LOW RANK TENSOR DECONVOLUTION

Anh-Huy Phan[‡], *Petr Tichavský*[•*], *Andrzej Cichocki*[‡†]

[‡]Brain Science Institute, RIKEN, Wakoshi, Japan
[†]Systems Research Institute PAS, Warsaw, Poland
[•]Institute of Information Theory and Automation, Prague, Czech Republic

ABSTRACT

In this paper, we propose a low-rank tensor deconvolution problem which seeks multiway replicative patterns and corresponding activating tensors of rank-1. An alternating least squares (ALS) algorithm has been derived for the model to sequentially update loading components and the patterns. In addition, together with a good initialisation method using tensor diagonalization, the update rules have been implemented with a low cost using fast inversion of block Toeplitz matrices as well as an efficient update strategy. Experiments show that the proposed model and the algorithm are promising in feature extraction and clustering.

Index Terms— tensor decomposition, tensor deconvolution, tensor diagonalization, CANDECOMP/PARAFAC

1. INTRODUCTION

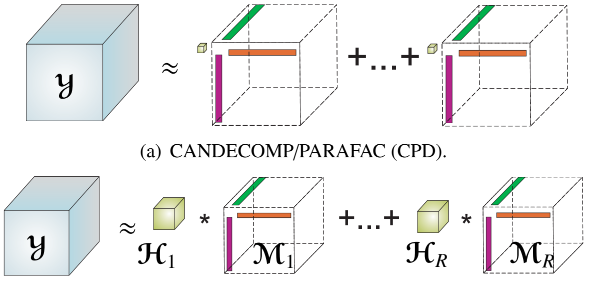
Tensor decomposition especially the CANDECOMP/PARAFAC tensor decomposition (CPD) has found a wide range of applications in variety of areas such as in chemometrics, telecommunication, data mining, neuroscience, blind source separation [1–4]. One of important applications of CPD is to extract hidden loading components which can provide physical insight of the source data. In order to deal with shifting factors in sequential data such as time series or spectra data, Harshman et. al [5] proposed the shifted CPD. The model has been applied to inspecting neural activity [6]. FitzGerald et al. extended the shift model to nonnegative tensor factorisation and applied it to music separation [7]. Cemgil et al. [8] introduced a probabilistic framework for non-negative factor deconvolution for audio modeling. Some other existing shifting models have been considered in nonnegative matrix factorisation (NMF) such as the convolutive NMF [9–11], two way CNMF, or multichannel NMF [12, 13]. In this paper, we consider a novel tensor deconvolution problem whose major aim is to represent multiway data by replicative patterns and activating maps following a convolutive model, that is

$$\mathcal{Y} \approx \mathcal{H}_1 * \mathcal{M}_1 + \cdots + \mathcal{H}_R * \mathcal{M}_R \qquad (1)$$

where “$*$″ denotes the tensor convolution, $\mathcal{Y}$ is of size $I_1 \times I_2 \times I_3$, $\mathcal{H}_r$ and $\mathcal{M}_r$ are tensors of size $J_{r1} \times J_{r2} \times J_{r3}$, and $K_{r1} \times K_{r2} \times K_{r3}$ with $J_{rn} + K_{rn} - 1 = I_n$, $1 \leq J_{rn} \leq I_n$, respectively. Although the roles of $\mathcal{H}_r$ and $\mathcal{M}_r$ are interchangeable because of commutativity of the convolution, we often consider patterns of relatively small sizes and $K_n \geq J_n$. More specifically, we consider a rank constrained model of (1) in which $\mathcal{M}_r$ are rank-1 tensors, for $r = 1, \ldots, R$, i.e., $\mathcal{M}_r = \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$, where $\boldsymbol{a}_r \in \mathbb{R}^{K_{r1}}$, $\boldsymbol{b}_r \in \mathbb{R}^{K_{r2}}$ and $\boldsymbol{c}_r \in \mathbb{R}^{K_{r3}}$ are vectors of unit length, and “$\circ$” represents the outer product. The low rank tensor deconvolution in (1) is rewritten as

$$\mathcal{Y} \approx \mathcal{H}_1 * (\boldsymbol{a}_1 \circ \boldsymbol{b}_1 \circ \boldsymbol{c}_1) + \cdots + \mathcal{H}_R * (\boldsymbol{a}_R \circ \boldsymbol{b}_R \circ \boldsymbol{c}_R) \qquad (2)$$

and illustrated in Fig. 1.



(a) CANDECOMP/PARAFAC (CPD).

(b) Low rank tensor deconvolution.

