Adaptive Low Rank Approximation for Tensors

Xiaofei Wang
Northeast Normal University
Renmin Street 5268, Changchun, China
wangxf341@nenu.edu.cn

Carmeliza Navasca
University of Alabama at Birmingham
1300 University Blvd, Birmingham, AL 35294
cnavasca@uab.edu

Abstract

In this paper, we propose a novel framework for finding low rank approximation of a given tensor. This framework is based on the adaptive Lasso with coefficient weights for sparse computation in tensor rank detection. We also provide an algorithm for solving the adaptive Lasso model problem for tensor approximation. In a special case when each weight equals to one, the convergence of the algorithm and the probabilistic consistency of the sparsity have been addressed [15]. The method is applied to background extraction and video compression problems.

1. Introduction

Computer vision problems often require processing and analyzing multidimensional data in face and object databases [6], surveillance videos [2, 16] and 3D/4D CT/fMRI images [19]. Tensors which are multiway arrays are natural representation of multidimensional data. Recent works [6, 11, 17] in computer vision use tensor based algorithms which decompose a tensor data into a sum of rank-one tensors. This tensor decomposition is referred to the canonical polyadic (CP) decomposition. Although several techniques [3, 9] can handle this decomposition, most of them need a priori tensor rank estimates, and a low rank approximation computation of tensor.

We consider a low rank approximation problem of tensors:

$$\min_{\mathcal{B}} \text{ rank}(\mathcal{B}) \quad \text{s.t. } \|\mathcal{A}-\mathcal{B}\|_F^2 \leq \varepsilon \tag{1}$$

for a given tensor $\mathcal{A}$ and a nonnegative regularization parameter ε. This approximation problem is actually a sparse recovery problem with an l_0-norm term. As in compressive sensing [5], the original l_0-minimization is replaced by an l_1-regularization problem. Advantages of this regularization for tensors are in detecting the rank of a given tensor due to sparsity and in mitigating the ill-posedness of the best low rank approximation of tensors since the l_1-regularization term provides a restriction on the boundedness of variables. The l_1-regularization problem formulated in [15] is

$$\min_{\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha}} \frac{1}{2}\|\mathcal{A}-[\boldsymbol{\alpha};\mathbf{X},\mathbf{Y},\mathbf{Z}]_R\|_F^2+\lambda\|\boldsymbol{\alpha}\|_1 \tag{2}$$

where $\mathbf{N}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = 1$. Here $\mathbf{N}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = 1$ is a normalization constraint and $\mathcal{B} = [\boldsymbol{\alpha};\mathbf{X},\mathbf{Y},\mathbf{Z}]_R = \sum_{r=1}^{R}\alpha_r\mathbf{x}\circ\mathbf{y}\circ\mathbf{z}$ is the R summands of rank one tensors. The symbol $\circ$ denotes the outer product. The matrices $\mathbf{X},\mathbf{Y},\mathbf{Z}$ are concatenation of the vectors $\mathbf{x}_r,\mathbf{y}_r,\mathbf{z}_r$ where $r=1,\cdots,R$, respectively. This formulation led to practical computation of low rank tensor decomposition.

In this paper, we propose a more general optimization framework in the model by using an adaptive method (known as adaptive Lasso [20]). The new formulation is the following

$$\min_{\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha}} \frac{1}{2}\|\mathcal{A}-[\boldsymbol{\alpha};\mathbf{X},\mathbf{Y},\mathbf{Z}]_R\|_F^2+\lambda\sum_{r=1}^{R}\omega_r|\alpha_r| \tag{3}$$

where $\boldsymbol{\omega} = (\omega_1,\cdots,\omega_R)^T$ is a vector of known positive weights and $\mathbf{N}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = 1$. We call this optimization form (3) as the adaptive low rank approximation of tensor. Observe that the model (2) is recovered from the adaptive low rank approximation of tensor if $\omega_r = 1$ for all r.

1.1. Adaptive Lasso

The Lasso problem [14] is the ℓ_1 regularization of the least-square method (ℓ_1 penalized linear regression). Given a vector $y\in\mathbb{R}^n$, a matrix $\boldsymbol{X}\in\mathbb{R}^{n\times p}$ with a tuning parameter $\lambda\geq 0$, the Lasso estimate can be defined as

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1. \tag{4}$$

The solution to the Lasso problem is unique when $\boldsymbol{X}$ is full rank. Otherwise, (4) can have multiple solutions when $\boldsymbol{X}$ is rank deficient. Due to the nature of ℓ_1 penalty and on the value of the tuning parameter λ, the solutions to the Lasso problem have many coefficients set exactly to zero.

