Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2008, Article ID 939567, 13 pages
doi:10.1155/2008/939567

Research Article

Fast Nonnegative Matrix Factorization Algorithms Using Projected Gradient Approaches for Large-Scale Problems

Rafal Zdunek[1] and Andrzej Cichocki[2,3,4]

[1] *Instiute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland*
[2] *Laboratory for Advanced Brain Signal Processing, Brain Science Institute RIKEN, Wako-shi, Saitama 351-0198, Japan*
[3] *Institute of Theory of Electrical Engineering, Measurement and Information Systems, Faculty of Electrical Engineering, Warsaw University of Technology, 00-661 Warsaw, Poland*
[4] *Systems Research Institute, Polish Academy of Science (PAN), 01-447 Warsaw, Poland*

Correspondence should be addressed to Rafal Zdunek, rafal.zdunek@pwr.wroc.pl

Received 15 January 2008; Revised 18 April 2008; Accepted 22 May 2008

Recommended by Wenwu Wang

Recently, a considerable growth of interest in projected gradient (PG) methods has been observed due to their high efficiency in solving large-scale convex minimization problems subject to linear constraints. Since the minimization problems underlying nonnegative matrix factorization (NMF) of large matrices well matches this class of minimization problems, we investigate and test some recent PG methods in the context of their applicability to NMF. In particular, the paper focuses on the following modified methods: projected Landweber, Barzilai-Borwein gradient projection, projected sequential subspace optimization (PSESOP), interior-point Newton (IPN), and sequential coordinate-wise. The proposed and implemented NMF PG algorithms are compared with respect to their performance in terms of signal-to-interference ratio (SIR) and elapsed time, using a simple benchmark of mixed partially dependent nonnegative signals.

1. Introduction and Problem Statement

Nonnegative matrix factorization (NMF) finds such nonnegative factors (matrices) $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{I\times J}$ and $\mathbf{X} = [x_{jt}] \in \mathbb{R}^{J\times T}$ with a $a_{ij} \geq 0$, $x_{jt} \geq 0$ that $\mathbf{Y} \cong \mathbf{AX}$, given the observation matrix $\mathbf{Y} = [y_{it}] \in \mathbb{R}^{I\times T}$, the lower-rank J, and possibly other statistical information on the observed data or the factors to be estimated.

This method has found a variety of real-world applications in the areas such as blind separation of images and nonnegative signals [1–6], spectra recovering [7–10], pattern recognition and feature extraction [11–16], dimensionality reduction, segmentation and clustering [17–32], language modeling, text mining [25, 33], music transcription [34], and neurobiology (gene separation) [35, 36].

Depending on an application, the estimated factors may have different interpretation. For example, Lee and Seung [11] introduced NMF as a method for decomposing an image (face) into parts-based representations (parts reminiscent of features such as lips, eyes, nose, etc.). In blind source separation (BSS) [1, 37, 38], the matrix $\mathbf{Y}$ represents the observed mixed (superposed) signals or images, $\mathbf{A}$ is a mixing operator, and $\mathbf{X}$ is a matrix of true source signals or images. Each row of $\mathbf{Y}$ or $\mathbf{X}$ is a signal or 1D image representation, where I is a number of observed mixed signals and J is a number of hidden (source) components. The index t usually denotes a sample (discrete time instant), where T is the number of available samples. In BSS, we usually have $T \gg I \geq J$, and J is known or can be relatively easily estimated using SVD or PCA.

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

Obviously, NMF is not unique in general case, and it is characterized by a scale and permutation indeterminacies.

These problems have been addressed recently by many researchers [39–42], and in this paper, the problems will not be discussed. However, we have shown by extensive computer simulations that in practice with overwhelming probability we are able to achieve a unique nonnegative factorization (neglecting unavoidable scaling and permutation ambiguities) if data are sufficiently sparse and a suitable NMF algorithm is applied [43]. This is consistent with very recent theoretical results [40].

The noise distribution is strongly application-dependent, however, in many BSS applications, a Gaussian noise is expected. Here our considerations are restricted to this case, however, the alternative NMF algorithms optimized to different distributions of the noise exist and can be found, for example, in [37, 44, 45].

NMF was proposed by Paatero and Tapper [46, 47] but Lee and Seung [11, 48] highly popularized this method by using simple multiplicative algorithms to perform NMF. An extensive study on convergence of multiplicative algorithms for NMF can be found in [49]. In general, the multiplicative algorithms are known to be very slowly convergent for large-scale problems. Due to this reason, there is a need to search more efficient and fast algorithms for NMF. Many approaches have been proposed in the literature to relax these problems. One of them is to apply projected gradient (PG) algorithms [50–53] or projected alternating least-squares (ALS) algorithms [33, 54, 55] instead of multiplicative ones. Lin [52] suggested applying the Armijo rule to estimate the learning parameters in projected gradient updates for NMF. The NMF PG algorithms proposed by us in [53] also address the issue with selecting such a learning parameter that is the steepest descent and also keeps some distance to a boundary of the nonnegative orthant (subspace of real nonnegative numbers). Another very robust technique concerns exploiting the information from the second-order Taylor expansion term of a cost function to speed up the convergence. This approach was proposed in [45, 56], where the mixing matrix $\mathbf{A}$ is updated with the projected Newton method, and the sources in $\mathbf{X}$ are computed with the projected least-squares method (the fixed point algorithm).

In this paper, we extend our approach to NMF that we have initialized in [53]. We have investigated, extended, and tested several recently proposed PG algorithms such as the oblique projected Landweber [57], Barzilai-Borwein gradient projection [58], projected sequential subspace optimization [59, 60], interior-point Newton [61], and sequential coordinate-wise [62]. All the presented algorithms in this paper are quite efficient for solving large-scale minimization problems subject to nonnegativity and sparsity constraints.

The main objective of this paper is to develop, extend, and/or modify some of the most promising PG algorithms to a standard NMF problem and to find optimal conditions or parameters for such a class of NMF algorithms. The second objective is to compare the performance and complexity of these algorithms for NMF problems, and to discover or establish the most efficient and promising algorithms. We would like to emphasize that most of the discussed algorithms have not been implemented neither used till now or even tested before for NMF problems, but they have been rather considered for solving a standard system of algebraic equations: $\mathbf{A}\mathbf{x}(k) = \mathbf{y}(k)$ (for only $k = 1$) where the matrix $\mathbf{A}$ and the vectors $\mathbf{y}$ are known. In this paper, we consider a much more difficult and complicated problem in which we have two sets of parameters and additional constraints of nonnegativity and/or sparsity. So it was not clear till now whether such algorithms would work efficiently for NMF problems, and if so, what kind of projected algorithms is the most efficient? To our best knowledge only the Lin-PG NMF algorithm is widely used and known for NMF problems. We have demonstrated experimentally that there are several novel PG gradient algorithms which are much more efficient and consistent than the Lin-PG algorithm.

In Section 2, we briefly discuss the PG approach to NMF. Section 3 describes the tested algorithms. The experimental results are illustrated in Section 4. Finally, some conclusions are given in Section 5.

2. Projected Gradient Algorithms

In contrast to the multiplicative algorithms, the class of PG algorithms has additive updates. The algorithms discussed here approximately solve nonnegative least squares (NNLS) problems with the basic alternating minimization technique that is used in NMF:

$$\min_{\mathbf{x}_t \geq 0} D_F(\mathbf{y}_t || \mathbf{A}\mathbf{x}_t) = \frac{1}{2}||\mathbf{y}_t - \mathbf{A}\mathbf{x}_t||_2^2, \quad t = 1, \ldots, T, \tag{1}$$

$$\min_{\underline{\mathbf{a}}_i \geq 0} D_F(\underline{\mathbf{y}}_i || \mathbf{X}^T \underline{\mathbf{a}}_i) = \frac{1}{2}||\underline{\mathbf{y}}_i - \mathbf{X}^T \underline{\mathbf{a}}_i||_2^2, \quad i = 1, \ldots, I \tag{2}$$

or in the equivalent matrix form

$$\min_{x_{jt} \geq 0} D_F(\mathbf{Y} || \mathbf{A}\mathbf{X}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2, \tag{3}$$

$$\min_{a_{ij} \geq 0} D_F(\mathbf{Y}^T || \mathbf{X}^T \mathbf{A}^T) = \frac{1}{2}||\mathbf{Y}^T - \mathbf{X}^T \mathbf{A}^T||_F^2, \tag{4}$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_J] \in \mathbb{R}^{I \times J}$, $\mathbf{A}^T = [\underline{\mathbf{a}}_1, \ldots, \underline{\mathbf{a}}_I] \in \mathbb{R}^{J \times I}$, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{J \times T}$, $\mathbf{X}^T = [\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_J] \in \mathbb{R}^{T \times J}$, $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] \in \mathbb{R}^{I \times T}$, $\mathbf{Y}^T = [\underline{\mathbf{y}}_1, \ldots, \underline{\mathbf{y}}_I] \in \mathbb{R}^{I \times T}$, and usually $I \geq J$. The matrix $\mathbf{A}$ is assumed to be a full-rank matrix, so there exists a unique solution $\mathbf{X}^* \in \mathbb{R}^{J \times T}$ for any matrix $\mathbf{Y}$ since the NNLS problem is strictly convex (with respect to one set of variables $\{\mathbf{X}\}$).

