Matrix cofactorization for joint representation learning and supervised classification : application to hyperspectral image analysis

Adrien Lagrange, Mathieu Fauvel, Stéphane May, José M. Bioucas-Dias, Nicolas Dobigeon

Matrix cofactorization for joint representation learning and supervised classification – Application to hyperspectral image analysis☆

Adrien Lagrange[a,*], Mathieu Fauvel[b], Stéphane May[c], José Bioucas-Dias[e], Nicolas Dobigeon[a,d]

[a] University of Toulouse, IRIT/INP-ENSEEIHT Toulouse, BP 7122, Toulouse Cedex 7 31071, France
[b] CESBIO, University of Toulouse, CNES/CNRS/INRA/IRD/UPS, BPI 2801, Toulouse Cedex 9 31401, France
[c] CNES, DCT/SI/AP, 18 Avenue Edouard Belin, Toulouse 31400, France
[d] Institut Universitaire de France, France
[e] Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Lisbon 1049-001, Portugal

A B S T R A C T

Supervised classification and representation learning are two widely used classes of methods to analyze multivariate images. Although complementary, these methods have been scarcely considered jointly in a hierarchical modeling. In this paper, a method coupling these two approaches is designed using a matrix cofactorization formulation. Each task is modeled as a factorization matrix problem and a term relating both coding matrices is then introduced to drive an appropriate coupling. The link can be interpreted as a clustering operation over the low-dimensional representation vectors. The attribution vectors of the clustering are then used as features vectors for the classification task, i.e., the coding vectors of the corresponding factorization problem. A proximal gradient descent algorithm, ensuring convergence to a critical point of the objective function, is then derived to solve the resulting non-convex non-smooth optimization problem. An evaluation of the proposed method is finally conducted both on synthetic and real data in the specific context of hyperspectral image interpretation, unifying two standard analysis techniques, namely unmixing and classification.

Keywords:
Image interpretation
Supervised learning
Representation learning
Hyperspectral images
Non-convex optimization
Matrix cofactorization

1. Introduction

Numerous frameworks have been developed to efficiently analyze the increasing amount of remote sensing images [1,2]. Among those methods, supervised classification has received considerable attention leading to the development of current state-of-the-art classification methods based on advanced statistical tools, such as convolutional neural networks [3–5], kernel methods [6], random forest [7] or Bayesian models [8]. In the context of remote sensing image classification, these methods aim at retrieving the class of each pixel of the image given a specific class nomenclature. Within a supervised framework, a set of pixels is assumed to be annotated by an expert and subsequently used as examples through a learning process. Thanks to extensive research efforts of the community, classification methods have become very efficient. Nevertheless, they still face some challenging issues, such as the high dimension of the data, often coupled with the lack of training data [9]. Handling multi-modal and/or composite classes with intrinsic intra-variability is also a recurrent issue [10]: for instance, a class referred to as *building* can gather very dissimilar samples when metallic and tiled roofs are present in a scene. Besides, the resulting classification remains a high-level interpretation of the scene since it only gives a single class to summarize all information in a given pixel.

Hence, more recent works have emerged in order to provide a richer interpretation [11,12]. In particular, representation learning methods assume that the data results from the composition of a reduced number of elementary patterns. More precisely, the observed measurements can be approximated by mixtures of dictionary elements able to simultaneously capture the variability and redundancy in the dataset. Representation learning can be tackled from different perspectives, in particular known as dictionary

☆ Part of this work has been supported by Centre National d'Études Spatiales (CNES), Occitanie Region, EU FP7 through the ERANETMED JC-WATER program (project ANR-15-NMED-0002-02 MapInvPlnt), by the ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANITI) and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 681839 (project FACTORY)

* Corresponding author.
E-mail addresses: adrien.lagrange@enseeiht.fr (A. Lagrange), mathieu.fauvel@inra.fr (M. Fauvel), stephane.may@cnes.fr (S. May), bioucas@lx.it.pt (J. Bioucas-Dias), nicolas.dobigeon@enseeiht.fr (N. Dobigeon).

learning [13], source separation [14], compressive sensing [15], factor analysis [16], matrix factorization [17] or subspace learning [18]. Various models have been proposed to learn a dedicated representation relevant to the field of interest, differing by specific assumptions and/or constraints. Most of them attempt to identify a dictionary and a mixture function by minimizing a reconstruction error measuring the discrepancy between the chosen model and the dataset. For instance, non-negative matrix factorization (NMF) aims at recovering a linear mixture of non-negative elements with non-negative activation coefficients leading to additive part-based decompositions of the observations [19,20]. Contrary to a classification task, representation learning methods have generally the great advantage of being unsupervised. However, for particular purposes, they can be specialized to learn a representation suited for a particular task, e.g. classification or regression [21]. Thus, representation learning provides a rich yet compact description of the data whereas supervised classification offers a univocal interpretation based on prior knowledge from experts.

The idea of combining the representation learning and classification tasks has already been considered, mostly to use the representation learning method as a dimensionality reduction step prior to the classification [22], where the low-dimensional representation is used as input features. Nonetheless, some works introduce the idea of performing the two tasks simultaneously [23]. For example, the discriminative K-SVD algorithm associates a linear mixture model to a linear classifier [24]. At the end, the method tries to learn a dictionary well-fitted for the classification task, i.e., the learned representation minimizes the reconstruction error but also ensures a good separability of the classes. More intertwined frameworks can be also considered, as the one proposed in [25] where elements of the dictionary are class-specific. Joint representation learning and classification can be cast as a cofactorization problem. Both tasks are interpreted as individual factorization problems and constraints between the dictionaries and coding matrices associated with the two problems can then be imposed. These cofactorization-based models have proven to be highly efficient in many application fields, e.g. for text mining [26], music source separation [27], or image analysis [28,29].

However, most of the available methods tend to focus on classification results and generally oppose reconstruction accuracy and discriminative abilities of the models instead of designing a unifying hierarchical structure. Capitalizing on recent advances and a first attempt in [30] in a Bayesian setting, this paper proposes a particular cofactorization method, with a dedicated application to multivariate image analysis. The representation learning and classification tasks are related through the coding matrices of the two factorization problems. A clustering is performed on the low-dimensional representation and the clustering attribution vectors are used as coding vectors for the classification. This novel coupling approach produces a coherent and fully-interpretable hierarchical model. To solve the resulting non-convex non-smooth optimization problem, a proximal alternating linearized minimization (PALM) algorithm is derived, yielding guarantees of convergence to a critical point of the objective function [31].

The main contributions reported in this paper can be summarized as follows. A generic framework is proposed to demonstrate that two ubiquitous image analysis methods, namely supervised classification and representation learning, can be unified into a unique joint cofactorization problem. This framework is instanced for one particular application in the context of hyperspectral image analysis where supervised classification and spectral unmixing are performed jointly. The proposed method offers a comprehensive and meaningful analysis of the image as well as competitive quantitative results for the two considered tasks.

This paper is organized as follows. Section 2 defines the two factorization problems used to perform representation learning and classification and further discusses the joint cofactorization problem. It also details the optimization scheme developed to solve the resulting non-convex minimization problem. To illustrate the generic framework introduced in the previous section, an application to hyperspectral image analysis is conducted in Section 3 through the dual scope of spectral unmixing and classification. Performance of the proposed framework is illustrated thanks to experiments conducted on synthetic and real data in Section 4. Finally, Section 5 concludes the paper and presents some research perspectives to this work.

2. Proposed generic framework

The representation learning and classification tasks are generically defined as factorization matrix problems in Sections 2.1 and 2.2. To derive a unified cofactorization formulation, a third step consists in drawing the link between these two independent problems. In this work, this coupling is ensured by imposing a consistent structure between the two coding matrices corresponding to the low-dimensional representation and the feature matrices, respectively. As detailed in Section 2.3, it is expressed as a clustering task where the parameters describing the attribution to the clusters are the feature vectors, i.e. the coding matrix resulting from the classification task. Particular instances of these three tasks will be detailed in Section 3 for an application to multiband image analysis.

2.1. Representation learning

The fundamental assumption in representation learning is that the P considered L-dimensional samples, gathered in matrix $\mathbf{Y} \in \mathbb{R}^{L\times P}$, belong to a R-dimensional subspace such that $R \ll L$. The aim is then to recover this manifold, where samples can be expressed as combinations of elementary vectors, herein the column of the matrix $\mathbf{W} \in \mathbb{R}^{L\times R}$ sometimes referred to as dictionary. These samples can be subsequently represented thanks to the so-called coding matrix $\mathbf{H} \in \mathbb{R}^{R\times P}$. Formally, identifying the dictionary and the coding matrices can be generally expressed as a minimization problem

$$\min_{\mathbf{W},\mathbf{H}} \mathcal{J}_{\mathrm{r}}(\mathbf{Y}|\psi(\mathbf{W},\mathbf{H})) + \lambda_w \mathcal{R}_w(\mathbf{W}) + \imath_{\mathbb{W}}(\mathbf{W}) + \lambda_h \mathcal{R}_h(\mathbf{H}) + \imath_{\mathbb{H}}(\mathbf{H}) \tag{1}$$

where $\psi(\cdot)$ is a mixture function (e.g., linear or bilinear operator), $\mathcal{J}_{\mathrm{r}}(\cdot)$ is an appropriate cost function, for example derived from a β-divergence [32], $\mathcal{R}_{\cdot}(\cdot)$ denote penalizations weighted by the parameter $\lambda_{\cdot}$ and $\imath_{\cdot}(\cdot)$ is the indicator functions defined here on the respective sets $\mathbb{W} \subset \mathbb{R}^{L\times R}$ and $\mathbb{H} \subset \mathbb{R}^{R\times P}$ imposing some constraints on the dictionary and coding matrices.

In the case of a linear embedding adopted in this work, the mixture function writes

$$\psi(\mathbf{W},\mathbf{H}) = \mathbf{W}\mathbf{H}. \tag{2}$$

In this context, the problem (1) can be cast as a factor analysis driven by the cost function $\mathcal{J}_{\mathrm{r}}(\cdot)$. Depending on the applicative field, typical data-fitting measures include the Itakura-Saito, the Euclidean and the Kullback–Leibler divergences [32]. Assuming a low-rank model (i.e., $R \leq L$), specific choices for the sets $\mathbb{H}$ and $\mathbb{W}$ lead to various standard factor models. For instance, when $\mathbb{W}$ is chosen as the Stiefel manifold, the solution of (1) is given by a principal component analysis (PCA) [33]. When $\mathbb{W}$ and $\mathbb{H}$ impose nonnegativity of the dictionary and coding matrix elements, the problem is known as nonnegative matrix factorization [19,34].

Within a supervised context, the dictionary $\mathbf{W}$ can be chosen thanks to a end-user expertise or estimated beforehand. Without loss of generality but for the sake of conciseness, the framework

described in this paper assumes that this dictionary is known, possibly overcomplete as proposed in the experimental illustration described in Section 4. In this case, as in many applications, it makes sense to look for a sparse representation of the signal of interest to retrieve its most achievable compact representation [21,35]. Following this strategy, we propose to consider an ℓ_1-norm sparsity penalization on the coding vectors, leading to representation learning task defined by

$$\min_{\mathbf{H}} \mathcal{J}_{\mathrm{r}}(\mathbf{Y}|\mathbf{W}\mathbf{H}) + \lambda_h \|\mathbf{H}\|_1 + \iota_{\mathbb{H}}(\mathbf{H}) \tag{3}$$

where $\|\mathbf{H}\|_1 = \sum_{p=1}^{P} \|\mathbf{h}_p\|_1$ with $\mathbf{h}_p$ denoting the pth column of $\mathbf{H}$.

2.2. Supervised classification

To clearly define the classification task, let first introduce some key notations. The index subset of samples with an available groundtruth is denoted as $\mathcal{L}$ while the index subset of unlabeled samples is $\mathcal{U}$ such that $\mathcal{L} \cap \mathcal{U} = \emptyset$ and $\mathcal{L} \cup \mathcal{U} = \mathcal{P}$ with $\mathcal{P} \triangleq \{1, \ldots, P\}$. Classifying the unlabeled samples consists in assigning each of them to one of the C classes. This can be reformulated as the estimation of a $C \times P$ matrix $\mathbf{C}$ whose columns correspond to unknown C-dimensional attribution vectors $\mathbf{c}_p = \left[c_{1,p}, \ldots, c_{C,p}\right]^T$. Each vector is made of 0 except for $c_{i,p} = 1$ when the pth sample is assigned the ith class.

Numerous classification rules have been proposed in the literature [36]. Most of them rely on a $K \times P$ matrix $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_P]$ of features $\mathbf{z}_p$ $(p \in \mathcal{P})$ associated with each sample and derived from the raw data. Within a supervised framework, the attribution matrix $\mathbf{C}_{\mathcal{L}}$ and feature matrix $\mathbf{Z}_{\mathcal{L}}$ of the labeled data are exploited during the learning step, where $\cdot_{\mathcal{L}}$ denotes the corresponding submatrix whose columns are indexed by $\mathcal{L}$. For a wide range of classifiers, deriving a classification rule can be achieved by solving the optimization problem

$$\min_{\mathbf{Q}} \mathcal{J}_{\mathrm{c}}(\mathbf{C}_{\mathcal{L}}|\phi(\mathbf{Q}, \mathbf{Z}_{\mathcal{L}})) + \lambda_q \mathcal{R}_q(\mathbf{Q}) \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{C \times K}$ is the set of classifier parameters to be inferred, $\mathcal{R}_q(\cdot)$ refer to regularizations imposed on $\mathbf{Q}$ and $\mathcal{J}_{\mathrm{c}}$ is a cost function measuring the quality of the classification such as the quadratic loss [24] or cross-entropy [37]. Moreover, in (4), $\phi(\mathbf{Q}, \cdot)$ defines a element-wise nonlinear mapping between the features and the class attribution vectors parametrized by $\mathbf{Q}$, e.g., derived from a sigmoid or a softmax operators. In this work, the classifier is assumed to be linear, which leads to a vector-wise post-nonlinear mapping

$$\phi(\mathbf{Q}, \mathbf{Z}_{\mathcal{L}}) = \phi(\mathbf{Q}\mathbf{Z}_{\mathcal{L}}) \tag{5}$$

with

$$\phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_p)]. \tag{6}$$

Once the classifier parameters have been estimated by solving (4), the unknown attribution vectors $\mathbf{C}_{\mathcal{U}}$ can be subsequently inferred during the testing step by applying the nonlinear transformation to the corresponding predicted features $\hat{\mathbf{Z}}_{\mathcal{U}}$ associated with the unlabeled samples. The obtained outputs are relaxed attribution vectors $\hat{\mathbf{c}}_p = \phi(\mathbf{Q}\hat{\mathbf{z}}_p)$ $(p \in \mathcal{U})$ and the most probable predicted sample class can be computed as $\mathrm{argmax}_i\, c_{i,p}$.

