Tensor Decompositions, Alternating Least Squares and other Tales

Pierre Comon, Xavier Luciani, André L. F. de Almeida

Journal of Chemometrics, vol.23, pp.393–405, Aug. 2009

Tensor Decompositions, Alternating Least Squares and other Tales

P. Comon, X. Luciani and A. L. F. de Almeida

Special issue, *Journal of Chemometrics*
in memory of R. Harshman

August 16, 2009

Abstract

This work was originally motivated by a classification of tensors proposed by Richard Harshman. In particular, we focus on simple and multiple "bottlenecks", and on "swamps". Existing theoretical results are surveyed, some numerical algorithms are described in details, and their numerical complexity is calculated. In particular, the interest in using the ELS enhancement in these algorithms is discussed. Computer simulations feed this discussion.

1 Introduction

Richard Harshman liked to explain Multi-Way Factor Analysis (MFA) as one tells a story: with words, sentences, appealing for intuition, and with few formulas. Sometimes, the story turned to a tale, which required the belief of participants, because of the lack of proof of some strange –but correct– results.

MFA has been seen first as a mere nonlinear Least Squares problem, with a simple objective criterion. In fact, the objective is a polynomial function of many variables, where some separate. One could think that this kind of objective is easy because smooth, and even infinitely differentiable. But it turns out that things are much more complicated than they may appear to be at first glance. Nevertheless, the Alternating Least Squares (ALS) algorithm has been mostly utilized to address this minimization problem, because of its programming simplicity. This should not hide the inherently complicated theory that lies behind the optimization problem.

Note that the ALS algorithm has been the subject of much older tales in the past. In fact, it can be seen as a particular instance of the nonlinear Least Squares problem addressed in [28], where variables separate. The latter analysis gave rise in particular to the Variable Projection algorithm, developed by a great figure of Numerical Analysis, who also passed away recently, barely two months before Richard.

Tensors play a wider and wider role in numerous application areas, much beyond Chemometrics. Among many others, one can mention Signal Processing for Telecommunications [55] [23] [21], and Arithmetic Complexity [35] [62] [8], which are two quite different frameworks. One of the reasons of this success is that MFA can often replace Principal Component Analysis (PCA), when available data measurements can be arranged in a meaningful tensor form [57]. When this is not the case, that is, when the observation diversity is not sufficient (in particular when a 3rd order tensor has proportional matrix slices), one can sometimes resort to High-Order Statistics (HOS), allowing to build symmetric tensors of arbitrarily large order from the data [17].

In most of these applications, the decomposition of a tensor into a sum of rank-1 terms (cf. Section 2) allows to solve the problem, provided the decomposition can be shown to be essentially unique (i.e. unique up to scale and permutation). Necessary and sufficient conditions ensuring existence and essential uniqueness will be surveyed in Section 2.3. The main difficulty stems from the fact that actual tensors may not exactly satisfy the expected model, so that the problem is eventually an approximation rather than an exact decomposition.

Richard Harshman was very much attached to the efficiency of numerical algorithms when used to process actual sets of data. He pointed out already in 1989 [36] that one cause of slow convergence (or lack of convergence) can be attributed to "degeneracies" of the tensor (cf. Section 2.2). Richard Harshman proposed the following classification:

1. *Bottleneck.* A bottleneck occurs when two or more factors in a mode are almost collinear [49].

2. *Swamp.* If a bottleneck exists in all the modes, then we have what has been called by R. Harshman and others a "swamp" [43] [52] [49].

3. *CP-degeneracies* may be seen as particular cases of swamps, where some factors diverge and at the same time tend to cancel each other as the goodness of fit progresses [36] [47] [30].

CP-degeneracies occur when one attempts to approximate a tensor by another of lower rank, causing two or more factors to tend to infinity, and at the same time to almost cancel each other, giving birth to a tensor of higher rank (cf. Section 2.2). The mechanism of these "degeneracies" is now rather well known [47] [63] [58] [59] [14], and has been recently better understood [56] [33]. If one dimension is equal to 2, special results can be obtained by viewing third order tensors as matrix pencils [60]. According to [56], the first example of a rank-r tensor sequence converging to a tensor of rank strictly larger than r was exhibited by D.Bini as early as the seventies [3]. CP-degeneracies can be avoided if the set in which the best approximate is sought is closed. One solution is thus to define a closed subset of the set of lower-rank tensors; another is to define a superset, yielding the notion of border rank [62] [4] [8].

On the other hand, swamps may exist even if the problem is well posed. For example, Paatero reports in [47] precise examples where the trajectory from a well

conditioned initial point to a well-defined solution has to go around a region of higher rank, hence a "swamp". In such examples, the convergence to the minimum can be slowed down or even compromised.

One solution against bottlenecks that we developed together with R. Harshman just before he passed away was the Enhanced Line Search (ELS) principle [49], which we subsequently describe. This principle, when applied to the Alternating Least Squares (ALS) algorithm, has been shown in [49] to improve significantly its performances in the sense that it would decrease the risk to terminate in a local minimum, and more importantly also need an often much smaller number of iterations. Also note that an extension to the complex case has been recently proposed [45].

However, the additional computational cost per iteration was shown to be negligible only when the rank was sufficiently large compared to the dimensions; see condition (9). Nevertheless, we subsequently show that this condition is not too much restrictive, since one can always run a prior dimensional reduction of the tensor to be decomposed with the help of a HOSVD truncation, which is a compression means that many users practice for several years [24] [55] [7].

The goal of the paper is two-fold. First we give a summary of the state of the art, and make the distinction between what is known but attributed to usual practice, and thus belongs to the world of conjectures, and what has been rigorously proved. Second, we provide details on numerical complexities and show that ELS can be useful and efficient in a number of practical situations.

Section 2 defines a few notations and addresses the problems of existence, uniqueness, and genericity of tensor decompositions. Section 3 gathers precise update expressions appearing in iterative algorithms used in Section 5. We shall concentrate on Alternating Least Squares (ALS), Gradient descent with variable step, and Levenberg-Marquardt algorithms, and on their versions improved with the help of the Enhanced Line Search (ELS) feature, with or without compression, the goal being to face bottlenecks more efficiently. Section 4 gives orders of magnitude of their computational complexities, for subsequent comparisions in Section 5. In memory of Richard, we shall attempt to give the flavor of complicated concepts with simple words.

2 Tensor decomposition

2.1 Tensor rank and the CP decomposition

Tensor spaces. Let $\mathbb{V}^{(1)}$, $\mathbb{V}^{(2)}$, $\mathbb{V}^{(3)}$ be three vector spaces of dimension I, J and K respectively. An element of the vector space $\mathbb{V}^{(1)} \otimes \mathbb{V}^{(2)} \otimes \mathbb{V}^{(3)}$, where $\otimes$ denotes the outer (tensor) product, is called a tensor. Let us choose a basis $\{\mathbf{e}_i^{(\ell)}\}$ in each of the three vector spaces $\mathbb{V}^{(\ell)}$. Then any tensor $\mathcal{T}$ of that vector space of dimension IJK has coordinates T_{ijk} defined by the relation

$$\mathcal{T} = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} T_{ijk}\, \mathbf{e}_i^{(1)} \otimes \mathbf{e}_j^{(2)} \otimes \mathbf{e}_k^{(3)}$$

Multi-linearity. If a change of basis is performed in space $\mathbb{V}^{(\ell)}$, defined by $\mathbf{e}_i^{(\ell)} = \mathbf{Q}^{(\ell)}\,\mathbf{e}_i'^{(\ell)}$, for some given matrices $\mathbf{Q}^{(\ell)}$, $1 \leq \ell \leq d$ then the new coordinates T'_{pqr} of tensor $\mathcal{T}$ may be expressed as a function of the original ones as

$$T'_{pqr} = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} Q^{(1)}_{pi}\,Q^{(2)}_{qj}\,Q^{(3)}_{rk}\;T_{ijk}$$

This is a direct consequence of the construction of tensors spaces, which can be readily obtained by merely pugging the exporession of $\mathbf{e}_i'^{(\ell)}$ in that of $\mathcal{T}$. By convention, this multilinear relation between arrays of coordinates is chosen to be written as [56] [14]:

$$\mathbf{T}' = (\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \mathbf{Q}^{(3)}) \cdot \mathbf{T}$$

CP decomposition. Any tensor $\mathcal{T}$ admits a decomposition into a sum of rank-1 tensors. This decomposition takes the form below in the case of a 3rd order tensor:

$$\mathcal{T} = \sum_{p=1}^{F} \mathbf{a}(p) \otimes \mathbf{b}(p) \otimes \mathbf{c}(p) \tag{1}$$

where $\mathbf{a}(p)$, $\mathbf{b}(p)$ and $\mathbf{c}(p)$ denote vectors belonging to spaces $\mathbb{V}^{(1)}$, $\mathbb{V}^{(2)}$, $\mathbb{V}^{(3)}$.

Denote A_{ip} (resp. B_{jp} and C_{kp}) the coordinates of vector $\mathbf{a}(p)$ in basis $\{\mathbf{e}_i^{(1)}, 1 \leq i \leq I\}$, (resp. $\mathbf{b}(p)$ in $\{\mathbf{e}_j^{(2)}, 1 \leq j \leq J\}$ and $\mathbf{c}(p)$ in $\{\mathbf{e}_k^{(3)}, 1 \leq k \leq K\}$). Then one can rewrite this decomposition as

$$\mathcal{T} = \sum_{p=1}^{F} (\sum_{i=1}^{I} A_{ip}\mathbf{e}_i^{(1)}) \otimes (\sum_{j=1}^{J} B_{jp}\mathbf{e}_j^{(2)}) \otimes (\sum_{k=1}^{K} C_{kp}\mathbf{e}_k^{(3)})$$

or equivalently

$$\mathcal{T} = \sum_{ijk} (\sum_{p=1}^{F} A_{ip}B_{jp}C_{kp})\;\mathbf{e}_i^{(1)} \otimes \mathbf{e}_j^{(2)} \otimes \mathbf{e}_k^{(3)}$$

which shows that arrays of coordinates are related by

$$T_{ijk} = \sum_{p=1}^{F} A_{ip}\,B_{jp}\,C_{kp} \tag{2}$$

One may see this equation as a consequence of the multi-linearity property. In fact, if one denotes by $\mathbf{I}$ the $F \times F \times F$ three-way diagonal array, having ones on its diagonal, then (2) can be rewritten as

$$\mathbf{T} = (\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{I}$$

Note that, contrary to (1), equation (2) is basis-dependent.

Tensor rank. The *tensor rank* is the minimal number F of rank-1 terms such that the equality (1), or equivalently (2), holds true. Tensor rank always exists and is well defined. This decomposition can be traced back to the previous century with the works of Hitchcock [32]. It has been later independently referred to as the *Canonical Decomposition* (CANDECOMP) [9] or *Parallel Factor* (PARAFAC) [29] [31]. We shall simply refer to it via the acronym CP, as usual in this journal.

Points of terminology. The *order* of a tensor is the number of tensor products appearing in its definition, or equivalently, the number of its ways, i.e. the number of indices in its array of coordinates (2). Hence a tensor of order d has d *dimensions*. In Multi-Way Factor Analysis, the columns $\mathbf{a}(p)$ of matrix $\mathbf{A}$ are called *factors* [57] [34]. Another terminology that is not widely assumed is that of *modes* [34]. A mode actually corresponds to one of the d linear spaces $\mathbb{V}^{(\ell)}$. For instance, according to the classification sketched in Section 1, a bottleneck occurs in the second mode if two columns of matrix $\mathbf{B}$ are almost collinear.

