AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

DÉCOMPOSITION BOOLÉENNE DES TABLEAUX MULTI-DIMENSIONNELS DE DONNÉES BINAIRES : UNE APPROCHE PAR MODÈLE DE MÉLANGE POST NON-LINÉAIRE








THÈSE

présentée et soutenue publiquement le 14 décembre 2018 pour l'obtention du

Doctorat de l'Université de Lorraine

(mention automatique, traitement du signal et génie informatique)

par

MAMADOU DIOP

Composition du jury

Président :

Frederic HAMELIN	Professeur à l'Université de Lorraine

Rapporteurs :

Stéphane CANU	Professeur à l'Université de Normandie
Rémy BOYER	Professeur à l'Université de Lille

Examinateurs :

Chloé CLAVEL	MdC HDR à Telecom-ParisTech
David BRIE	Professeur à l'Université de Lorraine
Sebastian MIRON	MdC à l'Université de Lorraine

Invités :

Anthony LARUE	Expert Senior datascience chez Thales LAS
Antoine SOULOUMIAC	Ingénieur HDR au CEA-LIST

À Seynabou, Aîcha, Massamba et Nara

Qui attend longtemps au puits finira par y trouver un seau à puiser

Remerciements

Ce travail de thèse a constitué une expérience très enrichissante qui m'a permis de faire mes premiers pas dans le monde de la recherche. Il n'aurait pas été aussi fructueux sans l'aide de plusieurs personnes. Il me sera très difficile de remercier tout le monde car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener cette thèse à son terme.

Je tiens tout d'abord à adresser mes remerciements aux membres du CRAN que j'ai pu côtoyer tout au long de cette thèse. Une agréable équipe de chercheurs, d'enseignants, de doctorants, d'ingénieurs, de post-docs et de personnels avec laquelle j'ai passé des moments inoubliables au sein du CRAN.

Plus particulièrement, je tiens à remercier :

Mon directeur de thèse, **M. David Brie** pour son encadrement, ses conseils et ses recommandations tout au long de ce travail. Ses conseils m'ont permis de mieux cadrer mon travail et surtout de surmonter les moments difficiles dans cette thèse.

Mon co-directeur de thèse du CRAN, **M. Sebastian Miron** pour toute son aide. Je suis ravi d'avoir travaillé en sa compagnie car outre son appui scientifique, il a toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

Mes collègues de bureau : Clémence, Ludivine et Yingying qui m'ont toujours soutenu, conseillé et encouragé durant cette thèse. Je tiens aussi à remercier Eddy et Julien pour leurs conseils et les corrections apportées à cette thèse.

Cette thèse est le fruit d'une coopération continue entre le CRAN et le CEA. Ainsi, mes remerciements vont également au personnel du CEA.

Plus particulièrement, je tiens à remercier :

Mes deux co-encadreurs, **M. Anthony Larue** et **M. Antoine Souloumiac** pour leurs encadrements continus, pour les remarques constructives qu'ils m'ont fournies ainsi que pour leurs précieux conseils durant toute la période de ma thèse. Leurs conseils m'ont été d'une grande aide.

Mme Jenny Gresse pour son aide coté administratif, M. Jérôme GAUTHIER pour ses corrections et toute l'équipe du CEA Metz.

Ma gratitude, mon profond respect et mes remerciements vont à tous les membres du jury : rapporteurs, examinateurs et encadreurs, pour leur attention consacrée à l'égard de mon travail.

Ces remerciements ne peuvent s'achever, sans une pensée pour ma famille particulièrement mes parents et mes soeurs.

Enfin, je remercie ma chère épouse pour son soutien quotidien indéfectible et son enthousiasme contagieux à l'égard de mes travaux comme de la vie en général. Notre couple a grandi en même temps que mon projet scientifique, le premier servant de socle solide à l'épanouissement du second.

Table des matières

Introduction générale **9**

1 Décomposition des tableaux N-Dimensionnels de données binaires **13**
1.1 Décompositions tensorielles 15
1.1.1 Notions sur les tenseurs 15
1.1.2 Décompositions tensorielles 18
1.2 Factorisation des matrices à valeurs binaires 21
1.2.1 Factorisations sur l'ensemble des réels 22
1.2.2 Factorisations booléennes 23
1.2.3 Propriétés algébriques des produits booléens matriciels 24
1.2.4 Rang booléen 25
1.3 Algorithmes de factorisation booléenne de matrices binaires (FBMB) . . . 28
1.3.1 Un panorama rapide des algorithmes de FMB 28
1.3.2 Formulation du problème FBMB 29
1.3.3 Approches par relaxation réelle de la FBMB 30
1.3.4 Algorithmes « gloutons » pour la FBMB 33
1.4 Décomposition CP booléennes 36
1.4.1 Modèle BCP 36
1.4.2 Algorithmes de décomposition CP booléenne 38
1.5 Conclusions 40

2 FBMB par modèle de mélange Post Non-Linéaire (PNL) **43**
2.1 Principe de la FBMB par modèle de mélange post non-linéaire 44
2.2 L'algorithme PNL-PF 45
2.3 Condition d'identifiabilité du modèle FBMB 47
2.4 L'algorithme C-PNL-PF 50
2.5 Simulations Numériques 53
2.5.1 Performances dans le cas exact 53
2.5.2 Performances dans le cas d'un bruit binaire 54
2.6 La FBMB appliquée aux données Netflix 55
2.6.1 Données Netflix 55
2.6.2 Capacité explicative des algorithmes FBMB 58
2.6.3 Capacité de prédiction des préférences des utilisateurs 58
2.7 Conclusion sur la FMBM 63

3 Une approche par modèle de mélange Post-NonLinéaire de la décomposition canonique polyadique booléenne **65**
3.1 Unicité de la décomposition canonique polyadique booléenne 66
3.1.1 *Indépendance booléenne* et *rang booléen* d'une famille de vecteurs binaires 68
3.1.2 Conditions suffisantes d'unicité 70
3.2 Algorithmes de DCPB par modèle de mélange post non-linéaire 75
3.2.1 Modèle de mélange post non-linéaire de la DCPB 75
3.2.2 L'algorithme T-PNL-PF 76
3.2.3 L'algorithme BT-ADMM 78
3.3 Simulations numériques 82
3.4 Application à des données réelles 85
3.4.1 Données 85
3.4.2 Capacité explicative des algorithmes DCPB 86
3.5 Conclusion de la DCPB 92

Conclusion **95**

A Une annexe **107**
A.1 Calcul des règles de mise à jour de $\mathbf{W}$ et $\mathbf{H}$ pour l'algorithme PF 107
A.2 Calcul des règles de mise à jour de $\mathbf{W}$ et $\mathbf{H}$ pour l'algorithme TH 109
A.3 Calcul des règles de mise à jour de $\mathbf{W}$ et $\mathbf{H}$ pour l'algorithme PNL-PF . . 110
A.4 Calcul des règles de mise à jour sur $\tilde{\mathbf{W}}$, $\mathbf{W}$, $\bar{\mathbf{H}}$, $\mathbf{H}$, $\bar{\mathbf{V}}$, $\mathbf{V}$ pour l'algorithme BT-ADMM 112

Résumé **116**

Introduction

Ce travail s'inscrit dans le cadre d'une collaboration entre le Centre de Recherche en Automatique de Nancy (CRAN UMR 7039, Université de Lorraine, CNRS) et le Laboratoire d'Intégration de Systèmes et des Technologies (LIST) du CEA Tech. Il s'inscrit dans le plan de ressourcement technologique du CEA 2015, financé par l'Union Européenne dans le cadre du programme opérationnel FEDER-FSE Lorraine et Massif des Vosges 2014-2020.

Le cadre général de ce travail est l'analyse de tableaux multidimensionnels de données binaires et vise à développer des méthodes de décomposition adaptées à ce type de données. L'objectif de ces décompositions est de permettre d'extraire de ces tableaux de données les structures latentes, tout en tenant compte de la nature booléenne des données.

Les données binaires ne peuvent prendre que deux valeurs : 0 ou 1. Elles peuvent être fournies par des capteurs photoélectriques, détecteurs de proximité, boutons poussoirs, détecteurs de présence, système d'alerte, *etc.* Elle peuvent également correspondre à des attributs discrets [1], tels que l'expression des gènes [2], des sondages d'opinions, des données de connection à des sites internet [3], des symboles en communications numériques [4, 5], *etc.* L'organisation de ces données en tableaux, selon des facteurs spécifiques, pose le problème de leur décomposition : peut-on extraire de ces tableaux de données des structures caractéristiques de comportements spécifiques ?

La décomposition de données réelles organisées en tableaux 2D est un problème classique pour lequel les méthodes les plus courantes sont la décomposition en valeurs singulières (SVD) [6] et la factorisation en matrices non-négatives (NMF) [7, 8]. L'analyse des tableaux multidimensionnels fait, pour sa part, appel aux décompositions tensorielles qui généralisent au cas multidimensionnel les décompositions matricielles. Les tenseurs sont des objets qui apparaissent dans nombre de contextes et applications différentes. Initialement introduites dans le contexte de la psychométrie [9], les décompositions tensorielles ont été appliquées à un grand nombre d'autres problèmes : la chimiométrie [10, 11, 12, 13, 14, 15], le traitement du signal [16, 17, 18, 19, 20], l'algèbre linéaire numérique [21, 22, 23, 24], l'analyse numérique [25, 26, 27], l'exploration de données [28, 29, 30], l'analyse de graphes [31, 32, 33], la neuroscience [26, 34, 35], *etc.*

L'objectif de ce travail est d'étudier comment ces notions liées à la décomposition des tableaux à valeurs réelles peuvent s'adapter au cas des tableaux de données binaires. Le

cas de la décomposition de matrices binaires est celui qui a suscité le plus grand nombre de travaux [36, 37, 38, 39, 40, 41, 42, 43]. Une analyse de ces travaux montre cependant qu'un certain nombre de points reste à approfondir. Un premier point est la notion de rang de décomposition qui montre clairement qu'il n'y a un rang, mais des rangs de décomposition de matrices binaires. Un second aspect est relatif à l'unicité de telles décompositions qui n'a jamais été abordée. Un troisième aspect est relatif au développement d'algorithmes de décomposition des matrices binaires.

Le cas des décompositions tensorielles de tableaux binaires est pour sa part beaucoup moins étudié [44, 45] ; il se pose dans ce cas exactement le même type de questions relatives aux notions de rang, d'unicité et d'algorithmes.

Dans cette thèse, nous nous focalisons sur les décompositions booléennes de matrices et tenseurs binaires et sur la notion de rang booléen associée à ce type de décomposition. Nous développons des résultats relatifs à l'unicité de la décomposition booléenne et proposons un modèle de mélange post non-linéaire qui est équivalent aux décompositions booléennes matricielles et tensorielles. L'intérêt du modèle de mélange post non-linéaire est qu'il permet de développer des algorithmes efficaces pour le problème de l'approximation de rang faible des tableaux multidimensionnels binaires.

Le document de thèse est organisé en 3 chapitres.

Dans le premier chapitre, nous introduisons les principales notions liées aux décompositions tensorielles ainsi que les différents types de décompositions de tableaux à valeurs binaires ainsi que les rangs associés. Un panorama des algorithmes existants pour la décomposition de matrices et tenseurs à valeurs binaires est également dressé.

Dans le deuxième chapitre, nous introduisons l'approche par modèle de mélange Post Non-Linéaire (PNL) de la factorisation booléenne de matrices binaires proposée et développons une approche pour la factorisation des matrices binaires (FMB) s'appuyant sur une mise à jour multiplicative de type NMF. Nous donnons la condition nécessaire et suffisante pour assurer l'unicité de la factorisation booléenne en matrices binaires. Dans le cas où la factorisation n'est pas unique, nous proposons d'inclure un terme de régularisation supplémentaire. Des simulations numériques montrent que les approches proposées fournissent de meilleurs résultats que l'état de l'art. L'intérêt de ces approches est également illustré sur des données réelles.

Dans le troisième chapitre, nous généralisons les résultats du chapitre 2 au cas de la décomposition canonique polyadique booléenne de tableaux multidimensionnels (tenseurs). Nous étudions l'identifiabilité de cette décomposition et donnons la condition nécessaire et suffisante d'identifiabilité. Néanmoins, cette condition est difficilement vérifiable en pratique. C'est pourquoi nous proposons des conditions suffisantes plus facilement vérifiables, fondées sur une nouvelle notion d'indépendence booléenne. Nous généralisons l'algorithme de décomposition introduit au chapitre 2 (s'appuyant sur une mise à jour multiplicative, de type NMF) au cas multidimensionnel et proposons un nouvel algorithme,

plus efficace, fondé sur une stratégie de type AO-ADMM (*Alternating Optimization - ADMM*). Ces algorithmes sont comparés à ceux de l'état de l'art sur des données simulées et sur un jeu de données réelles.

Publications associées au travail de thèse

Articles parus

1. Mamadou Diop, Anthony Larue, Sebastian Miron, and David Brie. Factorisation en matrices binaires par modèle de mélange post non-linéaire. In *XXVIe Colloque GRETSI Traitement du Signal & des Images, GRETSI 2017*, 2017.
2. Mamadou Diop, Anthony Larue, Sebastian Miron, and David Brie. A post-nonlinear mixture model approach to binary matrix factorization. In *25th European Signal Processing Conference, EUSIPCO 2017*, 2017.

Articles soumis

1. Mamadou Diop, Sebastian Miron, Anthony Larue, and David Brie. Boolean decomposition of binary matrices using a post-nonlinear mixture approach. In *IEEE Transactions on Signal Processing*.
2. Mamadou Diop, Sebastian Miron, Antoine Souloumiac, and David Brie. Boolean CP decomposition of binary tensors : uniqueness and algorithm. In *ICASSP 2019, Brighton, UK, 12-17 May*, 2019.

Chapitre 1

Décomposition des tableaux N-Dimensionnels de données binaires

Sommaire

1.1 Décompositions tensorielles **15**
1.1.1 Notions sur les tenseurs 15
Opérations sur les tenseurs 16
1.1.2 Décompositions tensorielles 18
Décomposition de TUCKER 19
Décomposition CP 19
1.2 Factorisation des matrices à valeurs binaires **21**
1.2.1 Factorisations sur l'ensemble des réels 22
1.2.2 Factorisations booléennes 23
1.2.3 Propriétés algébriques des produits booléens matriciels 24
Propriétés par rapport à l'addition 24
Propriétés par rapport à la multiplication 24
Distributivité 25
1.2.4 Rang booléen 25
1.3 Algorithmes de factorisation booléenne de matrices binaires (FBMB) **28**
1.3.1 Un panorama rapide des algorithmes de FMB 28
1.3.2 Formulation du problème FBMB 29
1.3.3 Approches par relaxation réelle de la FBMB 30
L'algorithme ≪ Penalty Function ≫ (PF) 31
L'algorithme ≪ THresholding ≫ (TH) 31
Remarque : 32
1.3.4 Algorithmes ≪ gloutons ≫ pour la FBMB 33
L'algorithme ASSO 34
L'algorithme FC 35
1.4 Décomposition CP booléennes **36**

1.4.1 Modèle BCP 36
1.4.2 Algorithmes de décomposition CP booléenne 38
1.5 Conclusions **40**

1.1 Décompositions tensorielles

1.1.1 Notions sur les tenseurs

L'algèbre linéaire est un outil incontournable en calcul scientifique au centre duquel la notion de matrice permet de représenter la transformation d'un système de coordonnées en un autre. Dans ce cadre, la notion de rang apparaît comme fondamentale car elle permet de représenter une matrice comme la factorisation de matrices de dimension plus faible et correspond à la décomposition d'une matrice en termes de rang 1. Le rang s'interprète comme le nombre minimal de termes de rang 1 permettant de représenter la matrice. Une factorisation définit une relation bilinéaire entre les deux espaces. Les tenseurs ont été introduits à la fin du 19ème siècle avec le développement du calcul différentiel. Ils ont alors été omniprésents en physique pour exprimer des transformations multilinéaires entre espaces indépendamment des systèmes de coordonnées. Un tenseur est donc une application permettant de définir une transformation entre les coordonnées d'espaces linéaires. Dès que la base de représentation est fixée, un tenseur peut être assimilé à un tableau multidimensionnel dont chaque élément est accessible via N indices. C'est l'acceptation généralement adoptée. Une décomposition tensorielle est une représentation multilinéaire de cette transformation. Cependant contrairement au cas matriciel, il n'existe pas une, mais des décompositions tensorielles.

Dans cette partie, nous introduisons les définitions et les opérations relatives aux tenseurs ainsi que les principales notations utilisées dans ce manuscript. La plupart des définitions et notations utilisées sont issues de [50, 51, 52, 53]. Un tenseur est donc un tableau multidimensionnel à N indices dont l'*ordre* est le nombre de dimensions, également appelées *modes*. Ainsi, un tenseur d'ordre N à valeurs réelles est symbolisé par $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$, avec I_n, la dimension du $n^{\text{ième}}$ mode. Un tenseur d'ordre 0 est un *scalaire*, un tenseur du premier ordre est un *vecteur* et un tenseur du second ordre est une *matrice*. Les tenseurs d'ordre supérieur ou égal à 3 sont les *tenseurs d'ordre supérieur*. Un tenseur $\mathcal{X}$ d'ordre 3 de dimension $I_1 \times I_2 \times I_3$ est illustré dans la figure 1.1. Dans ce manuscrit, les scalaires seront notés par des lettres minuscules, par exemple x, les vecteurs par des lettres minuscules en gras, par exemple $\mathbf{x}$, les matrices seront désignées par des lettres majuscules en caractère gras comme $\mathbf{X}$ et les tenseurs d'ordre supérieur seront désignés par les lettres majuscules calligraphiques, par exemple, $\mathcal{X}$. Le $i^{\text{ième}}$ élément du vecteur $\mathbf{x}$ sera désigné par x_i, l'élément (i, j) de la matrice $\mathbf{X}$ par x_{ij} et l'élément $(i_1, i_2, \cdots i_N)$ du tenseur $\mathcal{X}$ d'ordre N sera noté par $x_{i_1 i_2 \cdots i_N}$. La $i^{\text{ième}}$ ligne de la matrice $\mathbf{X}$ est notée par $\mathbf{x}_{i:}$ et la $j^{\text{ième}}$ colonne de la matrice $\mathbf{X}$ est représentée par $\mathbf{x}_j$.

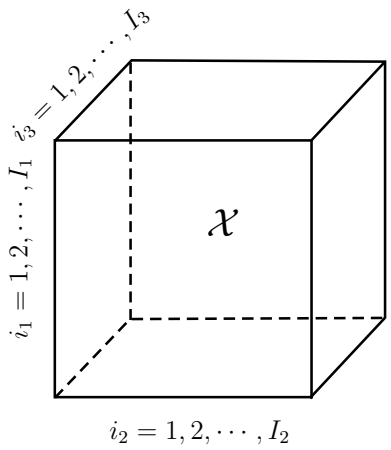


FIGURE 1.1 – Exemple d'un tenseur d'ordre 3, de dimension $I_1 \times I_2 \times I_3$.

Opérations sur les tenseurs

La *norme d'un tenseur* $\mathcal{X}$ d'ordre N ($\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$) est la racine carrée de la somme des carrés de tous ses éléments :

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N}^2}.$$

Un *tenseur d'ordre N est de rang 1*, s'il peut être écrit comme le produit tensoriel de N vecteurs, c'est-à-dire,

$$\mathcal{X} = \mathbf{a}^{\{1\}} \circ \mathbf{a}^{\{2\}} \circ \cdots \circ \mathbf{a}^{\{N\}}, \tag{1.1}$$

avec ($\circ$), le produit tensoriel. Cela signifie que chaque élément du tenseur est le produit des éléments des vecteurs correspondants : $x_{i_1 i_2 \cdots i_N} = a_{i_1}^{\{1\}} a_{i_2}^{\{2\}} \cdots a_{i_N}^{\{N\}}$. Dans le cas matriciel on retrouve la relation, $\mathbf{a}^{\{1\}} \circ \mathbf{a}^{\{2\}} = \mathbf{a}^{\{1\}} \mathbf{a}^{\{2\}T}$.

Le *dépliement* d'un tenseur est le processus qui permet de réordonner ou transformer les éléments d'un tenseur en matrices. Le dépliement d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots I_N}$ suivant le $n^{\text{ième}}$ mode est noté $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_2 \times \cdots I_N}$. Ainsi, dans [50], Kolda et Bader définissent la matricisation ou dépliement du tenseur $\mathcal{X}$ telle que l'élément $(i_1, i_2, \cdots, i_N)$ de $\mathcal{X}$ corresponds à l'élément (i_n, j) de $\mathbf{X}_{(n)}$ avec :

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \qquad \text{avec} \qquad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m.$$

Par exemple, un tenseur de taille $3 \times 4 \times 5$ peut être arrangé ou transformé en une matrice de taille 3×20 ou 4×15 ou encore 5×12 qui correspondent respectivement aux tailles des matrices $\mathbf{X}_{(1)}$, $\mathbf{X}_{(2)}$ et $\mathbf{X}_{(3)}$. La figure 1.2 permet d'illustrer le dépliement d'un tenseur d'ordre 3 de taille $I_1 \times I_2 \times I_3$ suivant les 3 modes.

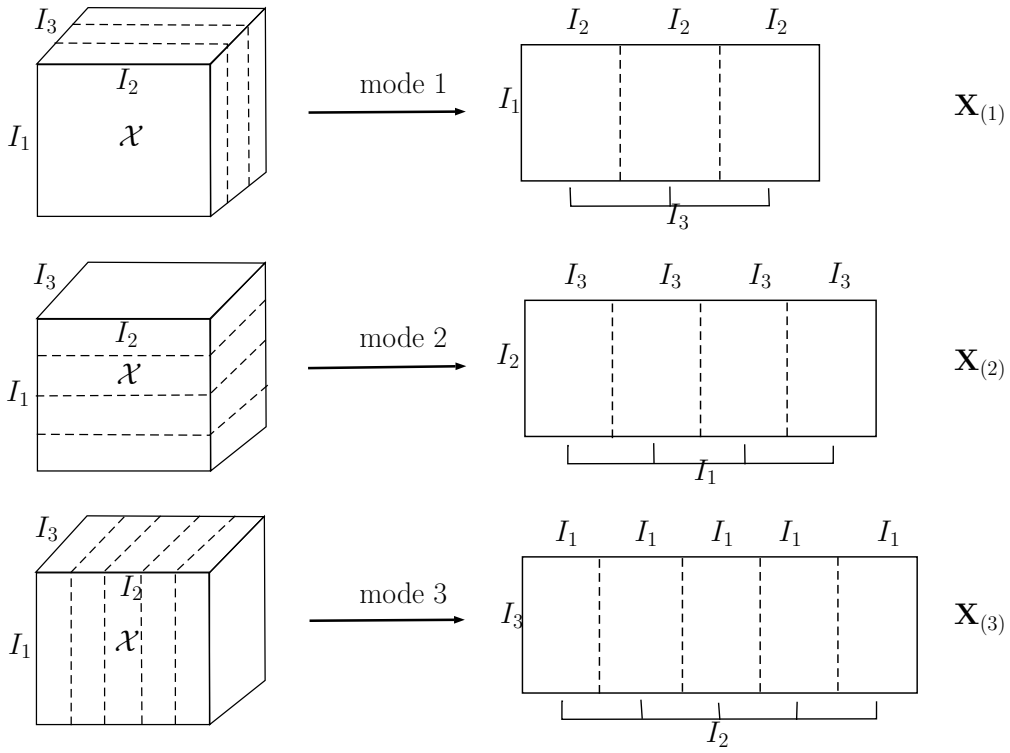


FIGURE 1.2 – Illustration du dépliement du tenseur $\mathcal{X}$ suivant les 3 modes.

Plusieurs produits matriciels sont utilisés pour représenter les opérations sur les tenseurs.

Le produit matriciel classique entre deux matrices $\mathbf{A} \in \mathbb{R}^{I\times K}$ et $\mathbf{B} \in \mathbb{R}^{K\times J}$ défini selon :

$$\mathbf{AB} = \sum_{k=1}^{K} \mathbf{a}_k \otimes \mathbf{b}_k = \sum_{k=1}^{K} \mathbf{a}_k \mathbf{b}_k^T, \tag{1.2}$$

permet de construire une matrice de dimension $I \times J$.

Le produit de Kronecker entre deux matrices $\mathbf{A} \in \mathbb{R}^{I\times J}$ et $\mathbf{B} \in \mathbb{R}^{M\times L}$, représenté par le symbole $(\otimes)$, construit une matrice de taille $(IM) \times (JL)$ selon :

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix} \tag{1.3}$$

$$= [\mathbf{a}_1 \otimes \mathbf{b}_1 \;\; \mathbf{a}_1 \otimes \mathbf{b}_2 \;\; \mathbf{a}_1 \otimes \mathbf{b}_3 \;\; \cdots \mathbf{a}_J \otimes \mathbf{b}_{L-1} \;\; \mathbf{a}_J \otimes \mathbf{b}_L]\,. \tag{1.4}$$

Le produit de Kronecker de deux vecteurs $\mathbf{a}_1 \in \mathbb{R}^I$ et $\mathbf{b}_1 \in \mathbb{R}^J$ donne : $\mathbf{a}_1 \otimes \mathbf{b}_1 = [a_{11}\mathbf{b}_1 \;\; a_{21}\mathbf{b}_1 \;\; \cdots a_{N1}\mathbf{b}_1]^T \in \mathbb{R}^{IJ}$.

Le produit de Khatri-Rao de deux matrices de même nombre de colonnes, représenté par le symbole $(\odot)$, est le produit de Kronecker entre les colonnes de ces matrices. Soient

$\mathbf{A}$ et $\mathbf{B}$ deux matrices de tailles respectives $N \times K$ et $P \times K$, leur produit de Khatri-Rao noté par $\mathbf{A} \odot \mathbf{B}$ de taille $(NP) \times K$ est défini par :

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \;\; \mathbf{a}_F \otimes \mathbf{b}_F \;\; \cdots \;\; \mathbf{a}_K \otimes \mathbf{b}_K]. \tag{1.5}$$

Le produit de Hadamard de deux ou plusieurs matrices/tenseurs de même taille, représenté par $(*)$, est la matrice/tenseur où chaque coefficient et le produit terme à terme de ces matrices/tenseurs. Ainsi dans le cas de deux matrices $\mathbf{A}$ et $\mathbf{B}$ de même taille $(I \times J)$ on a :

$$\mathbf{A} * \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix} \tag{1.6}$$

Le produit n-mode d'un tenseur $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ par une matrice $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ noté par $\mathcal{X} \times_n \mathbf{U}$ est un tenseur $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times J \times I_{n+1} \cdots \times I_N}$ dont les entrées sont données par :

$$y_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

Ce produit peut aussi être exprimé en termes de tenseurs dépliés :

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \quad \Leftrightarrow \quad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}. \tag{1.7}$$

1.1.2 Décompositions tensorielles

Les décompositions tensorielles ont pour origine les travaux de Hitchcock [54]. Elles se sont ensuite développées dans le domaine de la psychométrie où les travaux de Cattell [55] en 1944, Tucker dans les années 60, Carroll et Chang et Harshman [9, 56] dans les années 70 ont permis la construction des premiers modèles et algorithmes. Au cours des années 80, les premières applications en dehors du domaine de la psychométrie se développent dans le domaine de la chimiométrie [14]. Ces dernières années, elles ont été utilisées pour une variété d'applications telles que : le traitement du signal [18], [16], [17], la vision par ordinateur [57], [58], l'analyse numérique [25], [26],[27], l'exploration de données [28], [29], [30], l'analyse de graphes [31], [32], [33], les neurosciences [34], [35], [26].

Plusieurs types de décompositions tensorielles ont été proposés dans la littérature, cependant, les plus populaires sont la décomposition CP (pour Canonique Polyadique ou CANDECOMP-PARAFAC) proposée dans [9, 12, 13, 56] qui apparaît comme un cas particulier de la décomposition de TUCKER proposée dans [59, 60, 61, 62]. Ces décompositions se définissent pour des tenseurs d'ordre N. Pour simplifier la présentation, nous nous limitons aux tenseurs d'ordre 3.

Décomposition de TUCKER

La décomposition de TUCKER consiste à décomposer un tenseur en un tenseur cœur, de dimension réduite, multiplié par une matrice selon chacun des modes. Cette décomposition est généralement utilisée pour l'estimation de sous-espace, la compression et la réduction de dimension. Ainsi, la décomposition de TUCKER de $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, il s'agit de trouver les matrices appelées *facteurs*, $\mathbf{A}^{\{1\}} \in \mathbb{R}^{I_1 \times R_1}$, $\mathbf{A}^{\{2\}} \in \mathbb{R}^{I_2 \times R_2}$, $\mathbf{A}^{\{3\}} \in \mathbb{R}^{I_3 \times R_3}$ et un tenseur cœur $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{\{1\}} \times_F \mathbf{A}^{\{2\}} \times_3 \mathbf{A}^{\{3\}} = [\![\mathcal{G}; \mathbf{A}^{\{1\}}, \mathbf{A}^{\{2\}}, \mathbf{A}^{\{3\}}]\!]. \tag{1.8}$$

Un élément de $\mathcal{X}$ peut s'exprimer selon :

$$x_{i_1 i_2 i_3} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} a^{\{1\}}_{i_1 r_1} a^{\{2\}}_{i_2 r_2} a^{\{3\}}_{i_3 r_3}.$$

La figure 1.3 illustre cette décomposition.

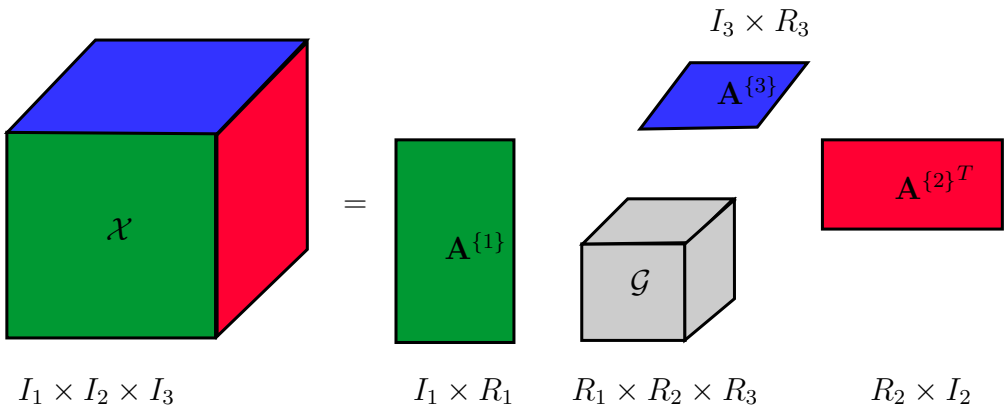


FIGURE 1.3 – Décomposition de TUCKER d'un tenseur d'ordre 3, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$

La décomposition de Tucker n'est pas unique. Lorsqu'on impose des contraintes d'orthogonalité sur les colonnes de $\mathbf{A}^{\{1\}}, \mathbf{A}^{\{2\}}, \cdots \mathbf{A}^{\{N\}}$, cette décomposition est unique. Elle est appelée décomposition en valeurs singulières d'ordre supérieur ou HOSVD (pour *higher-order singular value decomposition* en anglais) d'après les travaux de L. D. Lathauwer *et al.* [21]. La HOSVD peut être considérée comme une généralisation de la décomposition en valeurs singulières au cas tensoriel ($N \geq 3$). Cependant, elle ne généralise pas la notion de rang en tant que nombre minimum de termes de rang 1. On appelle rang multilinéaire de la HOSVD le triplet (R_1, R_2, R_3).

Décomposition CP

La décomposition CP (DCP) est un cas particulier de la décomposition de Tucker, avec un tenseur cœur $\mathcal{G}$ hyperdiagonal. Utilisée initialement en phonétique [9] et en psychométrie [56], la décomposition CP a regagné de l'intérêt dans les années 2000 dans les domaines liés au traitement du signal et analyse de données [63]. Elle a également été utilisée

dans plusieurs domaines : la neuroscience [64], la chimiométrie [15], les télécommunications [65], la séparation de source aveugle [66].

La décomposition CP décompose un tenseur en une somme de tenseurs de rang 1. Elle consiste à trouver les matrices $\mathbf{A}^{\{1\}} \in \mathbb{R}^{I_1 \times K}, \mathbf{A}^{\{2\}} \in \mathbb{R}^{I_2 \times K}, \mathbf{A}^{\{3\}} \in \mathbb{R}^{I_3 \times K}$ telles que :

$$\mathcal{X} = \sum_{k=1}^{K} \mathbf{a}_k^{\{1\}} \circ \mathbf{a}_k^{\{2\}} \circ \mathbf{a}_k^{\{3\}} = [\![\mathbf{A}^{\{1\}}, \mathbf{A}^{\{2\}}, \mathbf{A}^{\{3\}}]\!], \tag{1.9}$$

avec $\mathbf{A}^{\{n\}} = [\mathbf{a}_1^{\{n\}}, \mathbf{a}_F^{\{n\}}, \cdots, \mathbf{a}_K^{\{n\}}]$ et $n, = 1, \cdots, 3$.
La décomposition CP (1.9) peut s'écrire comme :

$$x_{i_1 i_2 \cdots i_N} = \sum_{k=1}^{K} a_{i_1 k}^{\{1\}} a_{i_2 k}^{\{2\}} a_{i_3 k}^{\{3\}}.$$

La figure 1.4 donne une représentation graphique de la décomposition. CP

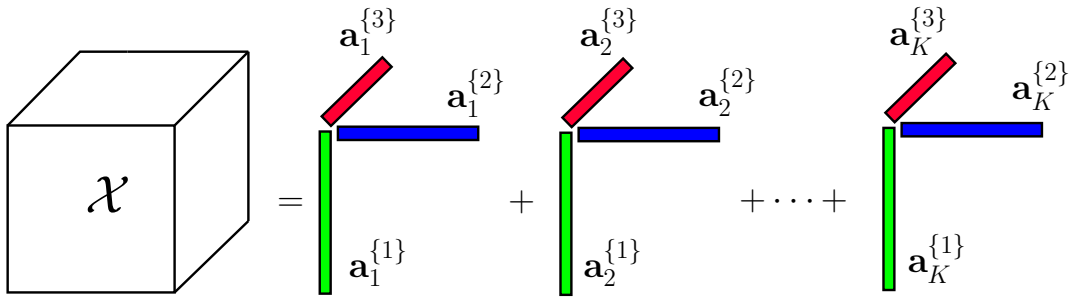


FIGURE 1.4 – Décomposition CP d'un tenseur d'ordre 3.

Le rang d'un tenseur $\mathcal{X}$ noté $rang(\mathcal{X})$ est le plus petit nombre K de tenseurs de rang 1 ou composants dont la somme est égale à $\mathcal{X}$ ou le plus petit entier positif qui permet de vérifier exactement (1.9). Au contraire de la HOSVD, des arguments théoriques peuvent remettre en cause l'utilisation de la DCP en tant qu'outil qu'approximation. En particulier, [53] montre qu'il s'agit d'un problème inverse mal posé mais qu'imposer une structure particulière au tenseur (non-négativité, symétrie) permet de le rendre bien posé. C'est un point important car il garantit qu'on peut choisir le rang de la décomposition en fonction du niveau de détail souhaité : un faible nombre de composants révèle les structures importantes tandis que l'utilisation d'un nombre élevé de composants permet de révéler les détails. Le choix K équivaut à un problème d'optimisation dans lequel on recherche le nombre minimal de composants qui expliquent au mieux la structure du tenseur pour un niveau d'erreur donné. Déterminer le rang K d'un tenseur est un problème NP complet [67].

La DCP de $\mathcal{X}$ (figure 1.4), peut être réécrite sous forme matricielle en utilisant le

dépliement de $\mathcal{X}$. Dans le cas d'un tenseur d'ordre 3, on a :

$$\mathbf{X}_{(1)} = \mathbf{A}^{\{1\}}(\mathbf{A}^{\{3\}} \odot \mathbf{A}^{\{2\}})^T, \tag{1.10}$$

$$\mathbf{X}_{(2)} = \mathbf{A}^{\{2\}}(\mathbf{A}^{\{3\}} \odot \mathbf{A}^{\{1\}})^T, \tag{1.11}$$

$$\mathbf{X}_{(3)} = \mathbf{A}^{\{3\}}(\mathbf{A}^{\{2\}} \odot \mathbf{A}^{\{1\}})^T, \tag{1.12}$$

avec $(\odot)$ le produit de Khatri-Rao.
Dans la suite du chapitre nous nous intéressons à la définition de décompositions matricielles et tensorielles de tableaux de données binaires qui sont le cœur de ce travail. Cependant au préalable et afin d'alléger les notations et de s'affranchir des indices sur $\mathbf{A}^{\{1\}}$ et $\mathbf{A}^{\{2\}}$, nous procédons à un changement de notation et écrirons les produits matriciels et décompositions tensorielles avec les symboles $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$. Ainsi, par exemple la factorisation matricielle sera notée $\mathbf{W}\mathbf{H}^T$ et une DCP sera notée $[\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]$.

1.2 Factorisation des matrices à valeurs binaires

Les données binaires sont de plus en plus présentes dans divers domaines. Elles peuvent être soit générées par des processus ou phénomènes à deux états (et naturellement codées sur les valeurs binaires $\{0, 1\}$), par exemple : des capteurs de proximité, des boutons-poussoirs, des interrupteurs électriques, des réponses oui/non dans les enquêtes, etc., ou obtenus artificiellement par une forte quantification (sur 1 bit) de données réelles.
Les techniques de quantification sur 1 bit et de compression sur 1 bit ont été initialement développées pour accélérer les convertisseurs numériques (voir par exemple, [68], [69]) mais elles représentent en même temps une solution intéressante au problème de déluge de données constaté ces dernières années dans de nombreux domaines d'application. En spectroscopie, par exemple, un expert peut être plus intéressé par la présence/absence d'un pic dans un spectre enregistré (ou séparé) à une longueur d'onde donnée, plutôt que par l'amplitude réelle de ce pic.

