Regularized Alternating Least Squares Algorithms for Non-negative Matrix/Tensor Factorization

Andrzej Cichocki* and Rafal Zdunek**

Laboratory for Advanced Brain Signal Processing,
RIKEN BSI, Wako-shi, Saitama 351-0198, Japan

Abstract. Nonnegative Matrix and Tensor Factorization (NMF/NTF) and Sparse Component Analysis (SCA) have already found many potential applications, especially in multi-way Blind Source Separation (BSS), multi-dimensional data analysis, model reduction and sparse signal/image representations. In this paper we propose a family of the modified Regularized Alternating Least Squares (RALS) algorithms for NMF/NTF. By incorporating regularization and penalty terms into the weighted Frobenius norm we are able to achieve sparse and/or smooth representations of the desired solution, and to alleviate the problem of getting stuck in local minima. We implemented the RALS algorithms in our NMFLAB/NTFLAB Matlab Toolboxes, and compared them with standard NMF algorithms. The proposed algorithms are characterized by improved efficiency and convergence properties, especially for large-scale problems.

1 Introduction and Problem Formulation

Nonnegative Matrix Factorization (NMF) and its multi-way extensions: Nonnegative Tensor Factorization (NTF) and Parallel Factor analysis (PARAFAC) models with sparsity and/or non-negativity constraints have been recently proposed as promising sparse and quite efficient representations of signals, images, or general data [1,2,3,4,5,6,7,8,9,10]. From a viewpoint of data analysis, NMF/NTF provide nonnegative and usually sparse common factors or hidden (latent) components with physiological meaning and interpretation [4,7,11]. NMF, NTF and SCA are used in a variety of applications, ranging from neuroscience and psychometrics to chemometrics [1,2,7,8,9,10,11,12,13,14].

In this paper we impose nonnegativity and sparsity constraints, and possibly other natural constraints, such as smoothness for the following linear model:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}, \tag{1}$$

* On leave from Warsaw University of Technology, Dept. of EE, Warsaw, Poland.
** On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.

D. Liu et al. (Eds.): ISNN 2007, Part III, LNCS 4493, pp. 793–802, 2007.

where $\boldsymbol{Y} \in \mathbb{R}^{I \times T}$ is a matrix of the observed data or signals, $\boldsymbol{A} \in \mathbb{R}_+^{I \times R}$ is a mixing or basis matrix, $\boldsymbol{X} \in \mathbb{R}_+^{R \times T}$ represents unknown sources or hidden (nonnegative and sparse) components, and $\boldsymbol{E} \in \mathbb{R}^{I \times T}$ represents a noise or error (residuum) matrix. Usually, in BSS applications: $T >> I \geq R$, and R is known or can be estimated using SVD. Our objective is to estimate the mixing (basis) matrix $\boldsymbol{A}$ and the sources $\boldsymbol{X}$, subject to nonnegativity and sparsity constraints.

The above model can be extended to the 3D PARAFAC2 or NTF2 model in which a given tensor $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times T \times K}$ is decomposed to a set of matrices $\boldsymbol{X}$, $\boldsymbol{D}$ and $\{\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_K\}$ with nonnegative entries [9,15,16,17]. The NTF2 model can be described as

$$\boldsymbol{Y}_k = \boldsymbol{A}_k \boldsymbol{D}_k \boldsymbol{X} + \boldsymbol{E}_k, \qquad (k = 1, 2, \ldots, K) \tag{2}$$

