ROBUST LOW-RANK TENSOR MODELLING USING TUCKER AND CP DECOMPOSITION

Niannan Xue[*], *George Papamakarios*[†], *Mehdi Bahri*[*], *Yannis Panagakis*[*‡], *Stefanos Zafeiriou*[*⋆]

[*] Imperial College London, UK [†] University of Edinburgh, UK [‡] Middlesex University, UK

ABSTRACT

A framework for reliable seperation of a low-rank subspace from grossly corrupted multi-dimensional signals is pivotal in modern signal processing applications. Current methods fall short of this separation either due to the radical simplification or the drastic transformation of data. This has motivated us to propose two new robust low-rank tensor models: Tensor Orthonormal Robust PCA (TORCPA) and Tensor Robust CP Decomposition (TRCPD). They seek Tucker and CP decomposition of a tensor respectively with l_p norm regularisation. We compare our methods with state-of-the-art low-rank models on both synthetic and real-world data. Experimental results indicate that the proposed methods are faster and more accurate than the methods they compared to.

Index Terms— Tensor Decomposition, Robust Principal Component Analysis, Tucker, CANDECOMP/PARAFAC

1. INTRODUCTION

In many real-world applications, input data are naturally representd by tensors (i.e., multi-dimensional arrays). Traditionally, such data would require vectorising before processing and thus destroy the inherent higher-order interactions. As a result, novel models must be developed to preserve the multilinear structure when extracting the hidden and evolving trends in such data. Typical tensor data are video clips, color images, multi-channel EEG records, etc.

In practice, important information usually lie in a (multilinear) low-dimensional space whose dimensionality is much lower dimensional space than observations.This is the essence of low-rank modelling. In this paper, we focus on the problem of recovering a low-dimensional multilinear structure from tensor data corrupted by gross corruptions.

Given a tensor $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, its tensor rank [1] is defined by the smallest r such that $\boldsymbol{\mathcal{L}} = \sum_{i=1}^{r} \boldsymbol{a}_i^{(1)} \circ \cdots \circ \boldsymbol{a}_i^{(N)}$, where $\circ$ denotes outer products among some Nr vectors $\boldsymbol{a}_i^{(1)}, \cdots, \boldsymbol{a}_i^{(N)}, 1 \leq i \leq r$. As such, robust low-rank tensor modelling seeks a decomposition $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{S}}$ for an N^{th}-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, where $\boldsymbol{\mathcal{L}}$ has a low tensor rank and $\boldsymbol{\mathcal{S}}$ is sparse. However, the tensor rank is usually intractable [2]. A common adjustment [3–5] is to use a convex combination of the n-ranks of $\boldsymbol{\mathcal{L}}$, that is $\gamma = \sum_{i=1}^{N} \alpha_i \text{rank}_i(\boldsymbol{\mathcal{L}})$, where $\alpha_i \geq 0$, $\sum_{i=1}^{N} \alpha_i = 1$ and $\text{rank}_i(\boldsymbol{\mathcal{L}})$ is the the column rank of the mode-i matricisation [6] of $\boldsymbol{\mathcal{L}}$. It is, therefore, natural to obtain the decomposition by solving optimisation problem (1)

$$\min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{S}}} \gamma + \lambda \|\boldsymbol{\mathcal{E}}\|_0 \quad \text{s.t. } \gamma = \sum_{i=1}^{N} \alpha_i \text{rank}_i(\boldsymbol{\mathcal{L}}),\ \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{S}}, \quad (1)$$

where $\|\boldsymbol{\mathcal{S}}\|_0$ is the l_0 norm of the vectorisation of $\boldsymbol{\mathcal{S}}$ and λ is a weighting parameter.

Here we present two novel robust tensor methods based on Tucker and CP decomposition that recover the latent low-rank component from noisy observations by relaxing (1), which is NP-hard. In section 2, we review relevant literature on matrix and tensor algorithms. In section 3, we explain our proposed tensor methods in detail. In section 4, we demonstrate the advantages of our models on both synthetic data and a real-world dataset. Finally, in section 5, we summarise our new models and point out possible future improvements.

