

TP Optimisation et Echantillonnage compressif.

Déroulement et notation du TP

1 Introduction

Récemment (début années 2006-présent), de nouveaux concepts et théorèmes ont été développés et risquent de révolutionner à relativement court terme la fabrication de certains appareils de mesure numériques (microphones, imageurs, analyseurs de spectres,...). Ces nouvelles techniques sont couramment appelées échantillonnage compressif, “compressive sampling” ou encore “compressed sensing”. Nous les décrivons rapidement ci-après.

1.1 Le théorème de Shannon

Aujourd’hui, presque tous les appareils de mesure reposent sur le théorème de Shannon. Celui-ci peut s’énoncer ainsi :

Théorème 1. *Soit $g : \mathbb{R} \rightarrow \mathbb{R}$ une fonction de $L^2(\mathbb{R})$. Si sa transformée de Fourier \hat{g} a un support contenu dans l’intervalle $[-f_M, f_M]$, alors en l’échantillonnant à une fréquence d’échantillonnage $f_e > 2f_M$, on peut la reconstruire exactement en utilisant une interpolation en sinus cardinal.*

Ce théorème est illustré sur la figure 1.1. Les instruments de mesures qui reposent sur ce théorème sont donc construits suivant le principe :

Filtre passe-bas \rightarrow Echantillonnage $> 2f_M \rightarrow$ Interpolation sinc

Pour beaucoup d’applications, ce principe présente deux défauts majeurs :

- Les signaux sont rarement naturellement à spectre borné, et on perd donc l’information haute-fréquence en effectuant un filtrage passe-bas.
- Pour beaucoup de signaux, il faut choisir une très haute fréquence d’échantillonnage pour obtenir un résultat satisfaisant. Ceci implique que les données à stocker ont une taille très importante et qu’il faut les compresser après coup (exemple : jpeg).

1.2 L’échantillonnage compressif

L’idée sous-jacente à l’échantillonnage compressif est de réaliser la compression dès l’acquisition. Nous détaillons ci-dessous une de ses nombreuses variantes.

1.2.1 Le principe

Soit $\Psi = [\psi_1, \psi_2, \dots, \psi_n]$ une base orthogonale de \mathbb{R}^n . Tout vecteur $u \in \mathbb{R}^n$ s’écrit comme une combinaison linéaire de la forme :

$$u = \Psi x = \sum_{i=1}^n x_i \psi_i \quad (1)$$

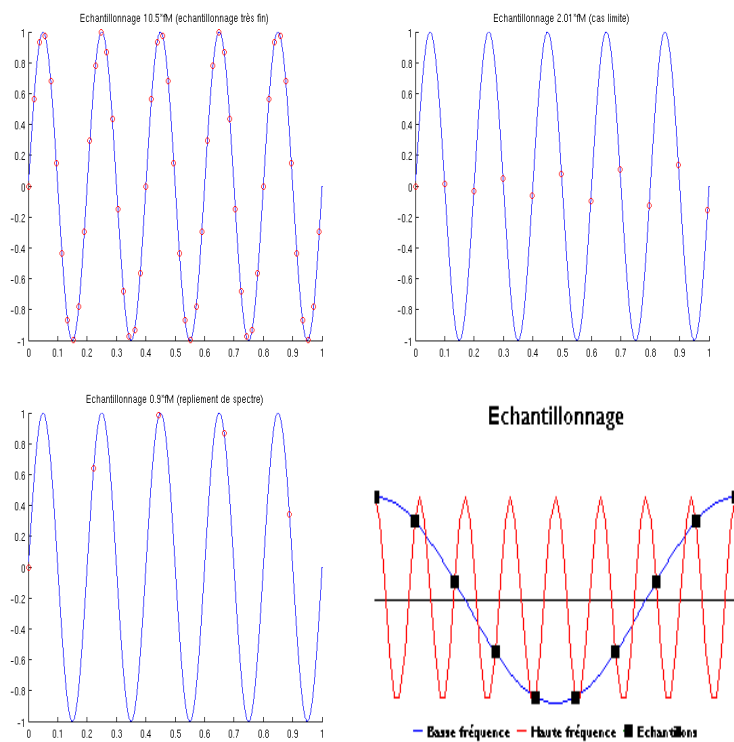


FIGURE 1 – Illustration du théorème de Shannon. Sur la figure en bas à droite, on voit qu'un mauvais échantillonnage ne permet pas d'identifier certaines fréquences. On parle de repliement de spectre.

où les vecteurs $\psi_i \in \mathbb{R}^n$, $1 \leq i \leq m$ sont généralement appelés “atomes” et les valeurs $x_i \in \mathbb{R}$ sont les coefficients de x dans la base Ψ . En notant $x = [x_1; \dots; x_n]$ (notations Matlab), on peut écrire $x = \Psi^* u$ puisque Ψ est orthogonale.

