Rank Approximation of a Tensor with Applications in Color Image and Video Processing

Ramin Goudarzi* Carmeliza Navasca† Da Yan‡

April 30, 2019

Abstract

We propose a block coordinate descent type algorithm for estimating the rank of a given tensor. In addition, the algorithm provides the canonical polyadic decomposition of a tensor. In order to estimate the tensor rank we use sparse optimization method using \( \ell_1 \) norm. The algorithm is implemented on single moving object videos and color images for approximating the rank.

1 Introduction

In 1927, Hitchcock [17, 18] proposed the idea of the polyadic form of a tensor, i.e., expressing a tensor, multilinear array, as the sum of a finite number of rank-one tensors. This decomposition is called the canonical polyadic (CP) decomposition; it is known as CANDECOMP or PARAFAC. It has been extensively applied to many problems in various engineering [30, 32, 1, 13] and science [38, 22]. Specifically, tensor methods have been applied in many multidimensional datasets in signal processing applications [7, 9, 11], color image processing [33, 19] and video processing [33, 4]. Most of these applications rely on decomposing a tensor data into its low rank form to be able to perform efficient computing and to reduce memory requirements. In computer vision, detection of moving objects in video processing relies on foreground and background separation, i.e. the separation of the moving objects called foreground from the static information called background, requires low rank representation

*Department of Mathematics, University of Alabama at Birmingham, Birmingham, AL 35294, USA, ramin@uab.edu.
†Department of Mathematics, University of Alabama at Birmingham, 1300 University Boulevard, Birmingham, AL, 35294, USA, cnavasca@uab.edu
‡Department of Computer Science, University of Alabama at Birmingham, Birmingham, AL 35294, USA, yanda@uab.edu.
of video tensor. In color image processing, the rgb channels in color image representation requires extensions of the matrix models of gray-scale images to low rank tensor methods. There are several numerical techniques for approximating a low rank tensor into its CP decomposition, but they do not give an approximation of the minimum rank. In fact, most low rank tensor algorithms require an a priori tensor rank to find the tensor decomposition. Several theoretical results on tensor rank can help, but they are limited to low-multidimensional and low order tensors.

In this work, the focus is on finding an estimation of the tensor rank and its rank-one tensor decomposition (CP) of a given tensor. There are also algorithms which give tensor rank, but they are specific to symmetric tensor decomposition over the complex field using algebraic geometry tools. Our proposed algorithm addresses two difficult problems for the CP decomposition: (a) one is that finding the rank of tensors is a NP-hard problem and (b) the other is that tensors can be ill-posed and failed to have their best low-rank approximations.

The problem of finding the rank of a tensor can be formulated as a constrained optimization problem.

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad \mathcal{X} = \sum_{r=1}^{R} \alpha_r (a_r \odot b_r \odot c_r)
\]

where \(\|\alpha\|_0\) represents the total number of non-zero elements of \(\alpha\). The rank optimization problem is NP hard and so to make it more tractable, the following formulation is used:

\[
\min_{A,B,C,\alpha} \frac{1}{2} \|\mathcal{X} - \sum_{r=1}^{R} \alpha_r (a_r \odot b_r \odot c_r)\|_F^2 + \gamma \|\alpha\|_1
\]

where \(\gamma > 0\) is the regularization parameter and the objective function is a composition of smooth and non-smooth functions. Our formulation includes a Tikhonov type regularization:

\[
\min \frac{1}{2} \|\mathcal{X} - \sum_{r=1}^{R} \alpha_r (a_r \odot b_r \odot c_r)\|_F^2 + \lambda (\|A\|_F^2 + \|B\|_F^2 + \|C\|_F^2) + \gamma \|\alpha\|_1.
\]

The added Tikhonov regularization has the effect of forcing the factor matrices to have the equal norm. Moreover, this formulation and its numerical methods described later give an overall improvement in the accuracy and thus, memory requirements of the tensor model found in [42].

1.1 Organization

Our paper is organized as follows. In Section 2, we provide some notations and terminologies used throughout this paper. In Section 3, we formulate an \(l_1\)-regularization optimization to the low-rank approximation of tensors. In Section
4, we describe a numerical method to solve the $l_1$-regularization optimization by using a proximal alternating minimization technique for the rank and an alternating least-squares for the decomposition. In Section 5, we provide an analysis of convergence of the numerical methods. The numerical experiments in Section 6 consist of simulated low rank tensor, color images and videos. Finally, our conclusion and future work are given in Section 7.

2 Preliminaries

We denote the scalars in $\mathbb{R}$ with lower-case letters ($a, b, \ldots$) and the vectors with lower-case letters ($a, b, \ldots$). The matrices are written as upper-case letters ($A, B, \ldots$) and the symbols for tensors are calligraphic letters ($\mathcal{A}, \mathcal{B}, \ldots$). The subscripts represent the following scalars: $(\mathcal{A})_{ijk} = a_{ijk}$, $(A)_{ij} = a_{ij}$, $(a)_{i} = a_{i}$ and the $r$-th column of a matrix $A$ is $a_{r}$. The matrix sequence is denoted $\{A_{k}\}$.

An $N$th order tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a multidimensional array with entries $(X)_{i_1 i_2 \cdots i_N} = x_{i_1 i_2 \cdots i_N}$ for $i_k \in \{1, \ldots, I_k\}$ where $k \in \{1, \ldots, N\}$. In particular, a third order tensor $X \in \mathbb{R}^{I \times J \times K}$ is a multidimensional array with entries $x_{ijk}$ for $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ and $k \in \{1, \ldots, K\}$.

