



**HAL**  
open science

## Some issues in computing the CP decomposition of NonNegative Tensors

Mohamad Jouni, Mauro Dalla Mura, Pierre Comon

► **To cite this version:**

Mohamad Jouni, Mauro Dalla Mura, Pierre Comon. Some issues in computing the CP decomposition of NonNegative Tensors. LVA/ICA 2018 - 14th International Conference on Latent Variable Analysis and Signal Separation, Jul 2018, Guildford, United Kingdom. pp.57-66, 10.1007/978-3-319-93764-9\_6 . hal-01784370v2

**HAL Id: hal-01784370**

**<https://hal.science/hal-01784370v2>**

Submitted on 17 Dec 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Some issues in computing the CP decomposition of NonNegative Tensors

Mohamad Jouni, Mauro Dalla Mura, and Pierre Comon\*

Univ. Grenoble Alpes, CNRS, Grenoble INP, Gipsa-Lab, 38000 Grenoble, France  
firstname.lastname@gipsa-lab.fr <http://www.gipsa-lab.grenoble-inp.fr>

**Abstract.** Tensor decompositions are still in the process of study and development. In this paper, we point out a problem existing in nonnegative tensor decompositions, stemming from the representation of decomposable tensors by outer products of vectors, and propose approaches to solve it. In fact, a scaling indeterminacy appears whereas it is not inherent in the decomposition, and the choice of scaling factors has an impact during the execution of iterative algorithms and should not be overlooked. Computer experiments support the interest in the greedy algorithm proposed, in the case of the CP decomposition.

## 1 Introduction

Tensors of order  $d$  are represented by data arrays with  $d$  indices, ( $d = 2$  for matrices). They provide unique features as they are a suitable data structure for representing multimodal or multisource data, in which each diversity is represented by one of the ways of the tensor. One of the most interesting applications of tensors is the Canonical Polyadic (CP) decomposition defined below, which aims at representing a tensor as a sum of *decomposable* rank one tensors, revealing relationships among its  $d$  ways.

*CP decomposition.* In this paper, we shall focus our attention on the CP decomposition of third order tensors. To begin with, a tensor  $\mathcal{D}$  is *decomposable* if it can be expressed as the outer product of vectors, *i.e.*:  $\mathcal{D}_{ijk} = a_i b_j c_k$ , which will be denoted compactly as  $\mathcal{D} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$ , where  $\otimes$  is the outer (tensor) product. Next, every real tensor  $\mathcal{T}$  of order  $d = 3$  and dimensions  $I \times J \times K$ , admits a CP Decomposition of the following form:

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

where  $\mathcal{D}(r) \stackrel{\text{def}}{=} \mathbf{a}(r) \otimes \mathbf{b}(r) \otimes \mathbf{c}(r)$ ,  $\mathbf{a}(r)$ ,  $\mathbf{b}(r)$  and  $\mathbf{c}(r)$  being real vectors, which can be stored in the so-called factor matrices,  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  respectively, of size  $I \times R$ ,  $J \times R$ , and  $K \times R$  respectively, and  $\lambda_r$  are real positive scalars. The CP

---

\* This work was supported in part by ERC Advanced Grant 2013-320594 “DECODA”.

decomposition reveals *tensor rank* when  $R$  is minimal, which will be assumed from now on; for instance, tensors  $\mathcal{D}(r)$  are of rank 1. Note that another writing of (1) in terms of factor matrices is  $T_{ijk} = \sum_{r=1}^R \lambda_r A_{ir} B_{jr} C_{kr}$ . In addition, because of the presence of  $\lambda_r$ , the columns of factor matrices may be normalized to 1.

At this stage, it is important to stress that *there is no scaling ambiguity* in the CP decomposition (1), contrary to what is sometimes claimed in the literature. Only the *representation* of tensors  $\mathcal{D}(r)$  by triplets of vectors is subject to this indeterminacy. In fact, by definition, tensors are precisely *equivalence classes* with respect to scaling [1–4]: the triplets  $(\mathbf{a}, \mathbf{b}, \mathbf{c})$  and  $(\alpha\mathbf{a}, \beta\mathbf{b}, \gamma\mathbf{c})$  represent *the same* tensor provided that  $\alpha\beta\gamma = 1$ .

