

Projet de Compressed sensing :
Stable rank one matrix completion is solved by two rounds of
semidefinite programming relaxation

Clara CARLIER

Étienne LE NAOUR

23 mars 2020

Table des matières

1	Introduction	2
2	Présentation du problème	2
3	Les limites des méthodes classiques	4
3.1	La propagation est instable	4
3.2	La minimisation de la norme nucléaire ne fonctionne pas	5
4	Résultats principaux	5
5	Implémentation et méthode numérique	8
5.1	Premier test pour une matrice 3×3	8
5.2	Deuxième test pour une matrice 5×5	9
6	Conclusion	9

1 Introduction

La complétion de matrices permet de comprimer et de manipuler des grandes bases de données. Elle peut également servir à déduire des informations à partir de quelques mesures, comme dans le filtrage par exemple. Dans l'article *Stable rank one matrix completion is solved by two rounds of semidefinite programming relaxation* de Augustin Cosse et Laurent Demanet, nous nous intéressons à la complétion de matrices de rang faible. L'objectif est de fournir un algorithme capable de résoudre de manière stable et complète ce problème dans le cas des matrices de rang 1.

2 Présentation du problème

Nous nous plaçons dans le cadre de matrice de rang 1. Nous introduisons la notation $\mathcal{M}(1; m \times n)$ désignant l'ensemble des matrices de taille $m \times n$ et de rang 1.

Le but est d'estimer correctement X_0 une matrice inconnue qui appartient à $\mathcal{M}(1; m \times n)$ sachant que nous avons à notre disposition $\mathcal{O}(m+n)$ des entrées de X_0 (possiblement corrompues par un bruit ε). Dans le cas sans bruit, le problème peut s'écrire sous la forme :

$$\text{Trouver } X \in \mathbb{R}^{m \times n} \text{ tel que : } \text{rang}(X) = 1 \quad \text{et} \quad X_{i,j} = (X_0)_{i,j} \quad \forall (i,j) \in \Omega \quad (1.1)$$

où Ω correspond à l'ensemble des observations/mesures.

La matrice X_0 peut s'écrire sous la forme $x_0 y_0^T$. En effet, nous avons :

$$X_0 = x_0 y_0^T = \begin{pmatrix} (x_0)_1 \\ (x_0)_2 \\ \vdots \\ (x_0)_m \end{pmatrix} \begin{pmatrix} (y_0)_1 & (y_0)_2 & \cdots & (y_0)_n \end{pmatrix} = \begin{pmatrix} (x_0)_1(y_0)_1 & (x_0)_1(y_0)_2 & \cdots & (x_0)_1(y_0)_n \\ (x_0)_2(y_0)_1 & (x_0)_2(y_0)_2 & \cdots & (x_0)_2(y_0)_n \\ \vdots & \vdots & \ddots & \vdots \\ (x_0)_m(y_0)_1 & (x_0)_m(y_0)_2 & \cdots & (x_0)_m(y_0)_n \end{pmatrix}$$

Il est intéressant de voir que la matrice à retrouver peut s'écrire sous cette forme car nous remarquons que dans le meilleur des cas nous avons besoin de $m+n$ mesures pour retrouver exactement la matrice X_0 .

Malheureusement, en général le problème (1.1) ne peut pas se résoudre directement. Par exemple, si aucune information n'est connue sur une ligne (resp. une colonne) donnée, il devient impossible de récupérer les entrées correspondantes à cette ligne (resp. cette colonne).

De plus, s'il y a une entrée égale à 0, alors toute la ligne ou la colonne est égale à 0 et le problème de complétion ne sera pas injectif. Pour illustrer ce cas, considérons une matrice $X \in \mathbb{R}^{m \times n}$ de la forme $X = xy^T$. Imaginons que nous avons mesuré la première ligne et la dernière colonne de X , soit $m+n-1$ mesures, et que nous avons obtenu des 0. Cela signifie que toutes les matrices X de la forme $X = xy^T$ avec $x_1 = 0$ et $y_n = 0$ sont des solutions au problème.

En raison de ces limites, nous allons par la suite nous concentrer sur les matrices de taille $m \times n$ de rang 1 et avec des entrées différentes de 0.

