

Nonnegative Matrix Factorization with Quadratic Programming

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, Wrocław University of Technology, Poland*

^c*Warsaw University of Technology, Poland*

Abstract

Nonnegative Matrix Factorization (NMF) solves the following problem: find such nonnegative matrices $\mathbf{A} \in \mathbb{R}_+^{I \times J}$ and $\mathbf{X} \in \mathbb{R}_+^{J \times K}$ that $\mathbf{Y} \cong \mathbf{AX}$, given only $\mathbf{Y} \in \mathbb{R}^{I \times K}$ and the assigned index J ($K \gg I \geq J$). Basically, the factorization is achieved by alternating minimization of a given cost function subject to nonnegativity constraints. In the paper, we propose to use Quadratic Programming (QP) to solve the minimization problems. The Tikhonov regularized squared Euclidean cost function is extended with a logarithmic barrier function (which satisfies nonnegativity constraints), and then using second-order Taylor expansion, a QP problem is formulated. This problem is solved with some trust-region subproblem algorithm. The numerical tests are performed on the blind source separation problems.

Key words: Nonnegative matrix factorization, Blind source separation, Quadratic programming

1 Introduction

Nonnegative Matrix Factorization (NMF) attempts to recover hidden non-negative structures or patterns from usually redundant data. This method has found a variety of real-world applications in the areas such as blind signal processing [16,35,43,48], spectra recovering [25,40–42], pattern recognition [45,46],

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

dimensionality reduction, segmentation or clustering [1,8,17–19,23,24,27,31–33,44], or neuro-biology (gene separation) [5,36].

NMF decomposes the data matrix $\mathbf{Y} = [y_{ik}] \in \mathbb{R}^{I \times K}$ as a product of two nonnegative matrices $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{I \times J}$ and $\mathbf{X} = [x_{jk}] \in \mathbb{R}^{J \times K}$, where $\forall i, j, k : a_{ij} \geq 0, x_{jk} \geq 0$. Although some matrix factorizations provide exact factors (i.e., $\mathbf{Y} = \mathbf{A}\mathbf{X}$), here we shall consider a factorization that is approximative in nature, i.e.,

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{V}, \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{I \times K}$ represents noise or error matrix.

Depending on an application, the hidden structures may have different interpretation. For example, Lee and Seung in [23] 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) [15], 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 k usually denotes a sample (discrete time instant), where K is a number of available samples. In BSS, we usually have $K \gg I \geq J$, 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.

The noise distribution is strongly application-dependent, however, in many BSS applications, a Gaussian noise is expected. Hence, here our considerations are restricted to this case, however, the alternative NMF algorithms optimized to different distributions of the noise can be found, e.g. in [10,15,47].

The basic approach to NMF is the alternating minimization of the specific function $D(\mathbf{Y}||\mathbf{A}\mathbf{X})$ described by the Algorithm 1.

Lee and Seung [23] were the first to apply the Algorithm 1 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 EMMML or Richardson-Lucy algorithm (RLA) [6,22,28,37] for minimization of the KL divergence, and the ISRA algorithm [34] which minimizes the Euclidean distance.

The multiplicative algorithms are known to have slow convergence and easily

Algorithm 1. NMF

Set Randomly initialize: $\mathbf{A}^{(0)}, \mathbf{X}^{(0)}$,
For $s = 1, 2, \dots$, until convergence **do**
 Step 1:
 $\mathbf{X}^{(s+1)} = \arg \min_{x_{jk} \geq 0} D(\mathbf{Y} || \mathbf{A}^{(s)} \mathbf{X})$
 Step 2:
 $\mathbf{A}^{(s+1)} = \arg \min_{a_{ij} \geq 0} D(\mathbf{Y} || \mathbf{A} \mathbf{X}^{(s+1)})$
End

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 [9,21,26] or projected Alternating Least-Squares (ALS) algorithms [2,12] instead of multiplicative ones. Another improvement concerns modification of a learning rate (relaxation parameter) to speed up the convergence, and better recovering zero-values entries (for sparse solutions). C.-J. Lin [26] suggested applying the Armijo rule to estimate the learning parameters in projected gradient updates for NMF. Also, the gradient algorithms given in [13] address the issue with selecting such a learning parameter that is the steepest decent and also keeps some distance to a boundary of nonnegative orthant. 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 [47], where the mixing matrix \mathbf{A} is updated with the projected Newton method, and the sources in \mathbf{X} are calculated with the projected least-squares method (the fixed point algorithm). The projection therein was performed in a very simple way, i.e. all negative entries are replaced by a small positive constant (eps). It is obvious that such a projection may introduce significant errors, artifacts, or instabilities, especially as the factors to be estimated are sparse. This issue for linear systems with a single right-hand side has been also discussed in [20,29,39].

In this paper, we extend the idea presented in [47] with Quadratic Programming (QP). The motivation for this approach was the Interior-Point Trust-Region (IPTR) based method which was proposed in [39], and latter, modified by Calvetti *et al* [7]. Rojas and Steihaug [39] considered a least-squares problem with nonnegativity and l_2 -norm constraints. They showed that such a problem can be transformed to a QP problem with a logarithmic barrier function. Then, they used the LSTRS method [38] to solve a trust-region subproblem in the QP problem. The LSTRS boils down a QP problem to a parameterized eigenvalue problem that is solved with the implicitly restarted Arnoldi method. Calvetti *et al* [7] solved a QP trust-region subproblem with a hybrid method based on the Lanczos bidiagonalization and QR factorization.

In our approach, we extend the IPTR method to a NMF problem, where

linear systems with multiple right-hand side must be solved. For a cost function expressed by the squared Euclidean distance, the Hessian has a regular small diagonal block structure, and hence, we use a different approach to solve the trust-region subproblem. Since such a Hessian is also positive definite, we proposed to apply the Q-less QR factorization, and then the upper triangular system is solved with the Gaussian elimination. The Q-less QR factorization is obtained with the Givens rotations, and it is implemented in LAPACK. Moreover, to reduce a computational cost even more, and to make the Hessian better conditioned for a sparse system matrix, we combine the QR factorization with an active-set method that is often used for convex QP.

We applied our new NMF algorithm to Blind Source Separation (BSS) problems in which mixed signals and images are used. The results demonstrate high performance of the proposed algorithm.

The paper is organized as follows: Section 2 presents the problem formulation, starting from a general approach of nonlinear programming, and finishing to formulation of the trust-region subproblem for NMF. This problem is then solved with the algorithm proposed in Section 3. The numerical results obtained for some BSS problems are illustrated in Section 4. Finally, Section 5 contains the conclusions and refers to the future works.

We will adopt the following notation throughout the paper: vectors are denoted by bold lower case letters, matrices by bold capital letters, and entries of vectors or matrices by corresponding italic lower case letters. The square brackets stand for a matrix created from all inside entries. For example, $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{I \times J}$ is a real matrix I by J whose the ij -th entry is a_{ij} . Alternatively, $[x_{jk}^{-1}]$ means the matrix which is element-wise inverse to the matrix \mathbf{X} . The diagonal matrix created from the vector \mathbf{x} will be denoted by $\text{diag}(\mathbf{x})$. The symbols $\text{vec}(\cdot)$ and $\text{Matrix}(\cdot)$ denote vectorization and matricization, respectively. We have $\mathbf{a} = \text{vec}(\mathbf{A}) = [a_{11}, a_{21}, \dots, a_{I1}, a_{12}, \dots, a_{IJ}]^T \in \mathbb{R}^{IJ}$, where $\mathbf{A} \in \mathbb{R}^{I \times J}$, and $\mathbf{A} = \text{Matrix}(\mathbf{a})$. Analogically, $\mathbf{a} = \text{vec}(\mathbf{A}^T) = [a_{11}, a_{12}, \dots, a_{1J}, a_{21}, \dots, a_{IJ}]^T \in \mathbb{R}^{IJ}$, where \mathbf{A}^T is a transposed matrix to \mathbf{A} . The vector of all ones will be expressed by $\mathbf{e} = [1, \dots, 1]^T$. The matrix $\mathbf{I}_I = \text{diag}(\mathbf{e}) \in \mathbb{R}^{I \times I}$ is an identity matrix of size I by I . The trace of \mathbf{A} is denoted by $\text{tr}(\mathbf{A})$.

