SWAMP REDUCING TECHNIQUE FOR TENSOR DECOMPOSITION

Carmeliza Navasca†, Lieven De Lathauwer‡§ and Stefan Kindermann¶

†Department of Mathematics, Clarkson University
PO Box 5815, Potsdam, NY 13699, USA
e-mail: cnavasca@gmail.com
‡Subfaculty Science and Technology, Katholieke Universiteit Leuven Campus Kortrijk,
E. Sabbelaan 53, 8500 Kortrijk, Belgium
e-mail: Lieven.DeLathauwer@kuleuven-kortrijk.be
§Electrical Engineering Department (ESAT-SCD), Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
e-mail: delathau@esat.kuleuven.be
¶Industrial Mathematics Institute, Johannes Kepler University Linz
Altenbergerstrasse 69, A-4040 Linz, Austria
e-mail: kindermann@indmath.uni-linz.ac.at

ABSTRACT

There are numerous applications of tensor analysis in signal processing, such as, blind multiuser separation-equalization-detection and blind identification. As the applicability of tensor analysis widens, the numerical techniques must improve to accommodate new data. We present a new numerical method for tensor analysis. The method is based on the iterated Tikhonov regularization and a parameter choice rule. Together these elements dramatically accelerate the well-known Alternating Least-Squares method.

1. INTRODUCTION

The purpose of this paper is to present a novel technique for tensor decomposition, namely the parallel factor (PARAFAC) analysis and block term decomposition (BTD). Tensor decomposition or multi-way analysis has been playing a major role in advancing signal processing and communications. For example, it has appeared in problems involving higher order statistics, with source independence, in blind multi-channel system identification, including the case where the number of sources exceeds the number of sensors, in source separation and in blind multiuser detection. In particular, Sidiropoulos et al. [28] have pioneered the application of tensor analysis to the blind multiuser separation-equalization-detection for direct-sequence code-division multiple access (DS-CDMA) systems where no space/time statistical independence is assumed. There have been numerous contributions thereafter; e.g. see [32] for PARAFAC and [25, 3] for BTD. Much of the success of applying tensor analysis is attributed to the Kruskal essential uniqueness property of PARAFAC [21] and BTD [11].

Moreover tensor decomposition has also been a vital tool in scientific computing [8, 16, 18], data analysis [2, 20], biomedical engineering [1, 14, 23] and chemometrics [29]. For more applications of tensor decomposition, see the survey paper of Kolda and Bader [19]. These tensor analysis techniques rely on numerical methods in optimization and numerical linear algebra. Current methods like the Alternating Least-Squares become inefficient and at times inadequate for these problems. Thus it is important to improve numerical methods for tensor analysis. In this paper, we introduce our regularized method for PARAFAC and BTD. We provide a new formulation for increasing the convergence rate of the well-known Alternating-Least-Squares method by applying the iterated Tikhonov regularization. Regularization methods have also appeared in the context of tensor decomposition; e.g. see [6], [7] for techniques in approximating nonnegative matrix. Our new formulation results in a new objective functional which requires a selection of the regularization parameter $\alpha$. We will describe our strategies for an easy and quick computation of the parameter $\alpha$. The implementation only requires only a few extra lines of coding so they are no more complicated than the original algorithms. This is another motivating factor for using regularization methods. They are extremely easy to implement unlike the purely algorithmic and complicated line search schemes [26, 24, 33] for accelerating the ALS algorithm.

1.1 Preliminaries

We denote the scalars in $\mathbb{R}$ and $\mathbb{C}$ with lower-case letters ($a, b, \ldots$) and the vectors with bold lower-case letters ($\mathbf{a}, \mathbf{b}, \ldots$). The matrices are written as bold upper-case letters ($\mathbf{A}, \mathbf{B}, \ldots$) and the symbol for tensors are calligraphic $\mathcal{T}$. The subscripts represent the following:

- $i_1 i_2 \ldots i_p$ is the $i_1 i_2 \ldots i_p$-th entry of a tensor $\mathbf{T}$.
- $\mathbf{A}_{i_1 i_2 \ldots i_p}$ is a submatrix of $\mathbf{T}$.

