Robust tensor regression with applications in imaging

Esa Ollila
Department of Signal Processing and Acoustics
Aalto University
Finland
esa.ollila@aalto.fi

Hyon-Jung Kim
Computing Sciences Unit
Tampere University
Finland
hyon-jung.kim@tuni.fi

***Abstract*—Tensor regression models have gained popularity in problems where covariates are tensors (multidimensional arrays) such as images. Tensor regression models are able to efficiently exploit the temporal and/or spatial structure of tensor covariates (e.g., in hyperspectral or fMRI images) by imposing a low-rank assumption on the parameter tensor. In this paper, we propose a robust tensor regression estimation method within the framework of Kruskal tensor regression model. We consider Huber's concomitant criterion for regression and scale as it offers a good tradeoff between robustness and computational feasibility. An efficient alternating minimization algorithm is proposed for estimating the unknown regression parameters. Our simulation studies with synthetic image signals illustrate that the proposed estimator performs similarly compared to benchmark method when errors are Gaussians but offers superior performance in heavy-tailed noise, while having similar computational complexity.**

***Index Terms*—PARAFAC, tensor regression, Huber's criterion, robustness, outliers**

I. Introduction

In several application domains, covariates are tensors (multidimensional arrays). This is especially the case in several imaging applications, where an acquired image may be considered as a covariate, and one would like to model the relationship of the image covariate to a response variable, which may be categorical or continuous. Such problem settings lead to the development of tensor regression or classification methods.

Tensor regression (TR) models (e.g., [1]–[3]) make use of tensor decompositions. Traditionally tensor decompositions have been used in psychometrics, chemometrics, or signal processing [4], [5]. Main uses of tensor decompositions are very similar to singular value decomposition (SVD). Namely, they often allow to approximate a tensor with one of a lower rank, thus providing effective compression (data reduction) as well as denoising.

In TR model, both the covariates and the regression parameter are tensors. A brute force approach that vectorizes the tensor covariates $\boldsymbol{\mathcal{X}}_i$ and then use traditional regression methods to vectorized covariates fail due to ultra high-dimensionality of the obtained regression model. Namely, mapping an $I \times J \times K$ tensor into an $(IJK) \times 1$ vector implies a high-dimensional linear regression model even when the dimensions I, J, and K of the tensor are only moderately large. Such vectorizing approach also completely ignores the structural information within the tensor, such as possible temporal and/or spatial correlations, low-rankness or sparsity which are often present in applications in which the tensor covariates are images.

One of the first TR models is the *Kruskal tensor regression (KTR) model* proposed by [1] which assumes a linear relationship $\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle$ between the tensor covariate $\boldsymbol{\mathcal{X}}_i$ and tensor regression parameter $\boldsymbol{\mathcal{B}}$, but imposes a rank-R CANDECOMP/PARAFAC decomposition (CPD) [6], [7] for $\boldsymbol{\mathcal{B}}$ in order to reduce the number of unknowns and for exploiting structure present in tensor covariates. Also other tensor decompositions have been considered, such as the Tucker tensor regression (TTR) model in [2], the low-rank orthogonally decomposable tensor regression (LODTR) model in [3], or the Bayesian estimation framework of the KTR model in [8].

In this paper, we propose a robust estimation method for linear KTR model that can cope with outliers and heavy-tailed error distributions. Thus far, robustness in tensor data analysis has been considered mainly from perspectives of finding robust estimators of tensor decomposition parameters (e.g., [9], [10]). This work bridges this gap and brings robust estimation [11], [12] to learning problems involving tensors. In our construction, we utilize the Huber's [11, Section 7.7] criterion proposed for joint estimation of regression and scale. One benefit of this approach (as compared to many other robust regression techniques) is that it can be computed efficiently using minimization majorization algorithm [13], thus providing a good tradeoff between robustness and computational feasibility.

The paper is structured as follows. Section II gives a brief review of basic concepts in tensor algebra and introduces the CPD. Section III introduces the KTR model and the proposed robust estimation method that minimizes Huber's criterion. Section IV provides simulation results while Section V concludes.