Here we present some standard definitions and relations in tensor analysis. The Kronecker product of two vectors $a \in \mathbb{R}^{I}$ and $b \in \mathbb{R}^{J}$ is denoted by $a \otimes b \in \mathbb{R}^{IJ}$:

\[ a \otimes b = (a_1 b^T \ldots a_I b^T)^T. \]

The Khatri-Rao (column-wise Kronecker) product (see [37]) of two matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{K \times J}$ is defined as

\[ A \odot B = (a_1 \otimes b_1 \ldots a_J \otimes b_J). \]

The outer product of three vectors $a \in \mathbb{R}^{I}$, $b \in \mathbb{R}^{J}$, $c \in \mathbb{R}^{K}$ is a third order tensor $X = a \circ b \circ c$ with the entries defined as follows:

\[ x_{ijk} = a_{i} b_{j} c_{k}. \]

**Definition 2.1 (vec)** Given a matrix $W \in \mathbb{R}^{I \times J}$, the function $\text{vec} : \mathbb{R}^{I \times J} \rightarrow \mathbb{R}^{I \cdot J}$ where $\text{vec}(W) = v$ is a vector of size $I \cdot J$ obtained from column-stacking the column vectors of $W$; i.e.

\[ \text{vec}(W)_l = v(l) = w_{ij} \]

where $l = j + (k - 1)J$.

The vectorization of a third order tensor $X \in \mathbb{R}^{I \times J \times K}$ is the process of transforming the tensor into a column vector, the $\text{vec} : \mathbb{R}^{I \times J \times K} \rightarrow \mathbb{R}^{IJK}$ map is defined as

\[ \text{vec}(X)_{\beta(i,j,k)} = x_{ijk} \]

where $\beta(i, j, k) = i + \{j - 1\}I + \{k - 1\}IJ$. Using the definitions above, we get

\[ \text{vec}(a \circ b \circ c) = c \otimes b \otimes a \]
Definition 2.2 (Mode-

Matricization is the process of reordering the elements of an \( N \)th order tensor into a matrix. The mode-

matricization of a tensor \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) is denoted by \( \mathcal{X}(n) \) and arranges the mode-

columns to be the columns of the resulting matrix. The mode-

column, \( x_{i_1 \cdots i_{n-1} i_n+1 \cdots i_N} \), is a vector obtained by fixing every index with the exception of the \( n \)th index.

If we use a map to express such matricization process for any \( N \)th order tensor \( \mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), that is, the tensor element \((i_1, i_2, \ldots, i_N)\) maps to matrix element \((i_n, j)\), then there is a formula to calculate \( j \):

\[
    j = 1 + \sum_{k=1 \atop k \neq n}^{N} (i_k - 1)J_k \quad \text{with} \quad J_k = \prod_{m=1 \atop m \neq n}^{k-1} I_m.
\]

For example, the tensor unfolding or matricization of a third order tensor \( \mathcal{X} \) is the process or rearranging the elements of \( \mathcal{X} \) into a matrix. The mode-

\( (n = 1, 2, 3) \) matricization is denoted by \( \mathcal{X}(n) \) and the elements of it can be expressed by the following relations:

\[
\mathcal{X}(1)(i,l) = x_{ijk}, \quad \text{where} \quad l = j + (k - 1)J_k \quad \text{and} \quad \mathcal{X}(1) \in \mathbb{R}^{I \times JK}
\]

\[
\mathcal{X}(2)(j,l) = x_{ijk}, \quad \text{where} \quad l = i + (k - 1)I_k \quad \text{and} \quad \mathcal{X}(2) \in \mathbb{R}^{J \times KI}
\]

\[
\mathcal{X}(3)(k,l) = x_{ijk}, \quad \text{where} \quad l = i + (j - 1)I_j \quad \text{and} \quad \mathcal{X}(3) \in \mathbb{R}^{K \times IJ}
\]

2.1 CP decomposition and the Alternating Least-Squares Method

In 1927, Hitchcock [17][18] proposed the idea of the polyadic form of a tensor, i.e., expressing a tensor as the sum of a finite number of rank-one tensors. Today, this decomposition is called the canonical polyadic (CP); it is known as CANDECOMP or PARAFAC. It has been extensively applied to many problems in various engineering [30, 32, 1, 13] and science [38, 22]. The well-known iterative method for implementing the sum of rank one terms is the Alternating Least-Squares (ALS) technique. Independently, the ALS was introduced by Carrol and Chang [6] and Harshman [15] in 1970. Among those numerical algorithms, the ALS method is the most popular one since it is robust. However, the ALS has some drawbacks. For example, the convergence of ALS can be extremely slow.

The CP decomposition of a given third order tensor \( \mathcal{X} \in \mathbb{R}^{I \times J \times K} \) factorizes it to a sum of rank one tensors:

\[
\mathcal{X} \approx \sum_{r=1}^{R} \alpha_r (a_r \circ b_r \circ c_r)
\]
For simplicity we use the notation \([A, B, C, \alpha]_R\) to represent the sum on the right hand side of the equation above, where \(A \in \mathbb{R}^{I \times R}\), \(B \in \mathbb{R}^{J \times R}\) and \(C \in \mathbb{R}^{K \times R}\) are called factor matrices.

\[A = [a_1 \ldots a_R], \quad B = [b_1 \ldots b_R], \quad C = [c_1 \ldots c_R]\]

The CP decomposition problem can be formulated as an optimization problem. Given \(R\) the goal is to find vectors \(a_r, b_r, c_r\), such that the distance between the tensor \(X\) and the sum of the outer products of \(a_r, b_r, c_r\) is minimized. The Frobenius norm (sum of squares of the entries) is mainly used to measure the distance.

\[
\min_{A,B,C,\alpha} \frac{1}{2} \|X - [A, B, C, \alpha]_R\|_F^2
\]  
(2.2)

