

Novel Multi-layer Non-negative Tensor Factorization with Sparsity Constraints

Andrzej CICHOCKI^{1*}, Rafal ZDUNEK^{1**}, Seungjin CHOI,
Robert PLEMMONS², and Shun-ichi AMARI¹

¹ RIKEN Brain Science Institute, Wako-shi, Japan

{a.cichocki@riken.jp}, <http://www.bsp.brain.riken.jp>

² POSTECH, Korea, <http://www.postech.ac.kr/> seungjin

³ Dept. of Mathematics and Computer Science, Wake Forest University, USA

{plemmons@wfu.edu} <http://www.wfu.edu/> plemmons

Abstract. In this paper we present a new method of 3D non-negative tensor factorization (NTF) that is robust in the presence of noise and has many potential applications, including multi-way blind source separation (BSS), multi-sensory or multi-dimensional data analysis, and sparse image coding. We consider alpha- and beta-divergences as error (cost) functions and derive three different algorithms: (1) multiplicative updating; (2) fixed point alternating least squares (FPALS); (3) alternating interior-point gradient (AIPG) algorithm. We also incorporate these algorithms into multilayer networks. Experimental results confirm the very useful behavior of our multilayer 3D NTF algorithms with multi-start initializations.

1 Models and Problem Formulation

Tensors (also known as n-way arrays or multidimensional arrays) are used in a variety of applications ranging from neuroscience and psychometrics to chemometrics [1–4]. Nonnegative matrix factorization (NMF), Non-negative tensor factorization (NTF), parallel factor analysis PARAFAC and TUCKER models with non-negativity constraints have been recently proposed as promising sparse and quite efficient representations of signals, images, or general data [1–14]. From a viewpoint of data analysis, NTF is very attractive because it takes into account spacial and temporal correlations between variables more accurately than 2D matrix factorizations, such as NMF, and it provides usually sparse common factors or hidden (latent) components with physical or physiological meaning and interpretation [4]. One of our motivation is to develop flexible NTF algorithms which can be applied in neuroscience (analysis of EEG, fMRI) [8, 15, 16].

The basic 3D NTF model considered in this paper is illustrated in Fig. 1 (see also [9]). A given tensor $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$ is decomposed as a set of matrices $\mathbf{A} \in$

* On leave from Warsaw University of Technology, Dept. of EE, Warsaw, Poland

** On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wrocław University of Technology, Poland

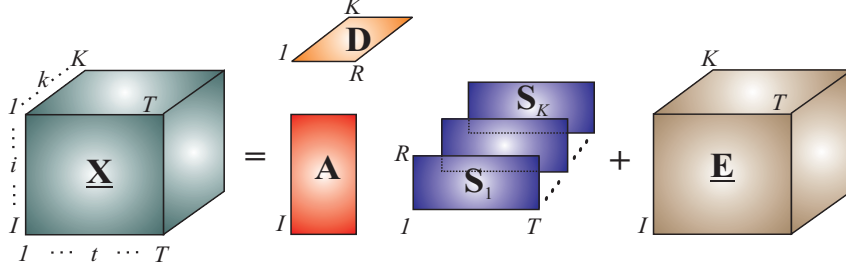


Fig. 1. NTF model that decomposes approximately tensor $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$ to set of nonnegative matrices $\mathbf{A} = [a_{ir}] \in \mathbb{R}_+^{I \times R}$, $\mathbf{D} \in \mathbb{R}^{K \times R}$ and $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_K\}$, $\mathbf{S}_k = [s_{rtk}] \in \mathbb{R}_+^{R \times T}$, $\underline{\mathbf{E}} \in \mathbb{R}^{I \times T \times K}$ is a tensor representing errors.

$\mathbb{R}_+^{I \times R}$, $\mathbf{D} \in \mathbb{R}_+^{K \times R}$ and the 3D tensor with the frontal slices $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_K\}$ with nonnegative entries. Here and elsewhere, \mathbb{R}_+ denotes the nonnegative orthant with appropriate dimensions. The three-way NTF model is given by

$$\mathbf{X}_k = \mathbf{A} \mathbf{D}_k \mathbf{S}_k + \mathbf{E}_k, \quad (k = 1, 2, \dots, K) \quad (1)$$

