Problèmes inverses et déconvolution

Ch. Dossal

Novembre 2008

1 Introduction

Nous proposons dans ce TD, de faire un survol de différentes approches de résolution de problèmes inverses linéaires et de nous concentrer un peu plus spécifiquement sur la déconvolution. Soit $A \in \mathcal{M}_{n,p}(\mathbb{R})$ une matrice réprésentant une application linéaire de $\mathbb{R}^p$ dans $\mathbb{R}^n$. Soit $y \in \mathbb{R}^p$ appartenant où non à l'image $Im(A)$ de A. On sait que $y = Ax + w$ est proche de l'image d'un certain x par A. Comment déterminer x tel que $Ax = y$ ou au moins $\|Ax - y\|$ est petit (pour une norme $\|\cdot\|$ à déterminer) ?
Si $y = Ax$ et si les colonnes $(a_i)_{i\leqslant p}$ forment une famille libre alors on peut retrouver x de manière fiable et certaine en inversant le système $y = Ax$.
En revanche si $y = Ax + w$ avec w non nul et aléaoire ou si les colonnes $(a_i)_{i\leqslant p}$ sont liées, ou si le système issu de A est mal conditionné retrouver x numériquement de manière certaine est impossible.
On doit donc trouver des stratégies qui proposent des estimateurs $\tilde{x}$ calculés à partir de y qui sont le plus proche possible du x inconnu recherché. La meilleure stratégie dépend de A et des informations dont on dispose sur x et sur w. Parfois, certaines informations ou *a priori* sont difficiles à exploiter directement algorithmiquement. Il faut alors proposer des solutions alternatives en approchant des problèmes justes mais complexes par des problèmes simplifiés mais résolubles.

2 Maximum A Posteriori (MAP)

Le Maximum A Posteriori (MAP) est un point de vue statistique qui englobe tout un ensemble de techniques, de Wiener à la minimisation l_1 en passant par le Matching Pursuit et la SVD. L'idée du MAP repose sur une question simple :
Parmi tous les x possibles qui expliquent mes données, quel est le $\tilde{x}$ le plus probable ?
La réponse apportée par le MAP est la suivante :
Le $\tilde{x}$ le plus probable est celui qui correspond au maximum de la densité de la loi de x conditionnellement aux observations y.
Faisons dès maintenant deux remarques :

- Pour calculer ce maximum, il faut connaitre les lois de x et de w ou au mimimum disposer d'un moyen de déterminer le maximum de la loi de x conditionnellent à y. Ce n'est pas toujours le cas. Dans nombre de situations on ne suppose même pas que x est une variable aléatoire mais nous verrons qu'un certain nombre de méthodes d'inversion sont totalement équivalentes à une approche MAP. C'est le cas de la régularisation de Tikhonov et de toutes les approches par minimisation ℓ_1.
- Considérer que le $\tilde{x}$ le plus probable est celui qui correspond au maximum de la densité $f(x|y)$ est un parti pris. On peut envisager d'autre choix parfois plus pertinent. On peut

par exemple, déterminer l'estimateur qui minimise un certain risque. L'approche MAP est une possibilité qui permet dans de nombreuses situations d'apporter une solution toute à fait raisonnable sans être l'arme absolue.

2.1 Rappels en 1D

Nous proposons d'abord une analyse en dimension 1 car tous les calculs d'optimisation peuvent être réalisée avec une simple dérivation. Les calculs en dimensions supérieures sont en fait analogue.
On considère Le problème monodimensionnel suivant :
Soit a un réel connu et

$$y = ax + w \tag{1}$$

où w est une réalisation d'une variable aléatoire réelle centrée de variance σ^2 indépendante de x.
Comment estimer x à partir de y ?
Si on cherche un estimateur $\tilde{x}$ de la forme $\tilde{x} = \lambda y$, comment choisir λ de manière optimale ?
Le lambda qui minimise le risque $E(\|x - \tilde{x}\|^2)$, (l'espérance étant calculée par rapport au bruit) vaut $\lambda = \dfrac{ax^2}{a^2x^2 + \sigma^2} = \dfrac{a}{a^2 + \frac{\sigma^2}{x^2}}$.
Le problème de ce λ optimal est qu'il dépend de x. Si on suppose maintenant que x est une variable aléatoire centrée de variance σ_1^2 alors le λ optimal minimisant le risque quadratique $r(\lambda)$ calculé cette fois ci relativement au bruit w et à x vaut : $\lambda = \dfrac{a\sigma_1^2}{a^2\sigma_1^2 + \sigma^2} = \dfrac{a}{a^2 + \frac{\sigma^2}{\sigma_1^2}}$.

