SPLIT GRADIENT METHOD FOR NONNEGATIVE MATRIX FACTORIZATION

*Henri Lantéri *, Céline Theys *, Cédric Richard * and Cédric Févotte†*

* Laboratoire Fizeau, Université de Nice Sophia-Antipolis
Observatoire de la Côte d'Azur, UMR CNRS 6525, Parc Valrose 06108 Nice France
E-mail: Henri.Lanteri, Celine.Theys, Cedric.Richard@unice.fr
† Laboratoire Traitement et Communication de l'Information (LTCI)
Unité Mixte de Recherche CNRS - TELECOM ParisTech, E-mail: fevotte@telecom-paristech.fr

ABSTRACT

This article deals with an extension of the split gradient method (SGM) applied to the optimization of any divergence between two data fields, under positivity and flux conservation constraints. SGM is guaranteed to converge for convex cost functions. A SGM-based algorithm is also derived to solve the nonnegative matrix factorization (NMF) problem. It is shown that the multiplicative algorithms that are usually used for NMF, under positivity constraints, are particular cases of SGM. Finally, to validate the algorithm, we propose an example of application to hyperspectral data unmixing.

1. INTRODUCTION

In the field of image reconstruction or deconvolution, the minimization of a Euclidean distance or a Kullback-Leibler divergence between noisy measurements and a linear model is usually performed, subject to positivity constraints, using multiplicative algorithms. Most of time, the latter are the well known Iterative Space Reconstruction Algorithm (ISRA) [3], and the Expectation Minimization (EM) [4] or Richardson Lucy (RL) [13, 14] algorithm. In the last ten years, a general algorithmic method, called Split Gradient Method (SGM) [9, 10], has been developed to derive multiplicative algorithms for minimizing any convex criterion under positivity constraints. It leads to ISRA and EM-RL algorithm as particular cases. SGM has recently been extended to take into account a flux conservation constraint [11].

During the last few years, many papers have been published in the field of Nonnegative Matrix Factorisation (NMF) with multiplicative algorithms [2, 6, 12]. This problem is closely related to the blind deconvolution one [5, 8]. The aim of this paper is to propose a unified framework based on SGM to derive algorithms for NMF, in multiplicative form or not.

2. NONNEGATIVE MATRIX FACTORIZATION

We consider here the problem of nonnegative matrix factorization (NMF), which is now a popular dimension reduction technique, employed for non-subtractive, part-based representation of nonnegative data. Given a data matrix $\mathbf{V}$ of dimension $F \times N$ with nonnegative entries, the NMF consists of seeking a factorization of the form

$$\mathbf{V} \approx \mathbf{WH} \tag{1}$$

where $\mathbf{W}$ and $\mathbf{H}$ are nonnegative matrices of dimensions $F \times K$ and $K \times N$, respectively. Dimension K is usually chosen such that $FK + KN \ll FN$, hence reducing the data dimensionality. The factorization (1) is usually sought through the minimization problem

$$\min_{\mathbf{W},\mathbf{H}} \mathscr{D}(\mathbf{V}, \mathbf{WH}) \quad \text{s.t.} \quad [\mathbf{W}]_{ij} \geq 0, [\mathbf{H}]_{ij} \geq 0 \tag{2}$$

with $[\mathbf{V}]_{ij}$ and $[\mathbf{WH}]_{ij}$ the (i, j)-th entries of $\mathbf{V}$ and $\mathbf{WH}$, respectively. In the above expression, $\mathscr{D}(\mathbf{V}, \mathbf{WH})$ is a cost function defined by

$$\mathscr{D}(\mathbf{V}, \mathbf{WH}) = \sum_{ij} d([\mathbf{V}]_{ij}, [\mathbf{WH}]_{ij}) = \sum_{ij} d_{ij} \tag{3}$$

