On Revealing Replicating Structures in Multiway Data: A Novel Tensor Decomposition Approach

Anh Huy Phan[1], Andrzej Cichocki[1,⋆], Petr Tichavský[2,⋆⋆],
Danilo P. Mandic[3,⋆⋆⋆], and Kiyotoshi Matsuoka[4]

[1] Brain Science Institute - RIKEN, Japan
[2] Institute of Information Theory and Automation, Czech Republic
[3] Imperial College, London, United Kingdom
[4] Kyushu University of Technology, Japan

Abstract. A novel tensor decomposition is proposed to make it possible to identify replicating structures in complex data, such as textures and patterns in music spectrograms. In order to establish a computational framework for this paradigm, we adopt a multiway (tensor) approach. To this end, a novel tensor product is introduced, and the subsequent analysis of its properties shows a perfect match to the task of identification of recurrent structures present in the data. Out of a whole class of possible algorithms, we illuminate those derived so as to cater for orthogonal and nonnegative patterns. Simulations on texture images and a complex music sequence confirm the benefits of the proposed model and of the associated learning algorithms.

Keywords: tensor decomposition, tensor product, pattern analysis, nonnegative matrix decomposition, structural complexity.

1 Problem Formulation

Estimation problems for data with self-replicating structures, such as images, various textures and music spectrograms require specifically designed approaches to identify, approximate, and retrieve the dynamical structures present in the data. By modeling data via summations of Kronecker products of two matrices (scaling and pattern matrices), Loan and Pitsianis [1] established an approximation to address this problem. Subsequently, Nagy and Kilmer [2] addressed 3-D image reconstruction from real-world imaging systems in which the point spread function was decomposed into a Kronecker product form, Bouhamidi and Jbilou [3] used Kronecker approximation for image restoration, Ford and Tyrtyshnikov focused on sparse matrices in the wavelet domain [4], while the extension to tensor data was addressed in [5].

[⋆] Also affiliated with the EE Dept., Warsaw University of Technology and with Systems Research Institute, Polish Academy of Science, Poland.

[⋆⋆] The work of P. Tichavsky was supported by Ministry of Education, Youth and Sports of the Czech Republic through the project 1M0572 and by Grant Agency of the Czech Republic through the project 102/09/1278.

[⋆⋆⋆] The work of Danilo P. Mandic has been supported by a Royal Society International Joint Project with Japan.

F. Theis et al. (Eds.): LVA/ICA 2012, LNCS 7191, pp. 297–305, 2012.

It is important to note that at present, the Kronecker approximation [1] is limited to 2-D structures which are required to have the same dimension. In this paper, we generalize this problem by considering replicas (or similar structures) for multiway data $\boldsymbol{\mathcal{Y}}$. To this end, we explain the tensor $\boldsymbol{\mathcal{Y}}$ by a set of patterns and their locations, while allowing the patterns to have different dimensions. In order to formulate mechanism of data replication, we define a new tensor product which is a generalization of the standard matrix Kronecker product, and is particularly suited for data with recurrent complex structures.

Definition 1 (Kronecker tensor product). *Let $\boldsymbol{\mathcal{A}} = [a_{\boldsymbol{j}}]$ and $\boldsymbol{\mathcal{B}} = [b_{\boldsymbol{k}}]$ be two N-dimensional tensors of size $J_1 \times J_2 \times \cdots \times J_N$ and $K_1 \times K_2 \times \cdots \times K_N$, respectively, $\boldsymbol{j} = [j_1, j_2, \ldots, j_N]$, $1 \leq j_n \leq J_n$ and $\boldsymbol{k} = [k_1, k_2, \ldots, k_N]$, $1 \leq k_n \leq K_n$. A Kronecker tensor product of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ is defined as an N-D tensor $\boldsymbol{C} = [c_{\boldsymbol{i}}] \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\boldsymbol{i} = [i_1, i_2, \ldots, i_N]$, $I_n = J_n K_n$ such that $c_{\boldsymbol{i}} = a_{\boldsymbol{j}}\, b_{\boldsymbol{k}}$, $i_n = k_n + (j_n - 1)K_n$, and is expressed as $\boldsymbol{C} = \boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$.*

Remark 1. *If $\boldsymbol{C}$ is partitioned into an $J_1 \times J_2 \times \cdots \times J_N$ block tensor, each block-$\boldsymbol{j}$ ($\boldsymbol{j} = [j_1, j_2, \ldots, j_N]$) can be written as $a_{\boldsymbol{j}}\boldsymbol{\mathcal{B}}$.*

In this article, we aim to solve the following problem:

Problem 1 (A New Tensor Decomposition). *Given an N-dimensional tensor $\boldsymbol{\mathcal{Y}}$ of size $I_1 \times I_2 \times \cdots \times I_N$, find smaller scale tensors $\boldsymbol{\mathcal{A}}_p$, $\boldsymbol{\mathcal{X}}_p$, $p = 1, ..., P$ such that*

$$\boldsymbol{\mathcal{Y}} \approx \sum_{p=1}^{P} \boldsymbol{\mathcal{A}}_p \otimes \boldsymbol{\mathcal{X}}_p. \tag{1}$$

We term sub-tensors $\boldsymbol{\mathcal{X}}_p$ of size $K_{p1} \times K_{p2} \times \cdots \times K_{pN}$ as patterns, while $\boldsymbol{\mathcal{A}}_p$ of dimensions $J_{p1} \times J_{p2} \times \cdots \times J_{pN}$ such that $I_n = J_{pn}\, K_{pn}$, are called intensities (see Remark 1).