Remarque : Quand σ tend vers 0, λ tend vers a^{-1}.
Prenons maintenant une approche MAP.
On considère le même problème (1) mais cette fois ci on suppose que x est une variable aléatoire gaussienne centrée de variance σ_1^2 et w une variable gaussienne, centrée de variance σ_2^2 indépendante de x.
On a ainsi $w = y - ax$. La densité $f(x, w)$ de la loi du couple (x, w) est ainsi

$$f(x, w) = Ce^{-\frac{\|x\|^2}{2\sigma_1^2}} e^{-\frac{\|w\|^2}{2\sigma_2^2}} \tag{2}$$

où C est une constante connue.
Si on conditionne la loi du couple (x, w) par rapport à $y = ax + w$ on obtient

$$f(x|y) = Ce^{-\frac{\|x\|^2}{2\sigma_1^2}} e^{-\frac{\|y-ax\|^2}{2\sigma_2^2}} \tag{3}$$

On en déduit que le $\tilde{x}$ réalisant le MAP est le minimiseur de la fonctionnelle suivante :

$$F(x) = \|ax - y\|^2 + \frac{\sigma_2^2}{\sigma_1^2}\|x\|_2^2. \tag{4}$$

Le minimiseur $\tilde{x}$ vaut donc $\tilde{x} = \lambda y$ avec

$$\lambda = \frac{a\sigma_1^2}{a^2\sigma_1^2 + \sigma_2^2} = \frac{a}{a^2 + \frac{\sigma_2^2}{\sigma_1^2}}.$$

L'estimateur est ainsi le même que celui obtenu par minimisation du risque quadratique.

2.2 Généralisation aux dimensions supérieures

Si on considère maintenant le problème général

$$y = Ax + w \tag{5}$$

où $A \in \mathcal{M}_{n,p}(\mathbb{R})$ et que l'on considère que x suit une loi gaussienne centrée de matrice de covariance K_1 et que w est une réalisation d'une variable aléatoire indépendante de x et suivant une loi gaussienne centrée de matrice de covariance K_2 alors l'estimateur MAP $\tilde{x}$ est alors le minimiseur de la fonctionnelle

$$F(x) = x^t K_1 x + (y - Ax)^t K_2 (y - Ax) \tag{6}$$

Ce qui donne, si w est un bruit blanc gaussien de variance σ_2^2 et si x est un bruit blanc gaussien de variance σ_1^2 :

$$F(x) = \frac{1}{2\sigma_2^2}\|y - Ax\|_2^2 + \frac{1}{2\sigma_1^2}\|x\|_2^2$$

Si maintenant on fait l'hypothèse que w est une réalisation d'un bruit blanc gaussien de variance σ_2^2 et que x est une réalisation d'une variable aléatoire de loi Laplacienne, ie de densité $f(x) = C_\gamma e^{-\gamma\|x\|_1}$, l'estimateur MAP $\tilde{x}$ est alors défini comme le minimiseur de

$$F(x) = \frac{1}{2\sigma_2^2}\|y - Ax\|^2 + \gamma\|x\|_1 \tag{7}$$

D'une manière générale, dans nombre de situations on peut écrire l'estimateur MAP comme le minimiseur d'une fonctionelle. Si cette dernière est strictement convexe on sait que l'estimateur est ainsi défini de manière unique. C'est toujours le cas par exemple pour (6) et très souvent pour (7).
Nous verrons par la suite que les deux types de fonctionnelles (6) et (7) peuvent être utilisées dans les problèmes inverses.

3 Régularisation de Tikhonov et SVD

On se place à nouveau dans le cas général où $y = Ax + w$, où $A \in \mathcal{M}_{n,p}(\mathbb{R})$ et w est non nul. Cette fois ci, on ne suppose pas particulièrement que x est une réalisation d'un processus aléatoire.
Une manière d'estimer x à partir de y est de chercher un $\tilde{x}$ qui a une énergie faible et qui a une image proche de y. On propose donc de minimiser la fonctionnelle suivante dépendant d'un paramètre T.

