CY Tech

Victor VITCHEFF
Antoine LE DU
Yasmine CHENOUF

Le compressive sensing : un moyen de compression, d'envoi et de decompression de données

Mini projet réalisé dans le cadre du module de mathématiques en deuxième année d'ingénieur

Sous la direction de Mr BOURHATTAS

CY TECH
Campus du parc
95000 Cergy Pontoise

Table des matières

Introduction

De plus en plus de données sont crées et ont besoin d'être taitées autour de nous. Dans ce projet, on s'intéresse à comment en envoyer à un utilisateur. Plus précisément, on s'intéresse à des signaux. La taille de ces signaux est souvent très grande, le but sera donc de simplifier ces signaux pour les envoyer à un utilisateur. Pour cela, il existe déjà des outils comme la réduction de dimension avec l'ACP. Dans ce module, on s'intéresse à un nouveau module appelé le compressive sensing. Soit un signal x de taille n. Le but sera de le réduire en le représentant comme $x = D.\alpha$. α est appelé représentation parcimonieuse de x et D est le dictionnaire. Chaque vecteur admet une représentation parcimonieuse dans le dictionnaire D. La représentation parcimonieuse est un vecteur contenant une majorité et quelques coefficients. Il est de la taille d'une ligne du dictionnaire D. On dit que ce vecteur est s-parcimonieux, cela signifie qu'il possède s coefficients non nuls.

Dans ce projet, on présentera d'abord des algorithmes et des solutions pour trouver une représentation parcimonieuse d'un vecteur. Puis dans un second temps, on appliquera un algorithme nommé k-svd qui permet de trouver le dictionnaire le plus adapté pour un signal donné en entrée. Enfin, nous introduirons les matrices de mesures pour essayer de se concentrer que sur certaines mesures du signal d'entrée X. On appliquera ces 3 algorithmes sur un ensemble de signaux données dans un fichier ".xlsx".

I Représentation Parcimonieuse

I.1 Dictionnaire orthonormée

On suppose ici que l'on a des dictionnaires orthonormées, c'est à dire que pour : $X = D.\alpha$ avec X signal d'entrée, D le dictionnaire et α la représentation parcimonieuse de x, on sait que $\alpha = (D^*)^T.X$. Cela nous donne donc directement une représentation parcimonieuse. A noter que D^* représente la matrice conjugué de D et D^T est la matrice transposée de D. Un dictionnaire orthonormée que l'on a déjà étudié en cours est le dictionnaire de la transformée en cosinus discrète (DCT). On note C cette matrice et elle est définie ainsi :

$$\forall (i,j) \in \mathbf{N} \times \mathbf{N} \text{ et } (i,j) \in [0; N-1], \text{avec N la taille de la matrice } C$$
$$C[i,j] = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } i = 0 \\ \sqrt{\frac{2}{N}} \times \cos(\frac{2j+1}{2N} i\pi) & \text{sinon.} \end{cases}$$

Une autre matrice, cette fois ci, avec des valeurs complexes, est la matrice de la transformée de Fourier noté F et définie ainsi :

$$\forall (p,q) \in \mathbf{N} \times \mathbf{N} \text{ et } (p,q) \in [0; N-1], \text{avec N la taille de la matrice } F \text{ :}$$
$$F[p,q] = \frac{1}{\sqrt{N}} \exp(\frac{-2pq}{N} i\pi)$$

On se propose maintenant de prouver que la matrice F est bien une base orthonormée. Pour cela il suffit de prouver que F est une matrice orthogonale. Il faut prouver que :

$$\forall (p,q) \in \mathbf{N} \times \mathbf{N} \text{ et } (p,q) \in [0; N-1]$$
$$< F_{.,p}, F_{.,q} >= \begin{cases} 1 & \text{si } p = q \\ 0 & \text{si} p \neq q \end{cases}$$