The rank  $R$  of the CP decomposition (1) is of particular interest in applications since it is related to the intrinsic dimensionality of multilinear data. Furthermore, the CP decomposition, contrary to other tensor decompositions, *e.g.*, Tucker’s or High-Order Singular Value Decomposition (HOSVD), enjoys uniqueness if the rank is not too large [5–7]. Uniqueness is of utmost importance since it eventually allows physical interpretation of relationships among the ways of a tensor.

## 2 Motivation

*Nonnegativity.* When the observation tensor  $\mathcal{T}$  contains only real nonnegative entries, it is suitable to impose decomposable tensors  $\mathcal{D}(r)$  to also be nonnegative. By doing this, we define a *nonnegative rank*,  $R^+$ , which may be larger than  $R$ . This is actually already true for matrices (tensors of order 2). In fact, Herbert E. Robbins exhibited a simple example of a  $5 \times 5$  matrix having rank 3 but nonnegative rank 4; see [8, 4] for its expression. It is thus necessary to define the nonnegative CP decomposition of a nonnegative tensor as:

$$\mathcal{T} = \sum_{r=1}^{R^+} \lambda_r \mathbf{a}(r) \otimes \mathbf{b}(r) \otimes \mathbf{c}(r), \quad (2)$$

where  $a_i(r) \in \mathbb{R}^+$ ,  $b_j(r) \in \mathbb{R}^+$  and  $c_k(r) \in \mathbb{R}^+$ ,  $\forall(i, j, k, r)$ .

There are many applications where nonnegativity is relevant, as to provide better interpretable results when dealing with variables related to physical quantities such as luminance in images, spectra or chemical concentrations [9, 10]. There exist many algorithms aiming at computing the CP decomposition of nonnegative tensors [9, 11]. However, due to measurement noise or modeling errors, the tensor to decompose may not be nonnegative or may have a too large rank, hence requiring to be approximated. It turns out that, given any real tensor  $\mathcal{T}$  of rank  $R$ , it is fortunately always possible to find a best nonnegative approximation of  $\mathcal{T}$  of given nonnegative rank  $R^+$ . This problem is indeed well-posed [12, 13] (which would not be the case in  $\mathbb{R}$  instead of  $\mathbb{R}^+$ ).

*Projection onto the nonnegative orthant:* In the nonnegative CP decomposition (2), all quantities are nonnegative. For instance, vector  $\mathbf{a}(r)$  belongs to the nonnegative orthant  $(\mathbb{R}^+)^I$ . In iterative algorithms, this constraint is ensured at each iteration by projecting a computed value onto the nonnegative orthant. This is where the problem shows up. In fact, projecting  $\mathcal{D}(r)$  or its building vectors  $\{\mathbf{a}(r), \mathbf{b}(r), \mathbf{c}(r)\}$  do not yield the same result. Since this observation is already true for matrices, a simple example will be most convincing.

*Example.* Take the matrix  $\mathbf{M}$  below, of rank 1. Now its projection  $\mathbf{M}^+$  has rank 2. So it is preferred to project its supporting vectors  $\{\mathbf{a}, \mathbf{b}\}$  instead. The obtained vectors are  $\{\mathbf{a}^+, \mathbf{b}^+\}$  and yield a matrix of nonnegative rank equal to 1:

$$\mathbf{M} = \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ -1 \end{pmatrix} = \mathbf{a} \otimes \mathbf{b}, \quad \mathbf{M}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{a}^+ \otimes \mathbf{b}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}.$$

The problem is that vectors  $\{\mathbf{a}, \mathbf{b}\}$  are not uniquely defined. We could have taken  $\{-\mathbf{a}, -\mathbf{b}\}$  without changing  $\mathbf{M}$ . Should we do that, we obtain instead:

$$\mathbf{M} = \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \mathbf{a} \otimes \mathbf{b}, \quad \mathbf{M}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{a}^+ \otimes \mathbf{b}^+ = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

We see that the projected matrix of nonnegative rank 1 is not the same. This issue comes from the fact that no care has been taken of the scaling indeterminacies (which reduce to sign indeterminacies thanks to the use of factors  $\lambda_r$ ) inherent in the representation of a rank-1 tensor by a triplet of vectors.

