Flexible Component Analysis for Sparse, Smooth, Nonnegative Coding or Representation

Andrzej Cichocki[⋆], Anh Huy Phan, Rafal Zdunek[⋆⋆], and Li-Qing Zhang[⋆⋆⋆]

RIKEN Brain Science Institute, Wako-shi, Saitama, Japan
cia@brain.riken.jp

Abstract. In the paper, we present a new approach to multi-way Blind Source Separation (BSS) and corresponding 3D tensor factorization that has many potential applications in neuroscience and multi-sensory or multidimensional data analysis, and neural sparse coding. We propose to use a set of local cost functions with flexible penalty and regularization terms whose simultaneous or sequential (one by one) minimization via a projected gradient technique leads to simple Hebbian-like local algorithms that work well not only for an over-determined case but also (under some weak conditions) for an under-determined case (i.e., a system which has less sensors than sources). The experimental results confirm the validity and high performance of the developed algorithms, especially with usage of the multi-layer hierarchical approach.

1 Introduction – Problem Formulation

Parallel Factor analysis (PARAFAC) or multi-way factorization models with sparsity and/or non-negativity constraints have been proposed as promising and quite efficient tools for processing of signals, images, or general data [1,2,3,4,5,6,7,8,9]. In this paper, we propose new hierarchical alternating algorithms referred to as the Flexible Component Analysis (FCA) for BSS, including as special cases: Nonnegative Matrix/Tensor Factorization (NMF/NTF), SCA (Sparse Components Analysis), SmoCA (Smooth Component Analysis). The proposed approach can be also considered as an extension of Morphological Component Analysis (MoCA) [10]. By incorporating nonlinear projection or filtering and/or by adding regularization and penalty terms to the local squared Euclidean distances, we are able to achieve nonnegative and/or sparse and/or smooth representations of the desired solution, and to alleviate a problem of getting stuck in local minima.

In this paper, we consider quite a general factorization related to the 3D PARAFAC2 model [1,5] (see Fig.1)

$$\boldsymbol{Y}_q = \boldsymbol{A}\boldsymbol{D}_q\widetilde{\boldsymbol{X}}_q + \boldsymbol{N}_q = \boldsymbol{A}\boldsymbol{X}_q + \boldsymbol{N}_q, \qquad (q = 1, 2, \ldots, Q) \tag{1}$$

[⋆] Dr. A. Cichocki is also with Systems Research Institute (SRI), Polish Academy of Science (PAN), and Warsaw University of Technology, Dept. of EE, Warsaw, Poland.

[⋆⋆] Dr. R. Zdunek is also with Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.

[⋆⋆⋆] Dr. L.-Q. Zhang is with the Shanghai Jiaotong University, China.

M. Ishikawa et al. (Eds.): ICONIP 2007, Part I, LNCS 4984, pp. 811–820, 2008.

where $\boldsymbol{Y}_q = [y_{itq}] \in \mathbb{R}^{I \times T}$ is the q-th frontal slice (matrix) of the observed (known) 3D tensor data or signals $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times T \times Q}$, $\boldsymbol{D}_q \in \mathbb{R}_+^{J \times J}$ is a diagonal scaling matrix that holds the q-th row of the matrix $\boldsymbol{D} \in \mathbb{R}^{Q \times J}$, $\boldsymbol{A} = [a_{ij}] = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_J] \in \mathbb{R}^{I \times J}$ is a mixing or basis matrix, $\widetilde{\boldsymbol{X}}_q = [\tilde{x}_{jtq}] \in \mathbb{R}^{J \times T}$ represents unknown normalized sources or hidden components in q-th slice, $\boldsymbol{X}_q = \boldsymbol{D}_q \widetilde{\boldsymbol{X}}_q = [x_{jtq}] \in \mathbb{R}^{J \times T}$ represents re-normalized (differently scaled) sources, and $\boldsymbol{N}_q = [n_{itq}] \in \mathbb{R}^{I \times T}$ represents the q-th frontal slice of the tensor $\underline{\boldsymbol{N}} \in \mathbb{R}^{I \times T \times Q}$ representing noise or errors. Our objective is to estimate the set of all matrices: $\boldsymbol{A}$, $\boldsymbol{D}_q$, $\widetilde{\boldsymbol{X}}_q$, subject to some natural constraints such as nonnegativity, sparsity or smoothness. Usually, the common factors, i.e., matrices $\boldsymbol{A}$ and $\widetilde{\boldsymbol{X}}_q$ are normalized to unit length column vectors and rows, respectively, and are often enforced to be as independent and/or as sparse as possible.

