Wang, Xinjue; Ollila, Esa; Vorobyov, Sergiy A.

# Nonnegative Structured Kruskal Tensor Regression

*Published in:*
2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2023

*DOI:*
10.1109/CAMSAP58249.2023.10403474

Published: 01/01/2023

*Document Version*
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

*Please cite the original version:*
Wang, X., Ollila, E., & Vorobyov, S. A. (2023). Nonnegative Structured Kruskal Tensor Regression. In *2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2023* (pp. 441-445). (2023 IEEE 9th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2023). IEEE. https://doi.org/10.1109/CAMSAP58249.2023.10403474

# NONNEGATIVE STRUCTURED KRUSKAL TENSOR REGRESSION

*Xinjue Wang, Esa Ollila, Sergiy A. Vorobyov*

Department of Information and Communications Engineering, Aalto University, Finland

## ABSTRACT

Many contemporary data analysis problems use tensors (multidimensional arrays) as covariates. For example, regression or classification tasks may need to be performed on a set of image covariates sampled from diffusion tensor imaging (DTI), functional magnetic resonance imaging (fMRI), or hyperspectral imaging (HSI). By enforcing a low-rank constraint on the parameter tensor, tensor regression models effectively leverage the temporal and spatial structure of tensor covariates. In this paper, we study Kruskal tensor regression with sparsity and smoothness inducing regularization and nonnegativity constraints. We solve the corresponding penalized nonnegative Kruskal tensor regression (KTR) problem using an efficient block-wise alternating minimization method. The efficiency of the proposed approach is illustrated via simulations.

***Index Terms***— Sparsity, PARAFAC, tensor regression, Kruskal tensor, fused LASSO

## 1. INTRODUCTION

Tensor decompositions are widely utilized in numerous fields such as signal/image processing, geophysics, neuroscience, and bioinformatics [1–5]. Their effectiveness arises from their ability to approximate a high-dimensional tensor with a low-rank approximation, thus offering efficient dimensionality reduction and denoising. The utilization of tensor regression (TR) models has captured significant interest during the past decade and several tensor regression techniques have been proposed in the literature, such as Tucker tensor regression [6], low-rank orthogonally decomposable tensor regression [7], Bayesian KTR [8], Bayesian low rank tensor ring completion [9], and tensor regression network [10].

In TR, the tensor regression parameter $\boldsymbol{\mathcal{B}}$, associated with tensor-valued covariates, is also a tensor of the same shape as the covariate. The KTR model [11], assumes a rank-R CANDECOMP/PARAFAC decomposition (CPD) [12, 13] on the tensor regression parameter $\boldsymbol{\mathcal{B}}$. This decomposition is used to minimize the number of unknown variables while leveraging the structure of the tensor covariate. Additionally, KTR assumes a linear relationship between the tensor covariates $\boldsymbol{\mathcal{X}}_i$ and the tensor regression parameter $\boldsymbol{\mathcal{B}}$.

In this paper, we consider the KTR model with regularization terms that enforce structure and smoothness. In many problems, the tensor covariates and regression outputs are positive valued and the regression coefficient, say $(\boldsymbol{\mathcal{B}})_{ijk}$, corresponding to $(i,j,k)$th feature $(\boldsymbol{\mathcal{X}})_{ijk}$ of the 3D-covariate tensor $\boldsymbol{\mathcal{X}}$ marks the presence/nonpresence as well as intensity of that image signature. This motivates us to constrain each element of $\boldsymbol{\mathcal{B}}$ to be non-negative. On the other hand, tensor image covariates, such as HSI images, contain a lot of structure. For example, the 3rd dimension corresponding to different spectral bands often exhibits smooth variation. The spatial dimensions (first and second dimensions) of HSI image have rather constant profiles locally. These structures are expected to be expressed in the regression parameter $\boldsymbol{\mathcal{B}}$ as well. These observations motivate us to consider the fused LAS SO [14] and fused ridge regularizers on different dimensions of the tensor parameter $\boldsymbol{\mathcal{B}}$.

The paper is structured as follows. Section 2 gives a brief review of basic notations of tensor algebra used in the paper. Section 3 presents our proposed non-negative and sparse tensor regression method. The experimental results are shown in Section 4, and the final section concludes the paper.

