Extended SMART Algorithms for Non-negative Matrix Factorization

Invited Paper

Andrzej Cichocki[1,⋆], Shun-ichi Amari[2],
Rafal Zdunek[1,⋆⋆], Raul Kompass[1,⋆⋆⋆], Gen Hori[1], and Zhaohui He[1,†]

[1] Laboratory for Advanced Brain Signal Processing
[2] Amari Research Unit for Mathematical Neuroscience,
BSI, RIKEN, Wako-shi Japan

Abstract. In this paper we derive a family of new extended SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) algorithms for Non-negative Matrix Factorization (NMF). The proposed algorithms are characterized by improved efficiency and convergence rate and can be applied for various distributions of data and additive noise. Information theory and information geometry play key roles in the derivation of new algorithms. We discuss several loss functions used in information theory which allow us to obtain generalized forms of multiplicative NMF learning adaptive algorithms. We also provide flexible and relaxed forms of the NMF algorithms to increase convergence speed and impose an additional constraint of sparsity. The scope of these results is vast since discussed generalized divergence functions include a large number of useful loss functions such as the Amari α– divergence, Relative entropy, Bose-Einstein divergence, Jensen-Shannon divergence, J-divergence, Arithmetic-Geometric (AG) Taneja divergence, etc. We applied the developed algorithms successfully to Blind (or semi blind) Source Separation (BSS) where sources may be generally statistically dependent, however are subject to additional constraints such as non-negativity and sparsity. Moreover, we applied a novel multilayer NMF strategy which improves performance of the most proposed algorithms.

1 Introduction and Problem Formulation

NMF (Non-negative Matrix Factorization) called also PMF (Positive Matrix Factorization) is an emerging technique for data mining, dimensionality reduction, pattern recognition, object detection, classification, gene clustering, sparse nonnegative representation and coding, and blind source separation (BSS)

⋆ On leave from Warsaw University of Technology, Poland.
⋆⋆ On leave from Institute of Telecommunications, Teleinformatics and Acoustics, Wroclaw University of Technology, Poland.
⋆⋆⋆ Freie University, Berlin, Germany.
† On leave from the South China University, Guangzhou, China.

L. Rutkowski et al. (Eds.): ICAISC 2006, LNAI 4029, pp. 548–562, 2006.
© Springer-Verlag Berlin Heidelberg 2006

[1, 2, 3, 4, 5, 6]. The NMF, first introduced by Paatero and Trapper, and further investigated by many researchers [7, 8, 9, 10, 4, 11, 12], does not assume explicitly or implicitly sparseness, smoothness or mutual statistical independence of hidden (latent) components, however it usually provides quite a sparse decomposition [1, 13, 9, 5]. NMF has already found a wide spectrum of applications in PET, spectroscopy, chemometrics and environmental science where the matrices have clear physical meanings and some normalization or constraints are imposed on them (for example, the matrix $\boldsymbol{A}$ has columns normalized to unit length) [7, 2, 3, 5, 14, 15]. Recently, we have applied NMF with temporal smoothness and spatial constraints to improve the analysis of EEG data for early detection of Alzheimer's disease [16]. A NMF approach is promising in many applications from engineering to neuroscience since it is designed to capture alternative structures inherent in data and, possibly to provide more biological insight. Lee and Seung introduced NMF in its modern formulation as a method to decompose patterns or images [1, 13].

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

$$\boldsymbol{Y} = \boldsymbol{AX} + \boldsymbol{V}, \qquad \boldsymbol{A} \geq 0, \;\; \boldsymbol{X} \geq 0 \tag{1}$$

or equivalently $\boldsymbol{y}(k) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{v}(k), \;\; k = 1, 2, \ldots, N$ or in a scalar form as $y_i(k) = \sum_{j=1}^{n} a_{ij} x_j(k) + \nu_i(k), \;\; i = 1, \ldots, m, \;\;$ with $a_{ij} \geq 0$ and $x_{jk} \geq 0$ where $\boldsymbol{V} \in \mathbb{R}^{m \times N}$ represents the noise or error matrix (depending on applications), $\boldsymbol{y}(k) = [y_1(k), \ldots, y_m(k)]^T$ is a vector of the observed signals (typically positive) at the discrete time instants[1] k while $\boldsymbol{x}(k) = [x_1(k), \ldots, x_n(k)]^T$ is a vector of nonnegative components or source signals at the same time instant [17]. Due to additive noise the observed data might sometimes take negative values. In such a case we apply the following approximation: $\hat{y}_i(k) = y_i(k)$ if $y_i(k)$ is positive and otherwise $\hat{y}_i(k) = \varepsilon$, where ε is a small positive constant. Our objective is to estimate the mixing (basis) matrix $\boldsymbol{A}$ and sources $\boldsymbol{X}$ subject to nonnegativity constraints of all entries of $\boldsymbol{A}$ and $\boldsymbol{X}$. Usually, in BSS applications it is assumed that $N >> m \geq n$ and n is known or can be relatively easily estimated using SVD or PCA. Throughout this paper, we use the following notations: $x_j(k) = x_{jk}$, $y_i(k) = y_{ik}$ and $z_{ik} = [\boldsymbol{AX}]_{ik}$ means ik-th element of the matrix $(\boldsymbol{AX})$, and the ij-th element of the matrix $\boldsymbol{A}$ is denoted by a_{ij}.

The main objective of this contribution is to derive a family of new flexible and improved NMF algorithms that allow to generalize or combine different criteria in order to extract physically meaningful sources, especially for biomedical signal applications such as EEG and MEG.