As such, Problem 1 is a generalization of the 13th problem of Hilbert, which seeks to perform universal function approximation for a function of n variables by a number of functions of $(n-1)$ or fewer variables. This new tensor decomposition is different from other existing tensor/matrix decompositions such as the canonical polyadic decomposition (CP) [6], the Tucker decomposition (TD) [7] and the block component decomposition (BCD) [8], in that it models the relation between latent variables via links between factor matrices and core tensor(s) which can be diagonal (for CP) or dense tensors (for TD). In a particular case when all $\boldsymbol{\mathcal{A}}_p$, $p = 1, \ldots, P$ in (1) become vectors of size I_n or have only one non-singleton dimension, Problem 1 simplifies into BCD which finds only one factor matrix for each core tensor.

In the sequel, we introduce methods to solve Problem 1 with/without nonnegative constraints. Simulations on a music sequence and on complex images containing textures validate the proposed tensor decomposition.

2 Notation and Basic Multilinear Algebra

Throughout the paper, an N-dimensional vector will be denoted by an italic lowercase boldface letters, with its components in squared brackets, for example $\boldsymbol{i} = [i_1, i_2, \ldots, i_N]$ or $\boldsymbol{I} = [I_1, I_2, \ldots, I_N]$.

Definition 2 (Tensor unfolding [9]). *Unfolding a tensor* $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *along modes* $\boldsymbol{r} = [r_1, r_2, \ldots, r_M]$ *and* $\boldsymbol{c} = [c_1, c_2, \ldots, c_{N-M}]$ *where* $[\boldsymbol{r}, \boldsymbol{c}]$ *is a permutation of* $[1, 2, \ldots, N]$ *aims to rearrange this tensor to a matrix* $\mathbf{Y}_{\boldsymbol{r}\times\boldsymbol{c}}$ *of size* $\prod_{k=1}^{M} I_{r_k} \times \prod_{l=1}^{N-M} I_{c_l}$ *whose entries* (j_1, j_2) *are given by* $\mathbf{Y}_{\boldsymbol{r}\times\boldsymbol{c}}(j_1, j_2) = \boldsymbol{\mathcal{Y}}(\boldsymbol{i_r}, \boldsymbol{i_c})$*, where* $\boldsymbol{i_r} = [i_{r_1} \ldots i_{r_M}]$, $\boldsymbol{i_c} = [i_{c_1} \ldots i_{c_{N-M}}]$, $j_1 = \mathrm{ivec}(\boldsymbol{i_r}, \boldsymbol{I_r})$, $j_2 = \mathrm{ivec}(\boldsymbol{i_c}, \boldsymbol{I_c})$*, and* $\mathrm{ivec}(\boldsymbol{i}, \boldsymbol{I}) = i_1 + \sum_{n=2}^{N} (i_n - 1) \prod_{j=1}^{n-1} I_j$.

If $\boldsymbol{c} = [c_1 < c_2 < \cdots < c_{N-M}]$, then $\mathbf{Y}_{\boldsymbol{r}\times\boldsymbol{c}}$ simplifies into $\mathbf{Y}_{(\boldsymbol{r})}$, while for $r = n$ and $\boldsymbol{c} = [1, \ldots, n-1, n+1, \ldots, N]$, we have mode-$n$ matricization $\mathbf{Y}_{\boldsymbol{r}\times\boldsymbol{c}} = \mathbf{Y}_{(n)}$.

Definition 3 (Reshaping). *The reshape operator for a tensor* $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *to a tensor of a size specified by a vector* $\boldsymbol{L} = [L_1, L_2, \ldots, L_M]$ *with* $\prod_{m=1}^{M} L_m = \prod_{n=1}^{N} I_n$ *returns an M-D tensor* $\boldsymbol{\mathcal{X}}$*, such that* $\mathrm{vec}(\boldsymbol{\mathcal{Y}}) = \mathrm{vec}(\boldsymbol{\mathcal{X}})$*, and is expressed as*

$$\boldsymbol{\mathcal{X}} = \texttt{reshape}(\boldsymbol{\mathcal{Y}}, \boldsymbol{L}) \quad \in \mathbb{R}^{L_1 \times L_2 \times \cdots \times L_M}. \tag{2}$$