## 2. TENSOR ALGEBRA REVIEW AND NOTATIONS

We use $\boldsymbol{\mathcal{B}} = (b_{i_1 \cdots i_D})$ to denote a $D$-way tensor of size $I_1 \times \cdots \times I_D$. The *mode-d matricization* [15], $\mathbf{B}_{(d)}$, reshapes a tensor $\boldsymbol{\mathcal{B}}$ to a $I_d \times \prod_{d' \neq d} I_{d'}$ matrix such that the tensor's $(i_1, \ldots, i_D)$ element corresponds to the $(i_d, j)$ element of the matrix $\mathbf{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d}(i_{d'} - 1) \prod_{d'' < d', d'' \neq d} I_{d''}$. The vectorization operator $\mathrm{vec}(\cdot)$ maps a tensor into a vector by stacking the columns of $\mathbf{B}_{(1)}$ on top of each other, where $\mathbf{B}_{(1)}$ is the mode-1 matricization. The tensor inner product between two tensors of same size are also denoted by the inner product of their vectorized or mode-$d$ matricized counterparts as $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \langle \mathrm{vec}(\boldsymbol{\mathcal{A}}), \mathrm{vec}(\boldsymbol{\mathcal{B}}) \rangle = \langle \mathbf{A}_{(d)}, \mathbf{B}_{(d)} \rangle$, where the latter inner product for matrices can also be expressed compactly using matrix trace as $\langle \mathbf{A}_{(d)}, \mathbf{B}_{(d)} \rangle = \mathrm{tr}(\mathbf{A}_{(d)} \mathbf{B}_{(d)}^\top)$. The outer product of vectors $\mathbf{b}_d \in \mathbb{R}^{I_d}$, $d = 1, \ldots, D$ gives a rank-1 tensor $\mathbf{b}_1 \circ \mathbf{b}_2 \cdots \circ \mathbf{b}_D$ of size $I_1 \times \cdots \times I_D$, whose entries are $(\mathbf{b}_1 \circ \mathbf{b}_2 \cdots \circ \mathbf{b}_D)_{i_1 \cdots i_d} = \prod_{d=1}^{D} b_{d i_d}$.

The rank-$R$ CPD expresses a tensor as a linear combination of rank-1 tensors:

$$\boldsymbol{\mathcal{B}} \equiv [\![ \mathbf{B}_1, \ldots, \mathbf{B}_D ]\!] = \sum_{r=1}^{R} \boldsymbol{\beta}_{r1} \circ \cdots \circ \boldsymbol{\beta}_{rD}, \tag{1}$$

$$\mathbf{B}_d = (\boldsymbol{\beta}_{1d} \ \cdots \boldsymbol{\beta}_{Rd}) \in \mathbb{R}^{I_d \times R}, \ d = 1, \ldots, D. \tag{2}$$

A tensor admitting decomposition (1) is also referred to as a Kruskal tensor [15].

Consider two matrices $\mathbf{A} = (\mathbf{a}_1 \cdots \mathbf{a}_n) \in \mathbb{R}^{m \times n}$ and $\mathbf{B} = (\mathbf{b}_1 \cdots \mathbf{b}_q) \in \mathbb{R}^{p \times q}$. If $\mathbf{A}$ and $\mathbf{B}$ have the same number of columns $n = q$, then the Khatri-Rao product is defined as a columnwise Kronecker product $\mathbf{A} \odot \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \ldots \quad \mathbf{a}_n \otimes \mathbf{b}_n)$, where $\otimes$ denotes the Kronecker product. If $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ is rank-$R$ Kruskal tensor in (1), then [16]:

$$\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}} \rangle = \langle \mathbf{B}_d, \mathbf{X}_{(-d)} \rangle = \mathrm{vec}(\mathbf{B}_d)^\mathsf{T} \mathrm{vec}(\mathbf{X}_{(-d)}) \tag{3}$$

where $\mathbf{X}_{(-d)} = \mathbf{X}_{(d)} (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_{d+1} \odot \mathbf{B}_{d-1} \odot \cdots \odot \mathbf{B}_1)$.