The above system of linear equations can be represented in an equivalent scalar form as follows $y_{itq} = \sum_j a_{ij} x_{jtq} + n_{itq}$, or equivalently in the vector form $\boldsymbol{Y}_q = \sum_j \boldsymbol{a}_j \; \underline{\boldsymbol{x}}_{jq} + \boldsymbol{N}_q$, where $\underline{\boldsymbol{x}}_{jq} = [x_{j1q}, x_{j2q}, \ldots, x_{jTq}]$ are the rows of $\boldsymbol{X}_q$, and $\boldsymbol{a}_j$ are the columns of $\boldsymbol{A}$ $(j = 1, 2, \ldots, J)$. Moreover, using the row-wise unfolding, the model (1) can be represented by one single matrix equation:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{N}, \tag{2}$$

where $\boldsymbol{Y} = [\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_Q] \in \mathbb{R}^{I \times QT}$, $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_Q] \in \mathbb{R}^{J \times QT}$, and $\boldsymbol{N} = [\boldsymbol{N}_1, \boldsymbol{N}_2, \ldots, \boldsymbol{N}_Q] \in \mathbb{R}^{I \times QT}$ are block row-wise unfolded matrices[1]. In the special case, for $Q = 1$ the model simplifies to the standard BSS model used in ICA, NMF, and SCA . The majority of the well-known algorithms for the PARAFAC models work only if the following assumption $T >> I \geq J$ is held, where J is known or can be estimated using PCA/SVD. In the paper, we propose a family of algorithms that can work also for an under-determined case, i.e., for $T >> J > I$, if sources are enough sparse and/or smooth. Our primary objective is to estimate the mixing (basis) matrix $\boldsymbol{A}$ and the sources $\boldsymbol{X}_q$, subject to additional natural constraints such as nonnegativity, sparsity and/or smoothness constraints. To deal with the factorization problem (1) efficiently, we adopt several approaches from constrained optimization, regularization theory, multi-criteria optimization, and projected gradient techniques. We minimize simultaneously or sequentially several cost functions with the desired constraints using switching between two sets of parameters: $\{\boldsymbol{A}\}$ and $\{\boldsymbol{X}_q\}$.

[1] It should be noted that the 2D unfolded model, in a general case, is not exactly equivalent to the PARAFAC2 model (in respect to sources $\boldsymbol{X}_q$), since we usually need to impose different additional constraints for each slice q. In other words, the PARAFAC2 model should not be considered as a 2-D model with the single 2-D unfolded matrix $\boldsymbol{X}$. Profiles of the augmented (row-wise unfolded) $\boldsymbol{X}$ can only be treated as a single profile, while we need to impose individual constraints independently to each slice $\boldsymbol{X}_q$ or even to each row of $\boldsymbol{X}_q$. Moreover, the 3D tensor factorization is considered as a dynamical process or a multi-stage process, where the data analysis is performed several times under different conditions (initial estimates, selected natural constraints, post-processing) to get full information about the available data and/or discover some inner structures in the data, or to extract physically meaningful components.

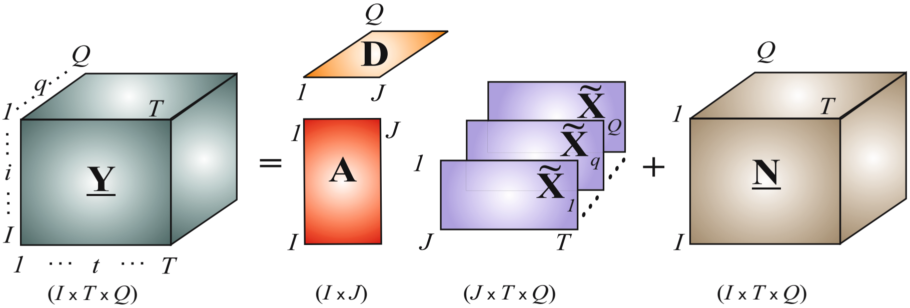
Q
q
1
i
I
1
t
T
Y
(I x T x Q)
=
D
Q
1
J
A
1
J
I
(I x J)
$\widetilde{\mathbf{X}}_Q$
$\widetilde{\mathbf{X}}_q$
$\widetilde{\mathbf{X}}_1$
1
J
T
(J x T x Q)
+
N
Q
1
T
I
(I x T x Q)

Fig. 1. Modified PARAFAC2 model illustrating factorization of 3D tensor into a set of fundamental matrices: $\boldsymbol{A}, \boldsymbol{D}, \{\widetilde{\boldsymbol{X}}_q\}$. In the special case, the model is reduced to standard PARAFAC for $\widetilde{\boldsymbol{X}}_q = \widetilde{\boldsymbol{X}}_1, \ \forall q$, or tri-NMF model for $Q = 1$.