Under the proposed formulation of the classification task, the learning and testing steps can be conducted simultaneously, a framework usually referred to as semi-supervised, with the beneficial opportunity to introduce additional regularizations and/or constraints on the submatrix of unknown attribution vectors $\mathbf{C}_{\mathcal{U}}$. The initial problem (4) is thus extended to the following one

$$\min_{\mathbf{Q}, \mathbf{C}_{\mathcal{U}}} \mathcal{J}_{\mathrm{c}}(\mathbf{C}|\phi(\mathbf{Q}\mathbf{Z})) + \lambda_q \mathcal{R}_q(\mathbf{Q}) + \lambda_c \mathcal{R}_c(\mathbf{C}) + \iota_{\mathbb{C}}(\mathbf{C}_{\mathcal{U}}) \tag{7}$$

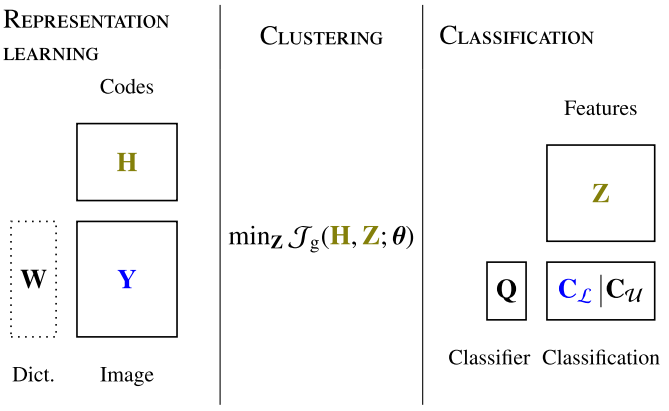


Fig. 1. Structure of the cofactorization model. Variables in *blue* stand for observations or available external data. Variables in *olive green* are linked through the clustering task here formulated as an optimization problem. The variable in a dotted box is assumed to be known or estimated beforehand in this work.

where $\mathbf{C} = [\mathbf{C}_{\mathcal{L}}\ \mathbf{C}_{\mathcal{U}}]$ and $\mathbb{C} \subset \mathbb{R}^{C \times |\mathcal{U}|}$ denotes a feasible set for the attribution matrix $\mathbf{C}_{\mathcal{U}}$. As discussed above, the cost function $\mathcal{J}_{\mathrm{c}}(\mathbf{C}|\hat{\mathbf{C}})$ measures the actual classification loss, i.e., the discrepancy between the attribution vector $\mathbf{C}$ of the training set and the attribution vectors $\hat{\mathbf{C}}$ predicted by the classifier. Two particular cases fitting this generic model are provided in Sections 3.2.1 and 3.2.2. The attribution vectors are defined as $\hat{\mathbf{C}} = \phi(\mathbf{Q}\hat{\mathbf{Z}})$ where $\phi(\cdot)$ is a nonlinear function applied to the output of a linear classifier. The regularization term $\mathcal{R}_q(\mathbf{Q})$ penalizes over the parameters of the classifiers. A typical example is a quadratic penalization which aims at avoiding overfitting, as conventionally done when optimizing neural networks and generally referred to as *weight decay* [38]. Finally, the regularization term $\mathcal{R}_c(\mathbf{C})$ penalizes over the attribution matrix. Typical examples include spatial regularizations such as total variation (TV) when dealing with image classification. The indicator function $\iota_{\mathbb{C}}(\mathbf{C}_{\mathcal{U}})$ enforces sum-to-one and non-negativity constraints such that each attribution vector $\mathbf{c}_p$ $(p \in \mathcal{U})$ can then be interpreted as a probability vector of belonging to each class. In such a case, the feasible set is chosen as $\mathbb{C} = \mathbb{S}_C^{|\mathcal{U}|}$ where

$$\mathbb{S}_C \triangleq \left\{ \mathbf{u} \in \mathbb{R}^C \middle| \forall k,\ u_k \geq 0 \text{ and } \sum_{k=1}^{C} u_k = 1 \right\}. \tag{8}$$

2.3. Coupling representation learning and classification

Up to this point, the representation learning and supervised classification tasks have been formulated as two independent matrix factorization problems given by (3) and (7), respectively. This work proposes to join them by drawing an implicit relation between two factors involved in these two problems. Inspired by hierarchical Bayesian models such as the one proposed in [30], both problems are coupled through the activation matrices $\mathbf{H}$ and $\mathbf{Z}$, as illustrated in Fig. 1. More precisely, the coding vectors in $\mathbf{H}$ are clustered such that the feature vectors in $\mathbf{Z}$ are defined as the attribution vectors to the K clusters. Ideally, clustering attribution vectors $\mathbf{z}_p$ are filled with zeros except for $z_{k,p} = 1$ when $\mathbf{h}_p$ is associated with the kth cluster. Thus, the vectors $\mathbf{z}_p$ $(p \in \mathcal{P})$ are assumed to be defined on the K-dimensional probability simplex $\mathbb{S}_K$ similarly defined as (8) and ensuring non-negativity and sum-to-one constraints. Many clustering algorithms can be expressed as optimization problem such as the well-known k-means algorithm and many of its variants [39,40]. Adopting this formulation, and denoting $\boldsymbol{\theta}$ the set of parameters of the clustering algorithm, the

Table 1
Overview of notations.

	Parameter
$P \in \mathbb{R}$	Number of observations
$L \in \mathbb{R}$	Dimension of observations
$C \in \mathbb{R}$	Number of classes
$K \in \mathbb{R}$	Number of features/clusters
$\mathcal{P} = \{1, \ldots, P\}$	Index set of observations
$\mathcal{L} \subset \mathcal{P}$	Index set of labeled samples
$\mathcal{L}_i \subset \mathcal{L}$	Index set of labeled samples in the ith class
$\mathcal{U} = \mathcal{P} \backslash \mathcal{L}$	Index set of unlabeled samples
$\mathbf{Y} \in \mathbb{R}^{L \times P}$	Observations
$\mathbf{W} \in \mathbb{R}^{L \times R}$	Dictionary
$\mathbf{H} \in \mathbb{R}^{R \times P}$	Coding matrix
$\mathbf{Q} \in \mathbb{C}^{C \times P}$	Classifier parameters
$\mathbf{C}_{\mathcal{L}} \in \mathbb{R}^{C \times \lvert\mathcal{L}\rvert}$	Attribution matrix of labeled data
$\mathbf{C}_{\mathcal{U}} \in \mathbb{R}^{C \times \lvert\mathcal{U}\rvert}$	Attribution matrix of unlabeled data
$\mathbf{C} = [\mathbf{C}_{\mathcal{L}}\ \mathbf{C}_{\mathcal{U}}]$	Class attribution matrix
$\mathbf{Z} \in \mathbb{R}^{K \times P}$	Cluster attribution matrix
$\boldsymbol{\theta} \in \boldsymbol{\Theta}$	Clustering parameters

clustering task can be defined as the minimization problem

$$\min_{\mathbf{Z},\boldsymbol{\theta}} \mathcal{J}_g(\mathbf{H}, \mathbf{Z}; \boldsymbol{\theta}) + \lambda_z \mathcal{R}_z(\mathbf{Z}) + \lambda_\theta \mathcal{R}_\theta(\boldsymbol{\theta}) + \imath_{\mathbb{S}_K^P}(\mathbf{Z}) + \imath_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \tag{9}$$

where $\boldsymbol{\Theta}$ defines a feasible set for the parameters $\boldsymbol{\theta}$.

It is worth noting that introducing this coupling term is one of the major novelty of the proposed approach. When considering task-driven dictionary learning methods, it is usual to intertwine the representation learning and the classification tasks by directly imposing $\mathbf{H} = \mathbf{Z}$ [24,41]. Since these methods generally rely on a linear classifier, one major drawback of such approaches is their unability to deal with non-separable classes in the low-dimensional representation space. In such cases, the underlying model cannot be discriminative and descriptive simultaneously and the resulting tasks become adversarial. When considering the proposed coupling term, the cluster attribution vectors $\mathbf{z}_p$ offer the possibility of linearly separating any group of clusters from the others. As a consequence, the model benefits from more flexibility, with both discriminative and descriptive abilities in a more general sense.

2.4. Global cofactorization problem

Unifying the representation learning task (3) and the classification task (7) through the clustering task (9) leads to the following joint cofactorization problem

$$\begin{aligned}\min_{\substack{\mathbf{H},\mathbf{Q},\mathbf{C}_{\mathcal{U}},\\ \mathbf{Z},\boldsymbol{\theta}}}\ & \lambda_0 \mathcal{J}_r(\mathbf{Y}|\mathbf{WH}) + \lambda_h \|\mathbf{H}\|_1 \\ & + \lambda_1 \mathcal{J}_c(\mathbf{C}|\phi(\mathbf{QZ})) + \lambda_q \mathcal{R}_q(\mathbf{Q}) + \lambda_c \mathcal{R}_c(\mathbf{C}) \\ & + \lambda_2 \mathcal{J}_g(\mathbf{H}, \mathbf{Z}; \boldsymbol{\theta}) + \lambda_z \mathcal{R}_z(\mathbf{Z}) + \lambda_\theta \mathcal{R}_\theta(\boldsymbol{\theta}) \\ & + \imath_{\mathbb{H}}(\mathbf{H}) + \imath_{\mathbb{S}_K^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}}) + \imath_{\mathbb{S}_K^P}(\mathbf{Z}) + \imath_{\boldsymbol{\Theta}}(\boldsymbol{\theta})\end{aligned} \tag{10}$$

where λ_0, λ_1 and λ_2 control the respective contribution of each task data-fitting term. All notations and parameter dimensions are summarized in Table 1. A generic algorithmic scheme solving the problem (10) is proposed in the next section.

2.5. Optimization scheme

The minimization problem defined by (10) is not globally convex. To reach a local minimizer, we propose to resort to the proximal alternating linearized minimization (PALM) algorithm introduced in [31]. This algorithm is based on proximal descent steps, which allows non-smooth terms to be handled. Moreover it is guaranteed to converge to a critical point of the objective function even in the case of non-convex problem. This means that, if the initialization is good enough, it is expected to likely converge to a solution close to the global optimum. To implement PALM, the problem (10) is rewritten in the form of an unconstrained problem expressed as a sum of a smooth coupling term $g(\cdot)$ and separable non-smooth terms $f_j(\cdot)$ ($j \in \{0, \ldots, 4\}$) as follows

$$\min_{\substack{\mathbf{H},\boldsymbol{\theta},\mathbf{Z},\\ \mathbf{Q},\mathbf{C}_{\mathcal{U}}}} f_0(\mathbf{H}) + f_1(\boldsymbol{\theta}) + f_2(\mathbf{Z}) + f_3(\mathbf{C}_{\mathcal{U}}) + g(\mathbf{H}, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q}) \tag{11}$$

where

$$f_0(\mathbf{H}) = \imath_{\mathbb{H}}(\mathbf{H}) + \lambda_h \|\mathbf{H}\|_1 \qquad f_2(\mathbf{Z}) = \imath_{\mathbb{S}_K^P}(\mathbf{Z})$$
$$f_1(\boldsymbol{\theta}) = \imath_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \qquad f_3(\mathbf{C}_{\mathcal{U}}) = \imath_{\mathbb{S}_K^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}})$$

and the coupling function is

$$\begin{aligned} g(\mathbf{H}, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q}) = {} & \lambda_0 \mathcal{J}_r(\mathbf{Y}|\mathbf{WH}) \\ & + \lambda_1 \mathcal{J}_c(\mathbf{C}|\phi(\mathbf{QZ})) + \lambda_q \mathcal{R}_q(\mathbf{Q}) + \lambda_c \mathcal{R}_c(\mathbf{C}) \\ & + \lambda_2 \mathcal{J}_g(\mathbf{W}, \mathbf{Z}; \boldsymbol{\theta}) + \lambda_z \mathcal{R}_z(\mathbf{Z}) + \lambda_\theta \mathcal{R}_\theta(\boldsymbol{\theta}). \end{aligned} \tag{12}$$

To ensure the stated guarantees of PALM, all $f_j(\cdot)$ have to be proper, lower semi-continuous function $f_j : \mathbb{R}^{n_j} \to (-\infty, +\infty]$, which ensures in particular that the associated proximal operator is well-defined. Additionally, sufficient conditions on the coupling function are that $g(\cdot)$ is a $\mathcal{C}^2$ function (i.e., with continuous first and second derivatives) and that its partial gradients are globally Lipschitz. For example, partial gradient $\nabla_{\mathbf{H}} g(\mathbf{H}, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q})$ should be globally Lipschitz for any fixed $\boldsymbol{\theta}$, $\mathbf{Z}$, $\mathbf{C}_{\mathcal{U}}$, $\mathbf{Q}$, that is

$$\begin{aligned} & \left\| \nabla_{\mathbf{H}} g(\mathbf{H}_1, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q}) - \nabla_{\mathbf{H}} g(\mathbf{H}_2, \boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q}) \right\| \\ & \quad \leq L_{\mathbf{H}}(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q}) \|\mathbf{H}_1 - \mathbf{H}_2\|, \quad \forall \mathbf{H}_1, \mathbf{H}_2 \in \mathbb{R}^{R \times P} \end{aligned} \tag{13}$$

where $L_{\mathbf{H}}(\boldsymbol{\theta}, \mathbf{Z}, \mathbf{C}_{\mathcal{U}}, \mathbf{Q})$, simply denoted $L_{\mathbf{H}}$ hereafter, is the Lipschitz constant. For sake of conciseness, we refer to [31] to get further details.

The main idea of the algorithm is then to update each variable of the problem alternatively using a proximal gradient descent. The overall scheme is summarized in Algorithm 1 . For a practical implementation, one needs to compute the partial gradients of $g(\cdot)$ explicitly and their Lipschitz constants to perform a gradient descent step, followed by a proximal mapping associated with the non-smooth terms $f_j(\cdot)$. The objective function is then monitored at each iteration and the algorithm is stopped when convergence is reached. Note that, when a specific penalization $\mathcal{R}_\cdot(\cdot)$ is non-smooth or non-gradient-Lipschitz, it is possible to move it into the corresponding independent term $f_j(\cdot)$ to ensure the required property of the coupling function $g(\cdot)$. This is for instance the case for the sparse penalization used over $\mathbf{H}$ which has been moved into $f_0(\cdot)$. Nonetheless, as mentioned above, the proximal operator associated with each $f_j(\cdot)$ is needed. Thus, even when the function consists of several terms, a closed-form expression of this operator should be known. Alternatively, one should be able to compose the proximal operators associated with each term of $f_j(\cdot)$ [42].

Algorithm 1: PALM.

