

Algorithme proximal efficace pour l'approximation Hankel rang faible

Florent CHATELAIN¹ Julien FLAMANT² Nicolas LE BIHAN¹

¹Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

²Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

Résumé – Cet article s'intéresse à l'approximation de rang faible de matrices structurées (SLRA) de type Hankel. Ce problème apparaît notamment en traitement du signal lorsque l'on s'intéresse à la décomposition d'un signal en somme d'harmoniques amorties. Nous montrons qu'il est possible de tirer parti de la structure Hankel rang faible afin d'établir un calcul Hankel efficace optimisant le coût mémoire et computationnel. Un algorithme proximal efficace est proposé en reformulant les contraintes Hankel du problème SLRA via une pénalisation appropriée. Les simulations numériques démontrent la pertinence de l'approche en termes d'erreur de reconstruction et de temps de calcul.

Abstract – This paper studies the structured low-rank approximation (SLRA) of Hankel matrices. This problem notably appears in signal processing for the decomposition of a signal into a sum of damped harmonics. We show that it is possible to take advantage of the Hankel low-rank structure to establish an efficient Hankel calculus optimizing the memory and computational cost. An efficient proximal algorithm is proposed by reformulating the Hankel constraints of the SLRA problem via an appropriate penalization. Numerical simulations show the relevance of the approach in terms of reconstruction error and computation time.

1 Introduction

L'information présente dans les données admet, dans de nombreux problèmes de traitement du signal, une représentation matricielle de petit rang. Cette matrice peut en plus exhiber une certaine structure liée aux symétries du problème. L'*approximation de rang faible sous contrainte* (SLRA) vise alors à récupérer cette information en minimisant, sous les contraintes appropriées de petit rang et de structure, la distance à la matrice des données. Dans ce papier, les contraintes considérées sont linéaires et définissent le sous-espace vectoriel des matrices à symétrie Hankel.

Ces matrices sont au cœur de méthodes de traitement du signal dites à *haute-résolution* afin d'estimer des paramètres fréquentiels d'un signal. En effet une matrice Hankel est de rang r lorsqu'elle encode un signal composé d'une somme de r sinusoides complexes (modulés exponentiellement), et cette propriété peut être utilisée afin d'identifier, aux erreurs numériques près, leurs paramètres. Lorsque le signal est corrompu par un bruit additif, de nombreuses méthodes ont alors été proposées dans la littérature pour résoudre le problème Hankel-SLRA, qui est formulé en (2), et remonter ainsi au signal utile. Ce problème est toujours d'actualité, en atteste le récent article de synthèse [5]. En particulier, de nombreuses méthodes d'optimisation ont été proposées ces dernières années pour améliorer les performances d'estimation. Par exemple, un algorithme de type ADMM est proposé dans [1], un autre de type Douglas-Rachford (qui inclut des contraintes supplémentaires de positivité) est introduit dans [3], une méthode de gradient dual est notamment étudiée dans [4], et une bibliothèque logicielle pour résoudre les problèmes SLRA a été proposée dans [8]. Ces méthodes itératives nécessitent toutes de stocker et de manipuler des matrices pleines. Le coût de stockage et de calcul est souvent rédhibitoire dès lors que les dimensions des signaux et des matrices ne sont pas suffisamment petites. Il s'avère pourtant possible d'exploiter la redondance des matrices Hankel afin de les stocker et de les manipuler de manière plus efficace. Ces idées ont été explorées

dans le contexte des méthodes de *singular spectrum analysis* (SSA) [6]; nous étudions ici leur potentiel dans le cadre de problèmes SLRA avec structure Hankel. En particulier, cet article présente un algorithme proximal à faible coût calculatoire et mémoire dédié au problème Hankel-SLRA. Les résultats sont étayés par des simulations qui permettent de retrouver les mêmes performances statistiques que l'état de l'art, pour un coût considérablement réduit. Ce résultat ouvre la porte au débruitage de matrices Hankel, et des signaux associés, en grandes dimensions.