2 Projected Gradient Local Least Squares Regularized Algorithm

Many algorithms for the PARAFAC model are based on Alternating Least Square (ALS) minimization of the squared Euclidean distance [1,4,5]. In particular, we can attempt to minimize a set of the following cost functions:

$$D_{Fq}(\boldsymbol{Y}_q || \boldsymbol{A}\boldsymbol{X}_q) = \frac{1}{2}\|\boldsymbol{Y}_q - \boldsymbol{A}\boldsymbol{X}_q\|_F^2 + \alpha_A J_A(\boldsymbol{A}) + \alpha_X J_{X_q}(\boldsymbol{X}_q), \qquad (3)$$

subject to some additional constraints, where $J_A(\boldsymbol{A})$, $J_{X_q}(\boldsymbol{X}_q)$ are penalty or regularization functions, and α_A and α_X are nonnegative coefficients controlling a tradeoff between data fidelity and *a priori* knowledge on the components to be estimated. A choice of the regularization terms can be critical for attaining a desired solution and noise suppression. Some of the candidates include the entropy, l_p-quasi norm and more complex, possibly non-convex or non-smooth regularization terms [11]. In such a case a basic approach to the above formulated optimization problem is alternating minimization or alternating projection: the specified cost function is alternately minimized with respect to two sets of the parameters $\{x_{jtq}\}$ and $\{a_{ij}\}$, each time optimizing one set of arguments while keeping the other one fixed [6,7].

In this paper, we consider a different approach: instead of minimizing only one global cost function, we perform sequential minimization of the set of local cost functions composed of the squared Euclidean terms and regularization terms:

$$D_{Fq}^{(j)}(\boldsymbol{Y}_q^{(j)} || \boldsymbol{a}_j \underline{\boldsymbol{x}}_{jq}) = \frac{1}{2}\|(\boldsymbol{Y}_q^{(j)} - \boldsymbol{a}_j \underline{\boldsymbol{x}}_{jq})\|_F^2 + \alpha_a^{(j)} \ J_a(\boldsymbol{a}_j) + \alpha_{X_q}^{(j)} \ J_x(\underline{\boldsymbol{x}}_{jq}), \quad (4)$$

for $j = 1, 2, \ldots, J, \ q = 1, 2, \ldots, Q$, subject to additional constraints, where

$$\boldsymbol{Y}_q^{(j)} = \boldsymbol{Y}_q - \sum_{r \neq j} \boldsymbol{a}_r \ \underline{\boldsymbol{x}}_{rq} = \boldsymbol{Y}_q - \boldsymbol{A}\boldsymbol{X}_q + \boldsymbol{a}_j \ \underline{\boldsymbol{x}}_{jq}, \qquad (5)$$

$\boldsymbol{a}_j \in \mathbb{R}^{I \times 1}$ are the columns of the basis mixing matrix $\boldsymbol{A}$, $\underline{\boldsymbol{x}}_{jq} \in \mathbb{R}^{1 \times T}$ are the rows of $\boldsymbol{X}_q$ which represent unknown source signals, $J_a(\boldsymbol{a}_j)$ and $J_x(\underline{\boldsymbol{x}}_{jq})$ are local penalty regularization terms which impose specific constraints for the estimated parameters, and $\alpha_a^{(j)} \geq 0$ and $\alpha_{X_q}^{(j)} \geq 0$ are nonnegative regularization parameters that control a tradeoff between data-fidelity and the imposed constraints.

The construction of such a set of local cost functions follows from the simple observation that the observed data can be decomposed as follows $\boldsymbol{Y}_q = \sum_{j=1}^{J} \boldsymbol{a}_j \underline{\boldsymbol{x}}_{jq} + \boldsymbol{N}_q, \ \forall q$. We are motivated to use of such a representation and decomposition, because $\underline{\boldsymbol{x}}_{jq}$ have physically meaningful interpretation as sources with specific temporal and morphological properties.

The penalty terms may take different forms depending on properties of the estimated sources. For example, if the sources are sparse, we can apply the l_p-quasi norm: $J_x(\underline{\boldsymbol{x}}_{jq}) = ||\boldsymbol{x}_{jq}||_p^p = (\sum_t |x_{jtq}|^p)^{1/p}$ with $0 < p \leq 1$, or alternatively we can use the smooth approximation $J_x(\underline{\boldsymbol{x}}_{jq}) = \left[\sum_t^T |x_{jtq}|^2 + \varepsilon\right]^{p/2}$, where $\varepsilon \geq 0$ is a small constant. In order to impose local smoothing of signals, we can apply the total variation (TV) $J_x(\underline{\boldsymbol{x}}_{jq}) = \sum_{t=1}^{T-1} |x_{j,t+1,q} - x_{j,t,q}|$, or if we wish to achieve a smoother solution: $J_x(\underline{\boldsymbol{x}}_{jq}) = \sum_{t=1}^{T-1} \sqrt{|x_{j,t+1,q} - x_{j,t,q}|^2 + \varepsilon}$, [12].