Definition 4 (Kronecker unfolding). *A* $(\boldsymbol{J} \times \boldsymbol{K})$ *Kronecker unfolding of* $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *with* $I_n = J_n K_n, \forall n$*, is a matrix* $\mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})}$ *of the size* $\prod_{n=1}^{N} J_n \times \prod_{n=1}^{N} K_n$ *whose entries* (j, k) *are given by*

$$\mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})}(j, k) = \boldsymbol{\mathcal{C}}(\boldsymbol{i}),$$

for all $\boldsymbol{j} = [j_1, \ldots, j_N]$, $j_n = 1, \ldots, J_n$, $\boldsymbol{k} = [k_1, \ldots, k_N]$, $k_n = 1, \ldots, K_n$, $n = 1, \ldots, N$ *and* $j = \mathrm{ivec}(\boldsymbol{j}, \boldsymbol{J})$*, and* $k = \mathrm{ivec}(\boldsymbol{k}, \boldsymbol{K})$, $\boldsymbol{i} = [i_1, \ldots, i_N]$, $i_n = k_n + (j_n - 1)K_n$.

Lemma 1 (Rank-1 Factorization). *Consider a tensor product* $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$ *where* $\boldsymbol{\mathcal{A}}$ *and* $\boldsymbol{\mathcal{B}}$ *have the dimensions as in Definition 1. Then a Kronecker unfolding* $\mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})}$ *is a rank-1 matrix*

$$\mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})} = \mathrm{vec}(\boldsymbol{\mathcal{A}})\ \mathrm{vec}(\boldsymbol{\mathcal{B}})^T. \tag{3}$$

Lemma 1 also provides a convenient way to compute and update $\boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$.

Lemma 2 (Implementation of the Kronecker unfolding). *Let* $\widetilde{\boldsymbol{\mathcal{C}}} = \mathrm{reshape}(\boldsymbol{\mathcal{C}}, \boldsymbol{L})$ *of* $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *following* $\boldsymbol{L} = [K_1, J_1, K_2, J_2, \ldots, K_N, J_N]$, $I_n = J_n K_n$, $n = 1, 2, \ldots, N$. *An* $(\boldsymbol{J} \times \boldsymbol{K})$ *Kronecker unfolding of* $\boldsymbol{\mathcal{C}}$ *is equivalent to a tensor unfolding* $\widetilde{\mathbf{C}}_{(\boldsymbol{r})} = \mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})}$ *where* $\boldsymbol{r} = [2, 4, \ldots, 2N]$.

Lemma 3 (Rank-P Factorization). *Let a tensor* $\boldsymbol{\mathcal{C}}$ *be expressed as a sum of P Kronecker products* $\boldsymbol{\mathcal{C}} = \sum_{p=1}^{P} \boldsymbol{\mathcal{A}}_p \otimes \boldsymbol{\mathcal{B}}_p$*, where* $\boldsymbol{\mathcal{A}}_p \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ *and* $\boldsymbol{\mathcal{B}}_p \in \mathbb{R}^{K_1 \times \cdots \times K_N}$, $p = 1, 2, \ldots, P$. *Then the Kronecker unfolding of* $\boldsymbol{\mathcal{C}}$ *is a matrix of rank-P, such that*

$$\mathbf{C}_{(\boldsymbol{J}\times\boldsymbol{K})} = \sum_{p=1}^{P} \mathrm{vec}\Big(\boldsymbol{\mathcal{A}}_p\Big)\ \mathrm{vec}\Big(\boldsymbol{\mathcal{B}}_p\Big)^T. \tag{4}$$

Lemmas 1 and 3 give us the necessary insight and physical intuition into methods for solving Problem 1, establishing that the Kronecker tensor decomposition of $\boldsymbol{\mathcal{Y}}$ is equivalent to factorizations of Kronecker unfoldings $\mathbf{Y}_{(\boldsymbol{J}\times\boldsymbol{K})}$. The algorithms for solving Problem 1 are presented in the subsequent section.

3 Decomposition Methods

The desired property of the tensor decomposition (1) is that not all patterns $\boldsymbol{\mathcal{X}}_p$ (and consequently intensities $\boldsymbol{\mathcal{A}}_p$) are required to have the same size. Assume that there are G pattern sizes ($G \leq P$) $K_{g1} \times K_{g2} \times \cdots \times K_{gN}$ ($g = 1, 2, \ldots, G$) corresponding to P patterns $\boldsymbol{\mathcal{X}}_p$ ($p = 1, 2, \ldots, P$). Patterns $\boldsymbol{\mathcal{X}}_p$ which have the same size are classified into the same group. There are G groups of pattern sizes whose indices are specified by $\mathcal{I}_g = \{p : \boldsymbol{\mathcal{X}}_p \in \mathbb{R}^{K_{g1} \times K_{g2} \times \cdots \times K_{gN}}\} = \{p_1^{(g)}, p_2^{(g)}, \ldots, p_{P_g}^{(g)}\}$, $\mathtt{card}\{\mathcal{I}_g\} = P_g, \sum_{g=1}^{G} P_g = P$. For simplicity, we assume that the first P_1 patterns $\boldsymbol{\mathcal{X}}_p$ ($p = 1, 2, \ldots, P_1$) belong to group 1, the next P_2 patterns ($p = P_1 + 1, \ldots, P_1 + P_2$) belong to group 2, and so on. The tensor decomposition (1) can now be rewritten as

