# Factorisation en matrices binaires par modèle de mélange post non-linéaire

Mamadou Diop, Anthony Larue, Sebastian Miron, David Brie

# Factorisation en matrices binaires par modèle de mélange post non-linéaire

Mamadou DIOP[1 2], Anthony LARUE[3], Sebastian MIRON[2], David BRIE[2]

[1]CEA Tech Grand-Est, 5 rue Marconi, Metz Technopôle, France

[2]Université de Lorraine, CRAN, UMR 7039, Vandœuvre, F-54506, France

[3]CEA LIST, 91 191 Gif sur Yvette

mamadou.diop@univ-lorraine.fr, mamadou.diop@cea.fr
anthony.larue@cea.fr
sebastian.miron@univ-lorraine.fr, david.brie@univ-lorraine.fr

**Résumé –** Dans cet article, nous étudions la Factorisation en Matrices Binaires (FMB) qui est un problème similaire à la NMF dans le cas des matrices à valeurs dans l'ensemble $\{0, 1\}$. Une condition nécessaire et suffisante d'identifiabilité pour le modèle FMB est fournie. Deux algorithmes fondés sur le modèle de mélange post non-linéaire proposé, qui garantit la binarité du produit matriciel, sont introduits et comparés à ceux de l'état de l'art.

**Abstract –** In this article, we study the Binary Matrix Factorization (BMF), which is similar to the NMF (Nonnegative Matrix Factorization) problem with a binarity constraint on the matrices. A necessary and sufficient condition for the identification of the BMF model is provided. Two algorithms based on the proposed post nonlinear mixture model, which guarantees the binarity of the matrix product, are presented and compared with the state of art.

## 1 Introduction

Les données binaires occupent aujourd'hui une place importante dans le domaine de l'analyse de données. Il s'agit des données qui ne peuvent prendre que deux valeurs discrètes, le plus souvent 0 ou 1. La factorisation de ces matrices en produit de matrices binaires a un grand nombre d'applications, parmi lesquelles nous pouvons citer : l'extraction d'attributs discrets en grande dimension [3], la classification des réponses des gènes [13], la découverte des motifs dans l'expression des gènes [10], les systèmes de recommandation [9], la reconstruction des caractères aléatoirement échantillonnés [7], etc.

Le problème de la FMB est similaire à celui de la factorisation en matrices non-négatives (NMF en anglais) [4, 5] avec des contraintes de binarité sur les matrices. Ainsi, la FMB consiste à factoriser une matrice $\mathbf{X}$ binaire ($\mathbf{X}_{ij} \in \{0, 1\}$) en deux matrices binaires $\mathbf{W}$ et $\mathbf{H}$ tel que : $\mathbf{X} \approx \mathbf{W} \odot \mathbf{H}^T$ ; $(\odot)$ est appelé produit matriciel binaire et sera défini dans la prochaine section.

Plusieurs algorithmes pour la FMB ont été proposés dans la littérature. Dans [8], Miettinen *et al.* a introduit un algorithme appelé ASSO qui exploite la corrélation entre les colonnes de la matrice de données $\mathbf{X}$. Yeredor s'est penché dans [11] sur l'analyse en composantes indépendantes, dans le cas où les sources sont binaires, avec l'addition qui est remplacée par l'opérateur logique OU-exclusif (XOR). Dans [12], Zhang *et al.* a étendu la NMF standard à la FMB. Ainsi, ils proposent deux algorithmes qui résolvent le problème de la NMF en prenant en compte la contrainte de binarité sur $\mathbf{W}$ et $\mathbf{H}$. Dans ce papier, nous proposons une approche inspirée du [12] mais fondée sur un modèle de mélange post-nonlinéaire, mieux adapté au problème de factorisation en matrices binaires.