## 3. NONNEGATIVE AND SPARSE TENSOR REGRESSION

### 3.1. The KTR Model

The KTR model [16] assumes a rank-R CPD for the parameter tensor $\boldsymbol{\mathcal{B}}$ in the linear model:

$$\begin{aligned} y_i &= f(\boldsymbol{\mathcal{X}}_i, \mathbf{z}_i; \boldsymbol{\mathcal{B}}, \boldsymbol{\beta}_0) + e_i, \\ &= \langle [\![ \mathbf{B}_1, \ldots, \mathbf{B}_D ]\!], \boldsymbol{\mathcal{X}}_i \rangle + \boldsymbol{\beta}_0^\top \mathbf{z}_i + e_i, \; i = 1, \ldots, N, \end{aligned} \tag{4}$$

where the learnable parameters are the vector covariate $\boldsymbol{\beta}_0 \in \mathbb{R}^{I_0}$ and the factor matrices $\{\mathbf{B}_d\} \in \mathbb{R}^{I_d \times R}$ of the Kruskal tensor regression parameter $\boldsymbol{\mathcal{B}} = [\![ \mathbf{B}_1, \ldots, \mathbf{B}_D ]\!]$, while $e_i, i = 1, ..., N$ are independent and identically distributed (i.i.d.) errors following normal distribution. An intercept can be added to the model by including 1 as the first element of vector covariates $\mathbf{z}_i \in \mathbb{R}^{I_0}, i = 1, \ldots, N$.

The degree of freedom (d.o.f.) of model (4) is

$$K = 1 + I_0 + R \sum_{d=1}^{D} (I_d - 1), \tag{5}$$

which is substantially smaller than $1 + I_0 + \prod_d I_d$ resulting by simply vectorizing $\boldsymbol{\mathcal{X}}_i$-s and then adopting conventional linear regression model for vectorized covariates.

Given the training data $\mathbb{T} = \{y_i, \boldsymbol{\mathcal{X}}_i, \mathbf{z}_i\}_{i=1}^N$ consisting of responses, $y_i \in \mathbb{R}$, tensor-valued predictors (covariates), $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, and conventional vector-valued covariates, $\mathbf{z}_i \in \mathbb{R}^{I_0}$, we aim at learning the estimators of $\boldsymbol{\theta} = \{\boldsymbol{\beta}_0, \{\mathbf{B}_d\}_{d=1}^D\}$ by minimizing the following regularized empirical risk

$$\begin{aligned} L(\boldsymbol{\theta}) &\equiv L(\boldsymbol{\beta}_0, \mathbf{B}_1, \ldots, \mathbf{B}_D) = L(\boldsymbol{\beta}_0, \{\mathbf{B}_d\}) = \\ &= \sum_{i=1}^{N} l(y_i, f(\boldsymbol{\mathcal{X}}_i, \mathbf{z}_i; \boldsymbol{\theta})) + \sum_{d=1}^{D} P_d(\mathbf{B}_d), \end{aligned} \tag{6}$$

where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ is the loss function, $P_d : \mathbb{R}^{I_d \times R} \to \mathbb{R}_{\geq 0}$ is the sparsity and structure/smoothness promoting regularization term acting on $d$th factor matrix. The corresponding optimization problem can be then formulated as

$$\min_{\boldsymbol{\beta}_0, \{\mathbf{B}_d\}} L(\boldsymbol{\beta}_0, \{\mathbf{B}_d\}) \text{ subject to } \mathbf{B}_d \geq 0, \forall d. \tag{7}$$

We illustrate the concept using the loss function

$$l(y_i, f(\boldsymbol{\mathcal{X}}_i, \mathbf{z}_i; \boldsymbol{\theta})) = \frac{1}{2}(y_i - \boldsymbol{\beta}_0^\mathsf{T} \mathbf{z}_i - \langle [\![ \mathbf{B}_1, ..., \mathbf{B}_D ]\!], \boldsymbol{\mathcal{X}}_i \rangle)^2 \tag{8}$$

which is the squared error loss in (6).

