ENHANCED LINE SEARCH APPLIED TO BLIND CHANNEL IDENTIFICATION : IDENTIFIABILITY CONDITIONS

Myriam Rajih and Pierre Comon

I3S laboratory
2000 route des Lucioles, B.P.121, F-06903 Sophia-Antipolis, France
phone: +33 4 9294 2793/2717, fax: +33 4 9294 2896, {rajih,comon}@i3s.unice.fr
www.i3s.unice.fr

ABSTRACT

The ALS algorithm, used to fit the PARAFAC model, sometimes needs a large number of iterations before converging. The slowness in convergence can be due to the large size of the data, or to the presence of degeneracies, etc. Several methods have been proposed to speed up the algorithm, some of which are compression [1], and Line Search [2]. [3] presents a novel method for speeding up the algorithm, Enhanced Line Search (ELS), that shows better results in simulations compared to the existing methods, especially in the case of degeneracy. This paper gives an application of ELS to blindly identify the mixing matrix of an Under-Determined Mixture (UDM) : algorithm ALESCAF, and states the identifiability conditions based on ALESCAF.

1. INTRODUCTION

PARAFAC can be viewed as one possible generalization of the Singular Value Decomposition (SVD) to higher orders. It was independently introduced by Harshman in 1970 [4], and the same year by Caroll and Chang [5] as CANDECOMP for CANonical DECOMPosition.

SVD consists of decomposing a matrix $\mathbf{X}_1$ of size $I \times J$ as a sum of rank one matrices :

$$\mathbf{X}_1 = \sum_{r=1}^{R} s_r A_r B_r^T$$

A_r and B_r are the columns of the matrices $\mathbf{A}$ and $\mathbf{B}$ respectively. s_r ($r = 1, ..., R$) are the singular values of $\mathbf{X}_1$ stored in $\mathbf{S}$, such as : $\mathbf{X}_1 = \mathbf{ASB}^T$. PARAFAC generalizes the SVD decomposition to three-way arrays. It consists of decomposing a tensor $\mathbf{X}$ of size $I \times J \times K$ as a sum of rank one tensors :

$$\mathbf{X} = \sum_{f=1}^{F} A_f \circ B_f \circ C_f \tag{1}$$

where matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are of size $I \times F$, $J \times F$, and $K \times F$ respectively, and $\circ$ is the outer product.

SVD reveals the rank R of the decomposed matrix $\mathbf{X}_1$, as the number of non zero singular values is equal to R. In other words, $\mathbf{X}_1$ cannot be represented with fewer than R factors. Equivalently, the rank F of the tensor $\mathbf{X}$ is defined as the minimum number of factors used in the rank one decomposition of $\mathbf{X}$.

SVD provides unique decomposition up to a permutation and a scale factor. Uniqueness of SVD is achieved because orthogonality constraints are imposed on the matrices $\mathbf{A}$ and $\mathbf{B}$. General unconstrained two-way decompositions (Principal Component Analysis, PCA) lack uniqueness, because introducing any non-singular matrix $\mathbf{P}$ keeps the fit of the data unchanged : $\mathbf{X}_1 = \mathbf{APP}^{-1}\mathbf{B}^T$, which means that the two-way model suffers a rotational indeterminacy that yields an infinite set of solutions. At the opposite, the general unconstrained PARAFAC model enjoys a uniqueness property under simple conditions summarized in the Kruskal theorem [6], hence its importance and its wide application. The three-way PARAFAC model is very popular in psychometrics and chemometrics where it was first used along with its extensions to higher orders [4] [5][7]. It also finds applications in the signal processing area [8] [9] [10].

Many algorithms propose a solution to fit the PARAFAC model, one of which is the Alternating Least Square (ALS) algorithm. The convergence of ALS was found to be very slow in some cases, typically when the size of the data is very large, or when two factors are almost collinear. Compression [1] and Line Search [2] are some of the solutions proposed to cope with the problem of slow convergence. We focus in this paper on the Line Search solution and present a novel method for speeding up ALS, that shows very good performance both in terms of the number of iterations and of the overall computational complexity.