II. Tensor algebra review

Let $\boldsymbol{\mathcal{B}} = (b_{i_1 \cdots i_D})$ denote a D-way tensor of size $I_1 \times \cdots \times I_D$. The *mode-$d$ matricization*, $\mathbf{B}_{(d)}$, maps a tensor $\boldsymbol{\mathcal{B}}$ into a $I_d \times \prod_{d' \neq d} I_{d'}$ matrix such that the $(i_1, \ldots, i_D)$ element of the tensor $\boldsymbol{\mathcal{B}}$ maps to the (i_d, j) element of the matrix $\mathbf{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d} (i_{d'} - 1) \prod_{d'' < d', d'' \neq d} I_{d''}$. Let $\mathrm{vec}(\cdot)$ denote a vectorization operator that transforms a tensor into a column

vector by stacking the columns of mode-1 matricization $\mathbf{B}_{(1)}$ on top of each other.

The inner product between two tensors of same size is defined as

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \sum_{i_1 \cdots i_D} a_{i_1 \cdots i_D} b_{i_1 \cdots i_D}.$$

One may express the tensor inner product using vectorization or d-mode matricization of the tensors 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 be expressed compactly using matrix trace as $\langle \mathbf{A}_{(d)}, \mathbf{B}_{(d)} \rangle = \mathsf{tr}(\mathbf{A}_{(d)} \mathbf{B}_{(d)}^\top)$. An outer product of vectors $\mathbf{b}_d \in \mathbb{R}^{I_d}$, $d = 1, \ldots, D$, is an $I_1 \times \cdots \times I_D$ rank-1 tensor, $\mathbf{b}_1 \circ \mathbf{b}_2 \cdots \circ \mathbf{b}_D$, with entries $(\mathbf{b}_1 \circ \mathbf{b}_2 \cdots \circ \mathbf{b}_D)_{i_1 \cdots i_d} = \prod_{d=1}^D b_{d i_d}$.

A tensor $\boldsymbol{\mathcal{B}}$ is said to admit a rank-R CANDECOMP/PARAFAC decomposition (CPD) [6], [7] if it can be expressed as a sum of R rank-1 tensors:

$$\begin{aligned} \boldsymbol{\mathcal{B}} &\equiv [\![\mathbf{B}_1, \ldots, \mathbf{B}_D]\!] = \sum_{r=1}^R \boldsymbol{\beta}_{r1} \circ \cdots \circ \boldsymbol{\beta}_{rD}, \\ \mathbf{B}_d &= (\boldsymbol{\beta}_{1d} \ \cdots \boldsymbol{\beta}_{Rd}) \in \mathbb{R}^{I_d \times R}, \quad d = 1, \ldots, D. \end{aligned} \tag{1}$$

Tensor admitting decomposition (1) is also called as Kruskal tensor.

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 the mp-by-n columnwise Kronecker product

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \ldots & \mathbf{a}_n \otimes \mathbf{b}_n \end{pmatrix},$$

where $\otimes$ denotes the Kronecker product. If $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ admits a rank-R decomposition (1), then [1]:

$$\begin{aligned} &\langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}} \rangle \\ &= \mathsf{tr}\big(\mathbf{B}_d (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_{d+1} \odot \mathbf{B}_{d-1} \odot \cdots \odot \mathbf{B}_1)^\top \mathbf{X}_{(d)}^\top \big) \\ &= \big\langle \mathbf{B}_d, \mathbf{X}_{(d)} (\mathbf{B}_D \odot \cdots \odot \mathbf{B}_{d+1} \odot \mathbf{B}_{d-1} \odot \cdots \odot \mathbf{B}_1) \big\rangle. \end{aligned}$$

III. Robust tensor regression

A. The Kruskal tensor regression (KTR) model

Given responses, $y_i \in \mathbb{R}$, and tensor-valued predictors (covariates), $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{I_1 \times \cdots \times I_D}$, and conventional vector-valued covariates, $\mathbf{z}_i \in \mathbb{R}^{I_0}$, the task is to learn a best possible predictor function $f(\mathbf{z}, \boldsymbol{\mathcal{X}})$ using the available training data $\mathbb{T} = \{(y_i, \mathbf{z}_i, \boldsymbol{\mathcal{X}}_i), i = 1, \ldots, N\}$.