Fig. 1. (a) Illustration of CANDECOMP/PARAFAC (CPD) as a tool to extract rank-1 patterns from multiway data $\mathcal{Y}$, and (b) low rank tensor deconvolution which represents the data by small patches $\mathcal{H}_r$ and maps $\mathcal{M}_r$, which are rank-1 tensors.

When data is a matrix, i.e., $I_3 = 1$, the tensor deconvolution becomes rank-1 blind matrix deconvolution proposed in [14]. In a particular case, when patterns $\mathcal{H}_r$ all are scalar, i.e., $J_{r1} = J_{r2} = J_{r3} = 1$, the tensor deconvolution with R patterns simplifies into the rank-R CP tensor decomposition. In general cases, we will show that this tensor deconvolution can be expressed as the rank-(J_1, J_2, J_3) block tensor decomposition [15] with Toeplitz factor matrices.

For simplicity, patterns $\mathcal{H}_r$ are supposed to be the same size, i.e., $J_{r1} = J_1$, $J_{r2} = J_2$ and $J_{r3} = J_3$ for $r = 1, \ldots, R$. It follows that loading components are also of the same length, i.e., $K_{r1} = K_1$, $K_{r2} = K_2$, $K_{r3} = K_3$ for $r = 1, \ldots, R$. In the following section, we will derive an ALS algorithm which sequentially updates loading components $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, $\boldsymbol{c}_r$ and the tensors $\mathcal{H}_r$. Application of the model is then demonstrated for feature extraction and clustering of the hand-written digits.

[*]The work of P. Tichavský was supported by the Czech Science Foundation through project No. 14–13713S.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

2. ALGORITHM

We define a linear operator $\mathbf{X} = \tau_J(\boldsymbol{x})$ which maps a vector $\boldsymbol{x}$ of length K to a lower triangular Toeplitz matrix $\mathbf{X}$ of size $I \times J$, $I = K + J - 1$ whose first column is $[\boldsymbol{x}^T, 0, \ldots, 0]^T$

$$\mathbf{X} = \tau_J(\boldsymbol{x}), \qquad \text{vec}(\mathbf{X}) = \mathbf{T}_{K,J}\, \boldsymbol{x} \tag{3}$$

where $\mathbf{T}_{K,J} = [\mathbf{I}_K, \mathbf{0}_{K\times J}, \ldots, \mathbf{I}_K, \mathbf{0}_{K\times J}, \mathbf{I}_K]^T$ of size $(K + J - 1)J \times K$ comprises only ones and zeros. The tensor decomposition in (2) can be expressed as

$$\mathcal{Y} \approx \sum_{r=1}^{R} \mathcal{H}_r \times_1 \mathbf{A}_r \times_2 \mathbf{B}_r \times_3 \mathbf{C}_r, \tag{4}$$

where “$\times_n$” denotes product between a tensor and a matrix along mode-n, $\mathbf{A}_r = \tau_{J_1}(\boldsymbol{a}_r)$, $\mathbf{B}_r = \tau_{J_2}(\boldsymbol{b}_r)$, $\mathbf{C}_r = \tau_{J_{r3}}(\boldsymbol{c}_r)$ are Toeplitz matrices of size $I_1 \times J_1$, $I_2 \times J_2$ and $I_3 \times J_3$, respectively.

2.1. Update of loading components $\boldsymbol{a}_r$

In order to solve the problem (4), we minimise the following cost function

$$\min \quad D = \frac{1}{2} \|\mathcal{Y} - \sum_{r=1}^{R} \mathcal{H}_r \times_1 \mathbf{A}_r \times_2 \mathbf{B}_r \times_3 \mathbf{C}_r\|_F^2 . \tag{5}$$

We denote by $\mathbf{Y}_{(1)}$ the mode-1 matricization of $\mathcal{Y}$. The approximation in (4) can be rewritten in matrix form as

$$\mathbf{Y}_{(1)} \approx \sum_{r=1}^{R} \mathbf{A}_r\, \mathbf{H}_{(1)}^{(r)}\, (\mathbf{C}_r \otimes \mathbf{B}_r)^T \quad , \tag{6}$$

where “$\otimes$” denotes the Kronecker product, and $\mathbf{H}_{(1)}^{(r)}$ are mode-1 matricizations of the tensors $\mathcal{H}_r$ for $r = 1, \ldots, R$. The cost function in (5) is therefore expressed in an equivalent form

$$D = \frac{1}{2} \| \text{vec}(\mathbf{Y}_{(1)}) - \sum_{r=1}^{R} \left((\mathbf{C}_r \otimes \mathbf{B}_r) \left(\mathbf{H}_{(1)}^{(r)}\right)^T \otimes \mathbf{I}_{I_1} \right) \mathbf{T}_{K_1,J_1} \boldsymbol{a}_r \|_F^2 \ ,$$