Afin de formaliser l'injectivité, l'application $\mathcal{R}_\Omega : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{|\Omega|}$ est introduite. Elle permet de récupérer les entrées observées de la matrice. Nous notons \mathcal{R}_Ω^1 la restriction de \mathcal{R}_Ω aux matrices de

rang 1 qui n'admettent pas de ligne ni de colonne nulles. Chercher à inverser cette application revient à se demander s'il est possible de récupérer la matrice X à partir des valeurs observées $\mathcal{R}_\Omega(X)$ tout en sachant qu'elle est de rang 1.

Pour la suite, nous avons besoin de réaliser quelques rappels sur les graphes.

Définition 1. *Un graphe est dit biparti si l'ensemble de ses sommets peut être divisé en deux sous-ensembles disjoints U et V tel que chaque arête ait une extrémité dans U et l'autre dans V .*

Définition 2. *Un graphe est dit connexe lorsque, quelques soient les sommets, il existe un chemin les reliant.*

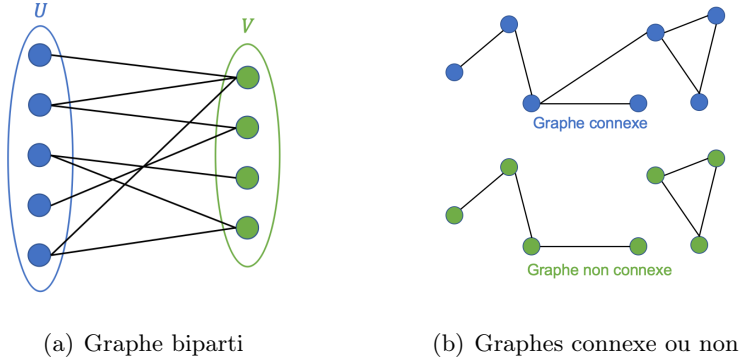


FIGURE 1 – Exemples de graphes

Soient U et V les ensembles d'indices des lignes et des colonnes de X . Notons $\mathcal{G}(U, V, \varepsilon)$ le graphe biparti associé au problème (1.1), où ε correspond à l'ensemble des arêtes du graphe. Nous avons $(i, j) \in \varepsilon$ si et seulement si $(i, j) \in \Omega$. Les sommets de \mathcal{G} sont annotés par l'indice de la colonne ou de la ligne correspondante. Les conditions de récupération de la matrice X sont liées aux propriétés de \mathcal{G} .

Lemme 1. \mathcal{R}_Ω est injective sur $\mathcal{M}(1; m \times n) \iff \mathcal{G}$ est connexe

Afin de donner un exemple, reprenons le cas où aucune information n'est connue sur une ligne ou une colonne donnée. Nous nous plaçons dans le cadre des matrices ayant deux lignes et trois colonnes, nous avons alors $U = \{1, 2\}$ et $V = \{1, 2, 3\}$. L'ensemble des mesures est donné par $\Omega = \{(1, 1), (1, 2), (2, 2)\}$. Ainsi le graphe biparti \mathcal{G} associé est donné par :

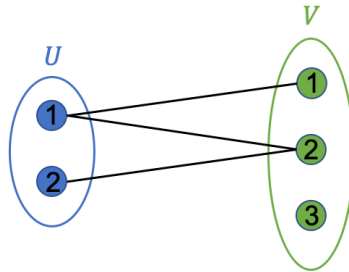


FIGURE 2 – Exemple

Dans ce cas-là, \mathcal{G} n'est pas connexe.

Quand ce lemme est respecté, l'article stipule que la complétion de matrice de rang 1 dans le cadre du problème (1.1) est réalisable (par exemple à travers une approche de propagation non linéaire). Cependant, cette solution n'est pas robuste lorsque du bruit est ajouté.

Lorsque le problème est corrompu par du bruit, une approche populaire consiste à minimiser la norme nucléaire qui est une approximation du rang mais il n'existe pas d'algorithme qui fournisse une estimation correcte (déterministe) de la stabilité. De plus, la norme nucléaire ne garantit pas la récupération de X_0 quand le bruit augmente.

3 Les limites des méthodes classiques

Les auteurs montrent que des méthodes classiques telles que la propagation ou encore la minimisation de la norme nucléaire ne sont pas toujours efficaces. Pour illustrer cela, ils utilisent des contre-exemples en prenant la matrice :

$$X_0 = \begin{pmatrix} 1 & ? \\ \delta & 1 \end{pmatrix}$$

3.1 La propagation est instable

La méthode de la propagation consiste à utiliser le fait que la matrice X_0 est de rang 1 pour retrouver les valeurs manquantes. Dans notre cas, si les mesures ne sont pas bruitées, nous devrions avoir :

$$X_{1,2} = \frac{X_{1,1}X_{2,2}}{X_{2,1}}$$

Nous retrouvons alors facilement que la valeur de ? est $1/\delta$.

