

# Méthode comprimée et distribuée de factorisation pondérée en matrices non-négatives pour les matrices de grande dimension

Matthieu PUIGT<sup>1</sup> Asmae EL HYANI<sup>1</sup> Kinan ABBAS<sup>2</sup> Gilles ROUSSEL<sup>1</sup> Guillaume CARON<sup>3,4</sup>

<sup>1</sup> Univ. Littoral Côte d'Opale, LISIC – UR 4491, F-62219 Longuenesse, France

<sup>2</sup> CNRS, ENS Lyon, Laboratoire de Physique – UMR 5672, F-69364 Lyon, France

<sup>3</sup> Univ. Picardie Jules Verne, MIS – UR 4290, F-80039 Amiens, France

<sup>4</sup> CNRS-AIST, JRL – IRL 3218, 305-0032 Tsukuba, Japon

**Résumé** – La factorisation pondérée en matrices non-négatives (WNMF) permet de factoriser une matrice de données tout en tenant compte de la confiance en chaque point de donnée. Lorsque les poids ne sont pas binaires, la plupart des méthodes proposées pour la WNMF sont peu adaptées au traitement de matrices de données volumineuses. Dans cet article, nous nous attaquons à ce problème en proposant une approche comprimée (par projections aléatoires) et distribuée de WNMF. Les projections aléatoires font partie des principales techniques pour accélérer le traitement de données de grande dimension. Elles ont été combinées à la WNMF dans un formalisme EM qui nécessitent certains calculs qui pourraient ne pas tenir dans la mémoire. Dans cet article, nous proposons une technique qui ne nécessite pas de tels calculs et qui permet de mettre à jour les facteurs matriciels de manière distribuée.

**Abstract** – Weighted Non-Negative Matrix Factorization (WNMF) allows factoring a data matrix while taking into account the confidence in each data point. When the weights are not binary, most WNMF methods are poorly suited to processing large data matrices. In this paper, we tackle this problem by proposing a compressed (by random projections) and distributed WNMF approach. Random projections are among the main techniques to process big data. They have been combined with WNMF in an EM formalism that requires some computations which might not fit in memory. In this paper, we propose a technique which does not require such computations and which can distribute the update rules of the factor matrices.

## 1 Introduction

La factorisation de matrices non négatives (NMF pour *Non-negative Matrix Factorization* en anglais) est une technique de réduction de dimensionnalité, appliquée avec succès à de nombreux domaines [6]. Lorsqu'une mesure de confiance<sup>1</sup>  $w_{ij} \in [0, 1]$  est liée à un point de données  $x_{ij}$ , la NMF peut être étendue à un cadre pondéré (WNMF pour *Weighted NMF* en anglais). La WNMF consiste à estimer deux matrices non-négatives  $G$  et  $F$  de dimensions respectives  $n \times p$  et  $p \times m$  (avec  $p \leq \min(n, m)$ ), à partir de<sup>2</sup>

$$W \odot X \simeq W \odot (G \cdot F), \quad (1)$$

où  $X$  et  $W$  sont des matrices non-négatives de dimension  $n \times m$  et  $\odot$  désigne le produit de Hadamard. La WNMF a été appliquée avec succès, par exemple, au filtrage collaboratif [19], au partitionnement de sources [2, 14], à l'étalonnage *in situ* de capteurs [4] ou au dématricage et au démélange multispectral conjoints [1]. Bien que la plupart des méthodes de WNMF sont peu adaptées au traitement de matrices de données volumineuses, des techniques rapides ont été proposées pour les pondérations binaires (c'est-à-dire pour gérer les données manquantes dans  $X$ ) [12] et à valeurs réelles [3, 14, 18]. En particulier, les travaux dans [14, 18] font appel à des stratégies d'accélération grâce aux projections aléatoires [5, 8].

