

HAMMOUDA Ilyes
JOUINI Mohamed-Aziz
KOUHANA Ava

ENSAE 2^{ème} année
Projet statistique appliquée



Fast image recovery under noisy compressed sensing

ENSAE-CREST

Encadrants : Alexandre TSYBAKOV
Mohamed NDAOUD

Table des matières

1	Remerciements	3
2	Introduction	4
3	Présentation théorique des algorithmes et résultats sur les données simulées	5
3.1	L'algorithme IHT	5
3.1.1	Présentation générale de l'algorithme IHT	5
3.1.2	Procédure améliorée de l'algorithme IHT	5
3.2	L'algorithme ISTA	7
3.3	Les résultats sur les données simulées	10
4	Décomposition en ondelettes	12
4.1	Présentation théorique	12
4.2	Approche suivie	15
4.2.1	Système de Haar	15
4.2.2	Résultats pratiques	16
5	Application sur les images	18
6	Pistes d'amélioration	19
7	Conclusion	20
A	Figures	22
A.1	Données simulées	22
A.1.1	Algorithmes de la librairie Pylops	22
A.1.2	Algorithmes implémentés par nous mêmes	23
A.2	Décomposition en ondelettes	28
B	Résultats sur les images	29
C	Preuves	33

1 Remerciements

Nous tenons à exprimer notre profonde gratitude envers le Professeur Alexandre Tsybakov et Monsieur Mohamed Ndaoud. Monsieur Tsybakov nous a prodigué son aide à travers une explication détaillée sur les enjeux du projet ainsi que sur les méthodes d'estimation que nous allions devoir employer. Mr Ndaoud nous a suivis tout au long du projet d'une façon quasi hebdomadaire. Leurs conseils avisés, leur expertise et leur disponibilité ont grandement contribué à l'avancement des travaux et résultats obtenus dans ce projet.

Nous tenons également à remercier les enseignants que nous avons suivis au cours de notre scolarité à l'ENSAE et nous ont ainsi permis d'avoir en notre possession le bagage mathématique nécessaire pour aborder ce travail et résoudre les problématiques scientifiques rencontrées tout au long de ce projet de long terme.

2 Introduction

Le but de ce projet est de mettre en œuvre une nouvelle procédure, caractérisée par sa vitesse de convergence qui permet de résoudre des problèmes de régression en grande dimension. Il a déjà été montré que l'algorithme Iterative Hard Thresholding (IHT) est parmi les algorithmes les plus rapides répondants à cette problématique. L'article [8] présente plus en détails des résultats comparatifs de l'algorithme IHT dans le cadre de modèles de statistique en grande dimension.

La procédure qu'on essayera d'explorer dans ce rapport trouve son utilité de plus en plus dans le contexte actuel. En effet, des bases de données avec un grand nombre de caractéristiques sont encore plus utilisées aujourd'hui. La représentation et l'analyse de telles bases de données demandent des techniques de statistique en grande dimension. Afin de rendre le problème d'estimation en grande dimension bien défini, des hypothèses supplémentaires sur le signal sous-jacent sont nécessaires. Ainsi, dans ce travail on supposera que le vecteur à estimer est sparse. En pratique une décomposition en ondelettes de notre signal peut être utilisée pour satisfaire cette hypothèse.

La partie algorithmique de ce projet s'est basée sur les résultats théoriques de l'article [9]. Une limite de la version standard de l'algorithme IHT, est que l'implémentation de ce dernier nécessite de connaître la vraie valeur du paramètre s , qui représente la sparsité d'un vecteur donné.¹ Or dans [9] on présente une nouvelle procédure qui permet d'abandonner cette contrainte de connaître s . L'algorithme développé dans ce projet, permettra ainsi d'atteindre des résultats statistiquement optimaux plus rapidement que les algorithmes classiques de LASSO. Il est connu que la régression par LASSO produit un biais qui est inévitable, c'est aussi le cas de certains estimateurs convexes par exemple l'estimateur SLOPE. Or l'algorithme IHT permet d'enlever ce biais dans certains cas [9]. Par ailleurs, la version implémentée dans ce projet a l'avantage d'être plus adaptative que les autres versions de l'algorithme IHT, comme elle ne nécessite pas de connaître à l'avance s . Ainsi, Ce projet essaie de répondre à la question suivante :

La version adaptative de l'algorithme IHT proposé dans [9] permet-elle, appliquée sur les photos, d'avoir une nouvelle procédure, plus rapide, de décompression des images ?

Dans ce rapport, on présentera dans un premier lieu les résultats théoriques ainsi que la nouvelle procédure pour l'implémentation de l'algorithme IHT ; on présentera aussi l'algorithme ISTA, qui nous a servi de base de comparaison. Cette comparaison a été effectuée dans un premier temps sur les données simulées. Dans un deuxième temps, on présentera la procédure de l'analyse en ondelette qu'on a utilisée afin de transformer les photos en vecteurs sparses. Puis on s'intéressera aux résultats obtenus lors de l'application de ces procédures sur les images. Finalement on exposera quelques pistes d'amélioration de l'application de la procédure IHT pour la décompression des images.

1. Dans cet article le paramètre s représente le nombre de composants non nuls d'un vecteur donné

3 Présentation théorique des algorithmes et résultats sur les données simulées

3.1 L'algorithme IHT

3.1.1 Présentation générale de l'algorithme IHT

D'abord on se place dans le cadre du modèle suivant :

$$Y = X\beta + \sigma\xi, \quad (1)$$

où Y représente l'observable, le vecteur signal β un signal s -sparse, $\|\beta\|_0 \leq s$ et ξ un vecteur centré de bruit. On note $\|\beta_0\|$ le nombre de coordonnées non nuls de β .

On pose l'algorithme suivant :

$$\hat{\beta}_0 = 0, \forall n \geq 0, \hat{\beta}_{n+1} = H_{\lambda_{n+1}} \left(\hat{\beta}_n + \frac{1}{n+1} X^T (Y - X\hat{\beta}_n) \right), \quad (2)$$

où $(\lambda_n)_n$ est une suite positive, et H_λ est l'opérateur de Hard Thresholding défini comme suit : $H_\lambda : R^p \rightarrow R^p$

$$\forall u \in R^p, \forall j = 1, \dots, p, \quad H_\lambda(u)_j = u_j \mathbf{1}\{|u_j| \geq \lambda\}. \quad (3)$$

Dans [4] on montre que si $\|\beta\|_0 \leq s$ cet algorithme converge vers un minimum local du problème d'optimisation suivant :

$$\min_{\beta} \frac{1}{2n} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_0^2 \quad (4)$$

Cette procédure peut être interprétée comme une descente de gradient projetée sur un ensemble non convexe.

3.1.2 Procédure améliorée de l'algorithme IHT

On se place désormais dans le cadre du modèle suivant :

$$Y = X\beta + \sigma\xi \quad (5)$$

où $X \in R^{n \times p}$ (dans l'implémentation de l'algorithme dans ce projet on considère que X est une matrice dont tous les éléments sont iid $\mathcal{N}(0, 1) \sim (0, 1)$), le bruit ξ est un vecteur centré 1-sous-gaussienne³ indépendant de X . On note P_β la distribution de (Y, X) dans le modèle (5) et E_β l'espérance correspondante et P_β est

2. On fait le choix du critère de régularisation avec la norme l_0 car presque toutes les images ont une représentation sparse dans la base des ondelettes[2].

3. $\forall \lambda \in R^n, E(e^{\langle \lambda, \xi \rangle}) \leq e^{\frac{\|\lambda\|_1^2}{2}}$

toujours s -sparse. Dans [9] on adopte une procédure de seuillage qui interpole, dans un sens, deux seuils classiques à savoir le plus grand s dans l'algorithme IHT et $\sigma\sqrt{\frac{2\log(p)}{n}}$ dans la procédure de régression linéaire par Lasso.

Dans le cadre de cette procédure, on posera des hypothèses supplémentaires qui permettent de restreindre la convexité forte et la régularité sur l'ensemble des vecteurs sparses. On commence d'abord par définir : $|S|$ le cardinal d'un ensemble S . M_S la matrice constituée des colonnes M_i où $i \in S$. Finalement pour une matrice semi-définie positive donnée, A , on note $\lambda_{\max}(A)$ (resp $\lambda_{\min}(A)$), la plus grande valeur propre (resp la plus petite valeur propre). Soit $s \in 1, \dots, p$, On pose L_s et $m_s > 0$ tel que :

$$\begin{aligned} L_s &= \max_{|S|=s} \lambda_{\max}(X_S^T X_S) \\ m_s &= \min_{|S|=s} \lambda_{\min}(X_S^T X_S) \\ \delta_s &= \left| 1 - \frac{L_s}{m_s} \right| \end{aligned}$$

On supposera par la suite dans ce travail que $s \leq \frac{p}{3}$ et que X satisfait l'hypothèse suivante : $\delta_{3s} \leq \frac{\delta}{2}$ pour un $0 \leq \delta < 1$.⁴

La procédure introduite dans [9] garantie, implicitement, à chaque itération la sparsité de l'estimateur sans avoir à choisir exactement s composantes. Afin de garantir un résultat statistiquement optimal et une convergence rapide, on définit la suite λ_m de la manière suivante :

$$\lambda_0 \in R, \forall m \in N, \lambda_m = \max(\kappa^{m/2} \lambda_0, \lambda_\infty), \text{ où } \kappa \text{ est une constante fixée.} \quad (6)$$

Il reste ainsi que λ_0 et λ_∞ à définir. En ce qui concerne le premier paramètre λ_0 , le choix de celui-ci n'a pas vraiment d'importance, on peut le choisir aussi grand qu'on veut, on aboutira au même résultat, moyennant plus d'itérations. Pour λ_∞ , qu'on pourrait interpréter comme étant une règle de sortie de l'algorithme, pratiquement on utilise la cross validation pour le déterminer. Une valeur théorique de ce dernier peut être donnée par, $\lambda_\infty = \sqrt{\frac{2\sigma^2 \log(ep/s)}{\|X\|_{2,\infty}^2}} 5$.

Ainsi, on peut définir l'algorithme IHT adaptatif comme suit :

On fait l'initialisation par $\hat{\beta}_0$ qui peut être n'importe quel estimateur au plus $2s$ -sparse, en particulier le vecteur nul. Par ailleurs dans [9] on propose $\hat{\beta}_0 = \hat{\beta}^{\tilde{m}}$ où

$$\tilde{m} = \operatorname{argmin}_{m \in [\hat{T}]} \left[\frac{1}{n} \|Y - X\hat{\beta}^m\|^2 + \frac{10\hat{\sigma}_m^2 |\hat{\beta}|_0 \log(ep/|\hat{\beta}|_0)}{n} \right]$$

où $\hat{\sigma}_m^2 = \frac{\|Y - X\hat{\beta}\|^2}{n}$. Pour $\hat{\sigma}_n$ on peut prendre n'importe quel estimateur consistant de σ .

$\hat{T} = \inf \left[m \geq 0, \lambda_m \leq \frac{4\hat{\sigma}_{\tilde{m}}}{\|X\|_{2,\infty}} \right]$, avec $\tilde{m} = \inf \left[m, \frac{m}{\lambda_m} \leq \frac{\hat{\sigma}_m}{\|X\|_{2,\infty}} \sqrt{160 \log(ep)} + 1 \right]$. On considère les m dans cet intervalle car a pour un niveau de seuillage λ_m au dessus de $\frac{\sigma}{\|X\|_{\infty,2}}$ on ne peut plus garantir que le résultat sera sparse.

4. On dira alors que X satisfait $RIP(3s, \frac{\delta}{2})$ plus généralement on dit que X satisfait $RIP(s, c)$ si $\delta_s \leq c$

5. La norme $\|X\|_{2,\infty}$ est définie par $\|X\|_{2,\infty} := \max \|X_j\|_2$ où X_j est la $j^{\text{ème}}$ colonne de X et $\|\cdot\|$ est la distance euclidienne

Finalement, on peut définir :

$$\forall n \geq 1, \tilde{\beta}_{n+1} = H_{\lambda_{n+1}} \left(\tilde{\beta}_n + \frac{1}{\|X\|_{2,\infty}^2} X^T (Y - X \tilde{\beta}_n) \right) \quad (7)$$

$$\lambda_m = \max(\kappa^{m/2} \lambda_0, \frac{\hat{\sigma}_m}{\|X\|_{2,\infty}} \sqrt{160 \log(ep)}), m = 1, \dots$$

Ainsi, on peut annoncer le théorème suivant :

Théorème 1 Soit $\epsilon \in [0, 1]$ et $\lambda_\infty^\epsilon := (1 + \sqrt{\epsilon})^{\frac{\sigma \sqrt{2 \log(ep/s)}}{\|X\|_{2,\infty}}}$. Si $\delta \leq \max(\epsilon, 1/400^2)$, $\lambda \geq \lambda_\infty^\epsilon$ et $\tilde{\beta}^m$ donné par (7) alors :

$$\forall m \geq 0; \limsup_{s/p \rightarrow 0 | \beta|_0 \leq s} P_\beta \left(\|\tilde{\beta}^m - \beta\| \geq (1 + 4\sqrt{\delta} + 100\delta^{m/2} 0(1)) \sqrt{s} \lambda \right) = 0 \quad (8)$$

Ce théorème est sharp dans le sens où si $\delta \rightarrow 0$ alors $\forall \epsilon \geq 0$ il existe un estimateur qui vérifie l'erreur minmax donnée par : $(1 + \epsilon)^{\frac{2\sigma^2 \log(ep/s)}{\|X\|_{2,\infty}^2}}$.

La démonstration de ce théorème sera donnée en annexe ainsi qu'une approximation théorique de λ_0 qui permettra de réduire le temps de calcul de cette procédure. Le tableau suivant présente un résumé comparatif des différents algorithmes.

Tableau 1 : Résumé des bornes minmax en régression en grande dimension où $a^* := \sigma \sqrt{\frac{2 \log(ep/s)}{n}}$ et où C_1, C_2, C_3, C_4 sont des constantes strictement positives. Les résultats minmax sont vrais pour $0 < \epsilon < 1$

3.2 L'algorithme ISTA

On se place maintenant dans le cadre d'un modèle identique à celui expliqué ci-dessus où l'on essaie de résoudre le problème d'optimisation suivant :

$$\min_{\beta} F(\beta) \text{ où } F(\beta) = \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1 \quad (9)$$

Le problème d'optimisation convexe ci-dessus peut être formulé comme un problème de programmation de cône de deuxième ordre et peut donc être résolu via des méthodes de point intérieur[3]. Cependant, dans la plupart des applications, en rapport avec le traitement de signal, le problème n'est pas seulement à grande échelle mais peut impliquer également des données de matrice dense, ce qui empêche souvent l'utilisation et l'avantage potentiel de méthodes sophistiquées de point intérieur[2].

