Accélération de la factorisation pondérée en matrices non-négatives par projections aléatoires

Farouk Yahaya, Matthieu Puigt, Gilles Delmaire, Gilles Roussel

Accélération de la factorisation pondérée en matrices non-négatives par projections aléatoires

Farouk YAHAYA, Matthieu PUIGT, Gilles DELMAIRE, Gilles ROUSSEL

Univ. Littoral Côte d'Opale, LISIC – EA 4491, 62221 Calais France *
{farouk.yahaya,matthieu.puigt,gilles.delmaire,gilles.roussel}@univ-littoral.fr

Résumé – Les projections aléatoires font partie des principales techniques pour accélérer le traitement de données de grande dimension. Elles ont notamment été appliquées à la Factorisation en Matrices Non-négatives (FMN). Cependant, elles ne peuvent pas être appliquées lorsqu'une pondération est associée aux données. Dans cet article, nous proposons une technique pour répondre à cette problématique. Celle-ci combine une stratégie de maximisation de l'espérance et des projections aléatoires. Nous montrons expérimentalement qu'elle permet d'accélérer significativement plusieurs techniques de la littérature, sous des contraintes douces.

Abstract – Random projections belong to the major techniques to process big data and have been successfully applied to, e.g., Nonnegative Matrix Factorization (NMF). However, they cannot be applied when weights are associated with the data. In this paper, we thus aim to solve this issue. We propose a novel framework which combines an expectation-maximization strategy with random projections. We experimentally show the proposed framework to significantly speed-up state-of-the-art NMF methods under mild conditions.

1 Introduction

Les méthodes de Factorisation en Matrices Non-négatives (FMN) sont parmi les techniques de traitement de données les plus populaires, depuis les travaux pionniers de [1,2]. En effet, les facteurs matriciels obtenus par FMN sont généralement plus facilement interprétables que ceux obtenus par des approches sans contrainte de signe [2] et la FMN a été appliquée avec succès à de nombreuses problématiques en traitement du signal et des images et en apprentissage statistique [3]. Dans nombre de problèmes tels le traitement d'images [4], la séparation de sources chimiques [5], le filtrage collaboratif [6] ou la complétion de matrices [7], une mesure de confiance w_{ij} est associée à chaque entrée x_{ij} de la matrice X à factoriser, supposée de taille $n \times m$ et de rang p connu. La Factorisation Pondérée en Matrices Non-négatives (FPMN) cherche à résoudre

$$W \circ X \simeq W \circ (G \cdot F), \tag{1}$$

où W est la matrice de poids associée à X, $\circ$ dénote le produit de Hadamard et G et F sont les facteurs non-négatifs à estimer.

De nombreux travaux portent aujourd'hui sur l'accélération de la FMN pour traiter des matrices de grandes dimensions [3]. Pour cela, plusieurs stratégies ont été proposées, parmi lesquelles les projections aléatoires [8]. L'accélération des méthodes de FPMN est tout aussi cruciale mais a pourtant été moins explorée dans la littérature. En particulier, à notre connaissance, appliquer des projections aléatoires au cas de matrices pondérées n'a jamais été considéré. Nous proposons donc une stratégie permettant d'utiliser de telles projections pour la FPMN.

La suite de cet article est structurée ainsi. La section 2 présente le concept de FMN compressée par projections aléatoires. Nous proposons notre stratégie de compression pour la FPMN dans la section 3 alors que nous étudions ses performances dans la section 4. Enfin, la conclusion et les perspectives de ces travaux sont présentés dans la section 5.

2 FMN utilisant les projections aléatoires

Nous présentons brièvement les principes de la FMN utilisant les projections aléatoires. Ces dernières sont devenues un outil très populaire pour l'apprentissage statistique des *big data* [9]. Basées sur le Lemme de Johnson-Lindenstrauss, elles supposent 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. Lorsque les données à traiter sont de grandes dimensions et de faible rang, il est alors possible de *compresser* ces données pour accélérer les calculs, au prix d'une légère erreur d'estimation.