<sup>1</sup>Notons que si la matrice  $W$  de poids n'est pas comprise entre 0 et 1 [2, 14], il est toujours possible de remettre à l'échelle cette matrice en la divisant par sa plus grande valeur.

<sup>2</sup>L'éq. (1) se réduit à la NMF lorsque  $W$  est une matrice de uns.

Conçues pour traiter les problèmes de *big data*, ces dernières constituent un outil de compression (généralement) agnostique aux données qui préserve en espérance les informations [8].

La combinaison des projections aléatoires avec la NMF (ou NMF comprimée, par analogie avec l'apprentissage comprimé [7]) a été proposée dans plusieurs travaux [15–17, 20]. À l'exception de [20] – qui approxime la matrice de données par une SVD randomisée moins coûteuse à stocker et qui propose des mises à jour *ad hoc* – les auteurs des articles ci-dessus considèrent la compression bilatérale, c'est-à-dire qu'ils réduisent la taille de  $X$  respectivement à gauche et à droite – avec des matrices  $L$  et  $R$  – pour mettre à jour  $F$  et  $G$ . Par exemple, en définissant  $L$  une matrice de compression, on peut définir les matrices compressées  $X_L \triangleq L \cdot X$  et  $G_L \triangleq L \cdot G$ . La mise à jour de  $F$  à partir de  $X_L$  et  $G_L$  nécessite alors beaucoup moins d'opérations qu'à partir de  $X$  et  $G$ . L'application de projections aléatoires à la WNMF a été envisagée par plusieurs auteurs [14, 18]. La stratégie proposée dans [14] est assez similaire à celle de [20] et ne peut gérer des valeurs nulles dans  $W$ , tandis que les auteurs de [18] ont introduit des projections aléatoires bilatérales dans un cadre de Maximisation de l'Espérance (EM). Dans ce dernier cas, même si la technique de WNMF comprimée s'avère accélérer significativement son homologue non-comprimé, la complétion de  $X$  (et le calcul des opérateurs  $L$  et  $R$ ) dans l'étape E constituent un goulot d'étranglement compensé par les gains en temps de calculs dans l'étape M.

Dans cet article, nous proposons donc une alternative aux stratégies précédentes. Plus précisément, en nous appuyant sur

les travaux dans [11], nous proposons une méthode WNMF comprimée et distribuée. Nous rappelons en section 2 les stratégies proposées dans la littérature pour résoudre la WNMF. Nous introduisons notre approche proposée dans la section 3 et nous illustrons ses performances dans la section 4. Enfin, la conclusion et des perspectives sont fournies en section 5.

## 2 Rappel des stratégies de WNMF

Nous rappelons maintenant les principales stratégies proposées pour résoudre l'équation (1) lorsque  $W$  est à valeurs réelles. Plus précisément, en considérant la norme de Frobenius  $\|\cdot\|_{\mathcal{F}}$  comme mesure de divergence, la WNMF cherche à estimer deux matrices  $G$  et  $F$  qui satisfont

$$\min_{G, F \geq 0} \frac{1}{2} \|W \odot X - W \odot (G \cdot F)\|_{\mathcal{F}}^2. \quad (2)$$

Comme pour la NMF, la stratégie classique de résolution de WNMF consiste à mettre à jour de manière alternée un facteur matriciel en fixant l'autre. À cette fin, trois cadres principaux ont été proposés.

Le premier consiste à dériver les règles de mise à jour de WNMF en tenant compte de la matrice de pondération [9, Ch. 6]. Cependant, il a été démontré que cette stratégie ne peut pas bénéficier des solveurs de gradient rapides [3], en raison du produit de Hadamard dans les expressions de gradient.

