

Représentations parcimonieuses des signaux et images

Parcimonie dans des dictionnaires

Notations

- On note $\langle \cdot, \cdot \rangle$ le produit scalaire usuel sur \mathbb{C}^N tel que $\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n}$ pour $x = (x_i)_{i=0, \dots, N-1} \in \mathbb{C}^N$ et $y = (y_i)_{i=0, \dots, N-1} \in \mathbb{C}^N$. On note $\|x\|^2 = \sum_{n=0}^{N-1} |x_n|^2$ pour $x \in \mathbb{C}^N$.
- On note δ^ℓ pour $\ell \in \{0, \dots, N-1\}$ le ℓ -ième vecteur de la base canonique de \mathbb{C}^N . Il s'agit du vecteur de \mathbb{C}^N tel que pour $n \in \{0, \dots, N-1\}$ on a $\delta_n^\ell = 1$ si $n = \ell$ et 0 sinon. La collection de vecteurs $\mathcal{B} = \{\delta^\ell, \ell \in \{0, \dots, N-1\}\}$ constitue la base canonique de \mathbb{C}^N .
- la famille de vecteurs $\tilde{\mathcal{E}} = \{\frac{1}{\sqrt{N}} e^\ell \in \mathbb{C}^N, \ell \in \mathbb{Z}\}$ tels que pour $\ell \in \mathbb{Z}$ et pour $n \in \{0, \dots, N-1\}$ la n -ième coordonnée du vecteur e^ℓ s'écrit

$$e_n^\ell = e^{\frac{2i\pi\ell n}{N}}$$

est appelée base de Fourier orthonormalisée discrète.

1 Introduction

Il n'est pas toujours facile ou possible de trouver une base dans laquelle un signal $x \in \mathbb{C}^N$ a une décomposition parcimonieuse. Considérons en effet l'exemple suivant :

$$(1) \quad x = \delta^{k_0} + \frac{e^{k_1}}{\sqrt{N}}$$

Ce signal n'est parcimonieux ni dans la base canonique, ni dans la base de Fourier discrète orthonormalisée.

Cependant si on considère la réunion des deux bases $\mathcal{D} = \{\delta^k, k = 0, \dots, N-1\} \cup \{\frac{e^k}{\sqrt{N}}, k = 0, \dots, N-1\}$ alors le signal x admet une décomposition très parcimonieuse sur \mathcal{D} , qui est (1).

Question 1:

Justifier cette affirmation en calculant et comptant les coefficients non nuls de décomposition de x dans la base canonique, dans la base de Fourier orthonormalisée et dans le dictionnaire \mathcal{D} . Quelle décomposition est la plus parcimonieuse ?

Par contre il nous faut constater que x admet une infinité de décompositions possibles sur les vecteurs de \mathcal{D} .

Question 2:

Justifier cette affirmation.

La question qui se pose est alors de trouver la décomposition de x la plus parcimonieuse parmi toutes les décompositions possibles de x sur les vecteurs de \mathcal{D} .

Nous allons voir que l'algorithme dit de Matching-Pursuit nous permet de répondre à la question posée.

2 Algorithme de Matching-Pursuit

Nous nous plaçons dans \mathbb{C}^N et considérons un système générateur noté $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ comportant $K \geq N$ vecteurs de \mathbb{C}^N , tels que pour tout $k = 0, \dots, K-1$ on a $\|\phi^k\| = 1$. Un tel système est appelé **dictionnaire** de \mathbb{C}^N .

L'algorithme de Matching-Pursuit fonctionne de la façon suivante :

1. Initialisation : $m := 0, R^0(x) = x$.
2. Par récurrence sur $m \in \mathbb{N}$
 - (a) Choix du meilleur vecteur $k_m = \arg \max_k |\langle R^m(x), \phi^k \rangle|$
 - (b) Mise à jour du « résidu » $R^{m+1}(x) = R^m(x) - \langle R^m(x), \phi^{k_m} \rangle \phi^{k_m}$

Application à un exemple :

Supposons que x est une combinaison linéaire avec peu de coefficients non nuls de vecteurs de la base canonique et de vecteurs de la base $\tilde{\mathcal{E}}$. Testons sur un exemple numérique et voyons quelle décomposition nous produit l'algorithme.