2.2 Existence

In practice, one often prefers to fit a multi-linear model of lower rank, $F < \text{rank}\{\mathcal{T}\}$, fixed in advance, so that we have to deal with an *approximation problem*. More precisely, it is aimed at minimizing an objective function of the form

$$\Upsilon(\mathbf{A},\mathbf{B},\mathbf{C}) = ||\mathcal{T} - \sum_{p=1}^{F} \mathbf{a}(p)\otimes\mathbf{b}(p)\otimes\mathbf{c}(p)||^2 \tag{3}$$

for a given tensor norm defined in $\mathbb{V}^{(1)}\otimes\mathbb{V}^{(2)}\otimes\mathbb{V}^{(3)}$. In general, the Euclidean norm is used. Unfortunately, the approximation problem is not always well posed. In fact, as pointed out in early works of R.Harshman [36] [30], the infimum may never be reached, for it can be of rank higher than F.

CP-degeneracies. According to the classification proposed by Richard Harshman and sketched in Section 1, CP-degeneracies can occur when one attempts to approximate a tensor by another of lower rank. Let us explain this in this section, and start with an example.

Example. Several examples are given in [56] [14], and in particular the case of a rank-2 tensor sequence converging to a rank-4 tensor. Here, we give a slightly more tricky example. Let $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ be three linearly independent vectors, in $\mathbb{R}^3$ for instance. Then the following sequence of rank-3 symmetric tensors

$$\mathcal{T}(n) = n^2(\mathbf{x}+\frac{1}{n^2}\,\mathbf{y}+\frac{1}{n}\,\mathbf{z})^{\otimes 3} + n^2(\mathbf{x}+\frac{1}{n^2}\,\mathbf{y}-\frac{1}{n}\,\mathbf{z})^{\otimes 3} - 2\,n^2\mathbf{x}^{\otimes 3}$$

converges towards the tensor below, which may be proved to be of rank 5 (despite its 6 terms) as n tends to infinity:

$$\mathcal{T}(\infty) = \mathbf{x}\otimes\mathbf{x}\otimes\mathbf{y} + \mathbf{x}\otimes\mathbf{y}\otimes\mathbf{x} + \mathbf{y}\otimes\mathbf{x}\otimes\mathbf{x} + \mathbf{x}\otimes\mathbf{z}\otimes\mathbf{z} + \mathbf{z}\otimes\mathbf{x}\otimes\mathbf{z} + \mathbf{z}\otimes\mathbf{z}\otimes\mathbf{x}$$

Rank 5 is the maximal rank that can achieve third order tensors of dimension 3. To conclude, tensor $\mathcal{T}(\infty)$ can be approximated arbitrarily well by rank-3 tensors $\mathcal{T}(n)$, but the limit is never reached since it has rank 5. Note that this particular example cannot be directly generated by Leibniz tensors described in [56], which could have been used equally well for our demonstration (note however that $\mathcal{T}(\infty)$ can probably be expressed as a Leibnitz operator after a multilinear change of coordinates).

Classification of third order tensors of dimension 2. Contrary to symmetric tensors in the complex field, whose properties have been studied in depth [5] [2] [20] [14] (see also Section 2.3) and utilized in various applications including Signal Processing [16], much less is known concerning non symmetric tensors [62] [8], and even less in the real field. However, the geometry of 2-dimensional 3rd order asymmetric tensors is fully known (symmetric or not, real or not). Since the works of Bini [3], other authors have shed light on the subject [63] [58] [47], and all the proofs can now be found in [26] [56]. Since the case of 2-dimensional tensors is rather well understood, it allows to figure out what *probably* happens in larger dimensions, and is worth mentioning.

The decomposition of $2 \times 2 \times 2$ tensors can be computed with the help of matrix pencils if one of the $2d = 6$ matrix slices is invertible (one can show that if none of them is invertible, then the tensor is necessarily of rank 1). But hyperdeterminants can avoid this additional assumption [56], which may raise problems in higher dimensions. Instead of considering the characteristic polynomial of the matrix pencil, $\det(\mathbf{T}_1 + \lambda\,\mathbf{T}_2)$, one can work in the projective space by considering the homogeneous polynomial $p(\lambda_1, \lambda_2) = \det(\lambda_1\,\mathbf{T}_1 + \lambda_2\,\mathbf{T}_2)$. This does not need invertibility of one matrix slice. It can be shown [56] to take the simple form:

$$p(\lambda_1, \lambda_2) = \lambda_1^2 \det \mathbf{T}_1 + \frac{\lambda_1\lambda_2}{2}(\det(\mathbf{T}_1 + \mathbf{T}_2) - \det(\mathbf{T}_1 - \mathbf{T}_2)) + \lambda_2^2 \det \mathbf{T}_2$$

It turns out that the discriminant $\Delta(\mathbf{T}_1, \mathbf{T}_2)$ of polynomial $p(\lambda_1, \lambda_2)$ is nothing else but Kruskal's polynomial mentioned in [63] [58] [47], which itself is due to Cayley [10]. As pointed out in [56], the sign of this discriminant is invariant under changes of bases, and tells us about the rank of $\mathcal{T}$: if $\Delta < 0$, $\text{rank}\{\mathcal{T}\} = 3$, and if $\Delta > 0$, $\text{rank}\{\mathcal{T}\} = 2$.

Thus the variety defined by $\Delta = 0$ partitions the space into two parts, which hence have a non zero volume. The closed set of tensors of rank 1 lies on this hypersurface $\Delta = 0$. But two other rare cases of tensors of rank 2 and 3 share this hypersurface: in particular, this is where tensors of rank 3 that are the limit of rank-2 tensors are located.

How to avoid degeneracies. In [56] [58] [33], it has been shown that CP-degeneracies necessarily involve at least two rank-1 terms tending to infinity. In addition, if all rank-1 terms tend to infinity, then the rank jump is the largest possible. In our example, all the three tend to infinity, and almost cancel with each other. This is the residual that gives rise to a tensor of higher rank.

If cancellation between diverging rank-1 tensors cannot occur, then neither CP-degeneracy can. This is in fact what happens with tensors with positive entries.

Thus, imposing the non negativity during the execution of successive iterations is one way to prevent CP-degeneracy, if tensors with non negative entries are concerned [40].

In order to guarantee the existence of a minimum, the subset of tensors of rank F must be closed, which is not the case of tensors with free entries, except if $F \leq 1$ [14]. Should this not be the case, one can perform the minimization either over a closed subset (e.g. the cone of tensors with positive entries [40]) or over a closed superset. The most obvious way to define this superset is to define the smallest closed subset containing all tensors of rank F, i.e. their adherence; such a superset is that of tensors with *border rank* at most F [62] [4] [8]. Another idea proposed in [44] is to add a regularization term in the objective function, which avoids the permutation-scale ambiguities to change when iterations progress. The effect is that swamps are better coped with.

However, no solution is entirely satisfactory, as pointed out in [56], so that this problem remains open in most practical cases.

2.3 Uniqueness

The question that can be raised is then: why don't we calculate the CP decomposition for $F = \text{rank}\{\mathcal{T}\}$? The answer generally given is that the rank of $\mathcal{T}$ is unknown. This answer is not satisfactory. By the way, if this were the actual reason, one could choose F to be larger than $\text{rank}\{\mathcal{T}\}$, e.g. an upper bound. This choice is not made, actually for uniqueness reasons. In fact, we recall hereafter that tensors of large rank do not admit a unique CP decomposition.

Typical and generic ranks. Suppose that the entries of a 3-way array are drawn randomly according to a continuous probability distribution. Then the *typical ranks* are those that one will encounter with non zero probability. If only one typical rank exists, it will occur with probability 1 and is called *generic*.

For instance, in the $2 \times 2 \times 2$ case addressed in a previous paragraph, there are two typical ranks: rank 2 and rank 3. In the complex field however, there is only one: all $2 \times 2 \times 2$ tensors generically have rank 2.

In fact the rank of a tensor obtained from real measurements is *typical* with probability 1 because of the presence of measurement errors with a continuous probability distribution. And we now show that the smallest typical rank is known, for any order and dimensions.

With the help of the so-called Terracini's lemma [68], sometimes attributed to Lasker [38] [16], it is possible to compute the *smallest* typical rank for any given order and dimensions, since it coincides with the generic rank (if the decomposition were allowed to be computed in the complex field) [42] [18] [5]. The principle consist of computing the rank of the Jacobian of the map associating the triplet $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ to tensor $\mathbf{T}$ defined in (2) [18]. The same principle extends to more general tensor decompositions [19]. The resulting ranks are reported in Tables 1 and 2. Values reported **in bold** correspond to smallest typical ranks that are now known only via this numerical computation.

K		2				3			4	
J		2	3	4	5	3	4	5	4	5
I	2	2	3	4	4	3	4	5	4	5
	3	3	3	4	5	5	**5**	5	**6**	**6**
	4	4	4	4	5	5	**6**	**6**	**7**	**8**
	5	4	5	5	5	5	**6**	**8**	**8**	**9**
	6	4	6	6	6	6	**7**	**8**	**8**	**10**
	7	4	6	7	7	7	**7**	**9**	**9**	**10**
	8	4	6	8	8	8	8	**9**	**10**	**11**
	9	4	6	8	9	9	9	**9**	**10**	**12**
	10	4	6	8	10	9	10	10	**10**	**12**
	11	4	6	8	10	9	11	11	**11**	**13**
	12	4	6	8	10	9	12	12	12	**13**

Table 1: Smallest typical rank for some third order unconstrained arrays. Frame: finite number of solutions. Underlined: exceptions to the ceil rule.

I		2	3	4	5	6	7	8	9
d	3	2	5	**7**	**10**	**14**	**19**	**24**	**30**
	4	**4**	**9**	**20**	**37**	**62**	**97**		

Table 2: Smallest typical rank for unconstrained arrays of order $d = 3$ and $d = 4$ and with equal dimensions I. Frame: finite number of solutions. Underlined: exceptions to the ceil rule.

One could believe that this generic rank could be computed by just counting the number of degrees of freedom in both sides of decomposition (2). If we do this job we find IJK free parameters in the LHS, and $F(I+J+K-2)$ in the RHS. So we can say that the "expected rank" of a generic tensor would be

$$\bar{F} = \left\lceil \frac{IJK}{I+J+K-2} \right\rceil \tag{4}$$

Now notice that the values given by (4) do not coincide with those given by Tables 1 and 2; generic ranks that differ from expected values are underlined. These values are *exceptions* to the ceil rule. For instance, the row $d = 3$ in Table 2 shows that there is a single exception, which is consistent with [39, p.110]. We conjecture that the "ceil rule" is correct almost everywhere, up to a finite number of exceptions. In fact, this conjecture has been proved in the case of symmetric tensors [14], thanks to Alexander-Hirschowitz theorem [2] [5]. In the latter case, there are fewer exceptions than in the present case of tensors with free entries. Results of [1] also tend to support this conjecture.

Another open problem is the following: what are the other typical ranks, when they exist? For what values of dimensions can one find several typical ranks, and indeed how many can there be? Up to now, only two typical ranks have been proved to exist for some given dimensions [65] [64] [67] [66] [56], not more.