Le cadre EM [19] propose deux étapes exécutées en alternance. L'étape E consiste à compléter la matrice de données  $X$ , notée  $X^{\text{comp}}$ , qui se présente comme suit :

$$X^{\text{comp}} = W \odot X + (\mathbb{1}_{n,m} - W) \odot (G \cdot F), \quad (3)$$

où  $\mathbb{1}_{n,m}$  est la matrice  $n \times m$  de uns. Ensuite, à l'étape M,  $G$  et  $F$  sont mises à jour à partir de  $X^{\text{comp}}$  à l'aide d'un solveur NMF, par exemple un solveur rapide [3] éventuellement combiné à des projections aléatoires [18]. Il convient toutefois de noter que le calcul de  $X^{\text{comp}}$  dans l'équation (3) peut être compliqué si la matrice de données ne tient pas en mémoire.

Enfin, les auteurs de [11] ont proposé une stratégie distribuée où les matrices facteurs sont mises à jour ligne par ligne ou colonne par colonne. En effet, si l'on considère la  $i$ -ième ligne de  $X$  et  $G$ , notées respectivement  $\mathbf{x}_i$  et  $\mathbf{g}_i$ , la relation (1) limitée à cette ligne peut être réécrite comme suit :

$$D_i \cdot \mathbf{x}_i \approx D_i \cdot F^T \cdot \mathbf{g}_i, \quad (4)$$

où  $D_i$  est la matrice diagonale contenant la  $i$ -ième ligne de  $\sqrt{W}$ . De même, en considérant la  $j$ -ième colonne de  $X$  et  $F$  (notées respectivement  $\mathbf{x}^j$  et  $\mathbf{f}^j$ ) et la matrice diagonale  $D^j$  contenant la  $j$ -ième colonne de  $\sqrt{W}$ , on obtient

$$D^j \cdot \mathbf{x}^j \approx D^j \cdot G \cdot \mathbf{f}^j. \quad (5)$$

La mise à jour de ces vecteurs lignes et colonnes peut ensuite être obtenue en appliquant une technique de moindres carrés non-négatifs [11]. De plus, ces problèmes étant indépendants, ces calculs peuvent être distribués / parallélisés.

## 3 Méthode proposée

Nous introduisons maintenant notre approche comprimée et distribuée lorsque  $X$  est de faible rang et de grande dimension, c.-à-d. (en reprenant les notations dans la section 1)

$p \ll \min(n, m)$ . Rappelons d'abord que lorsque les techniques de projection aléatoire sont combinées à la WNMF dans [18], le produit de Hadamard dans l'éq. (1) rend difficile la compression bilatérale de la NMF [15]. C'est la raison pour laquelle des matrices de compression sont calculées à chaque étape E dans [18]. Cette stratégie comprimée génère donc une perte de temps de calcul dans l'étape E (même si la charge de calcul peut être réduite en optimisant les techniques de compression [18]), compensée lorsque le nombre de mises à jour dans l'étape M est suffisamment important. Toutefois, développer une approche distribuée à partir de cette technique n'est pas intéressant, notamment parce que le calcul de  $X^{\text{comp}}$  dans l'étape E pourrait ne pas tenir dans la mémoire si les dimensions de  $X$  sont très élevées.

Aussi, dans cet article, nous cherchons plutôt à étendre le troisième concept rappelé dans la section 2. Lorsqu'on cherche à mettre à jour la colonne  $\mathbf{g}_i$  dans l'éq. (4), on considère un problème de moindres carrés non-négatifs sur-déterminé où la matrice  $D_i \cdot F^T$  est grande et mince (puisque  $p \ll m$ ). Il peut alors être intéressant d'appliquer un opérateur  $L_i$  de taille  $(p + \nu) \times m$  (avec  $\nu$  un entier tel que  $p + \nu \ll m$ ) à sa gauche, pour réduire la taille du problème (6). En définissant  $\mathbf{y}_i \triangleq L_i \cdot D_i \cdot \mathbf{x}_i$  et  $Z_i \triangleq L_i \cdot D_i \cdot F^T$ , l'éq. (4) s'écrit