The solution $\mathbf{x}_t^*$ to (1) satisfies the Karush-Kuhn-Tucker (KKT) conditions:

$$\mathbf{x}_t^* \geq 0, \qquad \mathbf{g}_X(\mathbf{x}_t^*) \geq 0, \qquad \mathbf{g}_X(\mathbf{x}_t^*)^T \mathbf{x}_t^* = 0, \tag{5}$$

or in the matrix notation

$$\mathbf{X}^* \geq 0, \qquad \mathbf{G}_X(\mathbf{X}^*) \geq 0, \qquad \mathrm{tr}\{\mathbf{G}_X(\mathbf{X}^*)^T \mathbf{X}^*\} = 0, \tag{6}$$

where $\mathbf{g}_X$ and $\mathbf{G}_X$ are the corresponding gradient vector and gradient matrix:

$$\mathbf{g}_X(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} D_F(\mathbf{y}_t || \mathbf{A}\mathbf{x}_t) = \mathbf{A}^T(\mathbf{A}\mathbf{x}_t - \mathbf{y}_t), \tag{7}$$

$$\mathbf{G}_X(\mathbf{X}) = \nabla_{\mathbf{X}} D_F(\mathbf{Y} || \mathbf{A}\mathbf{X}) = \mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}). \tag{8}$$

Similarly, the KKT conditions for the solution $\underline{\mathbf{a}}^*$ to (2), and the solution $\mathbf{A}^*$ to (4) are as follows:

$$\underline{\mathbf{a}}_i^* \geq 0, \qquad \mathbf{g}_A(\underline{\mathbf{a}}_i^*) \geq 0, \qquad \mathbf{g}_A(\underline{\mathbf{a}}_i^*)^T \underline{\mathbf{a}}_i^* = 0, \tag{9}$$

or in the matrix notation:

$$\mathbf{A}^* \geq 0, \qquad \mathbf{G}_A(\mathbf{A}^*) \geq 0, \qquad \mathrm{tr}\{\mathbf{A}^* \mathbf{G}_A(\mathbf{A}^*)^T\} = 0, \tag{10}$$

where $\mathbf{g}_A$ and $\mathbf{G}_A$ are the gradient vector and gradient matrix of the objective function:

$$\begin{aligned} \mathbf{g}_A(\underline{\mathbf{a}}_t) &= \nabla_{\underline{\mathbf{a}}_i} D_F(\underline{\mathbf{y}}_i || \mathbf{X}^T \underline{\mathbf{a}}_i) = (\mathbf{X}^T \underline{\mathbf{a}}_i - \underline{\mathbf{y}}_i), \\ \mathbf{G}_A(\mathbf{A}) &= \nabla_{\mathbf{A}} D_F(\mathbf{Y}^T || \mathbf{X}^T \mathbf{A}^T) = (\mathbf{A}\mathbf{X} - \mathbf{Y})\mathbf{X}^T. \end{aligned} \tag{11}$$

There are many approaches to solve the problems (1) and (2), or equivalently (3) and (4). In this paper, we discuss selected projected gradient methods that can be generally expressed by iterative updates:

$$\begin{aligned} \mathbf{X}^{(k+1)} &= P_\Omega[\mathbf{X}^{(k)} - \eta_X^{(k)} \mathbf{P}_X^{(k)}], \\ \mathbf{A}^{(k+1)} &= P_\Omega[\mathbf{A}^{(k)} - \eta_A^{(k)} \mathbf{P}_A^{(k)}], \end{aligned} \tag{12}$$

where $P_\Omega[\xi]$ is a projection of ξ onto a convex feasible set $\Omega = \{\xi \in \mathbb{R} : \xi \geq 0\}$, namely, the nonnegative orthant $\mathbb{R}_+$ (the subspace of nonnegative real numbers), $\mathbf{P}_X^{(k)}$ and $\mathbf{P}_A^{(k)}$ are descent directions for $\mathbf{X}$ and $\mathbf{A}$, and $\eta_X^{(k)}$ and $\eta_A^{(k)}$ are learning rules, respectively.

The projection $P_\Omega[\xi]$ can be performed in several ways. One of the simplest techniques is to replace all negative entries in ξ by zero-values, or in practical cases, by a small positive number ϵ to avoid numerical instabilities. Thus,

$$P[\xi] = \max\{\epsilon, \xi\}. \tag{13}$$

However, this is not the only way to carry out the projection $P_\Omega(\xi)$. It is typically more efficient to choose the learning rates $\eta_X^{(k)}$ and $\eta_A^{(k)}$ so as to preserve nonnegativity of the solutions. The nonnegativity can be also maintained by solving least-squares problems subject to the constraints (6) and (10). Here we present the exemplary PG methods that work for NMF problems quite efficiently, and we implemented them in the Matlab toolboxm, NMFLAB/NTFLAB, for signal and image processing [43]. For simplicity, we focus our considerations on updating the matrix $\mathbf{X}$, assuming that the updates for $\mathbf{A}$ can be obtained in a similar way. Note that the updates for $\mathbf{A}$ can be readily obtained solving the transposed system $\mathbf{X}^T\mathbf{A}^T = \mathbf{Y}^T$, having $\mathbf{X}$ fixed (updated in the previous step).

3. Algorithms

3.1. Oblique Projected Landweber Method

The Landweber method [63] performs gradient-descent minimization with the following iterative scheme:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \eta \mathbf{G}_X^{(k)}, \tag{14}$$

Set $\mathbf{A}$, $\mathbf{X}$, % Initialization
$\eta_j^{(X)} = \dfrac{2}{(\mathbf{A}^T\mathbf{A}\mathbf{1}_J)_j}$,
For $k = 1, 2, \ldots$, % Inner iterations for X
$\mathbf{G}_X = \mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y})$, % Gradient with respect to X
$\mathbf{X} \leftarrow P_\Omega[\mathbf{X} - \mathrm{diag}\{\eta_j\}\mathbf{G}_X]$, % Updating
End

Algorithm 1: (OPL).

where the gradient is given by (8), and the learning rate $\eta \in (0, \eta_{\max})$. The updating formula assures an asymptotical convergence to the minimal-norm least squares solution for the convergence radius defined by

$$\eta_{\max} = \frac{2}{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}, \tag{15}$$

where $\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ is the maximal eigenvalue of $\mathbf{A}^T\mathbf{A}$. Since $\mathbf{A}$ is a nonnegative matrix, we have $\lambda_{\max}(\mathbf{A}^T\mathbf{A}) \leq \max_j[\mathbf{A}^T\mathbf{A}\mathbf{1}_J]_j$, where $\mathbf{1}_J = [1, \ldots, 1]^T \in \mathbb{R}^J$. Thus the modified Landweber iterations can be expressed by the formula

$$\mathbf{X}^{(k+1)} = P_\Omega[\mathbf{X}^{(k)} - \mathrm{diag}\{\eta_j\}\mathbf{G}_X^{(k)}], \quad \text{where } \eta_j = \frac{2}{(\mathbf{A}^T\mathbf{A}\mathbf{1}_J)_j}. \tag{16}$$

In the obliqueprojected Landweber (OPL) method [57], which can be regarded as a particular case of the PG iterative formula (12), the solution obtained with (14) in each iterative step is projected onto the feasible set. Finally, we have Algorithm 1 for updating $\mathbf{X}$.

3.2. Projected Gradient

One of the fundamental PG algorithms for NMF was proposed by Lin in [52]. This algorithm, which we refer to as the Lin-PG, uses the Armijo rule along the projection arc to determine the steplengths $\eta_X^{(k)}$ and $\eta_A^{(k)}$ in the iterative updates (12). For the cost function being the squared Euclidean distance, $\mathbf{P}_X = (\mathbf{A}^{(k)})^T(\mathbf{A}^{(k)}\mathbf{X}^{(k)} - \mathbf{Y})$ and $\mathbf{P}_A = (\mathbf{A}^{(k)}\mathbf{X}^{(k+1)} - \mathbf{Y})(\mathbf{X}^{(k+1)})^T$.

For computation of $\mathbf{X}$, such a value of η_X is decided, on which

$$\eta_X^{(k)} = \beta^{m_k}, \tag{17}$$

where m_k is the first nonnegative integer m that satisfies

$$\begin{aligned} &D_F(\mathbf{Y}||\mathbf{A}\mathbf{X}^{(k+1)}) - D_F(\mathbf{Y}||\mathbf{A}\mathbf{X}^{(k)}) \\ &\quad \leq \sigma \nabla_X D_F(\mathbf{Y}||\mathbf{A}\mathbf{X}^{(k)})^T(\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}). \end{aligned} \tag{18}$$

The parameters $\beta \in (0, 1)$ and $\sigma \in (0, 1)$ decide about a convergence speed. In this algorithm we set $\sigma = 0.01, \beta = 0.1$ experimentally as default.

The Matlab implementation of the Lin-PG algorithm is given in [52].