Les notations ci-dessous seront utilisées dans ce papier :
- $\mathbf{X}$ : la matrice $\mathbf{X}$
- $\mathbf{x}_j$ : le $j^{\text{ième}}$ vecteur colonne de la matrice $\mathbf{X}$
- $\mathbf{X}_{ij}$ : l'élément $(i, j)$ de la matrice $\mathbf{X}$
- $\mathbf{X}_k$ : la $k^{\text{ième}}$ matrice (source) de rang 1 dans la décomposition de $\mathbf{X}$ $\left(\mathbf{X}_k = \mathbf{w}_k \odot \mathbf{h}_k^T\right)$

Le reste du papier est organisé comme suit : dans la section 2 nous définissons la FMB, présentons les algorithmes proposés dans [12] et donnons une condition théorique nécessaire et suffisante d'identifiabilité du modèle FMB. Dans la section suivante, nous introduisons le modèle de mélange post nonlinéaire avec les deux algorithmes proposés. Enfin, les performances des algorithmes proposés sont évaluées avec des simulations numériques dans la section 5 et des conclusions sont données dans la section 6.

## 2 Factorisation en matrices binaires

### 2.1 Problématique

Le problème direct de la FMB est de reconstrire une matrice binaire $\mathbf{X}$ ($\mathbf{X}_{ij} \in \{0, 1\}$) de taille $N \times M$ à partir de deux

matrices binaires $\mathbf{W}$ et $\mathbf{H}$ de tailles respectives $N \times K$ et $M \times K$ par :

$$\mathbf{X} = \mathbf{W} \odot \mathbf{H}^T. \quad (1)$$

$(\odot)$ est appelé produit matriciel binaire et est défini tel que [6, 1] : $\mathbf{X}_{ij} = \bigvee_{k=1}^{K} (\mathbf{W}_{ik} \wedge \mathbf{H}_{jk})$, où $(\vee)$ et $(\wedge)$ représentent respectivement les opérateurs logiques OU et ET.
Dans la suite du papier, avant d'analyser le problème inverse de la FMB, nous allons étudié la condition d'identifiabilité du modèle FMB (1).

## 2.2 Condition d'identifiabilité du modèle FMB

Dans ce papier, on entend par identifiabilité du modèle FMB, l'unicité de la décomposition (1), *i.e.,* s'il existe un autre couple de matrices $(\bar{\mathbf{W}}, \bar{\mathbf{H}})$ qui vérifie (1), c'est à dire :

$$\mathbf{X} = \mathbf{W} \odot \mathbf{H}^T = \bar{\mathbf{W}} \odot \bar{\mathbf{H}}^T, \quad (2)$$

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 2.1.** (Identifiabilité partielle)
Le modèle (1) est partiellement identifiable si une ou plusieurs colonnes de $\mathbf{W}$ et $\mathbf{H}$ 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 $supp(\mathbf{X}) = \{(i,j), \mathbf{X}_{ij} \neq 0\}$, le support de la matrice $\mathbf{X}$.

**Théorème 2.1** (Identifiabilité partielle)**.** *La $\ell^{ième}$ colonne de $\mathbf{W}$, $\mathbf{w}_\ell$ peut être estimée de façon unique à partir de $\mathbf{X}$ ssi :* $\forall\ n = 1, \ldots, N,$ *avec* $\mathbf{w}_\ell(n) \neq 0, supp\left(\mathbf{e}_n \odot \mathbf{h}_\ell^T\right) \not\subseteq \bigcup_{k \neq \ell}^{K} supp(\mathbf{X}_k)$

avec $\mathbf{e}_n = [0, \cdots, 1, \cdots, 0]^T$, le $n^{\text{ième}}$ vecteur de la base canonique de $\mathbb{R}^N$. Pour des raisons de place, la démonstration de ce théorème n'est pas donnée dans cette communication.

**Corollaire 2.1.1** (Identifiabilité du modèle FMB (1))**.** *Le modèle de la FMB* (1) *est identifiable ssi :* $\forall\ k = 1, \cdots, K,$ $\mathbf{w}_k$ *et* $\mathbf{h}_k$ *peuvent être estimées de façon unique à partir de* $\mathbf{X}$*,* i.e., *le théorème* (2.1) *est vérifié pour toutes les colonnes de* $\mathbf{W}$ *et* $\mathbf{H}$.