Using the Khatri-Rao product, the objective function in (2.2) can be stated in the following four equivalent forms:

\[
\frac{1}{2} \|X(1) - A \text{diag}(\alpha)(C \odot B)^T\|_F^2,
\]  
(2.3)

\[
\frac{1}{2} \|X(2) - B \text{diag}(\alpha)(C \odot A)^T\|_F^2,
\]  
(2.4)

\[
\frac{1}{2} \|X(3) - C \text{diag}(\alpha)(B \odot A)^T\|_F^2,
\]  
(2.5)

and

\[
\frac{1}{2} \|\text{vec}(X) - \text{vec}([A, B, C, \alpha]_R)\|_2^2
\]  
(2.6)

All the functions in (2.3), (2.4), (2.5) and (2.6) are linear least squares problems with respect to matrices \(A, B, C\) and vector \(\alpha\). To find approximations to \(A, B, C\), and \(\alpha\), these four optimization problems (2.3)-(2.6) are implemented iteratively and the minimizers are updated between each optimization problems (via Gauss-Seidel sweep) with a stopping criteria. This technique is called the Alternating Least Squares (ALS) Method. The ALS method is popular since it is robust and easily implementable. However, the ALS has some drawbacks. For example, the convergence of ALS can be extremely slow. Another drawback is the requirement of a tensor rank \(R\) before a CP decomposition is approximated. The next sections deal with tensor rank approximation.

3 Rank Approximation of a Tensor

The problem of finding the rank of a tensor can be formulated as a constrained optimization problem.

\[
\min_{\alpha} \|\alpha\|_0 \quad \text{s.t.} \quad X = [A, B, C, \alpha]_R
\]
where $\|\alpha\|_0$ represents the total number of non-zero elements of $\alpha$. Since the problem is NP hard (ref), we replace $\|\alpha\|_0$ by the $\ell_1$ norm of $\alpha$. The $\ell_1$ norm is defined as the sum of absolute value of the elements of $\alpha$. So the rank approximation problem can be written as

$$\min_{\alpha} \|\alpha\|_1 \quad \text{s.t.} \quad X = [A, B, C, \alpha]_R$$

In order to obtain a CP decomposition of the given tensor $X$ as well as the rank approximation, we formulate the rank approximation problem as follow:

$$\min_{A,B,C,\alpha} \frac{1}{2} \|X - [A, B, C, \alpha]\|_F^2 + \gamma \|\alpha\|_1$$

(3.1)

where $\gamma > 0$ is the regularization parameter. The objective function of the problem (3.1) is non-convex and non-smooth. However, it is a composition of a smooth and non-smooth functions.

Moreover, it is known that CP decomposition of a tensor is unique up to scaling and permutation of factor matrices. Note that

$$[A, B, C, \alpha]_R = [cA, c^{-1}B, C, \alpha]_R$$

for a nonzero scalar $c \in \mathbb{R}$. In order to overcome the scaling indeterminacy, we add a Tikhonov type regularization term to our objective function \[20\]. Let $f$ and $g$ be the following:

$$f(A, B, C, \alpha) = \frac{1}{2} \|X - [A, B, C, \alpha]\|_F^2$$

(3.2)

and

$$g(\alpha) = \gamma \|\alpha\|_1$$

(3.3)

which represent the fitting term and the $\ell_1$ regularization term in (3.1), then the rank approximation problem can be formulated as

$$\min f(A, B, C, \alpha) + \frac{\lambda}{2} (\|A\|_F^2 + \|B\|_F^2 + \|C\|_F^2) + g(\alpha).$$

(3.4)

The added Tikhonov regularization has the effect of forcing the factor matrices to have the equal norm. i.e.

$$\|A\|_F = \|B\|_F = \|C\|_F.$$ 

\[29\] Now let $\Psi$ represent the objective function in (3.4) collectively, then

$$\Psi : \mathbb{R}^{I(I+J+K+1)} \rightarrow \mathbb{R}^+$$

where

$$\Psi(A, B, C, \alpha) = f(A, B, C, \alpha) + \frac{\lambda}{2} (\|A\|_F^2 + \|B\|_F^2$$

$$+ \|C\|_F^2) + g(\alpha)$$

(3.5)

Let $\omega = (A, B, C, \alpha)$, when $B, C, \alpha$ are fix, we represent $f(\omega)$ by $f(A)$ and $\Psi(\omega)$ by $\Psi(A)$. 



6
4 Approximation of Tensor Decomposition with Tensor Rank

In this section we propose a block coordinate descent type algorithm for solving the problem (3.4). We consider four blocks of variables with respect to \(A, B, C\) and \(\alpha\). In particular, at each inner iteration, we solve the following minimization problems

\[
A^{k+1} = \arg\min_A \{ f(A, B^k, C^k, \alpha^k) + \frac{\lambda}{2} \| A \|_F^2 \} \tag{4.1}
\]

\[
B^{k+1} = \arg\min_B \{ f(A^{k+1}, B, C^k, \alpha^k) + \frac{\lambda}{2} \| B \|_F^2 \} \tag{4.2}
\]

\[
C^{k+1} = \arg\min_A \{ f(A^{k+1}, B^{k+1}, C, \alpha^k) + \frac{\lambda}{2} \| C \|_F^2 \} \tag{4.3}
\]

and

\[
\alpha^{k+1} = \arg\min_{\alpha} \{ L^\beta_k f(A^{k+1}, B^{k+1}, C^{k+1}, \alpha) + g(\alpha) \} \tag{4.4}
\]

where \(L^\beta_k f(\alpha)\) represents the proximal linearization \([3]\) of \(f\) with respect to \(\alpha\), namely

\[
L^\beta_k f(\alpha) = (\alpha - \alpha^k, \nabla f(\alpha^k)) + \frac{1}{2\beta_k} \| \alpha - \alpha^k \|^2
\]