whereas its gradients with respect to $\boldsymbol{a}_r$ for $r = 1, 2, \ldots, R$ are given by

$$\frac{\partial D}{\partial \boldsymbol{a}_r} = -\boldsymbol{w}_r + \sum_{s=1}^{R} \boldsymbol{\Phi}_{r,s}\, \boldsymbol{a}_s \,, \tag{7}$$

where vectors $\boldsymbol{w}_r$ of length K_1, and matrices $\boldsymbol{\Phi}_{r,s}$ of size $K_1 \times K_1$ are defined as

$$\begin{aligned}
\boldsymbol{w}_r &= \mathbf{T}_{K_1,J_1}^T \left(\mathbf{H}_{(1)}^{(r)}\, (\mathbf{C}_r \otimes \mathbf{B}_r)^T \otimes \mathbf{I}_{I_1} \right) \text{vec}(\mathcal{Y}) \\
&= \mathbf{T}_{K_1,J_1}^T\, \text{vec}\Big(\langle \mathcal{Y} \times_2 \mathbf{B}_r^T \times_3 \mathbf{C}_r^T, \mathcal{H}_r \rangle_{-1} \Big) \,, \qquad (8) \\
\boldsymbol{\Phi}_{r,s} &= \mathbf{T}_{K_1,J_1}^T\, (\mathbf{Z}_{r,s} \otimes \mathbf{I}_{I_1})\, \mathbf{T}_{K_1,J_1} \,, \qquad (9) \\
\mathbf{Z}_{r,s} &= \mathbf{H}_{(1)}^{(r)} \left(\mathbf{C}_r^T \mathbf{C}_s \otimes \mathbf{B}_r^T \mathbf{B}_s \right) \left(\mathbf{H}_{(1)}^{(s)} \right)^T \\
&= \langle \mathcal{H}_r \times_2 \mathbf{B}_s^T \mathbf{B}_r \times_3 \mathbf{C}_s^T \mathbf{C}_r, \mathcal{H}_s \rangle_{-1} \,. \qquad (10)
\end{aligned}$$

The notation $\langle \mathcal{X}, \mathcal{Y} \rangle_{-1} = \mathbf{X}_{(1)} \mathbf{Y}_{(1)}^T$ denotes contraction between two tensors along all modes but mode-1. Entries of vectors $\boldsymbol{w}_r$ are simply sums of entries on the main diagonal or on the k-sub diagonal of the $(I_1 \times J_1)$ matrices $\mathbf{W}_r = \langle \mathcal{Y} \times_2 \mathbf{B}_r^T \times_3 \mathbf{C}_r^T, \mathcal{H}_r \rangle_{-1}$ for $k = -1, \ldots -K_1 + 1$, that is

$$\boldsymbol{w}_r(k) = \mathbf{W}_r(k, 1) + \mathbf{W}_r(k + 1, 2) + \ldots + \mathbf{W}_r(k + J_1 - 1, J_1)\,,$$

for $k = 1, \ldots, K_1$. In addition, the matrices $\mathbf{Z}_{r,s}$ of size $J_1 \times J_1$ are efficiently computed through contraction along all modes but mode-1 between two tensors $\mathcal{H}_r \times_2 \mathbf{B}_s^T \mathbf{B}_r \times_3 \mathbf{C}_s^T \mathbf{C}_r$ and $\mathcal{H}_s$ of size $J_1 \times J_2 \times J_3$.

We define $\phi_j^{(r,s)}$ and $\phi_{-j}^{(r,s)}$ sums of all entries on the j-th super diagonal or j-th sub diagonal of the $(J_1 \times J_1)$ matrices $\mathbf{Z}_{r,s}$, for $j = 0, \ldots, J_1 - 1$, that is

$$\begin{aligned}
\phi_j^{(r,s)} &= z_{1,j+1}^{(r,s)} + z_{2,j+2}^{(r,s)} + \cdots + z_{J_1-j,J_1}^{(r,s)} \,, \qquad (11) \\
\phi_{-j}^{(r,s)} &= z_{j+1,1}^{(r,s)} + z_{j+2,2}^{(r,s)} + \cdots + z_{J_1,J_1-j}^{(r,s)} \,. \qquad (12)
\end{aligned}$$

