# THÈSE

## Pour obtenir le diplôme de doctorat

**Spécialité: Informatique**

**Préparée au sein de l'Université de Rouen-Normandie**

**Contributions à la décomposition de données multimodales avec des applications en apprentissage de dictionnaires et la décomposition de tenseurs de grande taille**


## Présentée et soutenue par
## Abraham Traoré


| Thèse soutenue publiquement le (date de soutenance) devant le jury composé de | | |
|---|---|---|
| Thirion-Moreau Nadège | Professeur/Université de Toulon | Rapporteur |
| Brie David | Professeur/Université de Lorraine | Rapporteur |
| Honeine Paul | Professeur/Université de Rouen | Examinateur |
| Kadri Hachem | maître de conference/Université Aix-Marseille | Examinateur |
| Rakotomamonjy Alain | Professeur/Université de Rouen | Directeur de thèse |
| Berar Maxime | maître de conference/Université de Rouen | Codirecteur de thèse |

**Thèse dirigée par Rakotomamonjy Alain et Berar Maxime, laboratoire LITIS-Université de Rouen**

# TABLE DES MATIÈRES


**1 Introduction** **9**

1.1 Contexte . . . . . 9

1.2 Motivation du travail de thèse . . . . . 11

1.2.1 Apprentissage de dictionnaires par décomposition de Tucker . . . . 11

1.2.2 Méthodes de décomposition de Tucker pour un tenseur de grande taille . . . . . 12

1.3 Structure du manuscrit . . . . . 13

1.4 Liste des articles issus de la thèse . . . . . 15

**2 Introduction aux tenseurs et à leurs décompositions** **16**

2.1 Généralités sur les tenseurs . . . . . 16

2.1.1 Définitions . . . . . 16

2.1.2 Algèbre relative aux tenseurs . . . . . 18

2.2 Décomposition de Candecomp Parafac . . . . . 21

2.2.1 Généralités . . . . . 21

2.2.2 Rang d'un tenseur . . . . . 23

2.2.3 Unicité de la décomposition *CP* . . . . . 24

2.2.4 Principal algorithme de décomposition . . . . . 25

2.3 Décomposition block term . . . . . 26

2.4 Décomposition de Tucker . . . . . 28

2.4.1 Généralités . . . . . 28

2.4.2 Unicité de la décomposition de Tucker . . . . . 29

2.4.3 Algorithmes classiques . . . . . 30

2.4.3.1 High Order Singular Value Decomposition HOSVD . . . . 30

2.4.3.2 High Order Orthogonal Iterations HOOI . . . . . 31

2.4.4 Variantes importantes de la décomposition de Tucker . . . . . 32
2.4.4.1 Décomposition en ligne de Tucker . . . . . 32
2.4.4.2 Techniques de décomposition de Tucker pour un tenseur de grande taille . . . . . 34
2.4.4.3 Apprentissage de dictionnaires multimodaux par décomposition de tenseurs . . . . . 35
2.5 Outils numériques pour la décomposition de tenseurs . . . . . 36
2.5.1 Matlab . . . . . 36
2.5.2 Python . . . . . 37
2.6 Conclusion . . . . . 37

**3 Apprentissage de dictionnaires par décomposition non-négative de Tucker 39**
3.1 Contexte . . . . . 39
3.2 Décomposition non-négative de matrices . . . . . 40
3.3 De la NMF à la décomposition non-négative de Tucker . . . . . 41
3.4 Cadre de classification . . . . . 43
3.5 Expériences numériques . . . . . 44
3.5.1 Données synthétiques . . . . . 45
3.5.2 DCase2013 data . . . . . 46
3.6 Conclusion . . . . . 47

**4 OTL, un algorithme d'apprentissage en ligne de dictionnaires multimodaux par décomposition de Tucker 49**
4.1 Contexte . . . . . 49
4.2 Apprentissage de dictionnaires multimodaux . . . . . 50
4.2.1 Apprentissage en ligne de dictionnaires multimodaux *OTL* . . . . . 51
4.2.2 Algorithme . . . . . 52
4.2.2.1 *Codage parcimonieux* . . . . . 53
4.2.2.2 *Descente de coordonnées par blocs* . . . . . 54
4.3 Résultat de convergence . . . . . 57
4.4 Extensions . . . . . 59
4.4.1 Extension minibatch . . . . . 59
4.4.2 Contraintes de non-négativité . . . . . 59
4.4.3 Contrainte d'orthogonalité . . . . . 60
4.5 Analyse de la complexité en temps . . . . . 61
4.6 Simulations numériques . . . . . 62

4.6.1 Prédiction de données spatio-temporelles . . . . . 63
4.6.1.1 Schéma de résolution . . . . . 64
4.6.1.2 Base de données . . . . . 64
4.6.1.2.1 Données synthétiques . . . . . 64
4.6.1.2.2 Données réelles . . . . . 65
4.6.1.3 Critère d'évaluation . . . . . 65
4.6.1.4 Résultats . . . . . 65
4.6.1.4.1 Données synthétiques . . . . . 65
4.6.1.4.2 Données réelles . . . . . 66
4.6.2 Apprentissage de dictionnaires multimodaux . . . . . 67
4.6.2.1 Régression sur des données synthétiques . . . . . 67
4.6.2.2 Apprentissage de dictionnaires multimodaux non-négatifs . 68
4.7 Conclusion . . . . . 69

**5 Singleshot, une approche par morceaux pour la décomposition de Tucker 72**

5.1 Contexte . . . . . 72
5.2 Singleshot : une approche par morceaux pour la décomposition de Tucker . 73
5.2.1 Présentation du problème . . . . . 73
5.2.2 Singleshot . . . . . 74
5.2.2.1 Dérivée par rapport à $\mathcal{G}$. . . . . 75
5.2.2.2 Dérivée par rapport à $\mathbf{A}^{(\mathfrak{p})}, \mathfrak{p} \neq \mathfrak{n}$ . . . . . 76
5.2.2.3 Dérivée par rapport à $\mathbf{A}^{(\mathfrak{n})}$. . . . . 76
5.2.3 *Singleshotinexact* . . . . . 77
5.2.4 Complexité pour les sous-tenseurs par rapport à un mode . . . . . 78
5.2.5 Variantes . . . . . 78
5.2.5.1 *Singleshotonline* . . . . . 78
5.2.5.2 Contraintes de non-négativité . . . . . 79
5.3 Analyse théorique . . . . . 79
5.3.1 Résultat de convergence pour *Singleshot* . . . . . 80
5.3.2 Résultat de convergence pour *Singleshotinexact* . . . . . 81
5.4 Simulations numériques . . . . . 82
5.4.1 Données expérimentales avec Singleshot/Singleshotinexact . . . . . 82
5.4.2 Singleshot/Singleshotinexact vs Tensorsketchonline . . . . . 85
5.4.3 Singleshotonline vs Tensorsketchonline . . . . . 86
5.4.4 Vérification des hypothèses . . . . . 88

5.4.4.1 Évolution du gradient par rapport à la variable $\mathbf{A}^{(N)}$ (deuxième hypothèse dans la section 5.2.3 pour *Singleshotinexact*) . . 88
5.4.4.2 Choix de $\mathcal{SET}_k$ pour le calcul du gradient inexact . . . . . 89
5.4.4.3 Influence du nombre de sous-tenseurs . . . . . 89
5.5 Conclusion . . . . . 90

**6 Conclusion et perspectives** **91**
6.1 Conclusion . . . . . 91
6.2 Perspectives . . . . . 92

**Annexes** **95**

**Annexe 1. Divers** **95**
6.3 Équivalence de problèmes pour TDL . . . . . 95
6.4 Sketch d'une matrice . . . . . 95

**Annexe 2. Analyse de convergence pour OTL** **97**
1 Rappel des définitions et notations supplémentaires . . . . . 97
1.1 Rappel des notations et raisonnement pour la convergence . . . . . 97
1.2 Notations supplémentaires . . . . . 98
2 Algorithme pour un nombre infini d'échantillons et d'itération . . . . . 99
3 Hypothèses . . . . . 100
4 Propriétés préliminaires . . . . . 100
5 Convergence par blocs . . . . . 105
5.1 Borne sur la norme de $\mathbf{A}_t^{(n)} - \mathbf{A}_{t+1}^{(n)} : \exists M > 0, \|\mathbf{A}_t^{(n)} - \mathbf{A}_{t+1}^{(n)}\|_F \leq \frac{M}{t}$ . 105
5.2 Le gradient de la fonction $f_n$ est Lipschitzienne . . . . . 106
5.3 La suite $f_n(\mathbf{A}_t^{(n)})$ converge . . . . . 108
5.4 $\mathbf{A}_t^{(n)}$ converge vers un point stationnaires de $f_n$ . . . . . 110
6 Convergence globale . . . . . 110

**Annexe 3. Analyse de convergence pour Singleshot** **113**
1 Diverses propriétés sur les tenseurs nécessaires pour l'analyse . . . . . 113
1.1 Rappel des notations et définitions . . . . . 113
1.2 Propriétés sur les sous-tenseurs . . . . . 114
1.3 Calcul des dérivées . . . . . 116
2 Analyse théorique . . . . . 120
2.1 Définitions et notations supplémentaires . . . . . 120

2.2 Définitions pour *Singleshotinexact* . . . . . 121
2.3 Définitions pour *Singleshot* . . . . . 122
2.4 Hypothèses . . . . . 122
2.4.1 Hypothèses communes . . . . . 123
2.4.2 Hypothèses spécifiques pour *Singleshot-inexact* . . . . . 123
2.4.3 Hypothèses spécifiques pour *Singleshot* . . . . . 123
3 Résultat théorique . . . . . 124
3.1 Convergence de *Singleshotinexact* . . . . . 124
3.2 Convergence de *Singleshot* . . . . . 128

**Annexe 4. Principe de Singleshot pour un sous-tenseur tiré par rapport à tous les modes** **136**

# Remerciements

Je remercie d'abord mes deux responsables de thèse pour la qualité de l'encadrement et la patience dont ils ont fait preuve pendant les trois années ainsi que les suggestions intéressantes qui m'ont aidé à établir une base de travail intéressante, solide et consistante.
Je remercie également les intervenants extérieurs qui ont accepté d'arbitrer ces travaux de recherche pour leurs remarques pointues.
J'adresse un grand merci aux membres du LITIS pour m'avoir chaleureusement accueilli au sein de leur équipe tout en faisant de telle sorte que je me sente le mieux possible tout au long du doctorat. Je tiens à remercier très spécialement le secrétariat du LITIS pour toute leur aide sur le plan administratif.
Enfin, je remercie toutes les personnes qui m'ont aidé de près ou de loin dans cette aventure.

# Abréviations et acronymes

**Tucker** : Tucker Decomposition

**Tucker2** : Tucker2 Decomposition

**CP** : Candecomp Parafac Decomposition

**HOSVD** : High Order Singular Value Decomposition

**HOOI** : High Order Orthogonal Iteration

**NNTD** : Non Negative Tucker Decomposition

**BTD** : Block Tensor Decomposition

**NMF** : Nonnegative Matrix Factorization

**PCA** : Principal Component Analysis

**SVD** : Singular Value Decomposition

**ADMM** : Alternating Direction Method of Multipliers

**BCD** : Block Coordinate Descent

**SVM** : Support Vector Machine

**TDL** : Tensor Dictionary Learning

**OTL** : Online Tensor Learning

$\mathbf{I}_N$ : ensemble des entiers consécutifs de 1 à N avec 1 et N inclus

$\mathbf{I}_{N\neq m}$ : ensemble des entiers consécutifs de 1 à N avec 1 et N inclus et m exclu

$\mathcal{X}$ : les tenseurs sont notés par des lettres majuscules calligraphiques

$\mathbf{A}$ : les matrices sont notées par des lettres majuscules en gras

$\mathbf{A}_{i,j}$ : représente la composante $(i, j)$ de la matrice $\mathbf{A}$

$\mathbf{A}_{i,:}$ : représente la $i^{ème}$ ligne de la matrice $\mathbf{A}$

$\mathbf{A}_{:,j}$ : représente la $j^{ème}$ colonne de la matrice $\mathbf{A}$

$\otimes$ : produit de **Kronecker** de deux matrices

$\otimes_{m\in\mathbf{I}_N}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \otimes .... \otimes \mathbf{A}^{(N)}$ : produit de **Kronecker** des N matrices $\mathbf{A}^{(1)}, ..\mathbf{A}^{(N)}$

$\otimes_{m\in\mathbf{I}_{N\neq n}}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \otimes ..\mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)}.. \otimes \mathbf{A}^{(N)}$ : produit de **Kronecker** des $N-1$ matrices $\mathbf{A}^{(1)}, .., \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, .., \mathbf{A}^{(N)}$

$\odot$ : produit de **Khatri-Rao** de deux matrices

$\odot_{m\in\mathbf{I}_N}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \odot \mathbf{A}^{(2)} \odot .. \odot \mathbf{A}^{(N)}$ : produit de **Khatri-Rao** des N matrices $\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}$

$\odot_{m\in\mathbf{I}_{N\neq n}}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \odot .. \odot \mathbf{A}^{(n-1)} \odot \mathbf{A}^{(n+1)}.. \odot \mathbf{A}^{(N)}$ : produit de **Khatri-Rao** des $N-1$ matrices $\mathbf{A}^{(1)}, ..., \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, ..\mathbf{A}^{(N)}$

$\mathtt{rang}$ : rang d'une matrice

$\mathtt{Trace}$ : trace d'une matrice carrée

$\star$ : produit de **Hadamard**

$\dagger$ : pseudo-inverse

$\langle , \rangle$ : produit scalaire (pour les matrices et les tenseurs)

$\#$ : cardinal d'un ensemble

$\mathsf{o}$ : produit dyadique de deux vecteurs

$\mathsf{Prox}_{\mathsf{f}}$ : opérateur proximal d'une fonction $\mathsf{f}$

$\equiv$ : équivalent

$\mathbb{R}$ : ensemble des nombres réels

$\mathbb{R}_+$ : ensemble des nombres réels non-négatifs

$\mathbb{C}$ : ensemble des nombres complexes

$\mathbb{N}$ : ensemble des entiers naturels

**MSE** : erreur moyenne quadratique

**RMSE** : écart quadratique moyen.

**MAP** : précision moyenne

**AE** : erreur d'approximation

# CHAPITRE 1

# INTRODUCTION

À l'aune d'une époque où on acquiert des données de taille de plus en plus importante, avec une complexité croissante de leur nature résultant de la multiplication des sources d'acquisition, le développement de nouveaux outils permettant de modéliser correctement ces spécificités devient une nécessité. Dans ce travail, on s'intéresse à un problème particulier qui est la décomposition de données multimodales qui sont des données caractérisées par plus de deux modes : par exemple, une image hyperspectrale en présente trois car pour la décrire dans un système de coordonnées, il faut définir un repère comportant trois axes (un pour la longueur, un pour la largeur, un pour les bandes spectrales). Les techniques de décomposition auxquelles on s'intéresse peuvent être vues comme une généralisation de certaines méthodes établies sur les matrices.

## 1.1 Contexte

La décomposition de matrices joue un rôle prépondérant dans le traitement de données issues de diverses disciplines via des techniques populaires telles que la décomposition en valeurs singulières (**SVD**) ou la décomposition non-négative de matrices (**NMF**) couvrant des applications telles que la classification, le débruitage d'images ou la reconstruction d'images avec des pixels manquants (aussi appelé **Inpainting**).

Cependant, la nature de certaines données est en contradiction avec la représentation matricielle, dans le sens où elles sont plutôt représentées par des outils mathématiques dont la définition est plus complexe à formaliser que celle des matrices. Par exemple, les données electroencephalographiques présentent divers modes : elles mesurent typiquement la valeur du signal pour un pas de temps, une fréquence et un électrode donné, donc pour

les représenter, il faut définir un système de coordonnées muni d'un repère à trois axes : un pour les fréquences, un pour les pas de temps et enfin, un dernier pour les électrodes. Partant de ce constat, certains auteurs se sont intéressés aux questions suivantes :

— Perd-on de l'information en réorganisant des données multidimensionnelles dans des matrices ?

— Si oui, que gagne-t-on à utiliser des techniques de décomposition de tenseurs plutôt que des méthodes de décomposition de matrices ?

D'un point de vue intuitif, la réponse à la première question ne peut être que oui et pour la seconde, la réponse naturelle semble être une meilleure interpretabilité. Des travaux effectués ont permis de répondre à ces questions en confirmant les réponses intuitives que l'on peut avoir.

Les deux décompositions de tenseurs les plus utilisées sont la décomposition de **Tucker** et la décomposition de **Parafac** (encore appelée **Canonique Polyadique** ou **CP**) proposée par **Frank Hitchcock** en 1927 [52], [53] et la décomposition de **Tucker** proposée par **Ledyard Tucker** en 1963 [117]. Ces deux décompositions demeurent les plus courantes avec le développement d'algorithmes efficaces pour les mettre en oeuvre en pratique.

L'objectif de la décomposition de **Tucker** est d'approximer un tenseur par le produit d'un autre tenseur avec un certain nombre de matrices. Associée à des contraintes d'orthogonalité, cette décomposition peut être vue comme le pendant multidimensionnel de la décomposition en valeurs singulières [74].

L'idée de la décomposition **CP** est d'approximer un tenseur donné par une somme de tenseurs de rang 1. Cette décomposition peut être vue comme la version multidimensionnelle de la meilleure approximation de rang $k$ d'une matrice.

L'intérêt pour les tenseurs est motivé par certains travaux expérimentaux qui ont prouvé, dans divers contextes, que traiter des données multidimensionnelles au moyen de tenseurs plutôt que de matrices donne un meilleur résultat aussi bien pour des tâches de régression que de classification.

Par exemple, dans [63], Yong-Deok et al. ont proposé un algorithme itératif inspiré des règles multiplicatives pour la décomposition de matrice et ont montré sur un problème de débruitage d'images, que la décomposition non-négative de **Tucker** donnait une meilleure qualité de reconstruction que la décomposition de matrices non-négative. Shengqi Tan et al. [114] ont proposé un algorithme de décomposition de tenseurs inspiré de la méthode **ADMM** [114] et ont établi que pour un problème de reconstruction tomographique, la

décomposition de tenseurs fournit un meilleur résultat que la décomposition de matrices. Zheng et al. [133] ont introduit un système de recommandation à base de tenseurs et ont prouvé que leur modèle capture des interactions plus complexes comparées à celles inférées via la décomposition non-négative de matrices **NMF**. Yoshii et al. [125] ont proposé une approche tensorielle pour la séparation de sources et démontré qu'elle donne de meilleurs résultats que **NMF**. Tamir Hazan et al. [51] ont présenté une approche pour la décomposition non-négative d'un tenseur d'ordre 3 et illustré, via un problème de classification d'image, que leur approche produisait de meilleurs scores de classification par rapport à la stratégie consistant à traiter les données sous forme matricielle. Gaoyun An et al. [4] ont introduit une approche itérative pour la décomposition non-négative de tenseur sous contrainte de faible rang et établi son efficacité par rapport à la **NMF** sur un problème de clustering.

D'un point de vue théorique, divers travaux ont aussi été entrepris et des garanties théoriques ont été établies en terme d'unicité de la décomposition aussi bien pour la décomposition de **Tucker** sous des contraintes d'orthogonalité [74] que la décomposition **CP** [66].

## 1.2 Motivation du travail de thèse

Dans ce travail, on s'intéresse à deux problématiques particulières :

— L'apprentissage de dictionnaires multimodaux par décomposition de **Tucker** ;

— La mise en place de méthodes de décomposition de **Tucker** efficientes en terme de gestion de mémoire et/ou en temps de calcul ;

Notre choix de nous restreindre à la décomposition de **Tucker** est motivé par le fait qu'elle fait partie des décompositions les plus courantes et peut facilement être adaptée pour effectuer d'autres décompositions (par exemple la décomposition **CP** est un cas particulier de la décomposition de **Tucker**).

### 1.2.1 Apprentissage de dictionnaires par décomposition de Tucker

Récemment, certains auteurs ont commencé à s'intéresser aux liens entre décomposition de tenseurs et apprentissage de dictionnaires. L'idée est d'apprendre des dictionnaires multimodaux qui correspondent à des dictionnaires par modes. Tan et al. [114] ont mis en oeuvre une technique permettant d'inférer des dictionnaires multimodaux via une

décomposition non-négative de tenseurs. Stevens et al. [110] ont introduit une méthode non-paramétrique pour l'inférence aussi bien des facteurs latents que du rang.
Les limites majeures de ces méthodes reposent principalement sur leur manque de généricité [13], c'est-à-dire qu'elles sont difficiles à adapter pour incorporer des contraintes ou imposent la connaissance préalable de certains facteurs [13].

### 1.2.2 Méthodes de décomposition de Tucker pour un tenseur de grande taille

Les approches standard pour la décomposition de **Tucker**, reposant sur des algorithmes itératifs inspirés de techniques bien établies dans le cadre de la décomposition de matrices et dont l'application se fait via une reformulation de la fonction de coût (pour le problème tensoriel) par une opération de matricisation, peuvent avoir une complexité en espace élevée (par exemple, les schémas de résolution numérique dans certains travaux [103] pour la décomposition non-négative de **Tucker** matérialisent le produit de **Kronecker** de $\mathsf{N}-\mathsf{1}$ matrices de saturation, $\mathsf{N}$ étant l'ordre du tenseur).
Certaines approches ont été proposées pour contourner ce problème. Elles se classent principalement en trois groupes :

- Méthodes sur le principe "diviser pour mieux régner" : ces méthodes considèrent un tenseur fixe et infèrent les facteurs latents en traitant séquentiellement des sous-ensembles de composantes de manière séquentielle plutôt que d'utiliser toute la donnée de manière simultanée. Des approches ont été proposées dans divers travaux [77], [13], [84].

  Les limitations majeures de ces approches concernent la garantie de convergence que bon nombre d'entre elles ne fournissent pas et le problème de vitesse de convergence n'est pas traité.
- Méthodes à coût intermédiaire faible : l'idée de ces approches repose sur l'utilisation simultanée (sans division préalable) de la donnée, mais avec des astuces permettant de réduire la complexité aussi bien en temps qu'en espace des calculs intermédiaires. Pour cette classe d'approches, des méthodes aussi bien déterministes qu'aléatoires ont été proposées [99], [40].

  Bien que ces méthodes réduisent le coût en espace des calculs, elles ne sont pas applicables lorsque toute la donnée ne tient pas en mémoire.
- Méthodes dites en ligne : ces méthodes considèrent un tenseur qui croît séquentiellement au fil du temps par rapport à un ou plusieurs modes. Hiroyuki et al. [59] ont proposé un schéma de résolution en ligne pour une tâche de complétion traduit via

un problème de minimisation sur une variété de Stiefel. Une seconde approche proposée dans [126] permet de résoudre des problèmes de prédiction de données spatio-temporelles.

Les limitations principales de ces méthodes reposent sur leur manque de généricité (c'est-à-dire la difficulté d'incorporer des contraintes) ou des garanties de convergence sous de fortes contraintes (par exemple, la méthode proposée dans [59] donne une garantie de convergence sous la contrainte dite de faible rang).

**Conclusion** : pour résumer, nous rencontrons globalement pour les classes de problèmes considérées les limitations suivantes :

- Manque de généricité (difficulté de prendre en compte certaines contraintes) ;
- Nécessité d'une connaissance à priori de certains facteurs ;
- Résultats de convergence sous des conditions contraignantes ou manque de vitesse de convergence.

L'objet de nos travaux est de proposer de nouvelles approches pour contourner les limites présentées ci-dessus et a donné lieu aux publications acceptées dont la liste est donnée dans la section 1.4 :

- Non-negative Tensor Dictionary Learning : **ESANN-2018**
- Online multimodal dictionary learning : **Conférence pour l'apprentissage statistique-2018 et Neurocomputing-2019**
- Singleshot, a scalable Tucker tensor decomposition : **Neurips-2019**

## 1.3 Structure du manuscrit

Le manuscrit est structuré de la manière suivante :

1. **Chapitre 2 : Introduction aux tenseurs et à leurs décompositions**.
   Ce chapitre décrit l'état de l'art sur la décomposition de tenseurs et introduit les notions et opérations classiques ainsi que quelques outils numériques en Matlab et Python.
2. **Chapitre 3** : **Apprentissage de dictionnaires par décomposition non-négative de Tucker**.
   Un des problèmes importants pour l'apprentissage de dictionnaires par décomposition de matrices non-négative **NMF** est la modélisation efficace des motifs temporels dans un problème d'apprentissage de caractéristiques pour des données présentant une structure séquentielle (d-dimension+temps), par exemple les spectrogrammes. Dans ce chapitre, nous présentons une approche simple basée sur la décomposition de tenseurs et interprétable en terme d'atomes de dictionnaires pour atténuer ce

problème dans le cadre d'une tâche de classification, nos contributions étant : la clarification du lien entre l'apprentissage de dictionnaire par **NMF** et la décomposition de **Tucker**, la proposition d'un problème de minimisation pour l'apprentissage de motifs temporels par décomposition non-négative de **Tucker** et enfin la mise en place de simulations numériques pour prouver le caractère compétitif des caractéristiques de classification apprises par décomposition de **Tucker** par rapport à la **NMF** associée au pooling.

3. **Chapitre 4 : OTL, un algorithme d'apprentissage en ligne de dictionnaires multimodaux par décomposition de Tucker**

   Nous proposons une approche en ligne pour l'apprentissage de dictionnaires multimodaux. La motivation sous-jacente de ce travail est la mise en place d'une méthode pour décomposer une séquence de tenseurs dans un environnement dynamique (les tenseurs sont acquis au fil du temps). Notre méthode permet d'inférer une base génératrice de facteurs latents en traitant les échantillons par petit nombre plutôt que de traiter toute la séquence en une seule fois, d'où la dénomination en ligne. Elle repose sur la descente de coordonnées par blocs, la descente de gradient ainsi que le calcul récursif du gradient avec les contributions suivantes : on introduit une technique en ligne générique (permettant d'incorporer facilement des contraintes classiques telles que la non-négativité ou la parcimonie) pour l'apprentissage de dictionnaires via la décomposition de **Tucker** avec un résultat de convergence (vers un point stationnaire du problème d'optimisation proposé) et des expériences numériques pour évaluer son efficacité.

4. **Chapitre 5 : Singleshot, une approche par morceaux pour la décomposition de Tucker**.

   Nous proposons un nouvel algorithme pour la décomposition de **Tucker** efficient en terme de complexité en espace. Étant donné un tenseur $\mathcal{X}$, l'algorithme proposé, nommé **Singleshot**, permet d'effectuer l'inférence des facteurs latents en traitant un seul sous-tenseur de $\mathcal{X}$ à la fois. Le principe de notre approche repose sur le calcul récursif des gradients et la mise à jour cyclique des variables, chaque mise à jour étant définie via une seule itération de descente de gradients. Une variante de **Singleshot**, nommée **Singleshotinexact** et nécessitant moins de temps de calcul est aussi introduite. Une troisième méthode, déduite de **Singleshotinexact** et désignée par **Singleshotonline**, est présentée pour la décomposition d'un tenseur qui croît par rapport à tous les modes sous la contrainte de la passe unique (chaque sous-ensemble de donnée est traité une seule fois). Enfin, pour **Singleshot** et **Singleshotinexact**, on établit une vitesse de convergence (vers un point stationnaire

pour **Singleshot** et un minimiseur pour **Singleshotinexact**) et l'n prouve l'efficacité des algorithmes mis en place via des simulations numériques sur des données réelles.

5. **Chapitre 6 : Conclusion et perspectives**
   Dans ce chapitre, nous faisons une résumé du travail effectué et présentons d'éventuelles pistes d'amélioration.

## 1.4 Liste des articles issus de la thèse

Les articles publiés dans cette thèse sont les suivants :

— Abraham Traoré, Maxime Berar, Alain Rakotomamonjy. Non-negative Tensor Dictionary Learning. European Symposium on Artificial Neural Networks, pages : 585-590, 2018

— Abraham Traoré, Maxime Berar, Alain Rakotomamonjy. Décomposition en ligne de tenseurs, Conférence pour l'apprentissage statistique, 2018

— Abraham Traoré, Maxime Berar, Alain Rakotomamonjy. Online multimodal dictionary learning. Neurocomputing, 2019

— Abraham Traoré, Maxime Berar, Alain Rakotomamonjy. Singleshot, a scalable Tucker tensor decomposition. Neural Information Processing Systems, 2019

CHAPITRE 2

# INTRODUCTION AUX TENSEURS ET À LEURS DÉCOMPOSITIONS

## 2.1 Généralités sur les tenseurs

Dans cette section, on introduit les définitions, les opérations classiques ainsi que certains modèles de décomposition standard pour les tenseurs. Quelques outils numériques sont aussi présentés. Les références sur lesquelles s'appuie ce chapitre sont les travaux effectués dans [66], [37], [78], [60], [64],[107].

### 2.1.1 Définitions

La définition suivante donne la définition formelle d'un tenseur qui peut être vu comme une généralisation des matrices avec la possibilité d'avoir plus de deux indices, le nombre d'indices utilisés pour le décrire étant l'ordre du tenseur.

**Définition 1** *: tenseur d'ordre N*
*Un tenseur d'ordre N est un tableau multidimensionnel défini sur le produit tensoriel de N espaces vectoriels, chaque espace vectoriel étant doté de son propre système de coordonnées.*

Les exemples les plus classiques de tenseurs sont : les matrices qui peuvent être interprétées comme des tenseurs d'ordre 2, les vecteurs qui peuvent être interprétés comme des tenseurs d'ordre 1 et les scalaires qui peuvent être interprétés comme des tenseurs d'ordre 0.

La définition suivante présente un sous-tenseur qui est un tenseur construit à partir d'un autre tenseur. Certains de nos travaux s'appuient sur cette notion.

**Définition 2** *: sous-tenseur par rapport à un mode*
*Soit* $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$ *un tenseur d'ordre N. Le* $i_n^{ième}$ *sous-tenseur* $\mathcal{X}$ *par rapport au mode n, noté* $\mathcal{X}_{i_n}^n \in \mathbb{R}^{I_1 \times .. \times I_{n-1} \times 1 \times I_{n+1} \times .. \times I_N}$ *est le tenseur d'ordre* $N$ *dérivé de* $\mathcal{X}$ *en fixant l'indice de numéro n à* $i_n$*, c'est-à-dire :*

$$(\mathcal{X}_{i_n}^n)_{i_1,..,i_{n-1},1,i_{n+1},..,i_N} = \mathcal{X}_{i_1,..,i_N}$$

La figure 2.1 donne pour un tenseur d'ordre 3, les sous-tenseurs par rapport aux premier, second et troisième modes qui peuvent donc être vus comme des matrices, couramment appelées coupes par certains auteurs [66].

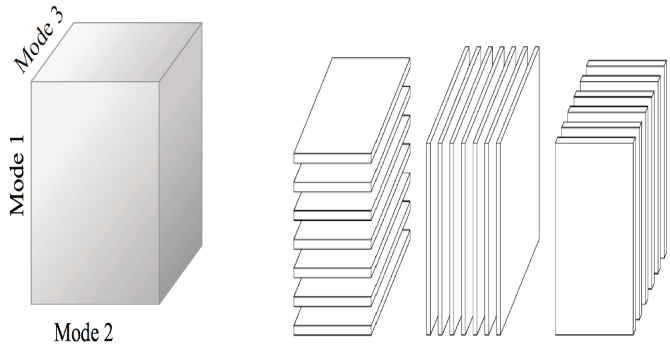


FIGURE 2.1 – De gauche à droite pour un tenseur d'ordre 3 [66] : sous-tenseurs par rapport au mode 1 (coupes horizontales), sous-tenseurs par rapport au mode 2 (coupes latérales), sous-tenseurs par rapport au mode 3 (coupes verticales)

La définition suivante introduit la notion de tenseur diagonal qui intervient dans certaines décompositions classiques.

**Définition 3** *: tenseur diagonal*
*Un tenseur diagonal est un tenseur cubique* $\mathcal{X} \in \mathbb{R}^{I \times ... \times I}$ *(c'est-à-dire un tenseur pour lequel tous les espaces vectoriels le définissant ont exactement la même dimension) dont les éléments extra-diagonaux (ceux qui ne sont pas de la forme* $\mathcal{X}_{i,...,i}, 1 \leq i \leq I$*) sont tous égaux à zéro :*

$$\mathcal{X}_{i_1,..i_N} = \begin{cases} \lambda_i & si\ i = i_1 = ...i_N \\ 0 & sinon. \end{cases}$$

### 2.1.2 Algèbre relative aux tenseurs

Dans cette section, nous introduisons les opérations algébriques qui seront utilisées dans le cadre de notre travail : la matricisation, le produit tenseur-matrice ainsi que les produits matriciels (**Kronecker**, **Khatri-Rao**, **Hadamard**, produit dyadique) qui interviennent dans certaines décompositions.

La matricisation est l'opération qui consiste à réarranger tous les éléments d'un tenseur dans une matrice et se fait par rapport à l'un des modes du tenseur.

**Définition 4** *: matricisation par rapport au mode n*
*Soit* $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$ *un tenseur d'ordre N. La forme matricielle de* $\mathcal{X}$ *par rapport au mode n est une matrice notée* $\mathbf{X}^{(n)}$ *ayant comme nombre de lignes* $I_n$ *et comme nombre de colonnes* $\prod_{k=1,k\neq n}^{N} I_k$ *et dont les entrées sont définies comme suit :*

$$\mathbf{X}^{(n)}_{i_n,j} = \mathcal{X}_{i_1,..,i_N} \text{ avec } j = \sum_{k=1,k\neq n}^{N} i_k \prod_{m=k+1}^{N} I_m \tag{2.1}$$

La figure 2.2 illustre la matricisation par rapport au premier mode d'un tenseur d'ordre 3.

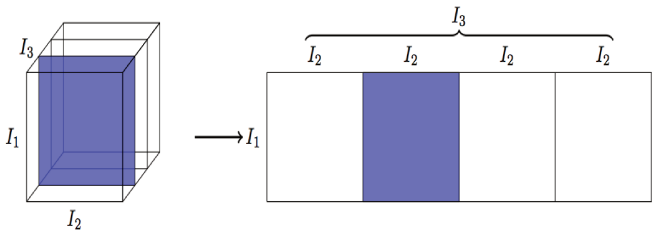


FIGURE 2.2 – Matricisation par rapport au premier mode pour un tenseur d'ordre 3[64]

Il existe plusieurs conventions pour l'opération de matricisation. La convention que nous considérons dans ce manuscrit est celle présentée par l'équation (2.1) et sur laquelle repose par exemple l'outil *Tensorly* [68].

Les définitions 5,6,7,8 introduisent des produits matriciels classiques sur lesquels reposent certaines décompositions courantes.

**Définition 5** *: Produit de* ***Hadamard***
*Considérons deux matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{M\times N}$*. Le produit de Hadamard de* $\mathbf{A}$ *et* $\mathbf{B}$*, noté* $\mathbf{A} \star \mathbf{B} \in \mathbb{R}^{M\times N}$*, est défini par* $(\mathbf{A} \star \mathbf{B})_{m,n} = \mathbf{A}_{m,n}\mathbf{B}_{m,n}$.

*Pour plusieurs matrices* $\mathbf{A}^{(m)} \in \mathbb{R}^{M\times N}, 1 \leq m \leq N$*, on considéra les notations suivantes :*

$$\star_{m\in\mathbf{I}_N}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \star .... \star \mathbf{A}^{(N)} \in \mathbb{R}^{M\times N} \tag{2.2}$$

$$\star_{m\in\mathbf{I}_{N\neq n}}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \star .. \star \mathbf{A}^{(n-1)} \star \mathbf{A}^{(n+1)} .. \star \mathbf{A}^{(N)} \in \mathbb{R}^{M\times N} \tag{2.3}$$

**Définition 6** *: Produit de* ***Kronecker***
*Le produit de* ***Kronecker*** *de deux matrices* $\mathbf{A} \in \mathbb{R}^{M\times N}$ *et* $\mathbf{B} \in \mathbb{R}^{K\times L}$*, noté* $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{MK\times NL}$*, est défini par :*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}_{1,1}\mathbf{B}, & & , \mathbf{A}_{1,N}\mathbf{B} \\ & & \\ & & \\ \mathbf{A}_{M,1}\mathbf{B}, & & , \mathbf{A}_{M,N}\mathbf{B} \end{bmatrix}$$

*avec* $\mathbf{A}_{m,n}, 1 \leq m \leq M, 1 \leq n \leq N$ *représentant les composantes de la matrice* $\mathbf{A}$*.*
*Cette définition peut se réécrire sous la forme d'indices par :*

$$(\mathbf{A} \otimes \mathbf{B})_{(m-1)K+k,(n-1)L+l} = \mathbf{A}_{m,n}\mathbf{B}_{k,l}$$

On notera le produit de **Kronecker** de plusieurs matrices $\mathbf{A}^{(m)} \in \mathbb{R}^{I_m\times J_m}, 1 \leq m \leq N$ par :

$$\otimes_{m\in\mathbf{I}_N}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \otimes .... \otimes \mathbf{A}^{(N)} \in \mathbb{R}^{(I_1\times ...\times I_N)\times(J_1\times ...\times J_N)} \tag{2.4}$$

$$\otimes_{m\in\mathbf{I}_{N\neq n}}\mathbf{A}^{(m)} = \mathbf{A}^{(1)} \otimes .. \otimes \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} \otimes ... \otimes \mathbf{A}^{(N)} \in \mathbb{R}^{(I_1\times ...\times I_{n-1}\times I_{n+1}\times ...\times I_N)\times(J_1\times ...\times J_{n-1}\times J_{n+1}\times ...\times J_N)} \tag{2.5}$$

**Définition 7** *: Produit de* ***Khatri-Rao***
*Soient* $\mathbf{A} \in \mathbb{R}^{N\times M}, \mathbf{B} \in \mathbb{R}^{K\times M}$ *deux matrices. Le produit de* ***Khatri-Rao****, noté* $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{KN\times M}$*, correspond à la matrice obtenue en faisant le produit de* ***Kronecker*** *des colonnes deux à deux, c'est-à-dire :*

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{A}_{:,1} \otimes \mathbf{B}_{:,1}, ..., \mathbf{A}_{:,M} \otimes \mathbf{B}_{:,M}]$$

*avec* $[,]$ *étant l'opérateur qui empile plusieurs vecteurs dans le sens de la colonne,* $\mathbf{A}_{:,j}$ *(respectivement* $\mathbf{B}_{:,j}$*) représentant la* $j^{ème}$ *colonne de la matrice* $\mathbf{A}$ *(respectivement* $\mathbf{B}$*) et* $\otimes$ *le produit de* ***Kronecker*** *(définition 6).*

Le produit de **Khatri-Rao** de plusieurs matrices $\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times R}, 1 \leq m \leq N$ sera alternativement noté par :

$$\odot_{m \in \mathbf{I}_N} \mathbf{A}^{(m)} = \mathbf{A}^{(1)} \odot .... \odot \mathbf{A}^{(N)} \in \mathbb{R}^{(I_1 \times .. \times I_N) \times R} \tag{2.6}$$

$$\odot_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(m)} = \mathbf{A}^{(1)} \odot .. \odot \mathbf{A}^{(n-1)} \odot \mathbf{A}^{(n+1)} .. \odot \mathbf{A}^{(N)} \in \mathbb{R}^{(I_1 \times .. \times I_{n-1} \times I_{n+1} \times ... I_N) \times R} \tag{2.7}$$

**Définition 8** *: produit dyadique de deux vecteurs*
*Le produit dyadique de vecteurs* $\mathbf{a} \in \mathbb{R}^N, \mathbf{b} \in \mathbb{R}^M$*, noté* $\mathbf{aob} \in \mathbb{R}^{N \times M}$*, est défini par :*

$$(\mathbf{aob})_{i,j} = \mathbf{a}_i \mathbf{b}_j$$

*Le produit dyadique de* $N$ *vecteurs* $\mathbf{a}_p \in \mathbb{R}^{I_p}, 1 \leq p \leq N$ *sera noté* $\mathbf{a}_1 o .... o \mathbf{a}_N \in \mathbb{R}^{I_1 \times ... \times I_N}$.

Certains algorithmes pour la décomposition de tenseurs reposent sur le produit tenseur-matrice qui est une opération dont la forme matricielle est tributaire des opérateurs de **Kronecker** et de **Khatri-Rao** présentés via les définitions 6 et 7.

**Définition 9** *: multiplication d'un tenseur par plusieurs matrices*
*Le produit d'un tenseur* $\mathcal{G} \in \mathbb{R}^{J_1 \times .. \times J_N}$ *avec des matrices* $\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times J_m}$*, noté par* $\mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} ... \times_N \mathbf{A}^{(N)}$*, donne un tenseur* $\mathcal{X} \in \mathbb{R}^{I_1 \times ..... \times J_N}$ *défini par :*

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} ... \times_N \mathbf{A}^{(N)} \iff \mathcal{X}_{i_1,..,i_N} = \sum_{j_1,..,j_N} \mathcal{G}_{j_1,..,j_N} \mathbf{A}^{(1)}_{i_1,j_1} ... \mathbf{A}^{(N)}_{i_N,j_N} \tag{2.8}$$

Pour des raisons de simplicité d'écriture, on adoptera pour le produit tenseur-matrice, les notations alternatives suivantes :

$$\mathcal{G}_{\times_1} \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} .. \times_N \mathbf{A}^{(N)} = \mathcal{G} \times_{p \in \mathbf{I}_{n-1}} \mathbf{A}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q \in \mathbf{I}_N^{n+1}} \mathbf{A}^{(q)} = \mathcal{G} \times_{m \in \mathbf{I}_N} \mathbf{A}^{(m)} \tag{2.9}$$

$$\mathcal{G}_{\times_1} \mathbf{A}^{(1)} .. \times_{n-1} \mathbf{A}^{(n-1)} \times_{n+1} \mathbf{A}^{(n+1)} .. \times_N \mathbf{A}^{(N)} = \mathcal{G} \times_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(m)} \tag{2.10}$$

NB : la notation $\mathbf{I}_N^{n+1}$ correspond à l'ensemble des entiers de $n+1$ et $N$ ($n+1$ et $N$ compris).
À la définition par indices du produit tenseur-matrice donnée par l'équation (2.8), on peut en substituer une basée sur les formes matricielles des tenseurs $\mathcal{X}$ et $\mathcal{G}$ donnée par (voir [65] pour les détails du calcul algébrique conduisant à cette expression) :

$$\mathbf{X}^{(n)} = \mathbf{A}^{(n)} \mathbf{G}^{(n)} \left( \mathbf{A}^{(1)} \otimes ... \otimes \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} \otimes ... \otimes \mathbf{A}^{(N)} \right)^T = \mathbf{A}^{(n)} \mathbf{G}^{(n)} \left( \otimes_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(m)} \right)^T \tag{2.11}$$

avec $\otimes$ correspondant au produit de **Kronecker** de deux matrices (voir la définition 6). Dans le cas d'un produit avec une seule matrice par rapport au mode $n$, la formule se simplifie :

$$\mathbf{X}^{(n)} = \mathbf{A}^{(n)}\mathbf{G}^{(n)}$$

Si $\mathcal{G} \in \mathbb{R}^{R\times\ldots\times R}$ est un tenseur diagonal dont toutes les composantes $\{\mathcal{G}_{r,\ldots,r}, 1 \leq r \leq R\}$ sont égales à $1$, la forme matricielle de $\mathcal{X}$ se réécrit sous la forme (voir [65] pour les détails du calcul algébrique) :

$$\mathbf{X}^{(n)} = \mathbf{A}^{(n)}\left(\mathbf{A}^{(1)} \odot \ldots\mathbf{A}^{(n-1)} \odot \mathbf{A}^{(n+1)}\ldots\mathbf{A}^{(N)}\right)^T = \mathbf{A}^{(n)}\left(\odot_{m\in\mathbf{I}_N\neq n}\mathbf{A}^{(m)}\right)^T \tag{2.12}$$

avec $\odot$ correspondant au produit de **Khatri-Rao** de deux matrices (voir la définition 7).

Nous rappelons ci-dessous les normes standard que sont les normes $\ell_1$ et de Frobenius.

**Définition 10** *: normes de Frobenius et $\ell_1$*
*Les normes de Frobenius et $\ell_1$ d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1\times..\times I_N}$, notées respectivement par $\|\mathcal{X}\|_F$ et $\|\mathcal{X}\|_1$, sont définies par :*

$$\|\mathcal{X}\|_F = \left(\sum_{1\leq i_1\leq I_1,..,1\leq i_N\leq I_N} \mathcal{X}^2_{i_1,..,i_N}\right)^{\frac{1}{2}}, \|\mathcal{X}\|_1 = \sum_{1\leq i_1\leq I_1,..,1\leq i_N\leq I_N} |\mathcal{X}_{i_1,..,i_N}|$$

## 2.2 Décomposition de Candecomp Parafac

### 2.2.1 Généralités

Cette décomposition a été introduite indépendamment par Hitchcock [52],[53] et par Cattell [20],[21]. Elle a ensuite été popularisée dans la communauté de la psychométrie par Carroll et Chang [18] ainsi que Harshman[49] avec des applications dans diverses disciplines telles que le traitement du signal [120], [121], [46].
Étant donné un tenseur $\mathcal{X} \in \mathbb{R}^{I_1\times\ldots\times I_N}$ d'ordre $N$, l'objectif est de le décomposer comme une somme finie de tenseurs de rang 1, c'est à dire :

$$\mathcal{X} = \sum_{r=1}^{R} \underbrace{\mathbf{A}^{(1)}_{:,r} o \ldots o \mathbf{A}^{(N)}_{:,r}}_{\text{tenseur de rang 1}} \tag{2.13}$$

Les vecteurs $\mathbf{A}^{(n)}_{:,r}$ représentent les colonnes de la matrice $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R}$ et l'opérateur $o$ le produit dyadique (voir la définition 8) de deux vecteurs.
Dans la suite, on se référera aux matrices $\mathbf{A}^{(n)}$ comme des matrices de saturation. Il est à

noter que cette décomposition existe toujours dans le sens où tout tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ peut s'écrire comme somme finie de tenseurs de rang 1 $\left\{ \mathbf{A}^{(1)}_{:,r} o \ldots o \mathbf{A}^{(N)}_{:,r} \right\}_{1 \leq r \leq R}$ [96],[32]. Un exemple de décomposition pour un tenseur d'ordre 3 est illustré par la figure 2.3

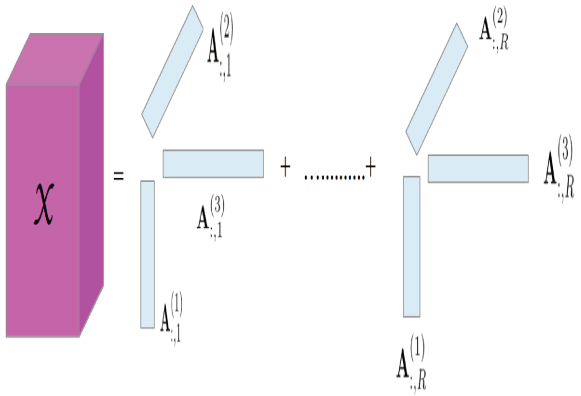


FIGURE 2.3 – Décomposition **CP** d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}, \mathcal{X} \approx \sum_{r=1}^{R} \mathbf{A}^{(1)}_{:,r} o \mathbf{A}^{(2)}_{:,r} o \mathbf{A}^{(3)}_{:,r}$ : $\mathbf{A}^{(1)}_{:,i} \in \mathbb{R}^{I_1}, \mathbf{A}^{(2)}_{:,i} \in \mathbb{R}^{I_2}, \mathbf{A}^{(3)}_{:,i} \in \mathbb{R}^{I_3}$

L'équation (2.13) peut être reformulée de diverses manières :

— Reformulation par composantes donnée par l'équation (2.14)

Par définition du produit dyadique comme produit des composantes (voir la définition 8), on a :

$$\mathcal{X}_{i_1,..,i_N} = \sum_{r=1}^{R} \mathbf{A}^{(1)}_{i_1,r} .. \mathbf{A}^{(N)}_{i_N,r} \tag{2.14}$$

— Reformulation en terme de produit tenseur-matrice

Par définition du produit tenseur-matrice (voir définition 9), la décomposition **CP** peut se réécrire sous la forme suivante (pour une illustration, voir figure 2.4) :

$$\mathcal{X} = \mathcal{I} \times_{n \in I_N} \mathbf{A}^{(n)}, \mathcal{I} \in \mathbb{R}^{R \times \ldots \times R}, \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R} \tag{2.15}$$

Dans cette équation, le tenseur $\mathcal{I}$ représente le tenseur identité défini par :

$$\mathcal{I}_{i_1,..,i_N} = \begin{cases} 1 & \text{si } i_1 = \ldots = i_N \\ 0 & \text{sinon.} \end{cases} \tag{2.16}$$

L'équivalence entre l'équation (2.15) et (2.13) résulte du fait que les expressions (2.15) et (2.14) sont équivalentes par définition du produit tenseur-matrice.

— Reformulation matricielle donnée par (2.17) : elle provient de calculs algébriques

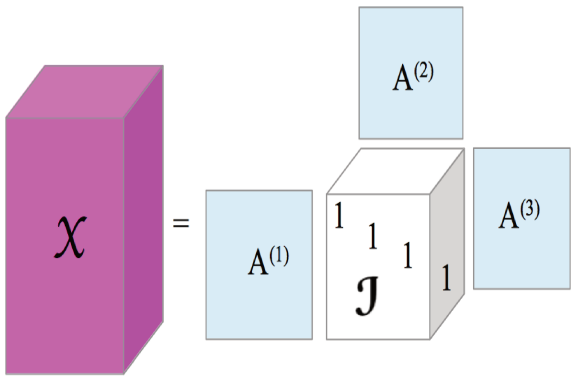


FIGURE 2.4 – Décomposition **CP** d'un tenseur $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ : $\mathbf{A}^{(1)} \in \mathbb{R}^{I\times R}, \mathbf{A}^{(2)} \in \mathbb{R}^{J\times R}, \mathbf{A}^{(3)} \in \mathbb{R}^{K\times R}$

(voir la définition 9 et les travaux effectués dans [65])

$$\mathbf{X}^{(n)} = \mathbf{A}^{(n)} \left(\odot_{m\in \mathbf{I}_{N\neq n}} \mathbf{A}^{(m)}\right)^{T}, 1 \leq n \leq N \tag{2.17}$$

Dans l'équation (2.17), l'opérateur $\odot$ correspond au produit de **Khatri-Rao**.

### 2.2.2 Rang d'un tenseur

Le plus petit entier $R$ tel que l'égalité $\mathcal{X} = \sum_{r=1}^{R} \mathbf{A}_{:,r}^{(1)} o ... o \mathbf{A}_{:,r}^{(N)}$ ou une de ses formes équivalentes (décomposition exacte) soit vérifiée est appelée le rang du tenseur. Le rang d'un tenseur est différent de celui d'une matrice. Quelques différences importantes entre le rang d'une matrice et d'un tenseur sont les suivantes :

— Un tenseur peut avoir deux valeurs de rang différentes suivant que ces entrées soient considérées à valeurs dans $\mathbb{R}$ (ensemble des nombres réels) ou dans $\mathbb{C}$ (ensemble des nombres complexes). Par exemple, considérons un tenseur $\mathcal{X} \in \mathbb{R}^{2\times 2\times 2}$ dont les premier et second sous-tenseurs (voir la définition 2) par rapport au mode 3 sont définies par :

$$\mathcal{X}^3_{:,:,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathcal{X}^3_{:,:,2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Le tenseur $\mathcal{X}$ est de rang 3 sur $\mathbb{R}$ et de rang 2 sur $\mathbb{C}$ [71]

— Une autre différence majeure résulte du fait qu'on ne peut pas calculer de manière simple le rang d'un tenseur quelconque, ce problème étant NP-difficile [66].

— La valeur du rang peut excéder les dimensions du tenseur [34].

— Plusieurs notions alternatives du rang ont été introduites telles que :

1. Le rang typique : qui se définit comme un entier tel que l'égalité décrite par

l'équation (2.13) soit vérifiée avec une probabilité strictement positive. Sa valeur n'est connue que dans des cas particuliers et un tenseur peut avoir plusieurs rangs typiques sur $\mathbb{R}$.

2. Le rang générique : se définit comme l'unique entier $R$ tel que l'équation (2.13) soit vérifiée avec une probabilité 1 [31]. Son existence est toujours assurée dans un corps qui est algébriquement clos [32]. Dans $\mathbb{C}$, il existe un unique rang typique qui correspond au rang générique [32].

3. Le rang symétrique qui correspond au rang d'un tenseur symétrique défini par :

$$\mathrm{rank}_S = \min \left\{ R : \mathcal{X} = \sum_{r=1}^{R} \underbrace{a_r o \ldots\ldots o a_r}_{N\text{fois}} \in \mathbb{C}^{I \times \ldots\ldots \times I} \right\} \tag{2.18}$$

$\mathrm{rank}_S$ admet une formule analytique établie par Alexander et Hirschowitz [2] et donnée par :

$$\mathrm{rank}_S = \lceil \frac{\binom{I+N-1}{N}}{I} \rceil \tag{2.19}$$

avec I étant la dimension commune des modes, $\lceil a \rceil$ le plus petit entier supérieur ou égal $a$ et $\binom{M}{k}$ le nombre de sous-ensembles à $k$ éléments d'un ensemble à $M$ éléments.

La formule donnée par l'équation (2.19) est valable pour $(N, I) \notin \{(3,5),(4,3),(4,4),(4,5)\}$ et pour les couples précédents, elle doit être augmentée de 1.

Par définition du rang comme étant le plus petit nombre de tenseurs de rang 1 dont la somme donne le tenseur d'intérêt, on a : $R \leq \mathrm{rank}_S$ avec $R$ représentant le rang. Il a été prouvé par Comon et al. [33] que cette inégalité est une égalité dans certains cas particuliers.

### 2.2.3 Unicité de la décomposition *CP*

L'unicité de la décomposition se définit par la combinaison de tenseurs de rang 1 dont la somme est égale au tenseur que l'on cherche à décomposer. On ne tient donc pas compte des indéterminations de permutation et d'échelle. L'indétermination de permutation fait référence au fait que les tenseurs de rang 1 dans la somme peuvent être réordonnés de manière arbitraire :

$$\mathcal{X} = \mathcal{I} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} = \mathcal{I} \times_1 \mathbf{A}^{(1)}\mathbf{P} \times_2 \mathbf{A}^{(2)}\mathbf{P} \times_3 \mathbf{A}^{(3)}\mathbf{P} \tag{2.20}$$

$\mathbf{P} \in \mathbb{R}^{R \times R}$ est une matrice de permutation et $\mathcal{I} \in \mathbb{R}^{R \times R \times R}$ représente le tenseur identité défini par l'équation (2.16).

L'indétermination d'échelle fait référence au fait que pour $\alpha, \beta, \theta \neq 0$ tel que $\alpha\beta\theta = 1$, on a :

$$\mathcal{X}_{i,j,k} = \sum_{r=1}^{R} \mathbf{A}^{(1)}_{i,r}\mathbf{A}^{(2)}_{j,r}\mathbf{A}^{(3)}_{k,r} = \sum_{r=1}^{R} \left(\alpha\mathbf{A}^{(1)}_{i,r}\right)\left(\beta\mathbf{A}^{(2)}_{j,r}\right)\left(\theta\mathbf{A}^{(3)}_{k,r}\right) \tag{2.21}$$

Considérons un tenseur $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ d'ordre 3 tel que : $\mathcal{X} = \sum_{r=1}^{R} \mathbf{A}^{(1)}_{:,r} o \mathbf{A}^{(2)}_{:,r} o \mathbf{A}^{(3)}_{:,r}$. Une condition suffisante d'unicité de cette décomposition est donnée par l'équation (2.22) :

$$k_{\mathbf{A}^{(1)}} + k_{\mathbf{A}^{(2)}} + k_{\mathbf{A}^{(3)}} \geq 2 \times R + 2 \tag{2.22}$$

Ce résultat d'unicité est dû à Kruskal [72], [70]. Dans l'équation (2.22), $k_{\mathbf{A}^{(1)}}$ représente l'entier maximal tel que $k_{\mathbf{A}^{(1)}}$ colonnes de $\mathbf{A}^{(1)}$ sont linéairement indépendantes. Les entiers $k_{\mathbf{A}^{(2)}}, k_{\mathbf{A}^{(3)}}$ sont définis de manière similaire. Ce résultat a été étendu par Sidiropoulos et Bro [105] pour un tenseur d'ordre quelconque N :

$$\sum_{n=1}^{N} k_{\mathbf{A}^{(n)}} \geq 2 \times R + (N - 1) \tag{2.23}$$

La condition suffisante donnée par (2.23) est aussi nécessaire pour $R \in \{2, 3\}$, mais pas pour $R > 3$ [14]. Pour d'autres conditions, consulter [66].