Now in many studies [8, 12], it has been confirmed that the Lasso does not possess oracle property [4]. The oracle property refers to the ability to correctly select the nonzero coefficients with probability converging to one, and that the estimators of the nonzero coefficients are asymptotically normal with the same means and covariance as they would have if the zero coefficients were known a priori. Zou [20] argued that it is unreasonable to force the coefficients to be equally penalized, and introduced a weighted ℓ_1 penalty with weights determined by an initial estimator; i.e.

$$\hat{\boldsymbol{\beta}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} w_j|\beta_j| \tag{5}$$

This is called the adaptive Lasso. If the weights ω_j are data-dependent and cleverly chosen, the adaptive Lasso has the oracle properties as shown in the following proposition [20]:

Proposition 1 *(Oracle properties) Suppose that* $\frac{\lambda_n}{\sqrt{n}} \to 0$ *and* $\lambda_n n^{\frac{\gamma-1}{2}} \to \infty$ *where* n *is the sample size. Then the adaptive Lasso estimates must satisfy the following:*

1. *Consistency in variable selection:* $\lim_n P(S_n^* = S) = 1$
2. *Asymptotic normality:* $\sqrt{n}(\hat{\boldsymbol{\beta}}_S^{*(n)} - \boldsymbol{\beta}_S^*) \to_d N(0, \sigma^2 \times C_{SS}^{-1})$

where $S = \{j : \beta_j^* \neq 0\}$ *and* C_{SS} *is the corresponding submatrix of* $C = \frac{1}{n}X^TX$.

Note that (5) is a convex optimization problem in which its global minima can be efficiently solved. Current efficient algorithms can be used to compute adaptive Lasso estimates.

1.2. Preliminaries

A bold lower-case letter $\mathbf{a}$ is denoted by a vector. The bold upper-case letter $\mathbf{A}$ represents a matrix and the symbol of tensor is a calligraphic letter $\mathcal{A}$. Tensors with three indices are third order tensors $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{I\times J\times K}$ with $1 \le i \le I, 1 \le j \le J$ and $1 \le k \le K$. For the clarity of the exposition, the discussion is limited to third order tensors, but all the methods proposed here can be applied to tensors of arbitrary high order.

A third-order tensor $\mathcal{A}$ has column, row and tube fibers, which are defined by fixing every index but one and denoted by $\mathbf{a}_{:jk}$, $\mathbf{a}_{i:k}$ and $\mathbf{a}_{ij:}$ respectively. Correspondingly, we can obtain three kinds $\mathbf{A}_{(1)}, \mathbf{A}_{(2)}$ and $\mathbf{A}_{(3)}$ of matricization of $\mathcal{A}$ according to respectively arranging the column, row, and tube fibers to be columns of matrices. We can also consider the vectorization for $\mathcal{A}$ to obtain a row vector $\mathbf{a}$ such the elements of $\mathcal{A}$ are arranged according to k varying faster than j and j varying faster than i, i.e., $\mathbf{a} = (a_{111}, \cdots, a_{11K}, a_{121}, \cdots, a_{12K}, \cdots, a_{1J1}, \cdots, a_{1JK}, \cdots)$.

The outer product $\mathbf{x}\circ\mathbf{y}\circ\mathbf{z} \in \mathbb{R}^{I\times J\times K}$ of three nonzero vectors $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ is called a rank-one tensor with elements $x_iy_jz_k$ for all the indices. A canonical polyadic (CP) decomposition of $\mathcal{A} \in \mathbb{R}^{I\times J\times K}$ expresses $\mathcal{A}$ as a sum of rank-one outer products:

$$\mathcal{A} = \sum_{r=1}^{R} \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r \tag{6}$$

where $\mathbf{x}_r \in \mathbb{R}^I, \mathbf{y}_r \in \mathbb{R}^J, \mathbf{z}_r \in \mathbb{R}^K$ for $1 \le r \le R$. Every outer product $\mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$ is called as a rank-one component and the number R is called as the rank-one component number of tensor $\mathcal{A}$. The minimal rank-one component number R such that the decomposition (6) holds is called the rank of tensor $\mathcal{A}$, and is denoted by rank($\mathcal{A}$). For any tensor $\mathcal{A} \in \mathbb{R}^{I\times J\times K}$, rank($\mathcal{A}$) has an upper bound $\min\{IJ, JK, IK\}$.

The CP decomposition (6) can be also written as:

$$\mathcal{A} = \sum_{r=1}^{R} \alpha_r \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r \tag{7}$$

where $\alpha_r \in \mathbb{R}$ is a rescaling coefficient of rank-one tensor $\mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$ for $r = 1, \cdots, R$. For convenience, we denote the row vector $(\alpha_1, \cdots, \alpha_R) \in \mathbb{R}^R$ as $\boldsymbol{\alpha}$, and rewrite the sum $\sum_{r=1}^{R} \alpha_r \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$ of (7) into $[\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R$, where $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_R) \in \mathbb{R}^{I\times R}$, $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_R) \in \mathbb{R}^{J\times R}$ and $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_R) \in \mathbb{R}^{K\times R}$ are called the factor matrices of tensor $\mathcal{A}$. It is often useful to add a constraint on the columns of factor matrices normalized to length one. We denote this constraint by $\mathbf{N}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = 1$, and call it as normalization constraint.