### 3.2. Penalties

**Fused LASSO (FL).** As sparsity and smoothness promoting regularizer $P_d(\mathbf{B}_d)$, we first consider the FL [14] penalty

$$\begin{aligned} P_d^{\text{FL}}(\mathbf{B}_d; \boldsymbol{\lambda}_d) &= \lambda_{d1} \|\mathbf{B}_d\|_1 + \lambda_{d2} \|\mathbf{D}_d \mathbf{B}_d\|_1 \\ &= \lambda_{d1} \sum_{r=1}^{R} \sum_{i=1}^{I_d} |(\mathbf{B}_d)_{i,r}| + \lambda_{d2} \sum_{r=1}^{R} \sum_{i=2}^{I_d} |(\mathbf{B}_d)_{i,r} - (\mathbf{B}_d)_{i-1,r}|, \end{aligned} \tag{9}$$

where $\lambda_{d1} \geq 0$ and $\lambda_{d2} \geq 0$ are penalty parameters, and $\mathbf{D}_d \in \mathbb{R}^{(I_d - 1) \times I_d}$ is a first-order difference operator

$$\mathbf{D}_d = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 & 0 \\ 0 & -1 & 1 & \ldots & 0 & 0 \\ & & & \vdots & & \\ 0 & 0 & 0 & \ldots & -1 & 1 \end{bmatrix}.$$

**Algorithm 1:** Nonnegative Fused LASSO Kruskal Tensor Regression ($\texttt{FL}^+\texttt{-KTR}$) method

**Input:** $\mathbb{T} = \{y_i, \boldsymbol{\mathcal{X}}_i, \mathbf{z}_i\}_{i=1}^N$, rank $R \in \mathbb{N}_0^+$, penalty parameters $(\lambda_{d1}, \lambda_{d2}) \in \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$, and stopping threshold $\delta$.

Compute initial $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^{I_0}$ and $\{\mathbf{B}_d^{(0)}\}_{d=1}^D \in \mathbb{R}^{I_d \times R}$ as LS-KTR estimates

**for** $n = 0, 1, \ldots, N_{\text{iter}}$ **do**

  **for** $d = 1, \ldots, D$ **do**

    Solve $\mathbf{B}_d^{(n+1)}$ as minimizer of (11) subject to non-negativity constraint $= \texttt{False}$.

  Compute $\boldsymbol{\beta}_0^{(n+1)}$ as the minimizer of

$$L(\boldsymbol{\beta}_0, \mathbf{B}_1^{(n+1)}, \ldots, \mathbf{B}_D^{(n+1)})$$

  **if** $\dfrac{|L(\boldsymbol{\theta}^{(n+1)}) - L(\boldsymbol{\theta}^{(n)})|}{|L(\boldsymbol{\theta}^{(n)})|} < \delta \vee n > N_{iter}$ **then**

    **return** $\hat{\boldsymbol{\beta}}_0 \leftarrow \boldsymbol{\beta}_0^{(n+1)}$ and $\{\hat{\mathbf{B}}_d\} \leftarrow \{\mathbf{B}_d^{(n+1)}\}$

Use the solution to re-initialize $\boldsymbol{\beta}_0^{(0)} = \hat{\boldsymbol{\beta}}_0$ and $\mathbf{B}_d^{(0)} = (\hat{\mathbf{B}}_d)_+$ and perform steps **3-5** until convergence with non-negativity constraint $= \texttt{True}$, where $(\cdot)_+ = \max(\cdot, 0)$ extracts positive part of its argument in an element-wise way.

**Output:** $\hat{\boldsymbol{\theta}}$, the stationary point of $L(\boldsymbol{\theta})$.

If $\lambda_{d1} = 0$, then one obtains the total variation (TV) penalty. The first penalty term is the LASSO penalty, which encourages sparsity in the coefficients, while the latter TV penalty encourages sparsity in their differences, i.e., similarity of neighboring coefficients.