2. MODEL AND NOTATION

We consider the three-way PARAFAC model of expression (1). This model can be written in a compact form using the Khatri-Rao product $\odot$ (column-wise Kronecker product) :

$$\mathbf{X}^{(I \times JK)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T$$

where $\mathbf{X}^{(I \times JK)}$ is the matrix of size $I \times JK$ obtained by unfolding the tensor $\mathbf{X}$ of size $I \times J \times K$ in the first mode. There exist several algorithms that fit the PARAFAC model. We focus on the most famous among all : the ALS algorithm. ALS consists of estimating one of the three matrices at each step by minimizing in the Least Square sense the error :

$$\Upsilon = \| \mathbf{X}^{(I\times JK)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T \|_F^2$$

where $\| \bullet \|_F$ is the Frobenius norm. With matrices $\mathbf{B}$ and $\mathbf{C}$ fixed to initial values, the estimate of $\mathbf{A}$ in the Least Square sense is given by :

$$\widehat{\mathbf{A}} = \mathbf{X}^{(I\times JK)}(\mathbf{Z}_a^{+})^T \qquad (2)$$

where $\mathbf{Z}_a = \mathbf{C} \odot \mathbf{B}$ and $(^{+})$ is the Moore-Penrose pseudo-inverse. We estimate matrices $\mathbf{B}$ and $\mathbf{C}$ in an equivalent way, with $\mathbf{Z}_b = \mathbf{A} \odot \mathbf{C}$ and $\mathbf{Z}_c = \mathbf{B} \odot \mathbf{A}$, and repeat the same steps until a convergence criterion is reached. Typically when the error Υ exhibits, between two iterations, a change smaller than a predefined threshold, which varies depending on the data. For simple data it can be set to 10^{-6} for example, but it should be smaller for difficult data, 10^{-10} for example.

Sometimes the convergence needs a very large number of iterations. Choosing good starting values can help on reaching the global minimum very quickly. But when the array has large dimensions, or when one dimension is very large compared to the other ones, initialization techniques do not solve the problem of slowness. The slowness of convergence also occurs when two factors are almost collinear. In [1] Bro proposes to compress the data in a smaller space so that the dimensions of the new array are reduced, thus reducing the complexity of ALS. Line Search was also proposed to speed up the convergence [11] [2]. This solution is discussed in the next section.

3. LINE SEARCH

It was noticed through simulations that, when the convergence is slow, there exist cycles of convergence defined by a unique direction. Within a given cycle, the loading factors evolve in the same direction to the final solution of that cycle. The following cycles exhibit the same scenario. The convergence within the cycle can take several iterations. To limit the number of iterations of a given cycle, Harshman and Bro propose to extrapolate. They propose to predict the value of the loading factors a certain number of iterations ahead by computing a kind of linear regression :

$$\mathbf{A}^{(new)} = \mathbf{A}^{(it-2)} + R_{LS}(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}) \qquad (3)$$

$\mathbf{A}^{(it-1)}$ is the estimate of matrix $\mathbf{A}$ obtained in the ALS iteration $(it-1)$, and $\mathbf{A}^{(new)}$ is the matrix that will be used in the it^{th} iteration instead of $\mathbf{A}^{(it-1)}$. $(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)})$ defines the direction of the cycle. Matrices $\mathbf{B}$ and $\mathbf{C}$ are obtained in an equivalent way using the same relaxation factor R_{LS}. Of course, extrapolation should be very simple and does not have sense if it requires more time than the corresponding iterations. This is the case when R_{LS} is given a fixed value (between 1.2 and 1.3) [4], or is set to $it^{1/3}$ [2].