The Khatri-Rao product of two matrices $\mathbf{X} \in \mathbb{R}^{I\times R}$ and $\mathbf{Y} \in \mathbb{R}^{J\times R}$ is defined as

$$\mathbf{X} \odot \mathbf{Y} = (\mathbf{x}_1 \otimes \mathbf{y}_1, \cdots, \mathbf{x}_R \otimes \mathbf{y}_R) \in \mathbb{R}^{IJ\times R},$$

where the symbol "$\otimes$" denotes the Kronecker product:

$$\mathbf{x} \otimes \mathbf{y} = (x_1y_1, \cdots, x_1y_J, \cdots, x_Iy_1, \cdots, x_Iy_J)^T.$$

Using this Khatri-Rao product, the decomposition (7) can be written in three matricization forms of tensor $\mathcal{A}$:

$$\mathbf{A}_{(1)} = \mathbf{X}\mathbf{D}(\mathbf{Z} \odot \mathbf{Y})^T, \mathbf{A}_{(2)} = \mathbf{Y}\mathbf{D}(\mathbf{Z} \odot \mathbf{X})^T, \tag{8}$$

$$\mathbf{A}_{(3)} = \mathbf{Z}\mathbf{D}(\mathbf{Y} \odot \mathbf{X})^T$$

where the matrix $\mathbf{D}$ is diagonal with elements of $\boldsymbol{\alpha}$.

2. Adaptive low rank approximation of tensor

We use an optimization framework to find a low rank tensor which can be calculated efficiently from an original given tensor. For any given error ε, the minimal rank of $\mathcal{B}$ such that $\|\mathcal{A}-\mathcal{B}\|_F^2 \le \varepsilon$ is no larger than rank$(\mathcal{A})$. The optimal solution $\hat{\mathcal{B}}$ is a low rank approximation of $\mathcal{A}$ with error ε.

We represent the tensor $\mathcal{B}$ as $\sum_{r=1}^{R} \alpha_r \mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r = [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R$ where R is an upper bound of the rank of $\mathcal{A}$ and columns of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ satisfy the normalization constraint $\mathbf{N}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = 1$. And the problem (1) is equivalent to the following constraint optimization problem with l_0-norm:

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{s.t. } \|\mathcal{A} - [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R\|_F^2 \le \varepsilon, \qquad (9)$$

where $\mathbf{N}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = 1$.

The problem (9) is equivalent to that of finding the rank of tensors when $\varepsilon = 0$, whose decision version is NP-hard [7]. Inspired by the theory of compressive sensing [5], we turn to the following optimization problem with l_1-norm to avoid the intractibility:

$$\min_{\mathbf{X},\mathbf{Y},\mathbf{Z},\boldsymbol{\alpha}} \frac{1}{2}\|\mathcal{A} - [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R\|_F^2 + \lambda \sum_{r=1}^{R} \omega_r |\alpha_r| \qquad (10)$$

where $\mathbf{N}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = 1, \lambda > 0$ and $\boldsymbol{\omega} = (\omega_1, \cdots, \omega_R)^T$ is a known positive weights vector. In this work, our algorithm is tailored for solving the problem (10).

We denote the objective function in (10) as

$$\Psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) : \mathbb{R}^{I\times R} \times \mathbb{R}^{J\times R} \times \mathbb{R}^{K\times R} \times \mathbb{R}^{R} \to \mathbb{R}^{+},$$

the approximation term $\frac{1}{2}\|\mathcal{A} - [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R\|_F^2$ by

$$f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) : \mathbb{R}^{I\times R} \times \mathbb{R}^{J\times R} \times \mathbb{R}^{K\times R} \times \mathbb{R}^{R} \to \mathbb{R}^{+},$$

and the regularized penalty term $\lambda \sum_{r=1}^{R} \omega_r |\alpha_r|$ as $g(\boldsymbol{\alpha})$. So $\Psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) = f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) + g(\boldsymbol{\alpha})$. There are four blocks $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}$ of variables in the function $\Psi(\bullet)$, and it is convex on one block for any fixed three blocks.

It is well known that the problem of finding a best rank-R approximation for tensors of order 3 or higher has no solution in general, due to the ill-posedness [13, 10] of the best low rank approximation of tensors. However, after introducing the l_1 penalty term $g(\boldsymbol{\alpha})$ to the low rank approximation term $f(\bullet)$, it is always attainable for the minimization of the objective function in (10). The following theorem shows the existence of the global optimal solution of problem (10).

Theorem 1 *The global optimal solution of problem (10) exists.*

2.1. Algorithm for tensor approximation

Here we describe an algorithm (ALRAT) of adaptive low rank approximation of tensor for computing the solution of problem (10). The idea of this ALRAT algorithm comes from the proximal alternating linearized minimization technique [1]. The objective function $\Psi(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ in (10) consists of two parts $f(\bullet)$ and $g(\bullet)$, where the approximation term $f(\bullet) = \frac{1}{2}\|\mathcal{A} - [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R\|_F^2$ and the regularized penalty term $g(\bullet) = \lambda \sum_{r=1}^{R} \omega_r |\alpha_r|$. The function $f(\bullet)$ is a real polynomial function on $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) \in \mathbb{R}^{I\times R} \times \mathbb{R}^{J\times R} \times \mathbb{R}^{K\times R} \times \mathbb{R}^{R}$ and the function $g(\bullet)$ is a non-differential continuous function on $\boldsymbol{\alpha}$.

