

Non-Negative Matrix Factorization with Quasi-Newton Optimization

Rafal ZDUNEK*, Andrzej CICHOCKI**

Laboratory for Advanced Brain Signal Processing
BSI, RIKEN, Wako-shi, JAPAN

Abstract. Non-negative matrix factorization (NMF) is an emerging method with wide spectrum of potential applications in data analysis, feature extraction and blind source separation. Currently, most applications use relative simple multiplicative NMF learning algorithms which were proposed by Lee and Seung, and are based on minimization of the Kullback-Leibler divergence and Frobenius norm. Unfortunately, these algorithms are relatively slow and often need a few thousands of iterations to achieve a local minimum. In order to increase a convergence rate and to improve performance of NMF, we proposed to use a more general cost function: so-called Amari alpha divergence. Taking into account a special structure of the Hessian of this cost function, we derived a relatively simple second-order quasi-Newton method for NMF. The validity and performance of the proposed algorithm has been extensively tested for blind source separation problems, both for signals and images. The performance of the developed NMF algorithm is illustrated for separation of statistically dependent signals and images from their linear mixtures.

1 Introduction and Problem Formulation

Non-negative matrix factorization (NMF) [1, 2, 3, 4, 5] decomposes the data matrix $\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(K)] \in \mathbb{R}^{M \times K}$ as a product of two matrices $\mathbf{A} \in \mathbb{R}^{M \times R}$ and $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K)] \in \mathbb{R}^{R \times K}$ having only non-negative elements. Although some decompositions or matrix factorizations provide an exact reconstruction data (i.e., $\mathbf{Y} = \mathbf{AX}$), we shall consider here decompositions which are approximative in nature, i.e.,

$$\mathbf{Y} = \mathbf{AX} + \mathbf{V}, \quad \mathbf{A} \geq 0, \quad \mathbf{X} \geq 0 \quad (1)$$

or equivalently $\mathbf{y}(k) = \mathbf{Ax}(k) + \mathbf{v}(k)$, $k = 1, 2, \dots, K$ or in a scalar form as $y_m(k) = \sum_{r=1}^R a_{mr}x_r(k) + \nu_m(k)$, $m = 1, \dots, M$, where $\mathbf{V} \in \mathbb{R}^{M \times K}$ represents noise or error matrix, $\mathbf{y}(k) = [y_1(k), \dots, y_M(k)]^T$ is a vector of the observed signals (typically nonnegative) at the discrete time instants¹ k while

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

** On leave from Warsaw University of Technology, POLAND

¹ The data are often represented not in the time domain but in a transform domain such as the time frequency domain, so index k may have different meaning.

$\mathbf{x}(k) = [x_1(k), \dots, x_R(k)]^T$ is a vector of components or source signals at the same time instant [6]. Our objective is to estimate the mixing (basis) matrix \mathbf{A} and sources \mathbf{X} subject to nonnegativity constraints all entries. Usually, in Blind Source Separation (BSS), to which NMF is applied in this paper, $K \gg M \geq R$ and R is known or can be relatively easily estimated using SVD or PCA. Through this paper, we use the following notations: $x_r(k) = x_{rk}$, $y_m(k) = y_{mk}$ and $z_{mk} = [\mathbf{A}\mathbf{X}]_{mk}$ means mk -element of the matrix $(\mathbf{A}\mathbf{X})$, the mr -th element of the matrix \mathbf{A} is denoted by a_{mr} .

The basic approach to NMF is alternating minimization or alternating projection: the specified loss function is alternately minimized with respect to two sets of the parameters $\{x_{rk}\}$ and $\{a_{mr}\}$, each time optimizing one set of arguments while keeping the other one fixed [7, 2, 6].

One of the NMF algorithms, which was proposed by Lee and Seung [2], alternatively minimizes the Kulback-Leibler (KL) divergence

$$D_{KL}(\mathbf{A}\mathbf{X}||\mathbf{Y}) = \sum_{mk} \left(y_{mk} \log \frac{y_{mk}}{[\mathbf{A}\mathbf{X}]_{mk}} + [\mathbf{A}\mathbf{X}]_{mk} - y_{mk} \right) \quad (2)$$

s. t. $x_{rk} \geq 0$, $a_{mr} \geq 0$, $\|\mathbf{a}_r\|_1 = \sum_{m=1}^M a_{mr} = 1$.

with multiplicative update rules based on a gradient descent approach [6]. This leads to the following algorithm

$$x_{rk} \leftarrow x_{rk} \frac{\sum_{m=1}^M a_{mr} (y_{mk}/[\mathbf{A}\mathbf{X}]_{mk})}{\sum_{q=1}^M a_{qr}}, \quad (3)$$

$$a_{mr} \leftarrow a_{mr} \frac{\sum_{k=1}^K x_{rk} (y_{mk}/[\mathbf{A}\mathbf{X}]_{mk})}{\sum_{p=1}^K x_{rp}}. \quad (4)$$

This algorithm extends (by alternating minimization) the well-known EMML or Richardson-Lucy algorithm (RLA) [8]. Another Lee-Seung algorithm minimizes the square Euclidean distance (Frobenius norm) with the same alternating approach.