Ainsi, les algorithmes de type ISTA et IHT trouvent tout leur intérêt. En effet ce type d'algorithme permet de converger vers la solution du problème posé ci-dessus par le biais d'itérations basées sur les produits scalaires et donc une puissance de calcul relativement simple. L'algorithme itératif shrinkage-thresholding algorithms (ISTA) est l'un des algorithmes les plus populaires de cette catégorie.

Plusieurs versions améliorées de l'algorithme ISTA ont été proposées dans la littérature notamment l'algorithme FISTA, qui est une version rapide de ce dernier. Cependant, dans ce projet nous avons décidé de ne pas s'intéresser à ces versions, même si au début du travail on a essayé de comparer les résultats de l'algorithme ISTA et FISTA. Étant donné qu'il n'y avait pas une grande différence sur la vitesse de convergence d'ISTA et FISTA et que de toute manière l'algorithme IHT converge plus rapidement que les deux autres, on a décidé de retenir uniquement l'algorithme ISTA pour faire la comparaison. D'une façon générale, l'algorithme ISTA est basé sur des itérations très simples, qui utilisent des produits scalaires faisant apparaître exclusivement X et X^T . Ainsi une itération de l'algorithme ISTA est donnée par [2] :

$$\hat{\beta}^0 = 0, \forall n \geq 0, \hat{\beta}^{n+1} = S_{\lambda t}(\hat{\beta}^n + 2t_n X^T(Y - X\hat{\beta}^n)). \quad (10)$$

Dans (17), la variable t_n représente le pas et S_λ représente l'opérateur de soft thresholding ie :

$$S_\lambda(x_i) = (|x_i| - \lambda)_+ \text{sgn}(x_i) = (x_i - \lambda \text{sgn}(x_i))_+.$$

Ainsi on peut écrire une étape de l'algorithme ISTA comme suit : $\hat{\beta}^{n+1} = S_{\lambda t}(G(\hat{\beta}^n))$ où $G(\cdot)$ représente une étape de la descente de gradient de l'ajustement de la fonction de coût de moindres carrés . Ainsi, on peut voir l'algorithme ISTA comme une généralisation des méthodes de descente de gradient classiques. Il s'inscrit alors dans le cadre des méthodes de premier ordre (FOM) ⁶. Dans le cadre de problèmes de régression en grande dimension, cette classe de méthodes est la plus pratique mais, généralement elles convergent lentement. Par exemple, dans le cadre de l'algorithme ISTA on montre dans [2] :

$$\text{pour un } k \text{ donné, } F(\hat{\beta}^k) - F(\hat{\beta}^*) \approx \mathcal{O}(\frac{1}{k})$$

Par ailleurs, dans [2] on propose de transformer le cadre (9) en un cadre plus général qui est le suivant :

$$\min_{\beta} (F(\beta) := f(\beta) + g(\beta), \beta \in R^n) \quad (11)$$

où $g : R^n \rightarrow R$: est une fonction continue et convexe, qu'on autorise à être non-lisse ⁷

, $f : R^n \rightarrow R$: est une fonction lisse ⁸, convexe, et à gradient Lipschitz continu

$\forall (x, y) \in R^{2n}, \|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|$, où $L(f)$ est une constante postive

On montre dans [2] que le problème défini ainsi admet une solution (c.a.d. $\text{argmin}_F \neq \emptyset$). Ainsi le problème (9) devient un cas particulier de (11), avec $f(\beta) = \|Y - X\beta\|$ et $g(\beta) = \lambda\|\beta\|_1$. Par ailleurs, la plus petite constante de Lipschitz dans ce cadre est donnée par $L(f) = 2\lambda_{(n)}(X^T X)$, où $\lambda_{(n)}(X^T X)$ représente la plus grande valeur propre de la matrice $X^T X$. Ainsi, on obtient un résultat sharp concernant la borne supérieure de l'algorithme ISTA donnée par le théorème suivant [2] :

6. Les méthodes du premier ordre (FOM) désignent de manière générale des méthodes itératives pour l'optimisation continue et (sous)différentiable qui utilisent principalement les informations du (sous)gradient de la fonction.

7. Une fonction est dite non lisse si elle n'est pas différentiable en un point de son domaine de définition.

8. indéfiniment différentiable

Théorème 2 Si on note $(\hat{\beta}^k)_k$ une suite générée par l'algorithme ISTA et $\hat{\beta}^*$ la solution théorique du problème (11) alors si on prend dans l'algorithme comme pas $t_k = \frac{1}{L(f)}$, on a :

$$\forall k \in N, F(\hat{\beta}^k) - F(\hat{\beta}^*) \leq \frac{L(f)\|\hat{\beta}^0 - \hat{\beta}^*\|^2}{2k} \quad (12)$$

Des résultats similaires ont été démontrés sur l'algorithme, qui est version rapide de l'algorithme FISTA. Un résultat similaire au théorème 2 peut être donné par [2] :

Théorème 3 Si on note $(\tilde{\beta}^k)_k$ une suite générée par l'algorithme FISTA et $\tilde{\beta}^*$ la solution théorique du problème (11) alors si on prend dans l'algorithme comme pas $t_k = \frac{1}{L(f)}$, on a :

$$\forall k \in N, F(\tilde{\beta}^k) - F(\tilde{\beta}^*) \leq \frac{2L(f)\|\tilde{\beta}^0 - \tilde{\beta}^*\|^2}{(k+1)} \quad (13)$$

3.3 Les résultats sur les données simulées

Afin de mener les premières analyses comparatives sur les deux algorithmes on commence par les tester sur des données simulées . Ainsi on se place dans le cadre du modèle théorique (1) avec les spécifications suivantes :

$$\begin{aligned} X_i &\sim \mathcal{N}(0, I_{p,p}), X \in R^{n \times p} \\ \beta_j &\sim \mathcal{B}(\mu), \beta \in R^{1 \times p}, \beta_j \text{ suit la loi de Bernoulli et où } \mu \approx \frac{s}{n} \\ \xi &\sim \mathcal{N}(0, I_{n,n}), \sigma \in R^{1 \times n} \end{aligned}$$

Ici la loi X_i est la i-ème colonne de X , $I_{p,p}$ est la matrice identité $p \times p$, β_j est la j-ème coordonnée de β et les β_j sont indépendants.

Dans le cadre de la représentation (1), β représente le vecteur sparse obtenu suite à la vectorisation d'une image avec les techniques d'ondelettes qu'on expliquera d'avantage dans la troisième section. X représente la matrice qui va servir pour permettre la compression de l'image et puis la décompression. En effet, dans le cadre d'une base d'images à très grande dimension, comme expliqué dans l'introduction, la démarche qu'on propose trouve son utilité dans ce cadre de base de données. Ainsi pour compresser ce genre de base de données, il suffirait de simuler une matrice normale, disons \tilde{X} et puis de calculer le produit matriciel suivant : $\tilde{X}\tilde{\beta}$, où $\tilde{\beta}$ représente le vecteur obtenu suite à la vectorisation d'une image donnée.

Cette partie nous a aussi été utile car elle nous a permis de vérifier que la validation croisée est suffisante pour calibrer les paramètres de nos algorithmes à savoir :

- le λ pour l'algorithme ISTA ainsi que le pas t ,
- le λ_∞ ⁹ pour l'algorithme IHT et d'analyser si la simplification faite sur le pas de l'algorithme IHT¹⁰.

Comme il a été mentionné dans notre rapport de mi-parcours que nous avons, dans un premier lieu, utilisé les algorithmes ISTA et FISTA préalablement définis dans la bibliothèque pylops, cependant comme l'illustrent bien les figures (1,2) les résultats obtenus ne correspondent pas à ceux mentionnés dans [2], ce qui suggère qu'il y a, potentiellement, des erreurs dans l'implémentation de ces algorithmes dans cette bibliothèque. Par ailleurs, on a pu remarquer que les algorithmes de la bibliothèque pylops sont optimisés, ce qui ne permet donc pas de comparer leur résultat à l'algorithme IHT que nous avons implémenté. Il a été donc décidé d'implémenter également l'algorithme ISTA, pour des raisons d'homogénéité des résultats. Par ailleurs, comme le montre la figure(4), nous obtenons des résultats similaires, pour l'algorithme ISTA, à celles obtenues dans [2].

9. Comme on l'a mentionné en haut une valeur théorique de ce paramètre peut être obtenu, cependant celle-ci suppose qu'on connaît la distribution de la loi de σ , or ceci n'est vrai que dans le cadre des données simulées. Ainsi, on utilise l'estimateur $\hat{\lambda}_\infty$, définit plus haut également, pour pouvoir définir l'ensemble sur lequel on effectue la procédure de validation croisée pour obtenir le paramètre qui minimise au plus $\|Y - X\hat{\beta}\|$

10. Comme dans l'article celle-ci a été prise par défaut égale à $\frac{1}{\|X\|_{2,\infty}}$

D'après [9], le λ_0 doit être, de préférable, de l'ordre de $\max(\frac{\|\beta\|}{\sqrt{s}}, \sigma \frac{\sqrt{\log(ep/s)}}{\|X\|_{2,\infty}})$. Par exemple on peut prendre :

$$\hat{\lambda}_0 = \max(\sqrt{\frac{\sum_{i=1}^s M_{(i)}^2}{s}}, \frac{\sigma}{\|X\|_{2,\infty}} \sqrt{40 \log(ep/s)}).$$

Alors dans ce cas on peut atteindre l'optimalité minimax au bout de m itérations seulement où :

$$\tilde{m} = \frac{\log(\max(2, \frac{s\lambda_0^2 \|X\|_{2,\infty}^2}{\sigma^2 s \log(ep/s)}))}{\log(1/\kappa)}.$$

La figure (3) illustre ce résultat. Par ailleurs, on calcule le coefficient de similarité cosinus de l'angle entre $(\beta$ et $\hat{\beta}^{\tilde{m}})$ et on obtient en moyenne :

$$S_C(\beta, \hat{\beta}^{\tilde{m}})^{11} = 0.99$$

La figure (9) ainsi que la figure (10), montrent le résultat qu'on a annoncé plus haut, selon lequel l'algorithme IHT garantit que le vecteur estimé est sparse à toutes les itérations.

11. S_C représente le coefficient de similarité cosinus tel que $S_C(A, B) = \frac{|A||B|}{\|A\|\|B\|}$, ce coefficient permet de calculer l'angle θ entre les vecteurs A et B . $S_C(A, B) \in [-1, 1]$ tel que si $S_C(A, B) = 1$ alors les vecteurs A et B sont colinéaires et donc similaires, $S_C(A, B) = 0$ alors les vecteurs A et B sont perpendiculaires et donc différents.

4 Décomposition en ondelettes

La pertinence des transformations en ondelettes (WT) pour l'analyse des images est bien établie. Une représentation en termes de contenu fréquentiel de régions locales sur une plage d'échelles fournit un cadre idéal pour l'analyse des caractéristiques d'une image, qui sont généralement de tailles différentes et peuvent souvent être caractérisées par leurs propriétés dans le domaine fréquentiel [5]. En effet les deux ancêtres de la transformation en ondelettes, à savoir les transformées dans le domaine fréquentiel('frequency domain transforms') et les modèles de représentation multirésolution, ont été largement utilisés dans le domaine de traitement des images. A titre d'exemple, les algorithmes de compression JPEG utilisent la transformation cosinus discrète.[14].

Par ailleurs, nous avons fait le choix d'utiliser ces techniques car elles permettent une réduction de la dimension inversible. En effet, une fois la réduction d'image a été effectuée, il reste possible de récupérer la version originale rapidement, c.a.d. les calculs ne sont pas très compliqués. Dans cette partie, on fera premièrement une présentation théorique de ces techniques d'analyse en ondelettes et notamment le processus d'inversion pour retrouver l'image initiale. Puis on présentera la démarche qu'on a effectuée dans notre projet et notamment les ondelettes de Haar que nous avons adoptées.

4.1 Présentation théorique

Le cas le plus simple dans la décomposition en ondelettes est de considérer une fonction à une dimension, disons $f : R \rightarrow R$, définie pour les nombres réels exclusivement. Il est vrai que cette fonction n'est pas trop représentative de notre problème de base, cependant comme pour les transformées de Fourier, il existe plusieurs variétés de la décomposition en ondelettes, qui dépendent de la dimension de l'espace initial du signal ainsi que de son domaine : que ce soit continu ou discret, fini ou infini. Une explication du cas complexe ainsi que le cas des signaux avec une dimension supérieure à un sera introduite par la suite. La décompensation en ondelettes consiste à trouver une représentation d'une fonction finie donnée, dans ce cas $f(x)$, en termes d'un ensemble de fonctions qui sont toutes des versions échelonnées et décalées d'une seule fonction donnée disons $\phi(x)$. Dans la littérature on utilise, des fonctions de base qui sont échelonnées par des puissances de 2, de sorte que la fonction $f(x)$ peut être donnée comme [15] :

$$f(x) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{mn} \phi_{mn}(x), \quad (14)$$

où les fonctions de base doivent vérifier la relation suivante :

$$\phi_{mn}(x) = 2^{-m/2} \phi(2^{-m}x - n). \quad (15)$$

Définie ainsi les coefficients de la décomposition en ondelettes sont l'ensemble $(c_{mn})_{mn}$, où n représente le position et m l'échelle. La différence fondamentale entre la décomposition en ondelettes et les séries de Fourier est que la première décrit les signaux en termes d'échelle plutôt que de fréquence. Ainsi l'utilité des ondelettes en traitement d'image découle du fait que celles-ci contiennent des caractéristiques à de nombreuses échelles.