### 2.2.4 Principal algorithme de décomposition

En supposant R fixé, la décomposition **CP** s'effectue en pratique en minimisant l'écart entre le tenseur qu'on cherche à décomposer et son approximation donnée par les facteurs latents qu'on cherche à inférer, ce qui revient à résoudre le problème de minimisation suivant :

$$\min_{\mathbf{A}^{(1)}\in\mathbb{R}^{I_1\times R},..,\mathbf{A}^{(N)}\in\mathbb{R}^{I_N\times R}} \ell(\mathcal{X}, \left\{\mathbf{A}^{(n)}\right\}_{1\leq n\leq N}) \tag{2.24}$$

La fonction de coût $\ell$ dans l'équation (2.24) est généralement mesurée par la divergence de Frobenius définie par :

$$\ell(\mathcal{X}, \left\{\mathbf{A}^{(n)}\right\}_{1\leq n\leq N}) = \frac{1}{2}\|\mathcal{X} - \sum_{r=1}^{R} \mathbf{A}^{(1)}_{:,r} o .... o \mathbf{A}^{(N)}_{:,r}\|_F^2 \tag{2.25}$$

Le problème qui consiste à trouver la meilleure approximation de rang R d'un tenseur est un problème mal posé [66] : ceci est principalement dû au fait que l'ensemble des tenseurs de rang au plus $R > 1$ n'est pas un ensemble fermé.

Cependant, certaines approches ont été proposées pour effectuer la décomposition **CP**

en pratique, l'une des plus courantes étant la minimisation alternée. Cette approche fut initialement proposée par Carroll et Chang [17] ainsi que Harshman [50]. Le principe consiste à minimiser la fonction de coût donnée par l'équation (2.25) par rapport à une variable quand toutes les autres sont fixées et ainsi de suite jusqu'à ce qu'un critère d'arrêt préalablement défini soit vérifié. Cette technique est résumée par l'algorithme suivant :

***Algorithme 1 : CP-ALS***

**Entrées** : tenseur à décomposer $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$, Rang $R$

iter $\leftarrow 0$ ;

Initialiser $\mathbf{A}^{(n)}_{iter}$

**Répéter**

  **Pour** $1 \leq n \leq N$ **faire**

$$\mathbf{A}^{(n)} = \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{X}^{(n)} - \mathbf{A}^{(n)} \left(\odot_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(m)}\right)^T \|_F^2$$

  **Fin Pour**

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

Le problème de minimisation présenté dans **Algorithme 1** admet une solution analytique donnée par [66] :

$$\mathbf{A}^{(n)} = \mathbf{X}^{(n)} \left(\mathbf{A}^{(N)} \odot .. \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n)} ... \odot \mathbf{A}^{(1)}\right) \mathbf{V}^{\dagger} \tag{2.26}$$

avec $\mathbf{V} = \mathbf{A}^{(1)T}\mathbf{A}^{(1)} \star .. \star \mathbf{A}^{(n-1)T}\mathbf{A}^{(n-1)} \star \mathbf{A}^{(n+1)T}\mathbf{A}^{(n+1)} \star .. \star \mathbf{A}^{(N)T}\mathbf{A}^{(N)}$, $\star$ le produit de **Hadamard** (voir la définition 5), $\mathbf{V}^{\dagger}$ représentant la pseudo-inverse [45] de $\mathbf{V}$.

## 2.3 Décomposition block term

L'idée sous-jacente à cette décomposition est similaire à la décomposition **CP** dans le sens où on cherche à écrire un tenseur comme une somme finie de produits tenseur-matrice, c'est-à-dire une somme d'approximations de **Tucker** (voir section 2.4). Précisément, étant donné un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times ... \times I_N}$, cette approche (illustrée par la figure 2.5), consiste à

effectuer l'approximation suivante :

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathcal{G}_r \times_{n \in \mathbf{I}_N} \mathbf{A}_r^{(n)} \tag{2.27}$$

avec : $\mathcal{G}_r \in \mathbb{R}^{J_{1,r} \times \ldots \times J_{N,r}}$ étant les tenseurs noyaux et $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_{n,r}}, 1 \leq n \leq N$ les matrices de saturation.

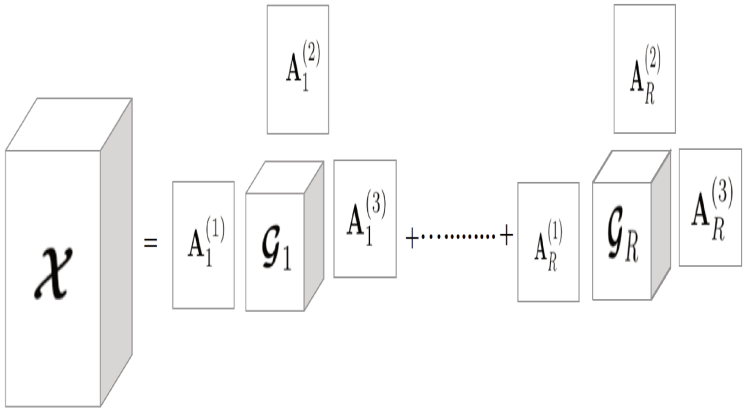


FIGURE 2.5 – Décomposition block term pour un tenseur $\mathcal{X}$ d'ordre 3 : $\mathcal{X} = \sum_{r=1}^{R} \mathcal{G}_r \times_1 \mathbf{A}_r^{(1)} \times_2 \mathbf{A}_r^{(2)} \times_3 \mathbf{A}_r^{(3)}$

Cette décomposition trouve des applications dans le filtrage de signaux [3], la transmission sans fil [87]. Le schéma de résolution général de ce problème est le suivant (pour plus d'information, consulter [92]) :

— Détermination des tenseurs noyaux $\mathcal{G}_r$ optimaux ;

— Résolution de problèmes de minimisation par rapport à une matrice $\mathbf{A}^{(m)}$ de saturation quand toutes autres sont fixées.

Dans la suite, on se focalisera principalement sur la décomposition de **Tucker**, ce choix étant motivé par le fait qu'elle peut être facilement adaptée pour effectuer d'autres décompositions populaires telles que la décomposition **CP** et présente diverses applications.

## 2.4 Décomposition de Tucker

### 2.4.1 Généralités

Cette décomposition, introduite dans [117], peut être vue comme le pendant multidimensionnel de la décomposition en composantes principales **PCA**. Étant donné un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times ... \times I_N}$ et N entiers $J_1, .., J_N$, l'objectif de la décomposition de **Tucker** est d'inférer des facteurs latents $\mathcal{G} \in \mathbb{R}^{J_1 \times ... \times J_N}$ et $\left\{\mathbf{A}^{(n)}\right\}_{1 \leq n \leq N}$, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ dont le produit permet d'approximer $\mathcal{X}$ (une figure illustrative de la décomposition de **Tucker** d'un tenseur d'ordre 3 est donnée par la figure 2.6.), c'est-à-dire :

$$\mathcal{X} \approx \mathcal{G} \times_{n \in \mathbf{I}_N} \mathbf{A}^{(n)} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} ... \times_N \mathbf{A}^{(N)} \tag{2.28}$$

Les diverses reformulations de l'équation (2.28) sont données par :

— Reformulation par composantes : elle provient de la définition du produit tenseur-matrice (voir la définition 9).

$$\mathcal{X}_{i_1,..,i_N} = \sum_{j_1,..,j_N} \mathcal{G}_{j_1,..,j_N} \mathbf{A}^{(1)}_{i_1,j_1} ... \mathbf{A}^{(N)}_{i_N,j_N} \tag{2.29}$$

— Reformulation matricielle : elle résulte de la définition 9 et de calculs algébriques effectués dans [65] :

$$\mathbf{X}^{(n)} \approx \mathbf{A}^{(n)} \mathbf{G}^{(n)} \left(\otimes_{m \in I_{N \neq m}} \mathbf{A}^{(m)}\right)^T = \mathbf{A}^{(n)} \mathbf{G}^{(n)} \left(\mathbf{A}^{(1)} \otimes .. \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} .. \otimes \mathbf{A}^{(N)}\right)^T \tag{2.30}$$

Dans l'équation (2.30), l'opérateur $\otimes$ correspond au produit de **Kronecker** de deux matrices.

Dans la littérature, le tenseur $\mathcal{G}$, qui est du même ordre que $\mathcal{X}$, est appelé **tenseur noyau** et les matrices $\mathbf{A}^{(n)}$ sont nommées **matrices de saturation**. Les matrices de saturation peuvent être vues comme les composantes principales relatives à chaque mode (le nombre de composantes principales pour le mode $n$ étant $J_n$) et les entrées de $\mathcal{G}$ comme le degré d'interaction entre les composantes [66]. Si $J_n < I_n, \forall n$, le tenseur $\mathcal{G}$ peut s'interpréter comme le version compressée de $\mathcal{X}$. Le $N-\mathtt{uplet}$ $\{J_n\}_{1 \leq n \leq N}$ est généralement appelé rang

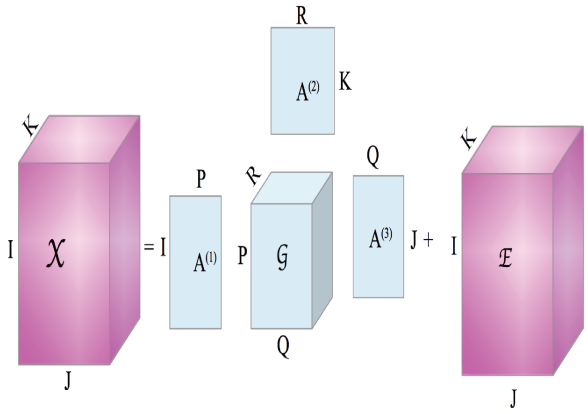


FIGURE 2.6 – Décomposition de **Tucker** d'un tenseur $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

de la décomposition.

Dans la suite, on se référera à $\mathcal{G}$ et $\left\{\mathbf{A}^{(n)}\right\}$ respectivement comme le **noyau** et les **matrices de saturation** sauf mention explicite du contraire.

A noter qu'un nombre important de travaux sur la décomposition de **Tucker** imposent des contraintes d'orthogonalité (ce choix étant probablement motivé par la propriété d'unicité) sur les matrices de saturation et exploitent le fait que le tenseur noyau soit une version compressée de $\mathcal{X}$ [124],[13] (cette dernière contrainte est généralement appelée "contrainte de faible rang"). Cependant, même si ces deux contraintes sont très courantes, certaines approches proposent d'autres types de contraintes comme par exemple la non-négativité des matrices de saturation et du tenseur noyau [30].

### 2.4.2 Unicité de la décomposition de Tucker

De manière générale, la décomposition de **Tucker** n'est pas unique. En effet, considérons $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathcal{G} \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ deux tenseurs et $\left\{\mathbf{A}^{(n)}\right\}_{1 \leq n \leq N}$ N matrices tels que :

$$\mathcal{X} \approx \mathcal{G} \times_{m \in \mathbf{I}_N} \mathbf{A}^{(m)}$$

Pour N matrices inversibles $\left\{\mathbf{B}^{(n)}\right\}_{1 \leq n \leq N}$ d'inverses respectifs $\left\{(B^{(m)})^{-1}\right\}_{1 \leq m \leq N}$, on a [66] :

$$\mathcal{X} \approx \left(\mathcal{G} \times_{m \in \mathbf{I}_N} \mathbf{B}^{(m)}\right) \times_{m \in \mathbf{I}_N} \left(\mathbf{A}^{(m)} \left(\mathbf{B}^{(m)}\right)^{-1}\right)$$

Pour contourner ce problème, on peut imposer certaines contraintes telles que l'orthogonalité, qui assure l'unicité des sous espaces générés par les facteurs latents [29].

### 2.4.3 Algorithmes classiques

Les deux principaux algorithmes utilisés pour la mise en œuvre de la décomposition de **Tucker** reposent sur la contrainte d'orthogonalité des matrices de saturation et considèrent une divergence de Frobenius. Plus précisément, ils reposent sur le problème de minimisation suivant [66] :

$$\min_{\mathcal{G}\in\mathbb{R}^{J_1\times\ldots\times J_N},\mathbf{A}^{(n)}\in\mathbb{R}^{I_n\times J_n},\mathbf{A}^{(n)T}\mathbf{A}^{(n)}=\mathbf{I}} \ell\left(\mathcal{X},\mathcal{G},\left\{\mathbf{A}^{(n)}\right\}_{1\leq n\leq N}\right) \tag{2.31}$$

avec :

$$\ell\left(\mathcal{X},\mathcal{G},\left\{\mathbf{A}^{(m)}\right\}_{1\leq m\leq N}\right) = \frac{1}{2}\|\mathcal{X}-\mathcal{G}\times_{m\in\mathbf{I}_N}\mathbf{A}^{(m)}\|_F^2 \tag{2.32}$$

#### 2.4.3.1 High Order Singular Value Decomposition HOSVD

Cette procédure de décomposition a été introduite en 1966 par Tucker [117]. Sa forme originelle est décrite pour la décomposition d'un tenseur d'ordre 3.

L'idée de **HOSVD** est de chercher les composantes qui capturent au mieux les variations par rapport à chaque mode. En pratique, l'approche consiste pour chaque mode $n$, à la détermination de la forme matricielle, puis à la déduction de la matrice de saturation via une décomposition en valeurs singulières. Ce principe est résumé par l'algorithme suivant :

***Algorithme 2 : High Order Singular value Decomposition HOSVD***

**Entrées** : tenseur à décomposer $\mathcal{X}\in\mathbb{R}^{I_1\times..\times I_N}$, $\{J_n\}_{1\leq n\leq N}$ : dimensions du noyau ;

**Sorties :** $\mathcal{G},\mathbf{A}^{(1)},....,\mathbf{A}^{(N)}$

**Pour** $1\leq n\leq N$ **faire**

- $\mathcal{Y}=\mathcal{X}\times_{m\in I_{N\neq n}}\mathbf{A}^{(m)T}$
- $\mathbf{U},\mathbf{S},\mathbf{V}=\mathbf{SVD}\left(\mathbf{Y}^{(n)}\right)$
- $\mathbf{A}^{(n)}\leftarrow$ les $J_n$ vecteurs propres associés aux $J_n$ plus grandes valeurs propres

**Fin Pour**

$\mathcal{G}\leftarrow\mathcal{X}\times_{m\in I_N}\mathbf{A}^{(m)T}$

#### 2.4.3.2 High Order Orthogonal Iterations HOOI

Cet algorithme est basé sur la minimisation alternée (minimisation de la fonction de coût par rapport une variable quand toutes les autres sont fixées) avec contraintes d'orthogonalité. Les origines de la minimisation alternée pour la décomposition de **Tucker** remontent à des travaux effectués en 1980 par Kroonenberg et De Leeuw [69] qui proposèrent une méthode alternée pour la décomposition de **Tucker** d'un tenseur d'ordre 3. Cette méthode fut ensuite étendue pour un tenseur d'ordre $N > 3$ par Kapteyn, Neudecker, et Wansbeek [57]. Ces travaux précédents furent ensuite améliorés par De Lathauwer, De Moor, et Vandewalle [75], notamment au niveau du calcul des vecteurs propres associés aux plus grandes valeurs propres résultant de la décomposition en valeurs singulières des formes matricielles. La méthode **HOOI** est résumée dans l'algorithme suivant :

***Algorithme 3 : High Order Orthogonal Iterations HOOI***

**Entrées** : tenseur à décomposer $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$, $\{J_n\}_{1 \leq n \leq N}$
**Sortie** : $\mathcal{G}, \mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}$
**Répéter**
  **Pour** $1 \leq n \leq N$ **faire**
    $\mathcal{Y} \leftarrow \mathcal{X} \times_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(m)T}$
    $\mathbf{U}, \mathbf{S}, \mathbf{V} = \mathbf{SVD}\left(\mathbf{Y}^{(n)}\right)$
    $\mathbf{A}^{(n)} \leftarrow$ les $J_n$ vecteurs propres des plus grandes valeurs propres
  **Fin Pour**
**jusqu'à ce que** (un critère d'arrêt soit vérifié)
$\mathcal{G} \leftarrow \mathcal{X} \times_{m \in I_N} \mathbf{A}^{(m)T}$

**Remarque 1** . *Les travaux effectués dans [74] donnent une explication convaincante à la question "pourquoi la décomposition de **Tucker** avec des contraintes d'orthogonalité peut être considérée comme le pendant multidimensionnel de la décomposition en valeurs singulières pour les matrices **SVD**" : pour un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$, ils assurent l'existence des facteurs latents $\mathcal{G} \in \mathbb{R}^{I_1 \times ... \times I_N}$ (associé à des contraintes d'orthogonalité par sous-tenseur, c'est-à-dire le produit scalaire de deux sous-tenseurs différents par rapport à n'importe quel mode est égal à zéro) et $\mathbf{A}^{(n)}$ matrices orthogonales (telles que $\mathbf{A}^{(n)T}\mathbf{A}^{(n)} = \mathbf{I}$) dont le produit est égal à $\mathcal{X}$.*

### 2.4.4 Variantes importantes de la décomposition de Tucker

On présente dans cette section les variantes de la décomposition de **Tucker** abordées dans le cadre de notre travail.

#### 2.4.4.1 Décomposition en ligne de Tucker

Dans une multitude d'applications, les données multimodales sont représentées par des séquences (une séquence étant un ensemble d'échantillons représentés par des tenseurs), spécialement dans des applications pour lesquelles les données sont acquises de manière séquentielle au fil du temps (e.g. données climatiques [123]). Une stratégie naturelle pour traiter ces séquences reviendrait à employer, après l'acquisition de chaque nouvelle observation, des décompositions standard. Cette stratégie présente une limite majeure, qui est que les opérations intermédiaires résultantes peuvent avoir une complexité en espace élevée (par exemple, certaines approches pour la décomposition non-négative de **Tucker** [103] matérialisent le produit de **Kronecker** de $N-1$ matrices de saturation, $N$ étant l'ordre du tenseur). Il est donc primordial de mettre en place des approches permettant de traiter les échantillons par petit nombre plutôt que de traiter toute la séquence en une seule fois.

Dans la littérature, les techniques de décomposition en ligne pour la décomposition de **Tucker** peuvent être classées en deux principales catégories. Pour la première classe de méthodes, le principe repose sur l'inférence des facteurs latents en traitant les observations dans la séquence une par une [118], [59]. Le principe de la seconde classe de méthodes repose sur le fait d'empiler toutes les observations (aussi bien celles déjà acquises que celles qui le sont nouvellement) et mettre à jour les facteurs latents de manière récursive. Le recours aux observations déjà acquises n'est pas requis pour le premier groupe de méthodes alors qu'il l'est pour le second.

On s'intéressera plutôt à la première classe de méthodes, notre objectif dans la suite de ce manuscrit étant la mise en place d'une nouvelle méthode utilisant chaque observation une seule fois, ce qui est intéressant car le maintien en mémoire des observations déjà acquises n'est pas nécessaire. Des techniques ont déjà été proposées pour cette classe d'approches. Thiernithi et al. ont présenté dans [118] une approche dans le cadre de l'acquisition comprimée, dont le principe est d'inférer les facteurs latents via des règles de mises à jour récursives. Étant donné deux séquences de tenseurs $\{\mathcal{X}_1, .., \mathcal{X}_T\}, \mathcal{X}_j \in \mathbb{R}^{I_1 \times ... \times I_N}$ et $\{\mathcal{G}_1, ..., \mathcal{G}_T\}, \mathcal{G}_j \in \mathbb{R}^{J_1 \times ... \times J_N}$, le problème résolu est le suivant :

$$\min_{\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} \frac{1}{2} \sum_{j=1}^{T} \beta_j \|\mathcal{X}_j - \mathcal{G}_j \times_1 \mathbf{A}^{(1)} .. \times_N \mathbf{A}^{(N)}\|_F^2 \tag{2.33}$$

avec $\beta_j > 0$ étant un paramètre d'oubli.
Les facteurs latents sont ensuite inférés via des statistiques résultant de la réformulation de la fonction donnée par l'équation (2.33) permettant de mettre à jour les facteurs latents via une descente de gradient et en traitant une seule observation à la fois.
La limite de cette approche résulte du fait qu'elle suppose que le tenseur noyau est fixé à l'avance, ce qui est assez restrictif de manière générale (car on n'a, la plupart du temps, aucune idée de sa forme), même si c'est une hypothèse naturelle en acquisition comprimée.
Une approche riemannienne (c'est à dire le problème d'optimisation associé est formulé sous forme d'un problème contraint, la contrainte étant définie par une variété) a été proposée par Hiroyuki et al. [59], le problème de minimisation résolu étant :

$$\min_{\mathcal{X}_{approx}} \|\mathcal{P}_\Omega(\mathcal{X}) - \mathcal{P}_\Omega(\mathcal{X}_{approx})\|_F^2 \tag{2.34}$$

sous la contrainte $\mathcal{X}_{approx} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}, \mathbf{A}^{(n)T}\mathbf{A}^{(n)} = \mathbf{I}, rank(\mathcal{X}_{approx}) = r$.
Dans l'équation (2.34), l'opérateur $\mathcal{P}_\Omega$ est l'opérateur qui met à zéro les composantes dont les indices se trouvent dans $\Omega$ et $\mathbf{I}$ représente la matrice identité.
Cette méthode impose l'orthogonalité sur les facteurs, ce qui n'est pas pertinent pour certaines tâches concrètes (par exemple, pour des données avec des composantes non-négatives, il est nécessaire d'imposer des contraintes de non-négativité sur les facteurs pour maintenir le sens physique [30]). Une autre approche, dont le principe repose sur la projection sur des sous-espaces aléatoires a été proposée par Yu et al. [126] dans le cadre de la prédiction de données spatio-temporelles. Cette approche repose sur l'hypothèse de faible rang (représentant une contrainte sur la nature de la donnée) qui n'est pas vérifiée en pratique pour toutes les données.
Globalement, la classe d'approches de décomposition en ligne de **Tucker** à laquelle on s'intéresse présente les limites suivantes :

- Fortes contraintes sur les facteurs latents comme l'orthogonalité [59], ce qui conduit à l'impossibilité de les étendre pour introduire d'autres types de contraintes telles que la non-négativité ;
- Connaissance préalable de certains facteurs de la décomposition [118], ce qui n'est pas acquis pour de nombreuses applications
- Contraintes fortes sur la nature de la donnée [59], ce qui réduit le cadre d'application des méthodes concernées.

#### 2.4.4.2 Techniques de décomposition de Tucker pour un tenseur de grande taille

La mise en place d'une approche efficace pour décomposer un tenseur de grande taille dans un cadre statique (c'est-à-dire que le tenseur est fixé) est un problème intéressant pour deux raisons principales : premièrement, on ne peut pas toujours charger en mémoire un tenseur d'intérêt et deuxièmement, le processus de décomposition peut résulter en un problème de gestion de mémoire [67].
Plusieurs méthodes déjà été établies pour ce problème. Zhe et al. [132] ont proposé une approche basée sur le problème suivant :

$$\min_{\mathcal{G},\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} .. \times_N \mathbf{A}^{(N)}\|_F^2 \tag{2.35}$$

sous la contrainte $\mathbf{A}^{(n)T}\mathbf{A}^{(n)} = \mathbf{I}, 1 \leq n \leq N$.
L'algorithme sous-jacent est une modification de **HOOI** avec des schémas de calcul permettant de distribuer les divers produits tenseur-matrice.
Une autre approche proposée par Becker et al. [13] repose sur la minimisation alternée avec l'incorporation d'opérateurs de Sketching (voir Annexe 1) pour la réduction de la dimension des problèmes :

— Problème résolu

$$\min_{\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times ... \times_N \mathbf{A}^{(N)}\|_F^2 \tag{2.36}$$

— Mise à jour de la matrice $\mathbf{A}^{(n)}$

$$\min_{\mathbf{A}} \|\mathbf{T}^{(n)} \left(\otimes_{m \in \mathbf{I}_{N \neq n}} \mathbf{A}^{(n)}\right) \mathbf{G}^{(n)} \mathbf{A}^T - \mathbf{T}^{(n)}(\mathbf{Y}^{(n)})\|_F^2$$

Dans cette équation, $\mathbf{T}^{(n)}$ représente un opérateur de Sketching.

— Mise à jour de $\mathcal{G}$

$$\min_{\mathcal{G}} \|\mathbf{T}^{\mathcal{G}} \left(\otimes_{m \in \mathbf{I}_N} \mathbf{A}^{(m)}\right) - \mathbf{T}^{\mathcal{G}} \operatorname{vec}(\mathcal{X})\|_F^2$$

avec $\mathbf{T}^{\mathcal{G}}$ représentant un opérateur de Sketching.

Plus généralement, les méthodes présentant des stratégies de décomposition pour un tenseur de grande taille peuvent être subdivisées principalement en deux groupes :

— Classe 1 : les méthodes reposant sur le principe du "diviser pour mieux régner"
Les méthodes dans cette classe permettent de réduire la complexité en espace en décomposant un tenseur d'intérêt via le traitement séquentiel de sous-tenseurs et se subdivisent en trois sous-classes principales :

1. les méthodes distribuées [132],[22],[95],[28]

2. les méthodes reposant sur le calcul du produit tenseur-matrice à la volée [16],[24],[13],[94]
3. les méthodes traitant séquentiellement des composantes tirées de manière aléatoire [85],[104],[41],[112],[131],[61],[130],[98],[81],[25],[76]

— Classe 2 : les méthodes à coût intermédiaire faible

Ces méthodes, utilisant toute la donnée, mettent en place des stratégies pour réduire le coût des calculs intermédiaires de la décomposition de **Tucker**. On peut ranger dans cette classe d'approches la méthode [13] décrite plus haut. D'autres approches aussi bien aléatoires [23],[88],[40],[106],[116],[129],[27] que déterministes [113], [10],[108], [11],[30], [56],[79],[91] ont également été proposées. Les limites principales de ces approches résultent du fait qu'elles utilisent toute la donnée d'un seul coup, ce qui conduit à leur non-applicabilité quand le tenseur à décomposer ne tient pas en mémoire.

Pour les approches de décomposition de **Tucker** pour un tenseur de grande taille, certaines méthodes assurent une garantie de convergence [13] ou établissent une borne sur l'erreur d'approximation [6]. Cependant, la vitesse de convergence pour la décomposition de **Tucker** d'un tenseur de grande taille n'est pas encore établie.

#### 2.4.4.3 Apprentissage de dictionnaires multimodaux par décomposition de tenseurs

Récemment, certains auteurs ont proposé des techniques d'apprentissage de dictionnaires par décomposition de tenseurs.

Soltani et al. [109] ont proposé une méthode consistant à inférer un dictionnaire tensoriel à partir du tenseur d'entraînement $\mathcal{Y} \in \mathbb{R}^{P\times T\times R}$ par résolution du problème suivant :

$$\min_{\mathcal{D},\mathcal{H}} \|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_F^2 + \lambda\|\mathcal{H}\|_1 + \mathbb{1}_{\mathbb{R}_+^{S\times T\times R}}(\mathcal{H}) + \mathbb{1}_{\mathbb{D}}(\mathcal{D}) \tag{2.37}$$

avec $\mathcal{D} \in \mathbb{R}^{P\times S\times R}, \mathcal{H} \in \mathbb{R}^{S\times T\times R}, \mathbb{D} = \left\{\mathcal{D} \in \mathbb{R}^{P\times S\times R} | \|\mathcal{D}\|_F^2 \leq \sqrt{p\times r}\right\}$ où l'opérateur * est défini par :

$$\mathcal{D} * \mathcal{H} = \mathtt{fold}\left(\mathtt{circ}\left(\mathcal{D}\right)\mathtt{unfold}\left(\mathcal{H}\right)\right)$$

Pour un tenseur $\mathcal{A} \in \mathbb{R}^{l\times m\times n}$ (dont les formes matricielles sont notées par $\mathbf{A}^{(m)}$), les opérateurs $\mathtt{unfold}, \mathtt{circ}, \mathtt{fold}$ sont définis comme suit :

$$\mathtt{fold}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} \\ . \\ . \\ \mathbf{A}^{(n)} \end{bmatrix}, \mathtt{circ}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n)} & .... & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & .... & \mathbf{A}^{(3)} \\ . & . & ... & . \\ \mathbf{A}^{(n)} & \mathbf{A}^{(n-1)} & .. & \mathbf{A}^{(1)} \end{bmatrix}, \mathtt{unfold} = \text{inverse de } \mathtt{fold}.$$

Ce problème est ensuite reformulé par séparation de variables et résolu par **ADMM**[15].

Une limite importante de cette technique découle du fait qu'elle n'est pas appropriée pour certains tâches telles que les problèmes de classification.

Wu et al. [122] ont proposé une approche pour la classification d'images consistant à l'apprentissage de deux matrices de dictionnaires spatiaux et une matrice de dictionnaires pour les exemples d'entraînement. L'idée de la décomposition est d'inférer les dictionnaires tout en cherchant à minimiser la dispersion intra-classe via le problème de minimisation suivant :

$$\min_{\mathbf{D}^W,\mathbf{D}^H,\mathbf{D}^N,\mathcal{X}_c^l} \sum_{c=1}^{C} \|\mathcal{Y}_c^l - \mathcal{X}_c^l \times_1 \mathbf{D}^W \times_2 \mathbf{D}^H \times_3 \mathbf{D}^N\|_F^2 \tag{2.38}$$

$$+\lambda \sum_{i\in\{W,H,N\}} \left( Tr\left(\mathbf{D}^{(i)T}\mathbf{S}_w^{(i)}\mathbf{D}^{(i)}\right) + \sum_{c=1}^{C}\sum_{l=1}^{L_c}\sum_{h\neq c}^{C}\sum_{q=1}^{L_h} \beta_{cl}^{hq} \| \left(\mathbf{D}_c^{l(i)}\right)^T \mathbf{D}_h^q \|_F^2 \right)$$

sous la contrainte de parcimonie des tenseurs $\mathcal{X}_c^l$

$\mathcal{Y}_c$ est le tenseur d'ordre 3 obtenu en empilant dans le sens du dernier mode les images appartenant à la classe c, les matrices $\mathbf{D}^W = [\mathbf{D}_1^{1(W)}, ..., \mathbf{D}_1^{L_1(W)}, .., \mathbf{D}_C^{1(W)}, ...\mathbf{D}_C^{L_C(W)}] \in \mathbb{R}^{d_w \times m_w}, \mathbf{D}^H = [\mathbf{D}_1^{1(H)}, ..., \mathbf{D}_1^{L_1(H)}, .., \mathbf{D}_C^{1(H)}, ...\mathbf{D}_C^{L_C(H)}] \in \mathbb{R}^{d_H \times m_H}$ représente les dictionnaires spatiaux, $\mathbf{S}_w$ correspond à une matrice de dispersion intra-classe. La principale limite de cette approche résulte du fait que le nombre de paramètres à régler est important.

Zubair et al. [136] ont proposé une approche basée sur la décomposition de **Tucker** classique avec une contrainte de parcimonie sur les entrées du tenseur noyau. Cette méthode n'impose de contraintes sur les facteurs latents, ce qui n'est pas pertinent dans certaines applications (par exemple pour la décomposition d'un tenseur avec des composantes non-négatives).

Les autres approches de la littérature présentent des limites similaires à celles mentionnées pour les méthodes présentées ci-dessus.

## 2.5 Outils numériques pour la décomposition de tenseurs

Dans cette section, nous présentons brièvement les outils numériques pour la décomposition de tenseur aussi bien en *Python* qu'en *Matlab*.

### 2.5.1 Matlab

Trois exemples d'outils en Matlab sont les suivants :

— The N-way Toolbox for MATLAB [5] : est un outil qui fournit une large gamme de fonctions et d'algorithmes pour la décomposition de données multimodales.

Plusieurs décompositions sont prises en compte, telles que : la décomposition **CP**, la décomposition de **Tucker**, la régression multilinéaire partielle, etc. Ces décompositions peuvent être effectuées avec certaines contraintes classiques telles que la non-négativité ou l'orthogonalité.

— MATLAB Tensor Toolbox [7],[8],[9] : propose un ensemble de fonctions pour supporter les opérations classiques sur les tenseurs telles que la multiplication d'un tenseur par une matrice ou la matricisation. Les décompositions **CP** et **Tucker** sont prises en charge via des approches basées sur la minimisation alternée. Il fournit également un support pour la gestion des tenseurs parcimonieux (c'est-à-dire contenant des composantes nulles).

— Tensorlab [119] : cet outil est l'un des plus complets actuellement avec une large gamme d'algorithmes pour un important nombre de factorisations (pour des tenseurs aussi bien réels que complexes) telles que : la décomposition **CP**, la décomposition de **Tucker**, la décomposition block term. Différents algorithmes pour les problèmes de factorisation sont proposés avec la possibilité d'inclure des contraintes populaires telles que la non-négativité ou l'orthogonalité.

### 2.5.2 Python

Les outils aboutis sont bien moins nombreux en Python qu'en Matlab. L'un des plus récents est Tensorly [68]. Tensorly fournit des algorithmes pour les décompositions **CP** et **Tucker** avec possibilité d'inclure des contraintes (non-négativité et orthogonalité). Il présente également un système de back-end pour représenter les données et effectuer de manière transparente des calculs numériques : NumPy, MXNet et PyTorch offrant ainsi que la possibilité (via MXNet et PyTorch) de faire du calcul sur GPU. L'implémentation de nos travaux reposent sur Tensorly.

## 2.6 Conclusion

Dans la suite, nous présentons les travaux effectués dans le cadre de cette thèse qui comprennent :

— Une approche pour l'apprentissage de dictionnaires par décomposition non-négative de **Tucker** pour laquelle nos contributions sont les suivantes : on clarifie le lien entre apprentissage de dictionnaires, **NMF** et décomposition de **Tucker** dans un contexte de classification, un problème de minimisation est proposé pour l'apprentissage de motifs temporels par décomposition de **Tucker** et l'efficacité de notre méthode est démontrée via des des expériences numériques .

— Une méthode pour l'apprentissage de dictionnaires en ligne via la décomposition de **Tucker** avec les contributions suivantes : on met en place une nouvelle approche pour l'apprentissage de dictionnaires multimodaux dont la convergence est prouvée pour le problème d'optimisation proposé et l'efficacité démontrée via des expériences numériques.

— Une technique pour la décomposition de **Tucker** d'un tenseur de grande taille dans un cadre statique (tenseur fixé) avec comme contributions : la proposition d'une nouvelle approche pour la décomposition de **Tucker** efficiente en terme de gestion de mémoire pour laquelle on établit une vitesse de convergence ainsi que son efficacité via des expériences numériques.

CHAPITRE 3

# APPRENTISSAGE DE DICTIONNAIRES PAR DÉCOMPOSITION NON-NÉGATIVE DE TUCKER

## 3.1 Contexte

L'apprentissage de dictionnaires est une technique standard utilisée dans divers domaines tels que le traitement du signal [54, 55], le débruitage d'images [26] ou la classification de signaux [101].
Quand la donnée que l'on cherche à traiter présente une structure particulière telle que la non-négativité, une pratique courante est d'incorporer une contrainte de non-négativité dans le problème d'apprentissage de dictionnaires, résultant sur un problème de décomposition non-négative de matrices dont l'objectif est d'approximer une matrice non-négative $\mathbf{S}$ par le produit de deux matrices non-négatives $\mathbf{W}$ et $\mathbf{H}$. Cette approximation se fait par la minimisation de fonctions de coût faisant intervenir différentes mesures, trois des plus courantes étant la divergence euclidienne [127], la divergence de Kullback-Leibler [89] ou encore la divergence de Itakura-Saito [42].
Un domaine d'application important de la **NMF** est l'analyse de signaux audio pour laquelle les signaux sont généralement représentés par leurs représentations temps-fréquence. Une limite importante de la **NMF** par minimisation de divergence euclidienne dans ce cadre est la difficulté à modéliser correctement les motifs temporels dans un signal audio dû au fait que les colonnes de la matrice de dictionnaire sont traitées indépendamment les unes des autres dans un problème d'apprentissage de dictionnaires, c'est-à-dire :

$$\mathbf{S} \approx \mathbf{WH} \Rightarrow \mathbf{S} \approx (\mathbf{WP})\left(\mathbf{P}^{-1}\mathbf{H}\right)$$

avec $\mathbf{P}$ étant une matrice de permutation, $\mathbf{P}^{-1}$ l'inverse de $\mathbf{P}$.
Certaines approches ont été proposées pour limiter l'ampleur de cette difficulté. Une variante de la **NMF**, introduite dans [44] pour la découverte de motifs temporels, exprime chaque matrice non-négative comme combinaison linéaire de patchs temps-fréquence. Cette solution conduit à des solutions sous-optimales car elle n'exploite pas la répétition des motifs temporels [47]. Une solution basée sur la décomposition de tenseurs a été suggérée dans [47], mais n'a pas été investiguée en profondeur. Notre objectif est de proposer une extension naturelle de la **NMF** basée sur la décomposition de **Tucker**. L'apprentissage de dictionnaires par décomposition de tenseurs a déjà été investiguée [102, 26]. La méthode présentée dans [109] ne prend pas en compte l'information de classe et celle établie dans [122] nécessite le réglage d'un grand nombre de paramètres. Les autres approches présentent des limitations similaires.
Notre méthode étend les approches existantes en permettant la sur-complétude des atomes de dictionnaires et en tirant parti de l'information de classe dans un cadre de classification.
Nos contributions sont les suivantes :

- On clarifie le lien entre l'apprentissage de dictionnaires, la **NMF** et la factorisation de tenseurs dans un contexte de classification ;
- On propose un problème de minimisation pour l'apprentissage de motifs temporels par décomposition non-négative de **Tucker** ;
- On effectue des simulations numériques sur des données synthétique et réelle et prouve que les caractéristiques (pour la classification) apprises par notre approche sont compétitives par rapport à la **NMF** associée à une opération de pooling.

## 3.2 Décomposition non-négative de matrices

L'objectif de la décomposition non-négative de matrices est d'approximer une matrice $\mathbf{S}$ non-négative (dont les entrées sont toutes supérieures ou égales à zéro) par le produit de deux matrices $\mathbf{W}$ et $\mathbf{H}$ toutes les deux non-négatives. L'approximation cherchée est donc la suivante (voir figure 3.1 pour une illustration) :

$$\mathbf{S} \approx \mathbf{WH}, \ \mathbf{H} \geq 0, \mathbf{W} \geq 0 \tag{3.1}$$

où les inégalités $\mathbf{H} \geq 0$, $\mathbf{W} \geq 0$ font référence à la non-négativité des composantes.
L'approximation (3.1) est effectuée en pratique par résolution du problème de minimisa-

tion suivant :

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} \ell\left(\mathbf{S},\widehat{\mathbf{S}}\right), \text{sous la contrainte } \widehat{\mathbf{S}} = \mathbf{WH} \tag{3.2}$$

avec $\ell$ étant une fonction de coût encore appelée divergence qui est généralement définie de telle sorte à éviter la divergence des composantes de $\mathbf{W}$ tout en assurant que certaines composantes de $\mathbf{H}$ sont nulles.

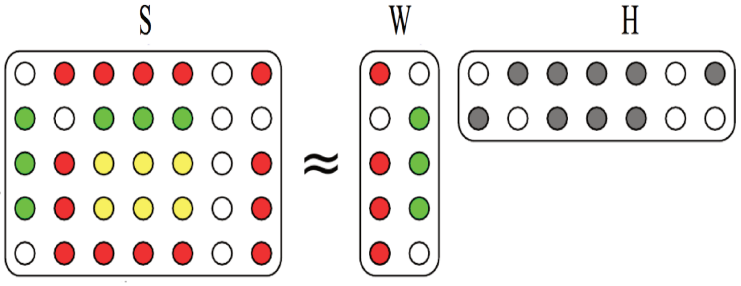


FIGURE 3.1 – Illustration de la décomposition non-négative de matrices. La couleur noire correspond aux composantes nulles et les autres à celles qui sont strictement positives et finies. On cherche donc à approximer une matrice $\mathbf{S}$ par le produit d'une matrice $\mathbf{W}$ non-négative dont les composantes sont finies et d'une matrice non-négative $\mathbf{H}$ dont certaines des composantes sont nulles.

## 3.3 De la NMF à la décomposition non-négative de Tucker

Étant donné un ensemble de $L$ signaux représentés par des spectrogrammes $\{\mathbf{S}_i\}_{1\leq i\leq L}$ (représentation de la puissance d'un signal en fonction du temps et de la fréquence) et un entier $K_f$ représentant le nombre d'atomes de dictionnaires, l'apprentissage de dictionnaires via la décomposition non-négative de matrices **NMF** est généralement formulé via le problème suivant (résumé par la figure 3.2) :

$$\min_{\mathbf{W}^{(f)}\in\mathbb{R}_+^{F\times K_f},\mathbf{H}_i\in\mathbb{R}_+^{K_f\times T}} \sum_{i=1}^{L} \|\mathbf{S}_i - \mathbf{W}^{(f)}\mathbf{H}_i\|_F^2 + \lambda_1\|\mathbf{W}^{(f)}\|_F^2 + \lambda\sum_{i=1}^{L}\|\mathbf{H}_i\|_1. \tag{3.3}$$

La matrice $\mathbf{W}^{(f)} \in \mathbb{R}_+^{F\times K_f}$, contenant les atomes de dictionnaires (représentés par ses colonnes), est appelée matrice de dictionnaires. Les matrices $\mathbf{H}_i$ sont des coefficients d'activation. On parle de dictionnaires sur-complets si $K \geq F$.

Dans l'équation (3.3), le second terme permet d'éviter la divergence des atomes de dictionnaires (c'est-à-dire qu'on essaie de faire de telle sorte que les composantes de $\mathbf{W}^{(f)}$ ne prennent pas des valeurs infinies) et le troisième permet d'inférer des coefficients d'acti-

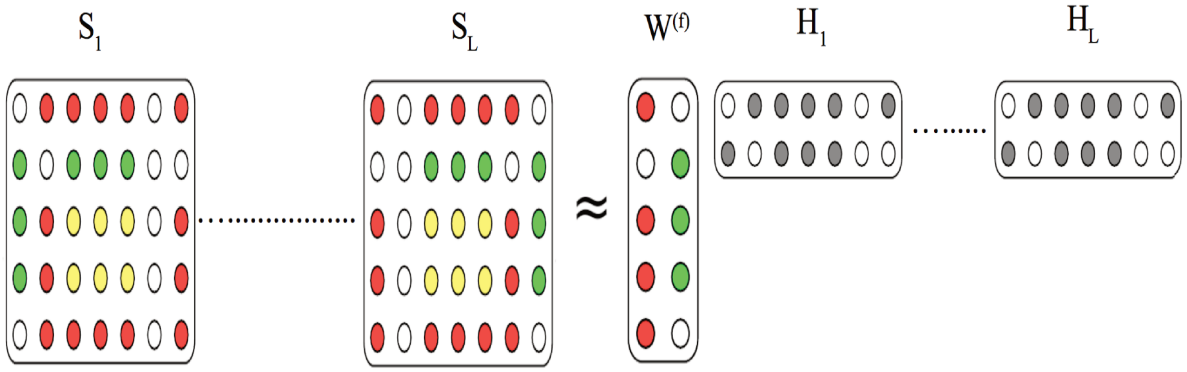


FIGURE 3.2 – Illustration de la décomposition non-négative pour un ensemble de matrices non-négatives $\{\mathbf{S}_k\}_{1\leq k\leq L}$. Les couleurs suivent la même logique que la figure 3.1 : couleur noire pour les composantes nulles et les autres couleurs pour les composantes non-nulles. Chaque matrice $\mathbf{S}_k$ est approximée par le produit deux matrices non-négatives $\mathbf{W}^{(f)}$ (dont toutes les composantes sont finies) et $\mathbf{H}_k$ (dont certaines des composantes sont nulles). La disposition des matrices $\mathbf{S}$ (ainsi que $\mathbf{H}_k$) signifie qu'elles sont empilées horizontalement)

vation parcimonieux (c'est-à-dire contenant des composantes nulles).

Considérons un tenseur d'ordre 3 $\mathcal{S} \in \mathbb{R}_+^{L\times F\times T}$ dont les sous-tenseurs par rapport au premier mode $\mathcal{S}_i^1$ représentent les spectrogrammes $\{\mathbf{S}_i\}_{i=1}^L$. Le problème (3.3) peut être réécrit sous la forme suivante :

$$\min_{\mathbf{W}^{(f)}\in\mathbb{R}_+^{F\times K_f},\mathcal{H}\in\mathbb{R}_+^{L\times K_f\times T}} \|\mathcal{S}-\mathcal{H}\times_2\mathbf{W}^{(f)}\|_F^2+\lambda_1\|\mathbf{W}^{(f)}\|_F^2+\lambda\|\mathcal{H}\|_1, \tag{3.4}$$

Dans l'équation (3.4), le tenseur $\mathcal{H}$ est un tenseur d'ordre 3 dont les sous-tenseurs par rapport au premier mode représentent les matrices $\{\mathbf{H}_i\}_{i=1}^L$.

L'équivalence entre les deux formulations données par les équations (3.3) et (3.4) est prouvée en Annexe 1.

Un problème important avec la **NMF** résulte du fait que tout ensemble d'atomes de dictionnaires $\left\{\mathbf{W}_{:,k}^{(f)}, 1\leq k\leq K_f\right\}$ est invariant par rapport à une permutation des colonnes de $\mathbf{S}_i$, ce qui gomme toute information temporelle au niveau des atomes de dictionnaires.

Dans l'objectif d'atténuer l'ampleur de ce problème, nous introduisons explicitement dans la formulation (3.4) une matrice $\mathbf{W}^{(t)}$ pour mieux localiser les motifs temporels présents dans les signaux. Cette matrice contient $K_t$ atomes de dictionnaires temporels qui seront associés aux $K_f$ atomes de dictionnaires fréquentiels de $\mathbf{W}^{(f)}$.

Ce modèle tensoriel revient donc à approximer chaque spectrogramme comme une combinaison linéaire de $K_f \times K_t$ éléments de base $\left\{\mathbf{W}_{:,p}^{(f)}(\mathbf{W}_{:,q}^{(t)})^\top\right\}$, permettant ainsi la surcomplétude si l'on choisit $K_f$ et $K_t$ tels que $F\times T \leq K_f\times K_t$. On obtient donc un nouveau

problème de minimisation donné par :

$$\min_{\mathbf{W}^{(f)},\mathbf{W}^{(t)},\mathcal{H}} \|\mathcal{S} - \mathcal{H} \times_2 \mathbf{W}^{(f)} \times_3 \mathbf{W}^{(t)}\|_F^2 + \lambda_1^f \|\mathbf{W}^{(f)}\|_F^2 + \lambda_1^t \|\mathbf{W}^{(t)}\|_F^2 + \lambda \|\mathcal{H}\|_1 \qquad (3.5)$$

$$\text{sous la contrainte} \quad \mathbf{W}^{(f)} \in \mathbb{R}_+^{F \times K_f}, \mathbf{W}^{(t)} \in \mathbb{R}_+^{T \times K_t}, \mathcal{H} \in \mathbb{R}_+^{L \times K_f \times K_t}.$$

Cette décomposition est connue sous le nom de **Tucker2** [66], avec des termes de pénalité dont la motivation émane de l'apprentissage de dictionnaires.
L'intuition derrière ce problème est qu'étant donné la minimisation de l'erreur d'ajustement et la contrainte de non-négativité, les structures temporelles de nos signaux émergeront via la matrice de dictionnaires temporels $\mathbf{W}^{(t)}$. Comme on peut s'y attendre, la taille du problème peut être très importante si on choisit $K_f$ et $K_t$ très grands.

## 3.4 Cadre de classification

Supposons que notre objectif étant de classer des signaux représentés par leurs spectrogrammes $\{\mathbf{S}_k\}_{1 \leq k \leq L}$ suivant un certain nombre d'étiquettes $\{m_k\}_{1 \leq k \leq L}$. Comme on cherche à exploiter les informations de classe dans la décomposition, on propose le problème de minimisation suivant :

$$\min_{\mathbf{W}^{(f)},\mathbf{W}^{(t)},\mathcal{H},\mathbf{B}^{(f)},\mathbf{B}^{(t)}} \|\mathcal{S} - \mathcal{H} \times_2 \mathbf{W}^{(f)} \times_3 \mathbf{W}^{(t)}\|_F^2 + \lambda_1^f \|\mathbf{W}^{(f)}\|_F^2 + \lambda_1^t \|\mathbf{W}^{(t)}\|_F^2 \qquad (3.6)$$

$$+\lambda \|\mathcal{H}\|_1 + \lambda_c \|\mathcal{C} - \mathcal{H} \times_2 \mathbf{B}^{(f)} \times_3 \mathbf{B}^{(t)}\|_F^2 + \lambda_2^f \|\mathbf{B}^{(f)}\|_F^2 + \lambda_2^t \|\mathbf{B}^{(t)}\|_F^2,$$

$$\text{sous la contrainte} \quad \mathbf{W}^{(f)} \in \mathbb{R}_+^{F \times K_f}, \mathbf{W}^{(t)} \in \mathbb{R}_+^{T \times K_t}, \mathcal{H} \in \mathbb{R}_+^{L \times K_f \times K_t}, \mathbf{B}^{(f)} \in \mathbb{R}_+^{K_f \times K_f}, \mathbf{B}^{(t)} \in \mathbb{R}_+^{K_t \times K_t},$$

Comme [101], le rôle des matrices $\mathbf{B}^{(f)}$ et $\mathbf{B}^{(t)}$ est de faire de telle sorte que les coefficients d'activation s'alignent sur l'information de classe portée par le tenseur $\mathcal{C} \in \mathbb{R}^{L \times K_f \times K_t}$ construit similairement à la méthode présentée dans [101]. Pour simplifier, supposons que $K_f$ et $K_t$ soient divisibles par le nombre de classes $M$. Le tenseur $\mathcal{C}$, contenant exclusivement $0$ et $1$, est défini par :

$$\mathcal{C}_{k, m_k \times \frac{J_f}{M} : (m_k+1) \times \frac{J_f}{M}, m_k \times \frac{J_t}{M} : (m_k+1) \times \frac{J_t}{M}} = \mathbf{1}_{[\frac{J_f}{M}, \frac{J_t}{M}]} \qquad (3.7)$$

Dans l'équation (3.7), $\mathbf{1}_{[P,Q]}$ représente la matrice de taille $P \times Q$ et dont toutes les composantes sont égales à 1 et $m_k$ représente la classe de l'exemple $k$.
Par exemple, supposons qu'on a trois exemples appartenant à trois classes différentes avec $K_f = K_t = 6$. Le tenseur $\mathcal{C} \in \mathbb{R}^{3 \times 6 \times 6}$ est défini comme suit :

$$\mathcal{C}_{1,:,:} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathcal{C}_{2,:,:} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathcal{C}_{3,:,:} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Dans ce cadre, le terme de régularisation va inciter les éléments $\left\{ \mathbf{W}^{(f)}_{:,1:\frac{J_f}{3}}, \mathbf{W}^{(t)}_{:,1:\frac{J_t}{3}} \right\}$ à se focaliser sur la reconstruction du signal de la classe 1. De même, les éléments $\left\{ \mathbf{W}^{(f)}_{:,\frac{J_f}{3}:2\times\frac{J_f}{3}}, \mathbf{W}^{(t)}_{:,\frac{J_t}{3}:2\times\frac{J_t}{3}} \right\}$ (respectivement $\left\{ \mathbf{W}^{(f)}_{:,2\frac{J_f}{3}:J_f}, \mathbf{W}^{(t)}_{:,2\frac{J_t}{3}:J_t} \right\}$) vont se concentrer sur la reconstruction du signal de la classe 2 (respectivement de la classe 3).

N'admettant pas de solutions analytiques, le problème d'apprentissage de dictionnaires (3.6) est résolu par minimisation alternée : on minimise la fonction de coût par rapport à une variable quand toutes les autres sont fixées jusqu'à ce qu'un critère d'arrêt préalablement défini soit vérifié. Chaque problème de minimisation résultant, de la forme $\min_{\mathbf{B}\geq 0} \|\mathbf{X} - \mathbf{AB}\|_F^2$ (obtenu par l'opération de matricisation de la décomposition de **Tucker**) est résolu par descente de gradient projeté [127]. Ce choix est motivé par l'efficacité de cette approche pour la résolution de problèmes de minimisation convexes sous des contraintes linéaires.

Après l'inférence des dictionnaires, on calcule les caractéristiques de classification en projetant les spectrogrammes (associés aux exemples d'entraînement et de test) sur les matrices de dictionnaires inférées indépendamment de l'information de classe. Notons respectivement $\mathbf{F}_i$ et $\mathbf{S}_i$ la caractéristique et le spectrogramme associé à l'exemple $i$. On a :

$$\mathbf{F}_i \leftarrow \texttt{vectoriser}(\mathcal{F}_i) \tag{3.8}$$

$$\mathcal{F}_i \leftarrow \underset{\mathcal{G}\in\mathbb{R}_+^{J_f\times J_t}}{\arg\min} \left\{ \|\mathbf{S}_i - \mathcal{G} \times_1 \mathbf{W}^{(f)} \times_2 \mathbf{W}^{(t)}\|_F^2 + \lambda\|\mathcal{G}\|_1 \right\} \tag{3.9}$$

**Remarque 2** *: l'équation*(3.8) *est utilisé aussi bien pour les caractéristiques d'entraînement que de test. Cependant, ce sont uniquement les exemples d'entraînement qui interviennent dans la décomposition de des dictionnaires* $\mathbf{W}^{(f)}$ *et* $\mathbf{W}^{(t)}$

## 3.5 Expériences numériques

Dans cette section, on appellera notre approche **TDL**. L'objectif de ces expériences est d'illustrer l'efficacité de notre approche tensorielle par rapport à la **NMF** dans l'appren-

tissage de motifs temporels à partir de la représentation temps-fréquence. Notre approche est comparée au problème de **NMF** supervisé présenté dans [101] suivi d'une opération de vectorisation (on réordonne les éléments de la matrice en vecteurs), d'une opération de max-pooling (on construit un vecteur dont les composantes représentent le maximum sur les colonnes) ou mean-pooling (le vecteur résultant est déduit on calculant la valeur moyenne sur les colonnes). Pour chaque approche, le nombre d'atomes de dictionnaires fréquentiels est choisi de manière identique pour toutes les méthodes (**NMF** et **TDL**) et les caractéristiques extraites sont données à un **SVM** à noyaux pour la tâche de classification. Tous les hyper-paramètres (ceux du **SVM** inclus) sont déterminés par validation croisée (5-fold) sur l'ensemble d'entraînement. Pour la **NMF**, les composantes des points initiaux sont tirées d'une distribution uniforme sur $[0, 1]$ et pour **TDL**, en résolvant des problèmes de **NMF** définis via les formes matricielles de $\mathcal{S}$ et $\mathcal{C}$.

