ENHANCED LINE SEARCH : A NOVEL METHOD TO ACCELERATE PARAFAC

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 [3], and Line Search [2]. In this paper, after a description of PARAFAC, we will present a novel method for speeding up the algorithm that shows better results in simulations compared to the existing methods, especially in the case of degeneracy. The paper gives an application of the method to blindly identify the mixing matrix of an Under-Determined Mixture (UDM), but it can be applied to any N-way decomposition problem.

1. INTRODUCTION

PARAFAC can be seen as a generalization of the two-way factor analysis to multi-way data. It was first introduced by Harshman in 1970 [8] based on the principle of parallel Proportional Profiles (PP) proposed by Cattell in 1944 [5]. The PP principle states that if two (or more) different two-way models are described by the same set of loading vectors and only proportions or weights change from one model to the other, then those loading vectors lead to a new model which is unambiguous with respect to (w.r.t.) rotation [2]. In other words, suppose that the matrix $\mathbf{X}_1$ can be modeled as :

$$\mathbf{X}_1 = \mathbf{a}_1\mathbf{b}_1^T c_{11} + \mathbf{a}_2\mathbf{b}_2^T c_{12} + ... + \mathbf{a}_F\mathbf{b}_F^T c_{1F} \quad (1)$$

$\mathbf{a}_f$ and $\mathbf{b}_f$ $(1 \leq f \leq F)$ being the columns of matrices $\mathbf{A}$ and $\mathbf{B}$. And suppose that another matrix $\mathbf{X}_2$ can be modeled using the same set of loading vectors only in different proportions described by c_{ij} :

$$\mathbf{X}_2 = \mathbf{a}_1\mathbf{b}_1^T c_{21} + \mathbf{a}_2\mathbf{b}_2^T c_{22} + ... + \mathbf{a}_F\mathbf{b}_F^T c_{2F} \quad (2)$$

Then, we can build a combined model ;

$$\mathbf{X}_k = \mathbf{A} diag(\mathbf{C}(k,:))\mathbf{B}^T, k = 1,2 \quad (3)$$

which can be alternatively written as : $X_{ijk} = \sum_f A_{if}B_{jf}C_{kf}$. The trilinear model is also known as CANDECOMP for CANonical DECOMPosition introduced by Caroll and Chang in 1970 [4]. The three-way PARAFAC model is very popular in psychometrics and chemometrics where it was first used along with its extension to higher orders [8] [4][12]. It also finds applications in the signal processing area [11] [6] [7]. While the two-way model suffers a rotational indeterminacy that yields an infinite set of solutions, the PARAFAC model enjoys a uniqueness property under simple conditions summarized in the Kruskal theorem [10], hence its importance. 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 [3] 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 in terms of the number of iterations.

2. MODEL AND NOTATION

We consider the three-way PARAFAC model of expression (3). This model can be written in a compact form using the Khatri-Rao product $\odot$ (column-wise Kronecker product) :

$$\mathbf{X}^{(I \times JK)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T \quad (4)$$

where matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are matrices of size $I \times F$, $J \times F$, and $K \times F$, and $\mathbf{X}^{(I \times JK)}$ 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 :

$$\Upsilon = \| \mathbf{X}^{(I \times JK)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T \|_F^2 \quad (5)$$

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 :

$$\widehat{\mathbf{A}} = \mathbf{X}^{(I \times JK)}(\mathbf{Z}_a^+)^T \quad (6)$$

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 [3] 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 [9] [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 senario. 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 :

$$\mathbf{A}^{(new)} = \mathbf{A}^{(it-2)} + R_{LS}(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}) \qquad (7)$$

$\mathbf{A}^{(it-1)}$ 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) [8], or is set to $it^{1/3}$ [2].