Moreover, $\mathbf{T}^\top$ is the transpose of a tensor $\mathbf{T}$.

Definition 1.1 The Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \mathbf{B} & a_{12} \mathbf{B} & \cdots \\ a_{21} \mathbf{B} & a_{22} \mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$  

Definition 1.2 The Khatri-Rao product of $\mathbf{A}$ and $\mathbf{B}$ is de-
fined in [27] as
\[ A \odot B = [A_1 \otimes B_1, A_2 \otimes B_2, \ldots]. \]
while the column-wise Khatri-Rao product is the special case,
\[ A \odot_c B = [a_1 \otimes b_1, a_2 \otimes b_2, \ldots] \]
when \( A = [a_1, a_2, \ldots, a_R] \) and \( B = [b_1, b_2, \ldots, b_R] \).

**Definition 1.3 (Mode-\(n\) vector)** Given a tensor \( \mathcal{T} \in \mathbb{C}^{I \times J \times K} \), there are three types of mode vectors, namely, mode-1, mode-2, and mode-3. There are \( J \cdot K \) mode-1 vectors that are of length \( I \) which are obtained by fixing the indices \( (j, k) \) while varying \( i \). Similarly, the mode-2 vector (mode-3 vector) is of length \( J \cdot (K \cdot R) \) obtained from the tensor by varying \( j \) (with fixed \( k, i \)) \( (i, j) \).

**Definition 1.4 (Mode-\(n\) rank)** The mode-\( n \) rank of a tensor \( \mathcal{T} \) is the dimension of the subspace spanned by the mode-\( n \) vectors.

The order of a tensor refers to the dimension of the index set. A matrix is a second-order tensor and a vector is a first-order tensor.

**Definition 1.5 (rank-(\(L,M,N)\))** A third-order tensor is rank-(\(L,M,N\)) if the mode-1 rank is \( L \), the mode-2 rank is \( M \) and the mode-3 rank is \( N \).

In the case when a third-order tensor has rank-(\(1,1,1\)), it is simply called a rank-1 tensor.

**Definition 1.6 (Tucker mode-\(n\) product)** Given a tensor \( \mathcal{T} \in \mathcal{C}^{I \times J \times K} \) and the matrices \( A \in \mathcal{C}^{I \times L}, B \in \mathcal{C}^{J \times M}, \) and \( C \in \mathcal{C}^{K \times N} \), then the Tucker mode-\( n \) products are the following:

\[ (\mathcal{T} \odot_1 A) = \sum_{i=1}^{L} t_i a_i \odot_1 A, \forall j, k, l \] (mode-1 product)
\[ (\mathcal{T} \odot_2 B) = \sum_{j=1}^{M} t_j b_j \odot_2 B, \forall i, k, l \] (mode-2 product)
\[ (\mathcal{T} \odot_3 C) = \sum_{k=1}^{N} t_k c_k \odot_3 C, \forall i, j, l \] (mode-3 product)

**2. TENSOR ANALYSIS**

The Parallel Factor Decomposition (PARAFAC) first appeared in [17] in the context of psychometrics. Independently, [9] introduced this decomposition as the Canonical Decomposition (CANDDECOMP) in phonetics. Given a tensor \( \mathcal{T} \in \mathcal{C}^{I \times J \times K} \), the PARAFAC decomposition is

\[ \mathcal{T} = \sum_{r=1}^{R} a_r \odot b_r \odot c_r \]