Note that each of the minimization problems in (4.1)-(4.4) is strictly convex, therefore \(A, B, C, \alpha\) are uniquely determined at each iteration. In fact, the subproblems in (4.1)-(4.3) are standard linear least squares problems with an additional Tikhonov regularization term. One can see by vectorization of the objective functions, for instance, the residual term in (2.3) can be written as follows

\[
\frac{1}{2} \| \text{vec}(X^{(1)}) - ((C \odot B) \text{diag}(\alpha^k)) \odot I) \text{vec}(A) \|_F^2
\]

Since the objective functions in (4.1)-(4.3) are strictly convex, the first order optimality condition is sufficient for a point to be minimum. In other words, the exact solutions of (4.1)-(4.3) can be given by the following normal equations

\[
A(E^k(E^k)^T + \lambda I) = X^{(1)}(E^k)^T,
\]

\[
B(F^k(F^k)^T + \lambda I) = X^{(2)}(F^k)^T,
\]

and

\[
C(G^k(G^k)^T + \lambda I) = X^{(3)}(G^k)^T
\]

where \(E^k = \text{diag}(\alpha^k)(C^k \odot B^k)^T\), \(F^k = \text{diag}(\alpha^k)(C^k \odot A^{k+1})^T\) and \(G^k = \text{diag}(\alpha^k)(B^{k+1} \odot A^{k+1})^T\).

To update \(\alpha\) in (4.4), we discuss the proximal operator first.
Definition 4.1 (proximal operator) Let \( g : \mathbb{R}^n \to \mathbb{R} \) be a lower semicontinuous convex function, then the proximal operator of \( g \) with parameter \( \beta > 0 \) is defined as follow
\[
\text{prox}_{\beta g}(y) = \arg \min_x \{ g(x) + \frac{1}{2\beta} \|x - y\|_2^2 \}. 
\] (4.5)

Using the proximal operator notation, the equation (4.4) is equivalent to
\[
\alpha^{k+1} = \text{prox}_{\beta g}(\alpha^k - \beta \nabla \alpha f(\alpha^k)).
\]
This is easy to verify because
\[
\begin{align*}
\alpha^{k+1} &= \text{prox}_{\beta g}(\alpha^k - \beta \nabla \alpha f(\alpha^k)) \\
&= \arg \min_{\alpha} \frac{1}{2\beta} \|\alpha - \alpha^k + \beta \nabla \alpha f(\alpha^k)\|_2^2 + g(\alpha) \\
&= \arg \min_{\alpha} \{ L_{f}^\beta + g(\alpha) \}.
\end{align*}
\]

Remark 4.2 The proximal operator in (4.5) is well-defined because the function \( g(\alpha) \) is continuous and convex. Using the vec operator, we have
\[
\begin{align*}
\text{vec}([A, B, C, \alpha]_R) &= \sum_{r=1}^R \alpha_r \text{vec}(a_r \circ b_r \circ c_r) \\
&= \sum_{r=1}^R \alpha_r (c_r \otimes b_r \otimes a_r) \\
&= M\alpha
\end{align*}
\] (4.6)

where \( M \in \mathbb{R}^{IJK \times R} \) is the matrix with columns \( c_r \otimes b_r \otimes a_r \). Therefore we can rewrite the objective function \( f \) as
\[
\frac{1}{2} \|\text{vec}(X) - M\alpha\|_2^2
\] (4.9)

It is easy to calculate the gradient of (4.9) with respect to \( \alpha \):
\[
\nabla \alpha f(A, B, C, \alpha) = M^T(M\alpha - \text{vec}(X))
\] (4.10)

This implies the Lipschitz continuity of the gradient of \( f \) with respect to \( \alpha \). The Lipschitz constant is \( Q_\alpha = \|M^T M\| \) so we must have
\[
\|\nabla \alpha f(\alpha_1) - \nabla \alpha f(\alpha_2)\| \leq Q_\alpha \|\alpha_1 - \alpha_2\|.
\] (4.11)

5 Analysis of Convergence

In this section, we study the global convergence of the proposed algorithm under mild assumptions. The Kurdyka-Lojasiewicz \([21, 26]\) property plays a key role in our analysis. We begin this section by stating the descent lemma.
Lemma 5.1 *(Descent Lemma)* Let $h : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable function, and $\nabla h$ is Lipschitz continuous with constant $L$, then for any $x, y \in \mathbb{R}^n$ we have

$$h(x) \leq h(y) + \langle x - y, \nabla h(y) \rangle + \frac{L}{2} \|x - y\|^2.$$ 

Next lemma provides the theoretical estimate for the decrease in the objective function after a single update $\alpha$.

Lemma 5.2 Suppose that $\alpha^{k+1}$ is obtained by the equation (4) and $0 < \beta^k < 1/Q^k$, where $Q^k$’s are defined in (4.11), then there is a constant $N^k > 0$ such that

$$\Psi(\alpha^k) - \Psi(\alpha^{k+1}) \geq N^k \|\alpha^{k+1} - \alpha^k\|^2. \quad (5.1)$$