The multiplicative descent algorithms are known to be very slowly-convergent and easily stuck in local minima. One of the way to speed up the convergence is to modify the learning rate in an iterative scheme. In this paper, we address this issue with second-order approximations of the loss function, i.e. with the quasi-Newton method.

2 Quasi-Newton Optimization

The KL divergence (2) is a particular case of the Amari alpha-divergence [9, 10, 11] defined as

$$D_A(\mathbf{A}\mathbf{X}||\mathbf{Y}) = \sum_{mk} y_{mk} \frac{(y_{mk}/z_{mk})^{\alpha-1} - 1}{\alpha(\alpha-1)} + \frac{z_{mk} - y_{mk}}{\alpha}, \quad z_{mk} = [\mathbf{A}\mathbf{X}]_{mk} \quad (5)$$

This case takes place if $\alpha \rightarrow 1$, and for $\alpha \rightarrow 0$ the dual KL can be derived. For $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger and Neyman's chi-square distances, respectively.

Applying the quasi-Newton method to (5), we have

$$\mathbf{X} \leftarrow \left[\mathbf{X} - [\mathbf{H}_{D_A}^{(\mathbf{X})}]^{-1} \nabla_{\mathbf{X}} D_A \right]_{\epsilon}, \quad \mathbf{A} \leftarrow \left[\mathbf{A} - [\mathbf{H}_{D_A}^{(\mathbf{A})}]^{-1} \nabla_{\mathbf{A}} D_A \right]_{\epsilon}, \quad (6)$$

where $\mathbf{H}_{D_A}^{(\mathbf{X})}$ and $\mathbf{H}_{D_A}^{(\mathbf{A})}$ are Hessians, $\nabla_{\mathbf{X}} D_A$ and $\nabla_{\mathbf{A}} D_A$ are gradients matrices for (5) with respect to \mathbf{X} and \mathbf{A} , respectively. The nonlinear operator $[\cdot]_{\epsilon} = \max\{\cdot, \epsilon\}$ enforces nonnegativity.

The gradients with respect to \mathbf{X} can be expressed as

$$\mathbf{G}_{D_A}^{(\mathbf{X})} = \nabla_{\mathbf{X}} D_A = \frac{1}{\alpha} \mathbf{A}^T (1 - (\mathbf{Y} ./ (\mathbf{A}\mathbf{X}))^{\alpha}) \in \mathbb{R}^{R \times K}, \quad (7)$$

where $./$ is a Hadamard division. The Hessian has the form: $\forall i \in \{1, \dots, R\}, j \in \{1, \dots, K\}$:

$$[\mathbf{H}_{D_A}^{(\mathbf{X})}]_{ij} = \frac{\partial^2 D_A}{\partial x_{rk} \partial x_{ij}} = \begin{cases} \sum_{m=1}^M \frac{a_{mr} y_{mk}^{\alpha} a_{mi}}{(\sum_{s=1}^R a_{ms} x_{sk})^{\alpha+1}}, & \text{for } j = k, i = s, \\ 0, & \text{otherwise} \end{cases}, \quad (8)$$

or in a block matrix

$$\mathbf{H}_{D_A}^{(\mathbf{X})} = \frac{1}{\alpha} \text{diag}\{[\mathbf{h}_k^{(\mathbf{X})}]_{k=1, \dots, K}\} \in \mathbb{R}^{RK \times RK} \quad (9)$$

where

$$\mathbf{h}_k^{(\mathbf{X})} = \mathbf{A}^T \text{diag}\{[\mathbf{Y}^{\alpha} ./ (\mathbf{A}\mathbf{X})^{\alpha+1}]_{*,k}\} \mathbf{A} \in \mathbb{R}^{R \times R}$$

Similarly for \mathbf{A} , we get

$$\mathbf{G}_{D_A}^{(\mathbf{A})} = \nabla_{\mathbf{A}} D_A = \frac{1}{\alpha} (1 - (\mathbf{Y} ./ (\mathbf{A}\mathbf{X}))^{\alpha}) \mathbf{X}^T \in \mathbb{R}^{M \times R}. \quad (10)$$

