

Dictionary-Based Tensor Canonical Polyadic Decomposition

Jérémy Emile Cohen ^{1b} and Nicolas Gillis ^{1b}

Abstract—To ensure interpretability of extracted sources in tensor decomposition, we introduce in this paper a dictionary-based tensor canonical polyadic decomposition, which enforces one factor to belong exactly to a known dictionary. A new formulation of sparse coding is proposed, which enables high-dimensional tensors dictionary-based canonical polyadic decomposition. The benefits of using a dictionary in tensor decomposition models are explored both in terms of parameter identifiability and estimation accuracy. Performances of the proposed algorithms are evaluated on the decomposition of simulated data and the unmixing of hyperspectral images.

Index Terms—Tensor, multiway analysis, sparse coding, constrained optimization, spectral unmixing.

I. INTRODUCTION

GIVEN a mixture of several components, a classic problem in signal processing is to separate the contribution of each of these components using solely the information contained in the data. This problem is known as blind source separation and has been a particularly widely studied topic for the last two decades [1]. When the available data are contained in a multiway array, that is, a table of three entries or more, blind source separation has been successfully achieved by means of tensor decomposition techniques in a large variety of applications, ranging from telecommunications [2] to chemometrics [3], spectral unmixing [4], neuroimaging [5], social sciences [6] and machine learning [7].

The key concept behind tensor decomposition methods is the linearity of blocks of parameters of interest such as spectra, concentrations or time signatures with respect to experimental parameters such as wavelength, pixel index, subject index or time. For the canonical polyadic decomposition (CPD) model studied in this paper, each block of parameters of interest depends respectively on only one experimental parameter.

As an illustration, tensor decomposition techniques can be used to perform source separation with hyperspectral images, a task often referred to as spectral unmixing. Hyperspectral

images are 2D images collected for a large number of wavelengths, and possibly along time. In this scenario, the blocks of parameters of interest are spectral signatures (characteristic responses of materials to light stimulation), abundances (relative concentrations) and time evolution.

For most applications, tensor decomposition models are however not exactly following a physical models because of their strong multilinearity assumptions. As a consequence, modeling error has to be accounted for when processing the results of the source separation using tensors. This matters when the goal of the source separation is to identify the components in the mixture. The estimated parameters of interest have to be compared with benchmarks, and modeling error induces error on the estimated parameters which may in turn deteriorate identification performances.

In this paper, we want to avoid splitting the identification procedure and the source separation procedure. It is shown that using a formalism inspired from sparse coding [8], merging source separation and identification is not only possible but also offers advantages in term of uniqueness properties of the tensor decomposition, and may reduce estimation error on identified factors provided the *a priori* information is accurate.

OUTLINE AND CONTRIBUTIONS

In this work, after discussing notations and vocabulary, we make the following contributions:

- Formalize high order tensor sparse coding by modifying the usual sparse coding formulation for matrices [8], [9]. We call the obtained tensor model dictionary canonical polyadic decomposition (DCPD).
- Provide tools for introducing flexibility in the DCPD formulation. This makes the DCPD model more suitable for practical problems.
- Study the identifiability of parameters of the DCPD for matrices and higher order tensors, and study the existence of a best low rank DCPD approximation.
- Develop greedy and continuous algorithms to compute the DCPD and its flexible variants.
- Check the identification performances of the DCPD model on synthetic data with respect to the CPD model. We also use DCPD in the matrix case to perform spectral unmixing under pure-pixel assumption of the Urban and Terrain data sets¹ and compare our results with state-of-the-art methods [10]–[13].

Manuscript received May 29, 2017; revised October 24, 2017; accepted November 16, 2017. Date of publication November 24, 2017; date of current version February 15, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cédric Févotte. The work of J. E. Cohen was supported by the F.R.S.-FNRS (incentive grant for Scientific Research F.4501.16). The work of N. Gillis was supported in part by the F.R.S.-FNRS (incentive grant for Scientific Research F.4501.16) and in part by the ERC under Starting Grant 679515. (Corresponding author: Jérémy Emile Cohen.)

The authors are with the University of Mons, Mons 7000, Belgium (e-mail: jeremy.cohen@umons.ac.be; nicolas.gillis@umons.ac.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2017.2777393

¹available at <http://www.agc.army.mil/>

TABLE I
BASIC DEFINITIONS FROM LINEAR ALGEBRA

$\mathcal{E} \otimes \mathcal{F}$: tensor product space, linear space mapped by \otimes from $\mathcal{E} \times \mathcal{F}$. $\mathbf{a} \otimes \mathbf{b}$: tensor product of two vectors, <i>i.e.</i> an element of $\mathcal{E} \otimes \mathcal{F}$, can be understood as an outer product of vectors if the tensor space is an array space. [14] $\mathbf{A} \odot \mathbf{B}$: Khatri-Rao (columnwise Kronecker) product of matrices [19]. $\mathbf{A} \boxtimes \mathbf{B}$: Hadamard product of matrices, <i>i.e.</i> , element-wise product [19].
--

TABLE II
SOME DEFINITIONS AND PROPERTIES OF MULTILINEAR OPERATORS

$\mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}$: an operator acting on a third order tensor. $(\mathbf{U} \otimes \mathbf{V} \otimes \mathbf{W}) \mathcal{T}$: application of \mathbf{U} on the first mode, \mathbf{V} on the second mode, \mathbf{W} on the third mode, also noted $\mathcal{T} \bullet_1 \mathbf{U} \bullet_2 \mathbf{V} \bullet_3 \mathbf{W}$.

NOTATION AND VOCABULARY

Among various notation habits in the multiway array processing community, we choose to follow notation from [14], [15], as presented in Tables I and II. Although our results will be applicable to tensors of any order, we focus in this paper on third order tensor in order to simplify the presentation. We call a **third-order** real $K \times L \times M$ **tensor** \mathcal{T} a vector from a tensor space $(\mathbb{R}^K \otimes \mathbb{R}^L \otimes \mathbb{R}^M, \otimes)$ with \otimes being a tensor product. A **three way array** is an element from $(\mathbb{R}^{K \times L \times M}, \otimes)$ where \otimes is the outer product. In other words, the set of arrays is a tensor space when the outer product is used as the tensor product. A higher order tensor is a vector from a tensor space featuring at least three linear subspaces.

In Table II, some useful properties of multilinear operators acting on tensors are specified. Multilinear operators generalize linear operators acting on vectors by defining linear operations on each vector space \mathbb{R}^N composing the tensor space. These multilinear operators also form a tensor space $(\mathbb{R}^{R_1 \times K} \otimes_{op} \mathbb{R}^{R_2 \times K} \otimes_{op} \mathbb{R}^{R_3 \times K}, \otimes_{op})$ where \otimes_{op} will be abusively denoted \otimes although it is not the outer product. Indeed, \otimes_{op} is defined by $(\mathbf{U} \otimes_{op} \mathbf{V})(\mathbf{a} \otimes \mathbf{b}) = \mathbf{U}\mathbf{a} \otimes \mathbf{V}\mathbf{b}$.

Given a $K \times L \times M$ tensor \mathcal{T} , its **canonical polyadic decomposition** (CPD) of rank R can be written as follows:

$$\mathcal{T} = \sum_{r=1}^R \mathcal{D}_r, \quad (1)$$

where \mathcal{D}_r are decomposable tensors of the form $\mathcal{D}_r = \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r$. The **rank** of \mathcal{T} is the minimal value of R such that (1) holds exactly, while we define the rank of a CPD model as the number of component R in that model. A CPD model or a tensor is said to be a **low-rank model** or a low-rank tensor if R is small with respect to all dimensions of the data or tensor.

Finding the CPD of a third-order tensor means finding rank-one tensors \mathcal{D}_r . Yet, each tensor \mathcal{D}_r may be defined by three vectors \mathbf{a}_r , \mathbf{b}_r and \mathbf{c}_r , only up to two scaling ambiguities; in fact, $\mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r = \alpha \mathbf{a}_r \otimes \beta \mathbf{b}_r \otimes \mathbf{c}_r / \alpha\beta, \forall \alpha, \beta \neq 0$.

Next, it is often convenient to store these vectors in matrices called **factors** as $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_R]$ and $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_R]$. This leads to a convenient writing:

$$\mathcal{T} = (\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}) \mathcal{I}_R, \quad (2)$$

where $\mathbf{A} \in \mathbb{R}^{K \times R}$, $\mathbf{B} \in \mathbb{R}^{L \times R}$ and $\mathbf{C} \in \mathbb{R}^{M \times R}$ are called factor matrices, and $\mathcal{I}_R = \sum_{r=1}^R \mathbf{e}_i \otimes \mathbf{e}_i \otimes \mathbf{e}_i$ belongs to $\mathbb{R}^{R \times R \times R}$ with \mathbf{e}_i a canonical basis vector of \mathbb{R}^R , that is, \mathcal{I}_R is a diagonal core tensor with only ones on the diagonal. A convenient interpretation of (2) is to see it as a change of basis. Indeed, (2) means that the vector \mathcal{T} is expressed by coefficients \mathcal{I}_R in the image of multilinear operator $\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C}$, with linear operators \mathbf{A} , \mathbf{B} and \mathbf{C} spanning respectively the first, second and third **mode** of \mathcal{T} . Note that model (2) now contains $2R$ scaling indeterminacies whereas definition (1) did not contain any.

Conditions on the dimensions of the tensor and rank of the decomposition are given in the literature [16]–[18] to ensure uniqueness of the factors in an unconstrained CP model, but only when noise is absent. When these conditions are satisfied, the factors in the CPD model are unique up to permutation ambiguity and the scaling described above. The model is then said to be **identifiable**. For higher order tensors, that is, arrays of three ways and more, these conditions are mild.

Matrices are a particular case of tensors with only two modes. One of the main differences of the CPD for matrices compared with the CPD of higher order tensors is that it is never identifiable without additional constraints as soon as $R > 1$. Note that, since the two modes of a matrix are the column space and the row space, the CPD of a matrix can be written as $\mathbf{M} = \mathbf{A}\mathbf{B}^T$.

A difficult problem not addressed in this manuscript is finding the rank of a tensor, in both an exact decomposition and an approximate decomposition scenario. A naive approach is to look at the singular value profiles of the unfoldings, but an interested reader may refer to [20], [21] and references therein.