At every iteration it, the "new" loading factors are used to compute the error :

$$\Upsilon^{(new)} = \| \mathbf{X}^{(I\times JK)} - \mathbf{A}^{(new)}(\mathbf{C}^{(new)} \odot \mathbf{B}^{(new)})^T \|_F^2 \qquad (4)$$

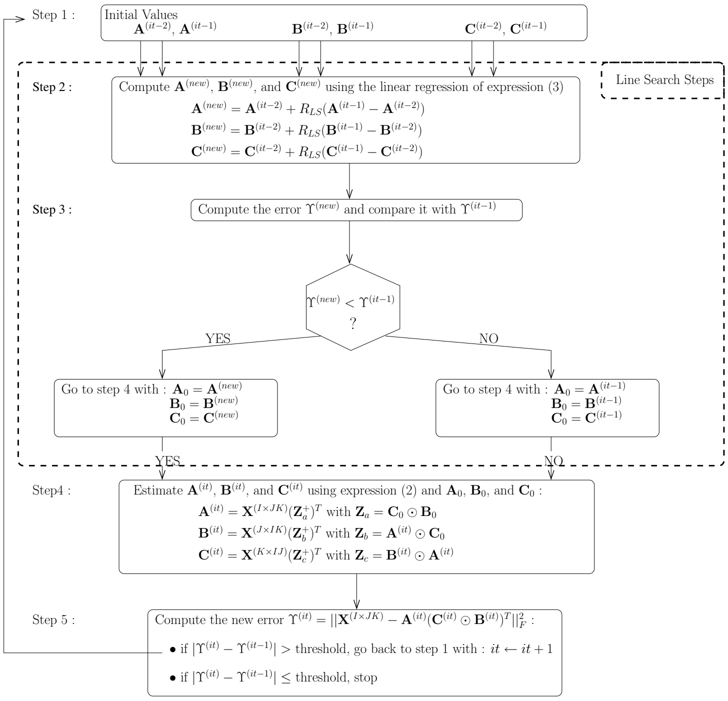

Fig. 1: Steps of the ALS algorithm with Line Search.

If $\Upsilon^{(new)} \geq \Upsilon^{(it-1)}$ this means that we went too far in the extrapolation because R_{LS} is too large. R_{LS} is decreased from $it^{1/n}$ to $it^{1/(n+1)}$ (n is set to 3 at the beginning of the simulation), and we take the loading factors of iteration $(it-1)$ instead of the "new" ones. However, if $\Upsilon^{(new)} < \Upsilon^{(it-1)}$ acceleration is accomplished and we gain some iterations.

The following graph in figure 1 describes the steps of the ALS algorithm with Line Search as proposed by Andersson and Bro in [12]. The dashed space corresponds to the Line Search part.

Line Search is applied after few iterations of the ALS algorithm in order to wait for the system to stabilize. In [12] "few" is set to 6 but it could be higher depending on the data. The relaxation factor R_{LS} is defined for iteration (it) by : $R_{LS} = it^{1/n}$, with n fixed to 3 at the beginning of the simulation. When the acceleration fails several times (5 times in [12]), R_{LS} is decreased to $it^{1/(n+1)}$ and $\mathbf{A}^{(it-1)}$, $\mathbf{B}^{(it-1)}$, and $\mathbf{C}^{(it-1)}$ are used to update the loading factors of the current iteration (it) as described by the graph in figure 1 at the end of step 3. However, when $\Upsilon^{(new)} < \Upsilon^{(it-1)}$ matrices $\mathbf{A}_0$, $\mathbf{B}_0$, and $\mathbf{C}_0$ are set to $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, and $\mathbf{C}^{(new)}$ respectively.

The fact that R_{LS} has a small value would suggest that the acceleration is not very efficient. This is not true since the effect of R_{LS} is re-conducted from one iteration to the other, leading in final to a noticeable reduction of the number of iterations as shown in figure 3. The model used in the simulation is exposed in section 5. The number of iterations necessary to reach convergence decreases from more than 10000 to 4907. However, it is still high. Therefore, it is of great interest to look for a novel method to reduce the time consumption of ALS significantly.

4. ENHANCED LINE SEARCH (ELS)

The Enhanced Line Search algorithm is described in [3], where it is shown that ELS performs better than ALS with Line Search both in terms of, complexity and, the total number of iterations needed to reach convergence.

We focus in this paper on the application of ELS to blindly identify the mixing matrix of an UDM, and on the identifiability conditions that ensure that the mixing matrix is unique up to scale and permutation factors.