$$\mathbf{y}_i \approx Z_i \cdot \mathbf{g}_i, \quad (6)$$

où  $\mathbf{y}_i$  est un vecteur colonne de  $p + \nu$  éléments et  $Z_i$  une matrice de taille  $(p + \nu) \times p$ . La matrice  $Z_i$  constitue une *esquisse* de taille réduite de  $D_i \cdot F^T$ . Ainsi, la mise à jour de  $\mathbf{g}_i$  à partir de l'éq. (6) prend en compte des matrices et vecteurs de taille plus réduite que sa variante non-comprimée. Reste à vérifier que  $L_i$  permet de conserver l'information contenue dans  $D_i \cdot F^T$ . Le cadre mathématique derrière les projections aléatoires [5, 8] permet de s'en assurer : le lemme de Johnson-Lindenstrauss indique en effet qu'un ensemble de points situé dans un espace euclidien de grande dimension peut être projeté dans un espace de plus faible dimension, tout en conservant approximativement les distances entre deux points arbitraires de l'ensemble. Le plongement linéaire pour réaliser cette projection peut être purement aléatoire [8].

Cependant, en particulier pour la NMF [15], un opérateur obtenu par projection aléatoire *structurée* – aussi connue sous le nom de puissance itérée randomisée<sup>3</sup> (RPI pour *Randomized Power Iterations* en anglais) [8] – offre une meilleure précision de calcul qu'un opérateur issu d'une projection aléatoire agnostique aux données. Une RPI calcule une base orthonormale randomisée de la matrice qu'elle cherche à compresser. Or, il se trouve que par construction de  $D_i$ , l'espace ligne de chaque matrice  $D_i \cdot F^T$  est inclus dans l'espace ligne de  $F^T$ . Dit autrement, au lieu de construire une matrice de compression  $L_i$  pour chaque matrice  $D_i \cdot F^T$ , on peut construire une unique matrice de compression – notée  $L_F$  – à partir de  $F^T$  et l'appliquer pour chacune des relations (6) à la place de  $L_i$ . La matrice orthonormale  $L_F$  s'obtient en calculant la décomposition QR<sup>4</sup>

$$L_F \triangleq \text{QR}((F^T \cdot F)^q \cdot F^T \cdot \Omega)^T, \quad (7)$$

<sup>3</sup>Il est à noter qu'il existe une variante plus robuste aux erreurs d'arrondis de calculs que la RPI, nommée RSI pour *Randomized Subspace Iterations*. En pratique, nous n'avons pas constaté de différence entre les deux approches dans nos expériences et nous ne présentons que les RPI par la suite.

<sup>4</sup>La décomposition QR permet d'approcher une matrice par le produit d'une matrice orthogonale – notée  $L_F$  dans l'éq. (7) – et une matrice triangulaire supérieure. Cette dernière n'est pas utilisée par la suite.

où  $q$  est un entier positif fixé (typiquement  $q = 4$  [15]) et  $\Omega$  une matrice aléatoire Gaussienne de taille  $p \times (p + \nu)$ . Il est à noter que  $L_F$  étant de signe quelconque, la non-négativité de  $y_i$  et  $Z_i$  dans l'éq. (6) n'est pas assurée, faisant de cette dernière un problème de régression semi-non-négatif.

De même, lorsqu'on cherche à mettre à jour les colonnes de  $F$  selon l'éq. (5), on peut définir l'opérateur de compression

$$L_G \triangleq \text{QR} \left( (G \cdot G^T)^q \cdot G \cdot \Omega \right)^T, \quad (8)$$

et déduire la version comprimée de (5) qui se lit

$$y^j \approx Z^j \cdot f^j, \quad (9)$$

où  $y^j \triangleq L_G \cdot D^j \cdot x^j$  et  $Z^j \triangleq L_G \cdot D^j \cdot G$ . Comme pour  $G$ , la mise à jour de  $f^j$  à partir de l'éq. (9) reste possible puisque  $L_G$  permet de conserver l'information contenue dans  $D^j \cdot G$ , alors que les dimensions en jeu dans l'éq. (9) sont plus petites que celles dans l'éq. (5). Le pseudo-code de la méthode proposée est fourni dans l'algorithme 1. Le critère d'arrêt de la boucle de la méthode peut être, par exemple, un nombre d'itérations, un temps de calcul autorisé ou une valeur cible de la norme (2).