Comme mentionné plus haut cette définition est limitée par rapport au problème qu'on essaye de résoudre dans cette partie : vectoriser une image¹² pour créer un vecteur sparse. Une façon simple d'aborder cette thématique consiste en l'utilisation de filtres de type : "quadrature mirror filters (QMF's)". Il s'agit de paires de filtres à une dimension qui divisent le signal en deux bandes de fréquences plus ou moins non superposées, à savoir une bande de fréquences passe-bas et une bande de fréquences passe-haut. Si (h_i) est la réponse impulsionnelle¹³ du filtre passe-bas, la réponse du filtre passe-haut correspondante est donnée par : [15] :

$$g_i = (-1)^i h_{1-i} \quad (16)$$

Par ailleurs, les bandes des fréquences passe-haut et passe-bas à une échelle m donnée sont :

$$c_i^m = \sum_j h_j c_{2i-j}^{m-1} \quad (17)$$

$$d_i^m = \sum_j g_j c_{2i-j}^{m-1} \quad (18)$$

Étant donné que le signal d'origine est égal à : $c_i^0 = f_i$

Le sous-échantillonnage par un facteur de 2 dans ces équations s'explique par le fait que le volume de données est préservé dans la transformation en ondelettes en effet : la moitié des coefficients de la transformation proviennent de la première bande passe-haut, un quart de la bande suivante (d_i^1) et ainsi de suite. Désormais, si on suppose que les filtres choisis forment une paire de type QMF alors une reconstitution parfaite de l'image est possible [7]¹⁴. Ainsi, en appliquant de manière répétée le même filtrage et la même décimation, le signal peut être décomposé en ensembles d'échantillons correspondant à des bandes de fréquences non chevauchantes avec une largeur de bande constante. En effet, comme les échantillons sont des transformés en ondelettes du signal, la reconstruction de celui-ci peut être obtenue par interpolation et sommation. En dimension deux la même procédure peut être suivie pour décompenser le signal en quatre ensembles à chaque étape : un ensemble de passe-bas et trois de passe-haut, basés sur des paires de filtres séparables dans les

12. Une image peut être défini comme une fonction à deux variables, $f(x, y)$, où x et y sont les coordonnées spatiales de l'image. L'amplitude de f à toute paire de coordonnées (x, y) est appelée intensité ou niveau de gris de l'image à ce point. Lorsque x , y et les valeurs d'amplitude de f sont toutes des quantités finies et discrètes, nous appelons l'image une image numérique ou digitale.

13. le signal de sortie obtenu quand l'entrée reçoit une variation soudaine et brève du signal [1]

14. Étant donné que cette procédure est parmi les plus anciennes qu'on a pu trouver dans la littérature, on expliquera dans la section suivante la procédure d'inversion et les ondelettes qu'on a choisis d'adopter

directions horizontale et verticale. On obtient alors [15]

$$c_{i,j}^m = \sum_k \sum_l h_k h_l c_{(2i-j)(2k-l)}^{m-1} \quad (19)$$

$$d_{i,j}^{m,2} = \sum_k \sum_l h_k g_l c_{(2i-j)(2k-l)}^{m-1} \quad (20)$$

$$d_{i,j}^{m,1} = \sum_k \sum_l g_k h_l c_{(2i-j)(2k-l)}^{m-1} \quad (21)$$

$$d_{i,j}^{m,2} = \sum_k \sum_l g_k g_l c_{(2i-j)(2k-l)}^{m-1} \quad (22)$$

Par ailleurs, une fonction donnée, disons $\psi(x)$ pour être admissible en tant que fonction d'ondelettes elle doit vérifier ces conditions : être centrée et localisée à la fois dans l'espace du temps et l'espace fréquentiel. [13]¹⁵.

Ainsi, une transformation d'ondelettes continue d'une séquence $(x_n)_{n \in [0, N-1]}$ et un espacement entre deux échantillons consécutifs du signal (x_n) dans le temps λ_t donnée par : [13]

$$W_n(s) = \sum_{i=0}^{N-1} x_i \bar{\psi}\left(\frac{(i-n)\lambda_t}{s}\right) \quad (16)$$

Pour faire une approximation de la transformation en ondelettes continue, la convolution ci-dessus doit être effectuée N fois. Cependant, afin d'accélérer la procédure, il est tout à fait préférable de faire les calculs dans un espace de Fourier. Ainsi le choix d'effectuer la convolution N fois est important (il est tout a fait possible de l'effectuer n fois ou $n < N$) car avec ce choix le théorème de convolution permet d'effectuer les N convolutions simultanément dans un espace de Fourier en utilisant la transformation de Fourier discrète (TFD) . La TFD de x_n est définie par [13] :

$$\hat{x}_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}; k \in [0, N-1].$$

Étant donné qu'en limite continue, la transformé de Fourier d'une fonction $\psi(t/s)$ est donné par $\hat{\psi}(s\omega)$, le théorème de convolution donne [13] :

$$W_n(s) = \sum_{n=0}^{N-1} \hat{x}_n \bar{\psi}(s\omega_n) e^{i\omega_k n \lambda_t}, \quad (23)$$

$$\omega_k = \frac{2\pi k}{N\lambda_t} k \leq \frac{N}{2},$$

15. Ces conditions sont nécessaires dans le cadre général de la décomposition en ondelettes d'un signal comme dans l'article on suppose qu'un signal qui peut être modélisé par une série temporelle

16. Comme ψ peut être une fonction complexe on prend son conjugué le cas échéant

$$\omega_k = \frac{-2\pi k}{N\lambda_t} k > \frac{N}{2}.$$

Par ailleurs, à chaque échelle s on normalise $\hat{\psi}$ tel que : $\hat{\psi}(s\omega_k) = (\frac{2\pi s}{\lambda_t})^{1/2} \hat{\psi}_0(s\omega_k)$ Comme mentionné plus haut, l'utilité de la décomposition en ondelettes est que cette dernière représente une réduction de dimension inversible, on peut alors reconstruire le signal x_n ainsi : [13]

$$x_n = \frac{\lambda_i \lambda_t^{1/2}}{C_\lambda \psi_0(0)} \sum_{j=0}^J \frac{Re(W_n(s_j))}{s_j^{1/2}}; s_i = s_0 2^{i\lambda_i}; J = \lambda_i^{(-1)} \log_2(N\lambda_t/s_0)$$

C_λ est une constante pour chaque ondelette et est égale à :

$$C_\lambda = \frac{\lambda_i \lambda_t^{1/2}}{\psi_0(0)} \sum_{j=0}^J \frac{Re(W_n(s_j))}{s_j^{1/2}}$$

4.2 Approche suivie

Le choix de l'ondelette dans la décomposition en ondelettes est arbitraire. Dans notre projet on a décidé d'adopter l'ensemble des fonctions introduites par Haar. En effet dans la littérature, ce système de fonctions est largement utilisé quand il s'agit de décomposition et analyse de signaux de type image.

4.2.1 Système de Haar

D'une façon générale, le système de Haar est un système de fonctions dans $L^p[0, 1]$, $p \in [0, \infty[$ qui prend des valeurs dans l'ensemble $\{0, 2^j; j \in N\}$. Ce système a la particularité que chaque fonction continue sur l'intervalle $[0, 1]$ peut être représentée par une série convergente en terme d'éléments de ce système. La fonction de Haar peut être définie de la manière suivante [12] :

$$haar(0, t) = 1, t \in [0, 1[; haar(1, t) = \begin{cases} 1 & \text{si } t \in [0, 1/2[\\ -1 & \text{si } t \in [1/2, 1[\end{cases}$$

$haar(k, 0) = \lim_{t \rightarrow 0+} haar(k, t)$, $haar(k, 1) = \lim_{t \rightarrow 1-} haar(k, t)$ aux points de discontinuité à l'intérieur de l'intervalle $]0, 1[, haar(k, t) = 1/2(haar(k, t-0) + haar(k, t+0))$ Par ailleurs la fonction d'ondelette de Haar peut être définie comme suit [12] :

$$\Psi : R \rightarrow R; \Psi(t) = \begin{cases} 1 & \text{si } t \in [0, 1/2[\\ -1 & \text{si } t \in [1/2, 1[\\ 0 & \text{sinon} \end{cases}$$

Étant donné que les fonctions de Haar ont la particularité que la i -ème peut être obtenue par la restriction des $(i-1)$ -ième autres fonctions à la moitié de l'intervalle où elle est différente de zéro, multiplié par $\sqrt{2}$ puis

la mettre à l'échelle sur l'intervalle $[0,1]$.¹⁷. Ainsi $\Psi_i^j(t) = \sqrt{2^j} \Psi(2^j t - i)$, $i \in [0, 2^j - 1]$, $j \in [0, \log_2 N - 1]$

On définit la matrice de Haar de façon itérative comme :

$$H(n) = \begin{bmatrix} H(n-1) & \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \\ 2^{(n-1)/2} I(n-1) & \otimes \begin{bmatrix} -1 & 1 \end{bmatrix} \end{bmatrix}, H(0) = 1.$$

Où $I(n-1)$ est la matrice identité de dimension $(n-1)$.

On a clairement $H(n) \neq H(n)^T \forall n > 1$. Par ailleurs $H(n)^{-1} = 2^{-n} H(n)^T$, l'opérateur \otimes représente le produit de Kronecker.

Soit F la représentation matricielle d'une image, la dimension de F est $N \times N = 2^n \times 2^n$ alors la transformation de Haar discrète est donné par :

$$S = \frac{1}{N^2} \cdot H(n) \cdot F \cdot H(n)^T \quad (24)$$

La transformation inverse est donnée par :

$$F = \frac{1}{N^2} \cdot H(n) \cdot S \cdot H(n)^T \quad (25)$$

Ainsi, on peut transformer l'écriture (24) et (25) en :

$$s(k, m) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \times haar(k, x) \times haar(m, y), \quad (26)$$

$$f(k, m) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} s(k, m) \times haar(k, x) \times haar(m, y). \quad (27)$$

D'où la matrice $S_{k,m} = s(k, m)$ et $F_{x,y} = f(x, y)$. Ainsi, la transformation de Haar est largement utilisée dans le domaine de la décomposition des images car elle ne nécessite pas des calculs compliqués.

4.2.2 Résultats pratiques

Dans un premier lieu on a utilisé une représentation matricielle de la photo en la décomposant sur trois couches rouge vert et bleu. On a ensuite effectué l'encodage avec les techniques des ondelettes et ensuite le décodage. Cependant, comme le montre la figure (12), cette démarche a du mal à restituer les couleurs originales au niveau du décodage. On a donc converti notre image du système RVB au système $YC_B C_R$ qui permet de mieux séparer la lumière de la chrominance par rapport à la représentation RVB [10]. Ainsi, l'image est décomposée en trois canaux : Y représente la composante luminosité d'une image donnée, C_B : contient de

17. Cette propriété est vrai pour toutes les fonctions de Haar sauf haar(0,t)

l'information sur la différence de la chrominance bleue par rapport au canal Y et finalement C_R :contient de l'information sur la différence de la chrominance rouge par rapport au canal Y. La figure (13) montre cette représentation pour une photo donnée.

Ainsi, une fois qu'on a représenté l'image dans ce repère $YC_B C_R$, la partie compression se fait de la manière suivante : premièrement on crée trois matrices dont chacune contient l'information de chaque canal(Y C_B et C_R). Puis, on effectue une décomposition en ondelettes pour chaque matrice. Ensuite, on transforme chaque matrice en un vecteur, tout en stockant les dimensions de base de chaque matrice pour finalement concaténer les trois vecteurs afin d'obtenir un seul vecteur sparse. La décompression se fait de la manière suivante : premièrement on sépare le vecteur sparse en trois vecteurs, ensuite on reconstitue les trois matrices obtenues dans la démarche de compression, puis on effectue la décomposition en ondelettes inverse qui retourne trois matrices qui représentent une approximation des canaux $YC_B C_R$ et finalement on superpose la représentation de ces trois matrices dans le repère $YC_B C_R$ pour obtenir l'image décompressée. La figure (14) représente le résultat de cette démarche. Pour analyser l'efficacité de ces deux démarches, on a utilisé deux indicateurs :

$$S(M) = \frac{\text{Nombre de composants non nuls du vecteur M}}{\text{Taille du vecteur M}}$$

$$Proxy(M) = \frac{\|M\|_1}{\|M\|_2 \times (\text{Taille du vecteur M})}$$

5 Application sur les images

Une fois qu'on a implémenté les différents algorithmes et l'analyse en ondelettes, on va finalement regrouper tous les travaux afin d'arriver à la finalité du projet, qui consiste à répondre à la question posée dans l'introduction du rapport. Ainsi, une fois qu'on a obtenu le vrai β suite à la décomposition en ondelettes en premier temps, on applique la compression $Y = X\beta$ où X est une matrice de normale standard de taille (n, p) , avec $n \ll p$. À l'aide des différents algorithmes on estime $\hat{\beta}$ auquel on applique la décomposition en ondelettes inverse pour reconstruire l'image. Finalement on suppose que la décomposition en ondelettes introduit un bruit, c'est pour cela, qu'on n'additionne pas au produit matriciel défini ci-dessus un vecteur de bruit Gaussien.

L'ISTA et l'IHT classique ont fourni des résultats satisfaisants quant à leur application sur les images. En témoignent les figures [15], [16] et [18] de l'annexe. Pour autant, l'IHT adaptatif n'a pas fonctionné à la hauteur de nos espérances sur les photos. La figure [17] montre que notre implémentation de l'IHT adaptatif a donné lieu à une reconstitution de la photo insatisfaisante. La photo est floue, illisible. Nous l'expliquons par l'analyse ondelette à partir duquel nous obtenons certes un vecteur très sparse mais pour lequel les valeurs non nulles restantes sont très concentrées autour de 0. Ce qui contredit l'hypothèse, \mathcal{O} explicitée en annexe, sur le signal sous laquelle on obtient les résultats théoriques de l'IHT adaptatif. On a essayé de remédier à ce problème en introduisant, artificiellement du bruit. Cependant, le résultat n'a pas été extraordinaire, d'autant plus que ça demande d'avantage de puissance de calcul, ce qui n'est pas très efficace en vue du problème d'optimisation qu'on essaye de résoudre.

Par ailleurs nous pouvons modifier cet algorithme adaptatif pour obtenir des résultats satisfaisants en remplaçant le Hard threshold par du soft threshold. On note que dans ce cas de figure, l'algorithme devient d'avantage une version accélérée et adaptative de l'ISTA qu'une version adaptative de l'IHT. La comparaison des résultats obtenus lors de la décompression des images grâce aux algorithmes ISTA et IHT classique obtenus dans la figure 18 montre que l'algorithme ISTA est légèrement plus performant. L'image obtenue avec l'algorithme ISTA est plus nette, plus contrastée qu'avec l'algorithme IHT classique. Cependant la figure [20] montre que l'algorithme IHT converge plus rapidement.

Enfin, durant l'application sur les images, nous nous sommes rendus compte que l'algorithme adaptatif pouvait fournir des résultats satisfaisants mais ce, à la condition que l'on effectue une compression d'image négligeable (par exemple passer d'un vecteur représentatif d'une image de 1000 lignes à un vecteur en 900 lignes). Le résultat de cette démarche est donné par la figure [19]. Nous notons néanmoins que ce résultat ne rend pas l'algorithme IHT adaptatif compétitif vis-à-vis de l'IHT et de l'ISTA classiques.

6 Pistes d'amélioration

Une limite qu'on a pu identifier dans notre démarche est que celle-ci requiert une puissance de calcul importante, à cause de l'utilisation des techniques de décomposition en ondelettes, quand il s'agit de photo de très grande dimension. En effet, les techniques de décomposition en ondelettes permettent une réduction de dimension dans le sens où on récupère des éléments sparsés après leur utilisation. Cependant, elles ne permettent pas de réduire la taille de la matrice qui représente l'image, bien au contraire si on parcourt toute la base des ondelettes, c'est ce qu'on a décidé de faire afin que le théorème de convolution soit applicable on augmente légèrement la taille de la matrice. Cependant, les éléments ajoutés sont tous égaux à zéro. Ceci pose problème au niveau de la simulation de la matrice normale qui va servir pour effectuer la compression.

