# Nonnegative Matrix Factorization with Constrained Second Order Optimization

Rafal ZDUNEK [a,b,*] Andrzej CICHOCKI [a,c]

[a] *Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Wako-shi, Saitama 351-0198, Japan*
[b] *Institute of Telecommunications, Teleinformatics, and Acoustics, Wroclaw University of Technology, Poland*
[c] *Warsaw University of Technology, Poland*

**Abstract**

Nonnegative Matrix Factorization (NMF) solves the following problem: find nonnegative matrices $\boldsymbol{A} \in \mathbb{R}_+^{M \times R}$ and $\boldsymbol{X} \in \mathbb{R}_+^{R \times T}$ such that $\boldsymbol{Y} \cong \boldsymbol{A}\boldsymbol{X}$, given only $\boldsymbol{Y} \in \mathbb{R}^{M \times T}$ and the assigned index $R$. This method has found a wide spectrum of applications in signal and image processing, such as blind source separation, spectra recovering, pattern recognition, segmentation or clustering. Such a factorization is usually performed with an alternating gradient descent technique that is applied to the squared Euclidean distance or Kullback-Leibler divergence. This approach has been used in the widely known Lee-Seung NMF algorithms that belong to a class of multiplicative iterative algorithms. It is well-known that these algorithms, in spite of their low complexity, are slowly-convergent, give only a positive solution (not nonnegative), and can easily fall in to local minima of a non-convex cost function. In this paper, we propose to take advantage of the second order terms of a cost function to overcome the disadvantages of gradient (multiplicative) algorithms. First, a projected quasi-Newton method is presented, where a regularized Hessian with the Levenberg-Marquardt approach is inverted with the Q-less QR decomposition. Since the matrices $\boldsymbol{A}$ and/or $\boldsymbol{X}$ are usually sparse, a more sophisticated hybrid approach based on the Gradient Projection Conjugate Gradient (GPCG) algorithm, which was invented by More and Toraldo, is adapted for NMF. The Gradient Projection (GP) method is exploited to find zero-value components (active), and then the Newton steps are taken only to compute positive components (inactive) with the Conjugate Gradient (CG) method. As a cost function, we used the $\alpha$-divergence that unifies many well-known cost functions. We applied our new NMF method to a Blind Source Separation (BSS) problem with mixed signals and images. The results demonstrate the high robustness of our method.

*Key words:* Nonnegative matrix factorization; Blind source separation; Quasi-Newton method; GPCG; Second order optimization; Fixed-Point algorithm

## 1. INTRODUCTION

Nonnegative Matrix Factorization (NMF) attempts to recover hidden nonnegative structures or patterns from usually redundant data. This technique has been successfully applied in many applications, e.g. in data analysis (pattern recognition, segmentation, clustering, dimensionality reduction) [1–14], signal and image processing (blind source separation, spectra recovering) [15–17], language modeling, text analysis [18], music transcription [5,16,19], or neuro-biology (gene separation) [20,21].

NMF decomposes the data matrix $\boldsymbol{Y} = [y_{mt}] \in \mathbb{R}^{M \times T}$ as a product of two nonnegative matrices $\boldsymbol{A} = [a_{mr}] \in \mathbb{R}^{M \times R}$ and $\boldsymbol{X} = [x_{rt}] \in \mathbb{R}^{R \times T}$, where

* Corresponding author.
*Email addresses:* zdunek@brain.riken.jp (Rafal ZDUNEK), cia@brain.riken.jp (Andrzej CICHOCKI).

$\forall m, r, t : a_{mr} \geq 0,\ x_{rt} \geq 0$. Although some matrix factorizations provide exact factors (i.e., $\boldsymbol{Y} = \boldsymbol{AX}$), here we shall consider a factorization that is approximative in nature, i.e.,

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{V}, \tag{1}$$

where $\boldsymbol{V} \in \mathbb{R}^{M \times T}$ represents a noise or error matrix.

Depending on an application, the hidden structures may have different interpretation. For example, Lee and Seung in [6] introduced NMF as a method to decompose an image (face) into parts-based representations (parts reminiscent of features such as lips, eyes, nose, etc.). In Blind Source Separation (BSS) [22], the matrix $\boldsymbol{Y}$ represents the observed mixed (superposed) signals or images, $\boldsymbol{A}$ is a mixing operator, and $\boldsymbol{X}$ is a matrix of true source signals or images. Each row of $\boldsymbol{Y}$ or $\boldsymbol{X}$ is a signal or 1D image representation, where $M$ is a number of observed mixed signals and $R$ is a number of hidden (source) components. The index $t$ usually denotes a sample (discrete time instant), where $T$ is a number of samples. In BSS, we usually have $T >> M \geq R$, and $R$ is known or can be relatively easily estimated using SVD or PCA.

Our objective is to estimate the mixing matrix $\boldsymbol{A}$ and sources $\boldsymbol{X}$ subject to nonnegativity constraints of all the entries, given $\boldsymbol{Y}$ and possibly the knowledge on a statistical distribution of noisy disturbances.

The basic approach to NMF, which is presented in Algorithm 1, is the alternating minimization of a specific cost function. In general, the cost function $D(\boldsymbol{Y}||\boldsymbol{AX})$ in Step 1 can be different than the function $\tilde{D}(\boldsymbol{Y}||\boldsymbol{AX})$ in Step 2, however, usually $D(\boldsymbol{Y}||\boldsymbol{AX}) = \tilde{D}(\boldsymbol{Y}||\boldsymbol{AX})$.

**Algorithm 1. General form of NMF**

**Set** Randomly initialize: $\boldsymbol{A}^{(0)}$, $\boldsymbol{X}^{(0)}$,

**For** $s = 1, 2, \ldots,$ until convergence **do**

Step 1:

$\boldsymbol{X}^{(s+1)} = \arg\min_{x_{rt} \geq 0} D(\boldsymbol{Y}||\boldsymbol{A}^{(s)}\boldsymbol{X})|_{\boldsymbol{X}^{(s)}}$

Step 2:

$\boldsymbol{A}^{(s+1)} = \arg\min_{a_{mr} \geq 0} \tilde{D}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}^{(s+1)})|_{\boldsymbol{A}^{(s)}}$

**End**

Lee and Seung [6] were the first to apply the Algorithm 1 separatively to two different cost functions: squared Euclidean distance (Frobenius norm) and Kullback-Leibler (KL) divergence. Using a gradient descent approach to perform Steps 1 and 2, they finally obtained multiplicative algorithms that were previously known in other applications as the EMML or Richardson-Lucy algorithm (RLA) [23–26] for minimization of the KL divergence, and the ISRA algorithm [27] which minimizes the Euclidean distance.

However, the multiplicative algorithms are known to be very slowly-convergent, give only a positive solution (not nonnegative), and easily get stuck in local minima. Many approaches have been proposed in the literature to relax these problems. One of them is to apply Projected Gradient (PG) algorithms [28–30] or projected Alternating Least-Squares (ALS) algorithms [31–33] instead of multiplicative ones. Another improvement concerns modification of the learning rate (relaxation parameter) to speed up the convergence, and better recovering zero-value entries (for sparse solutions). C.-J. Lin [29] suggested applying the Armijo rule to estimate the learning parameters in projected gradient updates for NMF. Also, Interior-Point Gradient (IPG) algorithms (see, e.g. [34,35]) address the issue with selecting such a learning parameter that is the steepest descent and also keeps some distance to a boundary of nonnegative orthant.

In this paper, we extend the idea presented in [36], which concerns exploiting the information from the second-order term in the Taylor expansion of a cost function to speed up the convergence. First, we discuss a projected quasi-Newton method in which a regularized Hessian with the Levenberg-Marquardt approach is inverted using the Q-less QR decomposition. An application of a quasi-Newton method to NMF has also been suggested in [30]. Then, we also propose to use an alternative approach that involves using the Gradient Projection Conjugate Gradient (GPCG) algorithm. To our best knowledge, the GPCG algorithm has not been applied to NMF problems so far. Originally, the algorithm was invented by More and Toraldo [37] to solve large-scale obstacle problems, elastic-plastic torsion problem, and journal bearing problems, and latter, Bardsley [38,39] applied it for solving large-scale problems in image reconstruction and restoration. This algorithm is based on the "reduced" Newton method which is only applied to update positive components (inactive). Due to the sparseness, a set of inactive components is not very large, which keeps a relatively low computational cost. Moreover, the reduced Hessian is not inverted directly, but it is used to compute gradient updates with the Conjugate Gradient (CG) method [40]. The CG iterates con-

verge within a finite number of iterations because the reduced Hessian is positive-definite. Then, the gradient updates are used with the Gradient Projection (GP) method to find zero-value components (active).

As a cost function, we used the $\alpha$-divergence [22,41,42] that unifies many well-known cost functions, and it adapts the algorithm to a statistical distribution of noise with only one parameter.