Cependant, cette méthode s'avère très peu efficace en présence de mesures bruitées. Imaginons à présent que nous avons :

$$\begin{cases} \tilde{X}_{1,1} = (X_0)_{1,1} + \varepsilon_{1,1} \\ \tilde{X}_{2,2} = (X_0)_{2,2} + \varepsilon_{2,2} \\ \tilde{X}_{2,1} = (X_0)_{2,1} + \varepsilon_{2,1} \end{cases}$$

Nous supposons également que $\varepsilon_{2,1} = -0.9\delta$. Nous obtenons alors :

$$\tilde{X}_{1,2} = \frac{\tilde{X}_{1,1} \tilde{X}_{2,2}}{\tilde{X}_{2,1}} = \frac{1 + \varepsilon_{1,1}\varepsilon_{2,2} + \varepsilon_{1,1} + \varepsilon_{2,2}}{\delta + \varepsilon_{2,1}} = \frac{1 + \mathcal{O}(\delta)}{0.1\delta} = 10 \times (X_0)_{1,2}$$

Nous constatons que l'erreur est importante en présence de bruit. Nous en concluons donc que cette méthode est efficace quand les mesures sont non bruitées et à éviter sinon.

3.2 La minimisation de la norme nucléaire ne fonctionne pas

Soit une matrice $A \in \mathbb{R}^{m,n}$, nous notons $\text{sp}(A)$ son spectre qui est un vecteur de $\mathbb{R}^{\min(m,n)}$ contenant les valeurs singulières de A . La norme p de Shatten est définie telle que :

$$\|A\|_{S_p} = \|\text{sp}(A)\|_{l_p} = \left(\sum_{i=1}^{\min(m,n)} \sigma_i(A)^p \right)^{1/p}$$

Nous avons alors :

- $\text{rang}(A) = \|A\|_{S_0}$
- $\|A\|_{S_1}$ désigne la norme nucléaire et représente simplement la somme des valeurs singulières.

La norme 1 de Shatten est l'enveloppe convexe de la norme 0 de Shatten, donc du rang (cf. cours chapitre 6 pour la preuve). À partir de ce constat, nous pouvons appliquer le principe de relaxation convexe en résolvant le programme suivant :

$$\text{Minimiser } \|X\|_{S_1} \quad \text{tel que } X_{i,j} = (X_0)_{i,j} \quad \text{pour } i, j \in \Omega$$

Cependant, prenons $X = \begin{pmatrix} 1 & 2/\delta \\ \delta & 1 \end{pmatrix}$. Nous remarquons que $\text{rang}(X) = 2$ tandis que $\text{rang}(X_0) = 1$. Néanmoins, nous avons $\|X\|_{S_1} < \|X_0\|_{S_1}$ pour δ suffisamment petit.

4 Résultats principaux

Tout d'abord, rappelons ce qu'est un *semidefinite programming* (SDP). D'après le cours, la définition est la suivante :

Définition 3. *Un problème d'optimisation est un semidefinite programming s'il peut se réécrire sous la forme :*

$$\min_{X \in S^p} \left(\langle C, X \rangle : \tilde{\mathcal{A}}(X) = b, X \succeq 0 \right) \quad (\text{SDP})$$

où $C \in S^p$, $\tilde{\mathcal{A}} : S^p \rightarrow \mathbb{R}^m$ est une application linéaire et $b \in \mathbb{R}^m$ où S^p désigne l'ensemble des matrices symétriques de taille $p \times p$.