Notations. Les vecteurs, matrices et opérateurs sont respectivement distingués par une écriture gras minuscule (\mathbf{a}), gras majuscule (\mathbf{A}) et scripte majuscule (\mathcal{A}). On note \mathbf{A}^H la matrice \mathbf{A} conjuguée-transposée et $\|\mathbf{A}\|_F = \sqrt{\text{trace}(\mathbf{A}^H \mathbf{A})}$ sa norme de Frobenius. L'opérateur "*" désigne le produit de convolution entre des vecteurs, "." est le produit terme à terme entre des matrices (ou des vecteurs) de même dimension alors que "×" dénote le produit matriciel usuel.

2 Problème Hankel-SLRA

Pour un nombre de lignes m fixé, le signal complexe $\mathbf{z} = (z_1, \dots, z_N) \in \mathbb{C}^N$ de longueur $N \geq m$ est encodé dans la matrice de Hankel suivante :

$$\mathbf{H} = \mathcal{H}\mathbf{z} \equiv \begin{pmatrix} z_1 & z_2 & z_3 & \cdots & \cdots & z_n \\ z_2 & z_3 & \ddots & \ddots & \ddots & \vdots \\ z_3 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & z_{N-1} \\ z_m & \cdots & \cdots & \cdots & z_{N-1} & z_N \end{pmatrix}, \quad (1)$$

où $n = N - m + 1$ et $\mathcal{H} : \mathbb{C}^N \rightarrow \mathbb{C}^{m \times n}$ est l'opérateur de Hankelisation associé. Les entrées de $\mathbf{H} \in \mathbb{C}^{m \times n}$ sont donc les $H_{ij} = z_{i+j-1}$ et la k -ème *anti-diagonale* de \mathbf{H} est constante et égale à z_k pour tout $k = 1, \dots, N$. On note

$\mathcal{H} = \{\mathcal{H}z : z \in \mathbb{C}^N\}$ le sous-espace vectoriel des matrices Hankel de $\mathbb{C}^{m \times n}$.

Étant donné un vecteur de données \mathbf{y} représenté par sa matrice Hankel $\mathbf{Y} = \mathcal{H}\mathbf{y} \in \mathbb{C}^{m \times n}$, le problème Hankel-SLRA vise à estimer une matrice Hankel $\widehat{\mathbf{X}} = \mathcal{H}\widehat{\mathbf{x}}$ de rang $r \ll m, n$ qui est obtenue comme :

$$\widehat{\mathbf{X}} = \arg \min_{\mathbf{X} \in \mathcal{H}} \|\mathbf{W} \cdot (\mathbf{Y} - \mathbf{X})\|_{\text{F}}^2, \quad \text{t.q. } \text{rang } \mathbf{X} \leq r, \quad (2)$$

où $\mathbf{W} = \mathcal{H}\mathbf{w} \in \mathbb{R}^{m \times n}$ est une matrice Hankel de pondération encodée par un vecteur $\mathbf{w} \in \mathbb{R}^N$. Sans perte de généralité, on suppose que $\|\mathbf{w}\|_{\infty} = 1$ si bien que $\max_{i,j} W_{ij} = 1$.

Un choix naturel pour le vecteur de pondération est $\mathbf{w} = \boldsymbol{\kappa}^{1/2} \equiv (\sqrt{\kappa_1}, \dots, \sqrt{\kappa_N})$ où les éléments κ_k sont les inverses des longueurs de chaque anti-diagonale d'une matrice de $\mathbb{C}^{m \times n}$:

$$\kappa_k = \begin{cases} 1/k & \text{si } 1 \leq k \leq p, \\ 1/p & \text{si } p \leq k \leq q, \\ 1/(m+n-k) & \text{si } q \leq k \leq N, \end{cases} \quad (3)$$

avec $p = \min(m, n)$, $q = \max(m, n)$ et $N = m + n - 1$. Dans ce cas la fonction objectif devient $\|\mathbf{W} \cdot (\mathbf{Y} - \mathbf{X})\|_{\text{F}}^2 = \|\mathbf{y} - \mathbf{x}\|_{\text{F}}^2$ et le problème (2) correspond à l'estimateur du maximum de vraisemblance (MV) pour débruiter $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$ lorsque le bruit $\boldsymbol{\epsilon}$ est gaussien complexe circulaire et i.i.d.¹

3 Calcul Hankel efficace

La notion d'*efficacité* est définie ici au sens d'un coût de stockage réduit ainsi que de calculs rapides. Pour représenter une matrice Hankel (1), seul le vecteur de longueur $N = m + n - 1$ des termes anti-diagonaux peut en effet être stocké. Par ailleurs deux opérations clés pour les algorithmes SLRA peuvent être calculées rapidement grâce à ce vecteur, sans jamais stocker ni manipuler de matrices pleines de $\mathbb{C}^{m \times n}$.