---

#### Algorithme 1 Approche comprimée et distribuée de WNMF

---

Initialiser  $G \geq 0$  et  $F \geq 0$ .

##### Répéter

Calculer  $L_F$  selon l'éq. (7)

Mettre à jour  $G$  ligne par ligne selon l'éq. (6)

Calculer  $L_G$  selon l'éq. (8)

Mettre à jour  $F$  colonne par colonne selon l'éq. (9)

##### Jusqu'à Convergence

---

Il est à noter que cette méthode possède certaines spécificités. Tout d'abord, contrairement aux méthodes comprimées de NMF [15, 16] et WNMF [14], les matrices de compression évoluent au cours du temps. Cette propriété est partagée avec [18], mis à part que (i) on ne calcule pas ici une matrice complète  $X^{\text{comp}}$  pour ensuite déduire les opérateurs de compression et (ii) les opérateurs de compression – obtenus à partir de  $G$  et  $F$  – sont moins coûteux à calculer que ceux à partir de  $X^{\text{comp}}$  dans [18]. Néanmoins, le coût calculatoire engendré pour obtenir les matrices de compression pourrait ne pas être compensé par le gain en calcul des mises à jour. De plus, comme pour [11], les mises à jour ligne par ligne ou colonne par colonne peuvent être distribuées sur plusieurs cœurs. Enfin, comme pour les autres méthodes de (W)NMF approchée [6], l'unicité de la solution n'est pas garantie.

## 4 Illustration expérimentale

Nous proposons dans cette section une illustration des performances de la stratégie proposée. Pour cela, nous proposons une expérience qui permet d'analyser la vitesse et la précision des algorithmes testés pour la complétion de matrices non-négatives de faible rang et pour la séparation de sources à partir de données partiellement observées. Alors que la première application se focalise sur l'estimation des valeurs manquantes de  $X$ , la seconde étudie la qualité d'estimation de  $G$  et  $F$ . Ce dernier cas est plus difficile car une bonne approximation de  $X$  n'implique pas forcément une bonne estimation des facteurs matriciels.

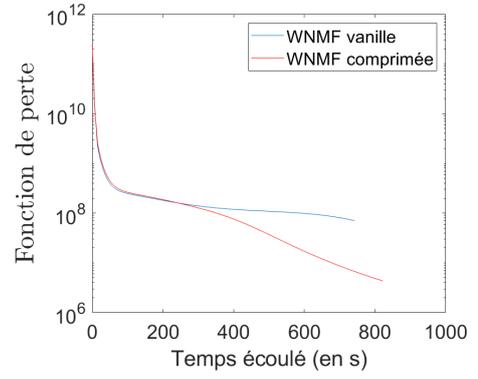


FIGURE 1 – Fonction de perte de la WNMF en fonction du temps

Pour cela, nous considérons des facteurs non-négatifs aléatoires  $G_{\text{theo}}$  et  $F_{\text{theo}}$ , avec  $n = m = 10000$  et  $p = 5$ . Leur produit fournit la matrice observée complète  $X_{\text{theo}}$  que nous échantillonons aléatoirement avec un taux de 50%. Nous comparons les performances de l'approche distribuée de WNMF avec et sans compression. La compression se fait par RPI avec  $q = 4$  et  $\nu = 10$ . La mise à jour des matrices se fait par descente de gradient de Nesterov avec 100 itérations [13].