Algorithm 1 Adaptive Low Rank Approximation of Tensor (ALRAT)

Input: A third order tensor $\mathcal{A}$, an upper bound R of rank$(\mathcal{A})$, a penalty parameter λ, a nonnegative weight vector $\boldsymbol{\omega}$ and a scale $s > 1$;

Output: A tensor $\hat{\mathcal{B}}$ and a estimated rank $\hat{R}$;

1: Give an initial tensor $\mathcal{B}^0 = [\boldsymbol{\alpha}^0; \mathbf{X}^0, \mathbf{Y}^0, \mathbf{Z}^0]_R$.

2: Update step:

a. Update matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$:

Compute $\mathbf{U}^{k+1}$ from $\boldsymbol{\alpha}^k, \mathbf{Y}^k, \mathbf{Z}^k$ by (11).

$\mathbf{X}_*^{k+1} = \mathbf{X}^k - \frac{1}{sc_k}(\mathbf{X}^k\mathbf{U}^{k+1} - \mathbf{A}_{(1)})\mathbf{U}^{k+1^T}$, where $c_k = \|\mathbf{U}^{k+1}\mathbf{U}^{k+1^T}\|_F$.

Compute $\mathbf{V}^{k+1}$ from $\boldsymbol{\alpha}^k, \mathbf{X}_*^{k+1}, \mathbf{Z}^k$ by (11).

$\mathbf{Y}_*^{k+1} = \mathbf{Y}^k - \frac{1}{sd_k}(\mathbf{Y}^k\mathbf{V}^{k+1} - \mathbf{A}_{(2)})\mathbf{V}^{k+1^T}$, where $d_k = \|\mathbf{V}^{k+1}\mathbf{V}^{k+1^T}\|_F$.

Compute $\mathbf{W}^{k+1}$ from $\boldsymbol{\alpha}^k, \mathbf{X}_*^{k+1}, \mathbf{Y}_*^{k+1}$ by (11).

$\mathbf{Z}_*^{k+1} = \mathbf{Z}^k - \frac{1}{se_k}(\mathbf{Z}^k\mathbf{W}^{k+1} - \mathbf{A}_{(2)})\mathbf{W}^{k+1^T}$, where $e_k = \|\mathbf{W}^{k+1}\mathbf{W}^{k+1^T}\|_F$.

Normalize every column in $\mathbf{X}_*^{k+1}, \mathbf{Y}_*^{k+1}$ and $\mathbf{Z}_*^{k+1}$ to one, and obtain updated matrices $\mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}$.

b. Update the row vector $\boldsymbol{\alpha}$:

Compute $\mathbf{Q}^{k+1}$ from $\mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}$ by (12).

$\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k - \frac{1}{s\eta_k}(\boldsymbol{\alpha}^k\mathbf{Q}^{k+1} - \mathbf{a})\mathbf{Q}^{k+1^T}$, where $\eta_k = \|\mathbf{Q}^{k+1}\mathbf{Q}^{k+1^T}\|_F$.

For all the indices i of $\boldsymbol{\alpha}^{k+1}$, use the soft shrinkage:

$$\alpha_i^{k+1} = \begin{cases} \alpha_i^{k+1} - \lambda\omega_i, & \text{if } \alpha_i^{k+1} > \lambda\omega_i \\ 0, & \text{if } -\lambda\omega_i \le \alpha_i^{k+1} \le \lambda\omega_i \\ \alpha_i^{k+1} + \lambda\omega_i, & \text{if } \alpha_i^{k+1} < -\lambda\omega_i \end{cases}$$

3: Denote the limitations by $\hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}, \hat{\boldsymbol{\alpha}}$, compute $\hat{\mathcal{B}} = [\hat{\boldsymbol{\alpha}}; \hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}]_R$ and count the number $\hat{R}$ of nonzero entries in $\hat{\boldsymbol{\alpha}}$.

4: **return** The tensor $\hat{\mathcal{B}}$ and the estimated rank $\hat{R}$.

Three kinds of matricization forms of tensor $\mathcal{B} = [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R$ can be written by using Khatri-Rao products as

$$\mathbf{B}_{(1)} = \mathbf{X}\mathbf{D}(\mathbf{Z} \odot \mathbf{Y})^T, \mathbf{B}_{(2)} = \mathbf{Y}\mathbf{D}(\mathbf{Z} \odot \mathbf{X})^T,$$

$$\mathbf{B}_{(3)} = \mathbf{Z}\mathbf{D}(\mathbf{Y} \odot \mathbf{X})^T$$

where $\mathbf{D} = diag(\alpha_1, \cdots, \alpha_R)$. We introduce three matrices for updating computation in Algorithm 1:

$$\mathbf{U} = \mathbf{D}(\mathbf{Z} \odot \mathbf{Y})^T, \mathbf{V} = \mathbf{D}(\mathbf{Z} \odot \mathbf{X})^T, \tag{11}$$

$$\mathbf{W} = \mathbf{D}(\mathbf{Y} \odot \mathbf{X})^T.$$

Thus $\mathbf{B}_{(1)} = \mathbf{X}\mathbf{U}$, $\mathbf{B}_{(2)} = \mathbf{Y}\mathbf{V}$ and $\mathbf{B}_{(3)} = \mathbf{Z}\mathbf{W}$. So the function $f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ can be written as three forms $\frac{1}{2}\|\mathbf{A}_{(1)} - \mathbf{X}\mathbf{U}\|_F^2 = \frac{1}{2}\|\mathbf{A}_{(2)} - \mathbf{Y}\mathbf{V}\|_F^2 = \frac{1}{2}\|\mathbf{A}_{(3)} - \mathbf{Z}\mathbf{W}\|_F^2$.