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

2 Extended Lee-Seung Algorithms and Fixed Point Algorithms

Although the standard NMF (without any auxiliary constraints) provides sparseness of its component, we can achieve some control of this sparsity as well as smoothness of components by imposing additional constraints in addition to non-negativity constraints. In fact, we can incorporate smoothness or sparsity constraints in several ways [9]. One of the simple approach is to implement in each iteration step a nonlinear projection which can increase the sparseness and/or smoothness of estimated components. An alternative approach is to add to the loss function suitable regularization or penalty terms. Let us consider the following constrained optimization problem:

Minimize:

$$D_F^{(\alpha)}(\boldsymbol{A}, \boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \alpha_A J_A(\boldsymbol{A}) + \alpha_X J_X(\boldsymbol{X}) \tag{2}$$

$$s.t. \;\; a_{ij} \geq 0, \quad x_{jk} \geq 0, \;\; \forall\, i, j, k, \tag{3}$$

where α_A and $\alpha_X \geq 0$ are nonnegative regularization parameters and terms $J_X(\boldsymbol{X})$ and $J_A(\boldsymbol{A})$ are used to enforce a certain application-dependent characteristics of the solution. As a special practical case we have $J_X(\boldsymbol{X}) = \sum_{jk} f_X(x_{jk})$, where $f(\cdot)$ are suitably chosen functions which are the measures of smoothness or sparsity. In order to achieve sparse representation we usually choose $f(x_{jk}) = |x_{jk}|$ or simply $f(x_{jk}) = x_{jk}$, or alternatively $f(x_{jk}) = x_{jk} \ln(x_{jk})$ with constraints $x_{jk} \geq 0$. Similar regularization terms can be also implemented for the matrix $\boldsymbol{A}$. Note that we treat both matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ in a symmetric way. Applying the standard gradient descent approach, we have

$$a_{ij} \leftarrow a_{ij} - \eta_{ij} \frac{\partial D_F^{(\alpha)}(\boldsymbol{A}, \boldsymbol{X})}{\partial a_{ij}}, \quad x_{jk} \leftarrow x_{jk} - \eta_{jk} \frac{\partial D_F^{(\alpha)}(\boldsymbol{A}, \boldsymbol{X})}{\partial x_{jk}}, \tag{4}$$

where η_{ij} and η_{jk} are positive learning rates. The gradient components can be expressed in a compact matrix form as:

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{A}, \boldsymbol{X})}{\partial a_{ij}} = [-\boldsymbol{Y}\boldsymbol{X}^T + \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T]_{ij} + \alpha_A \frac{\partial J_A(\boldsymbol{A})}{\partial a_{ij}}, \tag{5}$$

$$\frac{\partial D_F^{(\alpha)}(\boldsymbol{A}, \boldsymbol{X})}{\partial x_{jk}} = [-\boldsymbol{A}^T\boldsymbol{Y} + \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X}]_{jk} + \alpha_X \frac{\partial J_X(\boldsymbol{X})}{\partial x_{jk}}. \tag{6}$$

Here, we follow the Lee and Seung approach to choose specific learning rates [1, 3]:

$$\eta_{ij} = \frac{a_{ij}}{[\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T]_{ij}}, \quad \eta_{jk} = \frac{x_{jk}}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk}}, \tag{7}$$

that leads to a generalized robust multiplicative update rules:

$$a_{ij} \leftarrow a_{ij} \frac{\Big[\boldsymbol{Y}\,\boldsymbol{X}^T]_{ij} - \alpha_A \varphi_A(a_{ij})\Big]_\varepsilon}{[\boldsymbol{A}\,\boldsymbol{X}\,\boldsymbol{X}^T]_{ij} + \varepsilon}, \tag{8}$$

$$x_{jk} \leftarrow x_{jk} \frac{\Big[[\boldsymbol{A}^T\,\boldsymbol{Y}]_{jk} - \alpha_X\ \varphi_X(x_{jk})\Big]_\varepsilon}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk} + \varepsilon}, \tag{9}$$

where the nonlinear operator is defined as $[x]_\varepsilon = \max\{\varepsilon, x\}$ with a small positive ε and the functions $\varphi_A(a_{ij})$ and $\varphi_X(x_{jk})$ are defined as

$$\varphi_A(a_{ij}) = \frac{\partial J_A(\boldsymbol{A})}{\partial a_{ij}}, \quad \varphi_X(x_{jk}) = \frac{\partial J_X(\boldsymbol{X})}{\partial x_{jk}}. \tag{10}$$

Typically, $\varepsilon = 10^{-16}$ is introduced in order to ensure non-negativity constraints and avoid possible division by zero. The above Lee-Seung algorithm can be considred as an extension of the well known ISRA (Image Space Reconstruction Algorithm) algorithm. The above algorithm reduces to the standard Lee-Seung algorithm for $\alpha_A = \alpha_X = 0$. In the special case, by using the l_1-norm regularization terms $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for both matrices $\boldsymbol{X}$ and $\boldsymbol{A}$ the above multiplicative learning rules can be simplified as follows:

$$a_{ij} \leftarrow a_{ij} \frac{\Big[[\boldsymbol{Y}\,\boldsymbol{X}^T]_{ij} - \alpha_A\Big]_\varepsilon}{[\boldsymbol{A}\,\boldsymbol{X}\,\boldsymbol{X}^T]_{ij} + \varepsilon}, \qquad x_{jk} \leftarrow x_{jk} \frac{\Big[[\boldsymbol{A}^T\,\boldsymbol{Y}]_{jk} - \alpha_X\Big]_\varepsilon}{[\boldsymbol{A}^T\boldsymbol{A}\,\boldsymbol{X}]_{jk} + \varepsilon}, \tag{11}$$

with normalization in each iteration as follows $a_{ij} \leftarrow a_{ij} / \sum_{i=1}^{m} a_{ij}$. Such normalization is necessary to provide desired sparseness. Algorithm (11) provides a sparse representation of the estimated matrices and the sparseness measure increases with increasing values of regularization coefficients, typically $\alpha_X = 0.01 \sim 0.5$.

It is worth to note that we can derive as alternative to the Lee-Seung algorithm (11) a Fixed Point NMF algorithm by equalizing the gradient components of (5)-(6) (for l_1-norm regularization terms) to zero [18] :

$$\nabla_X D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{X} - \boldsymbol{A}^T\boldsymbol{Y} + \alpha_X = 0, \tag{12}$$

$$\nabla_A D_F^{(\alpha)}(\boldsymbol{Y}||\boldsymbol{A}\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{Y}\boldsymbol{X}^T + \alpha_A = 0. \tag{13}$$

These equations suggest the following fixed point updates rules:

$$\boldsymbol{X} \leftarrow \max\Big\{\varepsilon, \Big[(\boldsymbol{A}^T\boldsymbol{A})^+(\boldsymbol{A}^T\boldsymbol{Y} - \alpha_X)\Big]\Big\} = \Big[(\boldsymbol{A}^T\boldsymbol{A})^+(\boldsymbol{A}^T\boldsymbol{Y} - \alpha_X)\Big]_\varepsilon, \tag{14}$$

$$\boldsymbol{A} \leftarrow \max\Big\{\varepsilon, \Big[(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A)(\boldsymbol{X}\boldsymbol{X}^T)^+\Big]\Big\} = \Big[(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A)(\boldsymbol{X}\boldsymbol{X}^T)^+\Big]_\varepsilon, \tag{15}$$

where $[\boldsymbol{A}]^+$ means Moore-Penrose pseudo-inverse and *max* function is componentwise. The above algorithm can be considered as nonlinear projected Alternating Least Squares (ALS) or nonlinear extension of EM-PCA algorithm.