The KTR model [1] assumes a rank-R CPD for the parameter tensor $\boldsymbol{\mathcal{B}}$ in the linear model:

$$\begin{aligned} &y_i = \boldsymbol{\beta}_0^\top \mathbf{z}_i + \langle \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{X}}_i \rangle + e_i, i = 1, \ldots, N \\ &\text{subject to } \boldsymbol{\mathcal{B}} = [\![\mathbf{B}_1, \ldots, \mathbf{B}_D]\!], \end{aligned} \tag{2}$$

where the learnable parameters are the tensor $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ with CPD and vector $\boldsymbol{\beta}_0 \in \mathbb{R}^{I_0}$, while e_i-s are independent and identically distributed (i.i.d.) random error terms that are assumed to follow an unspecified distribution $F(e/\sigma)$ symmetric around zero, where $\sigma > 0$ denotes the unknown scale parameter. We also note that an intercept can be added to the tensor linear regression by including 1 as the first element of the vector covariate $\mathbf{z}_i$. The main benefit of KTR model (2) is its reduction of dimensionality of the parameter space. The degree of freedom (d.o.f.) is

$$K = 1 + I_0 + R \sum_{d=1}^D I_d \tag{3}$$

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 the vectorized covariates.

In the specific case, when the array covariates are matrices ($D = 2$) and rank is $R = 1$, so $\boldsymbol{\mathcal{B}} = \boldsymbol{\beta}_1 \circ \boldsymbol{\beta}_2 = \boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top$, then the model (2) can be expressed as a simple bilinear function of matrix covariates $\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{I_1 \times I_2}$ in the form

$$y_i = \boldsymbol{\beta}_0^\top \mathbf{z}_i + \boldsymbol{\beta}_1^\top \boldsymbol{\mathcal{X}}_i \boldsymbol{\beta}_2 + e_i.$$

This follows by noticing that $\langle \boldsymbol{\beta}_1 \circ \boldsymbol{\beta}_2, \boldsymbol{\mathcal{X}} \rangle = \mathsf{tr}(\boldsymbol{\beta}_1 \boldsymbol{\beta}_2^\top \boldsymbol{\mathcal{X}}^\top) = \langle \boldsymbol{\beta}_1, \boldsymbol{\mathcal{X}} \boldsymbol{\beta}_2 \rangle$.

If one assumes that the error terms follow a Gaussian distribution, $e_i \sim \mathcal{N}(0, \sigma^2)$, then minimizing the negative log-likelihood function is equivalent to minimization of the residual sum of squares (RSS) criterion,

$$\mathrm{RSS}(\tilde{\boldsymbol{\theta}}) = \sum_{i=1}^N (y_i - \boldsymbol{\beta}_0^\top \mathbf{z}_i - \langle [\![\mathbf{B}_1, \ldots, \mathbf{B}_D]\!], \boldsymbol{\mathcal{X}}_i \rangle)^2, \tag{4}$$

where $\tilde{\boldsymbol{\theta}}$ denotes the set of unknown parameters in (4), $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\beta}_0, \{\mathbf{B}_d\}_{d=1}^D\}$. The problem is non-convex, but a local minimum of (4) can be found by blockwise alternating least squares algorithm [1], which we refer to Kruskal Tensor Regression Least Squares (KTR-LS) estimator.

B. Robust tensor regression estimator

Consider instead a criterion function,

$$\begin{aligned} Q(\boldsymbol{\theta}) &\equiv Q(\sigma, \boldsymbol{\beta}_0, \mathbf{B}_1, \ldots, \mathbf{B}_D) \\ &= \sum_{i=1}^N \rho_c \left(\frac{y - \boldsymbol{\beta}_0^\top \mathbf{z}_i - \langle [\![\mathbf{B}_1, \ldots, \mathbf{B}_D]\!], \boldsymbol{\mathcal{X}}_i \rangle}{\sigma} \right) \end{aligned} \tag{5}$$

where $\boldsymbol{\theta}$ denotes the set of unknown parameters in (5), $\boldsymbol{\theta} = \{\sigma, \boldsymbol{\beta}_0, \{\mathbf{B}_d\}_{d=1}^D\}$, and $\rho_c(\cdot)$ is Huber's loss function [11], defined by