Initialize variables $\mathbf{H}^0$, $\boldsymbol{\theta}^0$, $\mathbf{Z}^0$, $\mathbf{C}_{\mathcal{U}}{}^0$ and $\mathbf{Q}^0$;
Set $\alpha > 1$;
while *stopping criterion not reached* **do**
  $\mathbf{H}^{k+1} \in \operatorname{prox}_{f_0}^{\alpha L_{\mathbf{H}}}(\mathbf{H}^k - \frac{1}{\alpha L_{\mathbf{H}}} \nabla_{\mathbf{H}} g(\mathbf{H}^k, \boldsymbol{\theta}^k, \mathbf{Z}^k, \mathbf{C}_{\mathcal{U}}^k, \mathbf{Q}^k))$;
  $\boldsymbol{\theta}^{k+1} \in \operatorname{prox}_{f_1}^{\alpha L_{\boldsymbol{\theta}}}(\boldsymbol{\theta}^k - \frac{1}{\alpha L_{\boldsymbol{\theta}}} \nabla_{\boldsymbol{\theta}} g(\mathbf{H}^{k+1}, \boldsymbol{\theta}^k, \mathbf{Z}^k, \mathbf{C}_{\mathcal{U}}^k, \mathbf{Q}^k))$;
  $\mathbf{Z}^{k+1} \in \operatorname{prox}_{f_2}^{\alpha L_{\mathbf{Z}}}(\mathbf{Z}^k - \frac{1}{\alpha L_{\mathbf{Z}}} \nabla_{\mathbf{Z}} g(\mathbf{H}^{k+1}, \boldsymbol{\theta}^{k+1}, \mathbf{Z}^k, \mathbf{C}_{\mathcal{U}}^k, \mathbf{Q}^k))$;
  $\mathbf{Q}^{k+1} \in \operatorname{prox}_{f_3}^{\alpha L_{\mathbf{Q}}}(\mathbf{Q}^k - \frac{1}{\alpha L_{\mathbf{Q}}} \nabla_{\mathbf{Q}} g(\mathbf{H}^{k+1}, \boldsymbol{\theta}^{k+1}, \mathbf{Z}^{k+1}, \mathbf{C}_{\mathcal{U}}{}^k, \mathbf{Q}^k))$;
  $\mathbf{C}_{\mathcal{U}}^{k+1} \in \operatorname{prox}_{f_4}^{\alpha L_{\mathbf{C}_{\mathcal{U}}}}(\mathbf{C}_{\mathcal{U}}^k - \frac{1}{\alpha L_{\mathbf{C}_{\mathcal{U}}}} \nabla_{\mathbf{C}_{\mathcal{U}}} g(\mathbf{H}^{k+1}, \boldsymbol{\theta}^{k+1}, \mathbf{Z}^{k+1}, \mathbf{C}_{\mathcal{U}}^k, \mathbf{Q}^{k+1}))$;
end
return $\mathbf{H}^{end}, \boldsymbol{\theta}^{end}, \mathbf{Z}^{end}, \mathbf{Q}^{end}, \mathbf{C}_{\mathcal{U}}^{end}$

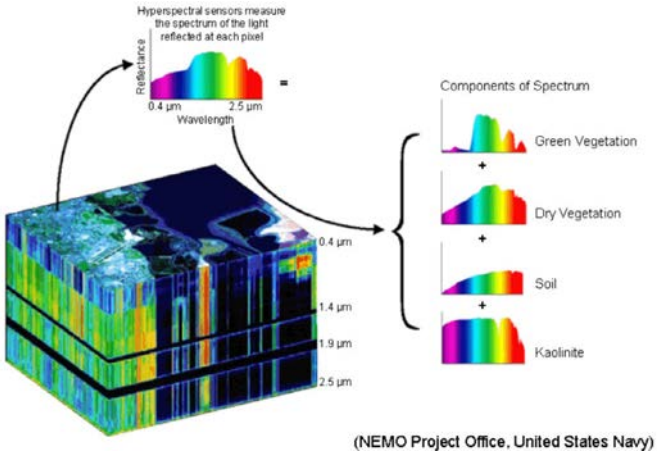


Fig. 2. Spectral unmixing concept (source US Navy NEMO).

3. Application: hyperspectral images analysis

A general framework has been introduced in the previous section. As an illustration, a particular instance of this generic framework is now considered, where explicit representation learning, classification and clustering are introduced. The specific case of hyperspectral images analysis is considered for this use case example.

Contrary to conventional color imaging which only captures the reflectance measure for three wavelengths (red, blue, green), hyperspectral imaging makes it possible to measure reflectance of the observed scene for several hundreds of wavelengths from visible to invisible domain. Each pixel of the image can thus be represented as a vector of reflectance, called spectrum, which characterizes the observed material.

One drawback of hyperspectral images is usually a weaker spatial resolution due to sensor limitations. The direct consequence of this poor spatial resolution is the presence of mixed pixels, i.e., pixels corresponding to areas containing several materials. Observed spectra are in this case the result of a specific mixture of the elementary spectra, called endmembers, associated with individual materials present in the pixel. The problem of retrieving the proportions of each material in each pixel is referred to as spectral unmixing [11]. This problem can be seen as a specific case of representation learning where the dictionary is composed of the set of endmembers standing for the endmember spectra and the coding matrix is the so-called abundance matrix containing the proportion of each material in each pixel.

Spectral unmixing is introduced as a representation learning task in Section 3.1. The specific classifier used for this application is then explained in Section 3.2 and finally Section 3.3 presents the clustering adopted to relate the abundance matrix and the classification feature matrix.

3.1. Spectral unmixing

As explained, each pixel of an hyperspectral image is characterized by a reflectance spectrum that physics theory approximates as a combination of endmembers, each corresponding to a specific material, as illustrated in Fig. 2. Formally, in this applicative scenario, the L-dimensional sample $\mathbf{y}_p$ denotes the L-dimensional spectrum of the pth pixel of the hyperspectral image ($p \in \mathcal{P}$). Each observation vectors $\mathbf{y}_p$ can be expressed as a function of the endmember matrix $\mathbf{W}$ (containing the R elementary spectra) and the abundance vector $\mathbf{h}_p \in \mathbb{R}^R$ with $R \ll L$.

In the case of the most commonly adopted linear mixture model, each observation $\mathbf{y}_p$ is assumed to be a linear combination of the endmember spectra $\mathbf{w}_r$ ($r = 1, \ldots, R$) corrupted by some noise, underlying the linear embedding (2). Assuming a quadratic data-fitting term, the cost function associated with the representation learning task in (1) is written

$$\mathcal{J}_{\mathrm{r}}(\mathbf{Y}|\mathbf{WH}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{WH}\|_{\mathrm{F}}^2. \tag{14}$$

The abundance vector $\mathbf{h}_p$ is usually interpreted as a vector of proportions describing the proportion of each elementary component in the pixel. Thus, to derive an additive composition of the observed pixels, a nonnegative constraint is considered for each element of the abundance matrix $\mathbf{H}$, i.e., $\mathbb{H} = \mathbb{R}_+^{R\times P}$. In this work, no sum-to-one constraint is considered since it has been argued that leaving this constraint out offers a better adaptation to possible changes of illumination in the scene [43]. Additionally, as the endmember matrix $\mathbf{W}$ is the collection of reflectance spectra of the endmembers, it is also expected to be non-negative. When this dictionary needs to be estimated, the resulting problem is a sparse non-negative matrix factorization (NMF) task. When the dictionary is known or estimated beforehand, the resulting optimization problem is the nonnegative sparse coding problem

$$\min_{\mathbf{H}} \frac{1}{2}\|\mathbf{Y} - \mathbf{WH}\|_{\mathrm{F}}^2 + \lambda_h\|\mathbf{H}\|_1 + \iota_{\mathbb{R}_+^{R\times P}}(\mathbf{H}) \tag{15}$$

where the sparsity penalization actually supports the assumption that only a few materials are present in a given pixel.

3.2. Classification

In the considered application, two loss functions associated with the classification problem have been investigated, namely quadratic loss and cross-entropy loss. One advantage of these two loss functions is that they can be used in a multi-class classification (i.e., with more than two classes). Moreover, this choice may fulfill the required conditions stated in Section 2.5 to apply PALM since, coupled with an appropriate $\phi(\cdot)$ function, both loss costs are smooth and gradient-Lipschitz according to each estimated variables.

3.2.1. Quadratic loss

The quadratic loss is the most simple way to perform a classification task and have been extensively used [25,44,45]. It is defined as

$$\mathcal{J}_{\mathrm{c}}(\mathbf{C}|\hat{\mathbf{C}}) = \frac{1}{2}\left\|\mathbf{CD} - \hat{\mathbf{C}}\mathbf{D}\right\|_{\mathrm{F}}^2 \tag{16}$$

where $\hat{\mathbf{C}}$ denotes the estimated attribution matrix. In (16), the $P \times P$ matrix $\mathbf{D}$ is introduced to weight the contribution of the labeled data with respect to the unlabeled one and to deal with the case of unbalanced classes in the training set. Weights are chosen to be inversely proportional to class frequencies in the input data. The weight matrix is defined as the diagonal matrix $\mathbf{D} = \mathrm{diag}[d_1, \ldots, d_P]$ with

$$d_p = \begin{cases} \sqrt{\frac{1}{|\mathcal{L}_i|}}, & \text{if } p \in \mathcal{L}_i; \\ \sqrt{\frac{1}{|\mathcal{U}|}}, & \text{if } p \in \mathcal{U}; \end{cases} \tag{17}$$

where $\mathcal{L}_i$ denotes the set of indexes of labeled pixels of the ith class ($i = 1, \ldots, C$). Thus, considering a linear classifier, the generic classification problem in (7) can be specified for the quadratic loss

$$\min_{\mathbf{Q},\mathbf{C}_{\mathcal{U}}} \frac{1}{2}\|\mathbf{CD} - \mathbf{QZD}\|_{\mathrm{F}}^2 + \lambda_c\mathcal{R}_c(\mathbf{C}) + \iota_{\mathbb{S}_c^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}}) \tag{18}$$

where no additional constraints nor penalization is applied to the classifier parameters $\mathbf{Q}$. Besides, when samples obey a spatially coherent structure, as it is the case when analyzing hyperspectral images, it is often desirable to transfer this structure to the classification map. Such a characteristics can be achieved by considering

a spatial regularization $\mathcal{R}_c(\mathbf{C})$ applied to the attributions vectors. Following this assumption, this work considers a regularized counterpart of the weighted vectorial total variation (vTV), promoting a spatially piecewise constant behavior of the classification map [46]

$$\|\mathbf{C}\|_{\text{vTV}} = \sum_{m,n} \beta_{m,n} \sqrt{\left\|[\nabla_{\text{h}}\mathbf{C}]_{m,n}\right\|_2^2 + \left\|[\nabla_{\text{v}}\mathbf{C}]_{m,n}\right\|_2^2 + \epsilon} \tag{19}$$

where $(m,\ n)$ are the spatial position pixel indexes and $[\nabla_{\text{h}}(\cdot)]_{m,n}$ and $[\nabla_{\text{v}}(\cdot)]_{m,n}$ stand for horizontal and vertical discrete gradient operators evaluated at a given pixel,[1] respectively, i.e.,

$$[\nabla_{\text{h}}\mathbf{C}]_{m,n} = \mathbf{c}_{(m+1,n)} - \mathbf{c}_{(m,n)}$$
$$[\nabla_{\text{v}}\mathbf{C}]_{m,n} = \mathbf{c}_{(m,n+1)} - \mathbf{c}_{(m,n)}.$$

The weights $\beta_{m,n}$ can be computed beforehand to adjust the penalizations with respect to expected spatial variations of the scene. They can be estimated directly from the image to be analyzed or extracted from a complementary dataset as in [47]. They will be specified during the experiments reported in Section 4. Moreover, the smoothing parameter $\epsilon > 0$ ensures the gradient-Lipschitz property of the coupling term $g(\cdot)$, as required in Section 2.5.

3.2.2. Cross-entropy loss

The quadratic loss has the advantage to be expressed simply and the associated Lipschitz constant of the partial gradients are trivially obtained. However, this loss function is known to be highly influenced by outliers which can result in a degraded predictive accuracy [48]. A more sophisticated way to conduct the classification task is to consider a cross-entropy loss

$$\mathcal{J}_{\text{c}}(\mathbf{C}|\hat{\mathbf{C}}) = -\sum_{p\in\mathcal{P}} d_p^2 \sum_{i\in\mathcal{C}} c_{i,p} \log\left(\hat{c}_{i,p}\right) \tag{20}$$

combined with a logistic regression, i.e., where the nonlinear mapping (5) is element-wise defined as

$$[\phi(\mathbf{X})]_{i,j} = \frac{1}{1+\exp(-x_{i,j})} = \text{sigm}(x_{i,j}) \tag{21}$$

with $i \in \{1,\dots,C\}$ and $p \in \mathcal{P}$. This classifier can actually be interpreted as a one-layer neural network with a sigmoid non-linearity. Cross-entropy loss is indeed a very conventional loss function in the neural network/deep learning community [38]. In the present case, the corresponding optimization problem can be written

$$\min_{\mathbf{Q},\mathbf{C}_{\mathcal{U}}} -\sum_{p\in\mathcal{P}} d_p^2 \sum_{i\in\mathcal{C}} c_{i,p} \log\left(\text{sigm}(\mathbf{q}_{i:}\mathbf{z}_p)\right)) + \lambda_q \mathcal{R}_q(\mathbf{Q}) + \lambda_c \|\mathbf{C}\|_{\text{vTV}} + \iota_{\mathbb{S}_C^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}}) \tag{22}$$

where $\mathbf{q}_{i:} \in \mathbb{R}^{1\times K}$ denotes the ith line of the matrix $\mathbf{Q}$. The penalization $\mathcal{R}_q(\mathbf{Q})$ is here chosen as $\mathcal{R}_q(\mathbf{Q}) = \frac{1}{2}\|\mathbf{Q}\|_{\text{F}}^2$ to prevent the loss function to artificially decrease when $\|\mathbf{q}_{i:}\|^2$ is increasing. This regularization has been extensively studied in the neural network literature where it is referred to as *weight decay* [38]. In (22), the regularization $\mathcal{R}_c(\mathbf{C}_{\mathcal{U}})$ applied to the attribution matrix is chosen again as a vTV-like penalization (see (19)).