We assume the observation model below :

$$\mathbf{x} = \mathbf{A}\,\mathbf{s} + \mathbf{w} \tag{5}$$

where array variables are distinguished from scalars by bold faces, $\mathbf{x}$ and $\mathbf{s}$ are random vectors of size P and N respectively, $\mathbf{A}$ is a $P \times N$ full rank matrix, and $\mathbf{w}$ accounts for modeling errors and additive noise. From now on, its presence is just ignored in the remaining, except when running computer experiments. The entries s_n of vector $\mathbf{s}$ are assumed to be non Gaussian and statistically independent.

For simplicity, we shall restrict our attention in this paper to real variables and mixture. As pointed out in [13], the immersion of the complex framework in a real framework of larger size introduces some additional constraints, which make the problem more difficult, but at the same time allow a better stability of the solution. Most of the reasoning developed in this paper applies to the complex case, up to some complication in the notation.

We also assume the following hypotheses:

H1 the columns of $\mathbf{A}$ are pairwise linearly independent.

H2 source distributions are unknown and non Gaussian

H3 the number N of sources is known

H4 the moments of the sources are unknown, but finite up to some order larger than N

Under **H1**, **H2**, and **H3**, $\mathbf{A}$ can be shown to be essentially unique [14, pp.311-313].

Algorithm ALESCAF (Alternate LEast Square identification based on the ChAracteristic Function) is subsequently described. ALESCAF is able to solve the derivative matching problem of [15] and [13] for $N > 2$ with the help of an ALS algorithm. It is based on the core functional equation below, which is a direct consequence of source independence:

$$\Psi_x(\mathbf{u}) = \sum_{n=1}^{N} \psi_n\Big(\sum_{p=1}^{P} A_{pn} u_p\Big) \tag{6}$$

where $\Psi_x(\mathbf{u})$ denotes the second c.f. of $\mathbf{x}$ defined as $\Psi_x(\mathbf{u}) = \log \mathrm{E}\{\exp(\mathbf{u}^{\mathsf{T}}\mathbf{x})\}$, and where $\psi_n(v)$ denotes the second c.f. of source s_n: $\psi_n(v) = \log \mathrm{E}\{\exp(v s_n)\}$. This core equation can be used in an open neighborhood Ω of the origin, where Ψ_x does not vanish.

4.1. Description of the algorithm ALESCAF

From (6), one can easily obtain that

$$\frac{\partial^3 \Psi_x(\mathbf{u})}{\partial u_i \partial u_j \partial u_p} = \sum_{n=1}^{N} A_{in} A_{jn} A_{pn}\, \psi_n^{(3)}\Big(\sum_q A_{qn} u_q\Big) \tag{7}$$

As in [13], take this equation on K points $\mathbf{u}[k] \in \Omega$. Then, storing the left hand side of (7) in a family of symmetric matrices $T_{ij}[p,k]$, and denoting $D_{kn} = \psi_n^{(3)}(\sum_q A_{qn} u_q[k])$, (7) can be arranged in compact form as

$$\mathbf{T}[p,k] = \mathbf{A}\,\mathrm{Diag}\{\mathbf{A}(p,:)\}\,\mathrm{Diag}\{\mathbf{D}(k,:)\}\,\mathbf{A}^{\mathsf{T}}, \tag{8}$$

with $1 \le p \le P$, $1 \le k \le K$, and where $\mathrm{Diag}\{\mathbf{v}\}$ denotes the diagonal matrix whose entries are those of vector $\mathbf{v}$. Expression (8) is a four way PARAFAC model and can be solved using the ALS algorithm described in section (2). This procedure constitutes algorithm ALESCAF and is able to compute $\mathbf{A}$ and $\mathbf{D}$ from symmetric matrices $\mathbf{T}[p,k]$ (the implicit dependence of $\mathbf{D}$ on $\mathbf{A}$ is ignored).

4.2. Identifiability

Essential uniqueness of $\mathbf{A}$ is achieved as Kruskal's inequality is verified [6]:

$$3r_k(\mathbf{A}) + r_k(\mathbf{D}) \ge 2\,\mathrm{rank}\{\mathbf{T}\} + 3 \tag{9}$$

where $r_k(\mathbf{A})$ is the Kruskal rank [6] of $\mathbf{A}$. Results are expected to be better when increasing the order of the statistics as we move away from the Parafac limit, but this need to be verified by simulations.