In the general case, $d(u, v)$ is a positive convex function that is equal to zero if $u = v$. An additional condition is the normalization of the columns of $\mathbf{W}$ and, as a direct consequence of (1), a constant-sum condition on the columns of $\mathbf{H}$. Minimization problem (2) becomes:

$$\min_{\mathbf{W},\mathbf{H}} \mathscr{D}(\mathbf{V}, \mathbf{WH}) \quad \text{s.t.} \quad [\mathbf{W}]_{ij} \geq 0, \quad [\mathbf{H}]_{ij} \geq 0,$$
$$\sum_i [\mathbf{W}]_{ij} = 1, \quad \sum_i [\mathbf{H}]_{ij} = \sum_i [\mathbf{V}]_{ij} \tag{4}$$

The constant-sum constraint is motivated by applications such as, for example, hyperspectral data unmixing. In this case, $\mathbf{W}$ is the matrix of basis spectra that are supposed to be normalized. To solve (2) and (4), one can use a minimization method of the SGM-type, alternatively on $\mathbf{W}$ and $\mathbf{H}$.

3. MINIMIZATION UNDER NON-NEGATIVITY CONSTRAINTS ONLY

SGM was initially formulated and developed to solve problem (2). Its Lagrangian function is given by:

$$\mathscr{L}(\mathbf{V}, \mathbf{WH}; \mathbf{\Lambda}, \mathbf{\Omega}) = \mathscr{D}(\mathbf{V}, \mathbf{WH}) - \langle \mathbf{\Lambda}, \mathbf{W} \rangle - \langle \mathbf{\Omega}, \mathbf{H} \rangle \tag{5}$$

where $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ are the matrices of positive Lagrange multipliers, and $\langle \cdot, \cdot \rangle$ is the inner product defined by:

$$\langle \mathbf{U}, \mathbf{V} \rangle = \sum_{ij} [\mathbf{U}]_{ij} [\mathbf{V}]_{ij} \tag{6}$$

The Karush-Kuhn-Tucker conditions must necessarily be satisfied at the optimum defined by $\mathbf{W}^*$, $\mathbf{H}^*$, $\mathbf{\Lambda}^*$, and $\mathbf{\Omega}^*$.

3.1 Minimization with respect to W

Minimization of (5) with respect to $\mathbf{W}$ leads to the following Karush-Kuhn-Tucker conditions for all i,j:

$$[\nabla_W \mathscr{L}(\mathbf{V},\mathbf{W}^*\mathbf{H};\mathbf{\Lambda}^*,\mathbf{\Omega})]_{ij}=0 \quad (7)$$
$$[\mathbf{\Lambda}^*]_{ij}\geq 0 \quad (8)$$
$$[\mathbf{W}^*]_{ij}\geq 0 \quad (9)$$
$$\langle \Lambda^*,\mathbf{W}^*\rangle = 0 \Leftrightarrow [\Lambda^*]_{ij}[\mathbf{W}^*]_{ij}=0 \quad (10)$$

Condition (7) immediately leads to

$$[\Lambda^*]_{ij}=[\nabla_W \mathscr{D}(\mathbf{V},\mathbf{W}^*\mathbf{H})]_{ij} \quad (11)$$

Condition (10) then becomes

$$\begin{aligned}&[\mathbf{W}^*]_{ij}[\nabla_W \mathscr{D}(\mathbf{V},\mathbf{W}^*\mathbf{H})]_{ij}=0\\ &\quad\Leftrightarrow [\mathbf{W}^*]_{ij}[-\nabla_W \mathscr{D}(\mathbf{V},\mathbf{W}^*\mathbf{H})]_{ij}=0\end{aligned} \quad (12)$$

where the extra minus sign in the last expression is just used to make a negative gradient descent direction of $\mathscr{D}(\mathbf{V},\mathbf{WH})$ apparent. To solve this equation iteratively, three points have to be noticed. The first one is that $\mathbf{M}\cdot(-\nabla_W\mathscr{D})$ is a gradient descent direction of $\mathscr{D}$ if $\mathbf{M}$ is a matrix with positive entries, where $\cdot$ denotes the Hadamard product. The second one is that $[-\nabla_W\mathscr{D}]_{ij}$ can always be decomposed as $[\mathbf{P}]_{ij}-[\mathbf{Q}]_{ij}$, where $[\mathbf{P}]_{ij}$ and $[\mathbf{Q}]_{ij}$ are positive entries, let us note that this decomposition is obviously not unique. Last but not least, the third one is that equations of the form $\varphi(\mathbf{W})=0$ can be solved with a fixed-point algorithm, under some conditions on function φ, by considering the problem $\mathbf{W}=\mathbf{W}+\varphi(\mathbf{W})$. Implementing this fixed-point strategy with equation (12) and using