Using the vectorization of tensors, we can vectorize every rank-one tensor of outer product $\mathbf{x}_r \circ \mathbf{y}_r \circ \mathbf{z}_r$ into a row vector $\mathbf{q}_r$ for $1 \le r \le R$, and denote a matrix consisting of all $\mathbf{q}_r$ for $1 \le r \le R$ by

$$\mathbf{Q} = (\mathbf{q}_1^T, \cdots, \mathbf{q}_R^T)^T. \tag{12}$$

Thus the function $f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ can be also written as $\frac{1}{2}\|\mathbf{a} - \boldsymbol{\alpha}\boldsymbol{Q}\|_F^2$, where $\mathbf{a}$ is a vectorization for tensor $\mathcal{A}$.

The algorithm starts from $(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k, \boldsymbol{\alpha}^k)$ and iteratively update variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and then $\boldsymbol{\alpha}$ in each loop. The update of $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ is based on the following optimization problems:

$$\begin{aligned}
\mathbf{X}_*^{k+1} &= \arg\min_{\mathbf{X}} \{\langle \mathbf{X} - \mathbf{X}^k, \nabla_{\mathbf{X}} f(\mathbf{X}^k, \mathbf{Y}^k, \mathbf{Z}^k, \boldsymbol{\alpha}^k)\rangle \\
&\qquad + \frac{sc_k}{2}\|\mathbf{X} - \mathbf{X}^k\|_F^2\} \\
\mathbf{Y}_*^{k+1} &= \arg\min_{\mathbf{Y}} \{\langle \mathbf{Y} - \mathbf{Y}^k, \nabla_{\mathbf{Y}} f(\mathbf{X}_*^{k+1}, \mathbf{Y}^k, \mathbf{Z}^k, \boldsymbol{\alpha}^k)\rangle \\
&\qquad + \frac{sd_k}{2}\|\mathbf{Y} - \mathbf{Y}^k\|_F^2\} \\
\mathbf{Z}_*^{k+1} &= \arg\min_{\mathbf{Z}} \{\langle \mathbf{Z} - \mathbf{Z}^k, \nabla_{\mathbf{Z}} f(\mathbf{X}_*^{k+1}, \mathbf{Y}_*^{k+1}, \mathbf{Z}^k, \boldsymbol{\alpha}^k)\rangle \\
&\qquad + \frac{se_k}{2}\|\mathbf{Z} - \mathbf{Z}^k\|_F^2\}
\end{aligned} \tag{13}$$

Notice that the penalty term $g(\bullet)$ vanishes in equations (13) since it is a function only relying on $\boldsymbol{\alpha}$.

Since the function $f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ can be written as three forms $\frac{1}{2}\|\mathbf{A}_{(1)} - \mathbf{X}\mathbf{U}\|_F^2 = \frac{1}{2}\|\mathbf{A}_{(2)} - \mathbf{Y}\mathbf{V}\|_F^2 = \frac{1}{2}\|\mathbf{A}_{(3)} - \mathbf{Z}\mathbf{W}\|_F^2$, we have the following gradient equations:

$$\begin{aligned}
\nabla_{\mathbf{X}} f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) &= (\mathbf{X}\mathbf{U} - \mathbf{A}_{(1)})\mathbf{U}^T, \\
\nabla_{\mathbf{Y}} f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) &= (\mathbf{Y}\mathbf{V} - \mathbf{A}_{(2)})\mathbf{V}^T, \\
\nabla_{\mathbf{Z}} f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) &= (\mathbf{Z}\mathbf{W} - \mathbf{A}_{(3)})\mathbf{W}^T.
\end{aligned} \tag{14}$$

By combining (13) and (14), the solutions of (13) have the update forms for $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ as shown in Algorithm 1.

After normalizing $\mathbf{X}_*^{k+1}, \mathbf{Y}_*^{k+1}, \mathbf{Z}_*^{k+1}$ into matrices $\mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}$ with unit columns, we consider to update $\boldsymbol{\alpha}$:

$$\begin{aligned}
\boldsymbol{\alpha}^{k+1} = \arg\min_{\boldsymbol{\alpha}} \{&\langle \boldsymbol{\alpha} - \boldsymbol{\alpha}^k, \nabla_{\boldsymbol{\alpha}} f(\mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}, \boldsymbol{\alpha}^k)\rangle \\
&+ \frac{s\eta_k}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^k\|^2 + \lambda \sum_{r=1}^{R} \omega_r |\alpha_r|\}.
\end{aligned} \tag{15}$$

Since the function $f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ can be written as $\frac{1}{2}\|\mathbf{a} - \boldsymbol{\alpha}\mathbf{Q}\|_F^2$, the gradient of $f(\bullet)$ on $\boldsymbol{\alpha}$ is

$$\nabla_{\boldsymbol{\alpha}} f(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}) = (\boldsymbol{\alpha}\mathbf{Q} - \mathbf{a})\mathbf{Q}^T.$$

So we can obtain the update form for $\boldsymbol{\alpha}$

In ALRAT (Algorithm 1), $\boldsymbol{\alpha}$ is updated by using the separate soft shrinkage $\mathcal{S}(\alpha_i) = \text{sgn}(\alpha_i)\max\{|\alpha_i| - \lambda\omega_i, 0\}$. Notice that in the regularization parameter λ is fixed in Algorithm 1, we can adaptively choose it for practical computation.