where $\boldsymbol{Y}_k = \boldsymbol{Y}_{:,:,k} = [y_{itk}]_{I \times T} \in \mathbb{R}^{I \times T}$ are frontal slices of $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times T \times K}$, K is a number of frontal slices, $\boldsymbol{A}_k = [a_{irk}]_{I \times R} \in \mathbb{R}_+^{I \times R}$ are the basis (mixing matrices), $\boldsymbol{D}_k \in \mathbb{R}_+^{R \times R}$ is a diagonal matrix that holds the k-th row of the $\boldsymbol{D} \in \mathbb{R}_+^{K \times R}$ in its main diagonal, and $\boldsymbol{X} = [x_{rt}]_{R \times T} \in \mathbb{R}_+^{R \times T}$ is a matrix representing the sources (or hidden nonnegative components or common factors), and $\boldsymbol{E}_k = \boldsymbol{E}_{:,:,k} \in \mathbb{R}^{I \times T}$ is the k-th frontal slice of the tensor $\underline{\boldsymbol{E}} \in \mathbb{R}^{I \times T \times K}$ representing error or noise depending upon the application. The objective is to estimate the set of nonnegative matrices $\{\boldsymbol{A}_k\}, (k, \ldots, K)$, $\boldsymbol{D}$ and $\boldsymbol{X}$, subject to some non-negativity constraints and other possible natural constraints such as sparseness and/or smoothness. Since the diagonal matrices $\boldsymbol{D}_k$ are scaling matrices they can be usually absorbed by the matrices $\boldsymbol{A}_k$ by introducing the column-normalized matrices $\boldsymbol{A}_k := \boldsymbol{A}_k \boldsymbol{D}_k$, so usually in BSS applications the matrix $\boldsymbol{X}$ and the set of scaled matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K$ need only to be estimated. It should be noted that the 3D PARAFAC2 and the corresponding NTF2 models[1] can be easily transformed to a 2D non-negative matrix factorization problem by unfolding (matricizing) tensors. Such 2D models are equivalent to a standard NMF model. In fact, the 3D PARAFAC2 model can be represented as column-wise unfolding. The unfolded system can be described by a single system of the matrix equation: $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}$, where $\boldsymbol{Y} = [\boldsymbol{Y}_1; \boldsymbol{Y}_2; \ldots; \boldsymbol{Y}_K] \in \mathbb{R}^{IK \times T}$ is a column-wise (vertical) unfolded matrix of all the frontal slices $\boldsymbol{Y}_k$, $\boldsymbol{A} = [\boldsymbol{A}_1; \boldsymbol{A}_2; \ldots; \boldsymbol{A}_K] \in \mathbb{R}_+^{IK \times R}$ is a column-wise unfolded matrix of the slices $\boldsymbol{A}_k$ representing (the frontal slices), and $\boldsymbol{E} = [\boldsymbol{E}_1; \boldsymbol{E}_2; \ldots; \boldsymbol{E}_K] \in \mathbb{R}^{IK \times T}$ is a column-wise unfolded matrix of errors.

Solutions of NMF algorithms may not be unique, therefore it is often required to impose additional data-driven natural constraints, such as sparsity or smoothness. Moreover, many existing algorithms for NMF are prohibitively slow and inefficient, especially for very large-scale problems. For large-scale problems a promising approach is to apply the Alternating Least Squares (ALS) algorithm [1,8]. Unfortunately, the standard ALS algorithm and its simple modifications suffer from unstable convergence properties, giving often not optimum solution,

[1] Analogously, NTF1 model described by a set of the matrix equations $\boldsymbol{Y}_k = \boldsymbol{A}\boldsymbol{D}_k\boldsymbol{X}_k + \boldsymbol{E}_k$, $k = 1, 2, \ldots, K$, can be transformed to the standard NMF problem by row-wise unfolding.

and they are characterized by high sensitivity to near-collinear data [1,4,8,10]. The main objective of this paper is to develop efficient and robust regularized ALS (RALS) algorithms. For this purpose, we exploit several approaches from constrained optimization and regularization theory, and propose additionally several heuristic algorithms.

2 Regularized ALS Algorithms

The most of known and used adaptive algorithms for NMF are based on alternating minimization of the squared Euclidean distance expressed by the Frobenius norm: $D_F(\boldsymbol{Y}||\boldsymbol{A},\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X}\|_F^2$, subject to nonnegativity constraints of all the elements in $\boldsymbol{A}$ and $\boldsymbol{X}$. Such a cost function is optimal for a Gaussian distributed noise [12,11].

In this paper we consider minimization of more general and flexible cost function that is a regularized weighted least-squares function with sparsity penalties:

$$D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{W}^{-1/2}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})\|_F^2 + \alpha_{As}\,||\boldsymbol{A}||_{L_1} + \alpha_{Xs}\,||\boldsymbol{X}||_{L_1} + \frac{\alpha_{Ar}}{2}\,||\boldsymbol{W}^{-1/2}\boldsymbol{A}\boldsymbol{L}_A||_F^2 + \frac{\alpha_{Xr}}{2}\,||\boldsymbol{L}_X\boldsymbol{X}||_F^2, \tag{3}$$

