

Recherche Linéaire Optimisée : Une Nouvelle Méthode pour Accélérer PARAFAC

Myriam RAJIH, Pierre COMON

Laboratoire I3S

2000 route des Lucioles, BP 121, F-06903 Sophia Antipolis, France

rajih@i3s.unice.fr, comon@i3s.unice.fr

Résumé – L’algorithme des moindres carrés alternés (ALS), utilisé dans la décomposition de PARAFAC, peut nécessiter un très grand nombre d’itérations avant de converger. Cette lenteur de convergence peut découler d’une taille importante des données, d’un mauvais conditionnement, ou de la présence de dégénérescences, etc. Plusieurs méthodes ont été proposées pour accélérer l’algorithme, notamment la compression préalable des données [3], et la recherche linéaire (Line Search LS) [3]. Après une brève description de PARAFAC, nous présenterons une nouvelle méthode pour accélérer ALS, qui, comparée aux méthodes existantes, a permis d’obtenir de meilleurs résultats lors des simulations. Nous proposerons ensuite une application de la nouvelle méthode à l’identification aveugle d’un mélange sous-déterminé (UDM), sachant qu’elle peut être appliquée à n’importe quelle décomposition d’ordre N .

1 Introduction

La décomposition de PARAFAC peut être considérée comme une généralisation de l’analyse de facteurs d’ordre deux aux tenseurs d’ordre N ($N \geq 3$). Elle fut introduite par Harshman en 1970 [8] qui s’est basé sur le principe des Profils Proportionnels et parallèles (PP) proposé par Cattell en 1944 [5]. Ce principe énonce que si deux (ou plusieurs) modèles d’ordre deux sont décrits par le même ensemble de vecteurs composants, tels que seuls les poids changent d’un modèle à l’autre, ces vecteurs composants conduisent alors à un nouveau modèle unique au sens de la rotation [2]. En d’autres termes, supposons que la matrice \mathbf{X}_1 puisse s’écrire sous la forme :

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

\mathbf{a}_f et \mathbf{b}_f ($1 \leq f \leq F$) étant les colonnes des matrices \mathbf{A} et \mathbf{B} respectivement, et supposons qu’une seconde matrice \mathbf{X}_2 puisse s’exprimer à l’aide des mêmes vecteurs composants précédents mais avec des poids différents décrits par c_{2j} (au lieu de c_{1j} précédemment) :

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

Alors nous pouvons ainsi construire un nouveau modèle :

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

qui peut être écrit autrement : $X_{ijk} = \sum_f A_{if} B_{jf} C_{kf}$. Ce modèle trilinéaire est aussi connu sous le nom de DECOM-Position CANonique (CANDECOMP) introduite par Carroll et Chang en 1970 [4]. Le modèle PARAFAC d’ordre trois et ses généralisations aux ordres supérieurs [8] [4][12] sont très populaires en psychométrie et en chimie, mais trouvent aussi des applications en traitement du signal [11] [6] [7]. Contrairement au modèle matriciel, le modèle PARAFAC a l’avantage d’être unique sous les conditions du théorème de Kruskal [10], d’où son importance. Il existe plusieurs algorithmes pour décomposer PARAFAC, le plus répandu étant ALS sur lequel nous nous concentrerons dans toute la suite. La convergence de l’algorithme ALS est parfois extrêmement lente. Cela peut se présenter lorsque la taille des données est assez grande, ou encore lorsque deux colonnes composantes sont presque colinéaires. La compression [3] et la recherche linéaire (LS) [2] sont deux solutions possibles pour remédier à ce problème. Nous nous concentrerons dans cet article sur la solution LS et nous proposerons une nouvelle méthode qui a permis de réduire le nombre d’itérations nécessaires à la convergence de ALS.