The Hessian has the form: $\forall i \in \{1, \dots, M\}, j \in \{1, \dots, R\}$:

$$[\mathbf{H}_{D_A}^{(\mathbf{A})}]_{ij} = \frac{\partial^2 D_A}{\partial a_{mr} \partial a_{ij}} = \begin{cases} \sum_{k=1}^K \frac{x_{rk} y_{mk}^{\alpha} x_{jk}}{(\sum_{s=1}^R a_{ms} x_{sk})^{\alpha+1}}, & \text{for } j = s, i = m, \\ 0, & \text{otherwise} \end{cases}, \quad (11)$$

or in a block matrix

$$\mathbf{H}_{D_A}^{(\mathbf{A})} = \frac{1}{\alpha} \text{diag}\{[\mathbf{h}_m^{(\mathbf{A})}]_{m=1, \dots, M}\} \in \mathbb{R}^{MR \times MR} \quad (12)$$

where

$$\mathbf{h}_m^{(\mathbf{A})} = \mathbf{X} \text{diag}\{[\mathbf{Y}^{\alpha} ./ (\mathbf{A}\mathbf{X})^{\alpha+1}]_{m,*}\} \mathbf{X}^T \in \mathbb{R}^{R \times R}$$

Since the Hessian is usually ill-conditioned, especially if we have sparse representations of the image to be estimated, some regularization of the Hessian is essential, which leads to a quasi-Newton iterations. We applied the Levenberg-Marquardt approach with a small fixed regularization parameter $\lambda = 10^{-12}$. Additionally we control the convergence by a slight relaxation of the iterative updates. To reduce substantially a computational cost, the inversion of the Hessian is replaced with the Q-less QR factorization computed with LAPACK. Thus the final form of the algorithm with the quasi-Newton algorithm is

$$\begin{aligned} \mathbf{X} &\leftarrow [\mathbf{X} - \gamma \mathbf{R}_{\mathbf{X}} \setminus \mathbf{W}_{\mathbf{X}}]_{\epsilon}, & \mathbf{A} &\leftarrow [\mathbf{A} - \gamma \mathbf{R}_{\mathbf{A}} \setminus \mathbf{W}_{\mathbf{A}}]_{\epsilon}, \\ \mathbf{W}_{\mathbf{X}} &= \mathbf{Q}_{\mathbf{X}}^T \nabla_{\mathbf{X}} D_A, & \mathbf{Q}_{\mathbf{X}} \mathbf{R}_{\mathbf{X}} &= \mathbf{H}_{D_A}^{(\mathbf{X})} + \lambda \mathbf{I}_{\mathbf{X}}, \\ \mathbf{W}_{\mathbf{A}} &= \mathbf{Q}_{\mathbf{A}}^T \nabla_{\mathbf{A}} D_A, & \mathbf{Q}_{\mathbf{A}} \mathbf{R}_{\mathbf{A}} &= \mathbf{H}_{D_A}^{(\mathbf{A})} + \lambda \mathbf{I}_{\mathbf{A}}, \end{aligned} \quad (13)$$

where $\mathbf{I}_{\mathbf{X}} \in \mathbb{R}^{RK \times RK}$, $\mathbf{I}_{\mathbf{A}} \in \mathbb{R}^{MR \times MR}$ are identity matrices, $\mathbf{R}_{\mathbf{X}}$ and $\mathbf{R}_{\mathbf{A}}$ are upper triangular matrices, and γ controls the relaxation. We set $\gamma = 0.9$. The \setminus in (13) means the Gaussian elimination.

For $\alpha \rightarrow 0$, the Amari alpha-divergence converges to the dual I-divergence (generalized K-L divergence), i.e.

$$\begin{aligned} D_{KL2}(\mathbf{Y} \parallel \mathbf{A}\mathbf{X}) &= \lim_{\alpha \rightarrow 0} D_A(\mathbf{A}\mathbf{X} \parallel \mathbf{Y}) \\ &= \sum_{mk} \left(z_{mk} \log \frac{z_{mk}}{y_{mk}} + y_{mk} - z_{mk} \right), \quad z_{mk} = [\mathbf{A}\mathbf{X}]_{mk}, \end{aligned} \quad (14)$$

and consequently the gradient and Hessian matrices simplify as follows:

– For \mathbf{X} :

$$\mathbf{G}_{D_{KL2}}^{(\mathbf{X})} = \nabla_{\mathbf{X}} D_{KL2} = \mathbf{A}^T \log((\mathbf{A}\mathbf{X}) ./ \mathbf{Y}) \in \mathbb{R}^{R \times K}, \quad (15)$$

and

$$\mathbf{H}_{D_{KL2}}^{(\mathbf{X})} = \text{diag}\{[\mathbf{h}_k^{(\mathbf{X})}]_{k=1, \dots, K}\} \in \mathbb{R}^{RK \times RK}, \quad (16)$$

where

$$\mathbf{h}_k^{(\mathbf{X})} = \mathbf{A}^T \text{diag}\{[1./(\mathbf{A}\mathbf{X})]_{*,k}\} \mathbf{A} \in \mathbb{R}^{R \times R}.$$

– For \mathbf{A} :

$$\mathbf{G}_{D_{KL2}}^{(\mathbf{A})} = \nabla_{\mathbf{A}} D_{KL2} = \log((\mathbf{A}\mathbf{X}) ./ \mathbf{Y}) \mathbf{X}^T \in \mathbb{R}^{M \times R}, \quad (17)$$

and

$$\mathbf{H}_{D_{KL2}}^{(\mathbf{A})} = \text{diag}\{[\mathbf{h}_m^{(\mathbf{A})}]_{m=1, \dots, M}\} \in \mathbb{R}^{MR \times MR}, \quad (18)$$

where

$$\mathbf{h}_m^{(\mathbf{A})} = \mathbf{X} \text{diag}\{[1./(\mathbf{A}\mathbf{X})]_{m,*}\} \mathbf{X}^T \in \mathbb{R}^{R \times R}.$$

In each alternating step, the l_1 -norm of the columns of \mathbf{A} are normalized to a unity, i.e. we have: $a_{mr} \leftarrow \frac{a_{mr}}{\sum_{m=1}^M a_{mr}}$.

3 Fixed-Point Algorithm

In our application, \mathbf{X} has much larger dimensions than \mathbf{A} , and hence, the computation of \mathbf{X} with the Newton method may be highly time-consuming or even intractable, even though the Hessian is very sparse. Let us assume some typical case: $M = 20$, $R = 10$, and $K = 1000$. Thus the Hessian $\mathbf{H}^{(\mathbf{A})}$ has size 200 by 200 with $MR^2 = 2 \times 10^3$ non-zero entries, but the size of $\mathbf{H}^{(\mathbf{X})}$ is 10^4 by 10^4 with $KR^2 = 10^5$ non-zero entries. For this reason, we do not apply the Newton method for updating \mathbf{X} . This can be also justified by the fact that the computation of \mathbf{A} needs to solve the system which is much more over-determined than for \mathbf{X} , and hence, this may be better done with the second order method since the information about the curvature of the cost function is exploited. In our approach, we apply the Newton method to the generalized cost function (Amari alpha-divergence).

In this paper, the sources \mathbf{X} are basically estimated with the EMLL and Fixed-Point (FP) algorithms.

In general, the FP algorithm solves a least-squares problem

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \right\} \quad (19)$$

with the Moore-Penrose pseudo-inverse of a system matrix, i.e. in our case, the matrix \mathbf{A} . Since in NMF $M \geq R$, we formulate normal equations as $\mathbf{A}^T \mathbf{A}\mathbf{X} = \mathbf{A}^T \mathbf{Y}$, and the least-squares solution of minimal l_2 -norm to the normal equations is $\mathbf{X}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y} = \mathbf{A}^+ \mathbf{Y}$, where \mathbf{A}^+ is the Moore-Penrose pseudo-inverse of \mathbf{A} . The cost function given by the square Euclidean distance as in (19) works the best with Gaussian noise (matrix \mathbf{V} in (1)), however, the computation of \mathbf{A} uses the Amari alpha-divergence which is optimal for a wide spectrum of signal distributions.

The computation of \mathbf{X} is usually improved with the prior knowledge about the source representations, such as sparsity and/or smoothing. The information about a structure of the estimated sources is usually incorporated to the cost function in the form of the additional term that regularizes the solution. Thus, the cost function in (19) is extended to the regularized squares Euclidean distance, and the problem to be solved becomes:

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \alpha_{\mathbf{X}} \Omega(\mathbf{X}) \right\}, \quad (20)$$

where $\Omega(\mathbf{X})$ is a regularization function, and $\alpha_{\mathbf{X}} (\geq)$ is a regularization parameter. The minimal-norm least-square solution to (20) is given by:

$$\mathbf{X}_{LS} = (\mathbf{A}^T \mathbf{A} + \alpha_{\mathbf{X}} \mathbf{C})^{-1} \mathbf{A}^T \mathbf{Y}, \quad (21)$$

where $\mathbf{C} \in \mathbb{R}^{R \times R}$ is some discrete representation of the regularization term $\Omega(\mathbf{X})$.