2 Problem formulation

The minimization problems in NMF (Step 1 and 2 in Algorithm 1) are nonlinear due to nonnegativity constraints. In this respect, we consider a general nonlinear problem which is then transformed to a QP problem with nonlinear constraints [4,11,30].

2.1 Nonlinear programming

Let $F(\mathbf{x}) = F(x_1, \dots, x_n) : \mathbb{R}^n \rightarrow \mathbb{R}$, where $\mathbf{x} \in \Omega \subseteq \mathbb{R}^n$, be nonlinear function, and Ω be a feasible set. Nonlinear programming can be expressed by the following nonlinear problem:

$$\min_{\mathbf{x} \in \Omega} F(\mathbf{x}), \quad (2)$$

subject to the constraints:

$$\hat{g}_i(\mathbf{x}) \geq \hat{d}_i : \quad i = 1, 2, \dots, m_1, \quad (3)$$

$$\check{g}_i(\mathbf{x}) \leq \check{d}_i : \quad i = m_1 + 1, \dots, m_2, \quad m_1 \leq m_2, \quad (4)$$

$$\bar{g}_i(\mathbf{x}) = \bar{d}_i : \quad i = m_2 + 1, \dots, m, \quad m_2 \leq m, \quad (5)$$

where $\hat{g}_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, $\check{g}_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, and $\bar{g}_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ are continuously differentiable functions at some point $\mathbf{x}^* \in \Omega$. If \mathbf{x}^* is a local minimum, then the following first-order Karush-Kuhn-Tucker (KKT) conditions are satisfied:

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\theta}^*, \boldsymbol{\nu}^*) = 0, \quad \boldsymbol{\lambda}^* = [\lambda_i^*], \quad \boldsymbol{\theta}^* = [\theta_i^*], \quad \boldsymbol{\nu}^* = [\nu_i^*], \quad (6)$$

$$\theta_i^*(\hat{g}_i(\mathbf{x}^*) - \hat{d}_i) = 0, \quad \lambda_i^*(\check{g}_i(\mathbf{x}^*) - \check{d}_i) = 0, \quad \nu_i^*(\bar{g}_i(\mathbf{x}^*) - \bar{d}_i) = 0, \quad (7)$$

$$\theta_i^* \leq 0, \quad \lambda_i^* \geq 0, \quad \nu_i^* \text{ is of arbitrary sign}, \quad (8)$$

where $\boldsymbol{\lambda}^*$, $\boldsymbol{\theta}^*$, and $\boldsymbol{\nu}^*$ are Lagrange multipliers at point \mathbf{x}^* , and

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\nu}) = & F(\mathbf{x}) + \sum_{i=1}^{m_1} \theta_i (\hat{g}_i(\mathbf{x}) - \hat{d}_i) \\ & + \sum_{i=m_1+1}^{m_2} \lambda_i (\check{g}_i(\mathbf{x}) - \check{d}_i) + \sum_{i=m_2+1}^m \nu_i (\bar{g}_i(\mathbf{x}) - \bar{d}_i), \end{aligned} \quad (9)$$

is a Lagrangian functional associated with the problem (2). From (6), we have

$$\nabla_x F(\mathbf{x}^*) + (\boldsymbol{\theta}^*)^T \nabla_x \hat{\mathbf{g}}(\mathbf{x}^*) + (\boldsymbol{\lambda}^*)^T \nabla_x \check{\mathbf{g}}(\mathbf{x}^*) + (\boldsymbol{\nu}^*)^T \nabla_x \bar{\mathbf{g}}(\mathbf{x}^*) = 0. \quad (10)$$

2.2 Quadratic programming

The step 1 in Algorithm 1 means to solve the linear multiple system: $\mathbf{A}\mathbf{X} = \mathbf{Y}$ with respect to \mathbf{X} , but the step 2 solves the transposed linear multiple system: $\mathbf{X}^T \mathbf{A}^T = \mathbf{Y}^T$ with respect to \mathbf{A} . Both systems are subjected to the nonnegativity constraints. Beside of these intrinsic constraints, we also admit some additional ones.

It is well-know that the least-squares solution $\mathbf{X}_{LS} = \mathbf{A}^+ \mathbf{Y}$ (without any constraints) to the perturbed system of linear equations given by (1), where \mathbf{A}^+ is the Moore-Penrose pseudo-inverse of \mathbf{A} , satisfies inequality: $\|\mathbf{X}_{LS}\|_F \gg \|\mathbf{X}_{exact}\|_F$, where \mathbf{X}_{exact} is an exact solution, i.e. for $\mathbf{V} = \mathbf{0}$. The stabilization of the solution can be achieved by some regularization which enforces $\|\mathbf{X}_{LS}\|_F \leq \delta_X$, where δ_X is assumed to be known from a prior information about the solution.

Thus we have the following constraints on the factors to be estimated: $\|\mathbf{X}\|_F \leq \delta_X$ and $x_{jk} \geq 0$, and $\|\mathbf{A}\|_F \leq \delta_A$ and $a_{ij} \geq 0$. With these constraints, we formulate the QP problem, separately for each minimization step in Alg. 1.

- *Step 1:* Let $F(\mathbf{x})$ in (2) be defined by a typical quadratic form that appears in a standard QP problem [4, p.152], [30, p.441]:

$$F(\mathbf{X}) = \frac{1}{2} \text{vec}(\mathbf{X})^T \mathbf{Q}_X \text{vec}(\mathbf{X}) + \text{tr}(\mathbf{C}_X^T \mathbf{X}), \quad \text{for } \mathbf{x} \leftarrow \mathbf{X}. \quad (11)$$

Note that $\text{tr}(\mathbf{C}_X^T \mathbf{X}) = \text{vec}(\mathbf{C}_X)^T \text{vec}(\mathbf{X})$. We assume that $\mathbf{Q}_X \in \mathbb{R}^{JK \times JK}$ is symmetric and positive definite matrix, $\mathbf{C}_X \in \mathbb{R}^{J \times K}$, and the constraints (3)–(5) have the form:

$$\begin{aligned} \hat{g}_{jk}(\mathbf{X}) &= x_{jk}, & \hat{d}_{jk} &= 0, & j &= 1, \dots, J, & k &= 1, \dots, K, \\ \check{g}(\mathbf{X}) &= \|\mathbf{X}\|_F, & \check{d} &= \delta_X, \\ \bar{g}(\mathbf{X}) &= 0, & \bar{d} &= 0. \end{aligned}$$

Our objective is to find the right matrices \mathbf{Q}_X and \mathbf{C}_X for our problems. The Lagrangian functional can be written as

$$\mathcal{L}(\mathbf{X}, \lambda_X, \boldsymbol{\Theta}_X) = F(\mathbf{X}) + \frac{\lambda_X}{2} \left(\|\mathbf{X}\|_F^2 - \delta_X^2 \right) + \text{tr}(\boldsymbol{\Theta}_X^T \mathbf{X}), \quad (12)$$

with the KKT conditions (6)–(8):

$$\nabla_X F(\mathbf{X}) + \lambda_X \mathbf{X} + \boldsymbol{\Theta}_X = 0, \quad \lambda_X \geq 0, \quad \theta_{jk}^{(X)} \leq 0, \quad \boldsymbol{\Theta}_X = [\theta_{jk}^{(X)}], \quad (13)$$

$$\lambda_X \left(\|\mathbf{X}\|_F^2 - \delta_X^2 \right) = 0, \quad \text{tr}(\mathbf{\Theta}_X^T \mathbf{X}) = 0. \quad (14)$$

Inserting (11) into (13), \mathbf{X} can be computed from

$$(\mathbf{Q}_X + \lambda_X \mathbf{I}_{JK}) \text{vec}(\mathbf{X}) = -\text{vec}(\mathbf{C}_X + \mathbf{\Theta}_X). \quad (15)$$

