A Gradient Based Method for Fully Constrained Least-Squares Unmixing of Hyperspectral Images

Jie Chen, Cédric Richard, Henri Lantéri, Céline Theys, Paul Honeine

A GRADIENT BASED METHOD FOR FULLY CONSTRAINED LEAST-SQUARES UNMIXING OF HYPERSPECTRAL IMAGES

Jie Chen *†*, Cédric Richard* †*, Henri Lantéri* †*, Céline Theys* †*, Paul Honeine**

* Institut Charles Delaunay, Université de Technologie de Troyes, CNRS, STMR, Troyes, France
†Université de Nice Sophia-Antipolis, CNRS, Observatoire de la Côte d'Azur, Nice, France

ABSTRACT

Linear unmixing of hyperspectral images is a popular approach to determine and quantify materials in sensed images. The linear unmixing problem is challenging because the abundances of materials to estimate have to satisfy non-negativity and full-additivity constraints. In this paper, we investigate an iterative algorithm that integrates these two requirements into the coefficient update process. The constraints are satisfied at each iteration without using any extra operations such as projections. Moreover, the mean transient behavior of the weights is analyzed analytically, which has never been seen for other algorithms in hyperspectral image unmixing. Simulation results illustrate the effectiveness of the proposed algorithm and the accuracy of the model.

Index Terms— Hyperspectral imagery, linear unmixing, estimation under constraints

1. INTRODUCTION

Hyperspectral imagery has been applied to a number of areas, including the environment, land use and agricultural monitoring. It can be used for more accurate and detailed information extraction than other types of remotely sensed data, as images provide rich spectral information to identify and distinguish between spectrally similar materials. This technology has received considerable attention.

In hyperspectral imagery, a pixel is a mixture of spectral components associated with a number of pure materials present in the scene. Although nonlinear mixture analysis has begun to find novel applications, linear spectral mixture analysis is still a widely studied approach to determine and quantify materials in sensed images, due to its simpler physical interpretation. In linear unmixing models, measured pixels can be decomposed as linear combinations of constituent spectra, the so-called endmembers, weighted by corresponding fractions named abundances. A key aspect in determining abundances is how to deal with the physical constraints, which are non-negativity and full-additivity. This problem is usually resolved by minimizing a cost function such as the least square criterion subject to these two constraints. In [1], the authors derive an iterative algorithm referred to as fully constrained least square algorithm, which adopts the sum-to-one constraint in the signature matrix and then performs the active set nonnegative least square algorithm of [2]. This method does not satisfy the full-additivity constraint exactly, and the active set method for non-negative least square includes substantial computational complexity. In [3], another class of algorithm is proposed where the full additivity constraint is ensured by projecting estimates at each iteration.

In this paper, we propose a gradient descent method which does not require any extra projection onto the space of constraints. It contains requirements for non-negativity and full-additivity into the coefficient update process. These two constraints are always fulfilled. Moreover, we derive an analytical analysis of the mean weight transient behavior of the proposed algorithm. To the best of our knowledge, this is unusual in the hyperspectral imaging community. Simulation results illustrate the effectiveness of the proposed algorithm and the accuracy of the model.

2. LINEAR MIXTURE MODEL

Linear spectral mixing model is largely used in hyperspectral imagery for determining and quantifying materials in sensed images. The physical assumption underlying this model is that each incident photon interacts with one Earth surface component only, and reflected spectra that are collected are not mixed with one another. Let L be the number of spectral bands, and $\boldsymbol{r} = [r_1, r_2, \ldots, r_L]^\top$ an L-by-1 column pixel of an hyperspectral image. Assume that $\boldsymbol{M}$ is the L-by-R target matrix of endmembers, denoted by $[\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_R]$, where each column $\boldsymbol{m}_i$ represents an endmember signature. Let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_R]^\top$ be a R-by-1 abundance column vector associated with the pixel $\boldsymbol{r}$. The linear mixing model is given by

$$\boldsymbol{r} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{n}$ is an additive white noise sequence. In order to estimate the abundance vector $\boldsymbol{\alpha}$, we consider here the least-squares cost function

$$J(\boldsymbol{\alpha}) = (\boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha})^\top(\boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha}) \tag{2}$$