2. RELATED WORK

RSTD [7] is a direct multi-linear extension of matrix principal component pursuit (PCP) [8]. It approximates (1) by replacing $\text{rank}_i(\boldsymbol{\mathcal{L}})$ and $\|\boldsymbol{\mathcal{S}}\|_0$ with convex surrogates $\|\boldsymbol{\mathcal{L}}_{(i)}\|_*$ and $\|\boldsymbol{\mathcal{S}}\|_1$ respectively, where $\|\boldsymbol{\mathcal{L}}_{(i)}\|_*$ is the nuclear norm of the mode-i matricisation of $\boldsymbol{\mathcal{L}}$ and $\|\boldsymbol{\mathcal{S}}\|_1$ is the l_1 norm of the vectorisation of $\boldsymbol{\mathcal{S}}$. As a result, it solves the following alternative objective

$$\min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{S}}} \sum_{i=1}^{N} \alpha_i \|\boldsymbol{\mathcal{L}}_{(i)}\|_* + \lambda \|\boldsymbol{\mathcal{S}}\|_1 \quad \text{s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{S}}. \quad (2)$$

An ALM solver can be found in [9]. It is also worth noting that under certain conditions RSTD is guaranteed to exactly recover the low-rank component [10].

Much recent research on subspace analysis for the matrix case has direct applicability to tensor data. The costly singular value decomposition step in classical PCP prohibits large-scale analysis. A general approach to mitigate this issue is to look for a factorisation of the low rank component $\boldsymbol{A}$. OR-PCA [11] uses a linear combination of the active subspace, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V},\ \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$, where bilinear factors $\boldsymbol{U} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{V} \in \mathbb{R}^{k \times n}$ are the principal components and the com-

⋆ Corresponding Author (s.zafeiriou@imperial.ac.uk). The work of Stefanos Zafeiriou was partially funded by EPSRC project EP/N007743/1 (FACER2VM).

ISBN 978-0-9928626-7-1 © EURASIP 2017

bination coefficients respectively and k is an upper bound of rank($\boldsymbol{A}$).

3. ROBUST LOW-RANK TENSOR MODELLING

3.1. Notation

We first introduce some notations used throughout the paper. Lowercase latin and greek letters denote scalars, e.g. r, γ. Bold lowercase latin letters denote vectors, e.g. $\boldsymbol{a}$. Bold uppercase latin letters denote matrices, e.g. $\boldsymbol{A}$. Bold uppercase calligraphic latin letters denote tensors, e.g. $\boldsymbol{\mathcal{L}}$. Bold uppercase greek letters denote operators on tensors and matrices, e.g. $\boldsymbol{\Theta}(\boldsymbol{\mathcal{S}})$, $\boldsymbol{\Phi}(\boldsymbol{U})$. $\langle \boldsymbol{A}, \boldsymbol{B}\rangle$ represents tr($\boldsymbol{A}^T\boldsymbol{B}$). $\|\boldsymbol{A}\|_F$ is the Frobenius norm. $\boldsymbol{A} \odot \boldsymbol{B}$ denotes the Khatri-Rao product between matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and $\boldsymbol{\mathcal{X}} \times_i \boldsymbol{U}$ is the i-mode product [6].

3.2. Soft and hard thresholding operators

For fixed $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, the optimal analytical solution for $\min_{\boldsymbol{\mathcal{Y}}} \ \kappa\|\boldsymbol{\mathcal{Y}}\|_1 + \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\|_F^2$ is given by the soft thresholding operation $\boldsymbol{\Theta}_\kappa(\boldsymbol{\mathcal{X}})$, where

$$\boldsymbol{\Theta}_\kappa(\boldsymbol{\mathcal{X}})_{\iota_1 \cdots \iota_N} = (\boldsymbol{\mathcal{X}}_{\iota_1 \cdots \iota_N} - \kappa)_+ - (-\boldsymbol{\mathcal{X}}_{\iota_1 \cdots \iota_N} \kappa)_+. \quad (3)$$

And for fixed $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, the optimal analytical solution for $\min_{\boldsymbol{Y}} \ \kappa\|\boldsymbol{X}\|_* + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|_F^2$ is given by the hard thresholding operation $\boldsymbol{\Phi}_\kappa(\boldsymbol{X})$, where

$$\boldsymbol{\Phi}_\kappa(\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{\Theta}_\kappa(\boldsymbol{S})\boldsymbol{V}^T, \quad (4)$$

for singular value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$.