### 3.5.1 Données synthétiques

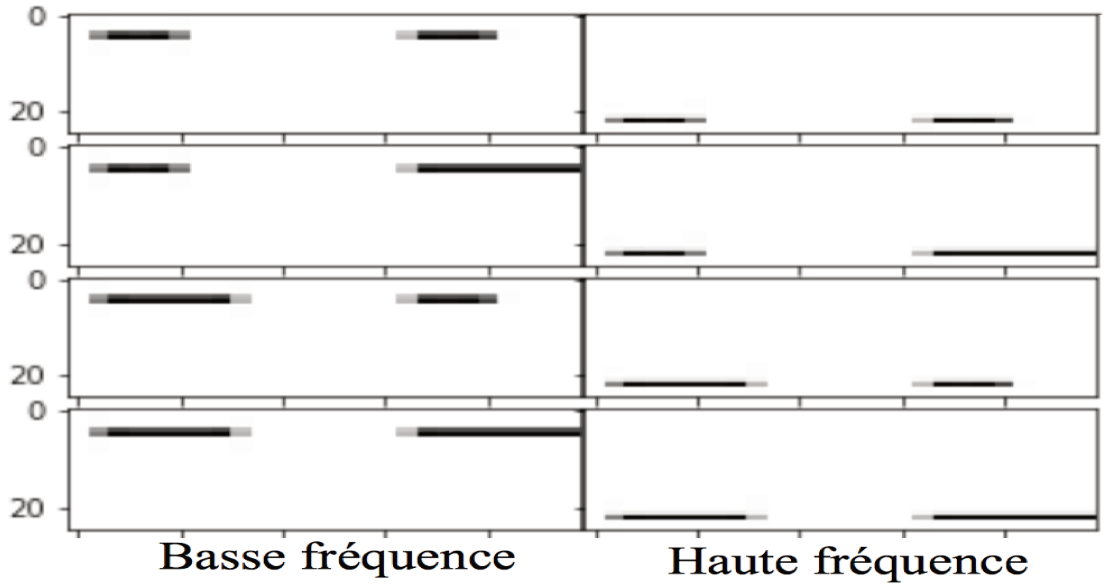


FIGURE 3.3 – Exemples de spectrogrammes (représentation de la puissance du signal en fonction du temps et de la fréquence) pour chaque classe de signaux. A gauche et à droite, on a des signaux haute fréquence et des signaux basse fréquence avec différentes longueurs dans le temps. L'axe des ordonnées (axe vertical) correspond aux fréquences et l'axe abscisses (axe horizontal) correspond au temps. Les niveaux de gris correspondent à l'énergie des motifs de fréquence qui présentent différentes durées

Cette base de données est composée de 400 exemples répartis équitablement entre 8 classes représentant des lieux. L'objectif est de classer les signaux, ce qui revient dans ce cadre à faire la différence entre deux lieux différents. Chaque signal est la somme de deux sinusoïdes présentant des durées diverses dans le temps. Chaque spectrogramme est dimensionné à $25 \times 25$ et bruité par un bruit uniforme (voir la figure 3.3 pour des exemples). Le critère d'évaluation considéré est la précision moyenne (voir [101] pour la définition) , qui

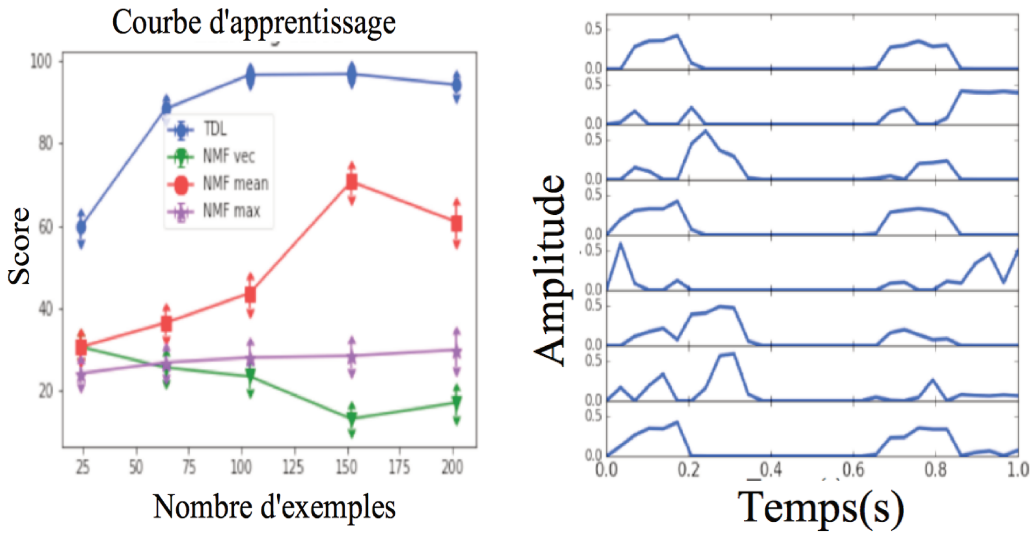


FIGURE 3.4 – Gauche : courbe d'apprentissage. **NMF** vec,**NMF** mean et **NMF** max : les caractéristiques sont déterminées par vectorisation, max-pooling, mean-pooling, Droite : exemple de dictionnaires temporels montrant les structures temporelles apprises

est un critère standard utilisé dans les problèmes de classification. La courbe d'apprentissage, donnant le taux de classification en fonction du nombre d'exemples d'entraînement utilisés est donnée par la figure 3.4 (chaque score de cette figure (partie gauche) correspond à une moyenne calculée sur 5 bruits aléatoires différents). Notre approche donne des caractéristiques de classification discriminantes avec juste 24 exemples d'entraînement (une moyenne de 3 exemples par classe) et produit de meilleurs résultats comparée à la méthode **NMF**. Cela prouve que notre approche capture mieux l'information temporelle car les données sont construites pour qu'elle soit discriminante. De plus, la figure 3.4 (à droite) montre bien deux motifs de fréquence séparés par un temps de latence exactement comme sur les spectrogrammes (voir la figure 3.3).

### 3.5.2 DCase2013 data

Cette base de données réelle a été introduite pour la classification de scènes audio. La base de données est composée de 10 classes représentant divers types de lieux (bus, bureau, etc...). L'objectif est de classer les signaux, c'est-à-dire faire la différence entre divers lieux. Les ensembles de test et d'entraînement contiennent chacun 100 exemples (avec 10 exemples par classe) avec des scènes de 30 secondes chacun. Même si les motifs temporels ne sont pas aussi marqués que dans le cas de la donnée synthétique (dû au fait que les spectrogrammes ont été compressés pour contourner le problème de complexité en espace), notre objectif est de prouver que **TDL** est toujours capable de capturer des informations temporelles pertinentes via les atomes de dictionnaires temporels. Les spectrogrammes sont calculés et dimensionnés à $60 \times 20$. Comme pour la première expérience, les paramètres

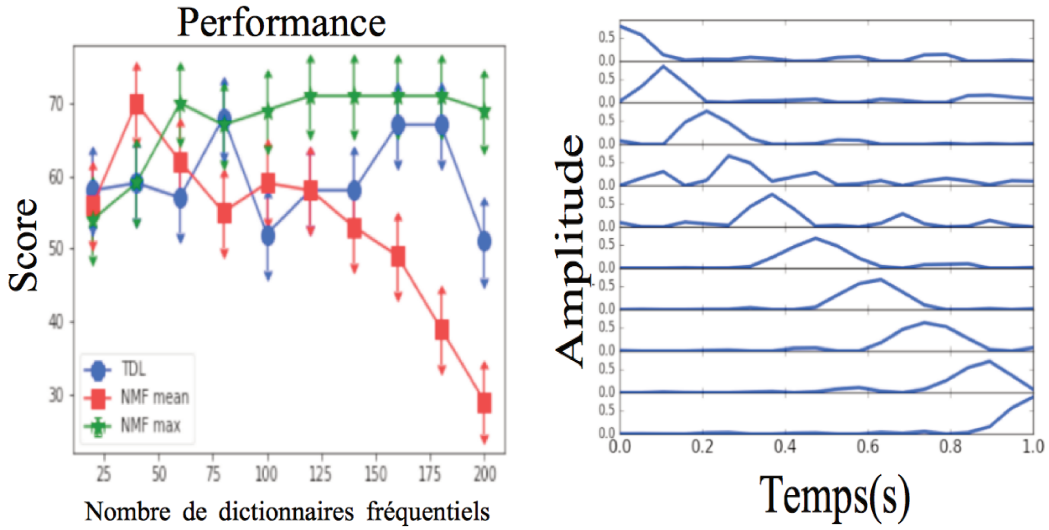


FIGURE 3.5 – Gauche : performance de **TDL** et **NMF** + pooling on DCase 2013. Droite : exemple de dictionnaires temporels.

sont réglés par validation croisée (5-fold). La figure 3.5 donne les scores de classification pour **TDL** et la **NMF** en fonction du nombre de dictionnaires fréquentiels avec 10 atomes temporels pour **TDL**.

On remarque que les performances sont approximativement égales avec un léger avantage pour la **NMF** associée au max-pooling. On remarque que dans ce cadre, notre approche se comporte comme une opération similaire au mean-pooling par rapport au temps dans le sens où l'on capture un temps moyen sur des zones locales (voir la figure 3.5 pour les motifs temporels appris par **TDL**), ce qui est en accord avec d'autres travaux (par exemple, Anh et al. ont montré dans [99] que les composantes temporelles capturent la durée moyenne du signal dans des données electro-encéphalographiques).

## 3.6 Conclusion

Dans ce chapitre, on a introduit une approche tensorielle d'apprentissage de dictionnaires sur-complets par décomposition non-négative de tenseurs pour l'apprentissage de motifs temporels dans des spectrogrammes. Cette nouvelle approche établit un lien clair entre l'apprentissage de dictionnaires et la décomposition de matrices. Les expériences numériques montrent que les caractéristiques apprises par notre approche sont compétitives par rapport à la **NMF** (associée au pooling). La principale limite de la méthode résulte du fait que les problèmes intermédiaires résultant de la minimisation alternée présentent une complexité élevée dûe au produit de **Kronecker** résultant de la propriété de matricisation de la décomposition de **Tucker**.

Pour contourner ce problème de complexité, on propose dans le chapitre suivant un algo-

rithme en ligne pour l'apprentissage de dictionnaires multimodaux.

CHAPITRE 4

# OTL, UN ALGORITHME D'APPRENTISSAGE EN LIGNE DE DICTIONNAIRES MULTIMODAUX PAR DÉCOMPOSITION DE TUCKER

## 4.1 Contexte

Dans ce chapitre, on s'intéresse à l'apprentissage de dictionnaires par décomposition en ligne de **Tucker** (en ligne signifiant dans le cadre de notre travail qu'on traite les observations par petit nombre plutôt que de les utiliser toutes de manière simultanée). Cette approche peut être vue comme une extension de la méthode **TDL** introduite dans le chapitre précédent dont l'une des limites réside dans le fait qu'elle a une complexité en espace élevée dû au produit de **Kronecker** provenant de la forme matricielle de la décomposition de **Tucker**.

Dans la littérature, les techniques en ligne pour la décomposition de **Tucker** peuvent être classées en deux catégories principales. Pour la première classe de méthodes, le principe repose sur l'inférence des facteurs latents en traitant de manière séquentielle des données en flux sans nécessité d'avoir recours aux observations déjà utilisées, c'est-à-dire chaque observation est traitée une seule fois [118], [59]. Le principe de la seconde classe de méthodes repose sur le fait d'empiler toutes les observations (aussi bien celles déjà acquises que celles qui le sont nouvellement) et mettre à jour les facteurs latents de manière récursive. La différence entre les deux classes de méthodes est que pour la première, chaque observation est traitée une seule fois.

On s'intéresse plutôt à la première classe d'approches pour laquelle des travaux ont déjà été mis en place [118], [59] (voir la section 2.4.4.1 pour plus de détails), notre objectif étant de mettre en place une méthode qui traite chaque observation une seule fois dans le

but d'avoir la complexité en espace la plus petite possible (étant donné que le maintien en mémoire des observations utilisées n'est pas nécessaire). Les méthodes de la littérature pour ce problème de décomposition en ligne présentent les limitations suivantes : elles ne sont pas assez génériques pour prendre en compte les spécificités de certaines données, impose des contraintes sur les tailles des facteurs latents, impose la connaissance préalable de certains facteurs latents. Ces méthodes manquent donc d'adaptabilité par rapport aux données.

Pour contourner ces problèmes, nous proposons une nouvelle approche en ligne pour inférer des dictionnaires via la décomposition de **Tucker**. Cette approche est basée sur la descente de coordonnées par blocs, la descente de gradients et des règles de mises à jour récursives. Nos contributions sont les suivantes :

— On propose une nouvelle approche pour l'apprentissage en ligne de dictionnaires multimodaux qui peut être facilement étendue pour incorporer certaines contraintes classiques telles que la non-négativité, l'orthogonalité, la parcimonie et notre approche ne fait aucune hypothèse sur la taille des facteurs latents.

— Convergence : on prouve, sous des conditions naturelles, la convergence de notre approche vers un point stationnaire du problème d'optimisation proposé.

— Des expériences numériques sont effectuées pour évaluer l'efficacité (capacité de reconstruction) de notre approche.

## 4.2 Apprentissage de dictionnaires multimodaux

Le problème d'intérêt dans ce travail est un cas particulier de la décomposition de **Tucker**. Étant donné un tenseur $\mathcal{X} \in \mathbb{R}^{T \times I_1 \times .. \times I_N}$ représentant un ensemble de $T$ observations de taille $I_1 \times ... \times I_N$ tirées d'une distribution de probabilité inconnue, l'objectif est d'effectuer l'approximation suivante :

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{I} \times_2 \mathbf{A}^{(1)} \times_2 ... \times_{N+1} \mathbf{A}^{(N)}, \mathcal{G} \in \mathbb{R}^{T \times J_1 \times .. \times J_N}, \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$$

Ce problème peut être spécifiquement formulé de la manière suivante :

$$\min_{\mathbf{A}^{(n)}, \{\mathcal{G}_t\}_{1 \leq t \leq T}} \sum_{i=1}^{T} \|\mathcal{X}_i - \mathcal{G}_i \times_{n \in I_N} \mathbf{A}^{(n)}\|_F^2 \tag{4.1}$$

avec $\mathcal{X}_i \in \mathbb{R}^{I_1 \times ... \times I_N}$ et $\mathcal{G}_i \in \mathbb{R}^{J_1 \times .. \times J_N}$ étant les $i^{\text{ième}}$ sous-tenseurs de $\mathcal{X}$ et $\mathcal{G}$ par rapport au premier mode. Il est à noter que les tenseurs $\mathcal{X}_i$ (respectivement $\mathcal{G}_i$) sont des tenseur d'ordre N. Dans la littérature, certains auteurs interprètent les matrices $\mathbf{A}^{(n)}$ comme des

matrices de dictionnaires et les composantes de $\mathcal{G}$ comme des coefficients d'activation [136].

Dans le cadre de l'apprentissage de dictionnaires classique, il est fréquent de chercher à inférer des coefficients parcimonieux (c'est-à-dire avec des composantes nulles) tout en évitant la divergence des atomes de dictionnaires [86]. Cette observation conduit au problème de minimisation suivant :

$$\min_{\{\mathbf{A}^{(m)}\}_{1\leq m\leq N},\{\mathcal{G}_t\}_{1\leq t\leq T}} \sum_{i=1}^{T} \left(\|\mathcal{X}_i - \mathcal{G}_i \times_{n\in I_N} \mathbf{A}^{(n)}\|_F^2 + \Omega_1(\mathcal{G}_i)\right) + \Omega_2(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) \tag{4.2}$$

La parcimonie et la bornitude sont respectivement induites via les fonctions de pénalité $\Omega_1$ et $\Omega_2$. On retrouve donc le problème de minimisation (voir l'équation (3.5)) pour **TDL**, l'approche introduite dans le chapitre précedent en choisissant $\Omega_1$ comme la norme $\ell_1$ et $\Omega_2$ comme la somme des carrés des normes de Frobenius et en fixant $\mathbf{A}^{(1)}$ à la matrice identité de taille $I_1 \times I_1$. Dans la suite, on suppose que $\Omega_2$ est différentiable et $\Omega_1$ admet un opérateur proximal [100].

### 4.2.1 Apprentissage en ligne de dictionnaires multimodaux *OTL*

Le problème traité dans ce papier est l'inférence de dictionnaires multimodaux en contournant le problème de complexité (en espace) émanant du fait de traiter toutes les observations en une seule fois. Pour résoudre ce problème pour une large séquence (une séquence étant une série de $T > 0$ observations, chacune représentant un tenseur), on propose une approche probabiliste, qui a déjà prouvé son efficacité pour la minimisation de fonctions non-convexes [86]. Supposons qu'on observe une sequence qui croît au fil du temps, les observations dans la séquence étant indépendantes et identiquement distribuées suivant une distribution de probabilité $\mathbb{P}$ sur l'espace des tenseurs de taille $I_1 \times ... \times I_N$. Notons par $\mathcal{X}_t \in \mathbb{R}^{I_1\times...\times I_N}$ le tenseur acquis à la date $t$. Notons par $\ell$ la fonction mesurant l'écart entre $\mathcal{X}_t$ et son approximation donnée par des facteurs latents fixés :

$$\ell(\mathcal{X}_t, \mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) = \min_{\mathcal{G}} \left\{ \frac{1}{2}\|\mathcal{X}_t - \mathcal{G} \times_{n\in\mathbf{I}_N} \mathbf{A}^{(n)}\|_F^2 + \Omega_1(\mathcal{G}) + \Omega_2\left(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}\right) \right\} \tag{4.3}$$

Cette expression est justifiée par le fait que notre objectif est d'approximer chaque observation par le produit entre un tenseur noyau et une base génératrice de facteurs latents, c'est-à-dire l'approximation cherchée est la suivante :

$$\mathcal{X}_t \approx \mathcal{G}_t \times_{m\in I_N} \mathbf{A}^{(m)} \tag{4.4}$$

L'idée sous-jacente de notre approche repose sur la mise à jour des matrices de dictionnaires de telle sorte que l'écart entre les observations $\{\mathcal{X}_t\}_{t\geq 1}$ et leurs approximations données par ces matrices soit faible. Un problème pertinent est donc :

$$\min_{\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} \left\{ f(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) \triangleq \mathbb{E}_{\mathcal{X}\backsim\mathbb{P}}\left(\ell\left(\mathcal{X},\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}\right)\right) \overset{p.s}{=} lim_{t\to\infty}\frac{1}{t}\sum_{j=1}^{t}\ell\left(\mathcal{X}_j,\mathbf{A}^{(1)},...,\mathbf{A}^{(N)}\right) \right\} \tag{4.5}$$

avec la fonction de coût $\ell$ définie par l'équation (4.3) et p.s signifiant presque-sûrement.

### 4.2.2 Algorithme

Étant donné que la distribution de probabilité $\mathbb{P}$ est inconnue, le problème (4.5) n'admet pas de solution analytique. Pour contourner ce problème, on remplace la fonction de coût du problème (4.5) par son estimateur donné par la moyenne empirique, ce qui correspond au problème suivant :

$$\min_{\mathbf{A}^{(1)},..\mathbf{A}^{(N)}} \left\{ \widehat{f}_t\left(\mathbf{A}^{(1)},..\mathbf{A}^{(N)}\right) \triangleq \frac{1}{t}\sum_{i=1}^{t}\ell\left(\mathcal{X}_i,\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}\right) \right\} \tag{4.6}$$

L'idée de notre approche est de mettre à jour les atomes de dictionnaires $\left\{\mathbf{A}^{(n)}\right\}_{1\leq n\leq N}$ en minimisant $\widehat{f}_t(\mathbf{A}^{(1)},..\mathbf{A}^{(N)})$ et en utilisant uniquement l'observation $\mathcal{X}_t$, les facteurs inférés de $\mathcal{X}_{t-1}$ et un certain nombre de statistiques (c'est-à-dire des fonctions de l'échantillon). Notons $\left\{\mathbf{A}_t^{(n)}\right\}_{1\leq n\leq N}$ les facteurs inférés de la minimisation de $\widehat{f}_t$. Supposons l'acquisition d'une nouvelle observation $\mathcal{X}_t$ à la date $t$. Les mises à jour des atomes de dictionnaires, inspirées de travaux précédemment effectués dans le cadre matriciel [62], sont effectuées via une stratégie en deux étapes :

- Le calcul du tenseur d'activation $\mathcal{G}_t$ en projetant $\mathcal{X}_t$ sur les facteurs latents inférés au pas de temps $t-1$. **Dans la suite, cette étape sera appelée étape de "codage parcimonieux"**
- La résolution du problème (4.6) par descente de coordonnées par blocs, c'est-à-dire par minimisation de la fonction de coût par rapport à une seule variable quand toutes les autres sont fixées. **Dans la suite, on se référera à cette étape par "descente de coordonnées par blocs"**

On désignera notre processus d'inférence de dictionnaires par **OTL** (Online Tensor Learning).

#### 4.2.2.1 *Codage parcimonieux*

Pour cette étape, on considère le problème de minimisation suivant :

$$\min_{\mathcal{G}} \underbrace{\frac{1}{2}\|\mathcal{X}_t - \mathcal{G} \times_{n\in\mathbf{I}_N} \mathbf{A}_{t-1}^{(n)}\|_F^2}_{\mathcal{O}(\mathcal{G})} + \Omega_1(\mathcal{G})$$

Au lieu de vectoriser cette expression comme c'est le cas pour certaines approches pour la décomposition de **Tucker** [103], ce qui peut conduire à des opérations avec des complexités en espace élevées, on propose d'appliquer directement le principe de la minimisation proximale dans l'espace des tenseurs.
La minimisation proximale est une technique pour minimiser la somme de deux fonctions, l'une étant différentiable et l'autre admettant un opérateur proximal [100]. Dans notre cas, la fonction différentiable est $\mathcal{O}$ et sa dérivée est donnée par l'équation (4.7) :

$$\frac{\partial \mathcal{O}}{\partial \mathcal{G}}(\mathcal{G}) = -\mathcal{X}_t \times_{n\in\mathbf{I}_N} \mathbf{A}_{t-1}^{(n)T} + \mathcal{G} \times_{n\in\mathbf{I}_N} \mathbf{A}_{t-1}^{(n)T}\mathbf{A}_{t-1}^{(n)} \tag{4.7}$$

Étant donné que le terme de pénalité $\Omega_1$ admet un opérateur proximal (par hypothèse) noté $\mathbf{prox}_{\Omega_1}$, l'algorithme du codage parcimonieux est donné par **Algorithme 4**.

***Algorithme 4 : codage parcimonieux***

**Entrées**

$\mathcal{X}_t$ : observation,
$\left\{\mathbf{A}_{t-1}^{(n)}\right\}_{1\leq n\leq N}$ : matrices de dictionnaire,
$\eta > 0$ : pas de descente

**Initialisation**

iter=0, $\mathcal{G}_{iter} = \mathcal{G}_0$

**Répéter**

$$\mathcal{G}_{iter+1} = \mathbf{prox}_{\eta\Omega_1}\left(\mathcal{G}_{iter} - \eta\frac{\partial \mathcal{O}}{\partial \mathcal{G}}(\mathcal{G}_{iter})\right) \tag{4.8}$$

avec la dérivée $\frac{\partial \mathcal{O}}{\partial \mathcal{G}}$ définie par l'équation (4.7)
$iter = iter + 1$

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

#### 4.2.2.2 *Descente de coordonnées par blocs*

On considère le problème de mise à jour des matrices de dictionnaires. Supposons que le tenseur d'activation $\mathcal{G}_t$ ainsi que les $n-1$ premières matrices de saturation $\left\{\mathbf{A}_{k+1}^{(p)}\right\}_{1\leq p\leq n-1}$ ($k$ étant le numéro de l'itération, une itération étant la mise à jour des $N$ matrices de saturation) sont déjà déterminés. La mise à jour de $\mathbf{A}^{(n)}$ est effectuée en résolvant le problème de minimisation suivant :

$$\mathbf{A}_{k+1}^{(n)} \leftarrow \underset{\mathbf{A}^{(n)}}{\arg\min}\, \widehat{f}_{n,t}(\mathbf{A}^{(n)}) \tag{4.9}$$

$$\widehat{f}_{n,t}(\mathbf{A}^{(n)}) = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|\mathcal{X}_i - \mathcal{G}_i \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \Omega_1(\mathcal{G}_i)$$
$$+\Omega_2(\mathbf{A}_{k+1}^{(1)},.,\mathbf{A}_k^{(n-1)},\mathbf{A}^{(n)},\mathbf{A}_k^{(n+1)}.,\mathbf{A}_k^{(N)})$$

Le problème (4.9) n'a pas de solution analytique. Étant donné la differentiabilité de la fonction de coût $\widehat{f}_{n,t}$, nous proposons une résolution numérique basée sur la descente de gradient. Avec ce choix, la matrice $\mathbf{A}^{(n)}$ peut être déterminée en utilisant $\mathcal{X}_t$ et les matrices $\left\{\mathbf{A}_{k+1}^{(p)}\right\}_{1\leq p\leq n-1}, \left\{\mathbf{A}_k^{(q)}\right\}_{n+1\leq q\leq N}$ à condition que certaines statistiques (fonctions de l'échantillon) soient mises à jour au fil du temps. En effet, la dérivée de $\widehat{f}_{n,t}$ est donnée par :

$$\frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}}\left(\mathbf{A}^{(n)}\right) = -\frac{1}{t}\sum_{i=1}^{t}\left(\widehat{\mathbf{X}}_i^{(n)}\mathbf{G}_i^{(n)T} - \mathbf{A}^{(n)}\mathbf{B}_i^{(n)}\mathbf{B}_i^{(n)T}\right) + \frac{\partial \Omega_2}{\partial \mathbf{A}^{(n)}}(..,\mathbf{A}_{k+1}^{(n-1)},\mathbf{A}^{(n)},\mathbf{A}_k^{(n+1)},..) \tag{4.10}$$

Dans l'équation (4.10), $\mathbf{G}_i^{(n)}$ représente la forme matricielle par rapport au mode n du tenseur $\mathcal{G}_i$, $\mathbf{B}_i^{(n)}$ et $\widehat{\mathbf{X}}_i^{(n)}$ les formes matricielles des tenseurs $\mathcal{B}_i$ et $\mathcal{X}_i$ définis par :

$$\mathcal{B}_i = \mathcal{G}_i \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{I} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}, \tag{4.11}$$

$$\widehat{\mathcal{X}}_i = \mathcal{X}_i \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)T} \times_n \mathbf{I} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)T} \tag{4.12}$$

$\mathbf{I}$ représente la matrice identité ($\mathbf{I}\in\mathbb{R}^{J_n\times J_n}$ pour (4.11) et $\mathbf{I}\in\mathbb{R}^{I_n\times I_n}$ pour (4.12))
La dérivée peut donc être réécrite sous la forme :

$$\frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}}(\mathbf{A}^{(n)}) = -\frac{P_t^{(n)}}{t} + \frac{\mathbf{A}^{(n)}Q_t^{(n)}}{t} + \frac{\partial \Omega_2}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_{k+1}^{(1)},.,\mathbf{A}_{k+1}^{(n-1)},\mathbf{A}^{(n)},\mathbf{A}_k^{(n+1)},.,\mathbf{A}_k^{(N)}) \tag{4.13}$$

avec :

$$\mathbf{P}_t^{(n)} = \sum_{i=1}^{t}\widehat{\mathbf{X}}_i^{(n)}\mathbf{G}_i^{(n)T}, \mathbf{Q}_t^{(n)} = \sum_{i=1}^{t}\mathbf{B}_i^{(n)}\mathbf{B}_i^{(n)T}$$

Il est trivial de voir que :

$$\mathbf{P}_t^{(n)} = \mathbf{P}_{t-1}^{(n)} + \widehat{\mathbf{X}}_t^{(n)} \mathbf{G}_t^{(n)T} \tag{4.14}$$

$$\mathbf{Q}_t^{(n)} = \mathbf{Q}_{t-1}^{(n)} + \mathbf{B}_t^{(n)} \mathbf{B}_t^{(n)T} \tag{4.15}$$

Étant donné que les séquences $\mathbf{P}_t^{(n)}$ et $\mathbf{Q}_t^{(n)}$ vérifient les équations (4.14) et (4.15), le calcul du gradient de $\widehat{f}_{n,t}$ nécessite uniquement la nouvelle observation $\mathcal{X}_t$ et les matrices de dictionnaires $\left\{\mathbf{A}_{k+1}^{(p)}\right\}_{1\leq p\leq n-1}, \left\{\mathbf{A}_k^{(q)}\right\}_{n+1\leq q\leq N}$. Concrètement, pour un pas de temps fixé, la dérivée de $\widehat{f}_{n,t+1}$ peut être déduite des statistiques $\mathbf{P}_t, \mathbf{Q}_t$, des matrices de dictionnaires et de l'observation $\mathcal{X}_{t+1}$. Cela est la clé de notre approche puisque l'inférence des facteurs $\left\{\mathbf{A}_t^{(n)}\right\}_{1\leq n\leq N}$ peut se faire de manière efficace en utilisant une unique observation qui est $\mathcal{X}_t$. Cette étape est résumée par l'**Algorithme 5**.

***Algorithme 5 : mise à jour de*** $\mathbf{A}^{(n)}$

**Entrées**

$\mathcal{X}_t$ : nouvelle observation,
$\left\{\mathbf{A}_{k+1}^{(p)}\right\}_{1\leq p\leq n-1}, \left\{\mathbf{A}_k^{(q)}\right\}_{n+1\leq q\leq N}$ : dictionnaires,
$\mathbf{P}_{t-1}^{(n)}, \mathbf{Q}_{t-1}^{(n)}$ : matrices de récurrence
$\mathcal{G}_t$ : code parcimonieux associé à $\mathcal{X}_t$,
$\eta > 0$ : pas de descente

**Initialisation**

$iter = 0$, $\mathbf{A}_0^{(n)}$ : valeur initiale,

**Répéter**

$\mathbf{A}_{iter+1}^{(n)} = \left(\mathbf{A}_{iter}^{(n)} - \eta \frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_{iter}^{(n)})\right)$,
$\frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_{iter}^{(n)})$ défini conformément aux équations (4.13), (4.14) et (4.15)
$iter = iter + 1$

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

Le processus global d'inférence des facteurs latents (coefficients d'activation et matrices de dictionnaires), donné par **Algorithme 6** et pour lequel on peut remarquer qu'il ne nécessite aucune contrainte sur les tailles des facteurs latents, est résumé comme suit : on tire au hasard une observation (correspondant à un tenseur de la séquence), on détermine le code parcimonieux correspondant à l'observation choisie et on met à jour les matrices

de dictionnaires.

***Algorithme 6 : OTL***

**Entrées**

$\{\mathcal{X}_1, .., \mathcal{X}_T\}$ : séquence d'observations,

**Initialisation**

$\mathbf{A}_0^{(n)}$ : valeurs initiales des matrices de dictionnaires,

$\mathcal{G}_0$ : valeur initiale pour le tenseur noyau,

$\mathbf{P}_0^{(n)}$,$\mathbf{Q}_0^{(n)}$ : valeurs initiales des statistiques

**Sortie** : $\mathbf{A}_T^{(1)}, .., \mathbf{A}_T^{(N)}$

**Répéter**

Acquisition de $\mathcal{X}_t$ suivant une distribution de probabilité $\mathbb{P}$

Codage parcimonieux

$$\mathcal{G}_t \leftarrow \arg\min \frac{1}{2}\|\mathcal{X}_t - \mathcal{G} \times_{n\in I_N} \mathbf{A}_{t-1}^{(n)}\|_F^2 + \Omega_1(\mathcal{G})$$

Descente de coordonnées par blocs

**Pour** $1 \leq n \leq N$ **faire**

Mise à jour des statistiques

$$\mathbf{P}_t^{(n)} = \mathbf{P}_{t-1}^{(n)} + \widehat{\mathbf{X}}_t^{(n)}\mathbf{G}_t^{(n)T}$$

$$\mathbf{Q}_t^{(n)} = \mathbf{Q}_{t-1}^{(n)} + \mathbf{B}_t^{(n)}\mathbf{B}_t^{(n)T}$$

avec $\mathbf{B}_i^{(n)}$ et $\widehat{\mathbf{X}}_i^{(n)}$ étant les formes matricielles des tenseurs $\mathcal{B}_i$ et $\mathcal{X}_i$ définies par les équations (4.11) et (4.12)

Mise à jour de $\mathbf{A}^{(n)}$

$$\mathbf{A}_{k+1}^{(n)} \leftarrow \arg\min_{\mathbf{A}^{(n)}} \widehat{f}_{n,t}(\mathbf{A}^{(n)})$$

$$\mathbf{A}_t^{(n)} \leftarrow \mathbf{A}_{k+1}^{(n)}$$

**Fin Pour**

$k \leftarrow k + 1$

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

## 4.3 Résultat de convergence

Notre résultat de convergence concerne l'exploration d'une double convergence asymptotique : par rapport au nombre d'observations qui croît par rapport au temps ainsi que le nombre d'itérations qui croît vers l'infini. On considère deux tours de boucles : un pour la *descente de coordonnées par blocs* et un pour le flux des observations. Notre raisonnement analytique repose sur les hypothèses naturelles suivantes :

— Les observations $\mathcal{X}_t$, représentant chacune un tenseur, sont tirées d'une distribution de probabilité avec un support compact : ceci est une hypothèse naturelle étant donné qu'on travaille uniquement avec des valeurs finies issues de capteurs.

— On choisit la pénalité $\ell_1$ pour le tenseur d'activation et la pénalité de Frobenius (ou pénalité $\ell_2$) pour les matrices de dictionnaires :

$$\Omega_1(\mathcal{G}) = \alpha\theta\|\mathcal{G}\|_1$$

$$\Omega_2(\mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}) = \frac{\alpha(1-\theta)}{2}\sum_{n=1}^{N}\|\mathbf{A}^{(n)}\|_F^2$$

$$\alpha > 0, 0 \leq \theta < 1$$

Ce choix est motivé par le fait que ces pénalités sont les plus courantes dans le cadre de l'apprentissage de dictionnaires. Les diverses valeurs de $\alpha$ et $\theta$ impactent la nature des contraintes :

1. $\alpha = 0$ correspond au problème sans contrainte (car les deux fonctions $\Omega_1$ et $\Omega_2$ sont nulles),
2. $\theta = 1, \alpha \neq 0$ revient à considérer uniquement des contraintes de parcimonie sur les coefficients d'activation ($\Omega_2$ est une fonction nulle, mais pas $\Omega_1$),
3. $\theta = 0, \alpha \neq 0$ équivaut à imposer uniquement la bornitude des matrices de dictionnaires ($\Omega_1$ est une fonction nulle, mais pas $\Omega_2$).

— Pour les problèmes de minimisation considérés, on suppose que le tenseur d'activation $\mathcal{G}$ et les matrices de dictionnaires appartiennent à des ensembles compacts : $\mathcal{G} \in \mathbb{K}_{\mathcal{G}}, \mathbf{A}^{(n)} \in \mathbb{K}_{I_n,J_n}$ : ceci correspond à une interprétation naturelle de la pénalité $\ell_1 - \ell_2$ (pénalité introduite par la somme $\Omega_1(\mathcal{G}) + \Omega_2\left(\mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}\right)$ ) ;

— Le problème de minimisation sous-jacent au *codage parcimonieux* (détermination du tenseur $\mathcal{G}$) admet une unique solution : cette hypothèse peut être imposée en considérant une pénalité $\ell_1 - \ell_2$ (c'est-à-dire de la forme $\lambda_1\|\mathcal{G}\|_1 + \lambda_2\|\mathcal{G}\|_F^2, \lambda_1, \lambda_2 > 0$) sur le tenseur d'activation. On ignore le terme de pénalité $\ell_2$ pour des raisons de simplicité et cela ne change strictement rien à notre raisonnement analytique.

— Pour la *descente de coordonnées*, on cherche $\mathbf{A}_{k+1}^{(n)}$ dans une boule centrée autour de $\mathbf{A}_k^{(n)}$ de rayon $\frac{1}{k^{\frac{1}{2}}}$, c'est-à-dire $\|\mathbf{A}_{k+1}^{(n)} - \mathbf{A}_k^{(n)}\|_F^2 \leq \frac{1}{k}$. Une manière naturelle d'inciter $\mathbf{A}_{k+1}^{(n)}$ à respecter cette contrainte est d'augmenter la fonction de coût pour le problème de mise à jour de $\mathbf{A}^{(n)}$ du terme $\beta\|\mathbf{A}^{(n)} - \mathbf{A}_{k+1}^{(n)}\|_F^2, \beta > 0$.

— Pour chaque valeur de $k$, $\mathbf{A}_k^{(n)}$ est un point intérieur de $\mathbb{K}_{I_n,J_n}$ ;

Sous ces hypothèses, la suite $\left\{\mathbf{A}_k^{(1)}, .., \mathbf{A}_k^{(N)}\right\}$ générée par **Algorithme 6** converge vers l'ensemble des points stationnaires de la fonction de coût du problème (4.5) quand $k$ tend vers $+\infty$. Plus précisément, on prouve que :

$$\forall 1 \leq n \leq N, \lim_{k\to\infty} \|\frac{\partial f}{\partial \mathbf{A}^{(n)}}\left(\mathbf{A}_k^{(1)}, ..., \mathbf{A}_k^{(N)}\right)\|_F = 0 \tag{4.16}$$

avec :

$$\mathbf{A}_{k+1}^{(n)} = \begin{cases} \lim_{t\to\infty} \mathbf{A}_{k+1,t}^{(n)} & \text{si } \mathbf{A}_{k+1,t}^{(n)} \text{ converge} \\ \arg\min_{\mathbf{A}^{(n)}} \widehat{f}_\infty(\mathbf{A}^{(n)}) & \text{dans le cas contraire} \end{cases}$$

$$\mathbf{A}_{k+1,t}^{(n)} = \underset{\mathbf{A}^{(n)}\in\mathbb{K}_{I_n,J_n}}{\arg\min}\ \widehat{f}_{n,t}(\mathbf{A}^{(n)})$$

$$\widehat{f}_\infty(\mathbf{A}^{(n)}) = \frac{1}{2}\mathrm{Trace}\left(\left(\tilde{\xi}_t + \frac{\alpha(1-\theta)}{2}\mathbf{I}\right)\mathbf{A}^{(n)T}\mathbf{A}^{(n)}\right) - \mathrm{Trace}\left(\mathbf{A}^{(n)}\tilde{\eta_t}\right)$$

L'expression $\mathrm{Trace}$ représente la trace d'une matrice. La suite $\tilde{\xi}_t$ et $\tilde{\eta_t}$ incarnent des points d'accumulation (dont l'existence est assurée par la bornitude) des suites $\xi_t$ et $\eta_t$ définies par :

$$\xi_t = \frac{1}{t}\sum_{i=1}^{t} \Gamma_i\Gamma_i^T, \eta_t = \frac{1}{t}\sum_{i=1}^{t} \Gamma_i\mathbf{X}_i^{(n)T}, \Gamma_i = \mathbf{G}_i^{(n)} \otimes_{q\in I_{n-1}} \mathbf{A}_{k+1}^{(q)T} \otimes_{p\in I_N^{n+1}} \mathbf{A}^{(p)T}$$

**Esquisse de la démonstration**

Pour établir notre résultat de convergence, on considère une stratégie en deux étapes :

**Première étape** : on démontre que la séquence $\left\{\mathbf{A}_{k+1,t}^{(n)}\right\}_{t\in\mathbb{N}}$ pour $k$ fixé converge vers un point stationnaire de la fonction $\mathbf{A}^{(n)} \to f(\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(N)})$, $f$ étant l'équation définie par (4.5). Ce travail est principalement inspiré du travail effectué dans [86], qui ne peut pas être appliqué directement car les deux fonctions de coût sont distinctes (pour notre problème, on a plusieurs matrices de dictionnaires tandis que dans [86], il y en a qu'une seule). Pour contourner ce problème, on établit certaines inégalités algébriques et utilise certaines propriétés connues sur le produit de **Kronecker**.

**Deuxième étape** : à partir de la première étape ainsi que des hypothèses mises en place, on prouve que $\left\{\mathbf{A}_k^{(1)}, ...\mathbf{A}_k^{(N)}\right\}$ convergence vers un point stationnaire de la fonction de coût donnée par l'équation (4.5).

Pour la démonstration complète du résultat de convergence, consultez Annexe 2.

**Remarque 3** *: l'idée de la double convergence asymptotique est inspirée des travaux effectués dans [82] qui établissent des bornes de convergence tributaires aussi bien du numéro de l'itération que du nombre d'échantillons.*

## 4.4 Extensions

L'objet de cette section est de proposer des extensions de **OTL** via l'incorporation de contraintes classiques telles que l'orthogonalité ou la non-négativité ainsi qu'une extension permettant de traiter plus d'un tenseur à la fois.

### 4.4.1 Extension minibatch

Jusqu'ici, on considérait une seule observation à la fois pour la mise à jour des facteurs. Cependant, on peut utiliser plusieurs tenseurs $\{\mathcal{X}_1, .., \mathcal{X}_\rho\}, \mathcal{X}_i \in \mathbb{R}^{I_1 \times ...; \times I_N}$ via la stratégie suivante :

- Appliquer l'algorithme **Algorithme 4** $\rho$ fois pour inférer les tenseurs d'activation $\mathcal{G}_r$ associées observation $\mathcal{X}_j, 1 \leq j \leq \rho$.
- Appliquer l'**Algorithme 5** en substituant aux équations (4.14) et (4.15) les expressions suivantes :

$$\mathbf{P}_t^{(n)} = \mathbf{P}_{t-1}^{(n)} + \sum_{r=1}^{\rho} \widehat{\mathbf{X}}_r^{(n)} \mathbf{G}_r^{(n)T} \tag{4.17}$$

$$\mathbf{Q}_t^{(n)} = \mathbf{Q}_{t-1}^{(n)} + \sum_{r=1}^{\rho} \mathbf{B}_r^{(n)} \mathbf{B}_r^{(n)T} \tag{4.18}$$

  $\widehat{\mathbf{X}}_r^{(n)}$ étant la forme matricielle par rapport au mode n du tenseur $\widehat{\mathcal{X}}_r$ défini par l'équation (4.12) et $\mathbf{B}_r^{(n)}$ la forme matricielle par rapport au mode $n$ du tenseur $\mathcal{B}_r$ défini par l'équation (4.11)

La motivation de cette extension est de réduire la complexité en temps de l'approche initiale en incorporant simultanément l'information apportée par plusieurs tenseurs tout en accélérant la décroissance de la fonction de coût.

### 4.4.2 Contraintes de non-négativité

Dans diverses applications, la donnée que l'on traite est non-négative par nature (c'est-à-dire qu'elle possède des entrées non-négatives). Pour la décomposition de données non-négatives, il est plus convenable d'imposer des contraintes de non-négativité sur tous

les facteurs latents (tenseur d'activation et matrices de dictionnaires) dans le but de maintenir l'interpretabilité physique [134]. Pour inférer des facteurs latents non-négatifs, on remplace toutes les étapes de mises à jour dans l'**Algorithme 4** et l'**Algorithme 5** par la descente de gradient projeté [128]. Ce choix est motivé par l'efficacité de cette approche pour trouver des minima locaux pour un problème de minimisation convexe avec des contraintes linéaires [128].

### 4.4.3 Contrainte d'orthogonalité

La contrainte d'orthogonalité joue un rôle important pour la décomposition de *Tucker* car elle assure l'unicité des sous-espaces générés par les facteurs latents [29]. Pour inférer des facteurs orthogonaux ( les matrices de dictionnaire vérifiant l'inégalité $\mathbf{A}^{(n)\mathsf{T}}\mathbf{A}^{(n)} = \mathrm{I}$), on applique la descente de gradient sur une variété de Stiefel définie par $\mathsf{St}(\mathrm{I}_n, \mathrm{J}_n) = \left\{\mathbf{A} \in \mathbb{R}^{\mathrm{I}_n \times \mathrm{J}_n}, \mathbf{A}^\mathsf{T}\mathbf{A} = \mathrm{I}\right\}$. Considérons le problème de mise à jour de $\mathbf{A}^{(n)}$ avec une contrainte d'orthogonalité et notons $\mathbf{A}^{(n)}_{\mathtt{iter}}$ la valeur de $\mathbf{A}^{(n)}$ à l'itération $\mathtt{iter}$. La différence avec la *descente de gradient* classique réside dans la stratégie de mise à jour définie par :

$$\mathbf{A}^{(n)}_{\mathtt{iter}+1} = \Psi_{\mathbf{A}^{(n)}_{\mathtt{iter}}}\left(-\eta \times \Pi_{\tau_{\mathbf{A}^{(n)}_{\mathtt{iter}}}(\mathsf{St}(\mathrm{I}_n,\mathrm{J}_n))}\left(\frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}}(\mathbf{A}^{(n)}_{\mathtt{iter}})\right)\right)$$

avec :

$$\frac{\partial \widehat{f}_{n,t}}{\partial \mathbf{A}^{(n)}} = -\frac{1}{t}\sum_{i=1}^{t}\left(\widehat{\mathbf{X}}^{(n)}_i \mathbf{G}^{(n)\mathsf{T}}_i - \mathbf{A}^{(n)}\mathbf{B}^{(n)}_i\mathbf{B}^{(n)\mathsf{T}}_i\right)$$

$\widehat{\mathbf{X}}^{(n)}_i$ et $\mathbf{B}^{(n)}$ représentant les formes matricielles par rapport au mode $n$ de $\widehat{\mathcal{X}}_i$ et $\mathcal{B}_i$ définies par les équations (4.12) et (4.11). L'opérateur $\Pi_{\tau_{\mathbf{A}^{(n)}_{\mathtt{iter}}}(\mathsf{St}(\mathrm{I}_n,\mathrm{J}_n))}$ correspond à la projection sur l'espace tangent de $\mathsf{St}(\mathrm{I}_n, \mathrm{J}_n))$ au point $\mathbf{A}^{(n)}_{\mathtt{iter}}$ et $\Psi_{\mathbf{A}^{(n)}_{\mathtt{iter}}}$ la rétraction sur $\mathsf{St}(\mathrm{I}_n, \mathrm{J}_n)$ au point $\mathbf{A}^{(n)}_{\mathtt{iter}}$. La rétraction sur une variété de Stiefel est calculée via la décomposition $\mathsf{QR}$ [1]. La projection de $\mathbf{A}$ sur l'espace tangent à $\mathsf{St}(p, q)$ au point $\mathbf{X}$ est donnée par [93] :

$$(\mathrm{I}_p - \mathbf{X}\mathbf{X}^\mathsf{T})\mathbf{A} + \frac{1}{2}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{A} - \mathbf{A}^\mathsf{T}\mathbf{X})$$

Le schéma de mise à jour pour les tenseurs d'activation $\mathcal{G}_t$ est identique à celui du cas sans contraintes.

**Remarque 4** . *Le résultat de convergence n'est pas établi pour l'extension minibatch ainsi que les variantes avec des contraintes de non-négativité et d'orthogonalité.*

## 4.5 Analyse de la complexité en temps

Dans cette section, on présente le gain en complexité de notre méthode ainsi certaines de ses variantes. Les complexités (en temps) par étape de mise à jour pour le **codage parcimonieux** et la **descente de coordonnées par blocs** sont données dans le tableau 4.1. De l'étude cette table, on observe que l'extension **OTLminibatch** nécessite moins de calcul par rapport à **OTL**. En effet, pour $\rho \geq 1$ nouvelles observations $\{\mathcal{X}_i\}_{1\leq i\leq \rho}$, considérons deux stratégies de mises à jour **StrategyI** et **StrategyII**, qui représentent respectivement les mises à jour des matrices de saturation en appliquant **OTLsingle** et son extension **OTLminibatch**. Pour le *codage parcimonieux*, la complexité de la mise à jour est de $\mathcal{O}(2\rho N \prod_{k=1}^{N} I_k J_k^2)$ pour les deux stratégies. Pour l'étape de **descente de coordonnées par blocs**, on a les observations suivantes :

- décomposition sans contraintes : **StrategyII** nécessite $(\rho-1)\left(I_n \prod_{k=1}^{N} J_k + 13 I_n J_n\right)$ opérations de moins que **StrategyI** par étape de mise à jour ;
- décomposition non-négative : le gain en terme de complexité de **StrategyII** par rapport à **StrategyI** est égal à $(\rho-1)\left(I_n \prod_{k=1}^{N} J_k + 14 I_n J_n\right)$ par étape de mise à jour ;
- décomposition avec contrainte d'orthogonalité : **StrategyII** nécessite $(\rho-1)\left(I_n J_n + J_n^2 + I_n \prod_{k=1}^{N} J_k\right) + (\rho-1) J_n^2 \prod_{k=1,k\neq n}^{N} I_k$ opérations de moins que **StrategyI** par étape de mise à jour.

Conformément à l'intuition, l'extension **OTLminibatch** nécessite donc moins de temps de calcul que **OTL** quel que soit la nature de la contrainte utilisée.

| **OTLsingle** | | |
|---|---|---|
| | codage parcimonieux | descente de coordonnées par blocs ($\mathbf{A}^{(n)}$) |
| Sans contrainte | $N \prod_{k=1}^{N} I_k J_k + 2N \prod_{k=1}^{N} I_k J_k^2 + 10 \prod_{k=1}^{N} J_k$ | $6 I_n J_n + I_n J_n^2 + J_n^2 + I_n \prod_{k=1}^{N} J_k + J_n^2 \prod_{k\neq n} I_k$ |
| Non-négativité | $N \prod_{k=1} I_k J_k + 2N \prod_{k=1}^{N} I_k J_k^2 + 6 \prod_{k=1}^{N} J_k$ | $7 I_n J_n + I_n J_n^2 + J_n^2 + I_n \prod_{k=1}^{N} J_k + J_n^2 \prod_{k\neq n} I_k$ |
| Orthogonalité | $N \prod_{k=1}^{N} I_k J_k + 2N \prod_{k=1}^{N} I_k J_k^2 + 10 \prod_{k=1}^{N} J_k$ | $7 I_n J_n + 6 I_n J_n^2 + 2 J_n^2 + I_n^2 (1 + 2 J_n) + I_n \prod_{k=1}^{N} J_k + J_n^2 \prod_{k\neq n} J_k$ |
| **OTLminibatch** | | |
| Sans contrainte | $\rho N \prod_{k=1}^{N} I_k J_k + 2\rho N \prod_{k=1}^{N} I_k J_k^2 + 10\rho \prod_{k=1}^{N} J_k$ | $5 I_n J_n + I_n J_n^2 + \rho(I_n J_n + I_n \prod_{k=1}^{N} J_k + J_n^2 + J_n^2 \prod_{k\neq n} I_k)$ |
| Non-négativité | $\rho N \prod_{k=1}^{N} I_k J_k + 2\rho N \prod_{k=1}^{N} I_k J_k^2 + 6\rho \prod_{k=1}^{N} J_k$ | $6 I_n J_n + I_n J_n^2 + \rho(I_n J_n + I_n \prod_{k=1}^{N} J_k + J_n^2 + J_n^2 \prod_{k\neq n} I_k)$ |
| Orthogonalité | $\rho N \prod_{k=1}^{N} I_k J_k + 2\rho N \prod_{k=1}^{N} I_k J_k^2 + 10\rho \prod_{k=1}^{N} J_k$ | $I_n^2 (1 + 2 J_n) + 6 I_n J_n + 6 I_n J_n^2 + J_n^2 + \rho(I_n J_n + I_n \prod_{k=1}^{N} J_k + J_n^2 + J_n^2 \prod_{k\neq n}^{N} I_k)$ |

TABLE 4.1 – Complexité en temps pour une étape de mise à jour : *descente de gradient* pour $\mathbf{A}^{(n)}$ et *descente de gradient proximale* pour le tenseur d'activation

**Remarque 5** . *Ces complexités ont été calculées en considérant que les fonctions* $\Omega_1$ *et* $\Omega_2$ *sont respectivement définies par les normes* $\ell_1$ *et de Frobenius. La contrainte d'orthogonalité est appliquée via la décomposition* QR *qui à son tour est calculée par l'algorithme*

***Modified Gram-Schmidt*** *[43].*

## 4.6 Simulations numériques

Dans cette section, on évalue l'efficacité (en terme d'erreur de reconstruction) de notre approche sur des données aussi bien synthétiques que réelles. Pour notre évaluation, on considère les méthodes suivantes :

- **TuckerBatch** : cette approche infère les facteurs en traitant tous les tenseurs de la séquence en une seule fois. Le schéma de résolution est basé sur la minimisation alternée : le problème de mise à jour des matrices de dictionnaires est résolu par la **descente de gradient** et pour le tenseur d'activation, le problème est résolu par la technique de minimisation proximale.
- **OTLsingle** : cela correspond à la méthode de décomposition en ligne proposée dans ce chapitre avec l'utilisation d'une seule observation à la fois ;
- **OTLminibatch** : cela correspond à l'extension de **OTLsingle** proposée avec l'utilisation de plusieurs observations à la fois ;
- **ALTO** : cette approche proposée dans [59], est une méthode de l'art pour la décomposition de **Tucker** en ligne. Sa différence avec **OTLsingle** est basée sur la stratégie de mise à jour des facteurs : projection sur des sous-espaces générés aléatoirement pour **ALTO**, résolution numérique de problèmes de minimisation pour **OTLsingle**. La méthode **ALTO** assure une convergence si les données vérifient l'hypothèse dite de faible rang [59] : le rang des formes matricielles des observations est inférieur à $\mathsf{R}$ avec $(\mathsf{R}, \mathsf{R}, \mathsf{R})$ étant la dimension du tenseur d'activation.

L'objectif de ces expériences est double :

- Premièrement, on illustre que notre approche donne des résultats similaires par rapport à la méthode qui consiste à traiter toutes les observations de la séquence en une seule fois avec beaucoup moins de temps de calcul.
- Deuxièmement, on montre le caractère générique de notre approche : on démontre numériquement que notre méthode donne des résultats similaires à **ALTO** dans un cadre favorable à **ALTO** et donne de meilleurs résultats quand certaines spécificités de **ALTO** ne sont pas vérifiées. Contrairement au premier point, notre intérêt concerne à ce niveau la précision de la décomposition.

On considère également les variantes de notre approche **OTL** avec des contraintes de non-négativité et d'orthogonalité. On se référera à ces décompositions comme **OTL** suivi du nom de la contrainte : **OTLsingle-unconstrained** pour la décomposition sans contrainte, **OTLsingle-nonnegative** pour la décomposition avec des contraintes de non-négativité,

**OTLsingle-ortho** pour les contraintes d'orthogonalité (les mêmes dénominations sont valables pour **OTLminibatch**). Les pénalités $\Omega_1$ et $\Omega_2$ sont respectivement définies par les normes $\ell_1$ et de Frobenius :

$$\Omega_1(\mathcal{G}) = \alpha\theta\|\mathcal{G}\|_1, \Omega_2(\mathbf{A}^{(1)}, ...., \mathbf{A}^{(N)}) = \frac{\alpha(1-\theta)}{2}\sum_{n=1}^{N}\|\mathbf{A}^{(n)}\|_F^2, \alpha > 0, 0 \leq \theta < 1$$

.

Les fonctions de pénalité verifient les hypothèses imposées sur $\Omega_1$ et $\Omega_2$ puisque $\Omega_1$ admet un opérateur proximal, qui est l'opérateur de *Soft thresholding* [100] et $\Omega_2$ est différentiable. La rétraction choisie pour l'incorporation des contraintes d'orthogonalité est celle qui est basée sur la décomposition QR [93], qui applique la contrainte en sélectionnant la matrice $\mathbf{Q}$ dans la décomposition QR. On définira comme "1 epoch" le fait de traiter toutes les observations de la séquence une seule fois et comme $m$ epochs le fait de traiter toutes les observations de la séquence exactement $m$ fois.

