

LINEAR-QUADRATIC AND POLYNOMIAL NON-NEGATIVE MATRIX FACTORIZATION; APPLICATION TO SPECTRAL UNMIXING

Inès Meganem^{1,2}, Yannick Deville¹, Shahram Hosseini¹, Philippe Déliot², Xavier Briottet², Leonardo T. Duarte³

¹IRAP, CNRS, Université de Toulouse, UPS-OMP. 14, av. Edouard Belin, 31400 Toulouse, France.

²ONERA. 2, av. Edouard Belin 31055 Toulouse, France.

³School of Applied Sciences, UNICAMP, Rua Pedro Zaccaria, 1300, CEP:13484-350, Limeira-SP, Brazil

{ ines.meganem, yannick.deville, shahram.hosseini } @ast.obs-mip.fr
{ philippe.deliot, xavier.briottet } @onera.fr, leonardo.duarte@fca.unicamp.br

ABSTRACT

In this article, we present a source separation method for linear-quadratic models. This class of mixing models is encountered in various real applications, such as hyperspectral unmixing for urban environments. Linear-quadratic mixing models are less studied in the literature than linear ones but there exist some methods for handling them, essentially Bayesian or based on Independent Component Analysis.

We propose a separation method based on Non-negative Matrix Factorization (NMF). This class of methods is well-suited for many applications where data is positive and satisfies a linear model. The originality of our work is that we here developed an extension of NMF suited to linear-quadratic models. We also show how we can extend this method to the more general case of polynomial models.

Our method for linear-quadratic models is tested with mixtures of artificial signals and yields attractive performance. We also apply it to hyperspectral unmixing, by testing it with artificial mixtures of reflectance spectra, which gives encouraging results.

1. INTRODUCTION

Our work focuses on source separation for instantaneous linear-quadratic mixing models. This class of models has been less studied in the literature than linear models. There however exist some proposed methods, essentially based on Independent Component Analysis (ICA)[2][4] or Bayesian approaches [3].

The study of the linear-quadratic mixing model is interesting because it concerns various real applications, such as hyperspectral unmixing in remote sensing [6][7]. The most used model in hyperspectral unmixing is the linear one, but it is only valid when the surface is flat and homogeneous. In urban environments for example, the 3D structure induces multiple scattering of light between surfaces and this yields a linear-quadratic model when only taking into account second-order interactions. In this article we are eventually interested in this application of linear-quadratic models. In this case, we aim at unmixing the reflectance spectra associated with the different materials composing the pixels in an image.

We first propose, as a source separation method, an approach based on Non-negative Matrix Factorization (NMF) [1], that we develop specifically for the linear-quadratic model. NMF methods are interesting when one works with positive data. Besides, the use of these methods remains pos-

sible when data do not satisfy the source independence condition required in ICA methods. In the case of hyperspectral unmixing, for example, the reflectance spectra are often correlated. We also show how our method for linear-quadratic models can be extended to polynomial ones.

To assess the performance of our linear-quadratic method we first present tests made with artificial signals. We then provide an example in hyperspectral unmixing, by showing results obtained with mixtures of reflectance spectra.

In Section 2 we first present our NMF algorithm for linear-quadratic mixing models. Then we provide an extension to the case of polynomial models. In Section 3 we show results obtained with artificial signals, then for hyperspectral unmixing, in the case of linear-quadratic models.

2. THE METHOD

2.1 Adaptation of the Linear-Quadratic model to NMF

In this section we focus on linear-quadratic mixing models. At this stage we only present mixtures of two sources \mathbf{s}_1 and \mathbf{s}_2 . For a given sample n , the observation (mixture) \mathbf{x}_i reads

$$x_i(n) = a_1(i)s_1(n) + a_2(i)s_2(n) + a_{1,2}(i)s_1(n)s_2(n). \quad (1)$$

This yields in matrix form

$$\mathbf{X} = \mathbf{A}\mathbf{S} \quad (2)$$

where :