Par conséquent, la Factorisation des Matrices Binaires (FMB) a récemment suscitée beaucoup d'intérêt et a déjà été utilisée avec succès dans de nombreux domaines tels que les télécommunications numériques [70], [5]. En effet dans [70], ils proposent une approche de maximum de vraisemblance pour séparer et estimer de multiples signaux numériques synchrones arrivant sur un réseau d'antennes. Dans [5], on utilise la factorisation d'une matrice de données, pour séparer à l'aveugle de multiples sources binaires dans un même canal de transmisssion.

Dans le domaine de la classification des réponses des gènes, les auteurs de [42] utilisent cette méthode de factorisation pour sélectionner les groupes de gènes qui présentent des comportements d'expression similaires dans un sous-ensemble d'échantillons appartenant à des classes spécifiques telles que certains types de tumeurs, afin d'identifier les structures locales des données de la matrice de micro-réseau [71]. Dans [72], ils appliquent la fac-

torisation binaire au problème de classification des données de panier de supermarché et à la classification de documents. Pour la classification du panier de supermarché, chaque transaction peut être représentée sous la forme d'un vecteur binaire où chaque élément indique si l'article ou le produit correspondant a été acheté ou non. Pour la classification de documents, chaque document peut être représenté comme un vecteur binaire où chaque élément indique si un mot ou un terme donné est présent ou non. Dans [2], ils appliquent cette factorisation pour la découverte de motifs pour les images d'expression de gènes de Drosophile. La factorisation est aussi utilisée pour l'extraction d'attributs discrets en grande dimension [73], la reconstruction de caractères alpha-numériques aléatoirement échantillonnés [74] ou dans les systèmes de recommandation [3].

L'objectif de cette section est de définir un produit matriciel permettant de représenter une matrice $\mathbf{X}$ binaire ($x_{ij} \in \{0, 1\}$), comme produit de deux matrices selon :

$$\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T, \tag{1.13}$$

où $\diamond$ représente un produit matriciel à préciser et K le rang de la décomposition.

En fonction de l'ensemble sur lequel est appliquée la décomposition, la factorisation et son rang K peuvent être définis de différentes manières. Les sections suivantes analysent différentes factorisations de matrices binaires et leurs rangs. On distingue deux types de factorisations : celles qui utilisent des opérations définies sur les réels et celles à base d'opérateurs logiques.

1.2.1 Factorisations sur l'ensemble des réels

La décomposition d'une matrice binaire en une somme de K matrices de rang 1 est définie par :

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T \triangleq \mathbf{W}\mathbf{H}^T. \tag{1.14}$$

En fonction de l'ensemble sur lequel est défini $\mathbf{w}_k$ et $\mathbf{h}_k$, le rang peut être défini de différentes façons selon les contraintes imposées aux facteurs.

Si les éléments de $\mathbf{w}_k$ et $\mathbf{h}_k$ sont à *valeurs réelles*, le rang de la décomposition de (1.14) correspond au *rang réel* classique que l'on notera $rang_{\mathbb{R}}(\mathbf{X})$. La FMB de $\mathbf{X}$ et le rang réel de cette factorisation matricielle peuvent être obtenus par des méthodes algébriques bien connues telles que la décomposition en valeurs singulières. La décomposition réelle est intéressante par le fait que de nombreuses méthodes de calculs efficaces. Cependant elle ne préserve pas la binarité des matrices facteurs et conduit à des résultats qui sont souvent difficiles à interpréter.

Si les éléments de $\mathbf{w}_k$ et $\mathbf{h}_k$ sont contraints à être *non-négatifs*, le problème (1.14) se résume au problème de la Factorisation en matrices non-négatives (ou NMF pour *Nonnegative Matrix Factorisation* en anglais) [7, 75] et son rang est le *rang non-négatif*, représenté par $rang_{\mathbb{R}_+}(\mathbf{X})$. Cette décomposition ne préserve pas la binarité des matrices

facteurs bien qu'elle ait déjà été utilisée dans ce cadre.

Si les éléments de $\mathbf{w}_k$ et $\mathbf{h}_k$ sont contraints à prendre des *valeurs binaires*, la décomposition de (1.14) est connue comme la décomposition binaire de $\mathbf{X}$ et le rang correspondant est le *rang binaire* [76] noté $rang_{\{0,1\}}(\mathbf{X})$. Dans ce cas, le rang binaire peut être interprété comme le nombre minimal de rectangles dont ces éléments sont à valeurs 1, qui sont à supports disjoints et dont l'union couvre de façon exacte le support $\mathbf{X}$. Ce type de décomposition préserve la nature binaire des résultats mais le nombre de termes de la décomposition est souvent grand.

1.2.2 Factorisations booléennes

Une alternative permettant de préserver la nature binaire des matrices facteur est de définir la multiplication de matrices à l'aide d'opération booléens. Ainsi la multiplication s'assimile au ET logique, noté ($\wedge$) tandis que l'addition peut être remplacée par le OU logique, noté ($\vee$) ou par le OU Exclusif noté ($\oplus$). On peut alors définir la Factorisation Booléenne des Matrices Binaires (FBMB) selon [77, 78] :

$$x_{ij} = \bigvee_{k=1}^{K} (w_{ik} \wedge h_{jk}). \tag{1.15}$$

En remarquant que dans le cas de données binaires, il y a équivalence entre le ET logique et la multiplcation, c'est-à-dire : $w_{ik} \wedge h_{jk} = w_{ik} h_{jk}$, on écrire :

$$\mathbf{X} = \bigvee_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T \triangleq \mathbf{W} \diamond \mathbf{H}^T, \tag{1.16}$$

Une FBMB alternative utilise l'opérateur OU Exclusif [79, 80] et conduit à la définition suivante :

$$\mathbf{X} = \bigoplus_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T \triangleq \mathbf{W} \boxtimes \mathbf{H}^T. \tag{1.17}$$

Exemple 1.2.1. « Sommes » de deux sources binaires corrélées
Cet exemple permet de mettre en évidence les différences entre les différents produits matriciels (1.14), (1.16) et (1.17) sur un exemple simple correspondant à la superposition de deux matrices de rang 1, que l'on désigne dans la suite par *sources binaires*. Si les sources binaires sont à supports disjoints, on dit que les sources sont non-corrélées, tandis que si elles sont à supports non disjoints, les sources sont dites corrélées. Dans le cas où les sources sont non-corrélées, les 3 produits matriciels sont équivalents car ils fournissent le même résultat. En revanche si les sources sont corrélées, les trois produits ne sont plus équivalents. Le produit réel (1.14) fournit un résultat qui n'est pas une matrice binaire.

Les produits booléens garantissent la binarité du produit mais le résultat est très différent.

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

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

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

1.2.3 Propriétés algébriques des produits booléens matriciels

Soient $(\mathbb{M}_b, \oplus, \boxtimes)$ l'ensemble des matrices binaires muni de l'opérateur d'addition $(\oplus)$ et de l'opérateur de multiplication $(\boxtimes)$ et $(\mathbb{M}_b, \vee, \diamond)$, l'ensemble des matrices binaires muni de $(\vee)$ et $(\diamond)$. Les propriétés algébriques de ces deux ensembles sont rappelées ci-dessous.

Propriétés par rapport à l'addition

— Elément neutre de l'addition $(\vee)$ et $(\oplus)$
La matrice nulle $\mathbf{0} \in \mathbb{M}_b(N, P)$ est l' élément neutre par rapport à $(\vee)$ et $(\oplus)$, c'est-à-dire : $\mathbf{A} \vee \mathbf{0} = \mathbf{A}$ et $\mathbf{A} \oplus \mathbf{0} = \mathbf{A}$

— Elément opposé de l'addition $(\vee)$ et $(\oplus)$
$\nexists\ \bar{\mathbf{A}}$ (élément opposé de $\mathbf{A}$) tel que : $\mathbf{A} \vee \bar{\mathbf{A}} = \mathbf{0}$, $\forall \mathbf{A}, \bar{\mathbf{A}} \in \mathbb{M}_b(N, P)$. Seule la matrice nulle est son propre opposé dans $\mathbb{M}_b(N, P)$.
$\exists\ \bar{\mathbf{A}}$ tel que : $\mathbf{A} \oplus \bar{\mathbf{A}} = \mathbf{0}$, $\forall \mathbf{A}, \bar{\mathbf{A}} \in \mathbb{M}_b(n, p)$, et dans ce cas, $\bar{\mathbf{A}} = \mathbf{A}$.

— Associativité $(\vee)$ et $(\oplus)$
$\mathbf{C} \vee (\mathbf{A} \vee \mathbf{B}) = (\mathbf{C} \vee \mathbf{A}) \vee \mathbf{B}$
$\mathbf{C} \oplus (\mathbf{A} \oplus \mathbf{B}) = (\mathbf{C} \oplus \mathbf{A}) \oplus \mathbf{B}$

— Commutativité de $(\vee)$ et $(\oplus)$
$\mathbf{A} \vee \mathbf{B} = \mathbf{B} \vee \mathbf{A}$, $\forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_b(N, P)$, avec $\mathbb{M}_b(N, P)$ qui repésente l'ensemble des matrices de taille $N \times P$.
$\mathbf{A} \oplus \mathbf{B} = \mathbf{B} \oplus \mathbf{A}$, $\forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_b(N, P)$.

$(\mathbb{M}_b, \oplus)$ est un groupe commutatif. En revanche $(\mathbb{M}_b, \vee)$ ne l'est pas. C'est un monoïde commutatif.

Propriétés par rapport à la multiplication

— L'élément neutre de $(\diamond)$ et $(\boxtimes)$
La matrice $\mathbf{I} \in \mathbb{M}_b(P, P)$ est l'élément neutre (à droite) par rapport à $(\diamond)$ et $\oplus$) c'est-

à-dire : $\mathbf{A} \diamond \mathbf{I} = \mathbf{A}$, $\forall \mathbf{A} \in \mathbb{M}_b(N, P)$. De même, $\mathbf{A} \boxtimes \mathbf{I} = \mathbf{A}$. De même $\mathbf{I} \in \mathbb{M}_b(N, N)$ est l'élément neutre à gauche.

— Associativité du « produit booléen » ($\diamond$) et ($\boxtimes$)
$\mathbf{C} \diamond (\mathbf{A} \diamond \mathbf{B}) = (\mathbf{C} \diamond \mathbf{A}) \diamond \mathbf{B}$, $\forall \mathbf{A} \in \mathbb{M}_b(N, P)$, $\mathbf{B} \in \mathbb{M}_b(P, D)$, $\mathbf{C} \in \mathbb{M}_b(R, N)$.
$\mathbf{C} \boxtimes (\mathbf{A} \boxtimes \mathbf{B}) = (\mathbf{C} \boxtimes \mathbf{A}) \boxtimes \mathbf{B}$, $\forall \mathbf{A} \in \mathbb{M}_b(N, P)$, $\mathbf{B} \in \mathbb{M}_b(P, D)$, $\mathbf{C} \in \mathbb{M}_b(R, N)$.

Donc, $(\mathbb{M}_b, \diamond)$ et $(\mathbb{M}_b, \boxtimes)$ sont des monoïdes.

Distributivité

— $\mathbf{C} \diamond (\mathbf{A} \vee \mathbf{B}) \neq (\mathbf{C} \diamond \mathbf{A}) \vee (\mathbf{C} \diamond \mathbf{B})$, $\forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_b(N, P)$, $\mathbf{C} \in \mathbb{M}_b(R, N)$

— $\mathbf{C} \boxtimes (\mathbf{A} \oplus \mathbf{B}) = (\mathbf{C} \boxtimes \mathbf{A}) \oplus (\mathbf{C} \boxtimes \mathbf{B})$, $\forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_b(N, P)$, $\mathbf{C} \in \mathbb{M}_b(R, N)$

la loi ($\diamond$) n'est pas distributive par rapport à ($\vee$), en revanche la loi ($\boxtimes$) est distributive par rapport à ($\oplus$).

Comme $(\mathbb{M}_b, \oplus)$ est un groupe commutatif, $(\mathbb{M}_b, \boxtimes)$ est un monoïde et que la loi ($\boxtimes$) est distributive par rapport à ($\oplus$), $(\mathbb{M}_b, \oplus, \boxtimes)$ est un anneau. Il possède des propriétés similaires à l'anneau des matrices réelles. Ce n'est pas le cas pour $(\mathbb{M}_b, \vee, \diamond)$ car $(\mathbb{M}_b, \vee)$ n'a pas d'élément opposé et la loi ($\diamond$) n'est pas distributive par rapport à ($\vee$). Le produit matriciel booléen ($\diamond$) en revanche est intéressant en termes de rang et d'interprétabilité.

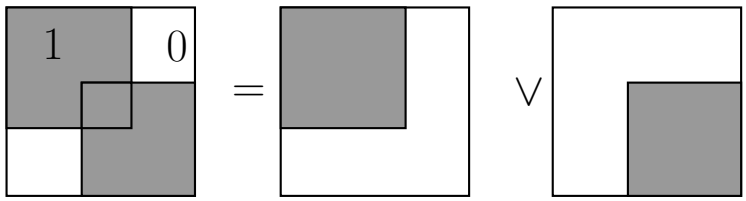
1.2.4 Rang booléen

Le rang booléen d'une matrice binaire $\mathbf{X}$ [81] appelé aussi rang de *Schein* [82] que l'on notera, $rang_{\mathbb{B}}(\mathbf{X})$ peut être défini comme le nombre minimum de matrices de rang 1 (pas forcément à supports disjoints) dont l'union des supports est égal au support de la matrice $\mathbf{X}$. C'est donc le rang associé à la factorisation 1.16. Un intérêt important de cette factorisation est que le rang booléen est toujours plus petit que les autres rangs non-négatifs [83].

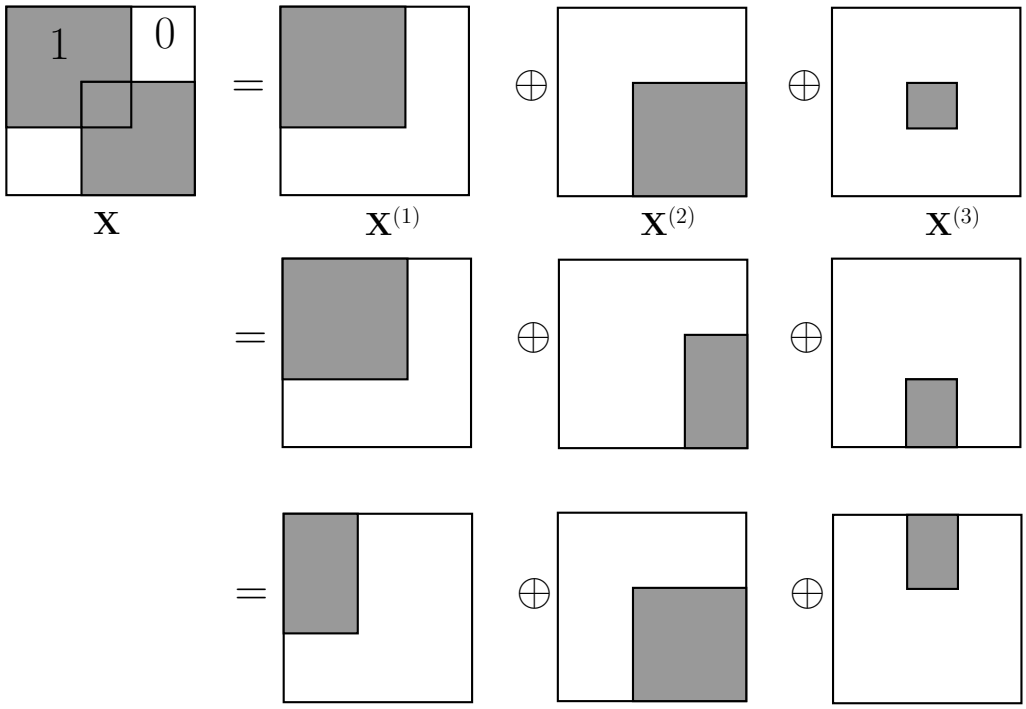
$$rang_{\mathbb{B}}(\mathbf{X}) \leq rang_{\mathbb{R}_+}(\mathbf{X}) \leq rang_{\{0,1\}}(\mathbf{X}). \tag{1.18}$$

A notre connaissance, il n'y a pas de relation prouvée entre le rang réel et le rang booléen. Cependant nous n'avons pas réussi à trouver de décomposition de matrices binaires dont le rang réel est inférieur au rang booléen. Le rang associé à la décomposition (1.17) est noté $rang_{\oplus}$. A notre connaissance, il n'y a pas non plus de relation prouvée pour ce rang. Pour illustrer les rangs associés aux différentes décompositions, nous proposons d'analyser quelques exemples simples.

Exemple 1.2.2. Considérons la matrice binaire $\mathbf{X}$ ci-desssous, avec la partie grise qui correspond aux éléments à 1 de la matrice et la partie coloriée en blanc où les éléments de la matrice valent 0.



(a) Décomposition de $\mathbf{X}$ utilisant le produit booléen ($\diamond$).



(b) Décomposition de $\mathbf{X}$ utilisant le produit booléen ($\boxtimes$).

FIGURE 1.5 – Décomposition de deux sources corrélées avec les différents produits matriciels booléens.

En observant les figures 1.5a et 1.5b, il apparaît que le produit matriciel booléen ($\diamond$), permet de décomposer $\mathbf{X}$ de façon unique (c'est-à-dire qu'il existe une unique paire de matrices $\mathbf{W}$ et $\mathbf{H}$ tel que : $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$), et que $rang_{\mathbb{B}}(\mathbf{X}) = 2$. En revanche, le rang associé au produit matriciel binaire ($\boxtimes$) est $rang_{\oplus}(\mathbf{X}) = 3$ et en plus, la décomposition n'est pas unique. Il est possible cependant de trouver des exemples où $rang_{\oplus}(\mathbf{X}) < rang_{\mathbb{B}}(\mathbf{X})$.

Exemple 1.2.3. : Le premier exemple considère le cas de sources non-corrélées (à support disjoint). Soit la matrice binaire de taille 4×4 :

$$\mathbf{X} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}. \tag{1.19}$$

La décomposition de $\mathbf{X}$ en un nombre minimum de termes de rang 1 peut être exprimée

comme :

$$\mathbf{X} = \underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(2)}}$$
$$= \underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} \vee \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(2)}}$$

En fait dans le cas de sources séparables toutes les factorisations sont équivalentes et

$$rang_{\mathbb{B}}(\mathbf{X}) = rang_{\oplus}(\mathbf{X}) = rang_{\mathbb{R}_+}(\mathbf{X}) = rang_{\{0,1\}}(\mathbf{X}). \tag{1.20}$$

Exemple 1.2.4. Le second exemple considère le cas de sources corrélées. Soit la matrice binaire de taille 3×3 :

$$\mathbf{X} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}. \tag{1.21}$$

Le rang réel de la matrice $\mathbf{X}$ est $rang_{\mathbb{R}}(\mathbf{X}) = 3$, et les termes de rang 1 obtenus par SVD sont donnés par :

$$\mathbf{X} = \underbrace{\begin{pmatrix} 0.6036 & 0.8536 & 0.6036 \\ 0.8536 & 1.2071 & 0.8536 \\ 0.6036 & 0.8536 & 0.6036 \end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix} 0.5 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \end{pmatrix}}_{\mathbf{X}^{(2)}} + \underbrace{\begin{pmatrix} -0.1036 & 0.1464 & -0.1036 \\ 0.1464 & -0.2071 & 0.1464 \\ -0.1036 & 0.1464 & -0.1036 \end{pmatrix}}_{\mathbf{X}^{(3)}}.$$

Cette décomposition n'est pas unique, et il y a une infinité de décompositions qui permettent de retrouver la matrice $\mathbf{X}$ avec un rang réel égal à 3. En outre, il est très difficile, voire impossible, d'interpreter les sources $\mathbf{X}^{(k)}$ à partir de la matrice binaire $\mathbf{X}$. On rencontre la même difficulté avec la décomposition non-négative et le rang correspondant $rang_{\mathbb{R}_+}$, la seule différence est que les valeurs de $\mathbf{X}^{(k)}, k = 1, \cdots, K$ sont non-négatives, ce qui, en général, conduit à une décomposition « plus interprétable ».

Le rang binaire de cette matrice $\mathbf{X}$ est $rang_{\{0,1\}}(\mathbf{X}) = 3$ et 2 exemples de décompositions

sont données (il en existe d'autres) :

$$\mathbf{X} = \underbrace{\begin{pmatrix} \mathbf{1} & 0 & 0 \\ \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix} 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 \end{pmatrix}}_{\mathbf{X}^{(2)}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(3)}}$$
$$= \underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(2)}} + \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(3)}}.$$

Cette décomposition est évidemment plus interprétable, car elle préserve la nature binaire des résultats cependant son rang est supérieur aux rangs réel et non-négatif et la décomposition n'est pas unique.

En remplaçant la somme arithmétique (+) par l'opérateur logique ($\vee$), on obtient comme décomposition booléenne de $\mathbf{X}$:

$$\mathbf{X} = \underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} \vee \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(2)}}.$$

Le rang booléen de cette décomposition est $rang_{\mathbb{B}}(\mathbf{X}) = 2$. De plus, la décomposition booléenne est unique ce qui n'est pas le cas pour les autres décompositions.

Dans la suite de ce manuscrit , nous utiliserons comme modèle de factorisation (1.16) pour les deux raisons suivantes : i) c'est, à notre sens la factorisation qui donne les résultats les plus interprétables, ii) le rang de cette décomposition est souvent le plus faible.

1.3 Algorithmes de factorisation booléenne de matrices binaires (FBMB)

1.3.1 Un panorama rapide des algorithmes de FMB

Au cours des deux dernières décennies, la majorité des résultats et des algorithmes pour la FMB provenait des domaines des statistiques, de la science des données et de l'apprentissage. Cependant, deux des premiers algorithmes de factorisation matricielle dont les facteurs sont binaires, sont issus du traitement du signal [70], [5], et ont été développés pour la séparation de sources dans le domaine des télécommunications numériques. Néanmoins, ces algorithmes considèrent le cas où un seul des deux facteurs (la matrice des sources) a des entrées binaires {-1, +1}, alors que les données et l'autre facteur sont à valeurs complexes. Une des premières méthodes de décomposition de matrices à valeurs binaires est

l'analyse en composantes principales logistiques (ou Logistic Principal Components Analysis (LPCA) en anglais)[36] qui utilise un modèle probabiliste pour les données. Basée sur une méthode bien connue en statistique, à savoir la régression logistique, l'algorithme de [36] a servi d'inspiration à plusieurs autres méthodes similaires, telles que celles proposées dans [37], [38], [39], [40], [84]. Un problème étroitement lié au LPCA, qui a récemment concentré des efforts de recherche, est la complétion de matrices binaires, où seul un sous-ensemble des entrées de la matrice binaire observée est supposé connu ([85], [86], [87], [88]).

Comme la FBMB est un problème NP-complet [83], diverses stratégies pour estimer et extraire itérativement les termes de rang 1 (sources) de la matrice d'observation ont été proposées. Notamment des approches de type glouton (ou gourmand) ([89], [73], [90], [83], [44], [91]). Une autre façon de contourner l'aspect NP-complet du problème de FMB est de relaxer ce problème sur l'orthant réel non-négatif ([92], [93], [42]). De cette façon, le problème discret est transformé en un problème d'optimisation continue, avec des contraintes qui forcent la binarité des résultats. Ces trois méthodes de FBMB utilisent des approches algorithmiques différentes ; elles sont cependant toutes basées sur un modèle de mélange utilisant le produit matriciel réel classique. Cela implique (comme expliqué dans la section suivante) que ces méthodes favorisent les décompositions en sources décorrélées, car le produit réel pénalise fortement les facteurs ayant des supports qui se chevauchent. Ceci représente un inconvénient majeur dans une grande classe d'applications où on cherche des comportements communs à plusieurs « populations ». De plus, en favorisant le non-chevauchement des sources, ces méthodes ont tendance à nécessiter un rang de décomposition élevé pour la décomposition des matrices binaires : c'est une caractéristique non souhaitable dans les applications d'approximation de rang faible et de compression de données.

1.3.2 Formulation du problème FBMB

Considérons à présent le modèle de factorisation booléenne de rang K d'une matrice à valeurs binaires. Dans le cas d'une factorisation exacte, cela s'exprime selon :

$$\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T = \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k^T. \tag{1.22}$$

Supposons que ce modèle exact est corrompu par du bruit ou des erreurs. En fonction de l'ensemble sur lequel la décomposition de la matrice binaire est effectuée, ce terme de bruit peut prendre différentes formes. Pour la factorisation en matrices binaires réelles (les éléments des facteurs $\mathbf{w}_k$ et $\mathbf{h}_k$ sont à valeur réelles), le terme de « bruit » est la matrice $\mathbf{N}$ à valeurs réelles, utilisée classiquement dans la plupart des applications de traitement du signal. Généralement, les entrées de $\mathbf{N}$ sont supposées être distribuées suivant une loi gaussienne de moyenne nulle. Pour la décomposition non-négative, les entrées de $\mathbf{N}$ seront réelles, non-négatives.

Dans le cas de la décomposition de $\mathbf{X}$ en termes binaires, et en particulier pour la FBMB, la matrice de « bruit », $\mathbf{N}$, est supposée être la réalisation d'une variable aléatoire tirée selon une loi binaire, par exemple une loi de Bernoulli, et que le modèle de dégradation est le suivant :

$$\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T \oplus \mathbf{N}. \tag{1.23}$$

Cela signifie que les éléments à 1 de la matrice de bruit $\mathbf{N}$ changent les éléments correspondants dans $\mathbf{W} \diamond \mathbf{H}^T$ quelque soit leurs valeurs. La figure 1.6 ci-après, permet d'illustrer l'ajout d'un bruit binaire dans la FBMB :

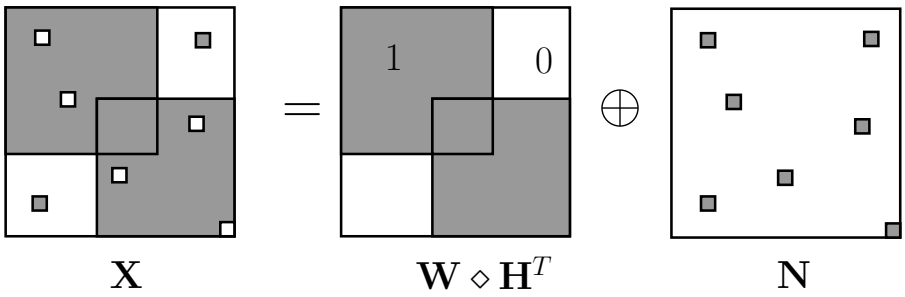


FIGURE 1.6 – Illustration du bruit binaire de la FBMB

Ainsi, le problème de FBMB approchée consiste à déterminer les matrices $\mathbf{W}$ et $\mathbf{H}$ qui satisfont :

$$\{\mathbf{W}, \mathbf{H}\} = \arg\min_{\mathbf{W},\mathbf{H}} \left\| \mathbf{X} - \mathbf{W} \diamond \mathbf{H}^T \right\|_F^2. \tag{1.24}$$

Il est à noter que, les données étant binaires, la norme de Frobenius dans (1.24) est équivalente aux normes ℓ_1 et ℓ_0 et que toutes correspondent au cardinal du support de la matrice en argument. Comme mentionné plus haut, le problème (1.24) est NP-complet [83]. Ainsi, plusieurs stratégies ont été proposées dans la littérature afin de contourner cette difficulté.

1.3.3 Approches par relaxation réelle de la FBMB

Plusieurs auteurs ([92], [93], [42]) ont proposé de remplacer le produit matriciel booléen, ($\diamond$), par le produit matriciel classique, ce qui revient à remplacer le problème (1.24) par le problème suivant :

$$\{\mathbf{W}, \mathbf{H}\} = \arg\min_{\mathbf{W},\mathbf{H}\in\{0,1\}} \|\mathbf{X} - \mathbf{W}\mathbf{H}^T\|_F^2, \tag{1.25}$$

où $\mathbf{W}\mathbf{H}^T$ est le produit matriciel réel classique. On parle alors de relaxation réelle de (1.24) qui peut être abordé comme un problème d'optimisation avec contrainte, plus facile à résoudre. Nous présentons dans la suite, les deux algorithmes de FBMB proposés dans [92] qui sont basés sur l'approche de relaxation réelle non-négative, à savoir l'algorithme PF (Penalty Function en anglais) et l'algorithme TH (THresholding en anglais).

L'algorithme « Penalty Function » (PF) consiste à imposer la contrainte de binarité par une fonction de pénalité (PF) et utilise une descente de gradient avec une mise à jour multiplicative similaire à la NMF [8], [7] pour minimiser la fonction de coût suivante :

$$J(\mathbf{W},\mathbf{H}) = \sum_{i,j}\left(\mathbf{X}_{ij} - (\mathbf{W}\mathbf{H}^T)_{ij}\right)^2 + \frac{1}{2}\lambda\sum_{i,k}\left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \frac{1}{2}\lambda\sum_{j,k}\left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2. \tag{1.26}$$

La contrainte de binarité sur $\mathbf{H}$ et $\mathbf{W}$ est imposée par les termes de pénalité $\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}$ et $\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}$. Les étapes de l'algorithme PF sont données dans la table algorithmique 1. En pratique, pour une convergence plus rapide de l'algorithme PF, $\mathbf{W}$ et $\mathbf{H}$ sont initialisées avec le résultat de l'algorithme de NMF [8]. À l'étape 3, $\mathbf{W}$ et $\mathbf{H}$ sont normalisées afin de forcer les valeurs de $\mathbf{W}_{ij}$ et $\mathbf{H}_{ij}$ à rester dans l'intervalle [0, 1].

Algorithm 1 : Penalty Function (PF)

Entrées $\mathbf{X}$, K, ε, λ
Sorties : $\mathbf{W}$, $\mathbf{H}$
PAS 1 : Initialisation de $\mathbf{W}$ et $\mathbf{H}$
PAS 2 : Normalisation
$\mathbf{W} \leftarrow \mathbf{W}\mathbf{D}_W^{-1/2}\mathbf{D}_H^{1/2}$, avec $\mathbf{D}_W = diag\left(\max(\mathbf{w}_1), \max(\mathbf{w}_F), \cdots, \max(\mathbf{w}_K)\right)$
$\mathbf{H} \leftarrow \mathbf{H}\mathbf{D}_H^{-1/2}\mathbf{D}_W^{1/2}$, $\mathbf{D}_H = diag\left(\max(\mathbf{h}_1), \max(\mathbf{h}_F), \cdots, \max(\mathbf{h}_K)\right)$
PAS 3 : Mise à jour $\mathbf{W}$, $\mathbf{H}$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{(\mathbf{X}^T\mathbf{W})_{jk} + 3\lambda\mathbf{H}_{jk}^2}{((\mathbf{W}\mathbf{H}^T)^T\mathbf{W})_{jk} + 2\lambda\mathbf{H}_{jk}^3 + \lambda\mathbf{H}_{jk}}$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik}\frac{(\mathbf{X}\mathbf{H})_{ik} + 3\lambda\mathbf{W}_{ik}^2}{((\mathbf{W}\mathbf{H}^T)\mathbf{H})_{ik} + 2\lambda\mathbf{W}_{ik}^3 + \lambda\mathbf{W}_{ik}}$$

PAS 4 : Critère d'arrêt
if $\sum_{i,j}\left((\mathbf{H}_{ij}^2 - \mathbf{H}_{ij})^2 + (\mathbf{W}_{ij}^2 - \mathbf{W}_{ij})^2\right) < \varepsilon$ **then**
break
else
retourner à PAS 3
end if

L'algorithme « THresholding » (TH) proposé aussi dans [92] , est fondé sur une procédure de seuillage des valeurs des matrices $\mathbf{W}$ et $\mathbf{H}$. Le principe de cette méthode est de trouver deux seuils w et h pour les deux matrices $\mathbf{W}$ et $\mathbf{H}$ qui minimisent le critère :

$$F(w,h) = \sum_{i,j}\left(\mathbf{X}_{ij} - \left(\theta(\mathbf{W} - \mathbf{1}_{N\times K}\cdot w)\cdot\theta(\mathbf{H} - \mathbf{1}_{M\times K}\cdot h)^T\right)_{ij}\right)^2, \tag{1.27}$$

où θ est la fonction de Heaviside appliquée élément à élément et $\mathbf{1}_{N\times K}$, la matrice de 1 de taille $N \times K$. Pour des raisons d'implémentation, la fonction de Heaviside est approchée par la fonction sigmoïde (1.28). Cette fonction est dérivable et possède un paramètre γ

qui permet de régler la pente.

$$\theta(x) \approx \phi(x) = \frac{1}{1+e^{-\gamma x}}. \tag{1.28}$$

L'algorithme TH est donné dans la table algorithmique 2.

Algorithm 2 : THresholding (TH)

Entrées : $\mathbf{X}$, K, γ
Sorties : $\mathbf{W}$, $\mathbf{H}$
PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$, $h^{(0)}$, $w^{(0)}$ et $iter = 1$
PAS 2 : Normalisation
$\mathbf{W} \leftarrow \mathbf{W}\mathbf{D}_W^{-1/2}\mathbf{D}_H^{1/2}$, avec $\mathbf{D}_W = diag\left(\max(\mathbf{w}_1), \max(\mathbf{w}_F), \cdots, \max(\mathbf{w}_K)\right)$
$\mathbf{H} \leftarrow \mathbf{H}\mathbf{D}_H^{-1/2}\mathbf{D}_W^{1/2}$, $\mathbf{D}_H = diag\left(\max(\mathbf{h}_1), \max(\mathbf{h}_F), \cdots, \max(\mathbf{h}_K)\right)$
PAS 3 : Calcul de la direction du gradient $\mathbf{g}^{(iter)}$ de $F(w,h)$
PAS 4 : Sélectionner le pas de gradient $\alpha^{(iter)}$
PAS 5 : Mise à jour
$w^{(iter)} = w^{(iter-1)} - \alpha^{(iter)}\mathbf{g}^{(iter)}(1)$
$h^{(iter)} = h^{(iter-1)} - \alpha^{(iter)}\mathbf{g}^{(iter)}(2)$
if le critère d'arrêt est satisfait
$\mathbf{W} = \theta(\mathbf{W} - w^{(iter)})$; $\mathbf{H} = \theta(\mathbf{H} - h^{(iter)})$; break
else
$iter \leftarrow iter + 1$, retourner à PAS 3
end

Remarque : les problèmes (1.25) et (1.24) sont équivalents si les termes de rang 1 ont des supports disjoints. En fait, (1.25) résout le problème de la FMB binaire au lieu de la FBMB. L'exemple suivant permettent d'illustrer le fait que ces approches peuvent néanmoins aussi donner de bons résultats si la corrélation des sources n'est pas « trop forte ».

Exemple 1.3.1. Considérons la FBMB suivante :

$$\underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}} = \underbrace{\begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}}_{\mathbf{X}^{(1)}} \vee \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}}_{\mathbf{X}^{(2)}}$$

Cette décomposition de rang 2 est unique. Supposons maintenant que nous effectuons

la décomposition avec un rang 2 de $\mathbf{X}$ en utilisant (1.25), alors dans ce cas nous obtenons :

$$\underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\0&0&\mathbf{1}&\mathbf{1}\end{pmatrix}}_{\mathbf{X}} \approx \underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\0&0&0&0\end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&\mathbf{1}&\mathbf{1}\\0&0&\mathbf{1}&\mathbf{1}\end{pmatrix}}_{\mathbf{X}^{(2)}}$$

$$\approx \underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\0&0&0&0\end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&\mathbf{1}\\0&0&0&\mathbf{1}\end{pmatrix}}_{\mathbf{X}^{(2)}}.$$

Dans ce cas, la « vraie » décomposition donne le même coût minimal ($\|\mathbf{X}-\mathbf{W}\mathbf{H}^T\|_F^2$=1) que l'autre décomposition binaire, ce qui veut dire que l'algorithme sortira après convergence l'une des deux décompositions de façon équivalente. Cependant, à mesure que la corrélation des deux sources augmente, la « vraie » solution de la décomposition booléenne de $\mathbf{X}$ ne correspond plus à un minimum de la fonction de coût (1.25), comme illustré par l'exemple suivant :

$$\underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\0&\mathbf{1}&\mathbf{1}&\mathbf{1}\end{pmatrix}}_{\mathbf{X}} = \underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\0&0&0&0\end{pmatrix}}_{\mathbf{X}^{(1)}} \vee \underbrace{\begin{pmatrix}0&0&0&0\\0&\mathbf{1}&\mathbf{1}&\mathbf{1}\\0&\mathbf{1}&\mathbf{1}&\mathbf{1}\\0&\mathbf{1}&\mathbf{1}&\mathbf{1}\end{pmatrix}}_{\mathbf{X}^{(2)}}.$$

Dans ce cas, la « vraie » solution donnée par la FBMB, donne un coût $\|\mathbf{X}-\mathbf{W}\mathbf{H}^T\|_F^2$=4, alors que la décomposition ci-dessous :

$$\underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\\mathbf{1}&\mathbf{1}&\mathbf{1}&\mathbf{1}\\0&\mathbf{1}&\mathbf{1}&\mathbf{1}\end{pmatrix}}_{\mathbf{X}} \approx \underbrace{\begin{pmatrix}\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\\mathbf{1}&\mathbf{1}&\mathbf{1}&0\\0&0&0&0\end{pmatrix}}_{\mathbf{X}^{(1)}} + \underbrace{\begin{pmatrix}0&0&0&0\\0&0&0&\mathbf{1}\\0&0&0&\mathbf{1}\\0&0&0&\mathbf{1}\end{pmatrix}}_{\mathbf{X}^{(2)}},$$

génère un coût plus faible $\|\mathbf{X}-\mathbf{W}\mathbf{H}^T\|_F^2 = 2$. Par conséquent, dans ce cas, les algorithmes fondés sur cette reformulation réelle du problème de la FBMB auront tendance à privilégier la deuxième décomposition.