3.3. Clustering

For the considered application, the conventional k-means algorithm has been chosen because of its straightforward formulation as an optimization problem. By denoting $\boldsymbol{\theta} = \{\mathbf{B}\}$ a $R \times K$ matrix collecting K centroids, the clustering task (9) can be rewritten as the following NMF problem [40]

$$\min_{\mathbf{Z},\mathbf{B}} \frac{1}{2}\|\mathbf{H} - \mathbf{B}\mathbf{Z}\|_{\text{F}}^2 + \lambda_z \mathcal{R}_z(\mathbf{Z}) + \iota_{\mathbb{S}_K^P}(\mathbf{Z}) + \iota_{\mathbb{R}_+^{R\times K}}(\mathbf{B}) \tag{23}$$

where $\mathcal{R}_z(\mathbf{Z})$ should promote $\mathbf{Z}$ to be composed of orthogonal lines. Combined with the nonnegativity and sum-to-one constraints, it would ensure that $\mathbf{z}_p$ is a vector of zeros except for its kth component equal to 1, i.e., meaning that the pth pixel belongs to the kth cluster. However, handling this orthogonality property within the PALM optimization scheme detailed in Section 2.5 is not straightforward, in particular because the proximal operator associated to this penalization cannot be explicitly computed. In this work, we propose to remove this orthogonality constraint since relaxed attribution vectors may be richer feature vectors for the classification task.

3.4. Multi-objective problem

Based on the quadratic and cross-entropy loss functions considered in the classification task, two distinct global optimization problems are obtained. When considering the quadratic loss of Section 3.2.1, the multi-objective problem (10) writes

$$\begin{aligned}\min_{\substack{\mathbf{H},\mathbf{Q},\mathbf{Z}\\ \mathbf{C}_{\mathcal{U}},\mathbf{B}}} &\frac{\lambda_0}{2}\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_{\text{F}}^2 + \lambda_h\|\mathbf{H}\|_1 + \iota_{\mathbb{R}_+^{R\times P}}(\mathbf{H})\\ &+ \frac{\lambda_1}{2}\|\mathbf{C}\mathbf{D} - \mathbf{Q}\mathbf{Z}\mathbf{D}\|_{\text{F}}^2 + \lambda_c\|\mathbf{C}\|_{\text{vTV}} + \iota_{\mathbb{S}_C^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}})\\ &+ \frac{\lambda_2}{2}\|\mathbf{H} - \mathbf{B}\mathbf{Z}\|_F^2 + \iota_{\mathbb{S}_K^P}(\mathbf{Z}) + \iota_{\mathbb{R}_+^{R\times K}}(\mathbf{B}).\end{aligned} \tag{24}$$

Instead, when considering the cross-entropy loss function proposed in Section 3.2.2, the optimization problem (10) is defined as

$$\begin{aligned}\min_{\substack{\mathbf{H},\mathbf{Q},\mathbf{Z}\\ \mathbf{C}_{\mathcal{U}},\mathbf{B}}} &\frac{\lambda_0}{2}\|\mathbf{Y} - \mathbf{W}\mathbf{H}\|_{\text{F}}^2 + \lambda_h\|\mathbf{H}\|_1 + \iota_{\mathbb{R}_+^{R\times P}}(\mathbf{H})\\ &- \frac{\lambda_1}{2}\sum_{p\in\mathcal{P}} d_p^2 \sum_{i\in\mathcal{C}} c_{i,p}\log\left(\text{sigm}(-\mathbf{q}_{i:}\mathbf{z}_p)\right))\\ &+ \frac{\lambda_q}{2}\|\mathbf{Q}\|_{\text{F}}^2 + \lambda_c\|\mathbf{C}\|_{\text{vTV}} + \iota_{\mathbb{S}_C^{|\mathcal{U}|}}(\mathbf{C}_{\mathcal{U}})\\ &+ \frac{\lambda_2}{2}\|\mathbf{H} - \mathbf{B}\mathbf{Z}\|_F^2 + \iota_{\mathbb{S}_K^P}(\mathbf{Z}) + \iota_{\mathbb{R}_+^{R\times K}}(\mathbf{B}).\end{aligned} \tag{25}$$

Both problems are particular instances of nonnegative matrix co-factorization [27,28]. To summarize, the hyperspectral pixel is first described as a combination of elementary spectra through the learning representation step, aka spectral unmixing. Then, assuming that there exist groups of pixels resulting from the same mixture of materials, a clustering is performed among the abundance vectors. And finally, attribution vectors to the clusters are used as feature vectors for the classification supporting the idea that classes are made of a mixture of clusters. For both multi-objective problems (24) and (25), all conditions required to the use of PALM algorithm described in Section 2.5 are met. Details regarding the two optimization schemes dedicated to these two problems are reported in the Appendix.

3.5. Complexity analysis

Regarding the computational complexity of the proposed Algorithm 1, deriving the gradients shows that it is dominated by matrix product operations. It yields that the algorithm has an overall computational cost in $\mathcal{O}(NK^2P)$ where N is the number of iterations.

4. Experiments

4.1. Implementation details

Before presenting the experimental results, it is worth clarifying the choices which have been made regarding the practical

[1] With a slight abuse of notations, $\mathbf{c}_{(m,n)}$ refers to the pth column of $\mathbf{C}$ where the pth pixel is spatially indexed by $(m,\ n)$.

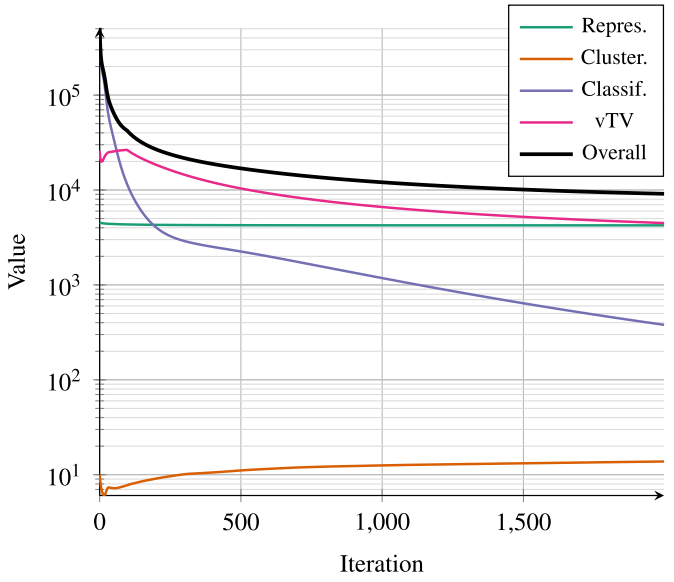


Fig. 3. Convergence of the various terms of objective function (representation learning, clustering, classification, vTV, total).

implementation of the proposed algorithms for the considered application. Important aspects are discussed below.

Convergence diagnosis and stopping rule – In all experiments conducted hereafter, the value of the objective function is monitored at each iteration to determine if convergence has been reached. The normalized difference between the last two consecutive values of the objective function is compared to a threshold and the algorithm stops when the criterion is smaller than this threshold (set as 10^{-4} for the conducted experiments). Fig. 3 shows one example of the behavior of the objective function along the iterations as well as the behavior of several terms composing this overall objective function. As it can be observed from the figure, the global objective function is decreasing over the iteration, which is theoretically ensured by the PALM algorithm.

Initialization – As PALM algorithm only ensures convergence to a critical point and not a global optimum, it remains sensitive to initialization, which needs to be carefully chosen to reach relevant solutions. The initialization of the parameters associated with the learning representation and clustering steps relies on the self-dictionary learning method proposed in [49]. This method proposes to use observed pixels of the image as dictionary elements. The underlying assumption is that the image contains pure pixels, i.e., composed of only a single material. Formally, the initial estimate $\mathbf{H}^0$ of $\mathbf{H}$ is chosen as

$$\mathbf{H}^0 = \underset{\mathbf{H}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{Y} - \tilde{\mathbf{Y}}\mathbf{H}\right\|_{\mathrm{F}}^2 + \alpha\|\mathbf{H}\|_{1,2} \tag{26}$$

where $\|\mathbf{H}\|_{1,2} = \sum_{r=1}^{R}\|\mathbf{h}_{r,:}\|_2$ promotes the use of a reduced number of pixels as dictionary elements and $\tilde{\mathbf{Y}}$ is a submatrix of $\mathbf{Y}$ containing the pixel candidates to be used as dictionary elements. Following the strategy similarly proposed in [49], this subset $\tilde{\mathbf{Y}}$ is built as follows: *i*) for each class of the training set, a k-means is applied to the labeled samples to identify J clusters, *ii*) within a given class, one candidate is retained from each cluster as the pixel the farthest away from the centers of the other clusters (in term of spectral angle distance). This procedure provides a subset $\tilde{\mathbf{Y}}$ composed of $J \times C$ spectrally diverse candidates extracted from the labeled samples.

Then, regarding the representation learning step, only active elements in $\tilde{\mathbf{Y}}$, i.e., those associated with non-zero rows in $\mathbf{H}^0$, are kept to define the dictionary $\mathbf{W}$. Finally, to initialize the variables involved in the clustering step, a k-means is conducted on $\mathbf{H}^0$ and the identified centroids are chosen as $\mathbf{B}^0$ while the corresponding attribution vectors define $\mathbf{Z}^0$. Finally, the classification parameters $\mathbf{Q}^0$ and attribution vectors $\mathbf{C}_{\mathcal{U}}^0$ are randomly initialized.

Weighting the vTV – As explained in Section 2.2, the classification is regularized by a weighted smooth vTV regularization. When all not fixed to the same value, the weights offer the possibility to account for natural boundaries in the observed scene, i.e., variations in the classification map are expected to be localized at the edges in the image. As in [47], an auxiliary dataset informing about the spatial structure of the image can be used to adjust these weights. Instead, in this work, we assume that no such external information is available. Thus these weights are directly computed from the hyperspectral image. More precisely, a virtually observed panchromatic image $\mathbf{y}_{\mathrm{PAN}} \in \mathbb{R}^P$, i.e. a single band image, is first synthetized by averaging the bands of the hyperspectral image $\mathbf{Y}$. Then, the weights are chosen as

$$\beta_{m,n} = \frac{\tilde{\beta}_{m,n}}{\sum_{p,q}\tilde{\beta}_{p,q}} \text{ with } \tilde{\beta}_{m,n} = \frac{1}{\left\|[\nabla \mathbf{y}_{\mathrm{PAN}}]_{m,n}\right\|_2 + \sigma} \tag{27}$$

where $\nabla(\cdot) = [\nabla_{\mathrm{h}}(\cdot)\ \nabla_{\mathrm{v}}(\cdot)]^T$ is the gradient operator and σ is an hyperparameter chosen as $\sigma = 0.01$ to avoid numerical problems and to control the adaptive weighting (the larger σ, the less variation in the weighting) [50].

Hyperparameter scaling – To balance the size and the dynamics of the matrices involved in the cofactorization problem, the hyperparameters λ_0 and λ_q in (24) and (25) have been set as

$$\lambda_0 = \frac{1}{L\|\mathbf{Y}\|_\infty^2}\tilde{\lambda}_0, \quad \lambda_q = \frac{P}{C}\tilde{\lambda}_q. \tag{28}$$

Then, for each experiment presented hereafter, the parameters $\tilde{\lambda}_\cdot$ have been empirically adjusted to obtain consistent results.

4.2. Synthetic hyperspectral image

Data generation – First, to assess the relevance of the proposed model, experiments have been conducted on synthetic images. These synthetic images have been generated using a real hyperspectral image which has been unmixed using the well-established unmixing method SUnSAL [51]. The extracted abundance maps and a set of 6 pure spectra from the hyperspectral library ASTER have been used to build a synthetic hyperspectral images with a realistic spatial organization. The resulting 100-by-250 pixel image presented in Fig. 4 is composed of $L = 385$ spectral bands. The image is associated with a classification groundtruth ($C = 4$) based on the groundtruth of the original real image and a subpart of this groundtruth is assumed known and therefore used as training dataset for the supervised classification step.

Moreover, in this experiment, the endmember matrix $\mathbf{W}$ comprises the 6 spectra actually used to generate the image. To evaluate the robustness of the method in a challenging scenario, these 6 initial endmember spectra are complemented with 9 endmembers not present in the image but very correlated with the 6 actually used ones. The endmember matrix is thus composed of $R = 15$ spectra depicted in Fig. 5.

Compared methods – The proposed methods with quadratic (Q) and cross-entropy (CE) classification losses, denoted respectively by Cofact-Q and Cofact-CE, have been compared with state-of-the-art classification and unmixing methods. First, one considered competing method is the random forest (RF) classifier, which has been extensively used for the hyperspectral image classification. Then, the convolutional neural network (CNN) proposed in [52] has also been tested. This CNN architecture, referred to as ResNet, is based on a residual network specifically designed for hyperspectral image classification. Additionally, the performance

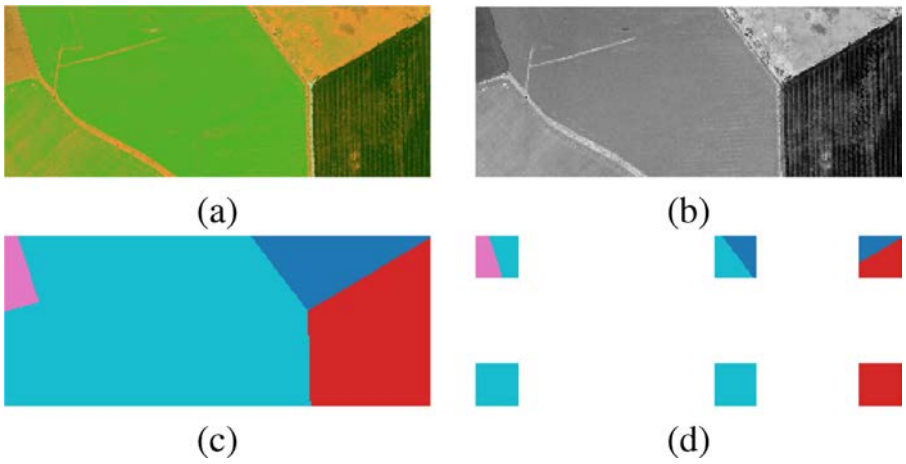


Fig. 4. Synthetic image: (a) colored composition of the hyperspectral image $\mathbf{Y}$, (b) panchromatic image $\mathbf{y}_{\text{PAN}}$, (c) classification ground-truth, (d) training set.