- $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_P]^T$, with P the number of observations ($P \geq 2$) and $\mathbf{x}_i = [x_i(1) \cdots x_i(K)]^T$, K the number of samples, indexed by n
- $\mathbf{S} = [\mathbf{s}_1 \ \mathbf{s}_2 \ \mathbf{s}_1 \odot \mathbf{s}_2]^T$, where $\mathbf{s}_j = [s_j(1) \cdots s_j(K)]^T$ (\odot stands for element-wise multiplication)
- $\mathbf{A} = \begin{bmatrix} a_1(1) & a_2(1) & a_{1,2}(1) \\ \vdots & \vdots & \vdots \\ a_1(P) & a_2(P) & a_{1,2}(P) \end{bmatrix}$ is the mixing matrix.

We assume that our data (sources and mixing coefficients) are positive. Considering that the third line in matrix \mathbf{S} corresponds to a pseudo-source equal to the product of the two real sources, Equation (2) will permit us to develop an NMF method, suited to linear-quadratic models. Hereafter, we will call this pseudo-source the "third source", but we have to keep in mind that our observations are actually mixtures of only two physical sources: \mathbf{s}_1 and \mathbf{s}_2 .

2.2 Our method: LQ-NMF

We here present our method developed for linear-quadratic mixtures, which is an extension of linear NMF methods.

Principle of NMF : Given a non-negative matrix V , Non-negative Matrix Factorization consists in finding non-negative matrix factors W and H that verify $V \approx WH$.

The way we wrote our mixtures in Equation (2) allows us to extend the principle of NMF methods, which is well-suited to linear mixtures, to the linear-quadratic model.

Different NMF methods have been proposed in the literature for linear mixtures (see [1]), based on different criteria, without or with constraints. They yield several updating rules for the matrix factors (they are iterative algorithms). We here chose the classical criterion of Frobenius norm (see Equation 3), with a simple gradient descent algorithm (see [1]). However, our mixing model implies new gradient calculations and new updating rules for the estimation of our factors A and S .

To make calculation easier we rewrite our matrices as follows

- $A = [A_a \ A_b]$, where $A_a = \begin{bmatrix} a_1(1) & a_2(1) \\ \vdots & \vdots \\ a_1(P) & a_2(P) \end{bmatrix}$
- and $A_b = \begin{bmatrix} a_{1,2}(1) \\ \vdots \\ a_{1,2}(P) \end{bmatrix}$
- $S = \begin{bmatrix} S_a \\ S_b \end{bmatrix}$, where $S_a = [s_1 \ s_2]^T$ and $S_b = [s_1 \odot s_2]^T$

Our criterion reads

$$J = \frac{1}{2} \|X - AS\|_F^2 = \frac{1}{2} \|X - A_a S_a - A_b S_b\|_F^2. \quad (3)$$

Our mixing model does not induce any modifications in gradient calculation with respect to the mixing matrix, since our criterion has the same dependence with respect to the mixing matrix A as in the linear case. Indeed, writing the criterion as follows

$$\begin{aligned} J &= \frac{1}{2} Tr((X - AS)(X - AS)^T) \\ &= \frac{1}{2} Tr(XX^T - XS^T A^T - ASX^T + ASS^T A^T) \end{aligned}$$

permits us to easily derive the gradient expression (see [8])

$$\begin{aligned} \frac{\partial J}{\partial A} &= \frac{1}{2} (-XS^T - XS^T + 2ASS^T) \\ &= (X - AS)S^T \end{aligned}$$

As stated above, we obtain the same result compared with the linear case. This yields the following update rule for the mixing matrix [1]

$$A^{(m+1)} = A^{(m)} - \alpha \frac{\partial J}{\partial A} \quad (4)$$

where α is a small positive learning rate.

