An Iterative Deflation Algorithm for Exact CP Tensor Decomposition

Alex Pereira da Silva, Pierre Comon, André L. F. de Almeida

AN ITERATIVE DEFLATION ALGORITHM FOR EXACT CP TENSOR DECOMPOSITION

Alex P. da Silva[1], *Pierre Comon*[1], *and André L. F. de Almeida*[2]

[1]GIPSA-Lab, 11 rue des Mathematiques F-38402 Saint Martin d'Hères cedex, France
[2]Federal University of Ceará, Campus do Pici, CP 6005 60455-970, Fortaleza, CE, Brazil
Email: Alex.Pereira-Da-Silva@gipsa-lab.grenoble-inp.fr

ABSTRACT

The Canonical Polyadic (CP) tensor decomposition has become an attractive mathematical tool these last ten years in various fields. Yet, efficient algorithms are still lacking to compute the full CP decomposition, whereas rank-one approximations are rather easy to compute. We propose a new deflation-based iterative algorithm allowing to compute the full CP decomposition, by resorting only to rank-one approximations. An analysis of convergence issues is included, as well as computer experiments. Our theoretical and experimental results show that the algorithm converges almost surely.

Index Terms— Deflation; rank-1 approximations; Canonical Polyadic; CanDecomp ; Parafac ; tensor decomposition; convergence

1. INTRODUCTION

Tensors play an important role in many applications such as chemometrics [1], blind source separation [2], data mining [3] and telecommunications [4]. The interest in resorting to tensors, compared to more standard matrix-based approaches, lies in the uniqueness of their decomposition into rank-one terms, now referred to as CP decomposition [5]. There exist iterative algorithms allowing to compute the CP decomposition, but none of them is entirely satisfactory. The most widely used is the Alternating Least Squares (ALS) [1], which is a simple iterative method that updates alternately the factor matrices of the CP decomposition. Note that the local convergence of ALS can be long, and that global convergence is not guaranteed [6, 7, 8]. There exists other iterative methods, namely those based on all-at-once estimation of the factor matices, such as the Conjugate Gradient and Levenberg-Marquardt methods [9, 8]. Hierarchical methods as described in [10] and [11] can be also used. In the former, the authors use a hierarchical ALS method only for decomposing non-negative tensors. Moreover, the performance of the algorithm strongly depends on initializations and the updates are made column-wise, which can be expensive. In the latter, the authors propose an interesting finite deflation algorithm, but this procedure only works when the rank of the tensor does not exceed the tensor dimensions. For both algorithms, no convergence study has been performed.

Since rank-one tensor approximations are easier to compute, one can be tempted by a deflation procedure, which consists of computing successive rank-one approximations followed by subtractions. The conventional deflation works well for matrices, but does not generally provide satisfactory results for tensors, as pointed out in [12]. In [10], the hierarchical algorithm is iterative and still based on deflations, but is implemented with care. The algorithm we proposed in this paper is based on similar ideas, and allows to exactly compute the CP decomposition by means of rank-one approximations only. The core of the paper is the analysis of convergence, showing that the CP decomposition is obtained almost surely. Our analysis is also corroborated by means of numerical examples.

The paper is organized as follows. In Section 2 several rank-one approximation algorithms are proposed. In Section 3, we describe our deflation algorithm. In Section 4 we discuss some issues about global convergence. Finally, computer results are reported in Section 5.

The notation employed is as follows: scalars are denoted by lowercase letters, we use calligraphic letters for tensors, boldface capital letters for matrices and boldface lowercase letters for vectors. Lastly, dimensions of tensors, matrices or vectors are denoted with plain capitals.