- *Step 2:* Now we solve the transposed system: $\mathbf{X}^T \mathbf{A}^T = \mathbf{Y}^T$, and our new variable of interest is $\text{vec}(\mathbf{A}^T)$. We assume that $F(\mathbf{x})$ in (2) is also defined by the similar quadratic form:

$$F(\mathbf{A}) = \frac{1}{2} \text{vec}(\mathbf{A}^T)^T \mathbf{Q}_A \text{vec}(\mathbf{A}^T) + \text{tr}(\mathbf{C}_A^T \mathbf{A}^T), \quad \text{for } \mathbf{x} \leftarrow \mathbf{A}, \quad (16)$$

where $\mathbf{Q}_A \in \mathbb{R}^{IJ \times IJ}$ is symmetric and positive definite matrix, $\mathbf{C}_A \in \mathbb{R}^{J \times I}$, and the constraints (3)–(5) have the form:

$$\begin{aligned} \hat{g}_{ij}(\mathbf{A}) &= a_{ij}, & \hat{d}_{ij} &= 0, & i &= 1, \dots, I, & j &= 1, \dots, J, \\ \check{g}(\mathbf{A}) &= \|\mathbf{A}\|_F, & \check{d} &= \delta_A, \\ \bar{g}(\mathbf{A}) &= 0, & \bar{d} &= 0. \end{aligned}$$

The Lagrangian functional has a similar form, i.e.

$$\mathcal{L}(\mathbf{A}, \lambda_A, \mathbf{\Theta}_A) = F(\mathbf{A}) + \frac{\lambda_A}{2} \left(\|\mathbf{A}\|_F^2 - \delta_A^2 \right) + \text{tr}(\mathbf{\Theta}_A^T \mathbf{A}), \quad (17)$$

and the KKT conditions (6)–(8) are as follows:

$$\nabla_A F(\mathbf{A}) + \lambda_A \mathbf{A} + \mathbf{\Theta}_A = 0, \quad \lambda_A \geq 0, \quad \theta_{ij}^{(A)} \leq 0, \quad \mathbf{\Theta}_A = [\theta_{ij}^{(A)}], \quad (18)$$

$$\lambda_A \left(\|\mathbf{A}\|_F^2 - \delta_A^2 \right) = 0, \quad \text{tr}(\mathbf{\Theta}_A^T \mathbf{A}) = 0. \quad (19)$$

Thus \mathbf{A} can be computed from

$$(\mathbf{Q}_A + \lambda_A \mathbf{I}_{IJ}) \text{vec}(\mathbf{A}^T) = -\text{vec}(\mathbf{C}_A + \mathbf{\Theta}_A^T). \quad (20)$$

Remark 1 *The solutions to both equations (15) and (20) can be regarded as Tikhonov regularized solutions, where the regularization parameters are given by λ_X and λ_A , respectively. These parameters can be adaptively estimated, e.g. by the Generalized Cross-Validation (GCV) technique [3]. It is also possible to apply the estimation technique based on Lanczos bidiagonalization, which is proposed in [7], assuming δ_X and δ_A can be roughly estimated by some prior information on the solutions.*

2.3 Trust-region subproblem

Following the proposition by Lee and Seung [23], we assume the cost function $D(\mathbf{Y}||\mathbf{A}\mathbf{X})$ in Algorithm 1 is expressed by the squared Euclidean distance:

$$D(\mathbf{Y}||\mathbf{A}\mathbf{X}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2, \quad (21)$$

which is optimal for a Gaussian noise. For the function (21), we have

$$\begin{aligned} D(\mathbf{Y}||\mathbf{A}\mathbf{X}) &= \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \\ &= \frac{1}{2} \text{tr}(\mathbf{X}^T \mathbf{A}^T \mathbf{A} \mathbf{X}) - \text{tr}(\mathbf{A}^T \mathbf{Y})^T \mathbf{X} + \frac{1}{2} \text{tr}(\mathbf{Y}^T \mathbf{Y}) \\ &= \frac{1}{2} \text{vec}(\mathbf{X})^T (\mathbf{I}_K \otimes \mathbf{A}^T \mathbf{A}) \text{vec}(\mathbf{X}) - \text{tr}(\mathbf{A}^T \mathbf{Y})^T \mathbf{X} + c, \end{aligned} \quad (22)$$

where $c = \frac{1}{2} \text{tr}(\mathbf{Y}^T \mathbf{Y})$, and \otimes stands for a Kronecker product. Neglecting the constant c and comparing (22) to (11), we have $\mathbf{Q}_X = \mathbf{I}_K \otimes \mathbf{A}^T \mathbf{A}$, $\mathbf{C}_X = -\mathbf{A}^T \mathbf{Y}$ in (11), and alternatively $\mathbf{Q}_A = \mathbf{I}_I \otimes \mathbf{X} \mathbf{X}^T$, $\mathbf{C}_A = -\mathbf{X} \mathbf{Y}^T$ in (16).

Neglecting the terms Θ_X and Θ_A , the solutions to (15) and (20) could be interpreted as usual Tikhonov regularized least-squares solutions. But the non-positive matrices Θ_X and Θ_A of Lagrange multipliers enforce the nonnegative solutions that are no longer the least-squares solutions. Unfortunately, the terms cannot be readily estimated to satisfy the corresponding equations (14) and (19). One of the approaches is to iteratively approximate the solutions to (15) and (20). However, in our case we get rid of the complicated duality gaps $\text{tr}(\Theta_X^T \mathbf{X})$ and $\text{tr}(\Theta_A^T \mathbf{A})$ with adding some penalty terms to the regularized cost function, which assures nonnegative solutions. There are several such penalty functions, e.g. quadratic penalty, logarithmic barrier, or exact penalty, whose properties have been discussed in [30]. Following the proposition by Rojas and Steihaug [39], we selected the logarithmic barrier functions:

$$\Psi_X(\mathbf{X}) = \theta_X \sum_{j=1}^J \sum_{k=1}^K \ln x_{jk}, \quad \Psi_A(\mathbf{A}) = \theta_A \sum_{i=1}^I \sum_{j=1}^J \ln a_{ij}, \quad (23)$$

where $\theta_X \leq 0$ and $\theta_A \leq 0$. The logarithmic barrier functions tend to relax ill-conditioning of Hessians. This approach considerably reduces a number of Lagrange multipliers – instead of matrices Θ_X and Θ_A , we must estimate only two parameters θ_X and θ_A . Thus the penalized cost function is as follows:

$$\begin{aligned}
D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) &= \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \frac{\lambda_X}{2} \left(\|\mathbf{X}\|_F^2 - \delta_X^2 \right) \\
&\quad + \frac{\lambda_A}{2} \left(\|\mathbf{A}\|_F^2 - \delta_A^2 \right) + \Psi_X(\mathbf{X}) + \Psi_A(\mathbf{A}),
\end{aligned} \tag{24}$$

for solving the minimization problems in Step 1 and 2. The penalty terms introduce strong nonlinearity, and to use QP the function $D_R(\mathbf{Y}||\mathbf{A}\mathbf{X})$ is approximated around points \mathbf{X} and \mathbf{A} with second-order Taylor expansions. Thus

$$\begin{aligned}
D_R(\mathbf{Y}||\mathbf{A}(\mathbf{X} + \Delta\mathbf{X})) &= D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) + \text{tr}(\mathbf{G}_X^T \Delta\mathbf{X}) \\
&\quad + \frac{1}{2}\text{vec}(\Delta\mathbf{X})^T \mathbf{H}_X \text{vec}(\Delta\mathbf{X}),
\end{aligned} \tag{25}$$

$$\begin{aligned}
D_R(\mathbf{Y}||(\mathbf{A} + \Delta\mathbf{A})\mathbf{X}) &= D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) + \text{tr}(\mathbf{G}_A^T \Delta\mathbf{A}) \\
&\quad + \frac{1}{2}\text{vec}(\Delta\mathbf{A})^T \mathbf{H}_A \text{vec}(\Delta\mathbf{A}),
\end{aligned} \tag{26}$$

where the corresponding gradients \mathbf{G}_X and \mathbf{G}_A with respect to \mathbf{X} and \mathbf{A} are expressed by:

$$\mathbf{G}_X = \nabla_X D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) = \mathbf{A}^T(\mathbf{A}\mathbf{X} - \mathbf{Y}) + \lambda_X \mathbf{X} + \theta_X [x_{jk}^{-1}] \in \mathbb{R}^{J \times K},$$

$$\mathbf{G}_A = \nabla_A D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) = (\mathbf{A}\mathbf{X} - \mathbf{Y})\mathbf{X}^T + \lambda_A \mathbf{A} + \theta_A [a_{ij}^{-1}] \in \mathbb{R}^{I \times J},$$

and the corresponding Hessians $\mathbf{H}_X \in \mathbb{R}^{JK \times JK}$ and $\mathbf{H}_A \in \mathbb{R}^{IJ \times IJ}$ have the forms:

$$\mathbf{H}_X = \nabla_X^2 D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) = \mathbf{I}_K \otimes \mathbf{A}^T \mathbf{A} + \lambda_X \mathbf{I}_{JK} - \theta_X \text{diag}(\text{vec}([x_{jk}^{-2}])),$$

$$\mathbf{H}_A = \nabla_A^2 D_R(\mathbf{Y}||\mathbf{A}\mathbf{X}) = \mathbf{I}_I \otimes \mathbf{X}\mathbf{X}^T + \lambda_A \mathbf{I}_{IJ} - \theta_A \text{diag}(\text{vec}([a_{ij}^{-2}])).$$

Then we formulate the following trust-region subproblems:

$$\min_{\Delta\mathbf{X}} D_R(\mathbf{Y}||\mathbf{A}(\mathbf{X} + \Delta\mathbf{X})), \quad \text{s.t. } \|\mathbf{X} + \Delta\mathbf{X}\|_F \leq \delta_X, \tag{27}$$

$$\min_{\Delta\mathbf{A}} D_R(\mathbf{Y}||(\mathbf{A} + \Delta\mathbf{A})\mathbf{X}), \quad \text{s.t. } \|\mathbf{A} + \Delta\mathbf{A}\|_F \leq \delta_A. \tag{28}$$

Following Rojas and Steihaug [39], the trust-region subproblems can be transformed to the following QP problems:

$$\min_{\mathbf{Z}_X} \left\{ \frac{1}{2} \text{vec}(\mathbf{Z}_X)^T (\mathbf{Q}_X) \text{vec}(\mathbf{Z}_X) + \text{tr}(\mathbf{C}_X^T \mathbf{Z}_X) \right\}, \quad \mathbf{Z}_X = \mathbf{X} + \Delta\mathbf{X}, \tag{29}$$

$$\min_{\mathbf{Z}_A} \left\{ \frac{1}{2} \text{vec}(\mathbf{Z}_A)^T (\mathbf{Q}_A) \text{vec}(\mathbf{Z}_A) + \text{tr}(\mathbf{C}_A^T \mathbf{Z}_A) \right\}, \quad \mathbf{Z}_A = \mathbf{A}^T + \Delta\mathbf{A}^T, \tag{30}$$

where

$$\mathbf{Q}_X = \mathbf{I}_K \otimes \mathbf{A}^T \mathbf{A} + \lambda_X \mathbf{I}_{JK} - \theta_X \text{diag}(\text{vec}([x_{jk}^{-2}])), \quad (31)$$

$$\mathbf{C}_X = 2\theta_X [x_{jk}^{-1}] - \mathbf{A}^T \mathbf{Y}, \quad \lambda_X \geq 0, \quad \theta_X \leq 0, \quad (32)$$

$$\mathbf{Q}_A = \mathbf{I}_I \otimes \mathbf{X} \mathbf{X}^T + \lambda_A \mathbf{I}_{IJ} - \theta_A \text{diag}(\text{vec}([a_{ij}^{-2}]^T)), \quad (33)$$

$$\mathbf{C}_A = 2\theta_A [a_{ij}^{-1}]^T - \mathbf{X} \mathbf{Y}^T, \quad \lambda_A \geq 0, \quad \theta_A \leq 0. \quad (34)$$

The QP problems given by (29) and (30) are therefore equivalent to the following basic QP problems:

$$\min_{\mathbf{X}} \left\{ \frac{1}{2} \text{vec}(\mathbf{X})^T \left(\mathbf{I}_K \otimes \mathbf{A}^T \mathbf{A} \right) \text{vec}(\mathbf{X}) - \text{tr}(\mathbf{Y}^T \mathbf{A} \mathbf{X}) \right\}, \quad (35)$$

s.t. $x_{jk} \geq 0, \quad \|\mathbf{X}\|_F \leq \delta_X$

$$\min_{\mathbf{A}} \left\{ \frac{1}{2} \text{vec}(\mathbf{A}^T)^T \left(\mathbf{I}_I \otimes \mathbf{X} \mathbf{X}^T \right) \text{vec}(\mathbf{A}^T) - \text{tr}(\mathbf{Y} \mathbf{X}^T \mathbf{A}^T) \right\}, \quad (36)$$

s.t. $a_{ij} \geq 0, \quad \|\mathbf{A}\|_F \leq \delta_A$

Remark 2 Assuming \mathbf{A} and \mathbf{X} are full rank matrices, both \mathbf{Q}_A and \mathbf{Q}_X are positive-definite but possibly very ill-conditioned, especially as \mathbf{A} and \mathbf{X} are very sparse. This is due to the case that $q_{tt}^{(X)} \rightarrow \infty$, where $t = j + (k - 1)J$, $\mathbf{Q}_X = [q_{mn}^{(X)}]$, $m, n \in \{1, \dots, JK\}$, if $x_{jt} \rightarrow 0$ and $|\theta_X| < \infty$. Also, if $a_{ij} \rightarrow 0$ and $|\theta_A| < \infty$, then $q_{tt}^{(A)} \rightarrow \infty$, where $t = j + (i - 1)J$, $\mathbf{Q}_A = [q_{mn}^{(A)}]$, $m, n \in \{1, \dots, IJ\}$. Assuming, the matrices \mathbf{Q}_A and \mathbf{Q}_X are positive-definite, the solutions \mathbf{X}^* and \mathbf{A}^* are global in the separate minimization problems in Algorithms 1. Obviously, the whole NMF alternating minimization does not provide a global solution.

3 Algorithm

The proposed QP based method can be used in many NMF applications. However, we restrict our further considerations to BSS problems for which we presented some results. In BSS applications, $K \gg I \geq J$, and hence, the computation of the matrix \mathbf{Q}_X in (31) may be highly time-consuming or even intractable, even though it is very sparse. Assuming some typical case: $I = 20$, $J = 10$, and $K = 1000$, the matrix \mathbf{Q}_A in (33) has size 200 by 200 with $IJ^2 = 2 \times 10^3$ non-zero entries. The size of \mathbf{Q}_X is 10^4 by 10^4 with $KJ^2 = 10^5$ non-zero entries. For this reason, we do not apply the QP for updating \mathbf{X} . This can be also justified by the fact that the computation of \mathbf{A} needs to solve the system which is much more over-determined than for \mathbf{X} , and hence, this may be better done with the second order method that exploits the information about the curvature of the cost function.

3.1 Updates for \mathbf{X}

For updating \mathbf{X} , we use a very simple Fixed-Point (FP) algorithm that solves a least-squares problem

$$\mathbf{X}_{LS} = \arg \min_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 \right\} \quad (37)$$

with the Moore-Penrose pseudo-inverse of the system matrix, i.e. in our case, the matrix \mathbf{A} , and then, the solution \mathbf{X}_{LS} is projected onto the feasible set $\Omega(\mathbf{X}) = \{x_{jk} : x_{jk} \geq \varepsilon\}$, where ε is a small constant ($\sim 10^{-9}$) for protecting the algorithm against some possible numerical instabilities. Thus

$$\mathbf{X}^* \leftarrow \max\{\varepsilon, \mathbf{A}^+ \mathbf{Y}\}, \quad (38)$$

where \mathbf{A}^+ is a Moore-Penrose pseudo-inverse of \mathbf{A} .