From definition of the matrices $\mathbf{T}_{K_1,J_1}$, we obtain that $\boldsymbol{\Phi}_{r,s}$ in (9) are banded Toeplitz matrices of size $K_1 \times K_1$ given as

$$\boldsymbol{\Phi}_{r,s} = \begin{bmatrix}
\phi_0^{(r,s)} & \phi_1^{(r,s)} & \cdots & \phi_{J_1-1}^{(r,s)} & & & \\
\phi_{-1}^{(r,s)} & \phi_0^{(r,s)} & \ddots & & \ddots & & \\
\vdots & \ddots & \ddots & & & & \\
\phi_{-J_1+1}^{(r,s)} & & & & & \ddots & \\
& \ddots & & & & & \phi_{J_1-1}^{(r,s)} \\
& & & & & & \vdots \\
& & & \ddots & & \ddots & \phi_1^{(r,s)} \\
& & & \phi_{-J_1+1}^{(r,s)} & \cdots & \phi_{-1}^{(r,s)} & \phi_0^{(r,s)}
\end{bmatrix} .$$

By setting the gradients in (7) with respect to all $\boldsymbol{a}_r$ to zeros, we derive the following update rule for $\boldsymbol{a} = \left[\boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_R^T\right]^T$

$$\boldsymbol{a} \leftarrow \boldsymbol{\Phi}^{-1}\, \boldsymbol{w}, \tag{13}$$

where $\boldsymbol{w} = \left[\boldsymbol{w}_1^T, \ldots, \boldsymbol{w}_R^T\right]^T$ is a vector of length $K_1 R$, and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_{r,s}]$ is an $R \times R$ partitioned matrix of Toeplitz matrices $\boldsymbol{\Phi}_{r,s}$ for $1 \le r, s \le R$. Since $\mathbf{Z}_{r,s} = \mathbf{Z}_{s,r}^T$, $\boldsymbol{\Phi}_{r,s} = \boldsymbol{\Phi}_{s,r}^T$ and $\boldsymbol{\Phi}_{r,r}$ are symmetric. It follows that $\boldsymbol{\Phi}$ is a symmetric matrix of size $(RK_1 \times RK_1)$. We will show that $\boldsymbol{\Phi}$ can be expressed as a block Toeplitz matrix after swapping its row and columns, and thereby its inversion in (13) can be efficiently performed using fast inversion methods, e.g., the block Levinson recursion algorithm [16], or the algorithm in [17].

We define a permutation matrix $\mathbf{P}_{K_1,R}$ such that $\text{vec}\left(\mathbf{X}_{K_1,R}^T\right) = \mathbf{P}_{K_1,R}\, \text{vec}(\mathbf{X}_{K_1,R})$ for any matrix $\mathbf{X}_{K_1,R}$ of size $K_1 \times R$. It can be verified that permutation of columns and rows of $\boldsymbol{\Phi}$ using $\mathbf{P}_{K_1,R}$ yields a block Toeplitz matrix of $(R \times R)$ matrices $\tilde{\boldsymbol{\Phi}}_{-j}$ and $\tilde{\boldsymbol{\Phi}}_j = \tilde{\boldsymbol{\Phi}}_{-j}^T$ for $j = 0, 1, \ldots, J_1 - 1$ given as

$$\tilde{\boldsymbol{\Phi}} = \mathbf{P}_{K_1,R}\, \boldsymbol{\Phi}\, \mathbf{P}_{K_1,R}^T = \begin{bmatrix}
\tilde{\boldsymbol{\Phi}}_0 & \cdots & \tilde{\boldsymbol{\Phi}}_{J_1-1} & & & \\
\vdots & \ddots & & \ddots & & \\
\tilde{\boldsymbol{\Phi}}_{-J_1+1} & & & & \ddots & \\
& \ddots & & & & \tilde{\boldsymbol{\Phi}}_{J_1-1} \\
& & \ddots & & \ddots & \vdots \\
& & & \tilde{\boldsymbol{\Phi}}_{-J_1+1} & \cdots & \tilde{\boldsymbol{\Phi}}_0
\end{bmatrix},$$

where

$$\tilde{\boldsymbol{\Phi}}_j = \begin{bmatrix}
\phi_j^{(1,1)} & \phi_j^{(1,2)} & \cdots & \phi_j^{(1,R)} \\
\phi_j^{(2,1)} & \phi_j^{(2,2)} & \cdots & \phi_j^{(2,R)} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_j^{(R,1)} & \phi_j^{(R,2)} & \cdots & \phi_j^{(R,R)}
\end{bmatrix}, \tag{14}$$

and $\phi_j^{(r,s)}$ are defined in (11) and (12). Using the above expression, the update rule in (13) is rewritten as

$$\boldsymbol{a} \leftarrow \mathbf{P}_{K_1,R}^T \left(\mathbf{P}_{K_1,R}\, \boldsymbol{\Phi}\, \mathbf{P}_{K_1,R}^T\right)^{-1} (\mathbf{P}_{K_1,R}\, \boldsymbol{w}) = \mathbf{P}_{K_1,R}^T \left(\tilde{\boldsymbol{\Phi}}^{-1}\, \tilde{\boldsymbol{w}}\right) . \quad (15)$$

This update rule requires a cost of $O(K_1^2 R^2)$ and needs only J_1 matrices $\tilde{\boldsymbol{\Phi}}_j$ of size $R \times R$, i.e., $J_1 R^2$ coefficients [16, 18].