$$[\mathbf{M}]_{ij}=\frac{1}{[\mathbf{Q}]_{ij}} \quad (13)$$

we obtain the following gradient-descent algorithm

$$[\mathbf{W}^{k+1}]_{ij}=[\mathbf{W}^k]_{ij}+\alpha_{ij}^k\frac{[\mathbf{W}^k]_{ij}}{[\mathbf{Q}]_{ij}^k}[-\nabla_W\mathscr{D}(\mathbf{V},\mathbf{W}^k\mathbf{H})]_{ij} \quad (14)$$

with α_{ij}^k a positive step size that allows to control convergence of the algorithm. Using the second point described above leads to

$$[\mathbf{W}^{k+1}]_{ij}=[\mathbf{W}^k]_{ij}+\alpha_{ij}^k\frac{[\mathbf{W}^k]_{ij}}{[\mathbf{Q}^k]_{ij}}\left([\mathbf{P}^k]_{ij}-[\mathbf{Q}^k]_{ij}\right) \quad (15)$$

that is,

$$[\mathbf{W}^{k+1}]_{ij}=[\mathbf{W}^k]_{ij}+\alpha_{ij}^k[\mathbf{W}^k]_{ij}\left(\frac{[\mathbf{P}^k]_{ij}}{[\mathbf{Q}^k]_{ij}}-1\right) \quad (16)$$

Let us determine the maximum value for the step size in order that $[\mathbf{W}^{k+1}]_{ij}\geq 0$, given $[\mathbf{W}^k]_{ij}\geq 0$. Note that, according to (15), a restriction may only apply if

$$[\mathbf{P}^k]_{ij}-[\mathbf{Q}^k]_{ij}<0 \quad (17)$$

since the other terms are positive. The maximum step size which ensures the positivity of $[\mathbf{W}^{k+1}]_{ij}$ is given by

$$(\alpha_{ij}^k)_{\max}=\frac{1}{1-\frac{[\mathbf{P}^k]_{ij}}{[\mathbf{Q}^k]_{ij}}} \quad (18)$$

which is strictly greater than 1. Finally, the maximum step size over all the components must satisfy

$$(\alpha^k)_{\max}\leq \min\{(\alpha_{ij}^k)_{\max}\} \quad (19)$$

This choice ensures the non-negativity of the components of $\mathbf{W}^k$ from iteration to iteration. Convergence of the algorithm is guaranteed by computing an appropriate step size, at each iteration, over the range $[0,(\alpha^k)_{\max}]$ by means of a simplified line search such as the Armijo rule for example. Finally, it is important to notice that the use of a step size equal to 1 leads to the very simple and well-known multiplicative form

$$[\mathbf{W}^{k+1}]_{ij}=[\mathbf{W}^k]_{ij}\frac{[\mathbf{P}^k]_{ij}}{[\mathbf{Q}^k]_{ij}} \quad (20)$$

Positiveness is satisfied if $[\mathbf{W}^0]_{ij}>0$, but convergence of the algorithm is not guaranteed.

3.2 Minimization with respect to H

Minimization with respect to $\mathbf{H}$ can be performed in the same way, using the decomposition

$$[-\nabla_H\mathscr{D}]_{ij}=[\mathbf{R}]_{ij}-[\mathbf{S}]_{ij} \quad (21)$$

where $[\mathbf{R}]_{ij}$ and $[\mathbf{S}]_{ij}$ are positive entries. The relaxed expression of the algorithm takes the form:

$$[\mathbf{H}^{k+1}]_{ij}=[\mathbf{H}^k]_{ij}+\beta_{ij}^k[\mathbf{H}^k]_{ij}\left(\frac{[\mathbf{R}^k]_{ij}}{[\mathbf{S}^k]_{ij}}-1\right) \quad (22)$$

Again, with a constant step size equal to 1, the algorithm takes a simple multiplicative form

$$[\mathbf{H}^{k+1}]_{ij}=[\mathbf{H}^k]_{ij}\frac{[\mathbf{R}^k]_{ij}}{[\mathbf{S}^k]_{ij}} \quad (23)$$

As previously, positiveness is satisfied if $[\mathbf{H}^0]_{ij}>0$ but convergence of the algorithm is not guaranteed.