3.3. Tensor Orthonormal Robust PCA

Generalisation of ORPCA to tensors corresponds to the following factorisation of the low-rank component $\boldsymbol{\mathcal{L}}$:

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{V}} \times_1 \boldsymbol{U}_1 \times \cdots \times_N \boldsymbol{U}_N \equiv \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i, \ \boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}, \quad (5)$$

which is exactly the HOSVD [12] of $\boldsymbol{\mathcal{L}}$ and the following relationship holds

$$\|\boldsymbol{\mathcal{L}}_{(i)}\|_* = \|\boldsymbol{\mathcal{V}}_{(i)}\|_*. \quad (6)$$

Based on the above, (2) can be re-written as

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}} \sum_{i=1}^N \alpha_i \|\boldsymbol{\mathcal{V}}_{(i)}\|_* + \lambda\|\boldsymbol{\mathcal{S}}\|_1, \\ &\text{s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i + \boldsymbol{\mathcal{S}}, \quad \boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}, \quad 1 \le i \le N. \end{aligned} \quad (7)$$

To separate variables, we make the substitution $\boldsymbol{\mathcal{V}}_{(i)} = \boldsymbol{J}_i$, to arrive at an equivalent problem:

$$\begin{aligned} &\min_{\boldsymbol{J}_i, \boldsymbol{\mathcal{S}}} \sum_{i=1}^N \alpha_i \|\boldsymbol{J}_i\|_* + \lambda\|\boldsymbol{\mathcal{S}}\|_1, \\ &\text{s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i + \boldsymbol{\mathcal{S}}, \quad \boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}, \\ &\quad \boldsymbol{\mathcal{V}}_{(i)} = \boldsymbol{J}_i, \quad 1 \le i \le N. \end{aligned} \quad (8)$$

To apply ADMM, the augmented Lagrangian of (8) is constructed first:

$$\begin{aligned} l(\boldsymbol{J}_i, \boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}, \boldsymbol{U}_i, \boldsymbol{\mathcal{Y}}, \boldsymbol{Z}_i) &= \sum_{i=1}^N \alpha_i \|\boldsymbol{J}_i\|_* + \lambda\|\boldsymbol{\mathcal{S}}\|_1 \\ &+ \langle \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i - \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Y}}\rangle + \sum_{i=1}^N \langle \boldsymbol{\mathcal{V}}_{(i)} - \boldsymbol{J}_i, \boldsymbol{Z}_i\rangle \\ &+ \frac{\mu}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i - \boldsymbol{\mathcal{S}}\|_F^2 + \sum_{i=1}^N \frac{\mu}{2}\|\boldsymbol{\mathcal{V}}_{(i)} - \boldsymbol{J}_i\|_F^2, \end{aligned} \quad (9)$$

where $\boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}$ has not been incorporated.

$\boldsymbol{J}_i$ is updated by the minimiser of $l(\boldsymbol{J}_i)$:

$$\begin{aligned} \boldsymbol{J}_i &= \arg\min_{\boldsymbol{J}_i} \alpha_i\mu^{-1}\|\boldsymbol{J}_i\|_* + \frac{1}{2}\|\boldsymbol{J}_i - (\boldsymbol{\mathcal{V}}_{(i)} + \frac{1}{\mu}\boldsymbol{Z}_i)\|_F^2 \\ &= \boldsymbol{\Phi}_{\alpha_i\mu^{-1}}(\boldsymbol{\mathcal{V}}_{(i)} + \frac{1}{\mu}\boldsymbol{Z}_i) \end{aligned} \quad (10)$$

$\boldsymbol{\mathcal{V}}$ is updated by the minimiser of $l(\boldsymbol{\mathcal{V}})$:

$$\begin{aligned} \boldsymbol{\mathcal{V}} &= \arg\min_{\boldsymbol{\mathcal{V}}} \langle \boldsymbol{\mathcal{V}}, -(\mu(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}}) + \boldsymbol{\mathcal{Y}}) \times_{i=1}^N \boldsymbol{U}_i^T\rangle \\ &+ \sum_{i=1}^N \langle \boldsymbol{\mathcal{V}} - \boldsymbol{\mathcal{J}}_i, \boldsymbol{\mathcal{Z}}_i\rangle + \frac{\mu}{2}\|\boldsymbol{\mathcal{V}}\|_F^2 + \sum_{i=1}^N \frac{\mu}{2}\|\boldsymbol{\mathcal{V}} - \boldsymbol{\mathcal{J}}_i\|_F^2, \end{aligned} \quad (11)$$

where we have used the fact that $\boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}$, the Frobenius norm is invariant under rotations and $\boldsymbol{\mathcal{J}}_i, \boldsymbol{\mathcal{Z}}_i$ are the inverse of mode-i matricisations, $\boldsymbol{J}_i, \boldsymbol{Z}_i$ respectively. To obtain $\boldsymbol{\mathcal{V}}$, setting the gradient of (11) to zero gives:

$$\boldsymbol{\mathcal{V}} = \frac{1}{N+1}((\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}) \times_{i=1}^N \boldsymbol{U}_i^T + \sum_{i=1}^N (\boldsymbol{\mathcal{J}}_i - \frac{1}{\mu}\boldsymbol{\mathcal{Z}}_i)). \quad (12)$$