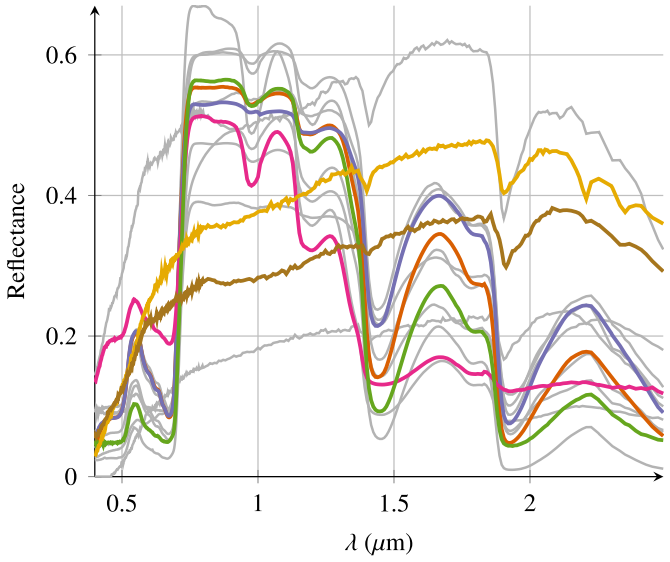


Fig. 5. Spectra used as dictionary $\mathbf{W}$. The 6 color spectra have been used to generate the semi-synthetic image (4 vegetation spectra and 2 soil spectra).

of the classification method proposed in [53] has been evaluated. This method, referred to as SSFPCA+SVM, relies on a so-called spectrally-segmented folded PCA (SSFPCA) as a feature extraction step, followed by a RBF-kernel SVM classifier. Finally, a multinomial logistic regression classifier (MLR) has also been applied directly on the observations. This classifier is equivalent to the classification term proposed in the Cofact-CE method. Thus it will illustrate the interest of using a representation learning step before performing the classification. Parameters of the RF and the SVM have been adjusted using cross-validation with a grid-search strategy and we used the implementations provided in the *scikit-learn* Python library [54]. The parameters of SSFPCA have been set based on the study provided in the original paper. The implementation and parameters proposed by the authors has been used for the ResNet method. All methods except ResNet have been run on a desktop computer with 16Gb of RAM and Intel(R) Xeon(R) CPU E5-1630 v4 @ 3.70GHz $\times$ 8 processor. Due to its high computational load, the ResNet method has been run on a DELL T630 server with 2 Intel(R) Xeon(R) CPU 2640 v4, 2 $\times$ 100Gb of RAM and a Nvidia GTX 1080 TI GPU.

Besides, two unmixing methods proposed in [51] has been tested, namely the fully constrained least squares (FC-SUnSAL) and the constrained sparse regression (CSR-SUnSAL). FC-SUnSAL basically relies on the same data fitting term (14) considered in the proposed cofactorization method, under non-negativity and sum-to-one constraints applied to the abundance vectors. Conversely, the CSR-SUnSAL problem removes the sum-to-one constraint and introduces a ℓ_1-norm penalization on the abundance vectors. It thus solves (15) where the associated regularization parameter λ_h is tuned using a grid-search strategy. These two methods use an augmented Lagrangian splitting algorithm to recover the abundance vectors. Additionally, these abundance vectors are subsequently used as input features of a MLR classifier. This classifier is linear and its combination with the CSR-SUnSAL unmixing algorithm, referred to as CSR-SUnSAL+MLR, yields a sequential counterpart of the proposed Cofact-CE method. In particular, comparing the resulting classification performance with the performance of Cofact-CE allows the benefit of introducing the clustering coupling term to be assessed.

Besides, the proposed method has been also compared with the discriminative K-SVD (D-KSVD) method proposed in [24]. The D-KSVD problem has strong similarities with the proposed cofactorization problem. Indeed, it corresponds to a ℓ_0-penalized representation learning and a classification with a quadratic loss. It aims at learning a dictionary suitable for the classification problem and performs a linear classification on the coding vectors. For this reason, the dictionary $\mathbf{W}$ is only used as an initialization for D-KSVD, while it remains fixed for the unmixing and proposed cofactorization methods. Similarly, the label consistent K-SVD (LC-KSVD) is also considered [25]. This model has been proposed as an improvement of D-KSVD where an additional term ensures that the dictionary elements are class-specific. Hyperparameters of D-KSVD and LC-KSVD have been manually adjusted in order to get the best results. When implementing the PALM algorithm proposed in Section 2.5, the normalized regularization parameters in (28) have been fixed as $\tilde{\lambda}_0 = 100$, $\lambda_1 = \lambda_2 = 1$, $\lambda_h = \lambda_q = 0.1$ and $\tilde{\lambda}_c = 10^{-3}$. Finally, the number of clusters has been set to $K = 10$. The influence of these parameters are empirically studied in the associated companion report [55].

Figure-of-merits – Several metrics are computed to quantify the quality of the classification and unmixing tasks. For classification, two widely-used metrics are used, namely Cohen's kappa and the averaged F1-score over all classes [56]. For unmixing, reconstruction error (RE) and root global mean squared error (RMSE) are computed as follows

$$\text{RE} = \sqrt{\frac{1}{PL}\left\|\mathbf{Y} - \mathbf{W}\hat{\mathbf{H}}\right\|_{\text{F}}^2},$$

Table 2
Synthetic data: unmixing and classification results.

Model	F1-mean	Kappa	RMSE($\hat{\mathbf{H}}$)	RE	Time (s)
Cofact-Q	0.911 ($\pm 3.5 \times 10^{-3}$)	0.893 ($\pm 3.5 \times 10^{-3}$)	0.0528 ($\pm 1.1 \times 10^{-4}$)	0.32 ($\pm 8.9 \times 10^{-4}$)	80 ($\pm$ 6)
Cofact-CE	0.899 ($\pm 5.4 \times 10^{-2}$)	0.880 ($\pm 6.2 \times 10^{-2}$)	0.0524 ($\pm 1.3 \times 10^{-4}$)	0.27 ($\pm 2.2 \times 10^{-3}$)	61 ($\pm$ 4)
MLR	0.873 ($\pm 2.6 \times 10^{-3}$)	0.882 ($\pm 2.3 \times 10^{-3}$)	N\A	N\A	92 ($\pm$ 14)
RF	0.913 ($\pm 1.4 \times 10^{-3}$)	0.907 ($\pm 1.3 \times 10^{-4}$)	N\A	N\A	0.9 ($\pm$ 0.08)
ResNet	0.913 ($\pm 1.6 \times 10^{-2}$)	0.943 ($\pm 4.6 \times 10^{-3}$)	N\A	N\A	220 ($\pm$ 12)[a]
SSFPCA+SVM	0.918 ($\pm 8.3 \times 10^{-4}$)	0.911 ($\pm 2.4 \times 10^{-3}$)	N\A	N\A	4.0 ($\pm$ 0.05)
FC-SUnSAL+MLR	0.893 ($\pm 6.4 \times 10^{-4}$)	0.912 ($\pm 3.7 \times 10^{-4}$)	0.120 ($\pm 3.1 \times 10^{-6}$)	0.37 ($\pm 5.1 \times 10^{-5}$)	6 ($\pm$ 0.3)
CSR-SUnSAL+MLR	0.888 ($\pm 1.0 \times 10^{-3}$)	0.911 ($\pm 5.0 \times 10^{-4}$)	0.125 ($\pm 3.0 \times 10^{-6}$)	0.36 ($\pm 4.2 \times 10^{-5}$)	9 ($\pm$ 0.5)
D-KSVD	0.520 ($\pm 3.1 \times 10^{-3}$)	0.653 ($\pm 3.4 \times 10^{-2}$)	N\A	0.23 ($\pm 4.1 \times 10^{-2}$)	382 ($\pm$ 9)
LC-KSVD	0.879 ($\pm 3.7 \times 10^{-4}$)	0.904 ($\pm 1.0 \times 10^{-4}$)	N\A	30.4 ($\pm 1.0 \times 10^{-4}$)	96 ($\pm$ 1)

[a] Based on a GPU implementation run on a computer cluster.

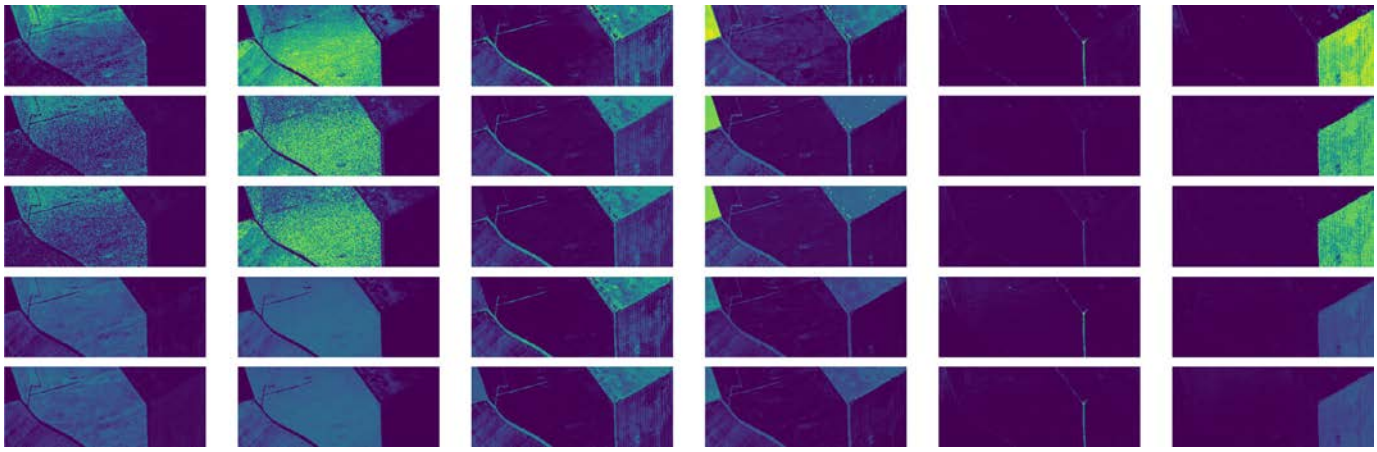


Fig. 6. Synthetic data: abundance maps of the 6 actual endmembers (from left to right): (1st row) ground-truth, (2nd row) Cofact-Q, (3rd row) Cofact-CE, (4rd row) FC-SUnSAL and (5th row) CSR-SUnSAL.

$$\mathrm{RMSE}(\hat{\mathbf{H}}) = \sqrt{\frac{1}{PR}\left\|\mathbf{H}_{\mathrm{true}} - \hat{\mathbf{H}}\right\|_{\mathrm{F}}^{2}} \tag{29}$$

where $\mathbf{H}_{\mathrm{true}}$ and $\hat{\mathbf{H}}$ are the actual and estimated abundance matrices. All these performance metrics are complemented with the computational times. Again, note that for all methods, similar computational framework have been considered except for the CNN-based algorithm whose complexity requires a specific GPU implementation embedded on a computer cluster.

Performance evaluation – Quantitative results obtained on the synthetic dataset are reported in Table 2 and are visually depicted in Figs. 7 and 6 for the classification and abundance maps, respectively. Metrics and their standard deviation have been computed over 20 trials. For each trial, a Gaussian white noise is added the observed image such that SNR = 30 dB. From these results, the proposed method appears to be competitive with the compared state-of-the-art methods. In term of classification results, even though the spatial regularization is very weak in this setting, the cofactorization methods are as good as the RF classifier, which is very satisfying since this latter classifier is one of the most prominent one to deal with HS images [57]. The ResNet algorithm shows similar accuracy in term of F1-score but seems to perform slightly better in term of kappa. However, classification results of FC-SUnSAL and CSR-SUnSAL show that a classifier using abundance vectors can already perform well on this toy example where classes are linearly separable. Similarly, the SSFPCA+SVM methods appears to give interesting results with this synthetic dataset. The MLR using directly the observations appears to be a little less accurate, which may result from the difficulty inherent to high-dimensional inputs. As for LC-KSVD, it performs slightly worse regarding the F1-mean score whereas results of D-KSVD are clearly the worst. In term of unmixing performance, FC-SUnSAL, CSR-SUnSAL, Cofact-Q and Cofact-CE obtain very similar REs. Note however this metrics only evaluates the quality of the reconstructed data. However, the RMSE is lower with the cofactorization methods and the abundance estimations provided by FC-SUnSAL and CSR-SUnSAL significantly degrade. Even if it is not possible to produce a quantitative evaluation of the representation learnt by D-KSVD and LC-KSVD, REs tends to show that D-KSVD successfully estimated a representation of the data (without being easily interpretable) whereas LC-KSVD seems to focus mostly on the discriminative power of the representation at the price of an inaccurate representation. Moreover, the results produced by LC-KSVD have been obtained by increasing the dimension of the representation R to 40 while the results obtained by the other methods have been obtained for $R = 15$ to get good classification performances. The rather poor performance obtained by these two dictionary learning methods, when compared to the proposed cofactorization model, can be explained by the lack of flexibility of the corresponding models which try to recover a descriptive and discriminative representation simultaneously. On the contrary, some flexibility is offered by the clustering step included in the proposed method. Finally, comparison in term of processing times shows that D-KSVD, LC-KSVD and the proposed cofactorization methods are significantly slower, which is expected since these methods conducts representation learning and classification jointly. Nonetheless, the cofactorization methods appears faster than D-KSVD and LC-KSVD. It should be also noted that it is necessary to tune manually the number of iterations when using the two latter methods. Conversely, standard convergence criterion can be implemented for the proposed optimization-based methods.

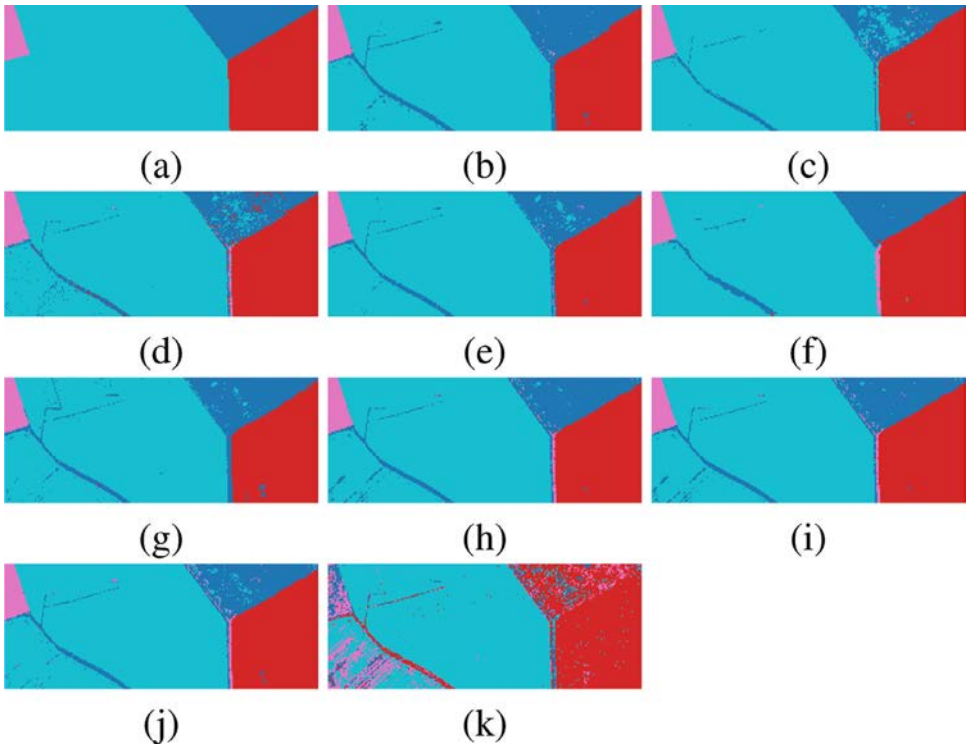


Fig. 7. Synthetic data, classification maps: (a) groundtruth, (b) Cofact-Q, (c) Cofact-CE, (d) MLR, (e) RF, (f) ResNet, (g) SSFPCA, (h) FC-SUnSAL+MLR, (i) CSR-SUnSAL+MLR, (j) LC-KSVD, (k) D-KSVD.