Before ending this section, let us compute $\nabla\mathscr{D}$ with respect to $\mathbf{H}$ and $\mathbf{W}$. It can be expressed in matrix form as follows:

$$\nabla_H\mathscr{D}=\mathbf{W}^T\mathbf{A} \qquad \nabla_W\mathscr{D}=\mathbf{A}\mathbf{H}^T \quad (24)$$

where $\mathbf{A}$ is a matrix whose (i,j)-th entry is given by:

$$[\mathbf{A}]_{ij}=\frac{\partial d_{ij}}{\partial[\mathbf{WH}]_{ij}} \quad (25)$$

Equations (20) (23), associated to (24) (25), lead to the multiplicative algorithms described in [2, 6, 12]. These are particular cases of the relaxed algorithms (15) (22), obtained by using a unit step size.

4. MINIMIZATION UNDER NON-NEGATIVITY CONSTRAINTS AND FLUX CONSERVATION

Let us now consider problem (4), which differs from (2) by additional flux constraints. We make the following variable changes:

$$[\mathbf{W}]_{ij}=\frac{[\mathbf{Z}]_{ij}}{\sum_m[\mathbf{Z}]_{mj}}; \quad (26)$$

$$[\mathbf{H}]_{ij} = [\mathbf{T}]_{ij} \times \frac{\sum_m [\mathbf{V}]_{mj}}{\sum_m [\mathbf{T}]_{mj}} \tag{27}$$

In so doing, the problem becomes unconstrained with respect to the flux. To ensure that the problem remains convex w.r.t. the new variables, the solution is searched in a domain where the denominator is a constant, it is precisely what is performed by our method. To deal with the non-negativity constraints, let us consider again the SGM algorithm and compute gradient with respect to new variables:

$$\frac{\partial \mathscr{D}}{\partial [\mathbf{Z}]_{lj}} = \sum_i \frac{\partial \mathscr{D}}{\partial [\mathbf{W}]_{ij}} \times \frac{\partial [\mathbf{W}]_{ij}}{\partial [\mathbf{Z}]_{lj}} \tag{28}$$

$$\frac{\partial \mathscr{D}}{\partial [\mathbf{T}]_{lj}} = \sum_i \frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{ij}} \times \frac{\partial [\mathbf{H}]_{ij}}{\partial [\mathbf{T}]_{lj}} \tag{29}$$

where, in a compact form,

$$\frac{\partial [\mathbf{W}]_{ij}}{\partial [\mathbf{Z}]_{lj}} = \frac{1}{\sum_m [\mathbf{Z}]_{mj}} \times (\delta_{li} - [\mathbf{W}]_{ij}) \tag{30}$$

$$\frac{\partial [\mathbf{H}]_{ij}}{\partial [\mathbf{T}]_{lj}} = \frac{\sum_m [\mathbf{V}]_{mj}}{\sum_m [\mathbf{T}]_{mj}} \times (\delta_{li} - \frac{[\mathbf{H}]_{ij}}{\sum_m [\mathbf{V}]_{mj}}) \tag{31}$$

with δ_{li} the Kronecker symbol. As a consequence, the components of (the opposite of) the gradient of $\mathscr{D}$ with respect to the new variables can now be written as

$$-\frac{\partial \mathscr{D}}{\partial [\mathbf{Z}]_{lj}} = \frac{1}{\sum_m [\mathbf{Z}]_{mj}} \left(\left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{W}]_{lj}} \right) - \sum_i [\mathbf{W}]_{ij} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{W}]_{ij}} \right) \right) \tag{32}$$

and

$$-\frac{\partial \mathscr{D}}{\partial [\mathbf{T}]_{lj}} = \frac{\sum_m [\mathbf{V}]_{mj}}{\sum_m [\mathbf{T}]_{mj}} \left(\left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{lj}} \right) - \frac{\sum_i [\mathbf{H}]_{ij}}{\sum_m [\mathbf{V}]_{mj}} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{ij}} \right) \right) \tag{33}$$