where $\mathbf{X}_k = \mathbf{X}_{::,k} \in \mathbb{R}_+^{I \times T}$ are the frontal slices of $\underline{\mathbf{X}} \in \mathbb{R}_+^{I \times T \times K}$, K is a number of vertical slices, $\mathbf{A} = [a_{ir}] \in \mathbb{R}_+^{I \times R}$ is the basis (mixing matrix) representing common factors, $\mathbf{D}_k \in \mathbb{R}_+^{R \times R}$ is a diagonal matrix that holds the k -th row of the matrix $\mathbf{D} \in \mathbb{R}_+^{K \times R}$ in its main diagonal, and $\mathbf{S}_k = [s_{rtk}] \in \mathbb{R}_+^{R \times T}$ are matrices representing sources (or hidden components), and $\mathbf{E}_k = \mathbf{E}_{::,k} \in \mathbb{R}^{I \times T}$ is the k -th vertical slice of the tensor $\underline{\mathbf{E}} \in \mathbb{R}^{I \times T \times K}$ representing errors or noise depending upon the application. Typically, for BSS problems $T \gg I \geq K > R$. The objective is to estimate the set of matrices \mathbf{A} , \mathbf{D} and $\{\mathbf{S}_1, \dots, \mathbf{S}_K\}$ subject to some non-negativity constraints and other possible natural constraints such as sparseness and/or smoothness on the basis of only $\underline{\mathbf{X}}$. Since the diagonal matrices \mathbf{D}_k are scaling matrices, they can usually be absorbed by the matrices \mathbf{S}_k by introducing row-normalized matrices $\tilde{\mathbf{S}}_k := \mathbf{D}_k \mathbf{S}_k$, hence $\mathbf{X}_k = \mathbf{A} \tilde{\mathbf{S}}_k + \mathbf{E}_k$. Thus in BSS applications the matrix \mathbf{A} and the set of scaled source matrices $\tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{S}}_K$ need only to be estimated. Throughout this paper, we use the following notation: the ir -th element of the matrix \mathbf{A} is denoted by a_{ir} , $x_{itk} = [\mathbf{X}_k]_{it}$ means the it -th element of the k -th frontal slice \mathbf{X}_k , $s_{rtk} = [\mathbf{S}_k]_{rt}$, $\tilde{\mathbf{S}} = [\tilde{\mathbf{S}}_1, \tilde{\mathbf{S}}_2, \dots, \tilde{\mathbf{S}}_K] \in \mathbb{R}_+^{R \times K T}$ is a row-wise unfolded matrix of the slices $\tilde{\mathbf{S}}_k$, analogously, $\tilde{\mathbf{X}} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_K] \in \mathbb{R}_+^{I \times K T}$ is a row-wise unfolded matrix of the slices \mathbf{X}_k and $\tilde{x}_{ip} = [\tilde{\mathbf{X}}]_{ip}$, $\tilde{s}_{rt} = [\tilde{\mathbf{S}}]_{rt}$.

2 Cost Functions and Associated NTF Algorithms

To deal with the factorization problem (1) efficiently we adopt several approaches from constrained optimization and multi-criteria optimization, where we minimize simultaneously several cost functions using alternating switching between

sets of parameters [5, 6, 11–13]. Alpha and Beta divergences are two complimentary cost functions [6, 7, 17]. Both divergences build up a wide class of generalized cost functions which can be applied for NMF and NTF [7, 8].

2.1 NTF Algorithms Using α -Divergence

Let us consider a general class of cost functions, called α -divergence [17, 6]:

$$D^{(\alpha)}(\bar{\mathbf{X}}||\mathbf{A}\bar{\mathbf{S}}) = \frac{1}{\alpha(\alpha-1)} \sum_{ip} (\bar{x}_{ip}^\alpha [\mathbf{A}\bar{\mathbf{S}}]_{ip}^{1-\alpha} - \alpha \bar{x}_{ip} + (\alpha-1)[\mathbf{A}\bar{\mathbf{S}}]_{ip}) \quad (2)$$

$$D_k^{(\alpha)}(\mathbf{X}_k||\mathbf{A}\mathbf{S}_k) = \frac{1}{\alpha(\alpha-1)} \sum_{itk} (x_{itk}^\alpha [\mathbf{A}\mathbf{S}_k]_{it}^{1-\alpha} - \alpha x_{itk} + (\alpha-1)[\mathbf{A}\mathbf{S}_k]_{it}). \quad (3)$$

We note that as special cases of the α -divergence for $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger's and Neyman's chi-square distances, respectively, while for the cases $\alpha = 1$ and $\alpha = 0$ the divergence has to be defined by the limits: $\alpha \rightarrow 1$ and $\alpha \rightarrow 0$, respectively. When these limits are evaluated one obtains for $\alpha \rightarrow 1$ the generalized Kullback-Leibler divergence (I-divergence) and for $\alpha \rightarrow 0$ the dual generalized KL divergence [6–8].