3.2 Updates for \mathbf{A}

The mixing matrix \mathbf{A} can be updated with solving only the QP problem (30). Let us assume the objective function of this problem be denoted by $\Phi(\mathbf{A})$, i.e. $\Phi(\mathbf{A}) = \frac{1}{2} \text{vec}(\mathbf{Z}_A)^T (\mathbf{Q}_A) \text{vec}(\mathbf{Z}_A) + \text{tr}(\mathbf{C}_A^T \mathbf{Z}_A)$. The stationary point \mathbf{A}^* of $\Phi(\mathbf{A})$ occurs if $\nabla_{\mathbf{A}} \Phi(\mathbf{A}^*) = 0$. Thus, the global solution can be obtained solving the following system of linear equations:

$$\mathbf{Q}_A \mathbf{z} = -\text{vec}(\mathbf{C}_A), \quad (39)$$

where $\mathbf{z} = \text{vec}(\mathbf{Z}_A) \in \mathbb{R}^{IJ}$.

The matrix \mathbf{Q}_A may be very ill-conditioned due to the reasons given in Remark 2. However, sufficiently small entries a_{ij} do not need to be updated in successive inner iterations that must be performed to estimate the parameter θ_A in (33) and (34). Removing such entries, a size of the system to be solved decreases, and the "reduced" system becomes better conditioned. To perform this operation, we use the same idea as in well-known active-set methods [30]. Let

$$\mathcal{I}(\mathbf{A}^T) = \left\{ t : a_t \geq \epsilon_A, \quad \text{where } a_t = [\text{vec}(\mathbf{A}^T)]_t \right\} \quad (40)$$

be an inactive-set for matrix \mathbf{A}^T , and ϵ_A be some threshold for entries a_{ij} . Now let us define the matrix $\mathbf{T} = \mathbf{I}_{[*,\mathcal{I}(\mathbf{A}^T)]} \in \mathbb{R}^{IJ \times N}$, where $\mathbf{I} = \mathbf{I}_{IJ}$, the

symbol $*$ means all rows, and N is the number of entries in $\mathcal{I}(\mathbf{A}^T)$. Thus, the "reduced" system can be expressed by:

$$(\mathbf{T}^T \mathbf{Q}_A \mathbf{T}) \mathbf{z}_R = -\mathbf{T}^T \text{vec}(\mathbf{C}_A), \quad (41)$$

where $\mathbf{z}_R = \mathbf{T}^T \mathbf{z} \in \mathbb{R}^N$. If \mathbf{A} is very sparse, we have $N \ll IJ$.

The matrix $\mathbf{Q}_A^{(R)} = \mathbf{T}^T \mathbf{Q}_A \mathbf{T}$ is positive definite, and for the cost function defined by (21), it has a regular block diagonal structure with small blocks. It means that a number of entries under the main diagonal is very small. Due to this we propose to factorize the matrix $\mathbf{Q}_A^{(R)}$ using the Q-less QR factorization which is implemented in LAPACK. Applying the Givens rotations to the under-diagonal entries the upper triangular matrix \mathbf{R} can be computed in relatively small number of arithmetic operations. Thus

$$\mathbf{Q}_A^{(R)} = \mathbf{Q} \mathbf{R}, \quad \text{where } \mathbf{w} = -\mathbf{Q}^T \mathbf{T}^T \text{vec}(\mathbf{C}_A). \quad (42)$$

The orthogonal matrix \mathbf{Q} is not computed explicitly in the Q-less QR factorization but only the vector \mathbf{w} is returned. The new system: $\mathbf{R} \mathbf{z}_R = \mathbf{w}$ with the upper triangular matrix \mathbf{R} can be efficiently solved with the Gaussian elimination.

The system (39) must be solved iteratively for increasing values of θ_A in (33) and (34). The iterative schedule for θ_A can be derived from (20). Combining (39), (33) and (34) with (20), we get

$$\text{vec}(\Theta_A^T) = 2\theta_A \text{vec}([a_{ij}^{-1}]^T) - \theta_A \text{diag}(\text{vec}([a_{ij}^{-2}]^T)) \mathbf{z}. \quad (43)$$

When $\mathbf{z} = \text{vec}(\mathbf{A}^T)$, the duality gap: $\text{tr}(\Theta_A^T \mathbf{A})$ in (17) can be approximated by $\text{vec}(\Theta_A^T)^T \text{vec}(\mathbf{A}^T) = \theta_A IJ$. Following Rojas and Steihaug [39], and Calvetti et al.[7], we use the following iterative rule for this parameter:

$$\begin{aligned} \tilde{\theta}_l^T &= \theta_{l-1} \left(2\text{vec}([\tilde{a}_{ij}^{-1}]^T) - \text{diag}(\text{vec}([\tilde{a}_{ij}^{-2}]^T)) \mathbf{z} \right), \quad \text{where } \tilde{\mathbf{A}} = \mathbf{A}_{l-1}, \\ \theta_l &= \frac{\rho}{IJ} \tilde{\theta}_l^T \text{vec}(\tilde{\mathbf{A}}^T), \end{aligned} \quad (44)$$

where \mathbf{A}_{l-1} is the matrix \mathbf{A} estimated in the $(l-1)$ -th inner iterative step, and $\rho \in (0, 1)$.

It is well-known that the second-order Taylor series as in (25) and (26) approximates the function of interest only in some small vicinity of a given point. This means that the solution to the problem (30) is more exact if the vicinity

$\Delta \mathbf{A}^T$ is smaller, i.e. $\|\Delta \mathbf{A}^T\|_F \rightarrow 0$. To satisfy this condition, and additionally, to ensure $\forall i, j : [\mathbf{A} + \Delta \mathbf{A}]_{ij} > 0$, \mathbf{A} is upgraded by the additive rule: $\mathbf{A} \leftarrow \mathbf{A} + \beta(\mathbf{Z} - \mathbf{A})$, where $\mathbf{Z} = (\text{Matrix}(\mathbf{z}))^T$ and \mathbf{z} is the solution to (39). The parameter β is estimated by the line-search rule [7,39] to ensure upgrades for \mathbf{A} be always positive.

Our algorithm for updating \mathbf{A} combines the trust-region algorithm [30, p.299] with the log-barrier algorithm [30, p.505]. It was also motivated by the *TRUST* $_{\mu}$ algorithm in [39] and the Constrained Tikhonov Regularization algorithm in [7]. Algorithm 2 is our final QP-NMF algorithm.

Following Rojas and Steihaug [39], Calvetti et al.[7], and our preliminary experiments, we set the following parameters in Algorithm 2: $\rho = 10^{-3}$, $\varepsilon_A = 10^{-6}$, $\epsilon = 10^{-9}$ (eps), $L = 4$, $\tau = 0.9995$, $\eta = 10^{-4}$. Except for λ_A , all other parameters can be fixed for all NMF problems.

In all the simulations, we normalized the rows of matrix \mathbf{Y} to unitary variances. This was motivated by practical applications where measured sensor signals have considerable different amplitudes for spatially distributed sensors.

Despite the assumption $\lambda_A \leq \delta_A$, the columns in the estimated matrix \mathbf{A} may have different lengths due to intrinsic scale and permutation ambiguities. This may lead to very badly scaled estimations for sources, which in consequence reduces the performance. To overcome this problem, we assumed the columns in \mathbf{A} to have a unit l_1 -norm.