To be physically meaningful, this model is subject to two constraints on $\boldsymbol{\alpha}$. The non-negativity constraint requires all the abundances to be nonnegative, that is, $\alpha_i \geq 0$ for all i. The full additivity constraint, also referred to as sum-to-one constraint, requires $\sum_{i=1}^{R} \alpha_i = 1$. The linear unmixing problem is cast as the following constrained optimization problem

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \tag{3}$$

$$\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, R \tag{4}$$

$$\sum_{i=1}^{R} \alpha_i = 1. \tag{5}$$

3. CONSTRAINED ALGORITHM WITH GRADIENT DESCENT METHOD

3.1. Solution with non-negativity constraints

In [4], we studied an algorithm for system identification involving only constraint (4), without the full-additivity constraint (5). For this problem, using Lagrange multiplier method, the Karush-Kuhn-Tucker conditions at the optimum were combined into the following expression

$$\alpha_i^o \, [-\nabla_\alpha J(\boldsymbol{\alpha})]_i = 0 \tag{6}$$

where the minus sign is just used to make a gradient descent of criterion $J(\boldsymbol{\alpha})$ apparent. Equations of the form $g(u) = 0$ can be solved with a fixed-point algorithm, under some conditions on function g, by considering the problem $u = u + g(u)$. Implementing this fixed-point strategy with (6) leads us to the component-wise gradient descent algorithm

$$\alpha_i(k+1) = \alpha_i(k) + \mu_i(k)\, \alpha_i(k)[-\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i \tag{7}$$

where $\mu_i(k) > 0$ is a positive step size that allows to control convergence. Suppose that $\alpha_i(k) > 0$. Non-negativity of $\alpha_i(k+1)$ is guaranteed if, and only if,

$$1 + \mu_i(k)\, [-\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i > 0 \tag{8}$$

If $[\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i < 0$, condition (8) is satisfied and non-negativity constraint does not impose any restriction on the step size. Conversely, if $[\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i > 0$, non-negativity of $\alpha_i(k+1)$ holds if

$$0 < \mu_i(k) \leq \frac{1}{[\nabla J(\boldsymbol{\alpha})]_i} \tag{9}$$

3.2. Algorithm for fully constrained problem

If non-negativity of the parameters is guaranteed at each iteration, we can make the following variable change to ensure that the sum-to-one constraint (5) is satisfied

$$\alpha_j = \frac{w_j}{\sum_{\ell=1}^{R} w_\ell} \tag{10}$$

With $w_i \geq 0$, the problem becomes unconstrained with respect to constraint (5). The partial derivative of the cost function J with respect to the new variables w_i can be expressed as follows

$$\frac{\partial J}{\partial w_i} = \sum_{j=1}^{R} \frac{\partial J}{\partial \alpha_j} \times \frac{\partial \alpha_j}{\partial w_i} \tag{11}$$

where

$$\begin{aligned}\frac{\partial \alpha_j}{\partial w_i} &= \frac{\frac{\partial w_j}{\partial w_i} \sum_{\ell=1}^{R} w_\ell - \frac{\partial \sum_{\ell=1}^{R} w_\ell}{\partial w_i} w_j}{\left(\sum_{\ell=1}^{R} w_\ell\right)^2} \\ &= \frac{\delta_{ij} - \alpha_j}{\sum_{\ell=1}^{R} w_\ell}\end{aligned} \tag{12}$$

The Kronecker symbol δ_{ij} results from the derivative $\partial w_j / \partial w_i$. Replacing (12) into (11), the negative partial derivative of J with respect to w_i can now be written as

$$-\frac{\partial J}{\partial w_i} = \frac{1}{\sum_{\ell=1}^{R} w_\ell} \left(-\frac{\partial J}{\partial \alpha_i} - \sum_{j=1}^{R} \alpha_j \left(-\frac{\partial J}{\partial \alpha_j} \right) \right) \tag{13}$$