### 4.6.1 Prédiction de données spatio-temporelles

La prédiction de données spatio-temporelle est la tâche consistant à prédire les valeurs futures d'une série temporelle multivariée. Pour cette tâche, on considère un modèle $Var(L)$ [59] qui est défini comme suit : étant donné un tenseur d'ordre 3 $\mathcal{X} \in \mathbb{R}^{P\times T\times M}$ (dont les modes représentent respectivement les lieux, le temps et les variables), ce modèle détermine un paramètre tensoriel $\mathcal{W}$ [59] via la résolution numérique d'un problème de minimisation donné par :

$$\min_{\mathcal{W}}\left\{f(\mathcal{W}) \triangleq \|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2 + \mu\sum_{m=1}^{M}\text{Trace}\left(\widehat{\mathbf{X}}_{:,:,m}^T\mathbf{S}\widehat{\mathbf{X}}_{:,:,m}\right)\right\} \tag{4.19}$$

sous la contrainte

$$\widehat{\mathbf{X}}_{:,t,m} = \mathbf{W}_{:,:,m}\mathbf{X}_{:,t,m}^L$$

$$\mathbf{X}_{:,t,m}^L = [\mathbf{X}_{:,t-1,m}^T, .., \mathbf{X}_{:,t-L,m}^T]^T, \text{rang}(\mathbf{W}^{(n)}) \leq R$$

Les matrices $\mathbf{W}_{:,:,m}$ et $\mathbf{W}^{(m)}$ correspondent respectivement au $m^{ième}$ sous-tenseur de $\mathcal{W}$ par rapport au troisième mode et la forme matricielle par rapport au mode $m$ de $\mathcal{W}$. Les fonctions $Trace$ et $rang$ représentent respectivement la trace et le rang d'une matrice. Le rôle de la matrice de similarité $\mathbf{S}$, prédéfinie, est d'assurer la consistance locale, c'est-à-dire que les variables auxquelles on s'intéresse ne sont pas significativement différentes pour deux lieux qui sont relativement proches. Le paramètre $L$ représente un pas de décalage ou nombre d'observations antérieures, c'est-à-dire la valeur à la date $t$ de la série

temporelle est tributaire des L valeurs antérieures dont les indices sont $t-1, ..., t-L$, d'où la dénomination "pas de décalage". Il est à noter que le problème ne peut pas être découpé en des problèmes de minimisation indépendants (par rapport aux variables $\{\mathbf{W}_{:,:,m}\}_{1\leq m\leq M}$) du fait des contraintes sur les rangs des formes matricielles de $\mathcal{W}$.

#### 4.6.1.1 Schéma de résolution

Une expression alternative de $f(\mathcal{W})$ est donnée par :

$$f(\mathcal{W}) = \sum_{t=1}^{T}\sum_{m=1}^{M} \|\mathbf{X}_{:,t,m} - \mathbf{W}_{:,:,m}\mathbf{X}^{L}_{:,t,m}\|_F^2 + \sum_{t=1}^{T}\sum_{m=1}^{M} \mathbf{z}_t \left(\mathbf{W}^{T}_{:,:,m}\mathbf{S}\mathbf{W}_{:,:,m}\right)\mathbf{z}_t^{T} \tag{4.20}$$

avec $\mathbf{z}_t = (\mathbf{X}^{L}_{:,:,m})^{T}_{t,:}$ représentant la ligne $t$ de la matrice $(\mathbf{X}^{L}_{:,:,m})^{T}$.
Une méthode de résolution numérique nommée **ALTO** a été proposée dans [59] pour le problème (4.19). Le principe est d'inférer le paramètre tensoriel $\mathcal{W}$ via une stratégie en deux étapes. La première étape consiste à découper le tenseur $\mathcal{X}$ en des sous-tenseurs par rapport aux premier et second modes et de mettre à jour séquentiellement le paramètre $\mathcal{W}$ (cela est possible grâce à l'équation (4.20)). Notons par $\mathcal{W}_t$ le $t^{\text{iè me}}$ paramètre tensoriel de la séquence. La seconde étape consiste à projeter $\mathcal{W}_t$ dans un espace de faible rang via des projections aléatoires en utilisant les facteurs inférés de $\mathcal{W}_{t-1}$. Donc, pour effectuer une comparaison plus juste entre notre méthode et **ALTO**, on choisit une approche en deux étapes : la première est identique au schéma de résolution de **ALTO** et la seconde est remplacée par **OTLsingle**. Ce schéma de résolution sera appelée **OTLsingle**. On considère une troisième approche appelée **TuckerBatch** suivant le même principe que **ALTO** avec la seconde étape qui est remplacée par la décomposition de **Tucker** standard.

#### 4.6.1.2 Base de données

##### 4.6.1.2.1 Données synthétiques

On considère une donnée synthétique $\mathcal{X} \in \mathbb{R}^{P\times T\times M}$ définie par :
$\mathcal{X}_{:,t,m} = \mathcal{W}_{:,:,m}\mathcal{Y}_{:,t,m} + \mathcal{E}_{:,t,m}$ avec $\mathcal{W} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}$, $\mathcal{G} \in \mathbb{R}^{R\times R\times R}$ étant un tenseur diagonal (tenseur pour lequel tous les éléments extra-diagonaux, c'est-à-dire ceux qui ne sont pas de la forme $\mathcal{G}_{r,r,r}, 1 \leq r \leq R$ sont nuls) dont le $r^{\text{iè me}}$ élément diagonal $\mathcal{G}_{r,r,r}$ est égal à $r$. Les entrées des matrices $\left\{\mathbf{A}^{(n)}\right\}_{1\leq n\leq 3}$ sont tirées d'une distribution uniforme sur $[0, 1]$ et le bruit $\mathcal{E}$ est un tenseur dont les entrées sont tirées selon une loi gaussienne centrée avec un écart type fixé à $\frac{1}{2}$. La matrice de similarité $\mathbf{S}$ est définie par : $\mathbf{S} = \mathbf{B}\mathbf{B}^{T} + 10^{-1} \times \mathbf{I}$ ($\mathbf{I}$ étant la matrice identité et $\mathbf{B}$ une matrice aléatoire dont les

entrées sont tirées selon une loi gaussienne centrée avec un écart type de $\frac{1}{2}$). Ce choix est motivé par le fait que le problème (4.19) est facile à résoudre quand la matrice de similarité est définie positive [59]. Pour une comparaison plus juste, tous les paramètres que les méthodes ont en commun sont choisis de manière identique. Les paramètres $\alpha$ et $\theta$ sont fixés respectivement $10^2$ et $10^{-2}$. Le tenseur activation ainsi que les entrées des matrices de dictionnaires sont tirées d'une distribution gaussienne standard. Le nombre d'epochs (1 epoch correspond "au fait de traiter toutes les observations de la séquence une fois") est fixé à 1.

#### 4.6.1.2.2 Données réelles

On utilise une base de données réelle contenant les notes pour différents types de lieux à Pittsburgh de 121 utilisateurs sur 1200 pas de temps repartis sur 4 mois.
La valeur du rang est déterminée par validation croisée (5-fold) parmi 10 valeurs de 3 à 12. Les hyperparamètres $\alpha$ et $\theta$ sont fixées à $10^2$ et $10^{-2}$ et le nombre d'epochs est fixé à 1.

### 4.6.1.3 Critère d'évaluation

Le critère d'évaluation est la **MSE** (mean square error ou erreur moyenne quadratique) et est défini par :

$$\mathbf{MSE} = \frac{1}{MT} \sum_{m=1}^{M} \sum_{t=L+1}^{T} \|\mathbf{X}_{:,t,m} - \mathbf{W}_{:,:,m}\mathbf{X}^{L}_{:,t,m}\|_F^2$$

Le paramètre tensoriel $\mathcal{W}$ est déterminé en utilisant un tenseur d'entraînement $\mathcal{X}_{train}$ et la **MSE** est évaluée utilisant un tenseur $\mathcal{X}_{test}$ différent de $\mathcal{X}_{train}$.

### 4.6.1.4 Résultats

#### 4.6.1.4.1 Données synthétiques

Les valeurs de la **MSE** et du temps de calcul sont déterminées en calculant la moyenne sur 10 bruits indépendants et reportées sur la figure 4.1 en fonction du pas de décalage L. Notre approche donne des résultats similaires comparées à **TuckerBatch** avec moins de temps de calcul et donne des erreurs plus faibles par rapport à **ALTO**. La méthode **ALTO** est moins coûteuse en temps de calcul par rapport aux autres méthodes car elle effectue des projections aléatoires tandis que les autres résolvent numériquement des problèmes de minimisation.

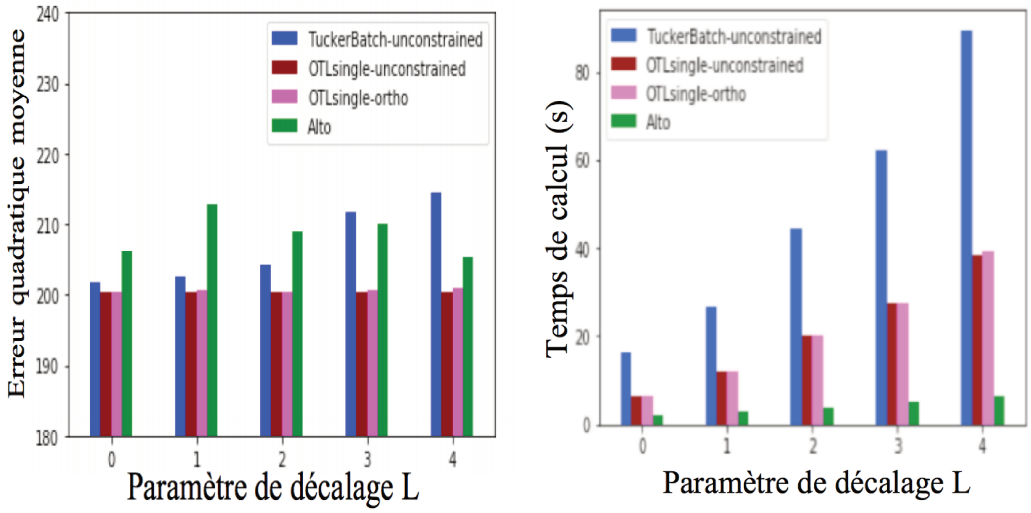


FIGURE 4.1 – Prédiction de données spatio-temporelles synthétiques : gauche (**MSE** avec 10 tirage indépendants de nombres aléatoires ), droite (*Temps de calcul*)

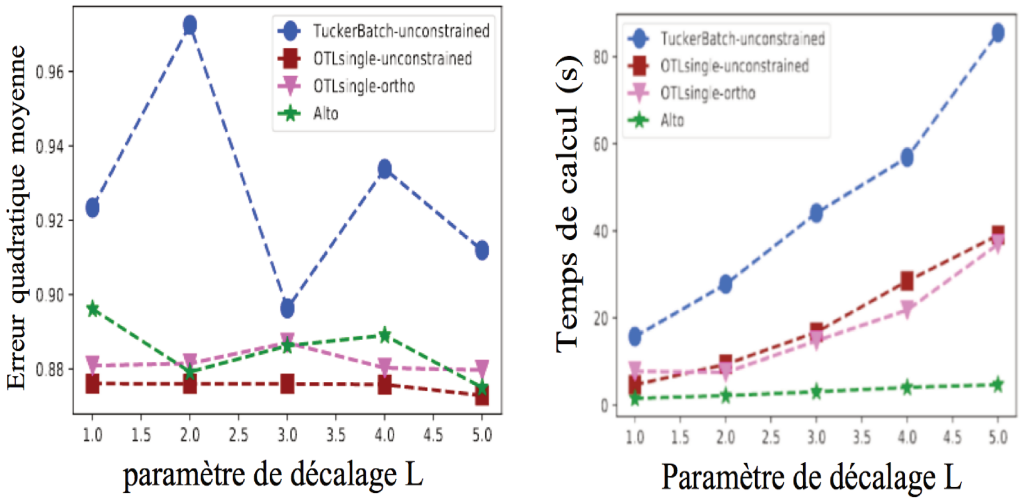


FIGURE 4.2 – Prédiction de données spatio-temporelles réelles : gauche (**MSE**), droite (**Temps de calcul**)

#### 4.6.1.4.2 Données réelles

La figure 4.2 fournit la **MSE** et le temps de calcul. Notre approche donne de meilleurs résultats par rapport à ses compétiteurs avec moins de temps de calcul que **TuckerBatch** et plus que **ALTO**. On remarque que le temps de calcul augmente avec le paramètre L dû au fait que l'augmentation de L revient à effectuer une régression sur plus d'observations antérieures.

### 4.6.2 Apprentissage de dictionnaires multimodaux

#### 4.6.2.1 Régression sur des données synthétiques

La donnée synthétique considérée est une séquence de 20.000 tenseurs d'ordre 3 de taille $30 \times 30 \times 30$ divisée en deux sous-ensembles $\mathbf{S}_{train}$ et $\mathbf{S}_{test}$ de cardinaux respectifs 15000 et 5000. L'ensemble $\mathbf{S}_{train}$ est utilisé pour l'inférence des matrices de dictionnaires et le critère d'évaluation est l'erreur sur l'ensemble de test $\mathbf{S}_{test}$. Chaque tenseur $\mathcal{X}_t$ (pour les ensembles d'entraînement et de test) est défini par :

$$\mathcal{X}_t = \mathcal{G}_t \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}$$

Le tenseur $\mathcal{G}_t \in \mathbb{R}^{R \times R \times R}$ est un tenseur d'ordre 3 dont les entrées sont tirées selon une loi gaussienne centrée avec un écart type de $\frac{1}{5}$. Les entrées des matrices $\mathbf{A}^{(1)} \in \mathbb{R}^{30 \times R}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{30 \times R}$, $\mathbf{A}^{(3)} \in \mathbb{R}^{30 \times R}$ sont tirées d'une gaussienne centrée avec un écart type de $\frac{1}{10}$. Pour l'inférence des matrices de dictionnaires, la valeur du rang est fixée à (R,R,R), c'est-à-dire la même valeur que celle utilisée pour la définition de la séquence de tenseurs pour éviter une comparaison biaisée. Pour une comparaison plus juste, les points initiaux sont choisis identiquement pour toutes les méthodes. Les tenseurs d'activation initiaux sont définis à partir d'une gaussienne centrée d'écart type $\frac{1}{10}$ et les matrices de dictionnaires de la même distribution avec un écart type $\frac{1}{100}$. Les pas de descente pour les étapes de **codage parcimonieux** et de **descente de coordonnées par blocs** sont fixés à $\eta = 10^{-5}$. Les paramètres $\alpha$ et $\theta$ sont respectivement fixés à $10^2$ et $10^{-2}$. Les algorithmes associés aux étapes de **codage parcimonieux** (**Algorithme 4**) et **la mise à jour d'une matrice** $\mathbf{A}^{(n)}$ (**Algorithme 5**) sont arrêtés quand l'erreur d'approximation est inférieure à $10^{-3}$ ou quand 20 itérations sont atteintes (une itération correspondant à la mise à jour de toutes les matrices de dictionnaires). Le nombre d'epochs est fixé à 1. Le critère d'évaluation est le **RMSE** (root mean square error ou écart quadratique moyen) sur l'ensemble de test et est défini par :

$$\text{RMSE} = \left( \frac{1}{T} \sum_{\mathcal{X}_t \in \mathbf{S}_{test}} \| \mathcal{X}_t - \mathcal{G}_t \times_1 \mathbf{A}^{(1)}_{out} \times_2 \mathbf{A}^{(2)}_{out} \times_3 \mathbf{A}^{(3)}_{out} \|_F^2 \right)^{\frac{1}{2}}$$

avec $T$ étant le cardinal $\mathbf{S}_{test}$, $\mathbf{A}^{(n)}_{out}$ les matrices de dictionnaires inférées de $\mathbf{S}_{train}$ et $\mathcal{G}_t$ la projection de $\mathcal{X}_t$ sur $\left\{ \mathbf{A}^{(n)}_{out} \right\}_{1 \leq n \leq 3}$.

Les résultats sont donnés par la figure 4.3. L'erreur de prédiction pour notre approche est trois fois moins importante que celle induite par **ALTO**. Cela est dû au fait que la donnée synthétique ne vérifie pas l'hypothèse dite de faible rang. En plus, notre approche

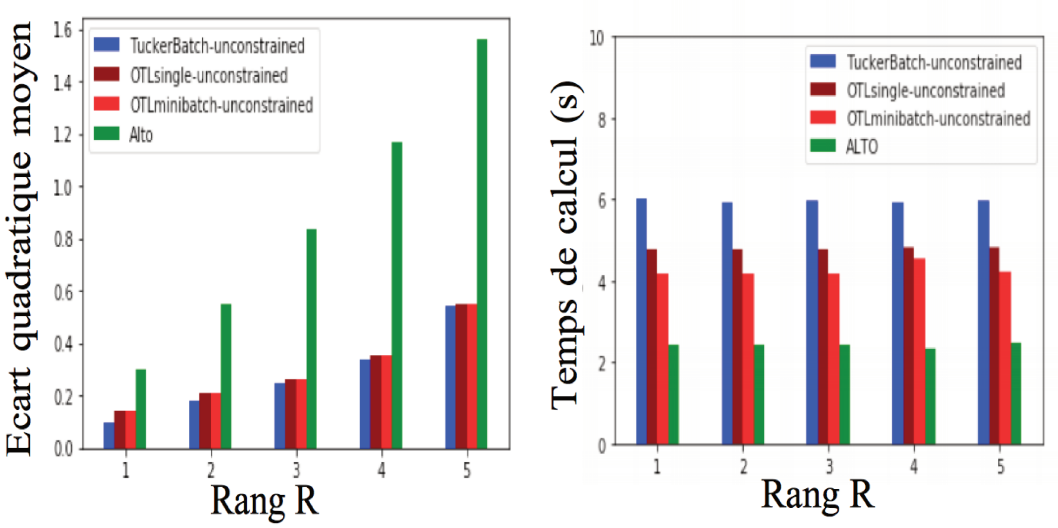


FIGURE 4.3 – Gauche : **RSME** avec six tirages indépendants de nombres aléatoires , Droite : *Temps de calcul* sur une échelle logarithmique

donne des résultats similaires à **TuckerBatch** avec moins de temps de calcul.

#### 4.6.2.2 Apprentissage de dictionnaires multimodaux non-négatifs

On compare notre approche aux compétiteurs considérés pour la tâche d'inpainting dont l'objectif est d'inférer les pixels manquants dans une image (voir la figure 4.5). Pour cette application, on construit un tenseur d'entraînement $\mathcal{S}_{train} \in \mathbb{R}^{64\times8\times8}$ en empilant suivant le premier mode des patchs d'image de taille $8 \times 8$. Le tenseur noyau ainsi que les matrices de saturation $\mathbf{A}_l \in \mathbb{R}^{8\times R}, \mathbf{A}_w \in \mathbb{R}^{8\times R}$, $\mathbf{A}_e \in \mathbb{R}^{64\times R}$ ($R$ étant un entier à déterminer) sont inférées de $\mathcal{S}_{train}$ via une décomposition non-négative. Chaque patch est reconstruit en estimant les coefficients parcimonieux obtenus via la projection des pixels non-manquants sur les facteurs latents $\mathbf{A}_l$ et $\mathbf{A}_w$, c'est-à-dire $\mathcal{P}_t^r = \mathcal{G}_t \times_1 \mathbf{A}_l \times_2 \mathbf{A}_w$ avec les composantes de $\mathcal{G}_t$ représentant les coefficients parcimonieux :

$$\mathcal{G}_t \leftarrow \underset{\mathcal{G}\geq0}{\arg\min} \|\mathcal{M} \star (\mathcal{X}_t - \mathcal{G} \times_1 \mathbf{A}_l \times_2 \mathbf{A}_w)\|_F^2 + \lambda\|\mathcal{G}\|_1$$

$\mathcal{M}$ est le tenseur définissant le masque et $\star$ représentant le produit composante par composante. Le paramètre $\lambda$ est fixé à $10^{-3}$ et le nombre d'epochs est fixé à 5. Le critère d'évaluation est le **PSNR** défini par :

$$\text{PSNR} = 10\log_{10}\left(\frac{m \times n \times 255^2}{\|\widehat{\mathbf{I}} - \mathbf{I}\|_F^2}\right)$$

Les variables $m, n, \mathbf{I}, \widehat{\mathbf{I}}$ représentent respectivement les tailles des images, les images réelle et reconstruite. Les résultats sont présentés sur les figures 4.4, 4.5 et 4.6.

Conformément à l'intuition, on remarque que la contrainte de non-négativité améliore la qualité de l'image en terme de **PSNR**, c'est-à-dire que **OTLsingle** avec des contraintes de non-négativité conduit à une meilleure valeur du **PSNR** que les approches qui n'incorporent pas cette contrainte. La même remarque est valable pour **OTLminibatch**. Il est à noter que toutes les approches donnent de meilleurs résultats par rapport à **ALTO**. Cela justifie empiriquement l'importance de la mise en place de méthodes pouvant être facilement adaptées pour prendre en compte les caractéristiques de la donnée d'intérêt.

La figure 4.4 montre la convergence de la fonction de coût par rapport au nombre d'epochs et confirme la convergence plus rapide de **OTLminibatch** (voir la figure 4.4 : gauche-bas et droite-bas) par rapport à **OTLsingle** ainsi que la convergence plus rapide de notre approche de notre approche comparée à **TuckerBatch** (voir la figure 4.4 : haut-droite).

## 4.7 Conclusion

Dans ce travail, nous avons proposé une approche de décomposition en ligne de **Tucker** nommée **OTL**, inspirée de méthodes de décomposition en ligne pour les matrices et utilisée pour l'apprentissage de dictionnaires multimodaux. Cette approche est assez flexible pour incorporer certaines contraintes classiques telles que l'orthogonalité, la non-négativité, la parcimonie. Un résultat théorique de convergence vers des points stationnaires est fourni avec des expériences numériques montrant l'efficacité de notre approche en terme de capacité de reconstruction. Une limite de cette approche résulte du fait qu'on suppose que les observations acquises séquentiellement au fil du temps ont la même taille (ce qui revient à considérer la décomposition d'un tenseur qui croît au fil du temps par rapport à un seul mode comme expliqué dans la section 4.2).

Dans le chapitre suivant, nous mettons en place une approche pour la décomposition d'un tenseur de grande taille dans un cadre statique (c'est-à-dire le tenseur d'intérêt est fixé) avec une extension pour décomposer un tenseur qui croît par rapport à tous les modes.

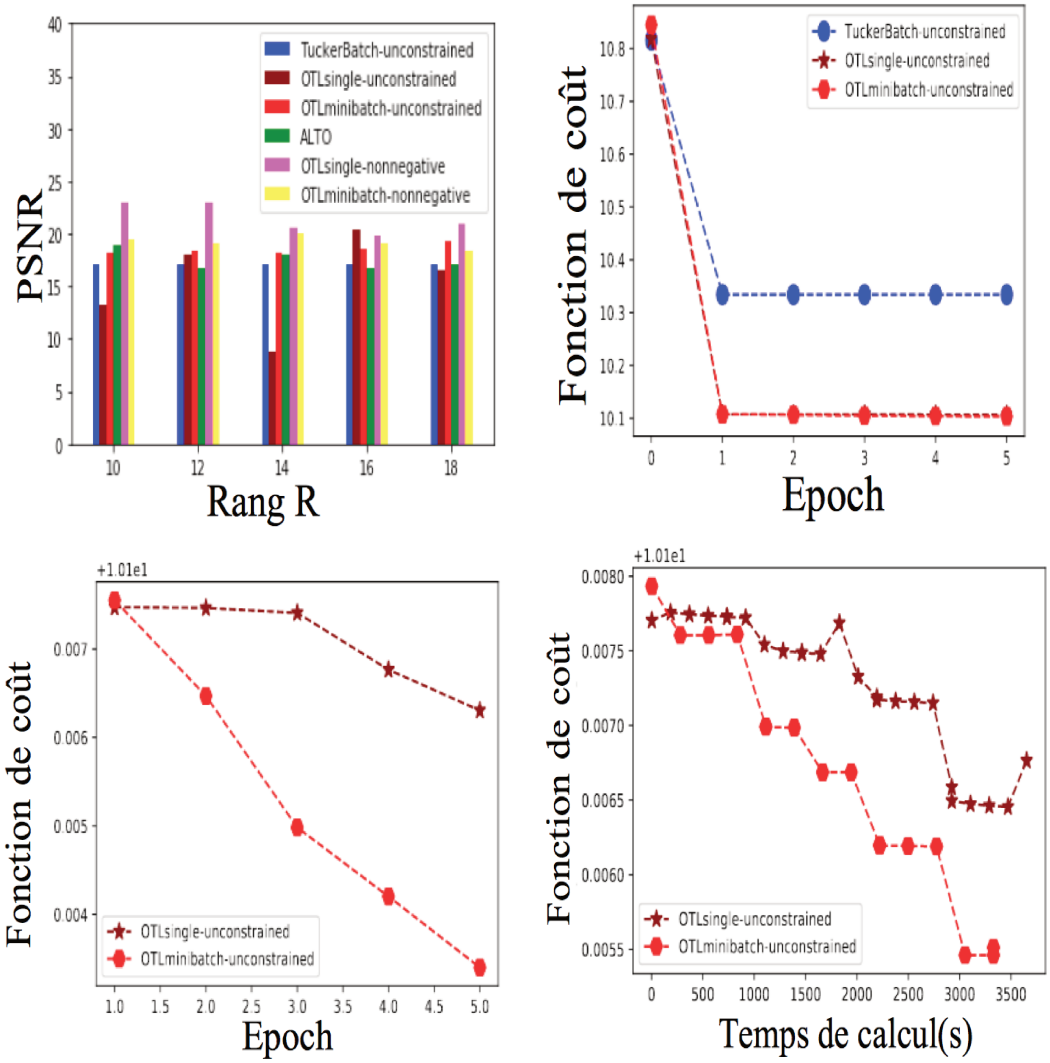


FIGURE 4.4 – **Haut-gauche** : **PSNR** (moyenne sur trois sélections aléatoires des pixels manquants , **Haut-droite** : fonction de coût sur une échelle logarithmique en fonction du nombre d objective fonction du nombre d'epochs pour la valeur du Rang égale à 16, **Bas-gauche** : fonction de coût sur une échelle logarithmique en fonction du nombre d'epochs pour la valeur du rang égale à 16, **Bas-droite** : fonction de coût en fonction du temps



FIGURE 4.5 – résultats pour la tâche d'inpainting de gauche à droite : image avec 50% de pixels manquants, image réelle, **OTLsingle-nonnegative**, **OTLsingle-unconstrained**

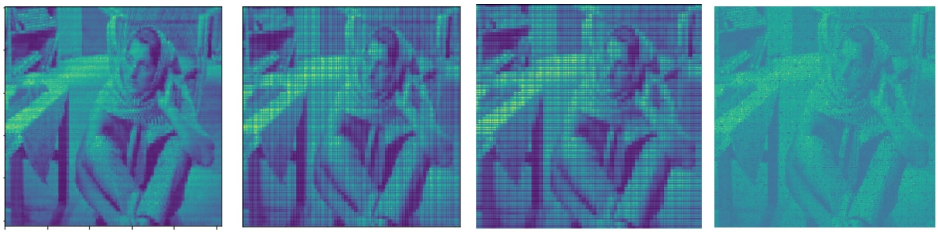

FIGURE 4.6 – résultats d'inpainting de gauche à droite : *OTLminibatch-nonnegative*, *OTLminibatch-unconstrained*, *Tucker-unconstrained*, *ALTO*

CHAPITRE 5

# SINGLESHOT, UNE APPROCHE PAR MORCEAUX POUR LA DÉCOMPOSITION DE TUCKER

## 5.1 Contexte

Le problème d'intérêt est la décomposition de **Tucker** d'un tenseur de grande taille dans un cadre statique (c'est-à-dire que le tenseur est fixé), qui est un problème intéressant pour deux raisons principales : premièrement, on ne peut pas toujours charger en mémoire un tenseur d'intérêt et deuxièmement, le processus de décomposition peut résulter en un problème de gestion de mémoire [67].
Les approches existantes pour ce problème reposent principalement sur deux philosophies différentes dont les détails sont fournis dans la section 2.4.4. La première repose sur le principe du diviser pour mieux régner (c'est-à-dire celles qui divisent la donnée en sous-parties) [132, 22, 95, 90, 85, 104, 41, 112, 131, 61, 130, 98, 81, 25, 76]. La seconde consiste à utiliser toute la donnée de manière simultanée tout en mettant en place des stratégies pour réduire la complexité des calculs intermédiaires [13, 23, 88, 40, 106, 116, 129, 113, 10, 108, 11, 134, 56, 79, 91].
Même si certains travaux proposent des résultats de convergence [13] ou établissent une borne sur l'erreur d'approximation [6], les méthodes existantes pour ce problème ne fournissent aucune vitesse de convergence pour le problème d'optimisation traité.
Par rapport aux limitations mentionnées ci-dessus, nos contributions sont les suivantes :

1. Nous proposons une approche efficiente en terme de gestion de memoire pour le problème de décomposition de **Tucker**, nommée **Singleshot** et basée sur la *descente de gradient par blocs* ainsi que le traitement séquentiel de sous-tenseurs. Cette approche peut être étendue pour incorporer des contraintes populaires telles que

la non-négativité.

2. Pour réduire la complexité en temps de **Singleshot**, on introduit une variante nommée **Singleshotinexact** dont l'idée repose sur l'utilisation d'un gradient inexact défini en retirant certains termes de l'expression du gradient. Nous proposons également une extension de cette approche pour la décomposition en ligne d'un tenseur qui croît au fil du temps par rapport à tous les modes.
3. D'un point de vue théorique, nous établissons pour **Singleshot** une vitesse de convergence ergodique $O(\frac{1}{\sqrt{K}})$ ($K$ étant le nombre maximum d'itérations). Pour **Singleshotinexact**, on prouve la convergence de notre algorithme vers l'ensemble des minimiseurs à la vitesse $\mathcal{O}\left(\frac{1}{k}\right)$, $k$ étant le numéro de l'itération.
4. Notre approche est soutenue par des expériences numériques sur des données réelles.

## 5.2 Singleshot : une approche par morceaux pour la décomposition de Tucker

Dans cette section, nous formalisons notre problème et présentons notre algorithme ainsi que ses extensions.

### 5.2.1 Présentation du problème

Le problème traité est la décomposition de **Tucker** défini via le problème d'optimisation suivant :

$$\min_{\mathcal{G},\mathbf{A}^{(1)},\cdots,\mathbf{A}^{(N)}} \left\{ \ell\left(\mathcal{G},\mathbf{A}^{(1)},\cdots,\mathbf{A}^{(N)}\right) \triangleq \frac{1}{2}\|\mathcal{X}-\mathcal{G}\times_{m\in\mathbf{I}_N}\mathbf{A}^{(m)}\|_F^2 \right\} \tag{5.1}$$

Notre objectif est l'inférence de facteurs latents via la décomposition de **Tucker** et le traitement séquentiel de sous-tenseurs. Nous proposons pour cela un raisonnement qui repose sur la réformulation du problème (5.1) en terme de sous-tenseurs tirés de $\mathcal{X}$ par rapport à un mode (qui est préalablement fixé). Considérons un entier $n$ (dans la suite, on se référera à $n$ comme le mode de division). En effet, la fonction de coût peut être reformulée de la manière suivante (*Propriété* 2 en Annexe 3) :

$$\sum_{i_n=1}^{I_n}\frac{1}{2}\|\mathcal{X}^n_{i_n}-\mathcal{G}\underset{\substack{m\\ m\in\mathbf{I}_N\neq n}}{\times_m}\mathbf{A}^{(m)}\times_n\mathbf{A}^{(n)}_{i_n,:}\|_F^2 \tag{5.2}$$

Plus généralement, la fonction $\ell$ définie par (5.1) peut se réécrire sous forme de sous-tenseurs construits par rapport à tous les modes (*Propriété* 3 en Annexe 3). Pour des

raisons de simplicité, on considère le cas de sous-tenseurs construits par rapport à un seul mode et le cadre général peut être déduit du principe expliqué en Annexe 4.

### 5.2.2 Singleshot

Étant donné que le problème (5.1) n'admet pas de solution analytique, nous proposons une approche de résolution basée sur la *descente de gradient par blocs* [115]. L'idée sous-jacente de la méthode repose sur la mise à jour cyclique des variables $\mathcal{G}, \mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}$. Chaque étape de mise à jour consiste en une seule itération de *descente de gradient* (ce qui est possible étant donné la différentiabilité par blocs de la fonction de coût $\ell$). Plus formellement, étant donné une itération $k$, $\mathcal{G}_k, \mathbf{A}_k^{(1)}, ..., \mathbf{A}_k^{(N)}$ les valeurs des facteurs latents, les derivées $\mathcal{D}_k^{\mathcal{G}}$ et $\mathbf{D}_k^p$ de $\ell$ par rapport au tenseur noyau et la $p^{\text{ième}}$ matrice de saturation respectivement évaluées à $\left(\mathcal{G}_k, \mathbf{A}_k^{(1)}, \cdots, \mathbf{A}_k^{(N)}\right)$ et $\left(\mathcal{G}_{k+1}, \mathbf{A}_{k+1}^{(1)}, \cdots, \mathbf{A}_{k+1}^{(p-1)}, \mathbf{A}_k^{(p)}., \mathbf{A}_k^{(N)}\right)$ sont données par :

$$\mathcal{D}_k^{\mathcal{G}} = \partial_{\mathcal{G}} \ell \left(\mathcal{G}_k, \mathbf{A}_k^{(1)}, ..., \mathbf{A}_k^{(N)}\right) = \partial_{\mathcal{G}} \ell \left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) \tag{5.3}$$

$$\mathbf{D}_k^p = \partial_{\mathbf{A}^{(p)}} \ell \left(\mathcal{G}_{k+1}, \mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(p-1)}, \mathbf{A}_k^{(p)}, .., \mathbf{A}_k^{(N)}\right) = \partial_{\mathbf{A}^{(p)}} \ell \left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}, \right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right), \tag{5.4}$$

Avec ces notations, notre approche **Singleshot** est résumée par **Algorithm 7**.
Une approche naïve de calcul des gradients pourrait engendrer un problème de gestion de mémoire. Dans ce qui suit, nous démontrons que les dérivées $\mathcal{D}_k^{\mathcal{G}}$ et $\mathbf{D}_k^p, 1 \leq p \leq N$ données respectivement par les équations (5.3) et (5.4) peuvent être déterminées en traitant un seul sous-tenseur à la fois, permettant l'application de l'**Algorithme 7** via le traitement séquentiel de sous-tenseurs. Pour cette démonstration, on distingue trois cas : dérivée par rapport au tenseur noyau, dérivée par rapport à $\mathbf{A}^{(p)}, p \neq n$ et dérivée par rapport $\mathbf{A}^{(n)}$ avec $n$ étant le mode de division. Le calcul des derivées est effectué via l'expression alternative de la fonction de coût $\ell$ définie par l'équation (5.2).

***Algorithme 7 : Singleshot***

**Entrées**

$\mathcal{X}$ : tenseur à décomposer,

$n$ : mode de division,

$\left\{\mathbf{A}_0^{(m)}\right\}_{1\leq m\leq N}$ : matrices de saturation initiales

**Sortie**

$\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_{1\leq m\leq N}$

**Répéter**

Calcul du pas optimal $\eta_k^{\mathcal{G}}$

Mise à jour du tenseur noyau $\mathcal{G}$ :

$$\mathcal{G}_{k+1} \leftarrow \mathcal{G}_k - \eta_k^{\mathcal{G}} \mathcal{D}_k^{\mathcal{G}}$$

**Pour** $1 \leq n \leq N$ **faire**

Calcul du pas optimal $\eta_k^p$

Mise à jour de la matrice de saturation :

$$\mathbf{A}_{k+1}^{(p)} \leftarrow \mathbf{A}_k^{(p)} - \eta_k^p \mathbf{D}_k^p$$

**Fin Pour**

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

#### 5.2.2.1 Dérivée par rapport à $\mathcal{G}$.

La dérivée de $\ell$ par rapport à $\mathcal{G}$ évaluée au point $\left(\mathcal{G}_k, \mathbf{A}_k^{(1)}, .., \mathbf{A}_k^{(N)}\right)$ est donnée par (voir *Propriété* 7 en Annexe 3) :

$$\mathcal{D}_k^{\mathcal{G}} = \sum_{i_n=1}^{I_n} \underbrace{\mathcal{R}_{i_n} \underset{m\in\mathbf{I}_{N\neq n}}{\times_m} \left(\mathbf{A}_k^{(m)}\right)^\top \times_n \left(\left(\mathbf{A}_k^{(n)}\right)_{i_n,:}\right)^\top}_{\theta_{i_n}} \tag{5.5}$$

avec $\mathcal{R}_{i_n} = -\mathcal{X}_{i_n}^n + \mathcal{G}_k \underset{m\in\mathbf{I}_{N\neq n}}{\times_m} \mathbf{A}_k^{(m)} \times_n \left(\mathbf{A}_k^{(n)}\right)_{i_n,:}$, $\theta_{i_n} = \mathcal{R}_{i_n} \underset{m\in\mathbf{I}_{N\neq n}}{\times_m} \left(\mathbf{A}_k^{(m)}\right)^\top \times_n \left(\left(\mathbf{A}_k^{(n)}\right)_{i_n,:}\right)^\top$

$\mathcal{D}_k^{\mathcal{G}}$ (donnée par l'équation (5.5)) est donc le dernier terme de la séquence récursive $\left\{(\mathcal{D}_k^{\mathcal{G}})^j\right\}_{1\leq j\leq I_n}$ définie par :

$\left(\mathcal{D}_k^{\mathcal{G}}\right)^j = \left(\mathcal{D}_k^{\mathcal{G}}\right)^{j-1} + \theta_j$, avec $\left(\mathcal{D}_k^{\mathcal{G}}\right)^0$ étant le tenseur nul (tenseur dont toutes les entrées sont nulles).
Étant donné que le terme additif $\theta_j$ ne dépend que d'un seul tenseur $\mathcal{X}_j^n$, le calcul de $\mathcal{D}_k^{\mathcal{G}}$ peut donc être mis en oeuvre via le traitement séquentiel d'un seul sous-tenseur $\mathcal{X}_j^n$ à la fois.

#### 5.2.2.2 Dérivée par rapport à $\mathbf{A}^{(p)}, p \neq n$

Le calcul de la dérivée de $\ell$ par rapport à $\mathbf{A}^{(p)}, p < n$ ($p > n$ suit le même principe) évaluée en $(\mathcal{G}_{k+1}, \mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(p-1)}, \mathbf{A}_k^{(p)}, .., \mathbf{A}_k^{(N)})$ conduit à la formule suivante (voir *Propriété* 9 en Annexe 3) :

$$\mathbf{D}_k^p = \sum_{i_n=1}^{I_n} \left( - \left(\mathbf{X}_{i_n}^n\right)^{(p)} + \mathbf{A}_k^{(p)} \mathbf{B}_{i_n}^{(p)} \right) \left(\mathbf{B}_{i_n}^{(p)}\right)^\top \tag{5.6}$$

Les matrices $(\mathbf{X}_{i_n}^n)^{(p)}$ et $\mathbf{B}_{i_n}^{(p)}$ représentent respectivement les formes matricielles par rapport au mode $p$ du $i_n^{ème}$ sous-tenseur $\mathcal{X}_{i_n}^n$ et le tenseur $\mathcal{B}_{i_n}$ défini par :

$$\mathcal{B}_{i_n} = \mathcal{G}_{k+1} \underset{m \in \mathbf{I}_{p-1}}{\times_m} \mathbf{A}_{k+1}^{(m)} \times_p \mathbf{Id} \underset{q \in \mathbf{I}_{N \neq n}^{p+1}}{\times_q} \mathbf{A}_k^{(q)} \times_n (\mathbf{A}_k^{(n)})_{i_n,:}$$

avec $\mathbf{Id} \in \mathbb{R}^{J_p \times J_p}$ étant la matrice identité. Avec un raisonnement identique à celui de la dérivée par rapport au tenseur noyau $\mathcal{G}$, la dérivée $\mathbf{D}_k^p$ peut être calculée en traitant un seul sous-tenseur à la fois.

#### 5.2.2.3 Dérivée par rapport à $\mathbf{A}^{(n)}$.

La dérivée de $\ell$ par rapport à la matrice $\mathbf{A}^{(n)}$ peut être déterminée en empilant dans le sens de la ligne les vecteurs $\partial_{\mathbf{A}_{j,:}^{(n)}} \ell$ représentant les dérivées par rapport aux lignes $\mathbf{A}_{j,:}^{(n)}$ de la matrice $\mathbf{A}^{(n)}$ et chacune des composantes $\partial_{\mathbf{A}_{j,:}^{(n)}} \ell$ dépend uniquement de $\mathcal{X}_j^n$. En effet, considérons $1 \leq j \leq I_n$. Dans l'expression de $\ell$ donnée l'équation (5.2), le seul terme qui dépend de $\mathbf{A}_{j,:}^{(n)}$ est le terme $\|\mathcal{X}_j^n - \mathcal{G} \underset{m \in \mathbf{I}_{n-1}}{\times_m} \mathbf{A}^{(m)} \times_n \mathbf{A}_{j,:}^{(n)} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2$. Donc la dérivée $\partial_{\mathbf{A}_{j,:}^{(n)}} \ell$ ne dépend que de $\mathcal{X}_j^n$ et est formulée comme suit (voir *Propriété* 8 en Annexe 3) :

$$\partial_{\mathbf{A}_{j,:}^{(n)}} \ell \left( \mathcal{G}, \left\{ \mathbf{A}^{(m)} \right\}_1^N \right) = - \left( (\mathbf{X}_j^n)^{(n)} - \mathbf{A}_{j,:}^{(n)} \mathbf{B}^{(n)} \right) \mathbf{B}^{(n)\top}$$

Les tenseurs $(\mathbf{X}_j^n)^{(n)} \in \mathbb{R}^{1 \times \prod_{k \neq n} I_k}$ et $\mathbf{B}^{(n)}$ représentent respectivement la forme matricielle par rapport au mode-n des tenseurs $\mathcal{X}_j^n$ et $\mathcal{B}$ défini par : $\mathcal{B} = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{Id} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}$.
Par conséquent, la dérivée de $\ell$ par rapport à $\mathbf{A}^{(n)}$ peut aussi être calculée en traitant un

seul sous-tenseur à la fois.

Toutes les composantes de $\mathcal{X}$ sont traitées pour **Singleshot** et notre astuce pour appliquer **Algorithme 7** de manière efficace consiste à reformuler la fonction de coût $\ell$ en terme de sous-tenseurs $\mathcal{X}^n_{i_n}$. En plus, si l'utilisation d'un sous-tenseur crée un problème de gestion de mémoire, le calcul récursif permettant la détermination des dérivées par rapport aux facteurs latents peut être appliqué à des sous-tenseurs de la forme $\mathcal{X}_{\theta_1,...,\theta_N}, \theta_m \subset \{1, 2, .., I_m\}$ avec $(\mathcal{X}_{\theta_1,..,\theta_N})_{i_1,..,i_N} = \mathcal{X}_{i_1,..,i_N}, (i_1, .., i_N) \in \theta_1 \times ... \times \theta_N$, $\times$ représentant le produit cartésien de deux ensembles (voir Annexe 4).

### 5.2.3 *Singleshotinexact*

Tandis que tous les sous-tenseus $\mathcal{X}^n_{i_n}$ sont utilisés pour **Singleshot**, on utilise plutôt un sous-ensemble de sous-tenseurs pour **Singleshotinexact** dans le but de réduire le temps de calcul des dérivées partielles. Le principe de **Singleshotinexact** est d'utiliser, pour le calcul des gradients, uniquement $B_k$ sous-tenseurs avec $B_k < I_n$ (pour **Singleshot**, on utilise $I_n$ sous-tenseurs puisque le tenseur $\mathcal{X}$ est entièrement traité). Considérons l'ensemble $\mathcal{SET}_k$ (de cardinal $B_k$) composé des indexes des sous-tenseurs utilisés pour **Singleshotinexact** à l'itération $k$.

L'algorithme de résolution numérique pour **Singleshotinexact** est identique à **Algorithme 7** à l'exception de la définition des gradients $\mathcal{D}^{\mathcal{G}}_k$ et $\mathbf{D}^p_k$ qui sont respectivement remplacés $\widehat{\mathcal{D}}^{\mathcal{G}}_k$ et $\widehat{\mathbf{D}}^p_k, p \neq n$ définis par :

$$\widehat{\mathcal{D}}^{\mathcal{G}}_k = \sum_{i_n \in \mathcal{SET}_k} \mathcal{R}_{i_n} \underset{m \in \mathbf{I}_{N \neq n}}{\times_m} \mathbf{A}^{(m)\top}_k \times_n \left( \left( \mathbf{A}^{(n)}_k \right)_{i_n,:} \right)^\top \tag{5.7}$$

$$\widehat{\mathbf{D}}^p_k = \sum_{i_n \in \mathcal{SET}_k} \left( - \left( \mathbf{X}^n_{i_n} \right)^{(p)} + \mathbf{A}^{(p)} \mathbf{B}^{(p)}_{i_n} \right) \left( \mathbf{B}^{(p)}_{i_n} \right)^\top \tag{5.8}$$

avec les pas de descente définis comme étant $\frac{\eta^{\mathcal{G}}_k}{B_k}$ et $\frac{\eta^p_k}{B_k}, 1 \leq p \leq N$.

Il est à noter que les dérivées $\widehat{\mathbf{D}}^n_k$ ($n$ étant le mode de division) est parcimonieux : **Singleshotinexact** correspond à la minimisation de la fonction de coût $\ell$ définie par (5.2) en supprimant les termes $\left\{ \|\mathcal{X}^n_j - \mathcal{G} \underset{m \in \mathbf{I}_{N \neq n}}{\times_m} \mathbf{A}^{(m)} \times_n \mathbf{A}^{(n)}_{j,:}\|^2_F \right\}$ avec $j \notin \mathcal{SET}_k$. donc, les lignes $\left( \widehat{\mathbf{D}}^n_k \right)_{j,:}, j \notin \mathcal{SET}_k$ sont toutes égales au vecteur nul car les termes dépendant de $\left\{ \mathbf{A}^{(n)}_{j,:} \right\}_{j \notin \mathcal{SET}_k}$ n'apparaissent pas dans la nouvelle fonction de coût.

### 5.2.4 Complexité pour les sous-tenseurs par rapport à un mode

Notons par $n$ le mode de division. Pour **Singleshot** et **Singleshotinexact**, on doit avoir en mémoire le tenseur $\mathcal{X}_j^n$ de taille $\prod_{m\in\mathbf{I}_{N\neq n}} I_m = I_1..I_{n-1}I_{n+1}..I_N$, les matrices $\left\{\mathbf{A}^{(m)}\right\}_{m\in\mathbf{I}_{N\neq n}}, \mathbf{A}^{(n)}_{i_n,:}$ et la valeur du gradient. La complexité en espace est donc $\prod_{m\in\mathbf{I}_{N\neq n}} I_m + \sum_{m\neq n} I_m J_m + J_n + \mathcal{AT}$, $\mathcal{AT}$ étant la complexité en espace du gradient. Pour la mise à jour du tenseur noyau $\mathcal{G}$, $\mathcal{AT} = \prod_{m\in\mathbf{I}_N} J_m$ et pour la matrice $\mathbf{A}^{(m)}$, $\mathcal{AT} = I_m J_m$.

### 5.2.5 Variantes

Nous présentons les variantes de notre approche pour la contrainte de non-négativité sur les facteurs latents ainsi la décomposition d'un tenseur qui croît par rapport à tous les modes.

#### 5.2.5.1 *Singleshotonline*

On considère le problème de décomposition d'un tenseur qui croît au fil du temps par rapport à chaque mode avec la contrainte d'une passe unique (c'est-à-dire qu'on traite une seule fois chaque sous-partie de donnée), justifiant ainsi le terme online. Ce problème important n'a pas été traité de manière exhaustive [13].
Considérons un tenseur d'ordre $N$ qui croît au fil du temps par rapport à chacun des modes et supposons qu'à la date $t$, il est sous la forme $\mathcal{X}_t \in \mathbb{R}^{I_{1,t}\times..\times I_{N,t}}$ et au pas de temps $t+1$, nous avons acquis $q$ nouveaux tenseurs par rapport au mode $n$ $Set_n = \left\{\mathcal{X}_q^n \in \mathbb{R}^{I_{1,t}\times..\times I_{n-1,t}\times 1\times I_{n+1,t}\times..\times I_{N,t}}, 1 \leq q \leq B_n\right\}$. Conformément à la cohérence mathématique pour la décomposition de **Tucker**, $\mathbf{A}^{(n)}$ est la seule matrice dont les dimensions changent : son nombre de lignes croît de $I_{n,t}$ à $I_{n,t} + B_n$.
Pour **Singleshotonline**, on propose un schéma de mise à jour identique à celui présenté pour **Singleshotinexact** dans la section 5.2.3, avec l'intuition que le problème en ligne peut être traité de manière efficace avec **Singleshotinexact**. Plus formellement, les dérivées par rapport au tenseur noyau $\mathcal{G}$ et la matrice de saturation $\mathbf{A}^{(p)}, p \neq n$, notées par $\widehat{\mathcal{D}}_k^{\mathcal{G}}$ et $\widehat{\mathbf{D}}_k^p$ sont déterminées exactement comme dans la section 5.2.3 en fixant $\mathcal{SET}_k$ à $Set_n$. La dérivée $\widehat{\mathbf{D}}_k^n \in \mathbb{R}^{(I_{n,t}+B_n)\times J_n}$ est définie par :

$$\widehat{\mathbf{D}}_k^n = [\underbrace{\mathbf{o}, \cdots, \mathbf{o}}_{I_{n,t}\text{vecteurs nuls}}, \mathbf{v}_1, .., \mathbf{v}_{B_n}], \tag{5.9}$$

avec $[.,...,.]$ étant l'opérateur qui empile les vecteurs dans le sens de la ligne , $\mathbf{o} \in \mathbb{R}^{1\times J_n}$ étant le vecteur nul et $\mathbf{v}_j$ la dérivée de $\ell_j$ évaluée à $(\mathbf{A}_k^{(n)})_{I_{n,t}+j,:}$, et $\ell_j$ définie par :

$$\ell_j(\mathbf{a}) = \frac{1}{2}\|\mathcal{X}_j^n - \mathcal{G}_{k+1} \underset{m\in\mathbf{I}_{n-1}}{\times_p} \mathbf{A}_{k+1}^{(m)} \times_n \mathbf{a} \underset{q\in\mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}_k^{(q)}\|_F^2, \mathcal{X}_j^n \in Set_n, \mathbf{a} \in \mathbb{R}^{1\times J_n}$$

On attire l'attention sur le fait que cette approche nécessite une unique passe sur les données et la mise à jour des facteurs au pas de temps $t+1$ ne nécessite pas du tout le recours à $\mathcal{X}_t$.
Cette approche traite une variété plus large de problèmes car elle permet à traiter un tenseur qui croît par rapport à tous les modes (et donc les observations séquentiellement acquises peuvent être de tailles différentes) et contourne une limitation de l'approche **OTL** introduite dans le chapitre 5 qui traite une séquence de tenseurs de taille égale.

#### 5.2.5.2 Contraintes de non-négativité

Pour inférer des facteurs latents avec des contraintes de non-négativité, on remplace toutes opérations de mise à jour par une itération de *descente de gradient projeté* [111] :

$$\mathcal{G}_{k+1} \leftarrow \max\left(\mathcal{G}_k - \eta_k^{\mathcal{G}}\mathcal{D}_k^{\mathcal{G}}, 0\right), \eta_k^{\mathcal{G}} > 0$$
$$\mathbf{A}_{k+1}^{(p)} \leftarrow \max\left(\mathbf{A}_k^{(p)} - \eta_k^p \mathbf{D}_k^p, 0\right), \eta_k^p > 0$$

## 5.3 Analyse théorique

Considérons le problème de minimisation donné par l'équation (5.1) :

$$\min_{\mathcal{G},\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} \ell\left(\mathcal{G}, \mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}\right)$$

Dans cette section, on note les dérivées partielles par $\partial_x \ell$ ($x$ étant la variable par rapport à laquelle on dérive). La dérivée de la fonction de coût $\ell$, notée par $\nabla\ell$ et définie par $(\partial_{\mathcal{G}}\ell, \partial_{\mathbf{A}^{(1)}}\ell..\partial_{\mathbf{A}^{(N)}}\ell)$, est un élément de $\mathbb{R}^{J_1\times..\times J_N} \times \mathbb{R}^{I_1\times J_1} \times ... \times \mathbb{R}^{I_N\times J_N}$ muni de la norme $\|\cdot\|_*$ définie par la somme des normes de Frobenius. En plus, on considère pour des raisons de simplicité d'écriture les notations alternatives de $\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right)$ données par : $\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right)$, $\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^p, \left\{\mathbf{A}^{(q)}\right\}_{p+1}^N\right)$. Pour les garanties théoriques dont les détails sont reportés dans Annexe 3, on considère les hypothèses suivantes :

**Hypothèse 1** *Bornitude uniforme. Les sous-tenseurs considérés sont uniformément bornés :*

$$\|\mathcal{X}_{i_n}^n\|_F \leq \rho, \forall 1 \leq i_n \leq I_n$$

**Hypothèse 2** *Bornitude des facteurs. On considère les domaines* $\mathcal{G} \in \mathbb{D}_g, \mathbf{A}^{(m)} \in \mathbb{D}_m$ *avec :*

$$\mathbb{D}_g = \{\|\mathcal{G}_a\|_F \leq \alpha\}, \mathbb{D}_m = \left\{\|\mathbf{A}_a^{(m)}\|_F \leq \alpha\right\}$$

### 5.3.1 Résultat de convergence pour *Singleshot*

Pour l'analyse de convergence, on considère les définitions suivantes pour les pas de descente $\eta_k^{\mathcal{G}}$ et $\eta_k^p$ à l'itération $k+1$ :

$$\eta_k^{\mathcal{G}} = \underset{\eta \in [\frac{\delta_1}{\sqrt{K}}, \frac{\delta_2}{\sqrt{K}}]}{\arg\min} \left(\eta - \frac{\delta_1}{\sqrt{K}}\right) \phi_g(\eta) \tag{5.10}$$

$$\eta_k^p = \underset{\eta \in [\frac{\delta_1}{\sqrt{K}}, \frac{\delta_2}{\sqrt{K}}]}{\arg\min} \left(\eta - \frac{\delta_1}{\sqrt{K}}\right) \phi_p(\eta) \tag{5.11}$$

$$\phi_g(\eta) = \ell\left(\mathcal{G}_k - \eta \mathcal{D}_k^{\mathcal{G}}, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)$$

$$\phi_p(\eta) = \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)} - \eta \mathbf{D}_k^p, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^N\right) - \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right)$$

avec $\delta_2 > \delta_1 > 0$ étant des paramètres à fixer par l'utilisateur.
La motivation sous-jacente aux problèmes (5.10), (5.11) est d'assurer la décroissance de la fonction de coût après chaque étape de mise à jour. Les problèmes de minimisation associés aux pas de descente $\eta_k^{\mathcal{G}}$ et $\eta_k^p$ sont bien définis étant donné que les facteurs les définissant sont déjà déterminés à l'étape de mise à jour de $\mathcal{G}_{k+1}$ et $\mathbf{A}_{k+1}^{(p)}$ en plus du fait qu'ils correspondent à des problèmes de minimisation de fonctions continues sur un compact. Parallèlement aux hypothèses *Hypothèse 1* et *Hypothèse 2*, ainsi que les définitions données par les équations (5.10), (5.11), on suppose que les inégalités suivantes sont vérifiées :

$$\frac{\delta_1}{\sqrt{K}} < \eta_k^{\mathcal{G}} \leq \frac{\delta_2}{\sqrt{K}}, \qquad \frac{\delta_1}{\sqrt{K}} < \eta_k^p \leq \frac{\delta_2}{\sqrt{K}} \tag{5.12}$$

Cela revient à considérer que les solutions des problèmes de minimisation définis par les équations (5.10) et (5.11) ne sont pas atteintes à la borne inférieure de l'intervalle de minimisation. Avec ces hypothèses et définitions, on établit, comme dans le cadre de problèmes de minimisation non-convexes standard [36], une vitesse de convergence

ergodique. Précisément, on prouve que :

$$\exists K_0 \geq 1, \forall K \geq K_0, \frac{1}{K}\sum_{k=0}^{K-1} \|\nabla \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2 \leq \frac{(N+1)\Delta}{\sqrt{K}} \tag{5.13}$$

avec $\Delta = \frac{\delta_2}{\delta_1^2}\left(2\Gamma + \alpha^{2N}\Gamma_g^2\delta_2^2 + \sum_{p=1}^{N}(1 + 2\Gamma + \alpha^{2N}\Gamma_p^2\delta_2^2)\right)$, $\Gamma, \Gamma_m, \Gamma_g \geq 0$.
Ce résultat prouve que **Singleshot** converge vers un point stationnaire, c'est-à-dire vers un point où le gradient est égal à 0 à la vitesse $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$. La convergence ergodique peut aussi être interprétée de la manière suivante [83] : si l'on choisit une itération $k$ avec une probabilité $\frac{1}{K}$, la derivée évaluée à $\left(\mathcal{G}_k, \mathbf{A}_k^{(1)}, .., \mathbf{A}_k^{(N)}\right)$ est bornée par $\frac{N\Delta}{\sqrt{K}}$.