Appliquées à la FMN, les projections aléatoires consistent à construire deux matrices de compression L et R, respectivement de taille $\nu \times n$ et $m \times \nu$ (avec $p \leq \nu \ll \min(n,m)$) et qui sont respectivement multipliées à gauche et à droite de X. Les matrices compressées, notées respectivement X_L and X_R, sont alors beaucoup plus petites que X et permettent d'accélérer les calculs de FMN. Cette stratégie de FMN Compressée (FMN-C) est décrite dans l'algorithme 1 [8], où $||\cdot||_{\mathcal{F}}$ dénote la norme de Frobenius. L et R n'ayant pas de contraintes de signe,

*F. Yahaya remercie la Région Hauts-de-France qui finance partiellement ses travaux. Les expériences présentées dans cet article ont été réalisées sur la plate-forme de calcul scientifique Calculco, gérée par le SCoSI de l'ULCO.

Algorithm 1 Stratégie de FMN-C

Initialiser G et F et construire L et R à partir de X.
Définir $X_L \triangleq L \cdot X$ et $X_R \triangleq X \cdot R$.
Répéter
  Définir $F_R \triangleq F \cdot R$.
  Mettre à jour $G = \arg\min_{\tilde{G}\geq 0} \left|\left|X_R - \tilde{G} \cdot F_R\right|\right|_{\mathcal{F}}$.
  Définir $G_L \triangleq L \cdot G$
  Mettre à jour $F = \arg\min_{\tilde{F}\geq 0} \left|\left|X_L - G_L \cdot \tilde{F}\right|\right|_{\mathcal{F}}$.
Jusqu'à un critère d'arrêt

les matrices X_L, G_L, X_R et F_R peuvent être à valeurs négatives. Comme G et F restent non-négatives, leurs mises à jour dans l'algorithme 1 s'obtiennent par factorisation en matrices semi-non-négatives [10]. Enfin, le critère d'arrêt peut être une erreur d'approximation de X, un nombre donné d'itérations ou un temps de calcul maximum. Parmi les stratégies proposées pour construire L et R, les projections aléatoires *structurées* [8], nommées RPI dans [9], sont bien adaptées à la FMN. A partir d'une matrice aléatoire gaussienne Ω_L, L est obtenue par décomposition QR de $(XX^T)^q \cdot X \cdot \Omega_L$, où q est un entier ($q = 4$ dans [8]). R est obtenue de manière similaire [8].

3 Méthode proposée

Nous proposons maintenant notre stratégie pour appliquer les projections aléatoires au problème de FPMN, fournissant ainsi des méthodes Compressées de FPMN (FPMN-C). Comme pour la FMN, la FPMN fait classiquement appel à des mises à jour alternées de G et F. Pour cela, deux stratégies permettent de prendre en compte W, c.-à-d. *(i)* conserver W dans le calcul des mises à jour [4] ou *(ii)* supprimer la matrice W dans une stratégie de Maximisation de l'Espérance (EM) [6]. Cette dernière suppose que les valeurs de W sont comprises [1] entre 0 et 1. L'étape E consiste à estimer les entrées incertaines de X :

$$X^{\text{comp}} = W \circ X + (\mathbb{1}_{n,m} - W) \circ (G \cdot F), \qquad (2)$$

où $\mathbb{1}_{n,m}$ est la matrice de taille $n \times m$ dont chaque entrée vaut 1. L'étape M consiste ensuite à appliquer des mises à jour de FMN à X^{comp} pour déduire G et F. Une fois que la FMN a convergé [6] ou après un nombre donné $\text{Max}_{\text{OutIter}}$ d'itérations [7], la matrice X^{comp} est mise à jour dans une nouvelle étape E, en utilisant les dernières estimées disponibles de G et F. En pratique, la stratégie EM founit des approches de FPMN moins sensibles à l'initialisation que celles obtenues par otimisation directe de (1) [6], tout en étant aussi voire plus rapides [7].