However, the third source is the product of s_1 and s_2 and this changes gradient calculation with respect to the sources,

compared with the case of a linear model. We derive the gradient using scalar calculation, we thus first write our criterion as follows

$$\begin{aligned} J &= \frac{1}{2} \sum_{i,n} [X - A_a S_a - A_b S_b]_{in}^2 \\ &= \frac{1}{2} \sum_{i,n} [x_{in} - (A_a S_a)_{in} - (A_b S_b)_{in}]^2 \\ &= \frac{1}{2} \sum_{i,n} [x_{in} - \sum_{j=1}^2 a_j(i) S_{jn} - a_{1,2}(i) S_{1n} S_{2n}]^2 \end{aligned}$$

We now calculate the gradient of J with respect to S_{1n} (with $S_{jn} = [S]_{jn}$)

$$\begin{aligned} \frac{\partial J}{\partial S_{1n}} &= \sum_i (x_{in} - \sum_{j=1}^2 a_j(i) S_{jn} - a_{1,2}(i) S_{1n} S_{2n}) \\ &\quad \times (-a_1(i) - a_{1,2}(i) S_{2n}) \\ &= -\sum_i a_1(i) (x_{in} - \sum_{j=1}^2 a_j(i) S_{jn} - a_{1,2}(i) S_{1n} S_{2n}) \\ &\quad - S_{2n} \sum_i a_{1,2}(i) (x_{in} - \sum_{j=1}^2 a_j(i) S_{jn} - a_{1,2}(i) S_{1n} S_{2n}) \end{aligned} \quad (5)$$

The gradient with respect to S_{2n} can be obtained easily by permuting, for S_{jn} and $a_j(i)$, indices 1 and 2 in (5).

Replacing scalar expressions by matrix products, we finally obtain

$$\begin{aligned} \frac{\partial J}{\partial S_{1n}} &= -[A_a^T (X - AS)]_{1n} - [S_a]_{2n} \times [A_b^T (X - AS)]_n \\ \frac{\partial J}{\partial S_{2n}} &= -[A_a^T (X - AS)]_{2n} - [S_a]_{1n} \times [A_b^T (X - AS)]_n \end{aligned}$$

This yields the following updating rules for the $(m+1)^{th}$ iteration

$$S_{1n}^{(m+1)} = S_{1n}^{(m)} - \alpha \frac{\partial J}{\partial S_{1n}} \quad (6)$$

$$S_{2n}^{(m+1)} = S_{2n}^{(m)} - \alpha \frac{\partial J}{\partial S_{2n}} \quad (7)$$

$$S_{3n}^{(m+1)} = S_{1n}^{(m+1)} S_{2n}^{(m+1)} \quad (8)$$

As shown by (8), for the third line of the matrix S , we just copy the element-wise product of the first two lines (our two effective sources). We thus force the third source to be in keeping with the reality in our mixtures.

Concerning updating rules in Equations (4), (6) and (7), applying the gradient descent algorithm in this way is not sufficient because it does not guarantee positivity. To solve this problem we chose the solution proposed in the literature (see [1]) which consists in comparing, at each iteration, the calculated update with a very small positive value ε . We keep the maximum between them as follows

$$A^{(m+1)} = \max\{A^{(m+1)}, \varepsilon\} \quad (9)$$

$$S_a^{(m+1)} = \max\{S_a^{(m+1)}, \varepsilon\} \quad (10)$$

This must be understood as a comparison of each element of the matrix with ε , and we only replace the negative elements

by ε . This way, we eliminate all negative matrix entries and guarantee the positivity needed in an NMF algorithm.

It is well-known that NMF methods do not lead to a unique solution [5] and do not guarantee convergence to a global minimum [1]. Besides performance can depend on matrix initialisation. To tackle this problem, additional constraints can be added to the initial criteria to be minimised, such as sparsity or smoothness constraints. These constraints depend on the data properties in the considered application and can improve the algorithm performance.

We here decided to stay in a general framework, i.e. without constraints, which can then be easily adapted to particular applications and whose results can thus be improved by adding some constraints.