### 5.3.2 Résultat de convergence pour *Singleshotinexact*

Pour cette analyse de convergence, on considère que le pas $\eta_k^N$ (pas de descente associé à la mise à jour de $\mathbf{A}^{(N)}$ à l'itération $k+1$) est défini par le problème de minimisation suivant :

$$\eta_k^N = \underset{\eta \in [\frac{1}{4K^\gamma}, \frac{1}{K^\gamma}]}{\arg\min}\left(\eta - \frac{1}{4K^\gamma}\right)\phi(\eta) \tag{5.14}$$

avec :
$\phi(\eta) = \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \tilde{\phi}(\eta)\right)$
$\tilde{\phi}(\eta) = \mathbf{A}_k^{(N)} - \frac{\eta}{B_k}\sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j$, $\partial_{\mathbf{A}^{(N)}}\ell_j$ étant la dérivée la fonction $\ell_j$ évaluée à $\mathbf{A}_k^{(N)}$,
$\ell_j(\mathbf{A}^{(N)}) = \frac{1}{2}\|\mathcal{X}_j^n - \mathcal{G}_{k+1} \underset{m \in \mathbf{I}_{n-1}}{\times_m} \mathbf{A}_{k+1}^{(m)} \times_n (\mathbf{A}_{k+1}^{(n)})_{j,:} \underset{q \in \mathbf{I}_{N-1}^{n+1}}{\times_q} \mathbf{A}_{k+1}^{(q)} \times_N \mathbf{A}^{(N)}\|_F^2$.

Les paramètres $\lambda > 0, \gamma > 1$ représentent des paramètres à fixer par l'utilisateur. En plus des hypothèses *Hypothèse 1* et *Hypothèse 2*, on considère les trois hypothèses additionnelles suivantes :

**1. Pas de descente pour la mise à jour de $\mathbf{A}^{(N)}$ :**

$$\frac{1}{4K^\gamma} < \eta_k^N \leq \frac{1}{\alpha^{2N}}$$

Cette hypothèse revient à considérer que la solution n'est pas atteinte à la borne inférieure de l'intervalle de minimisation

**2. Non-nullité de la derivée partielle par rapport à $\mathbf{A}^{(N)}$**

$$\partial_{\mathbf{A}^{(N)}}\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) \neq 0$$

Cette condition assure l'existence d'un ensemble $\mathcal{SET}_k$ tel que $\sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j \neq 0$ et l'ensemble que l'on considère pour le problème (5.14) est l'un de ces ensembles.

**3. Choix du nombre de sous-tenseurs $B_k$ :**

$$I_n \times \sqrt{\frac{1}{2} + \frac{1}{I_n}} \leq B_k, I_n > 2$$

La condition $I_n > 2$ assure $I_n\sqrt{\frac{1}{2}+\frac{1}{I_n}} < I_n$
Sur la base de ces hypothèses et définitions, on peut démontrer que la suite $\Delta_k = \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \ell_{\min}$ vérifie l'inégalité :

$$\forall k > k_0 = 1 + \frac{1}{\log(1+\lambda)}\log\left(\frac{1}{\log(1+\lambda)}\right), \Delta_k \leq \frac{\Delta_1 + \zeta(\lambda, \rho, \alpha, I_n)}{k - k_0} \tag{5.15}$$

avec log étant la fonction logarithmique, $\ell_{\min}$ étant la valeur minimale de $\ell$ qui est bien définie étant donné que $\ell$ est une fonction continue définie sur l'ensemble compact $\mathbb{D}_g \times \mathbb{D}_1 \times .... \times \mathbb{D}_N$ et $\zeta$ une fonction des paramètres $\lambda, \rho, \alpha, I_n$.
Ce résultat assure que la suite $\left\{\mathcal{G}_k, \mathbf{A}_k^{(1)}, ., \mathbf{A}_k^{(N)}\right\}$ converge vers l'ensemble des minimiseurs de la fonction $\ell$ à la vitesse $\mathcal{O}\left(\frac{1}{k}\right)$.

**Remarque 6** *: les problèmes définis par* (5.14)*,* (5.10) *et* (5.11) *dont les solutions sont globales et peuvent être résolus numériquement par des heuristiques (par exemple la* méthode du nombre d'or*), ne sont pas en contradiction avec la philosophie de notre approche puisque ces problèmes peuvent être résolus en traitant un seul sous-tenseur à la fois dû à l'expression alternative de $\ell$ donnée par l'équation* (5.2)*.*

## 5.4 Simulations numériques

Notre objectif est d'illustrer que, pour des tenseurs de petite taille, nos approches **Singleshot** et **Singleshotinexact** (ainsi que leurs variantes non-négatives) donnent des résultats comparables à ceux obtenus par certaines méthodes de l'état de l'art et réussissent à effecteur la tâche d'inférence pour des tenseurs de grande taille alors que certains compétiteurs n'y arrivent pas. Les méthodes considérées pour la comparaison sont : **GreedyHOOI** [124] qui est une extension avec garantie de convergence de la méthode standard **HOOI**, une approche récente basée sur la minimisation alternée avec l'incorporation d'opérateurs de Sketching [84] nommée **Tensorsketch** (sa version en ligne sera nommée **Tensorsketchonline**) [13] et des méthodes aléatoires [135] (**Algorithme 2** dans [135] nommées **Scalrandomtucker** et **Algorithme 1** dans [135] avec des contraintes de non-négativité nommée **posScalrandomtucker**).

### 5.4.1 Données expérimentales avec Singleshot/Singleshotinexact

Les bases de données considérées sont : *Movielens* [48] à partir de laquelle on construit un tenseur d'ordre 3 dont les modes représentent respectivement les pas de temps, les films et les utilisateurs et la base de données de messages électroniques *Enron*, à partir

de laquelle on construit un tenseur d'ordre 3 dont les modes représentent respectivement les expéditeurs, les destinataires et les mots les plus fréquents. Pour *Movielens*, on a mis en place un système de recommandation pour lequel on détermine l'erreur de test **MAP** (mean average precision) [58] sur un ensemble de test qui se définit comme la projection des données de test sur les facteurs latents inférés de l'ensemble d'entraînement (qui est différent de l'ensemble de test) et cette erreur est calculée en faisant la moyenne sur cinq divisions aléatoires dans la proportion $50-50$ en entraînement et en test. Pour *Enron*, on calcule l'erreur d'approximation **AE** définie par l'équation (5.16) sur un ensemble de test (de même taille que l'ensemble d'entraînement) pour un problème de régression.

$$\mathbf{AE} = \|\mathcal{X}_{\text{test}} - \mathcal{G}_s \times_1 \mathbf{A}^{(1)}_{\text{out}} \times_2 \mathbf{A}^{2}_{\text{out}} \times_3 \mathbf{A}^{(3)}_{\text{out}}\|_F \tag{5.16}$$

avec $\mathcal{G}_s$ étant la projection de $\mathcal{X}$ sur les matrices $\left\{\mathbf{A}^{(n)}_{\text{out}}, 1 \leq n \leq 3\right\}$ représentant les facteurs latents inférés de la décomposition.

Notre objectif étant d'analyser la capacité des diverses méthodes à décomposer un tenseur dont la taille croît, nous avons arbitrairement fixé la taille des tenseurs construits à partir de Movielens et Enron à $M \times M \times 200$ et $M \times M \times 610$, $M$ étant un paramètre à fixer. Les expériences sont effectuées sur une machine MacOs avec 32 Go de memoire (RAM).

Un autre objectif est de prouver la robustesse de notre approche par rapport aux hypothèses et définitions mises en place dans la section 5.3, ce qui est important car les problèmes de minimisation relatifs aux définitions des pas de descente peuvent avoir une complexité en temps élevée : cela motive le fait que pour **Singleshot**, les pas sont fixés à $10^{-6}$ et pour **Singeshotinexact**, ils sont respectivement fixés à $10^{-6}$ et $10^{-8}$ pour *Enron* et *Movielens*. Pour **Singleshotinexact**, les éléments dans $\mathcal{SET}_k$ sont générés aléatoirement sans remplacement (tous les éléments sont deux à deux distincts) et le nombre de sous-tenseurs est défini conformément à la troisième hypothèse dans la section 5.3.2. Pour les figures, les versions sans contraintes de **Singleshot** et **Singleshotinexact** seront suivies de "unconstrained" et "positive" pour les variantes non-négatives.

La figure 5.1 présente les résultats obtenus pour *Enron* et *Moovielens* pour **Singleshot** et **Singleshotinexact**. Les performances en terme de **MAP** et **AE** sont équivalentes pour les différentes méthodes. En terme de temps de calcul, **Scalrandomtucker** est la meilleure méthode. Cependant, les méthodes compétitrices n'arrivent pas décomposer des tenseurs pour lesquels $M = 4000$ (pour *Enron*) et $M \geq 2800$ (pour *Movielens*) à cause d'une erreur de mémoire, contrairement à notre approche même si le temps de calcul est élevé. On remarque aussi, comme prévu, que **Singleshotinexact** nécessite moins de

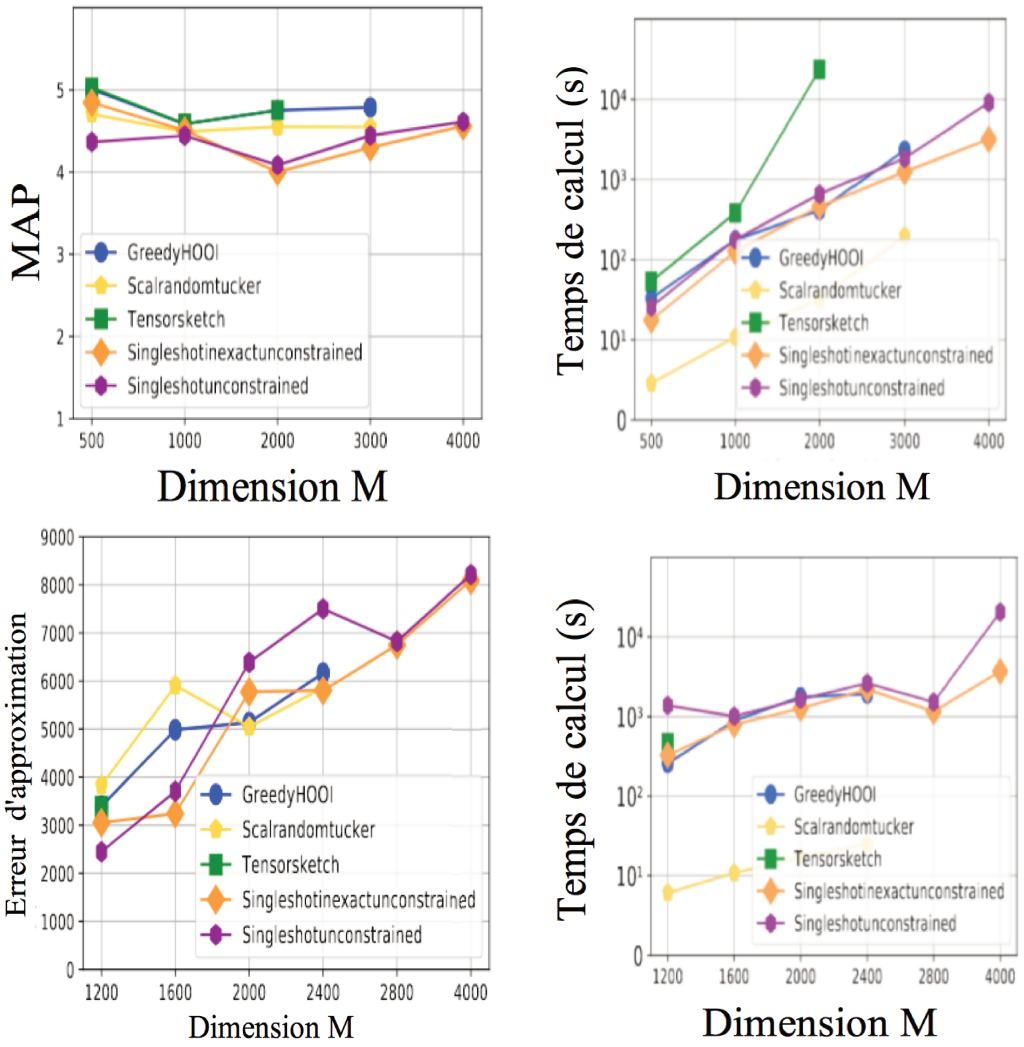


FIGURE 5.1 – Comparaison avec les versions sans contraintes de Singleshot/Singleshotinexact. Haut (de gauche à droite) : MAP et temps de calcul pour *Movielens*. Bas (de gauche à droite) : erreur d'approximation et temps de calcul pour *Enron*. Quand $\mathcal{M}$ croît, les marqueurs manquant signifient qu'on a une erreur mémoire. Le rang du tenseur noyau $\mathcal{G}$ est fixé à $(5, 5, 5)$

temps de calcul que **Singleshot** dû au fait que la première méthode utilise moins de termes dans le calcul des dérivées que la séconde.

La figure 5.2 donne l'erreur d'approximation et le temps de calcul pour **Singleshot** et **singleshotinexact** avec des contraintes de non-négativité et une approche aléatoire également avec des contraintes de non-négativité nommée **PosScalrandomtucker**. La qualité de décomposition est meilleure pour **Singleshotpositive** pour les deux bases de données *Movielens* et *Enron*. En plus, quand la taille du tenseur est petite, **PosScalrandomtucker** est très efficace en terme de temps de calcul par rapport à notre approche . Cependant, **PosScalrandomtucker** a du mal à décomposer un tenseur de grande taille dû à une erreur mémoire contrairement à nos approches **Singleshot** et **Singleshotinexact**.

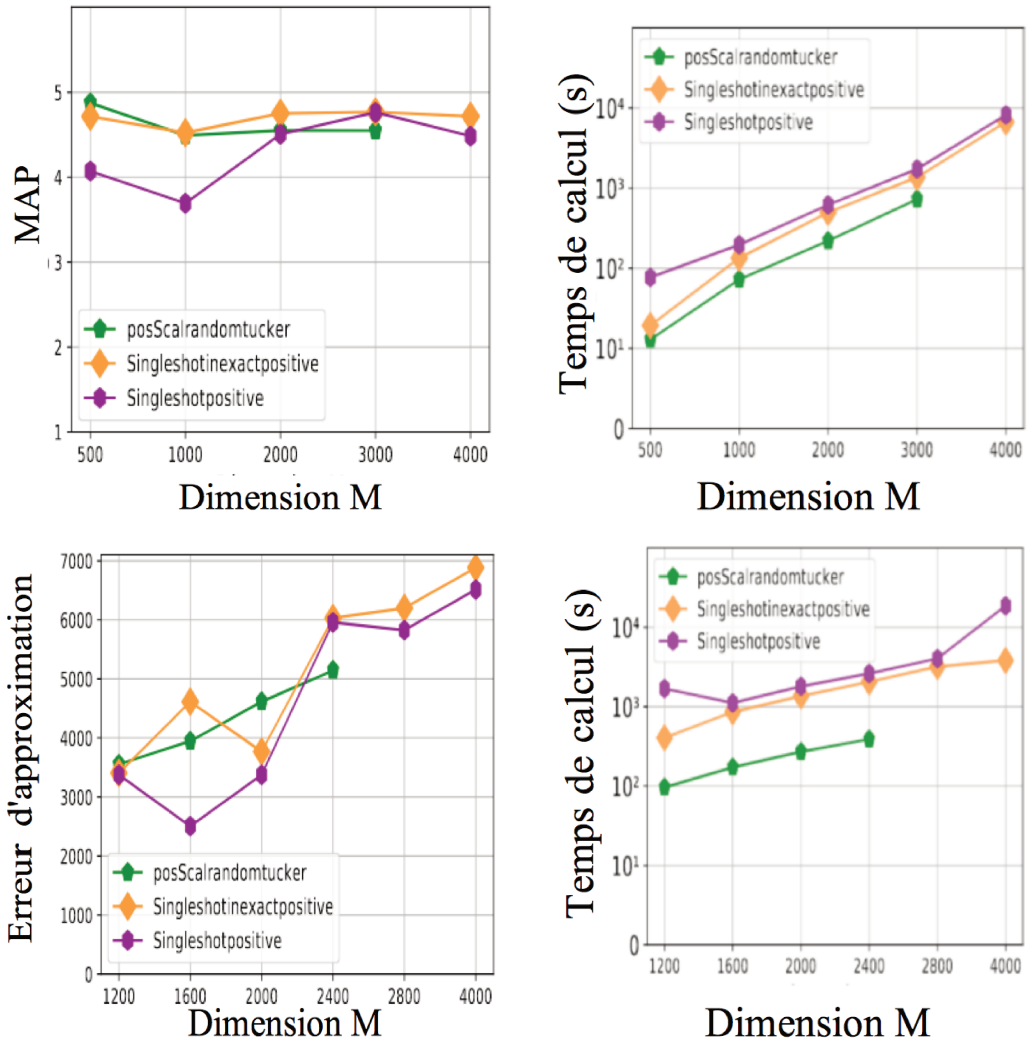


FIGURE 5.2 – Haut (de gauche à droite) : MAP et temps de calcul pour *Movielens*. Bas (de gauche à droite) : erreur d'approximation et temps de calcul pour *Enron*. Comme M croît, les marqueurs manquants font référence à une erreur de mémoire. Le rang du tenseur noyau $\mathcal{G}$ est fixé à $(5, 5, 5)$

### 5.4.2 Singleshot/Singleshotinexact vs Tensorsketchonline

Dans cette section, on compare nos approches **Singleshot** et **Singleshotinexact** à **Tensorsketchonline**, qui est une méthode récente ayant la même philosophie que notre approche dans le sens où elle traite de manière séquentielle des entrées tirées du tenseur à décomposer. Pour **Tensorsketchonline**, les données sont artificiellement divisées par rapport aux deux premiers modes.

Les résultats de la comparaison sont donnés par les figures 5.3 et 5.4 pour *Enron* et *Moovielens*. Toutes les méthodes donnent des erreurs de test similaires avec un léger avantage pour **Singleshot** et **Singleshotinexact** avec des contraintes de non-négativité sur les facteurs latents pour *Enron* (voir la figure 5.3) et les variantes sans contraintes **Singleshotinexactunconstrained** et **Singleshotunconstrained** pour *Moovielens* (voir la

figure 5.4). Cependant, notre approche nécessite moins de temps de calcul par rapport à **Tensorsketchonline** dû au fait qu'une minimisation alternée est plus coûteuse en temps que notre approche qui repose sur une descente de coordonnées alternée.

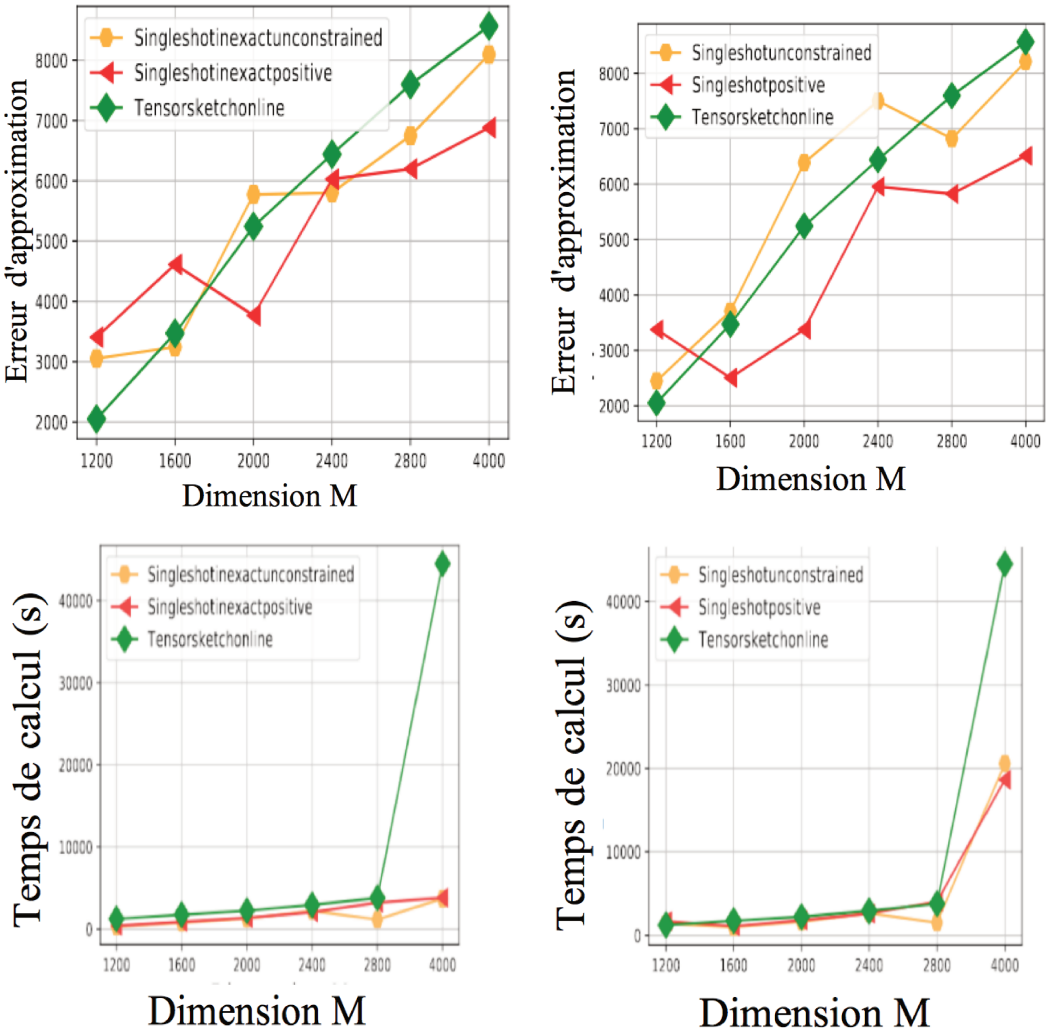


FIGURE 5.3 – Haut (de gauche à droite) : erreurs d'approximation pour **Singleshotinexact** et **Singleshot**. Bas (de gauche à droite) : temps de calcul pour **Singleshotinexact** et **Singleshot**. Le rang du tenseur noyau est fixé à $(5, 5, 5)$

### 5.4.3 Singleshotonline vs Tensorsketchonline

Pour illustrer l'efficacité de la variante en ligne de notre approche **Singleshotonline** (associée à des contraintes de non-négativité), on considère comme compétiteur la méthode **Tensorsketchonline** [13]. La première base de donnée utilisée est un tenseur d'ordre 3 de dimensions $20000 \times 2000 \times 50$ construit à partir de *Enron* qui est artificiellement divisé en sous-tenseurs par rapport aux premier et second modes. La seconde base de données considérée est un tenseur $\mathcal{X} \in \mathbb{R}^{15000 \times 2000 \times 60}$ construit à partir de *Moovielens* dont les modes sont respectivement le temps, les films, les utilisateurs : ce tenseur est divisé par

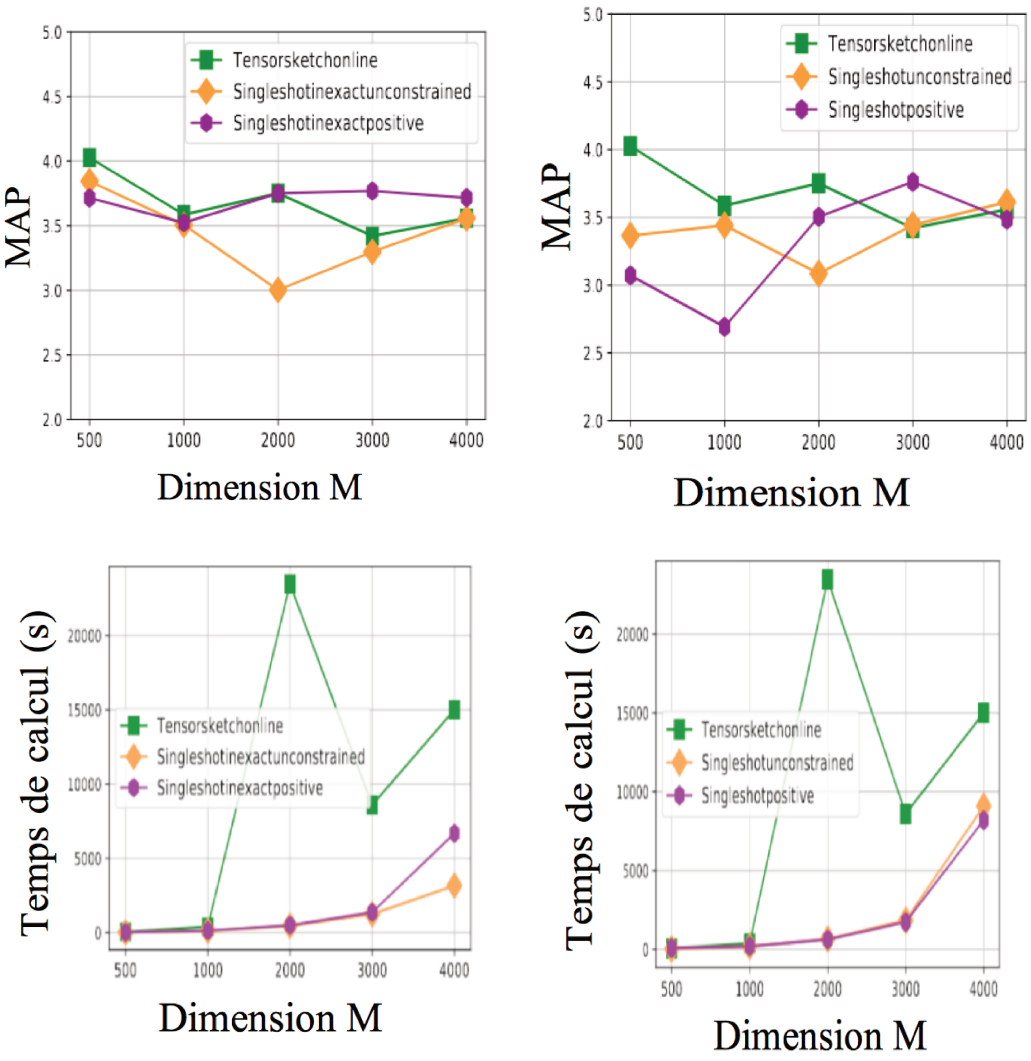


FIGURE 5.4 – Haut (de gauche à droite) : erreur de test dans le cas sans contraintes et avec des contraintes de non-négativité. Bas (de gauche à droite) : temps de calcul sans contraintes et avec des contraintes de non-négativité. Le rang du tenseur noyau est fixé à $(5, 5, 5)$

rapport au premier mode en deux ensembles, un pour l'entraînement et l'autre pour le test respectivement dans la proportion $70\% - 30\%$.

Pour *Enron*, la figure 5.5 montre l'erreur d'approximation et le temps de calcul. De même, pour *Movielens*, la figure 5.6 correspond à l'erreur de test et le temps de calcul. La valeur du rang est fixée à $(\mathsf{R}, \mathsf{R}, \mathsf{R})$. Notre méthode produit des erreurs moins importantes car elle prend en compte les contraintes sur les entrées des données (non-négativité) et est plus rapide car la mise à jour de chaque variable repose sur une itération de descente de gradient, ce qui est moins coûteux en temps de calcul qu'une minimisation complète pour chaque étape de mise à jour.

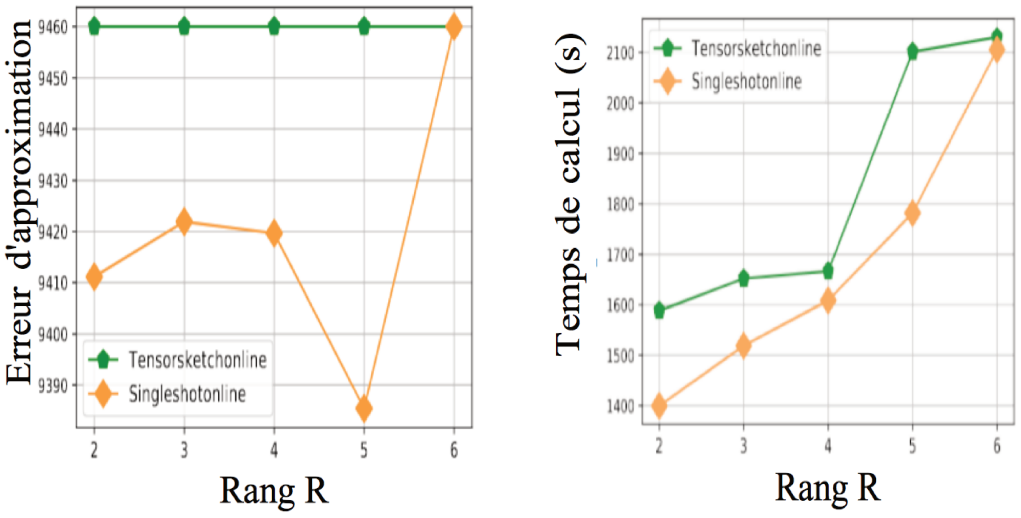


FIGURE 5.5 – Erreur d'approximation et temps de calcul pour **Tensorsketchonline** et **Singleshotonline**

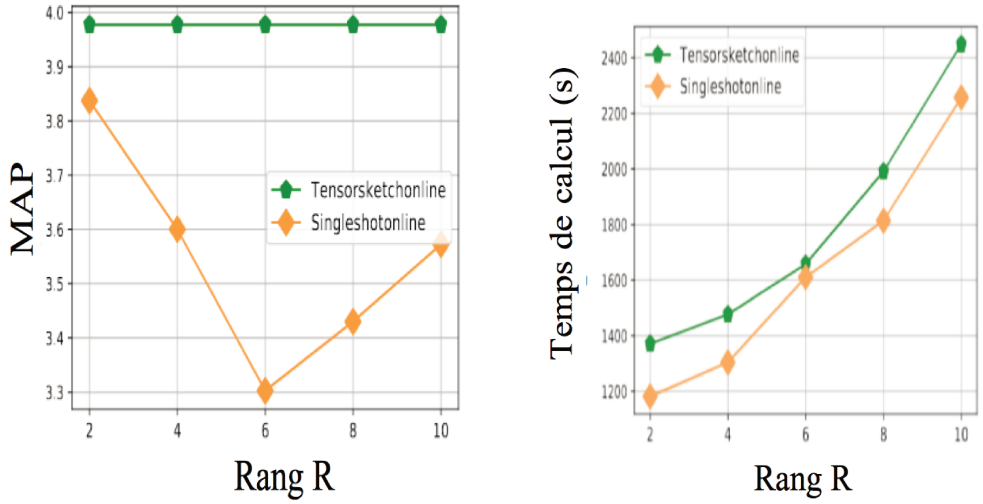


FIGURE 5.6 – Erreur de test et temps de calcul pour **Tensorsketchonline** et **Singleshotonline** pour Moovielens

### 5.4.4 Vérification des hypothèses

Dans cette section, on vérifie certaines hypothèses mises en place dans le cadre notre analyse de convergence.

#### 5.4.4.1 Évolution du gradient par rapport à la variable $\mathbf{A}^{(N)}$ (deuxième hypothèse dans la section 5.2.3 pour *Singleshotinexact*)

La figure 5.7, représentant les normes de $\partial_{\mathbf{A}^{(N)}} \ell \left( \mathcal{G}_{k+1}, \left\{ \mathbf{A}_{k+1}^{(m)} \right\}_1^{N-1}, \mathbf{A}_k^{(N)} \right)$ et de $\sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}} \ell_j$ montre que la deuxième hypothèse (présentée dans la section 5.3.2) pour **Singleshotinexact** est valide en pratique (le pas de descente $\eta_k^N$ est donc bien défini).

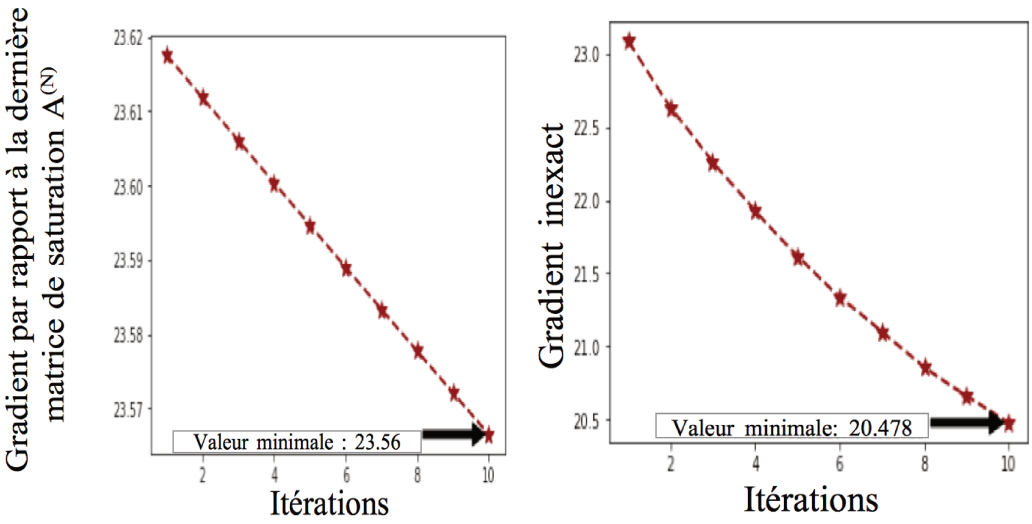


FIGURE 5.7 – Enron data set $\mathcal{X} \in \mathbb{R}^{1200\times1200\times200}$ : gauche (norme du gradient pour **Singleshot** sur une échelle logarithmique), droite (norme du gradient inexact pour **Singleshotinexact** sur une échelle logarithmique

#### 5.4.4.2 Choix de $\mathcal{SET}_k$ pour le calcul du gradient inexact

Une manière naturelle d'assurer la non-nullité du gradient inexact serait d'effectuer une recherche exhaustive pour déterminer l'ensemble $\mathcal{SET}_k$ de cardinal $B_k$ assurant cette condition. Cela n'est pas pratique même pour les petites valeurs de $I_n$ car dans le pire cas, cette stratégie nécessiterait $\frac{I_n!}{B_k!(I_n-B_k)!}$ calculs de gradient inexact. Pour avoir un ensemble convenable (assurant la non-nullité) $\mathcal{SET}_k$, on a prouvé numériquement (voir la figure 5.7) qu'il est suffisant d'effectuer une sélection aléatoire de sous-tenseurs.

#### 5.4.4.3 Influence du nombre de sous-tenseurs

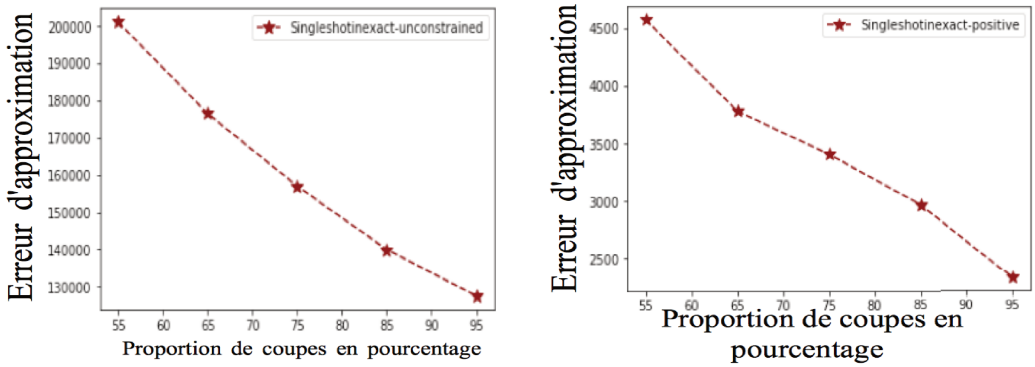


FIGURE 5.8 – Erreur d'approximation en fonction du nombre de sous-tenseurs pour **Singleshotinexact** avec le rang du tenseur noyau fixé à $(10, 10, 10)$ sur *Enron* : gauche (sans contraintes), droite (contrainte de non-négativité)

La figure 5.8 montre l'erreur d'approximation en fonction du nombre de sous-tenseurs utilisés pour **Singleshotinexact**. On remarque une dégradation de l'erreur d'approxima-

tion quand le nombre de sous-tenseurs décroît. Cela est cohérent avec l'intuition étant donné qu'un petit nombre de sous-tenseurs induit une erreur importante sur la direction de descente, conduisant ainsi loin de bons minima.

## 5.5 Conclusion

Nous avons introduit deux nouvelles approches pour la décomposition de **Tucker** d'un tenseur de grande taille nommées **Singleshot** et **Singleshotinexact**. Le principe sous-jacent à **Singleshot** repose sur la descente de gradient et la descente de coordonnées par blocs (mise à jour cyclique des variables), le calcul des dérivées étant effectué via le traitement séquentiel de sous-tenseurs (toutes les entrées du tenseur d'intérêt sont traitées pour cette approche, mais pas de manière simultanée). Le principe de **Singleshotinexact** est identique à celui de **Singleshot** à l'exception près qu'on utilise seulement certaines composantes de la donnée à décomposer. Une extension de **Singleshotinexact** a aussi été introduite pour la décomposition d'un tenseur qui croît par rapport à tous ses modes avec la contrainte de la passe unique (chaque observation est traitée une seule et unique fois).

D'un point de vue analytique, on a établi pour **Singleshot** (respectivement **Singleshotinexact**) la convergence vers un point stationnaire (respectivement minimiseur) avec une vitesse de convergence en $\mathcal{O}\left(\frac{1}{\sqrt{\mathsf{K}}}\right)$, $\mathsf{K}$ étant le nombre maximum d'itérations (respectivement en $\mathcal{O}\left(\frac{1}{\mathsf{k}}\right)$, $\mathsf{k}$ étant le numéro de l'itération).

Finalement, nous avons effectué des expériences numériques pour prouver l'efficacité de nos approches en terme de gestion de la complexité en espace ainsi que l'intérêt de mettre en place des variantes pouvant incorporer certaines contraintes populaires telles que la non-négativité.

# CHAPITRE 6

# CONCLUSION ET PERSPECTIVES

## 6.1 Conclusion

Dans ce travail, nous avons mis en place de nouvelles techniques pour l'apprentissage de dictionnaires et la décomposition d'un tenseur de grande taille via la décomposition de **Tucker**, le choix porté sur cette décomposition étant motivé par le fait qu'elle présente diverses applications et peut être facilement adaptée pour effectuer d'autres décompositions populaires (par exemple **CP**).

Dans le **chapitre 3**, on a développé une nouvelle méthode pour l'apprentissage de dictionnaires par décomposition non-négative de **Tucker**. Un rappel sur la décomposition non-négative de matrices est effectuée. De cette décomposition, une extension de la **NMF** pour la décomposition dans le cadre tensoriel, nommée **TDL** est présentée. La méthode de résolution numérique pour **TDL** est basée sur la minimisation alternée avec chaque problème intermédiaire, de la forme $\min_{\mathbf{B}\geq 0} \|\mathbf{X}-\mathbf{AB}\|_{\mathrm{F}}^2$, étant obtenu via la forme matricielle de la décomposition de **Tucker**. La limite de cette méthode provient du fait que ces problèmes intermédiaires peuvent avoir une complexité en espace élevée dû au produit de produit de **Kronecker** résultant de la forme matricielle du produit tenseur-matrice.
Pour contourner cette limitation, nous avons introduit dans le **chapitre 4** une approche en ligne nommée **OTL** dont le principe est d'inférer une base génératrice de facteurs latents à partir d'une séquence d'observations de tailles égales, les observations étant traitées en petit nombre, ce qui permet de réduire la complexité en espace par rapport à la stratégie qui consiste à utiliser toutes les observations de manière simultanée. Cette approche est inspirée de certains travaux effectués pour la décomposition en ligne de matrices [86] et fournit une garantie de convergence sous des conditions assez peu contraignantes et peut

être étendue pour inclure des contraintes populaires telles que :

— La parcimonie : appliquée sur les coefficients d'activation via l'introduction d'une pénalité $\ell_1$, avec le problème de minimisation sous-jacent résolu par une technique de minimisation proximale.

— la non-négativité : pertinente pour des observations présentant des composantes non-négatives, elle est mise en place grâce à la descente de gradient projeté.

— l'orthogonalité : utile pour assurer l'unicité des sous-espaces générés par les matrices de dictionnaires, on utilise pour cette contrainte une descente de gradient sur une variété de Stiefel.

Cependant, cette méthode n'est pas adaptée pour traiter des observations en flux (acquises séquentiellement au fil du temps) ayant des tailles différentes.

Enfin, nous avons établi dans le **chapitre 5** une approche désignée par **Singleshot** pour laquelle on a établi une vitesse de convergence en $\mathcal{O}(\frac{1}{\sqrt{\mathsf{K}}})$ avec $\mathsf{K}$ étant le nombre maximum d'itérations. L'algorithme proposé repose sur la descente de coordonnées (c'est-à-dire mise à jour cyclique des facteurs latents, chacune étant effectuée via une seule itération de *descente de gradient*) ainsi que le traitement séquentiel de sous-tenseurs, rendu possible possible grâce à la réformulation de la fonction de coût en utilisant des propriétés standard d'algèbre linéaire. Pour réduire la complexité en temps de **Singleshot**, nous avons mis en place **Singleshotinexact**, une variante dont l'idée repose sur le fait d'utiliser, pour l'inférence des facteurs latents, uniquement un certain nombre de sous-tenseurs et garantit une vitesse de convergence en $\mathcal{O}\left(\frac{1}{\mathsf{k}}\right)$, $\mathsf{k}$ étant le numéro de l'itération. De **Singleshotinexact** est déduite **Singleshotonline**, une troisième approche pour la décomposition d'un tenseur qui croît par rapport à tous les modes, permettant ainsi de traiter une gamme de problèmes plus large que **OTL** (introduit dans le *chapitre 4*) car les observations peuvent avoir des dimensions différentes.

## 6.2 Perspectives

Les travaux effectués, bien qu'adressant certaines limites de techniques existantes, présentent de nombreuses pistes d'amélioration :

— Une extension naturelle de certains de nos travaux (notamment **OTL** et **Singleshot**) serait la mise en place de versions supervisées (incorporant l'information de classes dans un problème d'apprentissage supervisé).

— Nous envisageons le développement d'approches en ligne pour des décompositions mixtes à la frontière entre **CP** et **Tucker** (par exemple, décomposition block term), ce problème n'étant pas très bien étudié dans la littérature.

— Pour des recherches futures, on compte également investiguer les liens potentiels existants entre décomposition de tenseur et réseaux de neurones comme ont tenté de faire certains travaux relativement récents. Par exemple, Jonah et al. [19] ont proposé une architecture de réseaux de neurones qui peut s'interpréter comme une décomposition non-négative de tenseur.

— Notre approche **OTL** n'offre de résultats de convergence que vers un point stationnaire et la vitesse de convergence n'est pas établie. De plus, elle ne tient pas compte de l'antériorité des données dans le sens où elle traite toutes données de manière équivalente alors qu'intuitivement, les données les plus récentes devraient avoir un poids plus important comparé aux données déjà acquises. Des améliorations possibles seraient donc d'établir une variante supplémentaire avec de meilleures garanties de convergence et/ou vitesse de convergence ainsi que d'associer aux observations des poids dont les valeurs sont d'autant plus importantes qu'elles sont récentes.

— D'un point de vue applicatif, la mise en oeuvre de **Singleshot** et **Singleshotinexact** est coûteux en temps. Cela peut être contourné en observant que les calculs de gradients/gradients inexacts peut être parallélisé, ceci étant simplement dû au fait qu'ils correspondent à la somme de termes qui sont tous indépendants les uns des autres (c'est-à-dire que deux termes différents peuvent être calculés indépendamment l'un de l'autre). L'étape suivante est donc bien évidemment la mise en place d'un script prenant en compte cette observation. Aussi, il peut être intéressant d'étudier l'impact de termes de pénalisation sur **Singleshot** et ses variantes.

— Les résultats de convergence pour **Singleshot/Singleshotinexact** ne sont valides que dans le cas « sans contraintes ». Une étude quantitative pour le cas « positif »parait donc être une suite naturelle. De plus, **Singleshotonline** n'offre aucune garantie de convergence, ni de garantie de reconstruction. La continuité évidente serait donc d'investiguer d'autres approches pour couvrir ces limites.

# Annexes

## 6.3 Équivalence de problèmes pour TDL

Soient $\mathcal{X} \in \mathbb{R}^{K \times I_2 \times I_3}, \mathcal{G} \in \mathbb{R}^{K \times J_2 \times I_3}$ deux tenseurs d'ordre 3 et $\mathbf{A}^{(2)} \in \mathbb{R}^{I_2 \times J_2}$. On a :

$$\|\mathcal{X} - \mathcal{G} \times_2 \mathbf{A}^{(2)}\|_F^2 = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}^{(2)}\mathbf{G}_k\|_F^2 \tag{6.1}$$

Dans l'équation (6.1), les matrices $\mathbf{X}_k$ et $\mathbf{G}_k$ représentent les sous-tenseurs par rapport au premier mode des tenseurs $\mathcal{X}$ et $\mathcal{G}$. **Démonstration** :
Notons $\alpha = \|\mathcal{X} - \mathcal{G} \times_2 \mathbf{A}^2\|_F^2$ et $\beta = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}^{(2)}\mathbf{G}_k\|_F^2$
Par définition du produit tenseur-matrice, on a :
$\alpha = \sum_{1 \leq k \leq K, 1 \leq i_2 \leq I_2, 1 \leq i_3 \leq I_3} \left( \mathcal{X}_{k,i_2,i_3} - \sum_{1 \leq j_2 \leq J_2} \mathcal{G}_{k,j_2,i_3} \mathbf{A}^{(2)}_{i_2,j_2} \right)^2$
Par définition du produit de deux matrices, on a :
$\beta = \sum_{k=1}^{K} \sum_{i_2,i_3} \left( (\mathbf{X}_k)_{i_2,i_3} - \sum_{1 \leq j_2 \leq J_2} \mathbf{A}^{(2)}_{i_2,j_2} (\mathbf{G}_k)_{j_2,i_3} \right)^2$
Par définition de $\mathbf{X}_k$ et $\mathbf{G}_k$, on a :
$\Rightarrow \beta = \sum_{k,i_2,i_3} \left( \mathcal{X}_{k,i_2,i_3} - \sum_{j_2} \mathcal{G}_{k,j_2,i_3} \mathcal{G}_{k,j_2,i_3} \mathbf{A}^{(2)}_{i_2,j_2} \right)^2$
$\Rightarrow \alpha = \beta$

## 6.4 Sketch d'une matrice

Etant donné une matrice $\mathbf{A} \in \mathbb{R}^{n \times d}$, le sketch de $\mathbf{A}$, noté $\tilde{\mathbf{A}} \in \mathbb{R}^{l \times d}$ ($l \leq n$), est une matrice plus petite que $\mathbf{A}$ en taille telle que $\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_F$ est petite. De manière équivalente, l'objectif de cette opération est de déterminer $\tilde{\mathbf{A}}$ telle que $\|\mathbf{A}\mathbf{x}\|_F$ et $\|\tilde{\mathbf{A}}\mathbf{x}\|_2$ soient proches pour tout $\mathbf{x} \in \mathbb{R}^d$.
Plusieurs approches ont été proposées en pratique pour calculer en pratique le sketch d'une matrice [39], [84] avec diverses applications telles que l'approximation faible rang

[39], analyse en composantes principales [84], la détection d'anomalies [35], etc.

# ANNEXE 2. ANALYSE DE CONVERGENCE POUR OTL

Pour l'analyse de convergence, on s'interesse à une double convergence asympotique dans le sens où l'on considère que le nombre d'itérations et le nombre d'échantillons tendent vers $+\infty$

## 1 Rappel des définitions et notations supplémentaires

### 1.1 Rappel des notations et raisonnement pour la convergence

$$\min_{\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}} f(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) \tag{6.2}$$

$$\begin{aligned} f(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) &= \mathbb{E}_{\mathbb{P}}(\ell(\mathfrak{X},\mathbf{A}^{(1)},..,\mathbf{A}^{(N)})) \\ &= \lim_{t\to\infty} f_t(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) \\ &= \lim_{t\to\infty} \frac{1}{t}\sum_{i=1}^{t} \ell(\mathfrak{X}_i,\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}) \end{aligned}$$

Pour $k$ fixé, on note par $\widehat{f}_{n,t}$ la fonction $\mathbf{A} \to \widehat{f}_t(\mathbf{A}_{k+1}^{(1)},..,\mathbf{A}_{k+1}^{(n-1)},\mathbf{A},\mathbf{A}_k^{(n+1)},..\mathbf{A}_k^{(N)})$ et considérons la suite $\mathbf{A}_{k+1,t}^{(n)}$ définie par :

$$\mathbf{A}_{k+1,t}^{(n)} \leftarrow \arg\min_{\mathbf{A}^{(n)}} \widehat{f}_{n,t}(\mathbf{A}^{(n)}) \tag{6.3}$$

avec $\widehat{f}_{n,t}$ définie par l'équation (4.9).

Dans un premier temps, on prouve que $\mathbf{A}_{k+1,t}^{(n)}$ converge vers l'ensemble des points stationnaires de la fonction définie par $\mathbf{A} \to f(\mathbf{A}_{k+1}^{(1)},..,\mathbf{A}_{k+1}^{(n-1)},\mathbf{A},\mathbf{A}_k^{(n+1)},..\mathbf{A}_k^{(N)})$. De ce résultat, on prouve que $\left\{\mathbf{A}_k^{(1)},..,\mathbf{A}_k^{(N)}\right\}$ converge vers l'ensemble des points stationnaires de la fonction $\left(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}\right) \to f(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)})$. Pour le problème de minimisation présenté via

l'équation (6.3), on remplace $\widehat{f}_{n,t}$ par $\widehat{f}_{n,t} - \frac{\alpha(1-\theta)}{2}(\sum_{p=1}^{n-1} \|\mathbf{A}_{k+1}^{(p)}\|_F^2 + \sum_{q=n+1}^{N} \|\mathbf{A}_k^{(q)}\|_F^2)$ puisque les termes soustraits sont des constantes, donc n'interviennent pas dans le problème de minimisation défini par l'équation (6.3) qui considère une minimisation par rapport à $\mathbf{A}^{(n)}$. Par conséquent, dans la suite, on s'interessera aux notations suivantes :

$$\ell(\mathcal{X}, \mathbf{A}^{(n)}) = \min_{\mathcal{G}} \frac{1}{2}\|\mathcal{X} - \mathcal{G} \times_{p \in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q \in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$$

$$f_n(\mathbf{A}^{(n)}) = \mathbb{E}_{\mathbb{P}}(\ell(\mathcal{X}, \mathbf{A}^{(n)}))$$

$$f_{n,t}(\mathbf{A}^{(n)}) = \frac{1}{t}\sum_{i=1}^{t} \ell(\mathcal{X}_t, \mathbf{A}^{(n)})$$

$$\begin{aligned}\widehat{f}_{n,t}(\mathbf{A}^{(n)}) &= \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\|\mathcal{X}_i - \mathcal{G}_i \times_{p \in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q \in I_N^{n+1}} \mathbf{A}_k^{(q)}|_F^2 + \alpha\theta\|\mathcal{G}_i\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2 \\ &= \frac{1}{2}\mathrm{Trace}(\xi_t \mathbf{A}^{(n)T}\mathbf{A}^{(n)}) - \mathrm{Trace}(\mathbf{A}^{(n)}\eta_t) + \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}\|\mathcal{X}_i\|_F^2 + \Omega_1(\mathcal{G}_i)) + \Omega_2(\mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}\right)\end{aligned}$$

Les suites $\xi_t$ et $\eta_t$ sont définies par :

$$\xi_t = \frac{1}{t}\sum_{i=1}^{t} \Gamma_i\Gamma_i^T, \eta_t = \frac{1}{t}\sum_{i=1}^{t} \Gamma_i \mathbf{X}_i^{(n)T}, \Gamma_i = \mathbf{G}_i^{(n)} \otimes_{p \in I_{n-1}} \mathbf{A}_{k+1}^{(p)T} \otimes_{q \in I_N^{n+1}} \mathbf{A}_k^{(q)T} \tag{6.4}$$

avec :

$$\mathcal{G}_i = \arg\min_{\mathcal{G}} \|\mathcal{X}_i - \mathcal{G} \times_{p \in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}_{i-1}^{(n)} \times_{q \in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$$

Les dérivées $f_n$ et $\ell$ en fonction de $\mathbf{A}^{(n)}$ seront notées respectivement par $\partial f_n$ et $\partial_2 \ell$.

## 1.2 Notations supplémentaires

La norme sur $\mathbb{R}^{m_1 \times n_1} \times .. \times \mathbb{R}^{m_K \times n_K}$ ($\mathbb{R}^{m_k \times n_k}$ étant l'ensemble des matrices de taille $m_k \times n_k$), notée par $\|\cdot\|_{\prod_{k \in I_K} \mathbb{R}^{m_k \times n_k}}$ est définie par :

$$\|\left(\mathbf{A}^{(1)}, .., \mathbf{A}^{(K)}\right)\|_{\prod_{k \in I_K} \mathbb{R}^{m_k \times n_k}} = \sum_{k=1}^{K} \|\mathbf{A}^{(k)}\|_F$$

Pour deux matrices $\mathbf{A} \in \mathbb{R}^{M\times N}$ et $\mathbf{B} \in \mathbb{R}^{M\times N}$, le produit scalaire est défini par :

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{1\leq m\leq M, 1\leq n\leq N} \mathbf{A}_{m,n}\mathbf{B}_{m,n}$$

Le supremum d'une fonction continue $f$ définie sur un domaine compact $\mathbb{D}$ sera noté par $\|f\|_\infty$ : $\|f\|_\infty = \sup_{\mathbf{x}\in\mathbb{D}} | f(\mathbf{x}) |$. La fonction signe est notée par $\mathtt{sign}$.

# 2 Algorithme pour un nombre infini d'échantillons et d'itération

Pour le cadre théorique, on suppose que le nombre d'itérations ainsi que le nombre d'échantillons sont infinis. Donc, notre algorithme se réécrit sous la forme :

***Algorithme 8 : OTL-infini***

**Entrées**

$\{\mathfrak{X}_1, .., \mathfrak{X}_t ....\}$ : séquence d'observations,

$\alpha > 0, 0 \leq \theta < 1$ : valeurs de paramètres

**Initialisation**

$\mathbf{A}_0^{(n)}$ : valeurs initiales des matrices de dictionnaires,

**Répéter**

  **Pour** $1 \leq k \leq K$ **faire**

    **Pour** $1 \leq n \leq N$ **faire**

$$\mathbf{A}_{k+1}^{(n)} = \begin{cases} \lim_{t\to\infty} \mathbf{A}_{k+1,t}^{(n)} & \text{si } \mathbf{A}_{k+1,t}^{(n)} \text{ converge} \\ \arg\min_{\mathbf{A}^{(n)}} \widehat{f}_\infty(\mathbf{A}^{(n)}) & \text{sinon} \end{cases}$$

    **Fin Pour**

  **Fin Pour**

**jusqu'à ce que** (un critère d'arrêt soit vérifié)

Dans la suite, les fonctions de pénalité $\Omega_1$ et $\Omega_2$ sont choisies comme étant les normes

$\ell_1$ et $\ell_2$. La fonction $\widehat{f}_\infty$ est définie par :

$$\widehat{f}_\infty(\mathbf{A}^{(n)}) = \frac{1}{2}\mathrm{Trace}\left(\left(\tilde{\xi}_t + \frac{\alpha(1-\theta)}{2}\mathbf{I}\right)\mathbf{A}^{(n)\mathsf{T}}\mathbf{A}^{(n)}\right) - \mathrm{Trace}\left(\mathbf{A}^{(n)}\tilde{\eta_t}\right)$$

avec $\tilde{\xi}_n$ et $\tilde{\eta_t}$ représentant les points d'accumulation des séquences $\tilde{\xi}_t$ et $\tilde{\eta_t}$ définies par l'équation (6.4) dont les existence sont assurées par la bornitude des suites $\xi_t$ et $\eta_t$.
**NB** : les différences entre les algorithmes 6 et 8 résultent du gap entre théorie et pratique : premièrement, tous les facteurs ne sont pas disponibles et deuxièmement, le code parcimonieux associé à l'observation $\mathcal{X}_t$ est déterminée via la projection sur $\left\{\mathbf{A}_{t-1}^{(1)}, .., \mathbf{A}_{t-1}^{(N)}\right\}$ au lieu de $\left\{\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{t-1}^{(n)}, \mathbf{A}_{k}^{(n+1)}, .., \mathbf{A}_{k}^{(N)}\right\}$ dans le but d'alléger le coût en terme de temps de calcul.