2.2. The choice of weights ω_i and regularized parameter λ

Inspired by Zou's work [20], one choice of weight ω_i is $\frac{1}{|\hat{\alpha}_i|^\gamma}$ where $\hat{\boldsymbol{\alpha}} = \{\hat{\alpha}_1, \cdots, \hat{\alpha}_R\}$ is the conventional solution of (10) without the penalized term $g(\boldsymbol{\alpha})$. For small $\hat{\alpha}_i$, the value of ω_i is large. This means that this adaptive method strengthens the penalization for relatively small $\hat{\alpha}_i$.

Another choice of weights is to consider the conventional Lasso with $\omega_i = 1$ for all i. In this case, the ALRAT algorithm has a simpler form in Step 2 (update step) of the row vector $\boldsymbol{\alpha}$. Specifically, for all the indices i of $\boldsymbol{\alpha}^{k+1}$, use the soft shrinkage:

$$\alpha_i^{k+1} = \begin{cases} \alpha_i^{k+1} - \lambda, & \text{if } \alpha_i^{k+1} > \lambda \\ 0, & \text{if } -\lambda \le \alpha_i^{k+1} \le \lambda \\ \alpha_i^{k+1} + \lambda, & \text{if } \alpha_i^{k+1} < -\lambda \end{cases}$$

In [15], ALRAT with $\omega_i = 1$ for all i is called as the LRAT algorithm. It was shown in [15] that under some assumptions, every limit point of the sequence generated by the LRAT algorithm is a critical point of the objective function satisfying the normalization constraint.

Proposition 2 *[15] If the Jump Assumption is only violated in finite loops, every limit point of the sequence generated by LRAT is a critical point* $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha})$ *of* $\Psi(\bullet)$ *such that* $\mathbf{N}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = 1$.

In this special case, we can further discuss the probabilistic consistency of algorithm, and consider how to choose the regularization parameter λ. For a given regularization parameter $\lambda > 0$, an optimal solution to problem (10) with all $\omega_i = 1$ is denoted by

$$(\hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}, \hat{\boldsymbol{\alpha}}) = \arg\min_{\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\alpha}} \frac{1}{2}\|\mathcal{A} - [\boldsymbol{\alpha}; \mathbf{X}, \mathbf{Y}, \mathbf{Z}]_R\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

where $\mathbf{N}(\mathbf{X},\mathbf{Y},\mathbf{Z}) = 1$. As shown in Section 2.1, we can construct a $R \times (I*J*K)$ matrix $\hat{\mathbf{Q}} = (\hat{\mathbf{q}}_1^T, \cdots, \hat{\mathbf{q}}_R^T)^T = ((\hat{\mathbf{X}} \odot \hat{\mathbf{Y}}) \odot \hat{\mathbf{Z}})^T$ from (12), and vectorize tensor $\mathcal{A}$ into a row vector $\mathbf{a}$.

For convenience, we introduce new notations $\mathbf{b}, \boldsymbol{\theta}, \mathbf{B}$ for $\mathbf{a}^T, \boldsymbol{\alpha}^T, \hat{\mathbf{Q}}^T$ respectively. Thus $\mathbf{b}$ and $\boldsymbol{\theta}$ are column vectors with dimension $I*J*K$ and R, and $\mathbf{B}$ is a $(I*J*K) \times R$ matrix. Furthermore, we have the following equality

$$\frac{1}{2}\|\mathcal{A} - [\boldsymbol{\alpha}; \hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}]_R\|_F^2 + \|\boldsymbol{\alpha}\|_1 = \frac{1}{2}\|\mathbf{b} - \mathbf{B}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1. \tag{16}$$

The optimal solution $\hat{\boldsymbol{\alpha}}^T$ for tensor approximation problem (10) is also an optimal solution $\hat{\boldsymbol{\theta}}$ of a standard l_1-regularized least square problem

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{b} - \mathbf{B}\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1. \tag{17}$$

Assume that $\mathbf{b}$ and $\mathbf{B}$ have a sparse representation structure as

$$\mathbf{b} = \mathbf{B}\boldsymbol{\theta}^* + \boldsymbol{\varepsilon}, \tag{18}$$

where $\boldsymbol{\theta}^*$ is a sparse signal with k non-zero entries ($k < R$) and $\boldsymbol{\varepsilon}$ is a vector with independent subgaussian entries of mean zero and parameter σ^2. The optimal solution $\hat{\boldsymbol{\theta}}$, which is also the $\hat{\boldsymbol{\alpha}}^T$, of problem (17) may become a suitable approximation for the real sparse signal $\boldsymbol{\theta}^*$ from the consistency theory of Lasso [18].