In a particular case when $J_1 = 1$, the update rule (15) can be simplified into a simple form given as

$$[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_R] \leftarrow [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_R]\, \mathbf{Z}^{-1} , \quad (16)$$

where $\mathbf{Z}$ is a matrix of size $(R \times R)$ whose entries $\mathbf{Z}(r, s)$ are given in (10). Loading components $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ are updated in the similar way.

2.2. Update of core tensors $\mathbf{H}_r$

In order to derive update rules for $\mathcal{H}_r$, we compute derivatives of the cost function in (5) with respect to $\mathcal{H}_r$. The derivatives are set to zero to obtain the following update rule

$$\begin{bmatrix} \text{vec}(\mathcal{H}_1) \\ \vdots \\ \text{vec}(\mathcal{H}_R) \end{bmatrix} \leftarrow \begin{bmatrix} \boldsymbol{\Psi}_{1,1} & \ldots & \boldsymbol{\Psi}_{1,R} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Psi}_{R,1} & \ldots & \boldsymbol{\Psi}_{R,R} \end{bmatrix}^{-1} \begin{bmatrix} \text{vec}(\mathcal{V}_1) \\ \vdots \\ \text{vec}(\mathcal{V}_R) \end{bmatrix} , \quad (17)$$

where $\mathcal{V}_r = \mathcal{Y} \times_1 \mathbf{A}_r^T \times_2 \mathbf{B}_r^T \times_r \mathbf{C}_r^T$ are tensors of size $J_1 \times J_2 \times J_3$, and $\boldsymbol{\Psi}_{r,s} = (\mathbf{C}_r^T \mathbf{C}_s) \otimes (\mathbf{B}_r^T \mathbf{B}_s) \otimes (\mathbf{A}_r^T \mathbf{A}_s)$. The partitioned matrix $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_{r,s}]$ in (17) is of size $R(J_1 J_2 J_3) \times R(J_1 J_2 J_3)$. In practice, we often seek relatively small patterns $\mathcal{H}_r$, the inversion $\boldsymbol{\Psi}^{-1}$ can be proceeded quickly.

2.3. Initialization and efficient implementation

For the noise-less case, we use the tensor diagonalization (TEDIA) [19] to seek for matrices which transform the data tensor $\mathcal{Y}$ to be diagonal or block diagonal form. Loading components $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ are then estimated using the Toeplitz matrix factorization [20]. For other cases, TEDIA is used to generate initial points for the deconvolution. The procedure is explained in more detail in the next section.

In update rules in (15) and (17), besides the cost due to solving linear systems, construction of vectors $\boldsymbol{w}_r$ in (8) and tensors $\mathcal{V}_r$ in (17) is expense with a cost of $O(I_1 I_2 I_3 \, \min(J_2, J_3))$. The computations are significantly time-consuming when the tensor sizes are large, because of large memory operations associated with tensor permutations [21]. Taking into account that the tensor product $\mathcal{F}_r = \mathcal{Y} \times_2 \mathbf{B}_r^T \times_3 \mathbf{C}_r^T$ is the common part involving in construction of $\mathcal{V}_r$ and $\boldsymbol{w}_w$. After updating $\boldsymbol{a}_r$, the tensors $\mathcal{V}_r$ are computed quickly as $\mathcal{V}_r = \mathcal{F}_r \times_1 \mathbf{A}_r$. Therefore we suggest to update patterns $\mathcal{H}_r$ after each update of loading components $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$. The update order is as follows: update $[\boldsymbol{a}_r]$, update $[\mathbf{H}_r]$, update $[\boldsymbol{b}_r]$, update $[\mathbf{H}_r]$, update $[\boldsymbol{c}_r]$, update $[\mathbf{H}_r]$, and so on.

Finally, a direct evaluation of the cost function (5) as a stopping criterion can be the most expensive step. Since we complete each iteration by updating $\mathcal{H}_r$ using (17), the cost value (5) is simply computed without construction of the approximate tensor to $\mathcal{Y}$ as

$$D = \frac{1}{2}\left(\|\mathcal{Y}\|_F^2 - \sum_{r=1}^{R} \text{vec}(\mathcal{V}_r)^T \, \text{vec}(\mathcal{H}_r) \right) . \quad (18)$$

Table 1. Clustering accuracy (%) using the low rank tensor deconvolution with rank $R = 2$. The values inside parentheses indicate the percentage of improvement of tensor deconvolution compared to the approach based on rank-2 CPD.