# 3 Hypothèses

— *Hypothèse I* : Les observations sont tirées d'une distribution de probabilité avec un support compact : cette hypothèse est naturelle car on travaille avec des valeurs finies enregistrées par des capteurs ;

— *Hypothèse II* : pour tous les problèmes de minimisation, on suppose que le tenseur d'activation ainsi que les matrices de dictionnaires appartiennent à des ensembles compacts : $\mathcal{G} \in \mathbb{K}_{\mathcal{G}}, \mathbf{A}^{(n)} \in \mathbb{K}_{I_n,J_n}$. Cela correspond à une interpreation naturelle de la pénalité $\ell_1 - \ell_2$.

— *Hypothèse III* : $\mathbf{C}_{:,j}\left(\mathrm{vec}(\mathcal{X}) - \mathbf{C}\,\mathrm{vec}(\mathcal{G})\right) = \theta\alpha\,\mathrm{vec}(\mathcal{G})_j)$ si $\mathtt{sign}(\mathrm{vec}(\mathcal{G})_j) \neq 0$
$|\mathbf{C}_{:,j}\left(\mathrm{vec}(\mathcal{X}) - \mathbf{C}\,\mathrm{vec}(\mathcal{G})\right)| \leq \alpha\theta$ sinon, avec : $\mathbf{C} = \left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\right)\otimes\mathbf{A}\otimes\left(\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\right)$
Cette condition assure l'unicité de la solution pour le problème de **Codage Parcimonieux**.

— *Hypothèse IV* : pour le problème de **Descente de Coordonnées**, on cherche $\mathbf{A}_{k+1}^{(n)}$ autour d'une boule centrée autour de $\mathbf{A}_k^{(n)}$ et de rayon $\frac{1}{k^{\frac{1}{2}}}$, c'est-à-dire telle que $\|\mathbf{A}_{k+1}^{(n)} - \mathbf{A}_k^{(n)}\|_F^2 \leq \frac{1}{k}$.

— *Hypothèse V* : $\forall k \in \mathbb{N}, \mathbf{A}_k^{(n)}$ est un point intérieur de l'ensemble $\mathbb{K}_{I_n,J_n}$.

# 4 Propriétés préliminaires

**Propriété 1** *Pour deux matrices* $\mathbf{A}, \mathbf{B}$ *telles que* $\|\mathbf{A}\|_F \leq \delta_1, \|\mathbf{B}\|_F \leq \delta_2$. *Il existe* $M > 0$ *tel que :*

$$\|\|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2\| \leq M_1\|\mathbf{A} - \mathbf{B}\|_F$$

Preuve :

$\|\mathbf{A}\|_F^2 = \|\mathbf{A}-\mathbf{B}+\mathbf{B}\|_F^2 = \|\mathbf{A}-\mathbf{B}\|_F^2 + 2 < \mathbf{A}-\mathbf{B}, \mathbf{B} > + \|\mathbf{B}\|_F^2$
$\Rightarrow \|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2 \leq \|\mathbf{A}-\mathbf{B}\|_F^2 + 2\|\mathbf{A}-\mathbf{B}\|_F\|\mathbf{B}\|_F$
$\Rightarrow \|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2 \leq \|\mathbf{A}-\mathbf{B}\|_F\|\mathbf{A}-\mathbf{B}\|_F + 2\delta_2\|\mathbf{A}-\mathbf{B}\|_F$
$\Rightarrow \|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2 \leq (\delta_1+\delta_2)\|\mathbf{A}-\mathbf{B}\|_F + 2\delta_2\|\mathbf{A}-\mathbf{B}\|_F$
Donc, on a :

$$\|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2 \leq (\delta_1+3\delta_2)\|\mathbf{A}-\mathbf{B}\|_F \leq \max(\delta_1+3\delta_2, \delta_2+3\delta_1)\|\mathbf{A}-\mathbf{B}\|_F \tag{6.5}$$

En permutant le rôle de $\mathbf{A}$ et $\mathbf{B}$, on a :

$$\|\mathbf{B}\|_F^2 - \|\mathbf{A}\|_F^2 \leq (\delta_2+3\delta_1)\|\mathbf{A}-\mathbf{B}\|_F \leq \max(\delta_1+3\delta_2, \delta_2+3\delta_1)\|\mathbf{A}-\mathbf{B}\|_F \tag{6.6}$$

En combinant (6.5) et (6.6), on a :
$\|\|\mathbf{A}\|_F^2 - \|\mathbf{B}\|_F^2\| \leq \max(\delta_1+3\delta_2, \delta_2+3\delta_1)\|\mathbf{A}-\mathbf{B}\|_F$

**Propriété 2** *Pour quatre matrices* $\mathbf{A}_1, \mathbf{B}_1 \in \mathbb{R}^{N\times M}$ *et* $\mathbf{A}_2, \mathbf{B}_2 \in \mathbb{R}^{M\times L}$ *telles que* $\|\mathbf{A}_1\|_F \leq \delta_1, \|\mathbf{B}_2\|_F \leq \delta_2$*, on a l'inégalite suivante :*
$\|\mathbf{A}_1\mathbf{A}_2 - \mathbf{B}_1\mathbf{B}_2\|_F \leq \delta_1\|\mathbf{A}_2-\mathbf{B}_2\|_F + \delta_2\|\mathbf{A}_1-\mathbf{B}_1\|_F$

Preuve :
$\|\mathbf{A}_1\mathbf{A}_2 - \mathbf{B}_1\mathbf{B}_2\|_F = \|\mathbf{A}_1(\mathbf{A}_2-\mathbf{B}_2+\mathbf{B}_2) - \mathbf{B}_1\mathbf{B}_2\|_F = \|\mathbf{A}_1(\mathbf{A}_2-\mathbf{B}_2) + \mathbf{A}_1\mathbf{B}_2 - \mathbf{B}_1\mathbf{B}_2\|_F$
$\Rightarrow \|\mathbf{A}_1\mathbf{A}_2 - \mathbf{B}_1\mathbf{B}_2\|_F \leq \delta_1\|\mathbf{A}_2-\mathbf{B}_2\|_F + \delta_2\|\mathbf{A}_1-\mathbf{B}_1\|_F$

**Propriété 3** *Si* $\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} .. \times_N \mathbf{A}^{(N)} = \mathcal{G} \times_{n\in I_N} \mathbf{A}^{(n)}$*, la forme matricielle de* $\mathcal{X}$ *par rapport au mode* $n$*, notée par* $\mathbf{X}^{(n)}$*, est définie par :*

$$\mathbf{X}^{(n)} = \mathbf{A}^{(n)}\mathbf{G}^{(n)}\left(\mathbf{A}^{(1)} \otimes .. \otimes \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} ... \otimes \mathbf{A}^{(N)}\right)^T = \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(n)T}$$

Ceci correspond à la propriété de matricisation d'une décomposition de **Tucker**.

**Propriété 4** *Notons* $g(\mathbf{A}^{(n)}) = \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} .. \times_n \mathbf{A}^{(n)} .. \times_N \mathbf{A}^{(N)}\|_F^2$*. La derivée de* $g$ *est donnée par :*

$$\begin{aligned}\partial g(\mathbf{A}^{(n)}) &= -2\left(\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\right)\left(\otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)}\right)\mathbf{G}^{(n)T} \\ &= -2\left(\widehat{\mathbf{X}}^{(n)}\mathbf{G}^{(n)T} - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\mathbf{B}^{(n)T}\right)\end{aligned}$$

*avec* $\mathbf{X}^{(n)}$ *et* $\mathbf{B}^{(n)}$ *étant les formes matricielles par rapport au monde* $n$ *des tenseurs* $\widehat{\mathcal{X}}$ *et* $\mathcal{B}$ *définis par :*
$\widehat{\mathcal{X}} = \mathcal{X} \times_{p\in I_{n-1}} \mathbf{A}^{(p)T} \times_n \mathbf{I} \times_{q\in I_N^{n+1}} \mathbf{A}^{(q)T}, \mathbf{I} \in \mathbb{R}^{I_n\times I_n}$ *: matrice identité*
$\mathcal{B} = \mathcal{G} \times_{p\in I_{n-1}} \mathbf{A}^{(p)} \times_n \mathbf{I} \times_{q\in I_N^{n+1}} \mathbf{A}^{(q)}, \mathbf{I} \in \mathbb{R}^{J_n\times J_n}$ *: matrice identité*

Preuve :

$g(\mathbf{A}^{(n)}+h) = \|\mathbf{X}^{(n)} - (\mathbf{A}^{(n)}+h)\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\|_F^2$

$\Rightarrow g(\mathbf{A}^{(n)}+h) = \|\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{\neq n}} \mathbf{A}^{(m)T} + h\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\|_F^2$

$\Rightarrow g(\mathbf{A}^{(n)}+h) = \|\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\|_F^2 + \|h\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\|_F^2 - 2\langle \mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}, h\mathbf{G}^{(n)} \otimes_{m\neq n} \mathbf{A}^{(m)T}\rangle$

$\Rightarrow g(\mathbf{A}^{(n)}+h) = g(\mathbf{A}^{(n)}) - 2\langle \left(\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\right) \otimes_{m\in I_{m\neq n}} \mathbf{A}^{(m)}\mathbf{G}^{(n)T}, h\rangle + O\left(\|h\|_F^2\right)$.

$\Rightarrow \partial g(\mathbf{A}^{(n)}) = -2\left(\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}\right) \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)}\mathbf{G}^{(n)T}$

$\Rightarrow \partial g(\mathbf{A}^{(n)}) = -2\mathbf{X}^{(n)}\left(\otimes_{m\in I_{N\neq n}}\mathbf{A}^{(m)}\right)\mathbf{G}^{(n)T} + 2\mathbf{A}^{(n)}\left(\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(n)T}\right)\left(\otimes_{m\in I_{N\neq n}}\mathbf{A}^{(n)}\mathbf{G}^{(n)T}\right)$

$\Rightarrow \partial g(\mathbf{A}^{(n)}) = -2\mathbf{X}^{(n)}\left(\otimes_{m\in I_{N\neq n}}\mathbf{A}^{(m)}\right)\mathbf{G}^{(n)T} + 2\mathbf{A}^{(n)}\left(\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(n)T}\right)\left(\mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(n)T}\right)^T$

( propriété du produit de **Kronecker**).

Par la *Propriété* 3, les formes matricielles par rapport au mode $n$ des tenseurs $\widehat{\mathcal{X}}$ et $\mathcal{B}$ sont données par :

$\widehat{\mathbf{X}}^{(n)} = \mathbf{X}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(n)}$ et $\mathbf{B}^{(n)} = \mathbf{G}^{(n)} \otimes_{m\in I_{N\neq n}} \mathbf{A}^{(m)T}$

Avec cette observation, il est trivial de voir que :

$\partial g(\mathbf{A}^{(n)}) = -2\widehat{\mathbf{X}}^{(n)}\mathbf{G}^{(n)T} + 2\mathbf{A}^{(n)}\mathbf{B}^{(n)}\mathbf{B}^{(n)T}$

**Propriété 5** *Si les matrices de dictionnaires et le tenseur noyau sont bornés,* $\widehat{f}_{n,t}$ *est bornée par une constante indépendante de* $t$.

Preuve :

Ceci est trivial avec l'inégalité triangulaire et *Hypothèse II*

**Propriété 6** *La fonction* $\widehat{f}_{n,t}$ *est strictement convexe et la matrice hessienne est bornée inférieurement (valeurs propres bornées inférieurement)*

Preuve :

$\widehat{f}_{n,t}(\mathbf{A}) = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\|\mathcal{X}_i - \mathcal{G}_i \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}_i\|_1 + \frac{(1-\theta)\alpha}{2}\|\mathbf{A}\|_F^2$, $0 \leq \theta < 1, \alpha > 0$.

**A. Convexité stricte**

$\widehat{f}_{n,t}(\mathbf{A}) - \frac{(1-\theta)\alpha}{2}\|\mathbf{A}\|_F^2$ est une fonction convexe. Donc, $\widehat{f}_{n,t}(\mathbf{A})$ est fortement convexe, impliquant ainsi la stricte convexité.

**B. Bornitude inférieure de la matrice hessienne**

$\widehat{f}_{n,t}(\mathbf{A}^{(n)}) = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\|\mathcal{X}_i - \mathcal{G}_i \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}_i\|_1 + \frac{(1-\theta)\alpha}{2}\|\mathbf{A}^{(n)}\|_F^2$

Par la **propriété** 4, on a :

$\partial\widehat{f}_{n,t}(\mathbf{A}^{(n)}) = \frac{1}{t}\sum_{i=1}^{t} \frac{1}{2}\left(-2\widehat{\mathbf{X}}_i^{(n)}\mathbf{G}_i^{(n)T} + 2\mathbf{A}^{(n)}\mathbf{B}_i^{(n)}\mathbf{B}_i^{(n)T}\right) + \alpha(1-\theta)\mathbf{A}^{(n)}$ avec :

$\mathbf{B}_i^{(n)} = \mathbf{G}_i^{(n)}\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T} \otimes_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)T}\right)$, $\widehat{\mathbf{X}}_i^{(n)} = \mathbf{X}_i^{(n)}\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)} \otimes_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}\right)$

Donc, on a :

$\partial\widehat{f}_{n,t}(\mathbf{A}^{(n)}) = \mathbf{A}^{(n)}\mathbf{S}_t - \mathbf{M}_t + \alpha(1-\theta)\mathbf{A}^{(n)}$ avec $\mathbf{S}_t = \frac{1}{t}\sum_{i=1}^{t}\mathbf{B}_i^{(n)}\mathbf{B}_i^{(n)T}$ , $\mathbf{M}_t = -\frac{1}{t}\sum_{i=1}^{t}\widehat{\mathbf{X}}_i^{(n)}\mathbf{G}_i^{(n)T}$
Par conséquent, la dérivée seconde par rapport à $\mathbf{A}^{(n)}$ donne :
$\partial^2\widehat{f}_{n,t}(\mathbf{A}^{(n)}) = \mathbf{S}_t^T \otimes \mathrm{I}_{I_n} + (1-\theta)\alpha\mathrm{I}_{I_nJ_n}, \mathrm{I}_{I_nJ_n} \in \mathbb{R}^{I_nJ_n \times I_nJ_n}$ et $\mathrm{I}_{I_n} \in \mathbb{R}^{I_n \times I_n}$ des matrices identités
$\Rightarrow \partial^2 f_{n,t}(\mathbf{A}^{(n)}) = \mathbf{S}_t \otimes \mathrm{I}_{I_n} + (1-\theta)\alpha\mathrm{I}_{I_nJ_n}$ (symétrie de la matrice $\mathbf{S}_t$)
Des calculs simples utilisant des propriétés connues sur le produit de **Kronecker** (produit mixte ou "mixed prodiuct" : $(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = \mathbf{AC}\otimes\mathbf{BD}$) donnent :
$\mathbf{S}_t \otimes \mathbf{I}_{I_n} = \frac{1}{t}\sum_{i=1}^{t}(\mathbf{B}_i^{(n)} \otimes \mathrm{I}_{I_n})(\mathbf{B}_i^{(n)} \otimes \mathrm{I}_{I_n})^T$
Donc, $\partial^2 f_{n,t}(\mathbf{A}^{(n)})$ est une matrice symétrique positive avec toutes ses valeurs propres bornées inférieurement par $(1-\theta)\alpha$.

**Propriété 7** *La fonction de perte $\ell$ définie dans le rappel des notations est une fonction Lipschitzienne par rapport à $\mathbf{A}^{(n)}$ et uniformément bornées.*

Preuve :

**A. Bornitude**

Ceci est trivial par l'inégalité triangulaire et la bornitude des facteurs latents..

**B. $\ell$ est Lipschitzienne pr rapport à $\mathbf{A}^{(n)}$**

La fonction $\mathbf{A}^{(n)} \to \frac{1}{2}\|\mathcal{X} - \mathcal{G}\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{A}^{(n)}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$ est différentiable et sa derivée est continue sur $\mathbb{R}^{J_1\times J_2\cdots\times J_N}\times\mathbb{R}^{I_n\times J_n}$ (car c'est un polynôme en les entrées de $\mathcal{G}$ et $\mathbf{A}^{(n)}$). Étant donné le problème de minimisation par rapport à $\mathcal{G}$ est un problème de minimisation sur un ensemble compact d'une fonction continue, il admet une solution qui est unique (par *Hypothèse III*). Par conséquent, par le théorème de Bonnan et Shapiro, the function lost function $\mathbf{A}^{(n)} \to \ell(\mathcal{X}, \mathbf{A}^{(n)}) = \min_{\mathcal{G}}\frac{1}{2}\|\mathcal{X} - \mathcal{G}\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{A}^{(n)}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$ est differentiable et nous avons par **Propriété** 4 :

$$\partial_2\ell(\mathcal{X}, \mathbf{A}^{(n)}) = -\left(\widehat{\mathbf{X}}^{(n)}\mathbf{G}_0^{(n)T}(\mathcal{X}, \mathbf{A}^{(n)}) - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\mathbf{B}^{(n)T}\right) + \alpha(1-\theta)\mathbf{A}^{(n)}$$

avec $\widehat{\mathbf{X}}^{(n)}$ et $\mathbf{B}^{(n)}$ étant les formes matricielles des tenseurs $\widehat{\mathcal{X}}$ et $\mathcal{B}$ définis par :
$\widehat{\mathcal{X}} = \mathcal{X}\times_{p\in I_{n-1}}\mathbf{A}^{(p)T}\times_n\mathbf{I}\times_{q\in I_N^{n+1}}\mathbf{A}^{(q)T}, \mathcal{B} = \mathbf{G}_0\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{I}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}$
$\mathcal{G}_0(\mathcal{X}, \mathbf{A}^{(n)}) = \arg\min_{\mathcal{G}}\|\mathcal{X} - \mathcal{G}\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{A}^{(n)}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$.
Comme la dérivée est bornée, $\ell(\mathcal{X}, .)$ est une fonction Lipschitzienne par le théorème des accroissements finis.

**Propriété 8** $\widehat{f}_{n,t}$ *est une fonction Lipschitzienne et bornée par une constante indepen-dente de* $t$.

Proof :
$\widehat{f}_{n,t}\left(\mathbf{A}^{(n)}\right) = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|\mathbf{X}_i^{(n)} - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\|_F^2 + \alpha\theta\|\mathcal{G}_i\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$ avec :
$\mathbf{B}^{(n)} = \mathbf{G}_i^{(n)}\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T}\otimes_{q\in I_N^{n+1}}\mathbf{A}_{k+1}^{(q)T}\right), \mathbf{X}_i^{(n)}$ et $\mathbf{G}_i^{(n)}$ étant les formes matricielles des tenseurs $\mathcal{X}_i$ et $\mathcal{G}_i$.
$\widehat{f}_{n,t}\left(\mathbf{A}_1^{(n)}\right) - \widehat{f}_{n,t}\left(\mathbf{A}_2^{(n)}\right) = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left(\|\mathbf{X}_i^{(n)} - \mathbf{A}_1^{(n)}\mathbf{B}^{(n)}\|_F^2 - \|\mathbf{X}_i^{(n)} - \mathbf{A}_2^{(n)}\mathbf{B}^{(n)}\|_F^2\right)$
$+\frac{\alpha(1-\theta)}{2}\left(\|\mathbf{A}_1^{(n)}\|_F^2 - \|\mathbf{A}_2^{(n)}\|_F^2\right)$
$\Rightarrow|\ \widehat{f}_{n,t}\left(\mathbf{A}_1^{(n)}\right) - \widehat{f}_{n,t}\left(\mathbf{A}_2^{(n)}\right)\ |\leq \frac{1}{t}\sum_{i=1}^{t} M_1\|\left(\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\right)\mathbf{B}^{(n)}\|_F + M_2\|\left(\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\right)\|_F$ (**Propriété** 1)
$\Rightarrow|\ \widehat{f}_{n,t}\left(\mathbf{A}_1^{(n)}\right) - \widehat{f}_{n,t}\left(\mathbf{A}_2^{(n)}\right)\ |\leq \frac{1}{t}\sum_{i=1}^{t} M_3\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F + M_2\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F$ (bornitude des facteurs)
$\Rightarrow|\ \widehat{f}_{n,t}\left(\mathbf{A}_1^{(n)}\right) - \widehat{f}_{n,t}\left(\mathbf{A}_2^{(n)}\right)\ |\leq (M_3 + M_2)\,\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F$

**Propriété 9** $f_{n,t}$ *est une fonction Lipschitzienne et bornée par une constante indépendante de* $t$.

Preuve : puisque la fonction de perte $\ell$ est Lipschitzienne par rapport à $\mathbf{A}^{(n)}$, la fonction $f_{n,t}(\mathbf{A}^{(n)}) = \frac{1}{t}\sum_{i=1}^{t}\ell(\mathcal{X}_i, \mathbf{A}^{(n)})$ l'est aussi (comme combinaison de linéaire avec des coefficients positifs de fonctions Lipschitziennes). Le raisonnement est identique pour la bornitude.

**Propriété 10** *Considerons la fonction* $\xi(\mathcal{X}, \mathbf{A}^{(n)}, \mathcal{G})$ *définie par :*
$\xi(\mathcal{X}, \mathbf{A}^{(n)}, \mathcal{G}) = \frac{1}{2}\|\mathcal{X} - \mathcal{G}\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{A}^{(n)}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}^{(n)}\|_F^2$.
*Notons par* $\Delta(\mathcal{G})$ *la fonction définie par :* $\Delta(\mathcal{G}) = \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}) - \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G})$. *La fonction* $\Delta$ *est Lipschitzienne.*

Preuve :
$\Delta(\mathcal{G}_\alpha) - \Delta(\mathcal{G}_\beta) = \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\alpha) - \left(\xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\beta) - \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\beta)\right)$
$\Rightarrow \Delta(\mathcal{G}_\alpha) - \Delta(\mathcal{G}_\beta) = \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\beta) - \left(\xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\beta)\right)$
$\Rightarrow|\ \Delta(\mathcal{G}_\alpha) - \Delta(\mathcal{G}_\beta)\ |\leq \underbrace{|\ \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\beta)\ |}_{FT}$
$+\underbrace{|\ \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_2, \mathbf{A}_2^{(n)}, \mathcal{G}_\beta)\ |}_{ST}$
$\xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\alpha) - \xi(\mathcal{X}_1, \mathbf{A}_1^{(n)}, \mathcal{G}_\beta) = \frac{1}{2}\|\mathbf{X}_1^{(n)} - \mathbf{A}_1^{(n)}\mathbf{G}_\alpha^{(n)}\mathbf{M}^{(n)}\|_F^2 + \alpha\theta\|\mathcal{G}_\alpha\|_1 + \frac{\alpha(1-\theta)}{2}\|\mathbf{A}_1^{(n)}\|_F^2 - \frac{1}{2}\|\mathbf{X}_1^{(n)} - \mathbf{A}_1^{(n)}\mathbf{G}_\beta^{(n)}\mathbf{M}^{(n)}\|_F^2 - \alpha\theta\|\mathcal{G}_\beta\|_1 - \frac{\alpha(1-\theta)}{2}\|\mathbf{A}_1^{(1)}\|_F$
$FT \leq M_1\|\mathbf{G}_\alpha^{(n)} - \mathbf{G}_\beta^{(n)}\|_F + \alpha\theta\|\|\mathcal{G}_\alpha\| - \|\mathcal{G}_\beta\|_1\|$ (bornitude+*Propriété* 1)
$\Rightarrow FT \leq M_1\|\mathcal{G}_\alpha - \mathcal{G}_\beta\|_F + M_2\|\mathcal{G}_\alpha - \mathbf{G}_\beta\|_F$ (seconde inégalité triangulaire+équivalence des normes). Un raisonnement sur $ST$ conclut la démonstration.

**Propriété 11** *Soit* $\mathbf{A}$ *une matrice avec des entrées aléatoire ,* $\exists\beta > 0, \|\mathbb{E}(\mathbf{A})\|_F \leq \beta\mathbb{E}(\|\mathbf{A}\|_F)$ *avec* $(\mathbb{E}(\mathbf{A}))_{i,j} = \mathbb{E}(\mathbf{A}_{i,j})$

**Preuve** :

Puisque l'ensemble des matrices est un espace de dimension finie et toutes les normes dans un espace vectoriel de dimension finie sont équivalentes, il existe $\beta_1, \beta_2 > 0$ (déterministes) telles que :

$\beta_1\|\mathbf{A}\|_F \leq \|\mathbf{A}\|_1 \leq \beta_2\|\mathbf{A}\|_F$ et $\beta_1\|\mathbb{E}(\mathbf{A})\|_F \leq \|\mathbb{E}(\mathbf{A})\|_1 \leq \beta_2\|\mathbb{E}(\mathbf{A})\|_F$

Donc, on a :

$\|\mathbb{E}(\mathbf{A})\|_F \leq \frac{1}{\beta_1}\|\mathbb{E}(\mathbf{A})\|_1 \leq \frac{1}{\beta_1}\mathbb{E}(\|\mathbf{A}\|_1) \leq \frac{\beta_2}{\beta_1}\mathbb{E}(\|\mathbf{A}\|_F)$

Les première et troisième inégalités proviennent de l'équivalence des normes et la séconde inégalité de l'inégalité triangulaire.

# 5 Convergence par blocs

Le raisonnement effectué dans cette section est inspiré de l'analyse effectuée dans [86]. L'objectif est de prouver que la suite $\mathbf{A}_{k+1,t}^{(n)}$ converge vers l'ensemble des points stationnaires de la fonction $\mathbf{A}^{(n)} \rightarrow f(\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(n)})$. Pour des raisons de simplicité, on note $\mathbf{A}_{k+1,t}^{(n)}$ par $\mathbf{A}_t^{(n)}$.

## 5.1 Borne sur la norme de $\mathbf{A}_t^{(n)} - \mathbf{A}_{t+1}^{(n)}$ : $\exists M > 0, \|\mathbf{A}_t^{(n)} - \mathbf{A}_{t+1}^{(n)}\|_F \leq \frac{M}{t}$

Preuve :

Comme $\widehat{f}_{n,t}(\mathbf{A}^{(n)})$ est strictement convexe avec une hessienne bornée inférieurement (valeurs propres bornées inférieurement) (*Propriété* 6), on a alors :

$$\widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) \geq \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) + M\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F^2 \tag{6.7}$$

En plus :

$$\begin{aligned}\widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) &= \widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) + \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) \\ &\leq \widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t+1}(\mathbf{A}_{t+1}^{(n)}) + \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) : \text{définition de } \mathbf{A}_{t+1}^{(n)} \\ \Rightarrow \widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) &\leq \widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t+1}(\mathbf{A}_{t+1}^{(n)}) + \widehat{f}_{n,t+1}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})\end{aligned}$$

Cela implique que :

$$\widehat{f}_{n,t}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) \leq Q_t(\mathbf{A}_{t+1}^{(n)}) - Q_t(\mathbf{A}_t^{(n)}), Q_t(\mathbf{A}^{(n)}) = \widehat{f}_{n,t}(\mathbf{A}^{(n)}) - \widehat{f}_{n,t+1}(\mathbf{A}^{(n)}) \tag{6.8}$$

Il est à noter que (par définition de $\mathbf{A}_t^{(n)}$) :

$$\mathbf{Q}_t(\mathbf{A}_{t+1}^{(n)}) - \mathbf{Q}_t(\mathbf{A}_t^{(n)}) \geq 0 \tag{6.9}$$

$\widehat{f}_{n,t+1}(\mathbf{A}^{(n)}) = \frac{t}{t+1}\widehat{f}_{n,t}(\mathbf{A}^{(n)}) + \frac{\alpha(1-\theta)}{2(t+1)}\|\mathbf{A}^{(n)}\|_F^2 + \frac{1}{2(t+1)}\|\mathcal{X}_{t+1} - \mathcal{G}_{t+1} \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q\in I_N^{n+1}} \mathbf{A}^{(q)}\|_F^2$

Donc, on a :

$Q_t(\mathbf{A}^{(n)}) = -\frac{1}{t}\widehat{f}_{n,t}(\mathbf{A}^{(n)}) + \frac{1}{t+1}\left(\frac{1}{2}\|\mathcal{X}_{t+1} - \mathcal{G}_{t+1} \times_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)} \times_n \mathbf{A}^{(n)} \times_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)}\|_F^2 + \alpha\theta\|\mathcal{G}_{t+1}\|_1\right) + \frac{1}{2(t+1)}\|\mathbf{A}^{(n)}\|_F^2$

$\Rightarrow Q_t(\mathbf{A}_1^{(n)}) - Q_t(\mathbf{A}_2^{(n)}) = \frac{1}{t}\left(\widehat{f}_{n,t}(\mathbf{A}_2^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_1^{(n)})\right) + \frac{1}{2(t+1)}\left(\|\mathbf{X}_{t+1}^{(n)} - \mathbf{A}_1^{(n)}\mathbf{B}^n\|_F^2 - \|\mathbf{X}_{t+1}^{(n)} - \mathbf{A}_2^{(n)}\mathbf{B}^n\|_F^2\right) + \frac{\alpha(1-\theta)}{2(t+1)}(\|\mathbf{A}_1^{(n)}\|_F^2 - \|\mathbf{A}_2^{(n)}\|_F^2), \mathbf{B}^{(n)} = \mathbf{G}_{t+1}^{(n)}\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T} \otimes_{q\in I_N^{n+1}} \mathbf{A}_k^{(q)T}\right)$

$\Rightarrow| Q_t(\mathbf{A}_1^{(n)}) - Q_t(\mathbf{A}_2^{(n)}) | \leq \frac{1}{t}\|\widehat{f}_{n,t}(\mathbf{A}_1^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_2^{(n)})\| + \frac{M}{2(t+1)}\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F$ (bornitude+*Propriété* 1)

$\Rightarrow Q_t(\mathbf{A}_{t+1}^{(n)}) - Q_t(\mathbf{A}_t^{(n)}) \leq \frac{M_2}{t}\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F + \frac{M}{2(t+1)}\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F \leq \frac{M_3}{t}\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F$

(*Propriété* 8+equation (6.9))

Donc, on a avec les équations (6.7) et (6.8) :

$M\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F^2 \leq \frac{M_3}{t}\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F$. En supposant que $\mathbf{A}_t^{(n)} \neq \mathbf{A}_{t+1}^{(n)}$, on a :

$\|\mathbf{A}_{t+1}^{(n)} - \mathbf{A}_t^{(n)}\|_F = O(\frac{1}{t})$

## 5.2 Le gradient de la fonction $f_n$ est Lipschitzienne

Preuve :

$f_n(\mathbf{A}^{(n)}) = \mathbb{E}_{\mathbb{P}}(\ell(\mathcal{X}, \mathbf{A}^{(n)}))$

La fonction $\mathbf{A} \to \ell(\mathcal{X}, \mathbf{A})$ est différentiable et sa dérivée est bornée (*Propriété* 7) par une constante K, qui est une fonction integrable. Par le théorème de dérivation sous intégrale, on a alors :

$\partial f_n(\mathbf{A}) = \mathbb{E}_{\mathbb{P}}\left(\partial_2\ell(\mathcal{X}, \mathbf{A})\right)$

$$\Rightarrow \partial f_n(\mathbf{A}) = \mathbb{E}_{\mathbb{P}}\left(\underbrace{-\left(\widehat{\mathbf{X}}^{(n)}\mathbf{G}_0^{(n)T}(\mathcal{X}, \mathbf{A}) - \mathbf{A}\mathbf{B}^{(n)}\mathbf{B}^{(n)T}\right) + (1-\theta)\alpha\mathbf{A}}_{\mathbf{V}(\mathbf{A})}\right)$$

avec : $\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}) = \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A})\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T} \otimes_{q\in I_N^{n+1}} \mathbf{A}_k^{(p)T}\right)$

$\mathbf{V}(\mathbf{A}_1) - \mathbf{V}(\mathbf{A}_2) = -2\widehat{\mathbf{X}}^{(n)}\left(\mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1) - \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_2)\right) + \alpha(1-\theta)(\mathbf{A}_1 - \mathbf{A}_2)$

$+2\left(\mathbf{A}_1\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}_1)\mathbf{B}^{(n)T}(\mathcal{X}, \mathbf{A}_1) - \mathbf{A}_2\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}_2)\mathbf{B}^{(n)T}(\mathcal{X}, \mathbf{A}_2)\right)$

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1) - \mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1) - \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_2)\|_F + 2\|\mathbf{A}_1\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}_1)\mathbf{B}^{(n)T}(\mathcal{X}, \mathbf{A}_1) - \mathbf{A}_2\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}_2)\mathbf{B}^{(n)T}(\mathcal{X}, \mathbf{A}_2)\|_F + \alpha(1-\theta)\|\mathbf{A}_1 - \mathbf{A}_2\|_F$

Comme toutes les matrices sont bornées, on a :

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1) - \mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1) - \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_2)\|_F + 2\alpha_2\|\mathbf{A}_1 - \mathbf{A}_2\|_F$

$+2\alpha_3\|\mathbf{B}_n(\mathcal{X}, \mathbf{A}_1)\mathbf{B}_n^T(\mathcal{X}, \mathbf{A}_1) - \mathbf{B}_n(\mathcal{X}, \mathbf{A}_2)\mathbf{B}_n^T(\mathcal{X}, \mathbf{A}_2)\|_F + \alpha(1-\theta)\|\mathbf{A}_1 - \mathbf{A}_2\|_F$ (*Propriété* 2)

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1) - \mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1) - \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1)\|_F + (2\alpha_2 + \alpha(1-\theta))\|\mathbf{A}_1 - \mathbf{A}_2\|_F +$

$2\alpha_3(\alpha_4\|\mathbf{B}_n(\mathcal{X}, \mathbf{A}_1) - \mathbf{B}_n(\mathcal{X}, \mathbf{A}_2)\|_F + \alpha_5\|\mathbf{B}_n^T(\mathcal{X}, \mathbf{A}_1) - \mathbf{B}_n^T(\mathcal{X}, \mathbf{A}_2)\|_F)$(*Propriété* 2)

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1) - \mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_1) - \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}_2)\|_F + (2\alpha_2 + \alpha(1-\theta))\|\mathbf{A}_1 - \mathbf{A}_2\|_F +$

$\alpha_6\|\mathbf{B}_n(\mathcal{X},\mathbf{A}_1)-\mathbf{B}_n(\mathcal{X},\mathbf{A}_2)\|_F$ (parce que le terme $\|\mathbf{B}^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{B}^{(n)}(\mathcal{X},\mathbf{A}_2)\|_F$ apparaît deux fois)

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2)\|_F+(2\alpha_2+\alpha(1-\theta))\|\mathbf{A}_1-\mathbf{A}_2\|_F+$
$\alpha_6\|\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T}\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(p)T}\right)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2)\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T}\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(p)T}\right)\|_F$
( en remplaccant l'expression de $\mathbf{B}^{(n)}(\mathcal{X},\mathbf{A})$)

$\Rightarrow \|\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_1\|\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2)\|_F+(2\alpha_2+\alpha(1-\theta))\|\mathbf{A}_1-\mathbf{A}_2\|_F$
$+\alpha_6\|(\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2))\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)T}\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(p)T}\right)\|_F$ (distributivité du produit de Kronecker)

Par la bornitude des matrices de dictionnaires, on a :

$$\|\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2)\|_F \leq \alpha_8\|\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2)\|_F+(2\alpha_2+\alpha(1-\theta))\|\mathbf{A}_1-\mathbf{A}_2\|_F \quad (6.10)$$

Par définition de $\partial f_n$, on a :

$\partial f_n(\mathbf{A}_1)-\partial f_n(\mathbf{A}_2)=\mathbb{E}_{\mathbb{P}}(\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2))$

$\Rightarrow \|\partial f_n(\mathbf{A}_1)-\partial f_n(\mathbf{A}_2)\|_F=\|\mathbb{E}_{\mathbb{P}}(\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2))\|_F$

Par *Propriété* 11,$\exists\beta>0$ tel que :

$\|\partial f_n(\mathbf{A}_1)-\partial f_n(\mathbf{A}_2)\|_F \leq \beta\mathbb{E}_{\mathbb{P}}(\|\mathbf{V}(\mathbf{A}_1)-\mathbf{V}(\mathbf{A}_2)\|_F$

Par l'inégalité (6.10), on a :

$\|\partial f_n(\mathbf{A}_1)-\partial f_n(\mathbf{A}_2)\|_F \leq M_1\mathbb{E}_{\mathbb{P}}(\|\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0^{(n)}(\mathcal{X},\mathbf{A}_2)\|_F)+M_2\|\mathbf{A}_1-\mathbf{A}_2\|_F$

Il siffit de prouver que $\mathbf{G}_0$ est Lipschitzienne pour prouver le résulat

Par définition, on a :

$$\begin{aligned}\mathbf{G}_0(\mathcal{X},\mathbf{A}) &\leftarrow \arg\min\frac{1}{2}\|\mathcal{X}-\mathcal{G}\times_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\times_n\mathbf{A}\times_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\|_F^2+\alpha\theta\|\mathcal{G}\|_1+\frac{\alpha(1-\theta)}{2}\|\mathbf{A}\|_F^2\\ &=\arg\min_{\mathcal{G}}\frac{1}{2}\operatorname{vec}(\mathcal{X})-\left(\otimes_{p\in I_{n+1}}\mathbf{A}_{k+1}^{(p)}\otimes\mathbf{A}\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\right)\operatorname{vec}(\mathcal{G})\|_F+\alpha\theta\|\operatorname{vec}(\mathcal{G})\|_1\end{aligned}$$

Étant donné *hypothèse III* ( unicité du problème de *Codage parcimonieux*), *Propriété* 10 avec le fait que le problème de *Codage Parcimonieux* est équivalent u problème de régression Lasso par vectorisation de la décomposition de **Tucker**, un raisonnement identique à celui effectué dans [86] : proposition 2 donne :

$\|\mathbf{G}_0(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0(\mathcal{X},\mathbf{A}_2)\|_F$
$\leq M_1\|\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\otimes\mathbf{A}_1\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\right)-\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\otimes\mathbf{A}_2\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\right)\|_F$
$=M_1\|\left(\otimes_{p\in I_{n-1}}\mathbf{A}_{k+1}^{(p)}\otimes(\mathbf{A}_1-\mathbf{A}_2)\otimes_{q\in I_N^{n+1}}\mathbf{A}_k^{(q)}\right)\|_F$ (distributivité du problème de Kronecker)

$\Rightarrow \|\mathbf{G}_0(\mathcal{X},\mathbf{A}_1)-\mathbf{G}_0(\mathcal{X},\mathbf{A}_2)\|_F \leq M_3\|\mathbf{A}_1-\mathbf{A}_2\|_F$, ce qui conclut la démonstration.

## 5.3 La suite $f_n(\mathbf{A}_t^{(n)})$ converge

**Première étape** : $\widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$ converge presque sûrement.

**Preuve**

$u_t = \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$

$u_{t+1} - u_t = \widehat{f}_{n,t+1}(\mathbf{A}_{t+1}^{(n)}) - \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) + \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$

$\Rightarrow u_{t+1} - u_t \leq \widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$ (par définition de $\mathbf{A}_{t+1}^{(n)}$)

$\widehat{f}_{n,t+1}(\mathbf{A}_t^{(n)}) = \frac{t}{t+1}\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) + \frac{1}{t+1}\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)})$ (par définition de $\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)})$ et $\mathcal{G}_{t+1}$)

Par conséquent, on a :

$$u_{t+1} - u_t \leq \frac{\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})}{t+1} = \frac{\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)}) + f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})}{t+1} \tag{6.11}$$

En notant $f_{n,t} \leq \widehat{f}_{n,t}$, on a :

$$u_{t+1} - u_t \leq \frac{\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1} \tag{6.12}$$

En conditionnant l'expression (6.11) par rapport à la filtration $\mathcal{F}_t$ (filtration déterminée par rapport à l'information antérieure à la date $t$), on a :

$\mathbb{E}(u_{t+1} - u_t|\mathcal{F}_t) \leq \frac{\mathbb{E}(\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)})|\mathcal{F}_t) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1} = \frac{\mathbb{E}(\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)})) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}$ (puisque $\ell(\mathfrak{X}_{t+1}, \mathbf{A}_t^{(n)})$ est indépendante de la filtration $\mathcal{F}_t$)

$\Rightarrow \mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t) \leq \frac{f_n(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}$(définition de $f_n(\mathbf{A}_t^{(n)})$)

$\Rightarrow \mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t) \leq \frac{\|f_n - f_{n,t}\|_\infty}{t+1}$,(bornitude des fonctions)

$\Rightarrow \mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t)^+ \leq \frac{\|f_n - f_{n,t}\|_\infty}{t+1}$(car $\frac{\|f_n - f_{n,t}\|_\infty}{t+1} \geq 0$)

Donc, on a :

$$\mathbb{E}_{\mathbb{P}}(\mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t)^+) \leq \frac{\mathbb{E}_{\mathbb{P}}\left(\|f_n - f_{n,t}\|_\infty\right)}{t+1} \tag{6.13}$$

La fonction de perte $\ell(\mathfrak{X}, \mathbf{A}^{(n)})$ est uniformément bornée et $\mathbb{E}_{\mathbb{P}}(\ell^2(\mathfrak{X}, \mathbf{A}^{(n)}))$ existe et est bornée (parce que la fonction de perte $\ell$ est uniformément bornée). Par conséquent, conformément à *Théorème IV* (voir en base de page la section sur "les divers résultats pour l'analyse de **OTL**"), $\exists M > 0$ tel que :

$\mathbb{E}_{\mathbb{P}}\left(\sqrt{t} \mid \frac{1}{t}\sum_{i=1}^{t} \ell(\mathfrak{X}_i, \mathbf{A}^{(n)}) - \mathbb{E}_{\mathbb{P}}(\ell(\mathfrak{X}, \mathbf{A}^{(n)}) \mid\right) \leq M$,$\mathfrak{X}$ suivant la distribution $\mathbb{P}, \forall \mathbf{A}^{(n)}$

$\Rightarrow \mathbb{E}_{\mathbb{P}}\left(\sqrt{t} \mid f_{n,t}(\mathbf{A}^{(n)}) - f_n(\mathbf{A}^{(n)}) \mid\right) \leq M, \forall \mathbf{A}^{(n)}$

$\Rightarrow \mathbb{E}_{\mathbb{P}}\left(\|f_{n,t} - f_n\|_\infty\right) \leq \frac{M}{\sqrt{t}}$

Cette inégalité associée avec (6.13) donne :

$\mathbb{E}_{\mathbb{P}}(\mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t)^+) \leq \frac{M}{\sqrt{t(t+1)}} \leq \frac{M}{t^{\frac{3}{2}}}$.

$\Rightarrow \sum_{t=1}^{\infty} \mathbb{E}_{\mathbb{P}}(\delta_t(u_{t+1} - u_t)) = \sum_{t=1}^{\infty} \mathbb{E}_{\mathbb{P}}(\mathbb{E}_{\mathbb{P}}(u_{t+1} - u_t|\mathcal{F}_t)^+) < \infty$($\delta_t$ définie par *Théorème*

*III* ). Par *Théorème III*, $u_t$ converge presque sûrement vers $u_\infty$.

**Deuxième étape** : $f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) \to 0$ presque sûrement

Par l'équation (6.11), on a :

$u_{t+1} - u_t \leq \frac{\ell(\mathcal{X}_{t+1}, \mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1} + \frac{f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})}{t+1}$

$\Rightarrow \mathbb{E}(u_{t+1} - u_t) \leq \mathbb{E}(\frac{f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})}{t+1})$ (l'espérance du premier terme est zéro car les observations dans l'échantillon sont censées être tirées de la même distribution)

$\Rightarrow \mathbb{E}_{\mathbb{P}}\left(\frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}\right) \leq \mathbb{E}_{\mathbb{P}}(u_t - u_{t+1})(*)$

$\Rightarrow \sum_{t=1}^{T-1} \mathbb{E}_{\mathbb{P}}(\frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}) \leq \sum_{t=1}^{T-1} \mathbb{E}_{\mathbb{P}}(u_t - u_{t+1}) = \mathbb{E}_{\mathbb{P}}(u_1 - u_T)(**)$

La suite $\{u_1 - u_T\}_T$ converge (**Première étape**). Puisqu'elle est bornée, par le théorème de la convergence dominée, la suite $\{\mathbb{E}_{\mathbb{P}}(u_1 - u_T)\}_T$ converge. Par conséquent, on déduit de l'inégalité (**) la convergence de la série numérique de terme général $\mathbb{E}_{\mathbb{P}}(u_t - u_{t+1})$. Comme les séries $\mathbb{E}_{\mathbb{P}}(\frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1})$ et $\mathbb{E}_{\mathbb{P}}(u_t - u_{t+1})$ sont non-négatives (la non-négativité des premier et second termes résulte de la définition et de (*)) et la série $\mathbb{E}_{\mathbb{P}}(u_t - u_{t+1})$ converge , par l'inégalité (*), la série $\mathbb{E}_{\mathbb{P}}(\frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1})$ converge. Par conséquent, on a :

$\sum_{t=1}^{\infty} \mathbb{E}_{\mathbb{P}}\left(\frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}\right) < \infty$

$\Rightarrow \mathbb{E}_{\mathbb{P}}(\sum_{t=1}^{\infty} \frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}) < \infty$ (par le théorème de Fubini par les variables aléatoires non-négatives)

Étant donné $\sum_{t=1}^{\infty} \frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1}$ est une variable aléatoire non-négative et son expérance est finie, cela implique (voir [38] : page 9) :

$$\sum_{t=1}^{\infty} \frac{\widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) - f_{n,t}(\mathbf{A}_t^{(n)})}{t+1} < \infty \text{presque sûrement} \tag{6.14}$$

Considérons deux suites $a_t, b_t$ définies par : $a_t = \frac{1}{t+1}, b_t = f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$

Il est complètement trivial de voir que les hypothèses de *Théorème II* sont vérifiées. Donc, on a :

$f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) \to 0$ presque sûrement quand $t \to \infty$

**Troisième étape** : $f_n(\mathbf{A}_t^{(n)})$ converge presque.

Par l'inégalité triangulaire et la définition de $|.|_\infty$, on a :

$$| f_n(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) | \leq \|f_n - f_{n,t}\|_\infty + | f_{n,t}(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) | \tag{6.15}$$

La classe de fonctions $\mathcal{F} = \left\{f : \Omega \times \mathbb{K}_{I_n,J_n} \to \mathbb{R}, f(\mathcal{X}, \mathbf{A}^{(n)}) = l(\mathcal{X}, \mathbf{A}^{(n)})\right\}, \Omega$ représentant l'ensemble des tenseurs à support compact. Puisque $\mathbb{K}_{I_n,J_n}$ est un compact et la classe de fonctions $\mathcal{F}$ vérifie *Théorème V* , on a :

$\lim_{t\to\infty} | \frac{1}{t} \sum_{i=1}^{t} l(\mathcal{X}_i, \mathbf{A}^{(n)}) - \mathbb{E}_{\mathbb{P}}(l(\mathcal{X}, \mathbf{A}^{(n)})) | = 0, \forall \mathbf{A}^{(n)}$

$\Rightarrow \lim_{t\to\infty} \mid \frac{1}{t}\sum_{i=1}^{t} l(\mathcal{X}_i, \mathbf{A}^{(n)}) - \mathbb{E}_{\mathbb{P}}(l(\mathcal{X}, \mathbf{A}^{(n)})) \mid = 0, \forall \mathbf{A}^{(n)}$

$\Rightarrow \lim_{t\to\infty} \mid f_{n,t}(\mathbf{A}^{(n)}) - f_n(\mathbf{A}^{(n)}) \mid = 0, \forall \mathbf{A}^{(n)}$

$\Rightarrow \lim_{t\to\infty} \|f_{n,t} - f_n\|_\infty = 0$

Par le théorème de Glivenko-Cantelli et la **Deuxième étape**, l'inégalité (6.15) donne :

$\mid f_n(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) \mid \to 0$ quand $t \to \infty$

$\Rightarrow f_n(\mathbf{A}_t^{(n)}) - \widehat{f}_{n,t}(\mathbf{A}_t^{(n)}) \to 0$ a.s. $t \to \infty$

$\Rightarrow f_n(\mathbf{A}_t^{(n)})$ converge presque sûrement quand $t \to \infty$ (puisque $\widehat{f}_{n,t}(\mathbf{A}_t^{(n)})$ converge presque sûrement par la **Première étape**)

## 5.4 $\mathbf{A}_t^{(n)}$ converge vers un point stationnaires de $f_n$

**Preuve** :

Avec *Hypothèse I*, *Hypothèse II*, *Hypothèse III*, un raisonnement identique à celui présenté dans [86] (*Proposition 4*) donne ce résultat de convergence.

# 6 Convergence globale

Dans cette section, on étend le résultat de la section précédente à l'ensemble des points stationnaires de la fonction $f$ considérée comme fonction de toutes les matrices de dictionnaires.

Avec la section 5, $\forall n, \mathbf{A}_t^{(n)}$ converge vers $\mathbf{A}_\infty^{(n)} = \mathbf{A}_{k+1}^{(n)}$, un point stationnaire de $f_n$ sur $\mathbb{K}_{I_n,J_n}$. Cela implique que la dérivée de $-\frac{\partial f_n}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_{k+1}^{(n)})$ appartient au cône normal de l'ensemble $\mathbb{K}_{I_n,J_n}$ au point $\mathbf{A}_{k+1}^{(n)}$. Puisque $\mathbf{A}_{k+1}^{(n)}$ est un point intérieur de $\mathbb{K}_{I_n,J_n}$ (*Hypothèse V*), il est bien connu que ce cône normal est réduit au singleton $\{0\}$. Par conséquenct :

$\frac{\partial f}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{k+1}^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(N)}) = 0$ puisque $\mathbf{A}^{(n)} \to f(.., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}_k^{(n+1)}, ..)$ et $\mathbf{A}^{(n)} \to f_n(\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(N)})$ sont proportionnels à une constante additive près.

On a prouvé au niveau de la section 5.2 que :

$\frac{\partial f(\mathbf{A})}{\partial \mathbf{A}^{(n)}}(\mathbf{A}^{(1)}, .., \mathbf{A}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}^{(n+1)}, .., \mathbf{A}^{(N)}) = \mathbb{E}_{\mathbb{P}}\left(\mathbf{V}(\mathbf{A}^{(1)}, .., \mathbf{A}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}^{(n+1)}, .., \mathbf{A}^{(N)})\right)$

$\mathbf{V}(\mathbf{A}^{(1)}, .., \mathbf{A}^{(n-1)}, \mathbf{A}^{(n)}, \mathbf{A}^{(n+1)}, .., \mathbf{A}^{(N)}) = -\left(\widehat{\mathbf{X}}^{(n)}\mathbf{G}_0^{(n)T}(\mathcal{X}, \mathbf{A}^{(n)}) - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\mathbf{B}^{(n)T}\right) + (1 - \theta)\alpha\mathbf{A}^{(n)}$, $\mathbf{B}^{(n)}(\mathcal{X}, \mathbf{A}^{(n)}) = \mathbf{G}_0^{(n)}(\mathcal{X}, \mathbf{A}^{(n)}) \otimes_{p\in I_{n-1}} \mathbf{A}_{k+1}^{(p)T} \otimes_{q\in I_N^{n+1}} \mathbf{A}_k^{(p)T}$

Puisque la fonction $\mathbf{A}^{(n)} \to \mathbf{G}_0(\mathcal{X}, \mathbf{A}^{(n)})$ est uniformement continue car Lipschitzienne (prouvé dans le section 4), la fonction $\mathbf{V}$ est aussi uniformément continue. Donc, on a :

$\forall \epsilon > 0, \exists \eta > 0, \forall x, y, \|x - y\|_{\prod_{n\in I_N} \mathbb{R}^{I_n \times J_n}} \leq \eta \Rightarrow \|\mathbf{V}(x) - \mathbf{V}(y)\|_F \leq \epsilon$

Posons $y = (\mathbf{A}_k^{(1)}, .., \mathbf{A}_k^{(n-1)}, \mathbf{A}_k^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(N)})$ et $x = (\mathbf{A}_{k+1}^{(1)}, .., \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{k+1}^{(n)}, \mathbf{A}_k^{(n+1)}, .., \mathbf{A}_k^{(N)})$

Avec *Hypothèse IV*, on a :

$\|x-y\|_{\prod_{n\in I_N}\mathbb{R}^{I_n\times J_n}} \leq \frac{n}{k^{\frac{1}{2}}}$

Par convergence de $\frac{n}{k^{\frac{1}{2}}}$ vers 0 quand k tend vers $+\infty$ ($n$ étant fixé) :

$\exists k_0 \in \mathbb{N}, \forall k \geq k_0, \frac{n}{k^{\frac{1}{2}}} \leq \eta$

Par conséquent, $\forall k \geq k_0$, on a :

$\|x-y\|_{\prod_{n\in I_N}\mathbb{R}^{I_n\times J_n}} \leq \eta$

$\Rightarrow \|\mathbf{V}(x)-\mathbf{V}(y)\|_F \leq \epsilon$

$\Rightarrow \mathbb{E}_{\mathbb{P}}\left(\|\mathbf{V}(x)-\mathbf{V}(y)\|_F\right) \leq \epsilon$

$\Rightarrow \|\mathbb{E}_{\mathbb{P}}(\mathbf{V}(x))-\mathbb{E}(\mathbf{V}(y))\|_F \leq \beta\epsilon$ (*Propriété* 11)

Par définition, $\mathbb{E}_{\mathbb{P}}(\mathbf{V}(x))=0$ et $\frac{\partial f}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_k^{(1)},..,\mathbf{A}_k^{(n-1)},\mathbf{A}_k^{(n)},\mathbf{A}_k^{(n+1)},..,\mathbf{A}_k^{(N)}) = \mathbb{E}_{\mathbb{P}}(\mathbf{V}(y))$.

Jusque là, on a prouvé que

$\forall \epsilon > 0, \exists k_0 \in \mathbb{N}, \forall k \geq k_0, \|\frac{\partial f}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_k^{(1)},..,\mathbf{A}_k^{(n-1)},\mathbf{A}_k^{(n)},\mathbf{A}_k^{(n+1)},..,\mathbf{A}_k^{(N)})\|_F \leq \beta\epsilon, \forall n$

$\Rightarrow \lim_{k\to\infty}\|\frac{\partial f}{\partial \mathbf{A}^{(n)}}(\mathbf{A}_k^{(1)},..,\mathbf{A}_k^{(n-1)},\mathbf{A}_k^{(n)},\mathbf{A}_k^{(n+1)},..,\mathbf{A}_k^{(N)})\|_F = 0$.

Cela signifie que plus $k$ croît, plus la suite $N-uplet\left\{\mathbf{A}_k^{(1)},..,\mathbf{A}_k^{(N)}\right\}$ devient proche de l'ensemble des points stationnaires de la fonction $\left\{\mathbf{A}^{(1)},..,\mathbf{A}^{(N)}\right\} \in \mathbb{R}^{I_1\times J_1}.. \times \mathbb{R}^{I_N\times J_N} \to f(\mathbf{A}^{(1)},..,\mathbf{A}^{(N)})$ : ce qui conclut la démonstration.

# *Divers résultats pour l'analyse de OTL.*

**Théorème I** : *De Bonnan et Shapiro*

Soit $f : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$. Supposons que pour tout $\mathbf{x} \in \mathbb{R}^p$, la fonction $\mathbf{u} \to f(x,u)$ est differentiable, et que $f$ et $\partial_2 f(\mathbf{x},u)$, la dérivée de la fonction $\mathbf{u} \to f(\mathbf{x},\mathbf{u})$ sont continues sur $\mathbb{R}^p \times \mathbb{R}^q$. Soit $\nu(u)$ la valeur optimale avec $\nu(\mathbf{u}) = \min_{\mathbf{x}\in C} f(\mathbf{x},\mathbf{u})$, où $C$ représente un sous-ensemble compact de $\mathbb{R}^p$. Alors, la fonction $\nu(\mathbf{u})$ est différentiable.