Proof. Recall that

$$L^k_f(\alpha) = (\alpha - \alpha^k, \nabla f(\alpha)) + \frac{1}{2 \beta^k} \|\alpha - \alpha^k\|^2,$$

and $\alpha^{k+1}$ is obtained by the equation

$$\alpha^{k+1} = \text{arg min}_\alpha \{L^k_f(\alpha) + g(\alpha)\},$$

therefore we must have

$$L^k_f(\alpha^{k+1}) + g(\alpha^{k+1}) \leq g(\alpha^k). \quad (5.2)$$

Since $\nabla f$ is Lipschitz continuous with constant $Q^k$, by the descent lemma we have

$$f(\alpha^{k+1}) \leq f(\alpha^k) + \langle \alpha^{k+1} - \alpha^k, \nabla f(\alpha^k) \rangle$$

$$+ \frac{Q^k}{2} \|\alpha^{k+1} - \alpha^k\|^2$$

with (5.2), the above inequality implies

$$f(\alpha^{k+1}) + g(\alpha^{k+1}) \leq f(\alpha^k) + g(\alpha^k)$$

$$- \left(1 - \beta^k \frac{Q^k}{2 \beta^k}\right) \|\alpha^{k+1} - \alpha^k\|^2$$

setting

$$N^k = \frac{1 - \beta Q^k}{2 \beta} > 0$$

proves the lemma. \qed

Remark 5.3 Suppose that $Q^k$’s are bounded from above by the constant $Q_\alpha$ in the previous lemma, then for fixed step-size $\beta$ where

$$0 < \beta < 1/Q_\alpha$$

9
we have
\[ \Psi(\alpha^k) - \Psi(\alpha^{k+1}) \geq \left( \frac{1 - \beta Q_\alpha}{2 \beta} \right) \| \alpha^{k+1} - \alpha^k \|^2 \]  
(5.3)
for each \( k = 1, 2, \ldots \).

**Definition 5.4**\[14\] A differentiable function \( h : \mathbb{R}^n \to \mathbb{R} \) is called strongly convex if there is a constant \( \mu > 0 \) such that
\[ h(x) - h(y) \geq \langle \nabla h(y), x - y \rangle + \frac{\mu}{2} \| x - y \|^2 \]
for any \( x, y \in \mathbb{R}^n \).

**Lemma 5.5** Suppose that \( A^{k+1} \) is obtained by equation (4.1), then we have
\[ \Psi(A^k) - \Psi(A^{k+1}) \geq \frac{\lambda}{2} \| A^k - A^{k+1} \|^2_F \]

**Proof.** Note that the objective functions in (4.1) is strongly convex with parameter \( \lambda \) and by the first-order optimality condition we must have
\[ \nabla_A f(A^{k+1}) + \lambda A^{k+1} = 0 \]
now the strong convexity of \( f + \| . \|^2_F \) yields
\[ f(A^k) + \frac{\lambda}{2} \| A^k \|^2_F - f(A^{k+1}) - \frac{\lambda}{2} \| A^{k+1} \|^2_F \]
\[ \geq \frac{\lambda}{2} \| A^k - A^{k+1} \|^2 \]
which implies
\[ \Psi(A^k) - \Psi(A^{k+1}) \geq \frac{\lambda}{2} \| A^k - A^{k+1} \|^2. \]
This proves the lemma. \( \square \)

**Remark 5.6** Similar results hold for the blocks \( B \) and \( C \), if they are updated by equations (4.2) and (4.3). In particular, we have that

The next theorem guarantees that the value of \( \Psi \) decreases monotonically at each iteration. This shows that the sequence \( \{ \omega^k \} \) generated by scheme (4.1), (4.2), (4.3) and (4.4) is monotonically decreasing in value,

**Theorem 5.7** (Sufficient decrease property) Let \( \Psi \) represent the objective function in (ref) and \( \omega^k = (A^k, B^k, C^k, \alpha^k) \), then we have
\[ \Psi(\omega^k) - \Psi(\omega^{k+1}) \geq \rho \| \omega^k - \omega^{k+1} \|^2 \]
(5.4)
for some positive constant \( \rho \). In addition we have
\[ \sum_{k=0}^{\infty} \| \omega^k - \omega^{k+1} \|^2 < \infty \]  
(5.5)
Proof. By lemmas 5.2 and 5.5 we have
\[
\Psi(\omega^k) - \Psi(\omega^{k+1}) \geq \frac{\lambda}{2} (\|A^k - A^{k+1}\|^2 + \|B^k - B^{k+1}\|^2 \\
+ \|C^k - C^{k+1}\|^2) + N_\alpha \|\alpha^k - \alpha^{k+1}\|^2
\]
setting \( \rho = \min\{\lambda/2, N_\alpha\} \) gives the first result. This shows that the sequence \( \{\Psi(\omega^k)\} \) generated by our algorithm is decreasing. The monotonicity of \( \{\Psi(\omega^k)\} \) with the fact that \( \Psi \) is bounded from below, implies \( \Psi(\omega^k) \to \inf \Psi = \Psi \) as \( k \to \infty \), next let \( n > 2 \) be a positive integer, then
\[
\sum_{k=0}^{n-1} \|\omega^k - \omega^{k+1}\|^2 \leq \frac{1}{\rho} \sum_{k=0}^{n-1} (\Psi(\omega^k) - \Psi(\omega^{k+1}))
\]
\[
= \frac{1}{\rho} (\Psi(\omega^0) - \Psi(\omega^n))
\]
letting \( n \to \infty \) proves the last statement. \( \square \)

Remark 5.8 The sequence \( \{\omega^k\} \) generated by the scheme (4.2)-(4.4) is bounded. The reason comes from the fact that unboundedness of \( \{\omega^k\} \) occurs when at least one of the blocks \( A, B, C \) or \( \alpha \) gets unbounded. This never happens due to the regularization terms in the objective function \( \Psi \) and the fact that \( \Psi(\omega^k) \) is non-increasing.