Table 3
AISA data: information about classes.

Class	Nb. of samples	Subclasses
Arable land	177,350	millet, rape, winter barley, winter wheat, oat
Forest	9274	forest
Grassland	25,399	meadow, pasture
Green fallowland	44,370	fallow treated last year, fallow with shrubs
Leguminosae	17,628	leguminosae
Reed	4776	reed
Row crops	79,737	maize, sunflowers

4.3. Real hyperspectral image

Description of the dataset – The Aisa dataset was acquired by the AISA Eagle sensor during a flight campaign over Heves, Hungary. It contains $L = 252$ bands ranging from 395 to 975nm. A set of $C = 7$ classes have been defined for a total of 358,534 referenced pixels, according to the class-wise repartition given in Table 3. To split the full dataset into two test and train subsets, special care has been taken to ensure that training samples are picked out from distinct areas than test samples. The polygons of the reference map are split in smaller polygons on a regular grid pattern and then 50% of the polygons are taken randomly for training and the remaining 50% for testing (see [58] for a similar procedure). Fig. 8 shows a colored composition of the image and the classification ground-truth. Several reasons justify the choice of this particular dataset. First, it is very challenging both in term of classification and unmixing mostly because the spectral signatures of the classes are very similar, leading in particular to very correlated endmember spectra in **W**. Secondly, the ground-truth associated to this image is composed of two levels of classification. Thus, an additional ground-truth is available where the 7 considered classes have been subdivided into 14 classes also detailed in Table 3. These subclasses could be compared to the clustering outputs obtained by the proposed cofactorization method, e.g., to verify either the clusters are consistent with the underlying subclasses.

Compared methods – The proposed algorithm is compared to the same methods introduced above. However, note that the D-KSVD method has experienced some difficulties to scale with the size of this new dataset, which is significantly bigger. Thus to obtain results in a decent amount of time, the algorithm has been interrupted prematurely, i.e., before convergence. Similarly, SVM classifier encounters the same difficulty for the training step and the SVM was finally trained using a subset of the training set (1 over 10 samples). For the proposed cofactorization method, regularization parameters have been set to $\tilde{\lambda}_0 = \tilde{\lambda}_1 = \tilde{\lambda}_2 = \tilde{\lambda}_c = 1$. and $\tilde{\lambda}_h = \tilde{\lambda}_q = 0.01$ and the number of clusters to $K = 30$. The initialization step described in Section 4.1 has been performed and the resulting dictionary **W** is depicted in Fig. 9 ($R = 13$). The same dictionary has been used for the compared unmixing methods.

Performance evaluation – All quantitative results are presented in Table 4. Metrics and their standard deviation have been computed over 5 trials. RMSE metrics have been removed since no groundtruth is available to assess the quality of the estimated abundance maps. RE is thus the only used figure-of-merit to assess the quality of the representation learning. Note however, as previously explained, RE does not directly evaluate the correctness of the abundance maps. In the present case, REs appear to be very similar for all algorithms. Contrary to the previous dataset, this is also the case for LC-KSVD, which can be explained by the fact that spectra are similar in the whole image and it is thus quite easy to get a very low RE with any estimated dictionary. This is the reason why qualitative evaluation remains interesting. Fig. 11 shows a subset of the estimated abundance maps. It is difficult to draw any incontestable conclusion but it is clear that, despite similar REs, significantly different result are obtained for each method. This behavior is strengthened by the very high correlation between the endmembers in this dataset, which may lead to

Fig. 8. AISA dataset: (a) colored composition of the hyperspectral image **Y**, (b) ground-truth [arable land: dark blue, forest: orange, grassland: red, fallowland: brown, leguminosae: pink, reed: green, row crops: light blue].

Table 4
AISA data: unmixing and classification results.

Model	F1-mean	Kappa	RE	Time (s)
Cofact-Q	0.503 ($\pm 4.7 \times 10^{-2}$)	0.652 ($\pm 2.5 \times 10^{-2}$)	0.310 ($\pm 1.6 \times 10^{-4}$)	7303 ($\pm$ 139)
Cofact-CE	0.697 ($\pm 4.5 \times 10^{-2}$)	0.759 ($\pm 3.5 \times 10^{-2}$)	0.310 ($\pm 1.4 \times 10^{-4}$)	4382 ($\pm$ 257)
MLR	0.497 ($\pm 7.3 \times 10^{-2}$)	0.482 ($\pm 7.7 \times 10^{-2}$)	N\A	2060 ($\pm$ 83)
RF	0.711 ($\pm 1.4 \times 10^{-2}$)	0.835 ($\pm 1.2 \times 10^{-2}$)	N\A	41 ($\pm$ 1)
ResNet	0.880 ($\pm 2.3 \times 10^{-2}$)	0.932 ($\pm 1.3 \times 10^{-2}$)	N\A	7576 ($\pm$ 555)[a]
SSFPCA+SVM	0.425 ($\pm 1.5 \times 10^{-2}$)	0.466 ($\pm 1.9 \times 10^{-2}$)	N\A	398 ($\pm$ 12)
FC-SUnSAL+MLR	0.344 ($\pm 3.1 \times 10^{-2}$)	0.433 ($\pm 3.8 \times 10^{-2}$)	0.298 ($\pm 1.9 \times 10^{-3}$)	512 ($\pm$ 96)
CSR-SUnSAL+MLR	0.535 ($\pm 5.0 \times 10^{-2}$)	0.618 ($\pm 8.0 \times 10^{-2}$)	0.304 ($\pm 2.0 \times 10^{-5}$)	529 ($\pm$ 61)
D-KSVD	0.224 ($\pm 2.1 \times 10^{-2}$)	0.406 ($\pm 9.9 \times 10^{-2}$)	0.303 ($\pm 7.6 \times 10^{-6}$)	10475 ($\pm$ 129)
LC-KSVD	0.350 ($\pm 3.2 \times 10^{-2}$)	0.594 ($\pm 3.0 \times 10^{-2}$)	0.303 ($\pm 4.0 \times 10^{-6}$)	3780 ($\pm$ 320)

[a] Based on a GPU implementation run on a computer cluster.

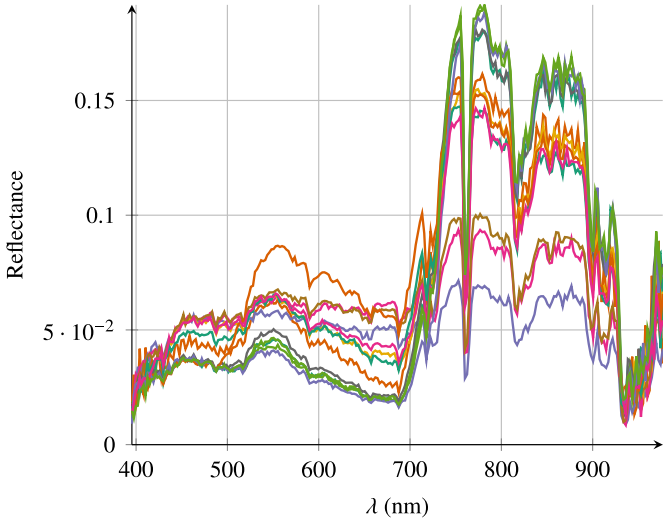


Fig. 9. AISA data: spectra used as the dictionary **W** identified by the self-dictionary method.

probable mismatch between endmember spectra. Nevertheless the Cofact methods seems to give slightly more consistent results. Indeed, edges in the abundance maps appear to be more consistent with boundaries observed in the hyperspectral image. Additionally, for the compared methods, some abundance maps seem to be influenced by the presence of two flight lines in the image. This phenomenon clearly appears in the abundance maps recovered by FC-SUnSAL (3rd row).

Concerning classification results, the results reported in Table 4 show that the classification maps recovered by the Cofact-CE is very closed to the one obtained by RF, whereas SSFPCA+SVM fails to provide reasonable results. As for the ResNet method, it clearly outperforms all the other methods. The better performance could be explained by the fact the neural network used convolutional layers which extract spatial context information. On the contrary, the other methods rely on pixelwise inputs with, at best, a spatial regularization which only promotes local regularity without benefiting from a richer description of the spatial context. Fig. 10 shows in particular that the cofactorization methods encounter some trouble distinguishing very similar classes, for example *grassland* (red) from *fallowland* (brown). Nevertheless, the obtained classification appears to be consistent and it seems reasonable to expect a lesser degradation of the classification results when considering less correlated spectral signatures. This confusion explains the less convincing results of the proposed method with quadratic loss. Besides, it is important to keep in mind that the objective of this work is not to propose the most efficient classification method but rather to propose a method that can give results of similar quality than some state-of-the-art methods, with the benefit of providing additional insights thanks to the joint representation learning. The results also show that the proposed method is beneficial to the classification since FC-SUnSAL+MLR, CSR-SUnSAL+MLR, MLR and Cofact-CE use the same classifier and the latter performs clearly better. The comparison between the representation learning-based algorithms is clear and the both Cofact methods perform better than LC-KSVD and D-KSVD.

In term of processing time, LC-KSVD, D-KSVD and the Cofact methods are clearly more time consuming. Nevertheless, all those methods provide more outputs than the other methods. The comparison between these methods seems to give an advantage for LC-KSVD. However, it should be noted that it is very difficult to monitor the convergence of LC-KSVD and D-KSVD since the value of the objective function over the iteration is not monotonic. The proposed algorithms and their implementations thus give a practical advantage since they do not need to be applied with different numbers of iterations to ensure good results.

One of very interesting feature of the Cofact method is the possibility of examining the clusters obtained as a byproduct. Given the formulation (23), the centroids **B** estimated by the Co-

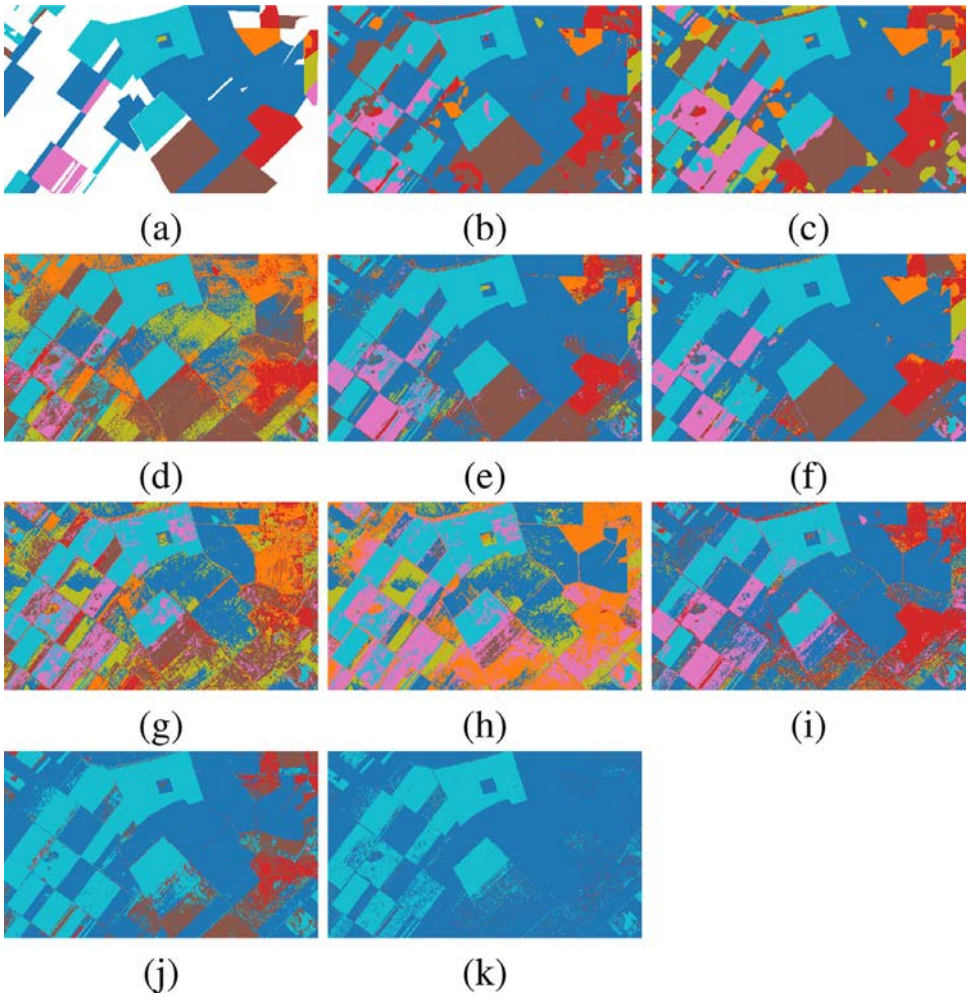


Fig. 10. AISA image, classification maps: (a) groundtruth, (b) Cofact-Q, (c) Cofact-CE, (d) MLR, (e) RF, (f) ResNet, (g) SSFPCA, (h) FC-SUnSAL+MLR, (i) CSR-SUnSAL+MLR, (j) LC-KSVD, (k) D-KSVD.