2. RANK-ONE APPROXIMATION

Let $\mathbb{K}$ denote the real or the complex field, and $\boldsymbol{\mathcal{T}}$ be a tensor in $\mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$. The best rank-one approximation is denoted $\phi(\boldsymbol{\mathcal{T}})$, and is formulated by the optimization problem:

$$\begin{aligned} \phi(\boldsymbol{\mathcal{T}}) = arg \min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}\|_F^2 \\ s.t.\ \text{rank}\{\boldsymbol{\mathcal{X}}\} = 1 \end{aligned} \tag{1}$$

This problem always has a solution because the set of rank-one tensors is known to be closed [13]. However, there

This work has been funded by the European Research Council under the 7th Framework Program FP7/2007-2013 Grant Agreement no. 320594, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) under the program Ciências sem Fronteiras.

is no algebraic method to find the exact solution to this problem, even if there exist efficient algorithms. For practical purposes, we shall now revue four possible ways to estimate the best rank-1 approximation. (i) First, the ALS algorithm is an option that guarantees global convergence for generic tensors in this case [14]. (ii) Another way to obtain a good estimate of the best rank-one approximation is based on truncating the higher order singular value decomposition (T-HOSVD); a rank-one approximation is constructed from the first column of each factor matrix [15]. In the case of rank-one approximations, rank and multilinear rank indeed coincide. (iii) A third way is to compute the dominant singular triplet of a sequence of tensors of decreasing order. (iv) Lastly, one can improve the previous approach by also computing orthogonal projections of a sequence of unfolding tensors of increasing order. The two latter procedures are described below.

ALGORITHM SEROA
(Sequential Rank-One Approximation)

The algorithm proceeds in $N-1$ steps, for a Nth order tensor.

Step 1: unfold tensor $\boldsymbol{\mathcal{T}}$ into a matrix $\mathbf{T}^{(1)}$ of size $I_1 \times I_2 I_3 \dots I_N$ and compute its SVD as $\mathbf{T}^{(1)} = \mathbf{U}^{(1)}\boldsymbol{\Sigma}^{(1)}\mathbf{V}^{(1)\mathsf{T}}$. Set the dominant left singular vector $\mathbf{u}_1$ as the 1st mode of $\boldsymbol{\mathcal{X}}$.

Step n, $1 < n < N-1$: reshape the dominant right singular vector of $\mathbf{T}^{(n-1)}$into a matrix $\mathbf{T}^{(n)}$ of size $I_n \times I_{n+1} \dots I_N$, and set the dominant left singular vector $\mathbf{u}_n$ of $\mathbf{T}^{(n)}$ as the nth mode of $\boldsymbol{\mathcal{X}}$.

Step $N-1$: reshape the dominant right singular vector of $\mathbf{T}^{(N-2)}$ into a $I_{N-1} \times I_N$ matrix $\mathbf{T}^{(N-1)}$. Set the dominant left and right singular vectors to $\mathbf{u}_{N-1}$ and $\mathbf{u}_N$, respectively.

The output is a suboptimal rank-one approximation $\boldsymbol{\mathcal{X}} = \hat{\phi}(\boldsymbol{\mathcal{T}}) = \lambda \cdot \otimes_{n=1}^{N} \mathbf{u}_n$, with $\lambda = \langle \boldsymbol{\mathcal{T}}, \otimes_{n=1}^{N} \mathbf{u}_n \rangle$.

ALGORITHM SEROAP
(Sequential Rank-One Approximation and Projection)

The algorithm proceeds in $2N-4$ steps, for a Nth order tensor.

1. Order Reduction-Fitting Phase:

Step n, $1 \le n < N-2$: reshape the dominant right singular vector of $\mathbf{T}^{(n)}$into a matrix $\mathbf{T}^{(n+1)}$ of size $I_{n+1} \times I_{n+2} \dots I_N$, and compute its SVD as $\mathbf{T}^{(n+1)} = \mathbf{U}^{(n+1)}\boldsymbol{\Sigma}^{(n+1)}\mathbf{V}^{(n+1)\mathsf{H}}$.

Step $N-2$: reshape the dominant right singular vector of $\mathbf{T}^{(N-2)}$ into a $I_{N-1} \times I_N$ matrix $\mathbf{T}^{(N-1)}$, and compute its SVD as $\mathbf{T}^{(N-1)} = \mathbf{U}^{(N-1)}\boldsymbol{\Sigma}^{(N-1)}\mathbf{V}^{(N-1)\mathsf{H}}$. Define $\mathbf{w}_{N-1} = \mathbf{v}_{N-1} \otimes \mathbf{u}_{N-1}$ as the Kronecker product between the dominant right and left singular vectors of $\mathbf{T}^{(N-1)}$.