This paper is organized as follows: Section 2 presents the quasi-Newton method in application to NMF. Section 3 shortly explains the modified Fixed-Point (FP) algorithm. In Section 4, we present the GPCG algorithm and its extension to NMF. The numerical results from a BSS problem of mixed signals and images are illustrated in Section 5. Finally, the conclusions and future work are presented in Section 6.

## 2. QUASI-NEWTON OPTIMIZATION

The $\alpha$-divergence [22,41–43] can be expressed as follows:

$$D_A(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{mt} y_{mt} \frac{(y_{mt}/z_{mt})^{\alpha-1}-1}{\alpha(\alpha-1)} + \frac{z_{mt}-y_{mt}}{\alpha},$$
$$z_{mt} = [\boldsymbol{AX}]_{mt}, \quad y_{mt} = [\boldsymbol{Y}]_{mt}. \tag{2}$$

The KL divergence can be obtained for $\alpha \to 1$, but if $\alpha \to 0$ the dual KL divergence can be derived. For $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger's, and Neyman's chi-square distances, respectively.

Applying the projected Newton method to (2), we have

$$\boldsymbol{X} \leftarrow \mathcal{P}_{\Omega_X}[\boldsymbol{X} - [\boldsymbol{H}_{D_A}^{(X)}]^{-1} \nabla_{\mathbf{X}} D_A], \tag{3}$$
$$\boldsymbol{A} \leftarrow \mathcal{P}_{\Omega_A}[\boldsymbol{A} - [\boldsymbol{H}_{D_A}^{(A)}]^{-1} \nabla_{\mathbf{A}} D_A], \tag{4}$$

where $\boldsymbol{H}_{D_A}^{(X)}$ and $\boldsymbol{H}_{D_A}^{(A)}$ are Hessians, $\nabla_{\mathbf{X}} D_A$ and $\nabla_{\mathbf{A}} D_A$ are gradient matrices for (2) with respect to $\boldsymbol{X}$ and $\boldsymbol{A}$, respectively. The applications $\mathcal{P}_{\Omega_X}[.]$ and $\mathcal{P}_{\Omega_A}[.]$ project from $\mathbb{R}^{R\times T}$ and $\mathbb{R}^{M\times R}$ into the corresponding feasible sets $\Omega_X \in \mathbb{R}_+^{R\times T}$ and $\Omega_A \in \mathbb{R}_+^{M\times R}$ which are defined as follows:

$$\Omega_X = \{\boldsymbol{X} \in \mathbb{R}^{R\times T} : \; x_{rt} \geq 0\}, \tag{5}$$

$$\Omega_A = \{\boldsymbol{A} \in \mathbb{R}^{M\times R} : \; a_{mr} \geq 0\}. \tag{6}$$

For $\alpha \neq 0$, the gradient $\boldsymbol{G}_{D_A}^{(X)} \in \mathbb{R}^{R\times T}$ with respect to $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{G}_{D_A}^{(X)} = \nabla_{\mathbf{X}} D_A = \frac{1}{\alpha} \boldsymbol{A}^T \left(\underline{1} - (\boldsymbol{Y} \oslash \boldsymbol{AX})^{\alpha}\right), \tag{7}$$

where $\oslash$ means a element-wise division. The Hessian has the form:

$$\boldsymbol{H}_{D_A}^{(X)} = \frac{1}{\alpha} \operatorname{diag}\{[\boldsymbol{h}_t^{(X)}]_{t=1,\ldots,T}\} \in \mathbb{R}^{RT\times RT}, \tag{8}$$

where

$$\boldsymbol{h}_t^{(X)} = \boldsymbol{A}^T \operatorname{diag}\{[\boldsymbol{Y}^{\alpha} \oslash (\boldsymbol{AX})^{\alpha+1}]_{*,t}\}\boldsymbol{A} \in \mathbb{R}^{R\times R}.$$

Similarly for $\boldsymbol{A}$, we get

$$\boldsymbol{G}_{D_A}^{(A)} = \nabla_{\mathbf{A}} D_A = \frac{1}{\alpha} \left(\underline{1} - (\boldsymbol{Y} \oslash \boldsymbol{AX})^{\alpha}\right) \boldsymbol{X}^T \in \mathbb{R}^{M\times R}, \tag{9}$$

and the Hessian $\boldsymbol{H}_{D_A}^{(A)} \in \mathbb{R}^{MR\times MR}$ is as follows:

$$\boldsymbol{H}_{D_A}^{(A)} = \frac{1}{\alpha} \operatorname{diag}\{[\boldsymbol{h}_m^{(A)}]_{m=1,\ldots,M}\}, \tag{10}$$

where

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

For the specific case, when $\alpha \to 0$, the $\alpha$-divergence converges to the dual KL divergence, i.e.

$$D_{KL2}(\boldsymbol{AX}||\boldsymbol{Y}) = \lim_{\alpha\to 0} D_A(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{mt} \left( z_{mt} \ln \frac{z_{mt}}{y_{mt}} + y_{mt} - z_{mt} \right), \tag{11}$$
$$z_{mt} = [\boldsymbol{AX}]_{mt},$$

and consequently, the gradient and Hessian matrices simplify as follows:

– For $\boldsymbol{X}$:

$$\boldsymbol{G}_{D_{KL2}}^{(X)} = \nabla_{\mathbf{X}} D_{KL2} = \boldsymbol{A}^T \ln(\boldsymbol{AX} \oslash \boldsymbol{Y}) \in \mathbb{R}^{R\times T}, \tag{12}$$

and

$$\boldsymbol{H}_{D_{KL2}}^{(X)} = \operatorname{diag}\{[\boldsymbol{h}_t^{(X)}]_{t=1,\ldots,T}\} \in \mathbb{R}^{RT\times RT}, \tag{13}$$

where

$$\boldsymbol{h}_t^{(X)} = \boldsymbol{A}^T \operatorname{diag}\{[1 \oslash (\boldsymbol{AX})]_{*,t}\}\boldsymbol{A} \in \mathbb{R}^{R\times R}.$$

– For $\boldsymbol{A}$:

$$\boldsymbol{G}_{D_{KL2}}^{(A)} = \nabla_{\mathbf{A}} D_{KL2} = \ln(\boldsymbol{AX} \oslash \boldsymbol{Y}) \boldsymbol{X}^T \in \mathbb{R}^{M\times R}, \tag{14}$$

and

$$\boldsymbol{H}^{(A)}_{D_{KL2}} = \operatorname{diag}\{[\boldsymbol{h}^{(A)}_m]_{m=1,\ldots,M}\} \in \mathbb{R}^{MR\times MR}, \quad (15)$$

where

$$\boldsymbol{h}^{(A)}_m = \boldsymbol{X}\operatorname{diag}\{[1 \oslash (\boldsymbol{AX})]_{m,*}\}\boldsymbol{X}^T \in \mathbb{R}^{R\times R}.$$

The $\alpha$-divergence unifies many well-known statistical distances, which makes our NMF algorithm more flexible to various distributions of noise. For $\alpha = 2$ we have the Pearson's distance which can be regarded as a normalized squared Euclidean distance. However, the basic Euclidean distance cannot be derived from the $\alpha$-divergence. This case may be very useful in practice, since a normally distributed noise happens very often. For the squared Euclidean distance:

$$D_F(\boldsymbol{Y}||\boldsymbol{AX}) = \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{AX}||^2_F, \quad (16)$$

the gradients and Hessians have the corresponding forms:

$$\boldsymbol{G}^{(X)}_{D_F} = \boldsymbol{A}^T(\boldsymbol{AX} - \boldsymbol{Y}) \in \mathbb{R}^{R\times T}, \quad (17)$$

$$\boldsymbol{G}^{(A)}_{D_F} = (\boldsymbol{AX} - \boldsymbol{Y})\boldsymbol{X}^T \in \mathbb{R}^{M\times R}, \quad (18)$$

$$\boldsymbol{H}^{(X)}_{D_F} = \boldsymbol{I}_T \otimes \boldsymbol{A}^T\boldsymbol{A} \in \mathbb{R}^{RT\times RT}, \quad (19)$$

$$\boldsymbol{H}^{(A)}_{D_F} = \boldsymbol{I}_M \otimes \boldsymbol{X}\boldsymbol{X}^T \in \mathbb{R}^{MR\times MR}, \quad (20)$$

where $\boldsymbol{I}_T \in \mathbb{R}^{T\times T}$ and $\boldsymbol{I}_M \in \mathbb{R}^{M\times M}$ are identity matrices, and $\otimes$ stands for a Kronecker product.

