Hyperspectral unmixing with spectral variability: Reducing sensitivity to learning parameters, by combining linear and nonlinear NMF algorithms

Yannick Deville and Guillaume Faury

[a]*IRAP (Institut de Recherche en Astrophysique et Planétologie), Université de Toulouse, UPS, CNRS, CNES, OMP, 14 avenue Edouard Belin, Toulouse F-31400, France*

Abstract

Unsupervised hyperspectral unmixing methods aim at extracting the spectra of pure materials that are present in a hyperspectral image. A current major challenge consists of extending these methods to the case when each class of pure materials involves spectra that have a significant variability over image pixels (called spectral variability or intraclass variability). Unmixing may essentially be reformulated as a matrix factorization problem, especially with nonnegativity constraints. Therefore, an attractive class of methods to solve this problem is (constrained) Nonnegative Matrix Factorization (NMF), which is a type of unsupervised machine learning methods. Whereas NMF results provided in the literature are most often limited to a single value of a performance parameter (SAM of estimated spectra), in the present paper we first analyze the sensitivity of performance to the selected adaptation (i.e. learning) gain and number of learning iterations. Due to the quite significant sensitivity thus observed for basic algorithms, it is difficult to automatically select their values of the above parameters, and overfitting may occur during learning. We solve these problems by introducing "hybrid NMF methods". They essentially consist of successively running several types of NMF methods (respectively intended for linear and nonlinear mixtures), with different but compatible spaces for their adaptive variables, with different cost functions, and with part of the adaptive variables of each method initialized with the final values provided by the previously executed method for its corresponding variables. These hybrid methods have a low sensitivity to the adaptation gain and to the number of learning iterations, and they avoid overfitting over a large range of values of these parameters.

Preprint submitted to Digital Signal Processing *April 8, 2025*

Keywords: Hyperspectral unmixing, spectral variability, bilinear mixture, nonnegative matrix factorization (NMF), machine learning, sensitivity of adaptation, overfitting

1. Introduction

Hyperspectral sensors have a limited spatial resolution. Therefore, when observing the Earth, each pixel of a hyperspectral image corresponds to a surface on Earth which is often covered by different pure materials. The reflectance spectrum of such a pixel is then a mixture of the spectra of the corresponding pure materials. A major data processing task then consists of unmixing observed spectra, in order to retrieve pure material spectra from them. For a survey of these blind, i.e. unsupervised, unmixing methods, see Ref. [2] for instance.

The simplest case is faced when the considered scene consists of a flat landscape that receives a homogeneous illumination and when the sunlight is directly reflected from the ground to the airborne or satellite sensor (and no intimate mixing occurs). In that case, the mixing transform is defined as follows. Its input consists of the spectra of all M pure materials present in the scene, each of which is an L-element column vector $\mathbf{r_m}$, where L is the number of spectral bands. The mixing transform, that combines these pure spectra to form the spectra $\mathbf{x_p}$ of the P pixels of the recorded image, is linear (and memoryless) and reads

$$\mathbf{x_p} = \sum_{m=1}^{M} c_{p,m}\mathbf{r_m} \qquad \forall\, p \in \{1, \ldots, P\} \tag{1}$$

where each coefficient $c_{p,m}$ is the abundance (i.e. fraction of surface) of the pure material with index m in the pixel with index p. In matrix form, the mixing model (1) reads

$$\mathbf{X} = \mathbf{CR} \tag{2}$$

where

$$\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_P}]^T \tag{3}$$

and $.^T$ stands for transpose, whereas

$$\begin{aligned} \mathbf{R} &= [\mathbf{r_1}, \ldots, \mathbf{r_M}]^T && (4)\\ \mathbf{C} &= [\mathbf{c_1}, \ldots, \mathbf{c_P}]^T. && (5) \end{aligned}$$

For each pixel p,

$$\mathbf{c_p} = [c_{p,1}, \ldots, c_{p,M}]^T \tag{6}$$

is an M-element column vector containing the set of mixing coefficients, i.e. abundances, associated with that pixel.

Both the matrix $\mathbf{R}$ of pure material spectra and the matrix $\mathbf{C}$ of abundances are usually unknown. Moreover, all their elements are nonnegative. Hence, estimating both matrices from the recorded data matrix is essentially[1] an instance of the very generic Nonnegative (linear) Matrix Factorization, or NMF, problem (see Ref. [5]). At first sight, this might suggest that one could solve this unmixing problem by using any of the numerous NMF algorithms available from the literature (see the survey [5]), especially including Paatero's alternating least-square method (see Ref. [35]), Lin's gradient-based algorithm (see Ref. [27]) and Lee and Seung's multiplicative approach (see Ref. [25], [26]). However, that unmixing/NMF problem is in fact more complex because, without additional hypotheses, linear NMF methods suffer from a major problem, which is inherent to the hypotheses (i.e. weak constraints) made in the definition of linear NMF: they have many solutions, e.g. in terms of all positions of the global minima of their cost functions (see Ref. [14, 18, 16, 32]), not to speak of their local minima. Therefore, when initialized with arbitrary values, the two adaptive matrices that respectively aim at estimating $\mathbf{R}$ and $\mathbf{C}$ are likely to converge towards non-satisfactory values with this type of iterative algorithms (see e.g. Ref. [24, 20] about the initialization of NMF). In the present paper, we propose to solve this problem by taking advantage of recently reported results for *nonlinear* mixtures. Moreover, we extend the proposed approach so as to handle the more difficult case of Earth observation scenes that involve so-called "spectral variability", that is defined below. It should be noted that the connections between possibly-nonnegative matrix factorization, denoted as (N)MF, and compressed sensing and/or sparsity have been explored in the literature (see e.g. Ref. [15, 17, 34, 38, 39, 42]), but this is a different aspect of (N)MF, that is not investigated hereafter.

The remainder of this paper is organized as follows. The proposed general methodology is detailed in Section 2. The considered data and performance criterion are presented in Section 3. Several practical algorithms resulting

[1] Earth observation data moreover have a constraint, called the "sum-to-one" property: see Ref. [2].

from the proposed approach are defined in Section 4, where their performance is also reported and analyzed. Finally, conclusions are drawn from this investigation in Section 5.

In all this investigation, we put the emphasis on the sensitivity of each considered unmixing method to the values of its parameters (adaptation gain, number of iterations during learning). We aim at developing methods that have a low sensitivity to these parameters because, in contrast, when their sensitivity is high, it is difficult to automatically select the values of the above parameters, especially because their adequate values may depend on the considered data and poor performance is obtained when inadequate parameter values are chosen. Moreover, we aim at developing unmixing methods that avoid overfitting. Overfitting is a well-known problem in machine learning algorithms, that takes the following form for the considered unsupervised unmixing methods: during the adaptation of the matrices involved in these algorithms, the parameter that defines performance (namely the Spectral Angle Mapper, or SAM) first improves, then remains almost constant over a certain number of adaptation iterations, but it then starts to degrade, because the trained system becomes too specialized. In the subsequent sections, and especially in Section 4, we provide a much more detailed description of these sensitivity and overfitting problems, and of the algorithms that we propose to solve them.

2. Proposed methodology

2.1. Mixtures without spectral variability

Beyond the type of scenes considered in Section 1, in the framework of Earth observation there exist other types of scenes that have more complex structures and that are therefore represented by more complicated data models. This especially includes so-called bilinear mixtures, i.e. mixing models where, in addition to the above-mentioned linear combinations of pure material spectra, there exist combinations of element-wise products of two different pure spectra (see e.g. Ref. [30] for this model and the associated constraints), i.e.

$$\mathbf{x_p} = \sum_{m=1}^{M} c_{p,m}\mathbf{r_m} + \sum_{m=1}^{M-1}\sum_{\mu>m}^{M} c_{p,(m,\mu)}\mathbf{r_m} \odot \mathbf{r}_\mu, \qquad \forall\ p \in \{1,\dots,P\} \tag{7}$$

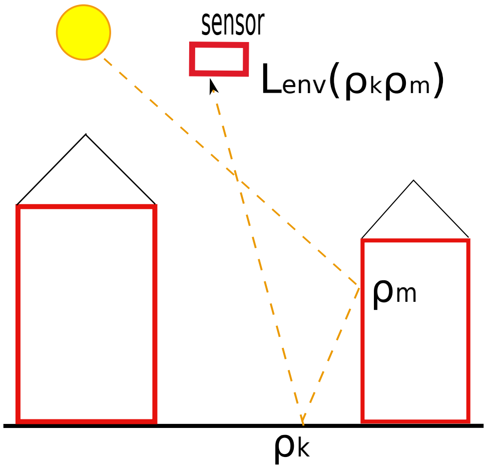

Figure 1: Light reflected twice in an urban scene (reprinted from Ref. [30]).

where $\odot$ stands for the element-wise product of vectors. As shown in Ref. [30], such mixtures e.g. occur in urban scenes where the light received by the satellite or airborne sensor consists of two types of contributions, namely: 1) light that first propagates from the sun to the Earth and that is then directly reflected by the Earth towards the sensor, as usual, and 2) light that first propagates from the sun to the Earth and that is then reflected twice before it reaches the sensor, e.g. because it is first reflected by a wall of a building to the ground and then from the ground to the sensor (see Fig. 1).

From a general Blind Source Separation (BSS) perspective (see e.g. Ref. [4, 6, 8, 11, 19, 29, 37]), it is natural to consider that the source signals involved in the mixing model (7) are the pure material spectra $\mathbf{r_m}$. That model is then nonlinear with respect to these sources (more precisely, it is second-order polynomial). However, remarkably, it can be reformulated as a *linear* model, with respect to so-called "extended sources", that consist of both the pure spectra $\mathbf{r_m}$ and their above element-wise products $\mathbf{r_m} \odot \mathbf{r}_\mu$ (see e.g. Ref. [11, 31]). More precisely, this model can be expressed in matrix form as

$$\mathbf{X} = \breve{\mathbf{C}}\breve{\mathbf{R}} \tag{8}$$

where

$$\breve{\mathbf{R}} = \begin{bmatrix} \mathbf{R} \\ \mathbf{R_b} \end{bmatrix} \tag{9}$$

with $\mathbf{R}$ still defined by (4) (linear part of the model) and where

$$\mathbf{R_b} = \left[\begin{array}{cccc} \mathbf{r_1} \odot \mathbf{r_2}, & \mathbf{r_1} \odot \mathbf{r_3}, & \cdots & \mathbf{r_{M-1}} \odot \mathbf{r_M} \end{array} \right]^T \tag{10}$$

contains the second-order terms $\mathbf{r_m} \odot \mathbf{r}_\mu$, with $\mu > m$, of the bilinear model (7), in the lexicographical order. Similarly,

$$\breve{\mathbf{C}} = \left[\begin{array}{cc} \mathbf{C} & \mathbf{C_b} \end{array} \right] \tag{11}$$

with $\mathbf{C}$ still defined by (5) and

$$\mathbf{C_b} = \left[\begin{array}{cccc} c_{1,(1,2)} & c_{1,(1,3)} & \cdots & c_{1,(M-1,M)} \\ \vdots & & & \vdots \\ c_{P,(1,2)} & c_{P,(1,3)} & \cdots & c_{P,(M-1,M)} \end{array} \right]. \tag{12}$$

Despite the initial nonlinear nature of the mixing model (7), the corresponding unmixing problem can thus be reformulated as an extended matrix factorization problem (8). More specifically, in the framework of Earth observation, all elements of $\breve{\mathbf{C}}$ and $\breve{\mathbf{R}}$ are nonnegative, so that (8) becomes an extended NMF problem. Quite interestingly, in Ref. [10] we showed that with only mild constraints, and without requesting nonnegativity, the bilinear matrix factorization problem (8) has no spurious solutions. This means that, given recorded data $\mathbf{X}$ that meet (8) and the associated constraints, the pure spectra involved in these data can be identified up to very limited indeterminacies, namely a permutation (and possibly scale factors, but they may be reduced by the additional properties available in the framework of Earth observation). These indeterminacies are also very common in the framework of linear instantaneous BSS. Beyond the data model, we also developed various bilinear (and linear-quadratic) extensions of NMF algorithms: see e.g. Ref. [1, 7, 10, 31].