A tool that needs to be introduced here is the unfolding of three-way arrays. By choosing three particular ways to cut the cube of data into slices and stacking the obtained matrices, it is possible to rewrite the CPD in a matrix format. We chose the unfolding defined in [15], yielding the following matrix format CPD :

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{A} (\mathbf{B} \odot \mathbf{C})^T, \\ \mathbf{T}_2 &= \mathbf{B} (\mathbf{A} \odot \mathbf{C})^T, \\ \mathbf{T}_3 &= \mathbf{C} (\mathbf{A} \odot \mathbf{B})^T. \end{aligned} \quad (3)$$

Unfolding \mathbf{T}_i contains all the vectors along the mode i . For instance, \mathbf{T}_1 contains all the $L \times M$ columns of \mathcal{T} , thus its column space is the subspace spanned on the first mode by \mathcal{T} , that is, the span of columns of \mathbf{A} .

II. DICTIONARY CPD : MODEL DEFINITION

An important property of low-rank models is that the computed factors bear physical meaning. Factors are interpretable when the multilinear relationship between the experimental parameters and the block of parameters of interest stems from a meaningful modeling, and when the model is identifiable.

The first condition is met in many applications of low-rank models, for instance in fluorescence spectroscopy where spectra and concentrations depend linearly, as a first-order approximation, on the emission wavelength, the excitation wavelength and the mixture index [22]. When mining a collection of text documents, low-rank factorization techniques can identify characteristic words and documents [23].

The identifiability condition may however not be verified in practice. For matrices, it is indeed well known that identifiability is never achieved without imposing additional constraints for R greater than one, since

$$M = AB^T = APP^{-1}B^T \quad (4)$$

for any invertible matrix P . In the higher-order case, conditions on the rank of the model and the dimensions of the data have been reported in the literature with some variations, but are mild enough to ensure identifiability of the CPD in most applications [16]–[18].

If the factors are meaningful, it may be of crucial importance to identify them. In the example of fluorescence spectroscopy, once emission and excitation spectra have been extracted from the data using a CPD model, the final step is to recognize which chemical compound is present in the mixtures by matching the extracted spectra with known spectra of known chemicals. The goal of the models presented below is to merge this identification step with the source separation procedure using the *a priori* information available on the factors.

A. Dictionary-Based CPD

Formally, assume the following relationship between a matrix $D \in \mathbb{R}^{L \times d}$ called the (over-complete) dictionary and factor B :

$$B = DS, \quad \|s_i\|_0 = 1 \text{ for } i \in \{1, \dots, R\}, \quad (5)$$

where $S \in \{0, 1\}^{d \times R}$ is a binary matrix, $\|s_i\|_0$ counts the number of non-zero values in the i th column of S , and d is much larger than R . Here S has exactly one 1 in each column and is a selection matrix that identifies, among the d atoms of the dictionary, R atoms present in factor B . The chosen columns of D that constitute B are given by the row index of all the ones in S . A priori, no restrictions on the atoms are assumed although some technical conditions to ensure identifiability in particular settings are discussed in Section IV.

Viewing (5) as a parameterization of B , a **dictionary CPD model (DCPD)** can be written as follows:

$$\begin{cases} \mathcal{T} = (A \otimes DS \otimes C) \mathcal{I}_R + \mathcal{E}, \\ \|s_i\|_0 = 1 \text{ for } i \in \{1, \dots, R\}, S \in \{0, 1\}^{d \times R}. \end{cases} \quad (6)$$

In this model, the parameters are A , S and C . Tensor \mathcal{E} represent the noise, and typically its entries follow i.i.d. known distributions such as a zero-mean Gaussian distribution. In that case, it is possible to derive the maximum likelihood estimator

of parameters A , S and C :

$$\begin{aligned} & \underset{A, S, C}{\operatorname{argmin}} \| \mathcal{T} - (A \otimes DS \otimes C) \mathcal{I}_R \|_F^2 \\ & \text{such that } \|s_i\|_0 = 1 \text{ for } i \in \{1, \dots, R\} \text{ and } S \in \{0, 1\}^{d \times R}. \end{aligned} \quad (7)$$

Of course, the DCPD does not apply in any situation where the CPD works. It is far from obvious that for a given application, a dictionary containing exactly all the factors on one mode is available. Even if such a dictionary is available, it is likely that some variability has to be accounted for between the atoms and the factors. Below, we try to tackle respectively cases where the relation (5) should not be exact, and cases where no dictionary is available.

B. Accounting for Variability

In spectral unmixing, two different approaches have been studied in the literature: the fully blind unmixing and the semi-blind approach, using already known spectra to compute regression [24]. However in this particular application, it is well known that the same material emits a slightly different spectrum depending on additional parameters not accounted for in a low-rank model. These variations are called spectral variability and are a major issue to using both blind unmixing and semi-blind unmixing [25]. Learning from this example, we emphasize that a naive DCPD as introduced above in (6) may be viewed as unrealistic.

Therefore, it may be necessary to introduce flexibility in the DCPD model. A first way is to generalize the relationship between factor B and the dictionary D :

$$B = f(DS, \theta), \quad (8)$$

for some function $f(x, \theta)$ mapping to $\mathbb{R}^{L \times R}$ and where θ is a random variable following some known probability $p(\theta)$. A simple instance of (8) is obtained by setting $f(x, \theta) = x + \theta$ and $p(\theta)$ is a Gaussian distribution of zero mean and known white covariance $\sigma_c \mathbf{I}_K \otimes \mathbf{I}_R$. Then (8) is a noisy version of (5) and interpreting (8) as *a priori* information on B , a maximum *a posteriori* estimator yields (MAP derivation similar to [26]):

$$\begin{aligned} & \underset{A, B, S, C}{\operatorname{argmin}} \| \mathcal{T} - (A \otimes B \otimes C) \mathcal{I}_R \|_F^2 + \frac{1}{\sigma_c^2} \| B - DS \|_F^2 \\ & \text{such that } \|s_i\|_0 = 1 \text{ for } i \in \{1, \dots, R\} \text{ and } S \in \{0, 1\}^{d \times R}. \end{aligned} \quad (9)$$

In the case where atoms can be grouped with sufficient group population, the user may want to identify the columns of factor B with a group of atoms rather than with a single element of D . This is the method used in [27] for hyperspectral imaging, where the dictionary contains multiple spectra for each class of mineral. In this scenario, a solution is to cluster the available atoms and use the centroids as columns of D along with inter-class covariance. Then comparing B to multiple classes of atoms with known averages and covariances amounts to penalizing using the Mahalanobis distance [27] in (9).

Dictionary-Based PARAFAC2: There are other ways to account for discrepancy between the dictionary and the true phenomena underlying the data. In particular, it is not unreasonable that this discrepancy depends on the other experimental parameters (that is, the first or third mode in our case). For instance, in hyperspectral imaging, a modification of the geometry of the ground due to weather conditions can modify the spectra. If the data is collected along time, then the spectrum of a particular material can relate to one atom in the dictionary but with a time-dependent discrepancy. Since modeling variability is a topic-dependent task, we only provide one instance of a modified DCPD that handles this task, to serve as a guide for further works.

A well-known modified CPD model that accounts for variability is the PARAFAC2 model [28], [29]. Here we show that the PARAFAC2 model can be adapted incorporating dictionary information. The model is the following: for all positive integer m smaller than M ,

$$\begin{cases} M_m = A \text{Diag}(\mathbf{c}_m) B_m^T, \\ B_m = P_m Q, \\ P_m^T P_m = I_R, \\ Q = DS, \\ \|\mathbf{s}_i\|_0 = 1 \text{ for } i \in \{1, \dots, R\}, \mathbf{S} \in \{0, 1\}^{d \times R}, \end{cases} \quad (10)$$

where $M_m \in \mathbb{R}^{K \times L}$ is the m th slice of data tensor \mathcal{T} along the third mode, \mathbf{c}_m is the m th row of a matrix of parameters \mathbf{C} , matrices P_m with orthogonal columns are unknown and Q is an unknown latent factor related to all the factors B_m . Dictionary D has sizes $D \times d$ where D can be different from M .

Here the three-way data is seen as a collection of matrices with shared first mode factor A and similar factors $B_m = P_m Q$. The matrix Q stands for a latent shared factor among matrices B_m . To include additional knowledge provided by the dictionary, it is assumed that this Q follows equation (5). Because the columns of P are orthogonal, the underlying hypothesis here is that all B_m have the same covariance matrix, which is a relaxed assumption with respect to the underlying hypothesis in (6), that is, $B_m = Q$ for all m .

An hidden advantage of the PARAFAC2 version of DCPD is that row dimension of the dictionary can be different from the number of samples on the second mode, since P_m does not need to be square. In practice this means that when using the PARAFAC2 dictionary model, the measured spectral bands can be different than the spectral bands at which atoms are sampled.

C. Self-Dictionary

If no dictionary is available on any of the factor, it is still possible in some specific cases to obtain the dictionary from the data. The topic of learning a dictionary from auxiliary data will be dealt with in future works. Rather, we focus here on the **separability** assumption [30], [31], that is when the data itself can be used as a dictionary. In the matrix case, the separability has been well studied in the context of nonnegative matrix

factorization (NMF), see next section for details. With the suggested formalism, the self-dictionary CPD for matrices is written as follows:

$$M = A (M^T S)^T, \quad (11)$$

where $\|\mathbf{s}_i\|_0 = 1$ for $i \in \{1, \dots, R\}$, $\mathbf{S} \in \{0, 1\}^{d \times R}$. A working hypothesis of this model for matrices is that the second mode factor is a subset of the rows of M .

A generalization of the separability assumption to higher order tensors is however not straightforward. Recently, an attempt was made to define ‘‘pure slices’’ [32], but another possible generalization is obtained by supposing the columns of factor B are contained in all the $K \times M$ columns of the unfolding matrix T_2 . The drawback of this model is the possibly very large number of correlated atoms in the obtained dictionary T_2 . There are various approaches to extend the separability assumption to high-order tensors, but this paper only deals with the case where $D = T_2$. Considering other variants is a direction for further research.

III. RELATED WORKS

A. Sparse Coding

Dictionary CPD presented in this paper can be seen as an extension of sparse coding for higher order tensors, see for instance [8], [9]. In sparse coding, a dictionary D is available, and the data is to be expressed as linear combinations of a small number of R atoms:

$$M = DX, \quad (12)$$

where X is imposed to have at most R non-zero rows. This constraint can be formulated using the ℓ_0/ℓ_q pseudo norm of X^T for any $q \geq 0$, where the ℓ_p/ℓ_q norm of a matrix is defined as

$$\|X\|_{\ell_p/\ell_q} = \|v\|_p, \text{ where } v_i = \|\mathbf{x}_i\|_q \forall i. \quad (13)$$

In fact, X has at most R non-zero rows if and only if $\|X^T\|_{\ell_0/\ell_q} \leq R$.