Number of decompositions. These tables not only give the most probable ranks, but allow to figure out when the decomposition may be unique. When the ratio in (2) is an integer, one can indeed hope that the solution is unique, since there are as many unknowns as free equations. This hope is partially granted: if the rank $\bar{F}$ is not an exception, then it is proved that there is a *finite number* of decompositions with probability 1. These cases are shown in a framebox in Tables 1 and 2. In all other cases there are *infinitely many* decompositions. In the symmetric case, stronger results have been obtained recently and prove that the decomposition is *essentially unique* with probability 1 if the dimension does not exceed the order [42]. In the case of non symmetric tensors, such proofs are still lacking.

So why do we go to the approximation problem? In practice, imposing a smaller rank is a means to obtain an essentially unique decomposition (cf. Section 2.4). In fact, it has been proved that symmetric tensors of any order $d \geq 3$ and of dimension 2 or 3 always have an essentially unique decomposition with probability 1 if their rank is strictly smaller than the generic rank [12]. Our conjecture is that this also holds true for non symmetric tensors: *all tensors with a sub-generic rank have a unique decomposition* with probability 1, up to scale and permutation indeterminacies.

Another result developed in the context of Arithmetic Complexity has been obtained by Kruskal in 1977 [35]. The proof has been later simplified [61] [37], and extended to tensors of order d higher than 3 [54]. The theorem states that essential uniqueness of the CP decomposition (equation (2) for $d = 3$) is ensured if

the sufficient condition below is satisfied:

$$2\,F + d - 1 \leq \sum_{\ell=1}^{d} \min(I_\ell, F)$$

where I_ℓ denotes the ℓth dimension. It is already known that this condition is not necessary. Our conjecture goes in the same direction, and our tables clearly show that Kruskal's condition on F above is much more restrictive than the condition that F is strictly smaller than generic. Another interesting result has been obtained in [22] for tensors having one large dimension. The condition of uniqueness is then less restrictive than the above.

2.4 Dimension reduction

When fitting a parametric model, there is always some loss of information; the fewer parameters, the larger the loss. Detecting the optimal number of free parameters to assume in the model, for a given data length, is a standard problem in parameter estimation [53]. If it is taken too small, there is a bias in the model; but if it is too large, parameters suffer from a large estimation variance. At the limit, if the variance of some parameters are infinite, it means that uniqueness is lost for the corresponding parameters. In our model (1), the number of parameters is controlled by the rank F and the dimensions. If the rank F is taken too small, the factors are unable to model the given tensor with enough accuracy; if it is too large, factors are poory estimated, or can even become non unique even if the permutation-scale ambiguity is fixed.

It is clear that assuming that the rank F is generic will lead either to large errors or to non unique decompositions. Reducing the rank is a means to eliminate part of the estimation variance, noise effects and measurement errors. Restoring the essential uniqueness of the model by rank reduction may be seen as a limitation of the estimation variance, and should not be seen as a side effect.

In this section, we propose to limit the rank to a subgeneric value and to reduce the dimensions. Statistical analyses leading to the optimal choices of dimensions and rank are out of our scope, and only numerical aspects are subsequently considered.

Unfolding matrices and multilinear rank. One way to reduce the dimensions of the problem, and at the same time reduce the rank of $\mathcal{T}$, is to truncate the Singular Value Decomposition (SVD) of unfolding matrices of array $\mathbf{T}$. Let's see how to do that. An order d array $\mathbf{T}$ admits d different *unfolding matrices*, also sometimes called *flattening matrices* (other forms exist but are not fundamentally different). Take a $I \times J \times K$ array. For k fixed in the third mode, we have a $I \times J$ matrix that can be denoted as $\mathbf{T}_{::k}$. The collection of these K such matrices can be

arranged in a $KI \times J$ block matrix:

$$\mathbf{T}_{KI\times J} = \begin{bmatrix} \mathbf{T}_{::1} \\ \vdots \\ \mathbf{T}_{::k} \\ \vdots \\ \mathbf{T}_{::K} \end{bmatrix}$$

The two other unfolding matrices, $\mathbf{T}_{IJ\times K}$ and $\mathbf{T}_{JK\times I}$, can be obtained in a similar manner as $\mathbf{T}_{KI\times J}$, by circular permutation of the modes; the latter contain blocks of size $J \times K$ and $K \times I$, respectively.

The rank of the unfolding matrix in mode p may be called the *p-mode rank*. It is known that tensor rank is bounded below by every mode-p rank [25]. The d-uple containing all p-mode ranks is sometimes called the *multilinear rank*.

Kronecker product. Let two matrices $\mathbf{A}$ and $\mathbf{H}$, of size $I \times J$ and $K \times L$ respectively. One defines the Kronecker product $\mathbf{A} \otimes \mathbf{H}$ as the $IK \times JL$ matrix [50] [51]:

$$\mathbf{A} \otimes \mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} a_{11}\mathbf{H} & a_{12}\mathbf{H} & \cdots \\ a_{21}\mathbf{H} & a_{22}\mathbf{H} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

Now denote $\mathbf{a}_j$ and $\mathbf{h}_\ell$ the columns of matrices $\mathbf{A}$ and $\mathbf{H}$. If $\mathbf{A}$ and $\mathbf{H}$ have the same number of columns, on can define the Khatri-Rao product [50] as the $IK \times J$ matrix:

$$\mathbf{A} \odot \mathbf{H} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{a}_1 \otimes \mathbf{h}_1 & \mathbf{a}_2 \otimes \mathbf{h}_2 & \cdots \end{pmatrix}.$$

The Khatri-Rao product is nothing else but the column-wise Kronecker product. Note that the Kronecker product and the tensor product are denoted in a similar manner, which might be confusing. In fact, this usual practice has some reasons: $\mathbf{A} \otimes \mathbf{H}$ is the array of coordinates of the tensor product of the two associated linear operators, in some canonical basis.

High-Order SVD (HOSVD). The Tucker3 decomposition has been introduced by Tucker [70] and studied in depth by L.DeLathauwer [24] [25]. It is sometimes referred to as the High-Order Singular Value Decomposition (HOSVD). A third order array of size $I \times J \times K$ can be decomposed into an all-orthogonal *core array* $\mathbf{T}_c$ of size $R_1 \times R_2 \times R_3$ by orthogonal changes of coordinates, where R_p denote the mode-p ranks, earlier defined in this section. One can write:

$$\mathbf{T} = (\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \mathbf{Q}^{(3)}) \cdot \mathbf{T}_c \;\Leftrightarrow\; \mathbf{T}_c = (\mathbf{Q}^{(1)\mathrm{T}}, \mathbf{Q}^{(2)\mathrm{T}}, \mathbf{Q}^{(3)\mathrm{T}}) \cdot \mathbf{T} \tag{5}$$

In order to compute the three orthogonal matrices $\mathbf{Q}^{(\ell)}$, one just computes the SVD of the three unfolding matrices of $\mathbf{T}$. The core tensor $\mathbf{T}_c$ is then obtained by the rightmost equation in (5). Now the advantage of the SVD is that it allows to compute the best low-rank approximate of a matrix, by just truncating the

decomposition (setting singular values to zero, below some chosen threshold); this is the well-known Eckart-Young theorem. So one could think that truncating the HOSVD in every mode would yield a best lower multilinear rank approximate. It turns out that this is not true, but generally good enough in practice.

By truncating the HOSVD to some lower multilinear rank, say (r_1, r_2, r_3) smaller than (R_1, R_2, R_3), one can reduce the tensor rank. But this is not under control, since tensor rank may be larger than all mode-ranks: $F \geq r_i$. In other words, HOSVD truncation may yield a sub-generic tensor rank, but the exact tensor rank will not be known. To conclude, and contrary to what may be found in some papers in the literature, there does not exist a generalization of the Eckart-Young theorem to tensors.

3 Algorithms

Various iterative numerical algorithms have been investigated in the literature, and aim at minimizing the fitting error (3). One can mention in particular the Conjugate Gradient [46], Gauss-Newton and Levenberg-Marquardt [69] algorithms.

Notations. It is convenient to define the vec operator that maps a matrix to a vector: $\mathbf{m} = \text{vec}(\mathbf{M})$, meaning that $m_{(i-1)J+j} = M_{ij}$. Also denote by $\boxdot$ the Hadamard entry-wise matrix product between matrices of the same size. With this notation, the gradient of Υ with respect to $\text{vec}(\mathbf{A})$ is given by:

$$\mathbf{g_A} = [\mathbf{I}_A \otimes (\mathbf{C}^{\mathrm{T}}\mathbf{C} \boxdot \mathbf{B}^{\mathrm{T}}\mathbf{B})]\text{vec}\mathbf{A}^{\mathrm{T}} - [\mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B})]^{\mathrm{T}}\text{vec}\mathbf{T}_{JK\times I}$$

and other gradients are deduced by circular permutation:

$$\begin{aligned}\mathbf{g_B} &= [\mathbf{I}_B \otimes (\mathbf{A}^{\mathrm{T}}\mathbf{A} \boxdot \mathbf{C}^{\mathrm{T}}\mathbf{C})]\text{vec}\mathbf{B}^{\mathrm{T}} - [\mathbf{I}_B \otimes (\mathbf{A} \odot \mathbf{C})]^{\mathrm{T}}\text{vec}\mathbf{T}_{KI\times J}\\ \mathbf{g_C} &= [\mathbf{I}_C \otimes (\mathbf{B}^{\mathrm{T}}\mathbf{B} \boxdot \mathbf{A}^{\mathrm{T}}\mathbf{A})]\text{vec}\mathbf{C}^{\mathrm{T}} - [\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})]^{\mathrm{T}}\text{vec}\mathbf{T}_{IJ\times K}\end{aligned}$$

Also define

$$\mathbf{p} = \begin{bmatrix} \text{vec}\mathbf{A}^{\mathrm{T}} \\ \text{vec}\mathbf{B}^{\mathrm{T}} \\ \text{vec}\mathbf{C}^{\mathrm{T}} \end{bmatrix}, \text{ and } \mathbf{g} = \begin{bmatrix} \mathbf{g_A} \\ \mathbf{g_B} \\ \mathbf{g_C} \end{bmatrix}$$

Lastly, we have the compact forms below for Jacobian matrices:

$$\begin{aligned}\mathbf{J_A} &= \mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B})\\ \mathbf{J_B} &= \mathbf{\Pi}_1\,[\mathbf{I}_B \otimes (\mathbf{A} \odot \mathbf{C})]\\ \mathbf{J_C} &= \mathbf{\Pi}_2\,[\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})]\end{aligned}$$

where $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are permutation matrices that put the entries in the right order.

The goal is to minimize the Frobenius norm of tensor $\mathbf{T} - (\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{I}$, which may be written in a non unique way as a matrix norm:

$$\Upsilon(\mathbf{p}) \stackrel{\text{def}}{=} ||\mathbf{T}_{JK\times I} - (\mathbf{B} \odot \mathbf{C})\mathbf{A}^{\mathrm{T}}||^2 \tag{6}$$

3.1 Gradient descent

The gradient descent is the simplest algorithm. The iteration is given by: $\mathbf{p}(k+1) = \mathbf{p}(k) - \mu(k)\,\mathbf{g}(k)$, where $\mu(k)$ is a stepsize that is varied as convergence progresses. Even with a good strategy of variation of $\mu(k)$, this algorithm has often very poor convergence properties. But they can be dramatically improved with the help of ELS, as will be demonstrated in Section 5.4.