Pour cela on rappelle que :

$$\forall (p,q) \in \mathbf{N} \times \mathbf{N} \text{ et } (p,q) \in [0; N-1]$$
$$< F_{.,p}, F_{.,q} >= (F^*_{.,q})^T \times F_{.,p}$$

On choisit $p, q \in \mathbf{N}$ et on commence la démonstration :

$$< F_{.,p}, F_{.,q} >= (F^*_{.,q})^T \times F_{.,p}$$

$$< F_{.,p}, F_{.,q} >= \sum_{k=0}^{N-1} F^*_{k,p} \times F_{k,q}$$

$$< F_{.,p}, F_{.,q} >= \sum_{k=0}^{N-1} \frac{1}{\sqrt{N}} \exp(\frac{2pk}{N} i\pi) \frac{1}{\sqrt{N}} \exp(\frac{-2qk}{N} i\pi)$$

car

$$(\exp(\frac{-2qk}{N} i\pi))^* = \exp(\frac{2qk}{N} i\pi)$$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N} \sum_{k=0}^{N-1} \exp(\frac{2ki\pi}{N}(p-q))$$

Si $p = q$ alors $p - q = 0$ et

$$\frac{1}{N}\sum_{k=0}^{N-1}\exp(0) = \frac{1}{N}N$$

Donc

$$< F_{.,p}, F_{.,q} >= 1$$

Si $p \neq q$ alors :

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}\sum_{k=0}^{N-1}(\exp(\frac{2i\pi}{N}(p-q)))^k$$

Il s'agit d'une série géométrique de raison $r = \exp(\frac{2i\pi}{N}(p-q)$ avec $r \neq 0$ car $p - q \neq 0$
Donc on a pour $p \neq q$:

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}(\frac{1-r^N}{1-r})$$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}(\frac{1-(\exp(\frac{2i\pi}{N}(p-q))^N}{1-\exp(\frac{2i\pi}{N}(p-q)})$$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}(\frac{1-\exp(\frac{2i\pi}{N}(p-q)N}{1-\exp(\frac{2i\pi}{N}(p-q)})$$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}(\frac{1-\exp(2i\pi(p-q))}{1-\exp(\frac{2i\pi}{N}(p-q)})$$

Or $2(p-q) \in \mathbf{Z}$ et on sait que :
$\exp(-2i\pi k) = 1, \forall k \in \mathbf{Z}$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N}(\frac{1-1}{1-\exp(\frac{2i\pi}{N}(p-q)})$$

$$< F_{.,p}, F_{.,q} >= \frac{1}{N} \times 0 = 0, \forall p \neq q$$

Donc finalement on a bien :

$$\forall (p,q) \in \mathbf{N} \times \mathbf{N} \text{ et } (p,q) \in [0; N-1]$$
$$< F_{.,p}, F_{.,q} >= \begin{cases} 1 & \text{si } p = q \\ 0 & \text{si} p \neq q \end{cases}$$

Finalement, comme F est orthogonale, alors c'est une base orthonormée.

On crée maintenant un signal et on se propose de trouver sa représentation parcimonieuse, d'abord dans la base C puis ensuite dans la base F. Pour se faire, on crée un signal X de taille $N = 500$ comme le produit de 2 signaux sinusoïdal (un venant de la fonction cosinus et l'autre de la fonction sinus). Une fois ces signaux obtenus, on calcule leur représentation parcimonieuse grâce à la conjuguée-transposée de C (ou de F). J'ai calculé la norme entre la différence de X, le signal, et le dictionnaire multiplié par la représentation parcimonieuse (que je vais nommer erreur). C'est à dire, en notant ϵ l'erreur :

$\epsilon = ||X - D.\alpha||$, α la représentation parcimonieuse de X et $D = C$ ou F en fonction de la base choisie.

Je réalise le tableau pour vous montrer les résultats suivants :

	ϵ en base C	ϵ en base F
Xobtenu avec $(D^*)^T$	0.145	1.172×10^{-2}

Il semble donc que la représentation en base F de la représentation parcimonieuse est plus proche de la réalité en base F qu'en base C. La précision en base C est supérieure à 0.1. Elle n'est pas précise mais reste cependant très négligable. Je pense que cela est du au premier point du signal, difficile à approcher car la première ligne de la base est une constante.