This model is equivalent to dictionary CPD for matrices by setting $SB^T = X$. Nevertheless there is a crucial difference between the two formulations:

- By splitting the variable X into two variables SB^T , the linear constraint $M = DX$ becomes non-convex with respect to S or B .
- The sparse coding formulation involves a large number of variables, $d \times m$ if M is n by m , whereas the DCPD formulation for matrices involves $(d + m) \times R$ parameters.

In summary, sparse coding involves much more parameters but yields a convex parametrization of the data.

In practice, because of the presence of noise, the constraint $M = DX$ is replaced by the minimization of $\|M - DX\|$ (which is convex for any norm $\|\cdot\|$). Since the mixed pseudo norm ℓ_0/ℓ_q is non-convex, most methods for computing sparse coding rely on convex relaxations. Row-sparsity of X can be achieved in different ways; in particular using convexifications based on the ℓ_1 norm, e.g., ℓ_1/ℓ_2 [9], [33], ℓ_1/ℓ_∞ [34], or using

more sophisticated models [35], [36]. As far as we know, it is the first time sparse coding is tackled using the reformulation $\mathbf{X} = \mathbf{S}\mathbf{B}^T$. Although the resulting problem cannot be relaxed easily, it involves significantly fewer variables hence will be applicable to large-scale problems.

Recently, Salhoun *et al.* suggested a direct extension of sparse coding for higher order tensors [37] in the particular context of harmonic retrievals, but an obvious difficulty is that the row-sparsity has to be imposed on Khatri-Rao products of factors.

B. NMF with Self Dictionary in Spectral Unmixing

Similarly, self dictionaries have been studied for matrices using the sparse coding formalism, by setting $\mathbf{D} = \mathbf{M}$. This is often called the pure-pixel assumption in spectral unmixing [24], since the self-dictionary model assumes some columns of the matrix \mathbf{M} are not mixtures of more than one column of \mathbf{B} . As far as we know, there are mainly two types of approaches to tackle (11):

- *Geometric* approaches that selects the atoms in the dictionary based on some geometric criteria, typically based on the volume of the convex hull of $\mathbf{M}^T\mathbf{S}$. These approaches include for example vertex component analysis (VCA) [11] and the successive projection algorithm (SPA) [13], [38]–[40]. They are usually fast, running in $\mathcal{O}(mnR)$ operations. However, they do not always select atoms leading to a small data fitting term $\|\mathbf{M} - \mathbf{A}(\mathbf{M}^T\mathbf{S})^T\|_F$ since, most of them do not take it into account directly, as they usually put an emphasis on some geometric properties of $\mathbf{M}^T\mathbf{S}$ (such as having a large volume). In particular, these methods are in general sensitive to outliers.
- *Sparse regression* approaches that are based on the sparse coding reformulation of (11)

$$\min_{\mathbf{X} \in \mathbb{R}_+^{d \times n}} \|\mathbf{M} - \mathbf{X}\mathbf{M}\|_F^2$$

such that \mathbf{X} has r non-zero columns,

and achieve column sparsity constraints on the scores \mathbf{X} in different ways [9], [33]–[36]. These methods have the advantage to better model (11) than geometric approaches as they take into account the data fitting term explicitly. They usually provide good solutions but are rather costly as an optimization problem in dn variables must be solved, where n stands for the column dimension of \mathbf{M} . In particular, since here $\mathbf{D} = \mathbf{M}^T$, we have $d = n$ hence n^2 variables. In hyperspectral unmixing, n is the order of millions and these approaches are impractical. Hence pixels have to be selected in a preprocessing step [41] (e.g., using a geometric approach). Moreover, the problem solved is an approximation of the original problem, which results may not be as close as desired to the solutions of the non-convex problem.

In Section V, we describe several algorithms to tackle the proposed formulation (11). They combine the advantages of the two types of approaches described above: they are fast, running

in $\mathcal{O}(mnR)$ operations, but taking explicitly the data fitting term $\|\mathbf{M} - \mathbf{X}\mathbf{M}\|_F^2$ into account.

C. Constrained Tensor Decompositions

The DCPD can be understood as a constrained CPD model. It is similar in spirit to computing CPD when a basis of representation is known for one of the factors, which typically happens when using Tucker Decomposition [42] or a basis of splines [43] for compression. Other linearly constrained tensor decomposition model are obtained when the components are linearly dependent [44], or when the factors are structured, e.g. Hankel or Toeplitz matrices [45].

From a constrained tensor decomposition perspective, the novelty of the present work is that the dictionary is overcomplete and is therefore not a basis of the factor space. Thus sparsity constraints are imposed on the coefficients and our approach is rather combinatorial. We have already published some preliminary results focusing on matrices and applications to spectral unmixing [46].

D. Parameterized Factors in Tensor Decomposition

Using a dictionary to help with recovering factor \mathbf{B} of the CPD is also closely related to parameterizing the columns of that factor. Parameterization is a viable option when an analytical formulation of the atoms of the dictionary is possible. The dictionary is then a continuous dictionary, see for instance [47, Section IV] and references therein. This continuous parameterization may yet not always be achievable in real-life applications. The motivation behind the two methods is however the same: reduce the number of degrees of freedom in the tensor decomposition model by providing a set of admissible solutions to improve estimation accuracy, and to restore identifiability in some pathological cases, some of which are discussed in the next section.

E. Abundances Estimation in Hyperspectral Unmixing

In the context of spectral unmixing of hyperspectral images, using a known library of spectra to estimate the second factor \mathbf{B} is a widely studied topic. Matrix \mathbf{A} contains the abundances and refers to relative concentrations of materials on each pixel. Except for the methods described above, the most widely used techniques to compute \mathbf{A} when \mathbf{D} is known are, to the best of our knowledge, MESMA [48], MELSUM [49], BSMA [50] and AutoMCU [51], which all rely on a more or less exhaustive search of all combinations of atoms in \mathbf{D} , except AutoMCU which only draws randomly a subset of possible atoms. For all possible combinations, abundances are computed, and the best abundances are those that minimize reconstruction error in addition to satisfying some interpretability criteria.

The proposed approach differs from these techniques since it merges blind source separation techniques for spectral unmixing, namely NMF, but featuring atom selection. To the best of our knowledge, our approach has not been described yet in the

spectral unmixing literature, and is bound to be computationally less expensive than an exhaustive search.

IV. IDENTIFIABILITY

The following section contains partial results on the identifiability of the DCPD model, cast in particular cases of interest. In propositions 1, 3 and 4 below, the assumption that no atoms are picked twice is made. This assumption is necessary for proving the identifiability of the DCPD parameters, but should not be necessarily imposed in the higher-order case. For instance, time series of hyperspectral images may require multiple abundance and time components for a single material when decomposed with the CPD model.

A. Matrix Case

It is well known that for matrices, the low rank CPD model is not identifiable because of the rotation ambiguity, see equation (4). However, when a dictionary is available for one of the modes, this rotation ambiguity may be fixed given some conditions on D :

Proposition 1: Let M be a real $n \times m$ matrix of rank R , and let D be a real $n \times d$ matrix with $\text{spark}(D) > R$ where $\text{spark}(D)$ is the minimum integer k such that at least one subset of k columns of D is rank-deficient. If there exist a full column-rank $S \in \{0, 1\}^{d \times R}$ with column sparsity set to 1, and $A \in \mathbb{R}^{n \times R}$ with nonzero columns such that $M = A(DS)^T$, then S and A are unique up to permutation ambiguity.

Proof: Since $\text{spark}(D) > R$ and S is full column rank, $B = DS$ has full column rank hence A is unique up to permutations in the decomposition $M = AB^T$ if B is unique up to permutations. Moreover, given $M = AB^T$, because A has no zero columns, the column space of B and the row space of M are equal. Such a B is built by selecting R atoms in D that both span and belong to the row space of M . Because the span of D is strictly larger than R , there is only one such set of R atoms. Thus B is unique up to permutation. ■

B. CPD and DCPD Uniqueness

For all applications where the identifiability of the CPD is usually verified, a natural question to ask is whether the DCPD will automatically be identifiable or if some additional conditions need to be checked before trying to use the DCPD model. It turns out that given the uniqueness of the CPD up to scaling and permutations of the factors, the only requirement to obtain uniqueness of the DPCD is the uniqueness of the factorization $B = DS$, which itself is quite simple to check.

Proposition 2: Let $\mathcal{T} \in \mathbb{R}^{K \times L \times M}$ be such that \mathcal{T} admits a unique rank R CPD up to scaling and permutations. Then if D does not contain collinear atoms and there exist S verifying (5), the DCPD is also unique up to scaling and permutations.

Proof: Because the CPD of $\mathcal{T} = (A \otimes B \otimes C)\mathcal{I}$ is unique, the DCPD is unique if and only if $\Omega = \{S | B = DS\}$ is a singleton. The set Ω is not empty by assumption, and if S_1, S_2 belong to Ω , then the columns they select in D have to be collinear (or equal if there was no scaling ambiguity on B).

By hypothesis this implies $S_1 = S_2\Pi$, where Π is a column permutation matrix. ■

The existence of S is what is really difficult to assess in practice. On the other hand, the dictionary may contain very correlated atoms, but not exactly collinear atoms, so that the DCPD as a model is identifiable in practice whenever the CPD is identifiable.

Moreover, using proposition 1, it is possible to derive a mild sufficient condition for uniqueness of DCPD.

Proposition 3: Let \mathcal{T} be a real $K \times L \times M$ tensor of rank R and D a real $L \times d$ matrix. If there exist a full column-rank $S \in \{0, 1\}^{d \times R}$ with column sparsity set to 1, and $A \in \mathbb{R}^{K \times R}$ and $C \in \mathbb{R}^{M \times R}$ such that $\mathcal{T} = (A \otimes DS \otimes C)\mathcal{I}_R$, if $\text{spark}(D) > R$ and if $A \odot C$ is full column-rank and A and C do not have zero columns, then S , A and C are unique up to permutation and scaling ambiguity.

Proof: By applying Proposition 1 to the rank R matrix $T_2 = DS(A \odot C)^T$, S and the Khatri-Rao product $P = A \odot C$ are unique up to permutation ambiguity. It remains to prove that the decomposition of P into $A \odot C$ is unique provided that the dimensions of the problem are fixed. Column-wise, we need to check that the factorization $\text{matr}(p_i) = a_i c_i^T$ is unique. This rank-one approximation is moreover up to scaling, and because this fixed matricization operator is an isomorphism, the decomposition of P is therefore unique. ■

An important remark is that Proposition 3 is true for any order, since rank-one decompositions are always unique up to scaling. Similar discussion on identifiability when one factor is unique can be found in [52] and references therein. Moreover, if A has many collinear columns as may be the case with spectra in spectral unmixing, if A has no zero column and C is full column rank, then $A \odot C$ is full column-rank and the DCPD model does not suffer from the rotation ambiguity inherent to the CPD with collinear columns in factors. Collinear columns in factors appear when multiple columns of a factor in one mode are necessary to express the evolution of only one physically meaningful component.