Assume that $\mathbf{B}$ is a full column rank matrix. Then the objective function in problem (17) is strictly convex, and the optimal solution $\hat{\boldsymbol{\theta}}$ to problem (17) is unique and exact $\hat{\boldsymbol{\alpha}}^T$. Denote S and $\hat{S}$ as the index set of non-zero entries in $\boldsymbol{\theta}^*$ and $\hat{\boldsymbol{\theta}}$ respectively. So the sparse signal $\boldsymbol{\theta}^*$ can be rewritten as $({\boldsymbol{\theta}_S^*}^T, \mathbf{0}^T)^T$ and the cardinality of S is k. According to the unknown set S, we can separate columns of the design matrix $\mathbf{B}$ as two parts $(\mathbf{B}_S, \mathbf{B}_{S^C})$, where S^C is the complement of S. Moreover, since $\mathbf{B}_S$ also have full column rank, there exists a unique solution $\hat{\boldsymbol{\theta}}_S$ by solving the restricted Lasso problem:

$$\min_{\boldsymbol{\theta}_S} \frac{1}{2}\|\mathbf{b} - \mathbf{B}_S\boldsymbol{\theta}_S\|_2^2 + \lambda\|\boldsymbol{\theta}_S\|_1. \tag{19}$$

Proposition 3 *[15] Suppose that the sparse structure (18) exists, the sparse signal $\boldsymbol{\theta}^* = ({\boldsymbol{\theta}_S^*}^T, \mathbf{0}^T)^T$ and $\mathbf{B}$ has full column rank. If there exist some parameters γ and μ where $0 < \gamma < 1$ and $\mu > 0$ such that $\|\mathbf{B}_{S^C}^T\mathbf{B}_S(\mathbf{B}_S^T\mathbf{B}_S)^{-1}\|_\infty \leq 1-\gamma$ and $\lambda_{min}(\mathbf{B}_S^T\mathbf{B}_S) \geq \mu$, we have that*

$$Pr\left(\{\hat{S} \subseteq S\} \bigcap \{\|\boldsymbol{\delta}_S\|_\infty \leq \frac{\lambda}{2\sqrt{\mu}} + \lambda\|(\mathbf{B}_S^T\mathbf{B}_S)^{-1}\|_\infty\}\right) \geq 1 - 2R\exp(-\frac{\lambda^2\gamma^2}{8\sigma^2}), \tag{20}$$

where $\hat{S}$ is the index set of non-zero entries in $\hat{\boldsymbol{\theta}}$, and $\boldsymbol{\delta}_S = \hat{\boldsymbol{\theta}}_S - \boldsymbol{\theta}_S^*$ and $\hat{\boldsymbol{\theta}}_S$ is the optimal solution of (19). Furthermore, if the lower bound of the absolute values of elements in $\boldsymbol{\theta}_S^*$ is larger than $\lambda(\frac{1}{2\sqrt{\mu}} + \|(\mathbf{B}_S^T\mathbf{B}_S)^{-1}\|_\infty)$, we have that

$$Pr(\{\hat{S} = S\}) \geq 1 - 2R\exp(-\frac{\lambda^2\gamma^2}{8\sigma^2}). \tag{21}$$

Proposition 3 tells us that if we want to recover the sparsity in (18) with a probability p, we should choose a λ such that $1 - 2R\exp(-\frac{\lambda^2\gamma^2}{8\sigma^2}) > p$ when we know the intrinsic parameters γ and σ^2. So to adaptively give a regularization parameter λ based on the data $\mathcal{A}$, we need to give two guesses on the intrinsic parameters γ and σ^2. We set λ to zero in the LART algorithm, and compute a estimated tensor $\hat{\mathcal{B}} = [\hat{\boldsymbol{\alpha}}; \hat{\mathbf{X}}, \hat{\mathbf{Y}}, \hat{\mathbf{Z}}]_R$ from the tensor data $\mathcal{A}$. The parameter σ^2 is estimated by using the variance $\hat{\sigma}^2$ of all the entries in the difference $\mathcal{A} - \hat{\mathcal{B}}$, and the parameter γ is set as $\hat{\gamma} = 1 - \max\{|\langle \mathbf{B}_i, \mathbf{B}_j\rangle| | i \neq j\}$, where $\mathbf{B}_i$ is the i-th column in $\mathbf{B} = (\hat{\mathbf{X}} \odot \hat{\mathbf{Y}}) \odot \hat{\mathbf{Z}}$. With regularization parameter $\hat{\lambda} = \frac{2}{\hat{\gamma}}\sqrt{2\hat{\sigma}^2\log(200R)}$, the result of our algorithm is shown by using the real data in the next Section.

3. Application to background extraction and video compression

We apply our method to one video dataset from Perception Test Images Sequences[1] (Institute for Infocomm Research, Singapore). This dataset consists of nine surveillance videos. For each video, we choose 220 consecutive frames for our experiments. All of the experiments are executed in C++ with OpenCV2.3.1 and run on a desktop computer with Intel i5 CPU 3.3GHz and 8Gb memory.