Based on all above available works, a preliminary version of our approach in the present paper may be defined as follows. We address the case of linear mixtures, since this is the most common situation and it therefore remains of high interest. In this framework, we aim at improving the performance of unmixing methods by moreover taking advantage of the above-mentioned results that are available for the more advanced case of bilinear mixtures. This appears to be relevant because the bilinear mixing model (8) is a superset of the linear model (2). But, concerning what can be transposed from bilinear mixtures to linear ones, one should distinguish between identifiability properties (i.e. indeterminacies) and practical unmixing algorithms. The above-mentioned absence of spurious solutions that was mathematically proved in Ref. [10] holds for mixtures which do possess a bilinear part, so that the recorded mixed data span a part of space that has a related structure: see Theorem 1 and its assumptions, especially Assumption 2, in Ref. [10]. In contrast, for linear mixtures, that part of space becomes degenerated and the above theorem does not apply, which is consistent with the already-known intrinsic identifiability issues of linear mixtures. The results borrowed hereafter from the case of bilinear mixtures therefore do not concern identifiability properties but practical unmixing algorithms. More precisely, since linear mixtures are a subset of bilinear ones, the unmixing algorithms that were developed for the latter mixtures also apply to the former ones. Besides, not only these two classes of algorithms are based on different cost functions, but they estimate variables that are different, although they are compatible: algorithms for linear mixtures only estimate the matrices $\mathbf{C}$ and $\mathbf{R}$ of (2), whereas algorithms for bilinear mixtures estimate the matrices $\breve{\mathbf{C}}$ and $\breve{\mathbf{R}}$ of (8), which are respectively extended forms of $\mathbf{C}$ and $\mathbf{R}$, since they contain $\mathbf{C}$ and $\mathbf{R}$ as submatrices: see (9) and (11). When applied to mixtures that are actually linear, the above two classes of algorithms, initially designed respectively for linear and bilinear mixtures, may therefore be expected to have *different* spurious global or local minima and to make their adaptive matrices evolve along *different* trajectories. We here aim at taking advantage of this property. A first approach that we propose to this end consists of two stages. First, we perform a series of iterations with an algorithm from one of the above two classes, where we adapt the matrices that it involves, i.e. the adaptive counterpart of $\{\mathbf{C}, \mathbf{R}\}$ or of $\{\breve{\mathbf{C}}, \breve{\mathbf{R}}\}$. That algorithm may thus e.g. converge to a local minimum of its cost function. Starting from this point in terms of adaptive matrices, we then perform a series of iterations with an algorithm from the other class (switching to the other set of adaptive matrices, as detailed below). Since this starting point of the *second* algorithm is possibly not a local minimum of its cost function, unlike for the cost function of the *first* algorithm, that second algorithm may be expected to escape from that local minimum and to indeed further modify the matrices that it updates, and thus to yield improved estimates. Building upon this first idea, several scenarios for combining the above two classes of algorithms or their extensions are defined in more detail and tested in the subsequent sections.

2.2. Mixtures with spectral variability

The above approach handles the generic "linear + bilinear NMF" data processing problem and hence e.g. its application to the standard unmixing scenario in Earth observation, based on the mixing model (2). Moreover, if focusing on that field of Earth observation, a more complex unmixing scenario is also of importance, to handle the problem of spectral (or intraclass) variability, that may be defined as follows, first considering linear mixtures. In the above description, it was implicitly assumed that a given type of pure material is defined by a single spectrum so that, if the complete considered image involves M pure materials, then all observed pixel spectra are mixtures of the same M pure material spectra. However, things are more difficult in many practical remote sensing configurations, because what we called a pure material above yields somewhat different spectra in all observations, i.e. in all pixels. This phenomenon is called spectral variability, or inctraclass variability. For instance, if various pixels contain roof tiles, the associated tile spectra are not the same for all these pixels (see e.g. Ref. [9, 36, 13]). One should then not think anymore in terms of a pure material defined by a single spectrum, but in terms of a class of materials (e.g. the class of roof tiles in urban areas), whose spectra are somewhat different but still generally closer to one another than to the spectra of other classes of materials (e.g. the asphalt class if again considering urban areas). In other words, the classes have no or limited overlap (see an example in Fig. 5 of Ref. [36]). Starting from an image involving M classes of pure materials and containing P pixels, one may therefore aim at estimating MP pure spectra, that is, representatives of all classes in each pixel.

A linear mixing model that accounts for spectral variability may be expressed as follows (see e.g. Ref. [9, 36] for this model and the associated constraints), building upon the model (1):

$$\mathbf{x_p} = \sum_{m=1}^{M} c_{p,m} \mathbf{r_m}(p) \quad \forall\, p \in \{1, \ldots, P\} \tag{13}$$

where $\mathbf{r_m}(p)$ is the spectrum associated with the mth class of pure materials and pixel p.

This more complex model may also be expressed in a compact matrix form, that reads

$$\mathbf{X} = \tilde{\mathbf{C}} \tilde{\mathbf{R}} \tag{14}$$

when using the following notations:

$$\mathbf{R}(p) = [\mathbf{r_1}(p), \ldots, \mathbf{r_M}(p)]^T \tag{15}$$

contains the M pure material spectra associated with pixel p. The matrix

$$\tilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R}(1) \\ \ldots \\ \mathbf{R}(P) \end{bmatrix} \tag{16}$$

gathers all above pure spectra for all pixels of the considered scene. $\tilde{\mathbf{C}}$ is a block-diagonal matrix that gathers all mixing coefficients (abundances) of the scene:

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{c_1}^T & 0\ldots0 & \ldots & 0\ldots0 \\ 0\ldots0 & \mathbf{c_2}^T & \ldots & 0\ldots0 \\ & & \ddots & \\ 0\ldots0 & 0\ldots0 & \ldots & \mathbf{c_P}^T \end{bmatrix} \tag{17}$$

where we again use (6).

Various unmixing methods were proposed in the literature to address mixing models that include spectral variability: see e.g. the surveys [43, 3]. In particular, a method that is of importance for the present paper (because of its nonlinear extension presented below) is the IP-NMF method that we proposed in Ref. [9, 36]. Briefly, this method derives estimates of the extended forms of the spectra and abundance matrices, namely $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{C}}$, that appear in the model (14). This is achieved by minimizing the cost function

$$J_{ip-nmf} = \frac{1}{2}\left\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{R}}\right\|_F^2 + \mu \sum_{m=1}^{M} Tr(Cov(\tilde{\mathbf{R}}_{\mathcal{C}_\mathbf{m}})) \tag{18}$$

that uses the following notations. Here, $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{C}}$ are in fact the *adaptive matrices* associated with the actual data present in the model (14), that are estimated by the IP-NMF method. Moreover, $\left\|.\right\|_F$ stands for the Frobenius norm of a matrix. The first term of (18), that is $\left\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{R}}\right\|_F^2$, is the extension, to the variability-based model (14), of the "quality of fit" parameter usually employed in NMF algorithms intended for the standard mixing model (2). The second term of (18) is a regularization term equal, up to the weightening factor μ, to the sum of terms $Tr(Cov(\tilde{\mathbf{R}}_{\mathcal{C}_\mathbf{m}})$ which measure, respectively for each class m of pure materials, the spread of the pure material spectra estimated for that class and all pixels of the scene (see details in Ref. [9, 36]). Increasing μ therefore allows one to limit the variability of the estimated pure material spectra. The proposed IP-NMF algorithm minimizes the cost function (18) by using an approach that includes a projected gradient descent for updating the adaptive version of $\tilde{\mathbf{R}}$ (more precisely, we use its IP-NMF-FCLSU version: see details in Ref. [9, 36]).

This approach therefore has the advantage of being highly versatile: for each class of pure materials with index m, all the estimates of the pure spectra $\mathbf{r_m}(p)$ in (13), respectively associated with all pixels p, may first be initialized independently, as desired, and then also evolve quite "freely" (as opposed to methods that would e.g. use a spatial model for them, with the drawback that one should have prior knowledge about a *relevant* spatial model): they are updated independently, however with the constraint that the above-mentioned second term $\mu \sum_{m=1}^{M} Tr(Cov(\tilde{\mathbf{R}}_{\mathcal{C}_\mathbf{m}}))$ of the cost function (18) forces them to have a limited spread, separately within each class m, so that they still form a class. Moreover, as stated above, the strength of this constraint can be freely adjusted by the user, by tuning the value of the parameter μ.

What is of major importance for the investigation reported here is that, in Ref. [13], we very recently introduced a bilinear extension of the mixing model (14) (and the importance of bilinear models, even without variability, was explained above in Section 2.1) and of the IP-NMF algorithm for identifying that model. More precisely, that extension also applies to linear-quadratic (LQ) mixtures, i.e. mixtures which also contain each element-wise product of a pure material spectrum by itself: $\mathbf{r_m}(p) \odot \mathbf{r_m}(p)$. Our general LQ unmixing method is therefore called LQIP-NMF. To our knowledge, this is the first reported unsupervised unmixing method that jointly handles spectral variability and LQ or bilinear mixing. A few partly related methods exist (see [21, 22, 41, 40]), but 1) they are restricted to supervised configurations, and/or 2) they consist of several successive stages, unlike our joint approach, so that they may yield a lower estimation accuracy, and/or 3) they use other types of nonlinear mixing models, whereas the importance of the bilinear model was shown above. Hereafter, we consider the bilinear version of our LQIP-NMF approach. The corresponding mixing model reads as follows (see

Ref [13] for the associated constraints):

$$\mathbf{x_p} \; = \; \sum_{m=1}^{M} c_{p,m}\mathbf{r_m}(p) + \sum_{m=1}^{M-1}\sum_{\mu>m}^{M} c_{p,(m,\mu)}\mathbf{r_m}(p) \odot \mathbf{r}_\mu(p), \qquad \forall\, p \in \{1,\ldots,P\}. \tag{19}$$

Remarkably, even that extended model, that includes both spectral variability and bilinear nonlinearity, can be reformulated with a matrix-product form, as follows. As a first step, we still distinguish between the first-order (i.e. linear) terms of the original mixing model and the second-order ones (i.e. with spectra products). This yields a first matrix-form model for the complete observed data matrix $\mathbf{X}$ composed of the vectors $\mathbf{x_p}$ for all pixels, rearranged as in (3). That model reads

$$\mathbf{X} = \tilde{\mathbf{C}}\tilde{\mathbf{R}} + \tilde{\mathbf{\Gamma}}\tilde{\mathbf{T}} \tag{20}$$

with the following notations:

- $\tilde{\mathbf{R}}$ is again defined by (15)-(16).
- $\tilde{\mathbf{C}}$ is again defined by (6) and (17).
- $\mathbf{T}(p) = [\mathbf{r_1}(p) \odot \mathbf{r_2}(p), \mathbf{r_1}(p) \odot \mathbf{r_3}(p), \ldots, \mathbf{r_{M-1}}(p) \odot \mathbf{r_M}(p)]^T$ contains all element-wise products of pure material spectra (involved in second-order components) for pixel p, in the same order for all pixels.
- The matrix

  $$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{T}(1) \\ \vdots \\ \mathbf{T}(P) \end{bmatrix} \tag{21}$$

  contains the element-wise products of pure material spectra for all pixels of the observed image.
- $\gamma_\mathbf{p} = [c_{p,(1,2)}, c_{p,(1,3)}, \ldots, c_{p,(M-1,M)}]^T$ is the vector of second-order mixing coefficients associated with $\mathbf{T}(p)$.