C. Existence of the Best Low-Rank DCPD

For third-order tensors, another important advantage of DCPD is that it makes the optimization problem well-posed. In fact, for CPD, the optimal solution may not exist as the feasible set is open; see for example [53].

Proposition 4: Let $\text{spark}(D) > R$ and impose that S is full column rank, then the optimal solution of (7) is attained.

Proof: Since $\text{spark}(D) > R$, for any full column rank S , $B = DS$ has rank R . Note that there are a finite number of such S . It remains to show that for any B of rank R , the infimum of

$$\inf_{A, C} \|\mathcal{T} - (A \otimes B \otimes C)\mathcal{I}_R\|_F^2$$

is attained. Without loss of generality, we can assume that $\|a_r\|_2 = 1$ for $1 \leq r \leq R$ by the scaling degree of freedom of each rank-one tensor $\mathcal{D}_r = a_r \otimes b_r \otimes c_r$. Using unfolding, the infimum of the above problem is attained if and only if the

infimum of

$$\inf_{\mathbf{A}, \|\mathbf{a}_r\|_2 = 1 \forall r, \mathbf{C}} \|\mathbf{T}_3 - \mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F^2$$

is attained. Moreover, since $\mathbf{C} = 0$ is a feasible solution, we can add the constraint

$$\|\mathbf{T}_3 - \mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F \leq \|\mathbf{T}_3\|_F$$

implying $\|\mathbf{C}(\mathbf{A} \odot \mathbf{B})^T\|_F \leq 2\|\mathbf{T}_3\|_F$. For $\text{rank}(\mathbf{B}) = R$ and $\|\mathbf{a}_r\|_2 = 1 \forall r$, we can show that

$$\sigma_R(\mathbf{A} \odot \mathbf{B}) \geq \sigma_R(\mathbf{B}) > 0,$$

where $\sigma_R(\mathbf{B})$ is the R th singular value of \mathbf{B} . In fact, denoting \mathbf{a}^j the j th row of \mathbf{A} ,

$$\begin{aligned} \sigma_R^2(\mathbf{A} \odot \mathbf{B}) &= \min_{\|x\|_2=1} \|(\mathbf{A} \odot \mathbf{B})x\|_2^2 \\ &= \min_{\|x\|_2=1} \sum_j \|B(x \boxtimes \mathbf{a}^j)\|_2^2 \\ &\geq \sigma_R^2(\mathbf{B}) \min_{\|x\|_2=1} \sum_j \|x \boxtimes \mathbf{a}^j\|_2^2 \\ &= \sigma_R^2(\mathbf{B}) \min_{\|x\|_2=1} \sum_{r=1}^R |x_r| \|\mathbf{a}_r\|_2^2 = \sigma_R^2(\mathbf{B}). \end{aligned}$$

Finally, $\|\mathbf{C}\|_F \leq 2 \frac{\|\mathbf{T}_3\|_F}{\sigma_{\min}(\mathbf{B})}$ hence the feasible set can be reduced to a compact set hence the infimum is attained since the objective function is continuous and bounded below. ■

V. DECOMPOSITION ALGORITHMS : GREEDY AND NON-GREEDY APPROACHES

In the literature of sparse approximation, two families of algorithms have been studied extensively: greedy approaches based on matching pursuit, and continuous approaches based on convex relaxations of the ℓ_0 pseudo norm [54], [55]. In the same spirit, we develop in the next two sections the two same kind of algorithms to attack (6). We will compare these approaches in Section VI.

This paper does not explicitly discuss algorithms for computing the CPD itself. In DCPD, factor matrices \mathbf{A} and \mathbf{C} can be estimated using any off-the-shelf CPD algorithm assuming \mathbf{B} is fixed. This can be done for instance using the alternating least squares procedure in the unconstrained case, or exact non-negative least squares for non-negative CPD [56], [57]. Therefore, below, only the estimation of \mathbf{S} and \mathbf{B} for fixed \mathbf{A} and \mathbf{C} is discussed.²

A. Greedy Algorithms: Matching Pursuit

Let us first provide an algorithm to compute the minimum (7) with respect to \mathbf{S} . Since the set of solution for \mathbf{S} is discrete, the underlying optimization problem is combinatorial. On the other hand, computing the unconstrained CPD can be done efficiently using alternating least squares (ALS). We want to take

advantage of both unconstrained CPD and greedy algorithms for computation efficiency when estimating respectively factors \mathbf{A} , \mathbf{B} , \mathbf{C} and the selection matrix \mathbf{S} .

This is what the matching pursuit-ALS (MPALS) described below does. The variable \mathbf{B} is injected in the problem to be a proxy of $\mathbf{D}\mathbf{S}$. The matrix \mathbf{B} is estimated using the least square update³

$$\widehat{\mathbf{B}} = \mathbf{T}_2 (\mathbf{A} \odot \mathbf{C})^\dagger = \mathbf{T}_2 (\mathbf{A} \odot \mathbf{C}) (\mathbf{A}^T \mathbf{A} \boxtimes \mathbf{C}^T \mathbf{C})^{-1}, \quad (14)$$

and $\widehat{\mathbf{S}}$ is then evaluated by choosing the closest atom in \mathbf{D} up to a scaling factor. Finally \mathbf{B} is reevaluated as $\widehat{\mathbf{B}} = \mathbf{D}\widehat{\mathbf{S}}$. In other words, \mathbf{B} is estimated through a projected least squares update, where the projection space is the set spanned by $\mathbf{D}\mathbf{S}$ for all \mathbf{S} with coefficients in $\{0, 1\}$ and column sparsity set to 1.

Like most projected ALS algorithms such as the ALS algorithm for NMF⁴ [58], convergence of the global MPALS algorithm, that is including the alternating least squares estimation of \mathbf{A} and \mathbf{C} , cannot be ensured since the cost function is not guaranteed to decrease at each step. Indeed, the least squares update of \mathbf{B} decreases the cost function, but the projection step increases it. Therefore MPALS is bound to have few provable results in term of convergence, contrary to continuous algorithms presented below. It is however a very simple algorithm to implement with no parameter to tune and it provides good results in both simulated and real data experiments reported in Section VI. In particular, we observed convergence in practice, and the cost function decreases for most iterations in simulations. The complexity of $\mathcal{O}(RKL M + RLd)$ operations per iteration of MPALS inside an ALS algorithm is about the same as plain ALS if the dictionary is not excessively large.

Note that the term greedy is a bit abusive since an atom chosen to belong to \mathbf{B} at some iteration of the global procedure may be discarded at a further iteration. What is greedy in MP-ALS is the procedure to choose atoms in the dictionary at each inner iteration. Also, MP-ALS can be easily adapted if the constraint on the number of elements in \mathbf{S} is modified to allow for more than one atom to be used to approximate the columns of \mathbf{B} , thus the borrowed name matching pursuit.

MPALS can also easily be adapted to tackle other similar optimization problems. If the factors are constrained to be non-negative, then the estimates of factors \mathbf{A} and \mathbf{C} can be obtained by nonnegative least squares as mentioned earlier, while factor \mathbf{B} should be nonnegative since the dictionary should be non-negative in this case. Also, if no column of \mathbf{D} may be selected twice, then after computing the scores $\max_j \frac{\langle \mathbf{B}_i, \mathbf{D}_j \rangle}{\|\mathbf{D}_j\|}$ for each column of \mathbf{B} , the atoms are assigned to each such column by solving an assignment problem in order to maximize the sum of the scores [59].

A smooth version of MPALS: MPALS features a projection of \mathbf{B} on the atoms of the dictionary at each outer iteration, which is a very rough way to impose the sparsity constraint on the

³The inverse is not actually computed, rather we solve the least square problem using any efficient solver. The bottleneck here is the large product $\mathbf{T}_2 (\mathbf{A} \odot \mathbf{C})$ since $R \ll \min(K, L, M)$.

⁴In Matlab, this is the 'als' algorithm of 'nmf'.

²Algorithms introduced in this section are available at <https://jeremy-cohen.jimdo.com/downloads/>

Algorithm 1: Matching Pursuit Alternating Least Squares.**INPUT:** array \mathcal{T} , factors \mathbf{A} and \mathbf{C} , dictionary \mathbf{D} . **B estimate:** $\mathbf{B} = \mathcal{T}_2(\mathbf{A} \odot \mathbf{C})(\mathbf{A}^\top \mathbf{A} \boxminus \mathbf{C}^\top \mathbf{C})^{-1}$ **S estimate:****for** i from 1 to R **do**

$$S_{j^*i} = 1 \iff j^* = \operatorname{argmax}_j \frac{\langle \mathbf{B}_i | \mathbf{D}_j \rangle}{\|\mathbf{D}_j\|}$$

end for **B reevaluation:** $\mathbf{B} = \mathbf{D}\mathbf{S}$ **OUTPUT:** Estimated scores \mathbf{S} and factor $\mathbf{B} = \mathbf{D}\mathbf{S}$.**Algorithm 2: Smooth Matching Pursuit Alternating Least Squares.****INPUT:** array \mathcal{T} , factors \mathbf{A} and \mathbf{C} , dictionary \mathbf{D} , coupling parameter $\lambda > 0$ and update rate $p > 1$. **B least squares estimate:**

$$\mathbf{B} = (\mathcal{T}_{(2)}(\mathbf{A} \odot \mathbf{C}) + \lambda \mathbf{D}\mathbf{S})(\mathbf{A}^\top \mathbf{A} \boxminus \mathbf{C}^\top \mathbf{C} + \lambda \mathbf{I}_{R \times R})^{-1}$$

 S estimate:**for** i from 1 to R **do**

$$S_{j^*i} = 1 \iff j^* = \operatorname{argmax}_j \frac{\langle \mathbf{B}_i | \mathbf{D}_j \rangle}{\|\mathbf{D}_j\|}$$

end for **λ update****if** $\|\mathbf{B} - \mathbf{D}\mathbf{S}\|_F^2 > 0.01\|\mathbf{B}\|_F^2$ **then**

$$\lambda = p\lambda$$

end if**OUTPUT:** Estimated scores \mathbf{S} and factor $\mathbf{B} = \mathbf{D}\mathbf{S}$.

scores \mathbf{S} . To obtain a smoother optimization algorithm, we suggest to enforce sparsity constraints in a continuous manner using the flexible formulation of the DCPD for factor \mathbf{B} , while using the projected factor $\mathbf{D}\mathbf{S}$ when estimating the other factors. We call this algorithm smooth MPALS (SMPALS), it is summarized in Algorithm 2. This algorithm also finds the minimum of (9) solving the flexible dictionary problem with Gaussian noise. MPALS is modified in a straightforward manner, by making the least squares update of \mathbf{B} depend on \mathbf{S} , and not reevaluate \mathbf{B} after evaluating \mathbf{S} . This means removing $\mathbf{B} = \mathbf{D}\mathbf{S}$ in Algorithm 1, and replacing \mathbf{B} first estimate with

$$\widehat{\mathbf{B}} = (\mathcal{T}_2(\mathbf{A} \odot \mathbf{C}) + \lambda \mathbf{D}\mathbf{S})(\mathbf{A}^\top \mathbf{A} \boxminus \mathbf{C}^\top \mathbf{C} + \lambda \mathbf{I}_{R \times R})^{-1} \quad (15)$$

where λ is a given parameter, set by the user at the beginning of the algorithm, meant to approach $\frac{1}{\sigma_{\mathcal{T}}^2}$ in (9).