Il faut noter ici que la stratégie EM remplace $W \circ X$ dans le problème de FPMN par X^{comp}, permettant d'appliquer n'importe quel solveur, par exemple, mises à jour multiplicatives (MU), moindres carrés alternés (ALS), ou gradient projeté (PG). Mais ces mises à jour nécessitent toujours d'utiliser entièrement les matrices X^{comp}, G et F, ce qui peut être coûteux si

1. Une telle hypothèse n'est pas contraignante, puisque toute matrice non-nulle W peut être normalisée de sorte que sa valeur maximale soit 1.

Algorithm 2 Structure de FPMN-C-EM proposée.

Initialiser G et F.
Répéter
  {**Etape E**}
  Calculer X^{comp} selon (2) et construire L et R.
  Définir $X_L^{\text{comp}} \triangleq L \cdot X^{\text{comp}}$ et $X_R^{\text{comp}} \triangleq X^{\text{comp}} \cdot R$
  {**Etape M**}
  Pour compt=1 **à** $\text{Max}_{\text{OutIter}}$ **faire**
    Définir $F_R \triangleq F \cdot R$.
    Mettre à jour $G = \arg\min_{\tilde{G}\geq 0} \left|\left|X_R^{\text{comp}} - \tilde{G} \cdot F_R\right|\right|_{\mathcal{F}}$.
    Définir $G_L \triangleq L \cdot G$
    Mettre à jour $F = \arg\min_{\tilde{F}\geq 0} \left|\left|X_L^{\text{comp}} - G_L \cdot \tilde{F}\right|\right|_{\mathcal{F}}$.
  Fin Pour
Jusqu'à un critère d'arrêt

elles sont de grande dimension. Pour résoudre ce problème, nous proposons de compresser X^{comp} – en utilisant la compression RPI ou sa variante RSI plus stable [9] – à chaque étape E, de telle sorte que les calculs de FMN dans l'étape M sont accélérés. Cependant, contrairement à la FMN non-pondérée, les matrices L et R ne sont pas construites une fois pour toutes mais après chaque estimation de X^{comp} dans l'étape E. Ainsi, le temps de calcul gagné dans l'étape M peut être perdu par la construction des matrices de compression dans l'étape E. Ce problème sera étudié dans la prochaine section.

La structure globale de la méthode de FPMN Compressée et utilisant la stratégie EM (FPMN-C-EM) est présentée dans l'algorithme 2. Cette approche consiste en une boucle externe alternant étapes E et M. Chaque étape M consiste en une boucle de FMN qui est répétée $\text{Max}_{\text{OutIter}}$ fois. Enfin, les mises à jour des matrices G et F peuvent être réalisées par tout type de solveur, par exemple par MU (FPMN-C-EM-MU), PG (FPMN-C-EM-PG), ALS (FPMN-C-EM-ALS) ou par une boucle interne d'itérations de Nesterov (FPMN-C-EM-Ne).

4 Validation expérimentale

Nous étudions dans cette section les 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 [7] 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.

Pour cela, nous répétons 15 fois l'expérience suivante : nous générons aléatoirement des facteurs non-négatifs G^{theo} et F^{theo}, avec $n = m = 10000$ et $p = 5$. Leur produit fournit la matrice observée complète X^{theo} que nous échantillonnons aléatoirement avec un taux variant de 10 à 90% (avec un pas de 20%). Nous comparons la stratégie FPMN-EM originale à celle pro-

posée (FPMN-C-EM), en mettant à jour les facteurs matriciels par MU, PG, ALS et Nesterov. Pour chaque test, chaque méthode est initialisée avec les mêmes facteurs F et G tirés aléatoirement et est lancée durant 60 s sur un PC portable équipé d'un processeur Intel Core i7-4800MQ Quad Core, 32 GB de mémoire RAM et Matlab R2018a. Pour chaque méthode, nous considérons deux critères de performance, à savoir l'erreur relative de reconstruction (RRE) portant sur la qualité d'estimation de X^{theo} :