3.2 Levenberg-Marquardt (LM)

Taking into account the second derivative (Hessian) allows faster local convergence. An approximation using only first order derivatives is given by the iteration below, known as Levenberg-Marquardt:

$$\mathbf{p}(k+1) = \mathbf{p}(k) - [\mathbf{J}(k)^{\mathrm{T}}\mathbf{J}(k) + \lambda(k)^2\,\mathbf{I}]^{-1}\,\mathbf{g}(k)$$

where $\mathbf{J}(k)$ denotes the Jacobian matrix $[\mathbf{J}_A, \mathbf{J}_B, \mathbf{J}_C]$ at iteration k and $\lambda(k)^2$ a positive regularization parameter. There exist several ways of calculating $\lambda(k)^2$, and this has an important influence on convergence. In the sequel, we have adopted the update described in [41]. Updates of $\mathbf{p}$ and λ^2 are controlled by the gain ratio γ:

$$\gamma \stackrel{\text{def}}{=} \Upsilon(k) - \Upsilon(k+1) \cdot (\hat{\Upsilon}(\mathbf{0}) - \hat{\Upsilon}(\Delta\mathbf{p}(k)))^{-1}$$

where $\hat{\Upsilon}(\Delta\mathbf{p}(k))$ is a second order approximation of $\Upsilon(\mathbf{p}(k) + \Delta\mathbf{p}(k))$. More precisely, each new iteration of the algorithm follows this simple scheme:

1. Find the new direction $\Delta\mathbf{p}(k) = -[\mathbf{J}(k)^{\mathrm{H}}\mathbf{J}(k) + \lambda(k)^2\,\mathbf{I}]^{-1}\,\mathbf{g}(k)$,
2. Compute $\mathbf{p}(k+1)$ and deduce $\Upsilon(k+1)$
3. Compute $\hat{\Upsilon}(\Delta\mathbf{p}(k))$
4. Compute γ.
5. If $\gamma \geq 0$, then $\mathbf{p}(k+1)$ is accepted, $\lambda(k+1)^2 = \lambda(k)^2 * max\left(\frac{1}{3}, 1-(2\gamma-1)^3\right)$ and $\nu = 2$. Otherwise, $\mathbf{p}(k+1)$ is rejected, $\lambda(k+1)^2 = \nu\lambda(k)^2$ and $\nu \leftarrow 2\nu$.

3.3 Alternating Least Squares (ALS)

The principle of the ALS algorithm is quite simple. We recall here the various versions that exist, since there are indeed several. For fixed $\mathbf{B}$ and $\mathbf{C}$, this is a quadratic form in $\mathbf{A}$, so that there is a closed form solution for $\mathbf{A}$. Similarly for $\mathbf{B}$ and $\mathbf{C}$. Consquently, we can write:

$$\begin{aligned}
\mathbf{A}^{\mathrm{T}} &= (\mathbf{B} \odot \mathbf{C})^{\dagger}\,\mathbf{T}_{JK\times I} \stackrel{\text{def}}{=} f_A(\mathbf{B},\mathbf{C}) \\
\mathbf{B}^{\mathrm{T}} &= (\mathbf{C} \odot \mathbf{A})^{\dagger}\,\mathbf{T}_{KI\times J} \stackrel{\text{def}}{=} f_B(\mathbf{C},\mathbf{A}) \\
\mathbf{C}^{\mathrm{T}} &= (\mathbf{A} \odot \mathbf{B})^{\dagger}\,\mathbf{T}_{IJ\times K} \stackrel{\text{def}}{=} f_C(\mathbf{A},\mathbf{B})
\end{aligned} \tag{7}$$

where $\mathbf{M}^{\dagger}$ denotes the pseudo-inverse of $\mathbf{M}$. When $\mathbf{M}$ is full column rank, we have $\mathbf{M}^{\dagger} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}$, which is the Least Squares (LS) solution of the overdetermined linear system. But it may happen that $\mathbf{M}$ is not full column rank. In

that case, not only there are more equations than unknowns, so that the system is impossible, but some equations are linked to each other so that there are also infinitely many solutions for some unknowns. The pseudo-inverse then gives the solution of minimal norm for those variables (i.e. null component in the kernel). In some cases, this may not be what we want: we may want to obtain full rank estimates of matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$.

Another well-known approach to singular systems is that of *regularization*. It consists of adding a small diagonal matrix $\mathbf{D}$, which allows to restore invertibility. Matrix $\mathbf{D}$ is often taken to be proportional to the identity matrix $\mathbf{D} = \eta\mathbf{I}$, η being a small real number. This corresponds to define another inverse of the form $\mathbf{M}^\dagger = (\mathbf{M}^{\mathrm{T}}\mathbf{M}+\eta\mathbf{D})^{-1}\mathbf{M}^{\mathrm{T}}$. With this definition, we would have three other relations, with obvious notation:

$$\mathbf{A}^{\mathrm{T}} = \bar{f}_A(\mathbf{B}, \mathbf{C}; \mathbf{D}), \quad \mathbf{B}^{\mathrm{T}} = \bar{f}_B(\mathbf{C}, \mathbf{A}; \mathbf{D}), \quad \mathbf{C}^{\mathrm{T}} = \bar{f}_C(\mathbf{A}, \mathbf{B}; \mathbf{D}) \tag{8}$$

Thus from (7) one can compute the updates below

ALS1. $\mathbf{B}_{n+1}=f_B(\mathbf{C}_n, \mathbf{A}_n)$, $\mathbf{C}_{n+1}=f_C(\mathbf{A}_n, \mathbf{B}_n)$, $\mathbf{A}_{n+1}=f_A(\mathbf{B}_n, \mathbf{C}_n)$

ALS2. $\mathbf{B}_{n+1}=f_B(\mathbf{C}_n, \mathbf{A}_n)$, $\mathbf{C}_{n+1}=f_C(\mathbf{A}_n, \mathbf{B}_{n+1})$, $\mathbf{A}_{n+1}=f_A(\mathbf{B}_{n+1}, \mathbf{C}_{n+1})$,

ALS3. $\mathbf{B}_{n+1}=\bar{f}_B(\mathbf{C}_n, \mathbf{A}_n; \eta\mathbf{I})$, $\mathbf{C}_{n+1}=\bar{f}_C(\mathbf{A}_n, \mathbf{B}_{n+1}; \eta\mathbf{I})$, $\mathbf{A}_{n+1}=\bar{f}_A(\mathbf{B}_{n+1}, \mathbf{C}_{n+1}; \eta\mathbf{I})$

Algorithm ALS1 is never utilized in practice except to compute a direction of search at a given point, and it is preferred to replace the values of the loading matrices by the most recently computed, as shown in ALS2. Algorithm ALS3 is the regularized version of ALS2. If the tensor $\mathbf{T}$ to be decomposed is symmetric, there is only one matrix $\mathbf{A}$ to find. At least two implementations can be thought of (more exist for higher order tensors):

ALS4. Soft constrained: $\mathbf{A}_{n+1} = f_A(\mathbf{A}_n, \mathbf{A}_{n-1})$

ALS5. Hard constrained: $\mathbf{A}_{n+1} = f_A(\mathbf{A}_n, \mathbf{A}_n)$

There does not exist any proof of convergence of the ALS algorithms, which may not converge, or converge to a local minimum. Their extensive use is thus rather unexplainable.

3.4 Enhanced Line Search (ELS)

The ELS enhancement is applicable to any iterative algorithm, provided the optimization criterion is a polynomial or a rational function. Let $\Delta\mathbf{p}(k)$ be the direction obtained by an iterative algorithm, that is, $\mathbf{p}(k+1) = \mathbf{p}(k) + \mu\,\Delta\mathbf{p}(k)$. ELS ignores the value of μ that the iterative algorithm has computed, and searches for the best stepsize μ that corresponds to the *global minimum* of (6):

$$\Upsilon(\mathbf{p}(k) + \mu\,\Delta\mathbf{p}(k)).$$

Stationary values of μ are given by the roots of a polynomial in μ, since Υ is itself a polynomial. It is then easy to find the global minimum once all the roots have

been computed, by just selecting the root yielding the smallest value of Υ [48]. Note that the polynomial is of degree 5 for third order tensors. The calculation of the ELS stepsize can be executed every every p iteration, $p \geq 1$, in order to spare some computations (cf. Section 4).

4 Numerical complexity

The goal of this section is to give an idea of the cost per iteration of various algorithms. In fact, to compare iterative algorithms only on the basis on the number of iterations they require is not very meaningful, because the costs per iteration are very different.

Nevertheless, only orders of magnitudes of computational complexities are given. In fact, matrices that are the Kronecker product of others are structured. Yet, the complexity of the product between matrices, the solution of linear systems or other matrix operations, can be decreased by taking into account their structure. And such structures are not fully taken advantage of in this section.

Singular Value Decomposition (SVD). Let the so-called reduced SVD of a $m \times n$ matrix $\mathbf{M}$ of rank r, $m \geq n \geq r$, be written as:

$$\mathbf{M} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^{\mathrm{T}} = \mathbf{U}_r\,\mathbf{\Sigma}_r\,{\mathbf{V}_r}^{\mathrm{T}}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, $\mathbf{\Sigma}$ is $m \times n$ diagonal, $\mathbf{U}_r$ is a $m \times r$ submatrix of $\mathbf{U}$, ${\mathbf{U}_r}^{\mathrm{T}}\mathbf{U}_r = \mathbf{I}_r$, $\mathbf{\Sigma}_r$ is $r \times r$ diagonal and $\mathbf{V}_r$ is $n \times r$, ${\mathbf{V}_r}^{\mathrm{T}}\mathbf{V}_r = \mathbf{I}_r$. The calculation of the diagonal matrix $\mathbf{\Sigma}$ needs $2mn^2 - 2n^3/3$ multiplications, if matrices $\mathbf{U}$ and $\mathbf{V}$ are not explicitly formed, but kept in the form of a product of Householder symmetries and Givens rotations [27]. Calculating explicitly matrices $\mathbf{U}_r$ and $\mathbf{V}_r$ requires additional $5mr^2 - r^3/3$ and $5nr^2 - r^3/3$ multiplications, respectively.

For instance if $r = n$, the total complexity for computing $\mathbf{\Sigma}$, $\mathbf{U}_n$ and $\mathbf{V}$ is of order $7mn^2 + 11n^3/3$. If $m \gg n$, this complexity can be decreased to $O(3mn^2)$ by resorting to Chan's algorithm [11] [15]. Even if in the present context this improvement could be included, it has not yet been implemented.