D'après le cours de Compressed Sensing sur la complétion de matrice (chapitre 6), le théorème 2.5 nous affirme que le problème de minimisation de la norme nucléaire pour le problème (1.1) (en remplaçant le rang par le norme nucléaire) est équivalent au problème suivant :

$$\begin{aligned} &\text{Minimiser } \text{Tr}(Y) + \text{Tr}(Z) \text{ avec } X \in \mathbb{R}^{m \times n}, Y \in S^m, Z \in S^n \\ &\text{tel que } \begin{pmatrix} Y & X \\ X^T & Z \end{pmatrix} \succeq 0 \text{ et } \mathcal{A}(X) = b \end{aligned}$$

Nous posons :

$$W = \begin{pmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{pmatrix}, \quad X_{1,1} = Y, \quad X_{2,2} = Z \quad \text{et} \quad X_{1,2} = X_{2,1}^T = X$$

Dans le contexte de la complétion de matrice de rang 1, comme $\text{Tr}(W) = \text{Tr}(X) + \text{Tr}(Z)$, alors le problème équivaut alors au SDP suivant :

$$\begin{aligned} &\text{Minimiser } \text{Tr}(W) \\ &\text{tel que } W = \begin{pmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{pmatrix} \succeq 0 \\ &\text{et } (X_{1,2})_{i,j} = (X_0)_{i,j}, \quad \forall (i, j) \in \Omega \end{aligned} \quad (1.12)$$

Quand $X_0 = x_0 y_0^T$, la matrice W est une approximation de la matrice de rang 1 :

$$W_0 = \begin{pmatrix} x_0 x_0^T & x_0 y_0^T \\ y_0 x_0^T & y_0 y_0^T \end{pmatrix} \quad (1.13)$$

Il est intéressant de noter que le problème (1.12) n'optimise que sur les monômes de bi-degrés (1, 1).

Par la suite, l'idée principale de l'article va être d'étendre ce type de formulation aux monômes de degrés supérieurs. En prenant $z_0 = (x_0, y_0)$ et en notant $z_0^{(2)} = \text{vect}(z_0^{\otimes 2}) = \text{vect}(z_0 \otimes z_0)$, nous allons nous intéresser aux matrice de la forme :

$$M_0 = \begin{pmatrix} 1 & z_0 & z_0^{(2)} \\ z_0 & z_0^{\otimes 2} & z_0 \otimes z_0^{(2)} \\ z_0^{(2)} & z_0^{(2)} \otimes z_0 & z_0^{(2)} \otimes z_0^{(2)} \end{pmatrix} \quad (1.14)$$

Nous avons alors $W_0 = z_0^{\otimes 2}$. Le nouveau problème à résoudre est alors donné par :

$$\text{Minimiser } \text{Tr}(M) \quad \text{tel que } \mathcal{A}(M) = b \text{ et } M \succeq 0 \quad (1.15)$$

De plus nous avons des **contraintes** sur M . En effet, nous avons les contraintes initiales du problème (1.1) ainsi que des contraintes dites "de structure" et "d'ordre supérieur". Ces deux derniers types de contraintes sont destinés à contraindre la matrice M à être de la forme de la matrice M_0 donnée par (1.14).

- **contraintes de structure** : elles sont dues à l'ajout des monômes dans (1.14), cela a crée des relations supplémentaires entre les entrées de M_0 , ces contraintes permettent de garantir les égalités nécessaires.
- **contraintes d'ordre supérieur** : cette contrainte permet de garantir que les produits des monômes x_0 et y_0 ne dépassent pas un certain degré imposé.

Par exemple, la contrainte de structure impose qu'à chaque endroit $(M_0)_{i_1, j_1}, (M_0)_{i_2, j_2}, \dots, (M_0)_{i_k, j_k}$ qui vaut $x_0 x_0^T \otimes x_0 y_0^T$ alors nos prédictions doivent être telles que : $M_{i_1, j_1} = M_{i_2, j_2} = \dots = M_{i_k, j_k}$. La contrainte d'ordre supérieur signifie dans notre cas que les monômes doivent être de degrés inférieur à 4.

Note : Nous nous apercevons que résoudre le problème (1.15) revient à résoudre le problème (1.12). Le passage d'un problème à l'autre se fait par des arguments algébrique. En effet, si la matrice W_0 est de rang 1, alors M_0 est de rang 1 aussi. Augmenter la dimension du problème a pour intérêt de faire apparaître les monômes de degrés supérieur à 1 et leur produit croisé. Nous supposons que c'est ce qui permet de retrouver plus facilement la matrice de rang initiale.

Le but du papier est de prouver que la formulation (1.15) est suffisante pour résoudre le problème de complétion de matrice de rang 1 quand le lemme 1 est respecté et de fournir un algorithme évolutif.