where \( a_r, b_r, c_r \) are the \( r \)-th columns of the matrices \( A \in \mathcal{C}^{I \times L}, B \in \mathcal{C}^{J \times M}, \) and \( C \in \mathcal{C}^{K \times N} \). The symbol ‘\( \odot \)’ denotes the usual outer product; i.e., if \( v, w \in \mathbb{R}^m \), then \( v \odot w = v \otimes w \). PARAFAC is a sum of rank-1 tensors. Another decomposition is the so-called Tucker decomposition [12, 30, 31]:

\[ \mathcal{T} = \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} (\mathcal{G})_{lmn} a_l \odot b_m \odot c_n \]

where \( (\mathcal{G})_{lmn} \) is a component of the tensor \( \mathcal{T} \in \mathbb{C}^{L \times M \times N} \) with full mode-1, mode-2 and mode-3 rank, \( a_l \) is \( l \)-th column of \( A \in \mathbb{C}^{I \times L}, b_m \) is the \( m \)-th column of \( B \in \mathbb{C}^{J \times M}, \) and \( c_n \) is the \( n \)-th column of \( C \in \mathbb{C}^{K \times N} \). The decomposition (2) is written more succinctly in terms of the Tucker mode-\( n \) product,

\[ \mathcal{T} = \mathcal{G} : A \odot_2 B \odot_3 C. \]

A normalized version of this formulation is referred to as the Higher-Order Singular Value Decomposition (HOSVD). In [12], the HOSVD has been shown as a generalization of the matrix SVD.

In the papers of [10, 11, 13], BTD is introduced. The following is a BTD in rank-(\(L_r, L_r, 1\)):

\[ \mathcal{T} = \sum_{r=1}^{R} E_r \odot c_r = \sum_{r=1}^{R} (A_r \cdot B_r^T) \odot c_r \]

where \( E_r \in \mathbb{C}^{I \times J} \), the rank \( \text{rank}(E_r) = L_r \) and \( E_r = A_r \cdot B_r^T \) and \( c_r \) is a column vector for \( r = 1, \ldots, R \). Note that \( A_r \in \mathbb{C}^{I \times L_r} \) with rank \( \text{rank}(A_r) = L_r \) and \( B_r \in \mathbb{C}^{J \times L_r} \) with rank \( \text{rank}(B_r) = L_r \). When \( L_r = L \) for all \( r = 1, \ldots, R \), it is called the BTD in rank-(\(L, L, 1\)). Furthermore, the BTD in rank-(\(L_r, L_r, 1\)) can be recast as

\[ \mathcal{T} = \sum_{r=1}^{R} D_r \odot_1 A_r \odot_2 B_r \odot_3 c_r \]

where \( D_r \in \mathbb{C}^{I \times J} \) is a full rank \( L_r \times L_r \) matrix. Lastly, the BTD in rank-(\(L, M, N\)) is

\[ \mathcal{T} = \sum_{r=1}^{R} \Gamma_r \odot_1 A_r \odot_2 B_r \odot_3 C_r \]

where \( \Gamma_r \in \mathbb{C}^{L \times M \times N} \) has full mode-1, mode-2 and mode-3 ranks, \( A_r \in \mathbb{C}^{I \times L}, B_r \in \mathbb{C}^{J \times M}, \) and \( C_r \in \mathbb{C}^{K \times N} \) for \( r = 1, \ldots, R \).

**2.1 Matrix And Vector Representation**

To apply the numerous well-developed tools in numerical linear algebra and optimization, matrix and vector representations of tensor are necessary. In the PARAFAC framework, the standard matricization of the tensors are in the directions of the left-right, front-back, and top-bottom to generate slices of matrices. Concatenating these sliced matrices allows us to build these long matrices:

\[ T^{I \times J} = (B \odot C)^T, \quad T^{K \times L} = (C \odot A)^T \]

and

\[ T^{L \times K} = (A \odot B)^C \]

where the superscripts of \( T \) reflect the matrix size. For the BTD in rank-(\(L_r, L_r, 1\)), the matricization results into these matrices

\[ T^{K \times I} = [B_1 \odot \ldots \odot B_R \odot C_1 \ldots C_1]^T. \]

\[ T^{K \times J} = [C_1 \odot \ldots \odot C_R \odot A_1 \ldots A_1]^T \]

and

\[ T^{L \times K} = [(A_1 \odot B_1)_{1_{L}} \ldots (A_1 \odot B_R)_{1_{L}}]^C \]

where \( 1_{L_r} \) is a column vector of 1’s of length \( L_r \).
3. ITERATED TIKHONOV REGULARIZATION