Nous programmons une version de l'algorithme 1 (abrégé MP) dont la formulation est adaptée à ce que nous voulons faire : calculer une décomposition de x sur les vecteurs de \mathcal{D} . Le vecteur sortie α correspond en effet aux coordonnées de x sur les vecteurs de \mathcal{D} calculées par l'algorithme, autrement dit l'algorithme calcule une décomposition $(\alpha_0, \alpha_1, \dots, \alpha_{K-1})$ telle que $x = \sum_{j=0}^{K-1} \alpha_j \phi^j + R(x)$ où $R(x)$ est un reste que l'on voudrait le plus petit possible et en particulier on souhaiterait $\|R(x)\| < \varepsilon$ où $\varepsilon > 0$ est une précision fixée à l'avance.

On considère ici le signal $x = (x_0, \dots, x_{N-1})$ tel que pour tout $n = 0, \dots, N-1$ avec $N = 1024$

$$(2) \quad x_n = 4\delta_n^{256} + 0.25\delta_n^{700} + \frac{16}{\sqrt{N}} \cos\left(2\pi * 25 * \frac{n}{N}\right)$$

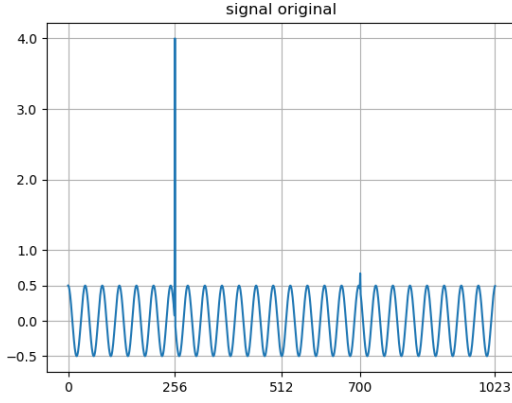
Comme on le voit sur les figures l'algorithme converge vite car au bout de 14 itérations on a $\|R^m(x)\| \leq 10^{-6}$. De plus on constate que l'algorithme de Matching-Pursuit retrouve les composantes de la décomposition parcimonieuse de x . Nous nous poserons la question de savoir si c'est toujours le cas si certaines conditions sont réunies.

Nous avons les propriétés suivantes

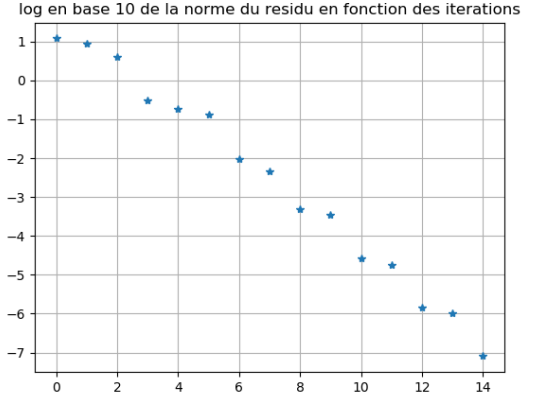
Proposition 1 Soit $x \in \mathbb{C}^N$ et \mathcal{D} un dictionnaire de \mathbb{C}^N . Avec les notations qui précèdent pour tout $m \in \mathbb{N}$ et $M \in \mathbb{N}$

1. $x = \sum_{m=0}^{M-1} \langle R^m(x), \phi^{k_m} \rangle \phi^{k_m} + R^M(x)$.
2. $R^{m+1}(x)$ et ϕ^{k_m} sont orthogonaux, et on a
 - (a) la suite $(\|R^m(x)\|)_{m \in \mathbb{N}}$ est décroissante.
 - (b)

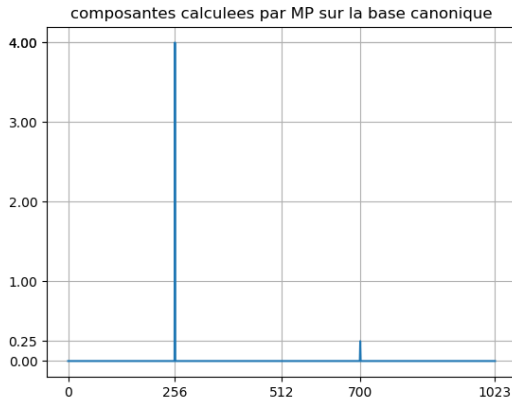
$$\|x\|^2 = \sum_{m=0}^{M-1} |\langle R^m(x), \phi^{k_m} \rangle|^2 + \|R^M(x)\|^2$$