Calcul matriciel et SVD tronquée. Les méthodes de décomposition en valeurs propres ou singulières (SVD) dites *sans matrice* demandent de savoir évaluer les produits matrice-vecteur, sans nécessairement stocker de manière explicite la matrice à décomposer. Ainsi, la méthode des puissances itérées, ou ses généralisations comme l'algorithme d'Arnoldi² [7], sont particulièrement utiles lorsque la matrice est si grande que son stockage et sa manipulation coûteraient beaucoup de mémoire et de temps de calcul.

Pour notre problème d'optimisation (2), ces méthodes itératives sont d'autant plus efficaces pour deux raisons : 1) elles permettent de calculer uniquement les vecteurs singuliers dominants nécessaires à l'approximation de rang faible, et 2) il est possible de calculer rapidement le produit matrice-vecteur, les matrices Toeplitz ou Hankel pouvant être étendues en des matrices circulantes. Si $\mathbf{b} = (b_1, \dots, b_n)$ est un vecteur colonne de \mathbb{C}^n et $\mathcal{H}z$ est la matrice Hankel donnée en (1), alors

¹Plus généralement, cette pondération Hankel permet de couvrir les problèmes de débruitage minimisant $\|\Delta(\mathbf{y} - \mathbf{x})\|_{\text{F}}^2$ pour toute matrice Δ diagonale (estimateur MV pour un bruit blanc gaussien non stationnaire), mais également le cas des données manquantes si $w_k = 0$.

²Ces algorithmes sont disponibles dans les langages de calcul scientifique courants, e.g. via la fonction `scipy.sparse.linalg.svds` en Python.

le produit $\mathcal{H}z \times \mathbf{x}$ peut se calculer comme les m premiers termes de la convolution circulaire entre le signal z et le signal

$$\tilde{\mathbf{b}} = (0, \dots, 0, b_n, b_{n-1}, \dots, b_1) \in \mathbb{C}^N,$$

qui est le vecteur \mathbf{b} retourné et précédé de $m - 1$ zéros.

Le théorème de convolution permet alors de calculer rapidement le produit matrice-vecteur à l'aide de transformée de Fourier rapide et son inverse, notées FFT et iFFT :

$$\mathcal{H}z \times \mathbf{b} = \text{iFFT} \left(\text{FFT}(z) \cdot \text{FFT}(\tilde{\mathbf{b}}) \right) \{ : m \},$$

où la notation $\{ : m \}$ désigne les m premiers termes du signal obtenu. Ceci permet de calculer le produit matrice-vecteur en seulement $O(N \log(N))$ opérations au lieu de $O(mn)$ par le calcul standard, et en stockant uniquement des signaux de longueur N au lieu de la matrice Hankel $m \times n$. Il est ainsi possible de construire un algorithme rapide afin de calculer, directement à partir du vecteur $z \in \mathbb{C}^N$, la SVD tronquée d'une matrice Hankel $\mathbf{H} = \mathcal{H}z$, notée

$$\text{SVD}_r \mathbf{H} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^{\text{H}},$$

où $\{\sigma_i\}$ désigne les r valeurs singulières dominantes de \mathbf{H} et $\{\mathbf{u}_i\}, \{\mathbf{v}_i\}$ les vecteurs singuliers à gauche et à droite associés.