Furthermore, using the Interior Point Gradient (IPG) approach an additive algorithm can be derived (which is written in a compact matrix form using MATLAB notations):

$$\boldsymbol{A} \leftarrow \boldsymbol{A} - \boldsymbol{\eta}_A * (\boldsymbol{A} \,./\, (\boldsymbol{A} * \boldsymbol{X} * \boldsymbol{X}')) \,.*\, ((\boldsymbol{A} * \boldsymbol{X} - \boldsymbol{Y}) * \boldsymbol{X}'), \tag{16}$$

$$\boldsymbol{X} \leftarrow \boldsymbol{X} - \boldsymbol{\eta}_X * (\boldsymbol{X}./(\boldsymbol{A}' * \boldsymbol{A} * \boldsymbol{X})) \,.*\, (\boldsymbol{A}' * (\boldsymbol{A} * \boldsymbol{X} - \boldsymbol{Y})), \tag{17}$$

where operators .* and ./ mean component-wise multiplications and division, respectively, and $\boldsymbol{\eta}_A$ and $\boldsymbol{\eta}_X$ are diagonal matrices with positive entries representing suitably chosen learning rates [19].

Alternatively, the mostly used loss function for the NMF that intrinsically ensures non-negativity constraints and it is related to the Poisson likelihood is based on the generalized Kullback-Leibler divergence (also called I-divergence):

$$D_{KL1}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left(y_{ik} \, \ln \frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} + [\boldsymbol{AX}]_{ik} - y_{ik} \right), \tag{18}$$

On the basis of this cost function we proposed a modified Lee-Seung learning algorithm:

$$x_{jk} \leftarrow \left(x_{jk} \, \frac{\sum_{i=1}^{m} a_{ij} \, (y_{ik}/[\boldsymbol{A}\,\boldsymbol{X}]_{ik})}{\sum_{q=1}^{m} a_{qj}} \right)^{1+\alpha_{sX}}, \tag{19}$$

$$a_{ij} \leftarrow \left(a_{ij} \, \frac{\sum_{k=1}^{N} x_{jk} \, (y_{ik}/[\boldsymbol{A}\,\boldsymbol{X}]_{ik})}{\sum_{p=1}^{N} x_{jp}} \right)^{1+\alpha_{sA}}, \tag{20}$$

where additional small regularization terms $\alpha_{sX} \geq 0$ and $\alpha_{sX} \geq 0$ are introduced in order to enforce sparseness of the solution, if necessary. Typical values of the regularization parameters are $\alpha_{sX} = \alpha_{sA} = 0.001 \sim 0.005$.

Raul Kompass proposed to apply beta divergence to combine the both Lee-Seung algorithms (11) and (19)-(20) into one flexible and elegant algorithm with a single parameter [10]. Let us consider beta divergence in the following generalized form as the cost for the NMF problem [10, 20, 6]:

$$D_K^{(\beta)}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left(y_{ik} \, \frac{y_{ik}^{\beta} - [\boldsymbol{AX}]_{ik}^{\beta}}{\beta(\beta+1)} + [\boldsymbol{AX}]_{ik}^{\beta} \frac{[\boldsymbol{AX}]_{ik} - y_{ik}}{\beta+1} \right) + \alpha_X \|\boldsymbol{X}\|_1 + \alpha_A \|\boldsymbol{A}\|_1, \tag{21}$$

where α_X and α_A are small positive regularization parameters which control the degree of smoothing or sparseness of the matrices $\boldsymbol{A}$ and $\boldsymbol{X}$, respectively, and l_1-norms $||\boldsymbol{A}||_1$ and $||\boldsymbol{X}||_1$ are introduced to enforce sparse representation of solutions. It is is interesting to note that for $\beta = 1$ we obtain the square Euclidean distance expressed by Frobenius norm (2), while for the singular cases $\beta = 0$ and $\beta = -1$ the beta divergence has to be defined as limiting cases as $\beta \to 0$ and $\beta \to -1$, respectively. When these limits are evaluated one gets for

$\beta \to 0$ the generalized Kullback-Leibler divergence (called I-divergence) defined by equations (18) and for $\beta \to -1$ the Itakura-Saito distance can be obtained:

$$D_{I-S}(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} \left[\ln(\frac{[\boldsymbol{AX}]_{ik}}{y_{ik}}) + \frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} - 1 \right]. \tag{22}$$

The choice of the β parameter depends on statistical distribution of data and the beta divergence corresponds to the Tweedie models [21, 20]. For example, the optimal choice of the parameter for the normal distribution is $\beta = 1$, for the gamma distribution is $\beta \to -1$, for the Poisson distribution $\beta \to 0$, and for the compound Poisson $\beta \in (-1, 0)$.

From the beta generalized divergence we can derive various kinds of NMF algorithms: Multiplicative based on the standard gradient descent or the Exponentiated Gradient (EG) algorithms (see next section), additive algorithms using Projected Gradient (PG) or Interior Point Gradient (IPG), and Fixed Point (FP) algorithms.