More precisely, we have the Theorem below.

Theorem

Model (5) is identifiable with the help of algorithm ALESCAF if the following conditions are all satisfied :

C1 *the unknown second characteristic functions of the sources and their derivatives up to order* 3 *exist, are finite and do not vanish in a neighborhood* Ω *of the origin*

C2 *the Kruskal rank [6] of the mixing matrix* $\mathbf{A}$ *verifies* $r_k(\mathbf{A}) = P$

C3 $r_k(\mathbf{D}) = N$

C4 *for a given* P *the number of sources must verify :* $N \le 3P - 3$

Proof:

- Condition [**C1**] allows us to derive expression (7) and build the four-way array $\mathbf{T}$.

- If conditions [**C2**], [**C3**], and [**C4**] are verified, then :

$$3r_k(\mathbf{A}) + r_k(\mathbf{D}) \geq 2\,\mathrm{rank}\{\mathbf{T}\} + 3 \qquad (10)$$

  which means that the uniqueness conditions of the four-way PARAFAC model is achieved, leading to the uniqueness of $\mathbf{A}$.

■

[**C3**] is easy to achieve by taking a large number of points on the grid $\Omega : K >> N$. But this increases the complexity of the ALS algorithm used to fit the PARAFAC model, as the fourth dimension of the tensor $\mathbf{T}$ will increase, leading to large size of the data. A good choice of the points on the grid can lead to an optimal value of K, thus reducing the complexity of computation.

Condition [**C2**] can be relaxed to :

C2' $\mathbf{A}$ is full row rank and its columns are pairwise linearly independent.

But this implies a limitation on the number of sources and/or an increase of complexity with the use of arrays of higher order. Let's take an example.

Example

We consider the following mixing matrix :

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$\mathbf{A}$ *is full row rank (rank(*$\mathbf{A}$*)=3) but is not full Kruskal rank (*$r_k(\mathbf{A}) = 2$*). Hence, relation (10), which ensures uniqueness of* $\mathbf{A}$*, is no longer verified. To cope with this problem we build a tensor of order five* $\mathbf{T}^{(5)}$ *by deriving one time entries of tensor* $\mathbf{T}$*. Then (10) becomes :*

$$4\,r_k(\mathbf{A}) + r_k(\mathbf{D}) \geq 2\,\mathrm{rank}\{\mathbf{T}^{(5)}\} + 4$$

and uniqueness is achieved. At the same time the complexity of the PARAFAC *model increases, since tensor* $\mathbf{T}^{(5)}$ *is of higher order (5 instead of 4) as well as the derivatives (order 4 instead of 3).*

In ALESCAF the number of sources N is limited by condition [**C4**]. Table 1 gives the maximum value of N as a function of P for $2 \leq P \leq 8$.

P	2	3	4	5	6	7	8
N	3	6	9	12	15	18	21

Table 1: Maximum number of sources N as a function of the number of sensors P using conditions of theorem 2.

5. COMPUTER RESULTS

Estimates of 4-way tensor $\mathbf{T}$ are computed in the following manner. First, all derivatives of $\Psi_x(\mathbf{u})$ of required order are formally expressed as a function of moments $\mu(\mathbf{n}, \mathbf{u}[k]) = \mathrm{E}\{\mathbf{x}^{\boldsymbol{n}} e^{\boldsymbol{x}^\mathsf{T}\boldsymbol{u}[k]}\}$. Then sample moments $\hat{\mu}(\mathbf{n}, \mathbf{u}[k]) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}[m]^{\boldsymbol{n}} e^{\boldsymbol{x}[m]^\mathsf{T}\boldsymbol{u}[k]}\}$ are computed, yielding eventually estimates of entries of $\mathbf{T}$ (here $\mathbf{x}^{\boldsymbol{n}}$ stands for $\Pi_p x_p^{n_p}$). Other useful expressions are reported in appendix. This formal approach to take the derivatives of $\Psi_x(\mathbf{u})$ is more accurate, but turns out to be more sensitive to derivations of the sample mean from zero. Another possible approach for computing the derivatives of $\Psi_x(\mathbf{u})$ is a numerical one. It consists of estimating the characteristic function $\Psi_x(\mathbf{u})$ of the observation over a set $(\mathbf{u}) \in \Omega$ containing the origin. We merely utilized the sample estimate:

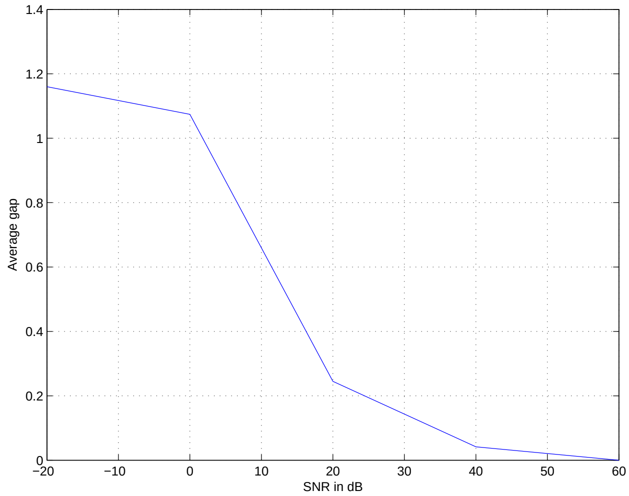

Fig. 2: Gap between estimated and actual mixing matrix for $(P, N) = (2, 3)$, with algorithm ALESCAF. Average gap values over 21 independent trials are plotted.

$$\hat{\Psi}_x(\mathbf{u}) = \log\left[\frac{1}{T}\sum_{t=1}^{T}\exp\{i\Re(\boldsymbol{x}(t)^\mathsf{T}\boldsymbol{u})\}\right] \qquad (11)$$

where vectors $\mathbf{x}(t)$ are realizations of the random variable $\mathbf{x}$. The successive derivatives can then be obtained from the values of $\hat{\Psi}_x(\mathbf{u})$ over a grid included in Ω by central finite differences.

The number of sensors is taken to be $P = 2$, and the number of sources is $N = 3$. Sources are BPSK, that is, they take their values in $\{-1, 1\}$ with equal probabilities. In both figures, we generate an infinite block of data by taking all the 2^3 possible combinations of $\{-1, 1\}$.

In figure 2 we report the influence of noise on ALESCAF algorithm. We start with SNR=60dB, check for convergence, and use the value of the corresponding loading matrices $\mathbf{A}$ and $\mathbf{D}$ to initialize the next ALESCAF algorithm for SNR=50dB and so on. By doing so, one expects to access ultimate performances, *i.e.* in actual situations, performances will be poorer in actual situations.

Figure 3 reports the gap between estimated and actual mixing matrix for six optimizations of ELS and compare them with ALS with Line Search and non accelerated ALS. We use the ALS implementation proposed by Andersson and Bro in [12] and replace the Line Search procedure by the six optimizations of ELS shown in figure 3 [3]. Noise is

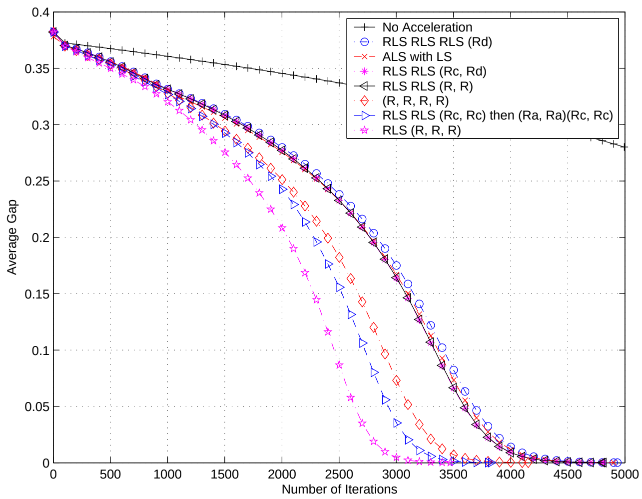

Fig. 3: Gap between estimated and actual mixing matrix for $(P, N) = (2, 3)$, with algorithm ALESCAF.

not taken into account, and we take 5000 as the maximum number of iterations. The figure shows that ELS is very useful for speeding up the convergence since the number of iterations needed to reach convergence decreases from 4200 when using ALS with Line Search, to 2900 when using optimization $R_{LS}(R, R, R)$ of ELS.