$$\boldsymbol{\mathcal{Y}} = \sum_{g=1}^{G} \sum_{p_g \in \mathcal{I}_g} \boldsymbol{\mathcal{A}}_{p_g} \otimes \boldsymbol{\mathcal{X}}_{p_g} + \boldsymbol{\mathcal{E}} = \sum_{g=1}^{G} \boldsymbol{\mathcal{Y}}^{(g)} + \boldsymbol{\mathcal{E}} = \widehat{\boldsymbol{\mathcal{Y}}} + \boldsymbol{\mathcal{E}}, \tag{5}$$

where $\boldsymbol{\mathcal{A}}_{p_g} \in \mathbb{R}^{J_{g1} \times J_{g2} \times \cdots \times J_{gN}}$, $\boldsymbol{\mathcal{X}}_{p_g} \in \mathbb{R}^{K_{g1} \times K_{g2} \times \cdots \times K_{gN}}$ and $\boldsymbol{\mathcal{Y}}^{(g)} = \sum_{p_g \in \mathcal{I}_g} \boldsymbol{\mathcal{A}}_{p_g} \otimes \boldsymbol{\mathcal{X}}_{p_g}$. According to Lemma 3, Kronecker unfoldings $\mathbf{Y}^{(g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)}$ with $\boldsymbol{K}_g = [K_{g1}, K_{g2}, \ldots, K_{gN}]$, $\boldsymbol{J}_g = [J_{g1}, J_{g2}, \ldots, J_{gN}]$ are rank-P_g matrices, that is

$$\mathbf{Y}^{(g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)} = \sum_{p_g \in \mathcal{I}_g} \operatorname{vec}\left(\boldsymbol{\mathcal{A}}_{p_g}\right) \operatorname{vec}\left(\boldsymbol{\mathcal{X}}_{p_g}\right)^T. \tag{6}$$

In order to estimate $\boldsymbol{\mathcal{A}}_{p_g}$ and $\boldsymbol{\mathcal{X}}_{p_g}$, $\forall p_g \in \mathcal{I}_g$, we define $\boldsymbol{\mathcal{Y}}^{(-g)} = \boldsymbol{\mathcal{Y}} - \sum_{h \neq g} \boldsymbol{\mathcal{Y}}^{(h)}$, and minimize the cost function

$$\begin{aligned} D(\boldsymbol{\mathcal{Y}} \| \widehat{\boldsymbol{\mathcal{Y}}}) &= \|\boldsymbol{\mathcal{Y}} - \widehat{\boldsymbol{\mathcal{Y}}}\|_F^2 = \|\boldsymbol{\mathcal{Y}}^{(-g)} - \boldsymbol{\mathcal{Y}}^{(g)}\|_F^2 = \|\mathbf{Y}^{(-g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)} - \mathbf{Y}^{(g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)}\|_F^2 \\ &= \|\mathbf{Y}^{(-g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)} - \sum_{p_g \in \mathcal{I}_g} \operatorname{vec}\left(\boldsymbol{\mathcal{A}}_{p_g}\right) \operatorname{vec}\left(\boldsymbol{\mathcal{X}}_{p_g}\right)^T \|_F^2. \end{aligned} \tag{7}$$

In general, without any constraints, the matrix decomposition in (7) or the tensor decomposition (1) are not unique, since any basis of the columnspace of the matrix $\mathbf{Y}^{(-g)}_{(\boldsymbol{J}_g \times \boldsymbol{K}_g)}$ in (7) can serve as $\operatorname{vec}\left(\boldsymbol{\mathcal{A}}_{p_g}\right)$, $p_g \in I_g$. One possibility to enforce uniqueness is to restrict our attention to orthogonal bases in which the scalar product of two patterns $\boldsymbol{\mathcal{X}}_p$, $\boldsymbol{\mathcal{X}}_q$, defined as a sum of the element-wise products of $\boldsymbol{\mathcal{X}}_p$, $\boldsymbol{\mathcal{X}}_q$, is zero for all $p \neq q$. Alternative constraints for nonnegative data $\boldsymbol{\mathcal{Y}}$, such as nonnegativity, can also be imposed on $\boldsymbol{\mathcal{A}}_p$ and $\boldsymbol{\mathcal{X}}_p$. In other words, by using the background physics to constrain all $\boldsymbol{\mathcal{A}}_q$ and $\boldsymbol{\mathcal{X}}_q$ in the other groups $q \notin \mathcal{I}_g$, we can sequentially minimize (7). These constraints do not have a serious effect on the generality of the proposed solutions as real world nonnegative data often exhibit a degree of orthogonality, and images are nonnegative.