In the alternating outer loop, $\widehat{\mathbf{B}}$ can be set either to the exact $\mathbf{D}\mathbf{S}$, or to the approximate version computed by (15). The first choice provides the algorithm we called SMPALS, while the second is fully flexible, therefore called Flex-MPALS. For Flex-MPALS, λ is also kept constant. In SMPALS, until $\|\mathbf{B} - \mathbf{D}\mathbf{S}\|_F^2 \geq 0.01\|\mathbf{B}\|_F^2$ is reached, the coupling strength λ increases by a multiplicative constant p . We chose a relatively aggressive choice for the increase $p = 1.1$ for third-order tensors. For matrices, even a higher value of $p = 1.5$ gave good results and allowed faster convergence.

Convergence of Proposed Algorithms: Although it is difficult to assess the convergence of MPALS, results for SMPALS and

Flex-MPALS can be derived. The convergence of SMPALS is guaranteed if λ is allowed to grow to infinity. Here is a sketch of the proof. The update of \mathbf{B} can be written as follows; see (15):

$$\mathbf{B} = \mathbf{D}\mathbf{S} + \mathcal{O}\left(\frac{1}{\lambda}\right). \quad (16)$$

This means that \mathbf{B} gets arbitrarily close to $\mathbf{D}\mathbf{S}$ as λ goes to infinity. This implies that, for λ sufficiently large, \mathbf{S} and \mathbf{B} will no longer be modified: if \mathbf{B} does not change sufficiently, \mathbf{S} does not change because of the discrete nature of the problem. In the mean time, the updates of \mathbf{A} and \mathbf{C} , that use $\mathbf{D}\mathbf{S}$, decrease the cost function and converge to a stationary point of the corresponding objective function (for \mathbf{S} fixed) [60].

The convergence of Flex-MPALS, *i.e.* with a fixed λ and no replacement \mathbf{B} by $\mathbf{D}\mathbf{S}$, is also guaranteed for a normalized dictionary. Indeed, Flex-MPALS is a block-coordinate descent algorithm where the blocks \mathbf{A} , \mathbf{B} , \mathbf{C} are updated using an optimal least squares estimate, while the estimated \mathbf{S} in Flex-MPALS minimizes $\|\mathbf{B} - \mathbf{D}\mathbf{S}\|_F^2$ if the atoms of the dictionary have unit norm. Therefore the cost function decreases at each step of Flex-MPALS hence convergence to some value (since it is bounded below by zero).

B. Continuous Approaches

As explained above, the original optimization problem underlying the DCPD model is combinatorial because of the ℓ_0 pseudo-norm and the fact that \mathbf{S} is binary. To develop continuous algorithms such as gradient descent, a first step is to derive a continuous relaxation for the DCPD formulation. The first relaxations that comes to mind when working on ℓ_0/ℓ_p pseudo-norm is a mixed norm ℓ_1/ℓ_2 which encourages sparsity column-wise. However in the DCPD model, \mathbf{S} can only have one non-zero coefficient per column, which makes a ℓ_1/ℓ_2 based optimization difficult to tune. Rather than mixed norm, we choose to use the ℓ_1 norm as a sparsity enhancing penalization on all the entries of \mathbf{S} , but under the constraint that the ℓ_2 norm of the columns of \mathbf{S} are set to one (which is not convex). Indeed, if $\mathcal{B}(1, \ell_2)$ is the unit ball of the ℓ_2 norm, the solutions to

$$\operatorname{argmin}_{x \in \mathcal{B}(1, \ell_2)} \|x\|_1 \quad (17)$$

are exactly located on the coordinate axes, which is also the constraint imposed on the columns of \mathbf{S} . In light of this remark, solely imposing a ℓ_1 penalization on coefficients of \mathbf{S} , that could be easily done with for instance ADMM, may not be sufficient to ensure the high level of column sparsity that is sought.

We chose to use the fast gradient from Nesterov [61] which can be understood as a proximal gradient descent with averaged steps. Although our problem is not convex, this method can still be applied and has been shown to work well in the non-convex case [62]. Fast gradient is used to solve the following subproblem in \mathbf{S}

$$\begin{cases} \min_{\mathbf{S}} \gamma(\mathbf{S}) = \frac{1}{2}\|\mathcal{T} - (\mathbf{A} \otimes \mathbf{D}\mathbf{S} \otimes \mathbf{C})\mathcal{I}_R\|_F^2 + \delta\|\mathbf{S}\|_1 \\ \text{such that } \|\mathbf{s}_i\|_2^2 = 1 \quad \forall i \leq R, \quad \mathbf{S} \geq 0, \end{cases} \quad (18)$$

Algorithm 3: Fast Gradient for Estimating \mathbf{S} .

INPUT: array \mathcal{T} , factors \mathbf{A} , \mathbf{C} , initial scores \mathbf{S} , dictionary \mathbf{D} , regularization parameter $\delta > 0$, fast gradient parameter $\alpha \in]0, 1]$.

step size computation:

$\epsilon_S =$ product of squared largest eigenvalues of $\mathbf{D}^T \mathbf{D}$ and $(\mathbf{A}^T \mathbf{A} \square \mathbf{C}^T \mathbf{C})$

while convergence criterion is not met **do**

gradient computation:

$$\mathbf{g}_S = \mathbf{D}^T \mathbf{D} \mathbf{S} (\mathbf{A}^T \mathbf{A} \square \mathbf{C}^T \mathbf{C}) - \mathbf{D}^T \mathbf{T}_2 (\mathbf{A} \odot \mathbf{C}) + \delta \mathbf{1}_{d \times R}$$

gradient descent correction:

$$\mathbf{S}_{old} = \mathbf{S}$$

$\mathbf{S} = \max(0, \mathbf{S} - \text{step } \mathbf{g}_S)$ where step is computed as in (20) to guarantee $s_i \neq 0$ for all i

normalize columns of \mathbf{S} using the ℓ_2 norm

$$\alpha_{old} = \alpha$$

$$\alpha = \frac{1}{2}(-\alpha_{old}^2 + \sqrt{\alpha_{old}^4 + 4\alpha_{old}})$$

$$\beta = \frac{\alpha_{old}(1-\alpha_{old})}{(\alpha_{old}^2 + \alpha)}$$

update:

$$\mathbf{S} = \mathbf{S} + \beta(\mathbf{S} - \mathbf{S}_{old})$$

$$\mathbf{B} = \mathbf{D} \mathbf{S}$$

end while

OUTPUT: Estimated factor and scores \mathbf{B} and \mathbf{S} .

for a given parameter δ fixed by the user, and where s_i is the i th column of \mathbf{S} . The non-negativity constraint on \mathbf{S} makes γ differentiable with respect to \mathbf{S} :

$$\frac{\partial \gamma}{\partial \mathbf{S}} = \mathbf{D}^T \mathbf{D} \mathbf{S} (\mathbf{A}^T \mathbf{A} \square \mathbf{C}^T \mathbf{C}) - \mathbf{D}^T \mathbf{T}_2 (\mathbf{A} \odot \mathbf{C}) + \delta \mathbf{1}_{d \times R} \quad (19)$$

and the gradients of (18) with respect to \mathbf{A} and \mathbf{C} can be found for instance in [63] (we set $\mathbf{B} = \mathbf{D} \mathbf{S}$) if an all-at-once optimization is sought. The normalization constraint on the columns of \mathbf{S} is imposed by normalization of \mathbf{S} at each iteration. To avoid all values of a column of \mathbf{S} to be non-positive, which would make the normalization meaningless, the gradient step is constrained as follows to ensure all columns of \mathbf{S} after the gradient update have at least one positive entry:

$$\text{step} = \min \left(\frac{1}{\epsilon_S}, \min_j \max_{i | g_S(i,j) > 0} \frac{S(i,j)}{g_S(i,j)} - 10^{-12} \right) \quad (20)$$

where ϵ_S is the Lipschitz constant and g_S is the gradient of the cost function with respect to \mathbf{S} . $M(i, j)$ refers to the entry of matrix \mathbf{M} indexed by (i, j) .

The global optimization procedure we suggest is therefore a mixture of ALS and fast gradient, denoted ALS-FG and summarized in Algorithm 3. An important remark is that this algorithm is to be used inside an outer loop consisting of least squares problems. To ensure a smooth transition from the unconstrained problem to enforcing the dictionary, the penalty coefficient δ is linearly increased up from zero to a maximum value at the last outer iteration specified by the user. We observed that using the fast gradient improves convergence speed with respect to a simple coordinate descent, but other methods can also be implemented to solve the subproblem. Again, a flexible version of

ALS-FG tackling optimization problem (9) can be derived from Algorithm 3 by adapting the gradient of γ with respect to \mathbf{S} and using \mathbf{B} as a variable.

a) *Stopping Criterion:* Choosing the stopping criterion is application-dependent, and we let this parameter be tuned by interested users. Nevertheless, a baseline that we used in simulations below is to compute the residual error E_i at iteration i of the global optimization procedure, every few iterations, and to check whether the error is still decreasing enough by computing $\frac{|E_i - E_{i-1}|}{E_i}$. We set the number of iterations of the fast gradient inside ALS-FG to 10.

b) *Normalization:* An inherent ambiguity of the CPD model which is also present in the DCPD model is the scaling ambiguity, that is, the norm of columns of the factors is not determined solely by the model. It is possible to fix the norm of factors in both MPALS and ALS-FG by normalizing the columns of factor \mathbf{A} after each update of this factor, but this is not mandatory.

c) *Initialization:* Because the global optimization problems underlying the DCPD model and its flexible counterparts are non-convex and the proposed optimization algorithms work locally, it is *a priori* crucial to use a good initial point. If any information is available on the factors like non-negativity, it should be used in the initialization procedure. However, in the general case where no such information is available, one possible strategy is to compute an unconstrained CPD and use the obtained factors as an initialization for MPALS and ALS-FG. In the experiments below, we observed that indeed MPALS is highly sensitive to initialization. In particular, using a single random initialization leads in many cases to poor results for third-order tensors. Note however that is worked well for hyperspectral images.