4 Numerical results

The proposed QP-NMF algorithm has been extensively tested for many difficult benchmarks for signals and images with various statistical distributions. The simulation results confirmed that the developed algorithm is very robust for BSS problems. We illustrate here two typical BSS problems. The performance of the algorithm is also estimated with a quantity measure: Signal-to-Interference Ratio (SIR). The four statistically dependent nonnegative signals shown in Fig. 1(a) have been mixed by very ill-conditioned, Hilbert matrix ($\mathbf{A} \in \mathbb{R}^{8 \times 4} : a_{ij} = (i + j - 1)^{-1}$) with condition number $\kappa = 4428$. Fig. 1(b) presents the mixed signals. Fig. 1(c) illustrates the best-SIR estimate that is obtained with the standard Lee-Seung NMF algorithm that minimizes the squared Euclidean distance (21). The best-SIR estimate means such estimates that have the highest mean-SIR value among all the Monte Carlo (MC) samples. For this test, we perform the MC analysis with 100 trials. The standard Lee-Seung NMF algorithm fails to give satisfactory estimates of the original

Algorithm 2. QP-NMF

Set $\rho \in (0, 1), \varepsilon, \varepsilon_A, \lambda_A, L, \tau, \eta$
Randomly initialize: \mathbf{A} ,

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

Step 1: $\mathbf{X} \leftarrow \max\{\varepsilon, (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y}\}$,

Step 2: $\mathbf{I} = \mathbf{I}_{IJ} \in \mathbb{R}^{IJ \times IJ}, \mathbf{A} = \mathbf{E} \in \mathbb{R}^{I \times J}$ – matrix of all ones
 $\bar{\mathbf{Q}} = \mathbf{I}_I \otimes \mathbf{X} \mathbf{X}^T + \lambda_A \mathbf{I}_{IJ}, \bar{\mathbf{C}} = \mathbf{Y} \mathbf{X}^T$,
 $\mathbf{G}_0 = \nabla_{\mathbf{A}} D(\mathbf{Y} \| \mathbf{A} \mathbf{X}) = (\mathbf{Y} - \mathbf{A} \mathbf{X}) \mathbf{X}^T, \theta = -\frac{\rho}{IJ} \text{tr}(\mathbf{G}_0 \mathbf{A}^T)$,

For $l = 1, 2, \dots, L$ % Inner iterations

$\mathbf{Q}_A \leftarrow \bar{\mathbf{Q}} - \theta \text{diag}(\text{vec}([a_{ij}^{-2}]^T))$,
 $\mathbf{C}_A \leftarrow 2\theta [a_{ij}^{-1}]^T - \bar{\mathbf{C}}^T$,
 $\mathbf{a} \leftarrow \text{vec}(\mathbf{A}^T), \mathbf{a} = [a_n] \in \mathbb{R}^{IJ}, n = 1, \dots, IJ$,
 $\mathcal{I} = \{t : a_t \geq \varepsilon_A\}$, % Inactive set
 $\mathbf{T} = \mathbf{I}_{[*], \mathcal{I}}, \tilde{\mathbf{a}} = \mathbf{T}^T \mathbf{a}, \tilde{\mathbf{Q}} = \mathbf{T}^T \mathbf{Q}_A \mathbf{T}, \tilde{\mathbf{C}} = -\mathbf{T}^T \text{vec}(\mathbf{C}_A)$,
 $[\mathbf{R}, \mathbf{w}] \leftarrow \text{lqr}(\tilde{\mathbf{Q}}, \tilde{\mathbf{C}})$, % Q-less QR factorization
 $\tilde{\mathbf{z}} = \mathbf{R} \setminus \mathbf{w}$, % Gaussian elimination
 $\tilde{\mathbf{h}} = \tilde{\mathbf{z}} - \tilde{\mathbf{a}}, \beta = \min \left\{ 1, \tau \min_n \frac{\tilde{a}_n}{\tilde{h}_n} \right\}$,
 $\tilde{\mathbf{a}} \leftarrow \tilde{\mathbf{a}} + \beta \tilde{\mathbf{h}}$,
 $\tilde{\boldsymbol{\theta}} \leftarrow \theta \left(2 \text{vec}([\tilde{a}_{ij}^{-1}]^T) - \text{diag}(\text{vec}([\tilde{a}_{ij}^{-2}]^T)) \tilde{\mathbf{z}} \right)$,
 $\theta \leftarrow \left(\frac{\rho}{IJ} \right) \tilde{\boldsymbol{\theta}}^T \tilde{\mathbf{a}}$,
 $\mathbf{a} = [a_n]$, where $a_n = \begin{cases} \tilde{a}_n, & \text{if } n \in \mathcal{I}, \\ 0, & \text{otherwise} \end{cases}$
 $\mathbf{A} \leftarrow (\text{Matrix}(\mathbf{a}))^T$,
if $|\boldsymbol{\theta}^T \tilde{\mathbf{a}}| < \eta \|\mathbf{A}\|_F$
Break

End

End % Inner iteration

$a_{ij} \leftarrow \frac{a_{ij}}{\sum_{q=1}^I a_{qj}}$, % Normalization to unit l_1 -norm columns

End % Main NMF loop (alternating)

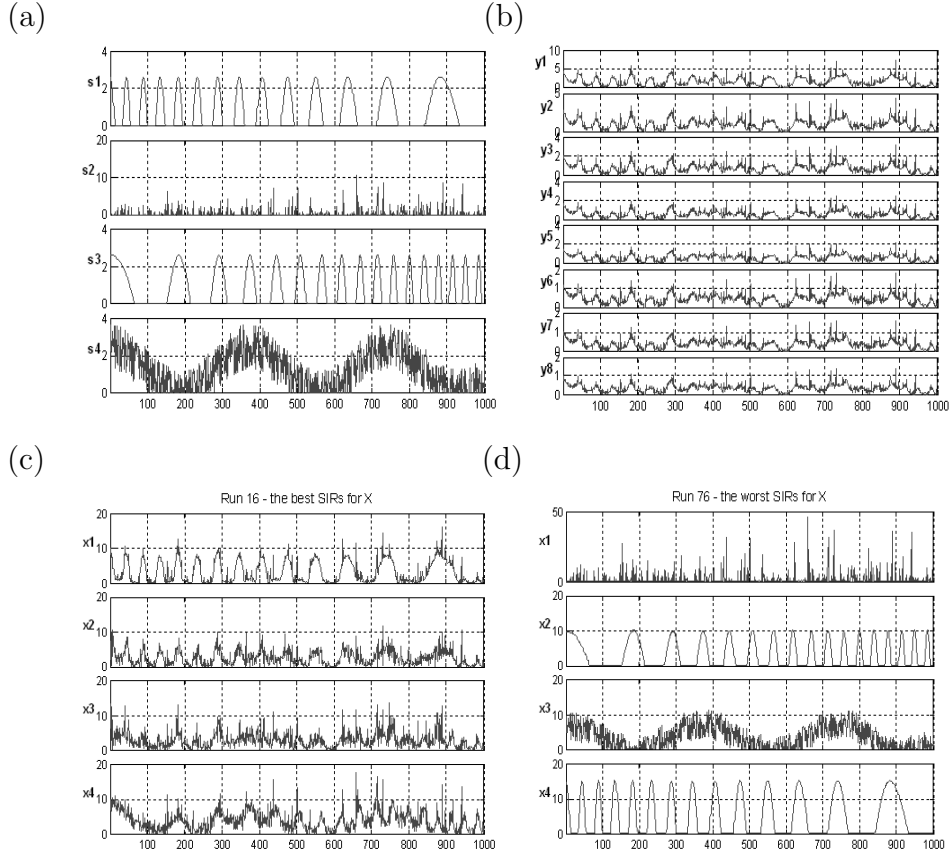


Fig. 1. Example 1: (a) Original 4 source signals; (b) Observed 8 mixed signals with Hilbert mixing matrix $\mathbf{A} \in \mathbb{R}^{8 \times 4}$; (c) Estimated source signals (the best Monte Carlo sample) using the standard Lee-Seung algorithm for the squared Euclidean distance (SIR = 10.2, -1.9, 2.2, 4.8 [dB], respectively); (d) Estimated source signals (the worst Monte Carlo sample) using our QP-NMF algorithm for $\lambda_A = 2000$ (SIR = 36.6, 40.1, 29.1 18.3 [dB], respectively).

sources. The result shown in Fig. 1(d) is obtained with our QP-NMF algorithm (Algorithm 2) for $\lambda_A = 2000$. In this case, we present the worst-SIR estimate, which means that all other estimates are better than this case.