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

**Remark 1** *All the Hessians $\boldsymbol{H}^{(X)}_{D_A}$, $\boldsymbol{H}^{(A)}_{D_A}$, $\boldsymbol{H}^{(X)}_{D_{KL2}}$, $\boldsymbol{H}^{(A)}_{D_{KL2}}$, $\boldsymbol{H}^{(X)}_{D_F}$ and $\boldsymbol{H}^{(A)}_{D_F}$ have a block-diagonal structure. For $\alpha > 0$, the Hessian given by (8) is not positive-definite if $\exists t,\ \forall m : y_{mt} = 0$. Similarly, if $\exists m,\ \forall t : y_{mt} = 0$, the Hessian (10) is also not positive-definite. This case may happen if $\boldsymbol{A}$ and $\boldsymbol{X}$ are very sparse. Also, if $\boldsymbol{Y}$ has many zero-value entries, the Hessians may be semi-positive definite. For the Hessians (13), (15), (19) and (20) the normalization of the columns in $\boldsymbol{A}$ in each alternating step keeps their positive-definiteness, however, they can be still very ill-conditioned, especially in early updates. Thus, to avoid a breakdown of Newton iterations, some regularization of the Hessian is essential, which leads to quasi-Newton iterations. We applied the Levenberg-Marquardt approach with the exponentially deceasing regularization parameter: $\lambda = \bar{\lambda}^{(H)}_0 + \lambda^{(H)}_0 \exp\{-\tau^{(H)}k\}$. Such regularization substantially reduces possible ill-conditioning of the Hessian (random initialization) during initial iterations when the updates are still very far from the solution.*

Additionally, we control the convergence by a slight relaxation of the iterative updates. To reduce a computational cost substantially, the inversion of the Hessian is replaced with the Q-less QR factorization computed with LAPACK. Thus

$$\boldsymbol{H}^{(X)}_{D_A} + \lambda\boldsymbol{I}_X = \boldsymbol{Q}_X\boldsymbol{R}_X, \qquad \boldsymbol{W}_X = \boldsymbol{Q}^T_X\nabla_{\mathbf{X}}D_A,$$

$$\boldsymbol{H}^{(A)}_{D_A} + \lambda\boldsymbol{I}_A = \boldsymbol{Q}_A\boldsymbol{R}_A, \qquad \boldsymbol{W}_A = \boldsymbol{Q}^T_A\nabla_{\mathbf{A}}D_A,$$

where $\boldsymbol{R}_X$ and $\boldsymbol{R}_A$ are upper triangular matrices, and $\boldsymbol{Q}_X$ and $\boldsymbol{Q}_A$ are orthogonal matrices that are not explicitly computed with the Q-less QR factorization. The final form of the algorithm with the quasi-Newton algorithm is

$$\boldsymbol{X} \leftarrow \mathcal{P}_{\Omega_X}[\boldsymbol{X} - \gamma\boldsymbol{R}^{-1}_X\boldsymbol{W}_X], \quad (21)$$

$$\boldsymbol{A} \leftarrow \mathcal{P}_{\Omega_A}[\boldsymbol{A} - \gamma\boldsymbol{R}^{-1}_A\boldsymbol{W}_A], \quad (22)$$

where $\boldsymbol{I}_X \in \mathbb{R}^{RT\times RT}$, $\boldsymbol{I}_A \in \mathbb{R}^{MR\times MR}$ are identity matrices, and $\gamma$ controls the relaxation. We set $\gamma = 0.9$. Since the matrices $\boldsymbol{R}_X$ and $\boldsymbol{R}_A$ are upper-triangular, the computational complexity of the algorithm (21) and (22) is the lowest with the Gaussian elimination, however, a direct inversion or even a pseudo-inversion may be more suitable for ill-conditioned NMF problems.

## 3. FIXED-POINT ALGORITHM

In our application, $\boldsymbol{X}$ has much larger dimensions than $\boldsymbol{A}$, and hence, the computation of $\boldsymbol{X}$ 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 $T = 1000$. Thus the Hessian $\boldsymbol{H}^{(A)}$ has size 200 by 200 with $MR^2 = 2\times10^3$ non-zero entries, but the size of $\boldsymbol{H}^{(X)}$ is $10^4$ by $10^4$ with $TR^2 = 10^5$ non-zero entries. For this reason, we do not apply the Newton method for updating $\boldsymbol{X}$. This can be also justified by the fact that the computation of $\boldsymbol{A}$ needs to solve the system which is much more over-determined than for $\boldsymbol{X}$, 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 this paper, the nonnegative components in $\boldsymbol{X}$ are basically estimated with the modified Fixed-Point (FP) algorithm that solves a regularized least-squares problem

$$\boldsymbol{X}^* = \arg\min_X \left\{ \frac{1}{2}||\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}||_F^2 + \frac{\lambda_X}{2} J(\boldsymbol{X}) \right\}. \quad (23)$$

The regularization term

$$J(\boldsymbol{X}) = \sum_t (||\boldsymbol{x}_t||_1)^2 = \mathrm{tr}\{\boldsymbol{X}^T \boldsymbol{E} \boldsymbol{X}\},$$

where $\boldsymbol{x}_t$ is the $t$-th column of $\boldsymbol{X}$, and $\boldsymbol{E} \in \mathbb{R}^{R \times R}$ is a matrix of all ones entries, enforces sparsity in the columns of $\boldsymbol{X}$ and it is motivated by the diversity measure [1] used in the M-FOCUSS algorithm [44]. Let $\Psi(\boldsymbol{X})$ be the objective function in (23). Thus the stationary point of $\Psi(\boldsymbol{X})$ is reached when

$$\nabla_{\mathbf{X}} \Psi(\boldsymbol{X}) = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{X} - \boldsymbol{Y}) + \lambda_X \boldsymbol{E}\boldsymbol{X} = 0,$$

which leads to the following regularized least-squares solution

$$\boldsymbol{X}_{LS} = (\boldsymbol{A}^T\boldsymbol{A} + \lambda_X \boldsymbol{E})^{-1} \boldsymbol{A}^T \boldsymbol{Y}.$$

Then, to satisfy the nonnegativity constraints, $\boldsymbol{X}_{LS}$ is projected into $\Omega_X$ that is defined by (5). The cost function given by the regularized square Euclidean distance as in (23) works the best with a Gaussian noise (matrix $\boldsymbol{V}$ in (1)), however, the computation of $\boldsymbol{A}$ uses the $\alpha$-divergence which is optimal for a wide spectrum of signal distributions.

We set the regularization parameter $\lambda_X$ in (23) according to the exponential rule, i.e.

$$\lambda_X = \lambda_X^{(k)} = \lambda_0 \exp\{-\tau k\}, \quad (24)$$

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. Larger parameter $\lambda_0$ and smaller $\tau$ should give better results but at the cost of high increase in a number of alternating steps.

## 4. REDUCED NEWTON OPTIMIZATION

The GPCG is a hybrid nested iterative method for nonnegatively constrained convex optimization. In this section, we apply the modified GPCG to perform the Steps 1 and 2 in Algorithm 1, where the cost functions $D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X})$ and $\tilde{D}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X})$ are defined by the $\alpha$-divergence (2) for $\alpha > 0$, and by the dual KL divergence (11) for $\alpha = 0$. In NMF, we solve two separated problems:

$$\boldsymbol{X}^* = \arg\min_{x_{rt} \geq 0} D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}), \quad (25)$$

$$\boldsymbol{A}^* = \arg\min_{a_{mr} \geq 0} D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}). \quad (26)$$

We restrict the description of the GPCG only to the case of solving the problem (25). The problem (26) can be also similarly treated by applying the GPCG to the transposed system: $\boldsymbol{X}^T\boldsymbol{A}^T = \boldsymbol{Y}^T$, where $\boldsymbol{A}$ is unknown given $\boldsymbol{X}$ and $\boldsymbol{Y}$.

In the remainder of the paper, we use the same notation for a gradient and Hessian of the $\alpha$-divergence and dual KL divergence, i.e. $\boldsymbol{P}^{(X)} = \boldsymbol{G}_{D_A}^{(X)}$, $\boldsymbol{P}^{(A)} = \boldsymbol{G}_{D_A}^{(A)}$, $\boldsymbol{H}_X = \boldsymbol{H}_{D_A}^{(X)}$ and $\boldsymbol{H}_A = \boldsymbol{H}_{D_A}^{(A)}$ for $\alpha > 0$, and $\boldsymbol{P}^{(X)} = \boldsymbol{G}_{D_{KL2}}^{(X)}$, $\boldsymbol{P}^{(A)} = \boldsymbol{G}_{D_{KL2}}^{(A)}$, $\boldsymbol{H}_X = \boldsymbol{H}_{D_{KL2}}^{(X)}$ and $\boldsymbol{H}_A = \boldsymbol{H}_{D_{KL2}}^{(A)}$ for $\alpha = 0$.

The solution $\boldsymbol{X}^*$ of the constrained problem (25) should satisfy the KKT conditions, i.e.

$$\forall r,t: \ \frac{\partial}{\partial x_{mt}} D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}^*) = 0, \ \text{if } x_{mt}^* > 0, \quad (27)$$

$$\forall r,t: \ \frac{\partial}{\partial x_{mt}} D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}^*) > 0, \ \text{if } x_{mt}^* = 0. \quad (28)$$

The GPCG is a two-step algorithm: in the first step, a solution is updated with Gradient Projection (GP) iterations, and in the other step, only the gradient $\boldsymbol{P}^{(X)} = \nabla_{\mathbf{X}} D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X})$ is updated with the Newton iterations applied to solve a subproblem, i.e. the reduced system that is obtained from the original one by removing the components that satisfy (28). Hence, it is called the "reduced" Newton optimization.