Instead of applying the standard gradient descent method, we use the nonlinearly transformed gradient descent approach which can be considered as a generalization of the exponentiated gradient (EG):

$$\Phi(s_{rtk}) \leftarrow \Phi(s_{rtk}) - \eta_{rtk} \frac{\partial D_A(\mathbf{X}_k||\mathbf{A}\mathbf{S}_k)}{\partial \Phi(s_{rtk})}, \quad \Phi(a_{ir}) \leftarrow \Phi(a_{ir}) - \eta_{ir} \frac{\partial D_A(\bar{\mathbf{X}}||\mathbf{A}\bar{\mathbf{S}})}{\partial \Phi(a_{ir})},$$

where $\Phi(x)$ is a suitably chosen function.

It can be shown that such a nonlinear scaling or transformation provides a stable solution and the gradients are much better behaved in the Φ space. In our case, we employ $\Phi(x) = x^\alpha$ (for $\alpha \neq 0$) and choose the learning rates as follows

$$\eta_{rtk} = \alpha^2 \Phi(s_{rtk}) / (s_{rtk}^{1-\alpha} \sum_{i=1}^I a_{ir}), \quad \eta_{ir} = \alpha^2 \Phi(a_{ir}) / (a_{ir}^{1-\alpha} \sum_{p=1}^{KT} \bar{s}_{rp}), \quad (4)$$

which leads directly to the new learning algorithm ⁴: (the rigorous proof of local convergence similar to this given by Lee and Seung [13] is omitted due to a lack of space):

$$s_{rtk} \leftarrow s_{rtk} \left(\frac{\sum_{i=1}^I a_{ir} (x_{itk}/[\mathbf{A}\mathbf{S}_k]_{it})^\alpha}{\sum_{q=1}^I a_{qr}} \right)^{1/\alpha}, \quad (5)$$

$$a_{ir} \leftarrow a_{ir} \left(\frac{\sum_{p=1}^{KT} (\bar{x}_{ip}/[\mathbf{A}\bar{\mathbf{S}}]_{ip})^\alpha \bar{s}_{rp}}{\sum_{q=1}^{KT} \bar{s}_{rq}} \right)^{1/\alpha}. \quad (6)$$

⁴ For $\alpha = 0$ instead of $\Phi(x) = x^\alpha$, we have used $\Phi(x) = \ln(x)$, which leads to a generalized SMART algorithm: $s_{rtk} \leftarrow s_{rtk} \prod_{i=1}^I (x_{itk}/[\mathbf{A}\mathbf{S}_k]_{it})^{\eta_{r a_{ir}}}$ and $a_{ir} \leftarrow a_{ir} \prod_{p=1}^{KT} (\bar{x}_{ip}/[\mathbf{A}\bar{\mathbf{S}}]_{ip})^{\tilde{\eta}_{r \bar{s}_{rp}}}$ [7].

The sparsity constraints are achieved via suitable nonlinear transformation in the form $s_{rtk} \leftarrow (s_{rtk})^{1+\gamma}$ where γ is a small coefficient [6].

2.2 NTF Algorithms using β -Divergence

The β -divergence can be considered as a general complimentary cost function to α -divergence defined above [6, 7]. Regularized β -divergences for the NTF problem can be defined as follows:

$$D^{(\beta)}(\bar{\mathbf{X}}\|\mathbf{A}\bar{\mathbf{S}}) = \sum_{ip} \left(\bar{x}_{ip} \frac{\bar{x}_{ip}^\beta - [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta}{\beta(\beta+1)} + [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta \frac{[\mathbf{A}\bar{\mathbf{S}}]_{ip} - \bar{x}_{ip}}{\beta+1} \right) + \alpha_A \|\mathbf{A}\|_{L1}, \quad (7)$$

$$D_k^{(\beta)}(\mathbf{X}_k\|\mathbf{A}\mathbf{S}_k) = \sum_{it} \left(x_{itk} \frac{x_{itk}^\beta - [\mathbf{A}\mathbf{S}_k]_{it}^\beta}{\beta(\beta+1)} + [\mathbf{A}\mathbf{S}_k]_{it}^\beta \frac{[\mathbf{A}\mathbf{S}_k]_{it} - x_{itk}}{\beta+1} \right) + \alpha_{S_k} \|\mathbf{S}_k\|_{L1}, \quad (8)$$