In order to derive a flexible NMF learning algorithm, we compute the gradient of (21) with respect to elements of matrices $x_{jk} = x_j(k) = [\boldsymbol{X}]_{jk}$ and $a_{ij} = [\boldsymbol{A}]_{ij}$. as follows

$$\frac{\partial D_K^{(\beta)}}{\partial x_{jk}} = \sum_{i=1}^{m} a_{ij} \ \left([\boldsymbol{AX}]_{ik}^{\beta} - y_{ik} \ [\boldsymbol{AX}]_{ik}^{\beta-1} \right) + \alpha_X, \tag{23}$$

$$\frac{\partial D_K^{(\beta)}}{\partial a_{ij}} = \sum_{k=1}^{N} \left(\ [\boldsymbol{AX}]_{ik}^{\beta} \ - y_{ik} [\boldsymbol{AX}]_{ik}^{\beta-1} \right) \ x_{jk} + \alpha_A. \tag{24}$$

Similar to the Lee and Seung approach, by choosing suitable learning rates:

$$\eta_{jk} = \frac{x_{jk}}{\sum_{i=1}^{m} a_{ij} [\boldsymbol{AX}]_{ik}^{\beta}}, \quad \widetilde{\eta}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{N} [\boldsymbol{AX}]_{ik}^{\beta} \, x_{jk}}, \tag{25}$$

we obtain multiplicative update rules [10, 6]:

$$x_{jk} \leftarrow x_{jk} \frac{[\sum_{i=1}^{m} a_{ij} \ (y_{ik} / [\boldsymbol{AX}]_{ik}^{1-\beta}) - \alpha_X]_{\varepsilon}}{\sum_{i=1}^{m} a_{ij} \ [\boldsymbol{AX}]_{ik}^{\beta}}, \tag{26}$$

$$a_{ij} \leftarrow a_{ij} \ \frac{[\sum_{k=1}^{N} (y_{ik} / [\boldsymbol{AX}]_{ik}^{1-\beta}) \ y_{jk} - \alpha_A]_{\varepsilon}}{\sum_{k=1}^{N} [\boldsymbol{AX}]_{ik}^{\beta} \ x_{jk}}, \tag{27}$$

where again the rectification defined as $[x]_{\varepsilon} = \max\{\varepsilon, x\}$ with a small ε is introduced in order to avoid zero and negative values.

3 SMART Algorithms for NMF

There are two large classes of generalized divergences which can be potentially used for developing new flexible algorithms for NMF: the Bregman divergences

and the Csiszár's φ-divergences [22, 23, 24]. In this contribution we limit our discussion to the some generalized entropy divergences.

Let us consider at beginning the generalized K-L divergence dual to (18):

$$D_{KL}(\boldsymbol{AX}||\boldsymbol{Y}) = \sum_{ik} \left([\boldsymbol{AX}]_{ik} \ln \left(\frac{[\boldsymbol{AX}]_{ik}}{y_{ik}} \right) - [\boldsymbol{AX}]_{ik} + y_{ik} \right) \tag{28}$$

subject to nonnegativity constraints (see Eq. (18)) In order to derive the learning algorithm let us apply multiplicative exponentiated gradient (EG) descent updates to the loss function (28):

$$x_{jk} \leftarrow x_{jk} \exp\left(-\eta_{jk} \frac{\partial D_{KL}}{\partial x_{jk}} x_{jk} \right), \quad a_{ij} \leftarrow a_{ij} \exp\left(-\eta_{ij} \frac{\partial D_{KL}}{\partial a_{ij}} a_{ij} \right), \tag{29}$$

where

$$\frac{\partial D_{KL}}{\partial x_{jk}} = \sum_{i=1}^{m} (a_{ij} \ln[\boldsymbol{AX}]_{ik} - a_{ij} \ \ln y_{ik}) \tag{30}$$

$$\frac{\partial D_{KL}}{\partial a_{ij}} = \sum_{k=1}^{N} (x_{jk} \ln[\boldsymbol{AX}]_{ik} - x_{jk} \ \ln y_{ik}) \,. \tag{31}$$

Hence, we obtain the simple multiplicative learning rules:

$$x_{jk} \leftarrow x_{jk} \exp\left(\sum_{i=1}^{m} \overline{\eta}_{jk} a_{ij} \ln \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right) \right) = x_{jk} \prod_{i=1}^{m} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\overline{\eta}_{jk} a_{ij}} \tag{32}$$

$$a_{ij} \leftarrow a_{ij} \exp\left(\sum_{k=1}^{N} \widetilde{\eta}_{ij} x_{jk} \ln \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right) \right) = a_{ij} \prod_{k=1}^{N} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\widetilde{\eta}_{ij} x_{jk}} \tag{33}$$

The nonnegative learning rates $\overline{\eta}_{jk}$ and $\widetilde{\eta}_{ij}$ can take different forms. Typically, for simplicity and in order to guarantee stability of the algorithm we assume that $\overline{\eta}_{jk} = \overline{\eta}_j = \omega \left(\sum_{i=1}^{m} a_{ij} \right)^{-1}$, $\widetilde{\eta}_{ij} = \widetilde{\eta}_j = \omega \left(\sum_{k=1}^{N} x_{jk} \right)^{-1}$, where $\omega \in (0, 2)$ is an over-relaxation parameter. The EG updates can be further improved in terms of convergence, computational efficiency and numerical stability in several ways.

In order to keep weight magnitudes bounded, Kivinen and Warmuth proposed a variation of the EG method that applies a normalization step after each weight update. The normalization linearly rescales all weights so that they sum to a constant. Moreover, instead of the exponent function we can apply its re-linearizing approximation: $e^u \approx \max\{0.5, 1 + u\}$. To further accelerate its convergence, we may apply individual adaptive learning rates defined as $\eta_{jk} \leftarrow \eta_{jk} c$ if the corresponding gradient component $\partial D_{KL} / \partial x_{jk}$ has the same sign in two consecutive steps and $\eta_{jk} \leftarrow \eta_{jk} / c$ otherwise, where $c > 1$ (typically $c = 1.02 - 1.5$) [25].

The above multiplicative learning rules can be written in a more generalized and compact matrix form (using MATLAB notations):

$$\boldsymbol{X} \leftarrow \boldsymbol{X} \ .* \ \exp\left(\boldsymbol{\eta_X} \ .* \ (\boldsymbol{A}' * \ln(\boldsymbol{Y}./(\boldsymbol{A} * \boldsymbol{X} + \epsilon))) \right) \tag{34}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} \ .* \ \exp\left(\boldsymbol{\eta}_A \ .* \ (\ln(\boldsymbol{Y}./(\boldsymbol{A} * \boldsymbol{X} + \epsilon)) * \boldsymbol{X}') \right), \tag{35}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} * \mathrm{diag}\{1./sum(\boldsymbol{A}, 1)\}, \tag{36}$$

where in practice a small constant $\varepsilon = 10^{-16}$ is introduced in order to ensure positivity constraints and/or to avoid possible division by zero, and $\boldsymbol{\eta}_A$ and $\boldsymbol{\eta}_X$ are non-negative scaling matrices representing individual learning rates. The above algorithm may be considered as an alternating minimization/projection extension of the well known SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) [26, 27]. This means that the above NMF algorithm can be extended to MART and BI-MART (Block-Iterative Multiplicative Algebraic Reconstruction Technique) [26].