Nous notons que bien entendu, ce problème peut se réécrire de la façon suivante lorsque les mesures sont bruitées (et c'est par ailleurs dans ce cadre que la méthode est intéressante).

En décomposant l'application \mathcal{A} en deux : \mathcal{A}_S la partie structurelle et $\tilde{\mathcal{A}}$ les contraintes restantes, nous pouvons réécrire le problème (1.15) sous la forme :

$$\text{Minimiser } \text{Tr}(M) \quad \text{tel que } \mathcal{A}_S(M) = 0, \quad \|\tilde{\mathcal{A}}(M) - \tilde{b}\| \leq \sigma \text{ et } M \succeq 0 \quad (1.16)$$

où $\tilde{b} = (X_0)_{i,j} + \varepsilon_{i,j}$ est la mesure du bruit. Cette nouvelle écriture est possible car les contraintes de structure ne sont pas affectées par le bruit.

Regardons à présent quels sont les principaux résultats de l'article.

Le premier théorème de l'article nous affirme que le problème (1.15) équivaut à résoudre le problème de complétion de matrice de rang 1 de façon efficace lorsque les conditions du lemme 1 sont respectées.

Théorème 1. *Le problème (1.1) dans le contexte du lemme 1 avec $X \in \mathcal{M}^*(1; m \times n)$ peut se résoudre exactement à l'aide de "two rounds of SDP relaxation" en minimisant la trace.*

Le deuxième théorème nous donne une borne supérieure sur l'erreur d'approximation de la matrice M_0 (1.14) par la solution au problème (1.16) (dans le cadre de mesures bruitées).

Théorème 2. *Soit M_0 la matrice de rang 1 donnée par (1.14) pour $z_0 \in \mathbb{R}^{m+n-1}$ et soit M solution de (1.16). Supposons que les conditions du lemme 1 sont vérifiées. Posons $\|\varepsilon\|_2 = \sqrt{\sum_{(i,j) \in \Omega} \varepsilon_{i,j}^2}$. Nous avons alors :*

$$\frac{\|M - M_0\|_F}{\|M_0\|_F} \leq c_0(m+n)^{7/2} \|\varepsilon\|_2$$

où c_0 ne dépend que des entrées de X_0 .

Une grande partie de l'article consiste à prouver ces deux théorèmes, nous ne nous attardons pas sur ces preuves dans la suite de notre rapport.

Nous avons choisi de simuler des problèmes de complétion de matrices et essayé de les résoudre avec les différentes méthodes introduites précédemment : à l'aide de W (1.12) et de M (1.15).

5 Implémentation et méthode numérique

Nous avons donc implémenté un code permettant de simuler une matrice X_0 dont nous souhaitons récupérer les entrées manquantes et qui vérifie $X_0 = x_0 y_0^T$. Nous avons ensuite construit les matrices W_0 et M_0 correspondantes à l'aide des formules (1.13) et (1.14) respectivement.

Afin de construire le problème, nous avons créé un masque qui est une matrice de 0 et de 1 de la même taille que X_0 . Les valeurs du masque sont simulées selon une loi de Bernoulli de paramètre 1 et une proportion p qui est à définir. À l'aide de ce masque, nous pouvons construire les matrices observées de X_0 , W_0 et M_0 . Pour ces deux dernières, nous construisons des matrices de masque plus grandes de sorte que :

$$\text{Masque } W_0 = \left(\begin{array}{c|c} 0 & \text{masque } X_0 \\ \hline (\text{masque } X_0)^T & 0 \end{array} \right)$$

$$\text{Masque } M_0 = \left(\begin{array}{c|c|c} 1 & 0 & \text{vec}(\text{masque } W_0)^T \\ \hline 0 & \text{masque } W_0 & 0 \\ \hline \text{vec}(\text{masque } W_0) & 0 & \text{masque } W_0 \otimes (\text{masque } W_0)^T \end{array} \right)$$

Enfin, pour résoudre le problème nous avons utilisé le package `cvxpy`.