**Fused ridge (FR).** Smoothness and shrinkage can also be promoted via the fused ridge penalty, which we define as

$$\begin{aligned} P_d^{\text{FR}}(\mathbf{B}_d; \boldsymbol{\lambda}_d) &= \lambda_{d1} \|\mathbf{B}_d\|_{\text{F}}^2 + \lambda_{d2} \|\mathbf{D}_d \mathbf{B}_d\|_{\text{F}}^2 \\ &= \lambda_{d1} \sum_{r=1}^{R} \sum_{i=1}^{I_d} (\mathbf{B}_d)_{i,r}^2 + \lambda_{d2} \sum_{r=1}^{R} \sum_{i=2}^{I_d} ((\mathbf{B}_d)_{i,r} - (\mathbf{B}_d)_{i-1,r})^2, \end{aligned} \tag{10}$$

where the first penalty is the conventional ridge regression (with $\|\cdot\|_{\text{F}}$ denoting the Frobenius norm) penalty that shrinks the coefficients towards zero, while the latter penalty enforces smoothness in neighboring coefficients.

### 3.3. Generic Alternating Algorithm

The optimization problem (6) is not jointly convex with respect to $\mathbf{B}_1, \ldots, \mathbf{B}_D$, but individually convex for each factor matrix $\mathbf{B}_d, d = 1, \ldots, D$, when the other matrices $\mathbf{B}_{d'}$ (where $d' = 1, \ldots, D$ and $d' \neq d$) are held fixed. Accordingly, a stationary solution can be obtained by an alternating block-wise minimization scheme, wherein each component of $\{\boldsymbol{\beta}_0, \mathbf{B}_1, \ldots, \mathbf{B}_D\}$ is updated one at a time, while the other components remain fixed, rendering subproblems convex. Then, in the $(n+1)$th iteration, we solve $\mathbf{B}_d$ as the minimizer of the convex criterion function

$$L(\boldsymbol{\beta}_0^{(n)}, \ldots, \mathbf{B}_{d-1}^{(n+1)}, \mathbf{B}_d, \mathbf{B}_{d+1}^{(n)}, \ldots, \mathbf{B}_D^{(n)}) + P_d(\mathbf{B}_d; \boldsymbol{\lambda}_d) \tag{11}$$

subject to the constraint $\mathbf{B}_d \geq 0$. For example, with the squared error loss and FL-penalty (9), the problem is essentially reduced

| SNR | | 20dB | | | | | 10dB | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Signal | $R$ | LS | FL\|FL$^+$ | FR\|FR$^+$ | FL$^+$-FR$^+$ | FR$^+$-FL$^+$ | LS | FL\|FL$^+$ | FR\|FR$^+$ | FL$^+$-FR$^+$ | FR$^+$-FL$^+$ |
| *Cross* | 1 | 17.27 | 15.90 \| **15.82** | 16.76 \| 16.32 | 16.11 | 16.00 | 18.39 | 16.07 \| **15.98** | 17.60 \| 17.06 | 16.64 | 16.41 |
| | 2 | 2.66 | 0.85 \| **0.73** | 2.58 \| 1.79 | 1.41 | 1.26 | 8.44 | 2.56 \| **2.50** | 7.31 \| 5.25 | 4.09 | 4.02 |
| | 3 | 4.51 | 0.77 \| **0.71** | 2.97 \| 1.95 | 1.28 | 1.33 | 14.84 | 2.45 \| **2.29** | 9.27 \| 6.02 | 4.03 | 3.84 |
| *Gradient* | 2 | 3.71 | **1.94** \| 2.01 | 3.21 \| 3.09 | 2.46 | 2.70 | 15.40 | **5.90** \| 5.99 | 12.50 \| 10.97 | 8.32 | 7.89 |
| | 3 | 5.75 | **1.68** \| 1.80 | 3.36 \| 3.29 | 2.62 | 2.79 | 21.54 | **6.12** \| 9.35 | 18.75 \| 17.43 | 10.29 | 10.81 |
| | 4 | 7.56 | **1.74** \| 1.98 | 3.82 \| 3.64 | 2.65 | 2.76 | 27.01 | 6.61 \| **6.27** | 22.99 \| 19.86 | 13.11 | 12.02 |
| *Floor* | 2 | 4.30 | 1.40 \| **1.39** | 2.46 \| 2.33 | 1.63 | 2.31 | 13.53 | 4.77 \| **4.48** | 11.72 \| 11.34 | 7.74 | 7.25 |
| | 3 | 5.88 | 1.36 \| **1.35** | 2.71 \| 2.55 | 1.55 | 2.48 | 19.02 | 5.42 \| **5.11** | 17.55 \| 16.57 | 12.23 | 9.63 |
| | 4 | 7.54 | 1.44 \| **1.39** | 3.10 \| 2.84 | 1.36 | 1.82 | 23.94 | 4.12 \| **4.09** | 20.05 \| 17.66 | 8.96 | 7.25 |

**Table 1**: The estimation results Err $= \|\hat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}\|_{\mathrm{F}}$ of different methods using KTR model with different ranks $R$. For parameter *cross*, the sample size was $N = 1000$, while for *gradient* and *floor* signal parameters the sample size was $N = 2000$.