Approximation Hankel. Le projeté orthogonal d'une matrice $\mathbf{A} \in \mathbb{C}^{m \times n}$ sur le sous-espace des matrices Hankel est défini comme

$$\mathcal{P}\mathbf{A} \equiv \arg \min_{\mathbf{H} \in \mathcal{H}} \|\mathbf{A} - \mathbf{H}\|_{\text{F}}.$$

Une base orthogonale de \mathcal{H} (pour le produit scalaire $\langle \mathbf{A}, \mathbf{B} \rangle = \text{trace } \mathbf{A}\mathbf{B}^{\text{H}}$ dont est issue la norme de Frobenius) est formée par les matrices anti-diagonales de $\mathbb{C}^{m \times n}$ et une expression explicite du projecteur est obtenue comme

$$\mathcal{P}\mathbf{A} = \mathcal{H}(\mathbf{a}) \quad (4)$$

où le k -ème élément du vecteur $\mathbf{a} \in \mathbb{C}^N$ est la moyenne des éléments de \mathbf{A} situés sur la k -ème anti-diagonale.

Lorsque $\mathbf{A} = \mathbf{u}\mathbf{v}^{\text{H}}$ est une matrice de rang 1, ce vecteur peut se calculer comme la convolution entre les vecteurs \mathbf{u} et \mathbf{v} , i.e. $\mathbf{a} = \boldsymbol{\kappa} \cdot (\mathbf{u} * \mathbf{v})$, les éléments du vecteur $\boldsymbol{\kappa} \in \mathbb{C}^N$ étant les inverses des longueurs de chaque anti-diagonale définis en (3). Le théorème de convolution assure encore que ce calcul peut être effectué par FFT en $O(N \log N)$ opérations.

Lorsque \mathbf{A} est une matrice de rang r telle que $\text{SVD}_r \mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^{\text{H}}$, par linéarité de l'opérateur \mathcal{P} , le projeté $\mathcal{P}\mathbf{A}$ est finalement encodé par le vecteur

$$\mathbf{a} = \boldsymbol{\kappa} \cdot \sum_{i=1}^r \sigma_i (\mathbf{u}_i * \mathbf{v}_i), \quad (5)$$

et peut donc se calculer en $O(rN \log N)$ opérations au lieu de $O(mn)$ par un calcul direct. Ceci est d'autant plus avantageux que m et n sont grands et que le rang r est faible. Par ailleurs, seul le vecteur $\mathbf{a} \in \mathbb{C}^N$ doit être stocké en plus des r vecteurs $\mathbf{u}_i \in \mathbb{C}^m$ et $\mathbf{v}_i \in \mathbb{C}^n$, $i = 1, \dots, r$, de la décomposition.

4 Algorithme Hankel-SLRA efficace

Il s'agit maintenant d'obtenir une solution au problème Hankel-SLRA (2) en utilisant les calculs efficaces précédents, et sans devoir stocker ni manipuler des matrices de $\mathbb{C}^{m \times n}$.

Méthode de Cadzow. L'algorithme de Cadzow est une méthode populaire qui consiste à projeter alternativement sur l'ensemble des matrices de rang faible et sur l'espace Hankel :

$$\mathbf{X}_{k+1} = \mathcal{P}(\text{SVD}_r \mathbf{X}_k),$$

jusqu'à un critère d'arrêt. Cette méthode simple peut être implémentée de manière efficace car elle ne dépend que des deux opérations efficaces présentées dans le paragraphe précédent. Elle présente néanmoins l'inconvénient de ne pas converger vers une solution locale du problème SLRA [5] et s'avère sous-optimale pour notre problème d'optimisation (2).

Solution SLRA. Le problème Hankel-SLRA peut être reformulé comme

$$\widehat{\mathbf{X}} = \arg \min_{\mathbf{X} \in \mathcal{H}} \frac{1}{2} \|\mathbf{W} \cdot (\mathbf{Y} - \mathbf{X})\|_{\text{F}}^2 + g(\mathbf{X}), \quad (6)$$

où la fonction $g(\mathbf{X})$ est la pénalité pour coder la contrainte de petit rang. Des choix classiques pour cette pénalisation sont la fonction indicatrice

$$g(\mathbf{X}) \equiv \delta_r(\mathbf{X}) = \begin{cases} 0 & \text{si } \text{rang } \mathbf{X} \leq r, \\ \infty & \text{sinon,} \end{cases} \quad (7)$$