- $\tilde{\mathbf{\Gamma}}$ is the resulting block-diagonal matrix of second-order mixing coefficients associated with $\tilde{\mathbf{T}}$ (i.e. with the complete image), defined as

$$\tilde{\mathbf{\Gamma}} = \begin{bmatrix} \gamma_{\mathbf{1}}{}^T & 0\ldots0 & \ldots & 0\ldots0 \\ 0\ldots0 & \gamma_{\mathbf{2}}{}^T & \ldots & 0\ldots0 \\ & & \ddots & \\ 0\ldots0 & 0\ldots0 & \ldots & \gamma_{\mathbf{P}}{}^T \end{bmatrix}.$$

Finally, this data model may be expressed in a more compact form, as

$$\mathbf{X} = \dot{\mathbf{C}}\dot{\mathbf{R}} \tag{22}$$

with the following notations:

- The matrix

$$\dot{\mathbf{R}} = \begin{bmatrix} \mathbf{R}(1) \\ \mathbf{T}(1) \\ \vdots \\ \mathbf{R}(P) \\ \mathbf{T}(P) \end{bmatrix} \tag{23}$$

contains the “extended spectra”, i.e. the pure material spectra (here again defined by (15)) and their element-wise products, for all the observed image.

- Besides, $\dot{\mathbf{C}}$ is the block-diagonal matrix of first-order and second-order mixing coefficients associated with $\dot{\mathbf{R}}$, defined as

$$\dot{\mathbf{C}} = \begin{bmatrix} [\mathbf{c_1}{}^T, \gamma_{\mathbf{1}}{}^T] & 0\ldots0 & \ldots & 0\ldots0 \\ 0\ldots0 & [\mathbf{c_2}{}^T, \gamma_{\mathbf{2}}{}^T] & \ldots & 0\ldots0 \\ & & \ddots & \\ 0\ldots0 & 0\ldots0 & \ldots & [\mathbf{c_P}{}^T, \gamma_{\mathbf{P}}{}^T] \end{bmatrix}. \tag{24}$$

$\dot{\mathbf{C}}$ is block-diagonal because each row of this matrix corresponds to one pixel and the (first-order and second-order) mixing coefficients associated with that pixel are non-zero only for the pure material spectra (or element-wise products of such spectra) associated with *that* pixel, as shown by (19). Physically, the first-order coefficients in a given vector among $\mathbf{c_1}^T, \mathbf{c_2}^T, \ldots$ represent the fractions of surface respectively associated with each class of pure materials in the considered pixel. Each second-order coefficient, corresponding to the indices m and μ, in a given vector among $\gamma_{\mathbf{1}}{}^T, \gamma_{\mathbf{2}}{}^T, \ldots$ defines the strength of the double light reflection over the two pure materials that have the pure spectra $\mathbf{r_m}(p)$ and $\mathbf{r}_\mu(p)$ involved in the term $\mathbf{r_m}(p) \odot \mathbf{r}_\mu(p)$ of (19).

As stated above, in Ref. [13] we developed the LQIP-NMF unmixing method to identify the above mixing model (22). Briefly, this method derives estimates of the extended spectra and mixing coefficient matrices $\dot{\mathbf{R}}$ and $\dot{\mathbf{C}}$ of the model (22) by minimizing a cost function $J_{lqip-nmf}$ that is derived from (18) by replacing its contribution $\left\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{R}}\right\|_F^2$ with its extended form $\left\|\mathbf{X} - \dot{\mathbf{C}}\dot{\mathbf{R}}\right\|_F^2$ (where $\dot{\mathbf{R}}$ and $\dot{\mathbf{C}}$ are in fact the *adaptive matrices* associated with the actual data present in the model (22)). The proposed LQIP-NMF algorithm minimizes this extended cost function by using an approach that includes a projected gradient descent for updating the adaptive version of $\dot{\mathbf{R}}$ (more precisely, we use its LQIP-NMF-FCLSU version: see details in Ref. [13]). For the sake of clarity, the pseudo-code of this LQIP-NMF algorithm is provided hereafter.

Algorithm: Linear-Quadratic Inertia-constrained Pixel-by-pixel NMF (LQIP-NMF)

```
$i \longleftarrow 1$
While ($J_{lqip-nmf} > Threshold$
and $i < Maximum\_number\_of\_iterations$):
  1. Update matrix $\tilde{\mathbf{R}}$:
     $\tilde{\mathbf{R}}^{(i+1)} \longleftarrow \tilde{\mathbf{R}}^{(i)} + \alpha_{\dot{\mathbf{R}}} \left( \tilde{\mathbf{C}}^{(i)T} \left( \mathbf{X} - \dot{\mathbf{C}}^{(i)} \dot{\mathbf{R}}^{(i)} \right) - \frac{2w}{P} \left( \mathbf{Id}_{PM} - \frac{1}{P} \mathbf{U} \right) \tilde{\mathbf{R}}^{(i)} + \left( \mathbf{Id}_{PM} \otimes 1_{1K} \right) \times \left( \left( \left( \mathbf{Id}_P \otimes \mathbf{F} \right) \left( \mathbf{Id}_P \otimes \left( 1_{M1} \otimes \mathbf{Id}_M \right) \right) \tilde{\mathbf{R}}^{(i)} \right) \odot \left( \left( \mathbf{Id}_P \otimes \left( 1_{M1} \otimes \mathbf{Id}_K \right) \right) \tilde{\mathbf{\Gamma}}^{(i)} \left( \mathbf{X} - \dot{\mathbf{C}}^{(i)} \dot{\mathbf{R}}^{(i)} \right) \right) \right) \right)$

     $\tilde{\mathbf{R}}^{(i+1)} \longleftarrow max(\tilde{\mathbf{R}}^{(i+1)}, \epsilon)$

  2. Update matrix $\tilde{\mathbf{T}}$:
     $\tilde{\mathbf{T}}^{(i+1)} \longleftarrow \left( (\mathbf{Id}_P \otimes \mathbf{G_1}) \tilde{\mathbf{R}}^{(i)} \right) \odot \left( (\mathbf{Id}_P \otimes \mathbf{G_2}) \tilde{\mathbf{R}}^{(i)} \right)$

  3. Update matrix $\tilde{\mathbf{C}}$:
     $\dot{\mathbf{C}}^{(i+1)} \longleftarrow \dot{\mathbf{C}}^{(i)} + \alpha_{\dot{\mathbf{C}}} \left( \left( \mathbf{X} - \dot{\mathbf{C}}^{(i)} \dot{\mathbf{R}}^{(i)} \right) \dot{\mathbf{R}}^{(i)T} \right)$.
     Then reset to zero all "off-block-diagonal elements" (see Ref. [13]).

  4. Post-process linear coefficients:
     For $p = 1$ to $P$
         $\mathbf{c_p}^{(i+1)} \longleftarrow max(\mathbf{c_p}^{(i+1)}, \epsilon)$
         $\mathbf{c_p}^{(i+1)} \longleftarrow \mathbf{c_p}^{(i+1)} / \sum_{m=1}^{M} c_{p,m}$

  5. Post-process quadratic coefficients:
     For $p = 1$ to $P$
         Update the vectors corresponding to all $m$ and $\nu$ as follows:
         $\mathbf{c_{p,(m,\nu)}}^{(i+1)} \longleftarrow max(\mathbf{c_{p,(m,\nu)}}^{(i+1)}, \epsilon)$
         $\mathbf{c_{p,(m,\nu)}}^{(i+1)} \longleftarrow min(\mathbf{c_{p,(m,\nu)}}^{(i+1)}, 0.5)$

  6. $i \longleftarrow i + 1$
```

In the present paper, we aim at taking advantage of the existence of both the linear version (14) and the bilinear version (22) of the mixing models accounting for spectral variability, and of the associated IP-NMF and LQIP-NMF unmixing algorithms, by extending the general approach for combining algorithms that we introduced in Section 2.1: to handle linear mixtures with spectral variability, we here consider the IP-NMF and LQIP-NMF algorithms, that estimate different variables (namely $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{C}}$, or $\dot{\mathbf{R}}$ and $\dot{\mathbf{C}}$), by optimizing different cost functions, and we aim at combining these algorithms, typically to allow one of them to escape from its local minima and thus to better estimate the above variables. The considered combinations of algorithms and their performance are presented below in Section 4, after defining the data used in tests, in Section 3. Before moving to that experimental part, we here summarize 1) the main features of the problem to be solved, that we introduced above, and 2) the particular relevance of the approach that we selected to solve this problem. The considered two data models are (14) and (22). Whereas (14) is naturally linear with respect to both the pure material spectra and the associated mixing coefficients, we stress that the model (22) has a special status: it originates from a model that is nonlinear, and more specifically bilinear, with respect to the pure material spectra, but we succeeded in reformulating it as a new model, namely (22), which is also *linear*, but now with respect to the above-defined *extended* spectra (and associated mixing coefficients). Thus, the two final data models that we face are linear with respect to each of the two types of unknowns, i.e. each of these models can be expressed as the product of two unknown matrices, as shown by (14) and (22). For each of these models, the estimation of the unknown matrices is therefore a matrix factorization problem, which is a type of linear Blind Source Separation (BSS) problem. One then has to choose which class of linear BSS methods one wants to use for solving the above two problems. The suitable class (or classes) depends on the properties of the considered data matrices. In the present investigation, the property met by the considered (extended) pure spectra and associated mixing coefficients is non-negativity, i.e. each matrix factorization problem that is faced is more specifically a non-negative matrix factorization (NMF) problem. Therefore, the algorithms used to solve these NMF problems in this paper are NMF algorithms. It should be noted that, in contrast, the other main two classes of linear BSS methods, that is Independent Component Analysis (ICA) and Sparse Component Analysis (SCA), are not suited here, because the considered data do not have the statistical independence or sparsity properties respectively required by ICA and SCA in the general framework that we want to address in this paper.

3. Considered data and performance criterion

3.1. Data

We aim at accurately, and hence numerically, evaluating the performance of the practical methods proposed further in this paper, especially in order to compare them in terms of convergence time and possible overfitting, estimation accuracy and sensitivity of these quantities to the parameter values of the learning algorithms. To this end, we process a matrix $\mathbf{X}$ of mixed spectra, we thus especially get the estimated pure spectra, and we compare these estimates to the actual values of these spectra. Implementing that protocol for a matrix $\mathbf{X}$ of *real*, i.e. measured, mixed data is constraining even when only considering the standard mixing model from the literature (i.e. the linear model without variability), because knowing the above actual values (i.e. ground truth) requests one to know all pure spectra involved in the considered scene. It is then infeasible to extend the above protocol to *real* data $\mathbf{X}$ that obey the much more complex data model (14) including variability that is considered hereafter because, due to that spectral variability, for each class of pure materials, one would need all versions of the corresponding spectrum for all pixels. To avoid that problem, we hereafter use mixed data $\mathbf{X}$ that are synthetic but realistic, since they combine real pure spectra that have variability, by using the model (14).