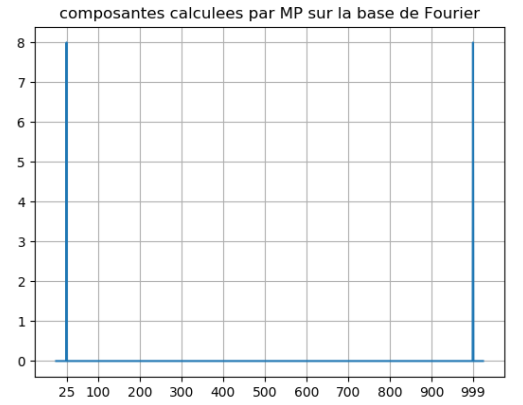
Signal original $x_n = 4\delta_n^{256} + 0.25\delta_n^{700} + \frac{16}{\sqrt{N}} \cos(2\pi * 25 * \frac{n}{N})$



$\|R^m(x)\|$ en fonction de m pour $m \leq 14$



Composantes calculées par MP sur la base canonique



Composantes calculées par MP sur la base de Fourier

FIGURE 1 – En haut à gauche le signal original, en haut à droite $\log(\|R_m(x)\|)$ en fonction des itérations, en bas à gauche les composantes calculées par MP sur la base canonique, et en bas à droite celles calculées sur la base de Fourier.

Algorithme 1 : Matching Pursuit.

Data : dictionnaire $D = [\phi^0, \dots, \phi^{K-1}] \in \mathbb{C}^{N,K}$, $x \in \mathbb{R}^N$, $\varepsilon > 0$, $M_{max} \in \mathbb{N}$

initialisation :

$r \leftarrow x$ [Initialisation du résidu];

$\alpha \leftarrow 0_K$ [Initialisation de α];

$m \leftarrow 0_K$ [Initialisation des itérations];

while $m \leq M_{max}$ et $\|r\| > \varepsilon$ **do**

$m \leftarrow m + 1$;
 $\forall m, c_m \leftarrow \langle r, \phi^m \rangle$;
 $k_m \leftarrow \arg \max_{m \leq K-1} |\langle r, \phi^m \rangle|$;
 $\alpha(k_m) \leftarrow \alpha(k_m) + c_{k_m}$;
 $r \leftarrow r - c_{k_m} \phi^{k_m}$

Result : $\alpha \in \mathbb{C}^K$, M nombre d'itérations.

Question 3:

1. Étant donné la matrice D du dictionnaire \mathcal{D} , un vecteur $y \in \mathbb{R}^N$, indiquer comment calculer sans boucle avec Python $\langle y, \phi^k \rangle$ pour tout $k = 0, \dots, K-1$.
2. Étant donné la matrice D du dictionnaire \mathcal{D} , un vecteur $\alpha \in \mathbb{C}^K$, indiquer comment calculer sans boucle avec Python $\sum_{i=0}^{K-1} \alpha_i \phi^i$.
3. Programmer l'algorithme de Matching Pursuit.

On cherchera en particulier à en programmer une version qui permet de vérifier qu'à chaque itération la norme du résidu est bien décroissante. Cela nous servira dans la phase de test du programme pour vérifier que la programmation de l'algorithme est correcte. Dans un deuxième temps on pourra s'en passer et garder une version qui ne renvoie pas la norme du résidu.

4. Le tester en utilisant le dictionnaire \mathcal{D} dont la matrice représentative dans la base canonique est produit par le code `creaDictBaseCanFourier.py` fourni sur Ametice et appliquant l'algorithme sur le vecteur $x = (x_0, \dots, x_{N-1})$ décrit par l'équation (2).
L'algorithme fournit-il la décomposition parcimonieuse recherchée ?

Remarque : Lorsque \mathcal{D} est une base orthonormale de \mathbb{C}^N l'algorithme retrouve la décomposition de x sur cette base en N itérations.

