009), 393–405.

[11] *I. S. Dhillon*: Fast Newton-type Methods for Nonnegative Matrix and Tensor Approximation. Talk given at the NSF Workshop, Future Directions in Tensor-Based Computation and Modeling, February 2009.

[12] *M. P. Friedlander, K. Hatz*: Computing non-negative tensor factorizations. Optim. Methods Softw. *23* (2008), 631–647.

[13] *L. Grippo, M. Sciandrone*: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Oper. Res. Lett. *26* (2000), 127–136.

[14] *L. Han, M. Neumann, U. Prasad*: Alternating projected Barzilai-Borwein methods for nonnegative matrix factorization. ETNA, Electron. Trans. Numer. Anal. *36* (2009–2010), 54–82, electronic only.

[15] *R. A. Harshman*: Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics *16* (1970), 1–84, `http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf`.

[16] *H. Kim, H. Park, L. Elden*: Non-negative tensor factorization based on alternating large-scale non-negativity-constrained least squares. Proceedings of IEEE 7th International Conference on Bioinformatics and Bioengineering (BIBE07), Vol. II. 2007, pp. 1147–1151.

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

[18] *D. D. Lee, H. S. Seung*: Learning the parts of objects by non-negative matrix factorization. Nature *401* (1999), 788–791.

[19] *L. Lim, P. Comon*: Nonnegative approximations of nonnegative tensors. Journal of Chemometrics *23* (2009), 432–441.

[20] *B. C. Mitchell, D. S. Burdick*: Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies. Journal of Chemometrics *8* (1994), 155–168.

[21] *M. Mørup, L. K. Hansen, S. M. Arnfred*: Algorithms for sparse nonnegative Tucker decompositions. Neural Comput. *20* (2008), 2112–2131.

[22] *D. Nion, L. De Lathauwer*: An enhanced line search scheme for complex-valued tensor decompositions. Application in DS-CDMA. Signal Process. *88* (2008), 749–755.

[23] *P. Paatero*: A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. Chemometrics and Intelligent Laboratory Systems *38* (1997), 223–242.

[24] *P. Paatero, U. Tapper*: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics *5* (1994), 111–126.

[25] *M. Rajih, P. Comon, R. A. Harshman*: Enhanced line search: a novel method to accelerate PARAFAC. SIAM J. Matrix Anal. Appl. *30* (2008), 1128–1147.

[26] *W. S. Rayens, B. C. Mitchell*: Two-factor degeneracies and a stabilization of PARAFAC. Chemometrics and Intelligent Laboratory Systems *38* (1997), 173–181.

[27] *R. T. Rockafellar*: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. *14* (1976), 877–898.

[28] *J.-P. Royer, N. Thirion-Moreau, P. Comon*: Computing the polyadic decomposition of nonnegative third order tensors. Signal Process. *91* (2011), 2159–2171.

[29] *A. Shashua, T. Hazan*: Non-negative tensor factorization with applications to statistics and computer vision. ICML 2005: Proceedings of the 22nd International Conference on Machine Learning. ACM, New York, 2005, pp. 792–799.

[30] *A. Smilde, R. Bro, P. Geladi*: Multi-way Analysis: Applications in the Chemical Sciences. Wiley, Chichester, 2004.

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

[32] MATLAB 7.5.0. The Mathworks (2008).

[33] Tensor Package. `http://www.i3s.unice.fr/~pcomon/TensorPackage.html`.

Authors' addresses: *Douglas Bunker*, *Lixing Han*, Department of Mathematics, University of Michigan-Flint, Flint, MI 48502, USA, e-mail: `lxhan@umflint.edu`; *Shuhua Zhang*, Research Center for Mathematics and Economics, Tianjin University of Finance and Economics, Tianjin 300222, China, e-mail: `szhang@tjufe.edu.cn`.