Algorithms resorting to projection include Alternating Nonnegative Least Squares (ANLS) [9], Projected and Compressed ANLS (ProCo) [14], or Alternating Direction Method of Multipliers (ADMM) [11], among others. Hard thresholding is the procedure in which it is the easiest to illustrate the occurring of the problem.

---

**Algorithm 1** Alternating Nonnegative Least Squares (ANLS)

---

**Require:**  $\mathcal{T}$ ,  $\mathbf{B} = \mathbf{B}[0]$ ,  $\mathbf{C} = \mathbf{C}[0]$

$t = 0$ ;

**while** stopping criterion is not met, **do**

$t = t + 1$

compute  $\mathbf{A}$  from  $\mathbf{B}[t-1]$  and  $\mathbf{C}[t-1]$ ;  $\mathbf{A}[t] \leftarrow \mathbf{A}^+$

compute  $\mathbf{B}$  from  $\mathbf{C}[t-1]$  and  $\mathbf{A}[t]$ ;  $\mathbf{B}[t] \leftarrow \mathbf{B}^+$

compute  $\mathbf{C}$  from  $\mathbf{A}[t]$  and  $\mathbf{B}[t]$ ;  $\mathbf{C}[t] \leftarrow \mathbf{C}^+$

normalize columns:  $\mathbf{A}[t] \leftarrow \mathbf{A}[t]\mathbf{\Lambda}_A^{-1}$ ;  $\mathbf{B}[t] \leftarrow \mathbf{B}[t]\mathbf{\Lambda}_B^{-1}$ ;  $\mathbf{C}[t] \leftarrow \mathbf{C}[t]\mathbf{\Lambda}_A\mathbf{\Lambda}_B$ ;

**end while**

normalize columns:  $\mathbf{C}[t] \leftarrow \mathbf{C}[t]\mathbf{\Lambda}^{-1}$

**return**  $\mathbf{A}[t]$ ,  $\mathbf{B}[t]$ ,  $\mathbf{C}[t]$ ,  $\mathbf{\Lambda}$

---

*ANLS*. One algorithm that has been widely used to compute CP decomposition (1) is the Alternating Least Squares (ALS) algorithm. ALS minimizes with respect to matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  in an alternating fashion, the loss:

$$\Phi = \sum_{ijk} [\mathcal{T}_{ijk} - \sum_{r=1}^R \lambda_r A_{ir} B_{jr} C_{kr}]^2. \quad (3)$$

Factor matrices are updated in turns during each iteration until a certain condition is attained (*e.g.* the number of iterations or a certain threshold on the reconstruction error). When a nonnegative decomposition is sought, each factor matrix can be projected onto the nonnegative orthant right after its calculation; this is the ANLS algorithm [9, p.47]. The pseudo-code is given in Alg.1.

### 3 Proposed approach

We illustrate the problem with hard thresholding (cf. Section 4), but our solution could also reveal useful in soft thresholding as well. The problem is worse when all entries in a column vector are set to zero; this prevents its normalization (as it would lead to a division by zero) or imposes an erroneous reduction of the rank (due to the arbitrary removal of the null columns). The solution we describe overcomes these two difficulties most of the time, up to negligible extraneous computation load. We propose to implement this in a procedure to be executed before projection. The concept goes as follows. Because of normalization, the scaling indeterminacy reduces merely to signs. In fact, in every decomposable tensor  $\mathcal{D}(r)$ , we have two variables,  $\epsilon, \eta \in \{-1, +1\}$ , which are to be used as sign flippers for the columns  $\mathbf{a}(r)$ ,  $\mathbf{b}(r)$  and  $\mathbf{c}(r)$  that are together involved in an outer product term, without changing the result of the outer product given by:

$$\mathbf{a}(r) \otimes \mathbf{b}(r) \otimes \mathbf{c}(r) = (\epsilon\eta\mathbf{a}(r)) \otimes (\epsilon\mathbf{b}(r)) \otimes (\eta\mathbf{c}(r)), \quad \forall (\epsilon, \eta) \in \{-1, +1\}. \quad (4)$$