Par ailleurs, si $\mathbf{u}_0 \in \mathbb{R}^q$, $f(.,\mathbf{u}_0)$ a un unique minimiseur $\mathbf{x}_0$, alors $\nu(u)$ est différentiable en $\mathbf{u}_0$ et $\partial\nu(\mathbf{u}_0) = \partial_2 f(\mathbf{x}_0,\mathbf{u}_0)$

**Theorème II** (voir *Lemma 8* dans [86])

Soient $a_n, b_n$ deux séquences de nombres réels telles que $\forall n$ $a_n \geq 0$,$b_n \geq 0$, $\sum_{n=1}^{\infty} a_n = \infty, \sum_{n=1}^{\infty} a_n b_n < \infty, \exists K > 0$ s.t $|b_{n+1}-b_n| \leq K a_n$. Alors, $\lim_{n\to\infty} b_n = 0$.

**Theorème III** (voir *Theorème 6* dans [86])

Soit $(\mathcal{W},\mathcal{F},\mathcal{P})$ un espace de probabilité. , $\{u_t, t \geq 0\}$ un processus stochastique et $\mathcal{F}_t$ déterminé par l'information jusqu'à la date $t$. Soit $\delta_t$ la suite définie par : $\delta_t = 1$ si $\mathbb{E}(u_{t+1}-u_t|\mathcal{F}_t) > 0$ sinon. Si pour tout $t$, $u_t \geq 0$ et $\sum_{t=1}^{\infty}\mathbb{E}(\delta_t(u_{t+1}-u_t)) < \infty$. Alors, $u_t$ est une quasi-martingale et converge presque-sûrement. En plus, $\sum_{t=1}^{\infty}\|\mathbb{E}(u_{t+1}-u_t|\mathcal{F}_t)\| < \infty$ presque sûrement.

**Théorème IV**(voir Van der Vaart, 1998, chap. 19.2, lemme 19.36 et exemple 19.7)Soit

$F = \{f_\theta : \beta \to \mathbb{R}, \theta \in \Theta\}$ un ensemble de fonctions mesurables indexé par un sous-ensemble borné $\Theta \subset \mathbb{R}^d$. Supposons qu'il existe une constante telle que :
$|f_{\theta_1}(x) - f_{\theta_2}(x)| \leq K|\theta_1 - \theta_2|$ pour tout $\theta_1, \theta_2 \in \Theta, x \in \beta$. Donc, $F$ est P-Donsker (voir Van der Vaart, 1998, chapitre 19.2).
Pour toute fonction dans $F$, définissons $\mathbb{P}_n f, \mathbb{P}_f$ et $\mathbb{G}_n(f)$ telles que : $\mathbb{P}_n f = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}_i)$, $\mathbb{P}f = \mathbb{E}_{\mathbf{X}}\left(f(\mathbf{X})\right), \mathbb{G}_n f = \sqrt{n}\left(\mathbb{P}_n f - \mathbb{P}f\right)$
Supposons aussi que pour toute fonction $f$, $\mathbf{P}f^2 \leq \delta^2$ et $\|f\|_\infty \leq M$ et les variables $\mathbf{X}_1, ..$ sont mesurables. Alors, on a : $\mathbb{E}_P\left(\|\mathbb{G}_n\|_F\right) = O(1), \|\mathbb{G}_n f\|_F = \sup_{f\in F}|\mathbb{G}_n f|$
**Théorème V** : de Wald (1949) et Le Cam (1953)
Soit $\mathcal{F} = \{f : \Omega \times \mathbb{T} \to \mathbb{R}\}$ où les fonctions $f : \Omega \times \mathbb{T} \to \mathbb{R}$, sont continues en $t$ presque sûrement par rapport à une probabilité $\mathbb{P}$ (définie sur $\Omega$).
Supposons que $\mathbb{T}$ est compact l'enveloppe $F$ de la fonction $f$ définie par $F(x) = \sup_{t\in\mathbb{T}} | f(x,t) |$ satisfaisant $\mathbb{P}(F) = \mathbb{E}_{\mathbb{P}}(F(\mathcal{X})) < \infty, \mathcal{X} \sim \mathbb{P}$. Alors, $N_{[]}(\epsilon, \mathcal{F}, L_1(\mathbb{P})) < \infty, \forall \epsilon > 0$ (par conséquent, $\mathcal{F}$ est $\mathbb{P}-$Glivenko-Cantelli.)

# ANNEXE 3. ANALYSE DE CONVERGENCE POUR SINGLESHOT

## 1 Diverses propriétés sur les tenseurs nécessaires pour l'analyse

### 1.1 Rappel des notations et définitions

Pour deux tenseurs $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, on note par $\langle , \rangle$ le produit scalaire entre deux tenseurs $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1,\cdots,i_N} \mathcal{X}_{i_1,\cdots,i_N} \mathcal{Y}_{i_1,..,i_N}$. La norme associée à ce produit scalaire est définie par $\|\mathcal{X}\|_F^2 = \sum_{i_1,\cdots,i_N} \mathcal{X}^2_{i_1,\cdots,i_N}$. Pour $N$ matrices $\left\{\mathbf{A}^{(n)}\right\}_1^N$, on note le produit de **Kronecker** $\mathbf{A}^{(1)} \otimes \cdots \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} \cdots \otimes \mathbf{A}^{(N)}$ par $\underset{m \in \mathbf{I}_N \neq n}{\otimes} \mathbf{A}^{(m)}$, $\otimes$ représentant le produit de **Kronecker** entre deux matrices. Le cardinal d'un ensemble $\theta$ est noté par $\#\theta$. Soit $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ une matrice et $\theta$ le sous-ensemble de $\{1, 2, ..., I_n\}$ (ensemble des entiers consécutifs de $1$ à $I_n$ avec $1$ et $I_n$ inclus). On note par $\mathbf{A}^{(n)}_{\theta,:}$ la matrice construite à partie de $\mathbf{A}^{(n)}$ en empilant dans le sens de la ligne les vecteurs $\left\{\mathbf{A}^{(n)}_{j,:}, j \in \theta\right\}$. Pour un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, on note par $\mathcal{X}^1_{\theta_1} \in \mathbb{R}^{\#\theta_1 \times I_2 \ldots \times I_N}$, un sous-tenseur défini par rapport au premier au premier mode en fixant les premiers indices aux valeurs données par $\theta_1$, $\mathcal{X}^{1,2}_{\theta_1,\theta_2} \in \mathbb{R}^{\#\theta_1 \times \#\theta_2 \times I_3 \ldots \times I_N}$ un sous-tenseur défini par rapport aux premier et second modes en fixant les premiers et second indices aux valeurs données par les deux ensembles $\theta_1$ et $\theta_2$ et ainsi de suite. Par exemple, on considère un tenseur d'ordre 4 $\mathcal{X} \in \mathbb{R}^{10 \times 15 \times 20 \times 30}$ : $\mathcal{X}^1_{\theta_1}, \theta_1 = \{1, 6, 8, 9\}$ est le tenseur dérivé de $\mathcal{X}$ en empilant par rapport au premier mode les tenseurs $\left\{\mathcal{X}^1_{e,:,:,:}, e \in \theta_1\right\}$, $\mathcal{X}^{1,2}_{\theta_1,\theta_2}, \theta_2 = \{1, 2, 4, 5\}$ est le tenseur tel que $\left(\mathcal{X}^{1,2}_{\theta_1,\theta_2}\right)_{i,j,k,l} = \mathcal{X}_{i,j,k,l}$ avec $(i, j) \in \theta_1 \times \theta_2$.

## 1.2 Propriétés sur les sous-tenseurs

L'objectif de cette section est d'introduire des propriétés utiles sur les sous-tenseurs.

La propriété donne une expression du produit tenseur-matrice en fonction de sous-tenseurs.

**Propriété 1 (Produit tenseur-matrice en fonction de sous-tenseurs)** .
*Soit* $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}, \mathcal{G} \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ *deux tenseurs et* $\left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times J_m}, 1 \leq m \leq N\right\}$ *N matrices telles que :*

$$\mathcal{X} = \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}$$

*Le* $n^{\text{ième}}$ *sous-tenseur de* $\mathcal{X}$ *par rapport au mode* $n$*, noté par* $\mathcal{X}^n_{i_n} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times 1 \times I_{n+1} .. \times I_N}$*, est égal à :*

$$\mathcal{X}^n_{i_n} = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}^{(n)}_{i_n,:} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}$$

*En plus, si* $\theta \subset \{1, 2, ....I_n\}$ *(ensemble des entiers consécutifs de* $1$ *à* $I_n$ *avec* $1$ *et* $I_n$ *inclus), on a :*

$$\mathcal{X}^n_\theta = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}^{(n)}_{\theta,:} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)} \in \mathbb{R}^{I_1 .. \times I_{n-1} \times \#\theta \times I_{n+1} .. \times I_N}, \# : \textit{cardinal}$$

**Preuve 1** *L'opérateur de forme matricielle par rapport au mode−n noté* $[\cdot]^{(n)}$ *transforme un tenseur* $\mathbb{R}^{I_1 \times \cdots \times I_N}$ *en une matrice* $\mathbb{R}^{I_n \times \prod_{k \neq n} I_k}$ *suivant un certain ordre. Notons par* $\mathsf{TZ}_n$ *l'opérateur inverse de* $[\cdot]^{(n)}$ *qui transforme une matrice dans* $\mathbb{R}^{I_n \times \prod_{k \neq n} I_k}$ *en un tenseur dans* $\mathbb{R}^{I_1 \times \cdots \times I_N}$*. Puisque* $\mathsf{TZ}_n$ *est l'opérateur inverse de* $[\cdot]^{(n)}$*, par définition de la forme matricielle pour la décomposition de* ***Tucker*** *[68] ainsi que le fait que les lignes de* $\mathbf{AB}$ *sont définies par* $\mathbf{A}_{i,:}\mathbf{B}$*), on a :*

$$\begin{aligned}
\mathcal{X}^n_{i_n} &= \mathsf{TZ}_n\left(\mathbf{X}^{(n)}_{i_n,:}\right) = \mathsf{TZ}_n\left(\mathbf{A}^{(n)}_{i_n,:}\mathbf{G}^{(n)}\left(\otimes_{m \in \mathbf{I}_N \neq n}\mathbf{A}^{(m)}\right)^\top\right) \\
&= \mathsf{TZ}_n\left(\left[\mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}^{(n)}_{i_n,:} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\right]^{(n)}\right) = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}^{(n)}_{i_n,:} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}
\end{aligned}$$

*Avec un raisonnement similaire, la seconde assertion est plus que triviale.*

La propriété suivante exprime la fonction de coût pour la décomposition de **Tucker** en fonction de sous-tenseurs tirés de $\mathcal{X}$ et utilise la propriété 1.

**Propriété 2 (fonction de coût et des sous-tenseurs par rapport à un mode)** .
*Soit* $\mathcal{X} \in \mathbb{R}^{I_1 \times .. \times I_N}$,$\mathcal{G} \in \mathbb{R}^{J_1 \times .. \times J_N}$ *deux tenseurs,* $\left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times J_m}, 1 \leq m \leq N\right\}$ *N matrices. L'égalité suivante est vérifiée :*

$$\|\mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 = \sum_{i_n=1}^{I_n} \|\mathcal{X}_{i_n}^n - \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2.$$

*En plus, si* $\{\theta_m\}_{1 \leq m \leq M}$ *est une partition* $\{1, 2, ..., I_n\}$*, on a :*

$$\|\mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 = \sum_{m=1}^{M} \|\mathcal{X}_{\theta_m}^n - \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}_{\theta_m,:}^{(n)} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2$$

*avec* $\mathcal{X}_{\theta_m}^n \in \mathbb{R}^{I_1 .. I_{n-1} \times \#\theta_m \times I_{n+1} \times ... \times I_N}$ *et* $\mathbf{A}_{\theta_m,:}^{(n)} \in \mathbb{R}^{\#\theta_m \times J_n}$.

**Preuve 2** *Par définition du carré de la norme de Frobenius, on a :*
$\|\mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 = \sum_{i_n=1}^{I_n} \|\mathcal{X}_{i_n}^n - \widehat{\mathcal{X}}_{i_n}^n\|_F^2$ *avec* $\widehat{\mathcal{X}}_{i_n}^n$ *étant le* $i_n^{ième}$ *sous-tenseur du tenseur* $\widehat{\mathcal{X}} = \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}$ *par rapport au mode* $n$
*Par la* Propriété *1, on a :* $\widehat{\mathcal{X}}_{i_n}^n = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}$*, ce qui conclut la démonstration.*
*La seconde assertion est triviale avec la seconde assertion de* Propriété *1.*

La propriété suivante affirme que la fonction de coût peut être exprimée en terme de sous-tenseurs par rapport à tous les modes, la conséquence étant qu'on peut les choisir aussi petits qu'on le désire.

**Propriété 3 (fonction de coût et sous-tenseurs par rapport à plusieurs modes)** *Soit* $\mathcal{X} \in \mathbb{R}^{I_1 \times ... \times I_N}$ *un tenseur et une partition* $\left\{\theta_{m_n}^{(n)}, 1 \leq m_n \leq M_n\right\}$ *de* $\{1, ..I_n\}$*. Donc, on a :*

$$\begin{aligned}\ell(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N) &= \|\mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 \\ &= \sum_{m_1=1}^{M_1} .. \sum_{m_q=1}^{M_q} .. \sum_{m_N=1}^{M_N} \|\mathcal{X}_{\theta_{m_1}^{(1)}, .., \theta_{m_N}^{(N)}} - \mathcal{G} \underset{p \in \mathbf{I}_N}{\times_p} \mathbf{A}_{\theta_{m_p}^{(p)},:}^{(p)}\|_F^2\end{aligned}$$

*avec le sous-tenseur* $\mathcal{X}_{\theta_{m_1}^{(1)}, .., \theta_{m_N}^{(N)}} \in \mathbb{R}^{\#\theta_{m_1}^{(1)} \times \cdots \times \#\theta_{m_N}^{(N)}}$ *dérivé de* $\mathcal{X}$ *dont les entrées sont* $\mathcal{X}_{i_1, \cdots, i_N}, (i_1, \cdots, i_N) \in \theta_{m_1}^{(1)} \times \cdots \times \theta_{m_N}^{(N)}$, $\mathbf{A}_{\theta_{m_p}^{(p)},:}^{(p)} \in \mathbb{R}^{\#\theta_{m_p}^{(p)} \times J_p}$

**Preuve 3** *Par la* Propriété *1 et en regroupant les indices, on a :*

$\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right) = \sum_{m_1=1}^{M_1} \|\mathcal{X}^1_{\theta^{(1)}_{m_1}} - \mathcal{G} \times_1 \mathbf{A}^{(1)}_{\theta^{(1)}_{m_1},:} \underset{m\in I_N^2}{\times_m} \mathbf{A}^{(m)}\|_F^2$. *En appliquant la* Propriété *1 par rapport au second mode au terme* $\|\mathcal{X}^1_{\theta^{(1)}_{m_1}} - \mathcal{G} \times_1 \mathbf{A}^{(1)}_{\theta^{(1)}_{m_1},:} \underset{m\in I_N^2}{\times_m} \mathbf{A}^{(m)}\|_F^2$, *on a :*

$\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right) = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \|\mathcal{X}^{1,2}_{\theta^{(1)}_{m_1},\theta^{(2)}_{m_2}} - \mathcal{G} \times_1 \mathbf{A}^{(1)}_{\theta^{(1)}_{m_1},:} \times_2 \mathbf{A}^{(2)}_{\theta^{(2)}_{m_2},:} \underset{m\in I_N^3}{\times_m} \mathbf{A}^{(m)}\|_F^2$.

*Avec le même raisonnement récursif, on a :*

$\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right) = \sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \dots \sum_{m_N=1}^{M_N} \|\mathcal{X}_{\theta^{(1)}_{m_1},\dots\theta^{(N)}_{m_N}} - \mathcal{G} \times_{q\in I_N} \mathbf{A}^{(q)}_{\theta^{(q)}_{m_q},:}\|_F^2$

## 1.3 Calcul des dérivées

Les deux propriétés 5 et 4 seront utilisées pour le calcul du gradient.

**Propriété 4 (Égalité importante intervenant dans le calcul de la derivée)** *Considérons 2 tensors* $\mathcal{X} \in \mathbb{R}^{I_1\times\dots\times I_N}, \mathcal{G} \in \mathbb{R}^{J_1\times\dots\times J_N}$ *et* $N$ *matrices* $\left\{\mathbf{A}^{(m)}\right\}_{1\le m\le N}$. *L'égalité suivante est vérifiée :*

$$\langle \mathcal{X}, \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle = \langle \mathcal{X} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}, \mathcal{G}\rangle$$

**Preuve 4** *Le produit scalaire de deux tenseurs étant égal au produit scalaire des formes matricielles, on a :*

$$\langle \mathcal{X}, \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle = \langle \mathbf{X}^{(n)}, \mathbf{A}^{(n)}\mathbf{G}^{(n)}\left(\otimes_{m\in\mathbf{I}_N\neq n}\mathbf{A}^{(m)}\right)^\top\rangle = Tr\left(\mathbf{X}^{(n)\top}\mathbf{A}^{(n)}\mathbf{G}^{(n)}\left(\otimes_{m\in\mathbf{I}_N\neq n}\mathbf{A}^{(m)}\right)^\top\right)$$

$$\Rightarrow \langle \mathcal{X}, \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle = Tr\left(\left(\otimes_{m\in\mathbf{I}_N\neq n}\mathbf{A}^{(m)}\right)\mathbf{G}^{(n)\top}\mathbf{A}^{(n)\top}\mathbf{X}^{(n)}\right) = \mathrm{Tr}\left(\mathbf{G}^{(n)\top}\mathbf{A}^{(n)\top}\mathbf{X}^{(n)}\left(\otimes_{m\in\mathbf{I}_N\neq n}\mathbf{A}^{(m)}\right)\right)$$

$$\Rightarrow \langle \mathcal{X}, \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle = Tr\left(\mathbf{G}^{(n)\top}\left[\mathcal{X} \times_{m\in\mathbf{I}_N} \mathbf{A}^{(m)\top}\right]^{(n)}\right) = \langle \mathbf{G}^{(n)}, \left[\mathcal{X} \times_{m\in\mathbf{I}_N} \mathbf{A}^{(m)\top}\right]^{(n)}\rangle$$

$$\Rightarrow \langle \mathcal{X}, \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle = \langle \mathcal{X} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}, \mathcal{G}\rangle$$

**Propriété 5 (Autre égalité intermédiaire pour le calcul du gradient )** *Considérons le tenseur* $\mathcal{G} \in \mathbb{R}^{J_1\times\dots\times J_N}$ *et* $N$ *matrices* $\left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m\times J_m}, 1 \le m \le N\right\}$. *L'égalité suivante est vérifiée :*

$$\left(\mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\right) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top} = \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}\mathbf{A}^{(m)}$$

**Preuve 5** *Il est bien connu que* $m \neq n$, $\mathcal{X} \times_n \mathbf{A} \times_m \mathbf{B} = \mathcal{X} \times_m \mathbf{B} \times_n \mathbf{A}$ *et* $\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} =$

$\mathcal{X} \times_n (\mathbf{BA})$ *(voir [66]). Donc, on a :*

$$\left(\mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\right) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top} = \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top} = \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}\mathbf{A}^{(m)}$$

La prochaine propriété donne l'expression de la dérivée partielle en fonction du tenseur noyau $\mathcal{G}$ ainsi que son caractère lipschitzien.

**Propriété 6 (Dérivée/caractère lipschitzien de la dérivée )** .
*La dérivée de la fonction* $\ell(\mathcal{G}) = \|\mathcal{X} - \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2$ *est donnée par :*

$$\partial_{\mathcal{G}}\ell(\mathcal{G}) = -2\left(\mathcal{X} - \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\right) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}.$$

*Supposons que* $\mathcal{G} \in \mathbb{D}_g$ *et* $\mathbf{A}^{(m)} \in \mathbb{D}_m, 1 \leq m \leq N$ *avec* $\mathbb{D}_g = \left\{\mathcal{G} \in \mathbb{R}^{J_1\times\ldots\times J_N} | \|\mathcal{G}\|_F \leq \alpha\right\}, \mathbb{D}_m = \left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m\times J_m} | \|\mathbf{A}^{(m)}\|_F \leq \alpha\right\}$*. Donc, la dérivée est lipschitzienne avec le paramètre* $2\prod_{m\in\mathbf{I}_N} \|\mathbf{A}^{(m)}\|_F^2$

**Preuve 6** *Dérivée et caractère lipschitzien.*
*1. Justification de l'expression de la dérivée*

$$\begin{aligned}
\ell(\mathcal{G}+\mathcal{H}) &= \|\mathcal{X} - (\mathcal{G}+\mathcal{H}) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 \\
&= \|\mathcal{X} - \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)} - \mathcal{H} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2 \\
&= \|\mathcal{R} - \mathcal{H} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\|_F^2
\end{aligned}$$

*avec* $\mathcal{R} = \mathcal{X} - \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}$ *le tenseur résiduel. Par définition du carré de la norme de Frobenius, on a :*
$\ell(\mathcal{G}+\mathcal{H}) = \ell(\mathcal{G}) - 2\langle\mathcal{R}, \mathcal{H} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\rangle + O(\|\mathcal{H}\|_F^2) = \ell(\mathcal{G}) - 2\langle\mathcal{R} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}, \mathcal{H}\rangle + O(\|\mathcal{H}\|_F^2)$
*(avec* Propriété *4).*
*La dérivée est donc :*

$$\partial_{\mathcal{G}}\ell(\mathcal{G}) = -2\left(\mathcal{X} - \mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}\right) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}$$

*2. Justification du caractère Lipschitzien*
*La dérivée peut être réecrite sous la forme (voir* Propriété *5) :*
$\partial_{\mathcal{G}}\ell(\mathcal{G}) = -2\mathcal{X} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top} + 2\mathcal{G} \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}\mathbf{A}^{(m)}$,
*La norme de la différence des dérivées est donnée par :*
$\|\partial_{\mathcal{G}}\ell(\mathcal{G}_1) - \partial_{\mathcal{G}}\ell(\mathcal{G}_2)\|_F = 2\|(\mathcal{G}_1 - \mathcal{G}_2) \underset{m\in\mathbf{I}_N}{\times_m} \mathbf{A}^{(m)\top}\mathbf{A}^{(m)}\|_F$

$$\Rightarrow \|\partial_{\mathcal{G}}\ell(\mathcal{G}_1)-\partial_{\mathcal{G}}\ell(\mathcal{G}_2)\|_F \leq 2\|\mathcal{G}_1-\mathcal{G}_2\|_F \prod_{m\in\mathbf{I}_N} \|\mathbf{A}^{(m)\top}\mathbf{A}^{(m)}\|_F \leq 2\|\mathcal{G}_1-\mathcal{G}_2\|_F \prod_{m\in\mathbf{I}_N} \|\mathbf{A}^{(m)}\|_F^2$$

On établit l'expression de la dérivée par rapport au tenseur $\mathcal{G}$ en fonction de sous-tenseurs.

**Propriété 7 (Dérivée par rapport à $\mathcal{G}$ en fonction des sous-tenseur )** .
*On considère la fonction $\ell$ définie par :*

$$\ell(\mathcal{G}) = \sum_{i_n=1}^{I_n} \|\mathcal{X}_{i_n}^n - \mathcal{G} \underset{m\in\mathbf{I}_{n-1}}{\times_m} \mathbf{A}^{(m)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{q\in\mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2$$

*La dérivée de $\ell$ par rapport à $\mathcal{G}$ est donnée par :*

$$\partial_{\mathcal{G}}\ell(\mathcal{G}) = -2\sum_{i_n=1}^{I_n} \left(\mathcal{X}_{i_n}^n - \mathcal{G} \underset{m\in\mathbf{I}_{N\neq n}}{\times_m} \mathbf{A}^{(m)} \times_n \mathbf{A}_{i_n,:}^{(n)}\right) \underset{m\in\mathbf{I}_{N\neq n}}{\times_m} \mathbf{A}^{(m)\top} \times_n \left(\mathbf{A}_{i_n,:}^{(n)}\right)^\top$$

**Preuve 7** *C'est une conséquence directe de* Propriété *6.*

La propriété 8 prouve le caractère lipschitzien de la dérivée par rapport à une matrice de saturation.

**Propriété 8 (Dérivée/caractère lipschitzien )** .
*Considérons la fonction $\ell(\mathbf{A}^{(n)}) = \|\mathcal{X} - \mathcal{G}\times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}\|_F^2$. La dérivée de $\ell(\cdot)$ est donné par :*

$$\partial\ell(\mathbf{A}^{(n)}) = -2\left(\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\right)\mathbf{B}^{(n)\top}$$

*avec $\mathbf{X}^{(n)}$,$\mathbf{G}^{(n)}$,$\mathbf{B}^{(n)}$ étant la forme matricielle par rapport au mode $n$ des tenseurs $\mathcal{X}$,$\mathcal{G}$, et $\mathcal{B}$ défini par :*

$$\mathcal{B} = \mathcal{G} \underset{p\in\mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{Id} \underset{q\in\mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)},$$

*avec $\mathbf{Id} \in \mathbb{R}^{J_n\times J_n}$ étant la matrice identité*
*Supposons que $\mathcal{G} \in \mathbb{D}_g$ et $\mathbf{A}^{(m)} \in \mathbb{D}_m, 1 \leq m \leq N$ avec $\mathbb{D}_g = \left\{\mathcal{G}\in\mathbb{R}^{J_1\times..\times J_N} | \|\mathcal{G}\|_F \leq \alpha\right\}, \mathbb{D}_m = \left\{\mathbf{A}^{(m)}\in\mathbb{R}^{I_m\times J_m} | \|\mathbf{A}^{(m)}\|_F \leq \alpha\right\}$. La dérivée $\partial g$ est lipschitzienne avec la borne $2\|\mathcal{G}\|_F^2 \prod_{m\in I_{N\neq n}} \|\mathbf{A}^{(m)}\|_F^2$*

**Preuve 8** *:Dérivée et caractère lipschitzien.*
*1. Expression de la dérivée*

*En introduisant les tenseurs* $\mathcal{R} = \mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)}$ *et* $\mathcal{B} = \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{Id} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}, \mathbf{Id} \in \mathbb{R}^{J_n \times J_n}$*, on a :*

$\ell(\mathbf{A}^{(n)} + \mathbf{H}) = \|\mathcal{X} - \mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)} - \mathcal{G} \underset{m \in \mathbf{I}_{N \neq n}}{\times_m} \mathbf{A}^{(m)} \times_n \mathbf{H}\|_F^2 = \|\mathcal{R} - \mathcal{B} \times_n \mathbf{H}\|_F^2$

$\Rightarrow \ell(\mathbf{A}^{(n)} + \mathbf{H}) = \|\mathcal{R}\|_F^2 - 2\left\langle \mathbf{R}^{(n)}, \mathbf{H}\mathbf{B}^{(n)} \right\rangle + \|\mathbf{H}\mathbf{B}^{(n)}\|_F^2$

$\Rightarrow \ell(\mathbf{A}^{(n)} + \mathbf{H}) = \ell(\mathbf{A}^{(n)}) - 2\left\langle \mathbf{R}^{(n)}\mathbf{B}^{(n)\top}, \mathbf{H} \right\rangle + \mathcal{O}\left(\|\mathbf{H}\|_F^2\right)$

*En conséquence et étant donné que* $\mathcal{G} \underset{m \in \mathbf{I}_N}{\times_m} \mathbf{A}^{(m)} = \mathcal{B} \times_n \mathbf{A}^{(n)}$*, la dérivée est donnée par :*

$$\partial \ell(\mathbf{A}^{(n)}) = -2\mathbf{R}^{(n)}\mathbf{B}^{(n)\top} = -2\left(\mathbf{X}^{(n)} - \mathbf{A}^{(n)}\mathbf{B}^{(n)}\right)\mathbf{B}^{(n)\top}$$

*2. Caractère lipschitzien de la dérivée*

$\|\partial \ell(\mathbf{A}_1^{(n)}) - \partial \ell(\mathbf{A}_2^{(n)})\|_F = 2\|\left(\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\right)\mathbf{B}^{(n)}\mathbf{B}^{(n)\top}\|_F \leq 2\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F\|\mathbf{B}^{(n)}\|_F^2$

$\|\mathbf{B}^{(n)}\|_F = \|\mathbf{G}^{(n)} \underset{m \in \mathbf{I}_{N \neq n}}{\otimes} \mathbf{A}^{(m)}\|_F \leq \|\mathcal{G}\|_F\| \underset{m \in \mathbf{I}_{N \neq m}}{\otimes} \mathbf{A}^{(m)}\|_F = \|\mathcal{G}\|_F \prod_{m \in \mathbf{I}_{N \neq m}} \|\mathbf{A}^{(m)}\|_F$ *(voir par exemple [73], page 433). Donc , on a :*

$\|\partial \ell(\mathbf{A}_1^{(n)}) - \partial \ell(\mathbf{A}_2^{(n)})\|_F \leq 2\|\mathbf{A}_1^{(n)} - \mathbf{A}_2^{(n)}\|_F\|\mathcal{G}\|_F^2 \prod_{m \in \mathbf{I}_{N \neq m}} \|\mathbf{A}^{(m)}\|_F^2$

On détermine, dans la propriété qui suit, l'expression de la dérivée de la fonction de coût par rapport à la matrice de saturation en fonction des sous-tenseurs.

**Propriété 9 (Dérivée par rapport à $\mathbf{A}^{(p)}, p \neq n$ et sous-tenseurs)** .
*Notons* $\ell(\mathbf{A}^{(p)}) = \sum_{i_n=1}^{I_n} \|\mathcal{X}_{i_n}^n - \mathcal{G} \underset{p \in \mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{q \in \mathbf{I}_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2$ *avec* $p < n$ *(*$p > n$ *suit exactement le même principe). La dérivée de* $f$ *est donnée par*

$$\partial \ell(\mathbf{A}^{(p)}) = -2 \sum_{i_n=1}^{I_n} \left(\left(\mathbf{X}_{i_n}^n\right)^{(p)} - \mathbf{A}^{(p)}\left(\mathbf{B}_{i_n}^{(p)}\right)\right)\left(\mathbf{B}_{i_n}^{(p)}\right)^\top$$

*avec* $\left(\mathbf{X}_{i_n}^n\right)^{(p)}$ *et* $\mathbf{B}_{i_n}^{(p)}$ *étant la forme matricielle par rapport au mode−p des tenseurs* $\mathcal{X}_{i_n}^n$ *et* $\mathcal{B}_{i_n}$*, avec* $\mathcal{B}_{i_n}$ *définis par :*

$$\mathcal{B}_{i_n} = \mathcal{G} \underset{m \in \mathbf{I}_{p-1}}{\times_m} \mathbf{A}^{(m)} \times_p \mathbf{Id} \underset{q \in \mathbf{I}_{n-1}^{p+1}}{\times_q} \mathbf{A}^{(q)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{r \in \mathbf{I}_N^{n+1}}{\times_r} \mathbf{A}^{(r)}.$$

**Preuve 9** *c'est une application directe de* Propriété *8.*

L'inégalité introduite par la propriété 10 sert pour le résultat de convergence de **Singleshotinexact**

**Propriété 10 (Inégalité intermédiaire pour *Singleshotinexact* ) .**
*Pour* $\lambda > 0$ *et* $k > k_0 = 1 + \frac{1}{\log(1+\lambda)} \log\left(\frac{1}{\log(1+\lambda)}\right), \frac{1}{(1+\lambda)^{k-1}} \leq \frac{1}{k-k_0}$ *avec* log *étant la fonction logarithmique, c'est-à-dire l'inverse de la fonction exponentielle notée* $e$.

**Preuve 10** *considérons la fonction à une variable définie par* $\ell(x) = x - e^{(x-1)\log(1+\lambda)} - k_0$ *avec* $x > k_0$.
*Sa derivée est donnée par :*
$\ell'(x) = 1 - (\log(1+\lambda))e^{(x-1)\times \ln(1+\lambda)}$.
*Etant donné que* $x > k_0$, *on a par définition de* $k_0$ *:*
$x > 1 + \frac{1}{\log(1+\lambda)} \log(\frac{1}{\log(1+\lambda)})$
$\Rightarrow x - 1 > \frac{1}{\log(1+\lambda)} \log(\frac{1}{\log(1+\lambda)})$
$\Rightarrow (x-1)\log(1+\lambda) \geq \log\left(\frac{1}{\log(1+\lambda)}\right)$ *(because* $\log(1+\lambda) > 0$ *puisque* $\lambda > 0$ *par définition)*
$\Rightarrow e^{(x-1)\log(1+\lambda)} \geq e^{\log\left(\frac{1}{\log(1+\lambda)}\right)}$ *(par définition de la fonction exponentielle* $e$ *qui est une fonction croissante)*
$\Rightarrow e^{(x-1)\log(1+\lambda)} \geq \frac{1}{\log(1+\lambda)}$ *(par définition de* $e$*)*
$\Rightarrow e^{(x-1)\log(1+\lambda)} \log(1+\lambda) \geq 1$ *(car* $\log(1+\lambda) > 0$ *puisque* $\lambda > 0$ *par définition)*
$\Rightarrow 0 \geq 1 - (\log(1+\lambda))\, e^{(x-1)\log(1+\lambda)}$
$\Rightarrow \ell'(x) \leq 0$ *: cela implique que* $\ell$ *est une fonctuon croissante sur le domaine* $x > k_0$.
*Par conséquent, pour un entier* $k > k_0$, *on a :*
$\ell(k) = k - e^{(k-1)\log(1+\lambda)} - k_0 \leq \ell(k_0) = -e^{(k_0-1)\log(1+\lambda)} < 0$
$\Rightarrow 0 < k - k_0 < e^{(k-1)\log(1+\lambda)} = (1+\lambda)^{k-1}$.

# 2 Analyse théorique

## 2.1 Définitions et notations supplémentaires

$$\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right) = \frac{1}{2}\|\mathcal{X} - \mathcal{G} \times_{m\in I_N} \mathbf{A}^{(m)}\|_F^2 \tag{16}$$

$$= \frac{1}{2}\sum_{i_n=1}^{I_n} \|\mathcal{X}_{i_n}^n - \mathcal{G} \underset{p\in\mathbf{I}_{n-1}}{\times_p} \mathbf{A}^{(p)} \times_n \mathbf{A}_{i_n,:}^{(n)} \underset{q\in I_N^{n+1}}{\times_q} \mathbf{A}^{(q)}\|_F^2 \tag{17}$$

On considère pour des raisons de simplicité d'écriture, les notations alternatives suivantes de $\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right)$ données par :

—

$$\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right); \ell\left(\mathcal{G}, \left\{\mathbf{A}^{(p)}\right\}_1^n, \left\{\mathbf{A}^{(q)}\right\}_{n+1}^N\right) \text{ et } \ell\left(\mathcal{G}, \left\{\mathbf{A}^{(p)}\right\}_1^{n-1}, \mathbf{A}^{(n)}, \left\{\mathbf{A}^{(q)}\right\}_{n+1}^N\right).$$

On garde la même notation pour toute fonction des $N+1$ variables $\left\{\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right\}$.

— La valeur minimale de $\ell$ est notée $\ell_{\min} = \ell\left(\mathcal{G}_m, \mathbf{A}_m^{(1)}, .., \mathbf{A}_m^{(N)}\right)$, avec

$$\left(\mathcal{G}_m, \mathbf{A}_m^{(1)}, \cdots, \mathbf{A}_m^{(N)}\right) \leftarrow \underset{\mathbb{D}_g \times \mathbb{D}_1 \times .... \times \mathbb{D}_N}{\arg\min} \ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right).$$

Une telle valeur existe sachant que le produit fini d'ensembles compacts est un ensemble compact et $\ell$ est continue car polynomiale.

— La fonction $\ell$ considérée comme une fonction de la variable $\mathcal{G}$ par $\ell(., \left\{\mathbf{A}^{(m)}\right\}_1^N)$. La même notation est utilisée pour les dérivées.

— Les dérivées partielles de $\ell$ en fonction de $\mathcal{G}$ (respectivement par rapport à $\mathbf{A}^{(p)}$) évalué à $\left(\tilde{\mathcal{G}}, \tilde{\mathbf{A}}^{(1)}, \cdots, \tilde{\mathbf{A}}^{(N)}\right)$ par $\partial_{\mathcal{G}}\ell\left(\tilde{\mathcal{G}}, \left\{\tilde{\mathbf{A}}^{(m)}\right\}_1^N\right)$ (respectivement $\partial_{\mathbf{A}^{(p)}}\ell\left(\tilde{\mathcal{G}}, \left\{\tilde{\mathbf{A}}^{(m)}\right\}_1^N\right)$).

— Les dérivées partielles de $\ell$ est notée par $\partial_x\ell$ ($x$ représentant la variable par rapport on dérive). Le gradient de $\ell$, noté par $\nabla\ell = (\partial_{\mathcal{G}}\ell, \partial_{\mathbf{A}^{(1)}}\ell, ..., \partial_{\mathbf{A}^{(N)}}\ell)$ est un élément de $\mathbb{R}^{J_1 \times \cdots \times J_N} \times \mathbb{R}^{I_1 \times J_1} \times \cdots \times \mathbb{R}^{I_N \times J_N}$ muni de la norme $\|\cdot\|_*$ défini comme la somme des normes de Frobenius des composantes :
$\|\nabla\ell(\tilde{\mathcal{G}}, \left\{\tilde{\mathbf{A}}^{(m)}\right\}_1^N)\|_* = \|\partial_{\mathcal{G}}\ell\left(\tilde{\mathcal{G}}, \left\{\tilde{\mathbf{A}}^{(m)}\right\}\right)\|_F + \sum_{p=1}^N \|\partial_{\mathbf{A}^{(p)}}\ell\left(\tilde{\mathcal{G}}, \left\{\tilde{\mathbf{A}}^{(m)}\right\}\right)\|_F$

On définit les grandeurs suivantes, qui sont toutes bien définies puisque $f$ est continue et le produit fini d'ensembles compacts est une ensemble compact :

— $\Gamma = \sup_{\mathbb{D}_g \times \mathbb{D}_1 \times \cdots \times \mathbb{D}_N} \ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right)$

— $\Gamma_g = \sup_{\mathbb{D}_g \times \mathbb{D}_1 \times .. \times \mathbb{D}_N} \|\partial_{\mathcal{G}}\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}\right)\|_F$

— $\Gamma_m = \sup_{\mathbb{D}_g \times \mathbb{D}_1 \times ... \times \mathbb{D}_N} \|\partial_{\mathbf{A}^{(m)}}\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}\right)\|_F$

Finalement, le nombre maximum d'itérations est noté par $K$.

## 2.2 Définitions pour *Singleshotinexact*

On considère la définition suivante pour $\eta_k^N$ (pas de descente pour $\mathbf{A}^{(N)}$ à l'itération $k+1$)

$$\eta_k^N = \underset{\eta \in [\frac{1}{4K^\gamma}, \frac{1}{K^\gamma}]}{\arg\min} \sigma(\eta)\,\Sigma(\eta) \tag{18}$$

avec :

$$\sigma(\eta) = \eta - \frac{1}{4K^\gamma}$$

$$\Sigma(\eta) = \lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \tilde{\mathbf{A}}^{(N)}(\eta)\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right)$$

$$\ell_j(\mathbf{A}^{(N)}) = \frac{1}{2}\|\mathcal{X}_j^n - \mathcal{G}_{k+1} \underset{m \in I_{n-1}}{\times_m} \mathbf{A}_{k+1}^{(m)} \times_n (\mathbf{A}_{k+1}^{(n)})_{j,:} \underset{q \in I_{N-1}^{n+1}}{\times_q} \mathbf{A}_{k+1}^{(q)} \times_N \mathbf{A}^{(N)}\|_F^2$$

$$\tilde{\mathbf{A}}^{(N)}(\eta) = \mathbf{A}_k^{(N)} - \eta \times \frac{1}{B_k} \sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j$$

avec $\partial_{\mathbf{A}^{(N)}}\ell_j$ représentant la dérivée de $\ell_j$ évaluée $\mathbf{A}_k^{(N)}$, et $\lambda > 0, \gamma > 0$ étant des paramètres à fixer par l'utilisateur.

Ce problème est bien défini pour deux raisons. La première est que les facteurs latents $\left\{\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_{1\leq m\leq N}, \mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_{1\leq m\leq N-1}\right\}$ sont connus à l'étape de mise à jour de $\mathbf{A}^{(N)}$ à l'itération $k+1$ (l'étape de mise à jour de $\mathbf{A}_{k+1}^{(N)}$) puisque les mises à jour des variables sont effectuées dans l'ordre ' mise à jour de $\mathcal{G}$', 'mise à jour de $\mathbf{A}^{(1)}$',..., 'mise à jour de $\mathbf{A}^{(N-1)}$', 'mise à jour $\mathbf{A}^{(N)}$' et on considère le problème de minimisation d'une fonction continue sur une compact.

La seconde est que par *Hypothèse 3.2* présentée en bas, il existe un ensemble $\mathcal{SET}_k$ tel que $\sum_{j\in\mathcal{SET}_k}\partial_{\mathbf{A}^{(N)}}\ell_j \neq 0$ (voir *Proprieté* 11) et l'ensemble que l'on considère est l'un des ensembles vérifiant cette non-nullité. Dans la suite, la matrice $\sum_{j\in\mathcal{SET}_k}\partial_{\mathbf{A}^{(N)}}\ell_j$ sera appelée par **Gradient Inexact**.

## 2.3 Définitions pour *Singleshot*

Pour *Singleshot*, on cosidère les notations suivantes pour les pas de descente $\eta_k^{\mathcal{G}}$ et $\eta_k^p$ à l'itération $k+1$ :

$$\eta_k^{\mathcal{G}} = \underset{\eta\in[\frac{\delta_1}{\sqrt{K}},\frac{\delta_2}{\sqrt{K}}]}{\arg\min}\left(\eta - \frac{\delta_1}{\sqrt{K}}\right)\Sigma_g(\eta) \tag{19}$$

$$\Sigma_g(\eta) = \ell\left(\mathcal{G}_k - \eta\mathcal{D}_k^{\mathcal{G}}, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) \tag{20}$$

$$\eta_k^p = \underset{\eta\in[\frac{\delta_1}{\sqrt{K}},\frac{\delta_2}{\sqrt{K}}]}{\arg\min}\left(\eta - \frac{\delta_1}{\sqrt{K}}\right) \tag{21}$$

$$\Sigma_p(\eta) = \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)} - \eta\mathbf{D}_k^p, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^N\right) - \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right) \tag{22}$$

avec $\delta_1 > 0, \delta_2 > 0$ étant des paramètres à fixer par l'utilisateur. Les problèmes de minimisation définissant $\eta_k^{\mathcal{G}}$ et $\eta_k^p$ sont bien définis puisque les facteurs impliqués dans leurs définitions sont déjà déterminés à l'étape de mise à jour de $\mathcal{G}_{k+1}$ et $\mathbf{A}_{k+1}^{(p)}$ et les problèmes considérés correspondent à des problèmes de minimisation de fonctions continues sur des ensembles compacts.

## 2.4 Hypothèses

L'analyse théorique de **Singleshotinexact** nécessite *Hypothèse 1*, *hypothèse 2*, *Hypothèse 3.1*, *Hypothèse 3.2*, *Hypothèse 3.3* et la définition du pas donnée par l'équation (18). L'analyse théorique de **Singleshot** nécessite *Hypothèse 1*, *Hypothèse 2*, *Hypothèse*

*4* et les problèmes de minimisation définis par les équations (19) et (21). Toutes les hypothèses sont preséntées ci-dessous.

### 2.4.1 Hypothèses communes

**Hypothèse 1 (Bornitude uniforme)** . *Les sous-tenseurs ont uniformément bornés :*

$$\|\mathcal{X}_{i_n}^n\|_F \leq \rho, \forall 1 \leq i_n \leq I_n$$

**Hypothèse 2 (Bornitude des facteurs)** . *On suppose que les facteurs sont bornés :*

$$\mathcal{G} \in \mathbb{D}_g = \left\{\mathcal{G} \in \mathbb{R}^{J_1 \times .. \times J_N} | \|\mathcal{G}\|_F \leq \alpha\right\}, \mathbf{A}^{(m)} \in \mathbb{D}_m = \left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times J_m} | \|\mathbf{A}^{(m)}\|_F \leq \alpha\right\}, 1 \leq m \leq N$$

### 2.4.2 Hypothèses spécifiques pour *Singleshot-inexact*

En plus de *Hypothèse 1* et *Hypothèse 2*, on considère les hypthèses additionnelles suivantes :

**Assumption 3.1 (Borne pour les pas de descente pour $\mathbf{A}^{(N)}$)**

$$\frac{1}{4K^\gamma} < \eta_k^N \leq \frac{1}{\alpha^{2N}}$$

**Assumption 3.2 (Non-nullité de la dérivée partielles par rapport $\mathbf{A}^{(N)}$)**

$$\partial_{\mathbf{A}^{(N)}} \ell \left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) \neq 0$$

**Assumption 3.3 (Choix sur le nombre de sous-tenseur)**

$$I_n \sqrt{\frac{1}{2} + \frac{1}{I_n}} \leq B_k, I_n > 2$$

### 2.4.3 Hypothèses spécifiques pour *Singleshot*

En plus des définitions données par les équations (19) et (21), on considère, en plus de *Hypothèse 1* et *Hypothèse 2*, l'hypothèse additionnelle suivante :

**Hypothèse 4**

$$\frac{\delta_1}{\sqrt{K}} < \eta_k^g, \eta_k^p \leq \frac{\delta_2}{\sqrt{K}}$$

Les trois inégalités reviennent à supposer que les solutions des problèmes de minimisation donnés par les équations (19) et (21) ne sont pas atteintes à la borne inférieure de l'intervalle de minimisation.

# 3 Résultat théorique

L'analyse théorique est motivée par le fait que les résultats de convergence existants pour la *Descente de Gradient par Blocs* ne peuvent être appliqués directement à notre cadre. Une grande partie des approches existantes utilisent la convexité [12] ou la convexité forte [80] de la fonction de coût, ce qui n'est pas vérifié dans notre cas notre problème correspond à un cadre non convexe. Des approches ont été proposées pour le cadre non-convexe, mais avec l'hypothèse de forte convexité pour la fonction vue comme fonction d'une seule de ses variables. Ces raisonnements ne peuvent s'appliquer dans notre cas car le cadre algorithmique est différent. Des approches ont également proposées sans aucune hypothèse de convexité, mais le cadre théorique ne peut être répliqué à notre cadre.
La difficulté du problème vient de l'absence de convexité. Pour contourner cette difficulté, on considère des définitions soigneusement choisies pour les différents pas de descente.

## 3.1 Convergence de *Singleshotinexact*

L'objectif de cette section est de prouver que la suite $\left\{\mathcal{G}_k, \mathbf{A}_k^{(1)}, \cdots, \mathbf{A}_k^{(N)}\right\}$ converge vers l'ensemble des minimiseurs de $\ell$. Avant d'établir ce résultat de convergence, on introduit les résultats préliminaires que sont *Propriété* 11, *Propriété* 12 et *Propriété* 13.

On prouve via la propriété 11 l'existence d'un *gradient inexact* non nul.

**Propriété 11 (Non-nullité du *gradient inexact*)** .
*Sous* Hypothèse 3.2*, la propriété suivante est vérifiée :*
$\exists \mathcal{SET}_k \subset \{1, 2, ...I_n\}, \sum_{j\in\mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j \neq 0$ *avec* $\partial_{\mathbf{A}^{(N)}}\ell_j$ *définie par l'équation* (18)

**Preuve 11** *On fait un raisonnement par contradiction. Supposons qu'on a* $\forall \mathcal{SET}_k \subset \{1, 2, ..., I_n\}, \sum_{j\in\mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j = 0$
*Choisissons* $\mathcal{SET}_k = \{1, 2, .., I_n\}$*. Donc, on a*

$$\sum_{j=1}^{I_n} \partial_{\mathbf{A}^{(N)}}\ell_j = \partial_{\mathbf{A}^{(N)}}\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) = 0,$$

*ce qui est en contradiction avec* Hypothèse 3.2

On établit une borne sur l'écart entre le gradient et le *gradient inexact*.

**Propriété 12 (Borne sur l'écart entre le gradient et le *gradient inexact*)** .
*Considérons l'écart entre le gradient et le* ***Gradient Inexact*** $\beta_k$ *défini par :*
$\beta_k = \frac{1}{B_k} \sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}} \ell_j - \partial_{\mathbf{A}^{(N)}} \ell \left( \mathcal{G}_{k+1}, \left\{ \mathbf{A}_{k+1}^{(m)} \right\}_1^{N-1}, \mathbf{A}_k^{(N)} \right)$ *avec* $\ell_j$ *et* $\partial_{\mathbf{A}^{(N)}} \ell_j$ *défini* (18).
*L'inégalité suivante est vérifiée :*

$$\|\beta_k\|_F^2 \leq 8 I_n^2 (\rho + \alpha^{N+1})^2 \alpha^{2N}$$

**Preuve 12** *Par définition de* $\beta_k$*, on a :*
$\|\beta_k\|_F^2 = \| \frac{1}{B_k} \sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}} \ell_j - \partial_{\mathbf{A}^{(N)}} \ell \left( \mathcal{G}_{k+1}, \left\{ \mathbf{A}_{k+1}^{(m)} \right\}_1^{N-1}, \mathbf{A}_k^{(N)} \right) \|_F^2$
$\Rightarrow \|\beta_k\|_F^2 = \| \left( \frac{1}{B_k} - 1 \right) \sum_{j \in \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}} \ell_j - \sum_{j \notin \mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}} \ell_j \|_F^2$
$\Rightarrow \|\beta_k\|_F^2 \leq \left( \left( 1 - \frac{1}{B_k} \right) \sum_{j \in \mathcal{SET}_k} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F + \sum_{j \notin \mathcal{SET}_k} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F \right)^2$ *car par définition* $\ell$ *contient tous les termes. Le terme de droite est obtenu par application de l'inégalité triangulaire. Comme* $(a+b)^2 \leq 2(a^2+b^2)$ *pour* $a, b \geq 0$ *et par application de l'inégalité de Cauchy-Schwartz, on a :*
$\|\beta_k\|_F^2 \leq 2 \left( 1 - \frac{1}{B_k} \right)^2 B_k \sum_{j \in \mathcal{SET}_k} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F^2 + 2(I_n - B_k) \sum_{j \notin \mathcal{SET}_k} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F^2$
*Étant donné que* $\sum_{j \in \mathcal{SET}_k} = \sum_{1 \leq j \leq I_n} - \sum_{j \notin \mathcal{SET}_k}$*, on a :* $\|\beta_k\|_F^2 \leq 2 \left( 1 - \frac{1}{B_k} \right)^2 B_k \sum_{j=1}^{I_n} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F^2$

$$+2 \underbrace{\left( (I_n - B_k) - \frac{(B_k - 1)^2}{B_k} \right)}_{\theta} \sum_{j \notin \mathcal{SET}_k} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F^2$$

*avec* $\theta = \frac{I_n B_k - B_k^2 - (B_k^2 - 2B_k + 1)}{B_k} \leq \frac{I_n B_k - 2B_k^2 + 2B_k - 1}{B_k} \leq \frac{I_n^2 - 2I_n^2 (\frac{1}{2} + \frac{1}{I_n}) + 2I_n - 1}{B_k} = -\frac{1}{B_k}$ *sous* Hypothèse 3.3.
*Donc, par* Hypothèse 1, Hypothèse 2 *et étant donné que* $B_k \leq I_n$*, on a :*
$\|\beta_k\|_F^2 \leq 2 \left( 1 - \frac{1}{B_k} \right)^2 B_k \sum_{j=1}^{I_n} \|\partial_{\mathbf{A}^{(N)}} \ell_j\|_F^2 \leq 2 \times 4 \times I_n \times I_n (\rho + \alpha^{N+1})^2 \alpha^{2N} \leq 8 I_n^2 (\rho + \alpha^{N+1})^2 \alpha^{2N}$

L'inégalite suivante est utile pour la convergence et utilise principalement le problème de minimisation définie par l'équation (18).

**Propriété 13 (Borne inférieure importante)** .
*L'inégalité suivante est vérifiée :*

$$\lambda \ell \left( \mathcal{G}_{k+1}, \left\{ \mathbf{A}_{k+1}^{(m)} \right\}_1^N \right) - \ell \left( \mathcal{G}_k, \left\{ \mathbf{A}_k^{(m)} \right\}_1^N \right) + \ell \left( \mathcal{G}_{k+1}, \left\{ \mathbf{A}_{k+1}^{(m)} \right\}_1^{N-1}, \mathbf{A}_k^{(N)} \right) \leq 0 \qquad (23)$$

**Preuve 13** *Le problème de minimisation définissat* $\eta_k^N$ *étant bien défini (voir équation*

(18) *et* Propriété *11), on a par définition d'un minimiseur :*

$$\left(\eta_k^N - \frac{1}{4K^\gamma}\right)\left(\lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \tilde{\mathbf{A}}^{(N)}(\eta_k^N)\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right)\right)$$
$$\leq 0 \times \left(\lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \tilde{\mathbf{A}}^{(N)}\left(\frac{1}{4K^\gamma}\right)\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right)\right)$$

*Puisque* $\eta_k^N > \frac{1}{4K^\gamma}$ *par* Hypothèse 3.1*, on a*
$\lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \tilde{\mathbf{A}}^{(N)}(\eta_k^N)\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) \leq 0$
$\Rightarrow \lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N}\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) + \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) \leq 0$ *(*)*
*La dernière implication (*) provient de la définition de* $\eta_k^N$ *car :*
$\tilde{\mathbf{A}}^{(N)}(\eta_k^N) = \mathbf{A}_k^{(N)} - \frac{\eta_k^N}{B_k}\sum_{j\in\mathcal{SET}_k} \partial_{\mathbf{A}^{(N)}}\ell_j = \mathbf{A}_{k+1}^{(N)}$ *avec* $\ell_j$ *définie par l'équation* (18)*)*

Le théorème suivant explique que la suite de facteurs latents $(\mathcal{G}_k, \mathbf{A}_k^{(1)}, .., \mathbf{A}_k^{(N)})$ converge vers un minimiseur de $\ell$. L'idée est de vérifier que la suite $\Delta_k = \ell\left(\mathcal{G}_k, \left\{\mathbf{A}^{(m)}\right\}_k\right) - \ell_{min}$ vérifie une inégalité de la forme $(1+\lambda)\Delta_{k+1} \leq \Delta_k + \frac{\zeta(\rho,\alpha,I_n)}{K^\gamma}$ et un raisonnement par récurrence conclut la démonstration.