L'échantillonnage compressif repose sur l'hypothèse suivante : le signal x que l'on souhaite mesurer est parcimonieux, c'est-à-dire que la majorité des coefficients x_i dans (1) sont nuls ou encore que

$$|\{x_i \neq 0, 1 \leq i \leq n\}| \ll n.$$

On va voir que cette hypothèse permet - dans certains cas - de réduire drastiquement le nombre de mesures par rapport au théorème de Shannon. La contrepartie est le besoin de résoudre un problème d'optimisation pour reconstruire la donnée. L'objectif de ce TP est de résoudre le problème d'optimisation résultant.

Le principe de l'acquisition du signal x est le suivant :

1. On effectue un *petit nombre* $m \ll n$ de produits scalaires du vecteur x inconnu $y_i = \langle a_i, x \rangle$, $1 \leq i \leq m$. Les vecteurs a_i dépendent du système d'acquisition. Cette opération peut s'écrire sous la forme condensée :

$$y = A^* x \quad \text{où} \quad y = [y_1; \dots; y_m] \quad \text{et} \quad A = [a_1, a_2, \dots, a_m].$$

2. On reconstruit le signal x en résolvant le problème contraint suivant :

$$\text{Trouver } x^* \in \arg \min_{x \in \mathbb{R}^n, A^* x = y} \|x\|_0 \quad (2)$$

où $\|\cdot\|_0$ est la fonction de comptage aussi appelée improprement “norme” l^0 . Celle-ci est définie par :

$$\|x\|_0 = |\{x_i \neq 0, 1 \leq i \leq m\}|.$$

Reformulée en français, l'idée est la suivante : on cherche le signal le plus parcimonieux, parmi les signaux qui peuvent donner lieu aux mesures y .

3. Après avoir trouvé x^* , on recouvre une approximation du vecteur u en calculant $u^* = \Psi x^*$.

1.2.2 Simplification du problème d'optimisation

Le problème (2) est un problème combinatoire NP-complet, ce qui signifie que trouver x peut demander un temps exponentiel en fonction de n , la dimension du signal. Pour le résoudre en pratique, il est souvent remplacé par :

$$\text{Trouver } x^* \in \arg \min_{x \in \mathbb{R}^n, A^* x = y} \|x\|_1 \quad (3)$$

où $\|x\|_1 = \sum_{i=1}^n |x_i|$ est la norme l^1 de x . On peut montrer dans certains cas que les solutions de (2) et de (3) sont identiques. Noter que le problème (3) est un problème de *programmation linéaire*.

Un appareil de mesure n'étant jamais parfait, il est impossible de mesurer exactement $y_i = \langle a_i, x \rangle$. Le vecteur y est bruité et la contrainte $A^* x = y$ est trop forte. Elle est donc généralement relaxée et le problème devient :

$$\text{Trouver } x^* \in \arg \min_{x \in \mathbb{R}^n} \|x\|_1 + \frac{\sigma}{2} \|A^* x - y\|_2^2 \quad (4)$$

Si σ tend vers $+\infty$, la solution du problème (4) tend vers une solution du problème (3). Nous allons résoudre les problèmes (3) et (4) dans ce TP.

Pour conclure cette introduction à l'échantillonnage compressif, notons que de façon similaire au théorème de Shannon, on dispose d'une condition de reconstruction exacte :

Théorème 2. *Supposons que :*

- $u = \sum_{i=1}^n x_i \psi_i \in \mathbb{R}^n$ avec $\|x\|_0 = s$.
- $F = [f_1, \dots, f_n]$ et $\Psi = [\psi_1, \dots, \psi_n]$ sont des matrices orthogonales.
- On tire m vecteurs de la base F de façon indépendante, suivant une même loi de probabilité $p_i \propto \|a_i\|_\infty$.
- On note $A^* = [a_1^*; \dots; a_m^*]$ la matrice d'échantillonnage ainsi obtenue.