It can be noticed that any shift of the form

$$\begin{array}{ccc} (-\partial \mathscr{D} / \partial [\mathbf{W}]_{ij})_s & \longleftarrow & (-\partial \mathscr{D} / \partial [\mathbf{W}]_{ij}) + \eta, \quad \forall (i,j) \\ (-\partial \mathscr{D} / \partial [\mathbf{H}]_{ij})_s & \longleftarrow & (-\partial \mathscr{D} / \partial [\mathbf{H}]_{ij}) + \mu, \quad \forall (i,j) \end{array}$$

leave equations (32) and (33) unchanged. Consequently, using

$$\eta = -\min_{ij} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{W}]_{ij}} \right) \quad \mu = -\min_{ij} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{ij}} \right)$$

does not modify the gradient of $\mathscr{D}$ with respect to the new variables $\mathbf{Z}$ and $\mathbf{T}$, but ensures the non-negativity of $(-\partial \mathscr{D} / \partial [\mathbf{W}]_{ij})_s$ and $(-\partial \mathscr{D} / \partial [\mathbf{H}]_{ij})_s$. Let us note that this particular decomposition allows to ensure that the denominator in 26 and 27 remains constant and then we are always in the convexity domain. We shall now apply the SGM method.

4.1 Minimization with respect to W

Consider the following gradient (32) decomposition

$$[-\nabla_Z \mathscr{D}]_{ij} = [\mathbf{P}]_{ij} - [\mathbf{Q}]_{ij} \tag{34}$$

that involves the non-negative entries defined as follows

$$[\mathbf{P}]_{ij} = \left(-\frac{\partial D}{\partial [\mathbf{W}]_{ij}} \right)_s \tag{35}$$

$$[\mathbf{Q}]_{ij} = [\mathbf{Q}]_{.j} = \sum_i [\mathbf{W}]_{ij} \left(-\frac{\partial D}{\partial [\mathbf{W}]_{ij}} \right)_s \tag{36}$$

The relaxed form of the minimization algorithm can be expressed as

$$[\mathbf{Z}^{k+1}]_{lj} = [\mathbf{Z}^k]_{lj} + \alpha^k [\mathbf{Z}^k]_{lj} \left(\frac{(-\partial \mathscr{D} / \partial [\mathbf{W}^k]_{lj})_s}{\sum_i [\mathbf{W}^k]_{ij} (-\partial \mathscr{D} / \partial [\mathbf{W}^k]_{ij})_s} - 1 \right)$$

We clearly have $\sum_l [\mathbf{Z}^{k+1}]_{lj} = \sum_l [\mathbf{Z}^k]_{lj}$, for all α^k. This allows to us to express the algorithm with respect to the initial variable $\mathbf{W}$, that is,

$$[\mathbf{W}^{k+1}]_{lj} = [\mathbf{W}]^k_{lj} + \alpha^k [\mathbf{W}]^k_{lj} \left(\frac{(-\partial \mathscr{D} / \partial [\mathbf{W}]^k_{lj})_s}{\sum_i [\mathbf{W}]^k_{ij} (-\partial \mathscr{D} / \partial [\mathbf{W}]^k_{ij})_s} - 1 \right)$$

Again, with a constant step size equal to 1, the algorithm takes a simple multiplicative form

$$[\mathbf{W}^{k+1}]_{lj} = [\mathbf{W}^k]_{lj} \frac{(-\partial \mathscr{D} / \partial [\mathbf{W}^k]_{lj})_s}{\sum_i [\mathbf{W}^k]_{ij} (-\partial \mathscr{D} / \partial [\mathbf{W}^k]_{ij})_s} \tag{37}$$