Le problème inverse correspondant au problème direct (1), est d'estimer à partir d'une matrice binaire $\mathbf{X}$ de taille $N \times M$, deux matrices binaires $\mathbf{W}$ et $\mathbf{H}$ de tailles respectives $N \times K$ et $M \times K$, avec $K$ le rang de la decomposition, tel que :

$$\{\mathbf{W}, \mathbf{H}\} = \underset{\mathbf{W},\mathbf{H} \in \{0,1\}}{argmin} \|\mathbf{X} - \mathbf{W} \odot \mathbf{H}^T\|_2^2. \quad (3)$$

Le problème (3) est NP complet [8] et par conséquent, pour développer des algorithmes efficaces pour ce problème inverse, des reformulations ont été proposées dans la littérature.

## 2.3 Le modèle de mélange linéaire FMB

Une reformulation de la FMB consiste à remplacer le produit matriciel binaire $(\odot)$ dans (1) par le produit matriciel classique. En d'autres termes, le problème direct (1) devient :

$$\mathbf{X} = \mathbf{W}\mathbf{H}^T. \quad (4)$$

Le problème inverse correspondant à ce problème direct devient :

$$\{\mathbf{W}, \mathbf{H}\} = \underset{\mathbf{W},\mathbf{H} \in \{0,1\}}{argmin} \|\mathbf{X} - \mathbf{W}\mathbf{H}^T\|_2^2. \quad (5)$$

Les deux algorithmes développés dans [12] résolvent ce problème. Le premier est basé sur une fonction de pénalité (PF [1]) et utilise une descente de gradient avec une mise à jour multiplicative similaire à la NMF pour minimiser la fonction de coût ci-dessous :

$$J(\mathbf{W}, \mathbf{H}) = \sum_{i,j} \left(\mathbf{X}_{ij} - (\mathbf{W}\mathbf{H}^T)_{ij}\right)^2 + \frac{1}{2}\lambda \sum_{i,k} \left(\mathbf{W}_{ik}^2 - \mathbf{W}_{ik}\right)^2 + \frac{1}{2}\lambda \sum_{j,k} \left(\mathbf{H}_{jk}^2 - \mathbf{H}_{jk}\right)^2 \quad (6)$$

Le second algorithme proposé dans [12], est basé sur une procédure de seuillage (TH [2]) 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 :

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

avec $$\theta(x) = \begin{cases} 1 & si \quad x > 0 \\ 0 & si \quad x \leq 0 \end{cases} \quad (7)$$

En remplaçant le problème binaire (3) par un problème réel avec des contraintes de binarité sur $\mathbf{W}$ et $\mathbf{H}$, le problème de la FMB est réduit à un problème classique d'optimisation sur l'ensemble des réels (5), plus facile à résoudre que le problème d'optimisation combinatoire (3). Cependant, en général, il n'y a pas de garantie à ce que la solution du problème inverse (5) soit équivalente au (3). C'est pourquoi, nous proposons dans la prochaine section, un modèle de mélange post non-linéaire qui approxime mieux le modèle original (1) et qui permet une meilleure estimation des matrices binaires $\mathbf{W}$ et $\mathbf{H}$.