2. Order Increase-Projection Phase:

Step n, $1 \le n < N-2$: define the matrix $\mathbf{W}^{(N-n+1)}$ of size $I_{N-n-1} \times I_{N-n} \dots I_N$, whose rows are the orthogonal projections of the rows of $\mathbf{T}^{(N-n-1)}$ onto $\mathbf{w}_{N-n}$. Reshape $\mathbf{W}^{(N-n+1)}$ into a vector $\mathbf{w}_{N-n+1}$.

Step $N-2$: define the matrix $\mathbf{W}^{(1)}$ of size $I_1 \times I_2 \dots I_N$,whose rows are the orthogonal projections of the rows of $\mathbf{T}^{(1)}$ onto $\mathbf{w}_2$.

The matrix $\mathbf{W}^{(1)}$ is the mode-1 unfolding of a suboptimal rank-one approximation $\boldsymbol{\mathcal{X}} = \hat{\phi}(\boldsymbol{\mathcal{T}}) = \boldsymbol{\mathcal{W}}$.

Remarks

1. In algorithms SeROA and SeROAP, dimensions can be permuted before proceeding. In particular, it can be attractive to choose I_1 to be the largest dimension, and to sort them in decreasing order.

2. The computational complexity of SeROA is smaller than that of the algorithm based on T-HOSVD, since the tensor order, and hence the dimensions of the associated unfolding matrices, decrease at each step.

3. T-HOSVD, SeROA and SeROAP algorithms terminate after a finite number of steps, whereas the ALS algorithm exhibits an unbounded complexity. For this reason, we do not consider ALS in the remainder.

3. COMPLETE EXACT CP DECOMPOSITION

In this section, we present the deflation-based CP decomposition Algorithm (DCPD), which calculates the exact CP tensor decomposition for general tensors. The algorithm follows the idea behind the solution described in [10] with the difference that each rank-one component is calculated directly using rank-one approximation procedures (1).

The algorithm works as follows. In the initialization phase, R rank-one components $\boldsymbol{\mathcal{X}}[1,1], \dots, \boldsymbol{\mathcal{X}}[R,1]$ are computed by successive rank-one approximations and subtractions. Since subtraction of a best rank-one approximation does not generally decrease tensor rank [12], there is a residue denoted by $\boldsymbol{\mathcal{E}}[R,1]$.

Then an iterative process starts. A first rank-one component is generated from the sum of that residue and tensor $\boldsymbol{\mathcal{X}}[1,1]$. A new residue, $\boldsymbol{\mathcal{E}}[1,2]$, is then determined. These operations are performed in first part of *while* loop.

The second *for* loop begins with the addition of the preceding residue with the second rank-one component. This procedure continues until all the remaining rank-one components are updated and a new residue $\boldsymbol{\mathcal{E}}[R,2]$ is generated in the end of this loop. The *while* loop continues until a stopping criterion $l = l_{max}$ is achieved. We have an exact tensor decomposition when $\|\boldsymbol{\mathcal{E}}\|_F \approx 0$. Note that $\phi(\cdot)$ will have to be replaced by a suboptimal solution $\hat{\phi}(\cdot)$, as one of those described in Section 2.

4. CONVERGENCE ANALYSIS

Some results on global convergence of DCPD using the best rank-one approximation can be delineated for general tensors.

input : $\mathcal{T} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data,
R: rank parameter.
output: $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(R)} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: rank one components, and $\mathcal{E} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: residue tensor.