$$\text{RRE} \triangleq \left|\left|X^{\text{theo}} - G \cdot F\right|\right|_{\mathcal{F}}^2 / \left|\left|X^{\text{theo}}\right|\right|_{\mathcal{F}}^2 ; \qquad (3)$$

et le rapport signal à interférence (SIR) :

$$\text{SIR} = \sum_{j=1}^{p} 10 \log_{10} \left(\left|\left| \hat{f}_j^{\text{coll}} \right|\right|^2 / \left|\left| \hat{f}_j^{\text{orth}} \right|\right|^2 \right), \qquad (4)$$

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 dans F^{theo}. Nous compressons les matrices par RSI, une variante plus stable des RPI [9], avec $q = 4$ [8]. Nous faisons varier le nombre $\text{Max}_{\text{OutIter}}$ d'itérations à 1, 50 et 100, pour mesurer son effet sur les performances de FPMN. Enfin, les paramètres de la boucle interne d'itérations utilisant le gradient de Nesterov sont choisis ainsi : nous arrêtons une boucle interne de calcul de gradient de Nesterov après au plus $\text{Max}_{\text{InIter}} = 500$ mises à jour ou si le gradient est 1000 fois inférieur à celui initialement calculé en début de boucle.

TABLE 1 – Temps de calcul CPU médian atteint par les différentes méthodes testées.

Méthode	1 étape E (en s)	1 boucle dans l'étape M (en s)
FPMN-C-EM-MU	6.6546	0.0043
FPMN-EM-MU	2.3932	0.2773
FPMN-C-EM-PG	6.4367	0.0718
FPMN-EM-PG	2.4802	1.3917
FPMN-C-EM-ALS	6.7039	0.0025
FPMN-EM-ALS	2.4462	0.2778
FPMN-C-EM-Ne	6.7359	0.0366
FPMN-EM-Ne	2.3894	0.3275

Le tableau 1 résume les temps médians de calcul nécessaires à chaque étape des approches classiques FPMN-EM et leurs extensions compressées que nous proposons, lorsque $\text{Max}_{\text{OutIter}} = 50$. Nous pouvons noter que les approches compressées ont besoin de presque 3 fois plus de temps que les approches classiques pour réaliser une étape E. Cela montre que le goulot d'étranglement de notre stratégie est le calcul des projections structurées qui est réalisé à chaque ré-estimation de la matrice X^{comp}. Cependant et comme attendu, le calcul d'une boucle dans l'étape M avec notre approche compressée est 9 à 110 fois plus rapide que la variante non-compressée, grâce à la réduction de la taille des matrices X^{comp} et G (ou F, selon la mise à jour). Cela implique notamment que plus $\text{Max}_{\text{OutIter}}$ est élevé et plus il y a un intérêt à appliquer des projections aléatoires structurées à la FPMN. Le tableau 1 indique aussi que les approches utilisant des gradients semblent moins bénéficier de l'accélération dûe à la compression que les autres techniques. Cela peut être respectivement du à la recherche du pas optimal dans la mise à jour PG, qui est aussi lente dans les deux approches, ou aux itérations de Nesterov dans la boucle interne, dont seul le nombre maximum est fixé [7] et qui peuvent donc significativement varier en fonction des tests et des méthodes.