5.1 Premier test pour une matrice 3×3

Nous avons réalisé un premier test sur une matrice de taille 3×3 . Pour illustrer nos résultats, nous donnons des valeurs précises obtenues sur une simulation. Nous avons comme matrice à estimer et observée :

$$X_0 = \begin{pmatrix} 333 & 2775 & 185 \\ 108 & 900 & 60 \\ 648 & 5400 & 360 \end{pmatrix} \quad \text{et} \quad X_0^{obs} = \begin{pmatrix} 333 & 2775 & 0 \\ 108 & 900 & 60 \\ 648 & 0 & 360 \end{pmatrix}$$

Nous observons alors 77.78% des entrées de la matrice X_0 . Les matrices W_0 et W_0^{obs} associées sont données par :

$$W_0 = \begin{pmatrix} 1369 & 444 & 2664 & 333 & 2775 & 185 \\ 444 & 144 & 864 & 108 & 900 & 60 \\ 2664 & 864 & 5184 & 648 & 5400 & 360 \\ 333 & 108 & 648 & 81 & 675 & 45 \\ 2775 & 900 & 5400 & 675 & 5625 & 375 \\ 185 & 60 & 360 & 45 & 375 & 25 \end{pmatrix} \quad \text{et} \quad W_0^{obs} = \begin{pmatrix} 0 & 0 & 0 & 333 & 2775 & 0 \\ 0 & 0 & 0 & 108 & 900 & 60 \\ 0 & 0 & 0 & 648 & 0 & 360 \\ 333 & 108 & 648 & 0 & 0 & 0 \\ 0 & 900 & 0 & 0 & 0 & 0 \\ 0 & 60 & 360 & 0 & 0 & 0 \end{pmatrix}$$

Dans ce cas-là, la matrice M_0 est de dimension $(1 + (u + v) + (u + v)^2) = 43 \times 43$. Nous n'allons évidemment pas l'écrire. Les solutions obtenues à l'aide de W_0 et M_0 sont les suivantes :

$$\text{Sol}_{W_0} = \begin{pmatrix} 333 & 2775 & 185 \\ 108 & 900 & 60 \\ 648 & 400 & 360 \end{pmatrix} \quad \text{et} \quad \text{Sol}_{M_0} = \begin{pmatrix} 332 & 2777 & 194 \\ 108 & 897 & 61 \\ 650 & 405 & 360 \end{pmatrix}$$

Avec W_0 , l'algorithme semble retrouver la valeur 185 mais pas 5400. Pourtant, une grande partie des entrées de la matrice sont observées. Il est étonnant qu'il ne retrouve pas cette valeur. Le problème en dimension 3×3 est que rapidement la matrice observée ne vérifie pas les conditions d'injectivité. Quand le pourcentage de valeurs observées diminue, les résultats deviennent rapidement mauvais car il y a rapidement deux voir trois 0 sur une ligne ou sur une colonne.

Avec M_0 , les valeurs prédites ne sont pas égales aux mesures mais restent tout de même assez proches des valeurs réelles. Cela ne nous étonne pas car, comme expliqué dans le notebook, nous n'avons pas réussi à inclure toutes les contraintes nécessaires (structure et plus haut degrés) lorsque nous introduisons le nouveau problème (1.15).

5.2 Deuxième test pour une matrice 5×5

Nous réalisons la même démarche que précédemment sur une matrice de taille 5×5 . Nous avons :

$$X_0 = \begin{pmatrix} 185 & 2923 & 2368 & 592 & 37 \\ 60 & 948 & 768 & 192 & 12 \\ 360 & 5688 & 4608 & 1152 & 72 \\ 45 & 711 & 576 & 144 & 9 \\ 375 & 5925 & 4800 & 1200 & 75 \end{pmatrix} \quad \text{et} \quad X_0^{obs} = \begin{pmatrix} 185 & 0 & 2368 & 0 & 37 \\ 0 & 948 & 0 & 192 & 0 \\ 0 & 5688 & 4608 & 1152 & 72 \\ 0 & 0 & 576 & 0 & 0 \\ 375 & 5925 & 4800 & 0 & 75 \end{pmatrix}$$

Nous observons alors 56% des entrées de X_0 . Les solutions obtenues sont alors :

$$\text{Sol}_{W_0} = \begin{pmatrix} 185 & 2923 & 2368 & 592 & 37 \\ 60 & 948 & 768 & 192 & 12 \\ 360 & 5688 & 4608 & 1152 & 72 \\ 45 & 711 & 576 & 144 & 9 \\ 375 & 5925 & 4800 & 1200 & 75 \end{pmatrix} \quad \text{et} \quad \text{Sol}_{M_0} = \begin{pmatrix} 166 & 2901 & 2340 & 402 & 37 \\ 63 & 1014 & 921 & 190 & 14 \\ 370 & 5677 & 4572 & 1137 & 71 \\ 30 & 702 & 569 & 97 & 9 \\ 378 & 5904 & 4756 & 1110 & 74 \end{pmatrix}$$

La solution obtenue avec W_0 retrouve toutes les valeurs réelles. Comme la matrice est de taille plus grande, le nombre de valeurs observées est plus élevé bien que le pourcentage soit plus petit. Cela doit nous permettre de récupérer des informations importantes et ainsi obtenir un résultat meilleur que pour la matrice 3×3 .