$\boldsymbol{\mathcal{S}}$ is updated by the minimiser of $l(\boldsymbol{\mathcal{S}})$:

$$\begin{aligned} \boldsymbol{\mathcal{S}} =& \arg\min_{\boldsymbol{\mathcal{S}}} \lambda\mu^{-1}\|\boldsymbol{\mathcal{S}}\|_1 \\ &+ \frac{1}{2}\|\boldsymbol{\mathcal{S}} - (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{\mathcal{Y}})\|_F^2 \\ =& \boldsymbol{\Theta}_{\lambda\mu^{-1}}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^N \boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}). \end{aligned} \quad (13)$$

$\boldsymbol{U}_i$ is updated by the minimiser of $l(\boldsymbol{U}_i)$ subject to $\boldsymbol{U}_i^T\boldsymbol{U}_i = \boldsymbol{I}$:

$$\begin{aligned} \boldsymbol{U}_i =& \arg\min_{\boldsymbol{U}_i} \frac{1}{2}\|\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{\mathcal{S}}_{(i)} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}_{(i)} - \boldsymbol{U}_i\boldsymbol{B}_i\|_F^2, \\ &\text{where } \boldsymbol{B}_i = (\boldsymbol{\mathcal{V}} \times_{j=1}^{i-1} \boldsymbol{U}_j \times_{j=i+1}^N \boldsymbol{U}_j)_{(i)}. \end{aligned} \quad (14)$$

If we have the following SVD

$$(\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{\mathcal{S}}_{(i)} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}_{(i)})\boldsymbol{B}_i^T = \boldsymbol{C}_i\boldsymbol{D}_i\boldsymbol{V}_i^T, \quad (15)$$

then according to the *Reduced Rank Procrustes Theorem* [13], the solution is given by

$$\boldsymbol{U}_i = \boldsymbol{C}_i\boldsymbol{V}_i^T. \quad (16)$$

The complete algorithm is presented in Algorithm 1.

ISBN 978-0-9928626-7-1 © EURASIP 2017

Algorithm 1 ADMM solver for TORPCA

Input: Observation $\boldsymbol{\mathcal{X}}$, parameter $\lambda > 0$, scaling $\kappa > 1$, weights α_i, ranks k_i

Initialise: $\boldsymbol{J}_i = \boldsymbol{Z}_i = \boldsymbol{0}$, $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{Y}} = \boldsymbol{0}$, $\boldsymbol{\mathcal{V}} = \boldsymbol{0}$, $\boldsymbol{U}_i =$ first k_i left singular vectors of $\boldsymbol{\mathcal{X}}_{(i)}$, $\mu > 0$
while not converged **do**
  for $i \in \{1, 2, \cdots, N\}$ **do**
    $\boldsymbol{J}_i = \boldsymbol{\Phi}_{\alpha_i \mu^{-1}}(\boldsymbol{\mathcal{V}}_{(i)} + \frac{1}{\mu}\boldsymbol{Z}_i)$
  end for
  $\boldsymbol{\mathcal{V}} = \frac{1}{N+1}((\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{S}} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}) \times_{i=1}^{N} \boldsymbol{U}_i^T + \sum_{i=1}^{N}(\boldsymbol{\mathcal{J}}_i - \frac{1}{\mu}\boldsymbol{\mathcal{Z}}_i))$
  $\boldsymbol{\mathcal{S}} = \boldsymbol{\Theta}_{\lambda\mu^{-1}}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^{N} \boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{\mathcal{Y}})$
  for $i \in \{1, 2, \cdots, N\}$ **do**
    $\boldsymbol{B}_i = (\boldsymbol{\mathcal{V}} \times_{j=1}^{i-1} \boldsymbol{U}_j \times_{j=i+1}^{N} \boldsymbol{U}_j)_{(i)}$
    $\boldsymbol{C}_i \boldsymbol{D}_i \boldsymbol{V}_i^T = (\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{\mathcal{S}}_{(i)} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}_{(i)})\boldsymbol{B}_i^T$
    $\boldsymbol{U}_i = \boldsymbol{C}_i \boldsymbol{V}_i^T$
  end for
  $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}} + \mu(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{V}} \times_{i=1}^{N} \boldsymbol{U}_i - \boldsymbol{\mathcal{S}})$
  for $i \in \{1, 2, \cdots, N\}$ **do**
    $\boldsymbol{Z}_i = \boldsymbol{Z}_i + \mu(\boldsymbol{\mathcal{V}}_{(i)} - \boldsymbol{J}_i)$
  end for
  $\mu = \mu \times \kappa$
end while

Return: $\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{S}}, \boldsymbol{U}_i$