3.1 Orthogonal Patterns

Solving the matrix decomposition in (7) with orthogonal constraints yields vectorizations $\operatorname{vec}\left(\boldsymbol{\mathcal{A}}_{p_g}\right)$ and $\operatorname{vec}\left(\boldsymbol{\mathcal{X}}_{p_g}\right)$ ($p_g \in \mathcal{I}_g$) that are proportional to P_g leading left and right singular vectors of $\mathbf{Y}^{(-g)}_{(\boldsymbol{J}\times\boldsymbol{K})} \approx \mathbf{U}\,\mathrm{diag}\{\boldsymbol{s}\}\,\mathbf{V}^T$, where $\mathbf{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{P_g}]$ and $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{P_g}]$, that is,

$$\mathcal{A}_{p_l^{(g)}} = \mathrm{reshape}\Big(s_l\,\boldsymbol{u}_l, \boldsymbol{J}_g\Big), \quad \boldsymbol{J}_g = [J_{g1}, J_{g2}, \ldots, J_{gN}], \tag{8}$$

$$\boldsymbol{\mathcal{X}}_{p_l^{(g)}} = \mathrm{reshape}\Big(\boldsymbol{v}_l, \boldsymbol{K}_g\Big), \quad \boldsymbol{K}_g = [K_{g1}, K_{g2}, \ldots, K_{gN}]. \tag{9}$$

If all the patterns have the same size, then $\boldsymbol{K}_p = \boldsymbol{K}, \forall p$, $\mathcal{A}_p$ and $\boldsymbol{\mathcal{X}}_p$ are reshaped from P leading left and right singular vectors of the Kronecker unfolding $\mathbf{Y}_{(\boldsymbol{J}\times\boldsymbol{K})}$.

3.2 Nonnegative Patterns

We shall now revisit Problem 1 and introduce nonnegative constraints in order to find nonnegative $\mathcal{A}_p$ and $\boldsymbol{\mathcal{X}}_p$ from a nonnegative tensor $\boldsymbol{\mathcal{Y}}$. Such a constrained problem can be solved in a manner similar to the previous problem, that is, $\mathcal{A}_p$ and $\boldsymbol{\mathcal{X}}_p$ are updated by minimizing the cost functions in (7). Note that we can employ straightforwardly update rules for nonnegative least squares approximation: the multiplicative update rules [10] and the ALS algorithms. In the following, we present the multiplicative update rules, which can be directly applied to (7) and have the form

$$\mathrm{vec}\Big(\mathcal{A}_{p_g}\Big) \leftarrow \mathrm{vec}\Big(\mathcal{A}_{p_g}\Big) \circledast \Big(\mathbf{Y}_{(\boldsymbol{J}_g\times\boldsymbol{K}_g)}\ \mathrm{vec}\Big(\boldsymbol{\mathcal{X}}_{p_g}\Big)\Big) \oslash \Big(\widehat{\mathbf{Y}}_{(\boldsymbol{J}_g\times\boldsymbol{K}_g)}\ \mathrm{vec}\Big(\boldsymbol{\mathcal{X}}_{p_g}\Big)\Big), \tag{10}$$

$$\mathrm{vec}\Big(\boldsymbol{\mathcal{X}}_{p_g}\Big) \leftarrow \mathrm{vec}\Big(\boldsymbol{\mathcal{X}}_{p_g}\Big) \circledast \Big(\mathbf{Y}^T_{(\boldsymbol{J}_g\times\boldsymbol{K}_g)}\ \mathrm{vec}\Big(\mathcal{A}_{p_g}\Big)\Big) \oslash \Big(\widehat{\mathbf{Y}}^T_{(\boldsymbol{J}_g\times\boldsymbol{K}_g)}\ \mathrm{vec}\Big(\mathcal{A}_{p_g}\Big)\Big). \tag{11}$$

Note that if all the patterns have the same size, the constrained Problem 1 becomes nonnegative matrix factorization of the Kronecker unfolding $\mathbf{Y}_{(\boldsymbol{J}\times\boldsymbol{K})}$. In a particular case when data $\mathbf{Y}$ is matrix and all patterns have the same size, Problem 1 simplifies into the matrix decomposition proposed in [1].

4 Simulations

The introduced algorithms were verified by comprehensive simulations on synthetic benchmark data and on real-world images with texture and music data.