$\mathcal{Y}[1,1] \leftarrow \mathcal{T}$;
for $r = 1$ *to* R *;* /* $l = 1$ */
do
  $\mathcal{X}[r,1] = \phi(\mathcal{Y}[r,1])$;
  $\mathcal{Y}[r+1,1] = \mathcal{Y}[r,1] - \mathcal{X}[r,1]$;
end
$\mathcal{E}[R,1] = \mathcal{Y}[R+1,1]$;
$l = 2$;
while $l \leq l_{max}$ **do**
  $\mathcal{Y}[1,l] \leftarrow \mathcal{X}[1,l-1] + \mathcal{E}[R,l-1]$;
  $\mathcal{X}[1,l] \leftarrow \phi(\mathcal{Y}[1,l])$;
  $\mathcal{E}[1,l] \leftarrow \mathcal{Y}[1,l] - \mathcal{X}[1,l]$;
  for $r = 2$ *to* R **do**
    $\mathcal{Y}[r,l] \leftarrow \mathcal{X}[r,l-1] + \mathcal{E}[r-1,l]$;
    $\mathcal{X}[r,l] \leftarrow \phi(\mathcal{Y}[r,l])$;
    $\mathcal{E}[r,l] \leftarrow \mathcal{Y}[r,l] - \mathcal{X}[r,l]$;
  end
end
foreach r **do**
  $\mathcal{X}^{(r)} \leftarrow \mathcal{X}[r, l_{max}]$
end
$\mathcal{E} \leftarrow \mathcal{E}[R, l_{max}]$;

Algorithm 1: DCPD algorithm

To start with, let's state some basic theoretical results.

Lemma 4.1 *Let $\mathcal{X}$ be a rank-one tensor and ϕ the best rank-one approximation function. Then, for any tensor $\mathcal{E}$:*

$$\|\mathcal{X} + \mathcal{E} - \phi(\mathcal{X} + \mathcal{E})\|_F \leq \|\mathcal{E}\|_F. \quad (2)$$

Proof. By definition, $\phi(\mathcal{X} + \mathcal{E})$ is a best rank-one approximation of $\mathcal{X} + \mathcal{E}$. In particular, $\mathcal{X}$ cannot be a strictly better rank-1 approximation than $\phi(\mathcal{X}+\mathcal{E})$, which means that $\|\mathcal{X} + \mathcal{E} - \phi(\mathcal{X} + \mathcal{E})\| \leq \|\mathcal{X} + \mathcal{E} - \mathcal{X}\| = \|\mathcal{E}\|$. ■

Let us look at the implications of equality in (2). To do that, we need the lemma below.

Lemma 4.2 *Let $\phi(\mathcal{T})$ be a best rank-one approximation of a tensor $\mathcal{T}$. Then $\|\mathcal{T} - \phi(\mathcal{T})\|_F^2 = \|\mathcal{T}\|_F^2 - \|\phi(\mathcal{T})\|_F^2$.*

Proof. The proof just needs the fact that the set of rank-one tensors is a linear cone. Any rank-one tensor can be written as $\mathcal{V} = \lambda\bar{\mathcal{V}}$ where $\|\bar{\mathcal{V}}\| = 1$. The best rank-1 approximate of $\mathcal{T}$ is a stationary point of $\Upsilon(\lambda, \bar{\mathcal{V}}) = \|\mathcal{T} - \lambda\bar{\mathcal{V}}\|_F^2$. The cancellation of the derivative $\partial\Upsilon/\partial\lambda$ yields $\lambda\|\bar{\mathcal{V}}\|^2 - \langle\mathcal{T}, \bar{\mathcal{V}}\rangle = 0$, and eventually $\lambda = \langle\mathcal{T}, \bar{\mathcal{V}}\rangle$. Now plug this back in the product $\langle\mathcal{T}-\lambda\bar{\mathcal{V}}, \lambda\bar{\mathcal{V}}\rangle$ and get $\lambda^*\langle\mathcal{T}, \bar{\mathcal{V}}\rangle - \lambda\lambda^*\|\bar{\mathcal{V}}\|^2 = 0$. Hence $\mathcal{T} - \mathcal{V}$ and $\mathcal{V}$ are orthogonal, and the result follows. ■

Corollary 4.3 *Let $\mathcal{X}$ be a rank-one tensor and $\mathcal{E}$ any general tensor. If $\|\mathcal{X} + \mathcal{E} - \phi(\mathcal{X} + \mathcal{E})\|_F = \|\mathcal{E}\|_F$, then $\langle\mathcal{E}, \mathcal{X}\rangle = 0$.*

Proof. By hypothesis, we have $\|\mathcal{X}+\mathcal{E}-\phi(\mathcal{X}+\mathcal{E})\|_F = \|\mathcal{X}+\mathcal{E}-\mathcal{X}\|_F$. This means that $\mathcal{X}$ reaches the same minimal value of the objective as $\phi(\mathcal{X} + \mathcal{E})$. Hence $\mathcal{X}$ is a best rank-one approximation, and Lemma 4.2 applies, which yields that $\mathcal{X} + \mathcal{E} - \mathcal{X} = \mathcal{E}$ is orthogonal to $\mathcal{X}$. ■

We are now ready to present the first important proposition about convergence. In the remainder, we shall denote by $\mathfrak{A}$ the DCPD algorithm.