for $i = 1, \dots, I$, $t = 1, \dots, T$, $k = 1, \dots, K$, $p = 1, \dots, KT$, where α_{S_k} and α_A are small positive regularization parameters which control the degree of sparseness of the matrices \mathbf{S} and \mathbf{A} , respectively, and the $L1$ -norms defined as $\|\mathbf{A}\|_{L1} = \sum_{ir} |a_{ir}| = \sum_{ir} a_{ir}$ and $\|\mathbf{S}_k\|_{L1} = \sum_{rt} |s_{rtk}| = \sum_{rt} s_{rtk}$ are introduced to enforce a sparse representation of the solution. It is interesting to note that in the special case for $\beta = 1$ and $\alpha_A = \alpha_{S_k} = 0$, we obtain the square Euclidean distance expressed by the Frobenius norm $\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2$, while for the singular cases, $\beta = 0$ and $\beta = -1$, the unregularized β -divergence has to be defined as limiting cases as $\beta \rightarrow 0$ and $\beta \rightarrow -1$, respectively. When these limits are evaluated one gets for $\beta \rightarrow 0$ the generalized Kullback-Leibler divergence (I-divergence) and for $\beta \rightarrow -1$ we obtain the Itakura-Saito distance.

The choice of the β parameter depends on a statistical distribution of the data and the β -divergence corresponds to the Tweedie models [17]. For example, the optimal choice of the parameter for the normal distribution is $\beta = 1$, for the gamma distribution is $\beta \rightarrow -1$, for the Poisson distribution $\beta \rightarrow 0$, and for the compound Poisson $\beta \in (-1, 0)$. By minimizing the above formulated β -divergences, we can derive various kinds of NTF algorithms: Multiplicative based on the standard gradient descent, Exponentiated Gradient (EG), Projected Gradient (PG), Alternating Interior-Point Gradient (AIPG), or Fixed Point (FP) algorithms. By using the standard gradient descent, we obtain the multiplicative update rules:

$$s_{rtk} \leftarrow s_{rtk} \frac{[\sum_{i=1}^I a_{ir} (x_{itk}/[\mathbf{A}\mathbf{S}_k]_{it}^{1-\beta}) - \alpha_{S_k}]_\varepsilon}{\sum_{i=1}^I a_{ir} [\mathbf{A}\mathbf{S}_k]_{it}^\beta}, \quad (9)$$

$$a_{ir} \leftarrow a_{ir} \frac{[\sum_{p=1}^{KT} (\bar{x}_{ip}/[\mathbf{A}\bar{\mathbf{S}}]_{ip}^{1-\beta}) \bar{s}_{rp} - \alpha_A]_\varepsilon}{\sum_{p=1}^{KT} [\mathbf{A}\bar{\mathbf{S}}]_{ip}^\beta \bar{s}_{rp}}, \quad (10)$$

where the half-wave rectification defined as $[x]_\varepsilon = \max\{\varepsilon, x\}$ with a positive small $\varepsilon = 10^{-16}$ is introduced in order to avoid zero and negative values.

In the special case, for $\beta = 1$, we can derive a new alternative algorithm referred as, FPALS (Fixed Point Alternating Least Squares) algorithm [8]:

$$\mathbf{S}_k \leftarrow \left[(\mathbf{A}^T \mathbf{A} + \gamma_A \mathbf{E})^+ (\mathbf{A}^T \mathbf{X}_k - \alpha_{S_k} \mathbf{E}_S) \right]_{\varepsilon}, \quad (11)$$

$$\mathbf{A} \leftarrow \left[(\bar{\mathbf{X}} \bar{\mathbf{S}}^T - \alpha_A \mathbf{E}_A) (\bar{\mathbf{S}} \bar{\mathbf{S}}^T + \gamma_S \mathbf{E})^+ \right]_{\varepsilon}, \quad (12)$$

where \mathbf{A}^+ denotes Moore-Penrose pseudo-inverse, $\mathbf{E} \in \mathbb{R}^{R \times R}$, $\mathbf{E}_S \in \mathbb{R}^{R \times T}$ and $\mathbf{E}_A \in \mathbb{R}^{I \times R}$ are matrices with all ones and the function $[\mathbf{X}]_{\varepsilon} = \max\{\varepsilon, \mathbf{X}\}$ is componentwise. The above algorithm can be considered as a nonlinear projected Alternating Least Squares (ALS) or nonlinear extension of the EM-PCA algorithm⁵.