4.1 Synthetic Data

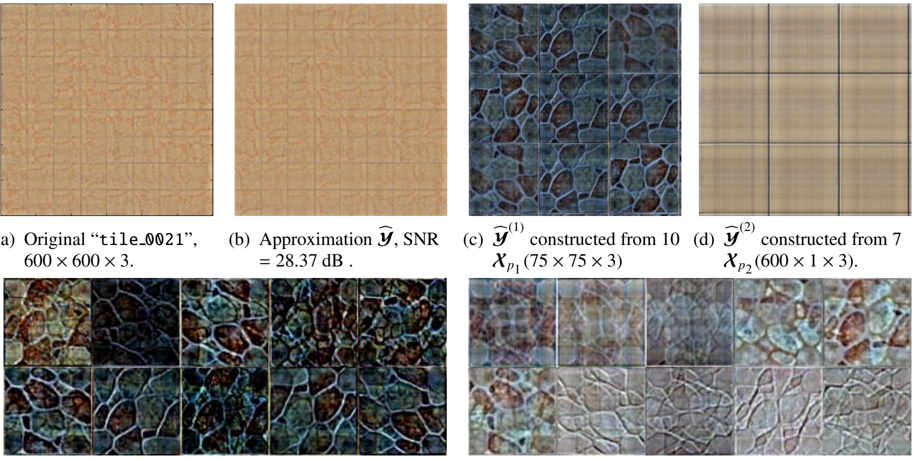
In the first set of simulations, we considered 3-D data of the size $90 \times 90 \times 12$ composed of 12 random nonnegative patterns of different sizes, as given in Table 1 (row 2). Our aim was to extract orthogonal and nonnegative patterns in 50000 iterations or until differences of successive relative errors (SNR) $-20\log_{10}\left(\frac{\|\boldsymbol{\mathcal{Y}}-\widehat{\boldsymbol{\mathcal{Y}}}\|_F}{\|\boldsymbol{\mathcal{Y}}\|_F}\right)$ are lower than 10^{-5}. Results (SNR) in Table 1 (the second row) show an average SNR = 110.16 dB over 100 runs for orthogonal decomposition, and an average SNR = 107.43 dB based on nonnegative patterns. The results confirm the validity of the proposed model and the excellent convergence of the proposed algorithms.

4.2 Analysis of Texture Images

The next set of simulations were performed on RGB textures "`tile_0021`" and "`metal_plate_0020`" taken from `http://texturelib.com`. Textures can be represented by 3-D tensors of pixels, or by 4-D tensors with additional modes for approximation and detail coefficients in the wavelet domain. For example, the image "`tile_0021`" of size $600 \times 600 \times 3$ is tiled by patterns $\boldsymbol{\mathcal{X}}_p$ of size $75 \times 75 \times 3$ as illustrated in Fig. 1(a). Detail coefficients of this image obtained by the biorthogonal wavelet transform formulate a 3-D tensor of size $300 \times 300 \times 3 \times 3$. The approximation coefficients can be independently decomposed or combined with the tensor of detail coefficients. Parameters of Kronecker decompositions such as the number of patterns and their dimensions are given in Table 1. Approximation errors (SNR (dB)) and ratio (%) between the number of fitting parameters and the number of data elements are also given in Table 1.

In Fig. 1, the image "`tile_0021`" was approximated by two groups of orthogonal and nonnegative patterns. Two nonnegative basis images corresponding to two groups of patterns are shown in Figs. 1(c), 1(d). The first group consists of 10 patterns $\boldsymbol{\mathcal{X}}_{p_1} \in \mathbb{R}_+^{75\times75\times3}$ (shown in Fig. 1(e)) expressing replicating structures, whereas the second group consists of 7 patterns of size $600 \times 1 \times 3$ representing the background as in Fig. 1(d). In addition, ten orthogonal patterns are shown in Fig. 1(f). For nonnegative patterns, each pattern in Fig. 1(e) represents a replicating structure in the image, whereas the orthogonal patterns in Fig. 1(f) were ranked according to the order of singular values which indicate detail level of patterns. Observe from Fig. 1(f) that the higher the order of the orthogonal patterns $\boldsymbol{\mathcal{X}}_p$, the more details these patterns comprise.

Results for decompositions of the color image "`metal_plate_0012`" are shown in Fig. 2. In the wavelet domain, we formulated a 3-D tensor for the approximation



(a) Original "`tile_0021`", $600 \times 600 \times 3$.

(b) Approximation $\widehat{\boldsymbol{\mathcal{Y}}}$, SNR = 28.37 dB .

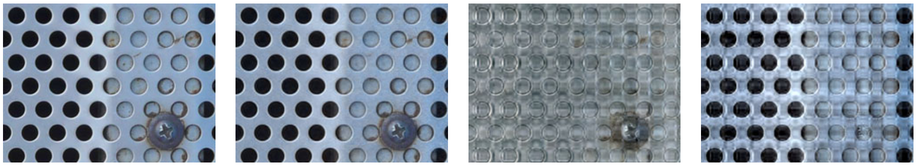
(c) $\widehat{\boldsymbol{\mathcal{Y}}}^{(1)}$ constructed from 10 $\boldsymbol{\mathcal{X}}_{p_1}$ $(75 \times 75 \times 3)$

(d) $\widehat{\boldsymbol{\mathcal{Y}}}^{(2)}$ constructed from 7 $\boldsymbol{\mathcal{X}}_{p_2}$ $(600 \times 1 \times 3)$.