There are many possibilities for defining $\Omega(\mathbf{X})$. For example, we have the basic Tikhonov regularization for $\Omega(\mathbf{X}) = \|\mathbf{X}\|_F^2$, which leads to $\mathbf{C} = \mathbf{I}_R$, where

$\mathbf{I}_R \in \mathbb{R}^{R \times R}$ is an identity matrix. This operator enforces a smooth solution. In many applications, $\Omega(\mathbf{X})$ is a first or second derivative of the solution, or it is given by the Total Variation (TV) term. For more regularization operators, e.g. see [12, 13, 14].

Due to sparse solutions in NMF, we introduce some new approach, i.e. we assume that $\mathbf{C} = \mathbf{E} \in \mathbb{R}^{R \times R}$, where \mathbf{E} is a matrix composed from all ones entries. The regularization parameter is set according to the exponential rule, i.e.

$$\alpha_{\mathbf{X}} = \alpha_{\mathbf{X}}^{(k)} = \alpha_0 \exp\{-\tau k\}, \quad (22)$$

where k is a number of alternating steps. This rule is motivated by a temperature schedule in the simulated annealing that steers the solution towards a global one. Thus, larger parameter α_0 and smaller τ should give better results but at the cost of high increase in a number of alternating steps. In our simulations, we set $\alpha_0 = 20$ and $\tau = 0.02$ for 1000 alternating steps.

Another simple approach that can be used for controlling sparsity of estimated variables is to apply nonlinear projections with suitable nonlinear monotonic functions. In this paper, the EMMML updates are modified by a very simple nonlinear transformation $x_{\tau k} \leftarrow (x_{\tau k})^{1+\alpha_{sX}}$, $\forall k$, where α_{sX} is a small coefficient, typically from 0.001 to 0.005, and it is positive or negative to increase or decrease the sparseness, respectively.

Since the Newton method does not ensure a nonnegative solution, we enforce a nonnegative solution through a very simple nonlinear projection as in (6), which is applied to both sets of the arguments (\mathbf{X} and \mathbf{A}). The same nonlinear projection is also applied to (21). Moreover, since \mathbf{E} is singular, and $\mathbf{A}^T \mathbf{A}$ may be very ill-conditioned, especially for sparse solutions, the inversion in (21) is done with the Moore-Penrose pseudo-inverse instead of the standard one. Thus the updating of \mathbf{X} in the $(k+1)$ -th alternating step is performed with the novel algorithm:

$$\mathbf{X}^{(k+1)} \leftarrow \max \left\{ \varepsilon, ([\mathbf{A}^T \mathbf{A}]^{(k)} + \alpha_{\mathbf{X}}^{(k)} \mathbf{E}) + [\mathbf{A}^T]^{(k)} \mathbf{Y} \right\}, \quad (23)$$

where $\mathbf{A}^{(k)}$ is the update of \mathbf{A} from the k -th alternating step.

4 Results

The proposed algorithms have been extensively tested for many benchmarks of signals and images. The original 4 images [Fig. 1(a)] and the original 9 signals [Fig. 2 (a)] have been mixed with uniform distributions, where $\mathbf{A}^{(i)} \in \mathbb{R}^{9 \times 4}$ and $\mathbf{A}^{(s)} \in \mathbb{R}^{18 \times 9}$ are dense mixing matrices for the images and signals, respectively. The mixtures are shown in Figs. 1 (b) and 2 (b). The results obtained with the traditional Lee-Seung algorithm (3) and (4), which has been applied to estimation of both \mathbf{A} and \mathbf{X} , are presented in Figs. 1 (c) and 2 (c). The separations are quantified with Signal-to-Interference Ratios (SIRs) that have the following