Furthermore, using the Alternating Interior-Point Gradient (AIPG) approach [18], another new efficient algorithm has been derived:

$$\mathbf{S}_k \leftarrow \mathbf{S}_k - \eta_{S_k} \mathbf{P}_{S_k}, \quad \mathbf{P}_{S_k} = \left(\mathbf{S}_k \oslash (\mathbf{A}^T \mathbf{A} \mathbf{S}_k) \right) \odot \left(\mathbf{A}^T (\mathbf{A} \mathbf{S}_k - \mathbf{X}_k) \right), \quad (13)$$

$$\mathbf{A} \leftarrow \mathbf{A} - \eta_A \mathbf{P}_A, \quad \mathbf{P}_A = \left(\mathbf{A} \oslash (\mathbf{A} \bar{\mathbf{S}} \bar{\mathbf{S}}^T) \right) \odot \left((\mathbf{A} \bar{\mathbf{S}} - \bar{\mathbf{X}}) \bar{\mathbf{S}}^T \right), \quad (14)$$

where the operators \odot and \oslash mean component-wise multiplication and division, respectively. The learning rates η_{S_k} and η_A are selected in this way to ensure the steepest descent, and on the other hand, to maintain non-negativity. Thus, $\eta_{S_k} = \min\{\tau \hat{\eta}_{S_k}, \eta_{S_k}^*\}$ and $\eta_A = \min\{\tau \hat{\eta}_A, \eta_A^*\}$, where $\tau \in (0, 1)$, $\hat{\eta}_{S_k} = \{\eta : \mathbf{S}_k - \eta \mathbf{P}_{S_k}\}$ and $\hat{\eta}_A = \{\eta : \mathbf{A} - \eta \mathbf{P}_A\}$ ensure non-negativity, and

$$\eta_{S_k}^* = \frac{\text{vec}(\mathbf{P}_{S_k})^T \text{vec}(\mathbf{A}^T \mathbf{A} \mathbf{S}_k - \mathbf{A}^T \mathbf{X}_k)}{\text{vec}(\mathbf{A} \mathbf{P}_{S_k})^T \text{vec}(\mathbf{A} \mathbf{P}_{S_k})}, \quad \eta_A^* = \frac{\text{vec}(\mathbf{P}_A)^T \text{vec}(\mathbf{A} \bar{\mathbf{S}} \bar{\mathbf{S}}^T - \bar{\mathbf{X}} \bar{\mathbf{S}}^T)}{\text{vec}(\mathbf{P}_A \bar{\mathbf{S}})^T \text{vec}(\mathbf{P}_A \bar{\mathbf{S}})}$$

are the adaptive steepest descent learning rates [8].

3 Multi-layer NTF

In order to improve the performance of all the developed NTF algorithms, especially for ill-conditioned and badly scaled data and also to reduce risk of getting stuck in local minima in non-convex alternating minimization computations, we have developed a simple hierarchical and multi-stage procedure combined together with multi-start initializations, in which we perform a sequential decomposition of nonnegative matrices as follows. In the first step, we perform the basic decomposition (factorization) $\mathbf{X}_k = \mathbf{A}^{(1)} \mathbf{S}_k^{(1)}$ using any available NTF algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition: $\mathbf{S}_k^{(1)} = \mathbf{A}^{(2)} \mathbf{S}_k^{(2)}$ using the same or

⁵ In order to drive the modified FPALS algorithm, we have used the following regularized cost functions: $\|\mathbf{X}_k - \mathbf{A} \mathbf{S}_k\|_F^2 + \alpha_{S_k} \|\mathbf{S}_k\|_{L_1} + \gamma_S \text{tr}\{\mathbf{S}_k^T \mathbf{E} \mathbf{S}_k\}$ and $\|\bar{\mathbf{X}} - \mathbf{A} \bar{\mathbf{S}}\|_F^2 + \alpha_A \|\mathbf{A}\|_{L_1} + \gamma_A \text{tr}\{\mathbf{A} \mathbf{E} \mathbf{A}^T\}$, where γ_S, γ_A are nonnegative regularization coefficients imposing some kinds of smoothness and sparsity.

different update rules, and so on. 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 NTF model has the form: $\mathbf{X}_k = \mathbf{A}^{(1)} \mathbf{A}^{(2)} \dots \mathbf{A}^{(L)} \mathbf{S}_k^{(L)}$, with the basis nonnegative matrix defined as $\mathbf{A} = \mathbf{A}^{(1)} \mathbf{A}^{(2)} \dots \mathbf{A}^{(L)}$. 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 novel approach is that the learning (update) process to find parameters of sub-matrices $\mathbf{S}_k^{(l)}$ and $\mathbf{A}^{(l)}$ is performed sequentially, i.e. layer by layer. This can be expressed by the following procedure [7, 8, 19]:

Outline Multilayer NTF Algorithm

Initialize randomly $\mathbf{A}^{(l)}$ and/or $\mathbf{S}_k^{(l)}$ and perform the alternating minimization till convergence:

$$\mathbf{S}_k^{(l)} \leftarrow \arg \min_{\mathbf{S}_k^{(l)} \geq 0} \left\{ D_k \left(\mathbf{S}_k^{(l-1)} \parallel \mathbf{A}^{(l)} \mathbf{S}_k^{(l)} \right) \right\}, \quad k = 1, \dots, K, \quad \bar{\mathbf{S}}^{(l)} = [\mathbf{S}_1^{(l)}, \dots, \mathbf{S}_K^{(l)}],$$

$$\mathbf{A}^{(l)} \leftarrow \arg \min_{\mathbf{A}^{(l)} \geq 0} \left\{ \tilde{D} \left(\bar{\mathbf{S}}^{(l-1)} \parallel \mathbf{A}^{(l)} \bar{\mathbf{S}}^{(l)} \right) \right\}, \quad [\mathbf{A}^{(l)}]_{ir} \leftarrow \left[a_{ir} / \sum_{i=1}^I a_{ir} \right]^{(l)},$$

$$\mathbf{S}_k = \mathbf{S}_k^{(L)}, \quad \mathbf{A} = \mathbf{A}^{(1)} \dots \mathbf{A}^{(L)},$$

where D_k and \tilde{D} are the cost functions (not necessary identical) used for estimation of \mathbf{S}_k and \mathbf{A} , respectively.

An open theoretical issue is to prove mathematically or explain more rigorously why the multilayer distributed NTF system with multi-start initializations results in considerable improvement in performance and reduces the risk of getting stuck in local minima. An intuitive explanation is as follows: the multilayer system provides a sparse distributed representation of basis matrices $\mathbf{A}^{(l)}$, so even a true basis matrix \mathbf{A} is not sparse it can be represented by a product of sparse factors. In each layer we force (or encourage) a sparse representation. We found by extensive experiments that if the true basis matrix is sparse, most standard NTF/NMF algorithms have improved performance (see next section). However, in practice not all data provides a sufficiently sparse representation, so the main idea is to model any data by cascade connections of sparse subsystems. On the other hand, such multilayer systems are biologically motivated and plausible.

4 Simulation Results

All the NMF algorithms presented in this paper have been extensively tested for many difficult benchmarks for signals and images with various statistical distributions of signals and additive noise, and also for preliminary tests with real EEG data. Due to space limitations we present here only comparison of proposed algorithms for a typical benchmark. The simulations results shown in Table 1 have been performed for the synthetic benchmark in which the nonnegative weakly statistically dependent 100 hidden components or sources (spectra) are

collected in 20 slices $\mathbf{S}_k \in \mathbb{R}_+^{5 \times 1000}$, each representing 5 different kind of spectra. The sources have been mixed by the common random matrix $\mathbf{A} \in \mathbb{R}_+^{10 \times 5}$ with a uniform distribution. In this way, we obtained the 3D tensor $\underline{\mathbf{X}} \in \mathbb{R}^{10 \times 1000 \times 20}$ of overlapped spectra. Table 1 shows the averaged SIR (standard signal to interference ratio) performance obtained from averaging the results from 100 runs of the Monte Carlo (MC) analysis for recovering of the original spectra \mathbf{S}_k and the mixing matrix \mathbf{A} for various algorithms and for different number of layers 1–5. (Usually, it assumed that $SIR \geq 20dB$ provides a quite good performance, and over $30dB$ excellent performance.) We have also applied and tested the

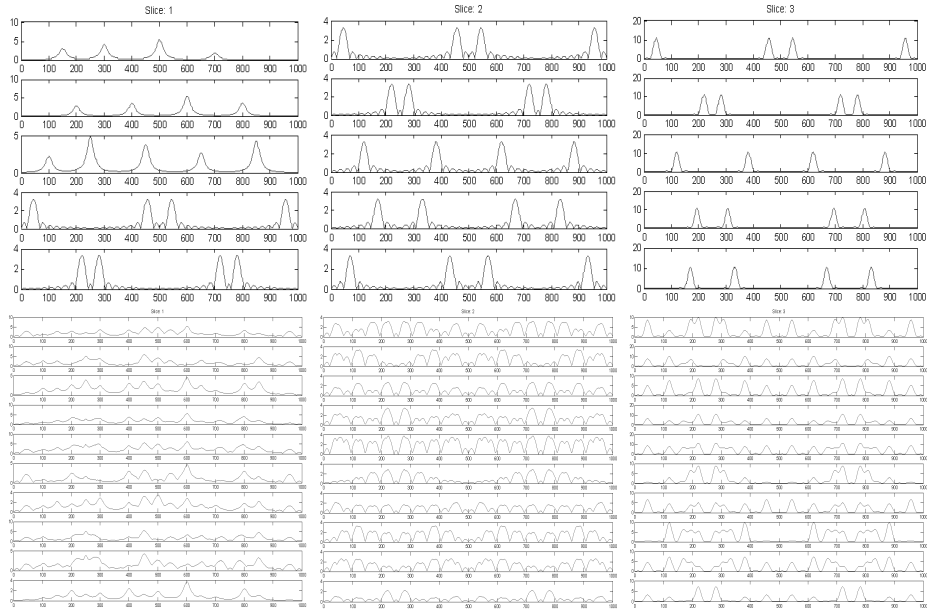


Fig. 2. Selected slices of: (top) the original spectra signals (top); mixed signals with dense mixing matrix $\mathbf{A} \in \mathbb{R}^{10 \times 5}$