2.3 Extension to polynomial models

The above method can be extended to a much more general model which is the polynomial one. We here show how this may be achieved, thanks to the following simple case of mixtures

$$\begin{aligned} x_i(n) = & a_1(i)s_1(n) + a_2(i)s_2(n) + a_3(i)s_3(n) \\ & + a_{1,2}(i)s_1(n)s_2(n) + a_{1,3}(i)s_1(n)s_3(n) \quad (11) \\ & + a_{2,3}(i)s_2(n)s_3(n) + a_{1,2,3}(i)s_1(n)s_2(n)s_3(n) \end{aligned}$$

As in Section 2.1, we can write this model in matrix form to adapt it to NMF methods. We thus obtain the same matrix form as in Equation (2) but with the following new definitions for matrices \mathbf{A} and \mathbf{S} :

- $\mathbf{S} = \begin{bmatrix} \mathbf{S}_a \\ \mathbf{S}_b \\ \mathbf{S}_c \end{bmatrix}$ where
 - $\mathbf{S}_a = [s_1 \quad s_2 \quad s_3]^T$
 - $\mathbf{S}_b = [s_1 \odot s_2 \quad s_1 \odot s_3 \quad s_2 \odot s_3]^T$
 - $\mathbf{S}_c = [s_1 \odot s_2 \odot s_3]^T$
- $\mathbf{A} = [\mathbf{A}_a \quad \mathbf{A}_b \quad \mathbf{A}_c]$, where
 - $\mathbf{A}_a = \begin{bmatrix} a_1(1) & a_2(1) & a_3(1) \\ \vdots & \vdots & \vdots \\ a_1(P) & a_2(P) & a_3(P) \end{bmatrix}$
 - $\mathbf{A}_b = \begin{bmatrix} a_{1,2}(1) & a_{1,3}(1) & a_{2,3}(1) \\ \vdots & \vdots & \vdots \\ a_{1,2}(P) & a_{1,3}(P) & a_{2,3}(P) \end{bmatrix}$
 - $\mathbf{A}_c = \begin{bmatrix} a_{1,2,3}(1) \\ \vdots \\ a_{1,2,3}(P) \end{bmatrix}$

With this notation, the criterion in this case reads

$$J = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 = \frac{1}{2} \|\mathbf{X} - \mathbf{A}_a \mathbf{S}_a - \mathbf{A}_b \mathbf{S}_b - \mathbf{A}_c \mathbf{S}_c\|_F^2 \quad (12)$$

Due to space limitation we here do not present gradient calculations for this criterion, but we want to make it clear that our method, initially proposed for linear-quadratic mixing models, can be easily extended to polynomial models if there is a need.

3. TEST RESULTS

3.1 Tests with artificial signals

We first present some results obtained with mutually independent 100-sample i.i.d. artificial signals uniformly distributed over $[0 \ 1]$. We here propose to study performance depending on the closeness of the initialisation with respect to the true data present in the mixtures.

Hence we initialise our NMF algorithm with \mathbf{A}^0 and \mathbf{S}^0 derived from the real data \mathbf{A} and \mathbf{S} by adding some noise, to make initialisation different from the real data. Thus, the initial error in our tests is controlled by changing the noise power. This permits us to compare the final estimation error $error_f$ with the initial error $error_i$ (between the initial and real data), defined respectively as:

$$error_f = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_F}{\|\mathbf{Y}\|_F} \quad (13)$$

$$error_i = \frac{\|\mathbf{Y} - \mathbf{Y}^0\|_F}{\|\mathbf{Y}\|_F} \quad (14)$$

where \mathbf{Y} is the matrix for which we compute the error, among \mathbf{S}_a , \mathbf{A}_a and \mathbf{AS} . The notation \mathbf{Y}^0 stands for initialisation and $\widehat{\mathbf{Y}}$ for the final estimate.

Note that, before calculating the errors over the sources and linear mixing coefficients, the estimates are normalized so as to have the same scale as in the true matrices (by dividing each row of $\widehat{\mathbf{S}}_a$ containing an estimated source by its power and multiplying it by the corresponding real source's power, and we do the same for columns of $\widehat{\mathbf{A}}_a$). This permits us not to take into account the influence of scale factors, since they correspond to well-known indeterminacies of NMF.