(e) 10 nonnegative $\boldsymbol{\mathcal{X}}_{p_1}$ $(75 \times 75 \times 3)$.

(f) 10 orthogonal $\boldsymbol{\mathcal{X}}_{p_1}$ $(75 \times 75 \times 3)$.

Fig. 1. Illustration for orthogonal and nonnegative pattern decompositions of the image "`tile_0021`". (b)-(d) reconstructed images and two basis images by 10 patterns of size 75×75×3 and 7 patterns of size $600 \times 1 \times 3$. (e)-(f) 10 nonnegative and orthogonal patterns.



(a) “`metal_plate_0012`”, 180 × 240 × 3.

(b) Approximation $\widehat{\mathcal{Y}}$, SNR = 28.38 dB.

(c) $\widehat{\mathcal{Y}}^{(1)}$ constructed from orthogonal DWT $\mathcal{X}_{p_1}$.

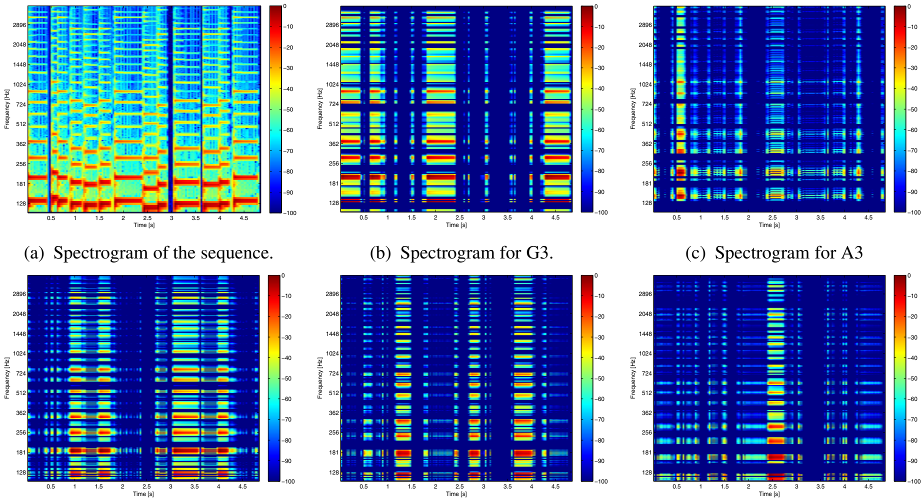
(d) $\widehat{\mathcal{Y}}^{(2)}$ constructed from orthogonal DWT $\mathcal{X}_{p_2}$.

Fig. 2. Approximation of “`metal_plate_0012`” in the wavelet domain

coefficients and a 4-D tensor comprising the details in the three orientations (horizontal, vertical, and diagonal). The two tensors were independently decomposed to find two groups of patterns whose sizes are given in Table 1 (row 4). The approximate image was then constructed from the basis patterns and achieved an SNR = 28.38 dB using 13.74 % of the number of entries. Figs. 2(c) and 2(d) visualize two basis images, each of which was constructed from one pattern group for the approximation coefficients and all the patterns for the detail coefficients.

4.3 Analysis of Patterns in Music

In this example, we decomposed a sampled song “London Bridge” composed of five notes A3, G3, F3, E3 and D3 played on a guitar for 5 seconds [10]. The log-frequency spectrogram $\mathbf{Y}$ (364 × 151), illustrated in Fig. 3(a), was converted from the



(a) Spectrogram of the sequence.

(b) Spectrogram for G3.

(c) Spectrogram for A3

(d) Spectrogram for F3.

(e) Spectrogram for E3.

(f) Spectrogram for D3.

Fig. 3. Log-frequency spectrograms of the music sequence and 5 basis nonnegative patterns corresponding to 5 notes G3, A3, F3, E3 and D3. The reconstructed signal has SNR = 20.78 dB.

Table 1. Parameters and results for orthogonal (Orho.) and nonnegative (NNG) pattern decompositions