(usually subject to additional constraints such as nonnegativity constraints) where $\boldsymbol{W} \in \mathbb{R}^{I\times I}$ is symmetric positive definite weighting matrix[2], $\alpha_{As} \geq 0$ and $\alpha_{Xs} \geq 0$ are parameters controlling a sparsity level of the matrices, and $\alpha_{Ar} \geq 0$, $\alpha_{Xr} \geq 0$ are regularization coefficients. The penalty terms $||\boldsymbol{A}||_{L_1} = \sum_{ir}|a_{ir}|$ and $||\boldsymbol{X}||_{L_1} = \sum_{rt}|x_{rt}|$ enforce sparsification in $\boldsymbol{A}$ and $\boldsymbol{X}$, respectively, and sparseness can be adjusted by α_{As} and α_{Xs}. The regularization matrices $\boldsymbol{L}_A$ and $\boldsymbol{L}_X$ are used to enforce a certain application-dependent characteristics of the solution. These matrices are typically unit diagonal matrices or discrete approximations to some derivative operator. Another option is to use the following setting: $\alpha_{Xr}\boldsymbol{L}_X^T\boldsymbol{L}_X = \boldsymbol{A}^T(\boldsymbol{I}-\boldsymbol{A}_S\boldsymbol{A}_S^T)\boldsymbol{A}$ where $\boldsymbol{A}_S$ contains the R first principal eigenvectors of the data covariance matrix $\boldsymbol{R}_Y = (\boldsymbol{Y}^T\boldsymbol{Y})/I = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^{\mathbf{T}}$ associated with the R largest singular values [2]. It is worth noting that both matrices $\boldsymbol{L}_X^T\boldsymbol{L}_X \in \mathbb{R}^{R\times R}$ and $\boldsymbol{L}_A\boldsymbol{L}_A^T \in \mathbb{R}^{R\times R}$ are in general symmetric and positive definite matrices.

The gradients of the cost function (3) with respect to the unknown matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ are expressed by

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X})}{\partial \boldsymbol{A}} = \boldsymbol{W}^{-1}(\boldsymbol{A}\boldsymbol{X}-\boldsymbol{Y})\boldsymbol{X}^T + \alpha_{As}\,\boldsymbol{S}_A + \alpha_{Ar}\,\boldsymbol{W}^{-1}\boldsymbol{A}\,\boldsymbol{L}_A\boldsymbol{L}_A^T, \tag{4}$$

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X})}{\partial \boldsymbol{X}} = \boldsymbol{A}^T\boldsymbol{W}^{-1}(\boldsymbol{A}\boldsymbol{X}-\boldsymbol{Y}) + \alpha_{Xs}\,\boldsymbol{S}_X + \alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X\,\boldsymbol{X}, \tag{5}$$

[2] $\boldsymbol{W}^{-1/2} = \boldsymbol{V}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^{\mathbf{T}}$ means in Matlab notation $\boldsymbol{W}^{-1/2} = inv(sqrtm(\boldsymbol{W}))$ and $\|\boldsymbol{W}^{-1/2}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})\|_F^2 = \mathrm{tr}\left\{(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})^T\boldsymbol{W}^{-1}(\boldsymbol{Y}-\boldsymbol{A}\boldsymbol{X})\right\}$.

where $\boldsymbol{S}_A = \text{sign}(\boldsymbol{A})$ and $\boldsymbol{S}_X = \text{sign}(\boldsymbol{X})$ [3]. In the particular case for a NMF problem the matrices $\boldsymbol{S}_A, \boldsymbol{S}_X$ will be transformed to the matrices $\bar{\boldsymbol{E}}_A$ and $\bar{\boldsymbol{E}}_X$ of the same dimension with all the entries equal to ones.

By equalizing the gradients (4)-(5) to zero, we obtain the following fixed-point regularized ALS algorithm

$$\boldsymbol{A} \leftarrow (\boldsymbol{Y}\boldsymbol{X}^T - \alpha_{As}\boldsymbol{W}\boldsymbol{S}_A)(\boldsymbol{X}\boldsymbol{X}^T + \alpha_{Ar}\,\boldsymbol{L}_A\boldsymbol{L}_A^T)^{-1} \tag{6}$$

$$\boldsymbol{X} \leftarrow (\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{A} + \alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X)^{-1}(\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{Y} - \alpha_{Xs}\,\boldsymbol{S}_X). \tag{7}$$

In order to achieve high performance, the regularization parameters $\alpha_{Ar} \geq 0$ and $\alpha_{Xr} \geq 0$ are usually not necessarily fixed but rather should be dynamically changed in time, depending how far we are from the desired solution. For example, we may gradually decrease exponentially the regularization coefficients during a convergence process. We found by computer experiments that quite a good performance for small-scale problems can be achieved by choosing $\alpha_{Ar}(k) = \alpha_{Xr}(k) = \alpha_0 \exp(-k/\tau)$ with typical values $\alpha_0 = 20$ and $\tau = 50$ and $\boldsymbol{L}_X^T\boldsymbol{L}_X \approx \bar{\boldsymbol{E}}_X$ where $\bar{\boldsymbol{E}}$ means a matrix with all ones entries[4]. For large-scale problems α_0 should be higher.