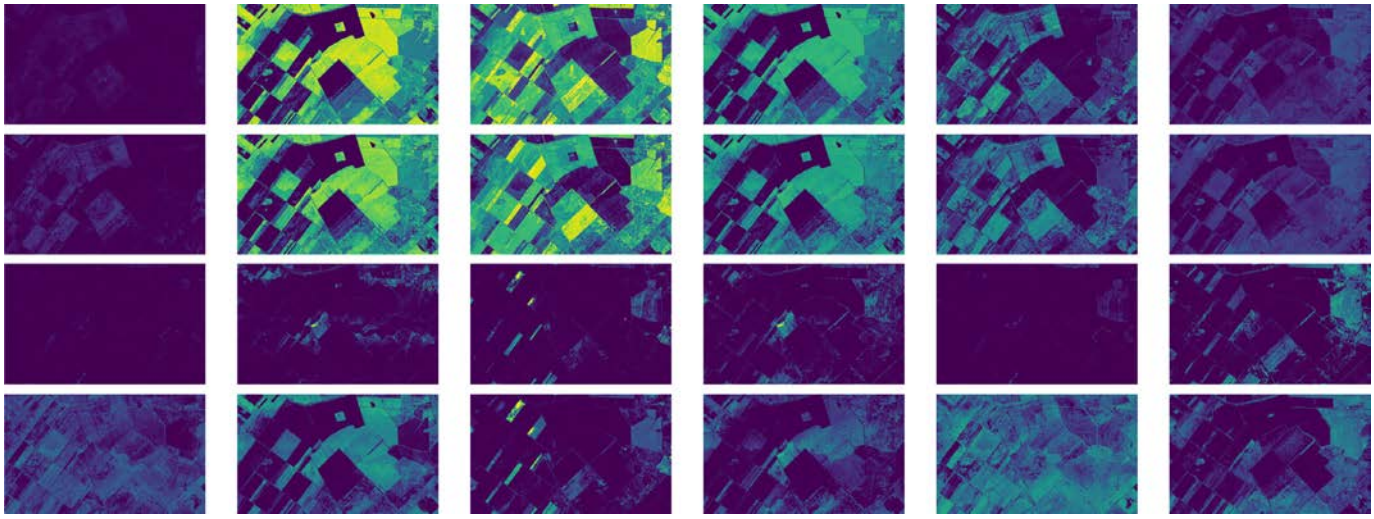


Fig. 11. AISA dataset, abundances map for the 6 components: (1st row) Cofact-Q, (2nd row) Cofact-CE, (3rd row) FC-SUnSAL and (4th row) CSR-SUnSAL.

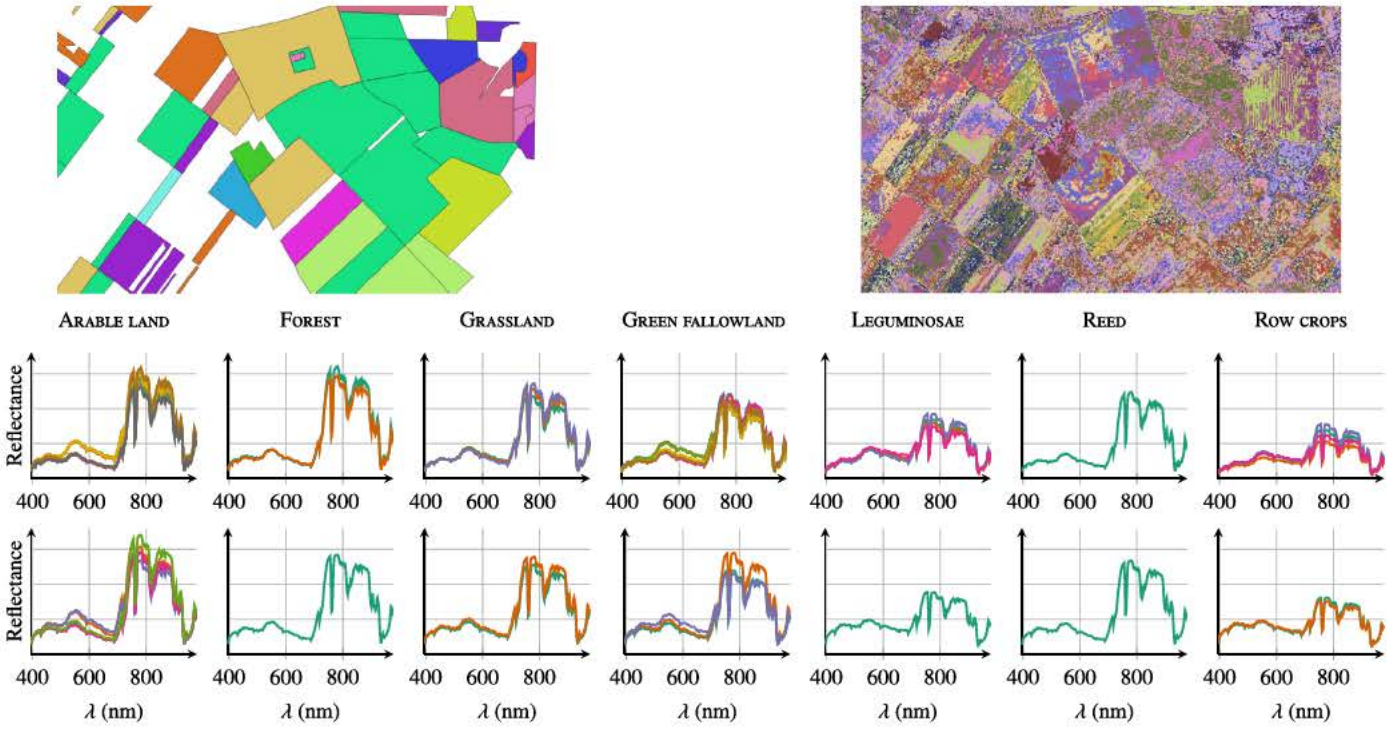


Fig. 12. AISA data: (1st row) Groundtruth map of subclasses and clustering recovered by Cofact-CE, (2nd row) for each class, spectral centroids of the clusters recovered by Cofact-CE composing the class, (3rd row) for each class, mean spectra of the groundtruth subclasses composing the class.

fact method can be interpreted as average behaviors of abundance vectors. Corresponding virtual spectral signatures can be obtained by right-multiplying the dictionary $\mathbf{W}$ by this estimated abundance-like matrix $\mathbf{B}$. The first line in Fig. 12 shows these spectral centroids for each cluster. Accessing this kind of information is precious in term of image interpretation since it offers the possibility of visualizing any class multi-modality. To illustrate, the second line of Fig. 12 shows the mean spectra associated with the subclass groundtruth. Clearly, both lines exhibit strong similarities, with spectral diversity (hence multi-modality) for the 1st, 3rd and 4rd classes. This illustrates the relevance of the clusters recovered by the proposed cofactorization method.

5. Conclusion and perspectives

This paper proposed a cofactorization model to unify a representation learning task and a classification task. The coding matrices associated with the two factorization problems, which respectively are the low-dimensional representations and the feature vectors, were related thanks to a clustering step. The low-dimensional representation vectors were clustered and the resulting attribution vectors were used as features vectors. These three tasks were jointly formulated as a non-convex non-smooth minimization problem, whose solution was approximated thanks to a PALM algorithm which ensured some convergence guarantees. The interest of considering a clustering task as a coupling process is threefold. First, it allows the learnt representation to be both descriptive and discriminative to ensure a low reconstruction error and a good separability of the classes, respectively. These two properties are often adversarial and the clustering term offers an additional degree of freedom to accommodate both properties. Secondly, instead of linearly separating the classes in the low-dimensional representation space, the resulting method achieves a non-linear classification relying on the coding vectors. The clustering term acts similarly as the well-known kernel trick since the coding vectors are mapped into a new representation space, the cluster attribution space, where classes are expected to be linearly separable. Finally, the clustering is very interesting to interpret the obtained results. For instance, analyzing the identified cluster centroids allows the end-user to characterize the possible class multi-modality. This model was instanced in an particular applicative scenario, namely hyperspectral image analysis, to jointly conduct unmixing and classification. It provided convincing results on synthetic and real data both quantitatively and qualitatively. Moreover, byproducts of the estimation appeared to be a relevant added value to interpret the obtained results.

To further improve the developed model, it would be particularly interesting to investigate the best way to learn an appropriate dictionary. For instance, it would be relevant to directly exploit the supervised information to get a better dictionary initialization. Moreover, updating the dictionary when solving the cofactorization problem would be also of interest. Another promising future work would consist in replacing the stage of the model dedicated to the classification task by a more advanced classifier. Indeed, when using the cross-entropy loss, this factorization model was interpreted as a single-layer neural network. A natural extension would be to leverage on a deeper architecture, while preserving the benefit of interpretability brought by the hierarchical representation of the data through the learning, clustering and classification steps. Finally, the genericity of the proposed approach should be assessed through the analysis of data from other applicative contexts where representation learning and classification play central roles, such as medical imaging of various modalities [16,59].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Adrien Lagrange: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft, Writing - review & editing. **Mathieu Fauvel:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision. **Stéphane May:** Conceptualization, Funding acquisition. **José Bioucas-Dias:** Conceptualization, Methodology. **Nicolas Dobigeon:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing, Supervision.

Acknowledgments

The authors would like to thank Olivier Gouvert and Prof. Cédric Févotte (IRIT, Univ. of Toulouse, CNRS, France) for fruitful discussions regarding this work. They also thank Juan Mario Haut and Pr. Antonio J. Plaza for providing the code associated with the ResNet method [52]. Part of the numerical experiments were conducted on the OSIRIM platform[2] of IRIT, supported by the CNRS, the FEDER, the Occitanie region and the French government.

Appendix A. Technical derivations

This appendix provides some details regarding the optimization schemes instanced for the proposed cofactorization model with the classification quadratic and cross-entropy losses.

A.1. Cofactorization model with quadratic loss function

Using notations consistent with (11), the smooth coupling term of the quadratic (Q) loss cost can be expressed as

$$g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) = \frac{\lambda_0}{2}\|\mathbf{Y}-\mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2 + \frac{\lambda_1}{2}\|\mathbf{C}\mathbf{D}-\mathbf{Q}\mathbf{Z}\mathbf{D}\|_{\mathrm{F}}^2 + \lambda_c\|\mathbf{C}\|_{\mathrm{vTV}} + \frac{\lambda_2}{2}\|\mathbf{H}-\mathbf{B}\mathbf{Z}\|_{\mathrm{F}}^2.$$

For a practical implementation, one needs to compute the partial gradients of $g(\cdot)$ explicitly and their Lipschitz constants to perform the gradient descent. Regarding the $\mathbf{H}$ and $\mathbf{B}$ variables, these computations are the same for the two models (quadratic and cross-entropy losses) and lead to

$$\nabla_{\mathbf{H}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) = \lambda_0(\mathbf{W}^t\mathbf{W}\mathbf{H}-\mathbf{W}^t\mathbf{Y}) + \lambda_2(\mathbf{H}-\mathbf{B}\mathbf{Z}), \tag{A.1}$$

$$\nabla_{\mathbf{B}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) = \lambda_2(\mathbf{B}\mathbf{Z}\mathbf{Z}^t-\mathbf{H}\mathbf{Z}^t), \tag{A.2}$$

Regarding the variables $\mathbf{Z}$, $\mathbf{Q}$ and $\mathbf{C}_{\mathcal{U}}$ involved in the classification step with quadratic loss, they writes

$$\begin{aligned}
\nabla_{\mathbf{Z}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= \lambda_2(\mathbf{B}^T\mathbf{B}\mathbf{Z}-\lambda_1\mathbf{B}^T\mathbf{H}) \\
&\quad + \lambda_1(\mathbf{Q}^T\mathbf{Q}\mathbf{Z}\mathbf{D}^2-\mathbf{Q}^T\mathbf{C}\mathbf{D}^2), \\
\nabla_{\mathbf{Q}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= \lambda_1(\mathbf{Q}\mathbf{Z}\mathbf{D}^2\mathbf{Z}^T-\mathbf{C}\mathbf{D}^2\mathbf{Z}^T), \\
\nabla_{\mathbf{C}_{\mathcal{U}}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= \lambda_c\nabla_{\mathbf{C}_{\mathcal{U}}}\|\mathbf{C}\|_{\mathrm{vTV}} + \lambda_1(\mathbf{C}_{\mathcal{U}}\mathbf{D}_{\mathcal{U}}^2-\mathbf{Q}\mathbf{Z}_{\mathcal{U}}\mathbf{D}_{\mathcal{U}}^2).
\end{aligned} \tag{A.3}$$

For sake of brevity, the gradient $\nabla\,.\,\|\cdot\|_{\mathrm{vTV}}$ of the vectorial TV regularization is not explicitly given. Readers are referred to [60] for further details.

All partial gradients are globally Lipschitz as functions of the corresponding partial variables. After basic matrix derivations, majorizations similar to (13) lead to the following Lipschitz constant

$$\begin{aligned}
L_{\mathbf{H}} &= \left\|\lambda_0\mathbf{W}^T\mathbf{W}+\lambda_2\mathbf{I}_R\right\|, \\
L_{\mathbf{B}}(\mathbf{Z}) &= \left\|\lambda_2\mathbf{Z}\mathbf{Z}^T\right\|, \\
L_{\mathbf{Z}}(\mathbf{B},\mathbf{Q}) &= \max_p \left\|\lambda_2\mathbf{B}^T\mathbf{B}+\lambda_1 d_p\mathbf{Q}^T\mathbf{Q}\right\|, \\
L_{\mathbf{Q}}(\mathbf{Z}) &= \left\|\lambda_1\mathbf{Z}\mathbf{D}^2\mathbf{Z}^T\right\|, \\
L_{\mathbf{C}_{\mathcal{U}}} &= \lambda_1\max_p d_p^2 + \lambda_c\frac{\sqrt{8}\max_p\beta_p}{\epsilon}.
\end{aligned} \tag{A.4}$$

A.2. Cofactorization model with cross-entropy loss function

When using cross-entropy as the classification loss function, the coupling term writes

$$\begin{aligned}
g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= \frac{\lambda_0}{2}\|\mathbf{Y}-\mathbf{W}\mathbf{H}\|_{\mathrm{F}}^2 \\
&\quad - \frac{\lambda_1}{2}\sum_{p\in\mathcal{P}} d_p^2\sum_{i\in\mathcal{C}} c_{i,p}\log\left(\mathrm{sigm}(-\mathbf{q}_{i:}\mathbf{z}_p)\right)) \\
&\quad + \frac{\lambda_q}{2}\|\mathbf{Q}\|_F^2 + \lambda_c\|\mathbf{C}\|_{\mathrm{vTV}} + \frac{\lambda_2}{2}\|\mathbf{H}-\mathbf{B}\mathbf{Z}\|_F^2
\end{aligned} \tag{A.5}$$

and the specific partial gradients are

$$\begin{aligned}
\nabla_{\mathbf{Z}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= -\frac{\lambda_1}{2}\mathbf{Q}^T\mathbf{G} \\
\nabla_{\mathbf{Q}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= -\frac{\lambda_1}{2}\mathbf{G}\mathbf{Z}^T + \lambda_q\mathbf{Q}, \\
\nabla_{\mathbf{C}_{\mathcal{U}}} g(\mathbf{H},\mathbf{B},\mathbf{Z},\mathbf{C}_{\mathcal{U}},\mathbf{Q}) &= \lambda_c\nabla_{\mathbf{C}_{\mathcal{U}}}\|\mathbf{C}_{\mathcal{U}}\|_{\mathrm{vTV}} \\
&\quad - \frac{\lambda_1}{2}\sum_{p\in\mathcal{P}} d_p^2\sum_{i\in\mathcal{C}}\log\left(\mathrm{sigm}(-\mathbf{q}_{i:}\mathbf{z}_p)\right))
\end{aligned} \tag{A.6}$$

where $\mathbf{G}$ is a $C\times P$ matrix with elements given by

$$g_{i,p} = \frac{d_p^2 c_{i,p}}{1+\exp(-\mathbf{q}_{i:}\mathbf{z}_p)}. \tag{A.7}$$