Proposition 4.4 *In algorithm $\mathfrak{A}$, if function ϕ is a best rank-one approximation, then $s_l = \|\mathcal{E}[R,l]\|_F$ is a monotonically decreasing sequence.*

Proof. In the second *for* loop of algorithm $\mathfrak{A}$, we have

$$\|\mathcal{E}[r,l]\|_F = \|\mathcal{Y}[r,l] - \mathcal{X}[r,l]\|_F$$

$$= \|\mathcal{X}[r,l-1] + \mathcal{E}[r-1,l] - \phi(\mathcal{X}[r,l-1] + \mathcal{E}[r-1,l])\|_F.$$

By Lemma 4.1, we conclude that

$$\|\mathcal{E}[r,l]\|_F \leq \|\mathcal{E}[r-1,l])\|_F.$$

In particular, $\|\mathcal{E}[R,l]\|_F \leq \|\mathcal{E}[R-1,l])\|_F$. Similarly, in the beginning of *while* loop, $\|\mathcal{E}[1,l+1]\|_F \leq \|\mathcal{E}[R,l]\|_F$. Thus, it is easy to see that $\|\mathcal{E}[R,l+1]\|_F \leq \|\mathcal{E}[R,l]\|_F$. ■

Proposition 4.4 does not guarantee that the residue $\|\mathcal{E}[R,l]\|_F$ converges to zero. Even when $\|\mathcal{E}[R,l]\|_F < \|\mathcal{E}[R,l+1]\|_F$, for all $l \geq 1$, the sequence (s_l) could indeed converge to a nonzero constant.

Remark 4 *Notice that when $rank(\mathcal{T}) > R$, the residue in algorithm $\mathfrak{A}$ never converges to zero.*

In order to construct more complete results of convergence, we introduce the following definition:

Definition 4.5 *Given a tensor $\mathcal{T}$, $\mathfrak{A}$ is (δ, R)-convergent if there exists δ, $0 \leq \delta \leq 1$, such that for all $l > 1$, $\|\mathcal{E}[R, l+1]\|_F \leq \delta\|\mathcal{E}[R,l]\|_F$.*

Remark 5 *For $\delta < 1$, after l iterations of while loop in $\mathfrak{A}$, $\|\mathcal{E}[R,l+1]\|_F \leq \delta^l\|\mathcal{E}[R,1]\|_F$. Hence $\|\mathcal{E}[R,l+1]\|_F \to 0$ when, $l \to \infty$.*

Let $\mathcal{T}$ be the set of all tensors with entries in $\mathbb{K}$. Define the sets: $\mathcal{T}^{(R)} = \{\mathcal{T} \in \mathcal{T} : \text{ rank}\{\mathcal{T}\} \leq R\}$ and $\mathcal{S}_\delta^{(R)} = \{\mathcal{T} \in \mathcal{T}^{(R)} : \mathfrak{A} \text{ is } (\delta, R)\text{-convergent for } \mathcal{T}\}$, for some $0 \leq \delta \leq 1$. Notice that $\mathcal{S}_{\delta_1}^{(R)} \subseteq \mathcal{S}_{\delta_2}^{(R)}$ for $\delta_1 \leq \delta_2$ and $\mathcal{S}_1^{(R)} = \mathcal{T}^{(R)}$ due to Definition 4.5.