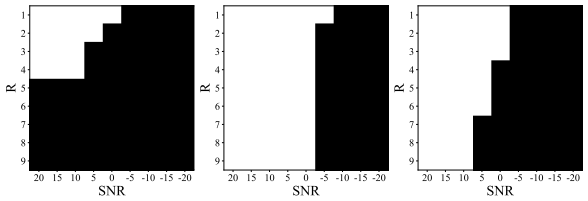
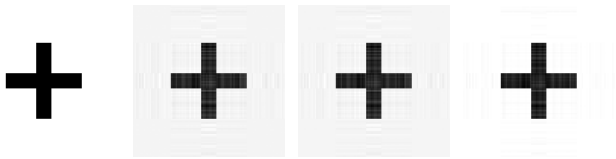



**Fig. 1**: Phase transition results achieved by LS (left), FL (middle) and FR (right) estimators when the true parameter $\boldsymbol{\mathcal{B}}$ is the image *cross* shown in Fig. 2(a) and $N = 1000$. SNR (X-axis, dB) increases with step size of 5dB, and Y-axis shows increasing rank $R$. White blocks indicate successful recovery while black block denote the failed one.



(a) Ground-Truth (b) LS (c) FR (d) FR$^+$

**Fig. 2**: The true signal parameter $\boldsymbol{\mathcal{B}}$ (*cross*) and estimated signals $\hat{\boldsymbol{\mathcal{B}}}$ obtained by different methods using $R = 2$ KTR model. SNR $=$ 20dB and $N = 1000$.

to solving the conventional fused LASSO regression under a non-negativity constraint, thanks to the linear form (3). We term the resulting blockwise alternating minimization scheme designed to solve the nonnegative fused KTR model as the FL$^+$-KTR method. This method is detailed in Algorithm 1. The acronym FR$^+$-KTR designates the case in which the FR penalty (10) is applied across all dimensions of factor matrices and is subject to a non-negativity constraint. Conversely, FL–KTR and FR-KTR refer to cases where no non-negativity constraint is imposed, signified by setting the non-negativity condition in Algorithm 1 as (= `False`).

## 4. NUMERICAL EXAMPLES

To visually demonstrate the effectiveness of our methods, we employ 2-D black-and-white images[1]. However, we acknowledge that these images do not fully represent the complexity and diversity of real-world applications.

The responses $y_i, i = 1, ..., N$ are produced using model (4), where $\boldsymbol{\beta}_0 = \mathbf{1} \in \mathbb{R}^5$, covariates $\boldsymbol{\mathcal{X}}_i, \mathbf{z}_i$ are generated randomly from $\mathcal{N}(0, 1)$ normal distributions as i.i.d. entries, and error terms $e_i, i = 1, ..., N$ are i.i.d. white Gaussian noise. We use three different structured 2D-signals $\boldsymbol{\mathcal{B}} \in [0, 1]^{I_1 \times I_2}$ having varying degrees of zero-values and smoothness/structure.

Penalty parameters $\{\lambda_{d1}, \lambda_{d2}\}_{d=1}^{D}$ are optimized as follows. For each dimension $d$, a grid $\{\lambda^{(i)}\}_{i=0}^{L}$ is defined with equispaced intervals on the log-scale: $\lambda^{(L)} = \epsilon\lambda^{(0)}$, $\lambda^{(j)} = \epsilon^{j/L}\lambda^{(0)} = \epsilon^{1/L}\lambda^{(j-1)}$, and thus $\log(\lambda^{(j-1)}) - \log(\lambda^{(j)}) = (\log(\lambda^{(0)}) - \log(\lambda^{(L)}))/L$, where $\lambda^{(0)}$ is the maximum value of the penalty parameter, and $L$ is the grid size parameter. We use $\lambda^{(0)} = 2 \cdot N$, $\epsilon = 5 \cdot 10^{-3}$ and $L = 6$ for all images. The stopping threshold of our algorithm is set as $\delta = 5 \times 10^{-4}$, $N_{\text{iter}} = 125$. The number of Monte-Carlo (MC) runs is 101 in all experiments[2], where each algorithm is implemented in `Python` using `JAX` [17] and `prox-TV` [18].