We here present results obtained with two mixing matrices and for each one we study the evolution of the final error $error_f$ as a function of the initial one $error_i$. Note that those errors have normalised values, as we divide by the true matrix norm (see (13) and (14)).

This study is interesting because it shows the algorithm robustness when initialisation is very different from the real data, knowing that NMF performance often depends on initialisation. Moreover, in many applications we can have prior information about data or some library of similar signals, so initialisation can take advantage of that.

3.1.1 Case 1

We here present results obtained with the following mixing matrix

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0.5 & 0.4 \\ 0.6 & 1 & 0.5 \end{bmatrix}$$

We show the average estimation error obtained after running the algorithms for 10 couples of sources and each one for 10 different initialisations (so the average result over 100 runs).

Figure 1 shows the source estimation error. As we can see, results are good compared with the initial errors, even if \mathbf{S}^0 is very different from the real data \mathbf{S} (high values of $error_i$). The total error (Equations (13) and (14) applied to \mathbf{AS}) stays below 10^{-2} (Figure 2) and this shows that the algorithm converges well. However, concerning the estimation of the mixing coefficients, if we compare the initial error with

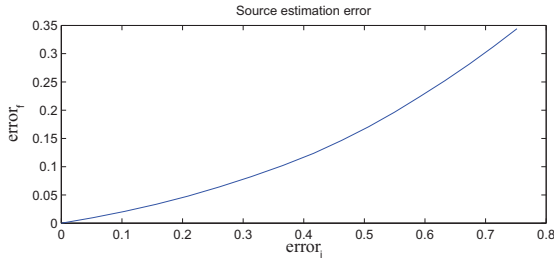


Figure 1: Case 1 - Source estimation error: $error_f$ vs $error_i$

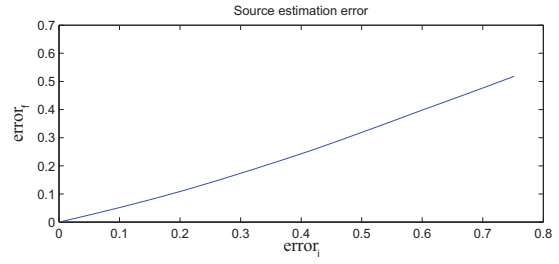


Figure 4: Case 2 - Source estimation error: $error_f$ vs $error_i$

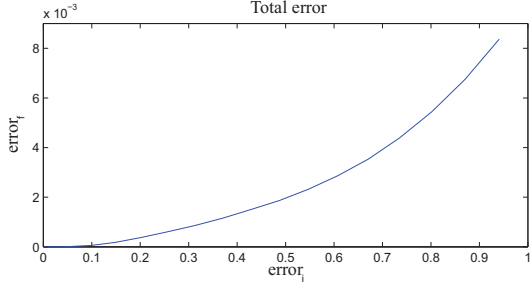


Figure 2: Case 1 - Total estimation error: $error_f$ vs $error_i$

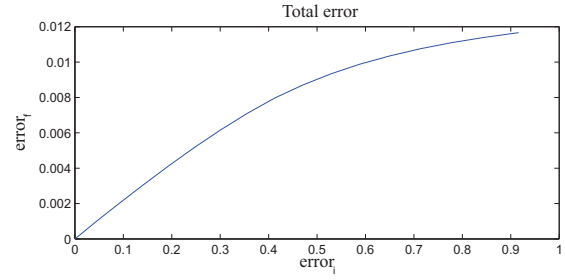


Figure 5: Case 2 - Total estimation error: $error_f$ vs $error_i$

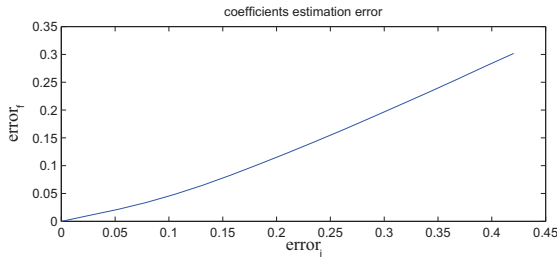


Figure 3: Case 1 - Mixing coefficient estimation error: $error_f$ vs $error_i$

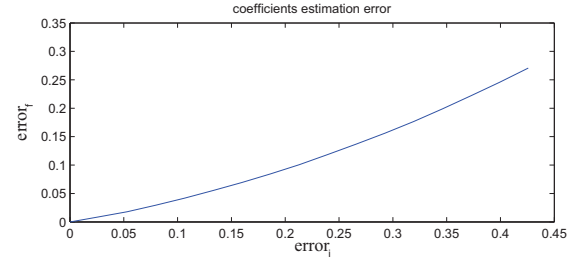


Figure 6: Case 2 - Mixing coefficient estimation error: $error_f$ vs $error_i$

the final one (Figure 3), we see that the improvement is less satisfying than in the case of the sources.