# 3 L'approche de la FMB par un modèle de mélange post non-linéaire

Nous proposons dans cette section, un modèle de mélange post non-linéaire qui est équivalent au modèle binaire (1) dans le cas où les matrices $\mathbf{W}$ et $\mathbf{H}$ sont exactement binaires. Ainsi, dans ce modèle, nous préservons l'idée de remplacer le produit matriciel binaire $(\odot)$ par le produit matriciel réel avec une contrainte de binarité sur $\mathbf{W}$ et $\mathbf{H}$, en plus de cela, nous introduisons une fonction non-linéaire qui garantit la binarité du produit $\mathbf{W}\mathbf{H}^T$. Pour cela, nous utilisons une fonction $\Phi(x)$ qui tend vers 0 quand $x = 0$ et vers 1 quand $x \geq 1$. Ainsi le problème direct s'écrit :

$$\mathbf{X} = \Phi\left(\mathbf{W}\mathbf{H}^T\right). \quad (8)$$

Dans ce papier, nous choisirons comme fonction $\Phi$, la fonction sigmoïde $\Phi(x) = \frac{1}{1+e^{-\gamma(x-0.5)}}$, avec un paramètre $\gamma$ qui permet d'ajuster la pente de la sigmoïde. Ainsi, le problème inverse correspondant au problème direct (8) s'écrit :

$$\{\mathbf{W}, \mathbf{H}\} = \underset{\mathbf{W},\mathbf{H}^T \in \{0,1\}}{argmin} \|\mathbf{X} - \Phi(\mathbf{W}\mathbf{H}^T)\|_2^2. \quad (9)$$

1. *Penalty Function* en anglais
2. *THreshold* en anglais

Pour résoudre ce problème, nous proposons un algorithme fondé sur une descente de gradient avec une mise à jour multiplicative similaire à l'algorithme PF proposé dans [12]. Cet algorithme minimise la fonction de coût ci-après, correspondant au problème inverse (9) :

$$G(\mathbf{W},\mathbf{H})=\frac{1}{2}\sum_{i,j}\left(\mathbf{X}_{ij}-\Phi(\mathbf{W}\mathbf{H}^T)_{ij}\right)^2+\frac{1}{2}\lambda\sum_{i,k}\left(\mathbf{W}_{ik}^2-\mathbf{W}_{ik}\right)^2 +\frac{1}{2}\lambda\sum_{j,k}\left(\mathbf{H}_{jk}^2-\mathbf{H}_{jk}\right)^2. \quad (10)$$

L'algorithme proposé minimisant la fonction de coût (10) (PNL-PF [3]) est resumé dans l'Algorithme 1.

**Algorithm 1** : PNL-PF

**Entrées :** $\mathbf{X}$, $K$, $Nb_{iter}$, $\varepsilon$
**Sorties :** $\mathbf{W}$, $\mathbf{H}$
*ÉTAPE 1 :* Initialisation
$\mathbf{W} \leftarrow rand(N,K)$, $\mathbf{H} \leftarrow rand(M,K)$
$p = 0$
*ÉTAPE 2 :* Mise à jour de $\mathbf{W}$ et $\mathbf{H}$

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

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

*ÉTAPE 3 :* Normalisation
$\mathbf{W} \leftarrow \mathbf{W}\mathbf{D}_W^{-1/2}\mathbf{D}_H^{1/2}$ $\quad \mathbf{H} \leftarrow \mathbf{H}\mathbf{D}_H^{-1/2}\mathbf{D}_W^{1/2}$
avec $\mathbf{D}_W = diag\left(\max(\mathbf{w}_1),\cdots,\max(\mathbf{w}_K)\right)$
$\mathbf{D}_H = diag\left(\max(\mathbf{h}_1),\cdots,\max(\mathbf{h}_K)\right)$
*ÉTAPE 4 :* Critère d'arrêt
$p \leftarrow p+1$
**SI** $p \geq Nb_{iter}$ ou
$\|\mathbf{X}-\Phi(\mathbf{W}\mathbf{H}^T)\|^2+\sum_{i,k}\left(\mathbf{W}_{ik}^2-\mathbf{W}_{ik}\right)^2+\sum_{j,k}\left(\mathbf{H}_{jk}^2-\mathbf{H}_{jk}\right)^2<\varepsilon$
**ALORS** break
**SINON**
Retourner à l'*ÉTAPE 2*
**Fin SI**