This formula covers all 4 combinations of sign flipping of vectors, without affecting the result of the original outer product. Now denote by  $\mathbf{a}'(r) = \epsilon\eta\mathbf{a}(r)$ ,  $\mathbf{b}'(r) = \epsilon\mathbf{b}(r)$ , and  $\mathbf{c}'(r) = \eta\mathbf{c}(r)$ , and:

$$\mathbf{a}^-(r) = \mathbf{a}'(r) \text{ where } \mathbf{a}'(r) < 0, \text{ and } 0 \text{ elsewhere} \quad (5)$$

$$\mathbf{b}^-(r) = \mathbf{b}'(r) \text{ where } \mathbf{b}'(r) < 0, \text{ and } 0 \text{ elsewhere} \quad (6)$$

$$\mathbf{c}^-(r) = \mathbf{c}'(r) \text{ where } \mathbf{c}'(r) < 0, \text{ and } 0 \text{ elsewhere.} \quad (7)$$

Vectors  $\mathbf{a}^+(r)$ ,  $\mathbf{b}^+(r)$  and  $\mathbf{c}^+(r)$  are defined in a similar manner, with positive entries. In particular,  $\mathbf{a}^+(r) + \mathbf{a}^-(r) = \mathbf{a}'(r)$ .

Given a triplet of vectors,  $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ , there are 4 possibilities to construct a non-negative decomposable tensor  $\mathcal{D}^{[4]}$  by just flipping their signs without changing the value of  $(\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c})$  and by setting to zero negative values in each vector (refer to Table.1).

| $(\epsilon, \eta)$     | $(+, +)$  | $(-, -)$  | $(+, -)$  | $(-, +)$  |
|------------------------|---|---|---|---|
| $\ell$                 | 1   | 2   | 3   | 4   |
| $\mathcal{D}^{[\ell]}$ | $\mathcal{D}^{[1]} =$<br>$\mathbf{a}^+ \otimes \mathbf{b}^+ \otimes \mathbf{c}^+$ | $\mathcal{D}^{[2]} =$<br>$\mathbf{a}^+ \otimes \mathbf{b}^- \otimes \mathbf{c}^-$ | $\mathcal{D}^{[3]} =$<br>$\mathbf{a}^- \otimes \mathbf{b}^+ \otimes \mathbf{c}^-$ | $\mathcal{D}^{[4]} =$<br>$\mathbf{a}^- \otimes \mathbf{b}^- \otimes \mathbf{c}^+$ |

**Table 1.** The 4 possibilities created by sign flipping.

where for the sake of convenience,  $\mathbf{a}^+$  stands for vector  $\mathbf{a}^{[\ell]}(r)^+$ , and similarly for  $\mathbf{a}^-$ ,  $\mathbf{b}^+$ ,  $\mathbf{b}^-$ ,  $\mathbf{c}^+$  and  $\mathbf{c}^-$ .

We are interested to know which combination would yield the minimal number of resets. Ultimately, we are concerned about (i) avoiding to set a whole vector to zero, which would lead to decrease the rank. This goal can mean “set as few entries to zero as possible”. And we also aim at (ii) minimizing the distance between the original tensor and its nonnegative approximation.

We explored several criteria. The first is to minimize  $\Phi_0 = \|\mathcal{T} - \sum_r \mathcal{D}^{[\ell]}(r)\|_2$ . This criterion is very costly to optimize, due to the large number of combinations. In fact, for every  $r$ , there are 4 possibilities to assign  $(\epsilon, \eta)$ , and this assignment can be different for each  $r$ . This would result in  $4^R$  possibilities to explore. This is why we propose two greedy algorithms searching for the optimal solution  $\mathcal{D}^{[\ell]}(r)$  independently for every  $r$ . One possibility is to minimize w.r.t.  $\ell$  the following product **for every  $r$  independently**, and for the  $L^2$  norm:

$$\Phi_1(\ell, r) = \|\mathcal{D}(r) - \mathcal{D}^{[\ell]}(r)\|_2^2. \quad (8)$$

Let us express this criterion for  $\ell = 1$ , without loss of generality. We have for any fixed  $r$  (that we drop for the sake of convenience):

$$\Phi_1(1, r) = \|\mathcal{D}(r)\|_2^2 + \|\mathcal{D}^{[1]}(r)\|_2^2 - 2 \sum_{ijk} a_i a_i^+ b_j b_j^+ c_k c_k^+. \quad (9)$$