Une piste d'amélioration possible est d'utiliser des techniques comme la Transformée de Fourier Multirésolution (TFM), puisqu'elles donnent à l'utilisateur une plus grande flexibilité dans la conception de modèles et de procédures d'estimation, que les techniques d'analyse en ondelettes standard. [6]

Une autre technique qu'on pourrait utiliser pour améliorer notre démarche est de la combiner avec les techniques d'échantillonnage compressé (Compressed Sensing). En effet, La théorie de l'échantillonnage compressé offre un cadre pour la détection et la compression simultanées de signaux. Elle établit qu'une image représentable sous forme d'une matrice sparse peut être récupérée à partir d'un ensemble de mesures hautement incomplet [11].

Il est vrai que les algorithmes IHT et ISTA dans leurs versions classiques ou améliorées permettent de réduire le bruit présent sur une image donnée, cependant les résultats pratiques qu'on a obtenus montrent que cette réduction du bruit n'est pas très efficace. Dans ce sens, les techniques d'échantillonnage compressé, comme la démarche décrite dans l'article [11], permettent d'améliorer la qualité d'une image donnée.

7 Conclusion

En conclusion, notre implémentation de l'algorithme adaptatif développé par Mr Ndaoud présente des résultats satisfaisants avec les données simulées. En ce sens, nous obtenons des résultats similaires à l'IHT classique comme le montre la figure [8] . Pour autant, l'analyse en ondelettes, qui est la technique la plus utilisée pour la vectorisation d'images et la réduction de dimension inversible, s'est révélée peu concluante lorsqu'appliquée à l'algorithme adaptatif.

Deux pistes d'améliorations principales sont possibles dans le cadre d'une poursuite de ce projet :

La première étant de remplacer le seuil dur de l'algorithme dit hard thresholding par un seuil mou dit soft. En effet, étant donné que ce dernier est moins catégorique que le hard thresholding, il est tout à fait compréhensible que l'IHT adaptatif ne converge pas vu que les composants du vecteur obtenu par la décomposition en ondelette sont assez concentrés autour de zéro comme le montre la figure [22]. Nous avons exploré cette piste et cela a fourni des résultats plus que satisfaisants. Nous gardons néanmoins à l'esprit que cela revient à créer un ISTA accéléré et adaptatif d'avantage qu'un IHT accéléré et adaptatif.

La deuxième serait de remplacer la méthode des ondelettes par d'autres méthodes telles que les méthodes exposées dans la sixième partie de ce rapport afin de corriger le problème de sparsité. Cela pourrait permettre de passer d'une image à un vecteur sparse dont les valeurs des coefficients non nuls ne seraient pas trop concentrées autour du zéro, et réduire d'avantage les dimensions des vecteurs obtenus suite à la vectorisation d'une image afin de réduire la complexité algorithmique et celle de simulation.

Références

- [1] https://fr.wikipedia.org/wiki/Réponse_impulsionnelle.
- [2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1) :183–202, 2009.
- [3] Aharon Ben-Tal and Arkadi Nemirovski. Lectures on modern convex optimization - analysis, algorithms, and engineering applications. In MPS-SIAM series on optimization, 2001.
- [4] Thomas Blumensath and Mike E. Davies. Iterative thresholding for sparse approximations. Journal of Fourier Analysis and Applications, 14(5) :629–654, Dec 2008.
- [5] A. Calway. Image analysis using a generalised wavelet transform. In IEE Colloquium on Applications of Wavelet Transforms in Image Processing, pages 8/1–8/4, 1993.
- [6] A. Calway. Image analysis using a generalised wavelet transform. In IEE Colloquium on Applications of Wavelet Transforms in Image Processing, pages 8/1–8/4, 1993.
- [7] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. Communications on Pure and Applied Mathematics, 41(7) :909–996, 1988.
- [8] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014.
- [9] Mohamed Ndaoud. Scaled minimax optimality in high-dimensional linear regression : A non-convex algorithmic regularization approach, 2020.
- [10] C. Emmy Premal and S. S. Vinsley. Image processing based forest fire detection using ycbcr colour model. In 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], pages 1229–1237, 2014.
- [11] Wael Saafin, Miguel Vega, Rafael Molina, and Aggelos K. Katsaggelos. Image super-resolution from compressed sensing observations. In 2015 IEEE International Conference on Image Processing (ICIP), pages 4268–4272, 2015.
- [12] Radomir S. Stanković and Bogdan J. Falkowski. The haar wavelet transform : its status and achievements. Computers Electrical Engineering, 29(1) :25–44, 2003.
- [13] Christopher Torrence and Gilbert P. Compo. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society, 79(1) :61–78, January 1998.
- [14] Gregory K. Wallace. The jpeg still picture compression standard. Commun. ACM, 34(4) :30–44, apr 1991.
- [15] R. Wilson. Wavelets ? In IEE Colloquium on Applications of Wavelet Transforms in Image Processing, pages 1/1–1/6, 1993.

A Figures

A.1 Données simulées

A.1.1 Algorithmes de la librairie Pylops

valeur de la fonction de coût en fonction du nombre d'itérations avec l'algorithme ISTA pour alpha égale à 0.00202020202020202

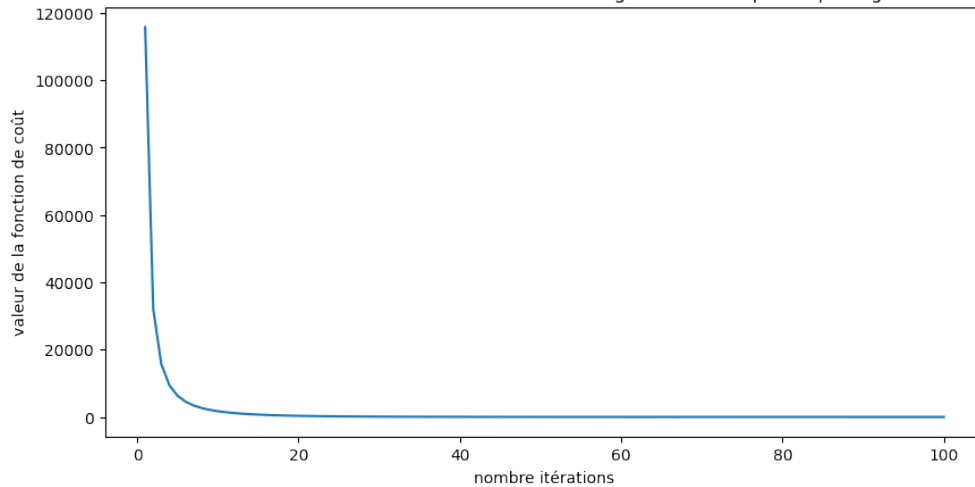


FIGURE 1 – La convergence de la fonction coût de l'algorithme ISTA implémenté dans la bibliothèque pylops

valeur de la fonction de coût en fonction du nombre d'itérations pour l'algorithme ISTA et FISTA pour alpha égale à 0.00202020202020202

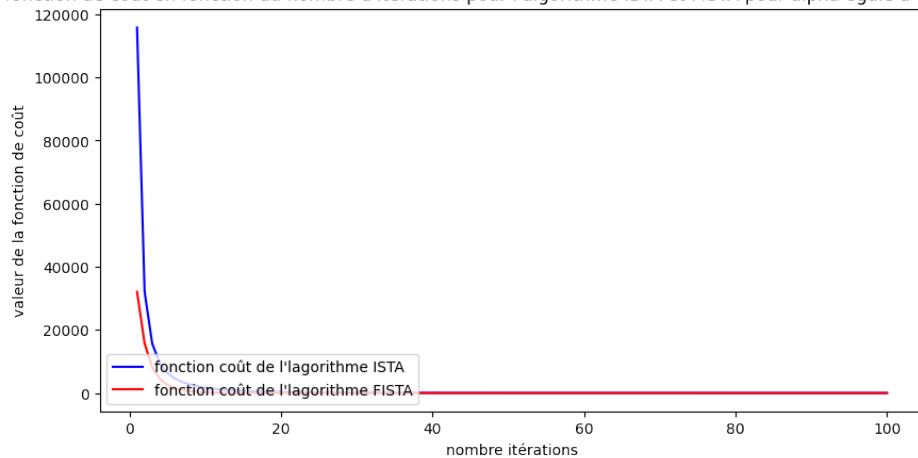


FIGURE 2 – comparaison entre la vitesse de convergence des algorithmes ISTA et FISTA implémentés dans la bibliothèque pylops

A.1.2 Algorithmes implémentés par nous mêmes

valeur de la fonction de coût en fonction du nombre d'itérations avec l'algorithme IHT après cross validation

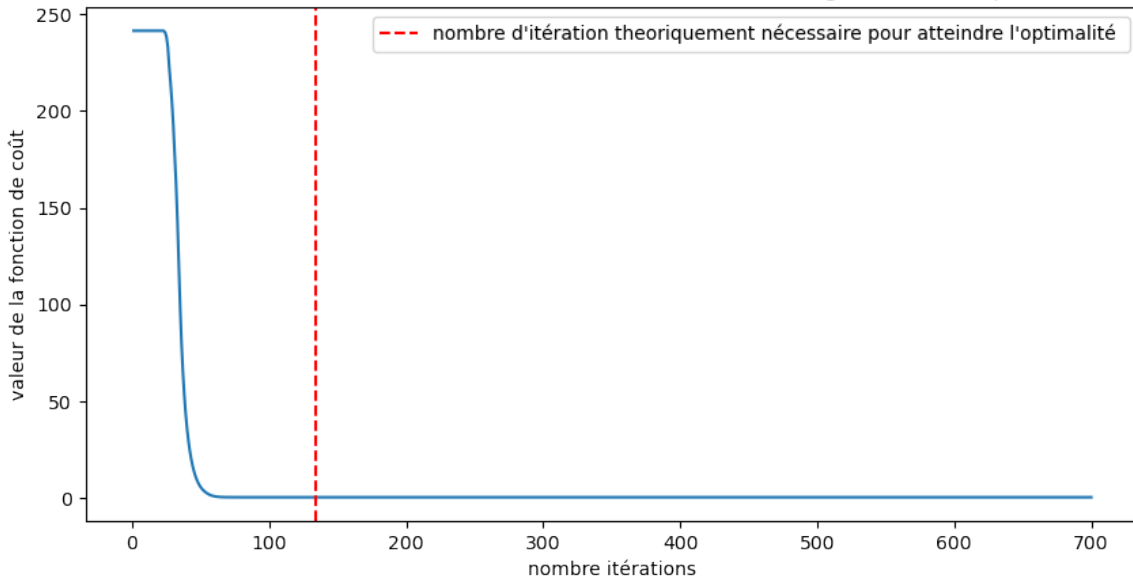


FIGURE 3 – La convergence de la fonction coût de l'algorithme IHT adaptatif

valeur de la fonction de coût en fonction du nombre d'itérations avec l'algorithme ISTA