An alternative approach is to keep the regularization parameters fixed and try to compensate (reduce) their influence by additional terms as the algorithm converges to the desired solution. For this purpose let us consider the following simple approach [4,10]. It is easy to note that the equation (7) can be re-written in the equivalent form as

$$(\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{A} + \alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X)\boldsymbol{X}_{new} = \boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{Y} - \alpha_{Xs}\,\boldsymbol{S}_X \tag{8}$$

In order to compensate the regularization term $\alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X\boldsymbol{X}_{new}$ we can add to the right-hand side the similar term $\alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X\boldsymbol{X}_{old}$ which gradually compensates the regularization term when $\boldsymbol{X} \to \boldsymbol{X}^*$, i.e.

$$(\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{A} + \alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X)\boldsymbol{X}_{new} = \boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{Y} - \alpha_{Xs}\,\boldsymbol{S}_X + \alpha_{Xr}\,\boldsymbol{L}_X^T\boldsymbol{L}_X\boldsymbol{X}_{old}$$

The magnitude of the bias (or influence of the regularization term) is a function of the difference between $\boldsymbol{X}_{old}$ and $\boldsymbol{X}_{new}$. As the algorithm tends to converge to the desired solution $\boldsymbol{X}^*$, this difference is gradually decreasing, and the effect of regularization and bias is smaller and smaller.

Hence, after simple mathematical manipulations our RALS algorithm can take the following general and flexible form:

$$\boldsymbol{A} \leftarrow (\boldsymbol{Y}\boldsymbol{X}^T - \alpha_{As}\boldsymbol{W}\boldsymbol{S}_A + \alpha_{Ar}\,\boldsymbol{A}\boldsymbol{L}_A\boldsymbol{L}_A^T)(\boldsymbol{X}\boldsymbol{X}^T + \alpha_{Ar}\,\boldsymbol{L}_A\boldsymbol{L}_A^T)^+, \tag{9}$$

$$\boldsymbol{X} \leftarrow (\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{A} + \alpha_{Xr}\boldsymbol{L}_X^T\boldsymbol{L}_X)^+(\boldsymbol{A}^T\boldsymbol{W}^{-1}\boldsymbol{Y} - \alpha_{Xs}\boldsymbol{S}_X + \alpha_{Xr}\boldsymbol{L}_X^T\boldsymbol{L}_X\boldsymbol{X}), \tag{10}$$

[3] $\text{sign}(\boldsymbol{X})$ means a componentwise sign operation (or its robust approximation) for each element in $\boldsymbol{X}$.

[4] In this case, to drive the RALS algorithm rigorously, we have used the following modified regularized cost functions: $0.5\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \alpha_X\|\boldsymbol{X}\|_{L_1} + 0.5\alpha_{Xr}\,\text{tr}\{\boldsymbol{X}^T\bar{\boldsymbol{E}}\boldsymbol{X}\} + \alpha_A\|\boldsymbol{A}\|_{L_1} + 0.5\alpha_{Ar}\,\text{tr}\{\boldsymbol{A}\bar{\boldsymbol{E}}\boldsymbol{A}^T\}$.

where $\boldsymbol{A}^{+}$ means Moore-Penrose pseudo-inverse of $\boldsymbol{A}$. It should be noted that the proposed algorithm for $\boldsymbol{W} = \boldsymbol{I}$ and for all the regularization coefficients setting to zero ($\alpha_{As} = \alpha_{Ar} = \alpha_{Xs} = \alpha_{Xr} = 0$) simplifies to the standard ALS. On the other hand, if we take all the regularization parameters equal to zero and $\boldsymbol{W} = \boldsymbol{R}_E = (\boldsymbol{E}\boldsymbol{E}^T)/I$, where the error matrix $\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}$ is evaluated in each iteration step, we obtain the extended BLUE (Best Linear Unbiased Estimated) ALS algorithm. Finally, in the special case when $\boldsymbol{W} = \boldsymbol{I}$ and matrices $\boldsymbol{L}_A\boldsymbol{L}_A^T$ and $\boldsymbol{L}_X^T\boldsymbol{L}_X$ are diagonal our algorithm is similar to the MALS (modified ALS) proposed by Wang et al in [10], and Hancewicz and Wang in [4].