si bien que (6) est équivalent au problème initial (2), ou encore sa relaxation convexe $g(\mathbf{X}) = \alpha \|\mathbf{X}\|_*$ où $\|\mathbf{X}\|_*$ est la norme nucléaire et $\alpha > 0$ le coefficient de pénalisation. L'algorithme du gradient proximal [2, ch. 10] est alors une méthode appropriée pour s'affranchir de la non-différentiabilité de la fonction $g(\mathbf{X})$, comme c'est le cas dans les deux exemples précédents. Néanmoins calculer l'opérateur proximal sur l'espace des matrices Hankel revient à calculer $\text{prox}_g(\mathbf{X}) = \arg \min_{\mathbf{H} \in \mathcal{H}} \frac{1}{2} \|\mathbf{H} - \mathbf{X}\|_{\text{F}}^2 + g(\mathbf{H})$ ce qui s'avère aussi compliqué que de résoudre le problème initial. Des alternatives consistent à regarder des méthodes duales afin de prendre en compte les contraintes Hankel et petit rang [1, 3, 4], ou encore à paramétrer explicitement le noyau de la matrice \mathbf{X} afin d'imposer la contrainte sur le rang [8]. Malheureusement les variables primales ou duales introduites dans ces algorithmes ne sont en général ni de rang faible, ni structurée Hankel. Il n'est alors plus possible de calculer la solution à l'aide de calculs rapides, ni sans stocker de matrices pleines.

Approche proposée. L'approche proposée consiste à relaxer la contrainte sur la structure Hankel de la solution afin de résoudre un problème composite de la forme :

$$\arg \min_{\mathbf{X} \in \mathbb{C}^{m \times n}} f_\rho(\mathbf{X}) + g(\mathbf{X}), \quad \text{avec} \quad (8)$$

$$f_\rho(\mathbf{X}) = \frac{1}{2} \|\mathbf{W} \cdot (\mathbf{Y} - \mathcal{P}\mathbf{X})\|_{\text{F}}^2 + \frac{1}{2}\rho \|\mathcal{P}_\perp \mathbf{X}\|_{\text{F}}^2,$$

où $\rho > 0$ est le paramètre de pénalisation, et $\mathcal{P}_\perp \mathbf{X} \equiv \mathbf{X} - \mathcal{P}\mathbf{X}$ désigne le projeté sur le supplémentaire de \mathcal{H} . Ce dernier terme pénalise d'autant plus la composante non Hankel de la solution que ρ est élevé, le problème Hankel-SLRA (6) étant obtenu à la limite où $\rho \rightarrow +\infty$, $g(\mathbf{X})$ étant la pénalité sur le rang introduite précédemment. La fonction f_ρ est convexe (par combinaison conique de fonctions convexes), différentiable avec pour gradient :

$$\nabla f_\rho(\mathbf{X}) = \mathbf{W}^2 \cdot (\mathcal{P}\mathbf{X} - \mathbf{Y}) + \rho \mathcal{P}_\perp \mathbf{X}, \quad (9)$$

où $\mathbf{W}^2 = \mathbf{W} \cdot \mathbf{W}$, dont la régularité est précisée ci-après.

Proposition 1 (Régularité de f_ρ). *Pour tout $\rho \geq 1$, le gradient de f_ρ est ρ -Lipschitz.*

Démonstration. Par hypothèse, $\max_{ij} W_{ij} = 1$ si bien que $\|\mathbf{W}^2 \cdot \mathcal{P}(\mathbf{X} - \mathbf{X}')\|_{\text{F}}^2 \leq \|\mathcal{P}(\mathbf{X} - \mathbf{X}')\|_{\text{F}}^2$ et d'après (9) :

$$\begin{aligned} \|\nabla f_\rho(\mathbf{X}) - \nabla f_\rho(\mathbf{X}')\|_{\text{F}}^2 &\leq \|\mathcal{P}(\mathbf{X} - \mathbf{X}')\|_{\text{F}}^2 + \rho^2 \|\mathcal{P}_\perp(\mathbf{X} - \mathbf{X}')\|_{\text{F}}^2, \\ &\leq \rho^2 \|\mathbf{X} - \mathbf{X}'\|_{\text{F}}^2, \end{aligned}$$

où la seconde ligne vient de $1 \leq \rho^2$ puis du théorème de Pythagore. \square