En pratique, pour une convergence plus rapide, $\mathbf{W}$ et $\mathbf{H}$ sont initialisées avec le résultat de l'algorithme NMF développé par *Lee et Seung* dans [4]. Dans l'étape 3, les matrices non-négatives $\mathbf{W}$ et $\mathbf{H}$ sont normalisées afin d'avoir des valeurs de $\mathbf{W}_{ij}$ et $\mathbf{H}_{ij}$ comprises dans l'intervalle [0, 1].
Cependant, le problème inverse (10) est en général, mal-posé. En d'autre termes, le modèle de mélange post non-linéaire n'est pas toujours identifiable. Pour pallier ce problème, une contrainte de plus est rajoutée. Ainsi nous proposons comme solution, de maximiser le support de chaque source de rang 1 : $\mathbf{X}_k = \mathbf{w}_k \odot \mathbf{h}_k^T$, $k = 1, ..., K$.

La fonction coût devient :

$$L(\mathbf{W},\mathbf{H})=\frac{1}{2}\sum_{i,j}\left(\mathbf{X}_{ij}-\Phi(\mathbf{W}\mathbf{H}^T)_{ij}\right)^2+\frac{1}{2}\lambda\sum_{j,k}\left(\mathbf{H}_{jk}^2-\mathbf{H}_{jk}\right)^2 +\frac{1}{2}\lambda\sum_{i,k}\left(\mathbf{W}_{ik}^2-\mathbf{W}_{ik}\right)^2+\lambda_1\frac{1}{\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)}. \quad (11)$$

3. *Post Nonlinear Penalty Function algorithm* en anglais

L'algorithme permettant de minimiser la fonction coût (11) (C-PNL-PF [4]) est identique à l'algorithme PNL-PF excepté l'étape de mise à jour de $\mathbf{W}$ et $\mathbf{H}$ qui est donnée ci-après :

$$\mathbf{H}_{jk} \leftarrow \mathbf{H}_{jk}\frac{\left(\mathbf{X}\frac{\partial}{\partial \mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk}+3\lambda\mathbf{H}_{jk}^2+\lambda_1\frac{\sum_k\left(\sum_i w_{ik}\right)}{\left(\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)\right)^2}}{\left(\frac{\partial}{\partial \mathbf{H}_{jk}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{jk}+2\lambda{\mathbf{H}_{jk}}^3+\lambda\mathbf{H}_{jk}}$$

$$\mathbf{W}_{ik} \leftarrow \mathbf{W}_{ik}\frac{\left(\mathbf{X}\frac{\partial}{\partial \mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik}+3\lambda\mathbf{W}_{ik}^2+\lambda_1\frac{\sum_k\left(\sum_j h_{kj}\right)}{\left(\sum_k\left(\sum_{i,j}\mathbf{W}_{ik}\mathbf{H}_{jk}\right)\right)^2}}{\left(\frac{\partial}{\partial \mathbf{W}_{ik}}\Phi(\mathbf{W}\mathbf{H}^T)\right)_{ik}+2\lambda{\mathbf{W}_{ik}}^3+\lambda\mathbf{W}_{ik}}$$

Dans la prochaine section, les différents algorithmes présentés dans cette communication seront comparés à l'aide des simulations numériques.