3.4. Tensor robust CP decomposition

Let $\boldsymbol{U}^{(i)} = [\boldsymbol{a}_1^{(i)}, \boldsymbol{a}_2^{(i)}, \cdots, \boldsymbol{a}_r^{(i)}]$, then we can express $\boldsymbol{\mathcal{L}}$ compactly as $\boldsymbol{\mathcal{L}} = \boldsymbol{U}^{(1)} \circ \boldsymbol{U}^{(2)} \circ \cdots \circ \boldsymbol{U}^{(N)}$. In particular, it can be shown that $\text{rank}_i(\boldsymbol{\mathcal{L}}) \leq \text{rank}(\boldsymbol{U}_i)$, for $1 \leq i \leq N$. So, it is beneficial to solve the following objective

$$\min_{\boldsymbol{U}_i, \boldsymbol{\mathcal{S}}} \sum_{i=1}^{N} \alpha_i \|\boldsymbol{U}_i\|_* + \lambda \|\boldsymbol{\mathcal{S}}\|_1, \quad \boldsymbol{\mathcal{X}} = \boldsymbol{U}_1 \circ \boldsymbol{U}_2 \circ \cdots \circ \boldsymbol{U}_N + \boldsymbol{\mathcal{S}}. \tag{17}$$

Again, we make the substitution $\boldsymbol{U}_i = \boldsymbol{J}_i$ before performing ADMM, which leads to the following problem

$$\begin{aligned} &\min_{\boldsymbol{J}_i, \boldsymbol{\mathcal{S}}} \sum_{i=1}^{N} \alpha_i \|\boldsymbol{J}_i\|_* + \lambda \|\boldsymbol{S}\|_1 \\ &\text{s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N + \boldsymbol{\mathcal{S}}, \;\; \boldsymbol{U}_i = \boldsymbol{J}_i, \;\; 1 \leq i \leq N. \end{aligned} \tag{18}$$

The corresponding augmented Lagrangian is

$$\begin{aligned} l(\boldsymbol{J}_i, \boldsymbol{U}_i, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{Z}_i) &= \sum_{i=1}^{N} \alpha_i \|\boldsymbol{J}_i\|_* + \lambda \|\boldsymbol{\mathcal{S}}\|_1 \\ &+ \langle \boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N - \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{Y}} \rangle + \sum_{i=1}^{N} \langle \boldsymbol{U}_i - \boldsymbol{J}_i, \boldsymbol{Z}_i \rangle \\ &+ \frac{\mu}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N - \boldsymbol{\mathcal{S}}\|_F^2 + \sum_{i=1}^{N} \frac{\mu}{2} \|\boldsymbol{U}_i - \boldsymbol{J}_i\|_F^2. \end{aligned} \tag{19}$$

Algorithm 2 ADMM solver for TRCPD

Input: Observation $\boldsymbol{\mathcal{X}}$, parameter $\lambda > 0$, scaling $\kappa > 1$, weights α_i, rank k

Initialise: $\boldsymbol{J}_i = \boldsymbol{U}_i =$ rand, $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{Y}} = \boldsymbol{0}$, $\boldsymbol{Z}_i = 0$, $\mu > 0$
while not converged **do**
  for $i \in \{1, 2, \cdots, N\}$ **do**
    $\boldsymbol{J}_i = \boldsymbol{\Phi}_{\alpha_i \mu^{-1}}(\boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{Z}_i)$
    $\tilde{\boldsymbol{U}}_i = (\boldsymbol{U}_N \odot \cdots \odot \boldsymbol{U}_{i+1} \odot \boldsymbol{U}_{i-1} \odot \cdots \odot \boldsymbol{U}_1)^T$
    $\boldsymbol{U}_i = ((\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{\mathcal{S}}_{(i)} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}_{(i)})\tilde{\boldsymbol{U}}_i^T + \boldsymbol{J}_i - \frac{1}{\mu}\boldsymbol{Z}_i)(\tilde{\boldsymbol{U}}_i \tilde{\boldsymbol{U}}_i^T + \boldsymbol{I})^{-1}$
  end for
  $\boldsymbol{\mathcal{S}} = \boldsymbol{\Theta}_{\lambda\mu^{-1}}(\boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N + \frac{1}{\mu}\boldsymbol{\mathcal{Y}})$
  $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}} + \mu(\boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N - \boldsymbol{\mathcal{S}})$
  for $i \in \{1, 2, \cdots, N\}$ **do**
    $\boldsymbol{Z}_i = \boldsymbol{Z}_i + \mu(\boldsymbol{U}_i - \boldsymbol{J}_i)$
  end for
  $\mu = \mu \times \kappa$
end while