Let us now compare the value of the estimation errors in the three figures at one point in the curves. For high values of initial error, for example the last point in the curves, we notice that the total error is small, so the algorithm converges well. However, for this point, the estimation errors for sources and coefficients are significant (about 0.3). From these results, it becomes clear that the minimisation of the criterion does not necessarily leads to a perfect source separation and estimation of mixing coefficients. This can be explained by the non-uniqueness of the NMF solution, and, as already mentioned, can be improved by adding additional constraints to our algorithm.

3.1.2 Case 2

We now present results obtained with more highly mixed signals, obtained with the following matrix

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0.9 & 0.8 \\ 0.9 & 1 & 0.9 \end{bmatrix}$$

We again provide here the average result over 100 runs. As we can see in Figure 4, results with this matrix are not as

good as in the previous case, even if we obtain similar results for the total error (the algorithm converges well). These results may be due to the fact that the rows of matrix \mathbf{A}_2 are more similar than in case 1, so more correlated, and this has an effect on NMF performance. Similarly, we will see in the next section that NMF performance can be sensitive to correlation between sources, which form the other factor in the matrix product involved in (2).

3.2 Application to hyperspectral unmixing

Results presented here are related to hyperspectral unmixing in remote sensing. The principle is to extract the reflectance spectra corresponding to the different materials composing the pixels in a hyperspectral image. As stated above, in some cases, such as in urban environments, the mixing model faced in these images is linear-quadratic, due to multiple scattering of light between different surfaces. To be more precise, we here only consider double scattering and neglect the higher-order interactions.

Note that, if we take into account reflections with higher order, a polynomial model (like that presented in Section 2.1) can be considered. However, higher-order terms are likely to be negligible in most remote sensing applications [6].

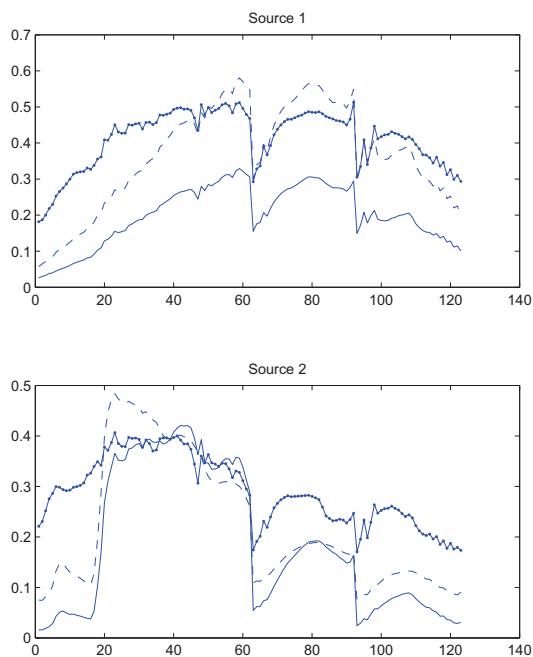


Figure 7: Spectra in mixtures (simple lines), their estimates (dashed lines) and the initialisations (thick dotted lines)

As data is positive in this case (reflectances and mixing coefficients), our NMF method is convenient. We here face a case in which a library of reflectance spectra of different material classes is available. We here show an example of results obtained with a couple of spectra mixed with the matrix \mathbf{A}_1 presented above.