More precisely, the reported tests involve $M = 3$ classes of pure materials, namely tiles, vegetation and asphalt, which is relevant for urban applications. For each class, a complete set of presumably pure spectra were manually extracted from a real urban hyperspectral image from the city of Toulouse (France). The features of this image are the same as in Ref. [13] and the considered area is shown in Fig. 2. The initial database of presumably pure spectra consists of 190 tile spectra, 55 vegetation spectra and 52 asphalt spectra. This set of spectra yields large intraclass variabilities. The angular variability is here separately evaluated for each of the 3 classes, by means of the Spectral Angle Mapper (SAM) (see Ref. [23]) with respect to the mean of the spectra of the considered class. The mean and maximum values of this SAM over a class are respectively equal to 2.64° and 8.53° for the tile class, 7.38° and 15.39° for the vegetation class, 2.73° and 10.05° for the asphalt class. In contrast, it is generally considered that spectra corresponding to the

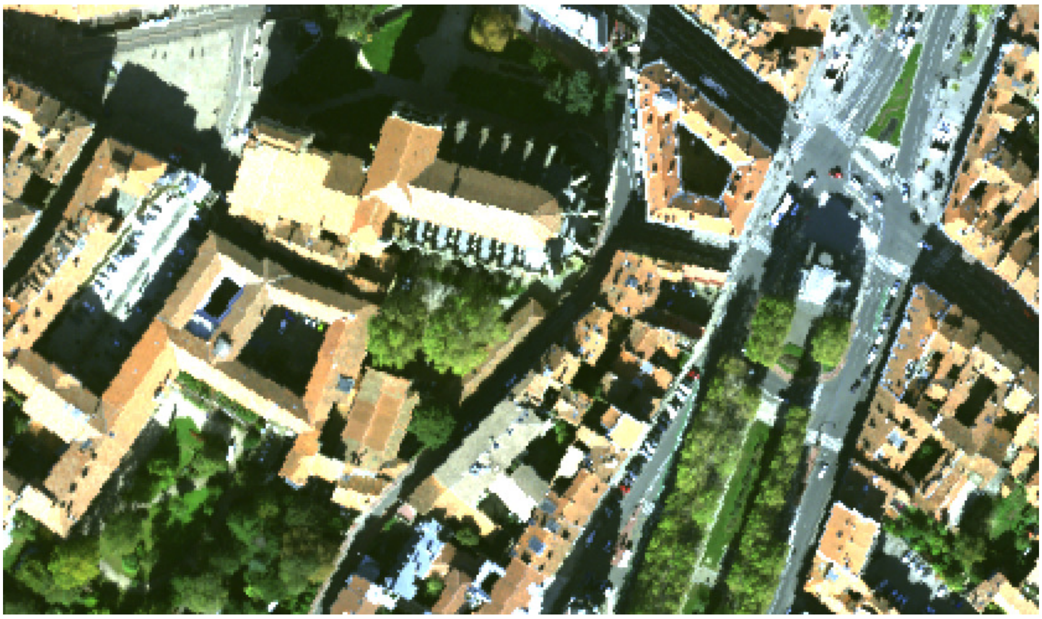

Figure 2: Image of the considered urban area (Toulouse, France) (Reprinted from Ref. [13]).

same class of materials yield SAMs limited to a few degrees. For instance, for default settings, the SAM-based classification method available in the commercial ENVI software (see Ref. [28]) considers that spectra from a class have a SAM lower than 0.1rad $\simeq 5.73°$, with respect to the representative of that class. The larger values faced in our initial database may result from the fact that a few spectra that were manually selected (based on the objects present in the considered scene) are in fact not completely pure.

The tests reported below were therefore performed with a subset of the above database, defined as follows. We selected a value f, within the range 0% to 100%, that defines the fraction of spectra that we want to keep, separately for each class of pure materials. For each such class, the spectra that are kept are those which yield a SAM, with respect to the mean spectrum of that complete class, that is lower than a bound. That bound has a correspondence with the parameter f, as explained in Ref. [13]. The parameter f thus allows us to control the variability of the considered data. We obtained a realistic variability by setting f to 80% (see Ref. [13]).

The subset of pure spectra selected with the above-defined approach is shown in Fig. 3. It is used as follows. For each pixel, one pure spectrum

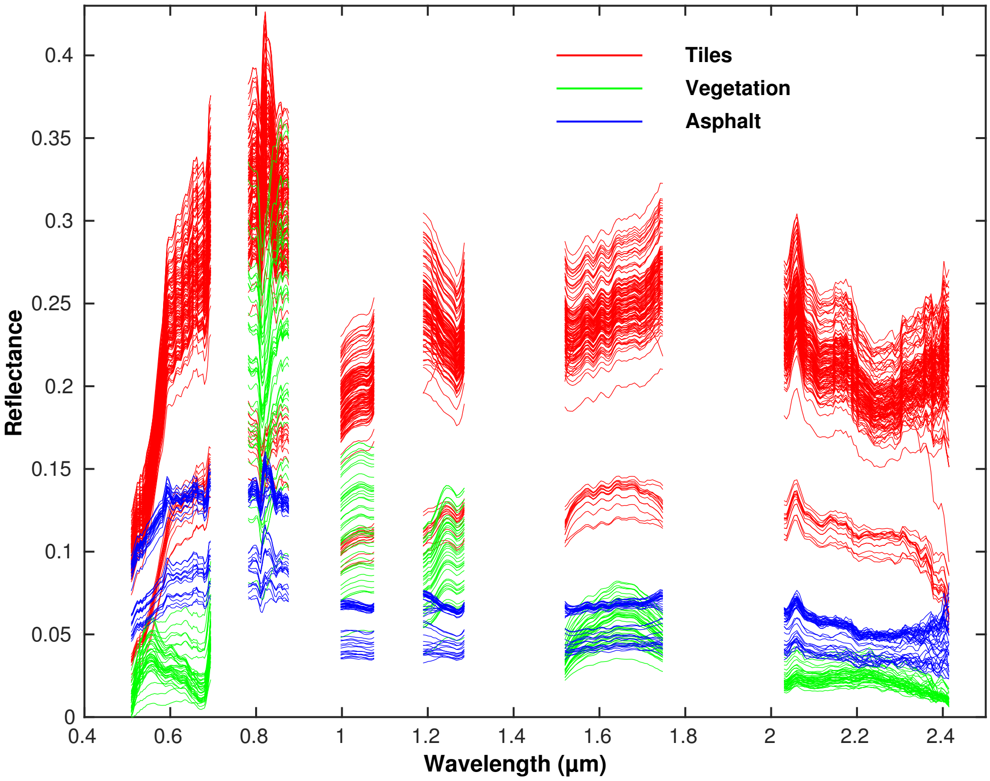

Figure 3: Final database of pure spectra with variability.

is randomly selected from the considered subset for each class, to derive the corresponding mixed spectrum of the data matrix $\mathbf{X}$, which contains 756 pixels. The mixing coefficient values used to this end in (14) are defined as follows. Some coefficients in $\tilde{\mathbf{C}}$ are set to zero, especially to create a few pure pixels (i.e. pixels with one coefficient equal to one, whereas all other coefficients are equal to zero). In each non-pure pixel, the non-zero mixing coefficients are arbitrarily selected and scaled so that they sum to one.

3.2. Performance criterion

To evaluate the performance of the tested unmixing methods, the quality of the estimated pure spectra is measured by comparing them to the actual pure spectra used to create the mixed matrix $\mathbf{X}$. This comparison is performed by using the above-mentioned SAM. This SAM parameter is first averaged over all $M = 3$ spectra and all image pixels. Moreover, for each considered configuration, 100 runs are performed with different randomly drawn data in each run. The SAM values provided hereafter are averages derived from all these runs.

For NMF-based methods, an alternative performance criterion that is sometimes used in the literature is the reconstruction error, mainly defined for linear mixtures without variability. Its extension to the linear model with variability, that is (14), reads $\left\|\mathbf{X} - \hat{\tilde{\mathbf{C}}}\hat{\tilde{\mathbf{R}}}\right\|_F$, where $\hat{\tilde{\mathbf{C}}}$ and $\hat{\tilde{\mathbf{R}}}$ are the estimates, provided by the considered unmixing algorithm, of the actual matrices $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ faced in the processed data. This reconstruction error may be misleading, especially because it does not detect overfitting during adaptation, as detailed in the analysis of the experimental behavior of the IP-NMF method in Section 4. To put it briefly, the reconstruction error only measures the quality of the estimate of the matrix product $\tilde{\mathbf{C}}\tilde{\mathbf{R}}$, whereas what is important in practice is the quality of the estimate of the matrix $\tilde{\mathbf{R}}$ of estimated spectra. The quality of the latter matrix is exactly what is evaluated by the above-mentioned mean SAM, and this is the reason why we hereafter use the mean SAM as the performance criterion, whereas we do not consider the reconstruction error. The same principle applies to the approach based on the bilinear mixing model with variability.

4. Practical algorithms and test results

In the present section, we first separately consider the IP-NMF and LQIP-NMF algorithms defined in Section 2.2, we define how we applied them to the data described in Section 3.1 and we present the performance thus obtained. This serves as the basis for then applying the general methodology proposed in Section 2.2, so as to introduce several new algorithms defined as combinations of IP-NMF and LQIP-NMF. A comparative analysis of the performance of all considered algorithms is provided. Moreover, we compare all these advanced methods with a standard method from the literature, namely VCA (see Ref. [33]).

4.1. Operation and performance of the IP-NMF algorithm

We first consider the IP-NMF algorithm, used to estimate the matrices $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ of the data model (14). The corresponding adaptive matrices of this algorithm must be initialized and this is performed as follows. The well-known VCA algorithm (see Ref. [33]) is first applied to the observed matrix $\mathbf{X}$. It here yields $M = 3$ pure spectra estimates for the complete image. They respectively correspond to the tile, vegetation and asphalt classes. These three

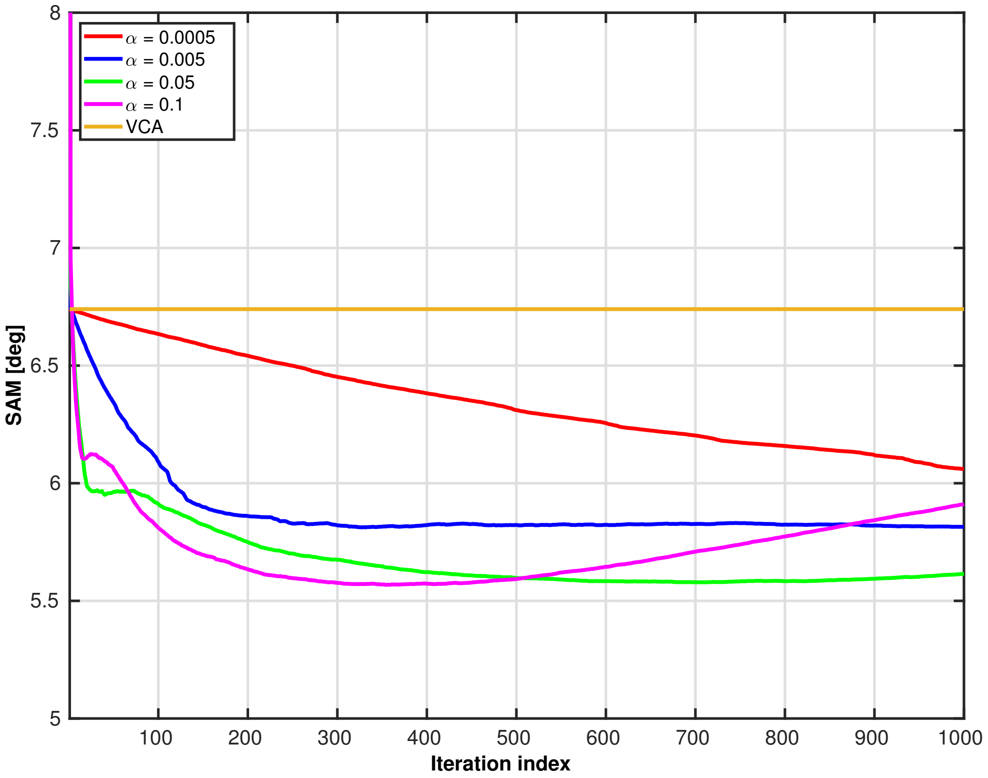

Figure 4: 1) IP-NMF algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations, 2) SAM of VCA algorithm.

spectra as used as the initial values of the pure material spectra estimates of IP-NMF for all pixels of the image, i.e. for the adaptive counterpart of (15) corresponding to all pixels p. Moreover, all the non-zero elements of the adaptive counterpart of $\tilde{\mathbf{C}}$ are initialized to $1/M = 1/3$ (so that they sum to one, as usual). The adaptive versions of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ are then updated with the rules of IP-NMF, with $\mu = 30$ in (18) and using the same adaptation gain for both matrices. This gain is denoted as α hereafter. Successively for several values of α, we investigate the evolution of the SAM of the estimated pure spectra, as defined in Section 3.2, throughout the iterations of IP-NMF.