Return: $\boldsymbol{U}_i, \boldsymbol{\mathcal{S}}$

$\boldsymbol{J}_i$ is updated by the minimiser of $l(\boldsymbol{J}_i)$:

$$\begin{aligned} \boldsymbol{J}_i &= \arg\min_{\boldsymbol{J}_i} \alpha_i \mu^{-1} \|\boldsymbol{J}_i\|_* + \frac{1}{2}\|\boldsymbol{J}_i - (\boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{Z}_i)\|_F^2 \\ &= \boldsymbol{\Phi}_{\alpha_i \mu^{-1}}(\boldsymbol{U}_i + \frac{1}{\mu}\boldsymbol{Z}_i). \end{aligned} \tag{20}$$

$\boldsymbol{U}_i$ is updated by the minimiser of $l(\boldsymbol{U}_i)$:

$$\begin{aligned} \boldsymbol{U}_i = \arg\min_{\boldsymbol{U}_i} &\langle \boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{U}_i \tilde{\boldsymbol{U}}_i - \boldsymbol{\mathcal{S}}_{(i)}, \boldsymbol{\mathcal{Y}}_{(i)} \rangle + \langle \boldsymbol{U}_i - \boldsymbol{J}_i, \boldsymbol{Z}_i \rangle \\ &+ \frac{\mu}{2}\|\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{U}_i \tilde{\boldsymbol{U}}_i - \boldsymbol{\mathcal{S}}_{(i)}\|_F^2 + \frac{\mu}{2}\|\boldsymbol{U}_i - \boldsymbol{J}_i\|_F^2, \end{aligned} \tag{21}$$

where $\tilde{\boldsymbol{U}}_i = (\boldsymbol{U}_N \odot \cdots \odot \boldsymbol{U}_{i+1} \odot \boldsymbol{U}_{i-1} \odot \cdots \odot \boldsymbol{U}_1)^T$

Setting the derivative of (21) to zero gives:

$$\begin{aligned} \boldsymbol{U}_i = &((\boldsymbol{\mathcal{X}}_{(i)} - \boldsymbol{\mathcal{S}}_{(i)} + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}_{(i)})\tilde{\boldsymbol{U}}_i^T \\ &+ \boldsymbol{J}_i - \frac{1}{\mu}\boldsymbol{Z}_i)(\tilde{\boldsymbol{U}}_i \tilde{\boldsymbol{U}}_i^T + \boldsymbol{I})^{-1}. \end{aligned} \tag{22}$$

$\boldsymbol{\mathcal{S}}$ is updated by the minimiser of $l(\boldsymbol{\mathcal{S}})$:

$$\begin{aligned} \boldsymbol{\mathcal{S}} &= \arg\min_{\boldsymbol{\mathcal{S}}} \lambda\mu^{-1}\|\boldsymbol{\mathcal{S}}\|_1 \\ &+ \frac{1}{2}\|\boldsymbol{\mathcal{S}} - (\boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N) + \frac{1}{\mu}\boldsymbol{\mathcal{Y}}\|_F^2 \\ &= \boldsymbol{\Theta}_{\lambda\mu^{-1}}(\boldsymbol{\mathcal{X}} - \boldsymbol{U}_1 \circ \cdots \circ \boldsymbol{U}_N + \mu^{-1}\boldsymbol{\mathcal{Y}}). \end{aligned} \tag{23}$$

The complete algorithm is described in Algorithm 2.

3.5. Complexity and convergence

For ease of exposition, we assume that $d_1, \cdots, d_N = \zeta$. For TORPCA, the most expansive calculation in each iteration is the i-mode product which has a time complexity of $O(Nr\zeta^N)$. For TRCPD, the dominant term is the chain of matrix outer products which costs $O(Nr\zeta^N)$. Note that both

ISBN 978-0-9928626-7-1 © EURASIP 2017

methods have lower complexity than RSTD whose complexity is $O(N\zeta^{N+1})$ due to SVD if $r < \zeta$.

Although both of our proposed tensor methods are non-convex, we have empirically found that the warm initialisation of using the first k_i left singular vectors of $\boldsymbol{\mathcal{X}}_{(i)}$ for $\boldsymbol{U}_i$ works well for TORPCA and uniform initialisation of $\boldsymbol{U}_i, 1 \leq i \leq k$ from $[0, 1]$ suffices for TRCPD (see Section 4).