Set $\mathbf{A}, \mathbf{X}, \alpha_{\min}, \alpha_{\max}, \boldsymbol{\alpha}^{(0)} \in [\alpha_{\min}, \alpha_{\max}] \in \mathbb{R}^T$ % Initialization
For $k = 1, 2, \ldots,$ % Inner iterations
$\boldsymbol{\Delta}^{(k)} = P_{\Omega}[\mathbf{X}^{(k)} - \alpha^{(k)} \nabla_X D_F(\mathbf{Y}||\mathbf{A}\mathbf{X}^{(k)})] - \mathbf{X}^{(k)}$,
$\boldsymbol{\lambda}^{(k)} = \arg\min_{\lambda_t^{(k)} \in [0,1]} D_F(\mathbf{Y}||\mathbf{A}(\mathbf{X} + \boldsymbol{\Delta}^{(k)} \mathrm{diag}\{\boldsymbol{\lambda}\}))$,
where $\boldsymbol{\lambda} = [\lambda_t] \in \mathbb{R}^T$,
$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \boldsymbol{\Delta}^{(k)} \mathrm{diag}\{\boldsymbol{\lambda}\}$,
$\boldsymbol{\gamma}^{(k)} = \mathrm{diag}\{(\boldsymbol{\Delta}^{(k)})^T \mathbf{A}^T \mathbf{A} \boldsymbol{\Delta}^{(k)}\}$,
If $\gamma_t^{(k)} = 0$: $\alpha_t^{(k+1)} = \alpha_{\max}$,
Else $\alpha_t^{(k+1)} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, [(\boldsymbol{\Delta}^{(k)})^T \boldsymbol{\Delta}^{(k)}]_{tt} / \gamma_t^{(k)}\}\}$,
End
End % Inner iterations

Algorithm 2: (GPSR-BB).

Set $\mathbf{A}, \mathbf{x}_t^{(0)}, p$ % Initialization
For $k = 1, 2, \ldots,$ % Inner iterations
$\mathbf{d}_1^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(0)}$,
$\mathbf{g}^{(k)} = \nabla_{\mathbf{x}_t} D_F(\mathbf{y}_t||\mathbf{A}\mathbf{x}_t)$,
$\mathbf{G}^{(p)} = [\mathbf{g}^{(k-1)}, \mathbf{g}^{(k-2)}, \ldots, \mathbf{g}^{(k-p)}] \in \mathbb{R}^{J \times p}$,
$w_k = \begin{cases} 1 & \text{if } k = 1, \\ 1/2 + \sqrt{1/4 + w_{k-1}^2} & \text{if } k > 1, \end{cases}$
$\mathbf{w}^{(k)} = [w_k, w_{k-1}, \ldots, w_{k-p+1}]^T \in \mathbb{R}^p$,
$\mathbf{d}_2^{(k)} = \mathbf{G}^{(p)} \mathbf{w}^{(k)}$,
$\mathbf{D}^{(k)} = [\mathbf{d}_1^{(k)}, \mathbf{d}_2^{(k)}, \mathbf{g}^{(k)}, \mathbf{G}^{(p)}]$,
$\boldsymbol{\alpha}_*^{(k)} = \arg\min_{\alpha} D_F(\mathbf{y}_t||\mathbf{A}(\mathbf{x}_t^{(k)} + \mathbf{D}^{(k)} \boldsymbol{\alpha}^{(k)}))$,
$\mathbf{x}^{(k+1)} = P_{\Omega}[\mathbf{x}^{(k)} + \mathbf{D}^{(k)} \boldsymbol{\alpha}_*^{(k)}]$
End % Inner iterations

Algorithm 3: (NMF-PSESOP).

3.3. Barzilai-Borwein Gradient Projection

The Barzilai-Borwein gradient projection method [58, 64] is motivated by the quasi-Newton approach, that is, the inverse of the Hessian is replaced with an identity matrix $\mathbf{H}_k = \alpha_k \mathbf{I}$, where the scalar α_k is selected so that the inverse Hessian has similar behavior as the true Hessian in the recent iteration. Thus,

$$\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)} \approx \alpha_k \left(\nabla_X D(\mathbf{Y}||\mathbf{A}^{(k)}\mathbf{X}^{(k+1)}) - \nabla_X D(\mathbf{Y}||\mathbf{A}^{(k)}\mathbf{X}^{(k)})\right). \tag{19}$$

In comparison to, for example, Lin's method [52], this method does not ensure that the objective function decreases at every iteration, but its general convergence has been proven analytically [58]. The general scheme of the Barzilai-Borwein gradient projection algorithm for updating $\mathbf{X}$ is in Algorithm 2.

Since $D_F(\mathbf{Y}||\mathbf{A}\mathbf{X})$ is a quadratic function, the line search parameter $\boldsymbol{\lambda}^{(k)}$ can be derived in the following closed-form formula:

$$\boldsymbol{\lambda}^{(k)} = \max\left\{0, \min\left\{1, \frac{\mathrm{diag}\{(\boldsymbol{\Delta}^{(k)})^T \nabla_X D_F(\mathbf{Y}||\mathbf{A}\mathbf{X})\}}{\mathrm{diag}\{(\boldsymbol{\Delta}^{(k)})^T \mathbf{A}^T \mathbf{A} \boldsymbol{\Delta}^{(k)}\}}\right\}\right\}. \tag{20}$$

The Matlab implementation of the GPSR-BB algorithm, which solves the system $\mathbf{AX} = \mathbf{Y}$ of multiple measurement vectors subject to nonnegativity constraints, is given in Algorithm 4 (see also NMFLAB).

3.4. Projected Sequential Subspace Optimization

The projected sequential subspace optimization (PSESOP) method [59, 60] carries out a projected minimization of a smooth objective function over a subspace spanned by several directions which include the current gradient and gradient from the previous iterations, and the Nemirovski directions. Nemirovski [65] suggested that convex smooth unconstrained optimization is optimal if the optimization is performed over a subspace which includes the current gradient $\mathbf{g}(\mathbf{x})$, the directions $\mathbf{d}_1^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(0)}$, and the linear combination of the previous gradients $\mathbf{d}_2^{(k)} = \sum_{n=0}^{k-1} w_n \mathbf{g}(\mathbf{x}_n)$ with the coefficients w_n, $n = 0, \ldots, k-1$. The directions should be orthogonal to the current gradient. Narkiss and Zibulevsky [59] also suggested to include another direction: $\mathbf{p}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$, which is motivated by a natural extension of the conjugate gradient (CG) method to a nonquadratic case. However, our practical observations showed that this direction does not have a strong impact on the NMF components, thus we neglected it in our NMF-PSESOP algorithm. Finally, we have Algorithm 3 for updating $\mathbf{x}_t$ which is a single column vector of $\mathbf{X}$.

The parameter p denotes the number of previous iterates that are taken into account to determine the current update.

The line search vector $\boldsymbol{\alpha}_*^{(k)}$ derived in a closed form for the objective function $D_F(\mathbf{y}_t||\mathbf{A}\mathbf{x}_t)$ is as follows:

$$\boldsymbol{\alpha}_*^{(k)} = -\left((\mathbf{D}^{(k)})^T \mathbf{A}^T \mathbf{A} \mathbf{D}^{(k)} + \lambda \mathbf{I}\right)^{-1} (\mathbf{D}^{(k)})^T \nabla_{\mathbf{x}_t} D_F(\mathbf{y}_t||\mathbf{A}\mathbf{x}_t). \tag{21}$$

The regularization parameter can be set as a very small constant to avoid instabilities in inverting a rank-deficient matrix in case that $\mathbf{D}^{(k)}$ has zero-value or dependent columns.

3.5. Interior Point Newton Algorithm

The interior point Newton (IPN) algorithm [61] solves the NNLS problem (1) by searching the solution satisfying the KKT conditions (5) which equivalently can be expressed by the nonlinear equations

$$\mathbf{D}(\mathbf{x}_t)\mathbf{g}(\mathbf{x}_t) = 0, \tag{22}$$

where $\mathbf{D}(\mathbf{x}_t) = \mathrm{diag}\{d_1(\mathbf{x}_t), \ldots, d_J(\mathbf{x}_t)\}$, $\mathbf{x}_t \geq 0$, and

$$d_j(\mathbf{x}_t) = \begin{cases} x_{jt} & \text{if } g_j(\mathbf{x}_t) \geq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{23}$$

```
% Barzilai-Borwein gradient projection (GPSR-BB) algorithm

%
function [X] = nmf_gpsr_bb(A,Y,X,no_iter)
%
% [X] = nmf_gpsr_bb(A,Y,X,no_iter) finds such matrix X that solves
% the equation AX = Y subject to nonnegativity constraints.
%
% INPUTS:
% A - system matrix of dimension [I by J]
% Y - matrix of observations [I by T]
% X - matrix of initial guess [J by T]
% no_iter - maximum number of iterations
%
% OUTPUTS:
% X - matrix of estimated sources [J by T]
%
% ###########################################################################
% Parameters
alpha_min = 1E-8; alpha_max = 1;
alpha = .1*ones(1,size(Y,2));
B = A'*A; Yt = A'*Y;

for k=1:no_iter

    G = B*X - Yt;
    delta = max(eps, X - repmat(alpha,size(G,1),1).*G) - X;
    deltaB = B*delta;
    lambda = max(0, min(1, -sum(delta.*G,1)./(sum(delta.*deltaB,1) + eps)));
    X = max(eps,X + delta.*repmat(lambda,size(delta,1),1));
    gamma = sum(delta.*deltaB,1) + eps;
    if gamma
        alpha = min(alpha_max, max(alpha_min, sum(delta.^2,1)./gamma ));
    else
        alpha = alpha_max;
    end
end
```

Algorithm 4

Applying the Newton method to (22), we have in the kth iterative step

$$(\mathbf{D}_k(\mathbf{x}_t)\mathbf{A}^T\mathbf{A} + \mathbf{E}_k(\mathbf{x}_t))\mathbf{p}_k = -\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t), \tag{24}$$

where

$$\mathbf{E}_k(\mathbf{x}_t) = \operatorname{diag}\{e_1(\mathbf{x}_t), \ldots, e_J(\mathbf{x}_t)\}. \tag{25}$$

In [61], the entries of the matrix $\mathbf{E}_k(\mathbf{x}_t)$ are defined by

$$e_j(\mathbf{x}_t) = \begin{cases} g_j(\mathbf{x}_t) & \text{if } 0 \le g_j(\mathbf{x}_t) < x_{jt}^{\gamma}, \text{ or } (g_j(\mathbf{x}_t))^{\gamma} > x_{jt}, \\ 1 & \text{otherwise,} \end{cases} \tag{26}$$

for $1 < \gamma \le 2$.