Question 4:

Démontrer la proposition 1.

Nous admettons le résultat suivant (voir [4] chapitre 12, section 12.3 Théorème 12.6, pour ceux qui veulent étudier la preuve)

Théorème 1 Avec les notations précédentes et $x \in \mathbb{C}^N$ il existe $0 < \beta(\mathcal{D}) \leq 1$ avec $\beta(\mathcal{D}) = \inf_{x \in \mathbb{C}^N, \|x\|=1} \sup_{j \in \{0, \dots, K-1\}} |\langle x, \phi^j \rangle|$ tel que le résidu à l'étape m vérifie

$$\|R^m(x)\| \leq (1 - \beta(\mathcal{D})^2)^m \|x\|^2.$$

En particulier $\|R^m(x)\| \rightarrow 0$ quand $m \rightarrow +\infty$, l'algorithme de Matching Pursuit converge et on a $x = \sum_{m=0}^{\infty} \langle R^m(x), \phi^{k_m} \rangle \phi^{k_m}$.

3 Conditions de recouvrement de décomposition parcimonieuse

Dans cette partie nous cherchons à expliquer le fait que l'algorithme de Matching-Pursuit a bien retrouvé la décomposition parcimonieuse du vecteur x donné par l'équation (2). Est-ce toujours le cas sous certaines conditions à préciser ? On va utiliser les mathématiques pour montrer un théorème qui indique que sous certaines hypothèses l'algorithme fait ce qu'on attend.

Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire de \mathbb{C}^N .

On note $\mu(\mathcal{D}) = \max_{i \neq j} |\langle \phi^i, \phi^j \rangle|$ la cohérence du dictionnaire. Plus $\mu(\mathcal{D})$ est petite plus le dictionnaire est dit « incohérent » et plus les vecteurs ϕ^j sont étrangers les uns aux autres. En particulier si $\mu(\mathcal{D}) = 0$ cela signifie que le dictionnaire forme une base orthonormale.

Question 5:

1. Dans le cas où \mathcal{D} est la réunion des vecteurs de la base canonique et de ceux de la base de Fourier calculer $\mu(\mathcal{D})$.
2. Dans le cas d'un dictionnaire \mathcal{D} quelconque comment calcule-t-on à l'aide de Python $\mu(\mathcal{D})$?

3.1 Conditions pour un recouvrement exact

Nous avons le résultat suivant.

Théorème 2 Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire de \mathbb{C}^N et $x \in \mathbb{C}^N$ tel que

$$(3) \quad x = \sum_{\ell \in \mathcal{L}} \alpha_\ell \phi^\ell$$

où \mathcal{L} est un sous-ensemble de $\{0, \dots, K-1\}$ tel que $\{\phi^\ell, \ell \in \mathcal{L}\}$ forme un système libre et $\alpha_\ell \neq 0$ pour tout $\ell \in \mathcal{L}$.

Si $\text{card}(\mathcal{L})\mu(\mathcal{D}) < \frac{1}{2}$ alors l'algorithme de Matching-Pursuit sélectionne à chaque itération un vecteur ϕ^ℓ avec $\ell \in \mathcal{L}$ et converge vers la décomposition (3).

Nous allons démontrer ce théorème.

Éléments de démonstration du théorème 2 : On considère $x \in \mathbb{C}^N$ qui vérifie (3) avec \mathcal{L} tel que $\{\phi^\ell, \ell \in \mathcal{L}\}$ forme un système libre.

Nous faisons une récurrence sur m , les itérations de l'algorithme.

On indique comment raisonner pour $m = 0$.

Considérons i_0 le plus petit indice dans \mathcal{L} tel que $|\alpha_{i_0}| = \max_{\ell \in \mathcal{L}} |\alpha_\ell|$.

Soit $j \in \{0, \dots, K-1\}$. Calculons $|\langle R^0(x), \phi^j \rangle|$.