High-Order SVD (HOSVD). Computing the HOSVD of a $I \times J \times K$ array requires three SVD of dimensions $I \times JK$, $J \times KI$ and $K \times IJ$ respectively, where only the left singular matrix is required explicitly. Assume that we want to compute in mode i the reduced SVD of rank R_i. Then the overall computational complexity for the three SVDs is

$$2IJK(I+J+K)+5(R_1^2JK+IR_2^2K+IJR_3^2)-2(I^3+J^3+K^3)/3-(R_1^3+R_2^3+R_3^3)/3$$

In addition, if the core tensor needs to be computed, singular matrices need to be contracted on the original tensor. The complexity depends on the order in which the contractions are executed. If we assume $I \geq J \geq K > R_i$, then it is better to contract the third mode first and the first mode last. This computation

requires $IR_1JK + R_1JR_2K + R_1R_2KR_3$ additional multiplications. If $R_i \approx F \ll min(I, J, K)$ one can simplify the overall complexity as:

$$2IJK(I + J + K) - 2(I^3 + J^3 + K^3)/3 + IJKF$$

Alternating Least Square (ALS). Let's look at the complexity of one iteration for the first mode, and consider the linear system below, to be solved for $\mathbf{A}$ in the LS sense:

$$\mathbf{M}\,\mathbf{A}^{\mathrm{T}} = \mathbf{T}$$

where the dimensions of $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{T}$ are $m \times r$, $r \times q$ and $m \times q$, respectively. First, matrix $\mathbf{M} = \mathbf{B} \odot \mathbf{C}$ needs to be calculated, which requires one Khatri-Rao product, that is, JKF multiplications. Then the SVD of matrix $\mathbf{M}$ of size $F \times JK$ needs $7JKF^2 + 11\,F^3/3$ multiplications, according to the previous paragraph. Last, the product $\mathbf{V}_F\,\mathbf{\Sigma}_F^{-1}\mathbf{U}_F{}^{\mathrm{T}}\,\mathbf{T}$ represents $FJKI + FI + F^2I$ multiplications.

The cumulated total of one iteration of ALS2 for the three modes is hence $(JK + KI + IJ)(7F^2 + F) + 3F\,IJK + (I + J + K)(F^2 + F) + 11\,F^3$, where the two last terms are often negligible. The complexity of ALS3 is not detailed here, but could be calculated following the same lines as LM in the next section.

Levenberg-Marquardt (LM). One iteration consists of calculating $\Delta\mathbf{p} \stackrel{\text{def}}{=} [\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I}]^{-1}\mathbf{g}$ where $\mathbf{J}$ is of size $m \times n$, $m \stackrel{\text{def}}{=} IJK$, $n \stackrel{\text{def}}{=} (I + J + K)F$. Of course, it is not suitable to compute the product $\mathbf{J}^{\mathrm{T}}\mathbf{J}$, nor the inverse matrix. Instead, one shall work with the $(m + n) \times n$ matrix $\mathbf{S} \stackrel{\text{def}}{=} [\mathbf{J},\, \lambda\mathbf{I}]^{\mathrm{T}}$, and consider the linear system $\mathbf{S}^{\mathrm{T}}\mathbf{S}\,\Delta\mathbf{p} = \mathbf{g}$.

As in ALS, the first step is to compute the Khatri-Rao products. Then each of the three gradients requires $(I + J + K)F^2 + IJKF$ multiplications. Next, the orthogonalization (QR factorization) of matrix $\mathbf{S}$ requires mn^2 multiplications, since making the orthogonal matrix explicit is not necessary. Finally, $\mathbf{g}$ is obtained by solving two triangular systems, each requiring an order of $n^2/2$ multiplications. Thus the overall complexity of one iteration of the algorithm is dominated by the orthogonalization step, which costs: $IJK(I + J + K)^2F^2$ multiplications.

Enhanced Line Search (ELS). Given a direction of search, the complexity of one ELS iteration is dominated by the calculation of the 6 coefficients of the polynomial in μ to be rooted. This complexity may be shown to be of order $(8F + 10)IJK$ [49]. But the constant coefficient is not needed here so that we may assume that $(8F + 9)IJK$ multiplications are required.

It is interesting to compare the complexity of ELS with those of ALS or LM. This gives simple rules of thumb. If the rank is smaller than the bound below, then the additional complexity of ELS is negligible compared to ALS2 [49]:

$$\left(\frac{1}{I} + \frac{1}{J} + \frac{1}{K}\right) F \gg 1 \tag{9}$$

Similarly, if $8 \ll (I + J + K)F$, ELS is negligible compared to LM, which in practice always happens. These rules are extremely simplified, but they still show that the

larger the rank F, the more negligible ELS; they also have the advantage to give quite reliable orders of magnitude when dimensions are large enough (i.e. at least 10).

5 Computer results

5.1 Performance measure

Performance evaluation is carried out in terms of error between the calculated and actual factor matrices. The problem would be easy if there were not an indetermination up to scale and permutation. In fact, it is well known that

$$(\mathbf{A}, \mathbf{B}, \mathbf{C}) \cdot \mathbf{I} = (\mathbf{A}\mathbf{\Lambda}_A\mathbf{P}, \mathbf{B}\mathbf{\Lambda}_B\mathbf{P}, \mathbf{C}\mathbf{\Lambda}_C\mathbf{P}) \cdot \mathbf{I}$$

For any $F \times F$ invertible diagonal matrices $\mathbf{\Lambda}_A$, $\mathbf{\Lambda}_B$, $\mathbf{\Lambda}_C$ satisfying $\mathbf{\Lambda}_A\mathbf{\Lambda}_B\mathbf{\Lambda}_C = \mathbf{I}$ and for any $F \times F$ permutation matrix $\mathbf{P}$. It is thus desirable that the performance measure be invariant under the action of these four matrices. First, it is convenient to define a distance between two column vectors $\mathbf{u}$ and $\mathbf{v}$, which is invariant to scale. This is given by the expression below:

$$\delta(\mathbf{u}, \mathbf{v}) = ||\mathbf{u} - \frac{\mathbf{v}^{\mathrm{T}}\mathbf{u}}{\mathbf{v}^{\mathrm{T}}\mathbf{v}}\,\mathbf{v}|| \tag{10}$$

Now we describe two ways of computing a scale-permutation invariant distance between two matrices $\mathbf{M}$ and $\hat{\mathbf{M}}$.

Optimal algorithm. For every $F \times F$ permutation $\mathbf{P}$, one calculates

$$\Delta(\mathbf{P}) = \sum_{k=1}^{K} \delta(\mathbf{u}(k), \mathbf{v}(k))$$

where $\mathbf{u}(k)$ and $\mathbf{v}(k)$ denote the kth column of matrices

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{bmatrix} \mathbf{P} \quad \text{et} \quad \hat{\mathbf{M}} \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\mathbf{A}} \\ \hat{\mathbf{B}} \\ \hat{\mathbf{C}} \end{bmatrix}$$

respectively. The error retained is the minimal distance obtained over all $F!$ possible permutations. Of course, this optimal algorithm is usable only for moderate values of F, say smaller than 9. For larger ranks, one must resort to a suboptimal algorithm, as the one described below.

Greedy algorithm. The goal is to avoid to describe all permutations. For doing this, one selects an hopefully relevant subset of all possible permutations. One first selects the column $\hat{\mathbf{m}}(j_1)$ of $\hat{\mathbf{M}}$ having the largest norm and one calculates its distance $\delta(\cdot)$ with the closest column $\mathbf{m}(k_1)$ of $\mathbf{M}$:

$$\Delta_1 = \min_{k_1} \delta(\mathbf{m}(k_1), \hat{\mathbf{m}}(j_1))$$

Then one deletes from matrices $\mathbf{M}$ and $\hat{\mathbf{M}}$ the columns k_1 and j_1 previously selected, and one keeps going with the remaining columns. At iteration n, one has to compute $F-n+1$ distances $\delta(\mathbf{m}(k_n), \hat{\mathbf{m}}(j_n))$. After F iterations, one eventually obtains a suboptimal distance Δ, defined either as

$$\Delta = \sum_{k=1}^{K} \Delta_k \text{ or as } \Delta^2 = \sum_{k=1}^{K} \Delta_k^2.$$

If $\mathbf{M}$ is $L \times F$, this greedy calculation of the error Δ requires an order of $3F^2L/2$ multiplications.

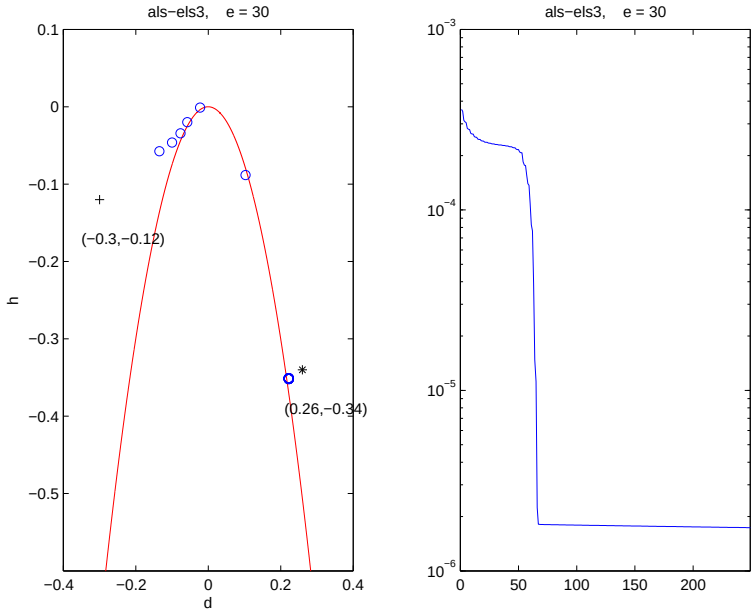

Figure 1: First case treated by ALS with ELS at every 3 iterations. Left: symbols '+', 'o' and '$*$' denote respectively the initial value, the successive iterations, and the global minimum, in the (d, h) plane. Right: value of the error as a function of iterations.

5.2 A $2 \times 2 \times 2$ tensor of rank 2

Paatero gave in his paper [47] a nice example of a $2 \times 2 \times 2$ tensor, fully parameterizable, hence permitting reproducible experiments. We use here the same parameterization. Its form is as follows:

$$\mathbf{T} = \left(\begin{array}{cc|cc} 0 & 1 & e & 0 \\ 1 & d & 0 & h \end{array} \right)$$

The discriminant defined in Section 2.2 is in the present case $\Delta = 4\,h + d^2 e$. As in [47], we represent the decompositions computed in the hyperplane $e = 30$. The

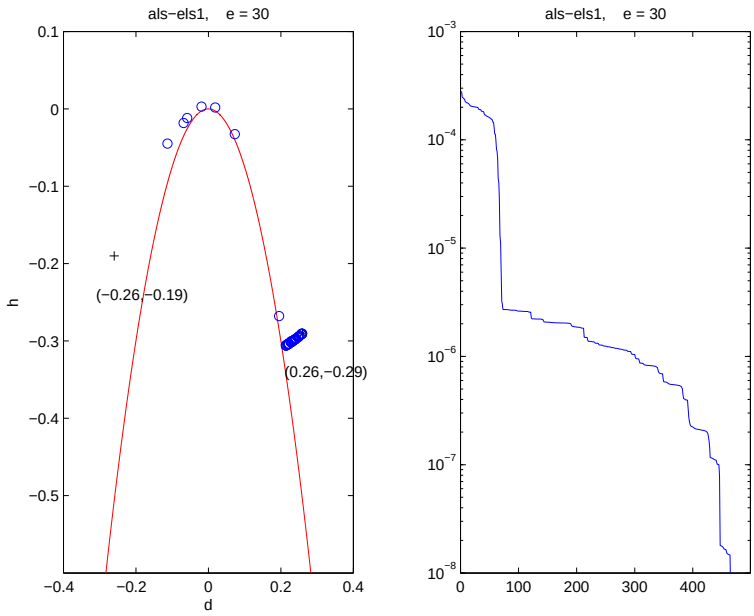

Figure 2: Second case treated by ALS with ELS at every iteration. Left: symbols '+', 'o' and '$*$' denote respectively the initial value, the successive iterations, and the global minimum, in the (d, h) plane. Right: value of the error as a function of iterations.

tensor to be decomposed are chosen to have a discriminant of $\Delta > 0$, so that they have a rank 2, but they lie close to the variety $\Delta = 0$, represented by a parabola in the subsequent figures.