The results obtained in the above-defined conditions are provided in Fig. 4. This shows that applying this single gradient-based algorithm leads to the general limitation of gradient algorithms: both the number of iterations required for reaching an almost constant SAM value and that SAM value itself significantly depend on the value of the adaptation gain α, and this does not allow one to easily apply this approach in a completely automated way.

It should be stressed that these tests not only address "standard conditions", i.e. with moderate values for the adaptation gain (mainly up to $\alpha = 0.05$) and number of iterations, but also the case when one makes the cumulative effect of the adaptation of the tunable matrices very large, by using both a high adaptation gain (mainly $\alpha = 0.1$) and a high number of iterations. In the latter case, the SAM value first decreases and then remains almost constant over a significant number of iterations, but it then starts to increase again (see mainly the part of Fig. 4 with $\alpha = 0.1$ and more than 400 iterations). This phenomenon may be interpreted as overfitting as will now be shown. Generally speaking, overfitting in adaptive / machine learning algorithms refers to situations when an excessive number of adaptation steps makes the considered adaptive system become too specialized. When continuing adaptation, the system keeps on improving its performance with respect to the performance parameter that it optimizes (or at least it keeps this parameter constant). However, other parameters that may be of interest but that are not taken into account in the considered cost function evolve differently: whereas they also improve during the first series of iterations, they then start to degrade as adaptation continues and overfitting occurs.

The above phenomenon is often considered for *supervised* machine learning algorithms, e.g. used for classification tasks (for an overview of supervised and unsupervised learning, including their application to classifications and their quantum extensions, see Ref [12]). In that case, the parameter that is monitored during training and that keeps on improving is the value of the cost function (or the rate of good classifications) over the training database. The parameter that is not monitored during training and that may start degrading again after a series of iterations is the value of the cost function (or the rate of good classifications) over a generalization database that is not seen by the training procedure.

In contrast, *unsupervised* machine learning algorithms do not rely on that concept of separate training and generalization databases and thus lead to other forms of overfitting. In particular, overfitting occurs as follows in the unmixing algorithm considered here. This adaptive algorithm aims at minimizing the cost function (18) and thus generally tends to reduce the value of its contribution $\left\|\mathbf{X} - \tilde{\mathbf{C}}\tilde{\mathbf{R}}\right\|_F$, which is nothing but the version tuned during training of the reconstruction error mentioned in Section 3.2. The unmixing algorithm thus tends to obtain matrix estimates $\hat{\tilde{\mathbf{C}}}$ and $\hat{\tilde{\mathbf{R}}}$ that are such that their *product* is close to its desired value $\mathbf{X}$. But this does not guarantee that each of the matrices $\hat{\tilde{\mathbf{C}}}$ and $\hat{\tilde{\mathbf{R}}}$ considered separately becomes close to its desired value $\tilde{\mathbf{C}}$ or $\tilde{\mathbf{R}}$. Instead, overfitting may occur, which here means that the adaptive counterparts of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ may first start to respectively get closer to the actual matrices $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ but then, after a series of iterations, they may start to evolve further from the actual matrices. This occurs when the algorithm starts to overfit the adaptive version of the product $\tilde{\mathbf{C}}\tilde{\mathbf{R}}$ (since this parameter is "seen" by the considered cost function), at the expense of making its two factors $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ separately evolve towards undesired values (since they are not *individually* taken into account by this learning procedure[2]). In other words, the proposed approach thus becomes too specialized on the quantity $\tilde{\mathbf{C}}\tilde{\mathbf{R}}$ that it aims at optimizing, at the expense of its generalization capabilities separately with respect to $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$. The above-defined phenomena are illustrated in Fig. 4 and 5: whereas the cost function used during adaptation keeps on decreasing throughout all iterations (see Fig. 5 for its average over 100 runs), the SAM that defines the quality of estimation of $\tilde{\mathbf{R}}$ starts increasing again after a large number of iterations and for a large adaptation gain, as discussed above (see Fig. 4).

It should be noted that, despite the above-defined limitations, IP-NMF, which is the simplest of our methods considered here, is already of interest because it yields a quite significant performance improvement as compared with the VCA method from the literature: whereas the SAM is equal to 6.7° for VCA, it goes down to 5.6° over a significant range of numbers of iterations, depending on α, for IP-NMF.

4.2. Operation and performance of the LQIP-NMF algorithm

We now consider the LQIP-NMF algorithm, used to estimate the matrices $\tilde{\mathbf{C}}$, $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{\Gamma}}$ (and $\tilde{\mathbf{T}}$, that is automatically derived from $\tilde{\mathbf{R}}$) of the data model (20). The adaptive counterparts of the matrices $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ are here initialized in the same way as for the IP-NMF algorithm. The elements of $\tilde{\mathbf{\Gamma}}$ are initially set to 0 (because they should then ideally converge to 0 for the actually linear

[2] The adaptive counterpart of $\tilde{\mathbf{R}}$ is adapted without knowing its target value, i.e. in an unsupervised way, because the actual value of $\tilde{\mathbf{R}}$ is unknown and has to be estimated. The same applies to $\tilde{\mathbf{C}}$. In contrast, the target value of their product $\tilde{\mathbf{C}}\tilde{\mathbf{R}}$ is known: it is the observed data matrix $\mathbf{X}$.

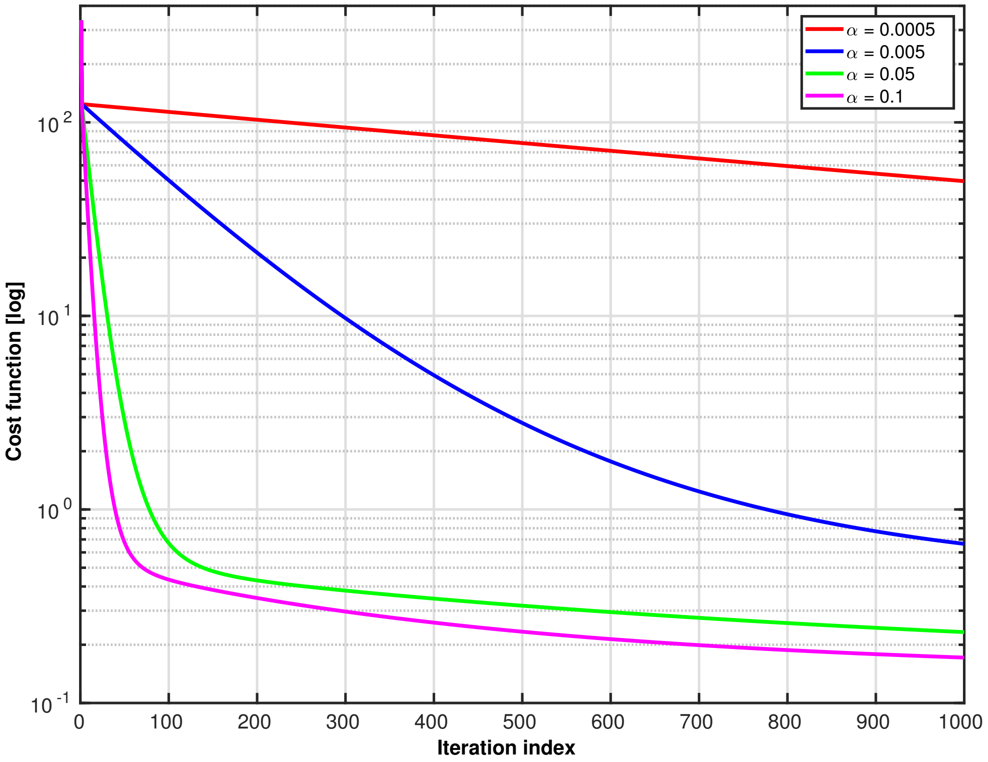

Figure 5: IP-NMF algorithm for various adaptation gains α: evolution of cost function throughout adaptation iterations.

mixture considered here; but they then evolve, according to the LQIP-NMF algorithm, during adaptation). The matrices $\tilde{\mathbf{C}}$, $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{\Gamma}}$ are then updated with the rules of LQIP-NMF, with $\mu = 30$ in the associated modified form of (18) and using the same adaptation gain for all these matrices. This gain is hereafter denoted as α.

We mainly investigate the evolution of the SAM of the estimated pure spectra throughout the iterations of LQIP-NMF, successively for several values of α. The results thus obtained are shown in Fig. 6. Moreover, we provide the evolution of the cost function during adaptation in Fig. 7. These figures show that LQIP-NMF qualitatively has the same behavior as IP-NMF. This not only includes the dependence of performance with respect to the number of iterations and adaptation gain, but also the existence of a minimum of SAM during adaptation for higher values of the adaptation gain (Fig. 6), again related to overfitting, i.e. with a cost function that then keeps on de-

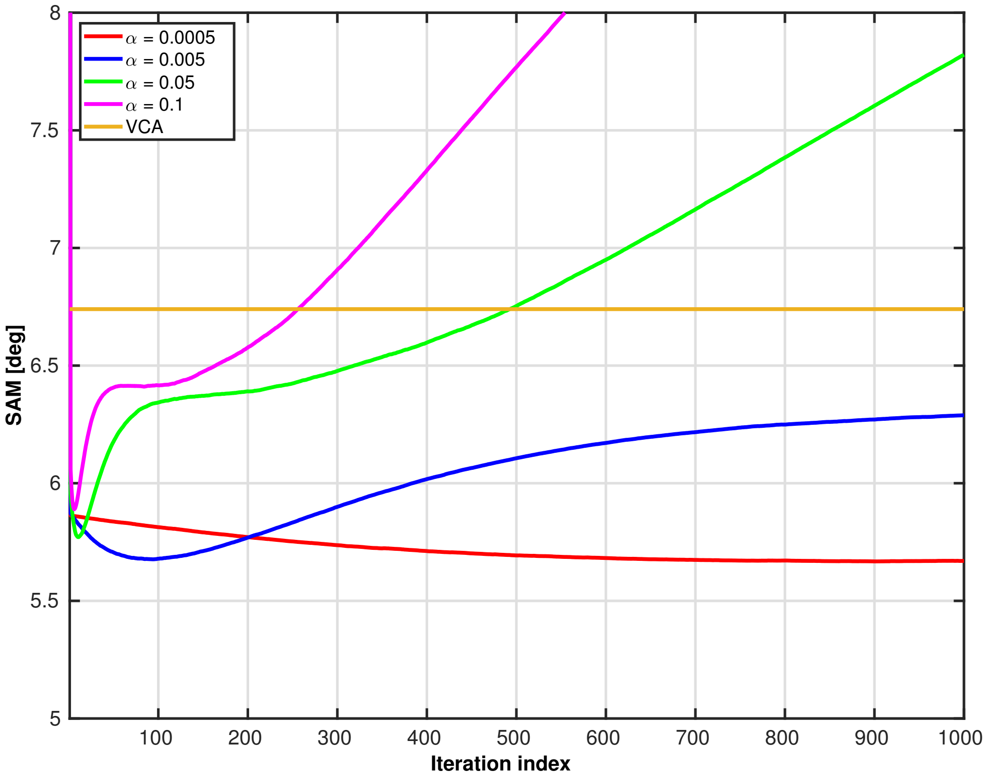

Figure 6: 1) LQIP-NMF algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations, 2) SAM of VCA algorithm.

creasing (Fig. 7). However, this phenomenon is more visible for LQIP-NMF, i.e. it appears for a larger range of values of the adaptation gain and with a lower number of iterations. This may be explained by the fact that the adaptive model of LQIP-NMF is more flexible, i.e. it contains additional adaptive variables as compared with IP-NMF, and the price to pay for this freedom is that LQIP-NMF is thus prone to more "rapidly" (when increasing the adaptation gain and/or in terms of number of iterations) evolve towards values of its adaptive variables that get too specialized, thus optimizing the cost function thanks to flexibility, but at the expense of degrading the SAM parameter.

Anyway, LQIP-NMF is also attractive because it yields a quite significant performance improvement as compared with VCA: whereas the SAM is equal to 6.7° for VCA, it goes down to 5.7° over a significant range of number of iterations, depending on α, for LQIP-NMF.