The gradients of the local cost function (4) with respect to the unknown vectors $\boldsymbol{a}_j$ and $\underline{\boldsymbol{x}}_{jq}$ are expressed by

$$\frac{\partial D_{Fq}^{(j)}(\boldsymbol{Y}_q^{(j)}||\boldsymbol{a}_j \, \underline{\boldsymbol{x}}_{jq})}{\partial \underline{\boldsymbol{x}}_{jq}} = \boldsymbol{a}_j^T \, \boldsymbol{a}_j \, \underline{\boldsymbol{x}}_{jq} - \boldsymbol{a}_j^T \, \boldsymbol{Y}_q^{(j)} + \alpha_{X_q}^{(j)} \, \Psi_x(\underline{\boldsymbol{x}}_{jq}), \tag{6}$$

$$\frac{\partial D_{Fq}^{(j)}(\boldsymbol{Y}_q^{(j)}||\boldsymbol{a}_j \, \underline{\boldsymbol{x}}_{jq})}{\partial \boldsymbol{a}_j} = \boldsymbol{a}_j \, \underline{\boldsymbol{x}}_{jq} \, \underline{\boldsymbol{x}}_{jq}^T - \boldsymbol{Y}_q^{(j)} \, \underline{\boldsymbol{x}}_{jq}^T + \alpha_a^{(j)} \, \Psi_a(\boldsymbol{a}_j), \tag{7}$$

where the matrix functions $\Psi_a(\boldsymbol{a}_j)$ and $\Psi_x(\underline{\boldsymbol{x}}_{jq})$ are defined as[2]

$$\Psi_a(\boldsymbol{a}_j) = \frac{\partial J_a^{(j)}(\boldsymbol{a}_j)}{\partial \boldsymbol{a}_j}, \qquad \Psi_x(\underline{\boldsymbol{x}}_{jq}) = \frac{\partial J_x^{(j)}(\underline{\boldsymbol{x}}_{jq})}{\partial \underline{\boldsymbol{x}}_{jq}}. \tag{8}$$

By equating the gradient components to zero, we obtain a new set of local learning rules:

$$\underline{\boldsymbol{x}}_{jq} \leftarrow \frac{1}{\boldsymbol{a}_j^T \, \boldsymbol{a}_j} \left(\boldsymbol{a}_j^T \, \boldsymbol{Y}_q^{(j)} - \alpha_{X_q}^{(j)} \, \Psi_x(\underline{\boldsymbol{x}}_{jq}) \right), \tag{9}$$

$$\boldsymbol{a}_j \leftarrow \frac{1}{\underline{\boldsymbol{x}}_{jq} \, \underline{\boldsymbol{x}}_{jq}^T} \left(\boldsymbol{Y}_q^{(j)} \, \underline{\boldsymbol{x}}_{jq}^T - \alpha_a^{(j)} \, \Psi_a(\boldsymbol{a}_j) \right), \tag{10}$$

for $j = 1, 2, \ldots, J$ and $q = 1, 2, \ldots, Q$.

However, it should be noted that the above algorithm provides only a regularized least squares solution, and this is not sufficient to extract the desired

[2] If the penalty functions are non-smooth, we can use sub-gradient instead of the gradient.

sources, especially for an under-determined case since the problem may have many solutions. To solve this problem, we need additionally to impose nonlinear projections $P_{\Omega_j}(\underline{\boldsymbol{x}}_{jq})$ or filtering after each iteration or each epoch in order to enforce that individual estimated sources $\underline{\boldsymbol{x}}_{jq}$ satisfy the desired constraints. All such projections or filtering can be imposed individually for the each source $\underline{\boldsymbol{x}}_{jq}$ depending on morphological properties of the source signals. The similar nonlinear projection $\widetilde{P}_{\Omega_j}(\boldsymbol{a}_j)$ can be applied, if necessary, individually for each vector $\boldsymbol{a}_j$ of the mixing matrix $\boldsymbol{A}$. Hence, using the projected gradient approach, our algorithm can take the following more general and flexible form:

$$\underline{\boldsymbol{x}}_{jq} \leftarrow \frac{1}{\boldsymbol{a}_j^T\,\boldsymbol{a}_j}(\boldsymbol{a}_j^T\boldsymbol{Y}_q^{(j)} - \alpha_{X_q}^{(j)}\Psi_x(\underline{\boldsymbol{x}}_{jq})), \qquad \underline{\boldsymbol{x}}_{jq} \leftarrow P_{\Omega_j}\{\underline{\boldsymbol{x}}_{jq}\}, \tag{11}$$

$$\boldsymbol{a}_j \leftarrow \frac{1}{\underline{\boldsymbol{x}}_{jq}\,\underline{\boldsymbol{x}}_{jq}^T}(\boldsymbol{Y}_q^{(j)}\underline{\boldsymbol{x}}_{jq}^T - \alpha_a^{(j)}\Psi_a(\boldsymbol{a}_j)), \qquad \boldsymbol{a}_j \leftarrow \widetilde{P}_{\Omega_j}\{\boldsymbol{a}_j\}; \tag{12}$$

where $P_{\Omega_j}\{\underline{\boldsymbol{x}}_{jq}\}$ means generally a nonlinear projection, filtering, transformation, local interpolation/extrapolation, inpainting, smoothing of the row vector $\underline{\boldsymbol{x}}_{jq}$. Such projections or transformations can take many different forms depending on required properties of the estimated sources (see the next section for more details).