Case 1. First consider the tensor defined by $(e, d, h) = (30, 0.26, -0.34)$, and start with the initial value $(e, d, h) = (30, -0.3, -0.12)$. In that case, neither the Gradient nor even LM converge towards the solution, and are stuck on the left side of the parabola. The ALS algorithm does not seem to converge, or perhaps does after a very large number of iterations. With the ELS enhancement executed every 3 iterations, ALS2 converges to a neighborhood of the solution (error of 10^{-6}) within only 60 iterations, as shown in Figure 1. Then, the algorithm is slowed down for a long time before converging eventually to the solution; the beginning of this plateau may be seen in the right graph of Figure 1. Note that it *does not* correspond to a plateau of the objective function. In this particular example, the gain brought by ELS is very significant at the beginning. On the other hand, no gain is brought at the end of the trajectory towards the goal, where ELS cannot improve ALS2 because the directions given by ALS2 are repeatedly bad.

Case 2. Consider now a second (easier) case given by $(e, d, h) = (30, 0.26, -0.29)$, with the initial value $(e, d, h) = (30, -0.26, -0.19)$. This time, the LM algorithm is

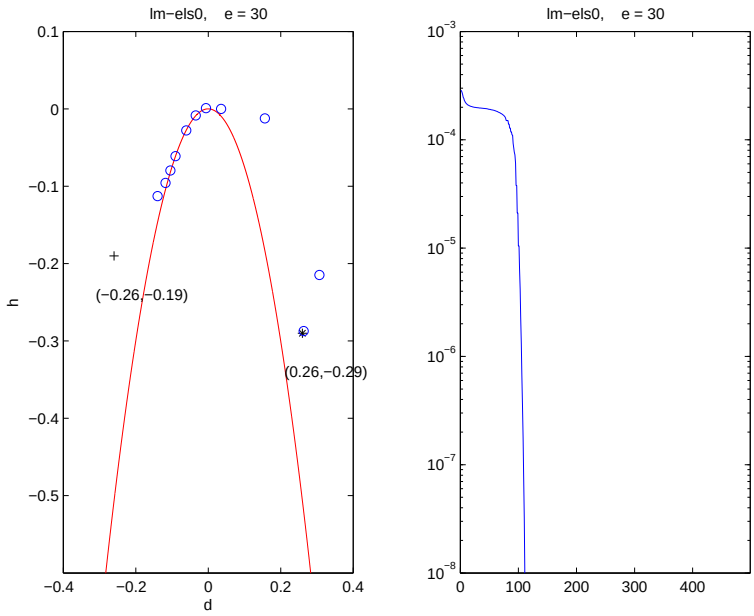

Figure 3: Second case treated by LM. Left: symbols '+', 'o' and '∗' denote respectively the initial value, the successive iterations, and the global minimum, in the (d, h) plane. Right: value of the error as a function of iterations.

the most efficient and converges within 100 iterations, as shown in Figure 3. The fastest ALS2 algorithm is this time the one using the ELS enhancement at every iteration, and converges within 400 iterations (cf. Figure 2).

In the second case, ALS2 with the ELS enhancement could avoid being trapped in a kind of "cycling" about the goal, whereas it could not do it in the first example. This is a limitation of ELS: if the directions provided by the algorithm are bad, ELS cannot improve much on the convergence. The same has been be observed with case 1: ELS does not improve LM since directions chosen by LM are bad (LM is not represented for case 1). The other observation one can make is that executing ELS at every iteration is not always better than at every p iterations, $p > 1$: this depends on the trajectory, and there is unfortunately no general rule. This will be also confirmed by the next computer experiments.

5.3 Random $30 \times 30 \times 30$ tensor of rank 20

In the previous section, the computational complexity per iteration was not an issue, because it is so small. Now we shall turn our attention to this issue, with a particular focus on the effect of *dimension reduction* by HOSVD truncation. As in the $2 \times 2 \times 2$ case, the initial value has an influence on convergence; one initial point is chosen randomly, the same for all algorithms to be compared.

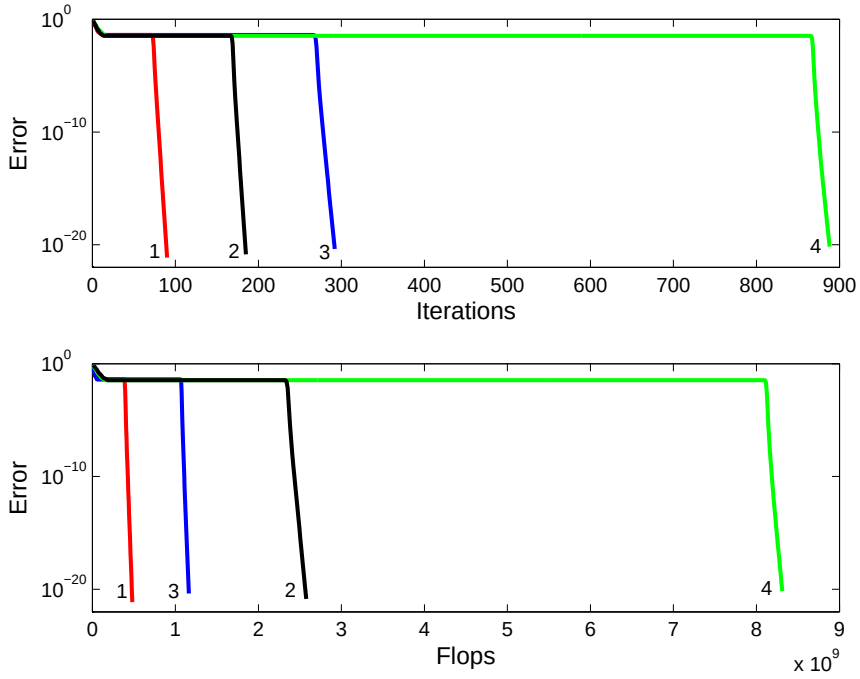

Figure 4: Reconstruction error of a $30 \times 30 \times 30$ random tensor of rank 20 as a function of the number of iterations (top) or the number of multiplications (bottom), and for various algorithms. 1: ALS + HOSVD + ELS, 2: ALS + ELS, 3: ALS + HOSVD, 4: ALS.

The tensor considered here is generated by three 30×20 factor matrices whose entries have been independently randomly drawn according to a zero-mean unit variance Gaussian distribution. In the sequel, we shall denote such a random drawing as $\mathbf{A} \sim \mathcal{N}_{30\times 20}(0, 1)$, in short. The correlation coefficient (absolute value of the cosine) between any pair of columns of these matrices was always smaller than 0.8, which is a way to check out that the goal to reach does not lie itself in a strong bottleneck.

The ALS algorithm is run either without ELS, or with ELS enhancement at every iteration. Figure 4 shows the clear improvement brought by ELS without dimension reduction (the computational complexity is 3 times smaller). If the dimensions are reduced to 20 by HOSVD truncation, we also observe a similar improvement: ALS runs better, but the additional complexity of ELS also becomes smaller (the rank is of same order as dimensions). Also notice by comparing top and bottom of Figure 4 that curves 2 and 3 do not appear in the same order: the dimension reduction gives the advantage to ALS in dimension 20 without ELS, compared to ALS with ELS in dimension 30. The advantage was reversed in the top part of the figure.

A reduction of the dimensions to a value smaller than 20, while keeping the rank equal to 20, is theoretically possible. However, we know that truncating the HOSVD is not the optimal way of finding the best $(20, 20, 20)$ multilinear rank approximation. In practice, we have observed that truncating the HOSVD yields reconstruction errors if the chosen dimension is smaller than the rank (e.g. 19 in the present case or smaller).

5.4 Random $5 \times 5 \times 5$ tensor of rank 4

The goal here is to show the interest of ELS, not only for ALS, but also for the Gradient algorithm. In fact in simple cases, the Gradient algorithm may be attractive, if it is assisted by the ELS enhancement.

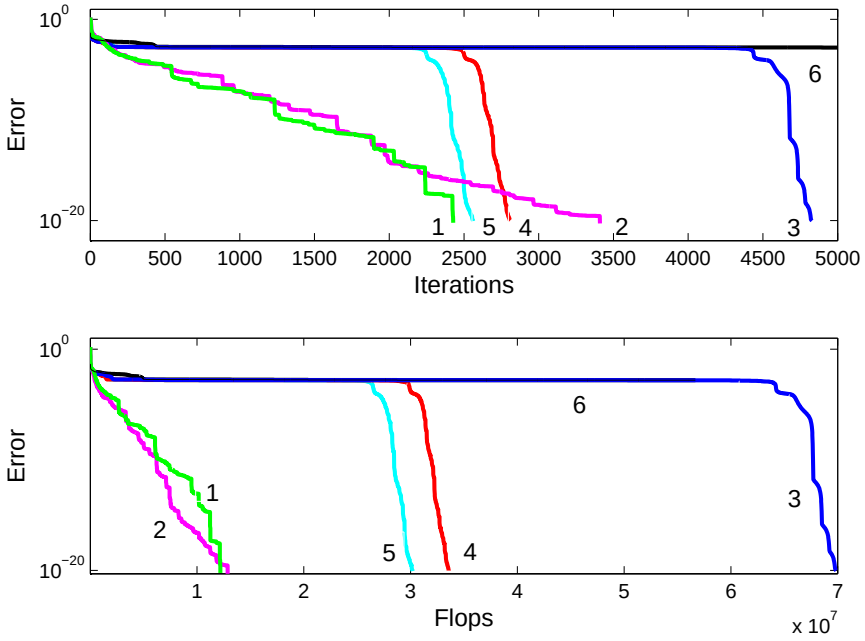

Figure 5: Reconstruction error of a $5 \times 5 \times 5$ tensor of rank 4 as a function of the number of iterations (top) or the number of multiplications (bottom), and for various algorithms. 1: Gradient + ELS with period 2, 2: Gradient + ELS with period 5, 3: ALS + ELS with period 1 (every iteration), 4: ALS + ELS with period 5, 5: ALS + ELS with period 7, 6: ALS.

We consider the easy case of a $5 \times 5 \times 5$ tensor of rank 4. The latter has been built from three 5×4 Gaussian matrices drawn randomly as before, each $\mathcal{N}_{5\times4}(0, 1)$. Results are reported in figure 5, and show that ELS performs similarly if executed every 2 iterations, or every 5. Results of the Gradient without ELS

enhancement are not shown, but are as poor as ALS. Convergence of the Gradient with ELS is reached earlier than ALS with ELS, but this is not always the case in general. The Gradient with ELS appears to be especially attractive in this example if computational complexity is considered.