2 Modèle et notations

Considérons le modèle PARAFAC d’ordre trois défini par l’expression (1). Ce modèle peut être écrit sous une forme compacte en utilisant le produit de Khatri-Rao \odot (produit Kronecker des colonnes) :

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

où les matrices \mathbf{A} , \mathbf{B} , et \mathbf{C} sont des matrices de taille $I \times F$, $J \times F$, et $K \times F$ respectivement, et $\mathbf{X}^{(I \times JK)}$ est la matrice de taille $I \times JK$ obtenue en dépliant le tenseur \mathbf{X} de taille $I \times J \times K$ suivant le premier mode. ALS consiste à estimer tour à tour une des trois matrices composantes, en minimisant au sens des moindres carrés l’erreur :

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

où $\|\bullet\|_F$ est la norme de Frobenius. \mathbf{B} et \mathbf{C} étant initialisées, l’estimée de \mathbf{A} est donnée par :

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

avec $\mathbf{Z}_a = \mathbf{C} \odot \mathbf{B}$ et $(\cdot)^+$ est la pseudo-inverse de Moore-Penrose. Nous estimons les matrices \mathbf{B} et \mathbf{C} de manière équivalente avec $\mathbf{Z}_b = \mathbf{A} \odot \mathbf{C}$ et $\mathbf{Z}_c = \mathbf{B} \odot \mathbf{A}$, et nous itérons jusqu’à satisfaction du critère de convergence. Typiquement, la convergence est atteinte lorsque l’erreur Υ affiche, entre deux itérations successives, un changement inférieur à un seuil prédéfini, lequel varie selon les données. Pour des données simples, le seuil peut être fixé à 10^{-6} par exemple, mais doit être plus petit pour des données plus complexes, 10^{-10} par exemple.

Il arrive que la convergence nécessite plusieurs milliers d’itérations. Un choix judicieux des valeurs initiales peut quelquefois aider à atteindre le minimum global très rapidement. Cependant, quand le tenseur a de grandes dimensions, ou quand deux facteurs sont presque colinéaires, les techniques d’initialisation ne permettent pas d’aller plus vite. Dans [3], Bro propose de compresser les données dans un espace réduit afin de réduire les dimensions du tenseur, et de là réduire la complexité de ALS. La solution LS a aussi été proposée pour accélérer la convergence [9] [2]. Cette solution est décrite dans la section qui suit.

3 Recherche Linéaire (LS)

Il a été noté lors des simulations que, lorsque la convergence est lente, il existe des cycles de convergence définis par une direction unique. Au cours d'un cycle donné, les facteurs composants évoluent dans la même direction jusqu'à la fin du cycle. Il en est de même pour les cycles suivants. La convergence à l'intérieur d'un cycle peut nécessiter plusieurs itérations. Afin de limiter le nombre d'itérations d'un cycle donné, Harshman et Bro proposent d'extrapoler ce comportement. Ils proposent de prédire la valeur du facteur composant après un certain nombre d'itérations en calculant une sorte de régression linéaire :

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

$\mathbf{A}^{(it-1)}$ est l'estimée de la matrice \mathbf{A} , obtenue à l'itération $(it-1)$ de ALS. A l'itération (it) , la nouvelle matrice $\mathbf{A}^{(new)}$ est utilisée à la place de $\mathbf{A}^{(it-1)}$. $(\mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)})$ définit la direction du cycle. Les matrices \mathbf{B} et \mathbf{C} sont obtenues de façon équivalente en utilisant le même facteur de relaxation R_{LS} . Pour être efficace, l'extrapolation ne doit pas engendrer plus de temps que les itérations qu'elle remplace. C'est justement le cas lorsque R_{LS} est une constante (entre 1.2 et 1.3) [8], ou lorsqu'il est fixé à $it^{1/3}$ [2].

A chaque itération (it) , les nouveaux facteurs sont utilisés pour calculer l'erreur :

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

Si $\Upsilon^{(new)} \geq \Upsilon^{(it-1)}$, cela veut dire que nous sommes allés trop loin lors de l'extrapolation et que R_{LS} est trop grand. Il est alors décrétement de $it^{1/n}$ à $it^{1/(n+1)}$, n étant fixé à 3 au début de la simulation, et les facteurs de l'itération $(it-1)$ sont utilisés au lieu des nouveaux : l'itération (it) est effectuée sans extrapolation. Par contre, si $\Upsilon^{(new)} < \Upsilon^{(it-1)}$ alors l'accélération est accomplie et nous gagnons quelques itérations.

LS est appliquée après un nombre d'itérations de l'algorithme ALS, afin de laisser le temps à ce dernier de se stabiliser. Dans [1], ce nombre est fixé à 6, mais il peut être augmenté selon les données.

La faible valeur de R_{LS} pourrait induire que l'accélération n'est pas efficace. Ceci n'est pas vrai car l'effet de R_{LS} est reconduit d'une itération à une autre, donnant lieu en final à une nette diminution du nombre d'itérations comme le montre la figure 4. Le modèle utilisé dans cette simulation est présenté dans la section 5. Le nombre d'itérations nécessaires à la convergence diminue de 10000 à 4200. Il reste toutefois assez élevé, d'où la nécessité de trouver une nouvelle méthode qui réduise significativement la complexité de ALS.