The last term can be rewritten as  $2(\mathbf{a}^\top \mathbf{a}^+)(\mathbf{b}^\top \mathbf{b}^+)(\mathbf{c}^\top \mathbf{c}^+)$ . Next, it is also equal to  $2\|\mathbf{a}^+\|^2 \|\mathbf{b}^+\|^2 \|\mathbf{c}^+\|^2$ , since  $\mathbf{a}^+$  and  $\mathbf{a}^-$  are orthogonal and  $\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-$ . This suggests another criterion to minimize w.r.t.  $\ell$ :

$$\Phi_2(\ell, r) = \|\mathbf{a}^-\| \cdot \|\mathbf{b}^-\| \cdot \|\mathbf{c}^-\| \quad (10)$$

Criteria  $\Phi_1$  and  $\Phi_2$  are easy to optimize w.r.t.  $(\epsilon, \eta)$ , *i.e.* w.r.t.  $\ell$ , and need negligible extraneous computation load.

## 4 A Toy Example

Consider the factor matrices:

$$\mathbf{A} = \begin{bmatrix} 0.8025 & 0.1914 \\ 0.0089 & 0.9106 \\ 0.5966 & 0.3662 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.0088 & 0.7495 \\ 1 & 0.6620 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0.7071 & 0 \\ 0.7071 & 0 \end{bmatrix}.$$

**Algorithm 2** Minimization of  $\Phi_1$  w.r.t. to  $\epsilon$  and  $\eta$  for fixed  $r$ 


---

**Require:**  $\mathbf{a}(r)$ ,  $\mathbf{b}(r)$ ,  $\mathbf{c}(r)$   
initialize  $\epsilon(\ell)$ ,  $\eta(\ell)$  for each possible  $\mathcal{D}^{[\ell]}$ ; (according to Table.1)  
normalize:  $\mathbf{a} \leftarrow \frac{\mathbf{a}(r)}{\|\mathbf{a}(r)\|}$ ;  $\mathbf{b} \leftarrow \frac{\mathbf{b}(r)}{\|\mathbf{b}(r)\|}$ ;  $\mathbf{c} \leftarrow \frac{\mathbf{c}(r)}{\|\mathbf{c}(r)\|}$ ;  
 $\ell = 0$ ;  
**while**  $\ell \leq 4$ , **do**  
 $\ell = \ell + 1$ ;  
 $\mathbf{a}' \leftarrow \epsilon(\ell) \eta(\ell) \mathbf{a}$ ;  $\mathbf{b}' \leftarrow \epsilon(\ell) \mathbf{b}$ ;  $\mathbf{c}' \leftarrow \eta(\ell) \mathbf{c}$ ;  
compute  $\mathbf{a}^+$ ,  $\mathbf{b}^+$ , and  $\mathbf{c}^+$ ;  
 $\Phi_1(\ell) \leftarrow \|\mathcal{D} - \mathcal{D}^{[\ell]}\|_2^2$ ;  
**end while**  
Find  $\ell_o = \arg \min_{\ell} \Phi_1(\ell)$ ;  
 $\mathbf{a}(r) \leftarrow \epsilon(\ell_o) \eta(\ell_o) \mathbf{a}(r)$ ;  $\mathbf{b}(r) \leftarrow \epsilon(\ell_o) \mathbf{b}(r)$ ;  $\mathbf{c}(r) \leftarrow \eta(\ell_o) \mathbf{c}(r)$ ;  
**return**  $\mathbf{a}(r)$ ,  $\mathbf{b}(r)$ ,  $\mathbf{c}(r)$ ;