5.5 Double bottleneck in a $30 \times 30 \times 30$ tensor

Now more difficult cases are addressed, and we consider a tensor with two double bottlenecks (with R. Harshman terminology). The tensor is built from three Gaussian matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ of size 30×4, and we have created two bottlenecks in the two first modes as follows. The first column $\mathbf{a}(1)$ of $\mathbf{A}$ is drawn randomly as $\mathcal{N}_{30}(0,1)$. Then the second column is set to be $\mathbf{a}(2) = \mathbf{a}(1) + 0.5\,\mathbf{v}(2)$, where $\mathbf{v}(2) \sim \mathcal{N}_{30}(0,1)$. The third and fourth columns of $\mathbf{A}$ are generated in the same manner, that is, $\mathbf{a}(4) = \mathbf{a}(3) + 0.5\,\mathbf{v}(4)$, where $\mathbf{a}(3) \sim \mathcal{N}_{30}(0,1)$ and $\mathbf{v}(4) \sim \mathcal{N}_{30}(0,1)$. Matrix $\mathbf{B}$ is independently generated exactly in the same manner. Vectors belonging to a bottleneck generated this way have a correlation coefficient of 0.89. On the other hand, matrix $\mathbf{C}$ is independently $\mathcal{N}_{30\times 4}(0,1)$, and its columns are not close to dependent.

Results reported in Figure 6 show that even when the rank is moderate compared to dimensions, ELS brings improvements to ALS, despite the fact that its additional complexity is not negligible according to Equation (9). Surprisingly, a dimension reduction down to the rank value does not change this conclusion. As previously, ELS with period 1 cannot be claimed to be always better than ELS with period $p > 1$.

ALS with ELS enhancement at every 5 iterations (solid line in Figure 7) benefits from an important computational complexity improvement thanks to the dimension reduction down to 4: its convergences needs 6.10^6 multiplications instead of 8.10^8. The improvement is even more impressive for ALS with ELS enhancement at every iteration (dashed line).

5.6 Swamp in a $30 \times 30 \times 30$ tensor of rank 4

In this last example, we have generated a tensor with two triple bottlenecks, that is, a tensor that lies in a swamp. Matrices $\mathbf{A}$ and $\mathbf{B}$ are generated as in Section 5.5, but matrix $\mathbf{C}$ is now also generated the same way.

This tensor is difficult to decompose, and ALS does not converge after 5000 iterations. With the help of ELS every 5 iterations, the algorithm converges with about a thousand of iterations. This was unexpected since ELS is not always able to escape from swamps. The reduction of dimensions to 4 allows to decrease the computational complexity by a factor of 50, even if the number of iterations has slightly increased.

6 Conclusion

Our paper was three-fold. Firstly we have surveyed theoretical results, and pointed out which are still conjectures. Secondly we have described several numerical algorithms and given their approximate complexities for large dimensions. Thirdly, we have compared these algorithms in situations that Richard Harshman had classified, and in particular *multiple bottlenecks* and *swamps*. Several conclusions can be drawn.

First, one can claim that dimension reduction by HOSVD truncation is always improving the results, both in terms of speed of convergence and computational complexity, provided the new dimensions are not smaller than the rank of the tensor to be decomposed.

Second, ELS has been shown to be useful to improve ALS2 or the Gradient descent, either as an help to escape from a local minimum or to speed up convergence. ELS has been often efficient in simple or multiple bottlenecks, and even in swamps. Its help ranges from excellent to moderate. On the other hand, ELS has never been able to help LM when it was stuck.

Third, ALS2 can still progress very slowly in some cases where the objective function is not flat at all. This is a particularity of ALS, and we suspect that it is "cycling". In such a case, ELS does not help because the directions defined by ALS2 are not good. One possibility would be to call for ALS1 to give other directions; this is currently being investigated.

Last, no solution has been proposed in cases of CP-degeneracies. In the later case, the problem is ill-posed, and the solution should be searched for by changing the modeling, i.e., either by increasing the rank or by imposing additional constraints that will avoid factor cancellations. Future works will be focussed on the two latter conclusions.

Acknowledgment. This work has been partly supported by contract ANR-06-BLAN-0074 "Decotes".

References

[1] H. ABO, G. OTTAVIANI, and C. PETERSON. *Induction for secant varieties of Segre varieties*, August 2006. arXiv:math/0607191.

[2] J. ALEXANDER and A. HIRSCHOWITZ. La methode d'Horace eclatee: application a l'interpolation en degre quatre. *Invent. Math.*, 107(3):585–602, 1992.

[3] D. BINI. Border rank of a $p \times q \times 2$ tensor and the optimal approximation of a pair of bilinear forms. In *Proc. 7th Col. Automata, Languages and Programming*, pages 98–108, London, UK, 1980. Springer-Verlag. Lecture Notes in Comput. Sci., vol. 85.

[4] D. BINI. Border rank of $m \times n \times (mn - q)$ tensors. *Linear Algebra Appl.*, 79:45–51, 1986.

[5] M. C. BRAMBILLA and G. OTTAVIANI. On the Alexander-Hirschowitz theorem. *Jour. Pure Applied Algebra*, 212:1229–1251, 2008.

[6] R. BRO. Parafac, tutorial and applications. *Chemom. Intel. Lab. Syst.*, 38:149–171, 1997.

[7] R. BRO and C. A. ANDERSSON. Improving the speed of multiway algorithms. part ii: Compression. *Chemometrics and Intelligent Laboratory Systems*, 42(1-2):105–113, 1998.

[8] P. BÜRGISSER, M. CLAUSEN, and M. A. SHOKROLLAHI. *Algebraic Complexity Theory*, volume 315. Springer, 1997.

[9] J. D. CARROLL and J. J. CHANG. Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition. *Psychometrika*, 35(3):283–319, September 1970.

[10] A. CAYLEY. On the theory of linear transformation. *Cambridge Math. J.*, 4:193–209, 1845.

[11] T. F. CHAN. An improved algorithm for computing the singular value decomposition. *ACM Trans. on Math. Soft.*, 8(1):72–83, March 1982.

[12] L. CHIANTINI and C. CILIBERTO. On the concept of $k-$sectant order of a variety. *J. London Math. Soc.*, 73(2), 2006.

[13] P. COMON. Tensor decompositions. In J. G. McWhirter and I. K. Proudler, editors, *Mathematics in Signal Processing V*, pages 1–24. Clarendon Press, Oxford, UK, 2002.

[14] P. COMON, G. GOLUB, L-H. LIM, and B. MOURRAIN. Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis Appl.*, 30(3):1254–1279, 2008.

[15] P. COMON and G. H. GOLUB. Tracking of a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, August 1990. (published from Stanford report 78NA-89-01, feb 1989).

[16] P. COMON and B. MOURRAIN. Decomposition of quantics in sums of powers of linear forms. *Signal Processing, Elsevier*, 53(2):93–107, September 1996. special issue on High-Order Statistics.

[17] P. COMON and M. RAJIH. Blind identification of under-determined mixtures based on the characteristic function. *Signal Processing*, 86(9):2271–2281, September 2006.

[18] P. COMON and J. ten BERGE. Generic and typical ranks of three-way arrays. In *Icassp'08*, Las Vegas, March 30 - April 4 2008.

[19] P. COMON, J. M. F. ten BERGE, L. DeLATHAUWER, and J. CASTAING. Generic and typical ranks of multi-way arrays. *Linear Algebra Appl.*, 2008. submitted.

[20] D. COX, J. LITTLE, and D. O'SHEA. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer Verlag, New York, 1992. 2nd ed. in 1996.

[21] 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*, 87(2):337–351, February 2007.

[22] L. DeLATHAUWER. A link between canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM Journal on Matrix Analysis*, 28(3):642–666, 2006.

[23] L. DeLATHAUWER and J. CASTAING. Tensor-based techniques for the blind separation of DS-CDMA signals. *Signal Processing*, 87(2):322–336, February 2007.

[24] L. DeLATHAUWER, B. de MOOR, and J. VANDEWALLE. A multilinear singular value decomposition. *SIAM Jour. Matrix Ana. Appl.*, 21(4):1253–1278, April 2000.

[25] L. DeLATHAUWER, 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.*, 21(4):1324–1342, April 2000.

[26] R. EHRENBORG. Canonical forms of two by two matrices. *Jour. of Algebra*, 213:195–224, 1999.

[27] G. H. GOLUB and C. F. VAN LOAN. *Matrix computations*. Hopkins Univ. Press, 1989.

[28] G. H. GOLUB and V. PEREYRA. Differentiation of pseudo-inverses, separable nonlinear least square problems and other tales. In M. Z. Nashed, editor, *Generalized Inverses and Applications*, pages 303–324. Academic Press, New York, 1976.

[29] R. A. HARSHMAN. Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970. http://publish.uwo.ca/ harshman.

[30] R. A. HARSHMAN. *An annoted bibliogrtaphy of articles on degenerate solutions or decompositions*, 2004. Unpublished, http://publish.uwo.ca/∼harshman/bibdegen.pdf.

[31] R. A. HARSHMAN and M. E. LUNDY. Parafac: Parallel factor analysis. *Computational Statistics and Data Analysis*, pages 39–72, 1994.

[32] F. L. HITCHCOCK. Multiple invariants and generalized rank of a p-way matrix or tensor. *J. Math. and Phys.*, 7(1):39–79, 1927.

[33] W. P. KRIJNEN, T. K. DIJKSTRA, and A. STEGEMAN. On the non-existence of optimal solutions and the occurrence of degeneracy in the Candecomp/Parafac model. *Psychometrika*, 73(3):431–439, September 2008.

[34] P. KROONENBERG. *Three mode Principal Component Analysis*. SWO Press, Leiden, 1983.

[35] J. B. KRUSKAL. Three-way arrays: Rank and uniqueness of trilinear decompositions. *Linear Algebra and Applications*, 18:95–138, 1977.

[36] J. B. KRUSKAL, R. A. HARSHMAN, and M. E. LUNDY. How 3-MFA data can cause degenerate Parafac solutions, among other relationships. In R. Coppi and S. Bolasco, editors, *Multiway Data Analysis*, pages 115–121. Elsevier Science, North-Holland, 1989.

[37] J. M. LANDSBERG. *Kruskal's theorem*, 2008. private communication.

[38] E. LASKER. Kanonische formen. *Mathematische Annalen*, 58:434–440, 1904.

[39] T. LICKTEIG. Typical tensorial rank. *Linear Algebra Appl.*, 69:95–120, 1985.

[40] L-H. LIM and P. COMON. Nonnegative approximations of nonnegative tensors. *J. Chemometrics*, 2008. this issue.

[41] K. MADSEN, H. B. NIELSEN, and O. TINGLEFF. *Methods for Non-Linear Least Squares Problems*. Informatics and mathematical Modelling. Technical University of Denmark, 2004.

[42] M. MELLA. Singularities of linear systems and the Waring problem. *Trans. Am. Math. Soc.*, 358(12), December 2006.

[43] B. C. MITCHELL and D. S. BURDICK. Slowly converging Parafac sequences: Swamps and two-factor degeneracies. *Jour. Chemometrics*, 8:155–168, 1994.

[44] C. NAVASCA, L. De LATHAUWER, and S. KINDERMANN. Swamp reducing technique for tensor decomposition. In *16th European Signal Processing Conference (Eusipco'08)*, Lausanne, August 25-29 2008.

[45] D. NION and L. DeLATHAUWER. An enhanced line search scheme for complex-valued tensor decompositions. application in DS-CDMA. *Signal Processing*, 88(3):749755, March 2008.

[46] P. PAATERO. The multilinear engine: A table-driven, least squares program for solving multilinear problems, including the n-way parallel factor analysis model. *Journal of Computational and Graphical Statistics*, 8(4):854–888, December 1999.