$$\rho_c(e) = \frac{1}{2} \times \begin{cases} |e|^2, & \text{for } |e| \le c \\ 2c|e| - c^2, & \text{for } |e| > c, \end{cases} \quad e \in \mathbb{R},$$

with (user defined) robustification parameter $c \in (0, \infty)$, and $\sigma > 0$ is the scale (dispersion) parameter of the error terms. In the case that the scale σ is known, then $Q(\boldsymbol{\theta})$ provides a robust alternative to RSS criterion. In practise, the scale σ of the error distribution is unknown, and thus we consider robust joint estimation of scale and the regression parameters and minimize the criterion function

$$L(\boldsymbol{\theta}) = Q(\sigma, \boldsymbol{\beta}_0, \mathbf{B}_1, \ldots, \mathbf{B}_D)\sigma + \alpha(N - K)\sigma, \tag{6}$$

with respect to $\boldsymbol{\theta}$, where $\alpha > 0$ is a fixed (known) consistency factor defined as [13]

$$\alpha = \frac{1}{2}\mathbb{E}[(\rho_c'(e))^2] = \frac{c^2}{2}(1 - F_{\chi_1^2}(c^2)) + \frac{1}{2}F_{\chi_3^2}(c^2), \qquad (7)$$

chosen to ensure that the obtained estimator of σ is Fisher-consistent for the unknown scale σ when the error terms have a Gaussian distribution, $e_i \sim \mathcal{N}(0, \sigma^2)$. Above in (7), $F_{\chi_k^2}(\cdot)$ denotes the c.d.f. of a chi-squared distribution with k degrees of freedom. One may view this cost function as a generalization of RSS criterion in (4), where the LS-loss $\rho_{\text{LS}}(e) = e^2$ in the summation is replaced by the loss function $\rho_{\text{Huber}}(e) = \sigma(\rho_c(e/\sigma) + \alpha(1 - K/N))$. Moreover, if the tensor covariates are all zeros, $\boldsymbol{\mathcal{X}}_i = \mathbf{0}\ \forall i$, then (6) reduces to original criterion considered by Huber [11, Section 7.7].

Although the minimization problem in (6) is not convex in $\mathbf{B}_1, \ldots, \mathbf{B}_D$ jointly, it is convex in $\mathbf{B}_d$ individually, when keeping other parameters fixed. This implies that the stationary solution can be found by alternating minimization (or coordinate descent) scheme, updating $(\boldsymbol{\beta}_0, \sigma), \mathbf{B}_1, \ldots, \mathbf{B}_D$, in turn while keeping other components fixed. These convex subproblems can be efficiently solved using minimization majorization (MM) algorithm developed for Huber's criterion in [13]. The resulting blockwise alternating minimization scheme for solving Kruskal tensor regression for Huber's criterion, referred to as KTR-Hub estimator, is given in Algorithm 1. Note that the solutions are found using multiple initializations in order to obtain an excellent local minimum. In our numerical examples, we run for $N_{rep} = 10$ random initialisations and do the same for alternating KTR-LS algorithm proposed in [1]. Note that steps 1, 3 and 4 in Algorithm 1 are solved using an MM algorithm of [13].

In Algorithm 1 one assumes that rank R is known and given as an input to the algorithm. This is not the case in practise, and an appropriate value of rank R should be adaptively chosen based on the training data. We tested an adaptation of Bayesian information criterion (BIC), so choosing a model that minimize the criterion $2L(\hat{\boldsymbol{\theta}}) + \log(N)K$, where K is the d.o.f. in the model given by (3) and $L(\hat{\boldsymbol{\theta}})$ serves as surrogate for negative log-likelihood of rank R KTR model in BIC criterion. However, this adaptation of BIC to tensor regression framework did not yield satisfactory results neither for KTS-LS or KTR-Hub methods. We leave the estimation of rank R as a topic for future work.