Remark 1. In practice, it is necessary to normalize the column vectors $\boldsymbol{a}_j$ or the row vectors $\underline{\boldsymbol{x}}_{jq}$ to unit length vectors (in the sense of the l_p norm ($p = 1, 2, ..., \infty$)) in each iterative step. In the special case of the l_2 norm, the above algorithm can be further simplified by neglecting the denominator in (11) or in (12), respectively. After estimating the normalized matrices $\boldsymbol{A}$ and $\widetilde{\boldsymbol{X}}_q$ (i.e., the normalized $\boldsymbol{X}_q$ to unit-length rows), we can estimate the diagonal matrices, if necessary, as follows:

$$\boldsymbol{D}_q = \mathrm{diag}\{\boldsymbol{A}^+\,\boldsymbol{Y}_q\,\widetilde{\boldsymbol{X}}_q^{+}\}, \qquad (q = 1, 2, \ldots, Q). \tag{13}$$

3 Flexible Component Analysis (FCA) – Possible Extensions and Practical Implementations

The above simple algorithm can be further extended or improved (in respect to a convergence rate and performance). First of all, different cost functions can be used for estimating the rows of the matrices $\boldsymbol{X}_q$ ($q = 1, 2, \ldots, Q$) and the columns of the matrix $\boldsymbol{A}$. Furthermore, the columns of $\boldsymbol{A}$ can be estimated simultaneously, instead one by one. For example, by minimizing the set of cost functions in (4) with respect to $\underline{\boldsymbol{x}}_{jq}$, and simultaneously the cost function (3) with normalization of the columns $\boldsymbol{a}_j$ to an unit l_2-norm, we obtain a new FCA learning algorithm in which the individual rows of $\boldsymbol{X}_q$ are updated locally (row by row) and the matrix $\boldsymbol{A}$ is updated globally (all the columns $\boldsymbol{a}_j$ simultaneously):

$$\underline{\boldsymbol{x}}_{jq} \leftarrow \boldsymbol{a}_j^T\boldsymbol{Y}_q^{(j)} - \alpha_{X_q}^{(j)}\Psi_x(\underline{\mathrm{x}}_{jq}), \qquad \underline{\boldsymbol{x}}_{jq} \leftarrow P_{\Omega_j}\{\underline{\boldsymbol{x}}_{jq}\}, \qquad (j = 1, \ldots, J), \tag{14}$$

$$\boldsymbol{A} \leftarrow (\boldsymbol{Y}_q\boldsymbol{X}_q^T - \alpha_A\Psi_A(\boldsymbol{A}))(\boldsymbol{X}_q\boldsymbol{X}_q^T)^{-1}, \quad \boldsymbol{A} \leftarrow \widetilde{P}_{\Omega}(\boldsymbol{A}), \quad (q = 1, \ldots, Q), \tag{15}$$

with the normalization (scaling) of the columns of $\boldsymbol{A}$ to an unit length in the sense of the l_2 norm, where $\Psi_A(\boldsymbol{A}) = \partial J_A(\boldsymbol{A})/\partial \boldsymbol{A}$.

In order to estimate the basis matrix $\boldsymbol{A}$, we can use alternatively the following global cost function (see Eq. (2)): $D_F(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = (1/2)\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \alpha_A J_A(\boldsymbol{A})$. The minimization of the cost function for a fixed $\boldsymbol{X}$ leads to the updating rule

$$\boldsymbol{A} \leftarrow \left[\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A \Psi_A(\boldsymbol{A})\right] (\boldsymbol{X}\boldsymbol{X}^T)^{-1}. \tag{16}$$

3.1 Nonnegative Matrix/Tensor Factorization

In order to enforce sparsity and nonnegativity constraints for all the parameters: $a_{ij} \geq 0,\ x_{jtq} \geq 0,\ \forall\, i, t, q$, we can apply the "half-way rectifying" element-wise projection: $[\boldsymbol{x}]_+ = \max\{\varepsilon, \boldsymbol{x}\}$, where ε is a small constant to avoid numerical instabilities and remove background noise (typically, $\varepsilon = [10^{-2} - 10^{-9}]$). Simultaneously, we can impose weak sparsity constriants by using the l_1-norm penalty functions: $J_A(\boldsymbol{A}) = ||\boldsymbol{A}||_1 = \sum_{ij} a_{ij}$ and $J_x^{(j)}(\underline{\boldsymbol{x}}_{jq}) = ||\underline{\boldsymbol{x}}_{jq}||_1 = \sum_t x_{jtq}$. In such a case, the FCA algorithm for the 3D NTF2 (i.e., the PARAFAC2 with nonnegativity constraints) will take the following form:

$$\underline{\boldsymbol{x}}_{jq} \leftarrow \left[\boldsymbol{a}_j^T \boldsymbol{Y}_q^{(j)} - \alpha_{X_q}^{(j)} \boldsymbol{1}\right]_+, \qquad (j = 1, \ldots, J), \ \ (q = 1, \ldots, Q), \tag{17}$$

$$\boldsymbol{A} \leftarrow \left[(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A \boldsymbol{1})(\boldsymbol{X}\boldsymbol{X}^T)^{-1}\right]_+, \tag{18}$$

with normalization of the columns of $\boldsymbol{A}$ in each iterative step to a unit length with the l_2 norm, where $\boldsymbol{1}$ means a matrix of all ones of appropriate size.