Une itération de l'algorithme du gradient proximal pour minimiser le nouveau problème composite (8) consiste maintenant à calculer

$$\mathbf{X}_{k+1} = \text{prox}_{t_k g}(f_\rho(\mathbf{X}_k) - t_k \nabla f_\rho(\mathbf{X}_k)). \quad (10)$$

Les résultats de convergence, voir par exemple [2, Ch. 10.4], assurent que le pas t_k peut être choisi constant comme $t_k = \frac{1}{\rho}$ d'après la proposition 1. L'argument de l'opérateur proximal devient

$$\mathbf{H}_k \equiv f_\rho(\mathbf{X}_k) - \frac{1}{\rho} \nabla f_\rho(\mathbf{X}_k) = \mathcal{P}\mathbf{X}_k - \frac{1}{\rho} \mathbf{W}^2 \cdot (\mathcal{P}\mathbf{X}_k - \mathbf{Y}),$$

et donc $\mathbf{H}_k \in \mathcal{H}$ par combinaison linéaire de matrices Hankel.

La fonction g est enfin choisie pour garantir que la solution du problème composite est de petit rang. On suppose par la suite que g est la fonction indicatrice définie en (7), qui est particulièrement adaptée lorsque le rang de la solution recherchée est connu. Notons cependant que cette pénalité n'est pas convexe et l'obtention d'un minimum local par un algorithme proximal n'est plus acquise³, même si en pratique l'algorithme proposé semble bien converger. Une alternative est d'utiliser la norme nucléaire : cette relaxation convexe garantit la convergence vers le minimum global qui est, en général, de petit rang.

Le théorème d'Eckart-Young stipule qu'une valeur de l'opérateur proximal est obtenue par SVD tronquée :

$$\mathbf{X}_{k+1} = \text{prox}_{\frac{1}{\rho} g} \mathbf{H}_k = \text{SVD}_r(\mathbf{H}_k).$$

Comme expliqué dans la partie 3, cette SVD peut être calculée efficacement car $\mathbf{H}_k \in \mathcal{H}$. Réciproquement, \mathbf{H}_k se déduit de l'approximation Hankel $\mathcal{P}\mathbf{X}_k$ qui peut aussi être calculée efficacement, \mathbf{X}_k étant par construction de petit rang. Pour un $\rho \geq 1$ fixé, on obtient donc un algorithme qui s'implémente de manière efficace.

Le problème (8) est d'autant plus facile que l'on relaxe la contrainte Hankel. Ainsi une petite valeur de ρ permet d'effectuer un plus grand pas $t_k = 1/\rho$ dans la descente de gradient de chaque itération (10). L'approche proposée consiste à partir du problème le plus simple lorsque $\rho = 1$, puis à augmenter progressivement la pénalité ρ . Ainsi si $a > 1$ est un facteur d'accroissement et $\rho_{\max} > 1$ une valeur limite liée à la tolérance sur la solution, on peut construire une suite de coefficients pour $i = 0, 1, \dots$

$$\rho_{i+1} = \min(\rho_{\max}, a\rho_i),$$

en partant de $\rho_0 = 1$, et résoudre successivement, pour chacun des ρ_i , le problème (8) en itérant selon (10) jusqu'à un critère d'arrêt comme un nombre maximal $M \in \mathbb{N}^*$ d'itérations. L'algorithme du gradient proximal (10) peut ainsi être redémarré à chaud en repartant de la solution précédente, et converger plus rapidement vers la solution. Ce dernier peut enfin être accéléré en utilisant la technique d'extrapolation de Nesterov [9, 2] à chaque coefficient ρ_i fixé. Tout ceci permet de définir l'Algorithme 1 pour résoudre le problème Hankel-SLRA (6)

³Il en est de même pour les méthodes de la littérature qui utilisent cette pénalité, par exemple [1] ou [3].

en construisant une séquence qui converge vers une solution Hankel du problème. Cet algorithme repose uniquement sur les calculs efficaces présentés dans la partie 3 : une valeur de l'opérateur proximal se déduit directement de la SVD tronquée d'une matrice Hankel, et les matrices dont on calcule les approximations Hankel sont par construction de petit rang. En particulier il n'est jamais nécessaire de stocker ni de manipuler des matrices pleines de $\mathbb{C}^{m \times n}$.