Sous ces conditions, si

$$m \geq C \left(\sum_{i=1}^n \|a_i\|_\infty^2 \right) s \log(n), \quad (5)$$

alors, la solution x^ du problème (3) correspond exactement au vecteur x avec une très grande probabilité.*

La quantité $\kappa(A) = \sum_{i=1}^n \|a_i\|_\infty^2$ est comprise entre 1 et n pour des matrices A orthogonales. Par exemple, si A est la transformée de Fourier discrète, $\kappa(A) = 1$ car toutes les colonnes de la matrice A ont un module égal à $1/\sqrt{n}$. L'inégalité (5) nous indique alors qu'il suffit d'un nombre de mesures proportionnel à $s \log(n)$ pour reconstruire x exactement. Ce résultat est important, car il indique que seule la dimension *intrinsèque* du signal compte pour l'échantillonnage. Dans beaucoup d'applications $s \log(n) \ll n$, et il faut donc bien moins de mesures qu'avec le théorème de Shannon.

Si au contraire A est la matrice identité, $\kappa(A) = n$. Dans ce cas, l'échantillonnage compressé n'a aucun intérêt.

Il existe de plus des résultats indiquant que la méthode est *robuste* au bruit.

2 Mise en place du TP

2.1 Installation

Pour ce TP, on aura besoin de 2 toolboxes extérieures.

- Pour commencer le TP, télécharger la *wavelet toolbox* de Rice. Elle se trouve ici : <https://github.com/ricedsp/rwt>. Installer la toolbox en tapant :
`rwt-master/compile.m`
- Télécharger le logiciel CVX ici <http://cvxr.com/cvx/download/>. Installer le logiciel en tapant :
`cvx/cvx_setup.m`.

Si ces deux étapes se sont bien passées, vous êtes prêts à affronter le TP.

2.2 Le problème

On s'intéresse à un problème pratique d'échantillonnage d'images en IRM. Dans ce cas, la matrice Ψ peut être choisie comme une transformée en ondelettes orthogonale. La matrice F correspond à la transformée de Fourier discrète

(orthogonale elle aussi). Effectuer une mesure $\langle f_i, u \rangle = \langle f_i, \Psi x \rangle = \langle \Psi^* f_i, x \rangle$ revient à évaluer un coefficient de Fourier discret de l'image. Notez qu'on identifie les images (2D) comme des vecteurs de \mathbb{R}^n plutôt que des matrices de taille $\sqrt{n} \times \sqrt{n}$.

En pratique, il est impossible de stocker les matrices F et Ψ pour des images de taille raisonnable¹. En effet, pour une petite image de 10^6 pixels, ces matrices contiennent 10^{12} éléments. Si les matrices sont codées avec 8 octets par coefficients, ceci représente 8 Téra-octets par matrice. Il ne faut donc jamais stocker ces matrices, mais utiliser les transformées rapides de Fourier et en ondelettes. La transformée de Fourier rapide 2D est appelée `fft2`. Son inverse est `ifft2`. La transformée en ondelettes rapide 2D est appelée `mdwt`. Son inverse est `midwt`. Pour tester ces transformées, vous pouvez lancer le script `TP_MiseEnPlace.m`.

Vous pouvez notamment essayer de changer le taux de compression et les images pour gagner un peu d'intuition.

3 Un peu de théorie

Soit $x \in \mathbb{R}^n$ une décomposition en ondelettes d'une image $u \in \mathbb{R}^n$. Pour échantillonner u on va prendre quelques-uns de ses coefficients de Fourier, au hasard, suivant la procédure décrite dans le théorème 2. Soit I l'ensemble des coefficients de cardinal m à garder dans F^*u . La sélection des coefficients de Fourier à préserver peut être effectuée de manière matricielle. Soit $(e_i)_{1 \leq i \leq n}$ les vecteurs de la base canonique. On considère la matrice $M = (e_i)_{i \in I} \in \mathbb{R}^{n \times m}$. Le vecteur $y = M^* F^* \Psi x \in \mathbb{R}^m$ contient tous les coefficients de Fourier sélectionnés de u . Par la suite, on note $A^* = M^* F^* \Psi \in \mathbb{R}^{m \times n}$.

Les questions suivantes sont importantes pour la suite du TP.

- Calculer le gradient de l'application $Q : x \mapsto \frac{1}{2} \|Ax - y\|_2^2$.
- Calculer la constante de Lipschitz de ∇Q .
- La transformée de Fourier discrète 2D et la transformée en ondelettes sont unitaires. Quelle est la pseudo-inverse de la matrice A ?
- Calculer l'opérateur proximal associé à la norme l^1 . Soit $\tau > 0$, celui-ci est défini par :

$$\begin{aligned} \text{prox}_{\tau \|\cdot\|_1}(x_0) &= \text{shrink}_\tau(x_0) \\ &= \arg \min_{x \in \mathbb{R}^n} \tau \|x\|_1 + \frac{1}{2} \|x - x_0\|_2^2. \end{aligned}$$

Shrink signifie seuillage en anglais. On parle de l'opérateur de seuillage doux, car il ramène les valeurs à 0.