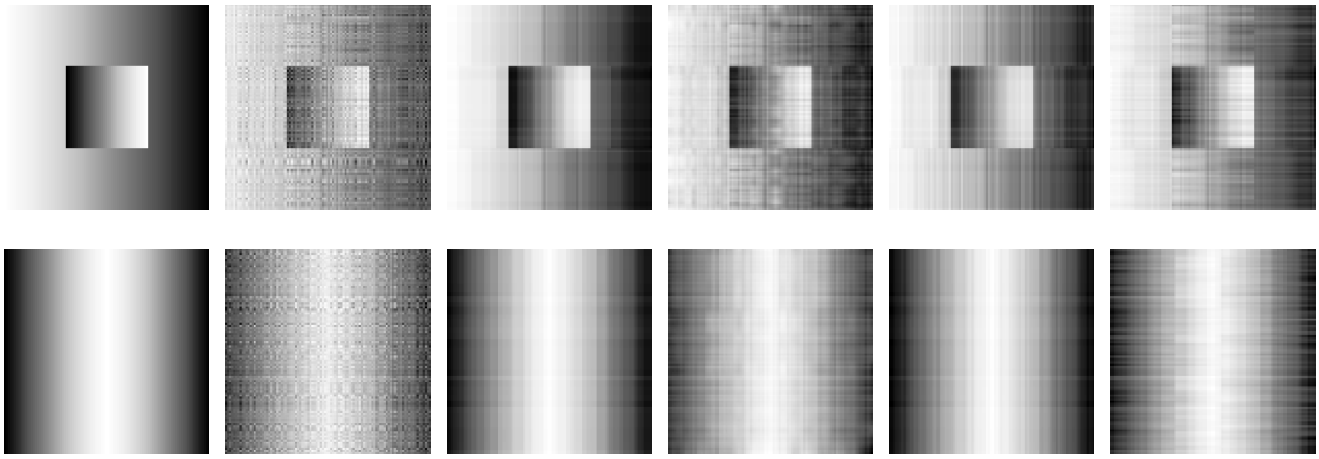
**Phase Transition.** We examine the influence of used KTR model rank $R$ and signal-to-noise ratio (SNR) on the performance of LS, FL, and FR estimators. The true parameter $\boldsymbol{\mathcal{B}}$ is the image *cross* shown in Fig. 2(a). The total sample length is $N = 1000$, the rank $R$ ranges from 1 to 10 and the SNR varies from $-20$ to 20 dB in increments of 5 dB. We consider the estimation successful if the normalized root mean squared estimation error meets the criterion $\|\hat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}\|_{\mathrm{F}}/\|\boldsymbol{\mathcal{B}}\|_{\mathrm{F}} \leq 1$. As illustrated in Fig.1, the FL estimator (middle) is capable of recovering a greater number of cases, including those with lower SNR and higher rank, in comparison to the LS (left) and FR (right) estimators.

**Effect of Nonnegativity.** Fig. 2 compares the estimation results produced by LS-, FR-, and FR$^+$-KTR methods when the signal $\boldsymbol{\mathcal{B}}$ is a *cross* image of size $100 \times 100$ as depicted in Fig. 2(a). Here we set SNR $=$ 20dB and $N = 1000$. Comparing Fig.(b), (c) and (d), it is obvious that the non-negativity constraint used by FR$^+$ method eliminates the erroneously predicted gray pixels in the signal image, thereby enhancing the visual quality of the estimation.