Algorithme 1 : SLRA par pénalisation Hankel

entrées : données \mathbf{y} et poids $\mathbf{w} \in \mathbb{C}^N$, dimensions (m, n)

paramètres : facteur $a > 1$, pénalité maximale $\rho_{\max} > 1$, nombre d'itérations $M \in \mathbb{N}^*$

initialisation : $\mathbf{X}_0 = \mathbf{0}$, $\rho_0 = 1$

itérations principales : pour $i = 0, 1, \dots$

i. redémarrage à chaud : $\mathbf{H}_0 = \mathbf{X}_i$, $\mathbf{Z}_0 = \mathbf{H}_0$, $t_0 = 1$

ii. itérations du problème (8) : pour $k = 0, \dots, M - 1$

1. $\mathbf{Z}_{k+1} = \text{prox}_{(1/\rho_i)g} \left(\mathbf{H}_k - \frac{1}{\rho_i} \mathbf{W}^{-2} \cdot (\mathbf{H}_k - \mathbf{Y}) \right)$,
2. $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$
3. $\mathbf{H}_{k+1} = \mathcal{P} \mathbf{Z}_{k+1} + \frac{t_k - 1}{t_{k+1}} (\mathcal{P} \mathbf{Z}_{k+1} - \mathcal{P} \mathbf{Z}_k)$.

iii. mise à jour : $\mathbf{X}_{i+1} = \mathbf{H}_M$, $\rho_{i+1} = \min(\rho_{\max}, a\rho_i)$

5 Simulations

On reprend ici le même scénario que celui proposé dans [1] où le signal $\mathbf{x} = (x_1, \dots, x_N)$ est la somme de $r = 4$ exponentielles complexes :

$$x_k = \sum_{i=1}^r c_i \exp(2\pi\nu_i t_k), \quad \text{pour } k = 1, \dots, N, \quad (11)$$

avec $t_k = -\frac{1}{2} + \frac{k-1}{N}$ et les paramètres suivants :

$$\begin{array}{c|c|c|c|c} c_i & e^{0.6i\pi} & \frac{2}{5}e^{1.12i\pi} & \frac{3}{2}e^{0.86i\pi} & \frac{7}{10}e^{1.87i\pi} \\ \hline \nu_i & 0.2+1.86i & -0.28+6.59i & 0.04+7.49i & -0.23+19.84i \end{array}.$$

Par construction la matrice Hankel $\mathbf{X} = \mathcal{H}\mathbf{x}$ est de rang $r = 4$. On observe le signal $\mathbf{y} = \mathbf{x} + \epsilon$ où les bruits ϵ_k sont gaussiens i.i.d., centrés, complexes circulaires, de variance σ^2 , et $\eta^2 = \frac{\|\mathbf{x}\|_F^2}{N\sigma^2}$ désigne le rapport signal sur bruit (RSB). Pour les méthodes SLRA, la matrice de poids est celle définie dans la partie 2 comme $\mathbf{W} = \mathcal{H}\boldsymbol{\kappa}^{1/2}$. Dans toutes les simulations, les paramètres de l'Algorithme 1 sont $a = 2$, $M = 30$, où $\rho_{\max} = 10^{-6}$ est la tolérance sur l'écart minimum entre les dernières meilleures solutions pour arrêter les itérations.

La Figure 1 présente les performances d'estimation pour les solutions SLRA calculées par ADMM [1] et par l'Algorithme 1 proposé, ainsi que pour la méthode de Cadzow. On peut constater que l'Algorithme 1 et ADMM ont bien des performances équivalentes et sont *statistiquement efficaces* pour les RSB élevés, la borne de Cramér-Rao étant alors atteinte, contrairement à la méthode de Cadzow.

Finalement les temps de calcul des algorithmes précédents sont comparés dans la Table 1 pour différentes tailles N du signal \mathbf{x} . La matrice Hankel $\mathbf{X} \in \mathbb{C}^{m \times n}$ est choisie (approximativement) carrée, ce qui est standard dans la littérature [5].