3 Implementation of RALS Algorithms for NMF

On the basis of the above consideration we have developed and implemented in MATLAB the following RALS algorithm for the NMF, especially suitable for large-scale problems:

Outline of the RALS algorithm for NMF

- 1a. Set the initial values of matrices $\boldsymbol{W}, \boldsymbol{L}_A, \boldsymbol{L}_X$ and parameters α_{As}, α_{Xs}, α_{Ar}, α_{Xr},
- 1b. Set the initial values of $\boldsymbol{A}$, (e.g., multi-start random initialization, or eigenvalue decomposition (SVD), ICA, or dissimilarity criteria [15,4,8],
- 2a. Calculate the new estimate of $\boldsymbol{X}_{new}$ from $\boldsymbol{Y}$ and $\boldsymbol{A}_{old}$ using iterative formula (10),(set $\boldsymbol{S}_X = \bar{\boldsymbol{E}}_X$),
- 2b. $\boldsymbol{X}_{new} = max\{\boldsymbol{X}_{new}, 0\}$ (set negative values to zero or alternatively to a small positive value, typically, $\varepsilon = 10^{-16}$). Impose additional optional natural constraints on rows of $\boldsymbol{X}$ such as low-pass filtering or smoothness,
- 3a. Calculate the new estimate of $\boldsymbol{A}_{new}$ from (9), (set $\boldsymbol{S}_A = \bar{\boldsymbol{E}}_A$),
- 3b. $\boldsymbol{A}_{new} = max\{\boldsymbol{A}_{new}, 0\}$ (set negative values to zero or to a small positive value ε). Impose some additional finite constraints such as clustering or smoothness,
- 3c. Normalize each column of $\boldsymbol{A}_{new}$ to unit length l_1-norm,
- 4. Repeat the steps (2) and (3) until convergence criterion is reached.

The above algorithm with a suboptimal set of the default parameters have been implemented in our NMFLAB and NTFLAB [15].

Further improvement of the RALS algorithm has been achieved by applying a hierarchical multi-layer system with multi-start initialization [13,15] which can be implemented as follows: In the first step, we perform the basic decomposition (factorization) $\boldsymbol{Y} = \boldsymbol{A}_1\boldsymbol{X}_1$ using the RALS algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition: $\boldsymbol{X}_1 = \boldsymbol{A}_2\boldsymbol{X}_2$ using the same or different set of parameters, and so on. We continue our factorization 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 model has the form: $\boldsymbol{Y} = \boldsymbol{A}_1\boldsymbol{A}_2 \cdots \boldsymbol{A}_L\boldsymbol{X}_L$, with the