It should be noted that the parameters (weights) $\{x_{jk}, a_{ij}\}$ are restricted to positive values, the resulting updates rules can be written:

$$\ln(x_{jk}) \leftarrow \ln(x_{jk}) - \eta_{jk} \frac{\partial D_{KL}}{\partial \ln x_{jk}}, \quad \ln(a_{ij}) \leftarrow \ln(a_{ij}) - \eta_{ij} \frac{\partial D_{KL}}{\partial \ln a_{ij}}, \tag{37}$$

where the natural logarithm projection is applied component-wise. Thus, in a sense, the EG approach takes the same steps as the standard gradient descent (GD), but in the space of logarithm of the parameters. In other words, in our current application the scalings of the parameters $\{x_{jk}, a_{ij}\}$ are best adapted in log-space, where their gradients are much better behaved.

4 NMF Algorithms Using Amari α-Divergence

It is interesting to note, that the above SMART algorithm can be derived as a special case for a more general loss function called Amari α-divergence (see also Liese & Vajda, Cressie-Read disparity, Kompass generalized divergence and Eguchi-Minami beta divergence)[2] [29, 28, 23, 22, 10, 30]):

$$D_A(\boldsymbol{Y}||\boldsymbol{AX}) = \frac{1}{\alpha(\alpha-1)} \sum_{ik} \left(y_{ik}^{\alpha} z_{ik}^{1-\alpha} - \alpha y_{ik} + (\alpha-1) z_{ik} \right) \tag{38}$$

We note that as special cases of the Amari α-divergence for $\alpha = 2, 0.5, -1$, we obtain the Pearson's, Hellinger and Neyman's chi-square distances, respectively, while for the cases $\alpha = 1$ and $\alpha = 0$ the divergence has to be defined by the limits $\alpha \to 1$ and $\alpha \to 0$, respectively. When these limits are evaluated one obtains for $\alpha \to 1$ the generalized Kullback-Leibler divergence defined by equations (18) and for $\alpha \to 0$ the dual generalized KL divergence (28).

The gradient of the above cost function can be expressed in a compact form as

$$\frac{\partial D_A}{\partial x_{jk}} = \frac{1}{\alpha} \sum_{i=1}^{m} a_{ij} \left[1 - \left(\frac{y_{ik}}{z_{ik}} \right)^{\alpha} \right], \quad \frac{\partial D_A}{\partial a_{ij}} = \frac{1}{\alpha} \sum_{k=1}^{N} x_{jk} \left[1 - \left(\frac{y_{ik}}{z_{ik}} \right)^{\alpha} \right]. \tag{39}$$

[2] Note that this form of α–divergence differs slightly with the loss function of Amari given in 1985 and 2000 [28, 23] by the additional term. This term is needed to allow de-normalized variables, in the same way that extended Kullback-Leibler divergence differs from the standard form (without terms $z_{ik} - y_{ik}$) [24].

However, instead of applying the standard gradient descent we use the projected (linearly transformed) gradient approach (which can be considered as generalization of exponentiated gradient):

$$\Phi(x_{jk}) \leftarrow \Phi(x_{jk}) - \eta_{jk}\frac{\partial D_A}{\partial \Phi(x_{jk})}, \quad \Phi(a_{ij}) \leftarrow \Phi(a_{ij}) - \eta_{ij}\frac{\partial D_A}{\partial \Phi(a_{ij})}, \tag{40}$$

where $\Phi(x)$ is a suitable chosen function.

Hence, we have

$$x_{jk} \leftarrow \Phi^{-1}\left(\Phi(x_{jk}) - \eta_{jk}\frac{\partial D_A}{\partial \Phi(x_{jk})}\right), \tag{41}$$

$$a_{ij} \leftarrow \Phi^{-1}\left(\Phi(a_{ij}) - \eta_{ij}\frac{\partial D_A}{\partial \Phi(a_{ij})}\right). \tag{42}$$

It can be shown that such nonlinear scaling or transformation provides stable solution and the gradients are much better behaved in Φ space. In our case, we employ $\Phi(x) = x^{\alpha}$ and choose the learning rates as follows

$$\eta_{jk} = \alpha^2 \Phi(x_{jk})/(x_{jk}^{1-\alpha}\sum_{i=1}^{m} a_{ij}), \quad \eta_{ij} = \alpha^2 \Phi(a_{ij})/(a_{ij}^{1-\alpha}\sum_{k=1}^{N} x_{jk}), \tag{43}$$

which leads directly to the new learning algorithm [3]: (the rigorous convergence proof is omitted due to lack of space)

$$x_{jk} \leftarrow x_{jk}\left(\frac{\sum_{i=1}^{m} a_{ij}\,(y_{ik}/z_{ik})^{\alpha}}{\sum_{q=1}^{m} a_{qj}}\right)^{1/\alpha}, \; a_{ij} \leftarrow a_{ij}\left(\frac{\sum_{k=1}^{N}(y_{ik}/z_{ik})^{\alpha}\, x_{jk}}{\sum_{t=1}^{N} x_{jt}}\right)^{1/\alpha} \tag{44}$$

This algorithm can be implemented in similar compact matrix form using the MATLAB notations:

$$\boldsymbol{X} \leftarrow \boldsymbol{X} .* \left(\boldsymbol{A}' * ((\boldsymbol{Y} + \varepsilon)./\,(\boldsymbol{A} * \boldsymbol{X} + \varepsilon)).^{\alpha}\right).^{1/\alpha}, \tag{45}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} .* \left(((\boldsymbol{Y} + \varepsilon)./\,(\boldsymbol{A} * \boldsymbol{X} + \varepsilon)).^{\alpha} * \boldsymbol{X}'\right).^{1/\alpha}, \tag{46}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{A} * \mathrm{diag}\{1./sum(\boldsymbol{A}, 1)\}.$$

Alternatively, applying the EG approach, we can obtain the following multiplicative algorithm:

$$x_{jk} \leftarrow x_{jk}\exp\left\{\overline{\eta}_{jk}\sum_{i=1}^{m} a_{ij}\left[\left(\frac{y_{ik}}{z_{ik}}\right)^{\alpha} - 1\right]\right\}, \tag{47}$$

$$a_{ij} \leftarrow a_{ij}\exp\left\{\widetilde{\eta}_{ij}\sum_{k=1}^{N}\left[\left(\frac{y_{ik}}{z_{ik}}\right)^{\alpha} - 1\right] x_{jk}\right\}. \tag{48}$$

[3] For $\alpha = 0$ instead of $\Phi(x) = x^{\alpha}$ we have used $\Phi(x) = \ln(x)$.