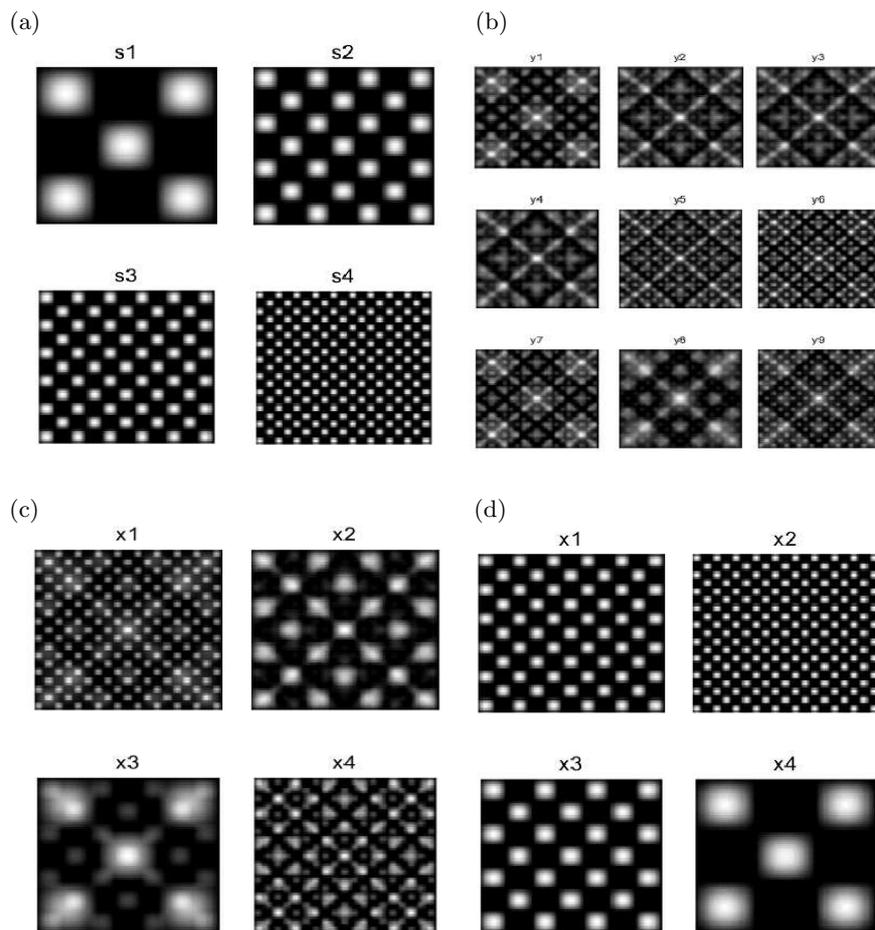


Fig. 1. Example 1: (a) Original 4 source images; (b) observed 9 mixed images; (c) Estimated source images using the standard Lee-Seung algorithm for KL function (2) (SIR = 5.5dB, 12.5dB, 9dB, 6dB, respectively); (d) Estimated source images using the new EMML-Newton algorithm for $\alpha = 2$ with nonlinear projection $\alpha_{sX} = 0.002$ with SIR=47dB, 45dB, 50dB, 44dB, respectively.

values: 5.5, 12.5, 9, 6 [dB] for the images, and 1, 7.6, 5.9, 4.9, 4.4, 9.8, 5.9, 10.7, 3.7 [dB] for the signals. Applying the quasi-Newton method only to estimation of \mathbf{A} , and the nonlinear transformation with $\alpha_{sX} = 0.002$ to the same Lee-Seung algorithm (EMML) we obtained much better results which are shown in Fig. 1 (d). The SIRs are as follows: 47, 45, 50, 44 [dB]. The similar performance can be obtained for signals. Furthermore, the results can be even much better if the sources \mathbf{X} are estimated with the Fixed-Point algorithm – see Fig. 2 (d). For this case, we have all the SIRs above 110 [dB], which is nearly a perfect result.