4.2 Minimization with respect to H

In an analogous way, consider the following gradient (33) decomposition

$$[-\nabla_T \mathscr{D}]_{ij} = [\mathbf{R}]_{ij} - [\mathbf{S}]_{ij} \tag{38}$$

that involves the non-negative entries given by

$$[\mathbf{R}]_{ij} = \frac{\sum_m [\mathbf{V}]_{mj}}{\sum_m [\mathbf{T}]_{mj}} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{ij}} \right)_s \tag{39}$$

$$[\mathbf{S}]_{ij} = S_{.j} = \frac{\sum_m [\mathbf{V}]_{m,j}}{\sum_m [\mathbf{T}]_{mj}} \sum_i \frac{[\mathbf{H}]_{ij}}{\sum_m [\mathbf{V}]_{mj}} \left(-\frac{\partial \mathscr{D}}{\partial [\mathbf{H}]_{ij}} \right)_s \tag{40}$$

This leads to the relaxed form of optimization algorithm with respect to variable $\mathbf{T}$, that is,

$$[\mathbf{T}^{k+1}]_{lj} = [\mathbf{T}^k]_{lj} + \alpha^k [\mathbf{T}^k]_{lj} \left(\frac{(-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{lj})_s}{\sum_i \frac{[\mathbf{H}^k]_{ij}}{\sum_m [\mathbf{V}]_{mj}} (-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{ij})_s} - 1 \right)$$

It can be seen that $\sum_l [\mathbf{T}^{k+1}]_{lj} = \sum_l [\mathbf{T}^k]_{lj}$, for all α^k, which implies that

$$[\mathbf{H}^{k+1}]_{lj} = [\mathbf{H}^k]_{lj} + \alpha^k [\mathbf{H}^k]_{lj} \left(\frac{(-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{lj})_s}{\sum_i \frac{[\mathbf{H}^k]_{ij}}{\sum_m [\mathbf{V}]_{mj}} (-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{ij})_s} - 1 \right)$$

The multiplicative form is obtained with a constant step size equal to 1, namely,

$$[\mathbf{H}^{k+1}]_{lj} = [\mathbf{H}^k]_{lj} \frac{(-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{lj})_s}{\sum_i [\mathbf{H}^k]_{ij} (-\partial \mathscr{D} / \partial [\mathbf{H}^k]_{ij})_s} \sum_m [\mathbf{V}]_{mj} \tag{41}$$

In the next section, we propose to illustrate this algorithm within the field of hyperspectral imaging.

5. CHOICE OF THE DESCENT STEP SIZE AND CONVERGENCE SPEED

On one hand, if the descent step size is fixed to one, there is no way to modify the convergence speed and the iterations number can be high, moreover, the convergence is not ensured but the algorithm takes a simple form. On the other hand, if the descent step size is searched by a simple rule, Armijo for example, the iterations number decreases but the duration of one iteration increases, from our experience, when the step size is computed, the overall gain is about ten or twenty percents and in this case the convergence is ensured.

6. SIMULATION RESULTS

Hyperspectral imaging has received considerable attention in the last few years. See for instance [1], [7] and references therein. It consists of data acquisition with high sensitivity and resolution in hundreds contiguous spectral bands, geo-referenced within the same coordinate system. With its ability to provide extremely detailed data regarding the spatial and spectral characteristics of a scene, this technology offers immense new possibilities in collecting and managing information for civilian and military application areas.

Each vector pixel of an hyperspectral image characterizes a local spectral signature. Usually, one consider that each vector pixel can be modeled accurately as a linear mixture of different pure spectral components, called endmembers. Referring to our notations, each column of $\mathbf{V}$ can thus be interpreted as a spectral signature obtained by linear mixing of the spectra of endmembers, i.e., the columns of $\mathbf{W}$. The problem is then to estimate the endmember spectra $\mathbf{W}$ and the abundance coefficients $\mathbf{H}$ from the spectral signatures $\mathbf{V}$.