Digits	1	2	3	4	5	6	7	8	9
0	98.5	90.5 (4)	94.5 (2)	94.5 (3.5)	84.5	90 (3)	98 (**9**)	97.5 (3.5)	91 (1)
1		89.5 (**39**)	92.5 (**40.5**)	97.5 (5)	83 (6)	95	92.5 (**15**)	86.5 (**33.5**)	94.5 (**23**)
2			89.5	92	84.5	91 (**27**)	89.5 (4.5)	72.5 (**20.5**)	87 (4.5)
3				98 (**9**)	67 (3.5)	97 (5.5)	94 (4.5)	81 (**28.5**)	89.5 (5)
4					87 (5)	91 (**20.5**)	93.5 (**36.5**)	95.5	67
5						65 (1.5)	92.5 (**10.5**)	67.5 (8.5)	58.5
6							94.5	96 (2)	96 (4)
7								93.5 (**23**)	69 (**17.5**)
8									78 (**13**)

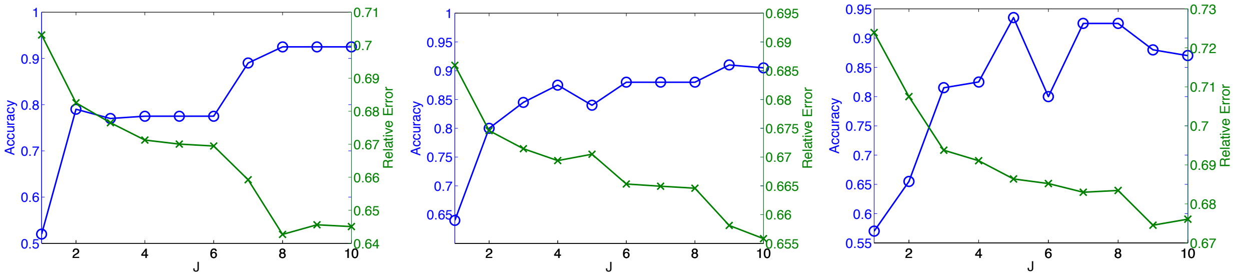
3. APPLICATIONS TO FEATURE EXTRACTION

This section introduces an application of tensor deconvolution to feature extraction. Assuming that slides $\mathbf{Y}_k$ for $k = 1, \ldots, I_3$ represent samples of certain entity, e.g., two-dimensional images. Then the third loading components $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_R$ explain relation between I_3 samples. In a particular case when $J_3 = 1$, i.e., patterns are matrices of size $J_1 \times J_2$, vectors $\boldsymbol{c}_r$, for $r = 1, \ldots, R$, represent R feature vectors associated with R basis components $\mathbf{H}_r * (\boldsymbol{a}_r \, \boldsymbol{b}_r^T) = \tau_{J_1}(\boldsymbol{a}_r)\, \mathbf{H}_r \, \tau_{J_2}(\boldsymbol{b}_r)^T$ of rank $\min(J_1, J_2)$, respectively. It is worth noting that CPD with rank-R also yields R feature vectors. However, its R basis components are only of rank-1, and do not explain complex structure as those in the tensor deconvolution.

We illustrate the tensor deconvolution in clustering applications on the MNIST handwritten digits[1]. We took the first 100 images of size 24 × 24 for each digit, and applied the tensor deconvolution to the data consisting of 200 images for each pair of digits, e.g., 0 and 1, 2 and 4. So far, there does not exist a method to determine the number of patterns, i.e. R, which is related to rank determination in the block tensor decomposition. However, one can select a suitable R by balancing the approximation error versus the number of parameters [3], or through a cross-validation technique. In this paper, the deconvolution estimated two common patterns $\mathbf{H}_1$ and $\mathbf{H}_2$ of size $J \times J \times 1$, where J varied in the range of $[1, 10]$

$$\mathcal{Y} \approx \sum_{r=1}^{R} \mathbf{H}_r * (\boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r) = \sum_{r=1}^{R} (\mathbf{H}_r * (\boldsymbol{a}_r \circ \boldsymbol{b}_r)) \circ \boldsymbol{c}_r . \quad (19)$$