The GP updates are as follows:

$$\boldsymbol{X} \leftarrow \mathcal{P}_{\Omega_X}[\boldsymbol{X} - \eta_p \boldsymbol{P}^{(X)}], \quad (29)$$

where $\mathcal{P}_{\Omega_X}[\xi]$ is a projection of $\xi$ onto the feasible set $\Omega_X$ defined by (5). The step length $\eta_p$ is inexactly estimated with the Armijo rule, i.e. $\eta_p = \beta^p \eta_0$, for $p = 1, 2, \ldots$, and $\beta \in (0,1)$. The iterations for $\eta_p$ are stopped at the first $p$ for which

$$D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) - D(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}(\eta_p)) \geq \frac{\mu}{\eta_p} ||\boldsymbol{X} - \boldsymbol{X}(\eta_p)||_F^2,$$

[1] The diversity measure $J^{(p,q)} = \sum_{i=1}^n (||\boldsymbol{x}[i]||_q)^p$, $p \geq 0$, $q \geq 1$, where $\boldsymbol{x}[i]$ is the $i$-th row of the matrix $\boldsymbol{X}$ $(n \times L)$, was introduced by Cotter et al.[44] to enforce sparsity in the rows of $\boldsymbol{X}$. Since we are more concerned with a sparsity column profile, we apply the measure to the columns in $\boldsymbol{X}$ instead of the rows, assuming $q = 1$ and $p = 2$.

and the initial step length $\eta_0$ is defined for each update as the Cauchy point:

$$\eta_0 = \frac{\boldsymbol{p}_X^T \boldsymbol{p}_X}{\boldsymbol{p}_X^T \boldsymbol{H}_X \boldsymbol{p}_X},$$

where $\boldsymbol{H}_X$ is the Hessian of $D(\boldsymbol{Y}||\boldsymbol{AX})$. The vector $\boldsymbol{p}_X$ is a vectorized version of the gradient matrix $\boldsymbol{P}^{(X)} = [p_{rt}] \in \mathbb{R}^{R \times T}$, i.e.

$$\boldsymbol{p}_X = \text{vec}(\boldsymbol{P}^{(X)}) = [p_{11}^{(X)}, p_{21}^{(X)}, \ldots, p_{R1}^{(X)}, p_{12}^{(X)}, \ldots, p_{RT}^{(X)}]^T \in \mathbb{R}^{RT}. \quad (30)$$

In the second step, only gradient $\boldsymbol{p}_X$ is updated with the standard CG method [40] that solves, so-called, the reduced system:

$$\boldsymbol{H}_R^{(X)} \tilde{\boldsymbol{p}}_X = -\boldsymbol{p}_R^{(X)}, \quad (31)$$

where

$$\boldsymbol{p}_R^{(X)} = \boldsymbol{D}^{(X)} \boldsymbol{p}_X,$$

$$\boldsymbol{H}_R^{(X)} = \boldsymbol{D}^{(X)} \boldsymbol{H}_X \boldsymbol{D}^{(X)} - \boldsymbol{D}^{(X)} + \boldsymbol{I}^{(X)},$$

are the reduced gradient and reduced Hessian. The diagonal matrix $\boldsymbol{D}^{(X)}$ consists of the indices of the inactive variables, i.e. the entries of the solution $\boldsymbol{X}$ that are positive at a given outer iterative step. Thus

$$\boldsymbol{D}^{(X)} = \text{diag}\{\boldsymbol{z}^{(X)}\},$$

where

$$\boldsymbol{z}_X = \text{vec}(\boldsymbol{Z}^{(X)}) = [z_{11}^{(X)}, z_{21}^{(X)}, \ldots, z_{R1}^{(X)}, z_{12}^{(X)}, \ldots, z_{RT}^{(X)}]^T \in \mathbb{R}^{RT},$$

$$\boldsymbol{Z}^{(X)} = [z_{rt}^{(X)}], \quad z_{rt}^{(X)} = \begin{cases} 1 \text{ if } & x_{rt} > 0, \\ 0 \text{ otherwise}, & \end{cases}$$

The matrix $\boldsymbol{I}^{(X)} \in \mathbb{R}^{RT \times RT}$ denotes an identity matrix. By applying the diagonal scaling, the zero-value components of the current GP update are not considered in evaluation of the gradient which will be used in the next GP update. In this way, the system (31) is very sparse.

**Remark 2** *The convergence of the CG is guaranteed in a finite number of iterations since $\boldsymbol{H}_R^{(X)}$ is positive-definite. This is because $D_A(\boldsymbol{Y}||\boldsymbol{AX})$ is convex for $\alpha \geq 0$ and $\boldsymbol{P}^{(X)}$ is a nonsingular matrix. The GP with the Armijo rule is always convergent to a stationary point of $D_A(\boldsymbol{Y}||\boldsymbol{AX})$ [38,39], which is a unique solution.*

**Algorithm 2. GPCG-NMF**

**Set** Random initialize: $\boldsymbol{A}$, $\boldsymbol{X}$,
$\beta \in (0,1)$, $\mu \in (0,1)$, $\gamma_{GP} \in (0,1)$,
**For** $s = 0, 1, \ldots,$ *% Alternating*
Step 1: Do **X-GPCG** iterations with Algorithm,
Step 2: Do **A-GPCG** iterations with Algorithm ,
**End** *% Alternating*

The solution $\tilde{\boldsymbol{p}}_X$ of (31) is then used in the GP iterations in (29), i.e. the following gradient matrix is created from the vector $\tilde{\boldsymbol{p}}_X$:

$$\boldsymbol{P}^{(X)} \leftarrow \text{Matrix}(\tilde{\boldsymbol{p}}_X) \in \mathbb{R}^{R \times T}. \quad (32)$$

The termination of both GP and CG iterations is achieved with the gradient descent criteria that are given in [37–39].

The Algorithm 2 is the simplified version of the GPCG adapted for NMF.

## 5. NUMERICAL RESULTS

The proposed NMF algorithms have been extensively tested for many difficult benchmarks for signals and images with various statistical distributions. Here we show two illustrative examples. The performance of the algorithms is also estimated with a quantity measure: Signal-to-Interference Ratio (SIR).

The five statistically dependent nonnegative signals shown in Fig. 1(a) have been mixed by randomly generated nonnegative well-conditioned matrix $\boldsymbol{A} \in \mathbb{R}^{5 \times 5}$ ($\text{cond}(\boldsymbol{A}) \simeq 10$) with a uniform distribution. The mixed signals are shown in Fig. 1(b). Using the standard multiplicative NMF algorithms we failed to estimate the original sources. Figs. 2(a) and 3(left) illustrate the results obtained with the standard Lee-Seung algorithm, referred here to as the EMML (Expectation Maximization Maximum Likelihood) for alternating minimization of the Kullback-Leibler divergence. The results shown in Figs. 2(b) and 3(right) are obtained with the hybrid algorithm: FP for $\boldsymbol{X}$ (Step 1) and quasi-Newton for $\boldsymbol{A}$ (Step 2). Fig. 3 presents the histograms from 100 mean-SIR samples generated with the Monte Carlo (MC) analysis for two different algorithms. In each MC run only initial matrices $\boldsymbol{A}^{(0)}$ and $\boldsymbol{X}^{(0)}$ were randomized. The worst case with the quasi-

**Algorithm 3. X-GPCG**

**For** $k = 0, 1, \ldots,$ *% Inner loop for* $\boldsymbol{X}$

Step 1: $\boldsymbol{P}^{(X)} \leftarrow -\nabla_{\mathbf{X}} D(\boldsymbol{Y}||\boldsymbol{AX})$,
$\mathrm{vec}(\boldsymbol{P}^{(X)}) = [p_{11}^{(X)}, p_{21}^{(X)}, \ldots,$
$p_{R1}^{(X)}, p_{12}^{(X)}, \ldots, p_{RT}^{(X)}]^T \in \mathbb{R}^{RT}$,
where $\boldsymbol{P}^{(X)} = [p_{rt}^{(X)}]$, *% Vectorization*
$\boldsymbol{H}_X = \nabla_{\mathbf{X}}^2 D(\boldsymbol{Y}||\boldsymbol{AX})$, *% Hessian*

Step 2: $\boldsymbol{X} \leftarrow \max\{\boldsymbol{X} + \eta_p \boldsymbol{P}^{(X)}, 0\}$, *% Projection*
where $\eta_p = \beta^p \eta_0$,
$\eta_0 = \dfrac{\mathrm{vec}(\boldsymbol{P}^{(X)})^T \mathrm{vec}(\boldsymbol{P}^{(X)})}{\mathrm{vec}(\boldsymbol{P}^{(X)})^T \boldsymbol{H}_X \mathrm{vec}(\boldsymbol{P}^{(X)})}$,
and $p = 0, 1, \ldots$ is the first non-negative integer for which:
$D(\boldsymbol{Y}||\boldsymbol{AX}(\eta_p)) - D(\boldsymbol{Y}||\boldsymbol{AX})) \leq$
$-\frac{\mu}{\eta_p}||\boldsymbol{X} - \boldsymbol{X}(\eta_p)||_F^2$,