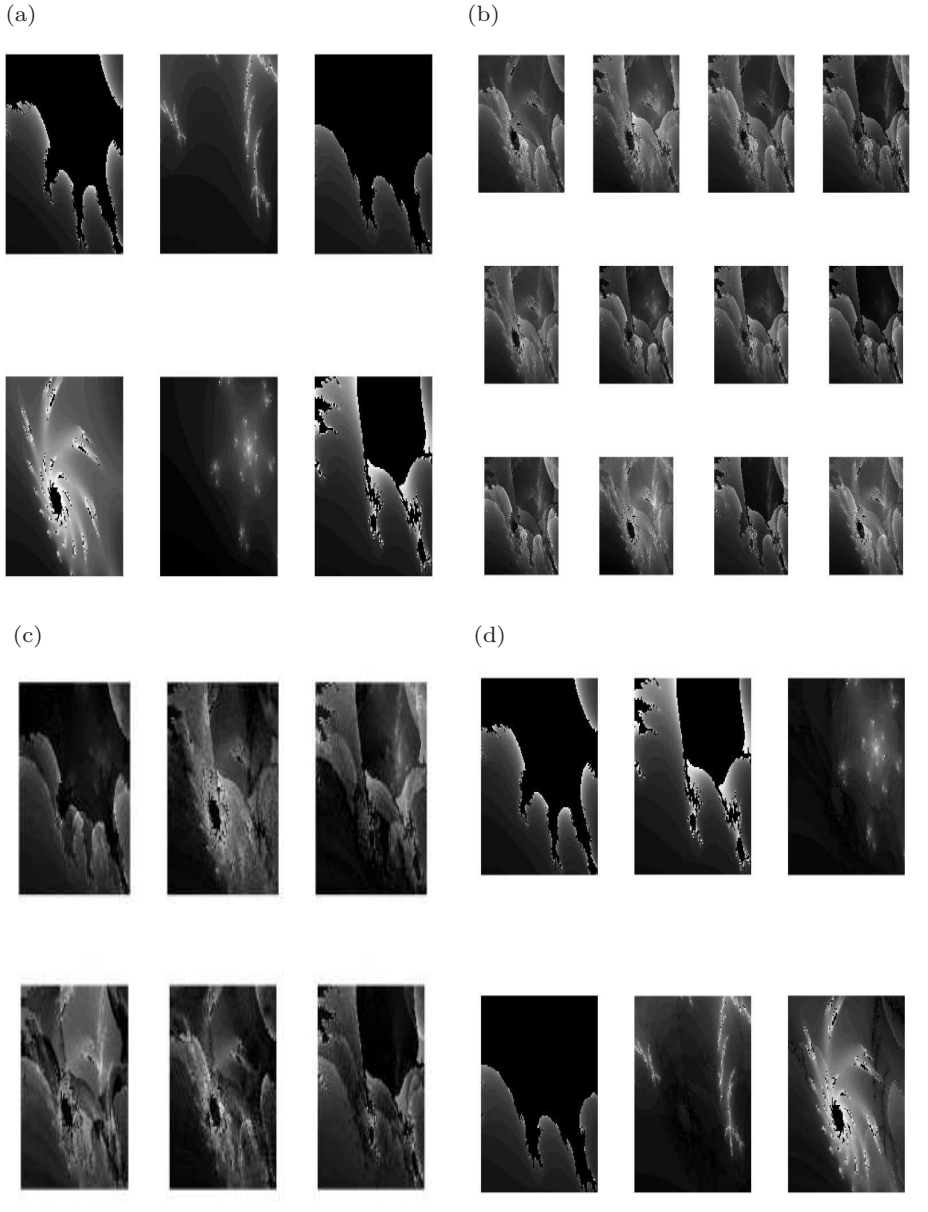

Fig. 1. Example 1: (a) Original 6 sources of Mandelbrot fractal images; (b) Observed 12 mixed images (uniformly distributed random mixing matrix); (c) Estimated sources using the standard Lee-Seung algorithm with Kullback-Leibler divergence (SIR = 6.4, 7.6, 0.2, 3.7, -0.2, 7.3 [dB], respectively); (d) Estimated source images using RALS algorithm (SIR = 50.61, 128.1, 17.6, 41, 16.6, 13.1 [dB], respectively) given by (6)–(7) with parameters: $\boldsymbol{W} = \boldsymbol{I}$ (identity), $\alpha_{Ar} = \alpha_{As} = \alpha_{Xs} = 0$, $\boldsymbol{L}_X^T \boldsymbol{L}_X = \bar{\boldsymbol{E}}$ and α_{Xr} given by the exponential rule with $\alpha_{Xr} = 20 \exp(-k/50)$.

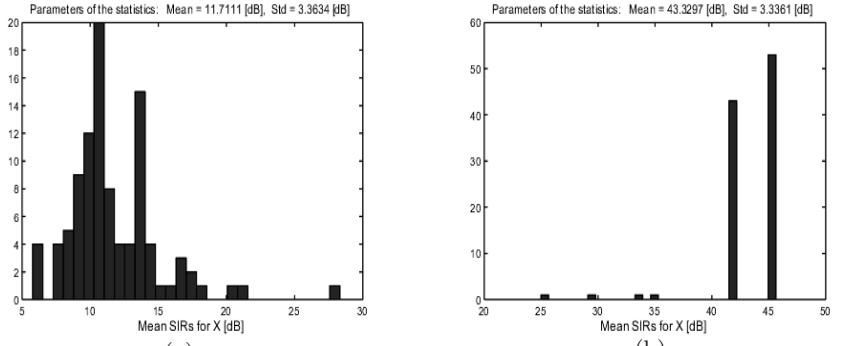

Fig. 2. Example 1: Histograms of 100 mean-SIR samples from Monte Carlo analysis performed with the algorithms: (a) standard ALS; (b) RALS with the same parameters as in Fig. 1.

basis nonnegative matrix defined as $\boldsymbol{A} = \boldsymbol{A}_1 \boldsymbol{A}_2 \cdots \boldsymbol{A}_L$. An open theoretical issue is to prove mathematically or explain more rigorously why the multilayer distributed NMF/NTF system results in considerable improvement in performance and reduces the risk of getting stuck at local minima. An intuitive explanation is as follows: the multilayer system provides a sparse distributed representation of basis matrix $\boldsymbol{A}$, which in general can be a dense matrix. So even a true basis matrix $\boldsymbol{A}$ is not sparse it can be represented by a product of sparse factors. In each layer we force (or encourage) a sparse representation. On the other hand, we found by extensive experiments that if the basis matrix is very sparse, most NTF/NMF algorithms have improved performance (see next section). However, not all real data provides sufficiently sparse representations, so the main idea is to model any data by a distributed sparse hierarchical multilayer system. It is also interesting to note that such multilayer systems are biologically motivated and plausible.