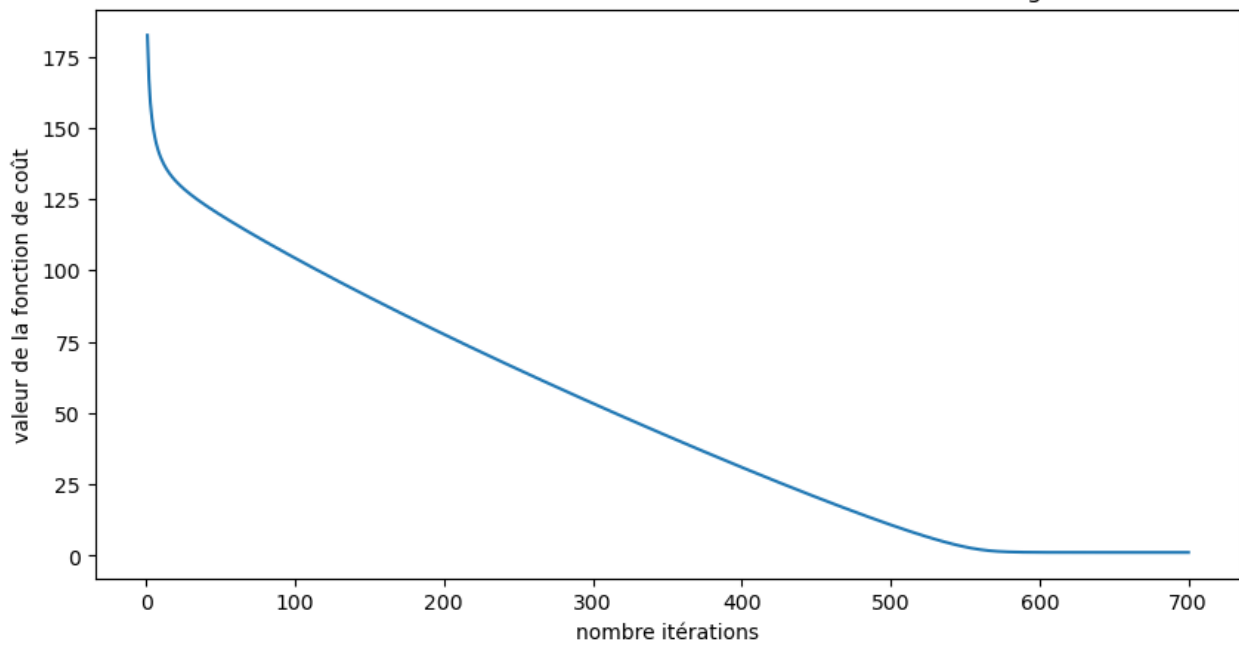


FIGURE 4 – La convergence de la fonction coût de l'algorithme ISTA

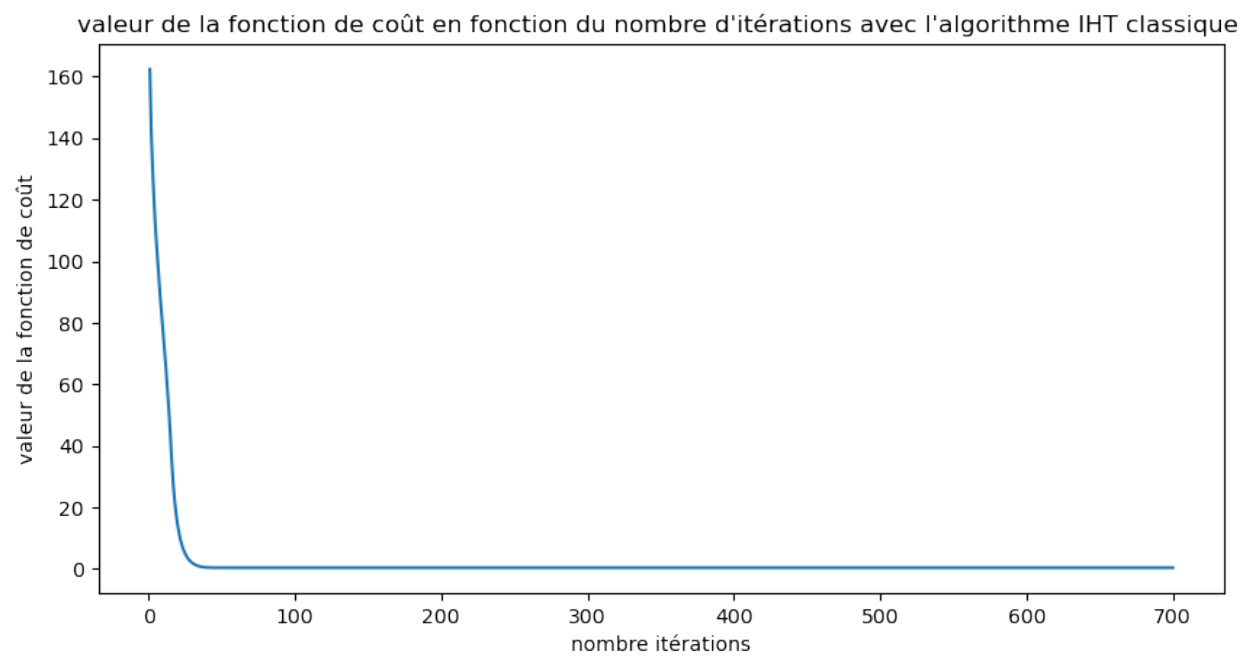


FIGURE 5 – La convergence de la fonction coût de l'algorithme IHT classique

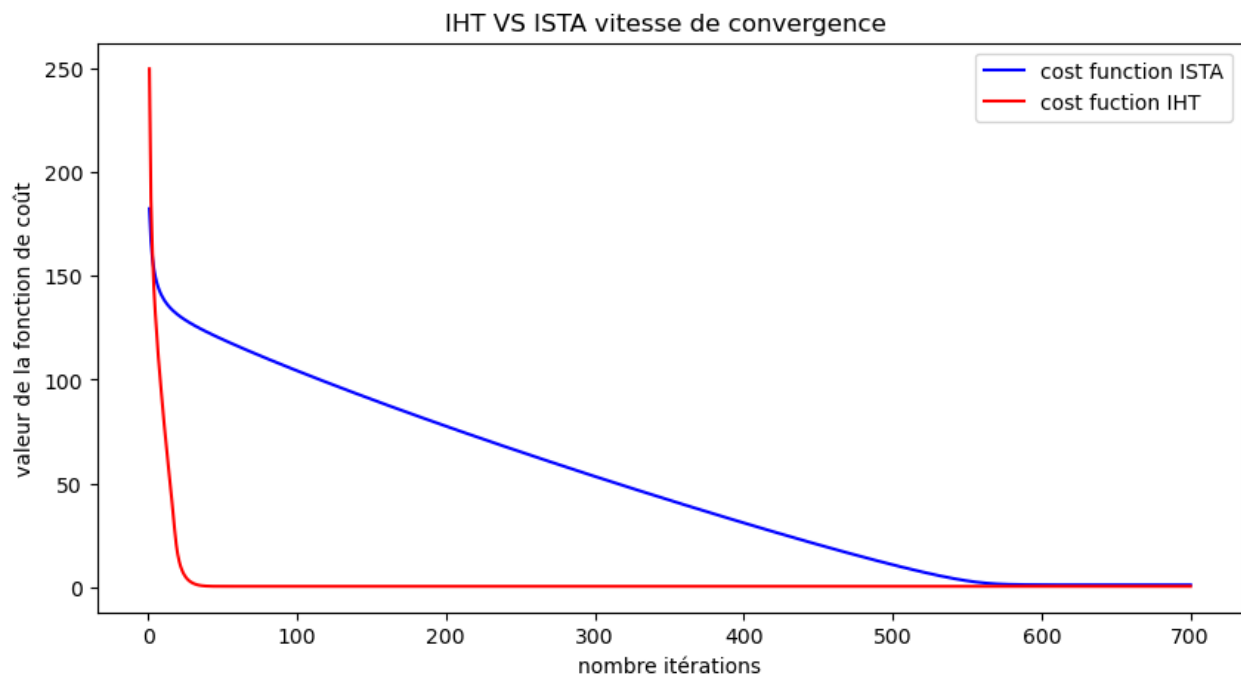


FIGURE 6 – comparaison entre la vitesse de convergence de l’algorithme ISTA et IHT adaptatif
valeur de la fonction de coût en fonction du nombre d’itérations pour l’algorithme ISTA et IHT classique

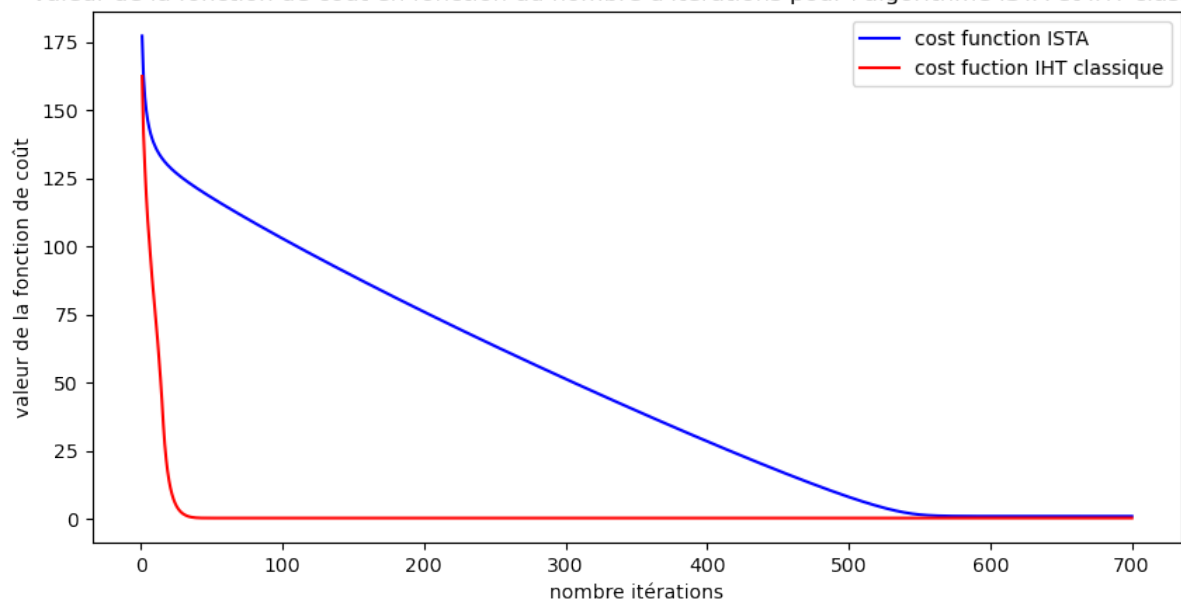


FIGURE 7 – comparaison entre la vitesse de convergence de l’algorithme ISTA et IHT classique

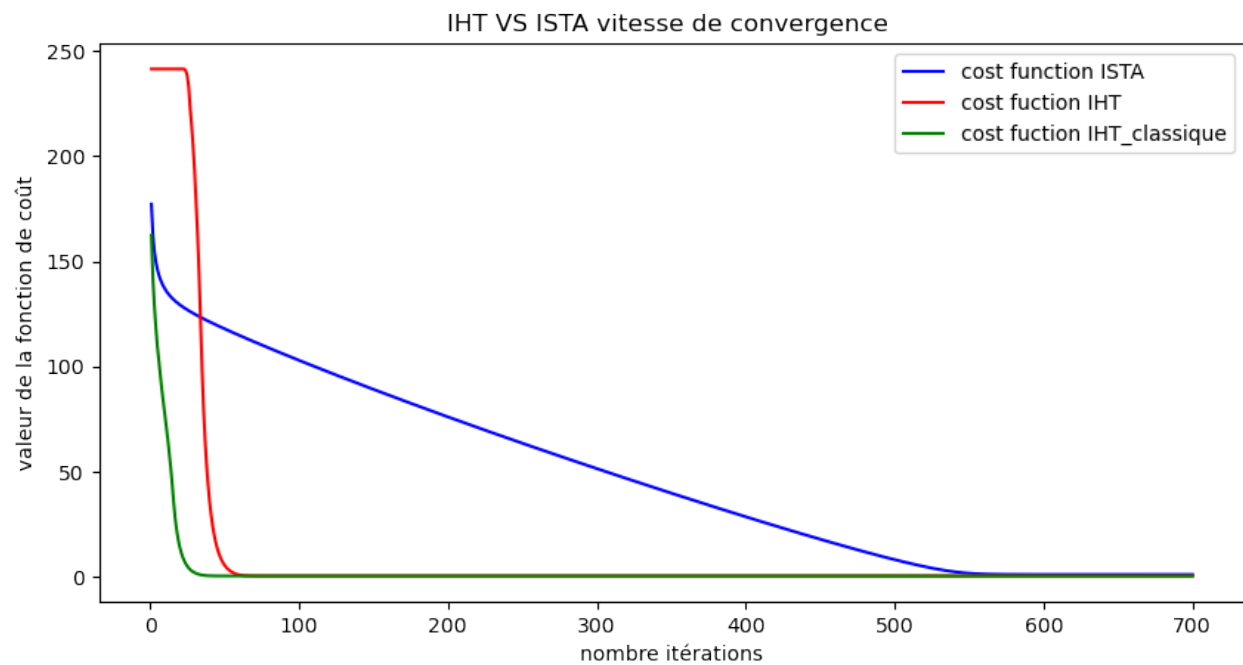


FIGURE 8 – comparaison entre la vitesse de convergence de l’algorithme ISTA et IHT classique et IHT adaptatif

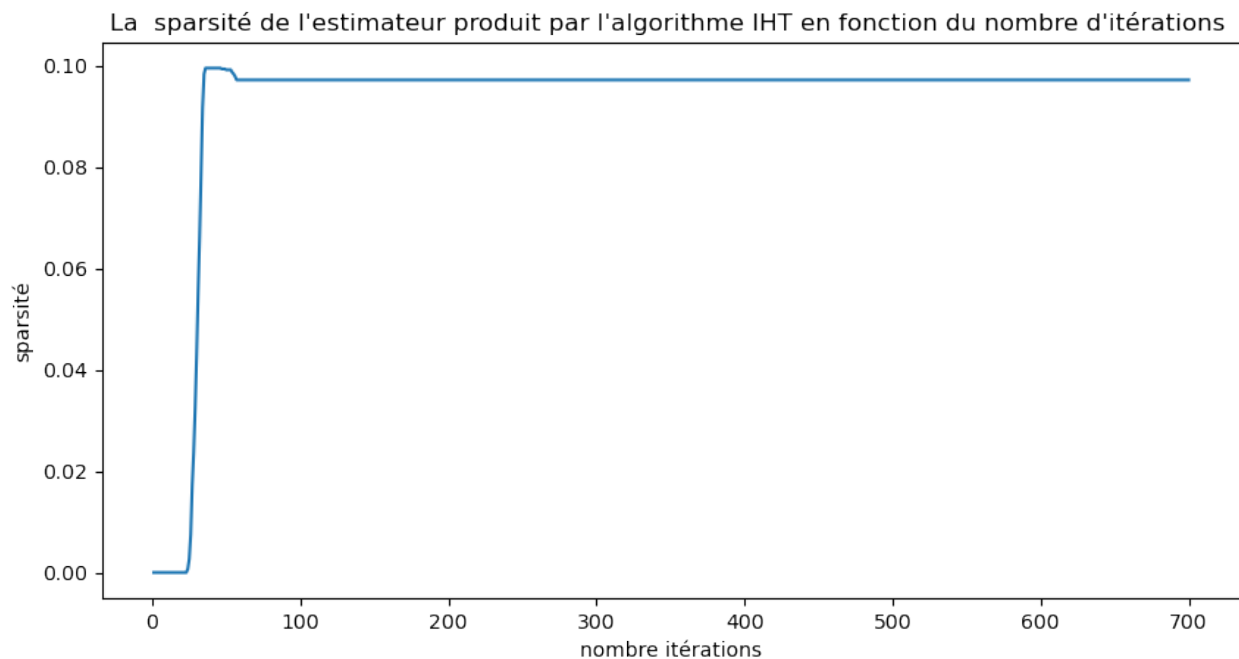


FIGURE 9 – La sparsité de l’estimateur produit par l’algorithme IHT adaptatif en fonction du nombre d’itérations

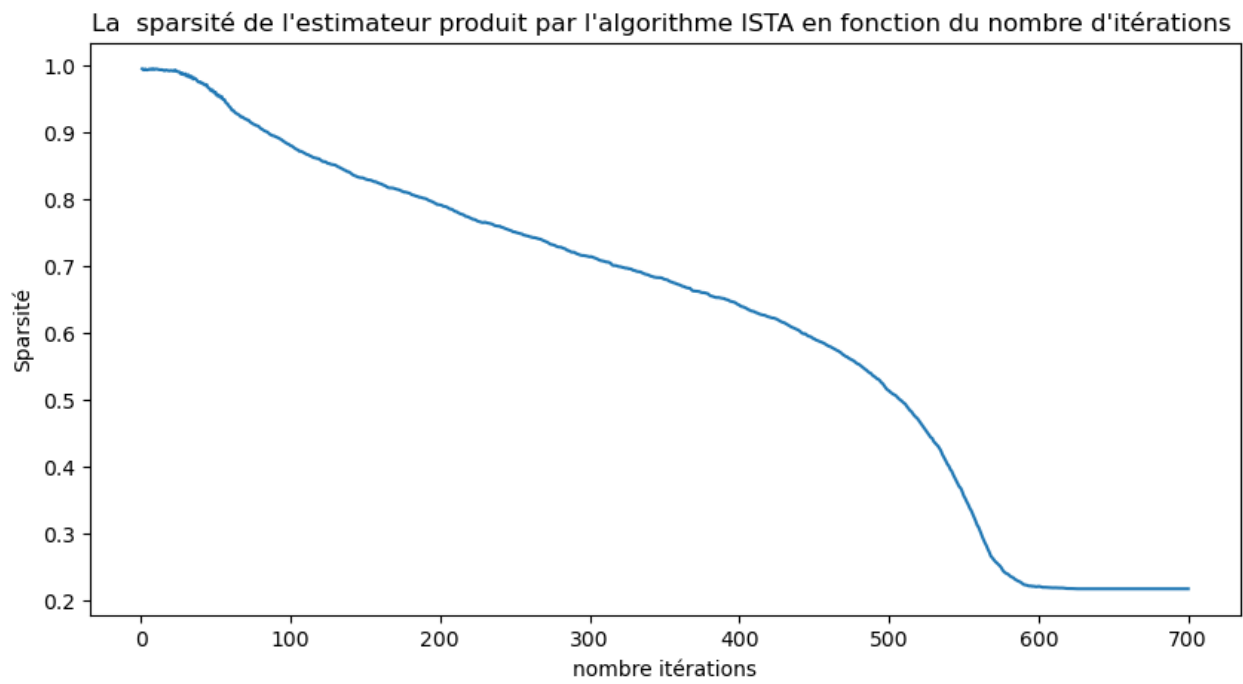


FIGURE 10 – La sparsité de l'estimateur produit par l'algorithme ISTA en fonction du nombre d'itérations