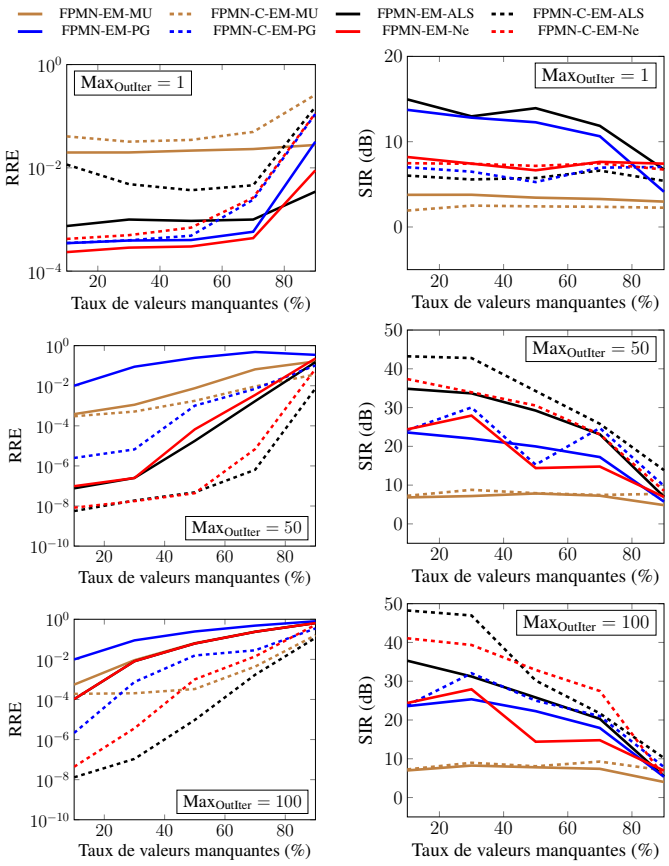

FIGURE 1 – RRE (à gauche) et SIR (à droite) versus le taux de valeurs manquantes et le nombre d'itérations par étape M.

La figure 1 montre les tracés des RRE et SIR médians obtenus par chaque méthode testée, en fonction du taux de valeurs manquantes et pour différentes valeurs du nombre d'itérations dans la boucle de l'étape M. Les tracés en trait plein (respectivement pointillé) montrent les performances des approches EM classiques (respectivement compressées) de FPMN. Nous nous attachons d'abord à l'erreur relative de reconstruction. Quand $\text{Max}_{\text{OutIter}} = 1$, les erreurs obtenues par les approches compressées sont plus élevées que celles obtenues avec les techniques non-compressées. Ce résultat est dû au fait qu'une seule mise à jour des facteurs matriciels est appliquée dans chaque étape M, ce qui implique (selon le tableau 1) que toutes les approches compressées génèrent moins d'itérations dans la boucle externe que les approches non-compressées. Cependant, les techniques faisant appel au gradient fournissent des RRE médians relativement similaires, qu'elles soient compressées ou non, quand au plus 50% des données sont manquantes. Quand $\text{Max}_{\text{OutIter}} = 50$, chaque approche faisant appel aux projections aléatoires

fournit un RRE bien plus faible que sa variante non-compressée. En particulier, les approches proposées utilisant les ALS ou le gradient de Nesterov surpassent toutes les approches testées. En fonction du taux de valeurs manquantes, leur RRE est 10 à 1000 fois inférieur à ceux, déjà bas, obtenus avec leurs variantes non-compressées. Quand $\text{Max}_{\text{OutIter}} = 100$, les performances atteintes avec les approches non-compressées sont sérieusement dégradées. C'est probablement du au fait qu'il y a trop d'itérations dans chaque étape M, eu égard au temps de calcul disponible. Cependant, les approches compressées présentent une chute de performance moins importante, mis à part la méthode utilisant les MU qui fournit des RRE légèrement plus bas et stables pour les valeurs de $\text{Max}_{\text{OutIter}}$ les plus élevées. Ces résultats montrent l'intérêt de la stratégie proposée qui permet d'accélérer significativement les itérations dans l'étape M et d'être relativement peu sensible aux valeurs élevées de $\text{Max}_{\text{OutIter}}$. Il est cependant à noter que l'erreur atteinte par toutes les méthodes quand 90% des données manquantes est plus faible avec une seule mise à jour des facteurs par étape M que lorsque ce nombre augmente, ce qui indique qu'il faudrait malgré tout chercher à optimiser la valeur de $\text{Max}_{\text{OutIter}}$ en fonction du taux de valeurs manquantes.