Proposition 4.6 *Let tensors be distributed within $\mathcal{T}^{(R)}$ according to an absolutely continuous probability measure. Then for any $\varepsilon > 0$, there exists δ, $0 \leq \delta < 1$, such that algorithm $\mathfrak{A}$ is (δ, R)-convergent with probability $(1 - \varepsilon)$.*

Proof. By the continuity of the measure, for any $\epsilon > 0$, there exists $0 \leq \delta < 1$, such that $\mu(\mathbb{S}_{\delta}^{(R)}) = 1 - \epsilon$. ■

Remark 6 *If the probability distribution is isotropic, an relationship can be stated between δ and ε, given by the ratio between the solid angle and the surface of the hypersphere.*

5. COMPUTER RESULTS

5.1. Performance of rank-one approximation functions

In this section, we compare the performance of three rank-one approximation functions, as described in Section 2. The table below shows the mean and variance of the rank-one approximation error for a sample of 10000 tensors in two different scenarios: $3 \times 3 \times 3$ rank-3 tensors, and $4 \times 4 \times 4$ rank-5 tensors.

	Scenario 1		Scenario 2	
Algorithm	mean	variance	mean	variance
T-HOSVD	0.8560	0.1355	2.3950	0.3733
SeROA	1.6113	0.2629	3.3500	0.4848
SeROAP	0.8226	0.1121	2.2667	0.2786

5.2. Performance of DCPD Algorithm

In order to produce satisfactory results with algorithm DCPD, we have defined the following stopping criteria: (i) $|\|\boldsymbol{\mathcal{E}}[R, l+1]\|_F - \|\boldsymbol{\mathcal{E}}[R, l]\|_F| < 10^{-10}$ and (ii) $l_{max} = 10000$. For various ranges of tensors, we plot the estimation error $E_{est} = \max_{k=1}^{100} \min_{\sigma} \sum_{r=1}^{R} \|\boldsymbol{\mathcal{X}}_r^{(k)} - \widetilde{\boldsymbol{\mathcal{X}}}_{\sigma(r)}^{(k)}\|_F$, where $\boldsymbol{\mathcal{X}}_r^{(k)}$ and $\widetilde{\boldsymbol{\mathcal{X}}}_{\sigma(r)}^{(k)}$ are the actual and estimated rth rank-one component, and $\sigma(\cdot)$ denotes a permutation.

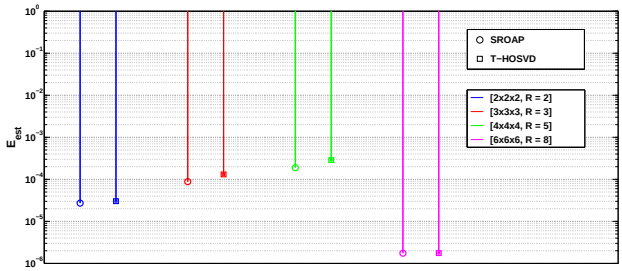

Fig. 1. Estimation error for DTD algorithm.

Figure 1 shows the maximum error E_{est} calculated for 100 random tensors for the following real tensor scenarios: $2 \times 2 \times 2$ rank-2 tensors, $3 \times 3 \times 3$ rank-3 tensors, $4 \times 4 \times 4$ rank-5 tensors, and $6 \times 6 \times 6$ rank-8 tensors. The simulations are performed with both rank-one approximations: T-HOSVD and SeROAP. Notice that in all scenarios the Kruskal uniqueness condition [16] is satisfied..

5.3. Example 1: $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{3\times3\times3}$, $rank(\boldsymbol{\mathcal{T}}) = 4$.

In the example below, the Kruskal condition is not satisfied. We take the example 3.1 from [11], whose mode-1 unfolding is given by

$$\mathbf{T}^{(1)} = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Here, for both T-HOSVD and SeROAP implementations of $\phi(\cdot)$, the algorithm converges to an exact decomposition in the first iteration ($\|\boldsymbol{\mathcal{E}}[4, 1]\|_F = 0$) in a few milliseconds. The first *for* loop in algorithm DCPD is enough to guarantee an exact solution. Contrary to [11], we do not need a sophisticated algorithm to decompose this tensor.

5.4. Example 2: $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{6\times6\times6}$, $rank(\boldsymbol{\mathcal{T}}) = 8$.

The residue $\|\boldsymbol{\mathcal{E}}[R, l]\|_F$ is plotted in Figure 2 as a function of iteration l. It illustrates the rate of convergence of DCPD for random $6 \times 6 \times 6$ tensors.