[1] `http://yann.lecun.com/exdb/mnist/`

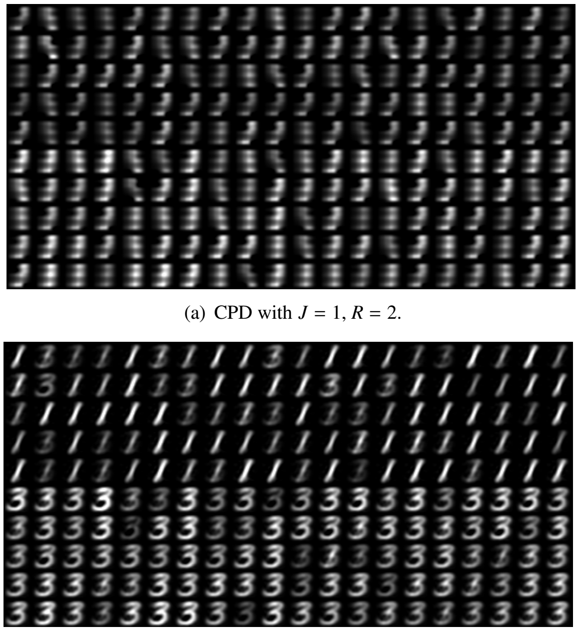


(a) Digits 1 and 3

(b) Digits 2 and 6

(c) Digits 4 and 7

Fig. 2. Illustration of tensor deconvolution with two patterns of size $J \times J \times 1$ for clustering of two digits 1 and 3, 2 and 6, 4 and 7. Relative approximation errors and clustering accuracies are obtained with pattern sizes $J = 1, 2, \ldots, 10$.



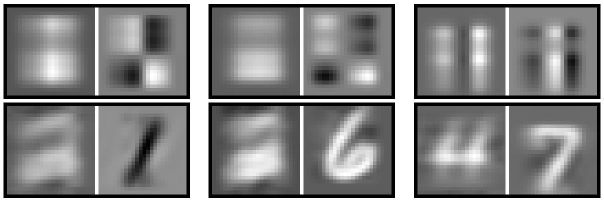
(a) CPD with $J = 1, R = 2$.

(b) Tensor deconvolution with $J = 8$ and $R = 2$

Fig. 3. Approximate images for two digits 1 and 3 using only two patterns when $J = 1$ and $J = 8$.

Parameters were initialised using the TEDIA algorithm (including the Tucker compression [22,23]) for two-sided block-diagonalization of the compressed tensor of size $RJ \times RJ \times 200$. The two unconstrained factor matrices $\mathbf{A}_r^{unc}$ and $\mathbf{B}_r^{unc}$ obtained by TEDIA were then factorised to yield the Toeplitz matrices $\tau_J(\boldsymbol{a}_r)$ and $\tau_J(\boldsymbol{b}_r)$ [20]. The initial patterns $\mathbf{H}_r$ were estimated such that they minimized the Frobenius norm $\|\mathcal{Y} - \sum_{r=1}^{R} \mathcal{H}_r \times_1 \tau_J(\boldsymbol{a}_r) \times_2 \tau_J(\boldsymbol{b}_r)\|_F^2$, with fixed $\boldsymbol{a}_r$ and $\boldsymbol{b}_r$. Finally, we completed the initialisation by performing best rank-1 approximations to mode-3 matricization of $\mathcal{H}_r$, i.e., approximations $\mathcal{H}_r \approx \mathbf{H}_r \circ \boldsymbol{c}_r$ for $r = 1, \ldots, R$.

In Fig. 2, the relative approximation errors and clustering accuracies using K-means are illustrated as functions of size J for pairs 1 and 3, 2 and 6, 4 and 7. The results for CPD with rank-2 are shown with $J = 1$. For these selected pairs of digits, we did not achieve good clustering accuracies using two features extracted by CPD. However, as seen on Fig. 2, when $J = 8, 9$ and 10 for digits 1 and 3, digits 2 and 6, and $J = 5$ for digits 4 and 7, we obtained much better clustering accuracies ($\geq 92.5\%$) using only two features estimated by the tensor deconvolution. The high performance of the tensor deconvolution can be explained by complex structure of basis images $\mathbf{F}_r = \mathbf{H}_r * (\boldsymbol{a}_r \, \boldsymbol{b}_r^T)$ illustrated in Fig. 4. Basis images of rank-1 by CPD, i.e., $\boldsymbol{a}_r \boldsymbol{b}_r^T$, do not express sufficient structure of digits. With the more complex basis images, tensor deconvolution achieved lower approximation errors, which are confirmed in both Fig. 2 and Fig. 3. It is clear that in Fig. 3, reconstructed digits 1 and 3 could not be distinguished by rank-2 CPD as compared with those by tensor deconvolution. The improvement was also observed in clustering of other digits as summarised in Table 1.



(a) Digits 1 and 3

(b) Digits 2 and 6

(c) Digits 4 and 7

Fig. 4. Illustration of two basis images extracted for digits 1 and 3, 2 and 6, 4 and 7 using rank-2 CPD and tensor deconvolution with $R = 2$ and $J_1 = J_2 = J$, $J_3 = 1$.