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

$$\Upsilon^{(new)} = \| \mathbf{X}^{(I\times JK)} - \mathbf{A}^{(new)}(\mathbf{C}^{(new)} \odot \mathbf{B}^{(new)})^T \|_F^2 \qquad (8)$$

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 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 idea of the Enhanced Line Search (ELS) consists of seeking the optimal relaxation factor R_{LS} that leads to the final solution of a given cycle in only one step. For iteration (it), let's define $\mathbf{G}_a^{(it)} = \mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}$ as the direction of the cycle for loading matrix $\mathbf{A}$. $\mathbf{G}_b^{(it)}$ and $\mathbf{G}_c^{(it)}$ are defined equivalently. Instead of fixing R_{LS} in expression (7), we look for the optimal triplet (R_a, R_b, R_c) that minimizes :

$$\Upsilon_{ELS} = \| \mathbf{X}^{(I\times JK)} - (\mathbf{A}^{(it-2)} + R_a\mathbf{G}_a^{(it)}) \Big((\mathbf{C}^{(it-2)} + R_c\mathbf{G}_c^{(it)}) \odot (\mathbf{B}^{(it-2)} + R_b\mathbf{G}_b^{(it)})\Big)^T \|_F^2 \qquad (9)$$

The optimal solution is obtained when we jointly minimize Υ_{ELS} w.r.t. the three different factors R_a, R_b, and R_c. In this case the problem consists of resolving a system of three polynomials in three unknowns which leads to a high numerical complexity. Solutions with less complexity are obtained by taking only two unknowns, or the same factor for all the matrices $R = R_a = R_b = R_c$. Some of the possible optimizations are listed below :

- (R_a, R_b, R_c) that gives the optimal solution
- (R, R, R_c) where we use the same factor for $\mathbf{A}$ and $\mathbf{B}$ and we minimize Υ_{ELS} w.r.t. two variables R and R_c
- (R, R, R) where we use the same factor for all matrices
- $R(R_b, R_c)$ where we use the relaxation factor of Line Search $R = it^{1/3}$ for matrix $\mathbf{A}$, and minimize (9) w.r.t. R_b and R_c
- $R(R, R)$ which is the same as $R(R_b, R_c)$ with $R_b = R_c$
- $R, R(R)$ where we optimize only w.r.t. to R_c

To make sure that extrapolation makes sense and requires less time than the corresponding iterations needed to reach the final solution of the cycle, we compute the complexity for each method and compare them. Let's take optimization (R, R, R) as an example. At each ALS iteration the following steps are performed :

1. Compute optimal relaxation factor R by minimizing expression (9). To do so, derive (9) w.r.t. R, and root the obtained polynomial of degree 5 in one unknown
2. Compute the new loading factors as in (7) and compute the corresponding error Υ_{new} given by expression (8)
3. Use $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, and $\mathbf{C}^{(new)}$ as starting values for the PARAFAC iteration instead of $\mathbf{A}^{(it-1)}$, $\mathbf{B}^{(it-1)}$, and $\mathbf{C}^{(it-1)}$, and estimate the first loading factors $\widehat{\mathbf{A}}$ as shown in (6)
4. Perform step 3. to estimate each of the remaining loading factors $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{C}}$

One (ALS+ELS) iteration corresponds to about $(F + 8F^2)(JK + IK + IJ) + 3FIJK + 11F^3 + 3F^2 + FIJK + 5^3 = \mathbf{2240}$ multiplications, when $I = 2$, $J = 3$, $K = 3$, and $F = 3$. Details of complexity computation can be found in the appendix. Without ELS, ALS requires $(F + 8F^2)(JK + IK + IJ) + 3FIJK + 11F^3 + 3F^2 = \mathbf{2061}$ multiplications.

When the ALS convergence is quick, say less than 1000 iterations, ELS is not of great help. However, when the convergence is very slow ELS makes the difference as it does not

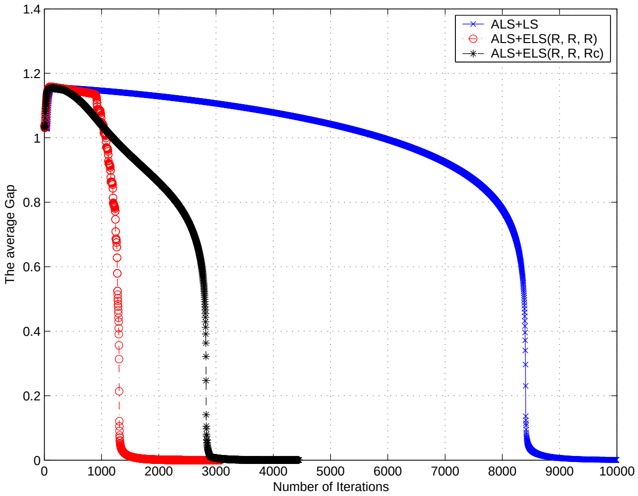

Figure 1: Gap between estimated and actual loading matrix $\mathbf{A}$ using ALS with Line Search and ELS with optimizations (R,R,R) and (R,R,R_c) for $\theta = \pi/60$.

require much more time than ALS alone. When the dimensions of the N-way array are very large and of order $O(I)$, both complexities are equivalent to $O(I^N)$. This is also the case when one dimension I_0 is very large compared to the other ones as both algorithms require $O(I_0)$.