For the implementation, the regularization parameter $\hat{\lambda}$ is set to $\frac{2}{\hat{\gamma}}\sqrt{2\hat{\sigma}^2\log(200R)}$ where $\hat{\sigma}^2$ and $\hat{\gamma}$ are computed. The upper bound R of rank is set to $\min\{I, J, K\}$, where I is the number of rows in one frame, J is the number of columns in one frame, and K is the number of frames in the video. All of the weights ω_i are set to one.

As shown in the first two columns of Figure 1, the proposed method for tensor approximation is applied to the background and foreground separation. The background information of the video is captured by the low rank approximation $\hat{\mathcal{B}}$ of the video tensor $\mathcal{A}$, while the foreground can be seen from the residual $\mathcal{A} - \hat{\mathcal{B}}$.

We notice that the complementary information of the low rank tensor $\hat{\mathcal{B}}$ is mainly from the extreme values of the residual $\mathcal{A} - \hat{\mathcal{B}}$. In other words, if we truncate the residual part at a value ϵ and keep both sides of extreme values, we can obtain a compression video tensor which is the low rank tensor $\hat{\mathcal{B}}$ plus the truncation tensor of $\mathcal{A} - \hat{\mathcal{B}}$. The corresponding

[1] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

compression ratio of video tensor $\mathcal{A} \in \mathbb{R}^{I\times J\times K}$ is

$$ratio = \frac{IJK}{(I+J+K)\hat{R}+n(\epsilon)} \tag{22}$$

where $\hat{R}$ is the estimated rank and $n(\epsilon)$ is the total number of voxels for the truncation tensor. The compression results are illustrated in the third column of Figure 1. The specific compression ratio and estimated rank for all the video tensors are shown in Table 1.

Table 1. Dimension, time cost, estimated rank and compression ratio of videos.

video	dimension	time(s)	rank	ratio
Bootstrap	120*160*220	893.97	45	2.03
Campus	128*160*220	1104.13	69	1.9
Curtain	128*160*220	1103.84	78	5.34
Escalator	130*160*220	1032.26	109	2.07
Fountain	128*160*220	1113.44	111	3.55
Hall	144*176*220	1528.29	87	2.04
Lobby	128*160*220	1093.99	128	9.37
ShoppingMall	256*320*220	7296.54	117	2.8
WaterSurface	128*160*220	1097.96	40	2.52

From Table 1, when the dimension of tensor is 128*160*220, the time cost of approximation algorithm is no more than 20 minutes. The estimated rank of WaterSurface is minimum. It is because those consecutive frames for WaterSurface may have a very low rank structure as their similarities of different frames. The Lobby tensor has a high estimated rank, and its compression rate is high up to 9.37. As shown in the 7th row of Figure 1, the compressed Lobby video can obtain a comparatively satisfactory effect compared to the original one while the compressed one only needs around one ninth memory space.

4. Future outlook

The proposed method for low rank approximation has many applications in analyzing multilinear signals. One interesting research direction is to build the relation between tensor computation and intelligent multiple objects tracking. Moreover, shedding light on the convergence rate of the algorithm will helpful in the theoretical study of the sparsity optimization problem.

Acknowledgement: This work is supported in part by NSF of China (Grants No. 11401092).

References

[1] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014. 3

[2] T. Bouwmans and E. Zahzah. Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014. 1

[3] I. Domanov and L. Lathauwer. Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition. *SIAM J. Matrix Anal. Appl.*, 35(2):636–660, 2014. 1

[4] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. 2

[5] S. Foucart and H. Rauhut. A mathematical introduction to compressive sensing. *Birkhäuser*, 2013. 1, 3

[6] N. Hao, M. Kilmer, K. Braman, and R. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM J. Imaging Sciences*, 6(1):437–463, 2013. 1

[7] C. Hillar and L.-H. Lim. Most tensor problems are np-hard. *Journal of the ACM*, 60(6), 2013. 3

[8] J. Huang, S. Ma, and C. Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, 18(4):1603–1618, 2008. 2

[9] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009. 1

[10] L.-H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics*, 23:432–441, 2009. 3

[11] C. Martin, R. Shafer, and B. Larue. An order-p tensor factorization with applications in imaging. *SIAM J. Sci. Comput.*, 35(1):A474–A490, 2013. 1

[12] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006. 2

[13] V. D. Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM J. Matrix Anal. Appl.*, 30(3):1084–1127, 2008. 3

[14] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc. B*, 58(1):267–288, 1996. 1

[15] X. F. Wang and C. Navasca. Low rank approximation of tensors via sparse optimization. http://arxiv.org/abs/1504.05273, 2015. 1, 4, 5

[16] B. Xin, Y. Tian, Y. Wang, and W. Gao. Background subtraction via generalized fused lasso foreground modeling. *CVPR*, 2015. 1

[17] 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(3):1758–1789, 2013. 1

[18] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2567, 2006. 5

[19] H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013. 1

[20] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. 1, 2, 4

Figure 1. The first column is the low rank part. The second column is the residual part. The third column is the compression result. The fourth column is the original frame.