It should be noticed that $\mathbf{G}$ depends on $\mathbf{Z}$, $\mathbf{Q}$ and $\mathbf{C}$ and is only introduced here to get compact notations. The following Lipschitz constants can be derived

$$\begin{aligned}
L_{\mathbf{Z}}(\mathbf{B},\mathbf{Q}) &= \lambda_1\sum_{p\in\mathcal{P}} d_p^2\sum_{i\in\mathcal{C}} c_{i,p}\left\|\mathbf{q}_{j:}\right\|_2^2 + \left\|\lambda_2\mathbf{B}\mathbf{B}^T\right\|, \\
L_{\mathbf{Q}} &= \lambda_1\sum_{p\in\mathcal{P}} d_p^2 + \lambda_{\mathrm{q}}, \\
L_{\mathbf{C}_{\mathcal{U}}} &= \lambda_c\frac{\sqrt{8}\max_p\beta_p}{\epsilon}.
\end{aligned} \tag{A.8}$$

A.3. Computing the proximal operators

For a practical implementation of the PALM algorithm, the proximal operators associated with each $f_j(\cdot)$ in (12) need to be computed. It is clear that all these functions are proper lower semi-continuous functions for both models instanced in Section 3.4. The involved indicator functions are defined on convex sets. Thus, their proximal operators can be expressed as projections. The projection on the non-negative quadrant is a simple thresholding of negative values. The projection on the simplices $\mathcal{S}$. can be conducted as detailed in [61]. The case of $f_0(\cdot)$ defined by a nonnegativity constraint complemented by a ℓ_1-norm sparsity promoting regularization is slightly more complex. It can be handled using a composition of proximal operators. As stated before, the proximal operator associated to the positivity constraint is the projection on the non-negative quadrant. The proximal operator associated with the ℓ_1-norm penalization is a soft-thresholding, i.e., $\mathrm{prox}^t_{\|\cdot\|_1}(x) = \mathrm{sign}(x)(|x|-\frac{1}{t})_+$ [62]. These two proximal operators satisfy the conditions exhibited in [42] required to be allowed to perform their compositions to get the proximal operator associated to $f_0(\cdot)$.

[2] http://osirim.irit.fr/site/en.

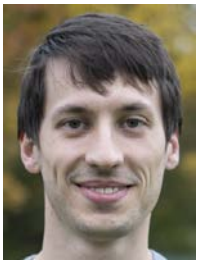
Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neucom.2019.12.068.

References

[1] J.A. Benediktsson, P. Ghamisi, Spectral-Spatial Classification of Hyperspectral Remote Sensing Images, Artech House, 2015.
[2] G. Moser, J. Zerubia, Mathematical Models for Remote Sensing Image Processing, Springer, 2018.
[3] Y. Chen, H. Jiang, C. Li, X. Jia, P. Ghamisi, Deep feature extraction and classification of hyperspectral images based on convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 54 (10) (2016) 6232–6251.
[4] Y. Yuan, J. Fang, X. Lu, Y. Feng, Remote sensing image scene classification using rearranged local features, IEEE Trans. Geosci. Remote Sens. 57 (3) (2019) 1779–1792.
[5] X. Lu, W. Ji, X. Li, X. Zheng, Bidirectional adaptive feature fusion for remote sensing scene classification, Neurocomputing 328 (2019) 135–146.
[6] G. Camps-Valls, L. Bruzzone, Kernel Methods for Remote Sensing Data Analysis, John Wiley & Sons, 2009.
[7] M. Belgiu, L. Drăguţ, Random forest in remote sensing: a review of applications and future directions, ISPRS J. Photogramm. Remote Sens. 114 (2016) 24–31.
[8] C. Chang, Statistical detection theory approach to hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (4) (2019) 2057–2074.
[9] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (1968) 55–63.
[10] L. Qi, X. Lu, X. Li, Exploiting spatial relation for fine-grained image classification, Patt. Recognit. 91 (2019) 47–55.
[11] 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 J. Sel. Top. Appl. Earth Observ. Remote Sens. 5 (2012) 354–379.
[12] O. Eches, N. Dobigeon, J.-Y. Tourneret, Enhancing hyperspectral image unmixing with spatial correlations, IEEE Trans. Geosci. Remote Sens. 49 (2011) 4239–4247.
[13] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311.
[14] M. Zibulevsky, B.A. Pearlmutter, Blind source separation by sparse decomposition in a signal dictionary, Neural Comput. 13 (4) (2001) 863–882.
[15] C.J.D. Porta, A.A. Bekit, B.H. Lampe, C. Chang, Hyperspectral image classification via compressive sensing, IEEE Trans. Geosci. Remote Sens. 57 (10) (2019) 8290–8303.
[16] Y.C. Cavalcanti, T. Oberlin, N. Dobigeon, S. Stute, M. Ribeiro, C. Tauber, Unmixing dynamic PET images with variable specific binding kinetics, Med. Image Anal. 49 (2018) 117–127.
[17] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (8) (2009) 30–37.
[18] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2765–2781.
[19] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788.
[20] D. Donoho, V. Stodden, When does non-negative matrix factorization give a correct decomposition into parts? in: Adv. in Neural Information Process. Systems, 2004, pp. 1141–1148.
[21] J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 791–804.
[22] X. Zheng, Y. Yuan, X. Lu, Dimensionality reduction by spatial–spectral preservation in selected bands, IEEE Trans. Geosci. Remote Sens. 55 (9) (2017) 5185–5197.
[23] Z. Zhang, W. Jiang, J. Qin, L. Zhang, F. Li, M. Zhang, S. Yan, Jointly learning structured analysis discriminative dictionary and analysis multiclass classifier, IEEE Trans. Neural Netw. Learn. Syst. 29 (8) (2018) 3798–3814.
[24] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2691–2698.
[25] Z. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent K-SVD, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1697–1704.
[26] C. Wang, D.M. Blei, Collaborative topic modeling for recommending scientific articles, in: Proc. ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, 2011, pp. 448–456.
[27] J. Yoo, M. Kim, K. Kang, S. Choi, Nonnegative matrix partial co-factorization for drum source separation, in: Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP), 2010, pp. 1942–1945.
[28] N. Yokoya, T. Yairi, A. Iwasaki, Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion, IEEE Trans. Geosci. Remote Sens. 50 (2) (2012) 528–537.
[29] N. Akhtar, A. Mian, Nonparametric coupled bayesian dictionary and classifier learning for hyperspectral classification, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2018) 4038–4050.
[30] A. Lagrange, M. Fauvel, S. May, N. Dobigeon, Hierarchical Bayesian image analysis: from low-level modeling to robust supervised learning, Pattern Recognit. 85 (2018) 26–36.
[31] J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, Math. Program. 146 (1–2) (2014) 459–494.
[32] C. Févotte, J. Idier, Algorithms for nonnegative matrix factorization with the beta-divergence, Neural Comput. 23 (9) (2011) 2421–2456.
[33] I. Jolliffe, Principal component analysis, in: International Encyclopedia of Statistical Science, Springer, 2011, pp. 1094–1096.
[34] P. Paatero, U. Tapper, Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (2) (1994) 111–126.
[35] A.M. Bruckstein, M. Elad, M. Zibulevsky, On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations, IEEE Trans. Inf. Theory 54 (11) (2008) 4813–4820.
[36] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Springer, New York, NY, 2009.
[37] D.M. Kline, V.L. Berardi, Revisiting squared-error and cross-entropy functions for training neural network classifiers, Neural Comput. Appl. 14 (4) (2005) 310–318.
[38] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, first ed., MIT press Cambridge, 2016.
[39] L. Condat, A convex approach to K-means clustering and image segmentation, in: Int. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2017, pp. 220–234.
[40] F. Pompili, N. Gillis, P.-A. Absil, F. Glineur, Two algorithms for orthogonal nonnegative matrix factorization with application to clustering, Neurocomputing 141 (2014) 15–25.
[41] X. Sun, N.M. Nasrabadi, T.D. Tran, Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints, IEEE Trans. Geosci. Remote Sens. 53 (8) (2015) 4457–4471.
[42] Y.-L. Yu, On decomposing the proximal map, in: Adv. in Neural Information Process. Systems, 2013, pp. 91–99.
[43] L. Drumetz, M.-A. Veganzones, S. Henrot, R. Phlypo, J. Chanussot, C. Jutten, Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability, IEEE Trans. Image Process. 25 (8) (2016) 3890–3905.
[44] Y. Zhang, M. Brady, S. Smith, Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm, IEEE Trans. Med. Imaging 20 (2001) 45–57.
[45] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Proc. IEEE Int. Conf. Computer Vision (ICCV), 2011, pp. 543–550.
[46] Y. Liu, F. Condessa, J.M. Bioucas-Dias, J. Li, P. Du, A. Plaza, Convex formulation for multiband image classification with superpixel-based spatial regularization, IEEE Trans. Geosci. Remote Sens. 56 (5) (2018) 2704–2721.
[47] T. Uezato, M. Fauvel, N. Dobigeon, Hyperspectral image unmixing with LiDAR data-aided spatial regularization, IEEE Trans. Geosci. Remote Sens. 56 (2) (2018) 4098–4108.
[48] P.J. Huber, Robust estimation of a location parameter, Ann. Math. Stat. 35 (1) (1964) 73–101.
[49] N. Gillis, R. Luce, A fast gradient method for nonnegative sparse regression with self dictionary, IEEE Trans. Image Process. 27 (1) (2018) 24–37.
[50] D.M. Strong, P. Blomgren, T.F. Chan, Spatially adaptive local-feature-driven total variation minimizing image restoration, in: SPIE Statistical and Stochastic Methods in Image Processing II, 3167, 1997, pp. 222–234.
[51] J.M. Bioucas-Dias, M.A. Figueiredo, Alternating direction algorithms for constrained sparse regression: application to hyperspectral unmixing, in: Proc. IEEE GRSS Workshop Hyperspectral Image SIgnal Process.: Evolution in Remote Sens. (WHISPERS), 2010, pp. 1–4.
[52] M.E. Paoletti, J.M. Haut, R. Fernandez-Beltran, J. Plaza, A.J. Plaza, F. Pla, Deep pyramidal residual networks for spectral–spatial hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. 57 (2) (2019) 740–754.
[53] M.P. Uddin, M.A. Mamun, M.A. Hossain, Effective feature extraction through segmentation-based folded-PCA for hyperspectral image classification, Int. J. Remote Sens. 40 (18) (2019) 7190–7220.
[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[55] A. Lagrange, M. Fauvel, S. May, J. Bioucas-Dias, N. Dobigeon, Matrix Cofactorization for Joint Representation Learning and Supervised Classification – Application to Hyperspectral Image Analysis. Complementary results, Technical Report, University of Toulouse, IRIT/INP-ENSEEIHT, France, 2019. http://dobigeon.perso.enseeiht.fr/papers/Lagrange_TechReport_2019.pdf
[56] R.G. Congalton, K. Green, Assessing the Accuracy of Remotely Sensed Data: Principles and Practices, CRC press, 2008.
[57] A. Stoian, V. Poulain, J. Inglada, V. Poughon, D. Derksen, Land cover maps production with high resolution satellite image time series and convolutional neural networks: adaptations and limits for operational systems, Remote Sens. 11 (17) (2019) 1986.
[58] A. Lagrange, M. Fauvel, M. Grizonnet, Large-scale feature selection with Gaussian mixture models for the classification of high dimensional remote sensing images, IEEE Trans. Comput. Imaging 3 (2) (2017) 230–242.
[59] L. Chaari, T. Vincent, F. Forbes, M. Dojat, P. Ciuciu, Fast joint detection-estimation of evoked brain activity in event-related fMRI using a variational approach, IEEE Trans. Med. Imaging 32 (5) (2013) 821–837.
[60] P. Getreuer, Rudin-Osher-Fatemi total variation denoising using split Bregman, Image Process. Line 2 (2012) 74–95.

[61] L. Condat, Fast projection onto the simplex and the l1 ball, Math. Program. 158 (1–2) (2016) 575–585.
[62] R. Jenatton, J. Mairal, G. Obozinski, F. Bach, Proximal methods for hierarchical sparse coding, J. Mach. Learn. Res. 12 (Jul) (2011) 2297–2334.



Adrien Lagrange received an Engineering degree in Robotics and Embedded Systems from ENSTA ParisTech, France, and the M.Sc. degree in Machine Learning from the Paris Saclay University, both in 2016.

He is currently a Ph.D. student at the National Polytechnic Institute of Toulouse. He is working on the subject of spectral unmixing and classification of hyperspectral images under the supervision of Nicolas Dobigeon and Mathieu Fauvel.

Mathieu Fauvel received the Ph.D. degrees in image and signal processing from the Grenoble Institut of Technology in 2007. From 2008 to 2010, he was a postdoctoral researcher with the MISTIS Team of the National Institute for Research in Computer Science and Control (INRIA). Since 2011, Dr. Fauvel has been an Associate Professor with the National Polytechnic Institute of Toulouse within the DYNAFOR lab (INRA). His research interests are remote sensing, pattern recognition, and image processing.

Stéphane May received an Engineering degree France in Telecommunications from National Institut of Telecommunications (Evry, France), in 1997. He is currently with the Centre National d'Études Spatiales (French Space Agency), Toulouse, France, where he is developing image processing algorithms and softwares for the exploitation of Earth observation images.

José Bioucas-Dias received the E.E., M.Sc., Ph.D., and "Agregado" degrees from Instituto Superior Técnico (IST), Technical University of Lisbon (TULisbon, now University of Lisbon), Portugal, in 1985, 1991, 1995, and 2007, respectively, all in electrical and computer engineering. Since 1995, he has been with the Department of Electrical and Computer Engineering, IST, where he was an Assistant Professor from 1995 to 2007 and an Associate Professor since 2007. Since 1993, he has also been a Senior Researcher with the Pattern and Image Analysis Group of the Instituto de Telecomunicações, which is a private non-profit research institution. His research interests include inverse problems, signal and image processing, pattern recognition, optimization, and remote sensing.

Nicolas Dobigeon received the Ph.D. degree in Signal Processing from the National Polytechnic Institute of Toulouse in 2012. He was a postdoctoral researcher with the Department of Electrical Engineering and Computer Science, University of Michigan (USA), from 2007 to 2008. Since 2008, he has been with the National Polytechnic Institute of Toulouse, currently with a Professor position. His research interests include statistical signal and image processing.