- 1er cas $j \notin \mathcal{L}$ on a

$$(4) \quad \begin{aligned} |\langle R^0(x), \phi^j \rangle| &= |\langle x, \phi^j \rangle| = \left| \left\langle \sum_{\ell \in \mathcal{L}} \alpha_\ell \phi^\ell, \phi^j \right\rangle \right| \\ &\leq |\alpha_{i_0}| \text{card}(\mathcal{L})\mu < \frac{|\alpha_{i_0}|}{2} \end{aligned}$$

- 2ème cas : $j = i_0$

$$(5) \quad \begin{aligned} |\langle R^0(x), \phi^{i_0} \rangle| &= |\langle x, \phi^{i_0} \rangle| \\ &= \left| \sum_{\ell \in \mathcal{L}} \alpha_\ell \langle \phi^\ell, \phi^{i_0} \rangle \right| = \left| \alpha_{i_0} + \sum_{\ell \in \mathcal{L}, \ell \neq i_0} \alpha_\ell \langle \phi^\ell, \phi^{i_0} \rangle \right| \\ &\geq |\alpha_{i_0}| - \left| \sum_{\ell \in \mathcal{L}, \ell \neq i_0} \alpha_\ell \langle \phi^\ell, \phi^{i_0} \rangle \right| \end{aligned}$$

$$(6) \quad > |\alpha_{i_0}| - \frac{|\alpha_{i_0}|}{2} = \frac{|\alpha_{i_0}|}{2}$$

Donc l'algorithme sélectionne à la première itération un vecteur ϕ^{k_0} avec $k_0 \in \mathcal{L}$.

Question 6

Compléter la démonstration du théorème 2.

Remarque :

En réalité sous l'hypothèse $\text{card}(\mathcal{L})\mu(\mathcal{D}) < 1/2$ le système de vecteurs $\{\phi^j, j \in \mathcal{L}\}$ est un système libre. On a en effet la proposition suivante.

Proposition 2 Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire de \mathbb{C}^N de cohérence $\mu(\mathcal{D})$ et \mathcal{L} un sous-ensemble de $\{0, \dots, K-1\}$.

Si $\text{card}(\mathcal{L})\mu(\mathcal{D}) < 1$ le système $\{\phi^j, j \in \mathcal{L}\}$ est un système libre.

Des éléments de démonstration de la proposition 2 sont donnés en annexe 5.1.

3.2 Au delà du théorème 2

Lorsque $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ est un dictionnaire constitué de la réunion de deux bases orthonormées, on peut obtenir un résultat légèrement plus favorable que celui du théorème 2 en écrivant la partition $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ telle que $\{\phi^\ell, \ell \in \mathcal{L}_1\}$ est un sous-ensemble de la première base alors que $\{\phi^\ell, \ell \in \mathcal{L}_2\}$ est un sous-ensemble de la deuxième.

On peut ainsi montrer sans trop d'efforts supplémentaires le théorème suivant

Théorème 3 Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire de \mathbb{C}^N qui est la réunion d'une base orthonormée \mathcal{B}_1 et d'une base orthonormée \mathcal{B}_2 . Soit $x \in \mathbb{C}^N$ tel que

$$(7) \quad x = \sum_{\ell \in \mathcal{L}} \alpha_\ell \phi^\ell$$

où \mathcal{L} est un sous-ensemble de $\{0, \dots, K-1\}$ tel que $\{\phi^\ell, \ell \in \mathcal{L}\}$ forme un système libre.

On pose $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ avec \mathcal{L}_1 et \mathcal{L}_2 tels que $\{\phi^\ell, \ell \in \mathcal{L}_1\} \subset \mathcal{B}_1$ alors que $\{\phi^\ell, \ell \in \mathcal{L}_2\} \subset \mathcal{B}_2$.

Si $\max(\text{card}(\mathcal{L}_1), \text{card}(\mathcal{L}_2)) \times \mu(\mathcal{D}) < \frac{1}{2}$ alors l'algorithme de Matching-Pursuit sélectionne à chaque itération un vecteur ϕ^ℓ avec $\ell \in \mathcal{L}$ et converge vers la décomposition (3).

Question 7: Facultatif

Faire la démonstration du théorème 3.

Question 8

On considère le cas d'un dictionnaire \mathcal{D} qui est la réunion de la base canonique \mathcal{B} et de la base de Fourier orthonormalisée $\tilde{\mathcal{E}}$ pour $N = 1024$.