5 Generalized SMART Algorithms

The main objective of this paper is to show that the learning algorithm (32) and (33) can be generalized to the following flexible algorithm:

$$x_{jk} \leftarrow x_{jk} \exp\left[\sum_{i=1}^{m} \overline{\eta}_{jk}\, a_{ij}\, \rho(y_{ik}, z_{ik})\right],\ a_{ij} \leftarrow a_{ij} \exp\left[\sum_{k=1}^{N} \widetilde{\eta}_{ij}\, x_{jk}\, \rho(y_{ik}, z_{ik})\right] \tag{49}$$

where the error functions defined as

$$\rho(y_{ik}, z_{ik}) = -\frac{\partial D(\boldsymbol{Y}||\boldsymbol{AX})}{\partial z_{ik}} \tag{50}$$

can take different forms depending on the chosen or designed loss (cost) function $D(\boldsymbol{Y}||\boldsymbol{AX})$ (see Table 1).

As an illustrative example let us consider the Bose-Einstein divergence:

$$BE_\alpha(\boldsymbol{Y}||\boldsymbol{AX}) = \sum_{ik} y_{ik} \ln\left(\frac{(1+\alpha)y_{ik}}{y_{ik}+\alpha z_{ik}}\right) + \alpha z_{ik} \ln\left(\frac{(1+\alpha)z_{ik}}{y_{ik}+\alpha z_{ik}}\right). \tag{51}$$

This loss function has many interesting properties:

1. $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = 0$ if $\boldsymbol{z} = \boldsymbol{y}$ almost everywhere.
2. $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = BE_{1/\alpha}(\boldsymbol{z}||\boldsymbol{y})$
3. For $\alpha = 1$, BE_α simplifies to the symmetric Jensen-Shannon divergence measure (see Table 1).
4. $\lim_{\alpha\to\infty} BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) = KL(\boldsymbol{y}||\boldsymbol{z})$ and for α sufficiently small $BE_\alpha(\boldsymbol{y}||\boldsymbol{z}) \approx KL(\boldsymbol{z}||\boldsymbol{y})$.

The gradient of the Bose-Einstein loss function in respect to z_{ik} can be expressed as

$$\frac{\partial BE_\alpha(\boldsymbol{Y}||\boldsymbol{AX})}{\partial z_{ik}} = -\alpha \ln\left(\frac{y_{ik}+\alpha z_{ik}}{(1+\alpha)z_{ik}}\right) \tag{52}$$

and in respect to x_{jk} and a_{ij} as

$$\frac{\partial BE_\alpha}{\partial x_{jk}} = -\sum_{i=1}^{m} a_{ij} \frac{\partial BE_\alpha}{\partial z_{ik}}, \qquad \frac{\partial BE_\alpha}{\partial a_{ij}} = -\sum_{k=1}^{N} x_{jk} \frac{\partial BE_\alpha}{\partial z_{ik}}. \tag{53}$$

Hence, applying the standard (un-normalized) EG approach (29) we obtain the learning rules (49) with the error function $\rho(y_{ik}, z_{ik}) = \alpha \ln((y_{ik}+\alpha z_{ik})/((1+\alpha)z_{ik}))$. It should be noted that the error function $\rho(y_{ik}, z_{ik}) = 0$ if and only if $y_{ik} = z_{ik}$.

6 Multi-layer NMF

In order to improve performance of the NMF, especially for ill-conditioned and badly scaled data and also to reduce risk to get stuck in local minima of non-convex minimization, we have developed a simple hierarchical and multi-stage

Table 1. Extended SMART NMF adaptive algorithms and corresponding loss functions

$a_{ij} \leftarrow a_{ij} \exp\left(\sum_{k=1}^{N} \widetilde{\eta}_{ij}\, x_{jk}\, \rho(y_{ik}, z_{ik})\right),$	$x_{jk} \leftarrow x_{jk} \exp\left(\sum_{i=1}^{m} \overline{\eta}_{jk}\, a_{ij}\, \rho(y_{ik}, z_{ik})\right)$
$a_j = \sum_{i=1}^{m} a_{ij} = 1, \ \forall j, \quad a_{ij} \geq 0$	$y_{ik} > 0, \ z_{ik} = [\boldsymbol{AX}]_{ik} > 0, \ x_{jk} \geq 0$
Minimization of loss function	Corresponding error function $\rho(y_{ik}, z_{ik})$
1. K-L I-divergence, $D_{KL}(\boldsymbol{AX}\|\|\boldsymbol{Y})$	
$\sum_{ik} \left(z_{ik} \ln \frac{z_{ik}}{y_{ik}} + y_{ik} - z_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{y_{ik}}{z_{ik}}\right)$
2. Relative A-G divergence $AG_r(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left((y_{ik} + z_{ik}) \ln\left(\frac{y_{ik} + z_{ik}}{2y_{ik}}\right) + y_{ik} - z_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right)$
3. Symmetric A-G divergence $AG(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$2\sum_{ik} \left(\frac{y_{ik} + z_{ik}}{2} \ln\left(\frac{y_{ik} + z_{ik}}{2\sqrt{y_{ik} z_{ik}}}\right) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{y_{ik} - z_{ik}}{2z_{ik}} + \ln\left(\frac{2\sqrt{y_{ik} z_{ik}}}{y_{ik} + z_{ik}}\right)$
4. Relative Jensen-Shannon divergence	
$\sum_{ik} \left(2y_{ik} \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right) + z_{ik} - y_{ik} \right)$	$\rho(y_{ik}, z_{ik}) = \frac{y_{ik} - z_{ik}}{y_{ik} + z_{ik}}$
5. Symmetric Jensen-Shannon divergence	
$\sum_{ik} y_{ik} \ln\left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right) + z_{ik} \ln\left(\frac{2z_{ik}}{y_{ik} + z_{ik}}\right)$	$\rho(y_{ik}, z_{ik}) = \ln\left(\frac{y_{ik} + z_{ik}}{2z_{ik}}\right)$
6. Bose-Einstein divergence $BE(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} y_{ik} \ln\left(\frac{(1+\alpha)y_{ik}}{y_{ik} + \alpha z_{ik}}\right) + \alpha z_{ik} \ln\left(\frac{(1+\alpha)z_{ik}}{y_{ik} + \alpha z_{ik}}\right)$	$\rho(y_{ik}, z_{ik}) = \alpha \ln\left(\frac{y_{ik} + \alpha z_{ik}}{(1+\alpha)z_{ik}}\right)$
7. J-divergence $D_J(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left(\frac{y_{ik} - z_{ik}}{2} \ln\left(\frac{y_{ik}}{z_{ik}}\right) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{1}{2} \ln\left(\frac{y_{ik}}{z_{ik}}\right) + \frac{y_{ik} - z_{ik}}{2z_{ik}}$
8. Triangular Discrimination $D_T(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\sum_{ik} \left\{ \frac{(y_{ik} - z_{ik})^2}{y_{ik} + z_{ik}} \right\}$	$\rho(y_{ik}, z_{ik}) = \left(\frac{2y_{ik}}{y_{ik} + z_{ik}}\right)^2 - 1$
9. Amari's α divergence $D_A(\boldsymbol{Y}\|\|\boldsymbol{AX})$	
$\frac{1}{\alpha(\alpha-1)} \sum_{ik} \left(y_{ik}^{\alpha} z_{ik}^{1-\alpha} - y_{ik} + (\alpha - 1)(z_{ik} - y_{ik}) \right)$	$\rho(y_{ik}, z_{ik}) = \frac{1}{\alpha} \left[\left(\frac{y_{ik}}{z_{ik}}\right)^{\alpha} - 1 \right]$