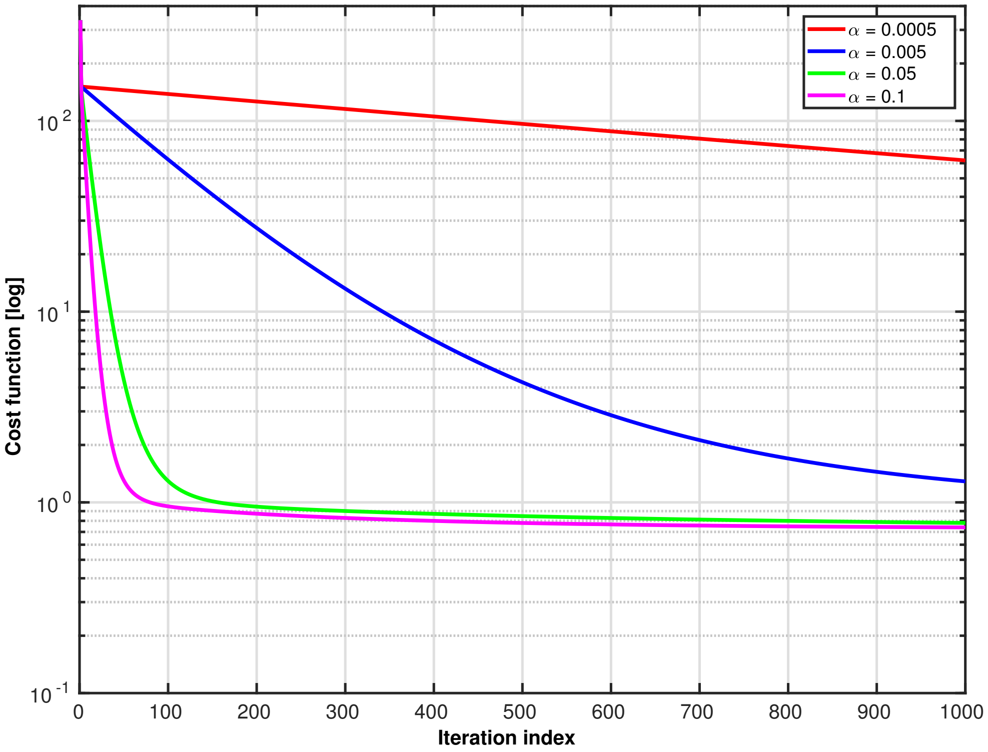

Figure 7: LQIP-NMF algorithm for various adaptation gains α: evolution of cost function throughout adaptation iterations

4.3. Two-stage hybrid methods

We now move to "hybrid methods", i.e. methods that combine IP-NMF and LQIP-NMF. We first describe their simplest versions, that start with a single run of one of the IP-NMF and LQIP-NMF methods, whose results are then used to initialize a single run of the other method. These methods are therefore stated to be two-stage methods. This approach yields two methods, depending on the order in which IP-NMF and LQIP-NMF are executed.

The first method, called Hybrid-1, first runs LQIP-NMF, initialized as in Section 4.2, and then operated with an adaptation gain fixed to 0.005 and a number of iterations fixed to 100. These values are selected based on two criteria. First, they are "standard values" that yield rather satisfactory performance for this method, a relatively limited sensitivity to parameter values (see Fig. 6) and a reasonable computational load thanks to a moderate number of iterations. Second, these values are intentionally selected without fine tuning them to obtain optimum performance, because we here aim at determining a performance level that can be obtained without a high sensitivity to optimized parameter values. The estimates of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ thus obtained with LQIP-NMF are then used to initialize the corresponding variables of IP-NMF. Adaptation is then performed with the rules of IP-NMF, successively considering various values for its adaptation gain. This method may therefore be seen as an extension of IP-NMF (since the SAM values eventually analyzed as the output of Hybrid-1 are those resulting from its second stage, i.e. from IP-NMF), that uses an advanced initialization provided by LQIP-NMF, instead of its plain initialization by VCA and standard abundance values considered in Section 4.1.

We here again investigate the evolution of SAM throughout adaptation. Its evolution during the first stage of Hybrid-1, i.e. when LQIP-NMF is executed, remains the same as in Fig. 6. In contrast, the evolution of SAM during the second stage of Hybrid-1 is new, because it corresponds to IP-NMF executed *with new initial values* for its adaptive matrices. The results thus obtained are provided in Fig. 8. This first shows that this Hybrid-1 method also yields a quite significant performance improvement as compared with VCA, with a SAM equal to 6.7° for VCA, and down to 5.5° over a quite significant range of numbers of iterations, depending on α, for Hybrid-1. As compared with the stand-alone version of IP-NMF (Section 4.1), Hybrid-1 also yields a slight improvement of the minimum SAM value (5.5° vs. 5.6°). Their sensitivity to parameter values (adaptation gain and number of iterations) and their overfitting effects are relatively similar. This may result from the fact, although the first stage of Hybrid-1 performed with LQIP-NMF allows an adaptation in a wider model space than with IP-NMF, the second stage of Hybrid-1 is performed with IP-NMF and therefore gets back to a more restricted class of models and hence to the limitations of IP-NMF.

We now consider the second hybrid method, called Hybrid-2, that first runs IP-NMF, initialized as in Section 4.1 and then operated with a given adaptation gain and a given number of iterations (two scenarios are considered for these two parameters, as detailed below). The estimates of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ thus obtained with IP-NMF are then used to initialize the corresponding variables of LQIP-NMF. As in its stand-alone version (Section 4.2), the adaptive variable corresponding to $\tilde{\mathbf{\Gamma}}$ is initially set to 0. The matrices $\tilde{\mathbf{C}}$, $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{\Gamma}}$ are then updated with the rules of LQIP-NMF, successively considering various values for its adaptation gain. This method may therefore be seen as an extension of LQIP-NMF (since the SAM values eventually analyzed as the

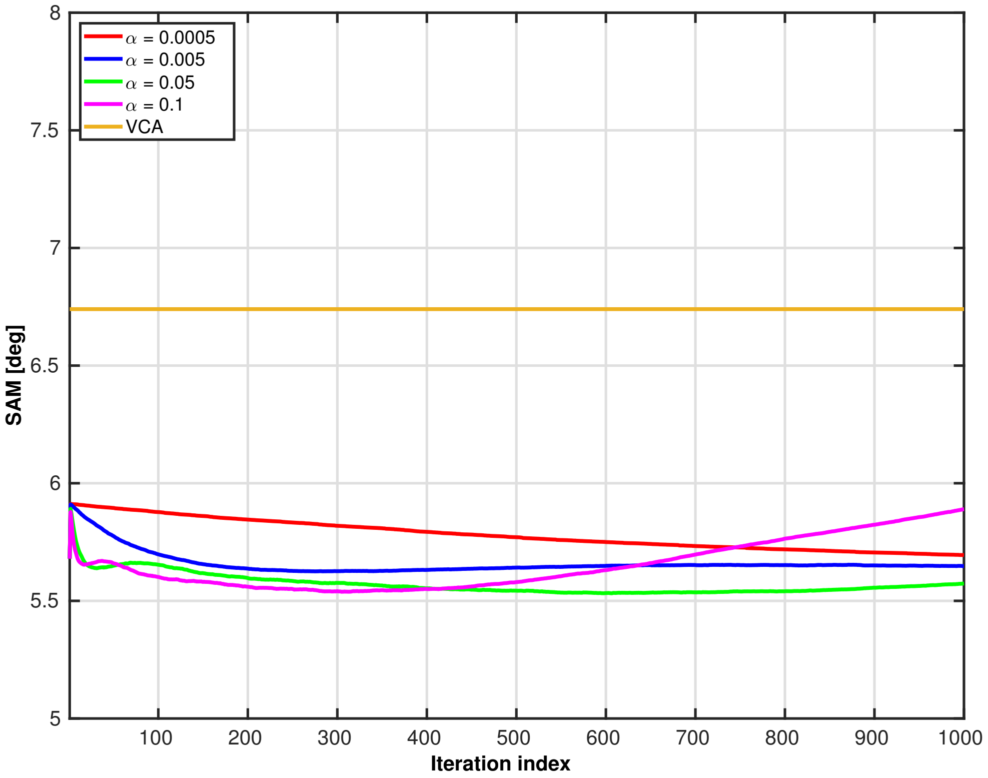

Figure 8: 1) second stage of Hybrid-1 algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations, 2) SAM of VCA algorithm.

output of Hybrid-2 are those resulting from its second stage, i.e. from LQIP-NMF), that uses an advanced initialization provided by IP-NMF, instead of its plain initialization by VCA and standard abundance values considered in Section 4.2.

We here again investigate the evolution of SAM throughout adaptation. Its evolution during the first stage of Hybrid-2, i.e. when IP-NMF is executed, remains the same as in Fig. 4. In contrast, the evolution of SAM during the second stage of Hybrid-2 is new, because it corresponds to LQIP-NMF executed *with new initial values* for its adaptive matrices. The results thus obtained are shown in Fig. 9 and 10, that correspond to different parameter values for the execution of IP-NMF. In the first scenario (Fig. 9), IP-NMF is executed with an adaptation gain fixed to 0.1 and a number of iterations fixed to 100. As in the first stage of the Hybrid-1 method, these parameter values are here selected by restricting ourselves to "standard values" that

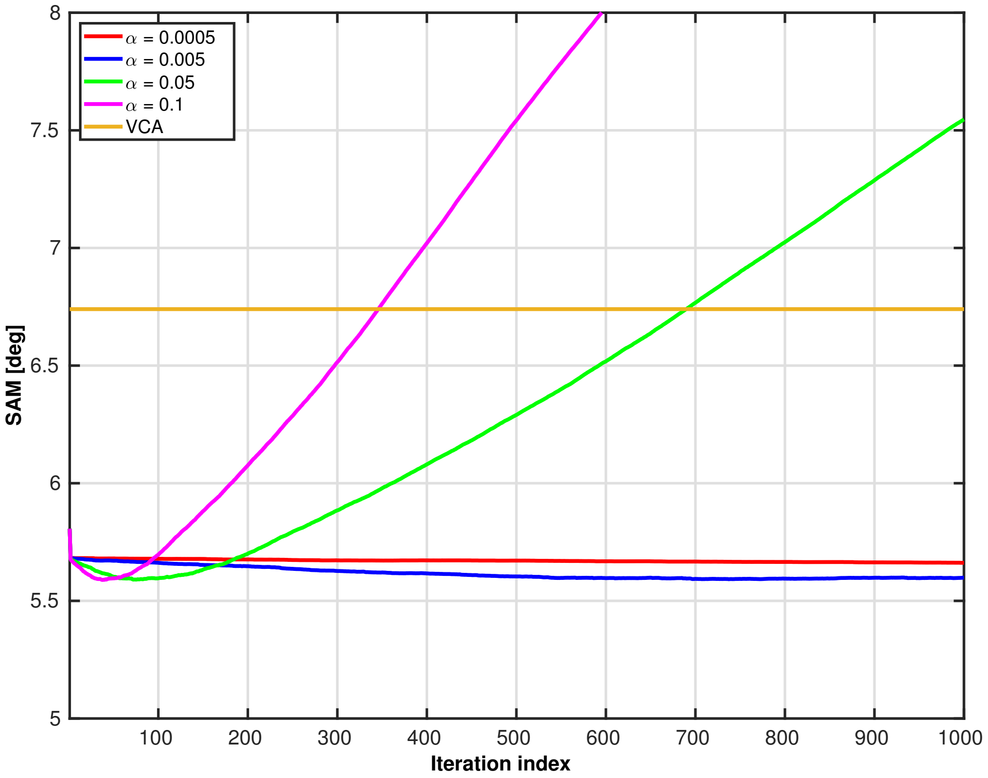

Figure 9: 1) second stage of Hybrid-2 algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations (with first scenario for first stage of Hybrid-2: see text), 2) SAM of VCA algorithm.

yield satisfactory but non-optimum performance for the IP-NMF method (see Fig. 4), with an acceptable sensitivity to parameter values and a reasonable computational load. In particular, although $\alpha = 0.1$ may be considered to be rather high, it is here selected in order to use a rather low number of iterations and to thus limit the computational load. A complementary approach is therefore also considered, in the second scenario (see Fig. 10): the adaptation gain of IP-NMF is then fixed to 0.05 and its number of iterations to 1000. The estimates of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ obtained with IP-NMF yield a somewhat better SAM for the first stage in this second scenario than in the first one (see Fig. 4), but the price to pay for that improved performance is a higher computational load, due to a higher number of iterations (10 times higher).