4 Recherche Linéaire Optimisée (ELS)

L'idée principale derrière ELS (Enhanced Line Search) est de chercher un facteur de relaxation R_{LS} optimal qui conduise à la solution finale du cycle en une seule itération. Définissons pour l'itération (it) , la direction du cycle pour la matrice composante \mathbf{A} comme $\mathbf{G}_a^{(it)} = \mathbf{A}^{(it-1)} - \mathbf{A}^{(it-2)}$. $\mathbf{G}_b^{(it)}$ et $\mathbf{G}_c^{(it)}$ sont définies de manière équivalente. Au lieu de fixer R_{LS} comme dans l'expression (3), nous cherchons le triplet optimal (R_a, R_b, R_c) qui minimise :

$$\Upsilon_{ELS} = \|\mathbf{X}^{(I \times JK)} - (\mathbf{A}^{(it-2)} + R_a \mathbf{G}_a^{(it)}) \left((\mathbf{C}^{(it-2)} + R_c \mathbf{G}_c^{(it)}) \odot (\mathbf{B}^{(it-2)} + R_b \mathbf{G}_b^{(it)}) \right)^T\|_F^2 \quad (5)$$

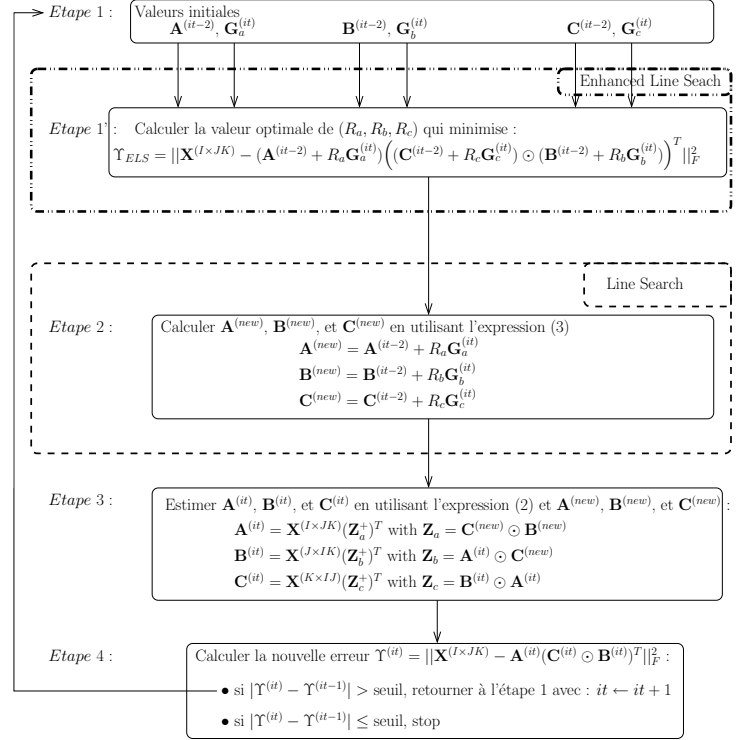


FIG. 1: Etapes de l'algorithme ALS avec la Recherche Linéaire Optimisée.

ELS est appliquée au début de l'algorithme ALS, comme le montre la figure (1), où l'étape 1 correspond à la partie ELS. Le facteur de relaxation R_{LS} n'est plus fixe comme c'était le cas pour LS, mais il est calculé à l'étape 1 de la figure (1), comme étant la valeur optimale qui donne la plus petite erreur Υ_{ELS} .