[47] P. PAATERO. Construction and analysis of degenerate Parafac models. *Jour. Chemometrics*, 14:285–299, 2000.

[48] M. RAJIH and P. COMON. Enhanced line search: A novel method to accelerate Parafac. In *Eusipco'05*, Antalya, Turkey, Sept. 4-8 2005.

[49] M. RAJIH, P. COMON, and R. HARSHMAN. Enhanced line search : A novel method to accelerate PARAFAC. *SIAM Journal on Matrix Analysis Appl.*, 30(3):1148–1171, September 2008.

[50] C. R. RAO. *Linear Statistical Inference and its Applications*. Probability and Statistics. Wiley, 1965.

[51] C.R. RAO and S. MITRA. *Generalized Inverse of Matrices and Its Applications*. New York: Wiley, 1971.

[52] W. S. RAYENS and B. C. MITCHELL. Two-factor degeneracies and a stabilization of Parafac. *Chemometrics Intell. Lab. Syst.*, 38:173–181, 1997.

[53] J. RISSANEN. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific Publ., London, 1989.

[54] N. D. SIDIROPOULOS and R. BRO. On the uniqueness of multilinear decomposition of N-way arrays. *Jour. Chemo.*, 14:229–239, 2000.

[55] N. D. SIDIROPOULOS, G. B. GIANNAKIS, and R. BRO. Blind PARAFAC receivers for DS-CDMA systems. *Trans. on Sig. Proc.*, 48(3):810–823, March 2000.

[56] V. De SILVA and L-H. LIM. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis Appl.*, 30(3):1084–1127, 2008.

[57] A. SMILDE, R. BRO, and P. GELADI. *Multi-Way Analysis*. Wiley, 2004.

[58] A. STEGEMAN. Degeneracy in Candecomp/Parafac explained for $p \times p \times 2$ arrays of rank $p + 1$ or higher. *Psychometrika*, 71(3):483–501, 2006.

[59] A. STEGEMAN. Degeneracy in Candecomp/Parafac and Indscal explained for several three-sliced arrays with a two-valued typical rank. *Psychometrika*, 72(4):601–619, 2007.

[60] A. STEGEMAN. Low-rank approximation of generic pxqx2 arrays and diverging components in the Candecomp/Parafac model. *SIAM Journal on Matrix Analysis Appl.*, 30(3):988–1007, 2008.

[61] A. STEGEMAN and N. D. SIDIROPOULOS. On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition. *Linear Algebra and Appl.*, 420:540–552, 2007.

[62] V. STRASSEN. Rank and optimal computation of generic tensors. *Linear Algebra Appl.*, 52:645–685, July 1983.

[63] J. M. F. ten BERGE. Kruskal's polynomial for $2 \times 2 \times 2$ arrays and a generalization to $2 \times n \times n$ arrays. *Psychometrika*, 56:631–636, 1991.

[64] J. M. F. ten BERGE. The typical rank of tall three-way arrays. *Psychometrika*, 65(5):525–532, September 2000.

[65] J. M. F. ten BERGE. Partial uniqueness in CANDECOMP/PARAFAC. *Jour. Chemometrics*, 18:12–16, 2004.

[66] J. M. F. ten BERGE and H. A. L. KIERS. Simplicity of core arrays in three-way principal component analysis and the typical rank of pxqx2 arrays. *Linear Algebra Appl.*, 294:169–179, 1999.

[67] J. M. F. ten BERGE and A. STEGEMAN. Symmetry transformations for square sliced three way arrays, with applications to their typical rank. *Linear Algebra Appl.*, 418:215–224, 2006.

[68] A. TERRACINI. Sulla rappresentazione delle forme quaternarie mediante somme di potenze di forme lineari. *Atti della R. Acc. delle Scienze di Torino*, 51, 1916.

[69] G. TOMASI and R. BRO. A comparison of algorithms for tting the Parafac model. *Comp. Stat. Data Anal.*, 50:1700–1734, 2006.

[70] L. R. TUCKER. Some mathematical notes for three-mode factor analysis. *Psychometrika*, 31:279–311, 1966.

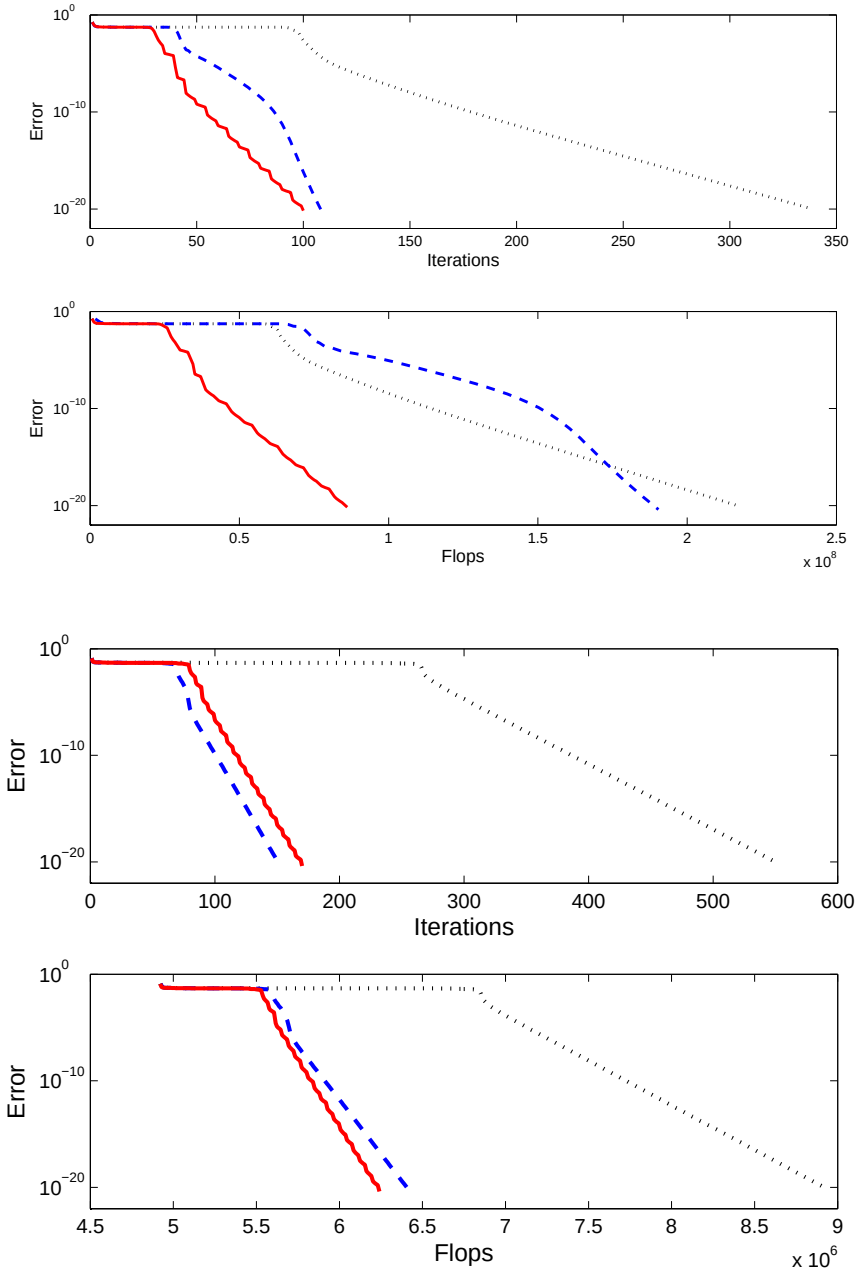

Figure 6: Reconstruction error of a $30 \times 30 \times 30$ tensor of rank 4 with 2 double bottlenecks as a function of the number of iterations or the number of multiplications, and for various algorithms. Solid: ALS + ELS with period 5, dashed: ALS + ELS with period 1 (at every iteration), dotted: ALS. Top: without dimension reduction, bottom: after reducing dimensions down to to 4.

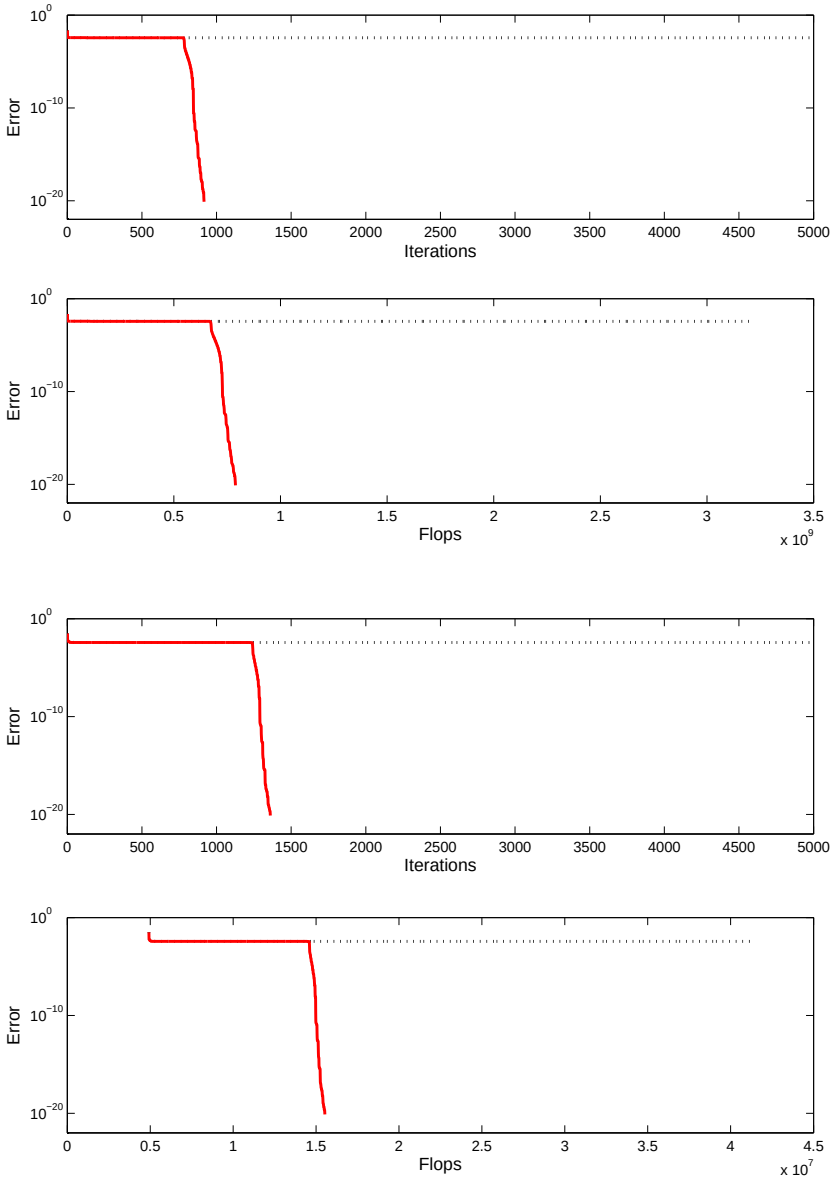

Figure 7: Reconstruction error of a $30 \times 30 \times 30$ of rank 4 with 2 triple bottlenecks as a function of the number of iterations (top) or the number of multiplications (bottom), and for various algorithms. Solid: ALS + ELS with period 5, dotted: ALS. Top: without dimension reduction, bottom: after reducing dimensions down to to 4.