1.3.4 Algorithmes « gloutons » pour la FBMB

La deuxième stratégie utilisée pour contourner le problème NP-complet (1.24) consiste à estimer et à soustraire itérativement les termes de rang 1 de $\mathbf{X}$, de manière « gloutonne ».

Plusieurs approches ont été proposées en utilisant différentes stratégies gloutonnes ([89], [73], [90], [83], [44], [91]). Nous présentons ci-après deux algorithmes de ce type.

L'algorithme ASSO (ou *ASSOciation rules algorithm* en anglais) résout le problème DBP (Discrete Basis Problem en anglais) a été introduit dans [83]. Il procède par déflation et recherche à chaque itération le terme de rang 1 qui minimise le résidu. L'implémentation de l'algorithme exploite la corrélation entre les colonnes de $\mathbf{X}$ à travers la notion d'association entre les colonnes i et j définie par $c(i \Rightarrow j, \mathbf{X}) = \frac{\mathbf{x}_i^T \mathbf{x}_j}{\mathbf{x}_i^T \mathbf{x}_i}$. Ces associations sont utilisées pour former des vecteurs candidats d'une base qui forment les colonnes de $\mathbf{H}$, par une sélection « gloutonne ». Les principales étapes d'ASSO sont résumées dans la table algorithmique 3.

Algorithm 3 : Asso

Entrées : $\mathbf{X}$, K, $\tau \in]0,1]$, w^+, w^-
Sorties : $\mathbf{W}$, $\mathbf{H}$
PAS 1 : Calcul de la matrice d'association $\mathbf{A}$
  for $i = 1, \cdots, M$ do
    for $j = 1, \cdots, M$ do
      $\mathbf{A}_{ij} = (c(j \Rightarrow i, \mathbf{X}) \geq \tau)$
    end
  end
PAS 2 : Construire de manière itérative $\mathbf{W}$ et $\mathbf{H}$
  $\mathbf{H} \leftarrow [\,], \mathbf{W} \leftarrow [\,]$
  for $k = 1, \cdots, K$ do
    $\{\mathbf{w}, \mathbf{h}\} = cover\,([\mathbf{H}\ \mathbf{h}], [\mathbf{W}\ \mathbf{w}], \mathbf{X}, w^+, w^-)$
    $\mathbf{H} \leftarrow [\mathbf{H}\ \mathbf{h}]$
    $\mathbf{W} \leftarrow [\mathbf{W}\ \mathbf{w}]$
  end

Le paramètre τ contrôle le niveau de confiance qui permet d'inclure un attribut au candidat du vecteur de base. Ainsi, à des valeurs différentes de τ on obtient différentes matrices d'association $\mathbf{A}$ qui sont calculées comme dans le PAS 1 de l'algorithme 3. Cela implique donc, différentes valeurs de la matrice $\mathbf{H}$, puisque les valeurs de $\mathbf{H}$ dépendent fortement de celles de la matrice d'association $\mathbf{A}$. Ainsi, l'exemple ci-dessous nous permet d'illustrer l'importance du paramètre τ dans cet algorithme.

Exemple 1.3.2. Considérons une matrice binaire de taille 3×3 :

$$\mathbf{X} = \begin{pmatrix} \mathbf{1} & \mathbf{1} & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} \end{pmatrix}; \quad \mathbf{X}(i \Rightarrow j) = \begin{pmatrix} 1 & 1 & 1/2 \\ 2/3 & 1 & 2/3 \\ 1/2 & 1 & 1 \end{pmatrix}$$

$$\mathbf{A}_1 = \underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}}_{\tau=1/2}, \quad \mathbf{A}_2 = \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}}_{\tau=2/3}, \quad \mathbf{A}_3 = \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}}_{\tau=1}$$

On peut bien voir sur cet exemple que l'algorithme est très sensible aux différentes valeurs du paramètre τ. La fonction de couverture (ou *cover*) est définie comme :

$$cover(\mathbf{H}, \mathbf{W}, \mathbf{X}, w^+, w^-) = w^+ \mid \{(i,j) : \mathbf{X}_{ij} = 1, (\mathbf{W} \diamond \mathbf{H})_{ij} = 1\} \mid \\ -w^- \mid \{(i,j) : \mathbf{X}_{ij} = 0, (\mathbf{W} \diamond \mathbf{H})_{ij} = 1\} \mid,$$

et peut être interprétée comme le « profit » de décrire $\mathbf{X}$ par $\mathbf{W}$ et $\mathbf{H}$.

L'algorithme FC (pour « *Formal Concept analysis* » en anglais) : il s'agit d'un autre algorithme de type « glouton » proposé dans [78]. Il est basé sur l'analyse de concepts formels (ou *Formal Concepts Analysis* (FCA) en anglais) ; un concept formel définit une relation binaire entre un ensemble d'objets et un ensemble d'attributs. Ici, la matrice des observations, $\mathbf{X}$ est interprétée comme une matrice objets-attributs, $\mathbf{W}$ et $\mathbf{H}$ sont interprétées respectivement comme des matrices de facteurs objets et de facteurs attributs. L'algorithme proposé est fondé sur un théorème indiquant que les décompositions optimales de $\mathbf{X}$ (celles ayant un rang K minimum) sont celles où les facteurs sont des concepts formels dans le sens de l'analyse de concepts formels.

Une reformulation « signal » de l'algorithme FC indique qu'il consiste à soustraire itérativement de $\mathbf{X}$, les sources (termes de rang 1), $\mathbf{X}^{(k)}$, dont les supports ont un chevauchement maximal avec le support de $\mathbf{X}$. Deux algorithmes sont proposés dans [78]. Le premier commence par calculer tous les concepts formels puis, à chaque itération, sélectionne celui maximisant l'intersection avec $\mathbf{X}$. Cet algorithme donne de bons résultats mais il est inefficace pour les grands ensembles de données. Le deuxième algorithme (qui sera utilisé pour la comparaison avec les algorithmes que nous proposons dans ce manuscrit), utilise une stratégie plus efficace et construit progressivement les facteurs en ajoutant séquentiellement des « colonnes prometteuses ».

Toutes les approches « gloutonnes » présentées dans cette section utilisent des procédures de déflation pour estimer de manière itérative les sources. Cette technique, tout comme celle de la relaxation réelle non-négative, garantit une récupération ou reconstruction correcte des « vraies » sources uniquement dans le cas non corrélé. Lorsque les sources sont corrélées, leur capacité à récupérer les « vraies » sources, dépendra fortement du degré de corrélation, ainsi que des paramètres de l'algorithme.

1.4 Décomposition CP booléennes

1.4.1 Modèle BCP

Le problème considéré est donc la décomposition CP booléenne (DCPB) d'un tenseur à valeurs binaires $\mathcal{X}$ d'ordre 3 selon le modèle :

$$\mathcal{X} = \bigvee_{k=1}^{K} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k) = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!] . \tag{1.29}$$

La DCPB des tenseurs binaires a été utilisée dans un certain nombres d'applications. Ainsi, dans [44], les auteurs considèrent des données d'entrée contenant des informations sur les étudiants et leur performance dans divers cours. La DCPB dans ce cas, peut aider à révéler des facteurs importants décrivant les compétences communes des élèves dans diverses conditions. La DCPB a aussi été appliquée dans les systèmes de recommandation ([94], [95]) pour lesquels on a des utilisateurs qui donnent des avis sur des films ou des articles sur une journée, un mois ou une année. L'idée dans cette application est de retrouver les utilisateurs qui regardent les mêmes films ou qui aiment les mêmes articles sur un mois ou une année ce qui permettra de suggérer à certains utilisateurs d'un groupe, les choses aimées par les autres membres de ce groupe.

Les algorithmes de DCPB qui ont été développés sont essentiellement des approches gloutonnes : ils sont présentés dans la section 1.4.2. Cependant, nous reprenons un exemple de [44] qui permet de bien illustrer l'intérêt d'une approche tensorielle.

Exemple 1.4.1. Un exemple d'application de la DCPB

	AL						CA						CI						BD						RE					
	co	cr	as	co	au	mo	co	cr	as	co	au	mo	co	cr	as	co	au	mo	co	cr	as	co	au	mo	co	cr	as	co	au	mo
a	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1
b	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1
c	1	1	1	1	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0
d	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1
e	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	1	1	0	0	0	0	1	1	0	0	1	1
f	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	0	0	1	1
g	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	1	1
h	0	0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

TABLE 1.1 – Tableau des compétences ou qualités des élèves dans divers cours

Le tableau 1.1 contient des informations sur les élèves et leurs performances dans divers cours. Dans ce tableau, {a, b, c, d, e, f, g et h} représente les différents étudiants, {co, cr, as, co, au, mo} est l'ensemble des qualités de ces étudiants : communication, créativité, assiduité, concentration, autonomie et motivation et {AL, CA, CI, BD, RE} représente les différents cours passés par les étudiants : algorithmes, calculs, circuits, bases de données

et réseaux. Ces données peuvent être représentées par un tenseur $\mathcal{X}$ binaire d'ordre 3 de dimension $8 \times 6 \times 5$ qui est illustré dans la figure ci-après :

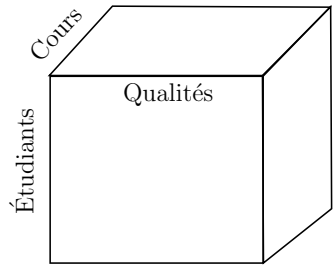


FIGURE 1.7 – Représentation du tenseur $\mathcal{X}$.

On peut reconstruire exactement la matrice de données observées, $\mathcal{X}$, avec un rang booléen $K = 3$. On obtient ainsi comme matrices $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$:

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

À partir des matrices $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$, on peut représenter sur la figure 1.8 les différents tenseurs de rang 1 estimés pour extraire les comportements communs entre les différents étudiants. Chaque ligne de cette figure représente le depliement des différentes sources (ou tenseurs de rang 1) selon le mode représentant les différents cours.

Ainsi, la première source nous permet de dire que les étudiants a, c, d, f et h sont assidus, concentrés, et motivés en algorithmes, calculs et en bases de données, la deuxième source nous dit que les étudiants b, c, e et f ont une bonne communication et sont créatifs en algorithmes, circuits, bases de données et réseaux, et enfin, la troisième source nous permet de dire que les étudiants a, b, d, e, f, g ont une bonne communication et sont créatifs, autonomes et motivés en algorithmes, circuits et réseaux.

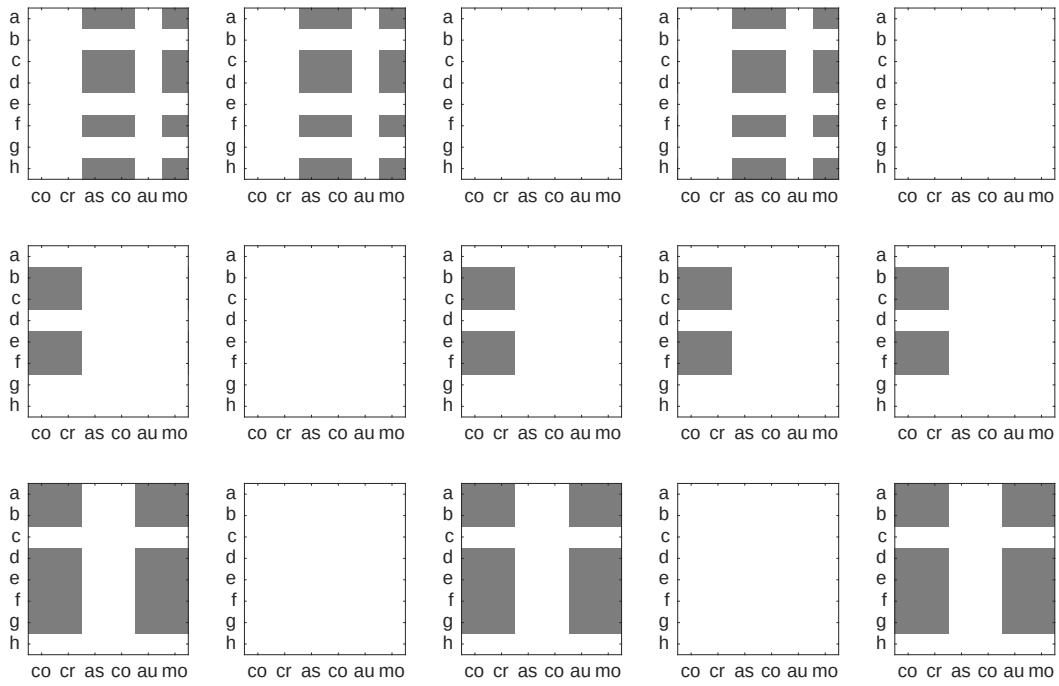


FIGURE 1.8 – Représentation des différents tenseurs de rang 1 estimés

1.4.2 Algorithmes de décomposition CP booléenne

Le problème de la DCPB (1.29) est d'estimer à partir d'un tenseur binaire $\mathcal{X}$, trois matrices $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ à valeurs binaires qui permettent de minimiser la fonction de coût suivante :

$$\{\mathbf{W}, \mathbf{H}, \mathbf{V}\} = \arg \min_{\mathbf{W},\mathbf{H},\mathbf{V}} \left\| \mathcal{X} - \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k \right\|_2^2. \tag{1.30}$$

Les expressions du modèle BCP correspondant aux 3 modes de dépliement du tenseur binaire $\mathcal{X}$ sont données par :

$$\mathbf{X}_{(1)} = \mathbf{W} \diamond (\mathbf{V} \odot \mathbf{H})^T, \tag{1.31}$$

$$\mathbf{X}_{(2)} = \mathbf{H} \diamond (\mathbf{V} \odot \mathbf{W})^T, \tag{1.32}$$

$$\mathbf{X}_{(3)} = \mathbf{V} \diamond (\mathbf{H} \odot \mathbf{W})^T. \tag{1.33}$$

($\diamond$) représente le produit booléen matriciel, et ($\odot$) représente le produit de Khatri-Rao ([96], [97]). Dans la littérature, deux algorithmes de type « glouton » ont été proposés pour résoudre ce problème par Miettinen [45] et Belohlavek et Vychodil [44]. Les deux approches reposent sur une approche de type ALS (Alternating Least Squares en anglais) pour résoudre le problème de la DCPB. Ainsi, la méthode ALS procède à une estimation alternée de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$, *i.e.*, en fixant à chaque fois deux matrices et en estimant la troisième. L'algorithme BCP-ALS pour (*Boolean CP Alternating least-squares* en anglais) est donné dans la table algorithmique 4.

Algorithm 4 : BCP-ALS algorithm

Entrées : $\mathcal{X}$, K
Sorties : $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$
PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$
$\mathbf{W} \leftarrow Asso\left(\mathbf{X}_{(1)}, K\right)$
$\mathbf{H} \leftarrow Asso\left(\mathbf{X}_{(2)}, K\right)$
$\mathbf{V} \leftarrow Asso\left(\mathbf{X}_{(3)}, K\right)$
PAS 2 : Mise à jour de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$
Répéter
$\mathbf{W} \leftarrow UpdateFactor\left(\mathbf{X}_{(1)}, \mathbf{W}, (\mathbf{V} \odot \mathbf{H})\right)$
$\mathbf{H} \leftarrow UpdateFactor\left(\mathbf{X}_{(2)}, \mathbf{H}, (\mathbf{V} \odot \mathbf{W})\right)$
$\mathbf{V} \leftarrow UpdateFactor\left(\mathbf{X}_{(3)}, \mathbf{V}, (\mathbf{H} \odot \mathbf{W})\right)$
Jusqu'à convergence

Dans la première partie de l'algorithme, on utilise l'algorithme « Asso » (voir algorithme 3) pour initialiser des matrices $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$. Selon [83], c'est préférable à une initialisation aléatoire qui converge généralement très rapidement vers un optimum local, et est donc incapable d'échapper à de mauvaises solutions initiales. Cependant, comme cette initialisation est déterministe, si l'algorithme « Asso » fournit une solution initiale inadéquate, l'algorithme BCP-ALS retournera une solution incorrecte.
Dans la deuxième partie de l'algorithme (PAS 2), les facteurs sont mis à jour en utilisant une heuristique « gloutonne ». Par exemple pour la mise à jour de $\mathbf{W}$, considérons chaque colonne de $\mathbf{X}_{(1)}$ séparément, et soit c une fonction telle que : $c(i) = 0$ si $(\mathbf{w} \diamond (\mathbf{V} \odot \mathbf{H}))_i = 1$, et $\mathbf{w}$, la $i^{\text{ème}}$ ligne de $\mathbf{W}$ avant la mise à jour. Nous pouvons maintenant faire la mise à jour des éléments de $\mathbf{w}$ tel que $cover(\mathbf{x}, (\mathbf{w} \diamond (\mathbf{V} \odot \mathbf{H})), \mathbf{c})$ est maximal, avec la fonction *cover* définie telle que :

$$cover(\mathbf{x}, \mathbf{z}, \mathbf{c}) = \sum_i \left(c(i)[x_i = 1] - c(i)[x_i = 0]\right)[z_i = 1].$$

En répétant cela pour chaque ligne de $\mathbf{W}$, nous obtenons la matrice de mise à jour (avec $\mathbf{H}$ et $\mathbf{V}$ fixes, nous pouvons mettre à jour chaque ligne de $\mathbf{W}$ indépendamment). Cette procédure est appelée *UpdateFactor*. Avec un tenseur de dimension $N_1 \times N_2 \times N_3$, l'algorithme BCP-ALS converge vers l'optimum local en $N_1 N_2 N_3$ étapes dans la plupart du temps, car il est garanti que chaque étape réduit les erreurs d'au moins 1 unité.

Dans le second algorithme de type « glouton », proposé dans [44], Belohlavek et Vychodil résolvent le problème de la DCPB (1.30) en utilisant les concepts formels ([98], [99]). Ainsi, ils proposent une décomposition de $\mathcal{X}$ qui consiste à trouver le minimum de concepts formels qui couvrent le support de $\mathcal{X}$. Cela revient à trouver une DCPB du tenseur binaire $\mathcal{X}$ en trois matrices binaires, une matrice de facteur d'objet $\mathbf{W}$, une matrice de facteur d'attribut $\mathbf{H}$ et une matrice de facteur de condition $\mathbf{V}$; le nombre de facteurs est le rang booléen de la DCPB. Si on revient à l'exemple 1.4.1, les objets correspondent

aux élèves, les attributs aux qualités des élèves et les conditions aux cours passés par les élèves.

1.5 Conclusions

Dans ce chapitre, nous avons tout d'abord introduit les notions et opérations sur les tenseurs et les décompositions tensorielles. Ensuite, nous avons étudié les algorithmes existants de factorisation des matrices à valeurs binaires. Puisque le problème de la FBMB est un problème NP-complet, des stratégies telles que la relaxation réelle du problème de la FBMB sur l'orthant non-négatif et l'approche de type glouton ont été proposées pour contourner ce problème. Cependant, ces algorithmes ne donnent pas de bons résultats dans le cas de sources corrélées, car, dans le cas des algorithmes reposant sur la relaxation réelle du problème de la FBMB, le produit matriciel classique utilisé n'est pas équivalent au produit matriciel booléen, et dans le cas de l'approche de type glouton, les algorithmes cherchent plutôt des sources décorrélées.
Nous avons aussi présenté et étudié les décompositions CP booléennes existantes . Ainsi, nous avons remarqué d'une part qu'il n'y a pas beaucoup d'algorithmes de décompositions de tenseurs binaires existants et la plupart de ces algorithmes sont de type glouton.
Ainsi, comme la plupart des algorithmes existants de la FBMB ne résolvent pas vraiment le problème de la FBMB car les modèles sur lesquels reposent ces algorithmes ne sont pas équivalents au modèle de la FBMB dans le cas de sources corrélées. Nous proposerons dans la section suivante, un nouveau regard sur la décomposition de la matrice binaire qui est équivalent au vrai modèle de la FBMB, même dans le cas corrélé.

Chapitre 2

FBMB par modèle de mélange Post Non-Linéaire (PNL)

Sommaire

2.1 Principe de la FBMB par modèle de mélange post non-linéaire **44**
2.2 L'algorithme PNL-PF **45**
2.3 Condition d'identifiabilité du modèle FBMB **47**
2.4 L'algorithme C-PNL-PF **50**
2.5 Simulations Numériques **53**
2.5.1 Performances dans le cas exact 53
2.5.2 Performances dans le cas d'un bruit binaire 54
2.6 La FBMB appliquée aux données Netflix **55**
2.6.1 Données Netflix 55
2.6.2 Capacité explicative des algorithmes FBMB 58
2.6.3 Capacité de prédiction des préférences des utilisateurs 58
2.7 Conclusion sur la FMBM **63**

2.1 Principe de la FBMB par modèle de mélange post non-linéaire

Dans cette section, nous proposons une nouvelle formulation du problème de FBMB basée sur un modèle de mélange *post non-linéaire* [100]. Dans ce nouveau modèle, nous préservons l'idée de [92], qui remplace le produit booléen matriciel ($\diamond$) par le produit réel classique avec des contraintes de binarité sur $\mathbf{W}$ et $\mathbf{H}$ mais nous introduisons une fonction non-linéaire Φ. Cette fonction peut être toute fonction $\Phi(x)$, qui est égalé à 0 quand l'argument de cette fonction est inférieur ou égal à 0 ($x \leq 0$), et qui est égale à 1 quand l'argument de cette fonction est supérieur ou égal à 1 ($x \geq 1$). L'intérêt de ce modèle est qu'il est équivalent à (1.16). Il apparaît comme une relaxation réelle exacte de la FBMB et à ce titre permet d'envisager le développement de méthode de factorisation qui repose sur des techniques d'optimisation continue.

Nous avons choisi la fonction sigmoïde dans la figure 2.1, qui est continûment dérivable et permet de garantir les contraintes sur la fonction non-linéaire $\Phi(x)$ lorsque le paramètre γ qui permet de régler la pente de la sigmoïde, est suffisamment grand.

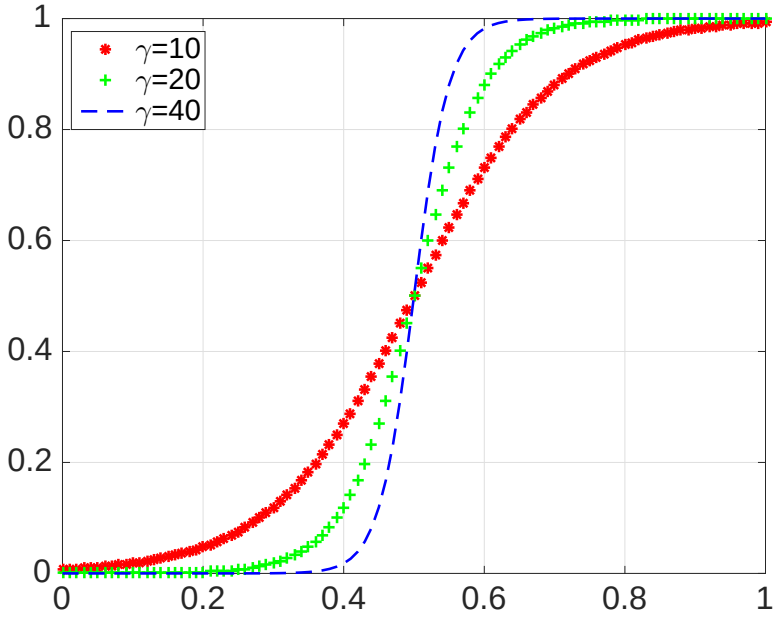


FIGURE 2.1 – Fonction sigmoïde $\Phi(x) = \frac{1}{1+e^{-\gamma(x-0.5)}}$.

Le modèle de mélange de la FBMB est donc donné par :

$$\mathbf{X} = \Phi\left(\mathbf{W}\mathbf{H}^T\right) \oplus \mathbf{N} \tag{2.1}$$

où Φ est appliquée élément à élément. Le problème inverse correspondant qu'on cherche

à résoudre peut être formulé comme suit :

$$\{\mathbf{W},\mathbf{H}\} = \arg\min_{\mathbf{W},\mathbf{H}\in\{0,1\}} \|\mathbf{X} - \Phi(\mathbf{W}\mathbf{H}^T)\|_F^2. \tag{2.2}$$

Ainsi, dans la prochaine section, nous allons étudier l'algorithme PNL-PF (pour *Post NonLinear Penalty Function* en anglais) qui résout (2.2).

2.2 L'algorithme PNL-PF

Pour résoudre (2.2), nous proposons l'algorithme PNL-PF (pour *Post NonLinear Penalty Function*) fondé sur la méthode de descente de gradient avec une mise à jour multiplicative, similaire à l'algorithme PF présenté dans la section 1.3.2. Cet algorithme minimise la fonction objective ci-après, $G(\mathbf{W},\mathbf{H})$, qui est similaire à la fonction (1.26), à laquelle nous avons intégré la fonction non linéaire Φ, pour garantir la binarité du produit matriciel $\mathbf{W}\mathbf{H}^T$.

$$G(\mathbf{W},\mathbf{H}) = \frac{1}{2}\sum_{i,j}\left(\mathbf{X}_{ij} - \Phi(\mathbf{W}\mathbf{H}^T)_{ij}\right)^2 + \frac{1}{2}\lambda\sum_{i,k}\left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \frac{1}{2}\lambda\sum_{j,k}\left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2. \tag{2.3}$$

Les étapes de l'algorithme proposé sont présentées dans la table algorithmique 5.

Algorithm 5 : Post NonLinear Penalty Function (PNL-PF)

Entrées : $\mathbf{X}$, K, Nb_{iter}, λ
Sorties : $\mathbf{W}$, $\mathbf{H}$
PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$ et $iter = 0$
PAS 2 : Normalisation
$\mathbf{W} \leftarrow \mathbf{W}\mathbf{D}_W^{-1/2}\mathbf{D}_H^{1/2}$
$\mathbf{H} \leftarrow \mathbf{H}\mathbf{D}_H^{-1/2}\mathbf{D}_W^{1/2}$
$iter \leftarrow iter + 1$
PAS 3 : Mise à jour de $\mathbf{W}$, $\mathbf{H}$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{\left(\mathbf{X}\frac{\partial}{\partial\mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 3\lambda\mathbf{H}_{jk}^2}{\left(\frac{\partial}{\partial\mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 2\lambda{\mathbf{H}_{jk}}^3 + \lambda\mathbf{H}_{jk}}$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik}\frac{\left(\mathbf{X}\frac{\partial}{\partial\mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik} + 3\lambda\mathbf{W}_{ik}^2}{\left(\frac{\partial}{\partial\mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik} + 2\lambda{\mathbf{W}_{ik}}^3 + \lambda\mathbf{W}_{ik}}$$

PAS 4 : Critère d'arrêt
if $iter \geq Nb_{iter}$ ou $G(\mathbf{W},\mathbf{H}) < \epsilon$ **then**
break
else
retourner au PAS 3
end if

Dans cet algorithme, l'initialisation et la normalisation se font comme dans l'algo-

rithme PF. La dérivation et les étapes pour obtenir les règles de mise à jour de $\mathbf{W}$ et $\mathbf{H}$ sont données dans l'annexe A.3.
Pour illustrer l'intérêt de cette approche, on considère quelques exemples simples et on compare les résultats de l'algorithme PNL-PF avec ceux fournis par les algorithmes de l'état de l'art 1.3.2. Lorsque les sources ne sont pas corrélées (Figure 2.2), tous les algorithmes estiment correctement $\mathbf{W}$ et $\mathbf{H}$. Cependant, lorsque les deux sources sont corrélées (Figure 2.3), seul l'algorithme PNL-PF donne de bons résultats. L'algorithme PNL-PF permet une reconstruction exacte de $\mathbf{X}$ avec 2 termes, alors que les algorithmes ASSO et FC sont incapables de produire une décomposition de rang 2, reproduisant exactement $\mathbf{X}$. Ils nécessitent une décomposition de rang 3 pour expliquer correctement les données.

L'exemple de la figure 2.4 illustre le comportement de notre algorithme (PNL-PF) dans le cas de 3 sources fortement corrélées. Nous pouvons observer que $\mathbf{H}$ est bien estimée et $\mathbf{X}$ est bien reconstruite bien que $\mathbf{W}$ ne soit pas estimée de façon correcte. Cela signifie qu'il existe un autre couple $(\bar{\mathbf{W}}, \bar{\mathbf{H}})$ différent du couple $(\mathbf{W}, \mathbf{H})$ qui permet de reconstruire correctement $\mathbf{X}$, *i.e.*, $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T = \bar{\mathbf{W}} \diamond \bar{\mathbf{H}}^T$. Le modèle n'est donc pas identifiable. L'objet de la prochaine section est d'étudier l'identifiabilité de la FBMB et de donner une condition nécessaire et suffisante d'identifiabilité pour le modèle $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$.

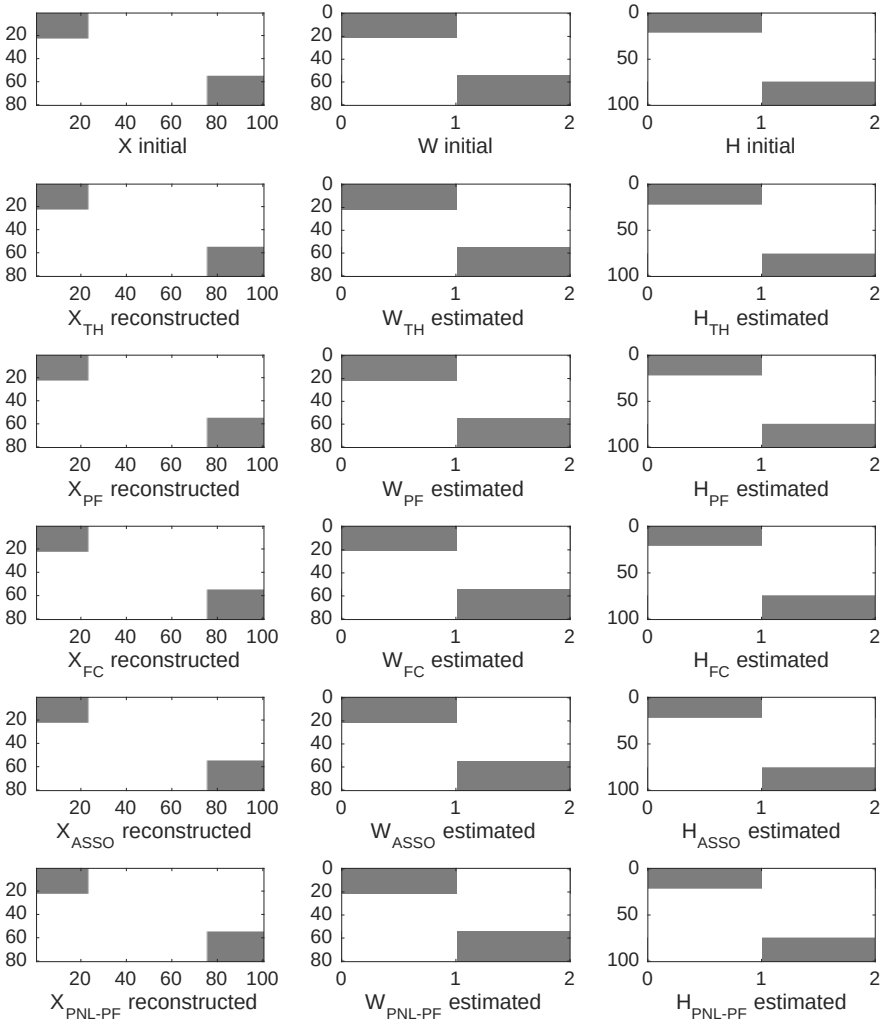


FIGURE 2.2 – Cas de deux sources non corrélées ($N = 80$, $M = 100$, $\tau = 1$(ASSO)).

2.3 Condition d'identifiabilité du modèle FBMB

Dans cette thèse, l'identifiabilité du modèle FBMB est comprise comme l'unicité de la décomposition $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$, *i.e.*, s'il existe un autre couple de matrices $(\bar{\mathbf{W}}, \bar{\mathbf{H}})$ qui vérifie :

$$\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T = \bar{\mathbf{W}} \diamond \bar{\mathbf{H}}^T, \tag{2.4}$$

alors les couples $(\mathbf{W}, \mathbf{H})$ et $(\bar{\mathbf{W}}, \bar{\mathbf{H}})$ sont les mêmes, à une permutation des colonnes de $\mathbf{W}$ et $\mathbf{H}$ près. Il faut noter que, dans le cas binaire, l'indétermination d'échelle, inhérente dans la factorisation des matrices réelles, n'est pas présente.

Définition 2.3.1. *(Identifiabilité partielle)*
Le modèle (2.2) *est partiellement identifiable si une ou plusieurs colonnes de* $\mathbf{W}$ *et (ou)* $\mathbf{H}$ *peuvent être estimées de façon unique à partir de* $\mathbf{X}$.

Notons $supp(\mathbf{x}) = \{i, \mathbf{x}_i \neq 0\}$ le support du vecteur $\mathbf{x}$ et $supp(\mathbf{X}) = \{(i,j), \mathbf{X}_{ij} \neq 0\}$,

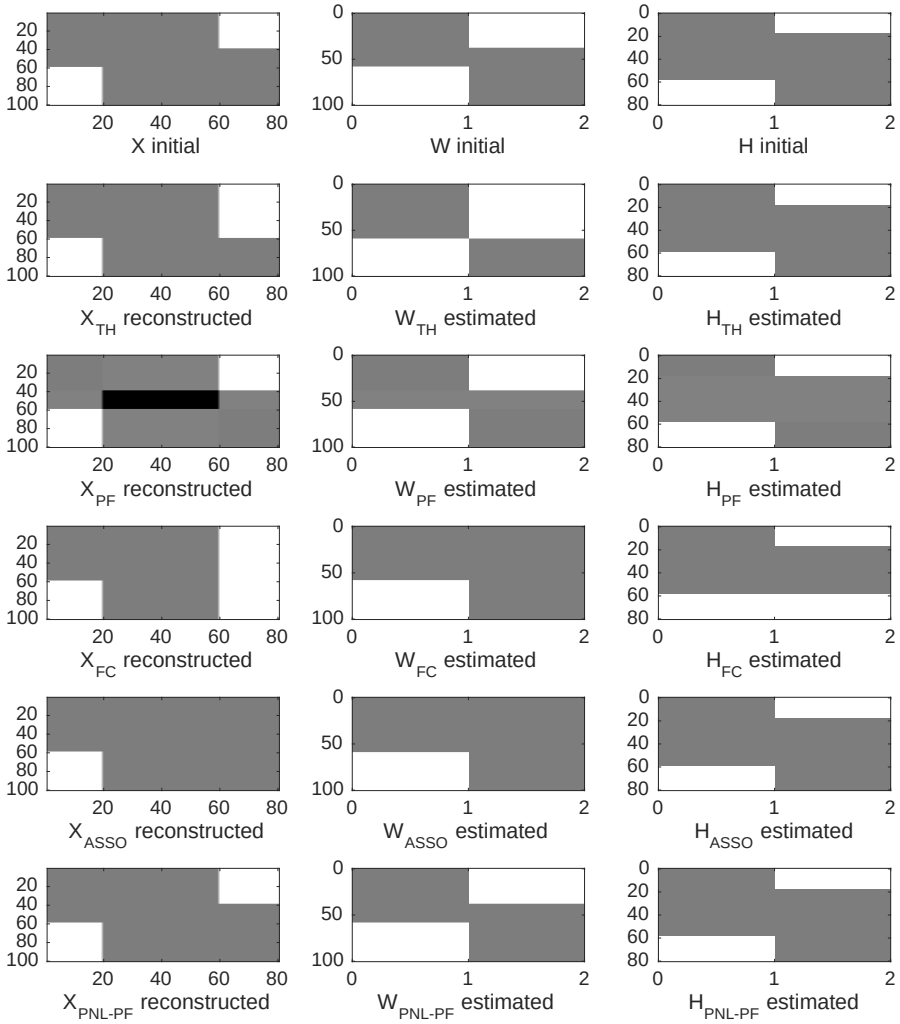


FIGURE 2.3 – Cas de deux sources corrélées ($N = 100$, $M = 80$, $\tau = 1$ (ASSO)).

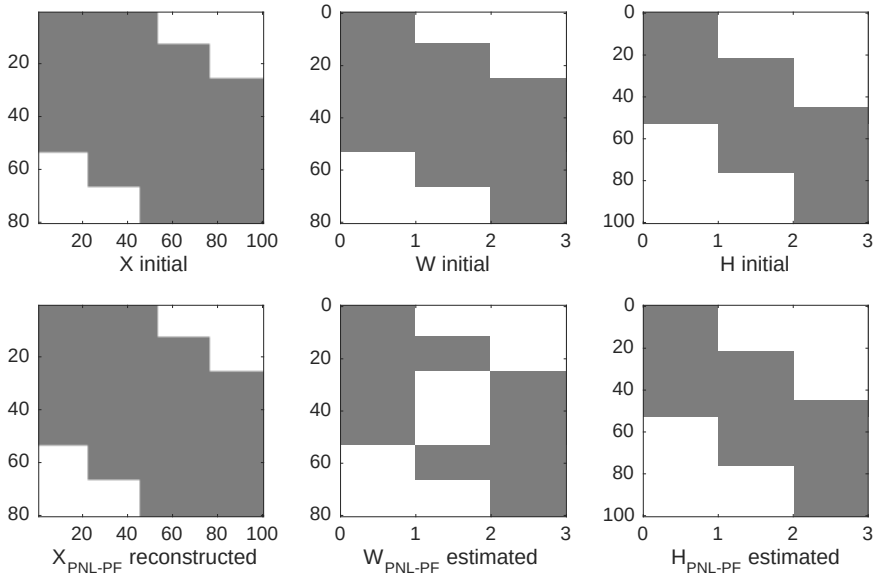


FIGURE 2.4 – Cas de trois sources corrélées ($N = 80$, $M = 100$).

le support de la matrice $\mathbf{X}$.