This first shows that this Hybrid-2 method also yields a quite significant performance improvement as compared with VCA, with a SAM equal to 6.7°

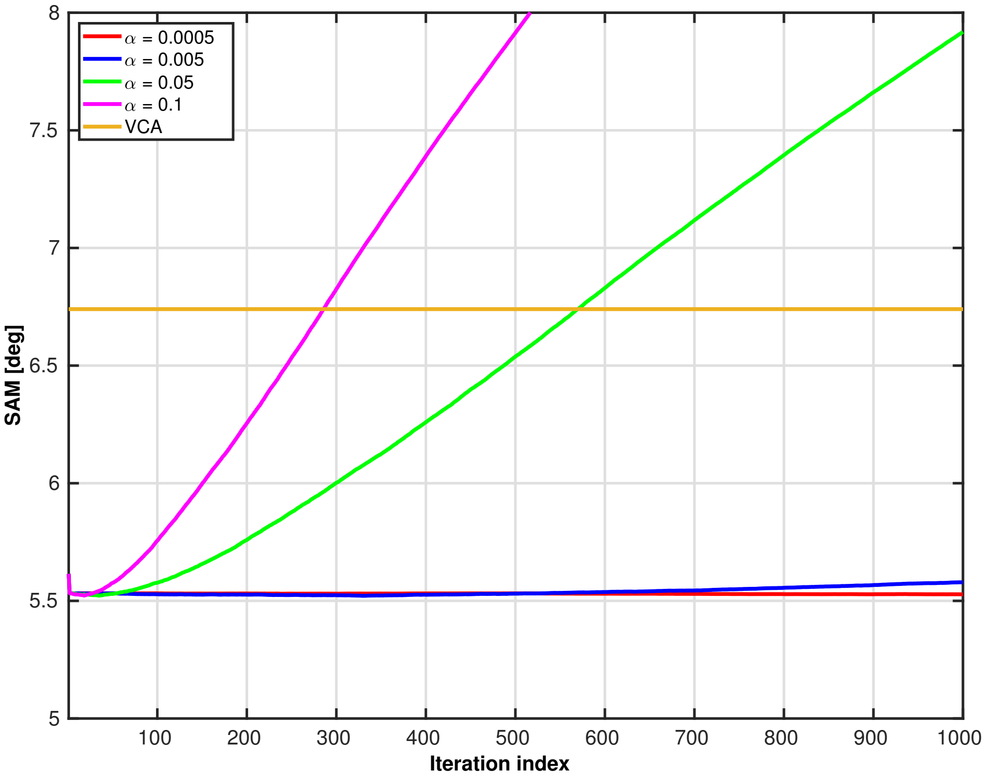

Figure 10: 1) second stage of Hybrid-2 algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations (with second scenario for first stage of Hybrid-2: see text), 2) SAM of VCA algorithm.

for VCA, and down to 5.6° (scenario 1) or 5.5° (scenario 2) over a significant range of number of iterations, depending on α, for Hybrid-2, and especially over *a very large number of iterations* for moderate values of α, namely up to $\alpha = 0.005$ for both scenarios. The latter comment also means that, in the considered conditions, Hybrid-2 has the major advantage of being quite insensitive to overfitting (during its two stages) when using a moderate adaptation gain during its second stage. Besides, for such adaptation gains, its SAM during its second stage varies much less than during the second stage of Hybrid-1 and than with the stand-alone version of IP-NMF, and it has better values. Overall, the good performance of Hybrid-2 may result from the fact that it combines the following two features: 1) its first adaptation stage already yields an accurate solution (by using IP-NMF instead of VCA) and then 2) its second stage keeps a memory of this type of solution and combines it with a flexible mixing model (the bilinear model of LQIP-NMF) that allows it to have good generalization properties, rather than being too specialized.

As compared with the stand-alone version of LQIP-NMF, Hybrid-2 can decrease SAM down to about the same range of values or somewhat below (especially for the second scenario), but moreover with the advantage of having a much lower sensitivity for a moderate adaptation gain in its second stage.

As an overall result, among the four methods studied at this stage, Hybrid-2 is the preferred one. Moreover, the two scenarios that we considered for the first stage of its operation yield relatively similar performance, which also shows its low sensitivity to the parameters of its first stage.

4.4. Multi-stage hybrid methods

The results reported in the previous sections suggest us to further extend the above approaches, especially Hybrid-2, thus creating an algorithm with more than two stages, that essentially consists of alternately using IP-NMF and LQIP-NMF. More precisely, the Hybrid-3 method that we here introduce successively (and alternately) runs each of the methods IP-NMF and LQIP-NMF twice, and thus consists of four stages, organized as follows. We first run the IP-NMF and then the LQIP-NMF stages once, as in the first scenario (to reduce the computational load) of Hybrid-2, here only considering the following parameter values for the LQIP-NMF stage of Hybrid-2: $\alpha = 0.005$ and a number of iterations equal to 100 (here again, to limit the computational load and without fine tuning all parameters). The third stage of Hybrid-3 then consists of running IP-NMF for the second time, with the same α and number of iterations as in the first stage (that is, $\alpha = 0.1$ and 100 iterations) and with its adaptive version of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ initialized to the values reached at the end of the second stage (i.e. provided by LQIP-NMF). Finally, the fourth stage of Hybrid-3 consists of running LQIP-NMF for the second time, with its adaptive version of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ initialized to the values reached at the end of the third stage (i.e. provided by IP-NMF), whereas the adaptive variable corresponding to $\tilde{\mathbf{\Gamma}}$ is initialized to its value reached at the end of the second stage (i.e. provided by the first run of IP-NMF). This method may therefore be seen as a second extension of LQIP-NMF (since the SAM values eventually analyzed as the output of Hybrid-3 are those resulting from its final (i.e. fourth) stage, i.e. from LQIP-NMF), that uses an even more refined initialization than the plain version of LQIP-NMF and than Hybrid-2: here, the initialization point received by the final execution of LQIP-NMF results from the evolution of adaptive variables throughout the previous three stages of the algorithm.

We here investigate how the SAM provided by Hybrid-3 depends on the value of α and on the number of iterations of LQIP-NMF during the fourth stage. The results thus obtained are provided in Fig. 11. This first shows that Hybrid-3 also yields a quite significant performance improvement as compared with VCA, with a SAM equal to 6.7° for VCA, and down to (slightly less than) 5.5° over a significant range of numbers of iterations, depending on α, for Hybrid-3, and especially over *a very large number of iterations* for moderate values of α, namely up to $\alpha = 0.005$. Here again, this means that Hybrid-3 has the major advantage of being quite insensitive to overfitting. Hybrid-3 should also be compared to the best of the previous four NMF-based methods. We therefore first compare Hybrid-3 to the scenario of Hybrid-2 that uses the same parameter values for the IP-NMF stages, namely those of scenario 1, that are not precisely tuned with respect to SAM. Hybrid-3 then yields a slight performance improvement, with a SAM equal to 5.5°, instead of 5.6° for Hybrid-2, over a wide range of parameter values. But, now comparing Hybrid-3 to the scenario of Hybrid-2 that uses different parameter values for the IP-NMF stages, namely those of scenario 2 that yield a somewhat better SAM for Hybrid-2, Hybrid-3 yields almost the same performance as that version of Hybrid-2. Still, Hybrid-3 may also be preferred to Hybrid-2 in the latter scenario, because it does not rely on the need to use optimized parameter values. Overall, the performance improvement provided by Hybrid-3 as compared with Hybrid-2 is limited. This is consistent with the fact that extending Hybrid-3 to more than two alternate runs of IP-NMF and LQIP-NMF did not yield a significant performance improvement in the other tests that we performed.

As a global conclusion, among the preferred proposed methods, scenario 1 of Hybrid-2 can be considered as the “standard” version and is the one that yields the lowest computational cost. The associated SAM can easily be decreased down to 5.6°. Beyond that version, scenario 2 of Hybrid-2 and Hybrid-3 yield some performance improvement (SAM around 5.5°), but at the expense of a higher computational load (due to the overall higher number of iterations).

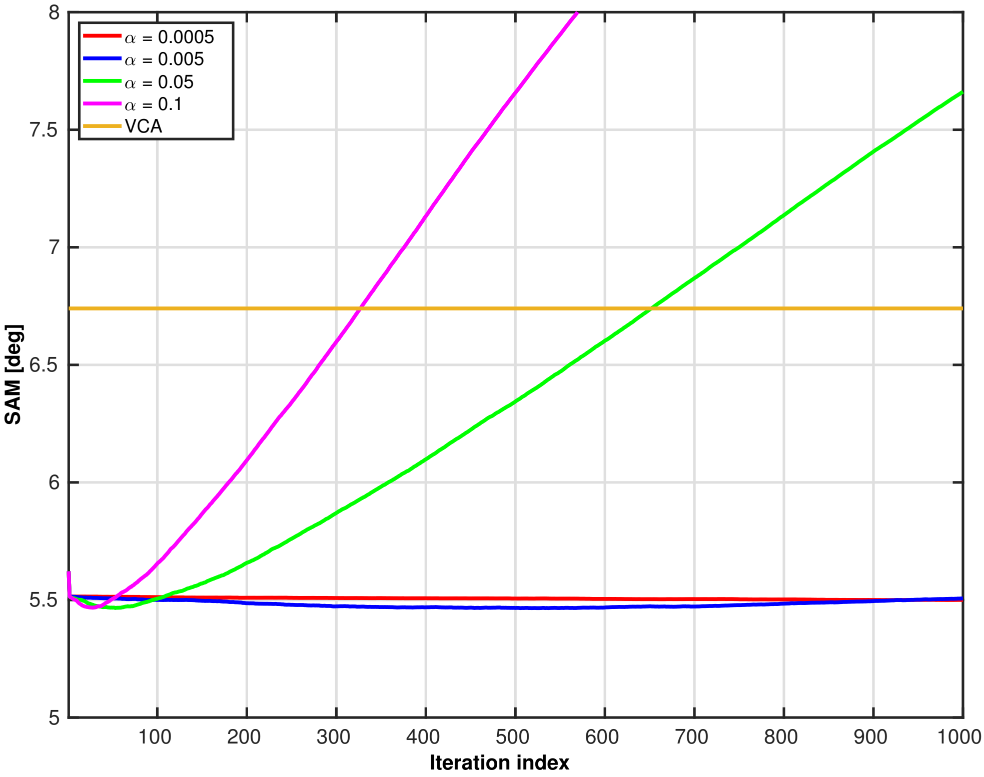

Figure 11: 1) fourth stage of Hybrid-3 algorithm for various adaptation gains α: evolution of SAM throughout adaptation iterations, 2) SAM of VCA algorithm.