6. CONCLUDING REMARKS

We presented identifiability conditions based on ELS, a novel method for accelerating the ALS algorithm used to fit the PARAFAC model [3]. The application of ELS to the Blind Channel Identification in an UDM showed very good results as ELS speeded up the convergence.

In future works, noise will be taken into account and a wider class of data blocks will be considered. Simulations will be run with large data size to confirm that ELS is also attractive in this case.

7. APPENDIX : FORMAL DERIVATIVES OF THE SECOND CHARACTERISTIC FUNCTION

We consider the case of a real 2×3 mixture, and compute the third order derivatives of $\Psi_x(\mathbf{u})$. We denote by E the first joint characteristic function of $\mathbf{x} = (x_1, x_2)$ ($\mathbf{u} = (u, v)$) :

$$E = \Phi_{(x_1,x_2)}(u,v) = E[exp(jux_1 + jvx_2)]$$

The successive derivatives of E with respect to u and v are given by :

$$\begin{aligned}
E_1 &= \frac{\partial \Phi_{(x_1,x_2)}(u,v)}{\partial u} = E[jx_1 exp(jux_1 + jvx_2)] \\
E_2 &= \frac{\partial^2 \Phi_{(x_1,x_2)}(u,v)}{\partial u^2} = E[-x_1^2 exp(jux_1 + jvx_2)] \\
E_3 &= \frac{\partial^3 \Phi_{(x_1,x_2)}(u,v)}{\partial u^3} = E[-jx_1^3 exp(jux_1 + jvx_2)]
\end{aligned}$$

$$\begin{aligned}
F_1 &= \frac{\partial \Phi_{(x_1,x_2)}(u,v)}{\partial v} = E[jx_2 exp(jux_1 + jvx_2)] \\
F_2 &= \frac{\partial^2 \Phi_{(x_1,x_2)}(u,v)}{\partial v^2} = E[-x_2^2 exp(jux_1 + jvx_2)] \\
F_3 &= \frac{\partial^3 \Phi_{(x_1,x_2)}(u,v)}{\partial v^3} = E[-jx_2^3 exp(jux_1 + jvx_2)]
\end{aligned}$$

$$\begin{aligned}
D_{11} &= \frac{\partial^2 \Phi_{(x_1,x_2)}(u,v)}{\partial u \partial v} = E[-x_1 x_2 exp(jux_1 + jvx_2)] \\
T_{21} &= \frac{\partial^3 \Phi_{(x_1,x_2)}(u,v)}{\partial u^2 \partial v} = E[-jx_1^2 x_2 exp(jux_1 + jvx_2)] \\
T_{12} &= \frac{\partial^3 \Phi_{(x_1,x_2)}(u,v)}{\partial u \partial v^2} = E[-jx_1 x_2^2 exp(jux_1 + jvx_2)]
\end{aligned}$$

As $\Psi_{\boldsymbol{x}}(u,v) = \log(\Phi_{\boldsymbol{x}}(u,v))$ we have :

$$\begin{aligned}
\frac{\partial \Psi_{\boldsymbol{x}}(u,v)}{\partial u} &= \frac{\partial \log(E)}{\partial u} = \frac{E_1}{E} \\
\frac{\partial^2 \Psi_{\boldsymbol{x}}(u,v)}{\partial u^2} &= \frac{\partial}{\partial u}\left(\frac{E_1}{E}\right) = \frac{E_2 E - E_1^2}{E^2} \\
\frac{\partial^3 \Psi_{\boldsymbol{x}}(u,v)}{\partial u^3} &= \frac{\partial}{\partial u}\left(\frac{E_2 E - E_1^2}{E^2}\right) = \frac{E_3 E^2 - 3E_2 E_1 E + 2E_1^3}{E^3} \\
\frac{\partial^3 \Psi_{\boldsymbol{x}}(u,v)}{\partial u^2 \partial v} &= \frac{\partial}{\partial v}\left(\frac{E_2 E - E_1^2}{E^2}\right) = \frac{T21 E^2 - E_2 F_1 E - 2E_1 D_1 E + 2E_1^2 F_1}{E^3}
\end{aligned}$$