---

**Algorithm 3** Modified ANLS

---

**Require:**  $\mathcal{T}$ ,  $\mathbf{B} = \mathbf{B}[0]$ ,  $\mathbf{C} = \mathbf{C}[0]$   
 $t = 0$ ;  
**while** stopping criterion is not met, **do**  
 $t = t + 1$   
compute  $\mathbf{A}$  from  $\mathbf{B}[t-1]$  and  $\mathbf{C}[t-1]$ ;  $\mathbf{A}[t]$   
compute  $\mathbf{B}$  from  $\mathbf{C}[t-1]$  and  $\mathbf{A}[t]$ ;  $\mathbf{B}[t]$   
compute  $\mathbf{C}$  from  $\mathbf{A}[t]$  and  $\mathbf{B}[t]$ ;  $\mathbf{C}[t]$   
 $r = 0$ ;  
**while**  $r < R$  **do**  
update  $\mathbf{a}(r)$ ,  $\mathbf{b}(r)$ , and  $\mathbf{c}(r)$  using Alg.2;  
 $r = r + 1$ ;  
**end while**  
 $\mathbf{A}[t] \leftarrow \mathbf{A}^+$ ;  $\mathbf{B}[t] \leftarrow \mathbf{B}^+$ ;  $\mathbf{C}[t] \leftarrow \mathbf{C}^+$ ;  
replace null columns in  $\mathbf{A}$ ,  $\mathbf{B}$  or  $\mathbf{C}$  by random values using e.g. absolute value of standard Gaussian distribution; (see Sect.5)  
normalize columns:  $\mathbf{A}[t] \leftarrow \mathbf{A}[t] \mathbf{A}_A^{-1}$ ;  $\mathbf{B}[t] \leftarrow \mathbf{B}[t] \mathbf{A}_B^{-1}$ ;  $\mathbf{C}[t] \leftarrow \mathbf{C}[t] \mathbf{A}_A \mathbf{A}_B$ ;  
**end while**  
normalize columns:  $\mathbf{C}[t] \leftarrow \mathbf{C}[t] \mathbf{A}^{-1}$   
**return**  $\mathbf{A}[t]$ ,  $\mathbf{B}[t]$ ,  $\mathbf{C}[t]$ ,  $\mathbf{A}$

---

When computing the CP Decomposition with Alg.1, after one update of  $\mathbf{A}$ , one of its columns became negative, and hence one of its columns got discarded, and the rank was decreased by 1. Note that, for the sake of conciseness, during the loop of updates only the columns of  $\mathbf{A}$  and  $\mathbf{B}$  are normalized and their norms multiply  $\mathbf{C}$ ; after the loop ends,  $\mathbf{C}$  is normalized and its column norms (containing that of  $\mathbf{A}$  and  $\mathbf{B}$ ) form the values of  $\mathbf{A}$  (cf. Alg.1).

**Algorithm 1: Standard ALS**

$$\mathbf{A} : \begin{bmatrix} 0.2311 & -0.0464 \\ 0.1891 & -0.0627 \\ 0.2178 & -0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2311 & 0 \\ 0.1891 & 0 \\ 0.2178 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 & \text{Undefined} \\ 0.5118 & \text{Undefined} \\ 0.5893 & \text{Undefined} \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 \\ 0.5118 \\ 0.5893 \end{bmatrix}$$

$$\mathbf{B} : \begin{bmatrix} 0.2962 & 0 \\ 1.0561 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2962 & 0 \\ 1.0561 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2701 & \text{Undefined} \\ 0.9628 & \text{Undefined} \end{bmatrix} \rightarrow \begin{bmatrix} 0.2701 \\ 0.9628 \end{bmatrix}$$

$$\mathbf{C} : \begin{bmatrix} 0 & 0 \\ 0.4978 & 0 \\ 0 & 0 \\ 1.3779 & 0 \\ 1.3779 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0.4978 & 0 \\ 0 & 0 \\ 1.3779 & 0 \\ 1.3779 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0.2018 & 0 \\ 0 & 0 \\ 0.5585 & 0 \\ 0.5585 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0.2018 \\ 0 \\ 0.5585 \\ 0.5585 \end{bmatrix}$$

At the end the output of standard ANLS results in:

$$\mathbf{A} = \begin{bmatrix} 0.8004 \\ 0.0249 \\ 0.5990 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0.0165 \\ 0.9999 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 0 \\ 0.0835 \\ 0 \\ 0.7046 \\ 0.7046 \end{bmatrix}$$

**Algorithm 2: Modified ANLS (using  $\Phi_1$ )**

$$\mathbf{A} : \begin{bmatrix} 0.2311 & -0.0464 \\ 0.1891 & -0.0627 \\ 0.2178 & -0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2311 & 0.0464 \\ 0.1891 & 0.0627 \\ 0.2178 & 0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 & 0.5015 \\ 0.5118 & 0.6772 \\ 0.5893 & 0.5384 \end{bmatrix}$$

$$\mathbf{B} : \begin{bmatrix} 0.3420 & 0.4425 \\ 1.1203 & 0.6211 \end{bmatrix} \rightarrow \begin{bmatrix} 0.3420 & 0.4425 \\ 1.1203 & 0.6211 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2919 & 0.5802 \\ 0.9564 & 0.8145 \end{bmatrix}$$

$$\mathbf{C} : \begin{bmatrix} 0 & 0 \\ -0.4978 & -7.6392 \\ 0 & 0 \\ 2.9691 & 11.1635 \\ 2.9691 & 11.1635 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 7.6392 \\ 0 & 0 \\ 2.9691 & 0 \\ 2.9691 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0.5392 \\ 0 & 0 \\ 1.2853 & 0 \\ 1.2853 & 0 \end{bmatrix}$$