We initialise \mathbf{S}^0 in our NMF algorithm with spectra belonging to a material class different from those of the sources. We thus address the situation when we do not know the materials present in our mixtures.

To initialise the mixing matrix, we apply a least square method with non-negativity constraint. Knowing \mathbf{S}^0 , we apply this method for each observation separately to estimate an initialisation \mathbf{A}^0 for the mixing matrix.

In Figure 7, we compare the real spectra present in our mixtures (simple lines) with their estimates (dashed lines) and the spectra used for initialisation (thick dotted lines). For this example we obtain a final estimation error of 0.19 for sources and 0.5 for the mixing coefficients, whereas the initial error is 0.37 for sources and is higher than 1 for coefficients. These errors are computed as in the previous section (i.e. using Eq. (13) and (14) with the same normalisation).

We have to stress that it does not matter here if we estimate spectra with a scale factor, because in most remote sensing applications, like classification, only the spectrum shape is important because it is specific to one material.

Note also that estimating reflectance spectra is a more complicated task, compared with the previous artificial signals, because spectra are correlated and even NMF methods are sensitive to this problem.

4. CONCLUSION AND FUTURE WORK

In this paper, we first proposed a source separation method based on Non-negative Matrix Factorization for linear-quadratic mixing models, by extending an approach which was previously reported for linear mixtures. We then showed how our work may be further extended to the more general case of polynomial models. As for results, we first presented examples with linear-quadratic mixtures of artificial signals. Then we showed how our method may be applied in hyperspectral unmixing where the linear-quadratic model is needed in some cases to express data.

This paper opens the way to several future investigations. Concerning the linear-quadratic model, we here presented our approach for only mixtures of two sources, so we intend to describe our method in more detail in the general case of N sources. It would be also interesting to detail the more general case of polynomial models.

In hyperspectral unmixing, we often face correlated sources, and this makes it more difficult to unmix the spectra. Therefore we need to study the behaviour of the proposed method in the case of correlated sources, to improve our algorithm and adapt it more to that case.

REFERENCES

- [1] A. Cichocki, R. Zdunek, A. H. Phan, S.-I. Amari. *Non-negative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, John Wiley and Sons, 2009.
- [2] S. Hosseini, Y. Deville, *Blind maximum likelihood separation of a linear-quadratic mixture*, Proceedings of the Fifth International Conference on Independent Component Analysis and Blind Signal Separation (ICA), Granada, Spain, 2004. *Erratum*: <http://arxiv.org/abs/1001.0863>.
- [3] L. T. Duarte, C. Jutten, S. Moussaoui, *Bayesian source separation of linear-quadratic and linear mixtures through a MCMC method*, Proceedings of IEEE MLSP, Grenoble, France, Sept. 2-4, 2009.
- [4] F. Mokhtari, M. Babaie-Zadeh, C. Jutten, *Blind separation of bilinear mixtures using mutual information minimization*, Proceedings of IEEE MLSP, Grenoble, France, Sept. 2-4, 2009.
- [5] S. Moussaoui, D. Brie, J. Idier, *Non-negative source separation: range of admissible solutions and conditions for the uniqueness of the solution*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2005, Vol.5, pp 289-292.
- [6] B. Somers, K. Cools, S. Delalieux, J. Stuckens, D. Van der Zande, W. W. Verstraeten, P. Coppin, *Nonlinear Hyperspectral Mixture Analysis for tree cover estimates in orchards*, Remote Sensing of Environment, vol. 113, no. 6, pp 1183 - 1193, 2009.
- [7] J. M. P. Nascimento, J. M. Bioucas-Dias. *Nonlinear Mixture Model for Hyperspectral Unmixing*, SPIE Conference on Image and Signal Processing for Remote Sensing XV 2009.
- [8] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, 2008, Technical University of Denmark, url: <http://matrixcookbook.com/>.