It should be noted the above algorithm can be easily extended to semi-NMF or semi-NTF in which only some sources $\underline{\boldsymbol{x}}_{jq}$ are nonnegative and/or the mixing matrix $\boldsymbol{A}$ is bipolar, by simple removing the corresponding "half-wave rectifying" projections. Moreover, the similar algorithm can be used for arbitrary bounded sources with known lower and/or upper bounds (or supports), i.e $l_{jq} \leq x_{jtq} \leq u_{iq},\ \forall t$, rather than $x_{jtq} \geq 0$, by using suitably chosen nonlinear projections which bound the solutions.

3.2 Smooth Component Analysis (SmoCA)

In order to enforce smooth estimation of the sources $\underline{\boldsymbol{x}}_{jq}$ for all or some preselected indexes j and q, we may apply after each iteration (epoch) the local smoothing or filtering of the estimated sources, such as the MA (Moving Average), EMA, SAR or ARMA models.

A quite efficient way of smoothing and denoising can be achieved by minimizing the following cost function (which satisfies multi-resolution criterion):

$$J(\underline{\boldsymbol{x}}_{jq}) = \sum_{t=1}^{T} (x_{jtq} - \widehat{x}_{jtq})^2 + \sum_{t=1}^{T-1} \lambda_{jtq}\ g_t\left(\widehat{x}_{j,t+1,q} - \widehat{x}_{jtq}\right), \tag{19}$$

where $\widehat{x}_{jtq}$ is a smoothed version of the actually estimated (noisy) x_{jtq}, $g_t(u)$ is a convex continuously differentiable function with a global minimum at $u = 0$, and λ_{jtq} are parameters that are data driven and chosen automatically.

3.3 Multi-way Sparse Component Analysis (MSCA)

In the sparse component analysis an objective is to estimate the sources $\underline{\boldsymbol{x}}_{jq}$ which are sparse and usually with a prescribed or specified sparsification profile, possibly with additional constraints like local smoothness. In order to enforce that the estimated sources are sufficiently sparse, we need to apply a suitable nonlinear projection or filtering which allows us adaptively to sparsify the data. The simplest nonlinear projection which enforces some sparsity to the normalized data is to apply the following weakly nonlinear element-wise projection:

$$P_{\Omega_j}(x_{jtq}) = \mathrm{sign}(x_{jtq})|x_{jtq}|^{1+\alpha_{jq}} \tag{20}$$

where α_{jq} is a small parameter which controls sparsity. Such nonlinear projection can be considered as a simple (trivial) shrinking. Alternatively, we may use more sophisticated adaptive local soft or hard shrinking in order to sparsify individual sources. Usually, we have the three-steps procedure: First, we perform the linear transformation: $\underline{\boldsymbol{x}}_w = \underline{\boldsymbol{x}}\boldsymbol{W}$, then, the nonlinear shrinking (adaptive thresholding), e.g., the soft element-wise shrinking: $P_\Omega(\underline{\boldsymbol{x}}_w) = \mathrm{sign}(\underline{\boldsymbol{x}}_w)\,[|\underline{\boldsymbol{x}}_w| - \delta]_+^{1+\delta}$, and finally the inverse transform: $\widehat{\underline{\boldsymbol{x}}} = P_\Omega(\underline{\boldsymbol{x}}_w)\boldsymbol{W}^{-1}$. The threshold $\delta > 0$ is usually not fixed but it is adaptively (data-driven) selected or it gradually decreases to zero with iterations. The optimal choice for a shrinkage function depends on a distribution of data. We have tested various shrinkage functions with gradually decreasing δ: the hard thresholding rule, soft thresholding rule, non-negative Garrotte rule, n-degree Garotte, and Posterior median shrinkage rule [13]. For all of them, we have obtained the promising results, and usually the best performance appears for the simple hard rule.

Our method is somewhat related to the MoCA and SCA algorithms, proposed recently by Bobin et al., Daubechies et al., Elad et al., and many others [10,14,11]. However, in contrast to these approaches our algorithms are local and more flexible. Moreover, the proposed FCA is more general than the SCA, since it is not limited only to a sparse representation via shrinking and linear transformation but allows us to impose general and flexible (soft and hard) constraints, nonlinear projections, transformations, and filtering[3]. Furthermore, in the contrast to many alternative algorithms which process the columns of $\boldsymbol{X}_q$, we process their rows which represent directly the source signals.