Let us now use the same rule as (7) for updating the non-negative entries $w_i(k)$. The component-wise update equation is given by

$$\begin{aligned}w_i(k+1) &= w_i(k) + \ldots \\ &+ \mu \frac{w_i(k)}{\sum_{\ell=1}^{M} w_\ell(k)} \left(-\frac{\partial J}{\partial \alpha_i(k)} - \sum_{j=1}^{R} \alpha_j(k) \left(-\frac{\partial J}{\partial \alpha_j(k)} \right) \right)\end{aligned} \tag{14}$$

It can be easily found that $\sum_{i=1}^{R} w_i(k+1) = \sum_{i=1}^{R} w_i(k)$ for all μ. The factor $(\sum_{\ell=1}^{M} w_\ell(k))^{-1}$ is thus constant and can be absorbed into μ. This yields

$$\begin{aligned}w_i(k+1) &= w_i(k) + \ldots \\ &+ \mu w_i(k) \left(-\frac{\partial J}{\partial \alpha_i(k)} - \sum_{j=1}^{R} \alpha_j(k) \left(-\frac{\partial J}{\partial \alpha_j(k)} \right) \right)\end{aligned} \tag{15}$$

Dividing by $\sum_{i=1}^{R} w_i(k+1)$ and $\sum_{i=1}^{R} w_i(k)$ the left and right sides of (15), respectively, and considering the variable change defined by (10), we obtain

$$\begin{aligned}\alpha_i(k+1) &= \alpha_i(k) + \ldots \\ &+ \mu \alpha_i(k) \left(-\frac{\partial J}{\partial \alpha_i(k)} - \sum_{j=1}^{R} \alpha_j(k) \left(-\frac{\partial J}{\partial \alpha_j(k)} \right) \right)\end{aligned} \tag{16}$$

We can verify that $\sum_{i=1}^{R} \alpha_i(k+1) = \sum_{i=1}^{R} \alpha_i(k)$, which means that the algorithm satisfies the sum-to-one constraint as long as the weight vector is initialized by any vector $\boldsymbol{\alpha}(0)$ such that $\sum_{i=1}^{R} \alpha_i(0) = 1$.

The gradient of the least-squares criterion (2) is given by

$$\nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \boldsymbol{M}^\top \boldsymbol{r} - \boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{\alpha}. \tag{17}$$

Considering now the component-wise update equation (16), we get

$$\begin{aligned} \boldsymbol{\alpha}(k+1) &= \boldsymbol{\alpha}(k) + \dots \\ &\quad + \mu \,\text{diag}\{\boldsymbol{\alpha}(k)\} \left[\nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) - \mathbf{1} \nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha})^\top \boldsymbol{\alpha}(k)\right] \end{aligned} \tag{18}$$

where $\mathbf{1}$ is the all-one vector, and diag$\{\cdot\}$ a diagonal matrix

4. MEAN WEIGHT TRANSIENT BEHAVIOR

In this section, we are interested in studying the transient behavior of the iteration governed by (18). As it is well known, it is rather challenging to study the performance of such iterative system. Therefore several simplifying assumption will be adopted in the analysis process. However, experiments will show that the simulated and predicted performance match almost perfectly. Firstly, we rewrite the expression (18) as

$$\begin{aligned} \boldsymbol{\alpha}(k+1) &= \boldsymbol{\alpha}(k) + \dots \\ &\quad + \mu \,\text{diag}\{\boldsymbol{\alpha}(k)\}(\boldsymbol{M}^\top \boldsymbol{r} - \boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{\alpha}(k)) \\ &\quad + \mu \,\text{diag}\{\boldsymbol{\alpha}(k)\} \mathbf{1} (\boldsymbol{M}^\top \boldsymbol{r} - \boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{\alpha}(k))^\top \boldsymbol{\alpha}(k) \end{aligned} \tag{19}$$