VI. EXPERIMENTS ON SIMULATED AND HYPERSPECTRAL DATA

A. Simulated Data Experiments

1) *Methodology:* A critical question yet unanswered is whether using a dictionary within the tensor factorization model improves on identification error, which is the percentage of columns of \mathbf{B} correctly matched to atoms of \mathbf{D} , with respect to only projecting the result of an unconstrained factorization on the set of atoms. Because information is added about factor \mathbf{B} in the model, the identification error is expected to be smaller by using DCPD, in particular if the simulations are made challenging in terms of signal to noise ratio (SNR), rank under-estimation and over-estimation, or conditioning of the factors.

If the SNR is high, if the tensor is well-conditioned and if the rank is known, unconstrained CPD may provide relatively low identification error because of the generic uniqueness of the solution. For this reason, we choose difficult scenarios to compare DCPD to unconstrained CPD with outputs identified using a projection on the set of atoms. The dictionary used is constituted of atoms as follows:

$$\mathbf{d}_k^i = |a_k \mathbf{u} + b_k + \nu_k^i \text{sinc} \left(\frac{\pi}{6} \mathbf{u} - e_k^i \right) + \mu_k^i (\text{tri}(\mathbf{u} - f_k^i + 2) - \text{tri}(\mathbf{u} - f_k^i - 2))| \quad (21)$$

where $\mathbf{u} = \{1, \dots, L\}$, $a_k, b_k, \nu_k^i, \mu_k^i, e_k^i, f_k^i$ are realizations of uniformly distributed random variables on respectively

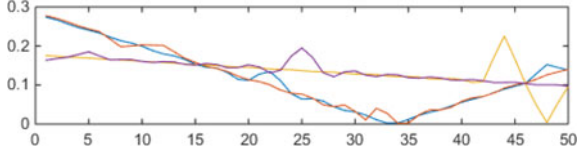


Fig. 1. Four dictionary atoms belonging to two different classes.

$[-1/L, 1/L], [-1, 1], [-1/4, 1/4], [-1/4, 1/4], \{1, \dots, L\}, \{1, \dots, L\}$ and tri is a triangular pulse function, of support $[-2, 2]$. Atoms are then normalized with the ℓ_2 norm. Index $k \in \{1, \dots, c\}$ is a class index, c being the number of classes, and $i \in \{1, \dots, \frac{d}{c}\}$ indexes individual atoms in each class. This means that atoms of the dictionary are a sum of a linear baseline common to groups of atoms and of two individual features; see Figure 1 for an example. This design is meant to bluntly echo spectral signatures of materials in hyperspectral imaging, where only a small number of features discriminate materials from the same family. For the experiments below, a single realization of the dictionary was used, with no collinear columns (as to satisfy Proposition 2), but it could not be checked whether its spark satisfied the condition stated in Proposition 1. However, $\text{spark}(\mathbf{D})$ is equal to the dimension of the atoms plus one with probability one, which ensures that the parameters of the DCPD models used in these simulations are identifiable almost surely.

The data are generated as follows: each entry of the factors \mathbf{A} and \mathbf{C} is drawn according to a unitary centered Gaussian distribution, and the columns of \mathbf{A} and \mathbf{C} are then normalized with the ℓ_2 norm, which makes these factors well-conditioned matrices. Matrix \mathbf{S} is fixed in all experiments so that only one atom of each class is contained in the columns of \mathbf{B} . White Gaussian noise of fixed variance σ is added to the data tensor.

Given the structure of the dictionary, the correlation among atoms of the same class can be quite high. The dictionary is constituted of a large number of atoms ($d = 1000$) subdivided in $c = 50$ groups. The dimensions of the tensors are $20 \times 50 \times 7$, yielding an inter-group correlation of atoms that reaches 0.999 at most. We choose a noise variance of $\sigma = 0.01$ which leads to an average SNR of about 11.5 dB, and set the rank to $R = 10$.

As explained above, the simulations focus on the impact on identification performance when the rank is wrongly estimated, and when one of the factor matrix, here \mathbf{C} , is ill-conditioned. Therefore, we chose to grid over an estimated rank Re ranging from 7 to 13 with a good conditioning of \mathbf{C} . The impact of the conditioning of \mathbf{C} is studied through a grid on a parameter ρ such that given a randomly-drawn well conditioned $\mathbf{C}^{(0)}$, it is modified as follows

$$\mathbf{C} \leftarrow \mathbf{C}^{(0)} \left(\rho \mathbf{I}_R + \frac{(1-\rho)}{R} \mathbf{1}_{R \times R} \right), \quad (22)$$

and is then normalized. When ρ is equal to 0, \mathbf{C} has column rank equal to 1, while its entries follow i.i.d. Gaussian distributions when ρ is 1. The estimated rank is set equal to the true rank when studying the ill-conditioned case. The number of realization in each setting is set to $N = 100$.

The CPD algorithm used as a baseline for comparison and for initialization is the NWAY toolbox [64]. We compare projected results of the NWAY toolbox with MPALS, SMPALS,

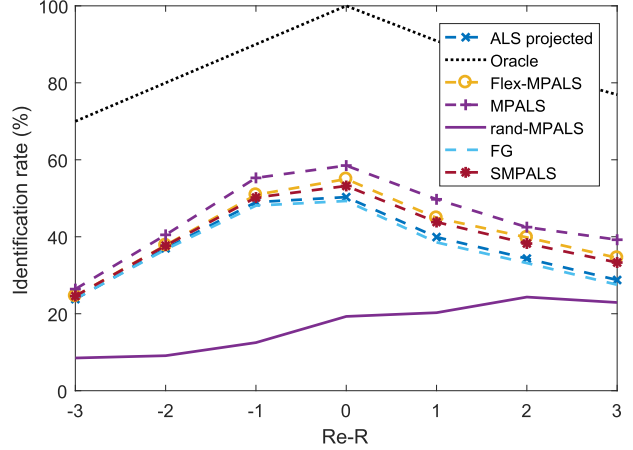


Fig. 2. Identification rates of various methods for varying estimated rank. True rank R is 10. The oracle is the best possible identification rate.

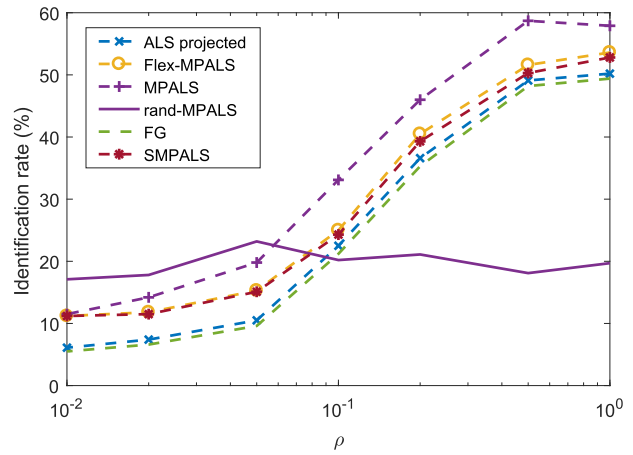


Fig. 3. Identification rates of various methods for various conditioning of factor matrix \mathbf{C} , tuned using parameter ρ defined in (22). True rank R is 10.

Flex-MPALS and ALS-FG. Coupling strength in Flex-MPALS is fixed to $\lambda = 0.04$ in this experiment. The value 0.04 provided spectra close to the dictionary, while allowing some flexibility. In practice, λ could be tuned for example in order to achieve a given relative error between the columns of \mathbf{B} of the corresponding matched atoms of the dictionary (for example 1% relative error). The number of iterations is fixed to a maximum of 1000, except for the rand-MPALS (MPALS initialized randomly) where the maximum iteration number is 5000. Algorithm also stops if the stopping criterion mentioned in the previous section goes below 10^{-4} .

Finally, in both cases we use the NWAY toolbox for initialization, except for the randomly initialized MPALS which is initialized with factors following the same distribution as the true factors.

2) *Results*: Figures 2 and 3 report the mean identification rate that measures the percentage of well-matched columns between \mathbf{B} and $\hat{\mathbf{B}}$. In the case where the rank and the estimated rank are miss-matched, the unmatched factors of \mathbf{B} and $\hat{\mathbf{B}}$ are considered miss-matched. This leads to a best possible identification rate given by $\frac{|Re-R|}{\max(Re, R)}$ and referred to as ‘‘oracle’’. Figure 4 reports the relative mean square error on factor \mathbf{B} given

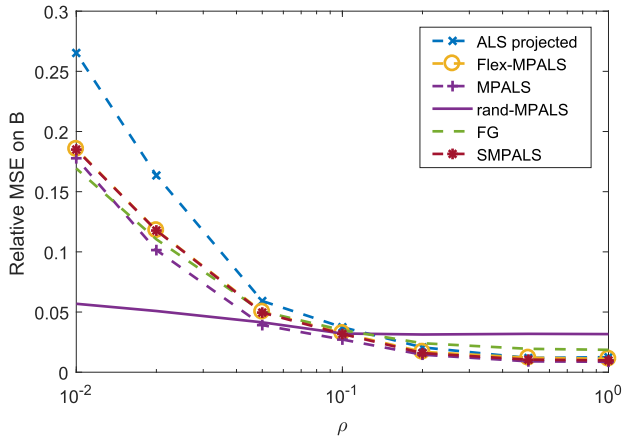


Fig. 4. Relative mean square error on factor \mathbf{B} that quantifies the distance between the selected atoms and the true ones, as a function of ρ .

TABLE III
AVERAGE RUNTIME OF VARIOUS ALGORITHMS OVER $N = 100$ REALIZATION
WHEN $R = Re$

Alg.	ALS	MPALS	Flex-MPALS
Time (s)	0.21	0.56	0.72
Alg.	r-MPALS	SMPALS	FG
Time (s)	2.75	0.73	7.62

by $\mathbb{E}[\frac{\|\mathbf{B}-\hat{\mathbf{B}}\Pi\|_F^2}{\|\mathbf{B}\|_F^2}]$ where Π is a permutation matrix computed to match the estimated factors to the true factors. Table III reports the mean run time of each algorithm in the first experiment when R equals Re .