# 4 Simulations numériques

Dans cette section, nous comparons les deux algorithms (PF, TH) présentés dans [12] basés sur un modèle de mélange linéaire (4) et les deux algorithmes (PNL-PF, C-PNL-PF) proposés dans ce papier qui sont basés sur un modèle de mélange post non-linéaire (8). Lorsque les sources sont décorrélées (à support disjoints), les différents modèles (1), (4) et (8) sont équivalents, ce qui n'est pas vrai cas dans le cas corrélé. Dans la première expérience (figure 1), nous comparons les différents algorithmes dans le cas de deux sources fortement corrélées mais identifiables, *i.e.,* chaque source vérifie le théorème 2.1. Nous pouvons voir que, l'algorithme TH n'estime pas bien les matrices $\mathbf{W}$, $\mathbf{H}$ et $\mathbf{X}$, l'algorithme PF estime bien $\mathbf{W}$ et $\mathbf{H}$ mais n'arrive pas à reconstruire correctement $\mathbf{X}$ à cause de la non- équivalence entre le modèle linéaire (4) utilisé dans ces algorithmes et le modèle original (4). Les deux algorithmes proposés estiment bien $\mathbf{X}$, $\mathbf{W}$ et $\mathbf{H}$.
La seconde expérience compare les algorithmes PF, PNL-PF et C-PNL-PF dans le cas où nous avons 3 sources corrélées avec une source non-identifiable (la source no. 2). Nous pouvons observer que l'algorithme PF n'arrive pas à estimer correctement les matrices $\mathbf{W}$ et $\mathbf{H}$ alors que l'algorithme PNL-PF estime bien $\mathbf{X}$ et $\mathbf{H}$ mais pas $\mathbf{W}$ (figure 2). Cela signifie que nous avons trouvé une autre solution ($\mathbf{W}$, $\mathbf{H}$) donnant le même minimum global du critère (10). L'algorithme C-PNL-PF quant à lui, arrive à estimer correctement toutes les matrices.

4. *Constaint Post Nonlinear Penalty Function algorithm* en anglais

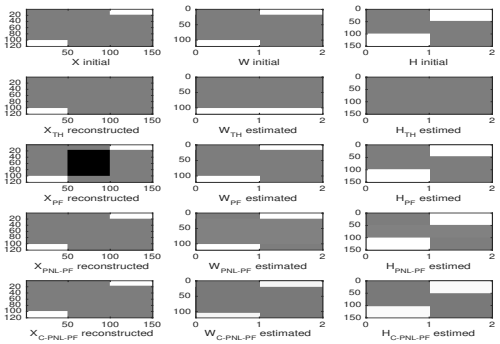



FIGURE 1 – Comparaison des algorithmes TH, PF, PNL-PF et C-PNL-PF dans le cas de 2 sources corrélées (N=120, M=150, $\gamma$ = 150, $\lambda$ = 800, $\lambda_1 = 10^7$)

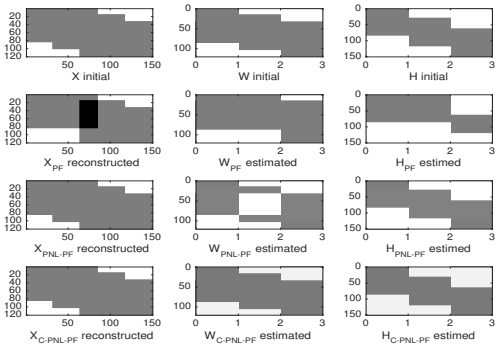
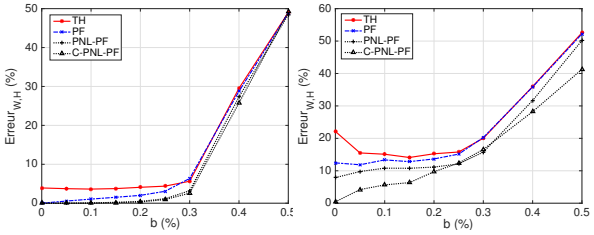



FIGURE 2 – Comparaison des algorithmes PF, PNL-PF et C-PNL-PF dans le cas de 3 sources corrélées, non-identifiables (N=120, M=150, $\gamma$ = 150, $\lambda$ =800, $\lambda_1 = 2 \cdot 10^8$)