Instead of studying $\boldsymbol{\alpha}(k)$ directly, we introduce the weight error vector

$$\boldsymbol{v}(k) = \boldsymbol{\alpha}(k) - \boldsymbol{\alpha}^* \tag{20}$$

with $\boldsymbol{\alpha}^* = \arg\min J(\boldsymbol{\alpha})$ the solution of the unconstrained problem. At each iteration, the residual error can be written as follows

$$\begin{aligned} \boldsymbol{e}(k) &= \boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha}(k) \\ &= \boldsymbol{n} - \boldsymbol{M}\boldsymbol{v}(k) \end{aligned} \tag{21}$$

Subtracting $\boldsymbol{\alpha}^*$ from both sides of (19) and considering the above expression, the following iterative equation with respect to the weight error vector $\boldsymbol{v}(k+1)$ is obtained after some mathematical calculation

$$\boldsymbol{v}(k+1) = \boldsymbol{v}(k) + \mu \boldsymbol{p}_1 - \mu \boldsymbol{p}_2 - \mu \boldsymbol{p}_3 + \mu \boldsymbol{p}_4 \tag{22}$$

where vectors $\boldsymbol{p}_1$ to $\boldsymbol{p}_4$ are defined by

$$\begin{aligned} \boldsymbol{p}_1 &= \text{diag}\{\boldsymbol{v}(k)\} \boldsymbol{M}^\top \boldsymbol{n} + \text{diag}\{\boldsymbol{\alpha}^*\} \boldsymbol{M}^\top \boldsymbol{n} \\ \boldsymbol{p}_2 &= \text{diag}\{\boldsymbol{v}(k)\} \boldsymbol{M}^\top \boldsymbol{M} \,\boldsymbol{v}(k) + \text{diag}\{\boldsymbol{\alpha}^*\} \boldsymbol{M}^\top \boldsymbol{M} \,\boldsymbol{v}(k) \\ \boldsymbol{p}_3 &= \boldsymbol{\alpha}^* \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \boldsymbol{n} + \boldsymbol{\alpha}^* \,\boldsymbol{v}(k)\, \boldsymbol{M}^\top \boldsymbol{n} + \boldsymbol{v}(k)\, \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \boldsymbol{n} \\ &\quad + \boldsymbol{v}(k)\, \boldsymbol{v}(k)^\top \boldsymbol{M}^\top \boldsymbol{n} \\ \boldsymbol{p}_4 &= \boldsymbol{\alpha}^* \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \boldsymbol{M} \,\boldsymbol{v}(k) + \boldsymbol{\alpha}^* \boldsymbol{v}^\top(k)\, \boldsymbol{M}^\top \boldsymbol{M} \,\boldsymbol{v}(k) \\ &\quad + \boldsymbol{v}(k)\, \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \,\boldsymbol{M}\boldsymbol{v}(k) + \boldsymbol{v}(k) \boldsymbol{v}^\top(k)\, \boldsymbol{M}^\top \boldsymbol{M} \boldsymbol{v}(k) \end{aligned}$$

Let us now analyze $\boldsymbol{v}(k)$ in terms of its expected value. Considering that $\boldsymbol{n}$ is zero-mean white noise, and using the independence of $\boldsymbol{v}(k)$ and $\boldsymbol{n}$, the expected values of $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$ are equal to 0. The expected values of $\boldsymbol{p}_2$ and $\boldsymbol{p}_4$ are given by

$$\begin{aligned} E[\boldsymbol{p}_2] &= \boldsymbol{M}^\top \boldsymbol{M} \text{diag}\{\boldsymbol{K}(k)\} + \text{diag}\{\boldsymbol{\alpha}^*\} \boldsymbol{M}^\top \boldsymbol{M} E[\boldsymbol{v}(k)] \\ E[\boldsymbol{p}_4] &= \boldsymbol{\alpha}^* \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \boldsymbol{M} \, E[\boldsymbol{v}(k)] + \boldsymbol{\alpha}^* \,\text{trace}\{\boldsymbol{K}(k)\, \boldsymbol{M}^\top \boldsymbol{M}\} \\ &\quad + \boldsymbol{K}(k)\, \boldsymbol{M}\, \boldsymbol{M}^\top \boldsymbol{\alpha}^* + \boldsymbol{K}(k)\, \boldsymbol{M}^\top \boldsymbol{M} E[\boldsymbol{v}(k)] \end{aligned}$$