procedure in which we perform a sequential decomposition of nonnegative matrices as follows: In the first step, we perform the basic decomposition (factorization) $\boldsymbol{Y} = \boldsymbol{A}_1\boldsymbol{X}_1$ using any available NMF algorithm. In the second stage, the results obtained from the first stage are used to perform the similar decomposition: $\boldsymbol{X}_1 = \boldsymbol{A}_2\boldsymbol{X}_2$ using the same or different update rules, and so on. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrarily many times until some stopping criteria are satisfied. In each step, we usually obtain gradual improvements of the performance. Thus, our model has the form: $\boldsymbol{Y} = \boldsymbol{A}_1\boldsymbol{A}_2 \cdots \boldsymbol{A}_L\boldsymbol{X}_L$, with the basis nonnegative matrix defined as $\boldsymbol{A} = \boldsymbol{A}_1\boldsymbol{A}_2 \cdots \boldsymbol{A}_L$. Physically, this means that we build up a system that has many layers or cascade connections of L mixing subsystems. The key point in our novel approach is that the learning (update) process to find parameters of sub-matrices $\boldsymbol{X}_l$ and $\boldsymbol{A}_l$ is performed sequentially, i.e. layer by layer[4]. In each step or each layer, we can use the same cost (loss) functions, and consequently, the same learning (minimization) rules, or completely different cost functions and/or corresponding update rules. This can be expressed by the following procedure:

(**Multilayer NMF Algorithm**)

Set: $\boldsymbol{X}_0 = \boldsymbol{Y}$,

For $l = 1, 2, \ldots\ L,$ **do** :

Initialize randomly $\boldsymbol{A}_l^{(0)}$ and/or $\boldsymbol{X}_l^{(0)}$,

For $k = 1, 2, \ldots, K$, **do** :

$$\boldsymbol{X}_l^{(k)} = \arg\min_{\mathbf{X}_l \geq 0} \left\{ D_l \left(\boldsymbol{X}_{l-1} || \boldsymbol{A}_l^{(k-1)} \boldsymbol{X}_l \right) \right\},$$

$$\boldsymbol{A}_l^{(k)} = \arg\min_{\mathbf{A}_l \geq 0} \left\{ \widetilde{D}_l \left(\boldsymbol{X}_{l-1} || \boldsymbol{A}_l \boldsymbol{X}_l^{(k)} \right) \right\},$$

$$\boldsymbol{A}_l^{(k)} \leftarrow \left[\frac{a_{ij}}{\sum_{i=1}^{m} a_{ij}} \right]_l^{(k)},$$

End

$$\boldsymbol{X}_l = \boldsymbol{X}_l^{(K)}, \quad \boldsymbol{A}_l = \boldsymbol{A}_l^{(K)},$$

End

7 Simulation Results

All the NMF algorithms discussed in this paper (see Table 1) have been extensively tested for many difficult benchmarks for signals and images with various statistical distributions. Simulations results confirmed that the developed algorithms are stable, efficient and provide consistent results for a wide set of parameters. Due to the limit of space we give here only one illustrative example: The five (partially statistically dependent) nonnegative source signals shown in Fig.1 (a) have been mixed by randomly generated uniformly distributed nonnegative

[4] The multilayer system for NMF and BSS is subject of our patent pending in RIKEN BSI, March 2006.