Using the symmetry between the variables u and v we obtain derivatives with respect to v :

$$\begin{aligned}
\frac{\partial \Psi_{\boldsymbol{x}}(u,v)}{\partial v} &= \frac{\partial \log(E)}{\partial v} = \frac{F_1}{E} \\
\frac{\partial^2 \Psi_{\boldsymbol{x}}(u,v)}{\partial v^2} &= \frac{\partial}{\partial v}\left(\frac{F_1}{E}\right) = \frac{F_2 E - F_1^2}{E^2} \\
\frac{\partial^3 \Psi_{\boldsymbol{x}}(u,v)}{\partial v^3} &= \frac{\partial}{\partial v}\left(\frac{F_2 E - F_1^2}{E^2}\right) = \frac{F_3 E^2 - 3F_2 F_1 E + 2F_1^3}{E^3} \\
\frac{\partial^3 \Psi_{\boldsymbol{x}}(u,v)}{\partial u \partial v^2} &= \frac{\partial}{\partial u}\left(\frac{F_2 E - F_1^2}{E^2}\right) = \frac{T12 E^2 - F_2 E_1 E - 2F_1 D_1 E + 2F_1^2 E_1}{E^3}
\end{aligned}$$

8. REFERENCES

[1] R. BRO and C. A. ANDERSSON, "Improving the speed of multiway algorithms. part ii : Compression," *Chemometrics and Intelligent Laboratory Systems*, , no. 42, pp. 105–113, 1998.

[2] R. BRO, *Multi-way Analysis in the Food Industry : Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, Amsterdam, 1998.

[3] M. RAJIH and P. COMON, "Enhanced line search : A novel method to accelerate parafac," *EUSIPCO Conference*, Sept 4–8 2005.

[4] R. A. HARSHMAN, "Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[5] J. D. CARROLL and J. J. CHANG, "Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, Sept. 1970.

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

[7] N. D. SIDIROPOULOS and R. BRO, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of Chemometrics*, vol. 14, pp. 229–239, 2000.

[8] L. DE LATHAUWER, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, K. U. Leuven, E. E. Dept. -ESAT, Belgium, 1997.

[9] P. COMON, "Blind channel identification and extraction of more sources than sensors," in *SPIE Conference*, San Diego, July 19-24 1998, pp. 2–13, republished in IEEE Trans. Sig. Proc., Jan. 2004.

[10] P. COMON and M. RAJIH, "Blind identification of under-determined mixtures based on the characteristic function," *ICASSP Conference*, Mar 18–23 2005.

[11] R. A. HARSHMAN, "Determination and proof of minimum uniqueness conditions for parafac1," *UCLA Working Papers in Phonetics*, vol. 22, pp. 111–117, 1972.

[12] C. A. ANDERSSON and R. BRO, "The n-way toolbox for matlab," *Chemometrics and Intelligent Laboratory Systems*, vol. 52, no. 9, pp. 1–4, Sept. 2000.

[13] P. COMON and M. RAJIH, "Blind identification of complex under-determined mixtures," in *ICA Conference*, Granada, Spain, Sept 22–24 2004.

[14] A. M. KAGAN, Y. V. LINNIK, and C. R. RAO, *Characterization Problems in Mathematical Statistics*, Probability and Mathematical Statistics. Wiley, New York, 1973.

[15] A. TALEB, "An algorithm for the blind identification of n independent signals with 2 sensors," in *Sixth International Symposium on Signal Processing and its Applications (ISSPA'01)*. 2001, IEEE.