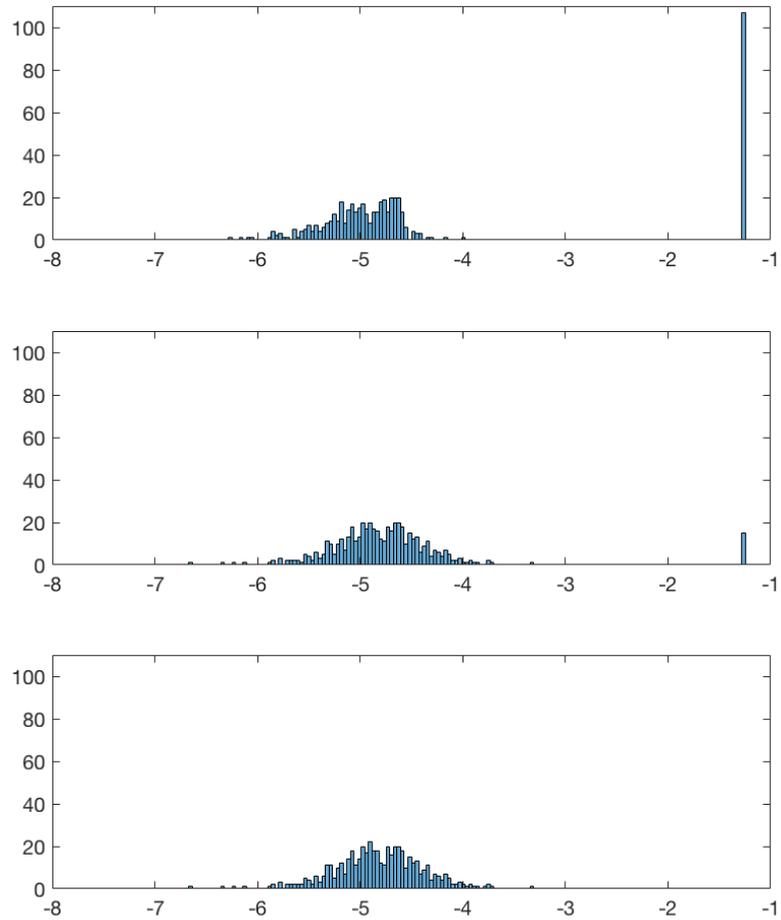
At the end the output of Modified ANLS results in:

$$\mathbf{A} = \begin{bmatrix} 0.8025 & 0.1914 \\ 0.0089 & 0.9106 \\ 0.5966 & 0.3662 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0.0088 & 0.7495 \\ 1 & 0.6620 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0.7071 & 0 \\ 0.7071 & 0 \end{bmatrix}$$

## 5 Computer results

500 realizations of  $10 \times 5$  matrices  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$  are drawn. The rank of the tensor that is tested is hence  $R = 5$ . Entries of factor matrices are the absolute value of i.i.d. drawn from a standard Gaussian distribution. On each realization, both ANLS and a modified version based on the minimization of  $\Phi_1$  are run.