Expliciter dans ce cas les hypothèses nécessaires

1. pour appliquer le théorème 2
2. pour appliquer le théorème 3.

On précisera dans chaque cas le nombre de vecteurs maximum qu'on peut avoir dans la décomposition de x pour être sûr que le Matching Pursuit retrouve cette décomposition.

3.3 Tests numériques

On se pose maintenant la question de savoir si ces théorèmes 2 et 3 ont des hypothèses optimales.

Plus précisément on cherche à savoir si le Matching Pursuit retrouve une décomposition parcimonieuse du type (3) même si $\text{card}(\mathcal{L})\mu(\mathcal{D}) \geq 1/2$ ou $\max(\text{card}(\mathcal{L}_1), \text{card}(\mathcal{L}_2)) \times \mu(\mathcal{D}) \geq \frac{1}{2}$.

Question 9

1. Écrire une fonction qui permet de générer aléatoirement un vecteur α_s de taille K ayant exactement s coordonnées non nulles ($s \leq K$), chacune de ces coordonnées étant la réalisation d'une variable aléatoire suivant une loi normale.
2. Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ le dictionnaire constitué de la réunion des vecteurs de la base canonique et de la base de Fourier orthonormalisée.

Calculer la décomposition sur les vecteurs du dictionnaire donnée par l'algorithme de Matching-Pursuit du vecteur x calculé à l'aide de $x = D\alpha_s$ où α_s est calculé par la fonction de la question précédente pour différentes valeurs de s .

Y-a-t-il des cas où l'algorithme retrouve bien après convergence α_s alors que l'hypothèse sur le produit $\text{card}(\mathcal{L})\mu(\mathcal{D}) < \frac{1}{2}$ du théorème 2 n'est pas vérifiée ?

3. Adapter l'étude pour voir si la condition $\max(\text{card}(\mathcal{L}_1), \text{card}(\mathcal{L}_2)) \times \mu(\mathcal{D}) < \frac{1}{2}$ du théorème 3 est optimale.

Remarque : Un théorème de Elad et Bruckstein montre que si $s < \sqrt{N}$ alors la décomposition parcimonieuse $x = D\alpha_s$ est la plus parcimonieuse possible et il n'y en a pas d'autre (voir détails en annexe 5.2).

4 Rendu du projet

Le projet consiste en

- Soit un fichier `notebook` codé en Python ou en R
- Soit un compte-rendu tapé en Latex dans un fichier `.pdf` et les codes associés. Dans ce cas il est indispensable que parmi les fichiers informatiques on trouve pour chaque partie pratique un fichier dont le titre permette de comprendre de quoi il s'agit et qu'il suffit d'exécuter pour illustrer informatiquement ce qui est indiqué dans le compte-rendu.

L'objectif du projet est l'étude du Matching-Pursuit et de l'optimalité des théorèmes 2 et 3.

En particulier on n'oubliera pas de détailler et commenter

- la présentation des techniques mathématiques utilisées
- les illustrations numériques de ces techniques.
- les résultats de restauration
- **Pour aller plus loin :** on pourra chercher à faire les exercices facultatifs ou à démontrer les résultats non explicitement demandés.

5 Annexe

5.1 Éléments de démonstration de la proposition 2

La démonstration repose sur l'étude de la matrice de Gram G du système $\{\phi^j, j \in \mathcal{L}\}$, c'est à dire de la matrice dont les coefficients sont $\langle \phi^i, \phi^j \rangle$ pour $i \in \mathcal{L}$ et $j \in \mathcal{L}$.

En effet on sait que G est inversible si et seulement si le système des $\{\phi^j, j \in \mathcal{L}\}$ est un système libre.

Donc montrer que $\{\phi^j, j \in \mathcal{L}\}$ est un système libre revient à montrer que G est inversible.

Montrer que G est inversible revient à montrer qu'elle n'a pas de valeurs propres nulles.

On peut donc étudier les valeurs propres de G . Le théorème suivant (dit de Gerschgorin) permet de conclure.

Théorème 4 Soit A une matrice à coefficients complexes de taille $n \times n$ et de terme général $(a_{ij})_{i,j \leq n-1}$. Pour i variant de 0 à $n-1$ on introduit le disque de Gershgorin correspondant

$$U_i = \{z \in \mathbb{C}, |a_{ii} - z| \leq \sum_{i \neq j} |a_{ij}|\}$$

Toute valeur propre de A appartient à au moins un des disques de Gerschgorin.