Algorithm 1: Kruskal tensor regression for Huber's criterion (KTR-Hub) using blockwise alternating minimization.

input : Response $\mathbf{y} \in \mathbb{R}^N$, covariates $\{\mathbf{z}_i, \boldsymbol{\mathcal{X}}_i\}_{i=1}^N \in \mathbb{R}^{I_0} \times \mathbb{R}^{I_1 \times \cdots \times I_D}$, and rank $R \in \mathbb{N}_0^+$ of CPD, threshold c

1 Solve $(\sigma^{(0)}, \boldsymbol{\beta}_0^{(0)})$ as the minimizer of

$$\sum_{i=1}^{N} \rho_c\Big(\frac{y_i - \boldsymbol{\beta}_0^\top \mathbf{z}_i}{\sigma}\Big)\sigma + \alpha(N-K)\sigma.$$

2 Draw $\{\mathbf{B}_d^{(0)}\}_{d=1}^D \in \mathbb{R}^{I_d \times R}$ random matrices.

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

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

3 Compute $\mathbf{B}_d^{(n+1)}$ as the minimizer of

$$Q(\sigma^{(n)}, \boldsymbol{\beta}_0^{(n)}, \ldots, \mathbf{B}_{d-1}^{(n+1)}, \mathbf{B}_d, \mathbf{B}_{d+1}^{(n)}, \ldots, \mathbf{B}_D^{(n)})$$

4 Solve $(\boldsymbol{\beta}_0^{(n+1)}, \sigma^{(n+1)})$ as the minimizer of

$$Q(\sigma, \boldsymbol{\beta}_0, \mathbf{B}_1^{(n+1)}, \ldots, \mathbf{B}_D^{(n+1)})\sigma + \alpha(N-K)\sigma$$

5 **if** $\dfrac{|L(\boldsymbol{\theta}^{(n+1)}) - L(\boldsymbol{\theta}^{(n)})|}{|L(\boldsymbol{\theta}^{(n)})|} < \epsilon$ **then**

  return

  $\hat{\boldsymbol{\theta}} \leftarrow (\sigma^{(n+1)}, \boldsymbol{\beta}_0^{(n+1)}, \mathbf{B}_1^{(n+1)}, \ldots, \mathbf{B}_D^{(n+1)})$

6 Repeat for N_{rep} (e.g., $N_{rep} = 10$) and choose the solution that yielded the minimum value for $L(\hat{\boldsymbol{\theta}})$.

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

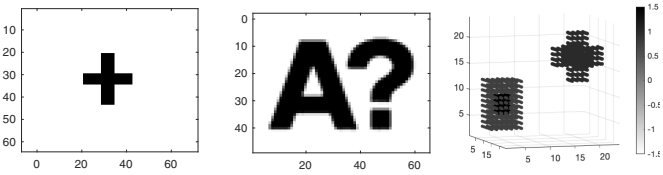

Fig. 1: True image signals $\hat{\boldsymbol{\mathcal{B}}}$.

IV. Numerical examples

The responses y_i are generated according to model (2), where $\boldsymbol{\beta}_o = \mathbf{1} \in \mathbb{R}^5$. The regular covariate $\mathbf{z}_i$ and the image covariate $\boldsymbol{\mathcal{X}}_i$ are randomly generated as having i.i.d. entries from $\mathcal{N}(0, 1)$ distribution. Then error terms e_i-s are i.i.d. and having different types of heavy-tailed distributions. The signal image $\boldsymbol{\mathcal{B}}$ is then learned through the linear association between y_i and the vector-tensor covariate pairs $(\mathbf{z}_i, \boldsymbol{\mathcal{X}}_i)$.

Figure 1 display the used image tensors $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times I_2}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. The first image ("cross"), of size 64×64, also used in the study in [1] is highly sparse with only few non-zero elements (all equal to 1) forming a shape of a cross appearing in the middle of the image. The proportion of 1-s is 5.3% among all $64^2 = 4096$ elements. The image in the middle displays Aalto University logo of size 46×65 while image on the right consisting of two nested cubes and a cross has. The 3D-signal has size $24 \times 24 \times 24$.