Théorème 2.3.1 (Identifiabilité partielle)**.** *La $\ell^{ième}$ colonne de $\mathbf{W}$, $\mathbf{w}_\ell$, peut être estimée de façon unique à partir de $\mathbf{X}$ ssi : $\forall\, n \in \{1, \ldots, N\}$ tel que $\mathbf{w}_\ell(n) \neq 0$, on a :*

$$supp\left(\mathbf{e}_n \mathbf{h}_\ell^T\right) \not\subseteq \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}),$$

avec $\mathbf{e}_n = [0, \cdots, 1, \cdots, 0]^T$, le $n^{ième}$ vecteur de la base canonique de $\mathbb{R}^N$.

Démonstration. L'équation $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$ peut être réécrite comme :

$$\mathbf{X} = \bigvee_{k=1}^{K} (\mathbf{w}_k \mathbf{h}_k^T) = \bigvee_{k=1}^{K} \mathbf{X}^{(k)} = \bigvee_{k\neq\ell}^{K} \mathbf{X}^{(k)} \vee \left(\mathbf{w}_\ell \mathbf{h}_\ell^T\right). \tag{2.5}$$

Supposons qu'il $\exists\, n \in \{1, \cdots, N\}$ avec $\mathbf{w}_\ell(n) \neq 0$ tel que :

$$supp\left(\mathbf{e}_n \mathbf{h}_\ell^T\right) \subseteq \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}).$$

Alors, à partir de (2.5), nous avons :
$supp(\mathbf{X}) = \bigcup_{k=1}^{K} supp(\mathbf{X}^{(k)}) = \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}) \cup supp(\mathbf{w}_\ell \mathbf{h}_\ell^T)$.
Comme $\mathbf{w}_\ell(n) \neq 0$, alors nous avons : $\mathbf{w}_\ell = \mathbf{w}_\ell \oplus \mathbf{e}_n \vee \mathbf{e}_n$, où $(\oplus)$ et $(\vee)$ représentent respectivement les opérateurs logiques (OU-Exclusif) et (OU). Ainsi, nous pouvons écrire :

$$supp(\mathbf{X}) = \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}) \cup supp((\mathbf{w}_\ell \oplus \mathbf{e}_n \vee \mathbf{e}_n)\, \mathbf{h}_\ell^T)$$

$$= \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}) \cup supp\left((\mathbf{w}_\ell \oplus \mathbf{e}_n)\mathbf{h}_\ell^T\right) \cup supp\left(\mathbf{e}_n \mathbf{h}_\ell^T\right).$$

Comme $supp\left(\mathbf{e}_n \mathbf{h}_\ell^T\right) \subseteq \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)})$, nous avons :
$supp(\mathbf{X}) = \bigcup_{k\neq\ell}^{K} supp(\mathbf{X}^{(k)}) \cup supp\left((\mathbf{w}_\ell \oplus \mathbf{e}_n)\mathbf{h}_\ell^T\right) \quad \Leftrightarrow \mathbf{X} = \bigvee_{k\neq\ell}^{K} \mathbf{X}^{(k)} \vee (\mathbf{w}_\ell \oplus \mathbf{e}_n)\mathbf{h}_\ell^T$.
Comme $\mathbf{w}_\ell \neq (\mathbf{w}_\ell \oplus \mathbf{e}_n)$, cela signifie qu'il existe un autre $\bar{\mathbf{w}}_\ell = \mathbf{w}_\ell \oplus \mathbf{e}_n \neq \mathbf{w}_\ell$ qui satisfait (2.4). Donc, $\mathbf{w}_\ell$ ne peut pas être estimé de façon unique. Une condition similaire peut aussi être dérivée pour les colonnes de $\mathbf{H}$.

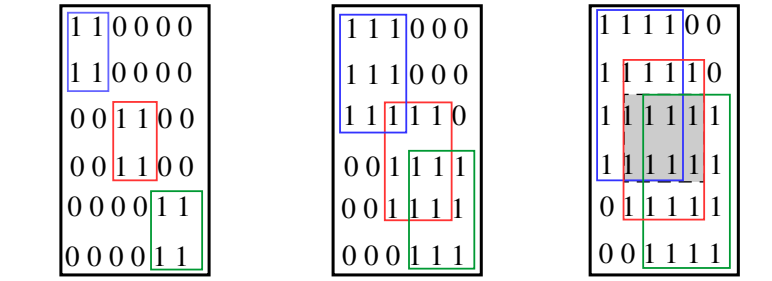
□

Intuitivement, le théorème 2.3.1 indique qu'un composant $\mathbf{X}^{(\ell)} = \mathbf{w}_\ell \mathbf{h}_\ell^T$ peut être estimé de façon unique à partir de $\mathbf{X}$ si aucune de ses colonnes et aucune de ses lignes n'a le support inclus dans l'union des supports de toutes les autres sources.
À partir du théorème 2.3.1, nous déduisons directement le corollaire suivant qui donne la condition nécessaire et suffisante pour l'unicité de la décomposition booléenne de matrices binaires.

Corollaire 2.3.1 (Identifiabilité du modèle FBMB). *Le modèle* $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$ *est identifiable ssi :* $\forall\, k = 1, \cdots, K$, $\mathbf{w}_k$ *et* $\mathbf{h}_k$ *peuvent être estimés de façon unique à partir de* $\mathbf{X}$, i.e., *le théorème* (2.3.1) *est vérifié pour toutes les colonnes de* $\mathbf{W}$ *et* $\mathbf{H}$.

Nous allons illustrer ce résultat sur l'identifibilité du modèle FBMB par l'exemple suivant :

Exemple 2.3.1. *:* Considérons 3 matrices binaires de taille 6×6 et leur 3 termes de rang 1, *i.e.,* $\mathbf{X} = \mathbf{X}^{(1)} \vee \mathbf{X}^{(2)} \vee \mathbf{X}^{(3)}$. Les supports de $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \mathbf{X}^{(3)}$ sont indiqués par les trois rectangulaires de couleurs différentes.



(a) Sources non-corrélées (Unique) (b) Sources corrélées (Unique) (c) Sources corrélées (Non unique)

A travers cet exemple, il apparaît que le problème inverse (2.2) est en général, mal posé car le modèle de la FBMB n'est pas toujours identifiable. Pour pallier ce problème, nous devons rajouter un terme de régularisation. Dans la prochaine section, nous introduisons l'algorithme C-PNL-PF (pour *Constraint Post NonLinear Penalty Function* en anglais) qui intègre une contrainte supplémentaire favorisant la reconstruction de sources de support maximum.

2.4 L'algorithme C-PNL-PF

Pour régulariser la décomposition PNL-PF, nous avons choisi de maximiser le support des termes de rang 1 ($\mathbf{X}^{(k)} = \mathbf{w}_k \mathbf{h}_k^T$). Le choix de cette régularisation peut sembler arbitraire mais il peut être justifié par le fait qu'il garantit la stabilité de la solution dans un scénario où la taille du support des différentes sources ou termes de rang 1 évolue lentement. Cela pourrait être par exemple, le cas pour les applications « en ligne », où nous visons à adapter en temps réel les facteurs $\mathbf{W}$ et $\mathbf{H}$, pour suivre les changements dans la matrice de données $\mathbf{X}$.

Pour illustrer ce point, considérons l'exemple suivant d'une FBMB avec 3 sources non corrélées, comme illustré sur la figure 2.5. Dans ce cas, la décomposition est unique. Supposons ensuite que la corrélation entre les sources 1 et 3 augmente progressivement jusqu'à ce que leurs supports se chevauchent (figure 2.6). La décomposition reste unique, et l'algorithme PNL-PF peut séparer les sources avec succès. Si nous continuons à augmenter la taille du support des sources 1 et 3, la décomposition devient non-unique. La figure 2.7 illustre trois décompositions admissibles pour ce cas. On peut voir que la décomposition

qui préserve la continuité de la solution (la dernière) est celle qui maximise les supports des termes de rang 1, ce qui justifie la contrainte. Cependant, en fonction de l'application visée, cette contrainte pourrait être remplacée par une autre plus adaptée à la nature particulière du phénomène étudié.

Ainsi, pour imposer cette contrainte de support maximum à la décomposition, nous modifions la fonction coût (2.3) en incluant un terme supplémentaire qui permet de mesurer la taille du support des termes de rang 1. Le problème d'optimisation s'écrit alors :

$$L(\mathbf{W},\mathbf{H}) = \frac{1}{2}\sum_{i,j}\left(\mathbf{X}_{ij} - \Phi(\mathbf{W}\mathbf{H}^T)_{ij}\right)^2 + \frac{1}{2}\lambda\sum_{j,k}\left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2 + \frac{1}{2}\lambda\sum_{i,k}\left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \beta\frac{1}{\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)} \tag{2.6}$$

L'algorithme PNL-PF avec la contrainte de support maximal appelé C-PNL-PF est présenté dans la table algorithmique 6.

Algorithm 6 : Constraint Post NonLinear Penalty Function (C-PNL-PF)

Entrées : $\mathbf{X}$, K, Nb_{iter}, λ, β
Sorties : $\mathbf{W}$, $\mathbf{H}$
PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$ et $iter = 0$
PAS 2 : Normalisation

$$\mathbf{W} \leftarrow \mathbf{W}\mathbf{D}_W^{-1/2}\mathbf{D}_H^{1/2}$$
$$\mathbf{H} \leftarrow \mathbf{H}\mathbf{D}_H^{-1/2}\mathbf{D}_W^{1/2}$$

PAS 3 : Mise à jour de $\mathbf{W}$, $\mathbf{H}$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{\left(\mathbf{X}\frac{\partial}{\partial\mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 3\lambda\mathbf{H}_{jk}^2 + \beta\frac{\sum_k\left(\sum_i\mathbf{W}_{ik}\right)}{\left(\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)\right)^2}}{\left(\frac{\partial}{\partial\mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 2\lambda{\mathbf{H}_{jk}}^3 + \lambda\mathbf{H}_{jk}}$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik}\frac{\left(\mathbf{X}\frac{\partial}{\partial\mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik} + 3\lambda\mathbf{W}_{ik}^2 + \beta\frac{\sum_k\left(\sum_j\mathbf{H}_{kj}\right)}{\left(\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)\right)^2}}{\left(\frac{\partial}{\partial\mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 2\lambda{\mathbf{W}_{ik}}^3 + \lambda\mathbf{W}_{ik}}$$

$iter \leftarrow iter + 1$
PAS 4 : Critère d'arrêt
if $iter \geq Nb_{iter}$ ou $L(\mathbf{W},\mathbf{H}) < \epsilon$ **then**
  break
else
  retourner au PAS 3
end if

Les règles de mise à jour dans le pas 2 sont obtenues de la même manière que pour l'algorithme PNL-PF.

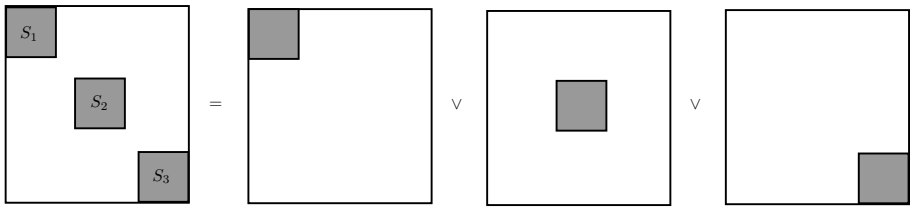


FIGURE 2.5 – Décomposition de 3 sources non-corrélées.

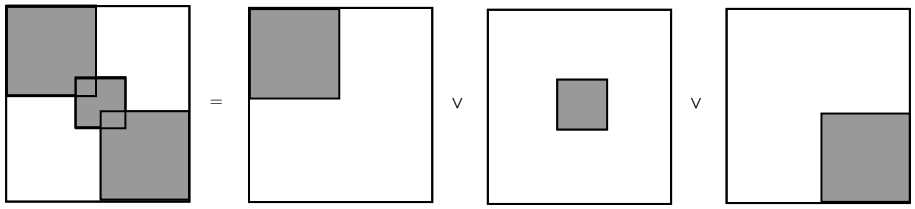


FIGURE 2.6 – Unicité de la décomposition de 3 sources corrélées.

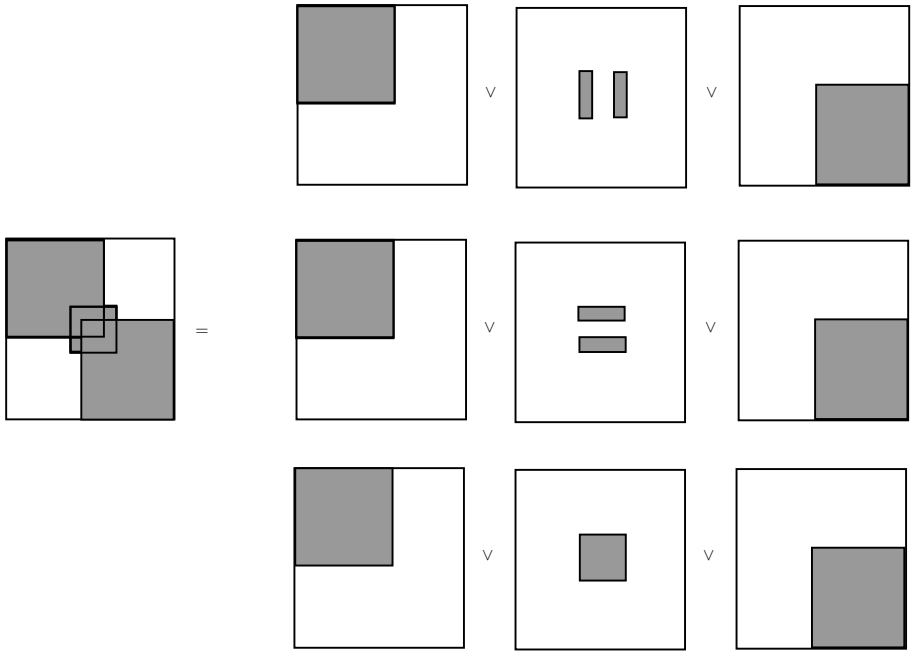


FIGURE 2.7 – Non-unicité de la décomposition de 3 sources corrélées.

La figure 2.8 compare les résultats des algorithmes PNL-PF et C-PNL-PF dans le cas de trois sources corrélées qui ne satisfont pas les conditions d'unicité du théorème 2.3.1. Nous pouvons observer que **W** et **H** sont correctement estimées avec C-PNL-PF, grâce à la contrainte de support maximum, ce qui n'est pas le cas pour l'algorithme PNL-PF.

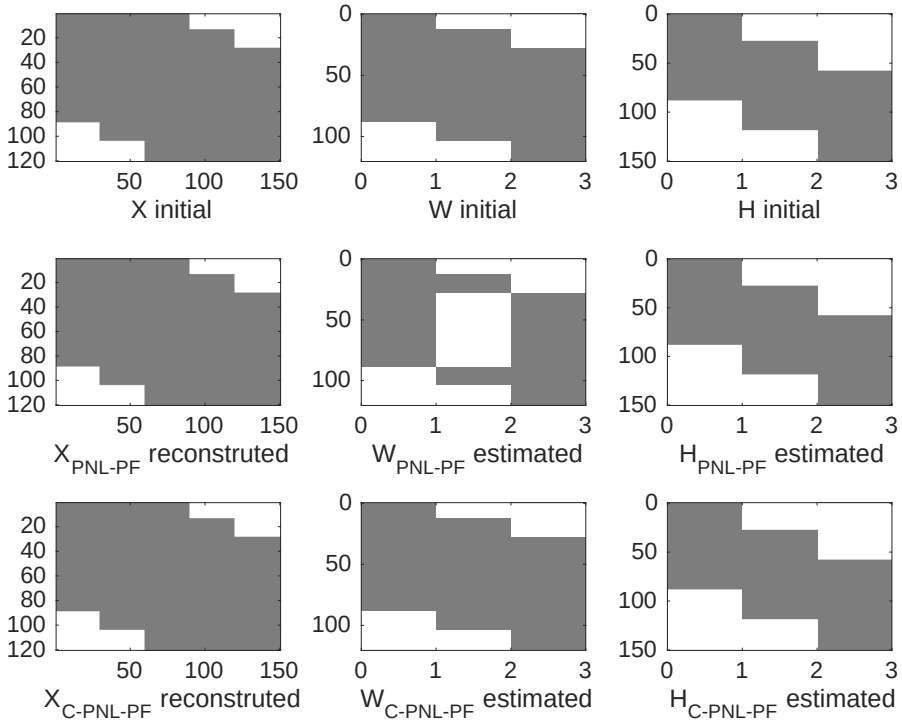


FIGURE 2.8 – Comparaison des algorithmes PNL-PF et C-PNL-PF pour une décomposition non-unique de 3 sources ($N = 80$, $M = 100$, $\beta = 5 \times 10^5$).

2.5 Simulations Numériques

Dans cette section, nous fournissons une analyse plus détaillée des performances des algorithmes PNL-PF et C-PNL-PF proposés, par rapport à d'autres approches de factorisation binaire, à travers des simulations numériques. Les performances sont étudiées dans le cas d'une factorisation exacte, puis dans le cas d'une factorisation bruitée.

2.5.1 Performances dans le cas exact

Nous commençons par évaluer les performances statistiques de ces algorithmes pour des problèmes de factorisation exacte. Nous simulons une matrice de données de 120×150 de rang 5 ; chaque colonne de **W** et **H** suit une loi de Bernoulli avec le paramètre p (%) représentant la probabilité des valeurs non nulles dans chaque colonne **W** et **H**. La qualité de l'estimation est évaluée par le taux d'erreur d'estimation des colonnes de **W** et **H** (moyenne [min, max]) sur 50 réalisations. Lorsque le paramètre de Bernoulli est petit, il y a une forte probabilité que la corrélation (chevauchement) entre les sources soit faible.

Lorsque p augmente, la corrélation des sources augmente. Les résultats indiqués dans le tableau 2.1 correspondent à $p = 25\%$, *i.e.*, à des sources binaires faiblement corrélées. Dans ce cas, les algorithmes (TH, PF, ASSO) donnent un faible taux d'erreur ($< 2\%$) avec un avantage à PF ; PNL-PF et C-PNL-PF donnent un taux d'erreur égal à 0%. L'algorithme ASSO donne un taux d'erreur nul sur $\mathbf{H}$ mais n'estime pas correctement $\mathbf{W}$ (le taux d'erreur sur $\mathbf{W}$ est égal à 1.41%, en moyenne), ce qui donne un taux moyen d'erreur de reconstruction sur $\mathbf{X}$ égal à 1.82%. Cela n'est plus vrai quand p augmente ; dans ce cas, la probabilité d'avoir des sources fortement corrélées est forte et seuls les algorithmes C-PNL-PF et PNL-PF donnent de faibles taux d'erreur (tableau 2.2).

	$\mathbf{X}$	$\mathbf{W}$	$\mathbf{H}$
TH	1.41[0.08, 3.46]	0.70[0.01, 2.03]	0.61[0.0, 2.05]
PF	0.13[0.0, 0.49]	0.06[0.0, 0.27]	0.06[0.0, 0.29]
ASSO	1.82[0.0, 6.51]	1.41[0.0, 4.97]	0.0[0.0, 0,0]
PNL-PF	0.0[0.0, 0.0]	0.0[0.0 0.0]	0.0[0.0, 0.0]
C-PNL-PF	0.0 [0.0, 0.0]	0.0[0.0 0.0]	0.0[0.0, 0.0]

TABLE 2.1 – La moyenne, le minimum et le maximum du taux d'erreur d'estimation (en %) de $\mathbf{W}$ et $\mathbf{H}$ et le taux d'erreur de reconstruction de $\mathbf{X}$, pour $p = 25\%$($N = 120$, $M = 150$, $\beta = 5 \times 10^6$).

	$\mathbf{X}$	$\mathbf{W}$	$\mathbf{H}$
TH	32.36[8.74, 49.48]	12.53[2.41, 19.60]	7.49[0.84, 14.09]
PF	32.96[12.20 52.15]	14.757[4.81, 23.04]	7.02[0.00, 15.67]
ASSO	22.09[3.97, 44.82]	8.51[2.00, 13.56]	1.98[0.00, 8.30]
PNL-PF	1.84[0.00, 7.08]	0.57[0.00, 7.08]	0.40[0.00, 1.65]
C-PNL-PF	1.68[0.00, 5.75]	0.50[0.00, 1.84]	0.36[0.00, 1.84]

TABLE 2.2 – La moyenne, le minimum et le maximum du taux d'erreur d'estimation (en %) de $\mathbf{W}$ et $\mathbf{H}$ et le taux d'erreur de reconstruction de $\mathbf{X}$, pour $p = 60\%$($N = 120$, $M = 150$, $\beta = 5 \times 10^6$).

2.5.2 Performances dans le cas d'un bruit binaire

Nous étudions ensuite le comportement de nos algorithmes en présence de bruit binaire qui correspond au modèle direct suivant : $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T \oplus \mathbf{N}$. La simulation du modèle de mélange est réalisée de façon similaire au cas exact. La matrice de bruit $\mathbf{N}$ est une matrice binaire aléatoire de la même taille que la matrice de données $\mathbf{X}$ qui suit une loi de Bernouilli avec b, la probabilité d'avoir des 1. Le SNR binaire ($SNR_{\mathbb{B}}$) est défini come le rapport entre $\|\mathbf{W} \diamond \mathbf{H}^T\|_F^2$ and $\|\mathbf{N}\|_F^2$:

$$SNR_{\mathbb{B}} = \frac{\left\|\mathbf{W} \diamond \mathbf{H}^T\right\|_F^2}{\|\mathbf{N}\|_F^2}$$

Nous calculons le taux moyen d'erreur d'estimation sur $\mathbf{H}$ $\left(\text{Err}_{\mathbf{H}} = \frac{\left\|\mathbf{H}-\hat{\mathbf{H}}\right\|_F^2}{MK} \times 100(\%)\right)$,

et sur $\mathbf{W}$ $\left(\text{Err}_{\mathbf{W}} = \frac{\|\mathbf{W}-\hat{\mathbf{W}}\|_F^2}{NK} \times 100(\%)\right)$ et le taux moyen d'erreur de reconstruction pour $\mathbf{X}$ $\left(\text{Err}_{\mathbf{X}} = \frac{\|\mathbf{X}-\hat{\mathbf{X}}\|_F^2}{NM} \times 100(\%)\right)$ sur 40 réalisations, en fonction du taux de bruit rajouté b. $\mathbf{W}$ et $\mathbf{H}$ sont générées comme précédemment, en suivant une distribution de Bernoulli, avec p, la probabilité d'avoir des 1 sur chaque colonne de $\mathbf{W}$ et $\mathbf{H}$. On peut voir que, lorsque les sources sont non corrélées, *i.e.*, $p = 0.25$ ou 25% (figure 2.9), tous les algorithmes, sauf ASSO, donnent des résultats satisfaisants pour un taux de bruit $b < 0.26$. Il est à noter que jusqu'à cette valeur de b, le $SNR_{\mathbb{B}} > 1$ (voir tableau 2.3). Au-delà de cette valeur, la « puissance » du bruit devient supérieure à la « puissance » du signal ($SNR_{\mathbb{B}} < 1$), ce qui entraîne une perte importante dans la structure des données.
Lorsque les sources sont fortement corrélées, ($p = 60\%$) (Figure 2.10), les algorithmes proposés, et en particulier C-PNL-PF, surpassent les autres. Il est à noter que dans le cas corrélé, il y a une source supplémentaire d'erreurs provenant de la non-unicité de la décomposition. Ce type d'erreur peut être géré par la contrainte de support maximale introduite dans l'algorithme C-PNL-PF.
Une autre conclusion qui peut être tirée des simulations effectuées est que les approches de type « greedy » comme ASSO, ne sont pas adaptées pour l'approximation de rang faible de données binaires « bruitées ». Elles ont été initialement conçues pour des applications dans le cas exact.

b	0.04	0.08	0.12	0.16	0.2	0.24	0.26	0.28	0.32	0.36	0.40
$SNR_{\mathbb{B}}$	6.95	3.48	2.31	1.69	1.38	1.10	1.05	0.96	0.85	0.74	0.69

TABLE 2.3 – Le SNR binaire ($SNR_{\mathbb{B}}$) en fonction du taux de bruit rajouté b pour $p = 25\%$.

2.6 La FBMB appliquée aux données Netflix

Cette section vise à évaluer le potentiel des algorithmes de factorisation proposés (PNL-PF et C-PNL-PF) pour l'analyse de l'ensemble de données Netflix ([101], [102]). Il est important de souligner que l'objectif n'est pas de proposer un nouveau système de recommandation, mais plutôt d'évaluer :

1. le pouvoir explicatif des algorithmes de factorisation,
2. la capacité de prédiction du modèle de rang faible correspondant.

2.6.1 Données Netflix

Les données Netflix considérées sont constituées d'une matrice de 17770 utilisateurs $\times$ 480189 films ; chaque utilisateur donne une note entre 1 et 5 aux films regardés, les éléments nuls de la matrice indiquant les films non classés ou non regardées. Pour pouvoir traiter les données dans un délai raisonnable, nous avons extrait une sous-matrice de 3000×5000 de la matrice d'origine. Puisque $98,93\%$ des données sélectionnées sont des

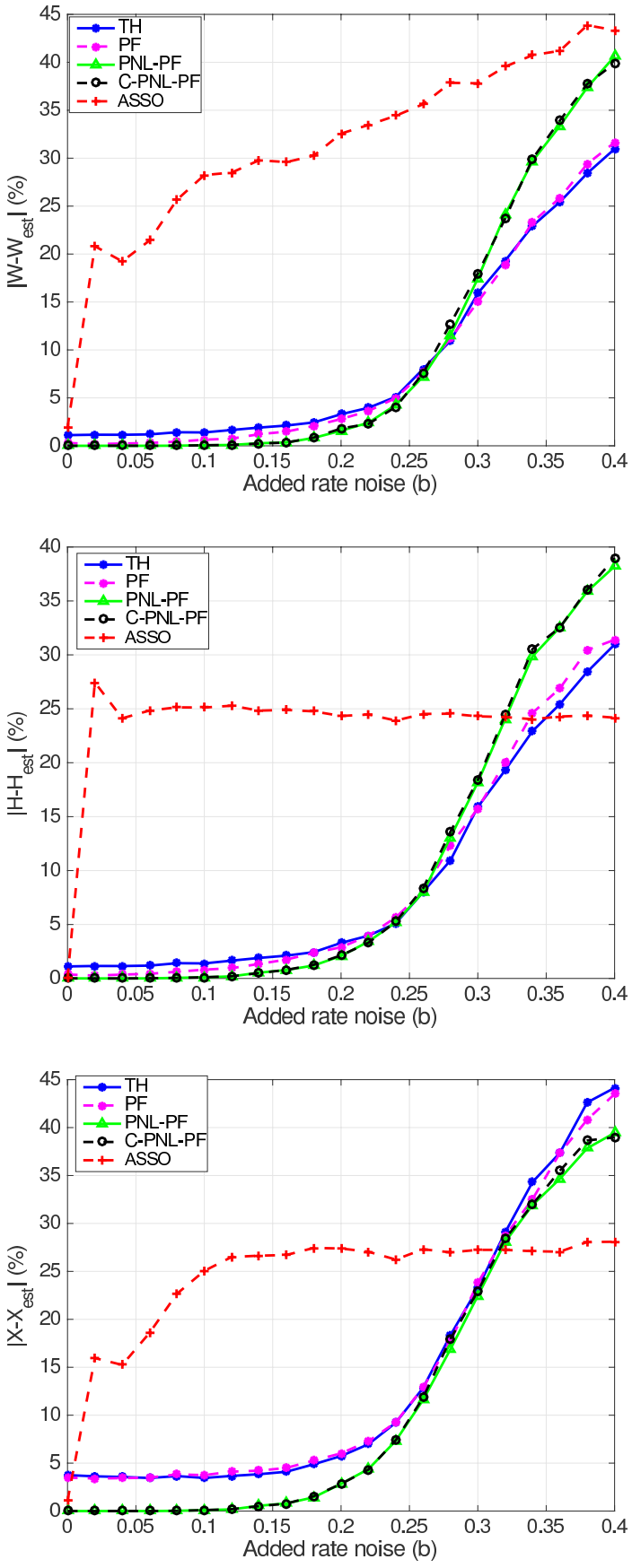


FIGURE 2.9 – Erreur d'estimation sur $\mathbf{W}$, $\mathbf{H}$ et de reconstruction sur $\mathbf{X}$ en fonction du taux de bruit b, pour $p = 25\%$ ($N = 80$, $M = 100$, $\beta = 10$).

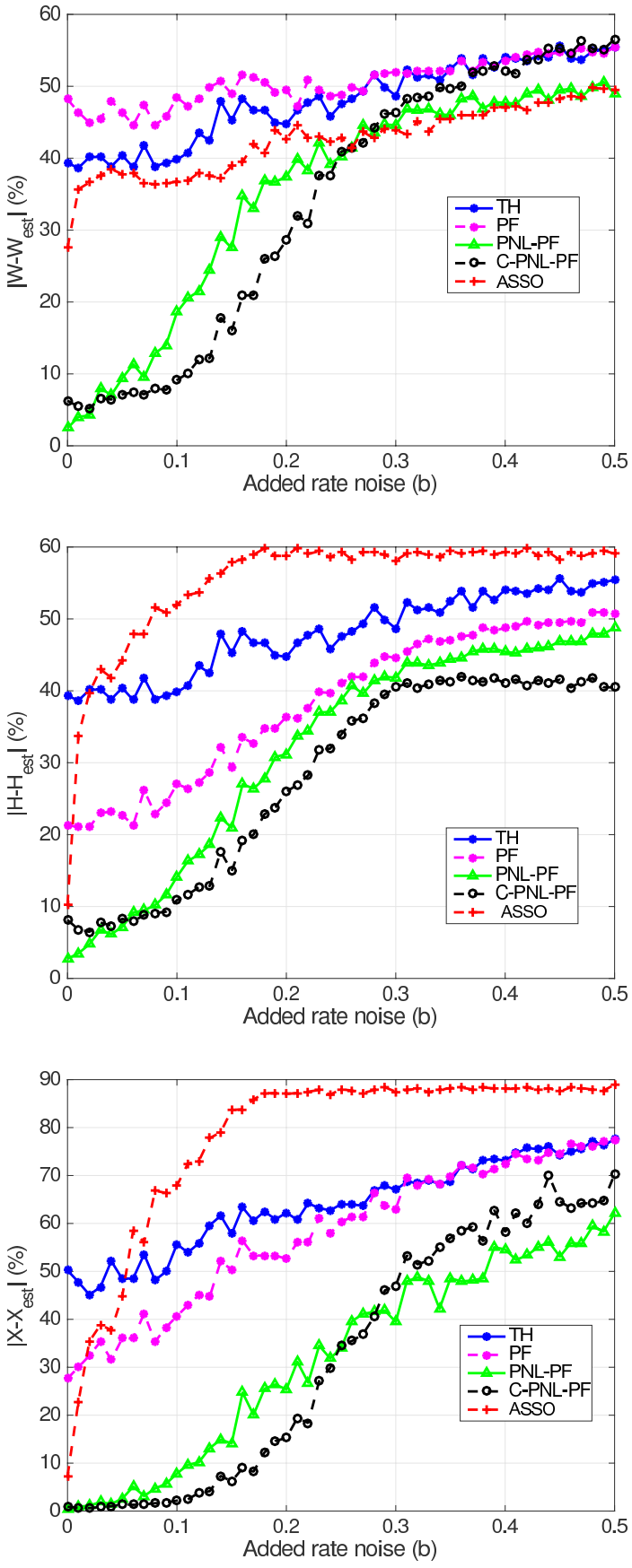


FIGURE 2.10 – Erreur d'estimation sur $\mathbf{W}$, $\mathbf{H}$ et de reconstruction sur $\mathbf{X}$ en fonction du taux de bruit b, pour $p = 60\%$ ($N = 80$, $M = 100$, $\beta = 6 \times 10^5$).

« 0 », nous les densifions en sélectionnant les utilisateurs qui ont donné un nombre d'avis plus grand que la moyenne, et les films qui ont plus de notes que la moyenne, ce qui donne une matrice de 502 × 1528. Pour appliquer la FBMB à cette matrice à valeurs entières, nous allons « binariser » cette matrice 502 × 1528 par seuillage, en mettant à « 1 » les éléments (notes) de cette matrice strictement supérieurs à 3, et à « 0 » tous les autres. La matrice binaire de taille réduite, notée $\mathbf{X}$ est ainsi représentée sur la figure 2.11a.

2.6.2 Capacité explicative des algorithmes FBMB

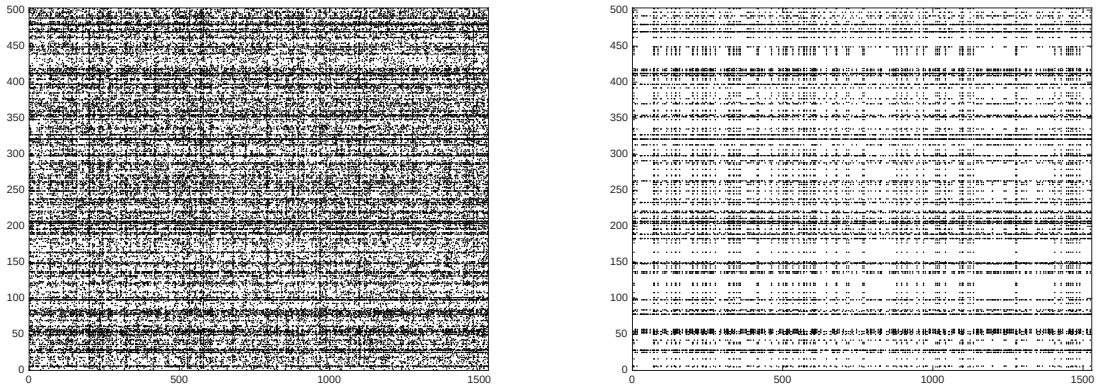
Dans cette section, nous étudions la capacité des différents algorithmes de FBMB d'expliquer les données avec un rang faible. La figure 2.12 montre le taux d'erreur d'approximation défini comme :

$$Err(k) = \frac{\|\mathbf{X} - \hat{\mathbf{X}}(k)\|_F^2}{\|\mathbf{X}\|_F^2} \times 100\ \%,$$

où $\hat{\mathbf{X}}(k)$ représente une approximation binaire de $\mathbf{X}$ avec un rang k. On peut voir que PNL-PF et C-PNL-PF expliquent 17,5% et 22% des données, respectivement avec un rang de décomposition égal à 4, alors que les autres sont inférieurs à 14%. Ces valeurs peuvent sembler faibles, au premier coup d'oeil, mais elles permettent de capturer la structure cohérente principale des données, comme illustré sur la figure 2.11b. Pour les rangs de décomposition supérieurs à 4, le taux d'erreur diminue plus lentement, ce qui indique qu'il n'y a plus de structure significative dans les données. Comme révélé par la FBMB de $\mathbf{X}$, il y a principalement 4 groupes cohérents de (Utilisateurs, Films) dans les données analysées, notés (U1, M1), (U2, M2), (U3, M3) et (U4, M4). Il est difficile d'analyser les relations entre les groupes (chevauchement, structure) simplement en inspectant visuellement les termes de rang 1. Il est possible de réarranger les lignes et les colonnes de la matrice de rang 4 (figure 2.11b), pour obtenir une meilleure vue du chevauchement et de la structure des différents groupes (voir la figure 2.13). Sur la base de cette nouvelle représentation, on peut identifier les utilisateurs dont les opinions sont peu discriminantes (*i.e.*, appartenant à plusieurs groupes) et les utilisateurs qui sont beaucoup plus sélectifs. De plus, il est possible d'identifier les films qui sont appréciés par une grande catégorie d'utilisateurs (les films qui font le « buzz ») et ceux qui sont spécifiques à un groupe.

2.6.3 Capacité de prédiction des préférences des utilisateurs

L'objectif de cette sous-section est double. Dans un premier temps, nous analysons la validité des groupes d'utilisateurs extraits par l'algorithme C-PNL-PF. Deuxièmement, nous étudions si ces groupes d'utilisateurs peuvent être utilisés pour prédire les préférences de film d'autres utilisateurs, qui n'ont pas été pris en compte dans la phase d'apprentissage. À cet égard, deux sous-matrices ont été extraites de la matrice de données $\mathbf{X}$ comme suit : la matrice $\mathbf{X}^a = \mathbf{X}_{(1:300,1:500)}$ qui est utilisée pour étudier les groupes d'utilisateurs



(a) Matrice $\mathbf{X}$ obtenue après « binarisation ». (b) Approximation de rang 4 de $\mathbf{X}$.

FIGURE 2.11 – Données Netflix binarisées et approximation de rang 4.w

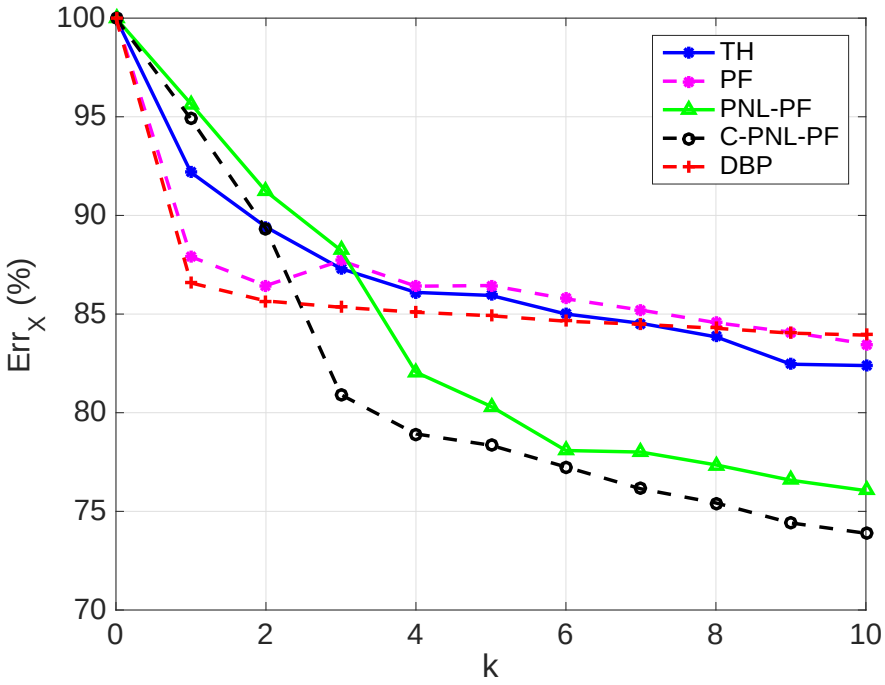


FIGURE 2.12 – Erreur d'approximation de $\mathbf{X}$ en fonction du rang k.