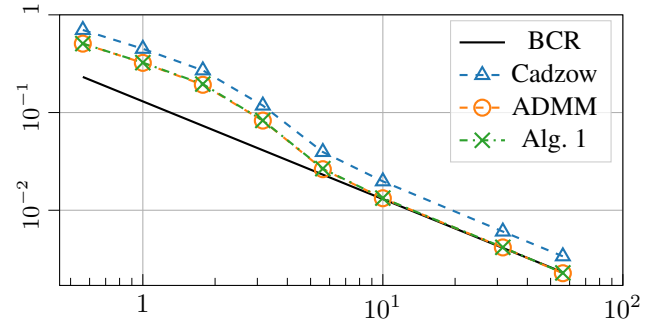


FIGURE 1 : Erreurs quadratiques moyennes $\|\mathbf{x} - \hat{\mathbf{x}}\|_F^2$ (1000 tirages de Monte-Carlo) vs RSB η^2 pour l'approximation Hankel en $r = 4$ composantes de la matrice encodée par le signal (11). Dimensions : $N = 256$, $m = 128$, et $n = 129$. Trait continu noir : borne de Cramér-Rao (BCR).

Tous les algorithmes sont codés en Python et exécutés sur un PC qui dispose de 6 CPU Intel cadencées à 3.70GHz et 32Go de RAM. Les algorithmes sont arrêtés en deçà d'une tolérance de 10^{-6} vers leurs solutions locales respectives (précalculées sur un grand nombre d'itérations). On constate que les méthodes *efficaces*, Cadzow et Alg. 1, ont un temps de calcul quasi linéaire⁴ en N . L'Algorithme 1 proposé permet ainsi d'obtenir une solution du problème Hankel-SLRA pour une matrice de dimension 32768×32769 en moins d'une minute. A l'inverse, ADMM a un coût générique en $O(N^3)$ qui devient vite rédhibitoire.

TABLE 1 : Temps d'exécution (s) pour différentes longueurs N du signal (11) et un RSB $\eta^2 = 1$ constant. Dimension des matrices Hankel : $m = N/2$, $n = N/2 + 1$.

N	2^8	2^{10}	2^{12}	2^{14}	2^{16}
Cadzow	0.04	0.08	0.22	0.54	2.97
Alg. 1	1.04	1.49	4.03	12.05	42.45
ADMM	3.46	141	2560	–	–

Références

- [1] Fredrik ANDERSSON, Marcus CARLSSON, Jean-Yves TOURNERET et Herwig WENDT : Frequency estimation based on Hankel matrices and the alternating direction method of multipliers. *In EUSIPCO 2013 - 21st European Signal Processing Conference*, Marrakech, Morocco, Sept. 2013.
- [2] Amir BECK : *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.
- [3] Laurent CONDAT et Akira HIRABAYASHI : Super-résolution d'impulsions positives par approximation Toeplitz de rang faible. *In GRETSI 2015 - XXVème Colloque francophone de traitement du signal et des images*, Lyon, France, septembre 2015.
- [4] Maryam FAZEL, Ting Kei PONG, Defeng SUN et Paul TSENG : Hankel matrix rank minimization with applications to system identification and realization. *SIAM Journal on Matrix Analysis and Applications*, 34(3):946–977, 2013.
- [5] Jonathan GILLARD et Konstantin USEVICH : Hankel low-rank approximation and completion in time series analysis and forecasting : a brief review. *Statistics and Its Interface*, A paraître.
- [6] Anton KOROBENIKOV : Computation and space-efficient implementation of SSA. *Statistics and Its Interface*, 3(3):357–368, 2010.
- [7] R. B. LEHOUCQ, D. C. SORENSEN et C. YANG : *ARPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1998.
- [8] Ivan MARKOVSKY et Konstantin USEVICH : Software for weighted structured low-rank approximation. *Journal of Computational and Applied Mathematics*, 256:278–292, 2014.
- [9] Yurii NESTEROV : A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269: 543–547, 1983.

⁴Pour les 'petites' valeurs de N le multithreading des fonctions de numpy n'est pas amorti, ce qui peut expliquer l'accroissement sublinéaire.