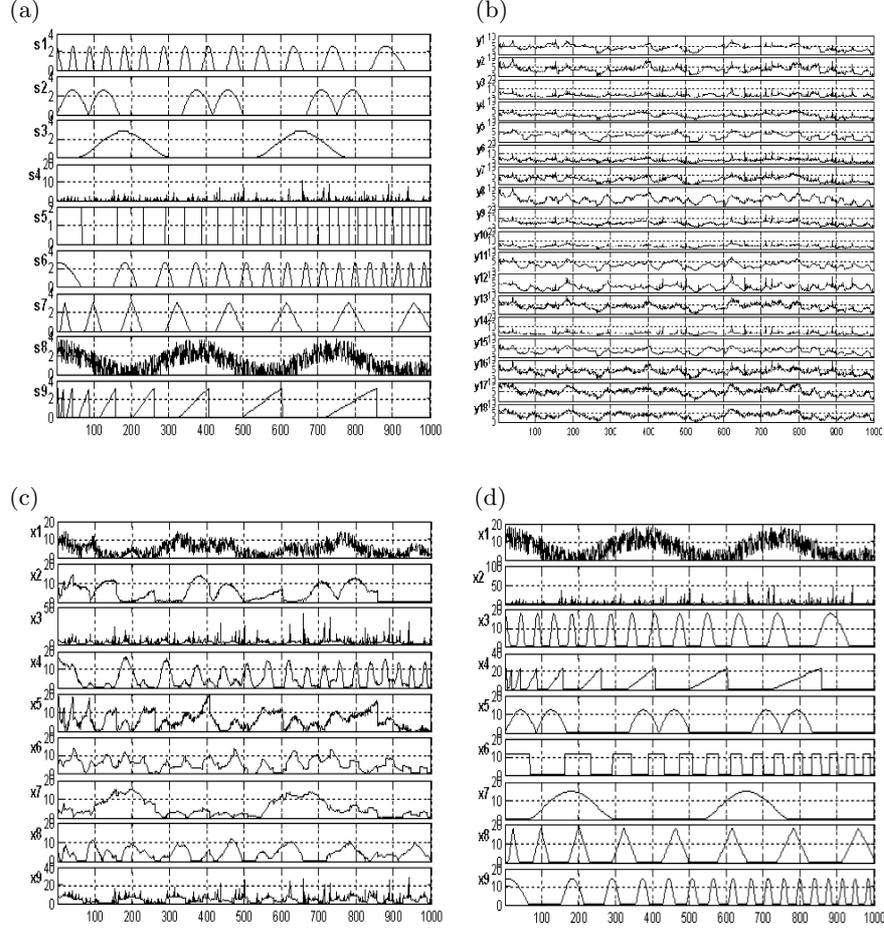


Fig. 2. Example 2: (a) Original 9 source signals; (b) observed 18 mixed signals; (c) Estimated source signals with the standard Lee-Seung algorithm for KL function (2) ($SIR = 1, 7.6, 5.9, 4.9, 4.4, 9.8, 5.9, 10.7, 3.7$ [dB], respectively); (d) Estimated source signals with new Fixed-Point – Quasi-Newton algorithm with $SIR = 130.8, 126.6, 135.9, 129.6, 135.9, 129.5, 133.7, 119.5, 137.4$ [dB], respectively.

For a sparse mixing matrix, the results can be even better. For estimating \mathbf{A} we set $\alpha = 2$, but satisfactory results can be obtained for $\alpha \in [-1, 2]$.

The cost functions (2) and (19) are convex with respect to only one set of the arguments (\mathbf{X} or \mathbf{A}). In the whole set of the arguments, both functions are non-convex, and hence, the alternating minimization may get stuck easily in local minima. To check how plausible are the single estimations given by the tested algorithms, the Monte Carlo analysis is carried out from 100 SIR samples in each case. We tested four different algorithms. Fig. 3 (a) presents the histogram of the SIR samples obtained with the traditional Lee-Seung algorithm. Applying