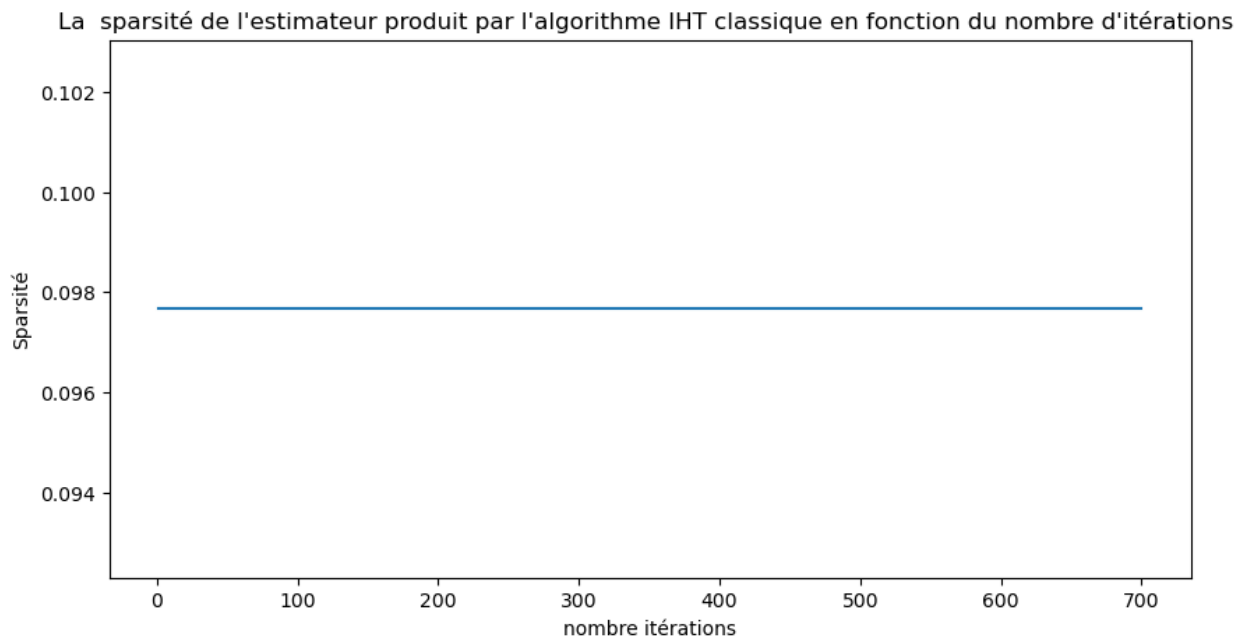


FIGURE 11 – La sparsité de l'estimateur produit par l'algorithme IHT classique en fonction du nombre d'itérations

A.2 Décomposition en ondelettes

Photo original VS photo obtenue avec la décomposition en ondelettes

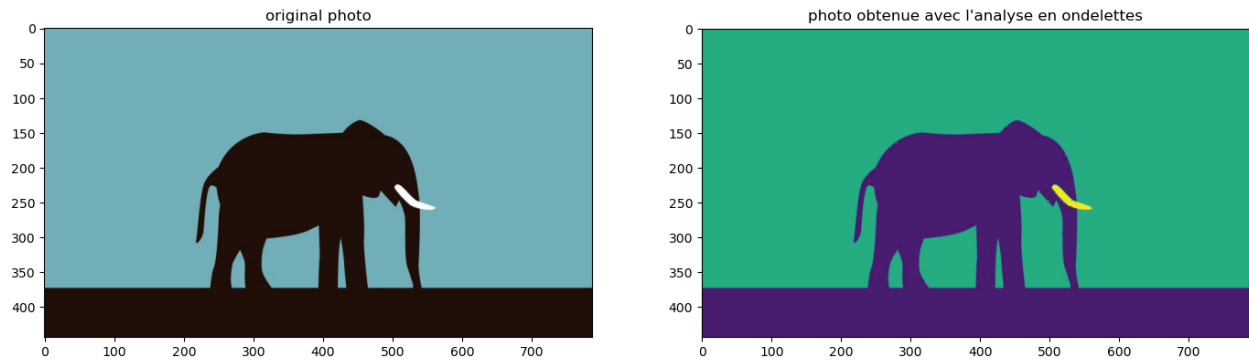


FIGURE 12 – le résultat en utilisant la représentation RVB

Les Couches de la décomposition de la photo

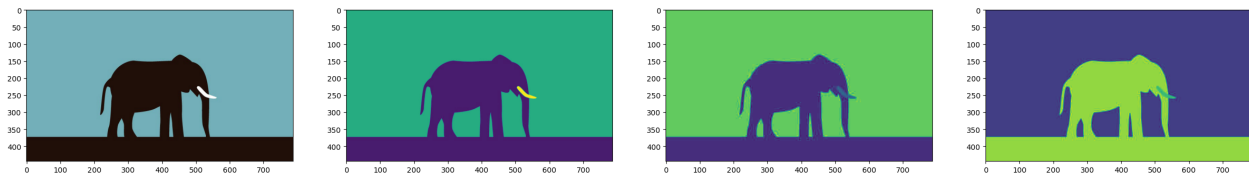


FIGURE 13 – La représentation $YC_B C_R$ d'une photo

Photo original VS photo obtenue avec le décomposition en ondelettes avec le système YCBCR

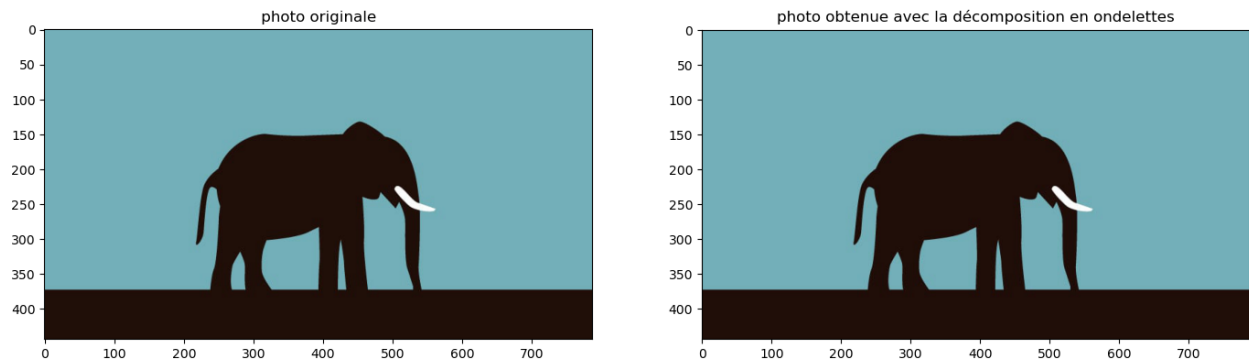


FIGURE 14 – le résultat en utilisant la représentation $YC_B C_R$

B Résultats sur les images

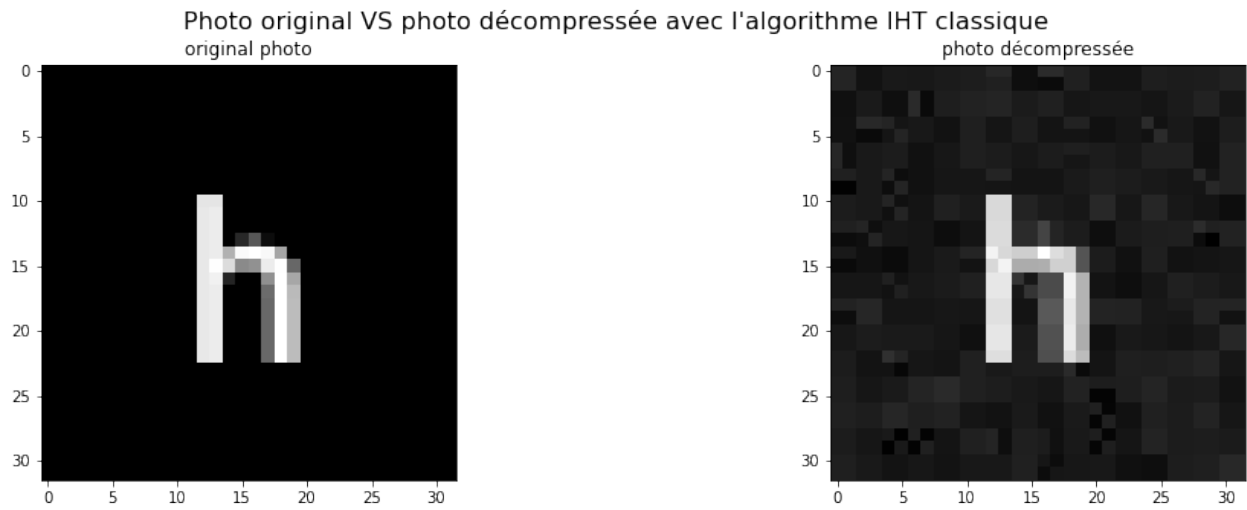


FIGURE 15 – le résultat de la décompression avec l'algorithme IHT classique

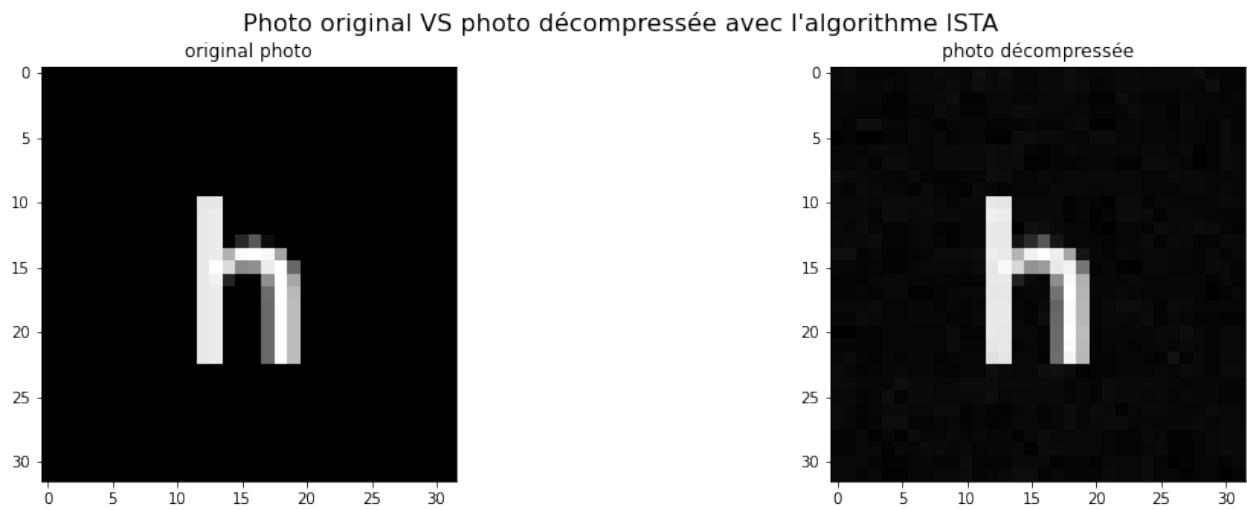


FIGURE 16 – le résultat de la décompression avec l'algorithme ISTA

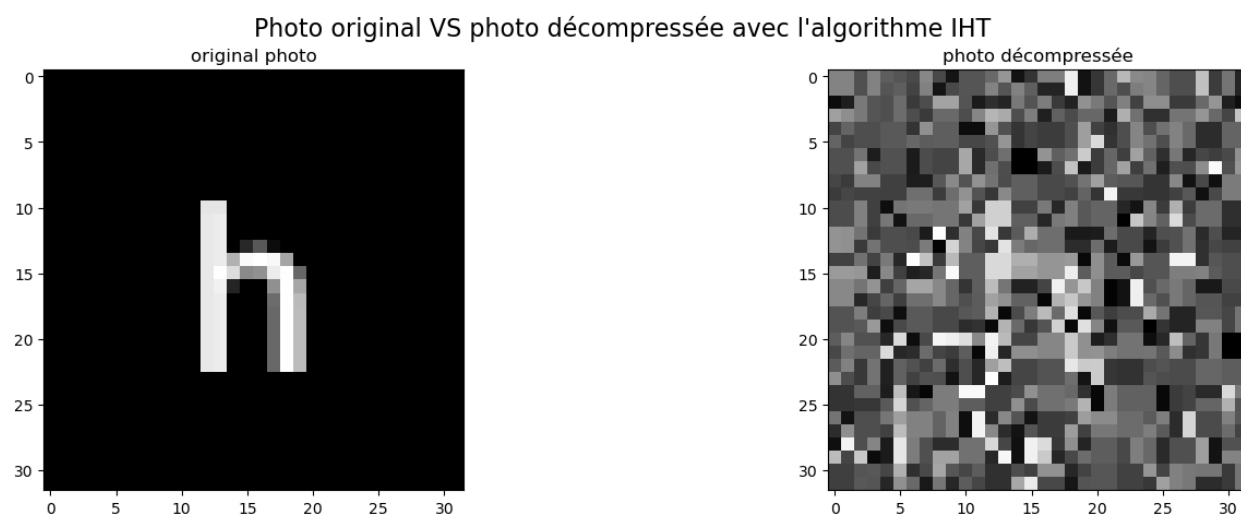


FIGURE 17 – le résultat de la décompression avec l'algorithme IHT adaptatif

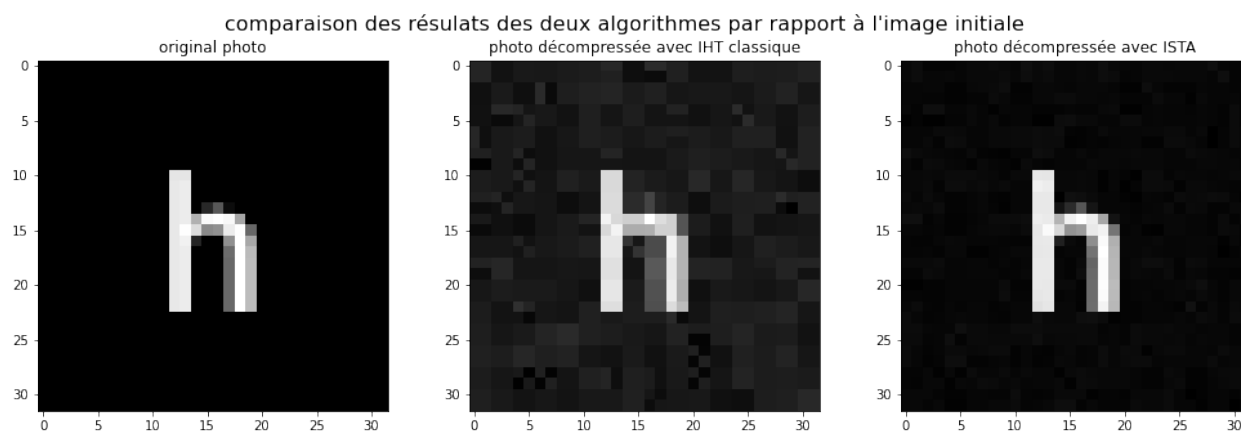


FIGURE 18 – comparaison entre les résultats fourni par l'algorithme IHT classique et l'ISTA