If the solution is degenerate, that is, $t = 1, \ldots, T$, $\exists j : x_{jt}^* = 0$, and $g_{jt} = 0$, the matrix $\mathbf{D}_k(\mathbf{x}_t)\mathbf{A}^T\mathbf{A} + \mathbf{E}_k(\mathbf{x}_t)$ may be singular. To avoid such a case, the system of equations has been rescaled to the following form:

$$\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{M}_k(\mathbf{x}_t)\mathbf{p}_k = -\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t) \tag{27}$$

with

$$\begin{aligned} \mathbf{M}_k(\mathbf{x}_t) &= \mathbf{A}^T\mathbf{A} + \mathbf{D}_k(\mathbf{x}_t)^{-1}\mathbf{E}_k(\mathbf{x}_t), \\ \mathbf{W}_k(\mathbf{x}_t) &= \operatorname{diag}\{w_1(\mathbf{x}_t), \ldots, w_J(\mathbf{x}_t)\}, \\ w_j(\mathbf{x}_t) &= (d_j(\mathbf{x}_t) + e_j(\mathbf{x}_t))^{-1}, \end{aligned} \tag{28}$$

for $\mathbf{x}_t > 0$. In [61], the system (27) is solved by the inexact Newton method, which leads to the following updates:

$$\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{M}_k(\mathbf{x}_t)\mathbf{p}_k = -\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t) + \mathbf{r}_k(\mathbf{x}_t), \tag{29}$$

$$\hat{\mathbf{p}}_k = \max\{\sigma, 1 - ||P_\Omega[\mathbf{x}_t^{(k)} + \mathbf{p}_k] - \mathbf{x}_t^{(k)}||_2\}(P_\Omega[\mathbf{x}_t^{(k)} + \mathbf{p}_k] - \mathbf{x}_t^{(k)}), \tag{30}$$

$$\mathbf{x}_t^{(k+1)} = \mathbf{x}_t^{(k)} + \hat{\mathbf{p}}_k, \tag{31}$$

where $\sigma \in (0, 1)$, $\mathbf{r}_k(\mathbf{x}_t) = \mathbf{A}^T(\mathbf{A}\mathbf{x}_t - \mathbf{y}_t)$, and $P_\Omega[\cdot]$ is a projection onto a feasible set Ω.

The transformation of the normal matrix $\mathbf{A}^T\mathbf{A}$ by the matrix $\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)$ in (27) means the system matrix $\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{M}_k(\mathbf{x}_t)$ is no longer symmetric and positive-definite. There are many methods for handling such systems of linear equations, like QMR [66], BiCG [67, 68], BiCGSTAB [69], or GMRES-like methods [70], however, they are more complicated and computationally demanding than, for example, the basic CG algorithm [71]. To apply the CG algorithm the system matrix in (27) must be converted to a positive-definite symmetric matrix, which can be easily done with normal equations. The methods like CGLS [72] or LSQR [73] are therefore suitable for such tasks. The transformed system has the form

$$\mathbf{Z}_k(\mathbf{x}_t)\widetilde{\mathbf{p}}_k = -\mathbf{S}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t) + \widetilde{\mathbf{r}}_k(\mathbf{x}_t), \tag{32}$$

$$\mathbf{S}_k(\mathbf{x}_t) = \sqrt{\mathbf{W}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)}, \tag{33}$$

$$\begin{aligned}\mathbf{Z}_k(\mathbf{x}_t) &= \mathbf{S}_k(\mathbf{x}_t)\mathbf{M}_k(\mathbf{x}_t)\mathbf{S}_k(\mathbf{x}_t)\\ &= \mathbf{S}_k(\mathbf{x}_t)\mathbf{A}^T\mathbf{A}\mathbf{S}_k(\mathbf{x}_t) + \mathbf{W}_k(\mathbf{x}_t)\mathbf{E}_k(\mathbf{x}_t),\end{aligned} \tag{34}$$

with $\widetilde{\mathbf{p}}_k = \mathbf{S}_k^{-1}(\mathbf{x}_t)\mathbf{p}_k$ and $\widetilde{\mathbf{r}}_k = \mathbf{S}_k^{-1}(\mathbf{x}_t)\mathbf{r}_k(\mathbf{x}_t)$.

Since our cost function is quadratic, its minimization in a single step is performed with combining the projected Newton step with the constrained scaled Cauchy step that is given in the form

$$\mathbf{p}_k^{(C)} = -\tau_k\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t), \quad \tau_k > 0. \tag{35}$$

Assuming $\mathbf{x}_t^{(k)} + \mathbf{p}_k^{(C)} > 0$, τ_k is chosen as being either the unconstrained minimizer of the quadratic function $\psi_k(-\tau_k\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t))$ or a scalar proportional to the distance to the boundary along $-\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t)$, where

$$\begin{aligned}\psi_k(\mathbf{p}) &= \frac{1}{2}\mathbf{p}^T\mathbf{M}_k(\mathbf{x}_t)\mathbf{p} + \mathbf{p}^T\mathbf{g}_k(\mathbf{x}_t)\\ &= \frac{1}{2}\mathbf{p}^T(\mathbf{A}^T\mathbf{A}+\mathbf{D}_k^{-1}(\mathbf{x}_t)\mathbf{E}_k(\mathbf{x}_t))\mathbf{p}+\mathbf{p}^T\mathbf{A}^T(\mathbf{A}\mathbf{x}_t^{(k)}-\mathbf{y}_t).\end{aligned} \tag{36}$$

Thus

$$\tau_k = \begin{cases}\tau_1 = \arg\min_{\tau}\psi_k(-\tau_k\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t))\\ \qquad\qquad \text{if } \mathbf{x}_t^{(k)} - \tau_1\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t) > 0,\\ \tau_2 = \theta\min_j\left\{\dfrac{x_{jt}^{(k)}}{(\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t))_j} : (\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t))_j > 0\right\}\\ \qquad\qquad \text{otherwise},\end{cases} \tag{37}$$

where $\psi_k(-\tau_k\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t)) = (\mathbf{g}_k(\mathbf{x}_t))^T\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t)/(\mathbf{D}_k \times (\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t))^T\mathbf{M}_k(\mathbf{x}_t)\mathbf{D}_k(\mathbf{x}_t)\mathbf{g}_k(\mathbf{x}_t)$ with $\theta \in (0,1)$. For $\psi_k(\mathbf{p}_k^{(C)}) < 0$, the global convergence is achieved if $\text{red}(\mathbf{x}_t^{(k+1)} - \mathbf{x}_t^{(k)}) \geq \beta$, $\beta \in (0,1)$ with

$$\text{red}(\mathbf{p}) = \frac{\psi_k(\mathbf{p})}{\psi_k(\mathbf{p}_k^{(C)})}. \tag{38}$$

The usage of the constrained scaled Cauchy step leads to the following updates:

$$\begin{aligned}\mathbf{s}_t^{(k)} &= t(\mathbf{p}_k^{(C)} - \widehat{\mathbf{p}}_k) + \widehat{\mathbf{p}}_k,\\ \mathbf{x}_t^{(k+1)} &= \mathbf{x}_t^{(k)} + \mathbf{s}_t^{(k)},\end{aligned} \tag{39}$$

with $t \in [0,1)$, $\widehat{\mathbf{p}}_k$ and $\mathbf{p}_k^{(C)}$ are given by (30) and (35), respectively, and t is the smaller square root (laying in $(0,1)$) of the quadratic equation:

$$\pi(t) = \psi_k(t(\mathbf{p}_k^{(C)} - \widehat{\mathbf{p}}_k) + \widehat{\mathbf{p}}_k) - \beta\psi_k(\mathbf{p}_k^{(C)}) = 0. \tag{40}$$

The Matlab code of the IPN algorithm, which solves the system $\mathbf{A}\mathbf{x}_t = \mathbf{y}_t$ subject to nonnegativity constraints, is given in Algorithm 5. To solve the transformed system (32), we use the LSQR method implemented in Matlab 7.0.