3.1 Alternating Least Squares

The problem we are solving is the following: we would like to recover the best tensor \( \hat{T} \in \mathbb{R}^{I \times J \times K} \) of rank-(\( L_r, L_r, 1 \)) or rank-(\( L_r, M_r, N_r \)) if it exists from the given a noisy tensor \( \tilde{T} \in \mathbb{R}^{I \times J \times K} \). Define the residual tensor as \( \mathcal{R} = \tilde{T} - \hat{T} \). The standard process is then to minimize the norm of the residual tensor in the least-square sense:

\[
\min \| A \|_F^2 = \min \| \tilde{T} - \hat{T} \|_F^2 \tag{10}
\]

where \( \| \cdot \|_F \) is the Frobenius norm. Here we discuss the recovery of the tensor decomposition in rank-(\( L_r, L_r, 1 \)), but the techniques are easily extended to the BTD in rank-(\( L_r, M_r, N_r \)). Recall the Frobenius norm of a tensor is defined as

\[
\| \mathcal{A} \|_F^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} |a_{ijk}|^2
\]

which is a direct extension of the Frobenius norm of a matrix. Here \( \mathcal{T} = \sum_{i=1}^{R} A_i B_i^T \circ c_r \) is tensor of rank-(\( L_r, L_r, 1 \)) where \( A_i \in \mathbb{R}^{L_r \times L_r}, B_i \in \mathbb{R}^{L_r \times L_r}, c_r \) is column vector of \( C \in \mathbb{R}^{r \times R} \). Then it follows that the optimization problem (10) is

\[
\min_{A,B} \| \mathcal{T} - \sum_{i=1}^{R} A_i B_i^T \circ c_r \|_F^2 \tag{11}
\]

where the minimization is over the partitioned matrices \( A \) and \( B \) and the matrix \( C \). Recall that \( A = [A_1 \ldots A_R], B = [B_1 \ldots B_R] \) and \( C = [c_1 \ldots c_R] \). Although there exists methods which directly approximate the factors \( A \), \( B \) and \( C \) through the least-squares formulation (10), it is more tractable and natural to approximate using these three sub-least squares problems:

\[
\begin{align*}
\min_{A} & \| \mathcal{T}^{JK \times I} - \mathcal{T}^{JK \times I} \|_F^2 = \min_{A} \| \mathcal{T}^{JK \times I} - QA^T \|_F^2 \\
\min_{B} & \| \mathcal{T}^{KI \times J} - \mathcal{T}^{KI \times J} \|_F^2 = \min_{B} \| \mathcal{T}^{KI \times J} - RB^T \|_F^2 \\
\min_{C} & \| \mathcal{T}^{IJ \times K} - \mathcal{T}^{IJ \times K} \|_F^2 = \min_{C} \| \mathcal{T}^{IJ \times K} - SC^T \|_F^2
\end{align*}
\tag{12}
\]

where \( \mathcal{T}^{JK \times I}, \mathcal{T}^{KI \times J}, \mathcal{T}^{IJ \times K} \) are the matricization of tensor \( \mathcal{T} \) through the top-bottom, left-right, front-back slices as in Section 2 and the matrices are

\[
Q = [B_1 \circ c_1 \ldots B_R \circ c_R], \quad R = [c_1 \circ A_1 \ldots c_R \circ A_R]
\]

and

\[
S = [(A_1 \circ B_1)L_1 \ldots (A_R \circ B_R)L_r].
\]