Step 3: $\boldsymbol{Z}^{(X)} = [z_{rt}^{(X)}]$, $z_{rt}^{(X)} = \begin{cases} 1 \text{ if } & x_{rt} > 0, \\ 0 \text{ otherwise,} \end{cases}$
$\boldsymbol{z}^{(X)} = \mathrm{vec}(\boldsymbol{Z}^{(X)}) = [z_{11}^{(X)}, z_{21}^{(X)}, \ldots,$
$z_{R1}^{(X)}, z_{12}^{(X)}, \ldots, z_{RT}^{(X)}]^T \in \mathbb{R}^{RT}$,
$\boldsymbol{D}^{(X)} = \mathrm{diag}\{\boldsymbol{z}^{(X)}\}$,

Step 4: $\boldsymbol{P}^{(X)} \leftarrow \nabla_{\mathbf{X}} D(\boldsymbol{Y}||\boldsymbol{AX})$, *% Gradient*

Step 5: $\boldsymbol{p}_R^{(X)} = \boldsymbol{D}^{(X)} \mathrm{vec}(\boldsymbol{P}^{(X)})$, *% Reduced grad.*
$\boldsymbol{H}_R^{(X)} = \boldsymbol{D}^{(X)} \boldsymbol{H}_X \boldsymbol{D}^{(X)} + \boldsymbol{I}^{(X)} - \boldsymbol{D}^{(X)}$,

Step 6: Solve: $\boldsymbol{H}_R^{(X)} \boldsymbol{p}_X = -\boldsymbol{p}_R^{(X)}$
with `CG` algorithm
$\boldsymbol{P}^{(X)} \leftarrow \mathrm{Matrix}(\boldsymbol{p}_X) \in \mathbb{R}^{R \times T}$,

Step 7: $\boldsymbol{X} \leftarrow \max\{\boldsymbol{X} + \eta_p \boldsymbol{P}^{(X)}, 0\}$, *% Projection*
where $\eta_p = \beta^p \eta_0, \quad \eta_0 = 1$,
and $p = 0, 1, \ldots$ is the first non-negative integer for which:
$D(\boldsymbol{Y}||\boldsymbol{AX}(\eta_p)) < D(\boldsymbol{Y}||\boldsymbol{AX})$,

**If** $D(\boldsymbol{Y}||\boldsymbol{AX}^{(k-1)}) - D(\boldsymbol{Y}||\boldsymbol{AX}^{(k)}) \leq$
$\gamma_{GP} \max\Big\{D(\boldsymbol{Y}||\boldsymbol{AX}^{(l-1)})$
$-D(\boldsymbol{Y}||\boldsymbol{AX}^{(l)})|\ l = 1, \ldots, k-1\Big\}$,
**Break**

**End If**

**End** *% Inner loop for* $\boldsymbol{X}$

**Algorithm 4. A-GPCG**

**For** $k = 0, 1, \ldots,$ *% Inner loop for* $\boldsymbol{A}$

Step 1: $\boldsymbol{P}^{(A)} \leftarrow -\nabla_{\mathbf{A}} D(\boldsymbol{Y}||\boldsymbol{AX}) \in \mathbb{R}^{M \times R}$,
$\mathrm{vec}(\boldsymbol{P}^{(A)}) = [p_{11}^{(A)}, p_{12}^{(A)}, \ldots,$
$p_{1R}^{(A)}, p_{21}^{(A)}, \ldots, p_{MR}^{(A)}]^T \in \mathbb{R}^{MR}$,
where $\boldsymbol{P}^{(A)} = [p_{mr}^{(A)}]$, *% Vectorization*
$\boldsymbol{H}_A = \nabla_{\mathbf{A}}^2 D(\boldsymbol{Y}||\boldsymbol{AX})$, *% Hessian*

Step 2: $\boldsymbol{A} \leftarrow \max\{\boldsymbol{A} + \eta_p \boldsymbol{P}^{(A)}, 0\}$, *% Projection*
where $\eta_p = \beta^p \eta_0$,
$\eta_0 = \dfrac{\mathrm{vec}(\boldsymbol{P}^{(A)})^T \mathrm{vec}(\boldsymbol{P}^{(A)})}{\mathrm{vec}(\boldsymbol{P}^{(A)})^T \boldsymbol{H}_X \mathrm{vec}(\boldsymbol{P}^{(A)})}$,
and $p = 0, 1, \ldots$ is the first non-negative integer for which:
$D(\boldsymbol{Y}||\boldsymbol{A}(\eta_p)\boldsymbol{X}) - D(\boldsymbol{Y}||\boldsymbol{AX}) \leq$
$-\frac{\mu}{\eta_p}||\boldsymbol{A} - \boldsymbol{A}(\eta_p)||_F^2$,

Step 3: $\boldsymbol{Z}^{(A)} = [z_{mr}^{(A)}]$, $z_{mr}^{(A)} = \begin{cases} 1 \text{ if } & a_{mr} > 0, \\ 0 \text{ otherwise,} \end{cases}$
$\boldsymbol{z}^{(A)} = \mathrm{vec}(\boldsymbol{Z}^{(A)}) = [z_{11}^{(A)}, z_{12}^{(A)}, \ldots,$
$z_{1R}^{(A)}, z_{21}^{(A)}, \ldots, z_{MR}^{(A)}]^T \in \mathbb{R}^{MR}$,
$\boldsymbol{D}^{(A)} = \mathrm{diag}\{\boldsymbol{z}^{(A)}\}$,

Step 4: $\boldsymbol{P}^{(A)} \leftarrow \nabla_{\mathbf{A}} D(\boldsymbol{Y}||\boldsymbol{AX})$, *% Gradient*

Step 5: $\boldsymbol{p}_R^{(A)} = \boldsymbol{D}^{(A)} \mathrm{vec}(\boldsymbol{P}^{(A)})$, *% Reduced grad.*
$\boldsymbol{H}_R^{(A)} = \boldsymbol{D}^{(A)} \boldsymbol{H}_A \boldsymbol{D}^{(A)} + \boldsymbol{I}^{(A)} - \boldsymbol{D}^{(A)}$,

Step 6: Solve: $\boldsymbol{H}_R^{(A)} \boldsymbol{p}_A = -\boldsymbol{p}_R^{(A)}$
with `CG` algorithm
$\boldsymbol{P}^{(A)} \leftarrow \mathrm{Matrix}(\boldsymbol{p}_A) \in \mathbb{R}^{M \times R}$,

Step 7: $\boldsymbol{X} \leftarrow \max\{\boldsymbol{A} + \eta_p \boldsymbol{P}^{(A)}, 0\}$, *% Projection*
where $\eta_p = \beta^p \eta_0, \quad \eta_0 = 1$,
and $p = 0, 1, \ldots$ is the first non-negative integer for which:
$D(\boldsymbol{Y}||\boldsymbol{A}(\eta_p)\boldsymbol{X}) < D(\boldsymbol{Y}||\boldsymbol{AX})$,

**If** $D(\boldsymbol{Y}||\boldsymbol{A}^{(k-1)}\boldsymbol{X}) - D(\boldsymbol{Y}||\boldsymbol{A}^{(k)}\boldsymbol{X}) \leq$
$\gamma_{GP} \max\Big\{D(\boldsymbol{Y}||\boldsymbol{A}^{(l-1)}\boldsymbol{X})$
$-D(\boldsymbol{Y}||\boldsymbol{A}^{(l)}\boldsymbol{X})|\ l = 1, \ldots, k-1\Big\}$,
**Break**

**End If**

**End** *% Inner loop for* $\boldsymbol{A}$

Newton algorithm has the mean-SIR value greater than 140 [dB]. Also, the standard deviation of the mean-SIRs with the quasi-Newton algorithm is very small (less than 1 [dB]). This suggests that our algorithm is rather convergent to the global minimum of the cost function [2]. In the experiments for a fixed number of alternating steps (usually 1500), we set

[2] Even a quadratic cost function $D(\boldsymbol{Y}||\boldsymbol{AX}) = ||\boldsymbol{Y} - \boldsymbol{AX}||_F$ with respect to both sets of arguments ($\boldsymbol{A}$ and $\boldsymbol{X}$) may have many local minima.