Fig. 3(a) displays the other two true signals $\boldsymbol{\mathcal{B}}$, referred to as *gradient* (top) and *floor* (bottom), both with size of $100 \times 100$. Fig. 3(b)-(d) show the median performance of LS, FL$^+$, and FR$^+$ estimators over 101 MC runs when SNR$=$ 20dB and $N = 2000$. Fig. 3(e)-(f) present KTR estimators employing unique penalties for different dimensions. Specifically, FL$^+$-FR$^+$ utilizes the FL penalty for the first

[1]The extension to more than 2 dimensions is straightforward, but requires more space for illustrations.

[2]Codes of our method are available at https://github.com/Yidingzaoshui/NSKTR.



(a) Ground-Truth (b) LS (c) FL$^+$ (d) FR$^+$ (e) FL$^+$-FR$^+$ (f) FR$^+$-FL$^+$

**Fig. 3**: Comparison of estimated signal parameter $\hat{\boldsymbol{\mathcal{B}}}$ using $R = 3$ KTR model. The shown estimate $\hat{\boldsymbol{\mathcal{B}}}$ correspond to one having median squared error over 101 MC simulations. SNR $= 20$dB and $N = 2000$.

dimension and the FR penalty for the second, while FR$^+$-FL$^+$ does the opposite. In both cases, the non-negativity constraint is enforced. This comparison illustrates the effect of applying distinct penalties on different dimensions. The resulting estimation patterns vary noticeably, as seen when comparing (c) and (d), as well as (e) and (f). Table 1 presents a performance comparison of the aforementioned five estimators. For both FL and FR methods, we also report the results without the non-negativity constraint. In each row, the best estimation value of the median errors is marked in bold. Overall, the FL and FL$^+$ estimators generally outperform the others. Particularly for the *cross*, which is highly sparse (having a large proportion of 0-values), the non-negativity constraint visibly improves the estimation. Conversely, for other signals such as *gradient* with fewer zeros, the benefits of imposing non-negativity are less pronounced.

## 5. CONCLUSION

This paper presents a nonnegative and sparse linear Kruskal tensor regression model. Four image signals are used to demonstrate its efficacy empirically. The practical utility of the nonnegativity constraint for sparse images is emphasized, and the impact of employing diverse regularization techniques on distinct dimensions is illustrated.

## 6. REFERENCES


[1] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, July. 2017.

[2] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[3] Y. Chen, W. He, N. Yokoya, and T.-Z. Huang, "Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3556–3570, Aug. 2020.

[4] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of eeg signals: A brief review," *J. Neurosci. Methods*, vol. 248, pp. 59–69, June. 2015.

[5] F. Huang, X. Yue, Z. Xiong, Z. Yu, S. Liu, and W. Zhang, "Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations," *Briefings Bioinf.*, vol. 22, no. 3, July. 2020.

[6] X. Li, D. Xu, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," *Stat. Biosci.*, vol. 10, no. 3, pp. 520–545, Mar. 2018.

[7] J. Poythress, J. Ahn, and C. Park, "Low-rank, orthogonally decomposable tensor regression with application to visual stimulus decoding of fMRI data," *J. Comput. Graphical Stat.*, vol. 31, no. 1, pp. 1–14, Aug. 2021.

[8] R. Guhaniyogi, S. Qamar, and D. B. Dunson, "Bayesian tensor regression," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2733–2763, Aug. 2017.

[9] Z. Long, C. Zhu, J. Liu, and Y. Liu, "Bayesian low rank tensor ring for image recovery," *IEEE Trans. Image Process.*, vol. 30, pp. 3568–3580, Mar. 2021.

[10] J. Kossaifi, Z. C. Lipton, A. Kolbeinsson, A. Khanna, T. Furlanello, and A. Anandkumar, "Tensor regression networks," *J. Mach. Learn. Res.*, vol. 21, no. 1, Jan. 2020.

[11] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

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

[13] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis," *UCLA Working Pap. Phonet.*, vol. 16, pp. 1–84, Dec. 1970.

[14] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. R. Stat. Soc., B: Stat. Methodol.*, vol. 67, pp. 91–108, Dec. 2005.

[15] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[16] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *J. Am. Stat. Assoc.*, vol. 108, no. 502, pp. 540–552, July. 2013.

[17] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.

[18] A. Barbero and S. Sra, "Fast newton-type methods for total variation regularization," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, Washington, USA, June 28 - July 2, 2011, pp. 313–320.