Les deux méthodes sont initialisées avec les mêmes facteurs matriciels  $F$  et  $G$  tirés aléatoirement et sont lancées durant 100 itérations sur un PC portable avec un processeur Intel Core i7-1365U (10 cœurs), 32 GB de mémoire RAM et Matlab R2021a. Nous considérons plusieurs critères de performance, c.-à-d. (i) l'évolution de la fonction de perte  $\|W \odot X - W \odot (G \cdot F)\|_{\mathcal{F}}^2$  au cours des itérations; (ii) le temps écoulé (en s), qui prend en compte les temps nécessaires aux lectures et écritures des calculs, au cours des itérations; (iii) l'erreur relative de reconstruction (RRE) finale, portant sur la qualité d'estimation de  $X_{\text{theo}}$  :

$$\text{RRE} \triangleq \|X_{\text{theo}} - G \cdot F\|_{\mathcal{F}}^2 / \|X_{\text{theo}}\|_{\mathcal{F}}^2; \quad (10)$$

(iv) le rapport signal à interférence (SIR)

$$\text{SIR} = \frac{1}{p} \sum_{j=1}^p 10 \log_{10} \left( \left\| \hat{f}^j{}^{\text{coll}} \right\|^2 / \left\| \hat{f}^j{}^{\text{orth}} \right\|^2 \right), \quad (11)$$

où  $\hat{f}^j{}^{\text{coll}}$  et  $\hat{f}^j{}^{\text{orth}}$  sont respectivement les parties colinéaire et orthogonale de  $\hat{f}^j$  (un vecteur source estimé) au vecteur théorique associé; et (v) le rapport erreur à mélange (MER) qui correspond au critère de SIR, mais calculé sur les colonnes de  $G$ .

La figure 1 présente l'évolution de la fonction de perte pour la WNMF vanille et comprimée dans l'illustration considérée. Étonnamment, l'approche comprimée est légèrement plus lente que sa variante vanille (en atteste le temps écoulé légèrement plus long au bout de 100 itérations) mais la fonction de perte est beaucoup plus faible, soulignant que l'approche comprimée fournit de meilleures estimations que sa variante vanille. Ce dernier résultat est visible sur le tableau 1 qui reprend l'ensemble des valeurs des critères de performance décrits ci-dessus. Néanmoins, nos prochains travaux chercheront à caractériser plus finement les performances de l'approche proposée.

Méthode	RRE	SIR (dB)	MER (dB)
vanille	$7.8 \cdot 10^{-5}$	16.7	16.8
comprimée	$4.9 \cdot 10^{-6}$	25.1	25.1

TABLEAU 1 – Performance atteinte par les méthodes testées sur l’illustration au bout de 100 itérations.

## 5 Conclusion

Dans cet article, nous avons proposée une approche comprimée et distribuée de WNMF qui évite le goulot d’étranglement de la stratégie EM précédemment proposée. Les futures directions de ces travaux sont nombreuses. D’une part, nous souhaitons analyser plus finement les performances de l’approche proposée. Il pourrait aussi être possible d’aller plus loin dans la distribution des calculs que ce qui a été proposé ici, par exemple en s’inspirant des travaux proposés dans [10] pour la NMF non-pondérée. Une application pourrait concerner l’extension de [1] à des données de vidéo. Par ailleurs, il serait intéressant d’en proposer des extensions informées et robustes, par exemple pour des applications où ces deux propriétés sont hautement désirables [2, 4].