4 Simulation Results

In order to confirm validity and high performance of the proposed algorithm we extensively tested it for various sets of free parameters and compared them with standard NMF algorithms. We illustrate the performance by giving only two examples. In the first example (see Fig. 1), we used 6 images which were mixed by a uniformly distributed randomly generated mixing matrix $\boldsymbol{A} \in \mathbb{R}^{12\times 6}$. We found by the Monte Carlo analysis performed for 100 runs that our RALS algorithm (with the exponentially decaying regularization term for the matrix $\boldsymbol{X}$) significantly outperforms the standard ALS (see Fig. 2).

In the second example, the 9 sparse nonnegative signals (representing synthetic spectra) have been mixed by the mixing matrix $\boldsymbol{A} \in \mathbb{R}^{18\times 9}$. Additive Gaussian noise with SNR = 20 dB has been added. In this case the standard

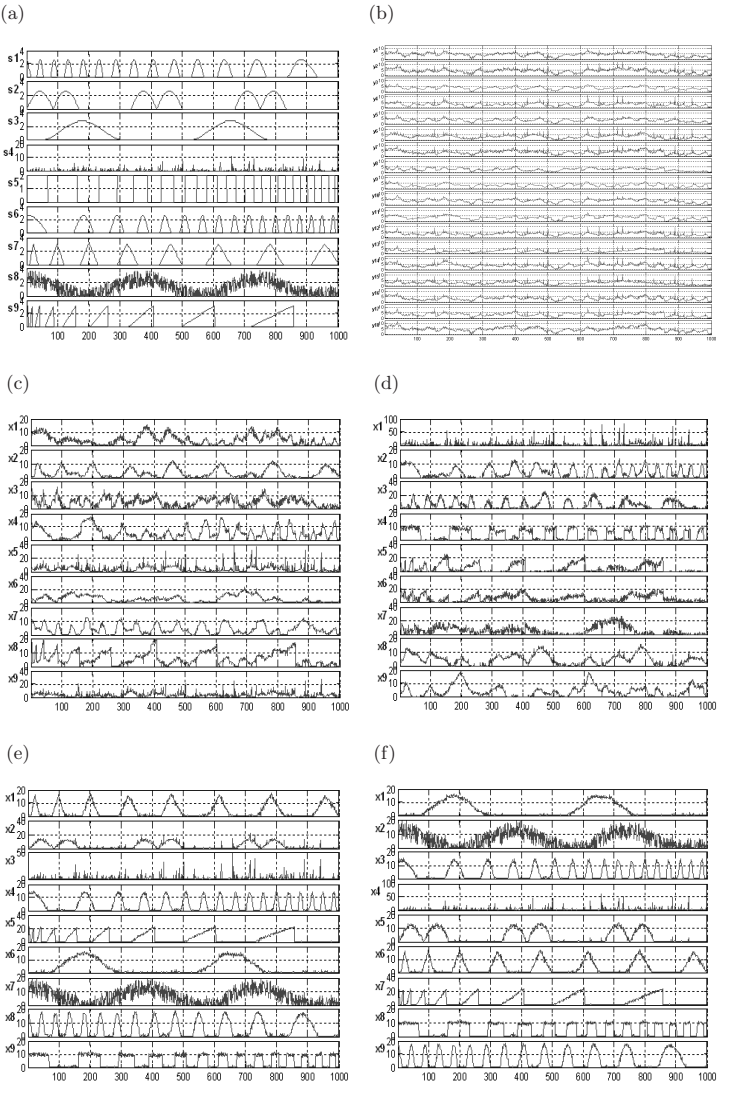

Fig. 3. Example 2: (a) Original 9 source signals; (b) Observed 18 mixed signals (uniformly distributed random mixing matrix) with SNR = 20 dB; (c) Estimated sources using the standard Lee-Seung algorithm with Kullback-Leibler divergence (SIR = 3.7, 8.2, 3.6, 6.1, 4.5, 2.9, 5.2, 5.8, 2.2 [dB], respectively) with 1 layer; (d) Estimated sources using the standard Lee-Seung algorithm with Kullback-Leibler divergence (SIR = 6.9, 6.6, 6.7, 18.2, 14, 8.7, 7.6, 5.8, 15.9 [dB], respectively) with 3 layers; (e) Estimated source images using the RALS algorithm (SIR = 18.2, 12.2, 21.1, 20.7, 22.5, 19.1, 21.3, 19.9 [dB], respectively) given by (9)–(10) with 1 layer and for the following parameters: $\boldsymbol{W} = \boldsymbol{R}_E$, $\alpha_{Ar} = \alpha_{As} = 0$, $\alpha_{Xs} = \alpha_{Xr} = 0.1$, $\boldsymbol{L}_X^T \boldsymbol{L}_X = \boldsymbol{I}$; (f) Estimated source images using the same RALS algorithm (SIR = 19.4, 17.4, 21.5, 22.6, 17.9, 18.5, 22.2, 21.6, 22.2 [dB], respectively) with 3 layers.