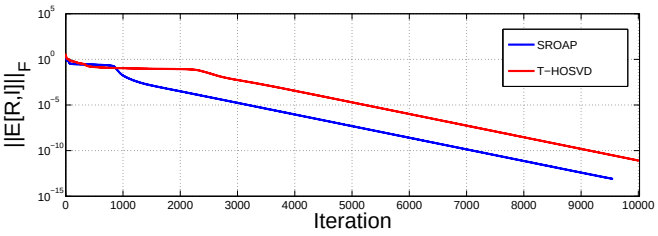

Fig. 2. Convergence rate of a $6 \times 6 \times 6$ rank-8 real random tensor.

6. CONCLUSION

We proposed a deflation algorithm (DCPD) delivering an exact CP decomposition of given rank, based on successive rank-one approximation. Best rank-one approximations are always well-posed and rather easy to compute; we also proposed several algorithms with this purpose (SeROA, SeROAP) and compared them to T-HOSVD. Our main contribution is a proof of convergence of algorithm DCPD, for a class of tensors of large measure. Computer experiments run on random tensors of fixed rank have confirmed our theoretical results.

7. REFERENCES

[1] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis*, Wiley, Chichester UK, 2004.

[2] P. Comon and C. Jutten, Eds., *Handbook of Blind Source Separation, Independent Component Analysis and Applications*, Academic Press, Oxford UK, Burlington USA, 2010, ISBN: 978-0-12-374726-6, hal-00460653.

[3] B. Savas, *Algorithms in Data Mining using Matrix and Tensor Metho ds*, Ph.D. thesis, Linköping Univ. Tech., 2008.

[4] A. L. F. De Almeida, G. Favier, and J. C. M. Mota, "Parafac-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization," *Signal Processing*, vol. 87, no. 2, pp. 337–351, Feb. 2007.

[5] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, pp. 105–122, 2000.

[6] N. Li, S. Kindermann, and C. Navasca, "Some convergence results on the regularized alternating least-squares method for tensor decomposition," *Lin. Algebra Appl.*, vol. 438, no. 2, pp. 796–812, 2013.

[7] A. Uschmajew, "Local convergence of the alternating least squares algorithm for canonical tensor approximation," *SIAM Jour. matrix Analysis*, vol. 33, no. 2, pp. 639–652, 2012.

[8] P. Comon, X. Luciani, and A. L. F. De Almeida, "Tensor decompositions, alternating least squares and other tales," *Jour. Chemometrics*, vol. 23, no. 7-8, pp. 393–405, Aug. 2009.

[9] P. Paatero, "A weighted non-negative least squares algorithm for three-way Parafac factor analysis," *Chemometrics Intell. Lab. Syst.*, vol. 38, pp. 223–242, 1997.

[10] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 3, pp. 708–721, 2009.

[11] A.-H. Phan, P. Tichavsky, and A. Cichocki, "Deflation method for Candecomp/Paarafac tensor decomposition," in *ICASSP*, Firenze, Italy, May 4-9 2014, pp. 6736–6740.

[12] A. Stegeman and P. Comon, "Subtracting a best rank-1 approximation does not necessarily decrease tensor rank," *Linear Algebra Appl.*, vol. 433, no. 7, pp. 1276–1300, Dec. 2010, hal-00512275.

[13] P. Comon, B. Mourrain, L-H. Lim, and G. Golub, "Genericity and rank deficiency of high order symmetric tensors," in *ICASSP'06*, Toulouse, May 14-19 2006.

[14] L. Wang and T. chu, "On the global convergence of the alternating least squares method for rank-one approximation to generic tensors," *SIAM J. Matrix Anal. Appl.*, vol. 35, pp. 1058–1072, 2014.

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1,R2,. . . RN) approximation of high-order tensors," *SIAM Jour. Matrix Ana. Appl.*, vol. 21, no. 4, pp. 1324–1342, Apr. 2000.

[16] P. Comon, "Tensors: a brief introduction," *IEEE Sig. Proc. Magazine*, vol. 31, no. 3, pp. 44–53, May 2014, hal-00923279.