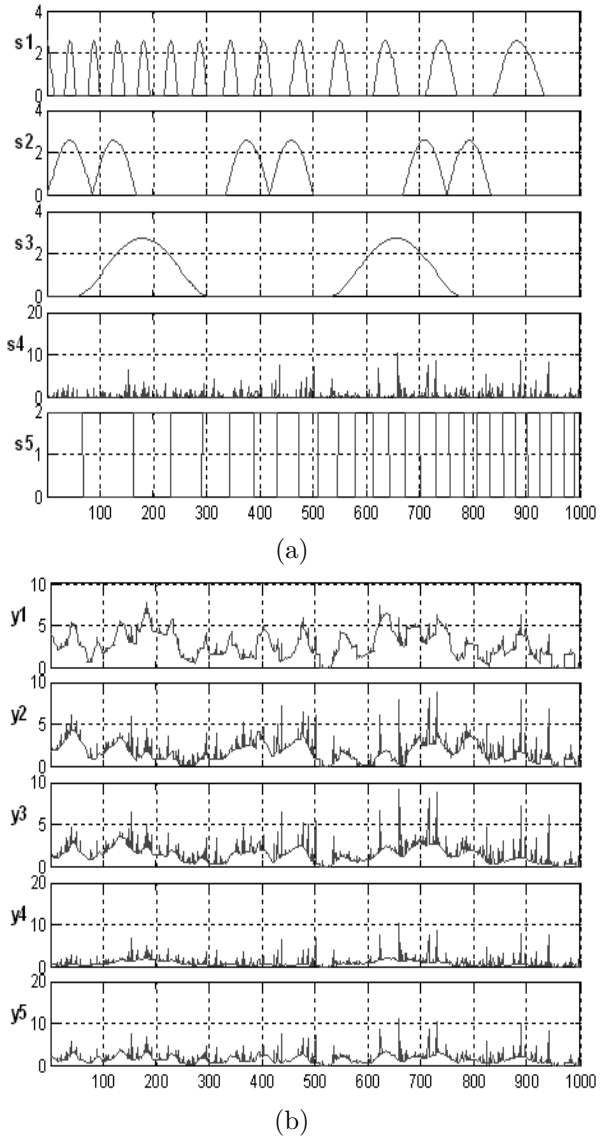


Fig. 1. (a) Original 5 source signals; (b) Observed 5 mixed signals with dense mixing matrix (noise-free data).

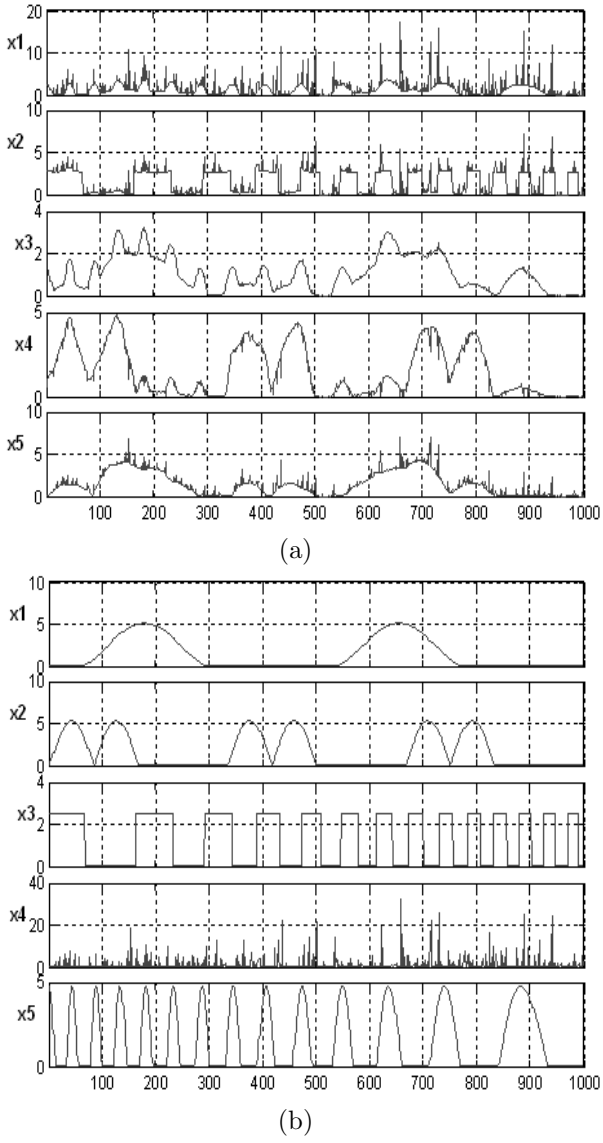


Fig. 2. Estimated sources with: (a) standard Lee-Seung algorithm (EMML) (SIR = 1.9, 11.4, 2.3, 11.4, 5.4 [dB], respectively); (b) FP for $\boldsymbol{X}$ (Step 1) and quasi-Newton for $\boldsymbol{A}$ (Step 2) (SIR = 141.7, 145.7, 138.8, 141.4, 154.2 [dB], respectively).

$\lambda_0 = 200$ and $\tau = 0.02$ in the exponential rule for computing $\boldsymbol{X}$. The parameters for the Levenberg-Marquardt regularization are set heuristically as follows: $\bar{\lambda}_0^{(H)} = 10^{-12}$, $\lambda_0^{(H)} = 10^8$ and $\tau^{(H)} = 1$ for the Euclidean distance, and $\bar{\lambda}_0^{(H)} = 10^{-12}$, $\lambda_0^{(H)} = 10^{15}$ and $\tau^{(H)} = 2$ for the $\alpha$-divergence.

We have also used four natural images (see Fig. 4 (a)) that have been mixed with another uniformly distributed random matrix $\boldsymbol{A} \in \mathbb{R}^{9\times 4}$ with $\text{cond}(\boldsymbol{A}) \simeq 4.5$. The mixtures are shown in Fig. 4 (b). Figs. 5 and 6 present the separation results. Again, the standard Lee-Seung algorithm fails to give satisfactory results, which is visible in Fig. 5 (a). The images separated with the hybrid algorithm: FP for $\boldsymbol{X}$ and quasi-Newton for $\boldsymbol{A}$ are illustrated in Fig. 5 (b). The results obtained with the GPCG applied for the Euclidean distance with 5 inner iteration ($k = 5$) are shown in Fig. 6. In this case, we have also used the multilayer technique that has been presented in [32,34,43]. Figs. 6 (a) and (b) show the images separated with the GPCG with one and three layers, respectively. The multi-

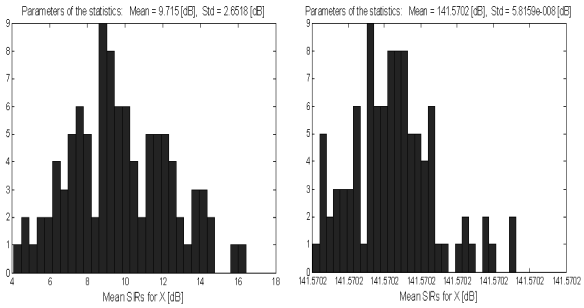


Fig. 3. Histograms from 100 mean-SIR samples generated with the following algorithms: (left) standard Lee-Seung algorithm (EMML); (right) FP for $\boldsymbol{X}$ (Step 1) and quasi-Newton for $\boldsymbol{A}$ (Step 2).

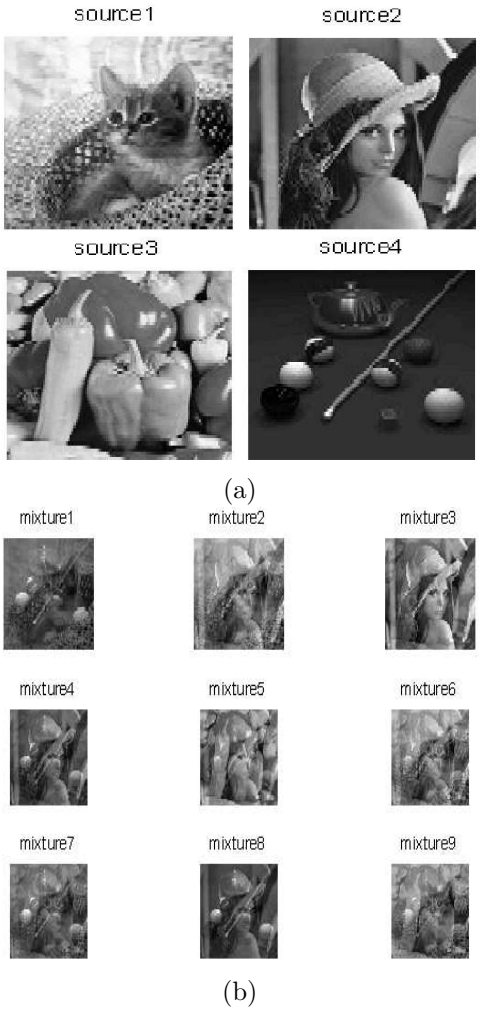


Fig. 4. (a) Original 4 source images; (b) observed 9 mixed images.