- Calculer l'opérateur proximal associé à la fonction :

$$\chi_X(x) = \begin{cases} 0 & \text{si } A^*x = y \\ +\infty & \text{sinon.} \end{cases}$$

L'opérateur prox_{χ_X} est la projection orthogonale de x sur X .

1. On sera tout de même obligés de stocker ces matrices pour utiliser CVX, pour de toutes petites images.

4 Points intérieurs et CVX

L'objectif de cette partie est de calculer la solution avec des routines standard reposant sur les méthodes de points intérieurs. Vous pouvez tester le programme CVX ou - si la toolbox d'optimisation de Matlab est disponible - avec la fonction Quadprog.

4.1 Travail à réaliser

1. Utiliser le package `cvx` ou la fonction `quadprog` pour résoudre (3) et/ou (4). Note : vous aurez besoin de construire la matrice A explicitement.
2. Comparer cette méthode aux autres méthodes que vous aurez implémentées.

5 Méthodes de gradient

Dans cette partie, on s'intéresse à la résolution du problème (4). On note $f_1(x) = \|x\|_1$ et $f_2(x) = \frac{\sigma}{2} \|A^*x - y\|_2^2$. Ainsi le problème à résoudre devient :

$$\min_{x \in \mathbb{R}^n} f(x) = f_1(x) + f_2(x).$$

5.1 Itération proximale

On rappelle que la descente proximale s'écrit :

- Choisir $\tau \in]0, \frac{2}{L}[$ où L est la constante de Lipschitz de ∇f_2 .
- Choisir $x^{(0)} \in \mathbb{R}^n$.
- $x^{(k+1)} = \text{prox}_{\tau f_1}(x^{(k)} - \tau \nabla f_2(x^{(k)}))$.

On peut montrer que la suite $(x^{(k)})_{k \in \mathbb{N}}$ converge vers x^* avec un taux de convergence de type :

$$f(x^{(k)}) - f(x^*) = O\left(\frac{L\|x^{(0)} - x^*\|_2^2}{k}\right).$$

5.2 Itération proximale accélérée

On rappelle que la descente proximale accélérée s'écrit :

- Choisir $\tau = \frac{1}{L}$ où L est la constante de Lipschitz de ∇f_2 .
- Choisir $x^{(0)} \in \mathbb{R}^n$ et poser $y^{(0)} = x^{(0)}$.
- $x^{(k)} = \text{prox}_{\tau f_1}(y^{(k)} - \tau \nabla f_2(y^{(k)}))$.
- $y^{(k)} = x^{(k)} + \frac{k-1}{k+2}(x^{(k)} - x^{(k-1)})$.

On peut montrer que la suite $(x^{(k)})_{k \in \mathbb{N}}$ converge vers x^* avec un taux de convergence de type :

$$f(x^{(k)}) - f(x^*) = O\left(\frac{L\|x^{(0)} - x^*\|_2^2}{k^2}\right).$$

5.3 Travail à réaliser

1. Implémenter la descente proximale et la tester.
2. Implémenter la descente proximale accélérée et la tester.
3. Evaluer les taux de convergence.
4. Comparer ces méthodes aux autres méthodes que vous aurez implémentées.

6 Méthodes de type Douglas-Rachford / ADMM

6.1 Principe

On note

$$f(x) = f_1(x) + f_2(x)$$

avec $f_1(x) = \|x\|_1$ et $f_2(x) = \chi_X(x)$. On rappelle que l'algorithme de Douglas-Rachford s'écrit sous la forme :

- Choisir $t > 0$ et $z^{(0)} \in \mathbb{R}^n$.
- $x^{(k)} = \text{prox}_{t f_1}(z^{(k-1)})$.
- $y^{(k)} = \text{prox}_{t f_2}(2x^{(k)} - z^{(k-1)})$.
- $z^{(k)} = z^{(k-1)} + y^{(k)} - x^{(k)}$.

Le paramètre t est une constante positive de scaling. On peut montrer que la suite des itérées $(x^{(k)})_{k \in \mathbb{N}}$ converge vers une solution x^* . De plus, sous certaines conditions techniques, la convergence est linéaire.

6.2 Travail à réaliser

1. Implémenter cet algorithme pour résoudre le problème (3) et/ou le problème (4).
2. Tester la méthode en changeant le paramètre t .
3. Evaluer son taux de convergence et le comparer aux autres méthodes.