$\lvert U_1 \rvert$	$\lvert U_F \rvert$	$\lvert U_3 \rvert$	$\lvert U_4 \rvert$
37	27	35	42

$\lvert U_1 \cap U_F \rvert$	$\lvert U_1 \cap U_3 \rvert$	$\lvert U_1 \cap U_4 \rvert$
10	17	15

$\lvert U_F \cap U_3 \rvert$	$\lvert U_F \cap U_4 \rvert$	$\lvert U_3 \cap U_4 \rvert$
15	16	19

$\lvert U_1 \cap U_F \cap U_3 \rvert$	$\lvert U_1 \cap U_F \cap U_4 \rvert$
17	9

$\lvert U_1 \cap U_3 \cap U_4 \rvert$	$\lvert U_F \cap U_3 \cap U_4 \rvert$
6	10

$\lvert U_1 \cap U_F \cap U_3 \cap U_4 \rvert$
5

TABLE 2.4 – Le cardinal des groupes d'utilisateurs et de leurs intersections.

$\lvert M_1 \rvert$	$\lvert M_F \rvert$	$\lvert M_3 \rvert$	$\lvert M_4 \rvert$
394	518	277	166

$\lvert M_1 \cap M_F \rvert$	$\lvert M_1 \cap M_3 \rvert$	$\lvert M_1 \cap M_4 \rvert$
83	61	34

$\lvert M_F \cap M_3 \rvert$	$\lvert M_F \cap M_4 \rvert$	$\lvert M_3 \cap M_4 \rvert$
66	77	44

$\lvert M_1 \cap M_F \cap M_3 \rvert$	$\lvert M_1 \cap M_F \cap M_4 \rvert$
5	9

$\lvert M_1 \cap M_3 \cap M_4 \rvert$	$\lvert M_F \cap M_3 \cap U_4 \rvert$
6	21

$\lvert M_1 \cap M_F \cap M_3 \cap M_4 \rvert$
3

TABLE 2.5 – Le cardinal des groupes de films et de leurs intersections.

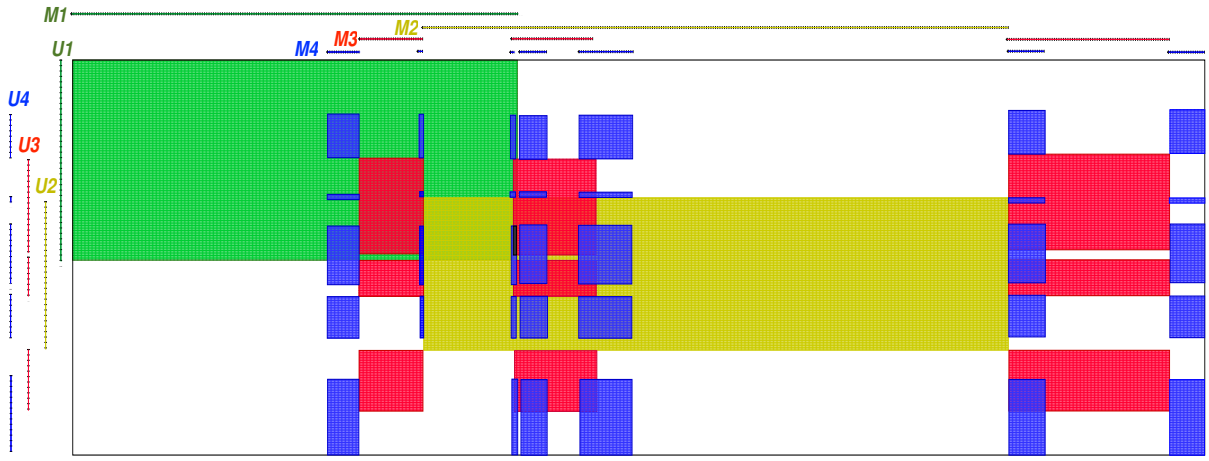


FIGURE 2.13 – Réorganisation de la matrice d'approximation de $\mathbf{X}$ (figure 2.11b) selon les 4 groupes (utilisateurs, films) estimés.

et la matrice $\mathbf{X}^b = \mathbf{X}_{(1:502,501:1528)}$ qui est utilisée pour la validation et la prédiction. L'apprentissage des groupes d'utilisateurs a été obtenu en effectuant une FBMB de rang 4 sur $\mathbf{X}^a$. Une fois les groupes d'utilisateurs U_i, $i = 1, \cdots, 4$ obtenus, nous estimons la réponse moyenne de chaque groupe d'utilisateurs par rapport aux films comme :

$$m_i = \mathbb{I}\left(\frac{1}{|U_i|}\sum_{k\in U_i}\mathbf{x}^b_{(k:)} > \rho\right),$$

où $\mathbb{I}$ est la fonction indicatrice et $0 < \rho \leq 1$. Ici, nous définissons $\rho = 0.7$. Pour évaluer si un utilisateur k a des préférences similaires à l'un des quatre groupes d'utilisateurs, nous avons évalué sa corrélation avec la réponse binaire moyenne définie telle que :

$$\mathbf{C}_{k,i} = \sum_{j=1}^{1028}\mathbf{X}^b_{k,j} \wedge m_i.$$

La validation du groupe d'utilisateurs a été faite en examinant les valeurs de $\mathbf{C}_{k,i}$, pour $k = 1, \ldots, 300$ tandis que la prédiction des préférences des utilisateurs a été faite en comparant les valeurs de $\mathbf{C}_{k,i}$ pour $k = 301, \ldots, 502$ à un seuil fixé à $\min_{k\in U_i}(\mathbf{C}_{k,i})$. Pour valider la capacité de prédiction des préférences des utilisateurs, les résultats ont été comparés à ceux obtenus en apprenant les groupes d'utilisateurs sur la matrice de données $\mathbf{X}^c = \mathbf{X}_{(1:502,1:500)}$.

Les résultats sont présentés à la figure 2.14. Sur la colonne de gauche, nous avons tracé les 4 termes de rang 1 appris sur $\mathbf{X}^1$ tandis que les diagrammes de la colonne du milieu sont les $\mathbf{C}_{k,i}, i = 1, \ldots, 4$ correspondants. Les utilisateurs de chaque groupe sont marqués par « $\circ$ » ($k = 1, \ldots, 300$) et les utilisateurs affectés à un groupe sont marqués par « $*$ »($k = 301, \ldots, 502$). La colonne de droite correspond aux $\mathbf{C}_{k,i}, i = 1, \ldots, 4$ avec les groupes d'utilisateurs appris sur $\mathbf{X}^c$ marqués par « $\circ$ ». Pour les groupes

d'utilisateurs 1, 3 et 4, les résultats sont satisfaisants pour les aspects de validation et de prédiction. En effet, pour la validation, presque tous les utilisateurs de ces groupes ont une valeur de $\mathbf{C}_{k,i}$ élevée et seulement très peu d'utilisateurs d'un groupe différent ont aussi un $\mathbf{C}_{k,i}$ élevé. Pour la prédiction, la plupart des utilisateurs affectés à un groupe sont affectés au même groupe d'utilisateurs lorsque l'apprentissage est effectué sur $\mathbf{X}^c$. Le cas du groupe d'utilisateurs 2 est quelque peu différent. Bien que la validation de ce groupe soit satisfaisante, il semble que la prédiction soit moins robuste. En effet, le second terme de la figure 2.14 a une structure beaucoup plus verticale (par rapport aux autres termes de rang 1), ce qui signifie que ce terme est plus approprié pour indiquer les films qui font le « buzz » que le comportement des utilisateurs.

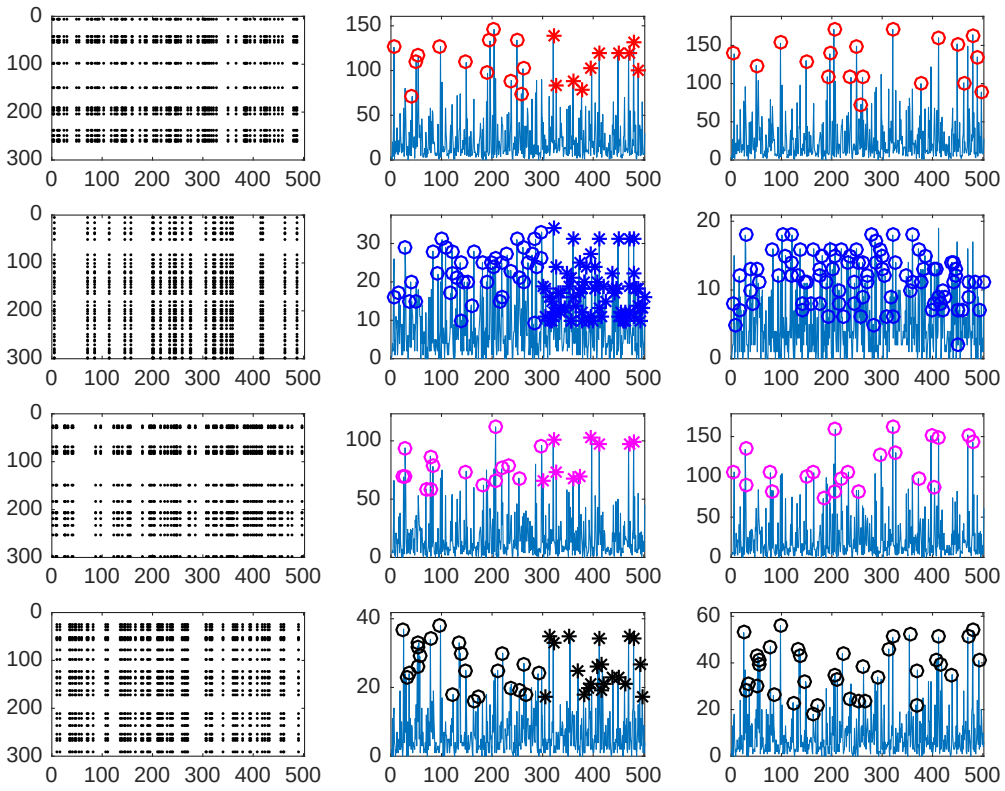


FIGURE 2.14 – Validation et prédiction.

2.7 Conclusion sur la FMBM

Dans cette section, nous avons introduit une nouvelle méthode pour la factorisation booléenne de matrices à valeurs binaires (FBMB) dans un cadre de séparation de sources. La méthode proposée est basée sur un modèle de mélange post non-linéaire qui est équivalent au modèle de mélange booléen lorsque les facteurs sont exactement binaires. Ce modèle tient explicitement compte de la corrélation des sources et, par conséquent, il donne des résultats plus interprétables dans le cas où les sources se chevauchent, par rapport aux autres algorithmes de l'état de l'art. De plus, dans le cas corrélé, on obtient avec ce modèle des approximations de matrices binaires plus précises par rapport aux autres méthodes similaires, ce qui est une propriété hautement souhaitable dans de nombreuses applications de traitement de signal et de compression de données. Une condition nécessaire et suffisante pour l'unicité de la FBMB a également été démontrée. L'approche proposée a été comparée à travers des simulations numériques avec d'autres algorithmes de l'état de l'art similaires et ses capacités explicatives et prédictives ont été illustrées sur l'ensemble de données Netflix.

Chapitre 3

Une approche par modèle de mélange Post-NonLinéaire de la décomposition canonique polyadique booléenne

Sommaire

3.1 Unicité de la décomposition canonique polyadique booléenne **66**
3.1.1 *Indépendance booléenne* et *rang booléen* d'une famille de vecteurs binaires 68
3.1.2 Conditions suffisantes d'unicité 70
3.2 Algorithmes de DCPB par modèle de mélange post non-linéaire . . . **75**
3.2.1 Modèle de mélange post non-linéaire de la DCPB 75
3.2.2 L'algorithme T-PNL-PF 76
3.2.3 L'algorithme BT-ADMM 78
Le choix du paramètre ρ 80
3.3 Simulations numériques **82**
3.4 Application à des données réelles **85**
3.4.1 Données 85
3.4.2 Capacité explicative des algorithmes DCPB 86
3.5 Conclusion de la DCPB **92**

Dans ce chapitre, nous généralisons l'approche pour la factorisation booléenne des matrices binaires introduite dans le chapitre 2.1 au cas tensoriel, et plus précisément à la Décomposition Canonique Polyadique Booléenne (DCPB) des tenseurs binaires d'ordre trois.

Dans un premier temps nous nous intéressons à l'unicité de la DCPB et généralisons la condition nécessaire et suffisante d'unicité introduite dans le chapitre 2 au cas tensoriel. Nous introduisons ensuite une condition suffisante pour la décomposition booléenne des tableaux de données binaires, plus facilement vérifiable pour des tableaux multidimensionnels que la condition nécessaire et suffisante. Cette condition est fondée sur une notion nouvelle d'*indépendance booléenne* des vecteurs binaires, que nous introduisons également dans ce chapitre.

Dans la deuxième partie du chapitre nous proposons un premier algorithme pour l'estimation de la DCPB comme une extension directe de l'algorithme matriciel PNL-PF. Un deuxième algorithme fondé sur une approche de type AO-ADMM (pour *Alternating optimization - Alternating Direction Method of Multiplier* en anglais), plus efficace d'un point de vue calculatoire, est ensuite proposé. Cet algorithme est mieux adapté au cas des décompositions multidimensionnelless et peut facilement se généraliser aux tenseurs d'ordre plus grand que 3. Ces algorithmes seront ensuite comparés en simulations avec des approches similaires de l'état de l'art et testés sur des données réelles.

3.1 Unicité de la décomposition canonique polyadique booléenne

Considérons la DCPB ci-dessous :

$$\mathcal{X} = \bigvee_{k=1}^{K} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k) = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!], \tag{3.1}$$

avec $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_K]$, $\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_K]$ et $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_K]$ et $(\circ)$ qui représente le produit tensoriel.

On dira que la DCPB est *unique* (ou que le modèle DCPB associé est *identifiable*) si pour toute autre décomposition $\mathcal{X} = [\![\bar{\mathbf{W}}, \bar{\mathbf{H}}, \bar{\mathbf{V}}]\!] = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]$, les triplets $(\bar{\mathbf{W}}, \bar{\mathbf{H}}, \bar{\mathbf{V}})$ et $(\mathbf{W}, \mathbf{H}, \mathbf{V})$ sont les mêmes, à une permutation conjointe des colonnes de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ près. Comme dans le cas matriciel, l'indétermination d'échelle n'existe pas dans le cas des tenseurs binaires.

Un premier résultat d'unicité pour la DCPB peut être obtenu en appliquant directement la condition nécessaire et suffisante d'unicité des décompositions booléennes matricielles 2.3.1, démontrée au chapitre précédent, aux trois dépliements du tenseur binaire $\mathcal{X}$ (voir (3.12), (3.13), (3.14)). Il en résulte que le $\ell^{\text{ième}}$ terme de rang 1 de la décomposition : $\mathcal{X}^{(\ell)} = \mathbf{w}_\ell \circ \mathbf{h}_\ell \circ \mathbf{v}_\ell$ est identifiable si et seulement si les trois conditions ci-dessous sont satisfaites conjointement :

$$\begin{cases} \forall\, n \in \{1,\ldots,N\} \text{ avec } \mathbf{w}_\ell(n) \neq 0 \text{ on a : } supp\left(\mathbf{e}_n(\mathbf{v}_\ell \otimes \mathbf{h}_\ell)^T\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{w}_k(\mathbf{v}_k \otimes \mathbf{h}_k)^T\right), \\ \forall\, m \in \{1,\ldots,M\} \text{ avec } \mathbf{h}_\ell(m) \neq 0 \text{ on a : } supp\left(\mathbf{e}_m(\mathbf{v}_\ell \otimes \mathbf{w}_\ell)^T\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{h}_k(\mathbf{v}_k \otimes \mathbf{w}_k)^T\right), \\ \forall\, p \in \{1,\ldots,P\} \text{ avec } \mathbf{v}_\ell(p) \neq 0 \text{ on a : } supp\left(\mathbf{e}_p(\mathbf{h}_\ell \otimes \mathbf{w}_\ell)^T\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{v}_k(\mathbf{h}_k \otimes \mathbf{w}_k)^T\right), \end{cases}$$

avec $\mathbf{e}_n$, $\mathbf{e}_m$ et $\mathbf{e}_p$ qui représentent respectivement le $n^{\text{ième}}$, $m^{\text{ième}}$ et $p^{\text{ième}}$ vecteur des bases canoniques de $\mathbb{R}^N$, $\mathbb{R}^M$ et $\mathbb{R}^P$. En repassant au formalisme tensoriel, la condition d'unicité suivante est obtenue :

Théorème 3.1.1 (Condition nécessaire et suffisante d'unicité partielle)**.** *Le $\ell^{ième}$ terme de rang 1 $\mathcal{X}^{(\ell)} = \mathbf{w}_\ell \circ \mathbf{h}_\ell \circ \mathbf{v}_\ell$ de la DCPB $\mathcal{X} = \underset{k=1}{\overset{K}{\vee}} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k)$ peut être estimé de façon unique à partir de $\mathcal{X}$ ssi :*

$$\begin{cases} \forall\, n \in \{1,\ldots,N\},\ on\ a : supp\left(\mathbf{e}_n \circ \mathbf{h}_\ell \circ \mathbf{v}_\ell\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k\right), \\ \forall\, m \in \{1,\ldots,M\},\ on\ a : supp\left(\mathbf{w}_\ell \circ \mathbf{e}_m \circ \mathbf{v}_\ell\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k\right), \\ \forall\, p \in \{1,\ldots,P\},\ on\ a : supp\left(\mathbf{w}_\ell \circ \mathbf{h}_\ell \circ \mathbf{e}_p\right) \not\subseteq \underset{k\neq\ell}{\overset{K}{\cup}} supp\left(\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k\right), \end{cases}$$

avec $\mathbf{e}_n$, $\mathbf{e}_m$ et $\mathbf{e}_p$ qui représentent respectivement le $n^{ième}$, $m^{ième}$ et $p^{ième}$ vecteur des bases canoniques de $\mathbb{R}^N$, $\mathbb{R}^M$ et $\mathbb{R}^P$.

Comme pour les matrices, le support d'un tenseur binaire est défini selon : $supp(\mathcal{X}) = \{(i,j,t),\ \text{tel que } x_{ijt} \neq 0\}$.

En d'autres mots, le théorème 3.1.1 dit que $\mathcal{X}^{(\ell)}$ peut être estimé de façon unique à partir de $\mathcal{X}$ si aucune tranche horizontale, latérale ou frontale du tenseur $\mathcal{X}^{(\ell)}$ n'est incluse dans l'union des supports des autres termes de rang 1. Cette condition, bien qu'intéressante d'un point point de vue théorique, est d'une utilité pratique assez limitée, car difficile à verifier. C'est pour cette raison que, par la suite, nous démontrons une condition d'unicité suffisante plus facile à vérifier dans la pratique.

Afin d'exprimer cette condition dans un language similaire à celui utilisé dans l'algèbre linéaire et multilinéaire classique, nous introduisons, dans la section suivante, un formalisme vectoriel fondé sur une nouvelle notion, celle d'*indépendance booléenne* des vecteurs à valeurs binaires.

3.1.1 *Indépendance booléenne* et *rang booléen* d'une famille de vecteurs binaires

Une analogie intéressante peut être établie entre les notions bien connues d'*indépendance linéaire* et de *rang* d'une famille de vecteurs à valeurs réelles et des concepts similaires pour les vecteurs à valeurs binaires, dans le cadre booléen. Ces nouveaux concepts serviront à formuler des conditions d'unicité pour la décomposition *booléenne* en termes de rang 1 des tenseurs binaires, qui sont similaires d'un point de vue formel aux conditions d'unicité pour les tenseurs à valeurs réelles.

Définition 3.1.1 (*Dépendance booléenne des vecteurs à valeurs binaires).* *Considérons une famille de K vecteurs binaires $Y = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$. On dit qu'un vecteur binaire non nul $\mathbf{u}$ est* booléennement dépendant *des vecteurs dans Y si :*

$$\mathbf{u} \vee (\bigvee_{k=1}^{K} \mathbf{y}_k) = \bigvee_{k=1}^{K} \mathbf{y}_k, \tag{3.2}$$

On notera $\mathbf{u} \overset{\mathbb{B}}{\sim} \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$. Dans le cas contraire, nous dirons que $\mathbf{u}$ est booléennement indépendant *de Y et nous noterons $\mathbf{u} \overset{\mathbb{B}}{\nsim} \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$.*

Il est facile de montrer que la définition (3.2) est équivalente à

$$supp(\mathbf{u}) \subseteq \bigcup_{k=1}^{K} supp(\mathbf{y}_k). \tag{3.3}$$

Il faut noter que, contrairement à la dépendance linéaire classique, la dépendance booléenne est asymétrique, *i.e.,* $\mathbf{u} \overset{\mathbb{B}}{\sim} \mathbf{y}$ n'implique pas forcément $\mathbf{y} \overset{\mathbb{B}}{\sim} \mathbf{u}$.

Définition 3.1.2 (*Famille booléenne libre).* *Nous dirons qu'une famille de vecteurs binaires $Y = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$ forme une* famille booléenne libre *si :*

$$\forall \ell \in \{1, \ldots, K\}, \text{ nous avons } \mathbf{y}_\ell \overset{\mathbb{B}}{\nsim} Y^{\{-\ell\}}, \tag{3.4}$$

avec $Y^{\{-\ell\}}$ l'ensemble des vecteurs de Y privé de l'élément $\mathbf{y}_\ell$.

Cette définition nous amène naturellement à la définition du rang booléen d'une famille de vecteurs binaires.

Définition 3.1.3 (*Rang booléen des colonnes d'une matrice binaire).* *Soit $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ la famille de vecteurs binaires formée par l'ensemble des colonnes d'une matrice $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_k]$. Nous définissons le rang booléen des colonnes de $\mathbf{X}$, noté $rg_{\mathbb{B}}(\mathbf{X})$, comme étant le nombre maximum de vecteurs qu'on peut extraire de $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\}$ de manière à former une famille booléenne libre.*

Il faut noter que, contrairement au cas classique des vecteurs à valeurs réelles, le *rang booléen des colonnes* d'une matrice binaire $\mathbf{X}$ est différent du *rang booléen de la décomposition* booléenne de $\mathbf{X}$, noté $rang_{\mathbb{B}}(\mathbf{X})$, définie dans la section 1.2.4. Cela est illustré par l'exemple ci-dessous :

Exemple 3.1.1. *Soit la matrice à valeurs binaires* $\mathbf{X} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$,

Il est facile de vérifier que $rang_{\mathbb{B}}(\mathbf{X}) = 3 \neq rg_{\mathbb{B}}(\mathbf{X}) = 2$.

Cependant, pour une factorisation booléenne $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$, on peut démontrer une relation d'inégalité entre le rang booléen de la décomposition de $\mathbf{X}$ et le rang booléen des colonnes de ses facteurs $\mathbf{W}$ et $\mathbf{H}$.

Proposition 3.1.1. *Considérons la FBMB de* $\mathbf{X}$ *donnée par :* $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$, *alors nous avons :*

$$rang_{\mathbb{B}}(\mathbf{X}) \geq \max\left(rg_{\mathbb{B}}(\mathbf{W}), rg_{\mathbb{B}}(\mathbf{H})\right). \tag{3.5}$$

Démonstration. Soit $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_K]$ et $\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_K]$ et soit $\max\left(rg_{\mathbb{B}}(\mathbf{W}), rg_{\mathbb{B}}(\mathbf{H})\right) = R$ avec $R \leq K$. Alors, il existe R valeurs $k_r \in \{1, \cdots, K\}$ tel que :

$$\forall\, r \in \{1, \cdots, R\},\ \text{on a } \mathbf{w}_{k_r} \overset{\mathbb{B}}{\nsim} W^{\{-k_r\}} \text{ ou } \mathbf{h}_{k_r} \overset{\mathbb{B}}{\nsim} H^{\{-k_r\}},$$

ce qui équivaut à dire que :

$$\forall\, r \in \{1, \cdots, R\},\ \text{on a } \begin{cases} \mathbf{w}_{k_r} \vee \left(\bigvee\limits_{k \neq k_r} \mathbf{w}_k\right) \neq \left(\bigvee\limits_{k \neq \ell} \mathbf{w}_k\right) \\ \text{ou} \\ \mathbf{h}_{k_r} \vee \left(\bigvee\limits_{k \neq k_r} \mathbf{h}_k\right) \neq \left(\bigvee\limits_{k \neq k_r} \mathbf{h}_k\right) \end{cases}.$$

Alors nous pouvons écrire que : $\forall\, r \in \{1, \cdots, R\},\quad \mathbf{w}_{k_r}\mathbf{h}_{k_r}^T \vee \left(\bigvee\limits_{k \neq k_r} \mathbf{w}_k\mathbf{h}_k^T\right) \neq \left(\bigvee\limits_{k \neq k_r} \mathbf{w}_k\mathbf{h}_k^T\right)$.

Il existe donc au moins R termes de rang 1 dont le support n'est pas inclus dans l'union des supports des autres termes de rang 1, ce qui équivaut à dire que $rang_{\mathbb{B}}(\mathbf{X}) \geq R$.

□

L'exemple suivant illustre ce résultat.

Exemple 3.1.2. *Soit* $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$ *avec* $\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$, $\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ *et*

$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$. *On peut vérifier que* $rg_{\mathbb{B}}(\mathbf{W}) = 2$ *et* $rg_{\mathbb{B}}(\mathbf{H}) = 3$.

Nous avons donc : $rang_{\mathbb{B}}(\mathbf{X}) \geq \max\left(rg_{\mathbb{B}}(\mathbf{W}), rg_{\mathbb{B}}(\mathbf{H})\right) = 3$.

3.1.2 Conditions suffisantes d'unicité

Dans cette section, nous allons tout d'abord démontrer des conditions suffisantes d'unicité pour la FBMB : $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$ en utilisant le nouveau formalisme introduit dans la section précédente.

Théorème 3.1.2 (Unicité partielle de la décomposition $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T$)**.** *Soit la factorisation booléenne de la matrice binaire* $\mathbf{X}$ *donnée par* $\mathbf{X} = \mathbf{W} \diamond \mathbf{H}^T = \bigvee_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T$*. Si*

$$\mathbf{w}_\ell \overset{\mathbb{B}}{\nsim} W^{\{-\ell\}},$$

où $W^{\{-\ell\}}$ *représente l'ensemble des vecteurs colonne de* W *privé de la colonne* $\mathbf{w}_\ell$*, alors* $\mathbf{h}_\ell$ *peut être estimé de façon unique à partir de* $\mathbf{X}$*.*

Démonstration. Supposons que $\mathbf{h}_\ell$ ne peut pas être estimé de façon unique à partir de $\mathbf{X}$. Alors il $\exists\ \bar{\mathbf{X}}^{(\ell)} = \mathbf{w}_\ell \bar{\mathbf{h}}_\ell^T \neq \mathbf{X}^{(\ell)} = \mathbf{w}_\ell \mathbf{h}_\ell^T$ avec $\bar{\mathbf{h}}_\ell = \mathbf{h}_\ell \vee \mathbf{h} \neq \mathbf{h}_\ell$ tel que :

$$\mathbf{X} = \bigvee_{k \neq \ell} \mathbf{X}^{(k)} \vee \mathbf{X}^{(\ell)} = \bigvee_{k \neq \ell} \mathbf{X}^{(k)} \vee \bar{\mathbf{X}}^{(\ell)} \tag{3.6}$$

À partir de (3.6) nous pouvons écrire :

$$\mathbf{X} = \bigvee_{k=1}^{K} \mathbf{w}_k \mathbf{h}_k^T = \mathbf{w}_\ell \mathbf{h}_\ell^T \vee (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T) = \mathbf{w}_\ell \bar{\mathbf{h}}_\ell^T \vee (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T)$$

$$= \mathbf{w}_\ell (\mathbf{h}_\ell \vee \mathbf{h})^T \vee (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T) = \mathbf{w}_\ell \mathbf{h}_\ell^T \vee \mathbf{w}_\ell \mathbf{h}^T \vee (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T) \tag{3.7}$$

Comme $\mathbf{h} \overset{\mathbb{B}}{\nsim} \mathbf{h}_\ell$, l'equation (3.7) implique :

$$\mathbf{w}_\ell \mathbf{h}^T \vee (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T) = (\bigvee_{k \neq \ell} \mathbf{w}_k \mathbf{h}_k^T),$$

ce qui équivaut à :

$$\mathbf{w}_\ell \vee (\bigvee_{k \neq \ell} \mathbf{w}_k) = (\bigvee_{k \neq \ell} \mathbf{w}_k) \text{ et } \mathbf{h} \vee (\bigvee_{k \neq \ell} \mathbf{h}_k) = (\bigvee_{k \neq \ell} \mathbf{h}_k).$$

La première égalité est équivalente à $\mathbf{w}_\ell \overset{\mathbb{B}}{\sim} W^{\{-\ell\}}$, ce qui termine la démonstration.

□

De façon similaire on peut démontrer que $\mathbf{w}_\ell$ est unique si $\mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}}$. Une conséquence directe de ce théorème est donnée par la proposition suivante :

Proposition 3.1.2. *La* $\ell^{ième}$ *source dans la décomposition de* $\mathbf{X}$*,* $\mathbf{X}^{(\ell)} = \mathbf{w}_\ell \mathbf{h}_\ell^T$ *peut être estimée de façon unique à partir de* $\mathbf{X}$ *si* $\mathbf{w}_\ell \overset{\mathbb{B}}{\nsim} W^{\{-\ell\}}$ *et* $\mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}}$*.*

Ce résultat, démontré dans le cas matriciel, peut être utilisé pour déduire une condition suffisante d'unicité dans le cas tensoriel. Avant de donner cette condition, nous avons

besoin du résultat suivant sur l'indépendance booléenne du produit de Kronecker des vecteurs binaires.

Lemme 3.1.1. *Soit* $\mathbf{H} = [\mathbf{h}_1 \cdots \mathbf{h}_K]$ *et* $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_K]$*, deux matrices de vecteurs binaires. Si* $\mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}}$ *ou* $\mathbf{v}_\ell \overset{\mathbb{B}}{\nsim} V^{\{-\ell\}}$*, alors* $(\mathbf{v}_\ell \otimes \mathbf{h}_\ell) \overset{\mathbb{B}}{\nsim} (V \odot H)^{\{-\ell\}}$*, avec* $(V \odot H)^{\{-\ell\}}$ *l'ensemble des vecteurs colonne de la matrice* $\mathbf{V} \odot \mathbf{H}$ *privé du vecteur* $\mathbf{v}_\ell \otimes \mathbf{h}_\ell$*.*

Démonstration. Supposons que $\mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} \underset{k\neq\ell}{\vee} \mathbf{h}_k$, alors nous pouvons décomposer $\mathbf{h}_\ell$ selon $\mathbf{h}_\ell = \mathbf{h} \vee \bar{\mathbf{h}}$, avec $\mathbf{h} \overset{\mathbb{B}}{\sim} \underset{k\neq\ell}{\vee} \mathbf{h}_k$ et $\bar{\mathbf{h}} \overset{\mathbb{B}}{\nsim} \underset{k\neq\ell}{\vee} \mathbf{h}_k$. On pourra donc écrire :

$$(\mathbf{v}_\ell \otimes \mathbf{h}_\ell) \vee [\underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k)] = [\mathbf{v}_\ell \otimes (\mathbf{h} \vee \bar{\mathbf{h}})] \vee [\underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k)] = (\mathbf{v}_\ell \otimes \mathbf{h}) \vee (\mathbf{v}_\ell \otimes \bar{\mathbf{h}}) \vee [\underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k)].$$

Comme $\underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k) \vee (\mathbf{v}_\ell \otimes \bar{\mathbf{h}}) \neq \underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k) \Rightarrow (\mathbf{v}_\ell \otimes \mathbf{h}_\ell) \vee [\underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k)] \neq \underset{k\neq\ell}{\vee} (\mathbf{h}_k \otimes \mathbf{v}_k)$, ce qui termine la démonstration.

□

Théorème 3.1.3 (Unicité partielle du modèle $\mathcal{X} = \overset{K}{\underset{k=1}{\vee}} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k)$)**.** *Considérons la DCPB de* $\mathcal{X}$ *donnée par* $\mathcal{X} = \overset{K}{\underset{k=1}{\vee}} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k) = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]$*. Si*

$$\begin{cases} \mathbf{w}_\ell \overset{\mathbb{B}}{\nsim} W^{\{-\ell\}} \ \textit{et} \ \mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}} \\ \qquad\qquad \textit{ou} \\ \mathbf{w}_\ell \overset{\mathbb{B}}{\nsim} W^{\{-\ell\}} \ \textit{et} \ \mathbf{v}_\ell \overset{\mathbb{B}}{\nsim} V^{\{-\ell\}} \\ \qquad\qquad \textit{ou} \\ \mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}} \ \textit{et} \ \mathbf{v}_\ell \overset{\mathbb{B}}{\nsim} V^{\{-\ell\}} \end{cases},$$

alors la $\ell^{\text{ième}}$ *source* $\mathcal{X}^{(\ell)} = \mathbf{w}_\ell \circ \mathbf{h}_\ell \circ \mathbf{v}_\ell$ *peut être estimée de façon unique à partir de* $\mathcal{X}$*.*

Démonstration. Si on applique le résultat de la proposition 3.1.2 aux dépliements du tenseur $\mathcal{X}$ suivant les 3 modes (voir (3.12), (3.13), (3.14)), on obtient que le terme $\mathcal{X}^{(\ell)}$ est identifiable si :

$$\mathbf{w}_\ell \overset{\mathbb{B}}{\nsim} W^{\{-\ell\}} \ \text{et} \ (\mathbf{v}_\ell \otimes \mathbf{h}_\ell) \overset{\mathbb{B}}{\nsim} (V \odot H)^{\{-\ell\}} \tag{3.8}$$

ou

$$\mathbf{h}_\ell \overset{\mathbb{B}}{\nsim} H^{\{-\ell\}} \ \text{et} \ (\mathbf{v}_\ell \otimes \mathbf{w}_\ell) \overset{\mathbb{B}}{\nsim} (V \odot W)^{\{-\ell\}} \tag{3.9}$$

ou

$$\mathbf{v}_\ell \overset{\mathbb{B}}{\nsim} V^{\{-\ell\}} \ \text{et} \ (\mathbf{h}_\ell \otimes \mathbf{w}_\ell) \overset{\mathbb{B}}{\nsim} (H \odot W)^{\{-\ell\}}. \tag{3.10}$$

En appliquant le lemme 3.1.1 aux trois équations ci-dessus, on obtient directement la condition d'unicité. □

Naturellement, on dira que DCPB est *unique* si la condition du théorème 3.1.3 est satisfaite pour tous les termes de rang 1 : $\mathcal{X}^{(\ell)} = \mathbf{w}_\ell \circ \mathbf{h}_\ell \circ \mathbf{v}_\ell$, avec $\ell = 1, \ldots, K$. Cela permet de formuler la condition d'unicité en termes de rang colonne des matrices des facteurs.

Corollaire 3.1.1 (Condition suffisante pour l'unicité de la DCPB)**.** *La DCPB donnée par :*

$$\mathcal{X} = \bigvee_{k=1}^{K} (\mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k)$$

est unique si

$$\begin{cases} rg_{\mathbb{B}}(\mathbf{W}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{H}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{W}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{H}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V}) = K \end{cases} .$$

Ce corollaire dit que, s'il existe au moins deux matrices facteur de $\mathcal{X}$ de rang colonne plein, alors la DCPB de $\mathcal{X}$ est unique.

Les figures 3.1 et 3.2 permettent d'illustrer la condition du corollaire 3.1.1 sur un tenseur $\mathcal{X}$ de taille $10 \times 8 \times 6$. Ainsi, dans la figure 3.1, nous pouvons voir que la condition du corollaire est vérifiée, car nous avons $rg_{\mathbb{B}}(\mathbf{W}) = K = 3$ et $rg_{\mathbb{B}}(\mathbf{H}) = K = 3$. Par contre dans la figure 3.2, ce corollaire n'est pas vérifié car nous avons $rg_{\mathbb{B}}(\mathbf{W}) = 2, rg_{\mathbb{B}}(\mathbf{H}) = 2$ et $rg_{\mathbb{B}}(\mathbf{V}) = 1$. Dans ce cas, on arrive à trouver un autre triplet $\{\bar{\mathbf{W}}, \bar{\mathbf{H}}, \bar{\mathbf{V}}\} \neq \{\mathbf{W}, \mathbf{H}, \mathbf{V}\}$ qui vérifie :

$$\mathcal{X} = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!] = [\![\bar{\mathbf{W}}, \bar{\mathbf{H}}, \bar{\mathbf{V}}]\!] .$$

Néanmoins, le corollaire 3.1.1 est une condition suffisante. Nous pouvons la relaxer en utilisant les partitions des matrices facteurs.