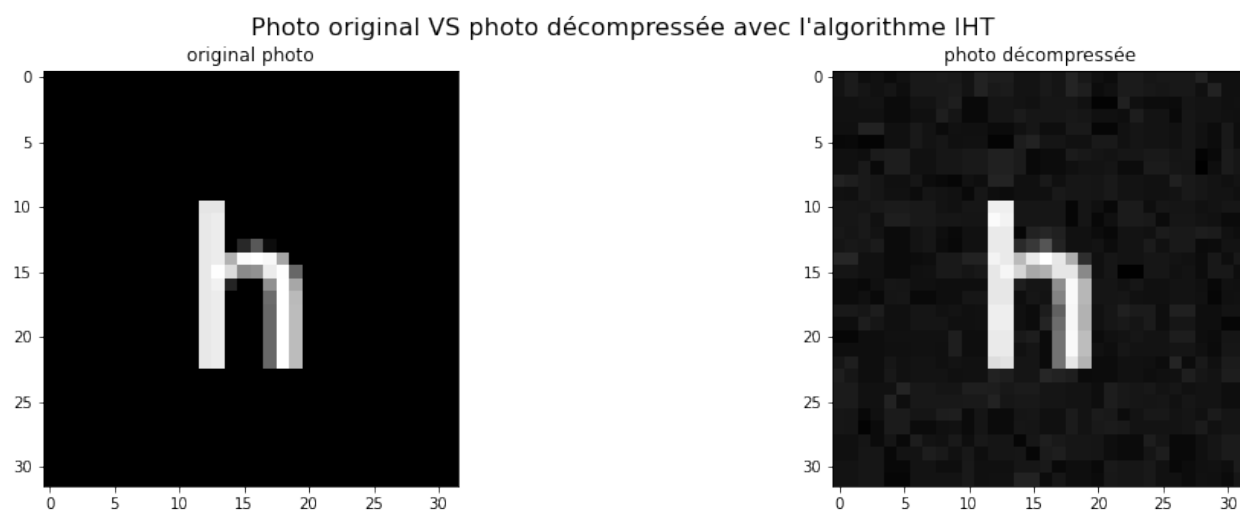


FIGURE 19 – le résultat de la décompression avec l'algorithme IHT adaptatif avec des dimensions proches des originales

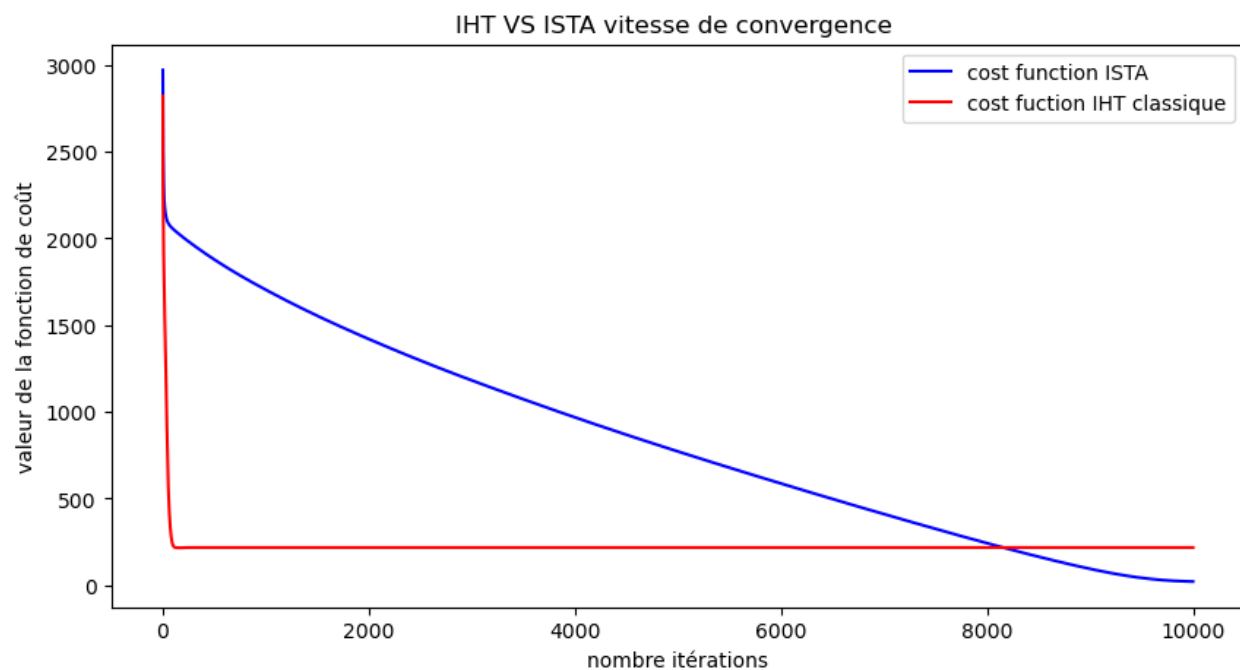


FIGURE 20 – Comparaison entre la convergence de la fonction coût de l'algorithme IHT classique et l'algorithme ISTA appliqué sur les photos

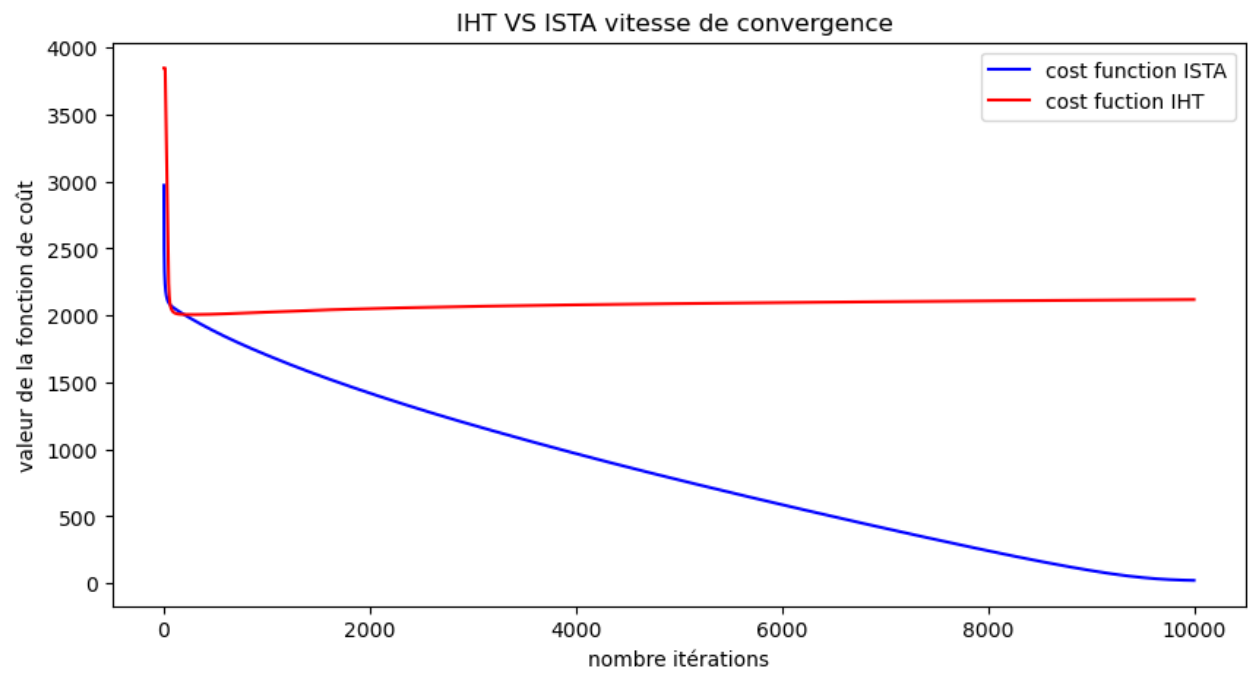


FIGURE 21 – Comparaison entre la convergence de la fonction coût de l’algorithme IHT adaptatif et l’algorithme ISTA appliqué sur les photos

répartition des coefficients obtenus suite à la décomposition en ondelettes

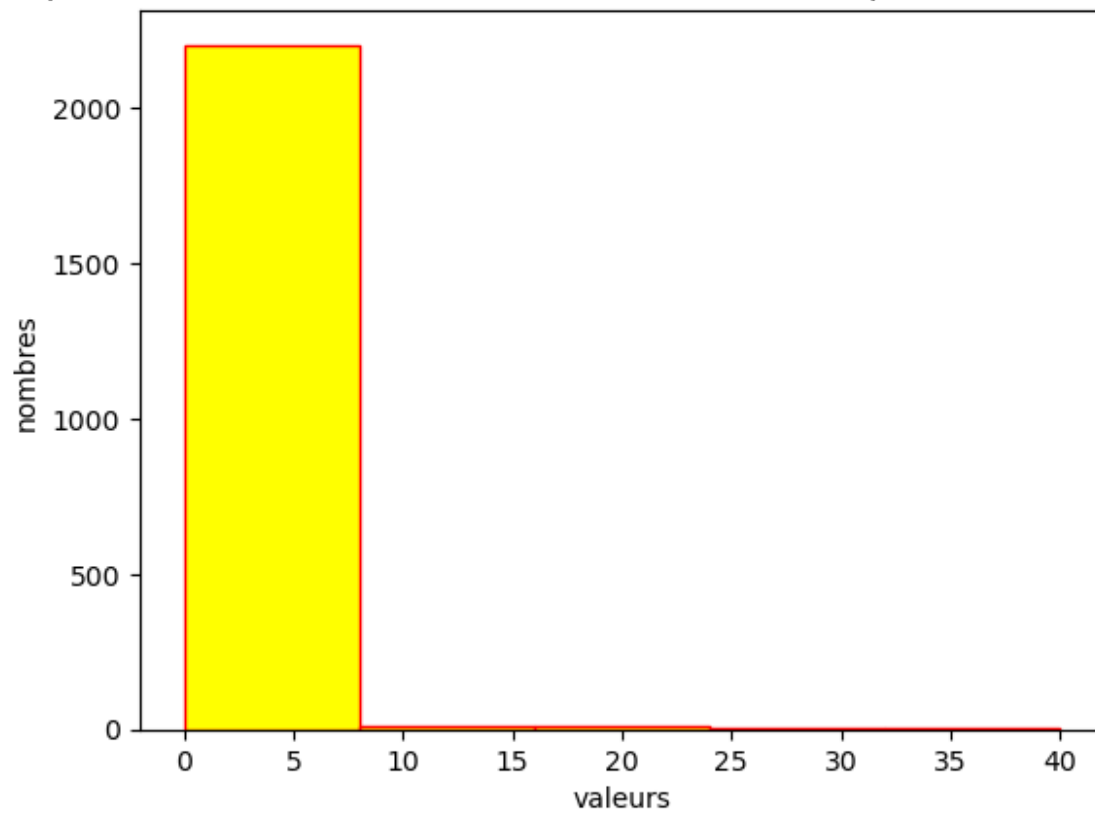


FIGURE 22 – Répartition des coefficients obtenus suite à la décomposition en ondelettes

C Preuves

Les preuves ci-dessus sont tirées de l'article [14]

$$\Xi := \frac{\sigma}{\|X\|_{2,\infty}^2} X^\top \xi \quad \text{et} \quad \Phi := \left(\frac{1}{\|X\|_{2,\infty}^2} X^\top X - \mathbf{I}_p \right)$$

Il est utile de remarquer que tant que X satisfait $\text{RIP}(3s, \delta/2)$, alors Φ est une contraction pour des vecteurs $3s$ -sparse. Ceci est reformulé dans le Lemme suivant que nous démontrons dans l'Annexe.

LEMME 1. Si X satisfait $\text{RIP}(3s, \delta/2)$ alors pour tous $S \subset \{1, \dots, p\}$ tel que $|S| \leq 3s$, nous avons $\lambda_{\max}(\Phi_{SS}) \leq \delta$.

Avant de présenter nos résultats, nous donnons un premier résultat qui est pertinent pour l'analyse de notre algorithme. Nous attirons l'attention du lecteur sur le fait que notre analyse est entièrement déterministe car nous nous plaçons dans un événement aléatoire bien choisi qui capture la complexité de notre modèle. Plus précisément, nous considérons l'événement

$$O = \left\{ \sum_{i=1}^s \Xi_{(i)}^2 \leq \frac{10\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2} \right\}.$$

En utilisant le Lemme 3 (cf. Annexe), l'événement O se produit avec une grande probabilité. Conditionnellement à l'événement O , le prochain Théorème montre qu'à chaque étape, l'estimateur correspondant est $2s$ -sparse et la fonction de substitution de l'erreur d'estimation, donnée par $s\lambda_m^2$, diminue exponentiellement.