where $\boldsymbol{K}(k) = E[\boldsymbol{v}(k)\, \boldsymbol{v}^\top(k)]$ is the covariance matrix of the weight error vector. Expected values $E[\boldsymbol{p}_2]$ and $E[\boldsymbol{p}_4]$ require second-order moments defined by $\mathbf{K}(n)$ in order to update $E[\boldsymbol{v}(k)]$. To simplify the analysis, we use the following separation approximation

$$\boldsymbol{K}(k) \approx E[\boldsymbol{v}(k)]\, E[\boldsymbol{v}^\top(k)]$$

This simplification allows us a more detailed study of the mean weight behavior using analytical methods. See [5] for a thorough discussion on this approximation. Extensive simulation results have shown that this approximation achieves adequate accuracy in modeling the mean behavior of the weights. Finally, we obtain the following update equation for the mean weight-error vector

$$\begin{aligned} E[\boldsymbol{v}(k+1)] &= E[\boldsymbol{v}(k)] + \dots \\ &\quad - \mu \big(\boldsymbol{M}^\top \boldsymbol{M} \,\text{diag}\{E[\boldsymbol{v}(k)]\, E[\boldsymbol{v}^\top(k)]\} \\ &\quad + \text{diag}\{\boldsymbol{\alpha}^*\} \boldsymbol{M}^\top \boldsymbol{M}\, E[\boldsymbol{v}(k)]\}\big) \\ &\quad + \mu (\boldsymbol{\alpha}^* \boldsymbol{\alpha}^{*\top} \boldsymbol{M}^\top \boldsymbol{M}\, E[\boldsymbol{v}(k)] \\ &\quad + \boldsymbol{\alpha}^* \text{trace}\{E[\boldsymbol{v}(k)]\, E[\boldsymbol{v}^\top(k)]\, \boldsymbol{M}^\top \boldsymbol{M}\} \\ &\quad + E[\boldsymbol{v}(k)]\, E[\boldsymbol{v}^\top(k)]\, \boldsymbol{M}\, \boldsymbol{M}^\top \boldsymbol{\alpha}^* \\ &\quad + E[\boldsymbol{v}(k)]\, E[\boldsymbol{v}^\top(k)]\, \boldsymbol{M}^\top \boldsymbol{M}\, E[\boldsymbol{v}(k)]\big) \end{aligned} \tag{23}$$

The behavior of the estimated abundance vector $\boldsymbol{\alpha}(k)$ is then described by $\boldsymbol{\alpha}(k) = \boldsymbol{\alpha}^* + \boldsymbol{v}(k)$. Simulations will be used to evaluate the model performance.

5. COMPUTER SIMULATIONS

5.1. Synthetic mixture of real spectra

The proposed learning algorithm was used for hyperspectral data unmixing, generated by linear combination of three pure materials with abundance $\boldsymbol{\alpha} = [0.3\; 0.5\; 0.2]$. These materials are grass, cedar and asphalt, with spectral signatures extracted from the USGC library. These spectra consist of 2151 bands covering wavelengths ranging from 0.35 to 2.5 μm. See Figure 1 (left). The data were corrupted by an additive Gaussian noise with SNR ≈ 20 dB. Figure 1 (middle and right) illustrates the convergence behavior of the algorithm for two different step sizes, averaged over 50 runs. The method clearly converges after several iterations. Note that the analytical