5.2 Théorème d'Elad et Bruckstein

Le résultat suivant est démontré dans l'article de M. Elad et A. M Bruckstein [2] "A Generalized Uncertainty Principle and Sparse Representation in Pairs of Bases" et disponible à la page <https://pdfs.semanticscholar.org/10e7/0e16e5a68d52fa2c9d0a452db9ed2f9403aa.pdf>.

Pour ceux qui sont motivés la démonstration est à leur portée (techniques d'algèbre linéaire et optimisation sous contrainte en dimension finie) avec comme toujours un peu de travail.

Théorème 5 Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire de \mathbb{C}^N qui est la réunion d'une base orthonormée \mathcal{B}_1 et d'une base orthonormée \mathcal{B}_2 . Soit $x \in \mathbb{C}^N$ tel que

$$(8) \quad x = \sum_{\ell \in \mathcal{L}} \alpha_\ell \phi^\ell$$

où \mathcal{L} est un sous-ensemble de $\{0, \dots, K-1\}$ et $\alpha_\ell \neq 0$ pour tout $\ell \in \mathcal{L}$. On note α le vecteur de \mathbb{C}^K de coordonnées α_ℓ pour $\ell \in \mathcal{L}$ et 0 sinon.

Si $\text{card}(\mathcal{L})\mu(\mathcal{D}) < 1$ alors la décomposition (8) est la plus parcimonieuse possible et il n'y en a pas d'autre.

Autrement dit si on suppose que $x \in \mathbb{C}^N$ vérifie aussi $x = \sum_{\ell' \in \mathcal{L}'} \beta_{\ell'} \phi^{\ell'}$ avec

- $\mathcal{L}' \subset \{0, \dots, K-1\}$
- $\beta_{\ell'} \neq 0$ pour tout $\ell' \in \mathcal{L}'$
- β le vecteur de \mathbb{C}^K de coordonnées $\beta_{\ell'}$ pour $\ell' \in \mathcal{L}'$ et 0 sinon

alors soit $\beta = \alpha$, soit $\text{card}(\mathcal{L}') > \text{card}(\mathcal{L})$.

Pour ce qui concerne la question 3.3 dans ce projet sur le Matching Pursuit, avec le cas qui nous intéresse où le dictionnaire est la réunion de la base canonique et de la base de Fourier, ce résultat signifie que dès que vous créez un vecteur α_s qui contient exactement s coordonnées non nulles avec $s < \sqrt{N}$ alors le vecteur $x = D\alpha_s$ n'a aucune autre représentation parcimonieuse possible dans le dictionnaire \mathcal{D} que α_s .

Si vous prenez $s \geq \sqrt{N}$ vous ne pouvez plus garantir qu'il n'y en a pas d'autre. En effet un autre résultat de Donoho et Huo [1] vous montre que dans le cas de notre dictionnaire \mathcal{D} on ne peut pas faire mieux que le théorème de Elad et Bruckstein. En effet si $s = \sqrt{N}$ Donoho et Huo montrent que pour certains $N = p^2$ où p est entier il est possible de trouver des vecteurs x qui ont deux représentations parcimonieuses différentes avec chacune exactement \sqrt{N} coefficients non nuls dans le dictionnaire \mathcal{D} (ici formé de la réunion de la base de Fourier et de Dirac). Là encore les démonstrations de leur résultat devraient vous être accessibles si vous êtes motivé-e et avez envie de les travailler.

Références

- [1] D. L. Donoho et X. Huo : Uncertainty principles and ideal atomic decomposition. IEEE Transactions on Information Theory, 47 :2845–2862, 2001.
- [2] M. Elad et A. Bruckstein : A generalized uncertainty principle and sparse representation in pairs of bases. IEEE Transactions On Information Theory, 48 :2558–2567, 2002.
- [3] R. Gribonval et P. Vandergheynst : On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. IEEE Transactions on Information Theory 52, 255-261,2006.
- [4] S. Mallat : A wavelet tour of signal processing. Academic Press, (2008)