Dans la troisième expérience, nous étudions le comportement des algorithmes au bruit binaire. Dans ce papier, nous définissons la donnée bruitée $\mathbf{X}_b$ comme la somme OU-exclusif ($\oplus$) entre la donnée $\mathbf{X}$ et le bruit binaire $\mathbf{B}$, qui est une matrice suivant une loi binomiale avec un paramètre $b$ qui indique le taux d'éléments non nuls dans $\mathbf{B}$. Ainsi dans la figure 3, nous étudions l'erreur moyenne d'estimation de $\mathbf{W}$ et $\mathbf{H}$ (Erreur$_{\mathbf{W},\mathbf{H}}$ [5]) (%) en fonction du bruit rajouté ($b$ (%)) sur une série de 30 réalisa-tions. Les sources sont générées comme dans la figure 2 avec un cas non identifiable (figure 3b) et un cas identifiable (figure 3a). Nous pouvons observer que dans le cas identifiable, les algorithmes PNL-PF et C-PNL-PF donnent une meilleure qualité d'estimation de $\mathbf{W}$ et $\mathbf{H}$ que les algorithmes PF et TH, grâce au modèle proposé. Dans le cas non-identifiable, grâce à la régularisation sur C-PNL-PF, nous estimons correctement $\mathbf{W}$ et $\mathbf{H}$ à bruit nul, ce qui n'est pas le cas pour les autres algorithmes. Même quand le taux de bruit $b$ augmente l'algorithme C-PNL-PF estime mieux $\mathbf{W}$ et $\mathbf{H}$.

5. $\text{Erreur}_{\mathbf{W},\mathbf{H}} = \dfrac{\frac{\sum_{i,k}(\mathbf{W}_{ik}-\hat{\mathbf{W}}_{ik})^2}{N\times K} + \frac{\sum_{j,k}(\mathbf{H}_{jk}-\hat{\mathbf{H}}_{jk})^2}{M\times K}}{2} \times 100$




(a) Cas identifiable (b) Cas non identifiable

FIGURE 3 – L'erreur d'estimation de $\mathbf{W}$ et $\mathbf{H}$ en fonction du taux de bruit rajouté $b$

# 5 Conclusion

Dans ce papier, nous avons étudié l'identifiabilité du modèle FMB et donné une condition nécessaire et suffisante pour l'identifiabilité. Nous avons ensuite proposé une nouvelle approche pour la factorisation des matrice binaires, fondée sur un modèle non-linéaire de mélange. Deux algorithmes basés sur ce modèle ont été proposés et comparés dans des simulations numériques aux algorithmes de l'état de l'art. Les algorithmes proposés donnent de meilleurs résultats notamment dans le cas corrélé, car le modèle non-linéaire utilisé est équivalent au modèle de mélange binaire. Dans le cas bruité, et plus particulièrement en présence des sources non-identifiables, l'algorithme C-PNL-PF a un meilleur comportement par rapport aux autres algorithmes, grâce à la contrainte de support maximal rajoutée.

**Remerciement :** Ce travail fait partie du 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.

# Références

[1] R. Belohlavek and V. 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.

[2] D. Cheng, Y. Zhao, and X. Xu. Matrix approach to boolean calculus. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 6950–6955. IEEE, 2011.

[3] M. Koyuturk, A. Grama, and N. 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.

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

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

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

[7] E. Meeds, Z. Ghahramani, R. M. Neal, and S. T. Roweis. Modeling dyadic data with binary latent factors. In *Advances in neural information processing systems*, pages 977–984, 2006.

[8] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering,*, 20(10) :1348–1362, 2008.

[9] E. Nenova, D. I. Ignatov, and A. V. Konstantinov. An FCA-based boolean matrix factorisation for collaborative filtering. *arXiv preprint arXiv :1310.4366*, 2013.

[10] B.-H. Shen, S. Ji, and J. Ye. Mining discrete patterns via binary matrix factorization. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 757–766. ACM, 2009.

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

[12] Z. Zhang, C. Ding, T. Li, and X. Zhang. Binary matrix factorization with applications. In *Seventh IEEE International Conference on Data Mining*, pages 391–400. IEEE, 2007.

[13] Z.-Y. Zhang, T. Li, C. Ding, X.-W. Ren, and X.-S. Zhang. Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, 20(1) :28–52, 2010.