Figure 2 shows results for cross image signal when $R = 2$ and the noise has an ε-contaminated Gaussian distribution, so e_i-s are generated from $\mathcal{N}(0, \sigma^2)$ with probability $1 - \varepsilon$ and from $\mathcal{N}(0, (\lambda\sigma)^2)$ with probability $\varepsilon \in [0, 1]$. We used parameters $\varepsilon = 0.1$, $\sigma^2 = 1$ and $\lambda = 20$. The sample length is $N = 500$. We computed the estimates $\hat{\boldsymbol{\mathcal{B}}}$ for 101 Monte-Carlo (MC) trials, and display the obtained estimates for smallest, median, and largest root squared error, defined by

$$\text{Err} = \|\hat{\boldsymbol{\mathcal{B}}} - \boldsymbol{\mathcal{B}}\|,$$

which is reported in the figure titles. We use threshold

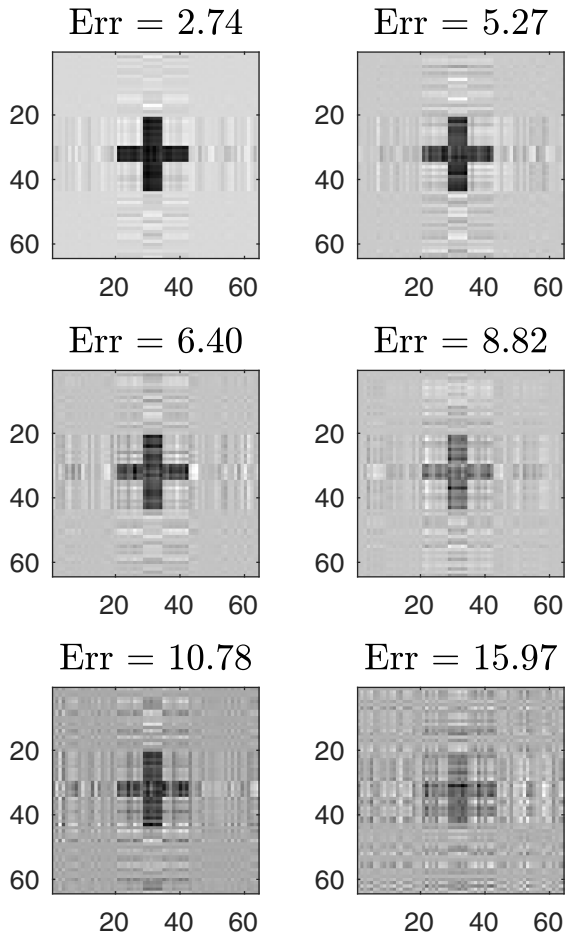

Fig. 2: Estimated signals $\hat{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{64\times 64}$ obtained by KTR-Hub (left panel) and KTR-LS (right panel) for $R = 2$. Smallest error (first row), median error (middle row) and largest error (bottom row) over 101 MC runs. The noise is ε-contaminated Gaussian ($\varepsilon = 0.1$, $\lambda = 20$), $N = 500$.

$c = 0.732$ in Huber's loss function throughout the studies. As can be noted, KTR-Hub has superior performance compared to KTR-LS. KTR-LS has large deviation in its performance, and its worst case result (Err $= 15.97$) is noisy with non-distinguishable structure while its best performance (Err $= 5.27$) is far from the best performance (Err $= 2.74$) of KTH-Hub. Figure 3 displays the obtained estimates that yield the median error over all 101 MC trials but now for ranks $R = 1$ and $R = 3$. As can be noted, both methods obtain unsatisfactory results, although KTR-Hub has slight advantage over KTR-LS. This example clearly illustrates that rank $R = 2$ is the most appropriate choice in this case. This does not come as a suprise since the cross image signal is perfectly recovered by rank-2 SVD. Thus choosing the rank optimally is important in the KTR model.

Figure 4 displays the results for Aalto logo image signal when the noise has a unit scale Cauchy distribution and rank $R = 4$ is used in the KTR model. Here we only show the signal image estimates corresponding to smallest error and median