## Références

- [1] K. ABBAS, M. PUIGT, G. DELMAIRE et G. ROUSSEL : Locally-rank-one-based joint unmixing and demosaicing methods for snapshot spectral images. part I : a matrix-completion framework. *IEEE Trans. Comput. Imaging*, 10:848–862, 2024.
- [2] G. DELMAIRE, M. OMIDVAR, M. PUIGT, F. LEDOUX, A. LIMEM, G. ROUSSEL et D. COURCOT : Informed weighted non-negative matrix factorization using  $\alpha\beta$ -divergence applied to source apportionment. *Entropy*, 21(3):253, 2019.
- [3] C. DORFFER, M. PUIGT, G. DELMAIRE et G. ROUSSEL : Fast nonnegative matrix factorization and completion using Nesterov iterations. *In Proc. LVA/ICA 2017*, volume LNCS 10179, pages 26–35, Feb. 2017.
- [4] C. DORFFER, M. PUIGT, G. DELMAIRE et G. ROUSSEL : Informed nonnegative matrix factorization methods for mobile sensor network calibration. *IEEE Trans. Signal Inf. Process. Netw.*, 4(4):667–682, 2018.
- [5] P. DRINEAS et M. W. MAHONEY : RandNLA : randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.
- [6] N. GILLIS : *Nonnegative matrix factorization*. SIAM, 2020.
- [7] R. GRIBONVAL, A. CHATALIC, N. KERIVEN, V. SCHELLEKENS, L. JACQUES et P. SCHNITER : Sketching data sets for large-scale learning : Keeping only what you need. *IEEE Signal Process. Mag.*, 38(5):12–36, 2021.
- [8] N. HALKO, P.-G. MARTINSSON et J. A. TROPP : Finding structure with randomness : Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- [9] N.-D. HO : *Non negative matrix factorization algorithms and applications*. Phd thesis, Université Catholique de Louvain, 2008.
- [10] R. KANNAN, G. BALLARD et H. PARK : MPI-FAUN : An MPI-based framework for alternating-updating non-negative matrix factorization. *IEEE Trans. Knowl. Data Eng.*, 30(3):544–558, 2017.
- [11] Y.-D. KIM et S. CHOI : Weighted nonnegative matrix factorization. *In Proc. ICASSP’09*, pages 1541–1544, 2009.
- [12] Y. KOREN, R. BELL et C. VOLINSKY : Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug 2009.
- [13] Y. NESTEROV : A method of solving a convex programming problem with convergence rate  $O(1/k^2)$ . *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- [14] B. C. SAPPER, S. YOUN, D. K. HENZE, M. CANAGARATNA, H. STARK et J. L. JIMENEZ : Positive matrix factorization of large real-time atmospheric mass spectrometry datasets using error-weighted randomized hierarchical alternating least squares. *Geoscientific Model Development*, 18(10):2891–2919, 2025.
- [15] M. TEPPER et G. SAPIRO : Compressed nonnegative matrix factorization is fast and accurate. *IEEE Trans. Signal Process.*, 64(9):2269–2283, May 2016.
- [16] F. WANG et P. LI : Efficient nonnegative matrix factorization with random projections. *In Proc. SIAM ICDM’10*, pages 281–292, 2010.
- [17] F. YAHAYA, M. PUIGT, G. DELMAIRE et G. ROUSSEL : Random projection streams for (weighted) nonnegative matrix factorization. *In Proc. IEEE ICASSP’21*, pages 3280–3284, 2021.
- [18] F. YAHAYA, M. PUIGT, G. DELMAIRE et G. ROUSSEL : A framework for compressed weighted nonnegative matrix factorization. *IEEE Trans. Signal Process.*, 72:4798–4811, 2024.
- [19] S. ZHANG, W. WANG, J. FORD et F. MAKEDON : Learning from incomplete ratings using non-negative matrix factorization. *In Proc. SIAM ICDM’06*, pages 549–553. SIAM, 2006.
- [20] G. ZHOU, A. CICHOCKI et S. XIE : Fast nonnegative matrix/tensor factorization based on low-rank approximation. *IEEE Trans. Signal Process.*, 60(6):2928–2940, June 2012.

## Remerciements

Ces travaux sont financés en partie par l’Alliance A2U dans le cadre de son atelier IA–Optimisation. Les expériences présentées dans cet article ont été réalisées sur la plate-forme de calcul scientifique Calculco, gérée par la DSI de l’ULCO.