Theorem 5.9 Let \( \{\omega^k\}_{k \in \mathbb{N}} \) be the sequence generated by our algorithm, then there exists a positive constant \( \rho > 0 \) such that for any \( k \in \mathbb{N} \) there is a vector \( \eta^{k+1} \in \partial \Psi(\omega^{k+1}) \) such that
\[
\|\eta^{k+1}\| \leq \rho \|\omega^k - \omega^{k+1}\|
\]
Proof. Let \( k \) be a positive integer. By equations (4.1), (4.2), (4.3) and the first order optimality condition we have
\[
\nabla_A f(A^{k+1}, B^k, C^k, \alpha^k) + \lambda A^{k+1} = 0,
\]
\[
\nabla_B f(A^{k+1}, B^{k+1}, C^k, \alpha^k) + \lambda B^{k+1} = 0,
\]
and
\[
\nabla_C f(A^{k+1}, B^{k+1}, C^{k+1}, \alpha^k) + \lambda C^{k+1} = 0
\]
define
\[
\eta_1^{k+1} = \nabla_A f(\omega^{k+1}) - \nabla_A f(A^{k+1}, B^k, C^k, \alpha^k)
\]
then \( \eta_1^{k+1} = \nabla_A \Psi(\omega^{k+1}) \). similarly we can define vectors \( \eta_2^{k+1}, \eta_3^{k+1} \). Next, by equation (4.4), we have that
\[
\alpha^{k+1} = \arg\min_{\alpha} \{L^g_f(A^{k+1}, B^{k+1}, C^{k+1}, \alpha) + g(\alpha)\} \quad (5.6)
\]
hence by the optimality condition, there exists \( u \in \partial g(\alpha^{k+1}) \) such that
\[
\nabla_\alpha f(\alpha^k) + \frac{1}{\beta^k} (\alpha^{k+1} - \alpha^k) + u^{k+1} = 0
\]
define
\[
\eta_4^{k+1} = \nabla_\alpha f(\omega^{k+1}) - \nabla_\alpha f(\alpha^k) + \frac{1}{\beta^k} (\alpha^k - \alpha^{k+1})
= \nabla_\alpha f(\omega^{k+1}) + u^{k+1}
\]
so \( \eta_4^{k+1} \in \partial \Psi_\alpha(\omega^{k+1}) \). From these facts we have that
\[
\eta_4^{k+1} = (\eta_1^{k+1}, \eta_2^{k+1}, \eta_3^{k+1}, \eta_4^{k+1}) \in \partial \Psi(\omega^{k+1})
\]
We now estimate the norm of \( \eta^{k+1} \). First note that by 5.8, \( \{\omega^k\} \) is bounded and the objective function (without the \( \ell_1 \) regularization term) is twice continuously differentiable, therefore as a consequence of mean value theorem, \( \nabla f \) must be Lipschitz continuous. Hence there must exist a constant \( P_1 \) such that
\[
\|\eta_1^{k+1}\| = \|\nabla_Af(\omega^{k+1}) - \nabla_Af(A^{k+1}, B^k, C^k, \alpha^k)\| \\
\leq P_1\|\omega^k - \omega^{k+1}\|
\]
similarly, constants \( P_2 \) and \( P_3 \) exist such that
\[
\|\eta_2^{k+1}\| \leq P_2\|\omega^k - \omega^{k+1}\|
\]
and
\[
\|\eta_3^{k+1}\| \leq P_3\|\omega^k - \omega^{k+1}\|
\]
setting \( \nu = \max\{P_1, P_2, P_3, P_4\} \), gives us the result. \( \square \)

Let \( f : \mathbb{R}^n \to \mathbb{R} \) be a continuous function. The function \( f \) is said to have Kurdyka-Łojasiewicz (KL) property at point \( \hat{x} \in \partial f \) if there exists \( \theta \in [0, 1) \) such that
\[
\frac{|f(x) - f(\hat{x})|}{\text{dist}(0, \partial f(x))} \leq \theta
\]
is bounded around \( \hat{x} \). A very rich class of functions satisfying the KL property is the semi-algebraic functions. These are functions where their graphs can be expressed as an algebraic set, that is
\[
\text{Graph}(f) = \bigcup_{i=1}^p \bigcap_{j=1}^q \{x \in \mathbb{R}^n : P_{ij} = 0, Q_{ij}(x) > 0\}
\]
where \( P_{ij} \)'s and \( Q_{ij} \)'s are polynomial functions and the graph of \( f \) is defined by
\[
\text{Graph}(f) = \{(x, y) \in \mathbb{R}^n \times \mathbb{R} : f(x) = y\}.
\]
Note that the univariate function \( g(x) = |x| \) is semialgebraic because

\[
\text{Graph}(g) = \{(x, y) : |x| = y\} = \{(x, y) : y - x = 0, x > 0\} \\
\cup \{(x, y) : y + x = 0, -x > 0\}
\]

The class of semi algebraic functions are closed under addition and composition \cite{2}. Hence The objective function in (3.5) is semialgebraic therefore it satisfies KL property.

Theorem 5.10 Suppose that \( \{\omega^k\}_{k \in \mathbb{N}} \) is the sequence generated by our algorithm, then \( \{\omega^k\}_{k \in \mathbb{N}} \) converges to the critical point of \( \Psi \).