Fig. 2 presents the histograms from 100 mean-SIR samples obtained for the data used in Example 1 (Fig. 1) using two different algorithms. The histogram shown in Fig. 2(a) is generated with the standard Lee-Seung NMF algorithm. Figs. 2(a)–(d) are obtained with Algorithm 2 for different values of λ_A . Note that raising this parameter the mean of the samples slightly decreases, but together with considerable reduction in a standard deviation of the samples. For $\lambda_A = 2000$, the worst mean-SIR sample has a value above 30 [dB], which is fully satisfactory. This suggests that our algorithm in this case is rather convergent to the global minimum of the cost function¹. This is justified by

¹ Even a quadratic cost function $D(\mathbf{Y}||\mathbf{A}\mathbf{X}) = \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F$ with respect to both sets of arguments (\mathbf{A} and \mathbf{X}) may have many local minima.

the fact that a higher value of λ_A ensures the stronger Tikhonov regularization, and the solution (matrix \mathbf{A}) is much smoother.

We have also used six natural images (faces) which are very statistically dependent. The original images shown in Fig. 3 (a) have been mixed with uniformly distributed random matrix $\mathbf{A} \in \mathbb{R}^{12 \times 6}$. The mixtures are shown in Fig. 3 (b). Again, the standard Lee-Seung algorithm fails to give satisfactory results,

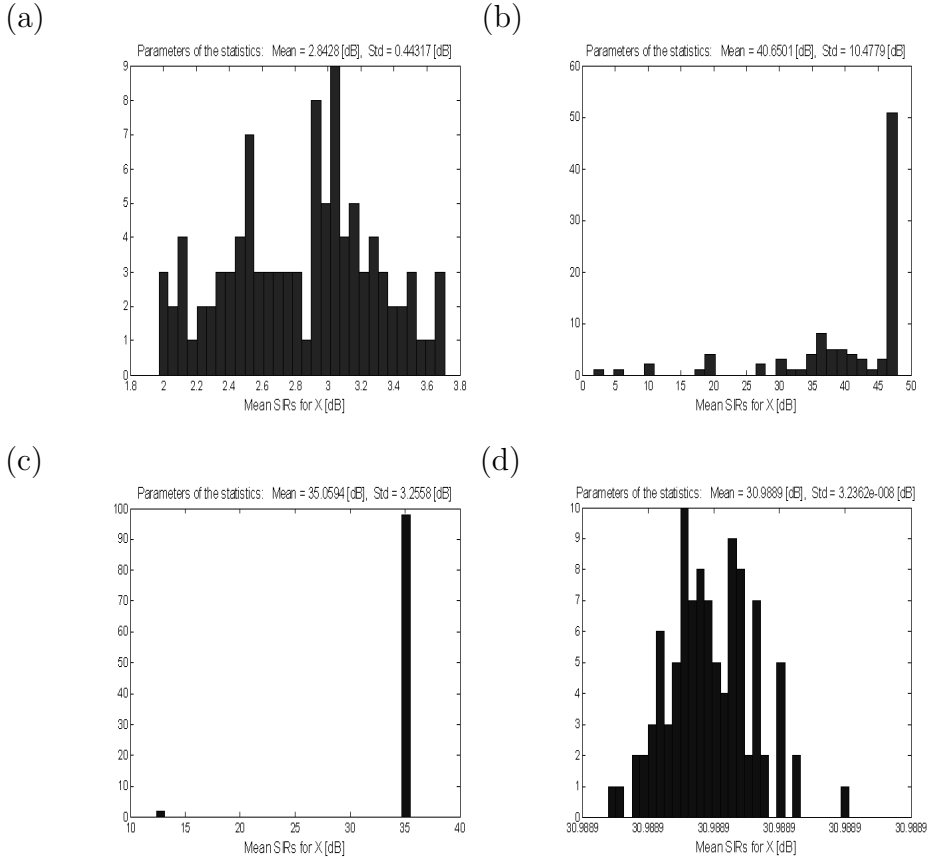


Fig. 2. Histograms from 100 mean-SIR samples obtained for Hilbert mixed signals (Fig. 1) using the following algorithms initialized with uniformly distributed random initial matrices \mathbf{A} : (a) the standard Lee-Seung algorithm for the squared Euclidean distance; (b) QP-NMF algorithm for $\lambda_A = 200$; (c) QP-NMF algorithm for $\lambda_A = 1000$; (d) QP-NMF algorithm for $\lambda_A = 2000$.

which is demonstrated in Fig. 3 (c). The images separated with Algorithm 2 are illustrated in Fig. 3 (d) for $\lambda_A = 100$.

5 Conclusions

The primary objective of this work has been to demonstrate that the second order optimization methods combined with the standard first-order techniques



Fig. 3. Example 2: (a) Original 6 source images; (b) Observed 12 mixed images (uniformly distributed random mixing matrix); (c) Estimated source images using the standard Lee-Seung algorithm for the squared Euclidean distance (SIR = 12.1, 12.4, 13, 14.9, 12.2, 14.7 [dB], respectively); (d) Estimated source images using our QP-NMF algorithm for $\lambda_A = 100$ (SIR = 28.5, 35.1, 41.8, 27.2, 21.4, 34 [dB], respectively).

provides very effective and reliable tools for the nonnegative matrix factorization problems.

We have developed a novel second-order NMF algorithm (Algorithm 2) that is based on Quadratic Programming (QP). This algorithm has been extensively tested for blind signal and image separation problems. Even for very ill-conditioned problems and statistically dependent sources we have received very good performance for various difficult benchmarks.

The features of the proposed algorithm include:

- High efficiency and reliability (this has been confirmed by extensive Monte Carlo simulations)
- Relatively small number of free parameters. Only one parameter notice-

able affects the results, i.e. the regularization parameter λ_A . In our future work, we will consider estimating the parameter by some in-line procedure (automatically from data). One of the approaches is to use the estimation technique proposed by [7].

- It is applicable for estimation of the matrix \mathbf{A} or \mathbf{X} , or both matrices, depending on dimensions of the problem.
- The proposed algorithm is very fast, and our both implementations of the QP-NMF and standard Lee-Seung algorithms have comparable computational costs (comparing elapsed times). The implementation of the QP-NMF algorithm can be found in the Appendix.
- Further extensions and improvements of the algorithm are possible, especially for various cost functions and additional constraints such as sparseness or smoothness.
- Using this algorithm with the multilayer technique [12,13], we obtained even better results.

We implemented the proposed algorithm in our software NMFLAB for Signal and Image Processing [14].

6 Appendix

Here we present the MATLAB code for the function that performs Step 2 in Algorithm 2.

```
% Nonnegative Matrix Factorization with Quadratic Programming
% The function iptrnr(Y,X,no_iter,lambdaA,rho,eta,epsil_A)
% returns the matrix A for the constrained system of
% linear equations: AX = Y

function A = iptrnr(Y,X,no_iter,lambdaA,rho,eta,epsil_A)

% INPUTS:
% > Y:      observation matrix [I by K] of mixed signals
% > X:      source component matrix [J by K]
% > no_iter: number of inner iterations
% > lambdaA: Tikhonov regularization parameter
% > rho:    auxiliary parameter for defining
%          the log barrier parameter
% > eta:    parameter in stopping rule for iterations
% > epsil_A: threshold for active-set variables

% OUTPUT:
% > A:      mixing matrix [I by J]
```

```

[J,K] = size(X); [I,Ky] = size(Y); M = I*J;

% Initialization
A = ones(I,J);
Q_bar = spalloc(M,M,M*J); Q_tilde = spalloc(M,M,M*J);

B = X*X'; C_bar = X*Y';
Q_bar = kron(speye(I),B); %Hessian of D(Y||AX) with respect to A
Go = (C_bar - B*A'); % Gradient of D(Y||AX) with respect to A^T
At = A';
theta = -(rho/(M))*abs(Go(:)'+At(:)); % initialization for
                                     %log barrier parameter

l = 0;
while l < no_iter % inner iterations

    l = l + 1;
    a = At(:);
    active = find(a < epsil_A); % active set
    inactive = find(a >= epsil_A); % inactive set

    if ~isempty(active)
        a_tilde = a;          a_tilde(active) = [ ];
        N = length(a_tilde);
        c_tilde = C_bar(:);   c_tilde(active) = [ ];
        c_tilde = 2*theta*1./a_tilde - c_tilde;
        Q_tilde = Q_bar;
        Q_tilde(:,active) = [ ]; Q_tilde(active,:) = [ ];
        Q_tilde = Q_tilde - theta*spdiags(1./a_tilde.^2,0,N,N);
    else
        a_tilde = a;
    Q_tilde = Q_bar - theta*spdiags(1./a.^2,0,M,M) + lambda*A*speye(M);
        c_tilde = 2*theta*(1./a) - C_bar(:);
    end

    [Q,R] = qr(Q_tilde,-c_tilde); % Q-less QR factorization
    z_tilde = R\Q; % Gaussian elimination
    h_tilde = z_tilde - a_tilde;
    beta = min(1, 0.9995*min(a_tilde./abs(h_tilde)));
    a_tilde = a_tilde + beta*h_tilde;

    theta_tilde = theta*(2*(1./a_tilde)-(1./a_tilde.^2).*z_tilde);
    theta = ( theta_tilde'*a_tilde);

    if ~isempty(active)