We can outline the FCA algorithm as follows:

1. Set the initial values of the matrix $\boldsymbol{A}$ and the matrices $\boldsymbol{X}_q$, and normalize the vectors $\boldsymbol{a}_j$ to an unit l_2-norm length.
2. Calculate the new estimate $\underline{\boldsymbol{x}}_{jq}$ of the matrices $\boldsymbol{X}_q$ using the iterative formula in (14).
3. If necessary, enforce the nonlinear projection or filtering by imposing natural constraints on individual sources (the rows of $\boldsymbol{X}_q$, $(q = 1, 2, \ldots, Q)$), such as nonnegativity, boundness, smoothness, and/or sparsity.

[3] In this paper, in fact, we use two kinds of constraints: the soft (or weak) constraints via penalty and regularization terms in the local cost functions, and the hard (strong) constraints via iteratively adaptive postprocessing using nonlinear projections or filtering.

4. Calculate the new estimate of $\boldsymbol{A}$ from (16), normalize each column of $\boldsymbol{A}$ to an unit length, and impose the additional constraints on $\boldsymbol{A}$, if necessary.
5. Repeat the steps (2) and (4) until the convergence criterion is satisfied.

3.4 Multi-layer Blind Identification

In order to improve the performance of the FCA algorithms proposed in this paper, especially for ill-conditioned and badly-scaled data and also to reduce the risk of getting stuck in local minima in non-convex alternating minimization, we have developed the simple hierarchical multi-stage procedure [15] combined together with multi-start initializations, in which we perform a sequential decomposition of matrices as follows. In the first step, we perform the basic decomposition (factorization) $\boldsymbol{Y}_q \approx \boldsymbol{A}^{(1)}\boldsymbol{X}_q^{(1)}$ using any suitable FCA algorithm presented in this paper. In the second stage, the results obtained from the first stage are used to build up a new tensor $\widehat{\underline{\boldsymbol{X}}}_1$ from the estimated frontal slices defined as $\widehat{\boldsymbol{Y}}_q^{(1)} = \boldsymbol{X}_q^{(1)}, \;\; (q = 1, 2, \ldots, Q)$. In the next step, we perform the similar decomposition for the new available frontal slices: $\widehat{\boldsymbol{Y}}_q^{(1)} = \boldsymbol{X}_q^{(1)} \approx \boldsymbol{A}^{(2)}\;\boldsymbol{X}_q^{(2)}$, using the same or different update rules. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrarily many times until some stopping criteria are satisfied. In each step, we usually obtain gradual improvements of the performance. Thus, our FCA model has the following form: $\boldsymbol{Y}_q \approx \boldsymbol{A}^{(1)}\boldsymbol{A}^{(2)} \cdots \boldsymbol{A}^{(L)}\boldsymbol{X}_q^{(L)}, \;\; (q = 1, 2, \ldots, Q)$ with the final components $\boldsymbol{A} = \boldsymbol{A}^{(1)}\boldsymbol{A}^{(2)} \cdots \boldsymbol{A}^{(L)}$ and $\boldsymbol{X}_q = \boldsymbol{X}_q^{(L)}$.

Physically, this means that we build up a system that has many layers or cascade connections of L mixing subsystems. The key point in our approach is that the learning (update) process to find the matrices $\boldsymbol{X}_q^{(l)}$ and $\boldsymbol{A}^{(l)}$ is performed sequentially, i.e. layer by layer, where each layer is initialized randomly. In fact, we found that the hierarchical multi-layer approach plays a key role, and it is necessary to apply in order to achieve satisfactory performance for the proposed algorithms.

4 Simulation Results

The algorithms presented in this paper have been tested for many difficult benchmarks for signals and images with various temporal and morphological properties of signals and additive noise. Due to space limitation we present here only one illustrative example. The sparse nonnegative signals with different sparsity and smoothness profiles are collected in with 10 slices $\boldsymbol{X}_q$ $(Q = 10)$ under the form of the tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{5\times1000\times10}$. The observed (mixed) 3D data $\underline{\boldsymbol{Y}} \in \mathbb{R}^{4\times1000\times10}$ are obtained by multiplying the randomly generated mixing matrix $\boldsymbol{A} \in \mathbb{R}^{4\times5}$ by $\underline{\boldsymbol{X}}$. The simulation results are illustrated in Fig. 2 (only for one frontal slice $q = 1$).