3.6. Sequential Coordinate-Wise Algorithm

The NNLS problem (1) can be expressed in terms of the following quadratic problem (QP) [62]:

$$\min_{\mathbf{x}_t \geq 0}\Psi(\mathbf{x}_t), \quad (t = 1,\ldots,T), \tag{41}$$

where

$$\Psi(\mathbf{x}_t) = \frac{1}{2}\mathbf{x}_t^T\mathbf{H}\mathbf{x}_t + \mathbf{c}_t^T\mathbf{x}_t, \tag{42}$$

with $\mathbf{H} = \mathbf{A}^T\mathbf{A}$ and $\mathbf{c}_t = -\mathbf{A}^T\mathbf{y}_t$.

The sequential coordinate-wise algorithm (SCWA) proposed first by Franc et al. [62] solves the QP problem given by (41) updating only single variable x_{jt} in one iterative step. It should be noted that the sequential updates can be easily done, if the function $\Psi(\mathbf{x}_t)$ is equivalently rewritten as

$$\begin{aligned}\Psi(\mathbf{x}_t) &= \frac{1}{2}\sum_{p\in\mathcal{I}}\sum_{r\in\mathcal{I}}x_{pt}x_{rt}(\mathbf{A}^T\mathbf{A})_{pr} + \sum_{p\in\mathcal{I}}x_{pt}(\mathbf{A}^T\mathbf{y}_t)_{pt}\\ &= \frac{1}{2}x_{jt}^2(\mathbf{A}^T\mathbf{A})_{jj} + x_{jt}(\mathbf{A}^T\mathbf{y}_t)_{jt}\\ &\quad + x_{jt}\sum_{p\in\mathcal{I}\setminus\{j\}}x_{pt}(\mathbf{A}^T\mathbf{A})_{pj} + \sum_{p\in\mathcal{I}\setminus\{j\}}x_{pt}(\mathbf{A}^T\mathbf{y}_t)_{pt}\\ &\quad + \frac{1}{2}\sum_{p\in\mathcal{I}\setminus\{j\}}\sum_{r\in\mathcal{I}\setminus\{j\}}x_{pt}x_{rt}(\mathbf{A}^T\mathbf{A})_{pr}\\ &= \frac{1}{2}x_{jt}^2h_{jj} + x_{jt}\beta_{jt} + \gamma_{jt},\end{aligned} \tag{43}$$

where $\mathcal{I} = \{1,\ldots,J\}$, and

$$\begin{aligned}h_{jj} &= (\mathbf{A}^T\mathbf{A})_{jj},\\ \beta_{jt} &= (\mathbf{A}^T\mathbf{y}_t)_{jt} + \sum_{p\in\mathcal{I}\setminus\{j\}}x_{pt}(\mathbf{A}^T\mathbf{A})_{pj}\\ &= [\mathbf{A}^T\mathbf{A}\mathbf{x}_t + \mathbf{A}^T\mathbf{y}_t]_{jt} - (\mathbf{A}^T\mathbf{A})_{jj}x_{jt},\\ \gamma_{jt} &= \sum_{p\in\mathcal{I}\setminus\{j\}}x_{pt}(\mathbf{A}^T\mathbf{y}_t)_{pt} + \frac{1}{2}\sum_{p\in\mathcal{I}\setminus\{j\}}\sum_{r\in\mathcal{I}\setminus\{j\}}x_{pt}x_{rt}(\mathbf{A}^T\mathbf{A})_{pr}.\end{aligned} \tag{44}$$

```
% Interior Point Newton (IPN) algorithm function
%
function [x] = nmf_ipn(A,y,x,no_iter)
%
% [x]=nmf_ipn(A,y,x,no_iter) finds such x that solves the equation Ax = y
% subject to nonnegativity constraints.
%
% INPUTS:
% A - system matrix of dimension [I by J]
% y - vector of observations [I by 1]
% x - vector of initial guess [J by 1]
% no_iter - maximum number of iterations
%
% OUTPUTS:
% x - vector of estimated sources [J by 1]
%
% ###########################################################################
% Parameters
s = 1.8; theta = 0.5; rho = .1; beta = 1;
H = A'*A; yt = A'*y; J = size(x,1);

% Main loop
for k=1:no_iter

g = H*x - yt; d = ones(J,1); d(g >= 0) = x(g >= 0);
ek = zeros(J,1); ek(g >=0 & g < x.^s) = g(g >=0 & g < x.^s);
M = H + diag(ek./d);
dg = d.*g;
tau1 = (g'*dg)/(dg'*M*dg); tau_2vec = x./dg;
tau2 = theta*min(tau_2vec(dg > 0));
tau = tau1*ones(J,1); tau(x - tau1*dg <= 0) = tau2;
w = 1./(d + ek); sk = sqrt(w.*d); pc = - tau.*dg;
Z = repmat(sk,1,J).*M.*repmat(sk',J,1);
rt = -g./sk;
[pt,flag,relres,iter,resvec] = lsqr(Z,rt - g.*sk,1E-8);
p = pt.*sk;
phx = max(0, x + p) - x;
ph = max(rho, 1 - norm(phx))*phx;
Phi_pc = .5*pc'*M*pc + pc'*g; Phi_ph = .5*ph'*M*ph + ph'*g;
red_p = Phi_ph/Phi_pc; dp = pc - ph;

if red_p >= beta
    t = 0;
else
    ax = .5*dp'*M*dp; bx = dp'*(M*ph + g);
    cx = Phi_ph - beta*Phi_pc;
    Deltas = sqrt(bx^2 - 4*ax*cx);
    t1 = .5*(-bx + Deltas)/ax; t2 = .5*(-bx - Deltas)/ax;
    t1s = t1 > 0 & t1 < 1; t2s = t2 > 0 & t2 < 1; t = min(t1, t2);
    if (t <= 0)
        if t1s
            t = t1s;
        else
            t = t2s;
        end
    end
end

sk = t*dp + ph;
x = x + sk;

end % for k
```

Algorithm 5

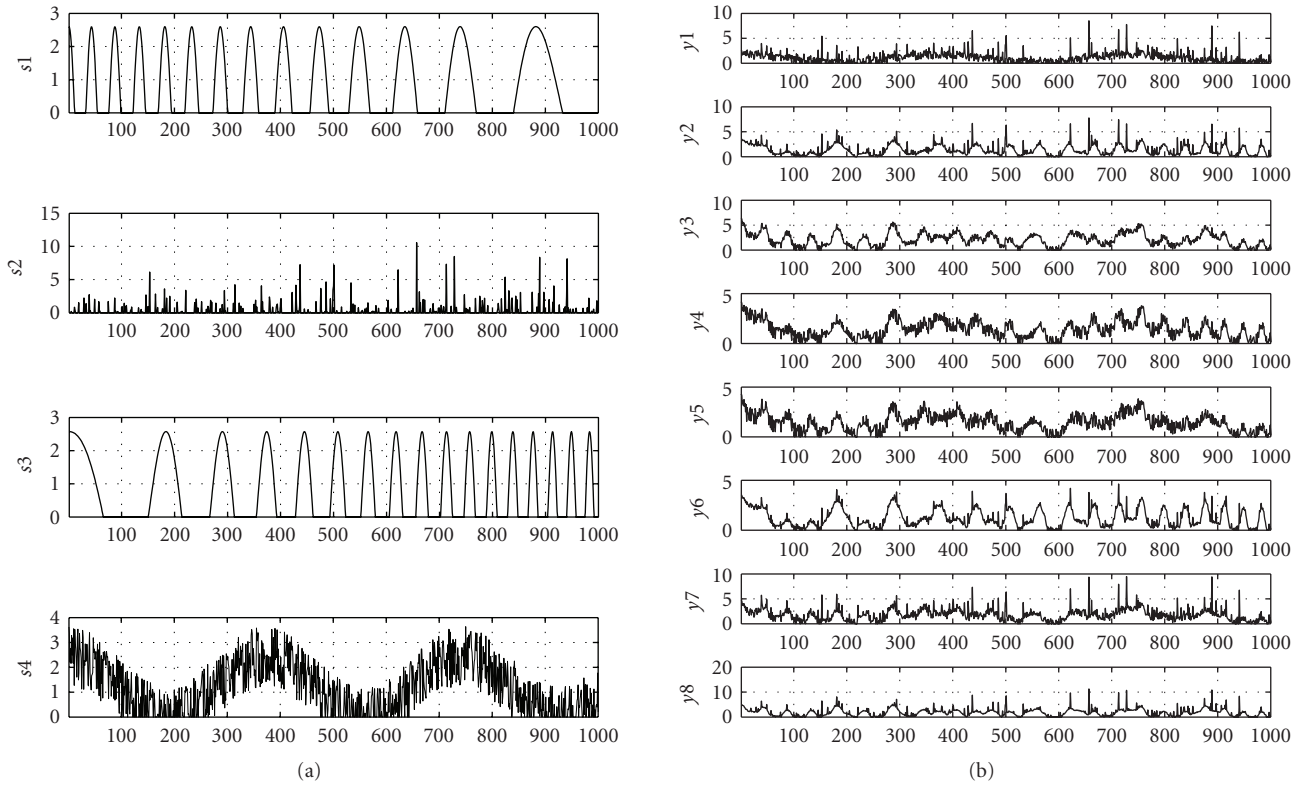

Figure 1: Dataset: (a) original 4 source signals, (b) observed 8 mixed signals.