4. CONCLUSIONS

The tensor deconvolution with rank-1 structures considered in this paper has shown advantage over the CP decomposition in providing complex structure basis patterns. The tensor deconvolution explains the data better than CPD, while keeping the same number of features R. With efficient implementation of update rules, initialisation and update strategy, our model and the proposed algorithm can be applied to feature extraction for other kinds of data. Matlab implementation is provided in the TENSORBOX package, and available online at: `http://www.bsp.brain.riken.jp/~phan/tensorbox.php`.

5. REFERENCES

[1] L.-H. Lim and P. Comon, "Blind multilinear identification," *CoRR*, vol. abs/1212.6663, 2012, preprint.

[2] A. Cichocki, R. Zdunek, A.-H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, Wiley, Chichester, 2009.

[3] A. Cichocki, D. P. Mandic, A.-H. Phan, C. Caiafa, G. Zhou, Q. Zhao, and L. De Lathauwer, "Tensor decompositions for signal processing applications. from two-way to multiway component analysis," *IEEE Signal Processing Magazine, accepted*, 2014.

[4] P. Comon, "Tensors: a brief introduction," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 44–53, 2014.

[5] R.A. Harshman, S. Hong, and M.E. Lundy, "Shifted factor analysis - Part I: Models and properties," *Journal of Chemometrics*, vol. 17, no. 7, pp. 363–378, 2003.

[6] M. Mørup, L. K. Hansen, S. M. Arnfred, L.-H. Lim, and K. H. Madsen, "Shift-invariant multilinear decomposition of neuroimaging data.," *NeuroImage*, vol. 42, no. 4, pp. 1439–1450, 2008.

[7] D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, 2008.

[8] A.T. Cemgil, U. Simsekli, and Y.C. Subakan, "Probabilistic latent tensor factorization framework for audio modeling," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, Oct 2011, pp. 137–140.

[9] P. Smaragdis, "Non-negative matrix factor deconvolution; Extraction of multiple sound sources from monophonic inputs," *Lecture Notes in Computer Science*, vol. 3195, pp. 494–499, 2004.

[10] P Smaragdis, "Convolutive speech bases and their application to speech separation," *IEEE Transactions of Speech and Audio Processing*, vol. 15, pp. 1–12, Jan 2007.

[11] R. Zdunek, "Improved convolutive and underdetermined blind audio source separation with MRF smoothing.," *Cognitive Computation*, vol. 5, no. 4, pp. 493–503, 2013.

[12] M. Mørup and M.N. Schmidt, "Sparse non-negative tensor factor double deconvolution (SNTF2D)for multi channel time-frequency analysis," Tech. Rep., Technical University of Denmark, DTU, 2006.

[13] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 3, pp. 550–563, March 2010.

[14] A.-H. Phan, P. Tichavský, A. Cichocki, and Z. Koldovský, "Low-rank blind nonnegative matrix deconvolution," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. 2012, pp. 1893–1896, IEEE.

[15] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms – Part III: Alternating least squares algorithms," *SIAM Journal of Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008, Special Issue Tensor Decompositions and Applications.

[16] H. Akaike, "Block Toeplitz matrix inversion," *SIAM Journal on Applied Mathematics*, vol. 24, no. 2, pp. 234–241, 1973.

[17] X.-G. Lv and T.-Z. Huang, "The inverses of block Toeplitz matrices," *Journal of Mathematics*, vol. 2013, no. ID 207176, pp. 8 pages, 1913.

[18] P. Alonso, J. M. Badía-Contelles, and A. M. Vidal, "Solving the block-Toeplitz least-squares problem in parallel," *Concurrency - Practice and Experience*, vol. 17, no. 1, pp. 49–67, 2005.

[19] P. Tichavský, A.-H. Phan, and A. Cichocki, "Non-orthogonal tensor diagonalization, a tool for block tensor decompositions," *arXiv*, vol. 1402.1673, 2014.

[20] E. Moulines, P. Duhamel, J. Cardoso, and S. Mayrargue, "Subspace methods for the blind identification of multichannel fir filters," *Signal Processing, IEEE Transactions on*, vol. 43, no. 2, pp. 516–525, Feb 1995.

[21] A.-H. Phan, P. Tichavský, and A. Cichocki, "Fast alternating LS algorithms for high order CANDECOMP/PARAFAC tensor factorizations," *Signal Processing, IEEE Transactions on*, vol. 61, no. 19, pp. 4834–4846, 2013.

[22] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1,R2,. . . ,RN) approximation of higher-order tensors," *SIAM Journal of Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[23] A.-H. Phan, A. Cichocki, and P. Tichavský, "On fast algorithms for orthogonal Tucker decomposition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6766–6770.