Algorithm 1 The BCD algorithm to approximate rank

\begin{algorithm}
\textbf{Input:} A third order tensor \( \mathcal{X} \), an upper bound \( R \) of rank(\( \mathcal{X} \)), and the regularization parameters \( \lambda > 0, \gamma > 0 \) and the fixed stepsize \( \beta \);

\textbf{Output:} An approximated tensor \( \mathcal{Y} \) with an estimated rank \( \hat{R} \);

1. Given initial guess \( \mathcal{X}^0 = [A^0, B^0, C^0, \alpha^0]_R \).
2. \textbf{while} stopping criterion not met \textbf{do}
   3. Update \( A \):
      \[
      E = \text{diag}(\alpha)(C \odot B)^T
      \]
      \[
      A = (\mathcal{X}^{(1)})(E + \lambda I)
      \]
   4. Update \( B \):
      \[
      F = \text{diag}(\alpha)(C \odot A)^T
      \]
      \[
      B = (\mathcal{X}^{(2)})(F + \lambda I)
      \]
   5. Update \( C \):
      \[
      G = \text{diag}(\alpha)(B \odot A)^T
      \]
      \[
      C = (\mathcal{X}^{(3)})(G + \lambda I)
      \]
   6. Update \( \alpha \):
   7. for \( r = 1 \) to \( R \) do
      8. \( M(:,r) = \text{vec}(a_r \odot b_r \odot c_r) \)
   9. end for
   10. \( y = \alpha - \beta(M^T(M\alpha - \text{vec}(\mathcal{X}))) \)
   11. for \( r = 1 \) to \( R \) do
      12. \( \alpha(r) = \begin{cases} 
      y(r) - \beta & y(r) > \beta \\
      0 & |y(r)| \leq \beta \\
      y(r) + \beta & y(r) < -\beta
      \end{cases} \)
   13. end for
14. end while
15. Create tensor \( \mathcal{Y} \) with factor matrices \( A, B, C \) and coefficients \( \alpha \).
16. Count the number of non-zero elements of \( \alpha \) and assign it to \( \hat{R} \).
17. \textbf{return} The tensor \( \mathcal{Y} \) with the estimated rank \( \hat{R} \).
\end{algorithm}
| Size of Tensor | $I, J, K = 5$ | $I, J, K = 7$ | $I, J, K = 10$ |
|---------------|--------------|--------------|--------------|
| Actual Rank   | 5            | 8            | 10           |
| Upper bound   | 10           | 15           | 20           |
| Estimated Rank| 5            | 8            | 12           |
| Residual error| $2.85e-1$    | $1.34e-1$    | $1.20e-1$    |
| Relative error| $5.17e-2$    | $1.05e-2$    | $5.00e-3$    |
| Time          | 2.23         | 3.86         | 6.39         |

Table 1: Rank Approximation

6 Numerical Experiment and Results

In this section we test our algorithm on tensors with different rank and dimensions. We randomly generate tensors with specified ranks and compare the performance of our algorithm with other available algorithms such as LRAT [42]. Next, we apply our algorithm on single moving object videos in order to extract the background and target object.

6.1 Tensor Rank Approximation

In this subsection we test the performance of our algorithm on randomly generated cubic tensors with various dimensions and various rank. The upper bound for the rank of tensors are set to be equal to $\min\{IJ, JK, IK\}$. The results are shown in TABLE I.

6.2 Comparison between LRAT and our algorithm

In this subsection, we compare the performance of our proposed algorithm to LRAT [42]. We generate a random cubic tensor $A \in \mathbb{R}^{5 \times 5 \times 5}$ where its rank is equal to five. The comparison is based on the residual function as well as the sparsity of vector $\alpha$. The upper bound for the rank of the tested tensor is set to be equal to ten for both algorithms.

6.3 Application in background extraction of single moving object videos

In this subsection we apply our algorithm to extract the background of videos. See Figure 1. The video example [4, 33] is a $48 \times 48 \times 51$ with rank 23 tensor. The relative residual error of $\|X - \sum_{r} \alpha_r a_r \circ b_r \circ c_r\|_F^2$ is $10^{-8}$. 
Figure 1: The original video \cite{4,33} is of the size $48 \times 48 \times 51$. Column 1 shows the original (11th, 16th, 49th) frames, column 2 shows the reconstruction (background) and column 3 shows the foreground (moving objects).
Figure 2: Residual Plot. The x-axis is the number of iterations and y-axis is the relative error term of $\|X - \sum\alpha_r a_r \odot b_r \odot c_r\|_F^2$ for the video example in Figure 1.

Figure 3: The performance of our algorithm on RGB image. The right image illustrates the compressed reconstructed version of the original image.
7 Conclusion

We presented the iterative algorithm for approximating tensor rank and CP decomposition based on a sparse optimization problem. Specifically, we apply a Tikhonov regularization method for finding the decomposition and a proximal algorithm for the tensor rank. We have also provided convergence analysis and numerical experiments on color images and videos. Overall, this new tensor sparse model and its computational method dramatically improve the accuracy and memory requirements.

References

[1] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, *Multiway analysis of epilepsy tensors*, Bioinformatics, 23 (13), pp. i10-i18, 2007.

[2] H. Attouch, J. Bolte, B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. Math. Program. Ser. A 137 (2013), 91-129.

[3] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Math. Program., vol. 146, pp. 459-494, (2014).

[4] T. Bouwmans, A. Sobral, S. Javed and S.-K. Jung, and E.-H. Zahzah, Decomposition into Low-rank plus Additive Matrices for Background/Foreground Separation: A Review for a Comparative Evaluation with a Large-Scale Dataset, Computer Science Review, 23 (2017), Page: 1-71

[5] J. Brachat, P. Comon, B. Mourrain and E. Tsigaridas *Symmetric Tensor Decomposition*, Linear Algebra and Applications 433, 11-12 (2010), pp. 1851-1872.

[6] J. Carrol and J. Chang. *Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of “Eckart-Young” Decomposition*. Psychometrika, 9, 267-283, 1970.

[7] P. Comon, G. Golub, L-H. Lim and B. Mourrain. *Symmetric tensors and symmetric tensor rank*. SIAM Journal on Matrix Analysis and Applications, 30(3), 1254-1279, 2008.