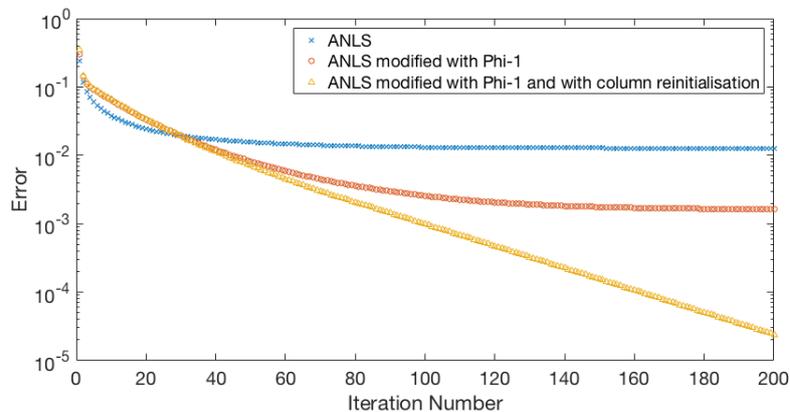
As can be seen in Fig. 1, 107 realizations out of 500 are unsuccessful with ANLS, that is, 107 realizations generate one fully negative column in a factor matrix which is then zeroed due to hard thresholding. This eventually leads to a decrease of the rank down to 4 and hence to a large reconstruction error (close



**Fig. 1.** Histograms of the error, in  $\log_{10}$  scale, obtained after 500 iterations. Top: ANLS. Middle: ANLS modified with  $\Phi_1$ . Bottom: ANLS modified with  $\Phi_1$  and with column reinitialization.

to  $10^{-1}$ ). Among those 107 pathological cases, our simple function described by Alg.2 could cope with 92 of them without a significant increase in complexity. However, 15 realizations remain unsolved, because they correspond to either one of two particular cases: (i) either one column, say  $\mathbf{a}(r_o)$ , is fully negative, and

the two others, namely  $\mathbf{b}(r_o)$  and  $\mathbf{c}(r_o)$  are fully positive, or (ii) all the three columns are fully negative.



**Fig. 2.** The average error obtained after 500 iterations, as a function of the number of iterations in ANLS (blue crosses), ANLS modified with  $\Phi_1$  (red circles), and ANLS modified with  $\Phi_1$  and with column reinitialization (yellow triangles).

In order to cope with the latter cases, a straightforward improvement was brought in Alg. 3, by drawing a fresh column vector (also the absolute value of i.i.d drawn from a standard Gaussian distribution) to replace null vectors when generated in the unsolved pathological cases, before normalizing the columns. The results can be seen in Fig.1 (bottom) and Fig.2, where all 15 cases were solved and the rank was preserved.

## 6 Concluding remarks

In this paper, we have emphasized the fact that rank-1 tensors should not be treated as a collection of vectors without care, and showed an illustration in the case of ANLS using hard thresholding. In the latter case, two modifications have been proposed to fix the problem. In future works, we plan to investigate applications to other algorithms such as ADMM, and/or using soft thresholding. The influence of noise would also deserve to be further addressed.

## References

1. J. R. Ruiz-Tolosa and E. Castillo. *From Vectors to Tensors*. Universitext. Springer, Berlin, Heidelberg, 2005.

2. J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. AMS publ., 2012.
3. W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Series in Computational Mathematics. Springer, Berlin, Heidelberg, 2012.
4. P. Comon. Tensors: a brief introduction. *IEEE Sig. Proc. Magazine*, 31(3):44–53, May 2014. special issue on BSS. hal-00923279.
5. N. D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of N-way arrays. *Jour. Chemo.*, 14:229–239, 2000.
6. J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions. *Linear Algebra and Applications*, 18:95–138, 1977.
7. L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven. An algorithm for generic and low-rank specific identifiability of complex tensors. *SIAM J. matrix Ana. Appl.*, 35(4):1265–1287, 2014.
8. J. Cohen and U. Rothblum. Nonnegative ranks, decompositions and factorizations of nonnegative matrices. *Lin. Alg. Appl.*, 190:149–168, 1993.
9. A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations*. Wiley, Chichester, 2009.
10. P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press, Oxford UK, Burlington USA, 2010. hal-00460653.
11. K.Huang, N. Sidiropoulos, and A. P. Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Trans. Signal proc.*, 64(19):5052–5065, 2016.
12. L-H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *Jour. Chemometrics*, 23:432–441, August 2009. hal-00410056.
13. Y. Qi, P. Comon, and L. H. Lim. Uniqueness of nonnegative tensor approximations. *IEEE Trans. Inf. Theory*, 62(4):2170–2183, April 2016. arXiv:1410.8129.
14. J. E. Cohen, R. Cabral Farias, and P. Comon. Fast decomposition of large nonnegative tensors. *IEEE Sig. Proc. Letters*, 22(7):862–866, July 2015. hal-01069069.