As expected the identification rate increases with respect to projected NWAY when using MPALS and its variants, with a gain ranging from a few percent to over ten percent. This is a direct but non-obvious consequence of the increased estimation performances. It can be seen that MPALS performs the best in almost all cases, especially when the factor \mathbf{C} is ill-conditioned. However, the wrong estimation of the rank does not impact the gain in performance between projected CPD and DCPD. Notably, SMPALS performs slightly worse than the naive MPALS.

An important result to observe is the steady performance of the randomly initialized MPALS over various values of ρ , both in terms of mean square error and identification rate. Because DCPD is supposed to be identifiable even when ρ tends to 0, the performance of MPALS should not depend too much on it, as observed with the random initialization. However, when the unconstrained CPD model is used as an initialization method, the results of MPALS do depend heavily on ρ . This shows that MPALS and all other proposed algorithm are very sensitive to initialization, a fact also supported by the experiment in the next section. As a consequence, a good initialization method is crucial for the performance of MPALS, this is a topic for further research.

Even though the identification rates evolve similarly for all methods in the ill-conditioning experiment, the mean square error on \mathbf{B} tells a different story. Since many atoms in the dictionary are very correlated, it is reasonable to assume that the mean square error on \mathbf{B} could be small even though all

TABLE IV
RECONSTRUCTION ERROR (%) FOR THE URBAN AND TERRAIN AIRPORT HSI

	Urban HSI		Terrain HSI	
	time(s)	Rel.err.	time(s)	Rel.err.
SPA	0.2	9.58	0.3	5.89
d-SPA	20	4.57	48	3.50
ds-SPA	21	4.67	58	3.37
nmf-SPA	13	5.37	18	3.75
VCA	2.0	13.07	1.6	18.61
d-VCA	16	4.73	46	3.29
ds-VCA	20	4.66	35	3.37
nmf-VCA	12	6.63	17	4.27
SNPA	9.5	9.63	13	5.76
d-SNPA	17	5.02	47	3.78
ds-SNPA	21	4.91	59	3.88
nmf-SNPA	13	5.14	19	4.47
H2NMF	8	5.81	11.7	5.09
d-H2NMF	17	4.05	47	3.49
ds-H2NMF	21	4.05	64	3.49
nmf-H2NMF	11	5.03	17	4.51
FGNSR-100	2.2	5.58	1.6	3.34
d-FGNSR-100	17	4.47	45	3.01
ds-FGNSR-100	20	4.47	58	3.01
nmf-FGNSR-100	12	6.37	19	3.31

Best Results are Highlighted in Bold. Computation Time Does not Include 500 Iterations of Nonnegative Least Squares Update for the Abundances After Each Method (Which Takes About 10 Seconds on Average).

atoms are wrongly identified. It can be observed on Figure 4 that on average, all proposed algorithms for computing the DCPD significantly outperform the projected ALS method when \mathbf{C} is ill-conditioned. As a conclusion, the atoms picked by MPALS and its variants are on average much closer to the true atoms than the ones picked using a projected ALS method. This is especially true for the randomly initialized MPALS.

Finally, from this experiment, we observe that computing the DCPD with a combinatorial greedy approach performs better than a continuous approach, since the performance of ALS-FG is relatively poor. We believe the reason for this is that the greedy approaches can escape the basin of attraction of a local minimum during the projection step, which a (standard) continuous approach will not be able to do. In fact, we observed that, in most cases, once an entry is set to zero by ALS-FG, it remains zero in the course of the iterations, which is not the case for MPALS and its variants.

B. Spectral Unmixing with the Self-Dictionary Model

For this communication on dictionary based tensor factorization models, we choose to try out the different models and algorithms on a well-known sparse coding problem, namely spectral unmixing under the pure-pixel assumption. As explained in Section II, the data itself can be used as the dictionary, and models described in this paper for matrices can be straightforwardly compared with state-of-the-art sparse coding approaches for spectral unmixing, namely the successive projection algorithm (SPA) [38], the successive nonnegative projection algorithm (SNPA) [10], the Hierarchical Clustering algorithm (H2NMF) [65] and FGNSR [41]. Below, the spectra contained in the data are used as atoms, so that the dictionary has row dimension equals to the number of spectral bands (≈ 150), and

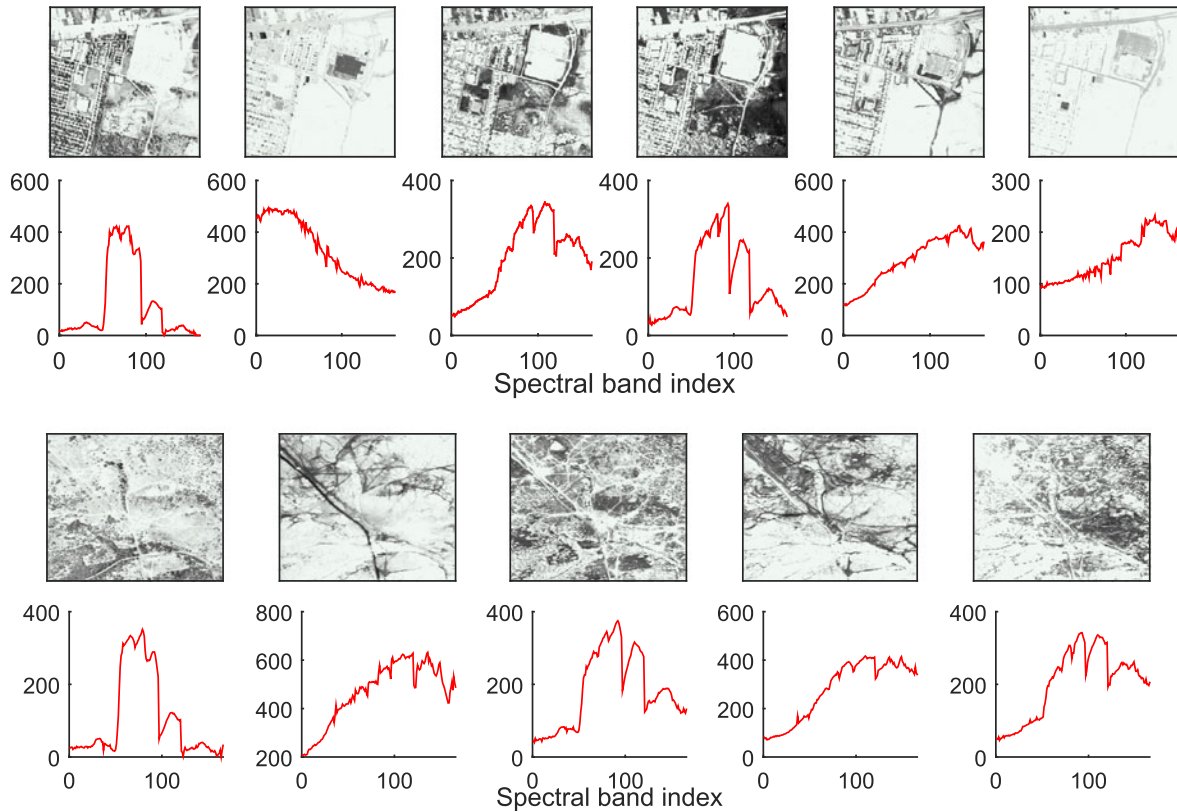


Fig. 5. Estimated spectra on the Urban HSI with d-H2NMF (top) and on the Terrain HSI with d-H2NMF (bottom).

number of atoms equals to the number of pixels in the hyper-spectral images (HSI) ($\approx 10^5$).

The two data sets that will be used to compare DCPD with these methods are Urban and Terrain. These two HSI satisfy approximately the pure pixel assumption. The dimensions of the HSI after vectorization of the pixel dimensions are respectively $307^2 \times 162$ and $(500 \times 307) \times 166$. We chose to decompose the two data sets with respective ranks $R_{\text{urban}} = 6$ and $R_{\text{terrain}} = 5$ according to what has been done previously in the literature; see [41] and the references therein. The maximal number of iterations of the DCPD algorithms is set to 50. Also because it does not perform well, results for the continuous fast gradient algorithm are not presented below.

We initialize the MPALS algorithm and its variants with each state-of-the-art algorithm with 10 additional A-HALS steps. For instance MPALS and SMPALS initialized with SPA are respectively denoted as d-SPA and ds-SPA. Results are presented in Table IV. Additionally, considering the performance of projected ALS in the multiway experiment above, the performance of projected NMF is also shown, for instance *nmf-SPA* refers to the A-HALS algorithm initialized with SPA, which endmembers output are projected onto the data points and which abundances are re-estimated using non-negative least squares.

The relative reconstruction error is used as a performance metric, since it is not possible to assess the identification performance without a ground-truth. Therefore, this experiment only studies the efficiency of the MPALS algorithm for minimizing the objective function.

In all cases, using either SMPALS or MPALS improves on the initial values for identified spectra. As shown already in the previous experiment, initialization plays an important role in the final reconstruction error, but at least using the DCPD model always refines the solutions, even when it is initially low like for FGNSR-100. Moreover, the DCPD algorithms are not excessively costly with respect to other state-of-the-art methods. There is no significant difference between MPALS and its smooth counterpart. Figure 5 shows the estimated abundances and spectra with the H2NMF-MPALS algorithm for both Urban and Terrain, and materials can be identified for each component by the user (no ground truth is available). Finally, Figure 6 shows the reconstruction error map on Urban and Terrain HSIs for the H2NMF-MPALS. Clearly the remaining error is not distributed as an i.i.d. Gaussian noise, which means the Frobenius norm used as the distance metric is not adapted. Also, most of the remaining error comes from rooftops and roads in the Urban HSI, a zone probably corrupted by large spectral variability.

Because Flex-MPALS does not require exactly that the pure pixel assumption is verified, we do not include it in this simulation. Indeed, using a reconstruction error criteria, Flex-MPALS would outperform the other methods, but that would not mean that obtained abundances and endmembers are better. On the other hand, the flexible model better tackles spectral variability since it can modify to some extent the spectra extracted from the pure pixels.⁵

⁵Interested readers will find the code for matrix Flex-MPALS online.