4. EXPERIMENTAL RESULTS

4.1. Implementation details

For stopping criteria, we use one of the KKT opimality conditions, $\frac{\|\boldsymbol{\mathcal{X}}-\boldsymbol{\mathcal{L}}-\boldsymbol{\mathcal{S}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F} < \delta$ and we have set $\delta = 10^{-7}$. The initial value of μ is set to 10^{-3}, which is geometrically increased by a factor of $\kappa = 1.2$ up to 10^9. The weights α_i are assumed equal.

4.2. Simulation

We first evaluate the performance of all algorithms on synthetic data. A low-rank tensor $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{100\times100\times100}$ is generated via $\boldsymbol{\mathcal{L}} = \boldsymbol{U}_1 \circ \boldsymbol{U}_2 \circ \boldsymbol{U}_3$, where elements of $\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3 \in \mathbb{R}^{100\times8}$ are independently sampled from the standard Gaussian distribution. The variance of $\boldsymbol{\mathcal{L}}$ is normalised to 1 afterwards. A sparse tensor $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{100\times100\times100}$ is constructed by uniform sampling from $[-10, 10]$. Then only 20% of the elements are kept, with others set to zero.

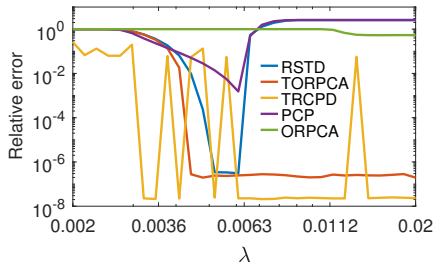

Fig. 1. Relative error from all algorithms for a range of λ.

Each tensor algorithm takes $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{S}}$ as input, whereas matrix algorithms take mode-1 matricisation of $\boldsymbol{\mathcal{X}}$ as input. Since $\text{rank}(\boldsymbol{\mathcal{L}}) \leq 8$, the rank k in TRCPD is set to 8 and the ranks k_i in TORPCA are all set to 8 because $\text{rank}_i(\boldsymbol{\mathcal{L}}) \leq 8$. The relative error $\frac{\|\boldsymbol{\mathcal{L}}-\tilde{\boldsymbol{\mathcal{L}}}\|_F}{\|\boldsymbol{L}\|_F}$ averaged over 5 trials against λ is plotted for the optimal $\tilde{\boldsymbol{\mathcal{L}}}$ in each algorithm in Fig 1. The total execution time for each algorithm versus λ is shown in Fig 2.

It is clear that tensor methods are superior to matrix-based methods. Particularly, TRCPD performs the best and TORPCA is also better than RSTD. Both TRCPD and TORPCA are stable in terms of λ whereas RSTD depends on tuning heavily. The execution time confirms our complexity analysis. Both of TORCPA and TRCPD are significantly faster than RSTD.

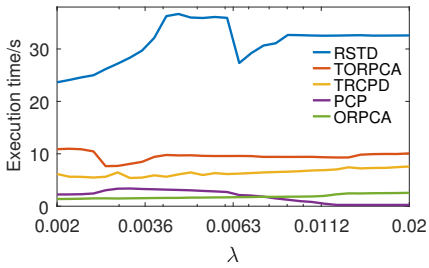

Fig. 2. Running time of all algorithms as λ varies.

Senario	Algorithm	$\lambda \in [10^{-4}, 10^{-1}]$	$k \in \{10, 20, \cdots, 200\}$	$\alpha \in \{0.1, 0.2, \cdots, 0.9\}$
Salt & Pepper	RSTD	0.0092	—	—
	TORPCA	0.2000	—	0.2
	TRCPD	0.0134	160	—
	ORPCA	0.0621	20	—
occlusion	RSTD	0.0076	—	—
	TORPCA	0.0190	—	0.2
	TRCPD	0.0017	50	—
	ORPCA	0.0300	40	—

Table 1. Optimal parameter choices for all algorithms used in different experiments.

4.3. Facial image denoising

It is well understood that a convex Lambertian surface, viz. faces, under distant and isotropic lighting has a low-rank underlying model. In light of this, we consider images of a fixed pose under different illuminations from the extended Yale B database for benchmarking. All 64 images for one person were studied. For matrix-based methods, 32556×64 observation matrices were formed by vectorising each 168×192 image. All images are also re-scaled such that every pixel lies in $[0, 1]$.

• **Salt & Pepper Noise** Salt & pepper noise is observed in real images, commonly caused by data transmission errors. To apply salt & pepper noise, we randomly set pixels to black (0) or white (1) with equal probability. This is close to the Laplacian noise hypothesis, where noise is heavy, non-Gaussian and potentially wide-ranging. We test an extreme case, where 60% of all the pixels are affected.