NMF algorithms completely failed to estimate the original sources while the RALS algorithm successfully estimates all the sources. We observed a considerable improvement in performance applying the multilayer procedure with 10 initializations in each layer.

We also performed the test on large-scale problems, increasing the number of observations in the second example to 1000. The mixing matrix $\boldsymbol{A} \in \mathbb{R}^{1000\times 9}$ was randomly chosen. For such a case we got the elapsed times and mean-SIRs given in Table 1. It should be noted than the ISRA and EMML algorithms (which are the basic Lee-Seung multiplicative algorithms that minimize the Frobenius norm and Kullback-Leibler divergence, respectively) failed to estimate the original components.

Table 1. Performance of the NMF algorithms for a large-scale problem with 1000 observations and 9 nonnegative components

Algorithm	Elapsed time [s]	Mean-SIRs [dB]
RALS	16.6	$SIR > 43$
ISRA	36	$SIR < 10$
EMML	81	$SIR < 16$

5 Conclusions and Discussion

In this paper we proposed the generalized and flexible cost function (controlled by sparsity penalty and flexible multiple regularization terms) that allows us to derive a family of robust and efficient alternating least squares algorithms for NMF and NTF. We proposed the method which allows us to automatically self-regulate or self-compensate the regularization terms. This is a unique modification of the standard ALS algorithm and to the authors' best knowledge, the first time this type of constraints has been combined together with the ALS algorithm for applications to NMF and NTF. The performance of the proposed algorithm is compared with the ordinary ALS algorithm for NMF. The proposed algorithm is shown to be superior in terms of performance, component resolution ability, speed and convergence properties, and ability to be used for large-scale problems. The proposed RALS algorithm may be also promising for other applications, such as Sparse Component Analysis and EM Factor Analysis because it overcomes the problems associated with ALS, i.e. the solution of RALS tends not to get trapped in local minima and will generally converges to the global desired solution.

References

1. 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.
2. Cichocki, A., Amari, S.: Adaptive Blind Signal And Image Processing (New revised and improved edition). John Wiley, New York (2003)

3. Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada (2005)
4. Hancewicz, T.M., Wang, J.H.: Discriminant image resolution: a novel multivariate image analysis method utilizing a spatial classification constraint in addition to bilinear nonnegativity. Chemometrics and Intelligent Laboratory Systems **77** (2005) 18–31
5. Heiler, M., Schnoerr, C.: Controlling sparseness in non-negative tensor factorization. Springer LNCS **3951** (2006) 56–67
6. Hoyer, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469
7. 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
8. Albright, R., Cox, J., Duling, D., Langville, A.N., Meyer, C.D.: Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical report, NCSU Technical Report Math 81706 (2006) submitted.
9. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis: Applications in the Chemical Sciences. John Wiley and Sons, New York (2004)
10. Wang, J.H., Hopke, P.K., Hancewicz, T.M., Zhang, S.L.: Application of modified alternating least squares regression to spectroscopic image analysis. Analytica Chimica Acta **476** (2003) 93–109
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401** (1999) 788–791
12. Cichocki, A., Zdunek, R., Amari, S.: Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. Springer LNCS **3889** (2006) 32–39
13. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended SMART algorithms for non-negative matrix factorization. Springer LNAI **4029** (2006) 548–562
14. Kim, M., Choi, S.: Monaural music source separation: Nonnegativity, sparseness, and shift-invariance. Springer LNCS **3889** (2006) 617–624
15. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
16. Cichocki, Zdunek, R., Choi, S., Plemmons, R., Amari, S.: Novel multi-layer nonnegative tensor factorization with sparsity constraints. In: Proc. 8-th International Conference on Adaptive and Natural Computing Algorithms, Warsaw, Poland (2007)
17. Cichocki, Zdunek, R., Choi, S., Plemmons, R., Amari, S.: Nonnegative tensor factorization using alpha and beta divergencies. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP07), Honolulu, Hawaii, USA (2007)