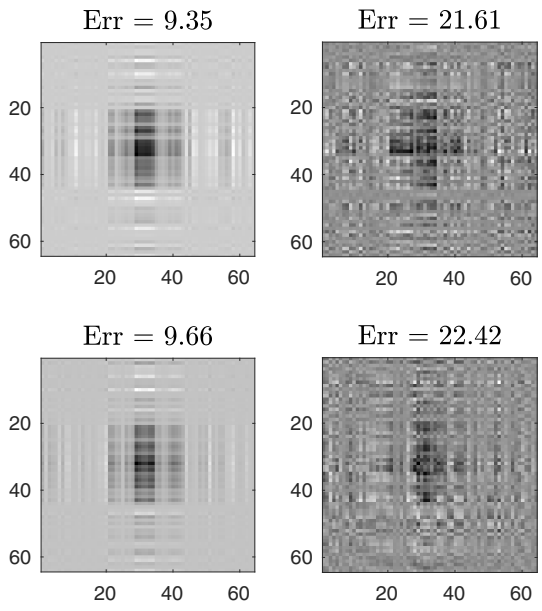

Fig. 3: Estimated signals $\hat{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{64\times 64}$ by KTR-Hub (left panel) and KTR-LS (right panel) regression for MC run that had median error over all 100 runs. Top row: $R = 1$, bottom row: $R = 3$. The noise is ε-contaminated Gaussian ($\varepsilon = 0.1$, $\lambda = 20$), $N = 500$.

error over 101 MC trials. As can be noted, such a spiky noise completely ruins the performance of KTR-LS method while the proposed KTR-Hub is able to reveal the underlying image signal. The shape of the logo is not well captured by KTR-LS even for its best run. The best performance obtained by KTR-Hub method is clear and on average (based on its median performance) the KTR-Hub is able to reveal the structure of the underlying signal. We also tested for rank $R = 3$ for which results were similar to $R = 4$ while $R = 5$ produced significantly worse results than $R = 4$.

Results for 3D-signal is reported in Figure 5 in the case that the noise follows ε-contaminated Gaussian ($\varepsilon = 0.1$, $\lambda = 20$) distribution and $N = 1000$. Again we show the signal image estimates corresponding to smallest error and median error over 101 MC trials. Here we used KTR-model of $R = 4$ which gave the best performance for both methods. Similar improvement in performance is observed for KTR-HUB over KTL-LS that was observed in Figure 2 with the same noise setting.

Table I compares the system running time (measured on a Macbook Pro laptop with a 2.3 GHz Intel Core i9) for computing the KTR-LS and KTR-Hub (using $N_{rep} = 10$ random initial trials) in all 3 studies. The reported times are averages over the 101 runs. It should be noted that both methods used similar settings (same initial guesses, N_{iter} and N_{rep} values, and convergence threshold $\epsilon = 5 \cdot 10^{-4}$). As can be noted, the KTR-Hub provides better running time in comparison to KTR-LS overall. This is because the alternating minimization algorithm converges faster in case of heavy-

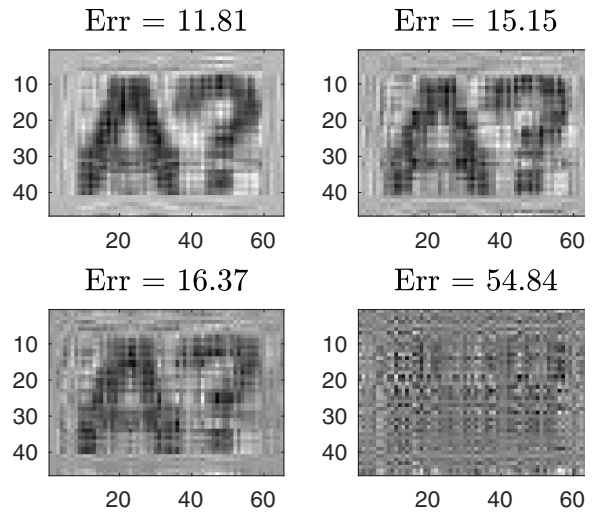

Fig. 4: Estimated signal $\hat{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{46\times 66}$ obtained by KTR-Hub (left panel) and KTR-LS (right panel) for Cauchy distributed ($\sigma = 1$) noise; $R = 4$, $N = 1000$, 101 MC runs. Top/bottom row: estimate for MC run with smallest/median error.