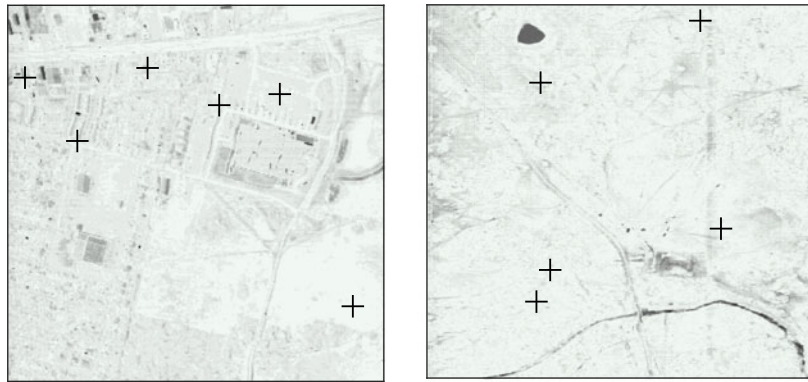


Fig. 6. Residual error maps on the Urban (left) and Terrain (right) HSIs for d-H2NMF. The black crosses mark the selected pure pixels.

VII. CONCLUSIONS

To jointly separate and identify sources using tensor canonical polyadic decomposition and a known dictionary, we introduced in this paper the DCPD model along with some flexible variants. Identifiability of the DCPD model parameters was discussed in both the matrix and the higher-order tensor cases. We proposed both greedy and continuous algorithms for DCPD and compared them on synthetic data sets and hyperspectral images. We observed that (i) the greedy algorithms provide in most cases better results, and that (ii) our DCPD formulation improves results with respect to standard CPD on the synthetic data sets, and with respect to spectral unmixing approaches based either on convex relaxation or geometric methods. A particularly promising direction for further research would be the design of efficient initialization schemes for the proposed greedy algorithms which are particularly sensitive to initialization.

An interesting continuation of this work would also be to compute the Cramér Rao bounds of the parameters in the DCPD to formally support what was presented in the simulation section. Moreover, in most application cases the dictionary is either unknown, or describes only a subset of the components. Thus a partial DCPD model should be investigated, as well as a dictionary learning scheme for tensors following the DCPD model.

ACKNOWLEDGMENT

The authors would like to thank Pierre Comon, Rodrigo Cabral Farias and Miguel A. Veganzones for early fruitful discussions on this work.

REFERENCES

- [1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. New York, NY, USA: Academic, 2010.
- [2] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. Signal. Process.*, vol. 48, no. 8, pp. 2377–2388, Aug. 2000.
- [3] R. Bro, "Multi-way analysis in the food industry: Models, algorithms, and applications," Ph.D. dissertation, Dept. Dairy and Food Sci., Univ. Copenhagen, Denmark, 1998.
- [4] M. A. Veganzones, J. E. Cohen, R. Cabral Farias, J. Chanussot, and P. Comon, "Nonnegative tensor CP decomposition of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2577–2588, May 2016.
- [5] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.
- [6] P. M. Kroonenberg, *Three-Mode Principal Component Analysis: Theory and Applications*, vol. 2. Leiden University: DSWO press, 1983.
- [7] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 2, 2016, Art. no. 16.
- [8] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
- [9] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1600–1607.
- [10] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM J. Imag. Sci.*, vol. 7, no. 2, pp. 1420–1450, 2014.
- [11] J. Nascimento and J. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.
- [12] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 231–239.
- [13] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithms for separable nonnegative matrix factorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 698–714, Apr. 2014.
- [14] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus (Series in Computational Mathematics)*. Berlin, Germany: Springer, 2012.
- [15] J. E. Cohen, "About notations in multiway array processing," arXiv: 1511.01306, 2015.
- [16] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.
- [17] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part II: Uniqueness of the overall decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 3, pp. 876–903, 2013.
- [18] J. M. Landsberg and G. Ottaviani, "Equations for secant varieties of veronese and other varieties," *Annali di Matematica Pura ed Applicata*, vol. 192, no. 4, pp. 569–606, 2013.
- [19] C. R. Rao, *Linear Statistical Inference and Its Applications (Series Probability and Statistics)*. Hoboken, NJ, USA: Wiley, 1965.
- [20] R. Bro and H. A. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *J. Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.
- [21] J. C. da Costa, M. Haardt, and F. Romer, "Robust methods based on the HOSVD for estimating the model order in PARAFAC models," in *Proc. 5th IEEE Sensor Array Multichannel Signal Process. Workshop*, 2008, pp. 510–514.
- [22] B. Valeur and M. N. Berberan-Santos, *Molecular Fluorescence: Principles and Applications*. Hoboken, NJ, USA: Wiley, 2012.
- [23] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [24] J. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

- [25] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 95–104, Jan. 2014.
- [26] R. Cabral-Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Trans. Signal. Process.*, vol. 64, no. 18, pp. 4830–4844, Sep. 2016.
- [27] Y. Guo and M. Berman, "A comparison between subset selection and l_1 regularisation with an application in spectroscopy," *Chemometrics Intell. Lab. Syst.*, vol. 118, pp. 127–138, 2012.
- [28] R. A. Harshman, "PARAFAC2: Mathematical and technical notes," *UCLA Working Papers Phonetics*, vol. 22, no. 3044, pp. 31–44(a), 1972.
- [29] H. A. Kiers, J. M. Ten Berge, and R. Bro, "PARAFAC2—Part I. A direct fitting algorithm for the PARAFAC2 model," *J. Chemometrics*, vol. 13, nos. 3/4, pp. 275–294, 1999.
- [30] D. L. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proc. Adv. Neural Inf. Process.*, vol. 16, MIT Press, 2004.
- [31] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization—Provably," in *Proc. 44th Annu. ACM Symp. Theory Comput.*, 2012, pp. 145–162.
- [32] A. M. S. Ang, Y. S. Hung, and Z. Zhang, "A non-negative tensor factorization approach to feature extraction for image analysis," in *Proc. IEEE Int. Conf. Digit. Signal Process.*, 2016, pp. 168–178.
- [33] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.
- [34] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3239–3252, Jul. 2012.
- [35] V. Bittorf, B. Recht, E. Ré, and J. Tropp, "Factoring nonnegative matrices with linear programs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1223–1231.
- [36] N. Gillis and R. Luce, "Robust near-separable nonnegative matrix factorization using linear optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 1249–1280, 2014.
- [37] S. Sahnoun, E.-H. Djermoune, D. Brie, and P. Comon, "A simultaneous sparse approximation method for multidimensional harmonic retrieval," *Signal Process.*, vol. 131, pp. 36–48, 2017.
- [38] U. Araújo, B. Saldanha, R. Galvão, T. Yoneyama, H. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometrics Intell. Lab.*, vol. 57, no. 2, pp. 65–73, 2001.
- [39] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1232–1249, Oct. 2003.
- [40] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4177–4193, Nov. 2011.
- [41] N. Gillis and R. Luce, "A fast gradient method for nonnegative sparse regression with self dictionary," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 24–37, Jan. 2018, doi: [10.1109/TIP.2017.2753400](https://doi.org/10.1109/TIP.2017.2753400).
- [42] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [43] M. E. Timmerman and H. A. Kiers, "Three-way component analysis with smoothness constraints," *Comput. Statist. Data Anal.*, vol. 40, no. 3, pp. 447–470, 2002.
- [44] G. Favier and A. de Almeida, "Overview of constrained PARAFAC models," *EURASIP J. Adv. Signal Process.*, vol. 142, no. 1, 2014.
- [45] J. H. M. Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, "Tensor CP decomposition with structured factor matrices: Algorithms and performance," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 4, pp. 757–769, Jun. 2016.
- [46] J. E. Cohen and N. Gillis, "A new approach to dictionary-based nonnegative matrix factorization," in *Proc. 25th Eur. Signal Process. Conf.*, 2017, pp. 493–497.
- [47] I. Domanov and L. De Lathauwer, "Generic uniqueness of a structured matrix factorization and applications in blind source separation," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 701–711, Jun. 2016.
- [48] D. A. Roberts, M. Gardner, R. Church, S. Ustin, G. Scheer, and R. Green, "Mapping chaparral in the Santa Monica mountains using multiple end-member spectral mixture models," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 267–279, 1998.
- [49] J.-P. Combe *et al.*, "Analysis of OMEGA/mars express data hyperspectral data using a multiple Endmember Linear Spectral Unmixing Model (MELSUM): Methodology and first results," *Planet. Space Sci.*, vol. 56, no. 7, pp. 951–975, 2008.
- [50] C. Song, "Spectral mixture analysis for subpixel vegetation fractions in the urban environment: How to incorporate endmember variability?" *Remote Sens. Environ.*, vol. 95, no. 2, pp. 248–263, 2005.
- [51] G. P. Asner, M. M. Bustamante, and A. R. Townsend, "Scale dependence of biophysical structure in deforested areas bordering the Tapajos National Forest, Central Amazon," *Remote Sens. Environ.*, vol. 87, no. 4, pp. 507–520, 2003.
- [52] I. Domanov, "Study of canonical polyadic decomposition of higher-order tensors," Ph.D. dissertation, Dept. Elect. Eng. (ESAT), KU Leuven, Leuven, Belgium, 2013.
- [53] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, 2008.
- [54] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
- [55] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.
- [56] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *J. Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.
- [57] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, 2012.
- [58] A. Cichocki and S.-i. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. vol. 1, New York, NY, USA: Wiley, 2002.
- [59] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1/2, pp. 83–97, 1955.
- [60] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss Seidel method under convex constraints," *Oper. Res. Lett.*, vol. 26, pp. 127–136, 2000.
- [61] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Doklady*, vol. 27, no. 2, 1983, pp. 372–376.
- [62] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Math. Program.*, vol. 156, nos. 1/2, pp. 59–99, 2016.
- [63] J.-P. Royer, N. Thirion-Moreau, and P. Comon, "Computing the polyadic decomposition of nonnegative third order tensors," *Signal Process.*, vol. 91, no. 9, pp. 2159–2171, 2011.
- [64] C. A. Andersson and R. Bro, "The N-way toolbox for MATLAB," *Chemometrics Intell. Lab. Syst.*, vol. 52, no. 1, pp. 1–4, 2000.
- [65] N. Gillis, D. Kuang, and H. Park, "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2066–2078, Apr. 2015.



Jérémie Emile Cohen received the Master's degree in applied mathematics from Ecole Centrale de Lyon, Écully, France, in 2013, and the Ph.D. degree in signal and image processing from Université Grenoble Alpes, Grenoble, France, in 2016.

He is currently a Postdoctoral Researcher with the University of Mons, Mons, Belgium, under an FNRS grant. His research interests include data mining using low-rank representation and tensor decompositions.



Nicolas Gillis received the Master's and Ph.D. degrees in applied mathematics from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 2007 and 2011, respectively. He is currently an Associate Professor with the Department of Mathematics and Operational Research, Faculté polytechnique, Université de Mons, Mons, Belgium. His research interests include optimization, numerical linear algebra, machine learning, and data mining.