La solution optimale est obtenue en minimisant conjointement Υ_{ELS} par rapport à R_a , R_b , et R_c . Cela revient à résoudre un système de trois polynômes en trois variables de degré global 5, et engendre une complexité importante. Des solutions à complexité réduite sont obtenues en minimisant Υ_{ELS} par rapport à deux variables seulement, ou encore en prenant le même facteur $R = R_a = R_b = R_c$. Quelques optimisations possibles sont citées ci-après :

1. (R_a, R_b, R_c) donne la solution optimale
2. (R, R, R_c) où nous utilisons le même facteur R pour \mathbf{A} and \mathbf{B} et nous minimisons Υ_{ELS} par rapport aux deux variables R et R_c
3. (R, R, R) où nous utilisons le même facteur pour les trois matrices
4. $R(R_b, R_c)$ où nous utilisons le facteur de relaxation $R = it^{1/3}$ de LS pour la matrice \mathbf{A} , et nous minimisons (5) par rapport à R_b et R_c
5. $R(R, R)$ est le même que $R(R_b, R_c)$ avec $R_b = R_c$
6. $R, R(R)$ où nous optimisons par rapport à R_c uniquement

Pour s'assurer que l'extrapolation est moins coûteuse que les itérations qu'elle remplace nous avons calculé les complexités de ALS et de l'optimisation 3 précédente. A chaque itération ALS, nous exécutons les étapes suivantes :

- Calculer le facteur de relaxation optimal R en minimisant l'expression (5). Pour cela, dériver (5) par rapport à R , et résoudre le polynôme de degré 5 en une inconnue obtenu

- Calculer les nouveaux facteurs $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, et $\mathbf{C}^{(new)}$ en utilisant l'expression (3), et calculer l'erreur correspondante Υ_{new} donnée par l'expression (4)
- Utiliser $\mathbf{A}^{(new)}$, $\mathbf{B}^{(new)}$, et $\mathbf{C}^{(new)}$ comme valeurs initiales pour l'itération PARAFAC au lieu de $\mathbf{A}^{(it-1)}$, $\mathbf{B}^{(it-1)}$, et $\mathbf{C}^{(it-1)}$, et estimer le premier facteur $\hat{\mathbf{A}}$ en utilisant la formule (2)
- Exécuter la troisième l'étape de nouveau pour estimer les autres facteurs $\hat{\mathbf{B}}$ et $\hat{\mathbf{C}}$, en utilisant $\hat{\mathbf{A}}$ et $\mathbf{C}^{(new)}$ pour estimer $\hat{\mathbf{B}}$, et $\hat{\mathbf{A}}$ et $\hat{\mathbf{B}}$ pour estimer $\hat{\mathbf{C}}$

Une itération (ALS+ELS) coûte environ $(F + 8F^2)(JK + IK + IJ) + 3FIJK + 11F^3 + 3F^2 + FIJK + 5^3 = 2240$ multiplications, lorsque $I = 2$, $J = 3$, $K = 3$, et $F = 3$. Le détail des calcul est reporté en annexe. (ALS) seul engendre $(F + 8F^2)(JK + IK + IJ) + 3FIJK + 11F^3 + 3F^2 = 2061$ multiplications, ce qui est équivalent en terme d'ordre de grandeur.

Lorsque ALS convergence rapidement, disons en moins de 1000 itérations, ELS n'est d'aucune utilité. Cependant, lorsque la convergence est très ralentie, ELS fait la différence, et cela à complexité équivalente. Lorsque les dimensions du tenseur d'ordre N sont très grandes et d'ordre $O(I)$ ($F \ll I$), les complexités de (ALS) et de (ALS+ELS) sont équivalentes et égales à $O(FI^N)$. C'est aussi le cas lorsqu'une dimension I_0 est beaucoup plus grande comparée aux autres dimensions puisque les deux algorithmes engendrent une complexité $O(I_0)$.

Il est intéressant de noter que l'erreur $\Upsilon^{(new)}$ est toujours inférieure à $\Upsilon^{(it-1)}$ quand les valeurs optimales de R_a , R_b , et R_c sont utilisées, comme dans les trois premières optimisations. Cependant, lorsque le facteur de relaxation est fixé, comme dans la méthode LS, $\Upsilon^{(new)}$ peut être supérieure à $\Upsilon^{(it-1)}$, ce qui veut dire que l'accélération peut échouer. Ceci explique la meilleure performance de ELS face à LS car l'accélération est garantie à chaque itération. Quelques optimisations possibles de ELS ont été implémentées. Les résultats de simulation montrent que ELS est très attrayante.

5 Résultats

Nous avons reporté sur les figures 2 et 3 l'impact de ELS sur ALS dans le cas de Dégénérescence de 2 Facteurs (2FD), où deux facteurs composants sont presque colinéaires de façon à ce que seule la contribution de l'un d'eux est prise en compte [2]. La présence de 2FD ralentit la convergence et peut engendrer l'apparition d'intervalles de convergence lente (swamps), où l'erreur Υ a besoin d'un très grand nombre d'itérations afin d'enregistrer une très faible diminution.