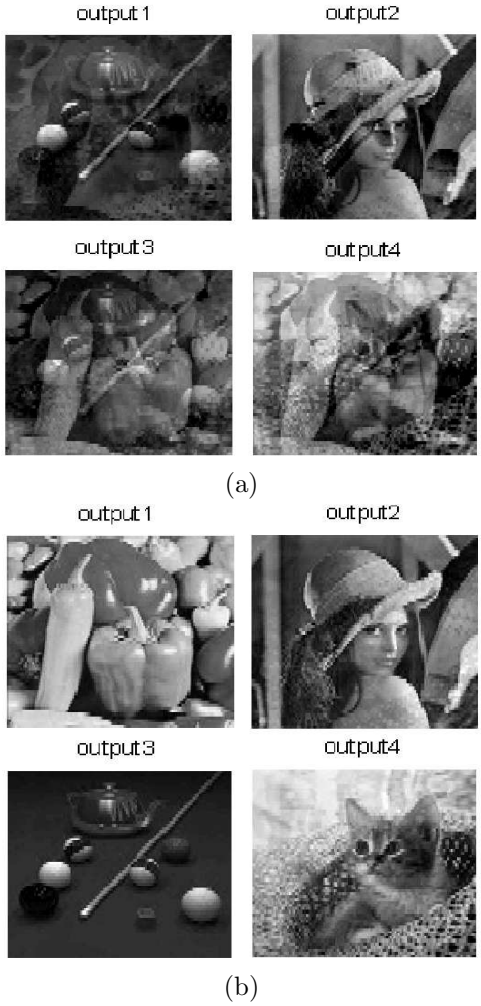


Fig. 5. Estimated sources with: (a) standard Lee-Seung algorithm (ISRA) (SIR = 9.5, 13.1, 8.3, 6.9 [dB], respectively); (b) FP for $\boldsymbol{X}$ (Step 1) and quasi-Newton for $\boldsymbol{A}$ (Step 2) (SIR = 39.6, 17.3, 32.5, 22.1 [dB], respectively).

layer technique also works very efficiently with the GPCG. However, the best results are obtained with the hybrid quasi-Newton and FP algorithm.

The simulations results confirmed that the developed algorithms are efficient and stable for a wide set of parameters, however the NMF problem cannot be too much ill-conditioned (especially mixing matrix $\boldsymbol{A}$). Additionally, to improve the ill-conditioning the rows in $\boldsymbol{Y}$ are scaled to have the unit variance.

## 6. CONCLUSIONS

In this paper, we proposed the second order methods to the NMF problem. We derived the new method for NMF, which uses the quasi-Newton it-

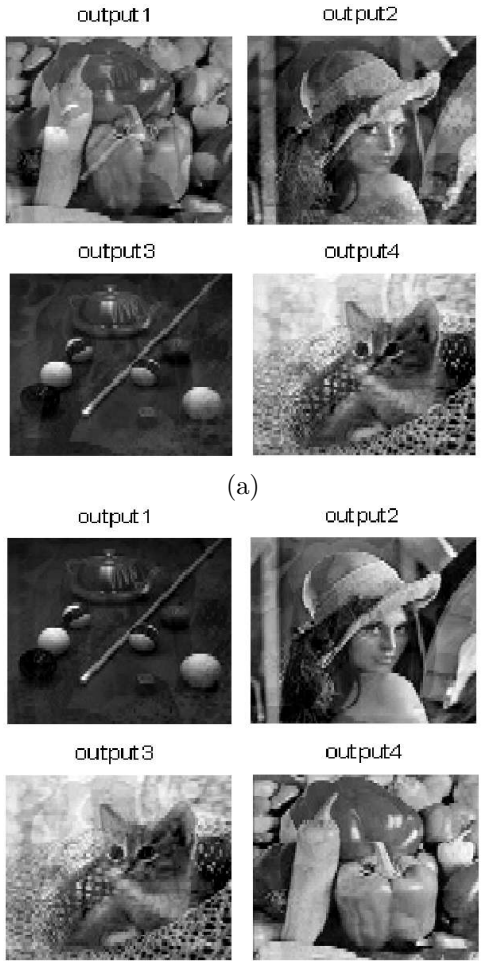


Fig. 6. Estimated sources with: (a) GPCG algorithm with one layer (SIR = 11.7, 12.9, 13.2, 19.5 [dB], respectively); (b) GPCG algorithm with three layers (SIR = 15.2, 18.5, 16.6, 18.6 [dB], respectively).

erates for updating $\boldsymbol{A}$ and the Fixed Point (FP) regularized least-squares algorithm for computing $\boldsymbol{X}$. As alternative approach, we have also developed the GPCG algorithm for NMF with the $\alpha$-divergence. Using synthetic data for BSS applications, we demonstrated the robustness and high performance of our new algorithms. We obtained the best results with the quasi-Newton FP algorithm. The GPCG gives slightly worse results but together with the multilayer technique it may be competitive to the quasi-Newton FP algorithm, especially for noisy or large-scale data. We have tested our algorithms on many difficult benchmarks of signals and images, and we believe that the GPCG algorithm can be also very useful for many NMF applications. The discussed algorithms have been implemented in the MATLAB toolbox: NMFLAB for Signal and Image Processing [45]. The Matlab source code of the hybrid algorithm: FP for computing $\boldsymbol{X}$ and quasi-Newton for $\boldsymbol{A}$ is included in Appendix.

## References


[1] D. Guillamet, J. Vitrià, B. Schiele, Introducing a weighted nonnegative matrix factorization for image classification, Pattern Recognition Letters 24 (14) (2003) 2447–2454.

[2] D. Guillamet, B. Schiele, J. Vitrià, Analyzing non-negative matrix factorization for image classification, in: 16th International Conference on Pattern Recognition (ICPR'02), Vol. 2, Quebec City, Canada, 2002, pp. 116–119.

[3] D. Guillamet, J. Vitrià, Classifying faces with nonnegative matrix factorization, in: Proc. 5th Catalan Conference for Artificial Intelligence, Castello de la Plana, Spain, 2002.

[4] J.-H. Ahn, S. Kim, J.-H. Oh, S. Choi, Multiple nonnegative-matrix factorization of dynamic PET images, in: ACCV, 2004.

[5] J. S. Lee, D. D. Lee, S. Choi, D. S. Lee, Application of nonnegative matrix factorization to dynamic positron emission tomography, in: 3rd International Conference on Independent Component Analysis and Blind Signal Separation, San Diego, CA, 2001, pp. 556–562.

[6] D. D. Lee, H. S. Seung, Learning the parts of objects by nonnegative matrix factorization, Nature 401 (1999) 788–791.

[7] H. Li, T. Adali, D. E. W. Wang, Non-negative matrix factorization with orthogonality constraints for chemical agent detection in raman spectra, in: IEEE Workshop on Machine Learning for Signal Processing, Mystic, USA, 2005.

[8] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, A. Pascual-Montano, Biclustering of gene expression data by non-smooth non-negative matrix factorization, BMC Bioinformatics 7 (78).

[9] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmean, R. Pascual-Marqui, Nonsmooth nonnegative matrix factorization (nsNMF), IEEE Trans. Pattern Analysis and Machine Intelligence 28 (3) (2006) 403–415.

[10] F. Shahnaz, M. Berry, P. Pauca, R. Plemmons, Document clustering using non-negative matrix factorization, Journal on Information Processing and Management 42 (2006) 373–386.

[11] O. Okun, H. Priisalu, Fast nonnegative matrix factorization and its application for protein fold recognition, EURASIP Journal on Applied Signal Processing 2006 (2006) Article ID 71817, 8 pages.

[12] Y. Wang, Y. Jia, C. Hu, M. Turk, Non-negative matrix factorization framework for face recognition, International Journal of Pattern Recognition and Artificial Intelligence 19 (4) (2005) 495–511.