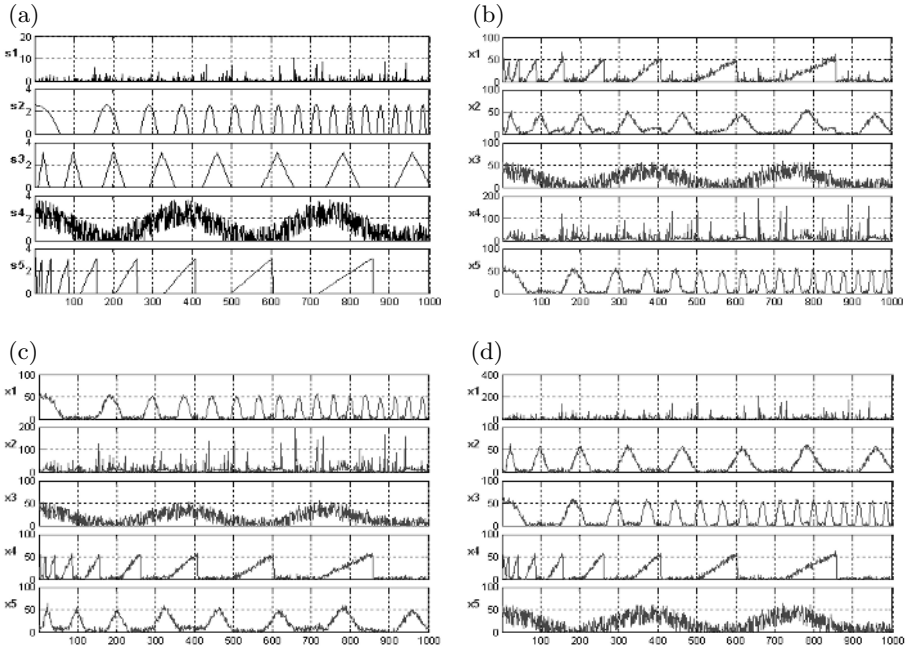


Fig. 1. Example 1: (a) The original 5 source signals; (b) Estimated sources using the standard Lee-Seung algorithm (8) and (9) with SIR = 8.8, 17.2, 8.7, 19.3, 12.4 [dB]; (c) Estimated sources using 20 layers applied to the standard Lee-Seung algorithm (8) and (9) with SIR = 9.3, 16.1, 9.9, 18.5, 15.8 [dB], respectively; (d) Estimated source signals using 20 layers and the new hybrid algorithm (15) with (49) with the Bose Shannon divergence with $\alpha = 2$; individual performance for estimated source signals: SIR = 15, 17.8, 16.5, 19, 17.5 [dB], respectively

matrix $\boldsymbol{A} \in \mathbb{R}^{50\times 5}$. To the mixing signals strong uniform distributed noise with SNR=10 dB has been added. Using the standard multiplicative NMF Lee-Sung algorithms we failed to estimate the original sources. The same algorithm with 20 layers of the multilayer system described above gives better results – see Fig.1 (c). However, even better performance for the multilayer system provides the hybrid SMART algorithm (49) with Bose-Einstein cost function (see Table 1) for estimation the matrix $\boldsymbol{X}$ and the Fixed Point algorithm (projected pseudo-inverse) (15) for estimation of the matrix $\boldsymbol{A}$ (see Fig.1 (d). We also tried to apply the ICA algorithms to solve the problem but due to partial dependence of the sources the performance was poor. The most important feature of our approach consists in applying multi-layer technique that reduces the risk of getting stuck in local minima, and hence, a considerable improvement in the performance of NMF algorithms, especially projected gradient algorithms.

8 Conclusions and Discussion

In this paper we considered a wide class of loss functions that allowed us to derive a family of robust and efficient novel NMF algorithms. The optimal choice of a loss function depends on the statistical distribution of the data and additive noise, so different criteria and algorithms (updating rules) should be applied for estimating the matrix $\boldsymbol{A}$ and the matrix $\boldsymbol{X}$ depending on *a priori* knowledge about the statistics of the data. We derived several multiplicative algorithms with improved performance for large scale problems. We found by extensive simulations that multilayer technique plays a key role in improving the performance of blind source separation when using the NMF approach.

References

[1] Lee, D.D., Seung, H.S.: Learning of the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791.

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

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

[4] Guillamet, D., Vitri'a, J., Schiele, B.: Introducing a weighted nonnegative matrix factorization for image classification. Pattern Recognition Letters **24** (2004) 2447 – 2454

[5] Li, H., Adali, T., W. Wang, D.E.: 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)

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

[7] Paatero, P., Tapper, U.: Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. Environmetrics **5** (1994) 111–126

[8] Oja, E., Plumbley, M.: Blind separation of positive sources using nonnegative PCA. In: 4th International Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan (2003)

[9] Hoyer, P.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457–1469.

[10] Kompass, R.: A generalized divergence measure for nonnegative matrix factorization, Neuroinfomatics Workshop, Torun, Poland (2005)

[11] Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: NIPS -Neural Information Proc. Systems, Vancouver Canada. (2005)

[12] Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (2006) http://www.wfu.edu/˜plemmons/papers.htm.

[13] Lee, D.D., Seung, H.S.: Algorithms for nonnegative matrix factorization. Volume 13. NIPS, MIT Press (2001)

[14] Novak, M., Mammone, R.: Use of nonnegative matrix factorization for language model adaptation in a lecture transcription task. In: Proceedings of the 2001 IEEE Conference on Acoustics, Speech and Signal Processing. Volume 1., Salt Lake City, UT (2001) 541–544
[15] Feng, T., Li, S.Z., Shum, H.Y., Zhang, H.: Local nonnegative matrix factorization as a visual representation. In: Proceedings of the 2nd International Conference on Development and Learning, Cambridge, MA (2002) 178–193
[16] Chen, Z., Cichocki, A., Rutkowski, T.: Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer's disease. In: IEEE International Conference on Acoustics, Speech, and Signal Processing,, ICASSP-2006, Toulouse, France (2006)
[17] Cichocki, A., Amari, S.: Adaptive Blind Signal And Image Processing (New revised and improved edition). John Wiley, New York (2003)
[18] Cichocki, A., Zdunek, R.: NMFLAB Toolboxes for Signal and Image Processing *www.bsp.brain.riken.go.jp*, JAPAN (2006)
[19] Merritt, M., Zhang, Y.: An interior-point gradient method for large-scale totally nonnegative least squares problems. Technical report, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, USA (2004)
[20] Minami, M., Eguchi, S.: Robust blind source separation by beta-divergence. Neural Computation **14** (2002) 1859–1886
[21] Jorgensen, B.: The Theory of Dispersion Models. Chapman and Hall (1997)
[22] Csiszár, I.: Information measures: A critical survey. In: Prague Conference on Information Theory, Academia Prague. Volume A. (1974) 73–86.
[23] Amari, S., Nagaoka, H.: Methods of Information Geometry. Oxford University Press, New York (2000)
[24] Zhang, J.: Divergence function, duality and convex analysis. Neural Computation **16** (2004) 159–195.
[25] Schraudolf, N.: Gradient-based manipulation of non-parametric entropy estimates. IEEE Trans. on Neural Networks **16** (2004) 159–195.
[26] Byrne, C.: Accelerating the EMML algorithm and related iterative algorithms by rescaled block-iterative (RBI) methods. IEEE Transactions on Image Processing **7** (1998) 100 – 109.
[27] Byrne, C.: Choosing parameters in block-iterative or ordered subset reconstruction algorithms. IEEE Transactions on Image Progressing **14** (2005) 321–327
[28] Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
[29] Amari, S.: Information geometry of the EM and em algorithms for neural networks. Neural Networks **8** (1995) 1379–1408.
[30] Cressie, N.A., Read, T.: Goodness-of-Fit Statistics for Discrete Multivariate Data. Springer, New York (1988)