Considérons le modèle PARAFAC d'ordre trois de l'expression (1) avec :

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

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

et \mathbf{C} est la matrice identité de taille 3×3 . La colinéarité est contrôlée à travers la variable θ qui est fixée à $\pi/60$.

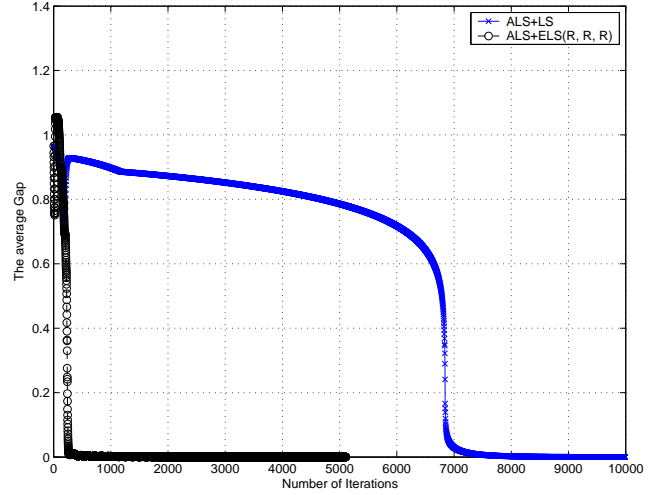


FIG. 2: Gap entre la vraie matrice composante \mathbf{A} et son estimée en utilisant (ALS+LS) et (ALS+ELS) avec l'optimisation (R, R, R) pour $\theta = \pi/60$.

Nous constatons à partir des figures 2 et 3 que ELS accélère la convergence puisqu'elle fait décroître le nombre d'itérations de 7100 à 350 !

Comme signalé dans le résumé, nous proposons une application de ELS à l'identification aveugle d'un UDM. Nous utilisons ELS pour accélérer ALESCAF, l'algorithme proposé en [7] qui consiste en l'identification aveugle d'un UDM basée sur la fonction caractéristique. En utilisant les notations de [7], ALESCAF conduit à un modèle PARAFAC d'ordre quatre :

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

Le tenseur \mathbf{T} contient les dérivées d'ordre trois de la fonction caractéristique conjointe des observations, calculée en K points de la grille Ω . La matrice \mathbf{D} est obtenue grâce à la propriété d'indépendance des sources, et ses entrées sont définies par :

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

où $1 \leq k \leq K$ et $1 \leq n \leq N$. \mathbf{A} est la matrice de canal de taille 2×3 à identifier.

Nous utilisons l'implémentation de ALS proposée par Andersson et Bro dans [1], et nous remplaçons la partie LS par les six optimisations de ELS citées dans la figure 4. Les trois sources sont BPSK et nous générons un bloc infini de données en prenant les 2^3 combinaisons possibles de $\{-1, 1\}$. Le bruit n'est pas pris en compte, et le nombre maximum d'itérations est fixé à 5000.

Nous reportons sur la figure 4 le gap entre la vraie matrice de canal et son estimée pour six optimisations de ELS, et nous les comparons avec (ALS+LS) et ALS non accéléré (sans LS). La figure montre, une fois de plus, que ELS est indispensable pour accélérer la convergence, puisque le nombre d'itérations nécessaires pour atteindre la convergence passe de 4200 avec (ALS+LS), à 2900 avec l'optimisation $R_{LS}(R, R, R)$ de ELS.

6 Conclusions

Nous avons présenté ELS, une nouvelle méthode pour accélérer l'algorithme ALS utilisé dans la décomposition du modèle PARAFAC, et nous l'avons comparée avec les méthodes

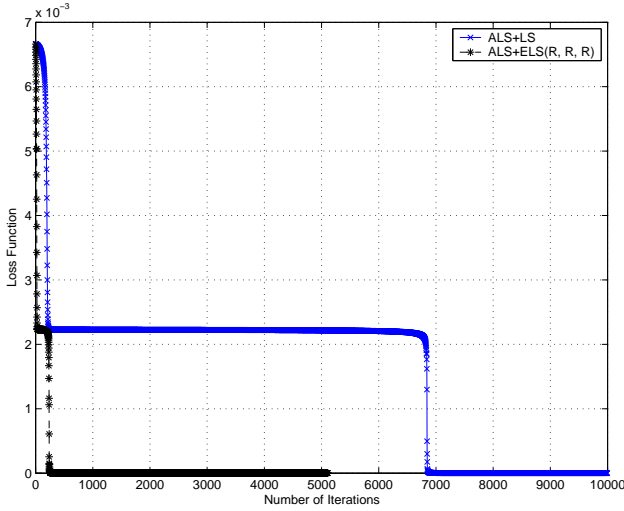


FIG. 3: L'erreur fonction Υ en fonction du nombre d'itérations pour ALS avec LS, et les optimisations (R, R, R) et (R, R, R_c) de ELS, avec $\theta = \pi/60$.