Considering the optimization of $\Psi(\mathbf{x}_t)$ with respect to the selected variable x_{jt}, the following analytical solution can be derived:

$$\begin{aligned} x_{jt}^{*} &= \arg\min \Psi\left(\left[x_{1t},\ldots,x_{jt},\ldots,x_{Jt}\right]^{T}\right) \\ &= \arg\min \frac{1}{2}x_{jt}^{2}h_{jj} + x_{jt}\beta_{jt} + \gamma_{jt} \\ &= \max\left(0, -\frac{\beta_{jt}}{h_{jj}}\right) \\ &= \max\left(0, x_{jt} - \frac{\left[\mathbf{A}^{T}\mathbf{A}\mathbf{x}_t\right]_{jt} + \left[\mathbf{A}^{T}\mathbf{y}_t\right]_{jt}}{\left(\mathbf{A}^{T}\mathbf{A}\right)_{jj}}\right). \end{aligned} \tag{45}$$

Updating only single variable x_{jt} in one iterative step, we have

$$x_{pt}^{(k+1)} = x_{pt}^{(k)}, \quad \forall p \in \mathcal{J} \setminus \{j\}, \qquad x_{jt}^{(k+1)} \neq x_{jt}^{(k)}. \tag{46}$$

Any optimal solution to the QP (41) satisfies the KKT conditions given by (5) and the stationarity condition of the following Lagrange function:

$$\mathcal{L}(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \frac{1}{2}\mathbf{x}_t^{T}\mathbf{H}\mathbf{x}_t + \mathbf{c}_t^{T}\mathbf{x}_t - \boldsymbol{\lambda}_t^{T}\mathbf{x}_t, \tag{47}$$

where $\boldsymbol{\lambda}_t \in \mathbb{R}^{J}$ is a vector of Lagrange multipliers (or dual variables) corresponding to the vector $\mathbf{x}_t$. Thus, $\nabla_{\mathbf{x}_t}\mathcal{L}(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \mathbf{H}\mathbf{x}_t + \mathbf{c}_t - \boldsymbol{\lambda}_t = \mathbf{0}$. In the SCWA, the Lagrange multipliers are updated in each iteration according to the formula

$$\boldsymbol{\lambda}_t^{(k+1)} = \boldsymbol{\lambda}_t^{(k)} + (x_{jt}^{(k+1)} - x_{jt}^{(k)})\mathbf{h}_j, \tag{48}$$

where $\mathbf{h}_j$ is the jth column of $\mathbf{H}$, and $\boldsymbol{\lambda}_t^{(0)} = \mathbf{c}_t$.

Finally, the SCWA can take the following updates:

$$\begin{aligned} x_{jt}^{(k+1)} &= \max\left(0, x_{jt}^{(k)} - \frac{\lambda_j^{(k)}}{\left(\mathbf{A}^{T}\mathbf{A}\right)_{jj}}\right), \\ x_{pt}^{(k+1)} &= x_{pt}^{(k)}, \quad \forall p \in \mathcal{J} \setminus \{j\} \\ \boldsymbol{\lambda}_t^{(k+1)} &= \boldsymbol{\lambda}_t^{(k)} + (x_{jt}^{(k+1)} - x_{jt}^{(k)})\mathbf{h}_j. \end{aligned} \tag{49}$$

4. Simulations

All the proposed algorithms were implemented in our NMFLAB, and evaluated with the numerical tests related to typical BSS problems. We used the synthetic benchmark of 4 partially dependent nonnegative signals (with only $T = 1000$ samples) which are illustrated in Figure 1(a). The signals are mixed by random, uniformly distributed nonnegative matrix

$\mathbf{A} \in \mathbb{R}^{8\times 4}$ with the condition number $\text{cond}\{\mathbf{A}\} = 4.11$. The matrix $\mathbf{A}$ is displayed in

$$\mathbf{A} = \begin{bmatrix} 0.0631 & 0.7666 & 0.0174 & 0.6596 \\ 0.2642 & 0.6661 & 0.8194 & 0.2141 \\ 0.9995 & 0.1309 & 0.6211 & 0.6021 \\ 0.2120 & 0.0954 & 0.5602 & 0.6049 \\ 0.4984 & 0.0149 & 0.2440 & 0.6595 \\ 0.2905 & 0.2882 & 0.8220 & 0.1834 \\ 0.6728 & 0.8167 & 0.2632 & 0.6365 \\ 0.9580 & 0.9855 & 0.7536 & 0.1703 \end{bmatrix}. \tag{50}$$

The mixing signals are shown in Figure 1(b).

Because the number of variables in $\mathbf{X}$ is much greater than in $\mathbf{A}$, that is, $I \times J = 32$ and $J \times T = 4000$, we test the projected gradient algorithms only for updating $\mathbf{A}$. The variables in $\mathbf{X}$ are updated with the standard projected fixed point alternating least squares (FP-ALS) algorithm that is extensively analyzed in [55].

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

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \right\} \tag{51}$$

with the Moore-Penrose pseudoinverse of a system matrix, that is, in our case, the matrix $\mathbf{A}$. Since in NMF usually $I \geq J$, 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 $\mathbf{X}_{\text{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y} = \mathbf{A}^+\mathbf{Y}$, where $\mathbf{A}^+$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$. The projected FP-ALS algorithm is obtained with a simple "half-rectified" projection, that is,

$$\mathbf{X} = P_\Omega[\mathbf{A}^+\mathbf{Y}]. \tag{52}$$

The alternating minimization is nonconvex in spite of the cost function being convex with respect to one set of variables. Thus, most NMF algorithms may get stuck in local minima, and hence, the initialization plays a key role. In the performed tests, we applied the multistart initialization described in [53] with the following parameters: $N = 10$ (number of restarts), $K_i = 30$ (number of initial alternating steps), and $K_f = 1000$ (number of final alternating steps). Each initial sample of $\mathbf{A}$ and $\mathbf{X}$ has been randomly generated from a uniform distribution. Each algorithm has been tested for two cases of inner iterations, that is, with $k = 1$ and $k = 5$. The inner iterations mean a number of iterative steps that are performed to update only $\mathbf{A}$ (with fixed $\mathbf{X}$, i.e., before going to the update of $\mathbf{X}$ and vice versa). Additionally, the multilayer technique [53, 54] with 3 layers ($L = 3$) is applied.

The multilayer technique can be regarded as multistep decomposition. In the first step, we perform the basic decomposition $\mathbf{Y} = \mathbf{A}_1\mathbf{X}_1$ using any available NMF algorithm, where $\mathbf{A}_1 \in \mathbb{R}^{I\times J}$ and $\mathbf{X}_1 \in \mathbb{R}^{J\times T}$ with $I \geq J$. In the second stage, the results obtained from the first stage are used to perform the similar decomposition: $\mathbf{X}_1 = \mathbf{A}_2\mathbf{X}_2$, where $\mathbf{A}_2 \in \mathbb{R}^{J\times J}$ and $\mathbf{X}_2 \in \mathbb{R}^{J\times T}$, using the same or different update rules, and so on. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrary 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 $\mathbf{Y} = \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_L\mathbf{X}_L$ with the basis matrix defined as $\mathbf{A} = \mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_L \in \mathbb{R}^{I\times J}$. Physically, this means that we build up a system that has many layers or cascade connection of L mixing subsystems.

There are many stopping criteria for terminating the alternating steps. We stop the iterations if $s \geq K_f = 1000$ or the following condition $\|\mathbf{A}^{(s)} - \mathbf{A}^{(s-1)}\|_F < \epsilon$ is held, where s stands for the number of alternating step, and $\epsilon = 10^{-5}$. Note that the condition (20) can be also used as a stopping criterion, especially as the gradient is computed in each iteration of the PG algorithms.