Many simulations have been performed to validate the proposed algorithm, eqs. (37) and (41). The experiment presented in this paper corresponds to 10 linear mixtures of 3 endmembers, the length of each spectrum being 826. The three endmembers used in this example were extracted from the ENVI library [15] and correspond to the spectra of the construction concrete, green grass, and micaceous loam. Equations (37) and (41) were implemented in the case of a Frobenius norm $\mathscr{D}$. Fig. 2 shows the estimated endmembers (columns of $\mathbf{W}$), and their abundance coefficients (rows of $\mathbf{H}$) after 12000 iterations, and compared them with the true values. Note that the initial values for $\mathbf{W}$ and $\mathbf{H}$ were chosen to satisfy the constraints, i.e., positivity, sum to one of the columns of $\mathbf{W}$. Fig. 1 shows the behaviour of the criterion $\mathscr{D}$ as a function of the number of iterations, and the 10 reconstructed spectra in comparison with the true ones. We clearly see that the curves coincide almost perfectly. The normalization of the columns of matrix $\mathbf{W}$, as well as the flux conservation between $\mathbf{V}$ and $\mathbf{H}$, are satisfied at each iteration. Let us note that $\mathbf{H}$ and $\mathbf{W}$ could be estimated up to a permutation of the columns of $\mathbf{W}$, and to an analogous permutation of the rows of $\mathbf{H}$.

7. AKNOWLEDGEMENTS

This work is supported in part by project ANR-09-JCJC-0073-01 TANGERINE (Theory and applications of nonnegative matrix factorization) and by PPF ISSO (Plan Pluri Formations Image et Signal en Sciences de l'Observation).

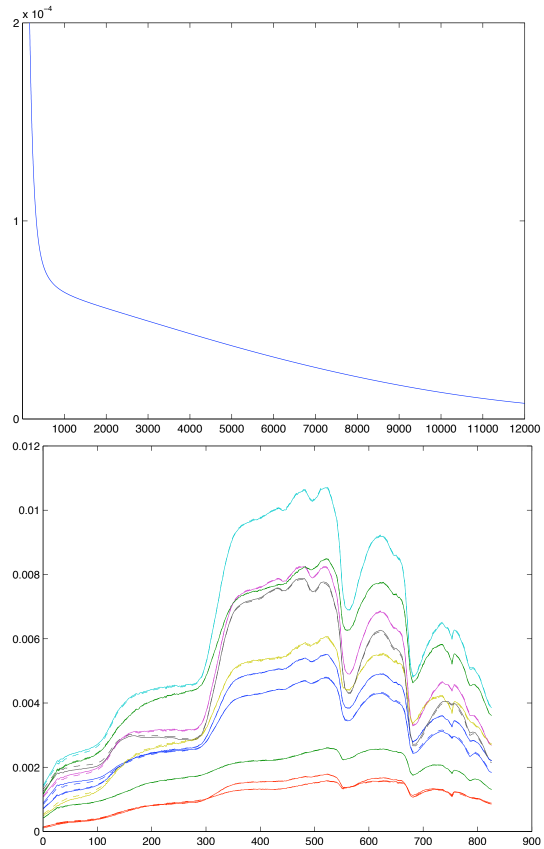

Figure 1: Frobenius $\mathscr{D}(\mathbf{V}, \mathbf{WH})$ as a function of the number of iterations. Columns of $\mathbf{V}$ at the end of the iterations, solid line for true values, dashed line for estimated values.

REFERENCES

[1] C. I. Chang. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*. Plenum Publishing Co., New York, 2003.

[2] A. Cichocki, R. Zdunek, and S. Amari. *Csiszár's Divergences for Non-negative Matrix Factorization: Family of New Algorithms*, volume 3889 of *Lectures Notes in Computer Science*. Springer Berlin/ Heidelberg, 2006.

[3] M. E. Daube-Witherspoon and G. Muehllehner. An iterative image space reconstruction algorithm suitable for volume ect. *IEEE Transactions on Medical Imaging*, (5):61–66, 1986.

[4] A. D. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the

EM algorithm. *Journal of the Royal statistic society*, (B 39):1–37, 1977.

[5] G. Desidera, B. Anconelli, M. Bertero, P. Boccacci, and M. Carbillet. Application of iterative blind deconvolution to the reconstruction of LBT LINC-NIRVANA images. *Astronomy and Astrophysics*, 452:727–753, 2006.

[6] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence with application to music analysis. *Neural Computation*, 2009.

[7] D. A. Landgrebe. *Signal Theory Methods in Multispectral Remote Sensing*. Wiley, New York, 2003.