À nouveau, nous observons le même problème que précédemment avec la solution obtenue à travers M_0 . Les valeurs ne sont pas très éloignées mais sont inexactes, sauf pour deux entrées.

6 Conclusion

L'article a pour but de résoudre la complétion de matrice de rang 1 lorsque les conditions du lemme 1 sont respectées et que l'on a suffisamment d'observations.

Les auteurs montrent dans un premier temps que les méthodes classiques de complétion ne sont pas adaptées dans le cadre de matrice de rang 1. Par exemple, même si la méthode de propagation est très efficace pour retrouver les valeurs inconnues de la matrice (lorsqu'on a suffisamment de mesures) lorsque les mesures ne sont pas bruitées, dans le cadre de mesures bruitées, cette méthode peut se tromper lourdement (exemple en introduction).

Dans un deuxième temps, les auteurs nous rappellent que l’enveloppe convexe du rang d’une matrice est sa norme nucléaire (somme des valeurs singulières ou encore sa trace) et dans ce cas, nous pouvons minimiser la norme nucléaire (sous contrainte des mesures connues) pour compléter la matrice. Les auteurs stipulent alors que résoudre ce problème est équivalent à résoudre le problème de semidefinite programming (1.12). L’équivalence entre ces deux problèmes a notamment été prouvé dans notre cours de compressed sensing (théorème 2.5 du chapitre correspondant). Nous nous apercevons que le nouveau problème revient à compléter une matrice de dimension plus grande qui prend en compte des interactions (de degré 1) entre les vecteurs ayant servi à générer la matrice de rang 1 initiale. Par la suite, les auteurs nous présentent une troisième façon équivalente de résoudre le problème initial, cette façon est la (1.15). Ceci revient à compléter une matrice qui est d’une dimension encore plus grande et qui tient compte des interactions de degré 4 au moins entre les vecteurs ayant servi à générer la matrice initiale. La procédure est la même que pour le problème précédent, nous cherchons à minimiser la trace de la matrice tout en gardant la matrice semi-définie positive et sous les contraintes de mesure, de plus nous rajoutons des contraintes de structure et de degré supérieur.

Dans un troisième temps, les auteurs énoncent les principaux résultats de l’article : le théorème 1 et le théorème 2. Le théorème 1 stipule que si le lemme 1 est respecté (l’injectivité dans les mesures est suffisante), alors la dernière façon présentée qui permet de résoudre le problème est efficace pour compléter la matrice initiale.

Le théorème 2, quant à lui, stipule qu’en utilisant la dernière méthode présentée, dans le cadre de mesures bruitées, il est possible de majorer l’erreur d’approximation de la matrice par une constante. **Par exemple, ceci n’est pas le cas dans la méthode de propagation.**

Par la suite, nous avons décidé d’essayer d’implémenter les deux méthodes : ‘first round SDP relaxation’ et ‘second round SDP relaxation’. Nous les avons testé sur des matrices volontairement incomplètes dont nous connaissons les valeurs réelles initiales. Malheureusement, nous n’avons pas réussi à implémenter correctement les contraintes de structure et celles de plus haut degré dans le ‘second round SDP relaxation’. Par conséquent, la méthode ‘first round SDP relaxation’ paraît plus efficace pour résoudre les problèmes de complétion que nous avons simulé mais cela n’est sûrement pas valable une fois ces contraintes ajoutées. Néanmoins, ces implémentations nous ont permises de voir comment fonctionnent ces programmes d’optimisation et notamment de remarquer que pour une matrice X_0 de taille relativement petite, la taille de la matrice M_0 peut être très grande à cause des produits tensoriels.

Pour conclure, nous aimerions à l’avenir réussir à correctement appliquer ces deux méthodes et notamment la deuxième en incluant toutes les contraintes qui vont avec afin de les appliquer sur des jeux de données réelles et non simulées.