• **Partial Occlusion** Partial occlusion is ubiquitous in visual information, which can usually be completed during human visual perception [14]. For the partial occlusion noise, we generate randomly sized patches at random locaions. The maximum dimension is 160 pixels and the occlusion is full of Salt & Pepper noise.

The successful application of various algorithms requires careful tuning of the algorithmic parameters. These include the penalty parameter λ, an estimate of $k = \text{rank}(\boldsymbol{\mathcal{L}})$ and $k_i = \text{rank}_i(\boldsymbol{\mathcal{L}}) = \lceil d_i \times \alpha \rceil$. The ranges of interest and the optimal choices are summarised in Table 1.

Reconstruction from salt & pepper noise is illustrated in the first row of Fig 3, where the first image in the sequence

ISBN 978-0-9928626-7-1 © EURASIP 2017

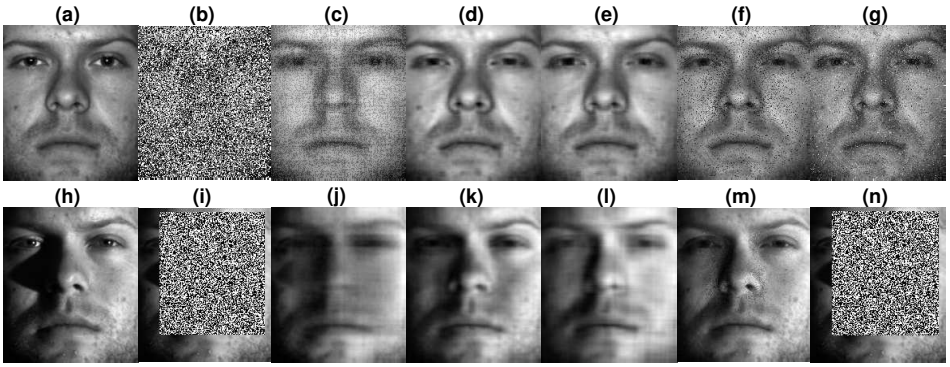

Fig. 3. Image Denoising Experiments: (**a**) & (**h**) are original images from the sequence. Salt & pepper is introduced as shown in (**b**) and occlusion is demonstrated in (**i**). (**c**) & (**j**) present recovery results for RSTD. (**d**) & (**k**) for TORPCA. (**e**) & (**l**) for TRCPD. (**f**) & (**m**) for PCP. And (**g**) & (**n**) for ORCPA.

is shown. RSTD and matrix-based methods fail to remove the introduced noise, whereas TORPCA and TRCPD are extremely promising such that no trail of noise can be seen. Recovery from partial occlusion is displayed in the second row of Fig 3. ORPCA has little effect. The region where noise was introduced is severely distorted in the recovered image of RSTD. Both TORPCA and TRCPD mananged to denoise the occlusion though they have an additional smoothing effect. PCP achieves the highest quality of recovery but there is still unremoved noise left in the image. This may be attributed to the fact that the nature of the occlusion is inherently in a matrix form.

5. CONCLUSIONS

In this paper, two novel methods, namely the TORPCA and TRCPD have been proposed in order to address the problem of robust low-rank tensor recovery. The proposed methods surpass existing tensor- and matrix-based methods in our experiments. We anticipate our work to lay foundations for more general signal processing problems. Future direction of research can extend our methods to hierarchical tensor decomposition.

REFERENCES

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

[2] J. Håstad, "Tensor rank is np-complete," *Journal of Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[3] S. Gandyand B. Recht and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, pp. 025010, 2011.

[4] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208 – 220, 2013.

[5] L. Yang, Z. Huang, and X. Shi, "A fixed point iterative method for low n-rank tensor pursuit," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2952 – 2962, 2013.

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

[7] Y. Li, J. Yan, Y. Zhou, and J. Yang, "Optimum subspace learning and error correction for tensors," in *European Conference on Computer Vision*, 2010.

[8] E.J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, 2011.

[9] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, 2014.

[10] B. Huang, C. Mu, J. Wright, and D. Goldfarb, "Provable models for robust low-rank tensor completion," *Pacific Journal of Optimisation*, vol. 11, no. 2, pp. 339–364, 2015.

[11] G. Liu and S. Yan, "Active subspace: Toward scalable low-rank learning," *Neural Computation*, vol. 24, no. 12, pp. 3371–3394, 2012.

[12] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, 2000.

[13] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, 2006.

[14] A.B. Sekuler and S.E. Palmer, "Perception of partly occluded objects: A microgenetic analysis," *Journal of Experimental Psychology: General*, vol. 121, no. 1, 1992.

ISBN 978-0-9928626-7-1