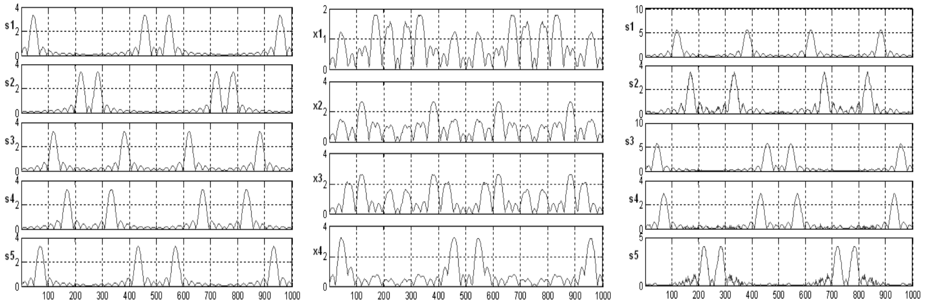

Fig. 2. (left) Original 5 spectra (representing $\boldsymbol{X}_1$); (middle) observed 4 mixed spectra $\boldsymbol{Y}_1$ generated by random matrix $\boldsymbol{A} \in \mathbb{R}^{4\times 5}$ (under-determined case); (right) estimated 5 spectra $\widehat{\boldsymbol{X}}_1$ with our algorithm given by (17)–(18), using 10 layers, and $\alpha_A = \alpha_{X_1}^{(j)} = 0.05$. Signal-to-Interference Ratios (SIR) for $\boldsymbol{A}$ and $\boldsymbol{X}_1$ are as follows: $SIR_A = 31.6, 34, 31.5, 29.9, 23.1$[dB] and $SIR_{X_1} = 28.5, 19.1, 29.3, 20.3, 23.2$[dB], respectively

5 Conclusions and Discussion

The main objective and motivations of this paper is to derive simple algorithms which are suitable both for under-determined (over-complete) and over-determined cases. We have applied the simple local cost functions with flexible penalty or regularization terms, which allows us to derive a family of robust FCA algorithms, where the sources may have different temporal and morphological properties or different sparsity profiles. Exploiting these properties and *a priori* knowledge about the character of the sources we have proposed a family of efficient algorithms for estimating sparse, smooth, and/or nonnegative sources, even if the number of sensors is smaller than the number of hidden components, under the assumption that the some information about morphological or desired properties of the sources is accessible.

This is an original extension of the standard MoCA and NMF/NTF algorithms, and to the authors' best knowledge, the first time such algorithms have been applied to the multi-way PARAFAC models. In comparison to the ordinary BSS algorithms, the proposed algorithms are shown to be superior in terms of the performance, speed, and convergence properties. We implemented the discussed algorithms in MATLAB [16]. The approach can be extended for other applications, such as dynamic MRI imaging, and it can be used as an alternative or improved reconstruction method to: the k-t BLAST, k-t SENSE or k-t SPARSE, because our approach relaxes the problem of getting stuck to in local minima, and provides usually better performance than the standard FOCUSS algorithms.

This research is motivated by potential applications of the proposed models and algorithms in three areas of the data analysis (especially, EEG and fMRI) and signal/image processing: (i) multi-way blind source separation, (ii) model reduction and selection, and (iii) sparse image coding. Our preliminary experimental results are promising. The models can be further extended by imposing additional, natural constraints such as orthogonality, continuity, closure, unimodality, local rank - selectivity, and/or by taking into account a prior knowledge about the specific components.

References

1. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis: Applications in the Chemical Sciences. John Wiley and Sons, New York (2004)
2. Hazan, T., Polak, S., Shashua, A.: Sparse image coding using a 3D non-negative tensor factorization. In: International Conference of Computer Vision (ICCV), pp. 50–57 (2005)
3. Heiler, M., Schnoerr, C.: Controlling sparseness in non-negative tensor factorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 56–67. Springer, Heidelberg (2006)
4. Miwakeichi, F., Martnez-Montes, E., Valds-Sosa, P., Nishiyama, N., Mizuhara, H., Yamaguchi, Y.: Decomposing EEG data into space−time−frequency components using parallel factor analysi. NeuroImage 22, 1035–1045 (2004)
5. Mørup, M., Hansen, L.K., Herrmann, C.S., Parnas, J., Arnfred, S.M.: Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. NeuroImage 29, 938–947 (2006)
6. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature 401, 788–791 (1999)
7. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing (New revised and improved edition). John Wiley, New York (2003)
8. Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada, pp. 283–290 (2005)
9. Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (in press, 2006)
10. Bobin, J., Starck, J.L., Fadili, J., Moudden, Y., Donoho, D.L.: Morphological component analysis: An adaptive thresholding strategy. IEEE Transactions on Image Processing (in print, 2007)
11. Elad, M.: Why simple shrinkage is still relevant for redundant representations? IEEE Trans. On Information Theory 52, 5559–5569 (2006)
12. Kovac, A.: Smooth functions and local extreme values. Computational Statistics and Data Analysis 51, 5155–5171 (2007)
13. Tao, T., Vidakovic, B.: Almost everywhere behavior of general wavelet shrinkage operators. Applied and Computational Harmonic Analysis 9, 72–82 (2000)
14. Daubechies, I., Defrrise, M., Mol, C.D.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Pure and Applied Mathematics 57, 1413–1457 (2004)
15. Cichocki, A., Zdunek, R.: Multilayer nonnegative matrix factorization. Electronics Letters 42, 947–948 (2006)
16. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)