It is worth noting that $\Upsilon^{(new)}$ is always smaller than $\Upsilon^{(it-1)}$ when we use optimal values for R_a, R_b, and R_c as it is the case for the first three optimizations. However, when we use a fixed relaxation factor as in Line Search, $\Upsilon^{(new)}$ can exceed $\Upsilon^{(it-1)}$, which means that the acceleration may fail. This can explain the fact that ELS performs better than ALS with Line Search. Some of the possible ELS optimizations have been implemented. Computer results show that ELS is very attractive.

5. COMPUTER RESULTS

In figures 1 and 2 we report the impact of ELS on ALS in the case of two Factor Degeneracy (2FD), where two of the loading factors are almost collinear such that the contribution of only one of them is considered [2]. The presence of 2FDs slows down the convergence and can lead to the occurrence of intervals called swamps, where the loss function Υ needs a large number of iterations in order to exhibit a very little decrease. We consider the three-way PARAFAC model of expression (3) with :

$$\mathbf{A} = \begin{pmatrix} 1 & cos(\theta) & 0 \\ 0 & sin(\theta) & 1 \end{pmatrix} \tag{10}$$

$$\mathbf{B} = \begin{pmatrix} 3 & \sqrt{2}cos(\theta) & 0 \\ 0 & sin(\theta) & 1 \\ 0 & sin(\theta) & 0 \end{pmatrix} \tag{11}$$

and $\mathbf{C}$ is the identity matrix of size 3×3. The collinearity is controlled through variable θ. We take $\theta = \pi/60$ in figures 1 and 2. The first and the second columns of each of the matrices $\mathbf{A}$ and $\mathbf{B}$ are almost collinear as θ is very close to zero $\theta \simeq 0.052$.

We notice from figures 1 and 2 that ELS speeds up the convergence as the number of necessary iterations decreases from 8664 to about 1300 when using ELS with optimization (R,R,R_c).

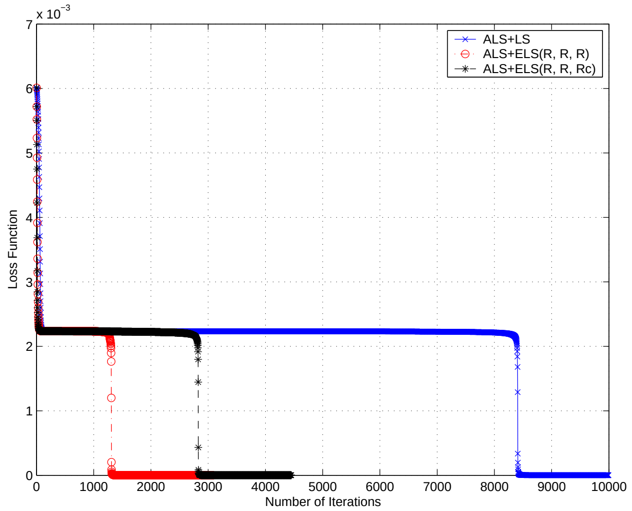

Figure 2: Loss function Υ as a function of the number of iterations for ALS with Line Search, and ELS with optimizations (R,R,R) and (R,R,R_c) for $\theta = \pi/60$.

As already pointed out, we propose an application of ELS to blindly identify a mixing matrix of an UDM. We use ELS to accelerate ALESCAF, the algorithm proposed in [7] for the Blind Channel Identification based on the characteristic function in an UDM. Using notations of [7], ALESCAF leads to a four-way PARAFAC model :

$$\mathbf{T}^{(P \times KP^2)} = \mathbf{A}(\mathbf{D} \odot \mathbf{A} \odot \mathbf{A})^T \tag{12}$$

Tensor $\mathbf{T}$ contains the third derivatives of the joint characteristic function of the observations computed at K points of the grid Ω. Matrix $\mathbf{D}$ is obtained from the independence property of the sources and its entries are defined as :

$$D_{kn} = \psi_n^{(3)}(\sum_q A_{qn} u_q[k]) \tag{13}$$

where $1 \leq k \leq K$ and $1 \leq n \leq N$. $\mathbf{A}$ is the channel matrix of size 2×3 to be identified.