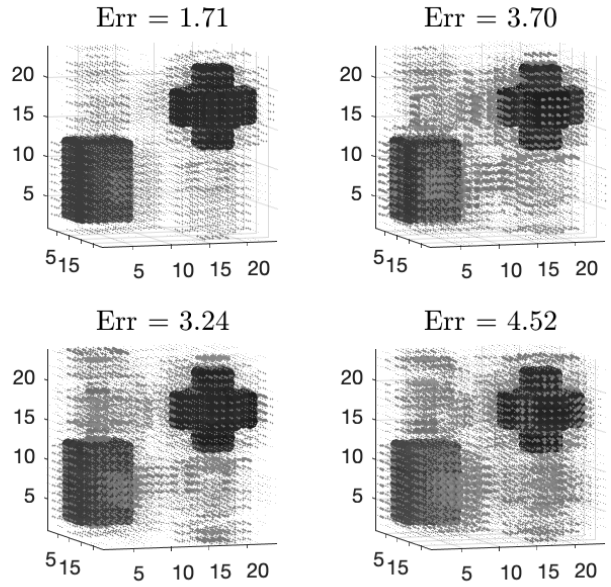

Fig. 5: Estimated signal $\hat{\boldsymbol{\mathcal{B}}} \in \mathbb{R}^{24\times 24\times 24}$ obtained by KTR-Hub (left panel) and KTR-LS (right panel) for rank $R = 4$. Top/bottom row: estimate corresponding to smallest/median error over all 101 MC trials. Noise is ε-contaminated Gaussian ($\varepsilon = 0.1$, $\lambda = 20$), $N = 1000$.

signal =	cross			logo			3D
$R =$	1	2	3	3	4	5	4
KTR-Hub	1.66	5.53	13.24	7.59	9.98	12.20	36.10
KTR-LS	1.48	7.58	17.10	6.91	11.78	19.55	48.54

TABLE I: System running times in seconds. Both methods used the same setting with same 10 initial random guesses

tailed errors and outliers.

V. Conclusions and Future Work

In this paper, we proposed robust estimation method based on Huber's criterion for linear Kruskal tensor regression (KTR) model [1]. The effectiveness of the method was tested in impulsive noise cases. Future research will focus on adaptive selection of rank R and adding sparsity enforcing penalties to the KTR-Hub criterion.

Applications for real-world sensing data is currently being investigated. For example, acquired HS images of an agricultural region can be used as 3D-image covariates in applications such as predicting the crop type or crop quality. Robust estimation is an important design criterion since HS images often contain outliers or missing data due to weather conditions such as clouds [14]. Tensor regression and classification methods have also been used in the analysis of high-frequency trading data [15] which is another application where the proposed method can be useful due to non-Gaussianity of the data.

References

[1] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013.

[2] X. Li, D. Xu, H. Zhou, and L. Li, "Tucker tensor regression and neuroimaging analysis," *Statistics in Biosciences*, vol. 10, no. 3, pp. 520–545, 2018.

[3] J. Poythress, J. Ahn, and C. Park, "Low-rank, orthogonally decomposable tensor regression with application to visual stimulus decoding of fMRI data," *Journal of Computational and Graphical Statistics*, pp. 1–14, 2021.

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

[5] 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 Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[6] 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, 1970.

[7] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis," *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.

[8] R. Guhaniyogi, S. Qamar, and D. B. Dunson, "Bayesian tensor regression," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2733–2763, 2017.

[9] H.-J. Kim, E. Ollila, V. Koivunen, and C. Croux, "Robust and sparse estimation of tensor decompositions," in *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP'13)*, Austin, USA, Dec 3–5 2013 2013, pp. 965–968.

[10] H.-J. Kim, E. Ollila, V. Koivunen, and H. V. Poor, "Robust iteratively reweighted lasso estimation for sparse tensor factorizations," in *IEEE Workshop on Statistical Signal Processing (SSP'14)*, Cold Coast, Australia, June 29-July 2 2014, pp. 452–455.

[11] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[12] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust statistics for signal processing*. Cambridge University Press, 2018.

[13] E. Ollila and A. Mian, "Block-wise minimization-majorization algorithm for huber's criterion: Sparse learning and applications," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2020, pp. 1–6.

[14] M. A. Veganzones, J. E. Cohen, R. C. Farias, J. Chanussot, and P. Comon, "Nonnegative tensor cp decomposition of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2577–2588, 2015.

[15] D. T. Tran, M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Tensor representation in high-frequency financial data for price change prediction," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–7.