**Théorème 1 (Convergence vers un minimiseur de *Singleshotinexact*)** .
*L'inégalité suivante est vérifiée :*

$$\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \ell_{min} \leq \left(\ell\left(\mathcal{G}_1, \left\{\mathbf{A}_1^{(m)}\right\}_1^N\right) - \ell_{min} + \frac{4I_n^2(\rho+\alpha^{N+1})^2\alpha^{2N}}{\lambda}\right) \times \frac{1}{k-k_0} \tag{24}$$
$$\forall k > k_0 = 1 + \frac{1}{\log(1+\lambda)}\log\left(\frac{1}{\log(1+\lambda)}\right)$$

**Preuve 14** *Notons par* $\ell_N$ *la fonction définie par* $\mathbf{A}^{(N)} \to \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}^{(N)}\right) - \ell_{min}$ *et* $\partial\ell_N\left(\mathbf{A}^{(N)}\right)$ *la derivée de* $\ell_N$*. Comme* $\mathbf{A}^{(N)} \to \partial\ell_N(\mathbf{A}^{(N)})$ *est lipschitzienne avec* $\alpha^{2N}$ *as paramètre de lipschitz (*Hypothèse 2 *et* Propriété *8), on a [83] :*

$$\ell_N\left(\mathbf{A}_{k+1}^{(N)}\right) \leq \ell_N\left(\mathbf{A}_k^{(N)}\right) + \langle\partial\ell_N\left(\mathbf{A}_k^{(N)}\right), \mathbf{A}_{k+1}^{(N)} - \mathbf{A}_k^{(N)}\rangle + \frac{\alpha^{2N}}{2}\|\mathbf{A}_{k+1}^{(N)} - \mathbf{A}_k^{(N)}\|_F^2 \tag{25}$$

*Par définition, on a :*
$\mathbf{A}_{k+1}^{(N)} - \mathbf{A}_k^{(N)} = -\eta_k^N\left(\partial f_N(\mathbf{A}_k^{(N)}) + \beta_k\right)$ *avec* $\beta_k = \frac{1}{B_k}\sum_{j\in\mathcal{SET}_k}\partial_{\mathbf{A}^{(N)}}\ell_j - \partial f_N(\mathbf{A}_k^{(N)})$, $\partial_{\mathbf{A}^{(N)}}\ell_j$ *étant la dérivée de la fonction* $\ell_j = \frac{1}{2}\|\mathcal{X}_j^n - \mathcal{G}_{k+1} \underset{p\in\mathbf{I}_{n-1}}{\times_p} \mathbf{A}_{k+1}^{(p)} \times_n \left(\mathbf{A}_{k+1}^{(n)}\right)_{j,:} \underset{q\in\mathbf{I}_{N-1}^{n+1}}{\times_q} \mathbf{A}_{k+1}^{(q)} \times_N \mathbf{A}^{(N)}\|_F^2$ *évaluée à* $\mathbf{A}_k^{(N)}$.

*Cette inégalité combinée avec l'inégalité* (25) *donne (comme* $\eta_k^N > 0$*) :*

$$\frac{\ell_N\left(\mathbf{A}_{k+1}^{(N)}\right) - \ell_N\left(\mathbf{A}_k^{(N)}\right)}{\eta_k^N} \leq -\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right)\|_F^2 - \langle\partial\ell_N\left(\mathbf{A}_k^{(N)}\right), \beta_k\rangle + \frac{\alpha^{2N}\eta_k^N}{2}\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right) + \beta_k\|_F^2$$

*En développant les termes* $\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right) + \beta_k\|_F^2$ *et en réarrangeant les termes, on a :*

$$\frac{\ell_N\left(\mathbf{A}_{k+1}^{(N)}\right) - \ell_N\left(\mathbf{A}_k^{(N)}\right)}{\eta_k^N} \leq \frac{\alpha^{2N}\eta_k^N - 2}{2}\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right)\|_F^2 + \frac{\alpha^{2N}\eta_k^N}{2}\|\beta_k\|_F^2 + (\alpha^{2N}\eta_k^N - 1)\langle\partial\ell_N\left(\mathbf{A}_k^{(N)}\right), \beta_k\rangle \tag{26}$$

*Puisque* $\alpha^{2N}\eta_k^N - 1 \leq 0$ *(par* Hypothèse 3.1*) et étant donné que la valeur absolue d'un réel* $a$ *est égal à* $-a$ *si* $a < 0$*, on a par l'inégalité de Cauchy-Schwartz inequality et l'inégalité* $2ab \leq a^2 + b^2$ *:*

$$(\alpha^{2N}\eta_k^N - 1)\langle\partial\ell_N\left(\mathbf{A}_k^{(N)}\right), \beta_k\rangle \leq (1 - \alpha^{2N}\eta_k^N)\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right)\|_F\|\beta_k\|_F \leq \frac{1 - \alpha^{2N}\eta_k^N}{2}(\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right)\|_F^2 + \|\beta_k\|_F^2) \tag{27}$$

*La combinaison des inégalités* (26) *et* (27) *donne après simplification :*

$$\frac{\ell_N\left(\mathbf{A}_{k+1}^{(N)}\right) - \ell_N\left(\mathbf{A}_k^{(N)}\right)}{\eta_k^N} \leq -\frac{1}{2}\|\partial\ell_N\left(\mathbf{A}_k^{(N)}\right)\| + \frac{1}{2}\|\beta_k\|_F^2 \leq \frac{1}{2}\|\beta_k\|_F^2$$

*Donc, on a :*

$$\ell_N\left(\mathbf{A}_{k+1}^{(N)}\right) \leq \ell_N\left(\mathbf{A}_k^{(N)}\right) + \frac{\eta_k^N}{2}\|\beta_k\|_F^2$$

*Considérons la suite définie par* $\Delta_k$ *définie par :* $\Delta_k = \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \ell_{\min}$*. En replaccant* $\ell_N$ *par son expression, la dernière inégalité donne :*

$$\Delta_{k+1} \leq \Delta_k + \left(\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\right) + \frac{\eta_k^N}{2}\|\beta_k\|_F^2 \tag{28}$$

*Par la* Propriété *13, on a :*

$$\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) \leq -\lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(p)}\right\}_1^N\right)$$

$$\Rightarrow \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N-1}, \mathbf{A}_k^{(N)}\right) - \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) \leq \underbrace{-\lambda\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(p)}\right\}_1^N\right) + \lambda\ell_{\min}}_{-\lambda\Delta_{k+1}}$$

*En combinant la dernière inégalité avec l'inégalité* (28)*, on a :*

$$(1 + \lambda)\Delta_{k+1} \leq \Delta_k + \frac{\eta_k^N}{2}\|\beta_k\|_F^2$$

*Par la* Propriété *12, la dernière inégalité donne :*

$$(1+\lambda)\Delta_{k+1} \leq \Delta_k + \frac{\eta_k^N}{2} \times (8I_n^2(\rho+\alpha^{N+1})^2\alpha^{2N})$$

*Étant donné que* $\eta_k^N \leq \frac{1}{K^\gamma}$*,la dernière inégalité implique :*

$$(1+\lambda)\Delta_{k+1} \leq \Delta_k + \frac{1}{2K^\gamma} \times (8I_n^2(\rho+\alpha^{N+1})^2\alpha^{2N})$$

*En introduisant ,* $\tilde{\epsilon}(\rho,\alpha,I_n) = 8I_n^2(\rho+\alpha^{N+1})^2\alpha^{2N}$*, cette équation récursive donne (par un raisonnment par recurrence) :*
$\Delta_k \leq \frac{\Delta_1}{(1+\lambda)^{k-1}} + \frac{\tilde{\epsilon}(\rho,\alpha,I_n)}{2K^\gamma}\sum_{m=1}^{k-1}\frac{1}{(1+\lambda)^m}$
$\Rightarrow \Delta_k \leq \frac{\Delta_1}{(1+\lambda)^{k-1}} + \frac{\tilde{\epsilon}(\rho,\alpha,I_n)}{2K^\gamma}\frac{1}{\lambda}\left(1-\frac{1}{(1+\lambda)^{k-1}}\right) \leq \frac{\Delta_1}{(1+\lambda)^{k-1}} + \frac{\tilde{\epsilon}(\rho,\alpha,I_n)}{2\lambda K^\gamma}$
*Étant donné que* $k \leq K$ *et* $> 1$*, on a :*

$$\Delta_k \leq \frac{\Delta_1}{(1+\lambda)^{k-1}} + \frac{\tilde{\epsilon}(\rho,\alpha,I_n)}{2\lambda} \times \frac{1}{k} \tag{29}$$

Par *Propriété* 10, on a donc :

$$\forall k > k_0 = 1 + \frac{1}{\log(1+\lambda)}\log\left(\frac{1}{\log(1+\lambda)}\right), \Delta_k \leq \frac{\Delta_1}{k-k_0} + \frac{\tilde{\epsilon}(\rho,\alpha,I_n)}{2\lambda} \times \frac{1}{k-k_0} \tag{30}$$

## 3.2 Convergence de *Singleshot*

L'objectif est d'établir pour *Singleshot* une convergence ergodique avec une vitesse de convergence en $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$. On établit préalablement les Propriétés 14, 15, 17, 16,18.

*Propriété 14* et *Propriété 15* prouvent que la fonction de coût décroît après chaque étape de mise à jour à condition que les pas de descente soient soigneusement définis.

**Propriété 14 (Décroissance de la fonction de coût après la mise à jour de $\mathcal{G}$ )**
.
*Sous l'*Hypothèse 4*, on a :*

$$\ell\left(\mathcal{G}_{k+1}, \left\{A_k^{(m)}\right\}_1^N\right) \leq \ell\left(\mathcal{G}_k, \left\{A_k^{(m)}\right\}_1^N\right)$$

**Preuve 15** *Notons par* $\ell_{\mathcal{G}}^k$ *la fonction* $\ell\left(\cdot, \left\{A_k^{(m)}\right\}_1^N\right) - \ell\left(\mathcal{G}_k, \left\{A_k^{(m)}\right\}_1^N\right)$*, on a par*

*définition* $\eta_k^{\mathcal{G}}$ *(équation* (19)*) et la définition d'un minimiseur :*

$$(\eta_k^{\mathcal{G}} - \frac{\delta_1}{K^{\frac{1}{2}}})\ell_{\mathcal{G}}^k\left(\mathcal{G}_k - \eta_k^{\mathcal{G}}\mathcal{D}_k^{\mathcal{G}}\right) \leq (\frac{\delta_1}{K^{\frac{1}{2}}} - \frac{\delta_1}{K^{\frac{1}{2}}})\ell_{\mathcal{G}}^k\left(\mathcal{G}_k - \frac{\delta_1}{K^{\frac{1}{2}}}\mathcal{D}_k^{\mathcal{G}}\right) = 0$$

*Puisque* $\eta_k^{\mathcal{G}} > \frac{\delta_1}{K^{\frac{1}{2}}}$ *par l'*Hypothèse 4*, on a*

$$\ell_{\mathcal{G}}^k\left(\mathcal{G}_k - \eta_k^{\mathcal{G}}\mathcal{D}_k^{\mathcal{G}}\right) \leq 0$$

*Par définition de* $\ell_{\mathcal{G}}^k$ *et étant donné que* $\mathcal{G}_{k+1} = \mathcal{G}_k - \eta_k^{\mathcal{G}}\mathcal{D}_k^{\mathcal{G}}$*, on a :*

$$\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) \leq \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)$$

**Propriété 15 (Décroissance de la fonction de coût après la mise à jour de $\mathbf{A}^{(p)}$)**
.
*Sous l'*Hypothèse 4*, on a :*

$$\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^p, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^N\right) \leq \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right)$$

**Preuve 16** *Le raisonnement est identique à celui de la* Propriété *14.*

Les deux propriétés suivantes utilisent la définition de **Algorithme 7**.

**Propriété 16 (Constrainte implicite sur $\mathbf{A}_{k+1}^{(p)}$)** .
*Sous l'*Hypothèse 4*,* $\mathbf{A}_{k+1}^{(p)}$*appartient à la boule centreée autour de* $\mathbf{A}_k^{(p)}$ *de rayon*$\frac{\beta}{K^{\frac{1}{2}}}$.

**Preuve 17** *Par définition, on a :*

$$\|\mathbf{A}_{k+1}^{(p)} - \mathbf{A}_k^{(p)}\|_F = \|\frac{\delta_2}{K^{\frac{1}{2}}}\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right)\|_F \leq \frac{\delta_2}{K^{\frac{1}{2}}}\Gamma_p \leq \frac{\delta_2}{K^{\frac{1}{2}}}\max\left(\Gamma_g, \Gamma_1, ..., \Gamma_N\right)$$

**Propriété 17 (Contrainte implicite sur $\mathcal{G}_{k+1}$)** .
*Sous l'*Hypothèse 4*,* $\mathcal{G}_{k+1}$ *apprtient à la boule centrée autour de* $\mathcal{G}_k$ *dont le rayon est donné par* $\frac{\beta}{K^{\frac{1}{2}}}$.

**Preuve 18** *Avec un raisonnement identique à celui de la* Propriété *16, on a :*

$$\|\mathcal{G}_{k+1} - \mathcal{G}_k\|_F \leq \frac{\delta_2}{K^{\frac{1}{2}}}\Gamma_g \leq \frac{\delta_2}{K^{\frac{1}{2}}}\max(\Gamma_g, \Gamma_1, ..., \Gamma_N)$$

On introduit la propriété 18 qui sera utile pour la preuve de la convergence de **Singleshot**

**Propriété 18 (résultat intermédiaire pour la convergence de Singleshot)** .
*Considérons la fonction* $g_p$ *définie par :*

$$g_p\left(\mathcal{G}, \mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}\right) = \|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}, \mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}\right)\|_F^2, 1 \leq p \leq N \tag{31}$$

*On considère la fonction* $g_p$ *définie sur l'ensemble* $\widehat{\mathbb{D}}_g \times \widehat{\mathbb{D}}_1 \times ... \times \widehat{\mathbb{D}}_N$ *avec* $\widehat{\mathbb{D}}_g = \left\{\mathcal{G} \in \mathbb{R}^{J_1 \times ... \times J_N} | \|\mathcal{G}\|_F < 2\alpha\right\}, \widehat{\mathbb{D}}_m = \left\{\mathbf{A}^{(m)} \in \mathbb{R}^{I_m \times J_m} | \|\mathbf{A}^{(m)}\|_F < 2\alpha\right\}$. *La fonctioon* $g_p$ *est Lipschitzienne (par conséquent, la restriction de* $g_p$ *sur l'ensemble* $\mathbb{D}_g \times \mathbb{D}_1 \times .... \times \mathbb{D}_N$ *est Lipschitzienne puisque* $\mathbb{D}_g \times \mathbb{D}_1 \times .... \times \mathbb{D}_N \subset \widehat{\mathbb{D}}_g \times \widehat{\mathbb{D}}_1 \times ... \times \widehat{\mathbb{D}}_N$ *)*

**Preuve 19** *en remarquant que la dérivée de* $g_p$ *est bornée sur* $\widehat{\mathbb{D}}_g \times \widehat{\mathbb{D}}_1 \times ... \times \widehat{\mathbb{D}}_N$ *et étant donné que le produit fini d'ensembles convexes (respectivement ouverts) est un ensemble convexe (respectivement ouvert), le* ***corollaire 2.31*** *dans [97] donne le caractère Lipschitzien de la fonction* $g_p$.

Les trois propriétés permettent de borner les dérivées partielles par rapport au tenseur noyau et aux matrices de saturation.

**Propriété 19 (Moyenne de la dérivée partielle par rapport à $\mathcal{G}$)** .

$$\forall K \geq 1, \frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2 \leq \frac{\frac{\delta_2}{\delta_1^2}(2\Gamma + \alpha^{2N}\Gamma_g^2\delta_2^2)}{K^{\frac{1}{2}}} \tag{32}$$

**Preuve 20** *Par Hypothèse 2 (bornitude des facteurs) et* Propriété *6 (caractère Lipschitzien de la dérivée), on a :*

$$\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_1, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right) - \partial_{\mathcal{G}}\ell\left(\mathcal{G}_2, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F \leq \|\mathcal{G}_1 - \mathcal{G}_2\|_F \prod_{m \in I_N}\|\mathbf{A}_k^{(m)}\|_F^2 \leq \alpha^{2N}\|\mathcal{G}_1 - \mathcal{G}_2\|_F$$

*Par conséquent,* $\partial_{\mathcal{G}}\ell\left(., \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)$ *est Lipschitzienne, ce qui conduit à l'inégalité [83] :*

$$\ell\left(\mathcal{G}_{k+1},\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\leq\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)+\langle\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right),\mathcal{G}_{k+1}-\mathcal{G}_k\rangle+\frac{\alpha^{2N}}{2}\|\mathcal{G}_{k+1}-\mathcal{G}_k\|_F^2$$

*Puisque* $\mathcal{G}_{k+1}=\mathcal{G}_k-\eta_k^{\mathcal{G}}\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k;\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)$ *par la définition de* $\Gamma_g$ *comme le supremum* $\|\partial_{\mathcal{G}}f\left(\mathcal{G},\{\mathbf{A}^{(m)}\}_1^N\right)\|_F$ *on* $\mathbb{D}_g\times\mathbb{D}_1\times...\times\mathbb{D}_N$*, on a :*

$$\eta_k^{\mathcal{G}}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)-\ell\left(\mathcal{G}_{k+1},\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)+\frac{\alpha^{2N}(\eta_k^{\mathcal{G}})^2}{2}\Gamma_g^2\quad(33)$$

*Par les* Propriétés *14 et 15, on a :* $\ell\left(\mathcal{G}_{k+1},\left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^N\right)\leq\ell\left(\mathcal{G}_{k+1},\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)$*. Cette inégalité combinée avec l'inégalité* (33) *donne :*

$$\eta_k^{\mathcal{G}}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)-\ell\left(\mathcal{G}_{k+1},\left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^N\right)+\frac{\alpha^{2N}\Gamma_g^2}{2}(\eta_k^{\mathcal{G}})^2$$

*En sommant cette inégalité, on :*

$$\sum_{k=0}^{K-1}\eta_k^{\mathcal{G}}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq\ell(\mathcal{G}_0,\left\{\mathbf{A}_0^{(m)}\right\}_1^N)-\ell\left(\mathcal{G}_K,\left\{\mathbf{A}_K^{(m)}\right\}_1^N\right)+\frac{\alpha^{2N}\Gamma_g^2}{2}\sum_{k=0}^{K-1}(\eta_k^{\mathcal{G}})^2$$

$$\Rightarrow\sum_{k=0}^{K-1}\eta_k^{\mathcal{G}}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq2\Gamma+\frac{\alpha^{2N}\Gamma_g^2}{2}\sum_{k=0}^{K-1}(\eta_k^{\mathcal{G}})^2\textit{(car }\Gamma\textit{ est le supremum de }f\textit{)}$$

*Puisque* $\frac{\delta_1}{K^{\frac{1}{2}}}<\eta_{g,k}\leq\frac{\delta_2}{K^{\frac{1}{2}}}$*, on a :*

$$\frac{1}{\sum_{k=0}^{K-1}\eta_k^{\mathcal{G}}}\leq\frac{1}{\delta_1K^{1/2}},\qquad\qquad\sum_{k=0}^{K-1}\frac{(\eta_k^{\mathcal{G}})^2}{\sum_{k=0}^{K-1}\eta_k^{\mathcal{G}}}\leq\frac{\delta_2^2}{\delta_1K^{1/2}}$$

*et*

$$\sum_{k=0}^{K-1}\frac{\eta_k^{\mathcal{G}}}{\sum_{k=0}^{K-1}\eta_k^{\mathcal{G}}}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\geq\frac{\delta_1}{\delta_2}\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2$$

*Avec les trois dernières inégalités, on a :*

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq\frac{\delta_2}{\delta_1^2}\frac{\left(2\Gamma+\frac{\alpha^{2N}\Gamma_g^2\delta_2^2}{2}\right)}{K^{\frac{1}{2}}}$$

**Propriété 20 (Moyenne de la dérivée partielle par rapport à $\mathbf{A}^{(p)}$)** .

$$\forall K \geq 1, \frac{1}{K}\sum_{k=0}^{K-1} \|\partial_{\mathbf{A}^{(p)}} \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)}, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right)\|_F^2 \leq \frac{\frac{\delta_2}{\delta_1^2}\left(2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)}{K^{\frac{1}{2}}}$$

**Preuve 21** *Notons* $\ell_k^{(p)}$ *la fonction* $\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \cdot, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right)$.

*Puisque le raisonnement est identque au précédent (*Propriété *19) jusqu'à la équation* (33) *et étant donné le caractère Lipschitzien de la dérivée par rapport à* $\mathbf{A}^{(m)}$ *avec également un paramètre égal à* $\alpha^{2N}$ *(*Propriété *8 et* Hypothèse 2*), on a :*

$$\eta_k^p \|\partial_{\mathbf{A}^{(p)}} \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)}, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right)\|_F^2 \leq -\ell_k^{(p)}\left(\mathbf{A}_{k+1}^{(p)}\right) + \ell_k^{(p)}\left(\mathbf{A}_k^{(p)}\right) + \frac{\alpha^{2N}\left(\eta_k^p\right)^2\Gamma_p^2}{2} \tag{34}$$

*Par les* Propriétés *14 et 15, on a :*

$$\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N}\right) \leq \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_{k+1}^{(p)}, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right) = \ell_k^{(p)}\left(\mathbf{A}_{k+1}^{(p)}\right),$$

*et*

$$\ell_k^{(p)}\left(\mathbf{A}_k^{(p)}\right) = \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)}, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right) \leq \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^{N}\right).$$

*En insérant les deux inégalités précédentes dans l'équation* (34)*, on a :*

$$\eta_k^p \|\partial_{\mathbf{A}^{(p)}} \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \mathbf{A}_k^{(p)}, \left\{\mathbf{A}_k^{(q)}\right\}_{p+1}^{N}\right)\|_F^2 \leq \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^{N}\right) - \ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{N}\right) + \frac{\alpha^{2N}\Gamma_p^2}{2}(\eta_k^p)^2$$

*De ce point, on effectue un raisonnement identique à celui effectué pour la* Propriété *19 (à partir de l'inégalité* (33)*) et obtient le résultat cherché.*

L'idée de la démonstration de cette propriété est similaire à la démonstration pour le critère séquentiel de continuité d'une fonction.

**Propriété 21 (Borne sur la dérivée par rapport à chaque mode )**

$$\exists K_p \geq 1, \forall K \geq K_p, \frac{1}{K}\sum_{k=0}^{K-1} \|\partial_{\mathbf{A}^{(p)}} \ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^{N}\right)\|_F^2 \leq \frac{\delta_2}{\delta_1^2 K^{\frac{1}{2}}}\left(1 + \left(2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)\right) \tag{35}$$

**Preuve 22** *La fonction* $g_p$ *définie par* $g_p\left(\mathcal{G}, \mathbf{A}^{(1)}, .., \mathbf{A}^{(N)}\right) = \|\partial_{\mathbf{A}^{(p)}} \ell\left(\mathcal{G}, \mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}\right)\|_F^2$ *étant lipschitzienne par la* Propriété *18, elle est uniformément continue. Donc, on a :*

$$\forall \epsilon > 0, \exists \eta > 0, \|x - y\|_* \leq \eta \Rightarrow |\|\partial_{\mathbf{A}^{(p)}} \ell(x)\|_F^2 - \|\partial_{\mathbf{A}^{(p)}} \ell(y)\|_F^2| \leq \epsilon \tag{36}$$

*Considérons* $y = (\mathcal{G}_{k+1}, \mathbf{A}_{k+1}^{(1)}, \cdots, \mathbf{A}_{k+1}^{(p-1)}, \mathbf{A}_k^{(p)}, \cdots, \mathbf{A}_k^{(N)})$ *et* $x = \left(\mathcal{G}_k, \mathbf{A}_k^{(1)}, \cdots, \mathbf{A}_k^{(N)}\right)$
$\|x-y\|_* = \|\mathcal{G}_{k+1}-\mathcal{G}_k\|_F + \|\mathbf{A}_{k+1}^{(1)}-\mathbf{A}_k^{(1)}\|_F + .... + \|\mathbf{A}_{k+1}^{(p-1)}-\mathbf{A}_k^{(p-1)}\|_F \leq \frac{p\beta}{K^{\frac{1}{2}}}$ *(par la* Propriété *17 et 16 )*

*Donc, on a :*

$$\|x - y\|_* \leq \frac{p\beta}{K^{\frac{1}{2}}} \tag{37}$$

*Comme* $\frac{p\beta}{K^{\frac{1}{2}}}$ *converge vers* $0$ *quand* $K \to \infty$, $\exists K_p, \forall K \geq K_p, \|x - y\|_* \leq \frac{p\beta}{K^{\frac{1}{2}}} \leq \eta$. *Par l'inégalité* (36), *on a :*

$\forall \epsilon > 0, \exists K_p, \forall K \geq K_p, |\|\partial_{\mathbf{A}^{(p)}}\ell(x)\|_F^2 - \|\partial_{\mathbf{A}^{(p)}}\ell(y)\|_F^2| \leq \epsilon$

$\Rightarrow \forall \epsilon > 0, \exists K_p, \forall K \geq K_p, \|\partial_{\mathbf{A}^{(p)}}\ell(x)\|_F^2 - \|\partial_{\mathbf{A}^{(p)}}\ell(y)\|_F^2 \leq \epsilon$ *(car* $a \leq |a|$*)*

$\Rightarrow \forall \epsilon > 0, \exists K_p, \forall K \geq K_p, \|\partial_{\mathbf{A}^{(p)}}\ell(x)\|_F^2 \leq \epsilon + \|\partial_{\mathbf{A}^{(p)}}\ell(y)\|_F^2$

$\Rightarrow \forall \epsilon > 0, \exists K_p, \forall K \geq K_p, \|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2 \leq \epsilon + \|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right)\|_F^2$

*La dernière inégalité est valable pour toute valeur de* $k \in \{1, 2, ..K\}$ *à condition que* $K \geq K_p$ *(on a deux indices* $k$ *et* $K$ *qui sont distincts).Donc, en sommant sur* $k$ *(et non* $K$*), on a :*

$\forall \epsilon > 0, \exists K_p \ \forall K \geq K_p, \frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(p)}\right\}_1^N\right)\|_F^2 \leq \epsilon$

$$+\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_{k+1}, \left\{\mathbf{A}_{k+1}^{(m)}\right\}_1^{p-1}, \left\{\mathbf{A}_k^{(q)}\right\}_p^N\right)\|_F^2$$

$\Rightarrow \forall \epsilon > 0, \exists K_p, \forall K \geq K_p, \frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N)\|_F^2 \leq \epsilon + \frac{\frac{\delta_2}{\delta_1^2}(2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2})}{K^{\frac{1}{2}}}$ *(par la* Propriété *20)*

$\Rightarrow \exists K_p \geq 1, \forall K \geq K_p, \frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2 \leq \frac{\frac{\delta_2}{\delta_1^2}}{K^{\frac{1}{2}}} + \frac{\frac{\delta_2}{\delta_1^2}(2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2})}{K^{\frac{1}{2}}}$

*Par conséquent, on a :*

$$\exists K_p \geq 1, \forall K \geq K_p, \frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2 \leq \frac{\delta_2}{\delta_1^2 K^{\frac{1}{2}}}\left(1 + \left(2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)\right) \tag{38}$$

**Théorème 2 (Convergence ergodique de Singleshot)** .

*L'inégalité suivante est vérifiée :*

$$\exists K_0 > 0, \forall K \geq K_0, \frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\ell\left(\mathcal{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2 \leq \frac{(N+1)\Delta}{K^{\frac{1}{2}}} \tag{39}$$

*avec*

$$\Delta = \frac{\delta_2}{\delta_1^2}\left(2\Gamma + \frac{\alpha^{2N}\Gamma_g^2\delta_2^2}{2} + \sum_{p=1}^{N}(1 + 2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2})\right)$$

**Preuve 23** *On établit une borne sur* $\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2$ *pour faire apparaître* $\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2, \frac{1}{K}\sum_{k=1}^{K}\|\partial_{\mathbf{A}^{(1)}}\ell(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N)\|_F^2,\cdots,\frac{1}{K}\sum_{k=1}^{K}\|\partial_{\mathbf{A}^{(N)}}f(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N)\|_F^2$, *qui sont des variables déjà bornées via les* Propriétés *19 et 21.*
*Par la convexité de la fonction* $g(x)=x^2$, *puisque* $\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*$ *est égale à la somme des normes de Frobenius des dérivées partielles évaluées à* $\mathcal{G}_k,\mathbf{A}_k^{(1)},\cdots,\mathbf{A}_k^{(N)}$, *on a :*

$$\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2\leq(N+1)\left(\|\partial_{\mathcal{G}}\ell(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N)\|_F^2+\sum_{p=1}^{N}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\right)$$

$$\Rightarrow\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2\leq\frac{N+1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}f\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2$$

$$+\frac{N+1}{K}\sum_{k=0}^{K-1}\frac{1}{K}\sum_{p=1}^{N}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2$$

*En permutant les deux sommes* $\sum$, *on a :*

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2\leq(N+1)\left(\underbrace{\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N)\|_F^2}_{\text{peut être bornée par la Propriété 19}}\right)\tag{40}$$

$$+\sum_{p=1}^{N}\underbrace{\sum_{k=0}^{K-1}\frac{1}{K}\|\partial_{\mathbf{A}^{(p)}}f\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2}_{\text{peut être borné par la Propriété 21}}$$

*Par* Propriété *21, on a pour chaque mode* $p$ *:*

$$\exists K_p\geq1,\forall K\geq K_p,\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathbf{A}^{(p)}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\leq\frac{\delta_2}{\delta_1^2K^{\frac{1}{2}}}\left(1+\left(2\Gamma+\frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)\right)\tag{41}$$

*Pour* $K\geq\max(K_p,1\leq p\leq N)$, *toutes les inégalités données par* (41) *sont vérifiées pour toutes les valeurs de* $p$. *Par conséquent, l'inégalité* (40) *donne :*
$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2\leq(N+1)\left(\frac{1}{K}\sum_{k=0}^{K-1}\|\partial_{\mathcal{G}}\ell\left(\mathcal{G}_k,\left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_F^2\right)$

$$+(N+1)\sum_{p=1}^{N}\frac{\delta_2}{\delta_1^2K^{\frac{1}{2}}}\left(1+\left(2\Gamma+\frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)\right)$$

$$\Rightarrow \frac{1}{K}\sum_{k=0}^{K-1} \|\nabla \ell\left(\mathfrak{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2 \leq (N+1)\left(\frac{\frac{\delta_2}{\delta_1^2}(2\Gamma + \frac{\alpha^{2N}\Gamma_g^2\delta_2^2}{2})}{K^{\frac{1}{2}}} + \sum_{p=1}^{N}\frac{\delta_2}{\delta_1^2 K^{\frac{1}{2}}}\left(1 + 2\Gamma + \frac{\alpha^{2N}\Gamma_p^2\delta_2^2}{2}\right)\right)$$

*La dernière implication résulte de la* Propriété *19. Finalement, on a :*

$$\forall K \geq \max(K_p, 1 \leq p \leq N), \frac{1}{K}\sum_{k=0}^{K-1} \|\nabla \ell\left(\mathfrak{G}_k, \left\{\mathbf{A}_k^{(m)}\right\}_1^N\right)\|_*^2 \leq \frac{(N+1)\Delta}{K^{\frac{1}{2}}} \tag{42}$$

# ANNEXE 4

Considerons un tenseur d'ordre 3 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Supposons

$$\{1, ..I_1\} = \theta_1^{(1)} \cup \theta_2^{(1)}, \qquad \{1, ..., I_2\} = \theta_1^{(2)} \cup \theta_2^{(2)}, \qquad \{1, ..., I_3\} = \theta_1^{(3)} \cup \theta_2^{(3)},$$

$\theta_1^{(n)} = \left\{1, \cdots, \text{Int}\left(\frac{I_n}{2}\right)\right\}$
$\theta_2^{(n)} = \left\{\text{Int}\left(\frac{I_n}{2}\right) + 1, \cdots, I_n\right\}$
Int(x) : plus grand entier inférieur ou égal à x
On sait que la fonction de coût $\ell$ se réécrit sous la forme suivante (voir *Propriété* 3) :

$$\begin{aligned}\ell\left(\mathcal{G}, \left\{\mathbf{A}^{(m)}\right\}_1^N\right) &= \|\mathcal{X} - \mathcal{G} \times_{m \in I_3} \mathbf{A}^{(m)}\|_F^2 \\ &= \sum_{m_1=1}^{2} \sum_{m_2=1}^{2} \sum_{m_3=1}^{2} \|\mathcal{X}_{\theta_{m_1}^{(1)}, \theta_{m_2}^{(2)}, \theta_{m_3}^{(3)}} - \mathcal{G} \times_1 \mathbf{A}^{(1)}_{\theta_{m_1}^{(1)},:} \times_2 \mathbf{A}^{(2)}_{\theta_{m_2}^{(2)},:} \times_3 \mathbf{A}^{(3)}_{\theta_{m_3}^{(3)},:}\|_F^2\end{aligned}$$

La dérivée de $\ell$ par rapport à $\mathbf{A}^{(1)}$ est égale à $[\partial_{\mathbf{A}^{(1)}_{\theta_1^{(1)},:}} \ell, \partial_{\mathbf{A}^{(1)}_{\theta_2^{(1)},:}} \ell]$ ([,] : étant l'opérateur d'empilement de vecteurs dans le sens de la ligne ).
La dérivée de $\ell$ en fonction de $\mathbf{A}^{(1)}_{\theta_1^{(1)},:}$ nécessite de traiter les sous-tenseurs $\left\{\mathcal{X}_{\theta_1^{(1)}, \theta_{m_2}^{(2)}, \theta_{m_3}^{(3)}}\right\}_{1 \leq m_2, m_3 \leq 2}$.
Le même raisonnement s'applique exactement pour toutes les autres dérivées.
Notre approche appliqué à cette formulation de la fonction de coût de sous-tenseurs par rapport à tous les modes nécessite le traitement de sous-tenseurs de la forme $\mathcal{X}_s \in \mathbb{R}^{\frac{I_1}{2} \times \frac{I_2}{2} \times \frac{I_3}{2}}$.
Plus généralement, pour $\mathcal{X} \in \mathbb{R}^{I_1 \times ... \times I_N}$ avec $I_n = \dot{\bigcup}_{k=1}^{p} \left\{(k-1) \times \text{Int}(\frac{I_n}{p}) + 1, ....k \times \text{Int}(\frac{I_n}{p})\right\}$ (union disjointe), notre approche nécessite le traitement de sous-tenseurs de taille $\frac{I_1}{p} \times .... \times \frac{I_N}{p}$.

# BIBLIOGRAPHIE


[1] P.-A. Absil and al. Optimization algorithms on matrix manifolds. *Princeton University Press, Princeton, NJ, USA*, 2008.

[2] J. Alexander and A. Hirschowitz. Polynomial interpolation in several variables. *J. Algebraic Geom*, 4 :201–222, 1995.

[3] A. De Almeida and al. The constrained block-parafac decomposition. presentation at tricap2006, chania, greece. *Available online at http ://www. telecom.tuc.gr/˜nikos/TRICAP2006main/TRICAP2006 Almeida.pdf*, 2006.

[4] Gaoyun An, Shuai Liu, and Qiuqi Ruan. A sparse neighborhood preserving non-negative tensor factorization algorithm for facial expression recognition. *Pattern Anal. Appl.*, 20(2) :453–471, 2017.

[5] C. A. Andersson and R. Bro. The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52 :1–4, 2000.

[6] Woody Austin, Grey Ballard, and Tamara G. Kolda. Parallel tensor compression for large-scale scientific data. *IEEE International Parallel and Distributed Processing Symposium*, 2016.

[7] B. W. Bader and T. G. Kolda. Algorithm 862 : Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Software*, 32 :635–653, 2006.

[8] B. W. Bader and T. G. Kolda. Efficient matlab computations with sparse and factored tensors. *SIAM J. Sci. Comput.*, 30 :205–231, 2007.

[9] B. W. Bader and T. G. Kolda. Matlab tensor toolbox, version 2.2. *Available at http ://csmr.ca.sandia.gov/˜tgkolda/TensorToolbox/*, 2007.

[10] Grey Ballard, Alicia Klin, and Tamara G. Kolda. Tuckermpi : A parallel c++/mpi software package for large-scale data compression via the tucker tensor decomposition. *arxiv*, 2019.

[11] Muthu Manikandan Baskaran, Benoît Meister, Nicolas Vasilache, and Richard Lethin. Efficient and scalable computations with sparse tensors. *IEEE Conference on High Performance Extreme Computing*, pages 1–6, 2012.

[12] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM J. OPTIM*, 23(4) :2037–2060, 2013.

[13] Stephen Becker and Osman Asif Malik. Low-rank tucker decomposition of large tensors using tensorsketch. *Advances in Neural Information Processing Systems*, pages 10117–10127, 2018.

[14] J. M. F. Ten Berge and N. D. Sidiriopolous. On uniqueness in candecomp/ parafac. *Psychometrika*, 67 :399–409, 2002.

[15] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1) :1–122, 2011.

[16] Cesar F. Caiafa and Andrzej Cichocki. Generalizing the column-row matrix decomposition to multiway arrays. *Linear Algebra and its Applications*, 433 :557–573, 2010.

[17] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35 :283–319, 1970.

[18] J. D. Carroll and J. J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition,. *Psychometrika*, 35 :1970, 283–319.

[19] Jonah Casebeer, Michèle Colomb, and Paris Smaragdis. Deep tensor factorization for spatially-aware scene decomposition. *Arxiv*, abs/1905.01391, 2019.

[20] B. Cattell. Parallel proportional profiles and other principles for determining the choice of factors by rotation. *Psychometrika*, 9 :267–283, 1944.

[21] R. B. Cattell. The three basic factor-analytic research designs—their interrelations and derivatives. *Psych. Bull.*, 49 :499–452, 1952.

[22] Venkatesan T. Chakaravarthy, Jee W. Choi, Douglas J. Joseph, Prakash Murali, Shivmaran S. Pandian, Yogish Sabharwal, and Dheeraj Sreedhar. On optimizing distributed tucker decomposition for sparse tensors. *Proceedings of the 2018 International Conference on Supercomputing*, pages 374–384, 2018.

[23] Maolin Che and Yimin Wei. Randomized algorithms for the approximations of tucker and the tensor train decompositions. *Advances in Computational Mathematics*, pages 1–34, 2018.

[24] Dongjin Choi, Jun-Gi Jang, and Uksong Kang. Fast, accurate, and scalable method for sparse coupled matrix-tensor factorization. *CoRR*, 2017.

[25] Dongjin Choi and Lee Sael. Snect : Scalable network constrained tucker decomposition for integrative multi-platform data analysis. *CoRR*, 2017.

[26] Andrzej Cichocki and Phan Anh-Huy. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. In *IEICE transactions on fundamentals of electronics, communications and computer sciences*, volume 92, pages 708–721, 2009.

[27] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo Mandic. Tensor networks for dimensionality reduction and large-scale optimization : Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9 :249–429, 01 2016.

[28] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo Mandic. Tensor networks for dimensionality reduction and large-scale optimizations. part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9 :249–429, 08 2017.

[29] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Phan Anh-Huy. Tensor Decompositions for Signal Processing Applications : From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2) :145–163, 2015.

[30] Andrzej Cichocki, Qibin Zhao, and Shengli Xie. Efficient nonnegative tucker decompositions : Algorithms and uniqueness. *IEEE Transactions on Image Processing*, 24(12) :4990 – 5003, 2015.

[31] Pierre Comon. Tensors : A brief introduction. *IEEE Signal Processing Magazine*, 31 :44–53, 2014.

[32] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3) :1254–1279, 2008.

[33] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3) :1254–1279, 2008.

[34] Pierre Comon and Mikael sorensen. Tensor diagonalization by orthogonal transforms. *Tehnical report, ISRN I3S-RR-2007-06-FR*, 2007.

[35] Graham Cormode, S. Muthukrishnan, Ke Yi, and Qin Zhang. Continuous sampling from distributed streams. *J. ACM*, 59(2) :10 :1–10 :25, 2012.

[36] Alistarh Dan and al. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, pages 5977–5987, 2018.

[37] Lieven De Lathauwer. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM J. Matrix Anal. Appl.*, 28(3) :642–666, 2006.

[38] Jean F. Delmas, Benjamin Jourdain, and Bernard Lapeyre. *Processus Aléatoires*.

[39] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast monte carlo algorithms for matrices ii : Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36(1) :158–183, 2006.

[40] Petros Drineas and Michael W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear Algebra and its Applications*, 420 :553–571, 2007.

[41] Dóra Erdös and Pauli Miettinen. Walk 'n' merge : A scalable algorithm for boolean tensor factorization. *IEEE 13th International Conference on Data Mining*, pages 1037–1042, 2013.

[42] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9) :2421–2456, 2011.

[43] WALTER GANDER. Algorithms for the qr-decomposition. *Research Report No. 80-02*, 1980.

[44] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. *Trans. Audio, Speech and Lang. Proc.*, 19(7) :2067–2080, 2011.

[45] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[46] Xijing Guo, Sebastian Miron, David Brie, Shihua Zhu, and Xuewen Liao. A can-decomp/parafac perspective on uniqueness of doa estimation using a vector sensor array. *IEEE Transactions on Signal Processing*, 59 :3475–3481, 2011.

[47] Hugo Van Hamme. An on-line nmf model for temporal pattern learning : Theory with application to automatic speech recognition. *LVA/ICA. LNCS*, 7191 :306–313, 2012.

[48] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets : History and context. *TiiS*, 5 :19 :1–19 :19, 2015.

[49] A. Harshman. Foundations of the parafac procedure : Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16 :1–84, 1970.

[50] R. A. Harshman. Foundations of the parafac procedure : Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16 :1–84, 1970.

[51] Tamir Hazan, Simon Polak, and Amnon Shashua. Sparse image coding using a 3d non-negative tensor factorization. *ICCV*, 1 :50–57, 2005.

[52] F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys.*, 6 :164–189, 1927.

[53] F. L. Hitchcock. Multilple invariants and generalized rank of a p-way matrix or tensor. *J. Math. Phys.*, 7 :39–79, 1927.

[54] Paul Honeine. Analyzing sparse dictionaries for online learning with kernels. *IEEE Transactions on Signal Processing*, 63 :6343–6353, 2015.

[55] Paul Honeine. Entropies of overcomplete kernel dictionaries. *Bulletin of Mathematical Sciences and Applications*, 16 :1–19, 2016.

[56] Inah Jeon, Evangelos E. Papalexakis, Uksong Kang, and Christos Faloutsos. Haten2 : Billion-scale tensor decompositions. *2015 IEEE 31st International Conference on Data Engineering*, pages 1047–1058, 2015.

[57] Arie Kapteyn, Heinz Neudecker, and Tom Wansbeek. An approach to n-mode components analysis. *Psychometrika*, 51(2) :269–275, 1986.

[58] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation : n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, 2010.

[59] Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion :a riemannian manifold preconditioning approach. *ICML*, 48 :1012–1021, 2016.

[60] Oguz Kaya. High performance parallel algorithms for tensor decompositions. *Distributed, Parallel,and Cluster Computing. Université de Lyon*, 2017.

[61] Oguz Kaya and Bora Uccar. High performance parallel algorithms for the tucker decomposition of sparse tensors. *2016 45th International Conference on Parallel Processing (ICPP)*, pages 103–112, 2016.

[62] Seung-Jun Kim. Online kernel dictionary learning. *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 103–107, 2015.

[63] Yong-Deok Kim and Seungjin Choi. Nonnegative tucker decomposition. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[64] Andreas Kodewitz. Methods for large volume image analysis applied to early detection of alzheimer's disease by analysis of fdg-pet scans. *Université d'Evry*, 2013.

[65] Tamara G. Kolda. Multilinear operators for higher-order decompositions. *Technical Report SAND2006-2081*, 2006.

[66] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3) :455–500, 2009.

[67] Tamara G. Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 363–372, 2008.

[68] Jean Kossaifi, Yannis Panagakis, Anima Anandkumar, and Maja Pantic. Tensorly : Tensor learning in python. *Journal of Machine Learning Research*, 20(26) :1–6, 2019.

[69] Pieter M. Kroonenberg and Jan de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1) :69–97, 1980.

[70] J. B. Kruskal. Three-way arrays : Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl*, 18 :95–138, 1977.

[71] J. B. Kruskal. Statement of some current results about three-way arrays. *manuscript, ATT Bell Laboratories, Murray Hill, NJ. Available at http ://three-mode.leidenuniv.nl/ pdf/k/kruskal1983.pdf*, 1983.

[72] J. B. Kruskal. Multiway data analysis. chapter Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays, pages 7–18. 1989.

[73] Amy N. Langville and William J. Stewart. The kronecker product and stochastic automata networks. *J. Comput. Appl. Math.*, 167(2) :429–447, 2004.

[74] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4) :1253–1278, 2000.

[75] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4) :1324–1342, 2000.

[76] Dongha Lee, Jaehyung Lee, and Hwanjo Yu. Fast tucker factorization for large-scale tensor completion. *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1098–1103, 2018.

[77] Jungwoo Lee, Dongjin Choi, and Lee Sael. Ctd : Fast, accurate, and interpretable method for static and dynamic tensor decompositions. In *PloS one*, volume 13, 2018.

[78] Na Li. Variants of als on tensor decompositions and applications. *Partial fulfillment of the requirements for the degree of Doctor of Philosophy, Clarkson University*, 2013.

[79] Xiaoshan Li, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 2013.

[80] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Mingyi Hong. On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *J. Mach. Learn. Res.*, 18(1) :6741–6764, 2017.

[81] Xinsheng Li, K. Selccuk Candan, and Maria Luisa Sapino. M2td : Multi-task tensor decomposition for sparse ensemble simulations. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1144–1155, 2018.

[82] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 2737–2745, 2015.

[83] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, pages 2737–2745, 2015.

[84] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 581–588, 2013.

[85] Michael W. Mahoney, Mauro Maggioni, and Petros Drineas. Tensor-cur decompositions for tensor-based data. In *SIGKDD*, pages 327–336, 2006.

[86] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. ICML '09, pages 689–696, 2009.

[87] C. Navasca, L. De Lathauwer, and S. Kindermann. Swamp reducing technique for tensor decomposition. In *Proc. 16th European Signal Processing Conference*, 2008.

[88] Carmeliza Navasca and Deonnia N. Pompey. Random projections for low multilinear rank tensors. In *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, pages 93–106, 2015.

[89] Duy Khuong Nguyen and Tu Bao Ho. Fast parallel randomized algorithm for nonnegative matrix factorization with kl divergence for large sparse datasets. *CoRR*, 2016.

[90] Jinoh Oh, Kijung Shin, Evangelos E. Papalexakis, Christos Faloutsos, and Hwanjo Yu. S-hot : Scalable high-order tucker decomposition. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 761–770, 2017.

[91] Sejoon Oh, Namyong Park, Lee Sael, and Uksong Kang. Scalable tucker factorization for sparse tensors - algorithms and discoveries. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1120–1131, 2018.

[92] Guillaume Olikier. Tensor approximation by block term decomposition. *Dissertation, Supervisors : Pierre-Antoine Absil, Lieven De Lathauwer., Reader : Yurii Nesterov*, 2017.

[93] Guillaume Olikier, P.-A. Absil, and Lieven De Lathauwer. Variable projection applied to block term decomposition of higher-order tensors. *LVA/ICA*, pages 139–148, 2018.

[94] Moonjeong Park, Jun-Gi Jang, and Lee Sael. Vest : Very sparse tucker factorization of large-scale tensors. *ArXiv*, abs/1904.02603, 2019.

[95] Namyong Park, Sejoon Oh, and U Kang. Fast and scalable method for distributed boolean tensor factorization. In *The VLDB Journal*, page 1–26, 2019.

[96] P.Comon, J.M.F. ten Berge, L. De Lathauwer, and J. Castaing. Generic and typical ranks of multi-way arrays. *Linear Algebra and its Applications*, 430(11–12) :2997–3007, 2009.

[97] Jean-Paul Penot. Elements of differential calculus (chapter 2 of the book : Calculus without derivatives). *Springer*, 2013.

[98] Ioakeim Perros, Robert Chen, Richard Vuduc, and J Sun. Sparse hierarchical tucker factorization and its application to healthcare. pages 943–948, 2015.

[99] Anh Huy Phan and Andrzej Cichocki. Extended hals algorithm for nonnegative tucker decomposition and its applications for multiway analysis and classification. *Neurocomput.*, 74(11) :1956–1969, 2011.

[100] Nicholas G. Polson, James G. Scott, and Brandon T. Willard. Proximal algorithms in statistics and machine learning. *Statistical Science*, 30(4) :559–581, 2015.

[101] A Rakotomamonjy. Supervised Representation Learning for Audio Scene Classification . *IEEE/ACM Transactions on ASLP*, 25 :1253–1265, 2017.

[102] Florian Roemer, Giovanni Del Galdo, and Martin Haardt. Tensor-based algorithms for learning multidimensional separable dictionaries. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3963–3967, 2014.

[103] Qiquan Shi, Yiu ming Cheung, and Qibin Zhao. Feature extraction for incomplete data via low-rank tucker decomposition. *ECML PKDD*, pages 564–581, 2017.

[104] Kijung Shin, Lee Sael, and U Kang. Fully scalable methods for distributed tensor factorization. *IEEE Trans. on Knowl. and Data Eng.*, 29(1) :100–113, January 2017.

[105] N. D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of n-way arrays. *J. Chemometrics*, 14 :229–239, 2000.

[106] Nicholas D. Sidiropoulos, Evangelos E. Papalexakis, and Christos Faloutsos. Parallel randomly compressed cubes ( paracomp ) : A scalable distributed architecture for big tensor decomposition. 2014.

[107] Nikos D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65 :3551–3582, 2017.

[108] Shaden Smith and George Karypis. Accelerating the tucker decomposition with compressed sparse tensors. In *Euro-Par*, 2017.

[109] Sara Soltani, Misha E. Kilmer, and Per Christian Hansen. A tensor-based dictionary learning approach to tomographic image reconstruction. *BIT Numerical Mathematics*, 56(4) :1425–1454, 2016.

[110] Andrew Stevens, Yunchen Pu, Yannan Sun, Gregory Spell, and Lawrence Carin. Tensor-dictionary learning with deep kruskal-factor analysis. In *AISTATS*, 2017.

[111] Dennis Sun and Cedric Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6201–6205, 2014.

[112] Jimeng Sun, Spiros Papadimitriou, Ching-Yung Lin, Nan Cao, Mengchen Liu, and Weihong Qian. Multivis : Content-based social network exploration through multiway visual analysis. In *SDM*, 2009.

[113] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos. Incremental tensor analysis : Theory and applications. *TKDD*, 2 :11 :1–11 :37, 2008.

[114] Shengqi Tan, Yanbo Zhang, Ge Wang, Xuanqin Mou, Guohua Cao, Zhifang Wu, and Hengyong Yu. Tensor-based dictionary learning for dynamic tomographic reconstruction. *Physics in Medicine and Biology*, 60 :2803–2818, 2015.

[115] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. In *Mathematical Programming*, volume 117, page 387–423, 2007.

[116] Charalampos E. Tsourakakis. Mach : Fast randomized tensor decompositions. In *SDM*, 2009.

[117] L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. *C.W. Harris (Ed.), Problems in Measuring Change, University of Wisconsin Press*, pages 122–137, 1963.

[118] Thiernithi Variddhisa and Danilo P. Mandic. Online multilinear dictionary learning for sequential compressive sensing. *CoRR, abs/1703.02492,*, 2017.

[119] Nico Vervliet, Otto Debals, and Lieven De Lathauwer. Tensorlab 3.0 — numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. *2016 50th Asilomar Conference on Signals, Systems and Computers*, pages 1733–1738, 2016.

[120] Xuan Vu, Caroline Chaux, Nadège Thirion-Moreau, Sylvain Maire, and Elfrida Mihaela Carstea. A new penalized nonnegative third-order tensor decomposition using a block coordinate proximal gradient approach : Application to 3d fluorescence spectroscopy. *Journal of Chemometrics*, 31(4) :e2859, 2017.

[121] Xuan Thanh Vu, Sylvain Maire, Caroline Chaux, and Nadège Thirion-Moreau. A new stochastic optimization algorithm to decompose large nonnegative tensors. *IEEE Signal Processing Letters*, 22(10) :1713–1717, 2015.

[122] Fei Wu, Xiao-Yuan Jing, Wangmeng Zuo, Ruiping Wang, and Xiaoke Zhu. Discriminant tensor dictionary learning with neighbor uncorrelation for image set based classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3069–3075, 2017.

[123] Jianpeng Xu, Jiayu Zhou, Pang-Ning Tan, Xi Liu, and Lifeng Luo. Wisdom : Weighted incremental spatio-temporal multi-task learning via tensor decomposition. *International Conference on Big Data*, pages 522–531, 2016.

[124] Yangyang Xu. On the convergence of higher-order orthogonal iteration. *Linear and Multilinear Algebra*, pages 2247–2265, 2017.

[125] Kazuyoshi Yoshii, Ryota Tomioka, Daichi Mochihashi, and Masataka Goto. Infinite positive semidefinite tensor factorization for source separation of mixture signals. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, pages 576–584, 2013.

[126] Rose Yu, Dehua Cheng, and Yan Liu. Accelerated online low-rank tensor learning for multivariate spatio-temporal streams. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning*, pages 238–247, 2015.

[127] Rafal Zdunek and Andrzej Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. *Intell. Neuroscience*, 2008 :3 :1–3 :13, 2008.

[128] Rafal Zdunek and Andrzej Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. *Intell. Neuroscience*, 2008 :3 :1–3 :13, 2008.

[129] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian sparse tucker models for dimension reduction and tensor completion. *CoRR*, 2015.

[130] Shandian Zhe, Yuan Qi, Youngja Park, Zenglin Xu, Ian Molloy, and Suresh Chari. Dintucker : Scaling up gaussian process models on large multidimensional arrays. In *AAAI*, 2016.

[131] Shandian Zhe, Zenglin Xu, Xinqi Chu, Yuan Qi, and Youngja Park. Scalable nonparametric multiway data analysis. In *AISTATS*, 2015.

[132] Shandian Zhe, Kai Zhang, Pengyuan Wang, Kuang-chih Lee, Zenglin Xu, Yuan Qi, and Zoubin Gharamani. Distributed flexible nonlinear tensor factorization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 928–936, 2016.

[133] Nan Zheng, Qiudan Li, Shengcai Liao, and Leiming Zhang. Flickr group recommendation based on tensor decomposition. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 737–738, 2010.

[134] Guoxu Zhou and al. Efficient nonnegative tucker decompositions : Algorithms and uniqueness. *IEEE Transactions on Image Processing*, 24(12) :4990–5003, 2015.

[135] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Decomposition of big tensors with low multilinear rank. *CoRR*, 2014.

[136] Syed Zubair and Wenwu Wang. Tensor dictionary learning with sparse tucker decomposition. *International conference on digital signal processing(DSP)*, pages 1–6, 2013.

**Titre** : contributions à la décomposition de données multimodales avec des applications en apprentissage de dictionnaires et la décomposition de tenseurs de grande taille

**Résumé** : dans ce travail, on s'interesse à des outils mathématiques spéciaux appelés tenseurs qui sont formellement définis comme des tableaux multidimensionnels définis sur le produit tensoriel d'espaces vectoriels (chaque espace vectoriel étant muni de son système de coordonnées), le nombre d'espaces vectoriels impliqués dans ce produit étant l'ordre du tenseur. L'intérêt pour les tenseurs est motivé par certains travaux expérimentaux qui ont prouvé, dans divers contextes, que traiter des données multidimensionnelles avec des tenseurs plutôt que des matrices donne un meilleur résultat aussi bien pour des tâches de régression que de classification.

Dans le cadre de la thèse, nous nous sommes focalisés sur une décomposition dite de **Tucker** et avons mis en place une méthode pour l'apprentissage de dictionnaires, une technique pour l'apprentissage en ligne de dictionnaires, une approche pour la décomposition d'un tenseur de grandes tailles et enfin une méthodologie pour la décomposition d'un tenseur qui croît par rapport à tous les modes. De nouveaux résultats théoriques concernant la convergence et la vitesse de convergence sont établis et l'efficacité des algorithmes proposés, reposant soit sur la minimisation alternée, soit sur la descente de gradients par coordonnées , est démontrée sur des problèmes réels

**Mots-clés** : tenseur, apprentissage de dictionnaires, descente de coordonnées, minimisation alternée

**Research topic** : contributions to multimodal data processing with applications to dictionary learning and large-scale decomposition

**Abstract** : in this work, we are interested in special mathematical tools called tensors, that are multidimensional arrays defined on tensor product of some vector spaces, each of which has its own coordinate system and the number of spaces involved in this product is generally referred to as order. The interest for these tools stem from some empirical works (for a range of applications encompassing both classification and regression) that prove the superiority of tensor processing with respect to matrix decomposition techniques.
In this thesis framework, we focused on specific tensor model named **Tucker** and established new approaches for miscellaneous tasks such as dictionary learning, online dictionary learning, large-scale processing as well as the decomposition of a tensor evolving with respect to each of its modes. New theoretical results are established and the efficiency of the different algorithms, which are based either on alternate minimization or coordinate gradient descent, is proven via real-world problems.

**Keywords** : tensor, dictionary learning, coordinate descent, alternate minimization.