existantes. Les simulations ont montré que ELS est très attrayante, particulièrement dans le cas de 2FD où elle a permis de diviser le nombre d'itérations par 20! L'application de ELS à l'identification aveugle d'un UDM a aussi donné de bons résultats puisque ELS a accéléré la convergence.

Dans les travaux futurs, le bruit sera pris en compte et une plus large classe de blocs de données sera considérée. Nous utiliserons des tenseurs à grandes dimensions afin de confirmer que ELS est aussi attractive dans ce cas.

7 Annexe

Pendant une itération de ALS, les opérations suivantes sont exécutées (la matrice $\hat{\mathbf{A}}$ est estimée en premier) :

1. Calculer le produit Khatri-Rao pour obtenir la matrice \mathbf{Z}_a . Cette opération coûte FJK multiplications
2. Calculer \mathbf{Z}_a^+ par SVD réduite de \mathbf{Z}_a , ce qui nécessite $7JKF^2 + \frac{11}{3}F^3$ multiplications
3. Estimer le facteur $\hat{\mathbf{A}}$ grâce à l'expression (2), ce qui nécessite $IJKF + JK F^2 + F^2$ multiplications

Les étapes précédentes sont exécutées pour l'estimation de chacun des trois facteurs, ce qui implique $(F + 8F^2)(JK + IK + IJ) + 3FIJK + 11F^3 + 3F^2$ multiplications pour une itération complète de ALS. La complexité générée par ELS est $FIJK + O((2N - 1)^3)$ lorsque nous choisissons l'optimisation 3. précédente pour un modèle PARAFAC d'ordre 3 ($N=3$).

En général, pour un tenseur d'ordre N de dimension $I_1 \times I_2 \times \dots \times I_N$, la complexité générée par ALS lorsque toutes les dimensions sont du même ordre $O(I)$ est : $\frac{11}{3}NF^3 + NF^2 + NFI^N + 8NF^2I^{N-1} + NF \frac{I^2(1-I^{N-2})}{1-I}$. Parallèlement, ELS nécessite $FI^N + O((2N - 1)^3)$ multiplications. On en déduit que ALS et (ALS+ELS) sont à complexité équivalente puisqu'ils nécessitent tous les deux $O(FI^N)$ multiplications, lorsque les dimensions sont grandes et de même ordre de grandeur $O(I)$. $O(I_0)$ multiplications sont nécessaires lorsqu'une dimension I_0 est très grande par rapport aux autres dimensions.

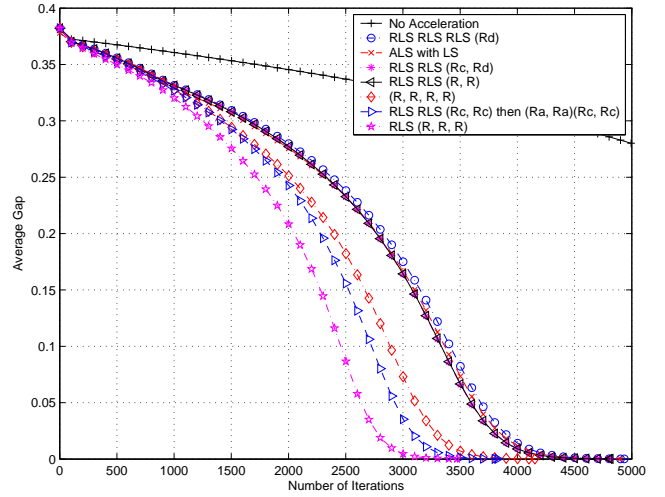


FIG. 4: Gap entre la vraie matrice de canal et son estimée pour $(P, N) = (2, 3)$, en utilisant l'algorithme ALESCAF.

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. RAJHI, "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.