I.2 Utilisation de l'algorithme OMP

On se propose maintenant d'utiliser l'algorithme OMP. Celui ci séléctionne les atomes du dictionnaire D les plus corrélés avec le résidu R qui est actualisé à chaque itération. Avec le même signal utilisé dans la sous section au dessus, on va chercher à trouver une nouvelle représentation parcimonieuse du signal X. L'algorithme OMP s'arrête lorsque la précision entrée en paramètre est inférieure à la norme du résidu ou si le nombre d'itérations k est supérieure au nombre d'itérations maximal. Ici, on choisit kmax=100. Je réalise un tableau pour vous montrer tous les résultats suivants (l'erreur ϵ est la même que dans la partie du dessus) :

	ϵ en base C	ϵ en base F
Xobtenu avec $(D^*)^T$	0.145	1.172×10^{-2}
X obtenu avec OMP	0.614	1.401
nombre d'itérations pour OMP	100	100

On remarque ici que la précision dans les 2 bases est supérieure à 0.1 dans les 2 cas. Cela signifie donc que les dictionnaires en base orthornomée n'ont pas l'air adapté à l'algorithme OMP. On décide d'utiliser cette fois ci des signaux de taille 100. Et on réalise le même tableau, on obtient :

	ϵ en base C	ϵ en base F
Xobtenu avec $(D^*)^T$	0.174	1.609×10^{-13}
X obtenu avec OMP	0.010	4.164
Nombre d'itérations pour OMP	91	100

Ce tableau nous montre que la précision de la l'algorithme OMP pour les dictionnaires orthonormées est très faible. En effet, ceux ci ne reflètent pas du tout la réalité. L'algorithme OMP cherche une corrélation entre les atomes de D (donc les colonnes) et le résidu qui correspond au début à X. Mais comme D est une base cela signifie que la corrélation est très peu existante. En fait le critère d'arrêt sera le nombre d'itérations. Pour que l'lagorithme soit précis, il faut choisir k très proche de la taille de signal et cela rend l'algorithme OMP très long pour les dictionnaires orthornomées.

On se propose maintenant de voir une nouvelle méthode de calcul de représentation parcimonieuse avec l'algorithme noté IRLS.

I.3 Algorithme IRLS

On se propose maintenant d'utiliser un nouvel algortihme qui se nomme IRLS. On remplace le problème ci dessous.

$$(P_0) : min||\alpha||_0 \text{ sous contrainte } D\alpha = x$$

Il est remplacé par une autre pseudo-norme et le problème est noté par :

$$(P_p) : min||\alpha||_p \text{ sous contrainte } D\alpha = x$$

On transforme ensuite cette norme en un problème des moindres carrées :

$$(P_2) : min||W\alpha||_2 \text{ sous contrainte} D\alpha = x \text{ et } W \text{ matrice diagonale}$$

On se propose de montrer que les 2 problèmes sont équivalents. Pour cela, on choisit :

$$\sum_{i=1}^{k} |\alpha_i|^p = \sum_{i=1}^{k} w_i \alpha_i^2 \text{ avec } w_i = (|\alpha_i|^2)^{\frac{p}{2}-1}$$

On se propose aussi de trouver les coefficients w_i. Pour cela, on part du problème P_p et on arrivera ensuite au problème P_2. On a :

$$(P_p) : min||\alpha||_p \text{ sous contrainte } D\alpha = x$$

$$(P_p) : min(\sum_{i=1}^{k} |\alpha_i|^p)^{\frac{1}{p}} \text{ sous contrainte } D\alpha = x$$