The algorithms have been evaluated with the signal-to-interference ratio (SIR) measures, calculated separately for each source signal and each column in the mixing matrix. Since NMF suffers from scale and permutation indeterminacies, the estimated components are adequately permuted and rescaled. First, the source and estimated signals are normalized to a uniform variance, and then the estimated signals are permuted to keep the same order as the source signals. In NMFLAB [43], each estimated signal is compared to each source signal, which results in the performance (SIR) matrix that is involved to make the permutation matrix. Let $\mathbf{x}_j$ and $\hat{\mathbf{x}}_j$ be the jth source and its corresponding (reordered) estimated signal, respectively. Analogically, let $\mathbf{a}_j$ and $\hat{\mathbf{a}}_j$ be the jth column of the true and its corresponding estimated mixing matrix, respectively. Thus, the SIRs for the sources are given by

$$\text{SIR}_j^{(X)} = -20 \log \left\{ \frac{\|\hat{\mathbf{x}}_j - \mathbf{x}_j\|_2}{\|\mathbf{x}_j\|_2} \right\}, \quad j = 1, \ldots, J, \ [\text{dB}] \tag{53}$$

and similarly for each column in $\mathbf{A}$ we have

$$\text{SIR}_j^{(A)} = -20 \log \left\{ \frac{\|\hat{\mathbf{a}}_j - \mathbf{a}_j\|_2}{\|\mathbf{a}_j\|_2} \right\}, \quad j = 1, \ldots, J, \ [\text{dB}]. \tag{54}$$

We test the algorithms with the Monte Carlo (MC) analysis, running each algorithm 100 times. Each run has been initialized with the multistart procedure. The algorithms have been evaluated with the mean-SIR values that are calculated as follows:

$$\begin{aligned} \overline{\text{SIR}}_X &= \frac{1}{J}\sum_{j=1}^{J} \text{SIR}_j^{(X)}, \\ \overline{\text{SIR}}_A &= \frac{1}{J}\sum_{j=1}^{J} \text{SIR}_j^{(A)}, \end{aligned} \tag{55}$$

for each MC sample. The mean-SIRs for the worst (with the lowest mean-SIR values) and best (with the highest mean-SIR values) samples are given in Table 1. The number k means the number of inner iterations for updating $\mathbf{A}$, and

Table 1: Mean-SIRs [dB] obtained with 100 samples of Monte Carlo analysis for the estimation of sources and columns of mixing matrix from noise-free mixtures of signals in Figure 1. Sources $\mathbf{X}$ are estimated with the projected pseudoinverse. The number of inner iterations for updating $\mathbf{A}$ is denoted by k, and the number of layers (in the multilayer technique) by L. The notation *best* or *worst* in parenthesis that follows the algorithm name means that the mean-SIR value is calculated for the best or worst sample from Monte Carlo analysis, respectively. In the last column, the elapsed time [in seconds] is given for each algorithm with $k = 1$ and $L = 1$.

Algorithm	Mean-SIR$_A$ [dB]				Mean-SIR$_X$ [dB]				Time
	$L = 1$		$L = 3$		$L = 1$		$L = 3$		
	$k = 1$	$k = 5$	$k = 1$	$k = 5$	$k = 1$	$k = 5$	$k = 1$	$k = 5$	
M-NMF (best)	21	22.1	42.6	37.3	26.6	27.3	44.7	40.7	
M-NMF (mean)	13.1	13.8	26.7	23.1	14.7	15.2	28.9	27.6	1.9
M-NMF (worst)	5.5	5.7	5.3	6.3	5.8	6.5	5	5.5	
OPL(best)	22.9	25.3	46.5	42	23.9	23.5	55.8	51	
OPL(mean)	14.7	14	25.5	27.2	15.3	14.8	23.9	25.4	1.9
OPL(worst)	4.8	4.8	4.8	5.0	4.6	4.6	4.6	4.8	
Lin-PG(best)	36.3	23.6	78.6	103.7	34.2	33.3	78.5	92.8	
Lin-PG(mean)	19.7	18.3	40.9	61.2	18.5	18.2	38.4	55.4	8.8
Lin-PG(worst)	14.4	13.1	17.5	40.1	13.9	13.8	18.1	34.4	
GPSR-BB(best)	18.2	22.7	7.3	113.8	22.8	54.3	9.4	108.1	
GPSR-BB(mean)	11.2	20.2	7	53.1	11	20.5	5.1	53.1	2.4
GPSR-BB(worst)	7.4	17.3	6.8	24.9	4.6	14.7	2	23	
PSESOP(best)	21.2	22.6	71.1	132.2	23.4	55.5	56.5	137.2	
PSESOP(mean)	15.2	20	29.4	57.3	15.9	34.5	27.4	65.3	5.4
PSESOP(worst)	8.3	15.8	6.9	28.7	8.2	16.6	7.2	30.9	
IPG(best)	20.6	22.2	52.1	84.3	35.7	28.6	54.2	81.4	
IPG(mean)	20.1	18.2	35.3	44.1	19.7	19.1	33.8	36.7	2.7
IPG(worst)	10.5	13.4	9.4	21.2	10.2	13.5	8.9	15.5	
IPN(best)	20.8	22.6	59.9	65.8	53.5	52.4	68.6	67.2	
IPN(mean)	19.4	17.3	38.2	22.5	22.8	19.1	36.6	21	14.2
IPN(worst)	11.7	15.2	7.5	7.1	5.7	2	1.5	2	
RMRNSD(best)	24.7	21.6	22.2	57.9	30.2	43.5	25.5	62.4	
RMRNSD(mean)	14.3	19.2	8.3	33.8	17	21.5	8.4	33.4	3.8
RMRNSD(worst)	5.5	15.9	3.6	8.4	4.7	13.8	1	3.9	
SCWA(best)	12.1	20.4	10.6	24.5	6.3	25.6	11.9	34.4	
SCWA(mean)	11.2	16.3	9.3	20.9	5.3	18.6	9.4	21.7	2.5
SCWA(worst)	7.3	11.4	6.9	12.8	3.8	10	3.3	10.8	

L denotes the number of layers in the multilayer technique [53, 54]. The notation $L = 1$ means that the multilayer technique was not used. The elapsed time [in seconds] is measured in Matlab, and it informs us in some sense about a degree of complexity of the algorithm.

For comparison, Table 1 contains also the results obtained for the standard multiplicative NMF algorithm (denoted as M-NMF) that minimizes the squared Euclidean distance. Additionally, the results of testing the PG algorithms which were proposed in [53] have been also included. The acronyms Lin-PG, IPG, RMRNSD refer to the following algorithms: projected gradient proposed by Lin [52], interior-point gradient, and regularized minimal residual norm steepest descent (the regularized version of the MRNSD algorithm that was proposed by Nagy and Strakos [74]). These NMF algorithms have been implemented and investigated in [53] in the context of their usefulness to BSS problems.

5. Conclusions

The performance of the proposed NMF algorithms can be inferred from the results given in Table 1. In particular, the results show how the algorithms are sensitive to initialization, or in other words, how easily they fall in local minima. Also the complexity of the algorithms can be estimated from the information on the elapsed time that is measured in Matlab.

It is easy to notice that our NMF-PSESOP algorithm gives the best estimation (the sample which has the highest best-SIR value), and it gives only slightly lower mean-SIR values than the Lin-PG algorithm. Considering the elapsed time, the PL, GPSR-BB, SCWA, and IPG belong to the fastest

algorithms, while the Lin-PG and IPN algorithms are the slowest.

The multilayer technique generally improves the performance and consistency of all the tested algorithms if the number of observation is close to the number of nonnegative components. The highest improvement can be observed for the NMF-PSESOP algorithm, especially when the number of inner iterations is greater than one (typically, $k = 5$).

In summary, the best and the most promising NMG-PG algorithms are NMF-PSESOP, GPSR-BB, and IPG algorithms. However, the final selection of the algorithm depends on a size of the problem to be solved. Nevertheless, the projected gradient NMF algorithms seem to be much better (in the sense of speed and performance) in our tests than the multiplicative algorithms, provided that we can use the squared Euclidean cost function which is optimal for data with a Gaussian noise.

References

[1] A. Cichocki, R. Zdunek, and S. Amari, "New algorithms for nonnegative matrix factorization in applications to blind source separation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 621–624, Toulouse, France, May 2006.

[2] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin, "Object characterization from spectral data using nonnegative matrix factorization and information theory," in *Proceedings of the AMOS Technical Conference*, pp. 1–12, Maui, Hawaii, USA, September 2004.

[3] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '06)*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 700–707, Charleston, SC, USA, March 2006.

[4] A. Ziehe, P. Laskov, K. Pawelzik, and K.-R. Mueller, "Nonnegative sparse coding for general blind source separation," in *Advances in Neural Information Processing Systems 16*, Vancouver, Canada, 2003.

[5] W. Wang, Y. Luo, J. A. Chambers, and S. Sanei, "Nonnegative matrix factorization for note onset detection of audio signals," in *Proceedings of the 16th IEEE International Workshop on Machine Learning for Signal Processing (MLSP '06)*, pp. 447–452, Maynooth, Ireland, September 2006.

[6] W. Wang, "Squared euclidean distance based convolutive nonnegative matrix factorization with multiplicative learning rules for audio pattern separation," in *Proceedings of the 7th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '07)*, pp. 347–352, Cairo, Egypt, December 2007.

[7] H. Li, T. Adali, W. Wang, and D. Emge, "Nonnegative matrix factorization with orthogonality constraints for chemical agent detection in Raman spectra," in *Proceedings of the 15th IEEE International Workshop on Machine Learning for Signal Processing (MLSP '05)*, pp. 253–258, Mystic, Conn, USA, September 2005.

[8] P. Sajda, S. Du, T. Brown, L. Parra, and R. Stoyanova, "Recovery of constituent spectra in 3D chemical shift imaging using nonnegative matrix factorization," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind (ICA '03)*, pp. 71–76, Nara, Japan, April 2003.

[9] P. Sajda, S. Du, T. R. Brown, et al., "Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *IEEE Transactions on Medical Imaging*, vol. 23, no. 12, pp. 1453–1465, 2004.

[10] P. Sajda, S. Du, and L. C. Parra, "Recovery of constituent spectra using nonnegative matrix factorization," in *Wavelets: Applications in Signal and Image Processing X*, vol. 5207 of *Proceedings of SPIE*, pp. 321–331, San Diego, Calif, USA, August 2003.