The three sub-least squares problems is implemented consecutively and iteratively. Each sub-least squares problem is updated immediately by the newly estimated matrix from the preceding sub-least squares problem much like the Gauss-Seidel scheme [5]. If instead

\[
Q = B \circ C, \quad R = C \circ A, \quad S = A \circ B
\]

and the noisy version of the matrices (6) are in (12), then this is a formulation for PARAFAC. These formulations are known as the Alternating Least-Squares (ALS) method which can be found in [29] for PARAFAC and [13] for BTD.

There are drawbacks of the ALS method. An artifact of the ALS algorithm is the so-called swamp. Swamp behavior occurs when there are exceedingly high number of iterations causing the convergence rate to slow down dramatically. Swamps have been observed when the component matrices are ill-conditioned or when collinearity occurs in the columns of these matrices.

3.2 Regularization Method

We reformulate the least squares sub-problems into three iterated Tikhonov sub-problems [15]:

\[
\begin{align*}
A_{i+1}^T &= \arg\min_A \| \tilde{T}^{JK \times I} - QA^T \|_F^2 + \alpha_i \| A^T - A_i^T \|_F^2 \tag{13} \\
B_{i+1}^T &= \arg\min_B \| \tilde{T}^{KI \times J} - RB^T \|_F^2 + \alpha_i \| B^T - B_i^T \|_F^2 \tag{14} \\
C_{i+1}^T &= \arg\min_C \| \tilde{T}^{IJ \times K} - SC^T \|_F^2 + \alpha_i \| C^T - C_i^T \|_F^2 \tag{15}
\end{align*}
\]

where \( \alpha_i \) is the regularization parameter and \( A_i, B_i \) and \( C_i \) are the approximation from the \( i \)th iteration. The regularization terms \( \alpha_i \| A - A_i \|, \alpha_i \| B - B_i \| \) and \( \alpha_i \| C - C_i \| \) are the fitting terms for the factors \( A, B \) and \( C \). It also forces the matrices \( A_i, B_i \) and \( C_i \) to have the same scalings and permutation of the previous iterates. Both PARAFAC and BTD are essentially unique. For PARAFAC, this means that the factors are unique up to scalings and permutation. For BTD, essential uniqueness includes conjugation. As a consequence, the objective functional is flat which causes a sluggish rate of convergence. Moreover, the regularization term allows a sub-optimal regularized solution to an otherwise ill-posed problem.

3.2.1 Parameter Choice

Iterated Tikhonov regularization can be understood in two ways: either \( \alpha \) is fixed and the iteration number is the main steering parameter or the iteration number is set \textit{a priori} and \( \alpha \) is tuned. Both strategies give a convergent regularization if the tuning parameter is based on the parameter choice rule. The parameter choice must be related to the noise level in the data according to a well-known result by Bakushinskii [4]. Otherwise a choice independent of the noise level cannot be expected to give convergent results in the worst case. In addition, the two schemes above can be combined by choosing a different regularization parameter at each iteration, e.g. a geometrically decaying sequence \( \alpha_i = q^i \) with \( 0 < q < 1 \). In fact, this is the strategy that we employed in the numerical examples in Section 4.

The parameter choice rule is based on Morozov’s discrepancy rule [22]: the \textit{a posteriori} choice of parameter \( \alpha \) requires no prior knowledge of the parameter but the knowledge of the noise level which is defined as the distance between given noisy tensor and unknown true tensor, i.e.