Ainsi, à partir des conditions (3.8), (3.9), (3.10) nous pouvons écrire que si

$$\forall \ell \in \{1, \cdots, K\}, \begin{cases} \mathbf{w}_\ell \overset{\mathbb{B}}{\preccurlyeq} W^{\{-\ell\}} \text{ et } (\mathbf{v}_\ell \otimes \mathbf{h}_\ell) \overset{\mathbb{B}}{\preccurlyeq} (V \odot H)^{\{-\ell\}} \\ \qquad\qquad ou \\ \mathbf{h}_\ell \overset{\mathbb{B}}{\preccurlyeq} H^{\{-\ell\}} \text{ et } (\mathbf{v}_\ell \otimes \mathbf{w}_\ell) \overset{\mathbb{B}}{\preccurlyeq} (V \odot W)^{\{-\ell\}} \\ \qquad\qquad ou \\ \mathbf{v}_\ell \overset{\mathbb{B}}{\preccurlyeq} V^{\{-\ell\}} \text{ et } (\mathbf{h}_\ell \otimes \mathbf{w}_\ell) \overset{\mathbb{B}}{\preccurlyeq} (H \odot W)^{\{-\ell\}} \end{cases} ,$$

alors la DCPB de $\mathcal{X}$ est unique. Si on écrit ces conditions en termes de rang colonne, on obtient que la décomposition est unique si :

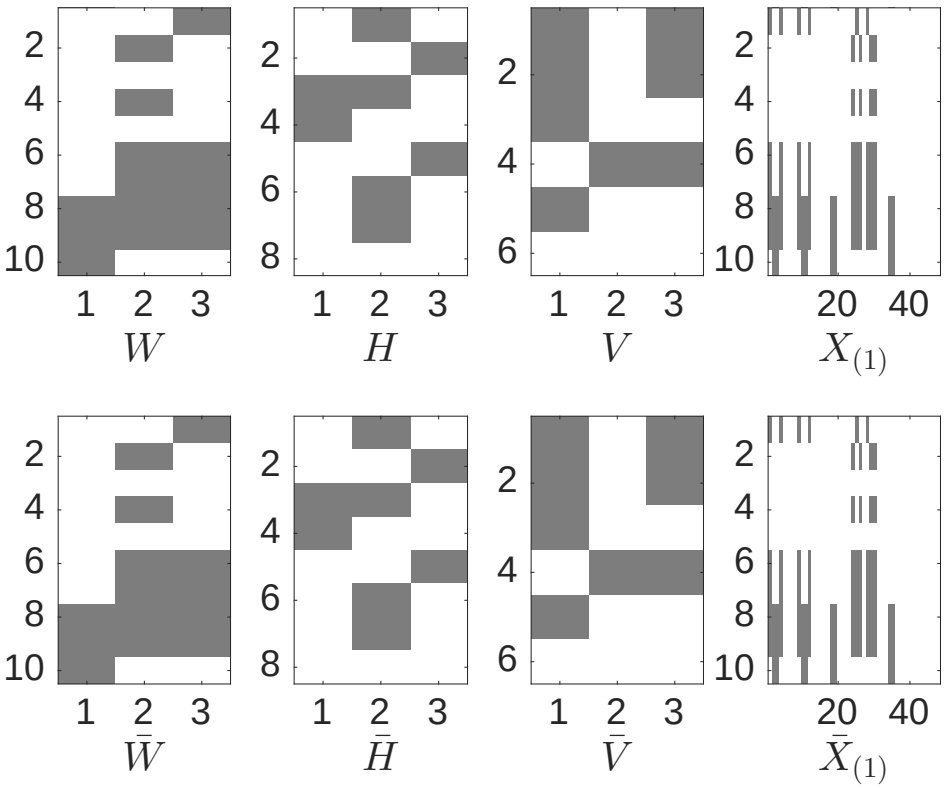


FIGURE 3.1 – Décomposition unique de $\mathcal{X}$.

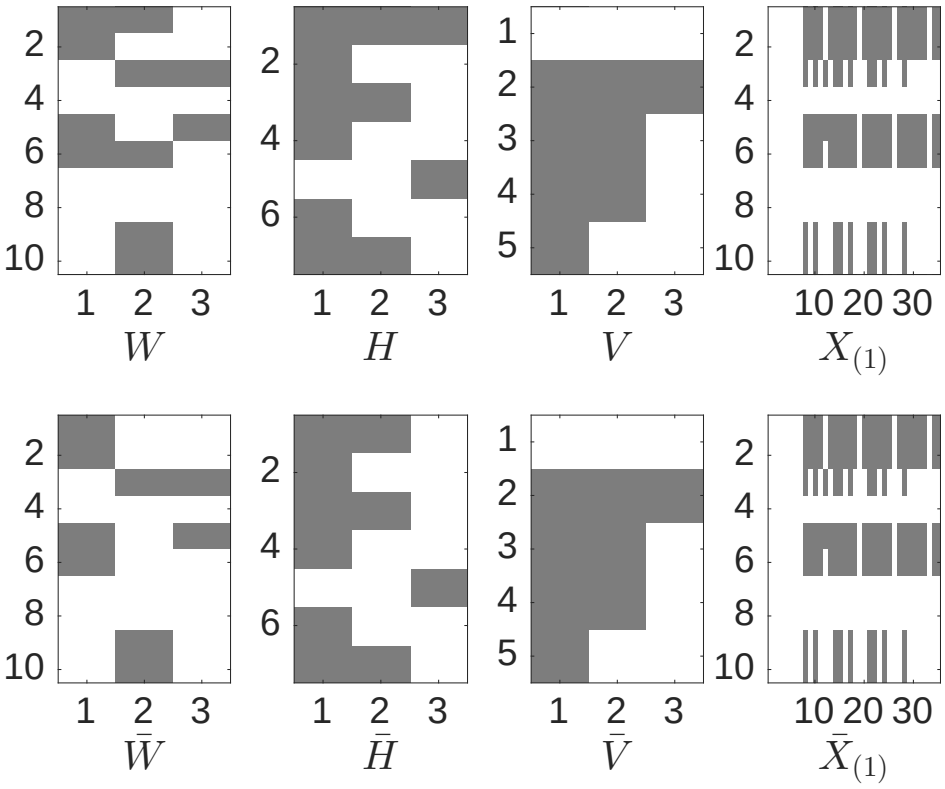


FIGURE 3.2 – Décomposition non-unique de $\mathcal{X}$.

$$\begin{cases} rg_{\mathbb{B}}(\mathbf{W}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{H}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{H}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{W}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{V}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{H} \odot \mathbf{W}) = K \end{cases}.$$

En s'appuyant sur le lemme 3.1.1, nous pouvons donner la condition ci-dessous sur les partitions de $\mathbf{H}$ et $\mathbf{V}$, pour que le produit de Khatri-Rao $\mathbf{V} \odot \mathbf{H}$ soit de rang plein.

Lemme 3.1.2 (Condition sur $\mathbf{H}$ et $\mathbf{V}$ pour que $rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{H}) = K$)**.** *Soient* $\mathbf{H}$ *et* $\mathbf{V}$ *deux matrices binaires de taille respective* $M \times K$ *et* $P \times K$*. Si, après une éventuelle permutation conjointe des colonnes de* $\mathbf{H}$ *et* $\mathbf{V}$*, il existe une partition* $\mathbf{H} = [\mathbf{H}_1|\mathbf{H}_2]$ *et* $\mathbf{V} = [\mathbf{V}_1|\mathbf{V}_2]$ *avec* $\mathbf{H}_1$ *et* $\mathbf{V}_1$ *de tailles respectives* $M \times K_1$ *et* $P \times K_1$*, et* $\mathbf{H}_2$ *et* $\mathbf{V}_2$ *de tailles respectives* $M \times (K - K_1)$ *et* $P \times (K - K_1)$*, alors* $rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{H}) = K$ *si :*

$$\begin{cases} \forall\ k_1 \in \{1, \cdots, K_1\},\ \mathbf{h}_{k_1} \overset{\mathbb{B}}{\approx} H^{\{-k_1\}} \ \textit{et}\ \forall\ k_2 \in \{K_1 + 1, \cdots, K\},\ \mathbf{v}_{k_2} \overset{\mathbb{B}}{\approx} V^{\{-k_2\}} \\ \qquad\qquad ou \\ \forall\ k_1 \in \{1, \cdots, K_1\},\ \mathbf{v}_{k_1} \overset{\mathbb{B}}{\approx} V^{\{-k_1\}} \ \textit{et}\ \forall\ k_2 \in \{K_1 + 1, \cdots, K\},\ \mathbf{h}_{k_2} \overset{\mathbb{B}}{\approx} H^{\{-k_2\}} \end{cases}.$$

En d'autres termes, le lemme 3.1.2 dit que si, pour $i, j \in \{1, 2\}, i \neq j$, chacune des colonnes de $\mathbf{H}_i$ et chacune des colonnes de $\mathbf{V}_j$ est booléennement indépendante respectivement des autres colonnes de $\mathbf{H}$ et des autres colonnes de $\mathbf{V}$, alors $\mathbf{V} \odot \mathbf{H}$ est de rang colonne plein.

Cela nous permet de donner une autre condition suffisante d'unicité de la DCPB, moins forte que la condition du théorème 3.1.3.

Théorème 3.1.4 (Condition suffisante d'unicité de la DCPB fondée sur les partitions)**.** *Soit la DCPB :* $\mathcal{X} = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]$*. Si*

$$\begin{cases} rg_{\mathbb{B}}(\mathbf{W}) = K,\ \textit{et}\ \mathbf{H}\ \textit{et}\ \mathbf{V}\ \textit{vérifient le lemme}\ 3.1.2 \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{H}) = K,\ \textit{et}\ \mathbf{V}\ \textit{et}\ \mathbf{W}\ \textit{vérifient le lemme}\ 3.1.2 \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{V}) = K,\ \textit{et}\ \mathbf{H}\ \textit{et}\ \mathbf{W}\ \textit{vérifient le lemme}\ 3.1.2 \end{cases},$$

alors la décomposition est unique.

Démonstration. La démonstration de ce théorème découle immédiatement de l'application du lemme 3.1.2 à la condition d'unicité sur les dépliements, *i.e.*,

$$\begin{cases} rg_{\mathbb{B}}(\mathbf{W}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{H}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{H}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{V} \odot \mathbf{W}) = K \\ \qquad\qquad ou \\ rg_{\mathbb{B}}(\mathbf{V}) = K \text{ et } rg_{\mathbb{B}}(\mathbf{H} \odot \mathbf{W}) = K \end{cases}.$$

□

L'exemple suivant permet d'illustrer le fait que la condition du théorème 3.1.4 est moins forte que celle du théorème 3.1.1 .

Exemple 3.1.3. *Considérons* $\mathcal{X} = [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]$, *avec une décomposition de rang* $K = 4$ *tel que :*

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}, \mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \mathbf{V} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Alors nous avons : $rg_{\mathbb{B}}(\mathbf{W}) = 4$, $rg_{\mathbb{B}}(\mathbf{H}) = 2$ *et* $rg_{\mathbb{B}}(\mathbf{V}) = 2$.
La condition du théorème 3.1.1 n'est pas vérifiée dans ce cas. Par contre, le théorème 3.1.4 est vérifié car, si on considère les deux partitions (rouge et bleu), chacune des colonnes en rouge de $\mathbf{H}_1$ *est booléennement indépendante des autres colonnes de* $\mathbf{H}$ *et chacune des colonnes en bleu de* $\mathbf{V}_2$ *est booléennement indépendante des autres colonnes de* $\mathbf{V}$. *La décomposition est donc unique.*

3.2 Algorithmes de DCPB par modèle de mélange post non-linéaire

3.2.1 Modèle de mélange post non-linéaire de la DCPB

Le problème de la DCPB est de retrouver, à partir d'un tenseur à valeurs binaires $\mathcal{X}$, trois matrices à valeurs binaires $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$, solution du problème de minimisation suivant :

$$\{\mathbf{W}, \mathbf{H}, \mathbf{V}\} = \arg \min_{\mathbf{W},\mathbf{H},\mathbf{V} \in \{0,1\}} \|\mathcal{X} - [\![\mathbf{W}, \mathbf{H}, \mathbf{V}]\!]\|_2^2 . \tag{3.11}$$

Les algorithmes développés pour réaliser la DCPB reposent sur une procédure de minimisation alternée qui s'apparente aux méthodes de moindres carrés alternées (ALS)

[13, 75]. Par dépliement du tenseur binaire $\mathcal{X}$ suivant les trois modes, la DCPB (3.11) peut s'exprimer selon :

$$\mathbf{X}_{(1)} = \mathbf{W} \diamond (\mathbf{V} \odot \mathbf{H})^T, \tag{3.12}$$

$$\mathbf{X}_{(2)} = \mathbf{H} \diamond (\mathbf{V} \odot \mathbf{W})^T, \tag{3.13}$$

$$\mathbf{X}_{(3)} = \mathbf{V} \diamond (\mathbf{H} \odot \mathbf{W})^T. \tag{3.14}$$

En adoptant le modèle de mélange post non-linéaire proposé dans la section 2.1, on peut réécrire (3.12), (3.13) et (3.14) selon :

$$\mathbf{X}_{(1)} = \Phi\left(\mathbf{W}(\mathbf{V} \odot \mathbf{H})^T\right), \tag{3.15}$$

$$\mathbf{X}_{(2)} = \Phi\left(\mathbf{H}(\mathbf{V} \odot \mathbf{W})^T\right), \tag{3.16}$$

$$\mathbf{X}_{(3)} = \Phi\left(\mathbf{V}(\mathbf{H} \odot \mathbf{W})^T\right). \tag{3.17}$$

Le problème 3.11 est résolu par minimisation alternée selon la procédure itérative :

Répéter

$$\begin{aligned} \mathbf{W} &= \underset{\mathbf{W}\in\{0,1\}}{\arg\min} \left\|\mathbf{X}_{(1)} - \Phi\left(\mathbf{W}(\mathbf{V} \odot \mathbf{H})^T\right)\right\|_F^2 \\ \mathbf{H} &= \underset{\mathbf{H}\in\{0,1\}}{\arg\min} \left\|\mathbf{X}_{(2)} - \Phi\left(\mathbf{H}(\mathbf{V} \odot \mathbf{W})^T\right)\right\|_F^2 \\ \mathbf{V} &= \underset{\mathbf{V}\in\{0,1\}}{\arg\min} \left\|\mathbf{X}_{(2)} - \Phi\left(\mathbf{V}(\mathbf{H} \odot \mathbf{W})^T\right)\right\|_F^2 \end{aligned} \tag{3.18}$$

jusqu'à convergence

Deux algorithmes sont proposés : le premier, appelé T-PNL-PF pour *Tensor Post Non-Lineair Penalty Function* repose sur une méthode de descente de gradient avec une mise à jour multiplicative similaire à l'approche PNL-PF présentée à la section 2.2. Le second repose sur une approche de type AO-ADMM [103, 104] que l'on spécialise au cas de la DCPB.

3.2.2 L'algorithme T-PNL-PF

Les fonctions de coût associées à chaque sous-problème de (3.18) sont remplacées par les fonctions de coût pénalisées suivantes :

$$J_1(\mathbf{W}) = \frac{1}{2}\left\|\mathbf{X}_{(1)} - \Phi\left(\mathbf{W}(\mathbf{V} \odot \mathbf{H})^T\right)\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{W} - \mathbf{W} * \mathbf{W}\right\|_F^2 \tag{3.19}$$

$$J_2(\mathbf{H}) = \frac{1}{2}\left\|\mathbf{X}_{(2)} - \Phi\left(\mathbf{H}(\mathbf{V} \odot \mathbf{W})^T\right)\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{H} - \mathbf{H} * \mathbf{H}\right\|_F^2 \tag{3.20}$$

$$J_3(\mathbf{V}) = \frac{1}{2}\left\|\mathbf{X}_{(3)} - \Phi\left(\mathbf{V}(\mathbf{H} \odot \mathbf{W})^T\right)\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{V} - \mathbf{V} * \mathbf{V}\right\|_F^2 \tag{3.21}$$

avec ($\odot$) qui représente le produit de Khatri-Rao et ($*$) le produit de Hadamard. Les termes de régularisation de la forme $\|\mathbf{W} - \mathbf{W} * \mathbf{W}\|_F^2$ permettent d'imposer la contrainte de binarité des matrices facteur. Le calcul des règles de mise à jour multiplicatives est similaire à celui de PNL-PF et conduit à l'algorithme T-PNL-PF qui est donné dans la table algorithmique 7.

Algorithm 7 : Tensor-Post NonLinear-Penalty Function (T-PNL-PF)

Entrées : $\mathcal{X}$, K, Nb_{iter}, ε

Sorties : $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$

PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$ et $iter = 0$

PAS 2 : Normalisation

$$\mathbf{W} = \Phi(\mathbf{W}),\ \mathbf{H} = \Phi(\mathbf{H}),\ \mathbf{V} = \Phi(\mathbf{V})$$

PAS 3 : Mise à jour de $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{\left(\mathbf{X}_{(1)} \frac{\partial}{\partial \mathbf{W}_{ik}} \Phi\left(\mathbf{W}\left(\mathbf{H} \odot \mathbf{V}\right)^T\right)\right)_{ik} + 3\lambda \mathbf{W}_{ik}^2}{\left(\frac{\partial}{\partial \mathbf{W}_{ik}} \Phi\left(\mathbf{W}\left(\mathbf{H} \odot \mathbf{V}\right)^T\right)\right)_{ik} + 2\lambda {\mathbf{W}_{ik}}^3 + \lambda \mathbf{W}_{ik}}$$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left(\mathbf{X}_{(2)} \frac{\partial}{\partial \mathbf{H}_{jk}} \Phi\left(\mathbf{H}\left(\mathbf{V} \odot \mathbf{W}\right)^T\right)\right)_{jk} + 3\lambda \mathbf{H}_{jk}^2}{\left(\frac{\partial}{\partial \mathbf{H}_{jk}} \Phi\left(\mathbf{H}\left(\mathbf{V} \odot \mathbf{W}\right)^T\right)\right)_{jk} + 2\lambda {\mathbf{H}_{jk}}^3 + \lambda \mathbf{H}_{jk}}$$

$$\mathbf{V}_{\ell k} \leftarrow \mathbf{V}_{\ell k} \frac{\left(\mathbf{X}_{(3)} \frac{\partial}{\partial \mathbf{V}_{\ell k}} \Phi\left(\mathbf{V}\left(\mathbf{H} \odot \mathbf{W}\right)^T\right)\right)_{\ell k} + 3\lambda \mathbf{V}_{\ell k}^2}{\left(\frac{\partial}{\partial \mathbf{V}_{\ell k}} \Phi\left(\mathbf{V}\left(\mathbf{H} \odot \mathbf{W}\right)^T\right)\right)_{\ell k} + 2\lambda {\mathbf{V}_{\ell k}}^3 + \lambda \mathbf{V}_{\ell k}}$$

PAS 4 : Critère d'arrêt

$$\hat{\mathcal{X}} = \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k$$

if $iter \geq Nb_{iter}$ or $\|\mathcal{X} - \hat{\mathcal{X}}\|_F^2 + \sum_{i,k} \left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \sum_{j,k} \left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2 + \sum_{\ell,k} \left(\mathbf{V}_{\ell,k}^2 - \mathbf{V}_{\ell,k}\right)^2 < \varepsilon$
then
  break
else
  $iter \leftarrow iter + 1$
  retourner au PAS 3
end if

L'initialisation de l'algorithme influe sur la qualité de l'estimation. De façon analogue au cas de PNL-PF, nous avons choisi d'initialiser $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ avec la NTF (*Non-negative Tensor Factorization*) [105] qui, en général, fournit une meilleure solution qu'une initialisation aléatoire. Dans l'étape de normalisation, nous utilisons la fonction sigmoïde pour imposer aux valeurs de $\mathbf{W}_{ij}$, $\mathbf{H}_{ij}$ et $\mathbf{V}_{ij}$ d'être comprises dans l'intervalle [0, 1]. Le pa-

ramètre γ de la sigmoïde augmente en fonction du nombre d'itération selon une règle déterministe : $\gamma^{(i+1)} = \alpha\gamma_{(i)},\ \alpha > 1$. La valeur de α a été fixé empiriquement à 1.04.

3.2.3 L'algorithme BT-ADMM

Le second algorithme proposé s'inspire de [104] où à chaque itération, chacun des sous-problèmes de (3.18) est résolu par une approche de type ADMM (*Alternating Direction Method of Multipliers*). Nous utilisons l'algorithme ADMM pour résoudre chaque sous-problème de 3.18. Ainsi, l'algorithme ADMM permet de résoudre des problèmes d'optimisation pouvant être écrits sous la forme :

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} f(x) + g(z) \\ &\text{s.c. } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \tag{3.22}$$

en itérant les étapes de mise à jour suivantes :

$$\begin{aligned} &\mathbf{x} \leftarrow \arg\min_{x}\ f(x) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}\right\|_F^2, \\ &\mathbf{z} \leftarrow \arg\min_{z}\ g(z) + \frac{\rho}{2}\left\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}\right\|_F^2 \\ &\mathbf{u} \leftarrow \mathbf{u} + (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) \end{aligned} \tag{3.23}$$

où $\mathbf{u}$ est une version mise à l'échelle des variables duales correspondant à la contrainte d'égalité $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}$, et ρ est choisi par l'utilisateur pour garantir la convergence. Les détails de l'algorithme ADMM et l'analyse de ses propriétés sont donnés dans [103]. La convergence de l'algorithme ADMM est garantie uniquement dans le cas convexe. Si l'on revient au problème de la DCPB, l'introduction de la fonction non-linéaire Φ fait perdre à chaque sous-problème la propriété de convexité et par suite la convergence de chaque sous-problème ne peut pas être garantie.
Considérons le problème de minimisation de la fonction coût :

$$\min_{\mathbf{W}}\ J_1(\mathbf{W}) = \frac{1}{2}\left\|\mathbf{X}_{(1)} - \Phi\left(\mathbf{W}(\mathbf{V}\odot\mathbf{H})^T\right)\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{W} - \mathbf{W} * \mathbf{W}\right\|_F^2.$$

En introduisant la variable auxiliaire $\bar{\mathbf{W}}$, on peut le réécrire sous la forme :

$$\begin{aligned} &\min_{\mathbf{W},\bar{\mathbf{W}}}\ \frac{1}{2}\left\|\mathbf{X}_{(1)} - \Phi\left(\bar{\mathbf{W}}\,(\mathbf{V}\odot\mathbf{H})^T\right)\right\|_F^2 + \frac{\lambda}{2}\left\|\mathbf{W} - \mathbf{W} * \mathbf{W}\right\|_F^2 \\ &\text{s.c. } \mathbf{W} = \bar{\mathbf{W}}, \end{aligned} \tag{3.24}$$

qui correspond exactement à un problème de la forme (3.22) dont la solution est obtenue en itérant les étapes de mise à jour :

$$\bar{\mathbf{W}} \leftarrow \arg\min_{\bar{\mathbf{W}}} \frac{1}{2} \left\| \mathbf{X}_{(1)} - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{W} - \bar{\mathbf{W}} + \mathbf{A} \right\|_F^2 \tag{3.25}$$

$$\mathbf{W} \leftarrow \arg\min_{\mathbf{W}} \ \frac{\lambda}{2} \left\| \mathbf{W} - \mathbf{W} * \mathbf{W} \right\|_F^2 + \frac{\rho}{2} \left\| \mathbf{W} - \bar{\mathbf{W}} + \mathbf{A} \right\|_F^2 \tag{3.26}$$

$$\mathbf{A} = \mathbf{A} + \mathbf{W} - \bar{\mathbf{W}}. \tag{3.27}$$

Compte tenu de la présence de la fonction non-linéaire Φ, la minimisation du premier terme (3.25) n'admet pas de solution explicite. L'idée est de procéder à une approximation quadratique locale de $\left\| \mathbf{X}_{(1)} - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) \right\|_F^2$ en utilisant un développement de Taylor au premier ordre de la fonction sigmoïde Φ selon : $\Phi(y) \approx \Phi(y_0) + \frac{\partial \Phi(y)}{\partial y}\Big|_{y_0} (y - y_0)$. Or

$$\begin{aligned}
\frac{\partial \Phi(y)}{\partial y} &= \frac{\partial}{\partial y} \left(\frac{1}{1 + e^{-\gamma(y-0.5)}} \right) = \frac{\gamma e^{-\gamma(y-0.5)}}{(1 + e^{-\gamma(y-0.5)})^2} \\
&= \frac{\gamma}{1 + e^{-\gamma(y-0.5)}} \left(1 - \frac{1}{1 + e^{-\gamma(y-0.5)}} \right) \\
&= \gamma \left(\frac{1}{1 + e^{-\gamma(y-0.5)}} \right) \left(\frac{1 + e^{-\gamma(y-0.5)} - 1}{1 + e^{-\gamma(y-0.5)}} \right) \\
&= \frac{\gamma e^{-\gamma(y-0.5)}}{(1 + e^{-\gamma(y-0.5)})^2} \\
&= \gamma \ \Phi(y) \left(1 - \Phi(y)\right)
\end{aligned}$$

En reportant cette expression dans le développement de Taylor, on obtient :

$$\Phi(y) \approx \Phi(y_0) - \gamma \Phi(y_0)(1 - \Phi(y_0)) y_0 + \gamma \left(\Phi(y_0) - \Phi(y_0)^2 \right) y$$

En reportant cette expression dans $\left\| \mathbf{X}_{(1)} - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) \right\|_F^2$, on obtient :

$$\left\| \mathbf{X}_{(1)} - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) \right\|_F^2 \approx \left\| \mathbf{X}_{(1)} - \mathbf{E}^w - \mathbf{F}^w * (\bar{\mathbf{W}} (\mathbf{V} \odot \mathbf{H})^T) \right\|_F^2$$

avec

$$\mathbf{E}^w = \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) - \gamma \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) * \left(\mathbf{1} - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right)\right) * \left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right)$$

$$\mathbf{F}^w = \gamma \left(\Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) - \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) * \Phi\left(\bar{\mathbf{W}} \left(\mathbf{V} \odot \mathbf{H}\right)^T\right) \right)$$

Notons que les expressions de $\mathbf{E}^w$ et $\mathbf{F}^w$ sont obtenues avec la valeur de $\bar{\mathbf{W}}$ obtenue à l'itération précédente et doivent être considérées comme des constantes. Ainsi on obtient une forme explicite pour $\bar{\mathbf{W}}$ solution de (3.25) (voir annexe A.4) :

$$\bar{\mathbf{W}} \leftarrow \frac{\left(\left((\mathbf{X}_{(1)} - \mathbf{E}^w) * \mathbf{F}^w \right) (\mathbf{V} \odot \mathbf{H}) \right) + \rho \left(\mathbf{W} + \mathbf{A} \right)}{\left((\mathbf{F}^w * \mathbf{F}^w) \left((\mathbf{V} \odot \mathbf{H}) * (\mathbf{V} \odot \mathbf{H}) \right) \right) + \rho \mathbf{1}}$$

$\mathbf{W}$ est obtenue avec une mise à jour multiplicative. La minimisation du problème (3.26) admet donc une solution explicite :

$$\mathbf{W} \leftarrow \mathbf{W} - \alpha_{iter} \left(\lambda \left(\mathbf{W} - 3\mathbf{W}^2 + 2\mathbf{W}^3 \right) + \rho \left(\mathbf{W} - \bar{\mathbf{W}} + \mathbf{A} \right) \right)$$

En procédant de même pour $\mathbf{H}$ et $\mathbf{V}$ on obtient l'algorithme BT-ADMM donné dans la table algorithmique 8.

L'initialisation et la normalisation de l'algorithme BT-ADMM se font comme dans l'algorithme T-PNL-PF, *i.e.,* l'initialisation avec les résultats de la NTF [105] et la normalisation avec la fonction sigmoïde Φ. Egalement, le paramètre γ qui permet de régler la pente de la fonction sigmoïde augmente avec le nombre d'itérations. Dans cet algorithme Nb_{iter} représente le nombre d'itération de la procédure de minimisation alternée tandis que $Nb_{interne}$ représente le nombre d'itérations de chaque procédure ADMM.

Le choix du paramètre ρ influe fortement sur la convergence de l'algorithme. Pour illustrer ce point, considérons l'exemple suivant. Le tenseur $\mathcal{X} = \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k$ est obtenu en tirant $\mathbf{w}_k$, $\mathbf{h}_k$ et $\mathbf{v}_k$ $(k = 1, \cdots, K)$ selon une loi de Bernoulli de paramètre $p = 0.6$. La figure 3.3 montre le taux d'erreur de reconstruction de $\mathcal{X}$ $(Error_X)$ en fonction du paramètre ρ. Le taux d'erreur de reconstruction est défini par :

$$Error_X = \frac{\left\| \mathbf{X}_{(1)} - \hat{\mathbf{X}}_{(1)} \right\|_F^2}{N_{(1)} M_{(1)}}, \tag{3.28}$$

avec $N_{(1)}$ et $M_{(1)}$ le nombre de lignes et de colonnes de $\mathbf{X}_{(1)}$. Nous pouvons observer que seules les grandes valeurs de ρ permettent de garantir la convergence de l'algorithme. Dans la suite la valeur de ρ est prise égale à 10^9.

Algorithm 8 : BT-ADMM

Entrées : $\mathcal{X}$, K, Nb_{iter}, $Nb_{interne}$, ρ, λ, ε

Sorties : $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$

PAS 1 : Initialisation de $\mathbf{W}$, $\mathbf{H}$, $\mathbf{V}$, $\bar{\mathbf{W}}$, $\bar{\mathbf{H}}$ et $\bar{\mathbf{V}}$

PAS 2 : Normalisation de $\bar{\mathbf{W}}$, $\mathbf{W}$, $\bar{\mathbf{H}}$, $\mathbf{H}$, $\bar{\mathbf{V}}$ et $\mathbf{V}$

for $t = 1 : Nb_{iter}$ **do**

  Mise à jour de $\bar{\mathbf{W}}$ et $\mathbf{W}$

  for $t_1 = 1 : Nb_{interne}$ **do**

    $\mathbf{E}^w = \Phi\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right) - \gamma\Phi\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right)\left(1 - \Phi\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right)\right)\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right)$

    $\mathbf{F}^w = \gamma\left(\Phi\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right) - \left(\Phi\left(\bar{\mathbf{W}}\left(\mathbf{V} \odot \mathbf{H}\right)\right)\right)^2\right)$

    $\bar{\mathbf{W}} \leftarrow \frac{\left(\left(\left(\mathbf{X}_{(1)} - \mathbf{E}^w\right) * \mathbf{F}^w\right)(\mathbf{V} \odot \mathbf{H})\right) + \rho(\mathbf{W} + \mathbf{A})}{\left(\left(\mathbf{F}^w * \mathbf{F}^w\right)\left((\mathbf{V} \odot \mathbf{H}) * (\mathbf{V} \odot \mathbf{H})\right)\right) + \rho\mathbf{1}}$

    $\mathbf{W} \leftarrow \mathbf{W} - \alpha_{iter}\left(\lambda\left(\mathbf{W} - 3\mathbf{W}^2 + 2\mathbf{W}^3\right) + \rho\left(\mathbf{W} - \bar{\mathbf{W}} + \mathbf{A}\right)\right)$

    $\mathbf{A} = \mathbf{A} + \mathbf{W} - \bar{\mathbf{W}}$

  end for

  Mise à jour de $\bar{\mathbf{H}}$ et $\mathbf{H}$

  for $t_2 = 1 : Nb_{interne}$ **do**

    $\mathbf{E}^h = \Phi\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right) - \gamma\Phi\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right)\left(1 - \Phi\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right)\right)\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right)$

    $\mathbf{F}^h = \gamma\left(\Phi\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right) - \left(\Phi\left(\bar{\mathbf{H}}\left(\mathbf{V} \odot \mathbf{W}\right)\right)\right)^2\right)$

    $\bar{\mathbf{H}} \leftarrow \frac{\left(\left(\left(\mathbf{X}_{(2)} - \mathbf{E}^h\right) * \mathbf{F}^h\right)(\mathbf{V} \odot \mathbf{W})\right) + \rho(\mathbf{H} + \mathbf{B})}{\left(\left(\mathbf{F}^h * \mathbf{F}^h\right)\left((\mathbf{V} \odot \mathbf{W}) * (\mathbf{V} \odot \mathbf{W})\right)\right) + \rho\mathbf{1}}$

    $\mathbf{H} \leftarrow \mathbf{H} - \alpha_{iter}\left(\lambda\left(\mathbf{H} - 3\mathbf{H}^2 + 2\mathbf{H}^3\right) + \rho\left(\mathbf{H} - \bar{\mathbf{H}} + \mathbf{B}\right)\right)$

    $\mathbf{B} = \mathbf{B} + \mathbf{H} - \bar{\mathbf{H}}$

  end for

  Mise à jour de $\bar{\mathbf{V}}$ et $\mathbf{V}$

  for $t_3 = 1 : Nb_{interne}$ **do**

    $\mathbf{E}^v = \Phi\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right) - \gamma\Phi\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right)\left(1 - \Phi\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right)\right)\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right)$

    $\mathbf{F}^v = \gamma\left(\Phi\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right) - \left(\Phi\left(\bar{\mathbf{V}}\left(\mathbf{H} \odot \mathbf{W}\right)\right)\right)^2\right)$

    $\bar{\mathbf{V}} \leftarrow \frac{\left(\left(\left(\mathbf{X}_{(3)} - \mathbf{E}^v\right) * \mathbf{F}^v\right)(\mathbf{H} \odot \mathbf{W})\right) + \rho(\mathbf{V} + \mathbf{C})}{\left(\left(\mathbf{F}^v * \mathbf{F}^v\right)\left((\mathbf{H} \odot \mathbf{W}) * (\mathbf{H} \odot \mathbf{W})\right)\right) + \rho\mathbf{1}}$

    $\mathbf{V} \leftarrow \mathbf{V} - \alpha_{iter}\left(\lambda\left(\mathbf{V} - 3\mathbf{V}^2 + 2\mathbf{V}^3\right) + \rho\left(\mathbf{V} - \bar{\mathbf{V}} + \mathbf{C}\right)\right)$

    $\mathbf{C} = \mathbf{C} + \mathbf{V} - \bar{\mathbf{V}}$

  end for

  $\bar{\mathbf{W}} = \Phi(\bar{\mathbf{W}})$, $\bar{\mathbf{H}} = \Phi(\bar{\mathbf{H}})$, $\bar{\mathbf{V}} = \Phi(\bar{\mathbf{V}})$, $\mathbf{W} = \Phi(\mathbf{W})$, $\mathbf{H} = \Phi(\mathbf{H})$, $\mathbf{V} = \Phi(\mathbf{V})$

  PAS 4 : critère d'arrêt

  $\hat{\mathcal{X}} = \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k$

  if $\|\mathcal{X} - \hat{\mathcal{X}}\|^2 + \sum_{i,k}\left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \sum_{j,k}\left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2 + \sum_{\ell,k}\left(\mathbf{V}_{\ell,k}^2 - \mathbf{V}_{\ell,k}\right)^2 < \varepsilon$ **then**

    break

  end if

end for

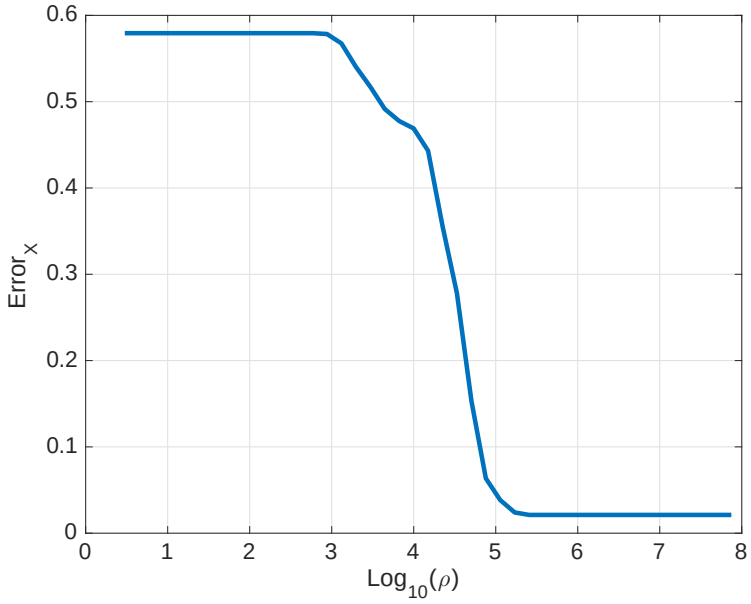


FIGURE 3.3 – Taux d'erreur de reconstruction de $\mathcal{X}$ en fonction du paramètre ρ ($N = 12$, $M = 9$, $P = 7$, $K = 3$, $Nb_{iter} = 100$, $Nb_{interne} = 10$) pour $p = 0.6$

3.3 Simulations numériques

Dans cette section, nous étudions le comportement de nos algorithmes à la présence de bruit binaire. Lors des simulations, le modèle direct ci-après est utilisé :

$$\mathcal{X}_n = \bigvee_{k=1}^{K} \mathbf{w}_k \circ \mathbf{h}_k \circ \mathbf{v}_k \oplus \mathcal{N} = \mathcal{X} \oplus \mathcal{N},$$

où $\mathcal{N}$ représente le tenseur de bruit binaire. Ainsi, $\mathcal{N}$ est la réalisation d'une variable de Bernoulli de paramètre b. Dans ces simulations, nous évaluons l'erreur de reconstruction sur $\mathcal{X}$ définie précédemment (3.28) et l'erreur d'estimation sur $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ ($Error_{W-H-V}$) définie selon :

$$Error_{W-H-V} = \frac{\frac{\left\|\mathbf{W}-\hat{\mathbf{W}}\right\|_F^2}{N\times K} + \frac{\left\|\mathbf{H}-\hat{\mathbf{H}}\right\|_F^2}{M\times K} + \frac{\left\|\mathbf{V}-\hat{\mathbf{V}}\right\|_F^2}{P\times K}}{3}$$

en fonction du taux de bruit rajouté (b) sur 50 réalisations. Chaque colonne de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ est générée comme précédemment, selon une loi de Bernoulli de paramètre p.

Dans ces simulations nous comparons les performances des algorithmes proposés (T-PNL-PF et BT-ADMM) avec les algorithmes de type « glouton » (BCP-ALS et T-FC) présentés dans la section 1.4.2.