developed algorithms for real-world EEG data and neuroimages. Due to space limitation these results will be presented in the conference and on our website.

5 Conclusions and Discussion

The main objective and motivation of this paper was to develop and compare leaning algorithms and compare their performance. We have extended the 3D non-negative matrix factorization (NMF) models to multi-layer models and found that the best performance is obtained with the FPALS and AIPG algorithms. With respect to the standard NTF (single layer) models, our model and

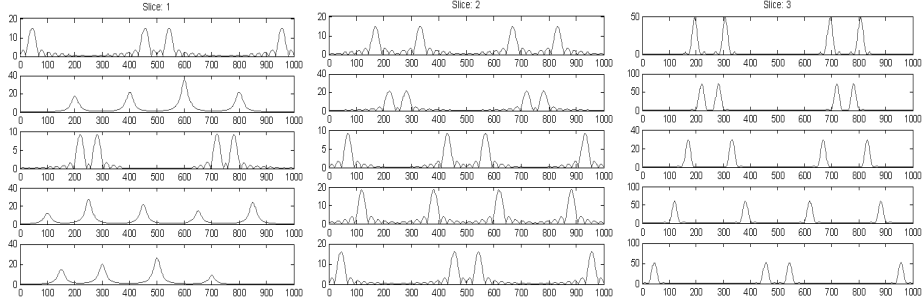


Fig. 3. Spectra signals estimated with the FPALS (11)–(12) using 3 layers for $\gamma_A = \gamma_S = \alpha_{S_k} = \alpha_A = 0$. The signals in the corresponding slices are scored with: SIRs(1-st slice) = 47.8, 53.6, 22.7, 43.3, 62; SIRs(2-nd slice) = 50, 52.7, 23.6, 42.5, 62.7; and SIRs(3-d slice) = 50.1, 55.8, 30, 46.3, 59.9; [dB], respectively.

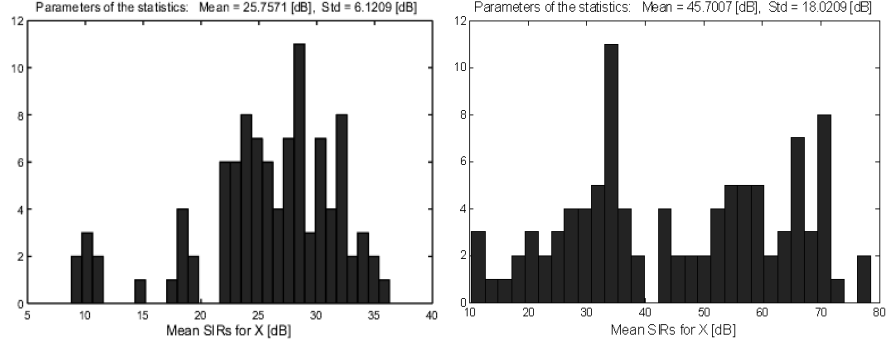


Fig. 4. Histograms of 100 mean-SIR samples from Monte Carlo analysis performed using the following algorithms with 5 layers: (left) Beta Alg. (9)–(10), $\beta = 0$; (right) FPALS (11)–(12) for $\gamma_A = \gamma_S = \alpha_{S_k} = \alpha_A = 0$.

Table 1. Mean SIRs in [dB] obtained from 100 MC samples for estimation of the columns in \mathbf{A} and the rows (sources) in \mathbf{S}_k versus the number of layers (Multi-layer technique), and for the selected algorithms.

ALGORITHMS: (Equations)	LAYERS (SIRs \mathbf{A})					LAYERS (SIRs \mathbf{S})				
	1	2	3	4	5	1	2	3	4	5
Alpha Alg. (5–6): $\alpha = 0.5$	9.1	15.6	19	21.8	24.6	7.8	13.5	16.5	18.9	21.2
Beta Alg. (9–10): $\beta = 0$	11.9	20.9	27.8	29.5	30.8	8.1	16.4	22.9	24.4	25.6
AIPG (13–14)	14	22.7	29	33.1	35.4	10.1	18	24.1	28.4	30.6
FPALS (11–12)	20.7	35	42.6	46	47.2	19.4	32.7	41.7	46.1	48.1

Table 2. Elapsed times (in seconds) for 1000 iterations with different algorithms.