Data	Size	Pattern Size	SNR (dB)		Ratio
		$(K_{g1} \times \cdots \times K_{gN}) \times P_g$	Ortho.	NNG	(%)
`random`	$90 \times 90 \times 12$	$(5 \times 5 \times 2) \times 2$ & $(6 \times 6 \times 3) \times 4$ & $(9 \times 9 \times 4) \times 6$	110.16	107.43	12.10
`tile_0021`	$600 \times 600 \times 3$	$(75 \times 75 \times 3) \times 10$ & $(600 \times 1 \times 3) \times 7$	29.69	28.37	17.24
	$300 \times 300 \times 3 \times 3$ (DWT, Detail Coefs.) $300 \times 300 \times 3$ (DWT, Approx. Coefs.)	$(20 \times 15 \times 1 \times 3) \times 20$ & $(300 \times 1 \times 1 \times 3) \times 3$ $(15 \times 15 \times 3) \times 40$ & $(300 \times 1 \times 3) \times 15$	27.84		9.48
`metal_plate_0012`	$180 \times 240 \times 3$	$(20 \times 20 \times 3) \times 15$ & $(180 \times 1 \times 3) \times 10$	27.58	25.35	21.16
	$90 \times 120 \times 3 \times 3$ (DWT, Detail Coefs.) $90 \times 120 \times 3$ (DWT, Approx. Coefs.)	$(5 \times 20 \times 1 \times 3) \times 3$ & $(90 \times 1 \times 1 \times 3) \times 10$ $(15 \times 15 \times 1) \times 20$ & $(90 \times 1 \times 1 \times 3) \times 5$	28.38		13.74
`guitar` music sequence	364×151 log-freq. spectrogram	$(4 \times 151) \times 5$ & $(2 \times 151) \times 4$ & $(7 \times 151) \times 2$	22.71	20.78	13.88

linear-frequency spectrogram in the frequency range from $f_0 = 109.4$ Hz (bin 8) to $f_l = f_s/2 = 4000$ Hz (bin 257) with 70 bins per octave. When there was no decomposition, the approximation error was 27.56 dB. The spectrogram was decomposed to find 11 patterns replicating along frequency (see row 5 in Table 1). Among the 11 log-frequency spectrograms $\widehat{\mathbf{Y}}^{(p)}$ constructed from $\mathbf{X}_p$, five spectrograms corresponding to five notes are illustrated in Figs. 3(b)-3(f). The approximate sequences (in the time domain) achieved SNR = 22.71 dB and 20.78 dB using orthogonal and nonnegative patterns, respectively. For this example, we may also apply the nonnegative matrix/tensor deconvolutions to seek for the similar patterns $\mathbf{X}_p$ replicating along frequency [11], however, the new tensor decomposition requires fewer fitting parameters.

5 Conclusions

A new tensor approximation has been proposed to identify and extract replicating structures from multiway data. By imposing a constraint on the replicating structures to be nonnegative or orthogonal, the model has been shown to significantly reduce the number of fitting parameters, compared with existing tensor/matrix factorizations. In a particular case when all the patterns have the same size, the new tensor decomposition simplifies into rank-P matrix factorization. This gives us a new insight and the ability to seek for hidden patterns by employing well-known matrix factorizations such as SVD and NMF. It has also been shown that a low-rank approximation by directly applying SVD or NMF to a data tensor results in common patterns which represent the background of the data, whereas factorization on the rearranged data extracts replicating structures. Simulation results for synthetic data, images and music sequence have shown that the proposed model and algorithms have the ability to extract desired patterns, and explain the data with relatively low approximation errors. Future extensions of the presented of this pattern decomposition will include approximating complex data by several subtensors instead of only two (scaling and pattern) tensors. One interesting implementation would be a multistage approach, in which patterns or scaling tensors are Kronecker products of subtensors.

References

1. Loan, C.V., Pitsianis, N.: Approximation with Kronecker products. In: Linear Algebra for Large Scale and Real Time Applications, pp. 293–314. Kluwer Publications (1993)
2. Nagy, J.G., Kilmer, M.E.: Kronecker product approximation for preconditioning in three-dimensional imaging applications. IEEE Transactions on Image Processing 15(3), 604–613 (2006)
3. Bouhamidi, A., Jbilou, K.: A Kronecker approximation with a convex constrained optimization method for blind image restoration. Optimization Letters, 1–14, doi: 10.1007/s11590-011-0370-7
4. Ford, J.M., Tyrtyshnikov, E.E.: Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related linear systems. SIAM J. Sci. Comput. 25, 961–981 (2003)
5. Hackbusch, W., Khoromskij, B.N., Tyrtyshnikov, E.E.: Hierarchical Kronecker tensor-product approximations. Journal of Numerical Mathematics 13(2), 119–156 (2005)
6. Harshman, R.: Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. UCLA Working Papers in Phonetics 16, 1–84 (1970)
7. Tucker, L.: Some mathematical notes on three-mode factor analysis. Psychometrika 31, 279–311 (1966)
8. De Lathauwer, L.: Decompositions of a higher-order tensor in block terms – Part I: Lemmas for partitioned matrices. SIAM J. Matrix Anal. Appl. 30(3), 1022–1032 (2008)
9. Bader, B., Kolda, T.: Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Transactions on Mathematical Software 32(4), 635–653 (2006)
10. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, ch.3. Wiley, Chichester (2009)
11. Phan, A.H., Tichavský, P., Cichocki, A., Koldovský, Z.: Low-rank blind nonnegative matrix deconvolution. In: Proc. IEEE ICASSP (2012)