Deux cas sont considérés : celui où les sources sont non ou peu corrélées (figures 3.4 et 3.5), *i.e.* $p = 0.3$ et le cas de sources corrélées (figures 3.6 et 3.7), *i.e,* $p = 0.65$.

Dans le cas non corrélé ($p = 0.3$), nos algorithmes donnent un taux d'erreur d'estimation plus faible jusqu'à un taux de bruit $b \leq 0.25$ (figure 3.4) et un taux d'erreur de reconstruction plus faible jusqu'à $b < 0.3$ (figure 3.5). Ceci peut s'expliquer par le choix du modèle de mélange post non-linéaire utilisé dans nos algorithmes. On peut également

remarquer qu'il y a une dégradation forte des performances lorsque $b > 0.1$. Ceci s'explique par le fait que lorsque chaque colonne de $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{V}$ est tirée selon une loi de Bernoulli, le tenseur résultant est creux (il y a un grand nombre de valeurs nulles). En effet chaque terme de rang 1 correspond à la réalisation d'une variable de Bernoulli de paramètre $0.3^3 = 0.0027$ ce qui est très faible par rapport à $b = 0.1$.

Dans le cas corrélé ($p = 0.65$), nous pouvons observer que nos algorithmes donnent un taux d'erreur de reconstruction, particulièrement un taux d'erreur d'estimation plus faible jusqu'à un taux d'erreur rajouté $b = 0.30$. En reprenant le même raisonnement que précédemment, chaque terme de rang 1 correspond à la réalisation d'une variable de Bernoulli de paramètre $0.65^3 = 0.275$ soit 100 fois plus que pour le cas $p = 0.3$.

Nous pouvons remarquer que les algorithmes de type « glouton » donnent de bons résultas dans le cas d'une factorisation exacte. Il faut noter aussi qu'au delà de $b = 0.25$ et $b = 0.3$ pour respectivement $p = 0.3$ et $p = 0.65$, nous obtenons des résultats similaires car à partir de cette valeur, le bruit binaire détruit la structure de rang faible de $\mathcal{X}$.

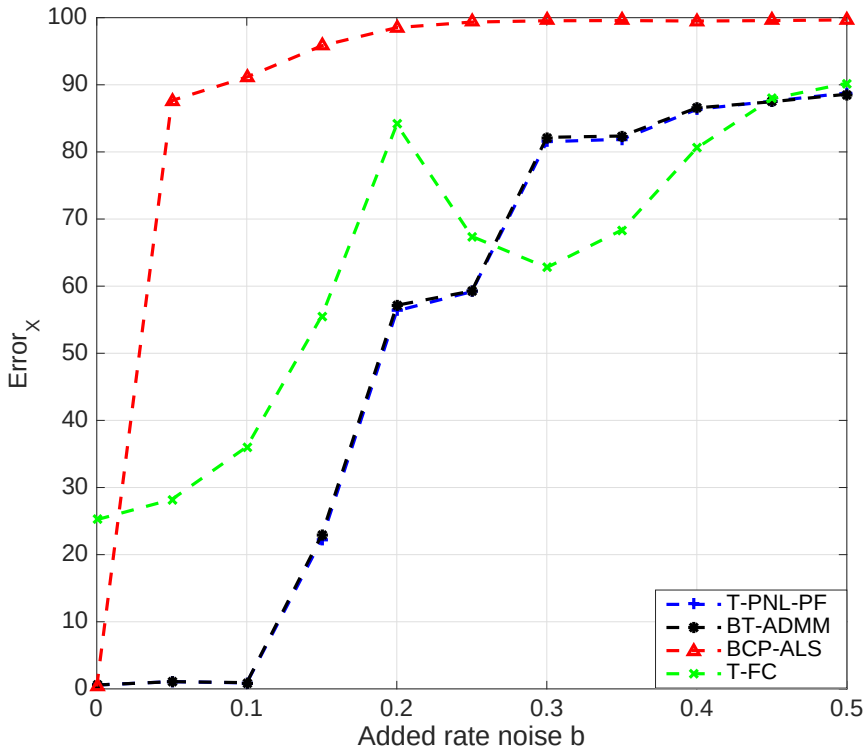


FIGURE 3.4 – Taux de reconstruction de $\mathcal{X}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.3$

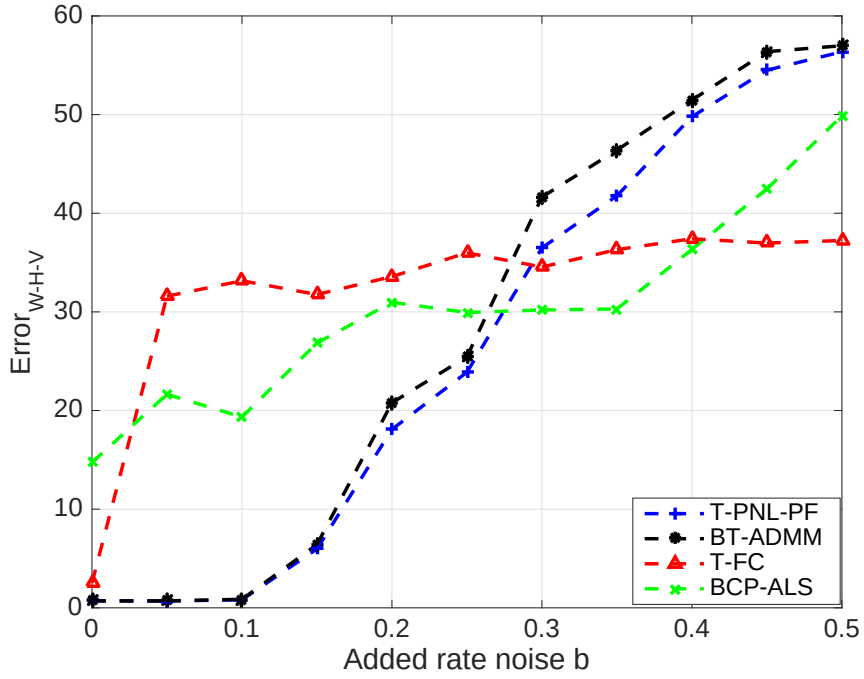


FIGURE 3.5 – Taux d'estimation de $\mathbf{W}$ $\mathbf{H}$ et $\mathbf{V}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.3$

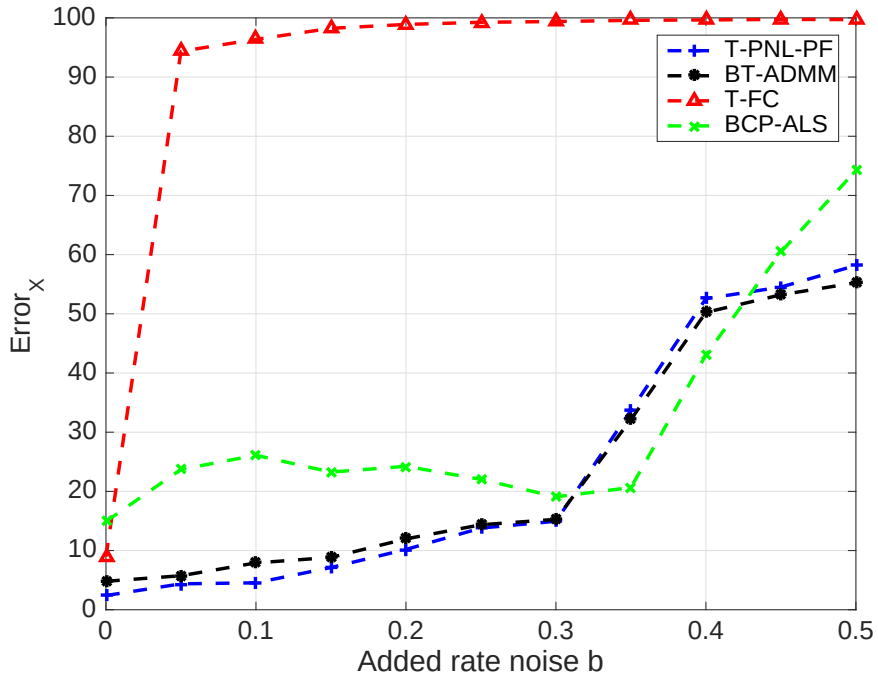


FIGURE 3.6 – Taux de reconstruction de $\mathcal{X}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.65$

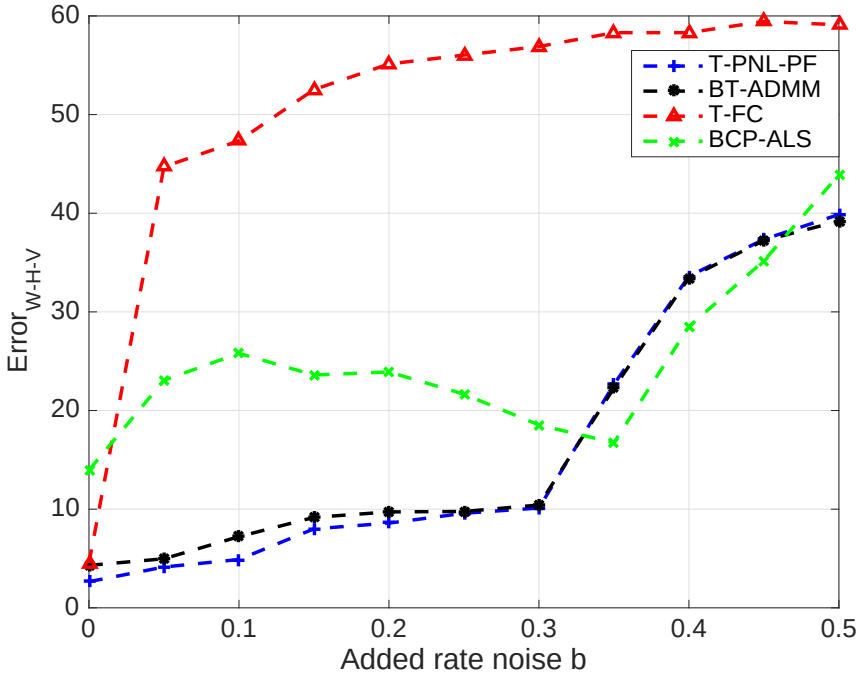


FIGURE 3.7 – Taux d'estimation de $\mathbf{W}$ $\mathbf{H}$ et $\mathbf{V}$ en fonction du taux de bruit rajouté b ($N = 12$, $M = 9$, $P = 7$, $K = 3$, $\rho = 10^9$) pour $p = 0.65$

3.4 Application à des données réelles

3.4.1 Données

Dans cette section, nous exploitons un ensemble de données qui décrit des interactions dans une école primaire (composée de 5 niveaux d'études) à Lyon en octobre 2009 en utilisant des capteurs de proximité qui détectent les relations de face à face (appelées *contacts* ici) des personnes qui les portent [106]. La population de l'école est composée de 241 enfants âgés de 6 à 12 ans, organisée en 10 classes (ou 5 niveaux) et 10 enseignants ; seuls 232 élèves ont participé à cette étude (voir tableau 3.1). Ainsi, cet ensemble de données présente des structures attendues (classes) mais également des structures potentielles moins visibles. Cela apparaît donc comme un ensemble de données idéal pour évaluer l'efficacité des algorithmes de DCPB.

Niveaux d'études	1		2		3		4		5		Enseignants
Classes	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	enseignants
Nombre d'individus	23	25	23	26	23	22	21	23	22	24	10

TABLE 3.1 – Le nombre d'élèves de chaque classe et de enseignants qui ont participé à l'étude

Les contacts sont enregistrés toutes les 20 secondes et un contact est établi entre deux personnes sur un intervalle de 20 secondes s'il y a un échange d'au moins 1 paquet radio pendant cet intervalle. Tant que leurs dispositifs RFID continuent à échanger au moins un paquet, pour chaque intervalle de 20 secondes, le contact est maintenu ; dans le cas où il n'y a pas d'échange radio sur un intervalle de 20 secondes, le contact est considéré comme rompu. Les données ont été collectées sur deux jours, de 8h45 à 17h20 le premier jour, de 8h30 à 17h05 le deuxième jour, et seules les interactions qui ont eues lieu dans les locaux de l'école (cours de récréation, salles de cours, escaliers et la cantine) ont été détectées. Les cours commencent à 8h30 et se terminent à 16h30. Il y a deux pauses (récréations) de 20 à 25 minutes vers 10h30 et 15h30 et la pause repas entre 12h et 14h. Puisque la cour de recréation et la cantine n'ont pas assez de capacité pour accueillir tous les élèves au même moment, seules deux ou trois classes ont des pauses (récréations) en même temps et le déjeuner est pris en deux tours consécutifs. Ces données qui représentent les interactions entre les personnes (élèves et enseignants) de cette étude pendant 2 jours peuvent être représentées par le tenseur ci-après :

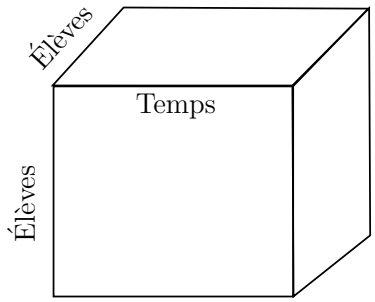


FIGURE 3.8 – Représentation du tenseur de données

Dans la littérature, ces données ont été analysées par factorisation en tenseurs non-négatifs (FTN) [107]. Ils montrent que la FTN est capable de récupérer la structure des classes. En particulier, ils montrent que les composantes ou sources de rang 1 peuvent être validées soit en tant que classes scolaires connues, soit en termes de modèles d'activités corrélées. Dans notre cas, nous appliquerons la DCPB à ces données afin de détecter les comportements communs des élèves d'une même classe ou de classes différentes pendant les moments de cours, repas, recréation, *etc.*

3.4.2 Capacité explicative des algorithmes DCPB

Pour appliquer la DCPB à ces données, nous allons réduire la dimension du tenseur $\mathcal{X}$. Pour cela, nous découperons l'axe du temps en tranches de 10 minutes, ce qui nous donnera un tenseur $\mathcal{X}$ de taille $242 \times 242 \times 105$. Pour chaque tranche de 10 minutes ℓ, si l'individu i a au moins un contact avec l'individu j alors $x_{ij\ell}$=1, sinon 0. Ensuite, nous considérerons qu'un individu i est toujours en contact avec lui même, ce qui revient à dire que $x_{ii\ell} = 1$, $\forall i = 1, \cdots, 242$.

Dans un premier lieu, nous ferons l'étude sur la première journée uniquement et ensuite sur la deuxième journée pour voir si nous arrivons à retrouver des comportements similaires lors de deux journées. Nous noterons respectivement $\mathcal{X}_1$ et $\mathcal{X}_2$, les tenseurs de données de la première et de la deuxième journée.

Pour commencer l'étude sur la première journée, nous déterminons le rang booléen de la décomposition de $\mathcal{X}_1$ qui permet de retrouver les comportements les plus représentatifs. Pour cela, nous traçons l'erreur de reconstruction de $\mathcal{X}_1$, définie comme dans (3.28), en fonction du rang booléen K de la décomposition de $\mathcal{X}_1$. A partir l'analyse de la courbe de la figure 3.9, nous considérons un rang égal à 15 car au-delà, le taux d'erreur diminue moins vite, ce qui indique une perte importante de structure dans les données.

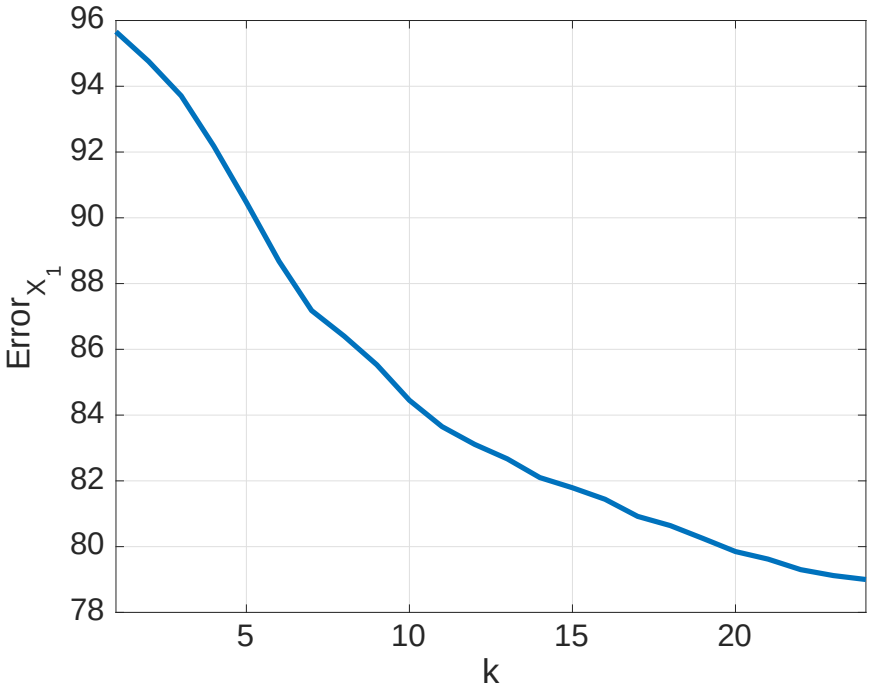


FIGURE 3.9 – Erreur d'approximation de $\mathcal{X}_1$ en fonction du rang booléen K

Pour cette décomposition booléenne de $\mathcal{X}_1$ avec un rang $K = 15$, nous représentons les différentes sources (ou tenseurs de rang 1) pour détecter des comportements similaires sur les individus lors de la première journée. Puisque ces tenseurs sont symétriques, (*i.e.*, si on pose $\mathcal{X}_1 = [\![\mathbf{W}_1, \mathbf{H}_1, \mathbf{V}_1]\!]$ alors nous avons ici $\mathbf{W}_1 = \mathbf{H}_1$), nous représenterons seulement les contacts actifs entre les individus en fonction du temps pour chaque source de rang 1, *i.e.,* $\mathbf{w}_1$ en fonction de $\mathbf{h}_1$, pour les 15 sources.

La figure 3.10 représente les différentes sources (ou tenseurs de rang 1) obtenus après décomposition de $\mathcal{X}_1$. Comme dans [107] et [106], nous arrivons à voir dans la figure 3.10, qu'au cours de la journée, il y a plus de contacts entre les élèves d'une même classe qu'entre ceux de classes différentes (voir sources 1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13 et 15). Dans certaines de ces sources ou classes, il est possible d'identifier les enseignants qui interviennent dans ces classes. C'est le cas des sources 1, 3, 7, 8, 10 et 13 où on arrive à

identifier les enseignants qui interviennent pendant toute la journée respectivement dans les classes 1B, 5B, 2B, 3A, 2A et 5A (les dernières lignes). Dans les sources 1, 5 et 11, nous avons, d'une part les contacts entre les élèves de la classe 1B, et d'autre part, dans les source 1 et 11 on peut voir aussi les contacts entre les élèves de cette classe, avec leurs enseignants. Dans la source 5, nous pouvons, par exemple, detecter les élèves très peu en contact avec leur professeur. Cependant, les élèves qui sont en contact dans la source 5 ne sont pas les mêmes que ceux de la source 11. En effet, nous avons dans cette classe 2 groupes d'élèves qui ont des comportements différents par rapport à leur professeur au cours de la journée. La source 2 nous permet de dire que la plupart des élèves des classes de $5^{\text{ième}}$ (5A et 5B) ont plus d'affinité (contacts) entre eux qu'avec les autres classes pendant les pauses de la première journée. Ainsi nous pouvons voir qu'ils sont ensemble pendant la pause de 10h30 et de midi.
Puisque les enfants des plus petites classes d'une part (de $1^{\text{ière}}$ à la $3^{\text{ième}}$) et ceux des plus grandes classes d'autre part (de $4^{\text{ième}}$ à la $5^{\text{ième}}$) prennent la pause de midi ensemble (selon [106]), nous pouvons observer dans la source 9 et 14 des contacts entre certains élèves de ces classes pendant la pause de midi.

Le tableau 3.2 représente les relations (notamment le nombre de contacts) entre les différentes classes et entre les classes et les enseignants de cette étude [106]. À partir de ce tableau, nous pouvons valider les comportements (sources) obtenus par la DCPB lors de la première journée. Ainsi, la diagonale de ce tableau nous permet de valider les résultats obtenus sur les sources 1, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13 et 15, qui stipulent que les contacts sont plus présents entre les individus d'une même classe. Egalement, le nombre de contacts entre les individus de la classe 1B (voir colonne 2 de ce tableau) et le fait que les enseignants ont plus de contacts avec la classe 1B qu'avec les autres classes (voir dernière colonne de ce tableau) font qu'on a plus de comportements différents entre les individus de la classe 1B (voir sources 1, 5 et 11). Il faut noter aussi qu'on arrive bien à identifier les enseignants qui ont plus de contacts avec les classes. Enfin, notre analyse nous a permis de dire que les élèves des petites classes (1 à 3) ont plus de contacts entre eux qu'avec les élèves des grandes classes.

Nous allons maintenant passer à la décomposition booléenne de $\mathcal{X}_2$, *i.e.,* l'étude sur la deuxième journée afin de voir si nous arrivons à retrouver les mêmes comportements entre les élèves d'une même classe ou/et ceux de classes différentes. Les tenseurs de rang 1 obtenus par la DCPB sur $\mathcal{X}_2$, avec le même rang que précédemment, sont représentés dans la figure 3.11. Ainsi, nous arrivons à retrouver le premier résultat de la première journée, *i.e.*, il y a plus de contacts entre les élèves d'une même classe (voir sources 1, 3, 4, 5, 6, 7, 8, 10, 11, 14 et 15) qu'entre classes différentes. Ensuite nous retrouvons des comportements différents entre les élèves de la classe 1B (sources 6 et 7). Aussi, nous arrivons bien à identifier la plupart des enseignants qui ont plus de contacts avec les classes, notamment ceux des classes 5A, 2B, 1B, 2A, 4A (voir respectivement sources 1, 5, 7, 10 et 14). Egalement, on retrouve les contacts (comportements communs) entre les

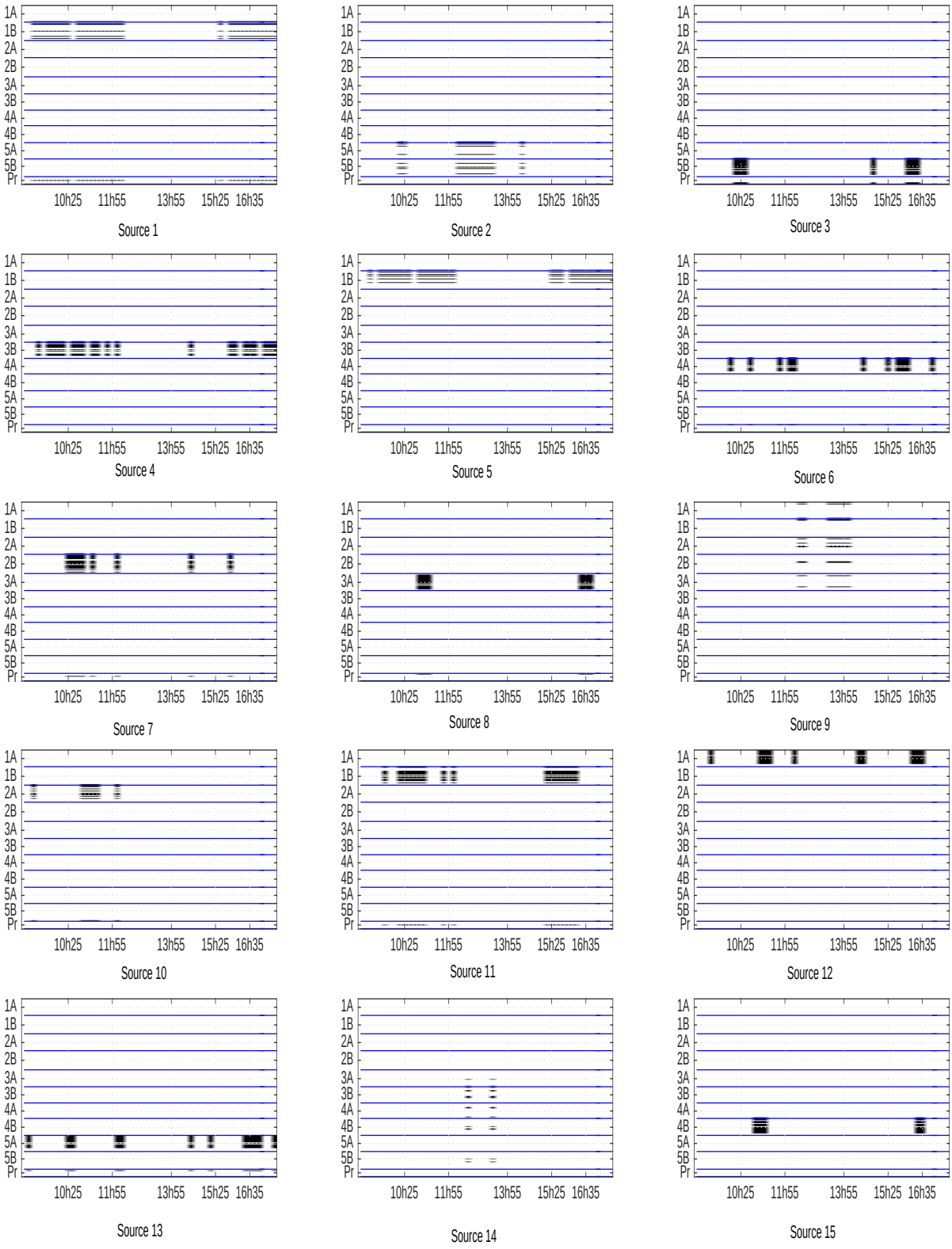


FIGURE 3.10 – Représentation des différentes sources ou tenseurs de rang 1 de $\mathcal{X}_1$

Classes	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	Profs
1A	4505	1051	594	625	560	286	83	160	57	105	149
1B	1051	9756	502	632	269	207	551	161	448	386	1084
2A	594	502	5401	1583	657	360	77	56	76	30	586
2B	625	632	1583	6270	712	373	119	36	41	54	508
3A	560	269	657	712	5537	2076	77	163	109	82	414
3B	286	207	360	373	2076	5926	248	193	154	219	282
4A	83	551	77	119	77	248	4496	828	351	745	382
4B	160	161	56	36	163	193	828	2843	119	346	168
5A	57	448	76	41	109	154	351	119	4913	1968	372
5B	105	386	30	54	82	219	745	119	1968	5025	273
Profs	149	1084	586	508	414	282	382	168	372	273	101

TABLE 3.2 – Tableau récapitulatif des contacts entre les classes pendant les 2 journées

élèves des petites classes et entre ceux des grandes classes pendant les pauses (voir sources 2, 9, 12 et 13).

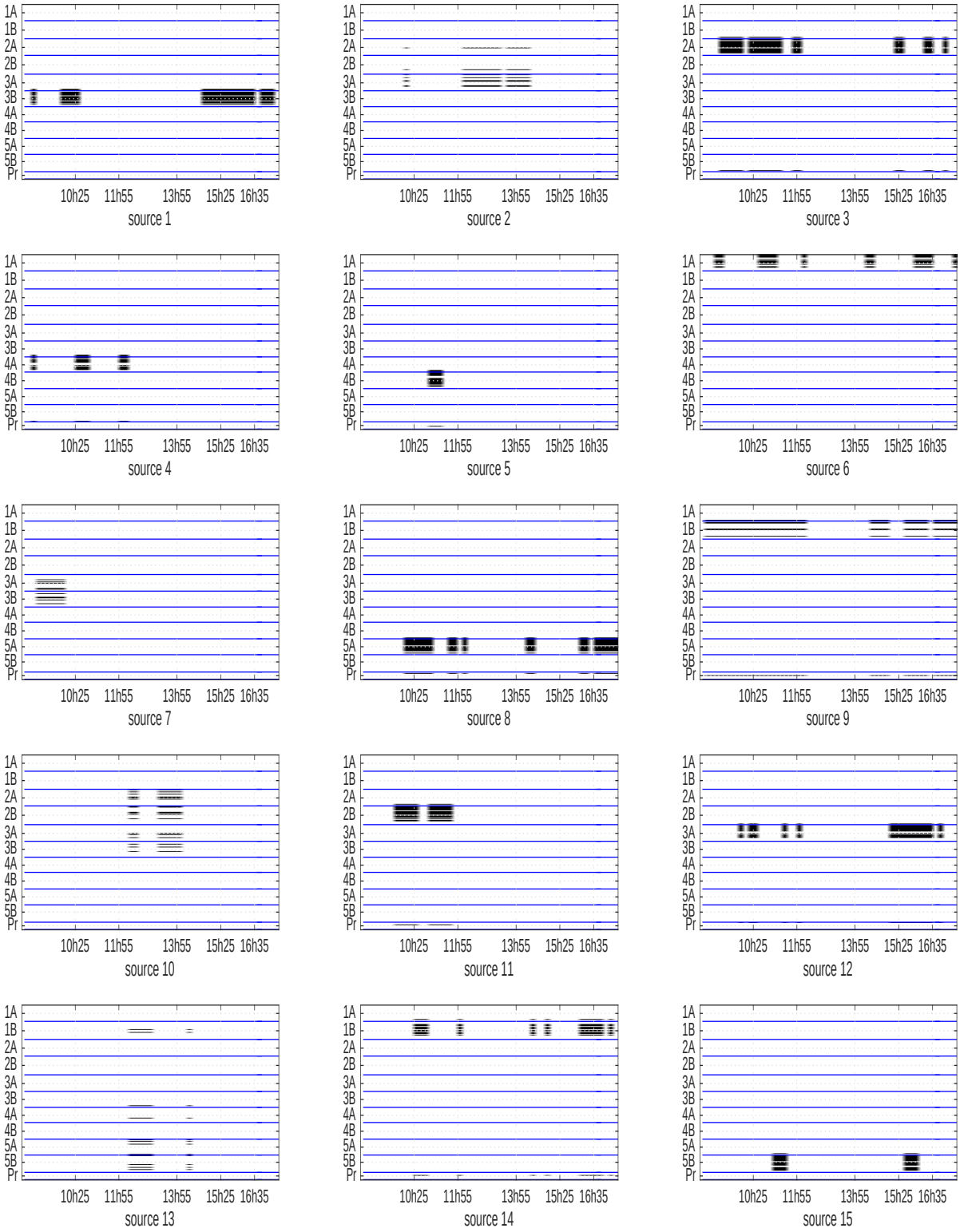


FIGURE 3.11 – Représentation des différentes sources ou tenseurs de rang 1 de $\mathcal{X}_2$

3.5 Conclusion de la DCPB

Nous avons introduit dans ce chapitre deux nouveaux algorithmes reposant sur le modèle de mélange post non-linéaire. Dans le premier algorithme, nous étendons l'algorithme de type multiplicatif proposé pour le cas matriciel au cas multidimensionnel et dans le deuxième algorithme, nous proposons un nouvel algorithme s'appuyant sur une stratégie de type AO-ADMM. Ces algorithmes sont comparés avec ceux de l'état de l'art. Nous avons constaté que les algorithmes proposés donnent de meilleurs résultats notamment dans le cas bruité. Nous avons aussi donné et démontré des conditions d'identifiabilité pour la DCPB. Ces résultats sont à notre connaissance totalement originaux et montrent que l'organisation de données binaires en tableaux multidimensionnels présente un réel intérêt puisque l'on pourra dans une grande majorité des cas, avoir des garanties sur le résultat fourni par la décomposition. L'application de la DCPB à des données réelles a permis d'illustrer cet intérêt sur un exemple réel.

Conclusion

Un premier apport de ce travail est de fournir des conditions d'identifiabilité des modèles de factorisation booléennes de matrices et tenseurs binaires. Ce type de résultats est important car il permet de donner des garanties sur les résultats de décompositions. A notre connaissance, ce travail est le premier à aborder cette problématique. Nous avons notamment pu montrer qu'une FBMB pouvait être unique sous des faibles contraintes. Ce résultat est à opposer au cas de la factorisation de matrices réelles. Le cas de décomposition CP booléenne est similaire et montre que l'unicité est garantie sous des conditions moins strictes que la décomposition CP réelle. Ainsi, il est possible d'avoir l'unicité de la décomposition CPB même en présence de facteurs colinéaires sur un des modes alors que cela est impossible dans le cas réel.

Un second apport de cette thèse est de proposer un nouveau modèle de mélange post non-linéaire pour la factorisation booléenne de matrices et tenseurs binaires. Ce modèle est équivalent au modèle de décomposition booléennes de tableaux binaires. Il peut être vu comme une relaxation réelle exacte du problème booléen qui permet le développement de méthodes de décomposition efficace. En particulier, dans le cas matriciel, il a conduit à proposer différents algorithmes (PNL-PF et C-PNL-PF) dont l'efficacité a été vérifiée sur des données réelles et simulées et qui donnent par ailleurs de meilleurs résultats surtout dans le cas de sources corrélées. Dans le cas tensoriel, deux algorithmes ont également été proposés (T-PNL-PF et BT-ADMM) qui, de façon similaire au cas matriciel, ont des performances qui surpassent celle de l'état de l'art.

Les perspectives ouvertes par ce travail sont nombreuses. A très court terme, les méthodes développées dans ce travail seront intégrées dans un démonstrateur développé en Python et destiné à l'analyse des données d'un système de production de grande dimension. A titre d'exemple, les algorithmes de factorisation booléennes matricielles ont d'ores et déjà été implantés et permettent le traitement de matrices de très grande taille (5000×5000) avec 16 initialisations NMF différentes en moins de 2 minutes. Une poursuite de ce travail est relative au développement de systèmes de recommandation s'appuyant sur les méthodes développées. Un autre aspect concerne le développement d'algorithmes aptes à traiter les flux de données en lignes. Ce travail ouvrent également des pistes de recherche pour l'analyse de données à valeurs discrètes : peut-on étendre les décompositions booléennes à ce type de données ? Comment définir les opérations matricielles et tensorielles permettant de bien prendre compte ce type de situations.

Bibliographie

[1] Mehmet Koyuturk, Ananth Grama, and Naren Ramakrishnan. Compression, clustering, and pattern discovery in very high-dimensional discrete-attribute data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(4) :447–461, 2005.

[2] Bao-Hong Shen, Shuiwang Ji, and Jieping Ye. Mining discrete patterns via binary matrix factorization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 757–766. ACM, 2009.

[3] Elena Nenova, Dmitry I Ignatov, and Andrey V Konstantinov. An FCA-based boolean matrix factorisation for collaborative filtering. *arXiv preprint arXiv :1310.4366*, 2013.

[4] Shilpa Talwar and Arogyaswami Paulraj. Blind separation of synchronous cochannel digital signals using an antenna array. ii. performance analysis. *IEEE Transactions on Signal Processing*, 45(3) :706–718, 1997.

[5] A Jan van der Veen. Analytical method for blind binary signal separation. In *13th International Conference on Digital Signal Processing Proceedings, DSP 97, 1997*, volume 1, pages 399–402. IEEE, 1997.

[6] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3) :37–52, 1987.

[7] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755) :788–791, 1999.

[8] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[9] Richard A Harshman. Foundations of the parafac procedure : Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[10] Age K Smilde, Yongdong Wang, and Bruce R Kowalski. Theory of medium-rank second-order calibration with restricted-tucker models. *Journal of Chemometrics*, 8(1) :21–36, 1994.

[11] René Henrion. Body diagonalization of core matrices in three-way principal components analysis : Theoretical bounds and simulation. *Journal of chemometrics*, 7(6) :477–494, 1993.

[12] Henk al Kiers. A three–step algorithm for candecomp/parafac analysis of large data sets with multicollinearity. *J. Chemometrics*, 12 :155–171, 1998.

[13] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2) :149–171, 1997.

[14] R Bro-Jorgensen et al. Multiway analysis in the food industry. models, algorithms and applications. 1998.

[15] Age Smilde, Rasmus Bro, and Paul Geladi. *Multi-way analysis : applications in the chemical sciences*. John Wiley & Sons, 2005.

[16] Lieven De Lathauwer, Josphine Castaing, and Jean-Franois Cardoso. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Transactions on Signal Processing*, 55(6) :2965–2973, 2007.

[17] Lieven De Lathauwer and Alexandre de Baynast. Blind deconvolution of DS-CDMA signals by means of decomposition in rank-$(1, l, l)$ terms. *IEEE Transactions on Signal Processing*, 56(4) :1562–1571, 2008.

[18] Lieven De Lathauwer and Bart De Moor. From matrix to tensor : Multilinear algebra and signal processing. In *Institute of Mathematics and its Applications Conference Series*, volume 67, pages 1–16. Citeseer, 1998.

[19] Lieven De Lathauwer and Joos Vandewalle. Dimensionality reduction in higher-order signal processing and rank-(r1, r2, ..., rn) reduction in multilinear algebra. *Linear Algebra and its Applications*, 391 :31–55, 2004.

[20] Derry FitzGerald, Matt Cranitch, and Eugene Coyle. Non-negative tensor factorisation for sound source separation. 2005.

[21] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4) :1253–1278, 2000.

[22] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM journal on Matrix Analysis and Applications*, 21(4) :1324–1342, 2000.

[23] Victor S Grigorascu and Phillip A Regalia. Tensor displacement structures and polyspectral matching. In *Fast reliable algorithms for matrices with structure*, pages 245–276. SIAM, 1999.

[24] Tamara G Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1) :243–255, 2001.

[25] Gregory Beylkin and Martin J Mohlenkamp. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16) :10246–10251, 2002.

[26] Gregory Beylkin and Martin J Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6) :2133–2159, 2005.

[27] Wolfgang Hackbusch and Boris N Khoromskij. Tensor-product approximation to operators and functions in high dimensions. *Journal of Complexity*, 23(4-6) :697–714, 2007.

[28] Evrim Acar, Seyit A Çamtepe, Mukkai S Krishnamoorthy, and Bülent Yener. Modeling and multiway analysis of chatroom tensors. In *International Conference on Intelligence and Security Informatics*, pages 256–268. Springer, 2005.