No. layers	Alpha Alg. (5–6) $\alpha = 0.5$	Beta Alg. (9–10) $\beta = 0$	AIPG (13–14)	FPALS (11–12)
1	23.7	4.7	11.8	3.8
3	49.3	11.3	32.8	10.3

proposed algorithms can give much better performance or precision in factorizations and better robustness against noise. Moreover, we considered a wide class of the cost functions which allows us to derive a family of robust and efficient NTF algorithms with only single parameter to tune (α or β). The optimal choice of the parameter in the cost function depends and on a statistical distribution of data and additive noise, thus different criteria and algorithms (updating rules) should be applied for estimating the basis matrix \mathbf{A} and the source matrices \mathbf{S}_k , depending on *a priori* knowledge about the statistics of noise or errors. We found by extensive simulations that the multi-layer technique combined together with multi-start initialization plays a key role in improving the performance of blind source separation when using the NTF approach. It is worth mentioning that we can use two different strategies. In the first approach presented in details in this contribution we use two different cost functions: A global cost function (using row-wise unfolded matrices: $\bar{\mathbf{X}}$, $\bar{\mathbf{S}}$ and 2D model $\bar{\mathbf{X}} = \mathbf{A}\bar{\mathbf{S}}$) to estimate the common factors, i.e., the basis (mixing) matrix \mathbf{A} ; and local cost functions to estimate the frontal slices \mathbf{S}_k , ($k = 1, 2, \dots, K$). However, it is possible to use a different approach in which we use only a set of local cost functions, e.g., $D_k = 0.5\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2$. In such a case, we estimate \mathbf{A} and \mathbf{S}_k cyclically by applying alternating minimization (similar to row-action projection in the Kaczmarz algorithm). We found that such approach also works well for the NTF model. We have motivated the use of proposed 3D NTF in three areas of data analysis (especially, EEG and fMRI) and signal/image processing: (i) multi-way blind source separation, (ii) model reductions and selection, and (iii) sparse image coding. Our preliminary experiments are promising.

The proposed models can be further extended by imposing additional, natural constraints such as smoothness, continuity, closure, unimodality, local rank - selectivity, and/or by taking into account a prior knowledge about specific 3D, or more generally, multi-way data.

Obviously, there are many challenging open issues remaining, such as global convergence, an optimal choice of the parameters and the model.

References

1. In Lathauwer, L.D., Comon, P., eds.: Workshop on Tensor Decompositions and Applications, CIRM, Marseille, France (2005)

2. Hazan, T., Polak, S., Shashua, A.: Sparse image coding using a 3D non-negative tensor factorization. In: International Conference of Computer Vision (ICCV). (2005) 50–57
3. Heiler, M., Schnoerr, C.: Controlling sparseness in nonnegative tensor factorization. In: ECCV. (2006)
4. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis: Applications in the Chemical Sciences. John Wiley and Sons, New York (2004)
5. Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (**2006**) submitted.
6. Cichocki, A., Zdunek, R., Amari, S.: Csiszar’s divergences for non-negative matrix factorization: Family of new algorithms. LNCS **3889** (2006) 32–39
7. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended SMART algorithms for non-negative matrix factorization. LNAI **4029** (2006) 548–562
8. Cichocki, A., Zdunek, R.: NMFLAB for Signal and Image Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
9. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
10. Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada (2005)
11. Hoyer, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469
12. Kim, M., Choi, S.: Monaural music source separation: Nonnegativity, sparseness, and shift-invariance. LNCS **3889** (2006) 617–624
13. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401** (1999) 788–791
14. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces, Copenhagen, Denmark, Proc. European Conf. on Computer Vision (ECCV) (2002) 447–460
15. Morup, 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** (2006) 938–947
16. Miwakeichi, F., Martinez-Montes, E., Valds-Sosa, P.A., Nishiyama, N., Mizuhara, H., Yamaguchi, Y.: Decomposing EEG data into spacetime-frequency components using Parallel Factor Analysis. NeuroImage **22** (2004) 1035–1045
17. Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
18. Merritt, M., Zhang, Y.: An interior-point gradient method for large-scale totally nonnegative least squares problems. J. Optimization Theory and Applications **126** (2005) 191–202
19. Cichocki, A., Zdunek, R.: Multilayer nonnegative matrix factorization. Electronics Letters **42** (2006) 947–948