Les tracés de la partie droite de la figure 1 montrent les SIR otbenus dans les mêmes conditions. Tout d'abord, alors qu'on pourrait s'attendre à obtenir de meilleures estimées de F lorsque plus d'entrées de X sont observées, ce n'est pas toujours le cas. Cela peut être du au fait que la FMN est NP-difficile [3] et qu'obtenir une solution unique au problème n'est pas garanti dans le cas général. Les performances médianes des méthodes testées peuvent alors s'en retrouver localement affectées. Comme pour les erreurs de reconstruction, les SIR obtenus quand $\text{Max}_{\text{OutIter}} = 1$ sont faibles. Lorsque $\text{Max}_{\text{OutIter}} = 50$, l'apport de la compression pour les approches utilisant les MU ou PG est limité. Au contraire, les autres extensions compressées testées sont clairement plus performantes que leur variante respective. Cependant, la FPMN-C-EM-ALS fournit des SIR plus élevés et semble donc plus stable à la non-unicité de la solution que la méthode utilisant les gradients de Nesterov. Lorsque $\text{Max}_{\text{OutIter}}$ augmente, les SIR obtenus sont étonnamment stables ou meilleurs, alors que les RRE étaient légèrement décroissants, et ce même pour les méthodes non-compressées. Au final, les approches compressées utilisant les ALS fournissent un excellent compromis performance/vitesse, à la fois en complétion de matrices et en séparation de sources. Ces premiers résultats très encourageants devront être validés sur des jeux de données réalistes.

5 Conclusion et perspectives

Nous avons proposé un nouveau cadre EM pour appliquer des projections aléatoires à la FPMN. Pourvu que le nombre d'itérations dans l'étape M soit assez important, les approches de FPMN-C sont notablement plus performantes que les techniques non-compressées qu'elles étendent. Le temps élevé de calcul de l'étape E est la limitation actuelle de notre stratégie et pourrait être réduit par l'utilisation de calculateurs spécifiques dédiés aux projections aléatoires. Nous chercherons aussi à étendre ces approches à un cadre informé [5]. Il est aussi à noter que nous avons supposé connaître parfaitement le rang de la matrice X à factoriser. En pratique, l'estimation de son rang reste problématique et peut être fournie par un expert [5] ou, de manière alternative pour l'application complétion de matrice, être relâchée par l'ajout d'une pénalisation de type norme nucléaire [11]. Nous étudierons aussi ces aspects à l'avenir.

Références

[1] P. Paatero and U. Tapper, "Positive matrix factorization : a non negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[2] D.D. Lee and H.S. Seung, "Learning the parts of objects by non negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] Y. X. Wang and Y. J. Zhang, "Nonnegative matrix factorization : A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, June 2013.

[4] N.-D. Ho, *Non negative matrix factorization algorithms and applications*, Phd thesis, Université Catholique de Louvain, 2008.

[5] A. Limem, G. Delmaire, M. Puigt, G. Roussel, and D. Courcot, "Non-negative matrix factorization under equality constraints—a study of industrial source identification," *Applied Numerical Mathematics*, vol. 85, pp. 1–15, Nov. 2014.

[6] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. SIAM ICDM'06*. SIAM, 2006, pp. 549–553.

[7] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Fast nonnegative matrix factorization and completion using Nesterov iterations," in *Proc. LVA/ICA'17*, 2017, vol. LNCS 10179, pp. 26–35.

[8] M. Tepper and G. Sapiro, "Compressed nonnegative matrix factorization is fast and accurate," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2269–2283, May 2016.

[9] N. Halko, P.-G. Martinsson, and J. A Tropp, "Finding structure with randomness : Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[10] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, 2010.

[11] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proc. ICCV'13*, 2013, pp. 2488–2495.