[8] H. Lantéri, C. Aime, H. Beaumont, and P. Gaucherel. Blind deconvolution using the Richardson-Lucy algorithm. In *European Symposium on Satellite and Remote Sensing*, 1994.

[9] H. Lantéri, M. Roche, and C. Aime. Penalized maximum likelihood image restoration with positivity constraints- multiplicative algorithms. *Inverse problems*, 18:1397–1419, 2002.

[10] H. Lantéri, M. Roche, O. Cuevas, and C. Aime. A general method to devise maximum likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Processing*, 54(81):945–974, 2001.

[11] H. Lantéri, C. Theys, F. Benvenuto, and D. Mary. Méthode algorithmique de minimisation de fonctions d'écart entre champs de données. Application à la reconstruction d'images astrophysiques. In *GRETSI*, 2009.

[12] D. D. Lee and H. S. Seung. Algorithms for Non-Negative Matrix Factorization. In *Advances in NIPS*, volume 13, pages 556–562, 2001.

[13] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomic Journal*, (79):745–754, 1974.

[14] W. H. Richardson. Bayesian based iterative method of image restoration. *Journal of the optical society of America*, 1(62):55–59, 1972.

[15] RSI (Research Systems Inc.). *ENVI User's guide Version 4.0*. Boulder, CO 80301 USA, Sept. 2003.

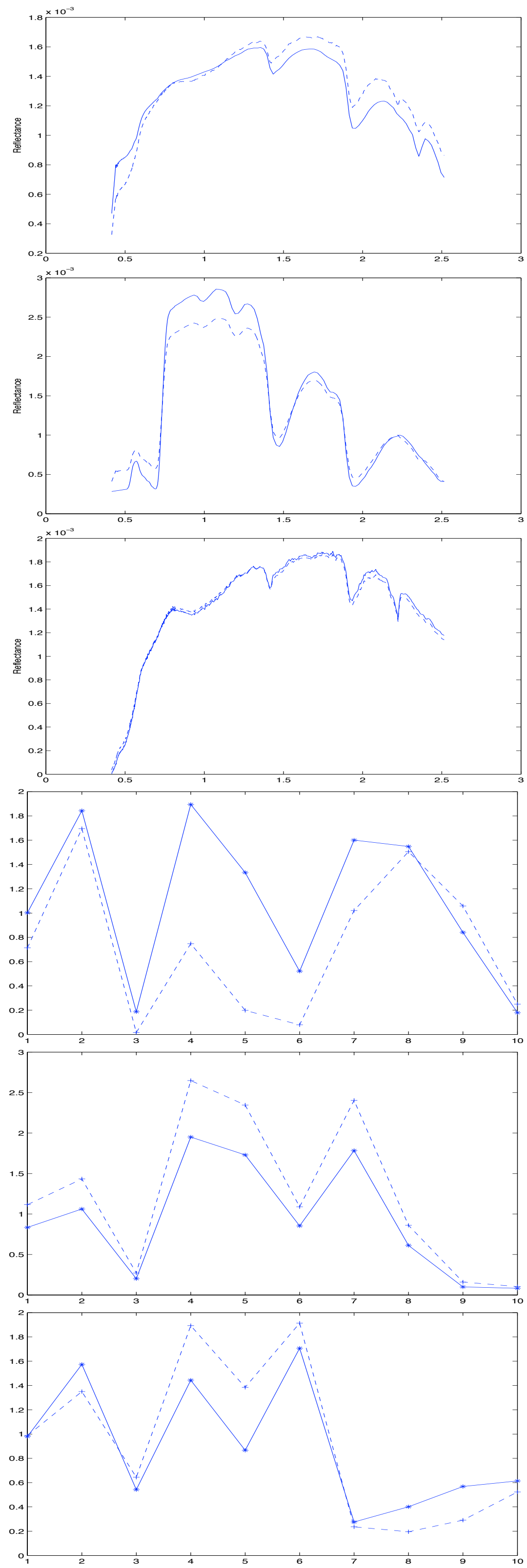

Figure 2: From top to bottom. Columns of **W** and rows of **H**. On each plot: solid line for true values, dashed line for estimated values.