[29] Evrim Acar, Seyit A Çamtepe, and Bülent Yener. Collective sampling and analysis of high order tensors for chatroom communications. In *International Conference on Intelligence and Security Informatics*, pages 213–224. Springer, 2006.

[30] Brett W Bader, Michael W Berry, and Murray Browne. Discussion tracking in enron email using PARAFAC. In *Survey of Text Mining II*, pages 147–163. Springer, 2008.

[31] Tamara Kolda and Brett Bader. The tophits model for higher-order web link analysis. In *Workshop on link analysis, counterterrorism and security*, volume 7, pages 26–29, 2006.

[32] Tamara G Kolda, Brett W Bader, and Joseph P Kenny. Higher-order web link analysis using multilinear algebra. In *Fifth IEEE International Conference on Data Mining*.

[33] Brett W Bader, Richard A Harshman, and Tamara G Kolda. Temporal analysis of semantic graphs using ASALSAN. In *IEEE International Conference on Data Mining, 2007. ICDM 2007.*, pages 33–42, 2007.

[34] Christian F Beckmann and Stephen M Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1) :294–311, 2005.

[35] Maarten De Vos, A Vergult, Lieven De Lathauwer, Wim De Clercq, Sabine Van Huffel, Patrick Dupont, Andre Palmini, and Wim Van Paesschen. Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone. *NeuroImage*, 37(3) :844–854, 2007.

[36] Andrew I Schein, Lawrence K Saul, and Lyle H Ungar. A generalized linear model for principal component analysis of binary data. In *AISTATS*, volume 3, page 10, 2003.

[37] Jan De Leeuw. Principal component analysis of binary data by iterated singular value decomposition. *Computational statistics & data analysis*, 50(1) :21–39, 2006.

[38] László Kozma, Alexander Ilin, and Tapani Raiko. Binary principal component analysis in the netflix collaborative filtering task. In *IEEE International Workshop on Machine Learning for Signal Processing, 2009. MLSP 2009.*, pages 1–6. IEEE, 2009.

[39] Seokho Lee, Jianhua Z Huang, and Jianhua Hu. Sparse logistic principal components analysis for binary data. *The annals of applied statistics*, 4(3) :1579, 2010.

[40] Zhaoyi Kang and Costas J Spanos. Sequential logistic principal component analysis (slpca) : Dimensional reduction in streaming multivariate binary-state system. In *13th International Conference on Machine Learning and Applications (ICMLA), 2014*, pages 171–177. IEEE, 2014.

[41] Petteri Pajunen. Blind separation of binary sources with less sensors than sources. In *International Conference on Neural Networks, 1997.,* volume 3, pages 1994–1997. IEEE, 1997.

[42] Zhong-Yuan Zhang, Tao Li, Chris Ding, Xian-Wen Ren, and Xiang-Sun Zhang. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, 20(1) :28, 2010.

[43] Pauli Miettinen and Jilles Vreeken. Model order selection for boolean matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 51–59. ACM, 2011.

[44] Radim Belohlavek and Vilem Vychodil. Optimal factorization of three-way binary data. In *IEEE International Conference on Granular Computing (GrC), 2010*, pages 61–66. IEEE, 2010.

[45] Pauli Miettinen. Boolean tensor factorizations. In *IEEE 11th International Conference on Data Mining (ICDM), 2011*, pages 447–456. IEEE, 2011.

[46] Mamadou Diop, Anthony Larue, Sebastian Miron, and David Brie. Factorisation en matrices binaires par modèle de mélange post non-linéaire. In *XXVIe Colloque GRETSI Traitement du Signal & des Images, GRETSI 2017*, 2017.

[47] Mamadou Diop, Anthony Larue, Sebastian Miron, and David Brie. A post-nonlinear mixture model approach to binary matrix factorization. In *25th European Signal Processing Conference, EUSIPCO 2017*, 2017.

[48] Mamadou Diop, Sebastian Miron, Anthony Larue, and David Brie. Boolean decomposition of binary matrices using a post-nonlinear mixture approach. In *IEEE Transactions on Signal Processing*.

[49] Mamadou Diop, Sebastian Miron, Antoine Souloumiac, and David Brie. Boolean CP decomposition of binary tensors : uniqueness and algorithm. In *ICASSP 2019, Brighton, UK, 12-17 May*, 2019.

[50] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3) :455–500, 2009.

[51] Tamara G Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08.*, pages 363–372. IEEE, 2008.

[52] Tamara G Kolda. Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories, 2006.

[53] Pierre Comon. Tensors : a brief introduction. *IEEE Signal Processing Magazine*, 31(3) :44–53, 2014.

[54] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics*, 6(1-4) :164–189, 1927.

[55] Raymond B Cattell. Parallel proportional profiles and other principles for determining the choice of factors by rotation. *Psychometrika*, 9(4) :267–283, 1944.

[56] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of " eckart-young" decomposition. *Psychometrika*, 1970.

[57] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles : Tensorfaces. In *European Conference on Computer Vision*, pages 447–460. Springer, 2002.

[58] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. 2003*, volume 2, pages II–93. IEEE, 2003.

[59] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3) :279–311, 1966.

[60] L Tucker. Implications of factor analysis of three-way matrices for measurement of changes, problems of measuring change (c. harris, ed.), 1963.

[61] Ledyard R Tucker. The extension of factor analysis to three-dimensional matrices. *Contributions to mathematical psychology*, 110119, 1964.

[62] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-(r1, r2, ..., rn) approximation of higher-order tensors.

[63] Nicholas 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(13) :3551–3582, 2017.

[64] Hanna Becker, Laurent Albera, Pierre Comon, Martin Haardt, Gwénaël Birot, Fabrice Wendling, Martine Gavaret, Christian-George Bénar, and Isabelle Merlet. Eeg extended source localization : tensor-based vs. conventional methods. *NeuroImage*, 96 :143–157, 2014.

[65] André LF de Almeida, Gérard Favier, João Cesar M Mota, and Raul L De Lacerda. Estimation of frequency-selective block-fading MIMO channels using PARAFAC modeling and alternating least squares. In *Fortieth Asilomar Conference on Signals, Systems and Computers, 2006. ACSSC'06.*

[66] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation : Independent component analysis and applications.* Academic press, 2010.

[67] J Hastad. Tensor rank is NP-Complete. *Journal of Algorithms*, 11(4) :644–654, 1990.

[68] Petros T Boufounos and Richard G Baraniuk. 1-bit compressive sensing. In *42nd Annual Conference on Information Sciences and Systems, 2008. CISS 2008.*, pages 16–21. IEEE, 2008.

[69] Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4) :2082–2102, 2013.

[70] Shilpa Talwar, Mats Viberg, and Arogyaswami Paulraj. Blind separation of synchronous co-channel digital signals using an antenna array. i. algorithms. *IEEE Transactions on Signal Processing*, 44(5) :1184–1197, 1996.

[71] Yizong Cheng and George M Church. Biclustering of expression data. In *ISMB*, volume 8, pages 93–103, 2000.

[72] Tao Li. A general model for clustering binary data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 188–197. ACM, 2005.

[73] Mehmet Koyutürk and Ananth Grama. Proximus : a framework for analyzing very high dimensional discrete-attributed datasets. In *Proceedings of the ninth ACM*

SIGKDD international conference on Knowledge discovery and data mining, pages 147–156. ACM, 2003.

[74] Edward Meeds, Zoubin Ghahramani, Radford M Neal, and Sam T Roweis. Modeling dyadic data with binary latent factors. In *Advances in neural information processing systems*, pages 977–984, 2006.

[75] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations : applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[76] Thomas Watson. Nonnegative rank vs. binary rank. *Chicago Journal OF Theoretical Computer Science*, 2 :1–13, 2016.

[77] R Duncan Luce. A note on boolean matrix theory. *Proceedings of the American Mathematical Society*, 3(3) :382–388, 1952.

[78] Radim Belohlavek and Vilem Vychodil. Discovery of optimal factors in binary data via a novel method of matrix decomposition. *Journal of Computer and System Sciences*, 76(1) :3–20, 2010.

[79] Daizhan Cheng, Yin Zhao, and Xiangru Xu. Matrix approach to boolean calculus. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011*, pages 6950–6955. IEEE, 2011.

[80] Arie Yeredor. Ica in boolean xor mixtures. In *Independent Component Analysis and Signal Separation*, pages 827–835. Springer, 2007.

[81] Valerie L Watts. Boolean rank of Kronecker products. *Linear Algebra and its Applications*, 336(1-3) :261–264, 2001.

[82] Kim Ki Hang. Boolean matrix theory and applications. *Pure and Applied Mathematics*, 70, 1982.

[83] Pauli Miettinen, Taneli Mielikainen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering,* 20(10) :1348–1362, 2008.

[84] Yanning Shen, Morteza Mardani, and Georgios B Giannakis. Online categorical subspace learning for sketching big data with misses. *IEEE Transactions on Signal Processing*, 65(15) :4004–4018, 2017.

[85] Tony Cai and Wen-Xin Zhou. A max-norm constrained minimization approach to 1-bit matrix completion. *The Journal of Machine Learning Research*, 14(1) :3619–3647, 2013.

[86] Mark A Davenport, Yaniv Plan, Ewout Van Den Berg, and Mary Wootters. 1-bit matrix completion. *Information and Inference : A Journal of the IMA*, 3(3) :189–223, 2014.

[87] Sonia A Bhaskar and Adel Javanmard. 1-bit matrix completion under exact low-rank constraint. In *49th Annual Conference on Information Sciences and Systems (CISS), 2015*, pages 1–6. IEEE, 2015.

[88] Vincent Cottet and Pierre Alquier. 1-bit matrix completion : Pac-bayesian analysis of a variational approximation. *Machine Learning*, 107(3) :579–603, 2018.

[89] Mehmet Koyutürk, Ananth Grama, and Naren Ramakrishnan. Algebraic techniques for analysis of large discrete-valued datasets. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 311–324. Springer, 2002.

[90] Haibing Lu, Jaideep Vaidya, and Vijayalakshmi Atluri. Optimal boolean matrix decomposition : Application to role engineering. In *IEEE 24th International Conference on Data Engineering, 2008. ICDE 2008.*, pages 297–306. IEEE, 2008.

[91] Huy Nguyen and Rong Zheng. Binary independent component analysis with or mixtures. *IEEE Transactions on Signal Processing,*, 59(7) :3168–3181, 2011.

[92] Zhongyuan Zhang, Chris Ding, Tao Li, and Xiangsun Zhang. Binary matrix factorization with applications. In *Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007*, pages 391–400. IEEE, 2007.

[93] Lopamudra Mukherjee, Sathya N Ravi, Vamsi K Ithapu, Tyler Holmes, and Vikas Singh. An NMF perspective on binary hashing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4184–4192, 2015.

[94] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[95] Steffen Rendle, Leandro Balby Marinho, Alexandros Nanopoulos, and Lars Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.

[96] Shuangzhe Liu. Matrix results on the Khatri-rao and tracy-singh products. *Linear Algebra and its Applications*, 289(1-3) :267–277, 1999.

[97] Wing-Kin Ma, Tsung-Han Hsieh, and Chong-Yung Chi. Doa estimation of quasi-stationary signals with less sensors than sources and unknown spatial noise covariance : a Khatri-rao subspace approach. *IEEE Transactions on Signal Processing*, 58(4) :2168–2180, 2010.

[98] Bernhard Ganter, Gerd Stumme, and Rudolf Wille. *Formal concept analysis : foundations and applications*, volume 3626. springer, 2005.

[99] Simon Andrews. In-close, a fast algorithm for computing formal concepts. 2009.

[100] Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE transactions on Signal Processing*, 47(10) :2807–2820, 1999.

[101] Robert M Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the Netflix prize. 2007. *Ref Type : Internet Communication*, 2008.

[102] James Bennett, Stan Lanning, et al. The Netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

[103] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1) :1–122, 2011.

[104] Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19) :5052–5065, 2016.

[105] Max Welling and Markus Weber. Positive tensor factorization. *Pattern Recognition Letters*, 22(12) :1255–1261, 2001.

[106] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8) :e23176, 2011.

[107] Laetitia Gauvin, André Panisson, and Ciro Cattuto. Detecting the community structure and activity patterns of temporal networks : a non-negative tensor factorization approach. *PloS one*, 9(1) :e86028, 2014.

Annexe A

Une annexe

A.1 Calcul des règles de mise à jour de W et H pour l'algorithme PF

Dérivation de $J(\mathbf{W},\mathbf{H})$ (2.3) par rapport à $\mathbf{H}$:

$$\frac{\partial J(\mathbf{W},\mathbf{H})}{\partial \mathbf{H}_{jk}} = -\sum_{j,k}\left(\mathbf{X}_{ij} - \frac{1}{1+e^{-\gamma((\mathbf{W}\cdot\mathbf{H}^T)_{ij}-0.5)}}\right)\cdot\frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} + \lambda(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk})\cdot(2\mathbf{H}_{jk}-1)$$

$$\text{mais } \frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} = -\frac{\frac{\partial}{\mathbf{H}_{jk}}\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2}$$

$$\frac{\partial}{\partial \mathbf{H}_{jk}}\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right) = -\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}$$

$$\text{alors } \frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} = \frac{\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2}$$

$$\frac{\partial J(\mathbf{W},\mathbf{H})}{\partial \mathbf{H}_{jk}} = -\sum_{i,j}\left(\mathbf{X}_{ij} - \frac{1}{1+e^{-\gamma((\mathbf{W}\cdot\mathbf{H}^T)_{ij}-0.5)}}\right)\cdot\frac{\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2} + 2\lambda{\mathbf{H}_{jk}}^3 - 3\lambda\mathbf{H}_{jk}^2 + \lambda\mathbf{H}_{jk}$$

Une simple mise à jour additive pour $\mathbf{H}$ ou $\mathbf{W}$ qui réduit la distance au carré peut être écrite comme :

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} - \alpha_{\mathbf{H}_{jk}}\frac{\partial}{\mathbf{H}_{jk}}J(\mathbf{W},\mathbf{H})$$

$$\text{avec } \alpha_{\mathbf{H}_{jk}} \leftarrow \frac{\mathbf{H}_{jk}}{\gamma\left(\left(\frac{e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)-0.5\right)}}{\left(1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)-0.5\right)}\right)^3}\right)^T\mathbf{W}\right)_{jk} + 2\lambda{\mathbf{H}_{jk}}^3 + \lambda\mathbf{H}_{jk}}$$

Ainsi, nous obtenons comme règles de mise à jour sur $\mathbf{H}$:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\gamma \left(\left(\frac{\mathbf{X} e^{-\gamma \cdot \left((\mathbf{W}\mathbf{H}^T) - 0.5 \right)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)} \right)^2} \right)^T \mathbf{W} \right)_{jk} + 3\lambda \mathbf{H}_{jk}^2}{\gamma \left(\left(\frac{e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)} \right)^3} \right)^T \mathbf{W} \right)_{jk} + 2\lambda {\mathbf{H}_{jk}}^3 + \lambda \mathbf{H}_{jk}}$$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left(\mathbf{X} \frac{\partial}{\partial \mathbf{H}_{jk}} \Phi(\mathbf{W}\mathbf{H}^T) \right)_{jk} + 3\lambda \mathbf{H}_{jk}^2}{\left(\frac{\partial}{\partial \mathbf{H}_{jk}} \Phi(\mathbf{W}\mathbf{H}^T) \right)_{jk} + 2\lambda {\mathbf{H}_{jk}}^3 + \lambda \mathbf{H}_{jk}}$$

En inversant les rôles de $\mathbf{W}$ et $\mathbf{H}$, nous pouvons facilement obtenir les règles de mise à jour sur $\mathbf{W}$:

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{\gamma \left(\left(\frac{\mathbf{X} e^{-\gamma \cdot \left((\mathbf{W}\mathbf{H}^T) - 0.5 \right)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)} \right)^2} \right) \mathbf{H} \right)_{ik} + 3\lambda \mathbf{W}_{ik}^2}{\gamma \left(\left(\frac{e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)} \right)^3} \right) \mathbf{H} \right)_{ik} + 2\lambda {\mathbf{W}_{ik}}^3 + \lambda \mathbf{W}_{ik}}$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{\left(\mathbf{X} \frac{\partial}{\partial \mathbf{W}_{ik}} \Phi(\mathbf{W}\mathbf{H}^T) \right)_{ik} + 3\lambda \mathbf{W}_{ik}^2}{\left(\frac{\partial}{\partial \mathbf{W}_{ik}} \Phi(\mathbf{W}\mathbf{H}^T) \right)_{ik} + 2\lambda {\mathbf{W}_{ik}}^3 + \lambda \mathbf{W}_{ik}}$$

A.2 Calcul des règles de mise à jour de W et H pour l'algorithme TH

Dans le PAS 3 de l'algorithme TH, la première composante du gradient $\mathbf{g}^{(iter)}$ est défini comme :

$$\mathbf{g}^{(iter)}(1) = \frac{\partial F(w,h)}{\partial w} = \frac{\partial \sum_{i,k} F(w,h)}{\partial \tilde{\mathbf{W}}_{ik}} . \frac{\partial \tilde{\mathbf{W}}_{ik}}{\partial w} \text{ avec } \tilde{\mathbf{W}}_{ik} = \phi(\tilde{\mathbf{W}}_{ik} - w)$$

$$\frac{\partial \sum_{i,j} F(w,h)}{\partial \tilde{\mathbf{W}}_{ik}} = \frac{\partial \left(\frac{1}{2} \sum_{i,j} (\mathbf{X}_{ij} - (\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)_{ij})^2 \right)}{\partial \tilde{\mathbf{W}}_{ik}} = -\sum_{i,k} \left((\mathbf{X}\tilde{\mathbf{H}})_{ik} - ((\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)\tilde{\mathbf{H}})_{ik} \right)$$

$$\frac{\partial \tilde{\mathbf{W}}_{ik}}{\partial w} = \frac{\partial \phi(\tilde{\mathbf{W}}_{ik} - w)}{\partial w} = \frac{\partial \frac{1}{1+e^{-\gamma(\tilde{\mathbf{W}}_{ik} - \mathbf{w})}}}{\partial w} = \frac{-\gamma e^{-\gamma(\tilde{\mathbf{W}}_{ik} - w)}}{\left(1 + e^{-\gamma(\tilde{\mathbf{W}}_{ik} - w)}\right)^2}$$

$$\Rightarrow \mathbf{g}^{(iter)}(1) = \sum_{i,k} \left((\mathbf{X}\tilde{\mathbf{H}})_{ik} - \left((\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)\tilde{\mathbf{H}} \right)_{ik} \right) \cdot \frac{\gamma e^{-\gamma(\tilde{\mathbf{W}}_{ik} - w)}}{\left(1 + e^{-\gamma(\tilde{\mathbf{W}}_{ik} - w)}\right)^2}$$

La deuxième composante de $\mathbf{g}^{(iter)}$ peut être définie comme :

$$\mathbf{g}^{(iter)}(2) = \frac{\partial F(w,h)}{\partial h} = \frac{\partial \sum_{j,k} F(w,h)}{\partial \tilde{\mathbf{H}}_{jk}} . \frac{\partial \tilde{\mathbf{H}}_{jk}}{\partial h} \text{avec } \tilde{\mathbf{H}}_{jk} = \phi(\tilde{\mathbf{H}}_{jk} - h)$$

$$\frac{\partial \sum_{j,k} F(w,h)}{\partial \tilde{\mathbf{H}}_{jk}} = \frac{\partial \left(\frac{1}{2} \sum_{i,j} \left(\mathbf{X}_{ij} - (\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)_{ij} \right)^2 \right)}{\partial \tilde{\mathbf{H}}_{jk}} = -\sum_{j,k} \left((\mathbf{X}^T\tilde{\mathbf{W}})_{jk} - \left((\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)^T\tilde{\mathbf{W}} \right)_{jk} \right)$$

$$\frac{\partial \tilde{\mathbf{H}}_{jk}}{\partial h} = \frac{\partial \phi(\tilde{\mathbf{H}}_{jk} - h)}{\partial h} = \frac{\partial \frac{1}{1+e^{-\gamma(\tilde{\mathbf{H}}_{jk} - h)}}}{\partial h} = \frac{-\gamma e^{-\gamma(\tilde{\mathbf{H}}_{jk} - h)}}{\left(1 + e^{-\gamma(\tilde{\mathbf{H}}_{jk} - h)}\right)^2}$$

$$\Rightarrow \mathbf{g}^{(iter)}(2) = \sum_{j,k} \left((\mathbf{X}^T\tilde{\mathbf{W}})_{ik} - \left((\tilde{\mathbf{W}}\tilde{\mathbf{H}}^T)^T\tilde{\mathbf{W}} \right)_{jk} \right) \cdot \frac{\gamma e^{-\gamma(\tilde{\mathbf{H}}_{jk} - h)}}{\left(1 + e^{-\gamma(\tilde{\mathbf{H}}_{jk} - h)}\right)^2}$$

Dans le PAS 4 $\alpha^{(iter)}$ peut être sélectionné pour minimiser $F(w,h)$.

A.3 Calcul des règles de mise à jour de W et H pour l'algorithme PNL-PF

Dérivation de $G(\mathbf{W},\mathbf{H})$ (2.3) par rapport à $\mathbf{H}$:

$$\frac{\partial G(\mathbf{W},\mathbf{H})}{\partial \mathbf{H}_{jk}} = -\sum_{j,k}\left(\mathbf{X}_{ij} - \frac{1}{1+e^{-\gamma((\mathbf{W}\cdot\mathbf{H}^T)_{ij}-0.5)}}\right)\cdot\frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} + \lambda(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk})\cdot(2\mathbf{H}_{jk}-1)$$

$$\text{mais } \frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} = -\frac{\frac{\partial}{\mathbf{H}_{jk}}\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2}$$

$$\frac{\partial}{\partial \mathbf{H}_{jk}}\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right) = -\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}$$

$$\text{alors } \frac{\partial\left(\frac{1}{1+e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}\right)}{\partial \mathbf{H}_{jk}} = \frac{\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2}$$

$$\frac{\partial J(\mathbf{W},\mathbf{H})}{\partial \mathbf{H}_{jk}} = -\sum_{i,j}\left(\mathbf{X}_{ij} - \frac{1}{1+e^{-\gamma((\mathbf{W}\cdot\mathbf{H}^T)_{ij}-0.5)}}\right)\cdot\frac{\gamma\mathbf{W}_{ik}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)_{ij}-0.5\right)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)_{ij}-0.5)}\right)^2} + 2\lambda{\mathbf{H}_{jk}}^3 - 3\lambda\mathbf{H}_{jk}^2 + \lambda\mathbf{H}_{jk}$$

Une simple mise à jour additive pour $\mathbf{H}$ ou $\mathbf{W}$ qui réduit la distance au carré peut être écrite comme :

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} - \alpha_{\mathbf{H}_{jk}}\frac{\partial}{\mathbf{H}_{jk}}J(\mathbf{W},\mathbf{H})$$

$$\text{avec } \alpha_{\mathbf{H}_{jk}} \leftarrow \frac{\mathbf{H}_{jk}}{\gamma\left(\left(\frac{e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)-0.5)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)-0.5)}\right)^3}\right)^T\mathbf{W}\right)_{jk} + 2\lambda{\mathbf{H}_{jk}}^3 + \lambda\mathbf{H}_{jk}}$$

Ainsi, nous obtenons comme règles de mise à jour sur $\mathbf{H}$:

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{\gamma\left(\left(\frac{\mathbf{X}e^{-\gamma\cdot\left((\mathbf{W}\mathbf{H}^T)-0.5\right)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)-0.5)}\right)^2}\right)^T\mathbf{W}\right)_{jk} + 3\lambda\mathbf{H}_{jk}^2}{\gamma\left(\left(\frac{e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)-0.5)}}{\left(1+e^{-\gamma\cdot((\mathbf{W}\mathbf{H}^T)-0.5)}\right)^3}\right)^T\mathbf{W}\right)_{jk} + 2\lambda{\mathbf{H}_{jk}}^3 + \lambda\mathbf{H}_{jk}}$$

$$\Updownarrow$$

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk} \frac{\left(\mathbf{X} \frac{\partial}{\partial \mathbf{H}_{jk}} \Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 3\lambda \mathbf{H}_{jk}^2}{\left(\frac{\partial}{\partial \mathbf{H}_{jk}} \Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk} + 2\lambda {\mathbf{H}_{jk}}^3 + \lambda \mathbf{H}_{jk}}$$

En inversant les rôles de $\mathbf{W}$ et $\mathbf{H}$, nous pouvons facilement obtenir les règles de mise à jour sur $\mathbf{W}$:

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{\gamma \left(\left(\frac{\mathbf{X} e^{-\gamma \cdot \left((\mathbf{W}\mathbf{H}^T) - 0.5 \right)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)}\right)^2} \right) \mathbf{H} \right)_{ik} + 3\lambda \mathbf{W}_{ik}^2}{\gamma \left(\left(\frac{e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)}}{\left(1 + e^{-\gamma \cdot ((\mathbf{W}\mathbf{H}^T) - 0.5)}\right)^3} \right) \mathbf{H} \right)_{ik} + 2\lambda {\mathbf{W}_{ik}}^3 + \lambda \mathbf{W}_{ik}}$$

$$\Updownarrow$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik} \frac{\left(\mathbf{X} \frac{\partial}{\partial \mathbf{W}_{ik}} \Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik} + 3\lambda \mathbf{W}_{ik}^2}{\left(\frac{\partial}{\partial \mathbf{W}_{ik}} \Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik} + 2\lambda {\mathbf{W}_{ik}}^3 + \lambda \mathbf{W}_{ik}}$$

A.4 Calcul des règles de mise à jour sur $\tilde{\mathbf{W}}$, $\mathbf{W}$, $\bar{\mathbf{H}}$, $\mathbf{H}$, $\bar{\mathbf{V}}$, $\mathbf{V}$ pour l'algorithme BT-ADMM

— Dérivation de $J_1(\bar{\mathbf{W}}) = \left\|\mathbf{X}_{(1)} - \mathbf{E}^w - \mathbf{F}^w * (\bar{\mathbf{W}}(\mathbf{V} \odot \mathbf{H})^T)\right\|_F^2 + \frac{\rho}{2}\left\|\mathbf{W} - \bar{\mathbf{W}} + \mathbf{A}\right\|_F^2$ par rapport à $\bar{\mathbf{W}}$

Pour faciliter l'écriture sur les dérivations, on pose $\mathbf{X} = \mathbf{X}_{(1)}$, $\mathbf{H} = \mathbf{V} \odot \mathbf{H}$ et $\mathbf{Y} = \mathbf{X}_{(1)} - \mathbf{E}^w$ alors nous pouvons écrire :

$$\frac{\partial J_1(\bar{\mathbf{W}})}{\partial \bar{\mathbf{W}}_{mn}} = \sum_{i,j}\left(\mathbf{Y}_{ij} - \mathbf{F}^w_{ij}(\bar{\mathbf{W}}\mathbf{H}^T)_{ij}\right)\frac{\partial}{\partial \bar{\mathbf{W}}_{mn}}\left(\mathbf{Y}_{ij} - \mathbf{F}^w_{ij}\sum_k \bar{\mathbf{W}}_{ik}\mathbf{H}_{kj}\right)$$

$$+\rho\sum_{i,k}\left(\mathbf{W}_{ik} - \bar{\mathbf{W}}_{ik} + \mathbf{A}_{ik}\right)\frac{\partial}{\partial \bar{\mathbf{W}}_{mn}}\left(\mathbf{W}_{ik} - \bar{\mathbf{W}}_{ik} + \mathbf{A}_{ik}\right)$$

$$= \sum_j\left(\mathbf{Y}_{mj} - \mathbf{F}^w_{mj}\sum_n \bar{\mathbf{W}}_{mn}\mathbf{H}_{nj}\right)\left(-\mathbf{F}^w_{mj}\mathbf{H}_{jn}\right) - \rho\sum_{m,n}\left(\mathbf{W}_{mn} - \bar{\mathbf{W}}_{mn} + \mathbf{A}_{mn}\right)$$

$$= \sum_j\left(-(\mathbf{Y}_{mj}\mathbf{F}^w_{mj})\mathbf{H}_{nj} + {\mathbf{F}^w_{mj}}^2\sum_n \bar{\mathbf{W}}_{mn}(\mathbf{H}_{nj})^2\right) - \rho\sum_{m,n}\left(\mathbf{W}_{mn} - \bar{\mathbf{W}}_{mn} + \mathbf{A}_{mn}\right)$$

$$= -\left((\mathbf{Y} * \mathbf{F}^w)\,\mathbf{H}^T\right)_{mn} + \left((\mathbf{F}^w * \mathbf{F}^w)\,(\mathbf{H} * \mathbf{H})^T\right)_{mn}\bar{\mathbf{W}}_{mn} + \rho\bar{\mathbf{W}}_{mn} - \rho\left(\mathbf{W}_{mn} + \mathbf{A}_{mn}\right)$$

En remplaçant $\mathbf{Y}$ et $\mathbf{H}$ par leur expression on obtient :

$$\bar{\mathbf{W}} \leftarrow \frac{\left(\left((\mathbf{X}_{(1)} - \mathbf{E}^w) * \mathbf{F}^w\right)(\mathbf{V} \odot \mathbf{H})\right) + \rho\left(\mathbf{W} + \mathbf{A}\right)}{\left((\mathbf{F}^w * \mathbf{F}^w)\left((\mathbf{V} \odot \mathbf{H}) * (\mathbf{V} \odot \mathbf{H})\right)\right) + \rho\mathbf{1}}$$

Si nous faisons les mêmes dérivations sur $\bar{\mathbf{H}}$ et $\bar{\mathbf{V}}$ nous obtenons les règles de mise à jour sur $\bar{\mathbf{H}}$ et $\bar{\mathbf{V}}$.

Table des figures

1.1 Exemple d'un tenseur d'ordre 3, de dimension $I_1 \times I_2 \times I_3$. 16
1.2 Illustration du dépliement du tenseur $\mathcal{X}$ suivant les 3 modes. 17
1.3 Décomposition de TUCKER d'un tenseur d'ordre 3, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ . . . 19
1.4 Décomposition CP d'un tenseur d'ordre 3. 20
1.5 Décomposition de deux sources corrélées avec les différents produits matriciels booléens. 26
1.6 Illustration du bruit binaire de la FBMB 30
1.7 Représentation du tenseur $\mathcal{X}$. 37
1.8 Représentation des différents tenseurs de rang 1 estimés 38

2.1 Fonction sigmoïde $\Phi(x) = \frac{1}{1+e^{-\gamma(x-0.5)}}$. 44
2.2 Cas de deux sources non corrélées ($N = 80$, $M = 100$, $\tau = 1$(ASSO)). . . . 47
2.3 Cas de deux sources corrélées ($N = 100$, $M = 80$, $\tau = 1$ (ASSO)). 48
2.4 Cas de trois sources corrélées ($N = 80$, $M = 100$). 48
2.5 Décomposition de 3 sources non-corrélées. 52
2.6 Unicité de la décomposition de 3 sources corrélées. 52
2.7 Non-unicité de la décomposition de 3 sources corrélées. 52
2.8 Comparaison des algorithmes PNL-PF et C-PNL-PF pour une décomposition non-unique de 3 sources ($N = 80$, $M = 100$, $\beta = 5 \times 10^5$). 53
2.9 Erreur d'estimation sur $\mathbf{W}$, $\mathbf{H}$ et de reconstruction sur $\mathbf{X}$ en fonction du taux de bruit b, pour $p = 25\%$ ($N = 80$, $M = 100$, $\beta = 10$). 56
2.10 Erreur d'estimation sur $\mathbf{W}$, $\mathbf{H}$ et de reconstruction sur $\mathbf{X}$ en fonction du taux de bruit b, pour $p = 60\%$ ($N = 80$, $M = 100$, $\beta = 6 \times 10^5$). 57
2.11 Données Netflix binarisées et approximation de rang 4.w 59
2.12 Erreur d'approximation de $\mathbf{X}$ en fonction du rang k. 59
2.13 Réorganisation de la matrice d'approximation de $\mathbf{X}$ (figure 2.11b) selon les 4 groupes (utilisateurs, films) estimés. 61
2.14 Validation et prédiction. 62

3.1 Décomposition unique de $\mathcal{X}$. 73
3.2 Décomposition non-unique de $\mathcal{X}$. 73
3.3 Taux d'erreur de reconstruction de $\mathcal{X}$ en fonction du paramètre ρ ($N = 12$, $M = 9$, $P = 7$, $K = 3$, $Nb_{iter} = 100$, $Nb_{interne} = 10$) pour $p = 0.6$ 82

3.4 Taux de reconstruction de $\mathcal{X}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.3$ 83
3.5 Taux d'estimation de $\mathbf{W}$ $\mathbf{H}$ et $\mathbf{V}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.3$ 84
3.6 Taux de reconstruction de $\mathcal{X}$ en fonction du taux de bruit rajouté b ($N = 20$, $M = 30$, $P = 10$, $K = 3$, $\rho = 10^9$) pour $p = 0.65$ 84
3.7 Taux d'estimation de $\mathbf{W}$ $\mathbf{H}$ et $\mathbf{V}$ en fonction du taux de bruit rajouté b ($N = 12$, $M = 9$, $P = 7$, $K = 3$, $\rho = 10^9$) pour $p = 0.65$ 85
3.8 Représentation du tenseur de données 86
3.9 Erreur d'approximation de $\mathcal{X}_1$ en fonction du rang booléen K 87
3.10 Représentation des différentes sources ou tenseurs de rang 1 de $\mathcal{X}_1$ 89
3.11 Représentation des différentes sources ou tenseurs de rang 1 de $\mathcal{X}_2$ 91

Liste des tableaux

1.1 Tableau des compétences ou qualités des élèves dans divers cours 36

2.1 La moyenne, le minimum et le maximum du taux d'erreur d'estimation (en %) de $\mathbf{W}$ et $\mathbf{H}$ et le taux d'erreur de reconstruction de $\mathbf{X}$, pour $p = 25\%$($N = 120$, $M = 150$, $\beta = 5 \times 10^6$). 54
2.2 La moyenne, le minimum et le maximum du taux d'erreur d'estimation (en %) de $\mathbf{W}$ et $\mathbf{H}$ et le taux d'erreur de reconstruction de $\mathbf{X}$, pour $p = 60\%$($N = 120$, $M = 150$, $\beta = 5 \times 10^6$). 54
2.3 Le SNR binaire ($SNR_{\mathbb{B}}$) en fonction du taux de bruit rajouté b pour $p = 25\%$. 55
2.4 Le cardinal des groupes d'utilisateurs et de leurs intersections. 60
2.5 Le cardinal des groupes de films et de leurs intersections. 60

3.1 Le nombre d'élèves de chaque classe et de enseignants qui ont participé à l'étude 85
3.2 Tableau récapitulatif des contacts entre les classes pendant les 2 journées . 90

List of Algorithms

1 : Penalty Function (PF) 31
2 : THresholding (TH) 32
3 : Asso 34
4 : BCP-ALS algorithm 39
5 : Post NonLinear Penalty Function (PNL-PF) 45
6 : Constraint Post NonLinear Penalty Function (C-PNL-PF) 51
7 : Tensor-Post NonLinear-Penalty Function (T-PNL-PF) 77
8 : BT-ADMM 81

Résumé

Cette thèse aborde le problème de la décomposition booléenne des tableaux multidimensionnels de données binaires par modèle de mélange post non-linéaire. Dans la première partie, nous introduisons une nouvelle approche pour la *factorisation booléenne en matrices binaires* (FBMB) fondée sur un modèle de mélange post non-linéaire. Contrairement aux autres méthodes de factorisation de matrices binaires existantes, fondées sur le produit matriciel classique, le modèle proposé est équivalent au modèle booléen de factorisation matricielle lorsque les entrées des facteurs sont exactement binaires et donne des résultats plus interprétables dans le cas de sources binaires corrélées, et des rangs d'approximation matricielle plus faibles. Une condition nécessaire et suffisante d'unicité pour la FBMB est également fournie. Deux algorithmes s'appuyant sur une mise à jour multiplicative sont proposés et illustrés dans des simulations numériques ainsi que sur un jeu de données réelles. La généralisation de cette approche au cas de tableaux multidimensionnels (tenseurs) binaires conduit à la factorisation booléenne de tenseurs binaires (FBTB). La démonstration de la condition nécessaire et suffisante d'unicité de la décomposition booléenne de tenseurs binaires repose sur la notion d'indépendance booléenne d'une famille de vecteurs. L'algorithme multiplicatif fondé sur le modèle de mélange post non-linéaire est étendu au cas multidimensionnel. Nous proposons également un nouvel algorithme, plus efficace, s'appuyant sur une stratégie de type AO-ADMM (*Alternating Optimization -ADMM*). Ces algorithmes sont comparés à ceux de l'état de l'art sur des données simulées et sur un jeu de données réelles.

Mots clés : factorisation de matrices binaires, factorisation de tenseurs binaires, produit matriciel booléen, rang booléen

Abstract

This work is dedicated to the study of boolean decompositions of binary multidimensional arrays using a post nonlinear mixture model. In the first part, we introduce a new approach for the boolean factorization of binary matrices (BFBM) based on a post nonlinear mixture model. Unlike the existing binary matrix factorization methods, the proposed method is equivalent to the boolean factorization model when the matrices are strictly binary and give thus more interpretable results in the case of correlated sources and lower rank matrix approximations compared to other state-of-the-art algorithms. A necessary and sufficient condition for the uniqueness of the BFBM is also provided. Two algorithms based on multiplicative update rules are proposed and tested in numerical simulations, as well as on a real dataset. The generalization of this approach to the case of binary multidimensional arrays (tensors) leads to the boolean factorisation of binary tensors (BFBT). The proof of the necessary and sufficient condition for the boolean decomposition of binary tensors is based on a notion of boolean independence of binary vectors. The multiplicative algorithm based on the post nonlinear mixture model is extended to the multidimensional case. We also propose a new algorithm based on an AO-ADMM (Alternating Optimization-ADMM) strategy. These algorithms are compared to state-of-the-art algorithms on simulated and on real data.

Keywords: binary matrix factorization, binary tensor factorization, boolean matrix product, boolean rank