## APPENDIX

```
% Second-order NMF algorithm
function [A,X] = nmf_newton(Y,R,CostFun,MaxIter,Alpha0,Tau,Alpha)
% INPUTS:
% Y:     Data matrix (M by T): (model Y = AX  s.t. nonnegativity constraints in A and X),
% R:     Low-rank,
% CostFun:  Cost function: 1 - Frobenius norm (default), 2 - Alpha-divergence,
% MaxIter:  Max. number of alternating steps (default: MaxIter = 1500),
% Alpha0:   Initial value of regular.parameter (default: Alpha0 = 100),
% Tau:    Damping factor (default: Tau = 50),
% Alpha:   Free parameter in alpha-divergence (default: Alpha = 1),
%
% OUTPUT:
% A and X:  Estimated NMF factors,
% ========================================================================
if (nargin < 7) | isempty(Alpha) | max(size(Alpha) > 1)
  Alpha = 1; fprintf(1,'Alpha = %f is set as default \n',Alpha); end
if (nargin < 6) | isempty(Tau) | max(size(Tau) > 1)
  Tau = 50; fprintf(1,'Tau = %f is set as default \n',Tau); end
if (nargin < 5) | isempty(Alpha0) | max(size(Alpha0) > 1)
  Alpha0 = 100; fprintf(1,'Alpha0 = %f is set as default \n',Alpha0); end
if (nargin < 4) | isempty(MaxIter) | max(size(MaxIter) > 1)
  MaxIter = 1500; fprintf(1,'MaxIter = %f is set as default \n',MaxIter);
end if (nargin < 3) | isempty(CostFun) | max(size(CostFun) > 1)
  CostFun = 1; disp('Frobenius norm is set as default'); end
if (nargin < 2) | isempty(R)
  disp('Rank of factorization must be given'); return; end
if isempty(Y) | isnan(Y) error('No data'); return; end
if min(min(Y)) < 0 % Test for negative values in Y
  disp('Some entries in Y are changed from negative to small positive');
  Y(Y< 0) = eps; end

% Scaling to unit-variance
mx = 1./sqrt(var(Y,0,2) + eps);  Y = repmat(mx,[1,size(Y,2),1]).*Y;

% Settings
k = 0; [M,T]=size(Y); I = eye(R); A = []; X = [];
lambda = 1E-12; % Levenberg-Marquardt regularization of Hessian
HA = spalloc(M*R,M*R,M*R^2); % Spase allocation for Hessian
while 1 A = rand(M,R); if cond(A) < 50 break; end; end % Initialization

% Alternatings
while k <=MaxIter
   k = k + 1;
   alpha_reg = Alpha0*exp(-k/Tau); % exponential rule for regular.param.
   if isnan(A)  disp('Matrix A is too much ill-conditioned. Try again.'); break; end
   if cond(A) > 1E6 alphaX = 1E-6; else alphaX = 0; end
   X = max(1E6*eps,pinv(A'*A + alpha_reg + alphaX*I)*A'*Y); %Updating of X
```

```
  switch CostFun
    case 1 % Frobenius norm

      hA = X*X';
      hA = hA + (lambda + 1E8*exp(-k))*eye(R); % Levenberg-Marquardt regularization
      GA = X*Y' - hA*A'; % Gradient
      HA = kron(speye(M),-hA); % Hessian

    case 2 % Alpha-divergence

      if Alpha == 0  % Dual Kullback-Leibler divergence
      Z = A*X+100*eps; Zx = 1./Z;
      GA = X*(log(Z./(Y+eps)))';  % Gradient
      for i = 1:M
        HA(((i-1)*R+1):i*R,((i-1)*R+1):i*R) = (X.*repmat(Zx(i,:),R,1))*X'; % Hessian
      end
      else  % Alpha-divergence
      Z = A*X+100*eps; Zx = ((Y+eps).^Alpha)./(Z.^(Alpha + 1));
        for i = 1:M
          HA(((i-1)*R+1):i*R,((i-1)*R+1):i*R) = (X.*repmat(Zx(i,:),R,1))*X'; % Hessian
        end
      GA = (1/Alpha)*X*(1-((Y+eps)./Z).^Alpha)';  % Gradient
      end
      HA = HA + (lambda + 1E15*exp(-2*k))*speye(M*R);
  end
[WA,RA] = qr(HA,GA(:));  % Q-less QR factorization
cond_RA = condest(RA); if isinf(cond_RA)
fprintf(1,'Upper-triangular matrix R is singular in %d iteration(s). Restart is needed.\n',k)
break; end
A = A'; A(:) = A(:) - .9*RA\WA;  % Newton iterations
A(A <= 0) = 1E2*eps;  A = A';
A = A*diag(1./sum(A,1));  % Normalization to unit L1-column norm
end  % For alternatings
A = repmat(1./mx,[1,size(A,2),1]).*A;  % De-scaling
```


[13] W. Liu, N. Zheng, Non-negative matrix factorization based methods for object recognition, Pattern Recognition Letters 25 (8) (2004) 893–897.

[14] M. W. Spratling, Learning image components for object recognition, Journal of Machine Learning Research 7 (2006) 793–815.

[15] P. Sajda, S. Du, T. R. Brown, R. S. D. C. Shungu, X. Mao, L. C. Parra, Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain, IEEE Trans. Medical Imaging 23 (12) (2004) 1453–1465.

[16] P. Sajda, S. Du, T. Brown, L. Parra, R. Stoyanova, Recovery of constituent spectra in 3d chemical shift imaging using nonnegative matrix factorization, in: 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, 2003, pp. 71–76.

[17] P. Sajda, S. Du, L. Parra, Recovery of constituent spectra using non-negative matrix factorization, in: Proceedings of SPIE – Volume 5207), Wavelets: Applications in Signal and Image Processing, 2003, pp. 321–331.

[18] I. S. Dhillon, D. M. Modha, Concept decompositions for large sparse text data using clustering, Machine Learning J. 42 (2001) 143–175.

[19] Y. C. Cho, S. Choi, Nonnegative features of spectro-temporal sounds for classification, Pattern Recognition Letters 26 (2005) 1327–1336.

[20] J.-P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, Vol. 101, PNAS, 2000, pp. 4164–4169.

[21] N. Rao, S. J. Shepherd, D. Yao, Extracting characteristic patterns from genome – wide expression data by non-negative matrix factorization, in: Proc. of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB 2004), Stanford, CA, 2004.

[22] A. Cichocki, R. Zdunek, S. Amari, Csiszar's divergences for non-negative matrix factorization: Family of new algorithms, LNCS 3889 (2006) 32–39.

[23] R. Richardson, Bayesian-based iterative method of

image restoration, J. Opt. Soc. Am. 62 (1) (1972) 55–59.
[24] L. Lucy, An iterative technique for the rectification of observed distributions, The Astronomical J. 79 (6) (1974) 745–754.
[25] K. Lange, R. Carson, EM reconstruction algorithms for emission and transmission tomography, J. Comp. Assisted Tomo. 8 (2) (1984) 306–316.
[26] C. Byrne, Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods, IEEE Transactions on Image Processing 7 (1998) 100 – 109.
[27] A. R. D. Pierro, On the relation between the ISRA and the EM algorithm for positron emission tomography, IEEE Trans. Medial Imaging 12 (2) (1993) 328–333.
[28] P. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research 5 (2004) 1457–1469.
[29] C.-J. Lin, Projected gradient methods for non-negative matrix factorization, Neural ComputationIn press.
URL http://www.csie.ntu.edu.tw/~cjlin
[30] M. T. Chu, F. Diele, R. Plemmons, S. Ragni, Optimality, computation, and interpretation of nonnegative matrix factorizations, SIAM Journal on Matrix Analysis and Applications (2004) submitted.
URL http://www.wfu.edu/~plemmons
[31] M. Berry, M. Browne, A. Langville, P. Pauca, R. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Computational Statistics and Data AnalysisSubmitted.
[32] A. Cichocki, R. Zdunek, Multilayer nonnegative matrix factorization, Electronics Letters 42 (16) (2006) 947–948.
[33] A. Cichocki, R. Zdunek, Regularized alternating least squares algorithms for non-negative matrix/tensor factorization, in: 4th International Symposium on Neural Networks, Nanjing, China, 2007, submitted.
[34] A. Cichocki, R. Zdunek, Multilayer nonnegative matrix factorization using projected gradient approaches, in: 13th International Conference on Neural Information Processing, Hong Kong, 2006.
[35] M. Merritt, Y. Zhang, An interior-point gradient method for large-scale totally nonnegative least squares problems, J. Optimization Theory and Applications 126 (1) (2005) 191–202.
[36] R. Zdunek, A. Cichocki, Non-negative matrix factorization with quasi-Newton optimization, LNAI 4029 (2006) 870–879.
[37] J. J. More, G. Toraldo, On the solution of large quadratic programming problems with bound constraints, SIAM J. Optimization 1 (1) (1991) 93–113.
[38] J. M. Bardsley, C. R. Vogel, Nonnegatively constrained convex programming method for image reconstruction, SIAM J. Sci. Comput. 4 (2004) 1326–1343.
[39] J. M. Bardsley, A nonnegatively constrained trust region algorithm for the restoration of images with an unknown blur, Electronic Transactions on Numerical Analysis 20 (2005) 139–153.
[40] M. R. Hestenes, E. Stiefel, Method of conjugate gradients for solving linear systems, J. Res. Nat. Bur. Standards 49 (1952) 409–436.
[41] S. Amari, Differential-Geometrical Methods in Statistics, Springer Verlag, 1985.
[42] N. A. Cressie, T. Read, Goodness-of-Fit Statistics for Discrete Multivariate Data, Springer, New York, 1988.
[43] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, Z. He, Extended SMART algorithms for non-negative matrix factorization, LNAI 4029 (2006) 548–562.
[44] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, Sparse solutions to linear inverse problems with multiple measurement vectors, IEEE Trans. Signal Processing 53 (7) (2005) 2477–2488.
[45] A. Cichocki, R. Zdunek, NMFLAB for Signal and Image Processing, Tech. rep., Laboratory for Advanced Brain Signal Processing, BSI RIKEN, Saitama, Japan (2006).
URL http://www.bsp.brain.riken.jp