Théorème 1. Supposons que β est s -sparse, c'est-à-dire que $|\beta|_0 \leq s$ et que X satisfait $\text{RIP}(3s, \delta/2)$. Nous notons par S le support de β . Soient $\lambda_0, \lambda_\infty > 0, 0 < \kappa < 1$, et définissons $(\hat{\beta}^m)_m$ et sa séquence de seuillage correspondante $(\lambda_m)_m$ comme dans (4)-(5). Supposons que $\delta < 1/36 \vee \kappa, \|\beta\| \leq \sqrt{s}\lambda_0$ et $\frac{\sigma\sqrt{40\log(ep/s)}}{\|X\|_{2,\infty}} \leq \lambda_\infty$. Si O est vérifié, alors pour tout m , nous avons

$$|\hat{\beta}_{S^c}^m|_0 \leq s$$

et

$$\|\hat{\beta}^m - \beta\|^2 \leq 9s\lambda_m^2$$

Preuve du Lemme 1. Rappelons que X satisfait $\text{RIP}(3s, \delta/2)$. Il est facile de constater que $m_{3s} \leq \|X\|_{2,\infty}^2 \leq L_{3s}$. Donc pour $S \in \{1, \dots, p\}$ tel que $|S| \leq 3s$, nous avons

$$\lambda_{\max}(\Phi_{SS}) \leq \left(1 - \frac{m_{3s}}{L_{3s}}\right) \vee \left(\frac{L_{3s}}{m_{3s}} - 1\right).$$

Comme $1 - \frac{m_{3s}}{L_{3s}} \leq \delta/2$, il reste à prouver que

$$\frac{L_{3s}}{m_{3s}} - 1 \leq \delta$$

En utilisant le fait que $m_{3s} \geq (1 - \delta/2)L_{3s}$, on obtient

$$\frac{L_{3s}}{m_{3s}} - 1 \leq \frac{\delta/2}{1 - \delta/2} \leq \delta$$

puisque $\delta \leq 1$.

Preuve du Théorème 1. Nous procédons par récurrence. Pour $m = 0$, le résultat est évident. Nous supposons maintenant le résultat vrai pour m et le prouvons pour $m + 1$. Dans ce qui suit, nous désignons par H^{m+1} le vecteur

$$H^{m+1} = \hat{\beta}^m + \frac{1}{\|X\|_{2,\infty}^2} X^\top (Y - X\hat{\beta}^m).$$

Remarquez que H^{m+1} peut être écrit sous la forme

$$H^{m+1} = \beta + \Phi(\beta - \hat{\beta}^m) + \Xi,$$

et que

$$\hat{\beta}^{m+1} = \mathbf{T}_{\lambda_{m+1}}(H^{m+1}).$$

Nous prouvons la première partie du résultat par l'absurde. Supposons que $|\hat{\beta}_{S^c}^{m+1}|_0 > s$. Il existe alors un sous-ensemble \tilde{S} de S^c tel que $|\tilde{S}|_0 = s$ et

$$s\lambda_{m+1}^2 \leq \sum_{i \in \tilde{S}} (H_i^{m+1})^2 \mathbf{1}\{|H_i^{m+1}| \geq \lambda_{m+1}\}$$

Puisque \tilde{S} n'est pas supporté sur S , alors nous avons

$$\sqrt{s}\lambda_{m+1} \leq \sqrt{\sum_{i \in \tilde{S}} \Xi_i^2} + \sqrt{\sum_{i \in \tilde{S}} \langle \Phi_i^\top, \beta - \hat{\beta}^m \rangle^2}.$$

Puisque β est s -parcimonieux et que $|\hat{\beta}_{S^c}^m|_0 \leq s$ alors $\beta - \hat{\beta}^m$ est au plus $2s$ -parcimonieux. De plus $|\tilde{S}| = s$. Ainsi, en utilisant le Lemme 1, nous avons que En utilisant l'hypothèse de récurrence et l'événement O , nous obtenons de plus que

$$\begin{aligned} \sqrt{s}\lambda_{m+1} &\leq \frac{\sqrt{10\sigma^2 s \log(ep/s)}}{\|X\|_{2,\infty}} + 3\delta\sqrt{s}\lambda_m \\ &\leq (1/2 + 3\sqrt{\delta})\sqrt{s}\lambda_{m+1} < \sqrt{s}\lambda_{m+1} \end{aligned}$$

aussilongtempsque

$\delta < 1/36$, ce qui est absurde. Donc $\left| \hat{\beta}_{S^c}^{m+1} \right|_{0 \leq s}$. Pour la deuxième partie, observez que $\forall i \in S$,

$$\hat{\beta}_i^{m+1} - \beta_i = -H_i^{m+1} \mathbf{1} \{ |H_i^{m+1}| \leq \lambda_{m+1} \} + \Xi_i + \left\langle \Phi_i^\top, \beta - \hat{\beta}^m \right\rangle$$

En utilisant les mêmes arguments qu'avant, nous avons

$$\left\| \hat{\beta}_S^{m+1} - \beta \right\| \leq \sqrt{s} \lambda_{m+1} + \frac{\sqrt{10\sigma^2 s \log(ep/s)}}{\|X\|_{2,\infty}} + \delta \left\| \hat{\beta}^m - \beta \right\|.$$

De plus sur S^c , nous avons

$$\left\| \hat{\beta}_{S^c}^{m+1} \right\| \leq \frac{\sqrt{10\sigma^2 s \log(ep/s)}}{\|X\|_{2,\infty}} + \delta \left\| \hat{\beta}^m - \beta \right\|$$

D'où

$$\left\| \hat{\beta}^{m+1} - \beta \right\| \leq \sqrt{s} \lambda_{m+1} + 2 \frac{\sqrt{10\sigma^2 s \log(ep/s)}}{\|X\|_{2,\infty}} + 2\delta \left\| \hat{\beta}^m - \beta \right\|.$$

En utilisant la définition de λ_m et l'hypothèse de récurrence, nous obtenons que

$$\left\| \hat{\beta}^{m+1} - \beta \right\| \leq \sqrt{s} \lambda_{m+1} + \sqrt{s} \lambda_{m+1} + 6\delta \sqrt{s} \lambda_m$$

Nous concluons que

$$\left\| \hat{\beta}^{m+1} - \beta \right\| \leq \sqrt{s} \lambda_{m+1} (2 + 6\sqrt{\delta}) \leq 3\sqrt{s} \lambda_{m+1}$$

Théorème 2. Soit $0 < \kappa < 1$. Supposons que $\delta \leq 1/36 \vee \kappa$ et que X satisfait RIP $(3s, \delta/2)$. Soit $\hat{\lambda}_0$ et $\hat{\lambda}_\infty$ donnés par (9) et (11), $(\lambda_m)_m$ soit la séquence correspondante d'estimateurs (5), et \hat{m} soit le moment d'arrêt (10). Alors les propriétés suivantes sont vérifiées

$$\begin{aligned} \sup_{|\beta|_0 \leq s} \mathbf{P}_\beta \left(\left\| \hat{\beta}^{\hat{m}} - \beta \right\|^2 \geq \frac{360\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2} \right) &\leq e^{-c_1 s \log(ep/s)}, \\ \sup_{|\beta|_0 \leq s} \mathbf{P}_\beta \left(\left| \hat{\beta}^{\hat{m}} \right|_0 \geq 2s \right) &\leq e^{-c_1 s \log(ep/s)}, \end{aligned}$$

et

$$\begin{aligned} \sup_{|\beta|_0 \leq s} \mathbf{P}_\beta \left(\hat{m} \geq 2 \log \left(\frac{5\|\beta\|^2 \|X\|_{2,\infty}^2}{\sigma^2 s \log(ep/s)} \vee 25 \right) / \log(1/\kappa) + 1 \right) \\ \leq e^{-c_1 s \log(ep/s)}, \end{aligned}$$

pour une constante absolue $c_1 > 0$.

Preuve du Théorème 2. Supposons que l'événement O se produit, alors en utilisant la Proposition 1, nous avons $\|\beta\|^2 \leq s\hat{\lambda}_0^2$. Nous pouvons alors appliquer le Théorème 1 et obtenir que

$$\forall m \geq 0, \quad \left\| \hat{\beta}^m - \beta \right\|^2 \leq 9 \left(s\hat{\lambda}_0^2 \kappa^{m/2} \vee \frac{40\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2} \right).$$

Avec le choix de \hat{m} , nous obtenons en outre en utilisant la Proposition 1 que

$$s\hat{\lambda}_0^2 \kappa^{\hat{m}/2} \leq \frac{40\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2}$$

D'où

$$\left\| \hat{\beta}^{\hat{m}} - \beta \right\|^2 \leq 360 \frac{\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2}$$

SEUILLEMENT DUR ITÉRATIF ADAPTATIF 25 Il s'ensuit que

$$\begin{aligned} \mathbf{P} \left(\left\| \hat{\beta}^{\hat{m}} - \beta \right\|^2 \geq 360 \frac{\sigma^2}{\|X\|_{2,\infty}^2} s \log(ep/s) \right) &\leq \\ \mathbf{P} \left(\sum_{i=1}^s \Xi_{(i)}^2 \geq \frac{10\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2} \right). \end{aligned}$$

Nous concluons en utilisant le Lemme 3. Nous procédons de manière similaire pour les déclarations restantes en utilisant la Proposition 1.

LEMME 3. Il existe $c > 0$ tel que

$$\mathbf{P} \left(\sum_{i=1}^s \Xi_{(i)}^2 \geq \frac{10\sigma^2 s \log(ep/s)}{\|X\|_{2,\infty}^2} \right) \leq e^{-cs \log(ep/s)}.$$

Preuve. En utilisant le Lemme 2, et la borne de l'union puisque

$$\binom{p}{s} \leq \left(\frac{ep}{s} \right)^s \leq e^{s \log(ep/s)}$$

LEMME 2. Supposons que ξ est un vecteur aléatoire 1-sous-Gaussien centré. Alors, pour tout $S \subset \{1, \dots, p\}$ tel que $|S| \leq s$, nous avons

$$\forall t \geq 0, \quad \mathbf{P} \left(\frac{1}{\|X\|_{2,\infty}^4} \sum_{i \in S} (X_i^\top \xi)^2 \geq \frac{s+t}{\|X\|_{2,\infty}^2} \right) \leq e^{-(t/8) \wedge (t^2/(64s))}.$$

Preuve : Remarquez que

$$\sum_{i \in S} (X_i^\top \xi)^2 = \|X_S^\top \xi\|^2$$

En utilisant le Théorème 2.1 dans [13], nous obtenons

$$\mathbf{P} \left(\|X_S^\top \xi\|^2 \geq \text{Tr}(X_S X_S^\top) + 2 \|X_S X_S^\top\|_F \sqrt{t} + 2\lambda_{\max}(X_S X_S^\top) t \right) \leq e^{-t}.$$

Nous avons que

$$\text{Tr}(X_S X_S^\top) \leq s \|X\|_{2,\infty}^2.$$

De plus, $\lambda_{\max}(X_S X_S^\top) \leq 2 \|X\|_{2,\infty}^2$ (en utilisant RIP) et $\|X_S X_S^\top\|_F \leq 2\sqrt{s} \|X\|_{2,\infty}^2$ puisque le rang de $X_S X_S^\top$ est au plus s . Par conséquent, nous concluons en utilisant l'inégalité de concentration ci-dessus que

$$\mathbf{P} \left(\frac{1}{\|X\|_{2,\infty}^4} \sum_{i \in S} (X_i^\top \xi)^2 \geq \frac{s + 4\sqrt{st} + 4t}{\|X\|_{2,\infty}^2} \right) \leq e^{-t}.$$

La borne finale est obtenue en considérant $u = 8\sqrt{t}(\sqrt{s} \vee \sqrt{t})$ et de manière équivalente $t = (u/8) \wedge (u^2/(64s))$.

Proposition 1. Soit β être s -parcimonieux et soit X satisfait $\text{RIP}(3s, \delta/2)$. Supposons que $\delta \leq 1/4$ et que l'événement O se produit. Alors

$$\left(\|\beta\| \vee \frac{\sigma \sqrt{10s \log(ep/s)}}{\|X\|_{2,\infty}} \right) \leq \sqrt{s} \hat{\lambda}_0 \leq 10 \left(\|\beta\| \vee \frac{\sigma \sqrt{10s \log(ep/s)}}{\|X\|_{2,\infty}} \right)$$

et

$$2 \log \left(\frac{\|\beta\|^2 \|X\|_{2,\infty}^2}{40\sigma^2 s \log(ep/s)} \vee 1/4 \right) / \log(1/\kappa) \leq \hat{m} - 1 \leq 2 \log \left(\frac{5\|\beta\|^2 \|X\|_{2,\infty}^2}{\sigma^2 s \log(ep/s)} \vee 25 \right) / \log(1/\kappa)$$

Remarquez que pour les grandes valeurs de $\|\beta\|$, $\hat{\lambda}_0$ est du même ordre que $\frac{\|\beta\|}{\sqrt{s}}$ avec une haute probabilité et que \hat{m} est de l'ordre de $\log \left(\frac{\|\beta\|^2 \|X\|_{2,\infty}^2}{\sigma^2 s \log(ep/s)} \right) / \log(1/\kappa)$. Nous sommes maintenant prêts à énoncer un résultat sur l'optimalité minimax de notre procédure. Nous choisirons également λ_∞ du même ordre que le seuil universel

$$\hat{\lambda}_\infty = \frac{\sigma \sqrt{40 \log(ep/s)}}{\|X\|_{2,\infty}}.$$

Preuve de la Proposition 1. Soit \tilde{S} un ensemble de taille s , alors

$$\|M_{\tilde{S}}\| \leq \|\beta_{\tilde{S}}\| + \delta \|\beta\| + \frac{\sigma \sqrt{10s \log(ep/s)}}{\|X\|_{2,\infty}}.$$

D'où

$$\|M_{\tilde{S}}\| \leq \|\beta\|(1 + \delta) + \frac{\sigma \sqrt{10s \log(ep/s)}}{\|X\|_{2,\infty}}$$

Alors

$$\sqrt{s}\hat{\lambda}_0 \leq 2(\sqrt{10}(1+\delta)+1) \left(\|\beta\| \vee \frac{\sigma\sqrt{10s\log(ep/s)}}{\|X\|_{2,\infty}} \right).$$

D'où

$$\sqrt{s}\hat{\lambda}_0 \leq 10 \left(\|\beta\| \vee \frac{\sigma\sqrt{10s\log(ep/s)}}{\|X\|_{2,\infty}} \right).$$

Nous avons également pour le vrai support S de β que

$$\|M_S\| \geq (1-\delta)\|\beta\| - \frac{\sigma\sqrt{10s\log(ep/s)}}{\|X\|_{2,\infty}}.$$

Si $\|\beta\| \leq \frac{\sigma\sqrt{40s\log(ep/s)}}{\|X\|_{2,\infty}}$ le résultat est trivial. Sinon $\|\beta\| > \frac{\sigma\sqrt{40s\log(ep/s)}}{\|X\|_{2,\infty}}$, et

$$\sqrt{s}\hat{\lambda}_0 \geq \sqrt{10}(1-\delta-1/2)\|\beta\| \geq \|\beta\|.$$

Le résultat pour \hat{m} est direct.