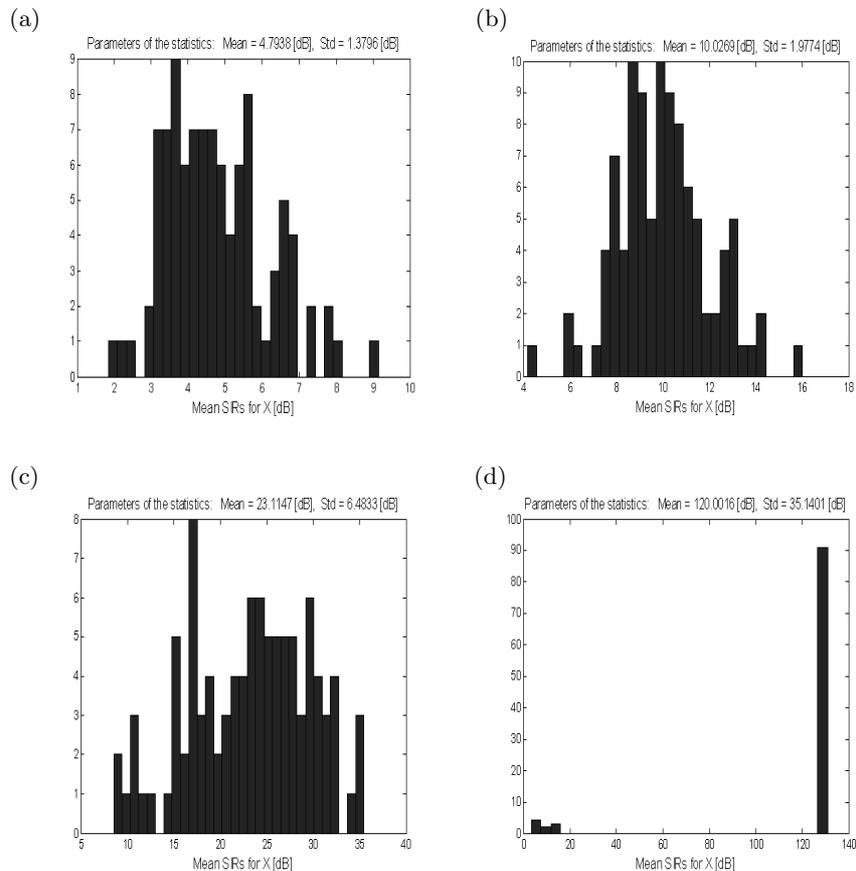


Fig. 3. Histograms from 100 SIR samples generated with the following algorithms initialized from uniformly distributed random initial matrices \mathbf{A} and \mathbf{X} : (a) \mathbf{A} – EMLL, \mathbf{X} – EMLL; (b) \mathbf{A} – Quasi-Newton, \mathbf{X} – EMLL; (c) \mathbf{A} – Quasi-Newton, \mathbf{X} – EMLL with 3 inner iterations and $\alpha_{sX} = 0.001$; (d) \mathbf{A} – Quasi-Newton, \mathbf{X} – Fixed-Point (with exponential rule for damping parameter);

the quasi-Newton only to estimation of \mathbf{A} , the mean-SIR performance increased more than twice – see Fig. 3 (b). Then, improving the estimation of \mathbf{X} with the nonlinear projection with $\alpha_{sX} = 0.001$, and using a few inner iterations for updating \mathbf{X} in each alternating step, the performance substantially goes up [Fig. 3 (c)]. However, the extremely good performance is obtained with the hybrid connection of the FP and quasi-Newton algorithm, which is illustrated in Fig. 3 (d). The global solution with the exponential rule is reached 90% times for 100 trials, and such a good performance is not possible to get with the other tested algorithms. However, the rest 10% trials are still quite far from the desired solution, and this problem will be analyzed in our further research.

5 Conclusions

In this paper, we proposed a new hybrid algorithm for NMF, which demonstrates a very good performance. We have confirmed by the extensive simulations that the proposed algorithm can successfully separate signals and images, especially if a suitable regularization/projection is applied. Changing parameter α in the Amari alpha-divergence, we can tune the algorithm to minimize the influence of noisy disturbances in data. The free parameters in the exponential rule (22) steer the updates towards the global solution. All the parameters can be estimated from data, but this issue will be a subject of our future research. The detailed description of our other algorithms for NMF can be found in [15].

References

- [1] Paatero, P., Tapper, U.: Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics* **5** (1994) 111–126
- [2] Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. *Nature* **401** (1999) 788–791
- [3] Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: *Neural Information Proc. Systems, Vancouver, Canada* (2005)
- [4] Chu, M., Plemmons, R.J.: Nonnegative matrix factorization and applications. *Bulletin of the International Linear Algebra Society* **34** (2005) 2–7
- [5] Pauca, V.P., Shahnaz, F., Berry, M.W., Plemmons, R.J.: Text mining using non-negative matrix factorizations. In: *Proc. SIAM Inter. Conf. on Data Mining, Orlando, FL* (2004)
- [6] Cichocki, A., Amari, S.: *Adaptive Blind Signal And Image Processing* (New revised and improved edition). John Wiley, New York (2003)
- [7] Amari, S.: Information geometry of the EM and em algorithms for neural networks. *Neural Networks* **8**(9) (1995) 1379–1408
- [8] Byrne, C.: Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods. *IEEE Transactions on Image Processing* **7** (1998) 100 – 109
- [9] Amari, S.: *Differential-Geometrical Methods in Statistics*. Springer Verlag (1985)
- [10] Cressie, N.A., Read, T.: *Goodness-of-Fit Statistics for Discrete Multivariate Data*. Springer, New York (1988)
- [11] Cichocki, A., Zdunek, R., Amari, S.: Csiszar’s divergences for non-negative matrix factorization: Family of new algorithms. *LNCS* **3889** (2006) 32–39
- [12] Cullum, J.: The effective choice of the smoothing norm in regularization. *Math. Comp.* **3** (1979) 149–170
- [13] Björck, Å.: *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia (1996)
- [14] Hansen, P.C.: *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia (1998)
- [15] Cichocki, A., Zdunek, R.: *NMFLAB for Signal Processing*. Technical report, Laboratory for Advanced Brain Signal Processing, BSI RIKEN, Saitama, Japan (2006) <http://www.bsp.brain.riken.jp>.