[8] P. Comon, *Tensor decompositions*, in Mathematics in Signal Processing V. J. G. McWhirter and I. K. Proudler, eds., Clarendon Press, Oxford, UK, 2002, pp. 1-24.

[9] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.
[10] I. Domanov and L.D. Lathauwer, **Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition**, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 636-660.

[11] L. De Lathauwer, J. Castaing and J-F. Cardoso **Fourth-order cumulant-based blind identification of underdetermined mixtures.** IEEE Transactions of Signal Processing, 55(6), June 2007.

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

[13] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmieri, and W. Van Paesschen. **Canonical decomposition of ictal EEG reliably detects the seizure onset zone.** Neuroimage, 37(3), 844-854, 2007.

[14] J. Hiriart-Urruty, C. Lemarchal, Fundamentals of Convex Analysis, Springer, 2001.

[15] R. A. Harshman. **Foundations of the PARAFAC procedure: Model and Conditions for an “Explanatory” Multi-code Factor Analysis.** UCLA Working Papers in Phonetics, 16, 1-84, 1970.

[16] C.J. Hillar and L.-H. Lim, **Most tensor problems are NP-hard**, Journal of the ACM, 60 (2013), no. 6, Art. 45.

[17] F. L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. Vol. 6, pp 164-189, 1927.

[18] F. L. Hitchcock, Multiple invariants and generalized rank of a p-way matrix or tensor, J. Math. Phys. Vol. 7 pp 39-79, 1927.

[19] K. Jackowski and B. Cyganek, A learning-based colour image segmentation with extended and compact structural tensor feature representation, Pattern Analysis and Applications, 20 (2) (2017) pp 401414

[20] T. G. Kolda, B. W. Bader. **Tensor decompositions and applications.** SIAM review. Vol. 51, No. 3. pp.455-500 (2009).

[21] K. Kurdyka, On gradient of functions definable on o-minimal structures. Ann. Inst. Fourier 48, 769-783 (1998).

[22] P.M. Kroonenberg, **Applied Multiway Data Analysis.** Wiley, 2008.

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

[24] J. B. Kruskal, Three-way arrays: rank and uniqueness or trilinear decompositions, with application to arithmetic complexity and statistics, Linear algebra and its applications, 18 (2), 95-138, 1977.
[25] J.M. Landsberg, Tensors: Geometry and Applications, AMS, Providence, Rhode Island, 2010.

[26] S. Lojasiewicz, Sur la géométrie semi- et sous- analytique. Ann. Inst. Forier 43, 1575-1595 (1993).

[27] C. Navasca, L.D. Lathauwer and S. Kindermann, Swamp reducing technique for tensor decomposition, in the 16th Proceedings of the European Signal Processing Conference, 2008.

[28] P. Novati and M. R. Russo, A GCV based Arnoldi-Tikhonov regularization method, BIT Numerical Mathematics, 54 (2014), pp. 501-521.

[29] P. Paatero, A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis, Chemometrics and intelligent laboratory systems, Vol. 38 (2) pp. 223-242, (1997).

[30] N.D. Sidiropoulos, G.B. Giannakis, and R. Bro, *Blind PARAFAC receivers for DS-CDMA systems*, IEEE Trans. on Signal Processing, 48 (3), 810-823, 2000.

[31] B. Savas and L. Elden, Handwritten digit classification using higher order singular value decomposition, Pattern Recogn., 40 (2007), pp. 993-1003.

[32] N. Sidiropoulos, R. Bro, and G. Giannakis, Parallel factor analysis in sensor array processing, IEEE Trans. Signal Processing, 48, 2377-2388, 2000.

[33] A. Sobral, T. Bouwmans and E.-H. Zahzah, Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing, CRC Press, Taylor and Francis Group, 2015.

[34] G. Schwarz, Estimating the dimension of a model, The annals of statistics, 6 (1978), pp. 461-464.

[35] N. Sidiropoulos, R. Bro, and G. Giannakis, Parallel factor analysis in sensor array processing, IEEE Transactions on Signal Processing, 48 (2000), pp. 2377-2388.

[36] M. Sorensen and L. De Lathauwer, Blind Signal Separation via Tensor Decomposition With Vandermonde Factor: Canonical Polyadic Decomposition, IEEE Transactions on Signal Processing, 61 (2013), pp. 5507-5519.

[37] A.K Smilde, R. Bro, and P. Geladi, Multi-way analysis with applications in the chemical science, John Wiley & Sons, 2004.

[38] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*, Chichester, U.K., John Wiley and Sons, 2004.

[39] A. Uschmajew, Local convergence of the alternating least squares algorithm for canonical tensor approximation, SIAM J. Matrix Anal. Appl. 33(2):639-652, 2012.
[40] E. van den Berg and M.P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, SIAM J. Sci. Comput., 31 (2008), pp. 890-912.

[41] M.J. Wainwright, Sharp thresholds for high-dimensional and noisy sparsity recovery using constrained quadratic programming (Lasso), IEEE Transactions on Information Theory, 55(2009), pp. 2183-2202.

[42] X. Wang, C. Navasca. Lowrank approximation of tensors via sparse optimization, Numerical Linear Algebra with Applications, vol. 25, 2018.

[43] Y. Xu, L. Yu, H. Xu, H. Zhang, and T. Nguyen, Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis, IEEE TRANSACTIONS ON IMAGE PROCESSING, 24 (4) (2015), pp 1315-1328.

[44] Y. Xu and W. Yin, A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion, SIAM J. Imaging Sciences, 6 (2013), pp. 1758-1789.

[45] P. Zhao and B. Yu, On model selection consistency of Lasso, Journal of Machine Learning Research, 7 (2006), pp. 2541-2567.