```

```

        an = zeros(M,1);   an(inactive) = a_tilde;   a = an;
    else
        a = a_tilde;
    end

    if abs(theta) < eta*norm(a) % stopping rule
        break
    end
    theta = (rho/sqrt(M))*theta;
    At = reshape(a,J,I);

end A = At';

A = A*diag(1./(sum(A,1)+eps)); % normalization

```

References

- [1] J-H. Ahn, S. Kim, J-H Oh, and S. Choi. Multiple nonnegative-matrix factorization of dynamic PET images. In *ACCV*, 2004.
- [2] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 2006. submitted.
- [3] Å. Björck. *Numerical Methods for Least-Squares Problems*. SIAM, Philadelphia, 1996.
- [4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, U.K., 2004.
- [5] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. volume 101, pages 4164–4169. *PNAS*, 2000.
- [6] C. Byrne. Accelerating the EMMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods. *IEEE Transactions on Image Processing*, 7:100 – 109, 1998.
- [7] D. Calvetti, B. Lewis, L. Reichel, and F. Sgallari. Tikhonov regularization with nonnegativity constraint. *Electronic Transactions on Numerical Analysis*, 18:153–173, 2004.
- [8] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. Pascual-Montano. Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, 7(78), 2006.

- [9] M. T. Chu, F. Diele, R. Plemmons, and S. Ragni. Optimality, computation, and interpretation of nonnegative matrix factorizations. Technical report, Departments of Mathematics and Computer Science, Wake Forest University, U.S.A., 2004.
- [10] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He. Extended SMART algorithms for non-negative matrix factorization. *LNAI*, 4029:548–562, 2006.
- [11] A. Cichocki and R. Unbehauen. *Neural Networks for Optimization and Signal Processing (New revised and improved edition)*. John Wiley & Sons, New York, 1994.
- [12] A. Cichocki and R. Zdunek. Multilayer nonnegative matrix factorization. *Electronics Letters*, 42(16):947–948, 2006.
- [13] A. Cichocki and R. Zdunek. Multilayer nonnegative matrix factorization using projected gradient approaches. *Pattern Analysis and Applications*, page in press, 2006.
- [14] A. Cichocki and R. Zdunek. NMFLAB for Signal and Image Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI RIKEN, Saitama, Japan, 2006. <http://www.bsp.brain.riken.jp>.
- [15] A. Cichocki, R. Zdunek, and S. Amari. Csiszar’s divergences for non-negative matrix factorization: Family of new algorithms. *LNCS*, 3889:32–39, 2006.
- [16] A. Cichocki, R. Zdunek, and S. Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP2006*, volume 5, pages 621–624, Toulouse, France, May 14–19 2006.
- [17] D. Guillamet, B. Schiele, and J. Vitrià. Analyzing non-negative matrix factorization for image classification. In *16th International Conference on Pattern Recognition (ICPR’02)*, volume 2, pages 116–119, Quebec City, Canada, August 2002.
- [18] D. Guillamet and J. Vitrià. Classifying faces with nonnegative matrix factorization. In *Proc. 5th Catalan Conference for Artificial Intelligence*, Castello de la Plana, Spain, 2002.
- [19] D. Guillamet, J. Vitrià, and B. Schiele. Introducing a weighted nonnegative matrix factorization for image classification. *Pattern Recognition Letters*, 24(14):2447–2454, 2003.
- [20] M. Hanke, J. G. Nagy, and C. Vogel. Quasi-Newton approach to nonnegative image restorations. *Linear Algebra and its Applications*, 316:223–236, 2000.
- [21] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.
- [22] K. Lange and R. Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comp. Assisted Tomo.*, 8(2):306–316, 1984.

- [23] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, October 1999.
- [24] J. S. Lee, D. D. Lee, S. Choi, and 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*, pages 556–562, San Diego, CA, December 2001.
- [25] H. Li, T. Adali, and D. Emge 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.
- [26] Ch-J. Lin. Projected gradient methods for non-negative matrix factorization. Technical report, Department of Computer Science, National Taiwan University, 2005.
- [27] W. Liu and N. Zheng. Non-negative matrix factorization based methods for object recognition. *Pattern Recognition Letters*, 25(8):893–897, 2004.
- [28] L. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical J.*, 79(6):745–754, 1974.
- [29] J. G. Nagy and Z. Strakos. Enforcing Nonnegativity in Image Reconstruction Algorithms. volume 4121 of *Mathematical Modeling, Estimation, and Imaging*, pages 182–190, 2000.
- [30] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.
- [31] O. Okun and H. Priisalu. Fast nonnegative matrix factorization and its application for protein fold recognition. *EURASIP Journal on Applied Signal Processing*, 2006:Article ID 71817, 8 pages, 2006.
- [32] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmean, and R. Pacual-Marqui. Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(3):403–415, 2006.
- [33] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons. Text mining using non-negative matrix factorizations. In *Proc. SIAM Inter. Conf. on Data Mining*, Orlando, FL, April 2004.
- [34] A. R. De Pierro. On the relation between the ISRA and the EM algorithm for positron emission tomography. *IEEE Trans. Medial Imaging*, 12(2):328–333, June 1993.
- [35] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin. Object characterization from spectral data using nonnegative matrix factorization and information theory. In *Proc. AMOS Tech. Conf.*, Maui, HI, September 2004.
- [36] N. Rao, S. J. Shepherd, and 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, August 2004.

- [37] R. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1):55–59, 1972.
- [38] M. Rojas, S. A. Santos, and D. C. Sorensen. A new matrix-free algorithm for the large-scale trust-region subproblem. *SIAM J. Optim.*, 11(3):611–646, 2000.
- [39] M. Rojas and T. Steihaug. An interior-point trust-region-based method for large-scale non-negative regularization. *Inverse Problems*, 18:1291–1307, 2002.
- [40] 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 *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 71–76, Nara, Japan, April 2003.
- [41] P. Sajda, S. Du, T. R. Brown, R. Stoyanova D. C. Shungu, Xiangling Mao, and 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):1453–1465, 2004.
- [42] P. Sajda, S. Du, and L. Parra. Recovery of constituent spectra using non-negative matrix factorization. In *Proceedings of SPIE*, volume 5207, pages 321–331. Wavelets: Applications in Signal and Image Processing, 2003.
- [43] M. N. Schmidt and M. Mørup. Nonnegative matrix factor 2-D deconvolution for blind single channel source separation. *LNCS*, 3889:700–707, 2006.
- [44] F. Shahnaz, M. Berry, P. Pauca, and R. Plemmons. Document clustering using non-negative matrix factorization. *Journal on Information Processing and Management*, 42:373–386, 2006.
- [45] M. W. Spratling. Learning image components for object recognition. *Journal of Machine Learning Research*, 7:793–815, 2006.
- [46] Y. Wang, Y. Jia, C. Hu, and M. Turk. Non-negative matrix factorization framework for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(4):495–511, 2005.
- [47] R. Zdunek and A. Cichocki. Non-negative matrix factorization with quasi-Newton optimization. *LNAI*, 4029:870–879, 2006.
- [48] A. Ziehe, P. Laskov, K. Pawelzik, and K.-R. Mueller. Non-negative sparse coding for general blind source separation. In *NIPS 2003*, Vancouver, Canada, 2003.