[11] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[12] W. Liu and N. Zheng, "Nonnegative matrix factorization based methods for object recognition," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 893–897, 2004.

[13] M. W. Spratling, "Learning image components for object recognition," *Journal of Machine Learning Research*, vol. 7, pp. 793–815, 2006.

[14] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Nonnegative matrix factorization framework for face recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 4, pp. 495–511, 2005.

[15] P. Smaragdis, "Nonnegative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '04)*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 494–499, Granada, Spain, September 2004.

[16] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[17] J.-H. Ahn, S. Kim, J.-H. Oh, and S. Choi, "Multiple nonnegative-matrix factorization of dynamic PET images," in *Proceedings of the 6th Asian Conference on Computer Vision (ACCV '04)*, pp. 1009–1013, Jeju Island, Korea, January 2004.

[18] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. Pascual-Montano, "Biclustering of gene expression data by non-smooth nonnegative matrix factorization," *BMC Bioinformatics*, vol. 7, article 78, pp. 1–18, 2006.

[19] D. Guillamet, B. Schiele, and J. Vitrià, "Analyzing nonnegative matrix factorization for image classification," in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR '02)*, vol. 2, pp. 116–119, Quebec City, Canada, August 2002.

[20] D. Guillamet and J. Vitrià, "Nonnegative matrix factorization for face recognition," in *Proceedings of the 5th Catalan Conference on Artificial Intelligence (CCIA '02)*, pp. 336–344, Castello de la Plana, Spain, October 2002.

[21] D. Guillamet, J. Vitrià, and B. Schiele, "Introducing a weighted nonnegative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, 2003.

[22] O. G. Okun, "Nonnegative matrix factorization and classifiers: experimental study," in *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP '04)*, pp. 550–555, Marbella, Spain, September 2004.

[23] O. G. Okun and H. Priisalu, "Fast nonnegative matrix factorization and its application for protein fold recognition," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 71817, 8 pages, 2006.

[24] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Non-smooth nonnegative matrix factorization (nsNMF)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403–415, 2006.

[25] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using nonnegative matrix factorizations," in *Proceedings of the 4th SIAM International Conference on Data Mining (SDM '04)*, pp. 452–456, Lake Buena Vista, Fla, USA, April 2004.

[26] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Journal on Information Processing & Management*, vol. 42, no. 2, pp. 373–386, 2006.

[27] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM '06)*, pp. 362–371, Hong Kong, December 2006.

[28] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 126–135, Philadelphia, Pa, USA, August 2006.

[29] R. Zass and A. Shashua, "A unifying approach to hard and probabilistic clustering," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 294–301, Beijing, China, October 2005.

[30] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," in *Proceedings of the 4th SIAM International Conference on Data Mining (SDM '04)*, pp. 234–245, Lake Buena Vista, Fla, USA, April 2004.

[31] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, "Minimum sum-squared residue co-clustering of gene expression data," in *Proceedings of the 4th SIAM International Conference on Data Mining (SDM '04)*, pp. 114–125, Lake Buena Vista, Fla, USA, April 2004.

[32] S. Wild, *Seeding nonnegative matrix factorization with the spherical k-means clustering*, M.S. thesis, University of Colorado, Boulder, Colo, USA, 2000.

[33] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.

[34] Y.-C. Cho and S. Choi, "Nonnegative features of spectro-temporal sounds for classification," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1327–1336, 2005.

[35] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 12, pp. 4164–4169, 2004.

[36] N. Rao and S. J. Shepherd, "Extracting characteristic patterns from genome-wide expression data by nonnegative matrix factorization," in *Proceedings of the IEEE Computational Systems Bioinformatics Conference (CSB '04)*, pp. 570–571, Stanford, Calif, USA, August 2004.

[37] A. Cichocki, R. Zdunek, and S. Amari, "Csiszár's divergences for nonnegative matrix factorization: family of new algorithms," in *Independent Component Analysis and Blind Signal Separation*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 32–39, Springer, New York, NY, USA, 2006.

[38] A. Cichocki, R. Zdunek, and S. Amari, "Nonnegative matrix and tensor factorization," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 142–145, 2008.

[39] D. Donoho and V. Stodden, "When does nonnegative matrix factorization give a correct decomposition into parts?" in *Advances in Neural Information Processing Systems 16*, Vancouver, Canada, 2003.

[40] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "Sparse nonnegative solution of a linear system of equations is unique," in *Proceedings of the 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP '08)*, St. Julians, Malta, March 2008.

[41] F. J. Theis, K. Stadlthanner, and T. Tanaka, "First results on uniqueness of sparse nonnegative matrix factorization," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO '05)*, Antalya, Turkey, September 2005.

[42] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen, "Theorems on positive data: on the uniqueness of NMF," *Computational Intelligence and Neuroscience*, vol. 2008, Article ID 764206, 9 pages, 2008.

[43] A. Cichocki and R. Zdunek, "NMFLAB for signal and image processing," Tech. Rep., Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan, 2006.

[44] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, "Extended SMART algorithms for nonnegative matrix factorization," in *Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC '06)*, vol. 4029 of *Lecture Notes in Computer Science*, pp. 548–562, Springer, Zakopane, Poland, June 2006.

[45] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with quasi-Newton optimization," in *Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC '06)*, vol. 4029 of *Lecture Notes in Computer Science*, pp. 870–879, Zakopane, Poland, June 2006.

[46] P. Paatero, "Least-squares formulation of robust nonnegative factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, no. 1, pp. 23–35, 1997.

[47] P. Paatero and U. Tapper, "Positive matrix factorization: a nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[48] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Advances in Neural Information Processing Systems 13*, pp. 556–562, Denver, Colo, USA, 2000.

[49] Ch.-J. Lin., "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.

[50] M. T. Chu, F. Diele, R. Plemmons, and S. Ragni, "Optimality, computation, and interpretation of nonnegative matrix factorizations," Tech. Rep., Departments of Mathematics and Computer Science, Wake Forest University, Winston-Salem, NC, USA, 2004.

[51] P. O. Hoyer, "Nonnegative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[52] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[53] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization using projected gradient approaches," *International Journal of Neural Systems*, vol. 17, no. 6, pp. 431–446, 2007.

[54] A Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization," *Electronics Letters*, vol. 42, no. 16, pp. 947–948, 2006.

[55] A. Cichocki and R. Zdunek, "Regularized alternating least squares algorithms for nonnegative matrix/tensor factorizations," in *Proceedings of the 4th International Symposium on Neural Networks on Advances in Neural Networks (ISNN '07)*, vol. 4493 of *Lecture Notes in Computer Science*, pp. 793–802, Springer, Nanjing, China, June 2007.

[56] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, pp. 1904–1916, 2007.

[57] B. Johansson, T. Elfving, V. Kozlov, Y. Censor, P.-E. Forssén, and G. Granlund, "The application of an oblique-projected Landweber method to a model of supervised learning," *Mathematical and Computer Modelling*, vol. 43, no. 7-8, pp. 892–909, 2006.

[58] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[59] G. Narkiss and M. Zibulevsky, "Sequential subspace optimization method for large-scale unconstrained problems," Tech. Rep. 559, Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel, October 2005.

[60] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 346–367, 2007.

[61] S. Bellavia, M. Macconi, and B. Morini, "An interior point Newton-like method for nonnegative least-squares problems with degenerate solution," *Numerical Linear Algebra with Applications*, vol. 13, no. 10, pp. 825–846, 2006.

[62] V. Franc, V. Hlaváč, and M. Navara, "Sequential coordinate-wise algorithm for the nonnegative least squares problem," in *Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns (CAIP '05)*, A. Gagalowicz and W. Philips, Eds., vol. 3691 of *Lecture Notes in Computer Science*, pp. 407–414, Springer, Versailles, France, September 2005.

[63] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*, Institute of Physics, Bristol, UK, 1998.

[64] Y.-H. Dai and R. Fletcher, "Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming," *Numerische Mathematik*, vol. 100, no. 1, pp. 21–47, 2005.

[65] A. Nemirovski, "Orth-method for smooth convex optimization," *Izvestiia Akademii Nauk SSSR. Tekhnicheskaia Kibernetika*, vol. 2, 1982 (Russian).

[66] R. W. Freund and N. M. Nachtigal, "QMR: a quasi-minimal residual method for non-Hermitian linear systems," *Numerische Mathematik*, vol. 60, no. 1, pp. 315–339, 1991.

[67] R. Fletcher, "Conjugate gradient methods for indefinite systems," in *Proceedings of the Dundee Biennial Conference on Numerical Analysis*, pp. 73–89, Springer, Dundee, Scotland, July 1975.

[68] C. Lanczos, "Solution of systems of linear equations by minimized iterations," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 1, pp. 33–53, 1952.

[69] H. A. van der Vorst, "Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992.

[70] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[71] M. R. Hestenes and E. Stiefel, "Method of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[72] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, Pa, USA, 1998.

[73] C. C. Paige and M. A. Saunders, "LSQR: an algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, 1982.

[74] J. G. Nagy and Z. Strakos, "Enforcing nonnegativity in image reconstruction algorithms," in *Mathematical Modeling, Estimation, and Imaging*, vol. 4121 of *Proceedings of SPIE*, pp. 182–190, San Diego, Calif, USA, July 2000.