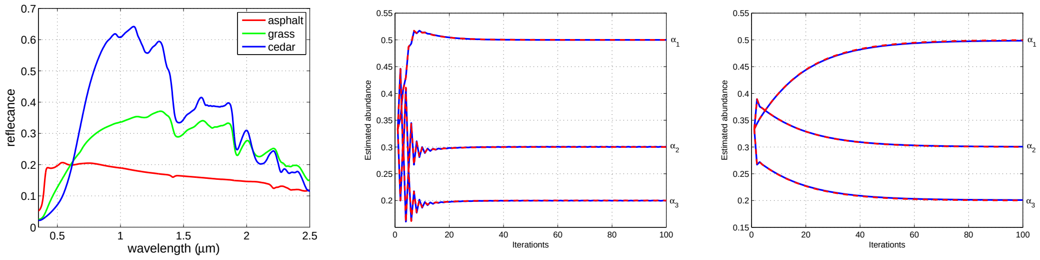

Fig. 1. Spectra of the three endmembers used for synthesizing data (left). Estimated abundances as a function of the iteration number, with two different step-sizes: $\mu = 0.1$ (middle), $\mu = 0.05$ (right). The red dashed curves represent the model output (23). The blue curves are the result of Monte-Carlo simulations.

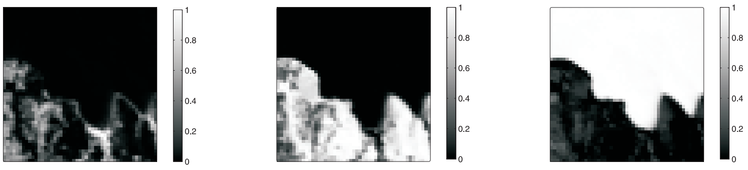

Fig. 2. Abundance maps estimated by the proposed algorithm (vegetation, soil and water, respectively)

model defined by equation (23) and simulated data match almost perfectly. These results clearly show that the derived model can be used for design purposes and, in particular, to derive a condition on the step-size for convergence.

For comparison, we generated a 50-by-50 hyperspectral image with abundance vectors $\boldsymbol{\alpha}_{ij}$ uniformly generated in the simplex defined by the positivity and sum-to-one constraints. The SNR of this image was set to 20 dB. We run the proposed algorithm, the fully-constrained algorithm described in [1], and the projected-gradient algorithm presented in [3]. The root mean square error

$$\text{RMSE} = \sqrt{\sum_{ij} \|\boldsymbol{\alpha}_{ij} - \hat{\boldsymbol{\alpha}}_{ij}\|^2 / NR}$$

was used to compare these algorithms, and the following results were obtained:

RMSE $_{\text{proposed}}$ = 0.0107
RMSE $_{[1]}$= 0.0103 (with $\delta = 0.1$)
RMSE $_{[1]}$= 0.0120 (with $\delta = 1$)
RMSE $_{[3]}$ = 0.0228

These results show the competitive performance of the presented algorithm compared with state-of-the-art methods.

5.2. Simulation on real data

The studied image is the scene over Moffett Field (CA, USA), captured by the airborne visible infrared imaging spectrometer (AVIRIS). A sub-image of size 50×50 pixels was chosen to evaluate the proposed algorithm. This scene is mainly composed of water, vegetation and soil. The endmembers were extracted by the VCA algorithm with $R = 3$. Figure 2 shows the estimated abundance maps. Note that several areas with dominant endmembers are clearly recovered.

6. CONCLUSION

This paper studied a new iterative algorithm for solving the problem of linear unmixing of hyperspectral images. Its transient mean weight behavior was analytically analyzed. Simulations conducted on synthetic data and real data have validated the algorithm and the proposed model. Future works will include detailed study of convergence, including convergence condition.

7. REFERENCES

[1] D. C. Heinz and C.-I Chang, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 39, pp. 529–545, 2001.

[2] C.L. Lawson and R.J. Hanson, *Solving least squares problems*, Society for Industrial Mathematics, 1995.

[3] C. Theys, N. Dobigeon, J.Y. Tourneret, and H. Lanteri, "Linear unmixing of hyperspectral images using a scaled gradient method," in *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE, 2009, pp. 729–732.

[4] J. Chen, C. Richard, P. Honeine, H. Lantéri, and C. Theys, "System identification under non-negativity constraints," in *Proc. of European Conference on Signal Processing*, Aalborg, Denmark, 2010.

[5] P.I. Hubscher and J.C.M. Bermudez, "An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm," *Signal Processing, IEEE Transactions on*, vol. 51, no. 3, pp. 664–671, 2003.