We use the ALS implementation proposed by Andersson and Bro in [1] and replace the Line Search procedure by the six optimizations of ELS shown in figure 3. The three sources are BPSK and we generate an "infinite block" of data by taking all the 2^3 possible combinations of $\{-1, 1\}$. Noise is not taken into account, and we take 5000 as the maximum number of iterations.

In figure 3 we report 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. The figure shows one more time 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.

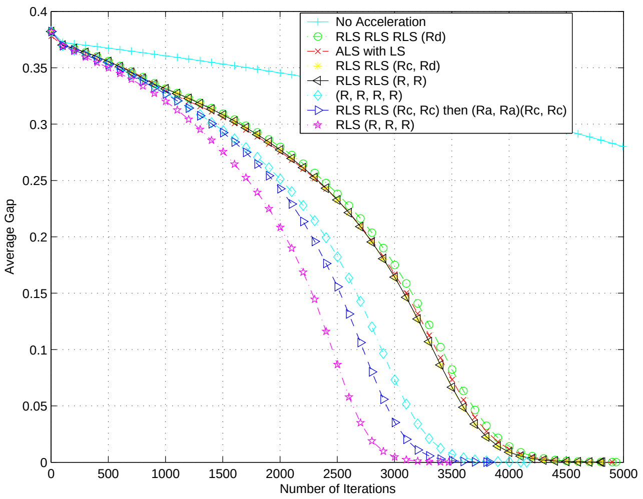

Figure 3: Gap between estimated and actual mixing matrix for $(P,N)=(2,3)$, with algorithm ALESCAF.

6. CONCLUDING REMARKS

We presented ELS, a novel method for accelerating the ALS algorithm used to fit the PARAFAC model, and we compared it with existing methods. Simulations showed that ELS is very attractive especially in the case of 2FDs where it made the number of iterations decrease by a factor 6. The application of ELS to the Blind Channel Identification in an UDM also 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

During one iteration of the ALS algorithm the following operations are performed (we first estimate matrix $\widehat{\mathbf{A}}$) :

1. Compute the Khatri-Rao product to obtain matrix $\mathbf{Z}^a$. This costs FJK multiplications

2. Compute $\mathbf{Z}_a^+$ by reduced SVD of $\mathbf{Z}_a$, which requires $7JKF^2+\frac{11}{3}F^3$ multiplications

3. Estimate the factor loading $\widehat{\mathbf{A}}$ as shown in expression (6), which requires $IJKF+JKF^2+F^2$ multiplications

The previous steps are performed for each of the three loading factors, then the global ALS iteration requires $(F+8F^2)(JK+IK+IJ)+3FIJK+11F^3+3F^2$multiplications. The complexity generated by ELS is $FIJK+O((2N-1)^3)$ when we choose optimization 3 for a three-way PARAFAC model (N=3).

In general, for a N-way array of size $I_1\times I_2\times\ldots\times I_N$ the complexity of ALS when all the dimensions are of the same order $O(I)$ is : $\frac{11}{3}NF^3+NF^2+NFI^N+8NF^2I^{N-1}+NF\frac{I^2(1-I^{N-2})}{1-I}$. Thus ELS requires $FI^N+O((2N-1)^3)$ multiplications. Then, both ALS and (ALS+ELS) requires $O(FI^N)$ multiplications when the dimensions are large and are of the same order $O(I)$. $O(I_0)$ multiplications are needed when one dimensions I_0 is very large compared to the other dimensions.

REFERENCES

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

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

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

[4] J. D. CARROLL, 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.

[5] R. B. CATTELL, "Parallel proportional profiles and other principles for determining the choice of factors by rotation", *Psychometrika*, vol. 9, pp. 267–283, 1944.

[6] 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.

[7] P. COMON, M. RAJIH, "Blind identification of underdetermined mixtures based on the characteristic function", *ICASSP Conference*, Mar 18–23 2005.

[8] 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.

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

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

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

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