\[
\| \mathcal{T} - \mathcal{R} \|_F^2 \leq \tau \cdot \sigma
\tag{16}
\]

where \( \tau > 1 \) is a factor to ensure that the residual is attained and \( \sigma \) is the noise level. The noise level is typically unknown but it must be assumed \textit{a priori}. Thus, the discrepancy rule defines \( \alpha \) or the iteration number as the optimal parameter when the residual has the same order as of the noise level. If \( \alpha \) is fixed and the iteration number is the parameter to
be tuned, then the discrepancy rule dictates the termination of the loop. It stops at the first iteration index where the inequality (16) is satisfied. If the iteration number is set and the iterated Tikhonov regularization is implemented to obtain a sequence of decreasing parameters $\alpha_i$, then the discrepancy principle essentially terminates the procedure at the largest $\alpha_i$ for which (16) holds. Similarly for the mixed method of geometrically decaying $\alpha_i$, we stop the iteration when (16) is satisfied. Typically for linear problems, there exists such a parameter $\alpha_i$ or maximum number of iterations such that (16) holds and the algorithm usually terminates. Moreover, it has been shown in [15] that the regularization together with this parameter choice rule result in a convergent method; i.e. as the noise level $\sigma \to 0$, the regularized solution will tend to the true one.

Also there is the possibility of an a priori parameter choice rule. For instance, one can choose

$$\alpha^* \approx \tau \cdot \sigma.$$  \hspace{1cm} (17)

However, there are cases when this choice does not give the optimal order of convergence in contrast to the discrepancy rule. See the pseudocode in Table 1. Notice only a few additional lines of coding is required.

### 4. NUMERICAL RESULTS

We consider two models, PARAFAC and BTD, to demonstrate the swamp shortening property of the iterated Tikhonov regularization. We start with simple PARAFAC model [26] with

$$A = \begin{bmatrix} 1 & \cos \theta & 0 \\ 0 & \sin \theta & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & \sqrt{2} \cos \theta & 0 \\ 0 & \sin \theta & 1 \end{bmatrix},$$

and $C = I_{1 \times 3}$. Notice that as $\theta \to 0$, the columns of $A$ and $B$ become collinear. In Figure 1, the plot shows the error $\| \mathcal{F} - \hat{\mathcal{F}} \|^2_F$ versus the number of iterations it takes to obtain an error of $1 \times 10^{-5}$ for $\theta = \frac{\pi}{6}$. At $\theta = \frac{\pi}{3}$ the iterated Tikhonov method takes 41 iterations to reach an error within $10^{-5}$ while the least-squares method need 683 iterations. As $\theta$ is decreased both methods need more iterations. For the iterated Tikhonov regularization, 69 and 311 iterations are needed to attain an error of $1 \times 10^{-5}$ for $\theta = \frac{\pi}{5}$ and $\theta = \frac{\pi}{12}$, respectively. However, ALS requires 12007 ($\theta = \frac{\pi}{5}$) and 12026 ($\theta = \frac{\pi}{6}$) iterations for the error to be within $1 \times 10^{-3}$.

In the second example, we find the BTD of the tensor $\mathcal{F} \in \mathbb{R}^{4 \times 4 \times 6}$ of rank $(2,2,1)$ with the component matrices $A_i, B_i \in \mathbb{R}^{4 \times 2}$ for $i = 1, 2, 3$ and $C \in \mathbb{R}^{6 \times 3}$. Here $A = [A_1 A_2 A_3]$ where

$$A_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 \cos \theta \\ 1 \cos \theta \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 \\ 0 \sin \theta \\ 1 \end{bmatrix},$$

$B = A$ and $C$ is randomly generated. The bottom graph in Figure 1 displays 2500 iterations for our method versus 68630 ALS iterations for the residual to be within $1 \times 10^{-7}$. The swamp regime lengthens as the columns become more collinear in the ALS case. In general, more iterations are required for both methods when the dimensions $I,J,K$ are increased.

Acknowledgments

L. De Lathauwer was supported in part by (1) Research Council K.U.Leuven: GOA-Ambiotics, CoE EF/05/006 Optimization in Engineering (OPTEC), CIFI, (2) F.W.O.: (a) project G.0321.06, (b) Research Communities ICCoS, AN-
REFERENCES