5. Conclusion

Hyperspectral unmixing is a major research topic within the field of Remote Sensing. Whereas various methods were previously proposed for handling the case of linear mixtures without variability, the emphasis is currently put on their extensions to (mainly linear) mixtures with variability, which provide a much more relevant model of the diversity of pure material spectra faced in real hyperspectral images. Linear hyperspectral unmixing is essentially a matrix factorization problem (expressing the observed data matrix as the product of an abundance fraction matrix and of a pure material spectra matrix), moreover with additional constraints that especially include the nonnegativity of the matrix factors. Its unsupervised version (i.e. with unknown pure spectra) is therefore a constrained Nonnegative Matrix Factorization (NMF) problem. NMF-based methods have therefore been proposed to solve this problem, and recently extended to handle the case of mixtures with spectral variability. Papers from the literature dealing with this topic usually only provide a single value of the SAM performance parameter achieved with the considered methods. In contrast, in the present paper, we first analyzed in detail how performance depends on the conditions of operation of such an NMF algorithm, namely IP-NMF, which moreover addresses the more difficult configuration, i.e. the one with spectral variability. The conditions of operation that we varied are the adaptation gain and the number of learning iterations. This analysis showed that, as with standard adaptive / machine learning methods, the performance of this NMF algorithm significantly depends on the selected adaptation gain and number of iterations, which makes it difficult to efficiently operate it with a fully automated choice of parameter values. Moreover, again as with usual machine learning methods, this approach was shown to possibly yield overfitting problems, whose form for the considered unsupervised processing task was detailed in this paper.

We significantly improved performance from all above points of view by developing "hybrid NMF methods". These methods essentially consist of successively running several types of NMF methods (respectively IP-NMF and its LQIP-NMF extension intended for nonlinear mixtures), with different but compatible spaces for their adaptive variables and different cost functions, and with part of the adaptive variables of each method initialized with their final values provided by the previously executed method. Thus, these variables converge to values that yield good generalization properties, because they combine the information successively extracted by each of the executed algorithms. The attractiveness of this approach appears in the low sensitivity of the performance of the proposed algorithms with respect both to the adaptation gain and to the number of learning iterations, which allows one to apply these hybrid methods in an automated way. Moreover, the SAM values obtained over a wide range of values of the above parameters are thus significantly better than with a completely different standard unmixing method from the literature, namely VCA, used to initialize our methods, with a mean SAM that decreases from 6.7° for VCA to 5.5° for our algorithms.

Beyond its specific implementations analyzed above, an important contribution in this paper is the proposed concept of hybrid algorithms that successively perform complementary optimizations of different but compatible variables and with different cost functions: this opens the way to a wide range of new methods for hyperspectral unmixing and, more generally, to various machine learning problems, to reduce the sensitivity to adaptation parameters and overfitting problems.

Declarations

Funding. The authors would like to thank the French ANR project "HYEP ANR 14-CE22-0016-01". The investigations reported in the present paper were not performed in the framework of that HYEP project itself, but they reused part of the software and databases that were previously developed during that HYEP project.

Conflicts of interest/Competing interests. The authors report no confict of interest.

Availability of data and material. The datasets generated during and/or analysed during the current study are not publicly available because they belong to the French ANR project "HYEP ANR 14-CE22-0016-01". However, they are available from the corresponding author on reasonable request.

Code availability. The codes generated during the current study are not publicly available because they belong to the French ANR project "HYEP ANR 14-CE22-0016-01". However, they are available from the corresponding author on reasonable request.

References

[1] F. Z. Benhalouche, Y. Deville, M. S. Karoui, A. Ouamri, "Hyperspectral unmixing based on constrained bilinear or linear-quadratic matrix factorization", Remote Sensing, vol. 13, issue 11, paper no. 2132, 2021.

[2] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, J. Chanussot, "Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 2, pp. 354-379, 2012.

[3] R. A. Borsoi, T. Imbiriba, J. C. M. Bermudez, C. Richard, J. Chanussot, L. Drumetz, J.-Y. Tourneret, A. Zare, C. Jutten, "Spectral variability in hyperspectral data unmixing. A comprehensive review", IEEE Geoscience and Remote Sensing Magazine, pp. 2-49, 2021.

[4] A. Cichocki, S.-I. Amari, “Adaptive blind signal and image processing. Learning algorithms and applications”, Wiley, Chichester, England, 2002.

[5] A. Cichocki, R. Zdunek, A. H. Phan, S.-I. Amari, Nonnegative matrix and tensor factorizations. Applications to exploratory multi-way data analysis and blind source separation, Wiley, Chichester, UK, 2009.

[6] P. Comon and C. Jutten Eds, “Handbook of blind source separation. Independent component analysis and applications”, Academic Press, Oxford, UK, 2010.

[7] Y. Deville, ”Matrix factorization for bilinear blind source separation: methods, separability and conditioning”, Proceedings of the 23rd European Signal Processing Conference (EUSIPCO 2015), pp. 1945-1949, Nice, France, Aug. 31 - Sept. 4, 2015.

[8] Y. Deville, “Blind source separation and blind mixture identification methods”, Wiley Encyclopedia of Electrical and Electronics Engineering, pp. 1-33, J. Webster (ed.), Wiley, 2016.

[9] Y. Deville, C. Revel, V. Achard, X. Briottet, “Application and extension of PCA concepts to blind unmixing of hyperspectral data with intra-class variability”, “Advances in Principal Component Analysis - Research and Development”, pp 225-252, G. R. Naik (ed), Springer, Singapore, 2018.

[10] Y. Deville, “From separability/identifiability properties of bilinear and linear-quadratic mixture matrix factorization to factorization algorithms”, Digital Signal Processing, vol. 87, pp. 21-33, April 2019.

[11] Y. Deville, L. T. Duarte, S. Hosseini, “Nonlinear blind source separation and blind mixture identification. Methods for bilinear, linear-quadratic and polynomial mixtures”, SpringerBriefs in Electrical and Computer Engineering, Springer Nature, 2021.

[12] Y. Deville, A. Deville, “New single-preparation methods for unsupervised quantum machine learning problems”, IEEE Transactions on Quantum Engineering, vol. 2, 2021, Article Sequence Number: 3104224, pp. 1-24. DOI: 10.1109/TQE.2021.3121797

[13] Y. Deville, G. Faury, V. Achard, X. Briottet, “An NMF-based method for jointly handling mixture nonlinearity and intraclass variability in hyperspectral blind source separation”, Digital Signal Processing, vol. 133, March 2023, paper no. 103838. DOI: https://doi.org/10.1016/j.dsp.2022.103838

[14] D. Donoho, V. Stodden, “When does Non-Negative Matrix Factorization give a correct decomposition into parts?” Advances in Neural Information Processing Systems 16 (Neural Information Processing Systems, NIPS 2003), Vancouver and Whistler, Canada, December 8-13, 2003.

[15] J. Eggert, E. Korner, “Sparse coding and NMF”, Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, vol. 4, pp. 2529-2533, Budapest, Hungary, 2004.

[16] N. Gillis, “Sparse and Unique Nonnegative Matrix Factorization Through Data Preprocessing”, Journal of Machine Learning Research, vol. 13, pp. 3349-3386, 2012.

[17] P. Hoyer, “Non-negative matrix factorization with sparseness constraints”, Journal of Machine Learning Research, vol. 5, pp. 1457-1469, 2004.

[18] K. Huang, N. D. Sidiropoulos, A. Swami, “Non-Negative Matrix Factorization Revisited: Uniqueness and Algorithm for Symmetric Decomposition”, IEEE Transactions on Signal Processing, vol. 62, no. 1, pp. 211-224, Jan. 1, 2014.

[19] A. Hyvärinen, J. Karhunen, E. Oja, “Independent Component Analysis”, Wiley, New York, 2001.

[20] A. Janecek, Y. Tan, “Using population based algorithms for initializing Nonnegative Matrix Factorization”. In: Y. Tan, Y. Shi, Y. Chai, G. Wang (eds), Advances in Swarm Intelligence (ICSI 2011). Lecture Notes in Computer Science, vol 6729. Springer, Berlin, Heidelberg, 2011.

[21] B. Koirala, Z. Zahiri, A. Lamberti, P. Scheunders, “Robust supervised method for nonlinear spectral unmixing accounting for endmember variability”, IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 9, pp. 7434-7448, Sept. 2021.

[22] B. Koirala, B. Rasti, Z. Bnoulkacem, P. Scheunders, “Nonlinear spectral unmixing using Bézier Surfaces”, IEEE Transactions on Geoscience and Remote Sensing, vol. 62, paper no. 5522416, 2024.

[23] F. A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, P. J. Barloon, A. F. H. Goetz, “The spectral image processing system (SIPS) - Interactive visualization and analysis of imaging spectrometer data”, Remote Sensing of Environment, vol. 44, pp. 145-163, 1993.

[24] A. N. Langville, C. D. Meyer, R. Albright, “Initializations for the Nonnegative Matrix Factorization”, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2006), 2006.

[25] D. D. Lee, H. S. Seung, “Learning the parts of objects by non-negative matrix factorization”, Nature, vol. 401, pp. 788-791, 1999.

[26] D. D. Lee, H. S. Seung, “Algorithms for non-negative matrix factorization”, Adv. Neural Info. Proc. Syst. 13, pp. 556-562, 2001.

[27] C.-J. Lin, “Projected gradient methods for Nonnegative Matrix Factorization”, Neural Computation, vol. 19, pp. 2756-2779, 2007.

[28] L3HARRIS Geospatial, ENVI software,
https://www.l3harrisgeospatial.com/docs/
spectralanglemapper.html

[29] S. Makino, T.-W. Lee, H. Sawada (Eds), “Blind speech separation”, Springer, Dordrecht, The Netherlands, 2007.

[30] I. Meganem, P. Déliot, X. Briottet, Y. Deville, S. Hosseini, ”Linear-quadratic mixing model for reflectances in urban environments”, IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 1, pp. 544-558, Jan. 2014.

[31] I. Meganem, Y. Deville, S. Hosseini, P. Déliot, X. Briottet, ”Linear-quadratic blind source separation Using NMF to unmix urban hyperspectral images”, IEEE Transactions on Signal Processing, vol. 62, no. 7, pp. 1822-1833, April 1, 2014.

[32] 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 ’05), pp. V-289 - V-292, Philadelphia, USA, March 19-23, 2005.

[33] J. M. P. Nascimento, J. M. Bioucas Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data”, IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 4, pp. 898-910, April 2005.

[34] K. O’Hanlon, M. D. Plumbley, “Learning overcomplete dictionaries with L0-sparse Non-negative Matrix Factorisation”, Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, pp. 977-980, Austin, TX, USA, 2013.

[35] P. Paatero, U. Tapper, P. Aalto, M. Kulmala, “Matrix factorization methods for analysing difussion battery data”, J. Aerosol Sci., vol. 22, suppl. 1, pp. S273-S276, 1991.

[36] C. Revel, Y. Deville, V. Achard, X. Briottet, C. Weber, “Inertia-constrained pixel-by-pixel nonnegative matrix factorisation: a hyperspectral unmixing method dealing with intra-class variability”, Remote Sensing, Vol. 10, Issue 11, 1706, Nov. 2018.

[37] A. Smilde, R. Bro, P. Geladi, “Multi-way analysis with applications in the chemical sciences”, Wiley, Chichester, England, 2004.

[38] Z. Wang, M. He, L. Wang, K. Xu, J. Xiao, Y. Nian, “Semi-NMF-based reconstruction for hyperspectral compressed sensing”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 4352-4368, 2020.

[39] T. Yang, S. Li, M. Song, C. Yu, H. Bao, “Low-rank and sparse NMF based on compression and correlation sensing for hyperspectral unmixing”, Infrared Physics & Technology, vol. 141, paper 105464, 2024.

[40] B. Yang, Z. Yin, “Spectral variability augmented multilinear mixing model for hyperspectral nonlinear unmixing”, IEEE Geoscience and Remote Sensing Letters, vol. 21, paper no. 5510405, 2024.

[41] Z. Yin, B. Yang, “Unsupervised nonlinear hyperspectral unmixing with reduced spectral variability via superpixel-based Fisher transformation”, Remote Sensing, vol. 15, paper no. 508, 2023.

[42] J. Yoo, S. Choi, “Matrix co-factorization on compressed sensing”, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1595-1602, Barcelona, Catalonia, Spain, 16-22 July 2011.

[43] A. Zare and K.C. Ho, “Endmember variability in hyperspectral analysis”, IEEE Signal Processing Magazine, vol. 31, no. 1, pp. 95-104, Jan. 2014.