D'après ce qu'on a posé au dessus :

$$(P_p) : min(\sum_{i=1}^{k} w_i \alpha_i^2)^{\frac{1}{p}} \text{ sous contrainte } D\alpha = x$$

$$(P_p) : min(\sum_{i=1}^{k} (|\alpha_i|^2)^{\frac{p}{2}-1} \alpha_i^2)^{\frac{1}{p}} \text{ sous contrainte } D\alpha = x$$

Si on pose $p = 2$ on a bien une équivalence avec le problème P_p. En effet :

$$(P_p) : min(\sum_{i=1}^{k} (|\alpha_i|^2)^{\frac{2}{2}-1} \alpha_i^2)^{\frac{1}{2}} \text{ sous contrainte } D\alpha = x$$

$$(P_p) : min(\sum_{i=1}^{k} |\alpha_i|^2 |\alpha_i|^{-2} \alpha_i^2)^{\frac{1}{2}} \text{ sous contrainte } D\alpha = x$$

$$(P_p) : min(\sum_{i=1}^{k} \alpha_i^2)^{\frac{1}{2}} \text{ sous contrainte } D\alpha = x$$

Ce qui signifie donc que :

$$(P_p) \Leftrightarrow (P_2)$$

On voit donc que W est une matrice diagonale ou les coefficients sont égales à $w_i = |\alpha_i|$ avec p=2. On va maintenant chercher une solution au problème P_2. Pour cela, on cherche α tel que :

$$x = DW\alpha$$

Donc en utilisant la pseudo inverse, on obtient :

$$\alpha = (DW)^{+} x$$

$$\alpha = (DW)^T (DW(DW)^T)^{-1} x$$

$$\alpha = W^T D^T (DWW^T D^T)^{-1} x$$

Comme on veut minimiser $W\alpha$ et non α, on multiplie par W devant à droite et à gauche et on a :

$$W\alpha = WW^T D^T (DWW^T D^T)^{-1} x$$

Si on pose $WW^T = Q$ alors Q est une matrice diagonale et on a bien :

$$W\alpha = QD^T (DQD^T)^{-1} x$$

A noter que cette solution existe si et seulement si WW^T est inversible.

L'algorithme suivant est ensuite implémenter dans le langage python et le fichier "code.py". On se propose maintenant de passer à l'lalgorithme du k-SVD et plus précisément à l'apprentissage de dictionnaire.

II Apprentissage de dictionnaire

On implémente dans le code python l'algorithme k-SVD qui permet d'optimiser un dictionnaire. A la fin de cette implémentation, on utilise l'algorithme pour créer un dictionnaire de 100 atomes sur les 108 signaux qui nous sont données. Pour calculer, la représentation parcimonieuse de chaque signal, on utilise l'algorithme OMP et l'algorithme IRLS.

Les résultats seront présentées dans la section suivante où je parlerai aussi des matrices de mesure et je ferai la décomposition de 3 signaux en représentation parcimonieuse puis je vais les reconstruire grâce au dictionnaire. Il sera aussi intéressant de mesurer l'erreur pour chacun des signaux.

III Matrice de mesure et Compressive Sensing

Pour finir, on choisit la matrice de mesure puis on calcule les 3 signaux pour chacun leur représentation parcimonieuse et on les compare avec eux mêmes au début.

Conclusion

Pour conclure, je pense que ce module de compressive sensing nous a beaucoup apporté. Nous n'avons malheuresement pas eu le temps de le terminer même si je pense que nous avons passé beaucoup de temps dessus. Nous avons été perdu dans le problème de dimension lors de la programmation en python. Nous avons trouvé cela intéressant de pouvoir programmer une nouvelle façon de chercher la représentation parcimonieuse notamment avec l'algorithme IRLS.

Pour finir, nous avons trouvé ce module intéressant et on pourra ensuite se demander s'il est possible de pouvoir mêler le compressive sensing à du chiffrement de fichiers. En effet, au lieu de se concentrer sur tous les pixels d'une image par exemple, on peut uniquement se concentrer sur la représentation parcimonieuse de ces pixels que l'on chiffre ensuite.