$$F_T(x) = \|y - Ax\|^2 + T^2\|x\|^2. \tag{8}$$

Le paramètre T est un paramètre de régularisation. Plus T est grand, plus l'énergie du minimiseur $\tilde{x}(T)$ est faible, plus T est petit, plus l'image $A\tilde{x}$ de $\tilde{x}$ par A est proche de y. En pratique, on choisit la valeur de T en fonction de l'amplitude supposée de w.
Pour une erreur $\|y - Ax\|^2$ donnée, $\tilde{x}$ est le vecteur d'énergie minimale.
On a vu dans la section précédente que ce type d'approche pouvait être vu comme un MAP.
Un des avantages de cette formulation est qu'il est possible de donner une solution analytique au problème de minimisation (8) grâce à la SVD.
En effet, la fonctionnelle F_T est différentiable en tout point et son gradient vaut

$$grad(F_T)(x) = -2A^t(y - Ax) + 2T^2x \tag{9}$$

Si on note λ_k les valeurs singulières de A, e_k **les vecteurs singuliers associés de norme 1** et $u_k = Ae_k$ on a

$$\tilde{x}(T) = \sum_k \frac{\langle y, u_k \rangle}{\lambda_k^2 + T^2} e_k \tag{10}$$

En effet si on note $\tilde{x}(T)$ le minimiseur de (8), on a par définition des e_k,

$$\tilde{x}(T) = \sum_{k \leqslant p} \langle \tilde{x}(T), e_k \rangle e_k = \sum_{k \leqslant p} x_k e_k$$

De plus par définition de $\tilde{x}(T)$, pour tout $k \leqslant p$, $\langle grad(F)(\tilde{x}(T)), e_k \rangle = 0$ et donc pour tout $k \leqslant p$

$$\begin{aligned} \langle A^t Ax + T^2 \tilde{x}(T), e_k \rangle &= \langle A^t y, e_k \rangle \\ \langle x, A^t Ae_k \rangle + T^2 \langle \tilde{x}(T), e_k \rangle &= \langle y, Ae_k \rangle \\ \lambda_k^2 x_k + T^2 x_k &= \langle y, u_k \rangle. \end{aligned}$$

d'où, pour tout $k \leqslant p$

$$x_k = \frac{\langle y, u_k \rangle}{\lambda_k^2 + T^2}$$

ce qui définit de manière unique le minimiseur $\tilde{x}(T)$.
On remarque que la limite de $\tilde{x}(T)$ quand T tend vers 0 correpond bien à l'inversion directe par SVD : $\tilde{x}(0) = \sum_{\lambda_k \neq 0} \frac{\langle y, u_k \rangle}{\lambda_k^2} e_k$ qui effectue une inversion sur l'orthogonal du noyau de A et qui fournit ainsi l'antécédent de la projection de y sur l'image de A de norme ℓ_2 minimale.

3.1 Exemple de régularisation par SVD et limites de l'approche

Nous allons considérer un exemple simple d'inversion utilisant la SVD et de la régularistion de Tikhonov et montrer l'intérêt et la limite de la méthode.
On note $(e_k)_{k \leqslant 2n}$ la base canonique de $\mathbb{R}^{2n}$, v_k la base canonique de $\mathbb{R}^{\ltimes}$ et A l'application linéaire de $\mathbb{R}^{2n}$ dans $\mathbb{R}^n$ définie par

$$Ae_k = \begin{cases} u_k = \frac{v_k}{k} & \text{si } k \leqslant n \\ 0 & \text{sinon.} \end{cases} \tag{11}$$

Un calcul rapide montre que les e_k sont des vecteurs singuliers de A associés aux valeurs singulières $\lambda_k = \frac{1}{k}$ si $k \leqslant n$ et $\lambda_k = 0$ si $k > n$.
Remarque : A deux changements de base prêts, l'un sur $\mathbb{R}^p$ et l'autre sur $\mathbb{R}^n$, on peut se ramener toute application linéaire B à une telle application linéaire A ?. On n'a pas nécessairement $p = 2n$, les valeurs singulières ne valent pas nécessairement $\frac{1}{k}$ mais le principe est le même.
L'application linéaire A est clairement non inversible.
Soit $y = (y_k)_{k \leqslant n} \in \mathbb{R}^n$, quel est l'antécédent de y proposé par la SVD ?
L'inversion par SVD propose un inverse en utilisant uniquement les e_k tq Ae_k est non nul de manière à minimiser l'énergie ici on aura donc :

$$\tilde{x}_{SVD} = \sum_{k \leqslant n} k y_k e_k \tag{12}$$

La régularisation de Tikhonov, minimiseur de (8), fournit une version *régularisée* de la version SVD standart :

$$\tilde{x}_{Tik} = \sum_{k \leqslant n} \frac{k y_k}{1 + T^2 k^2} e_k \tag{13}$$

Dans l'une comme dans l'autre version, l'estimé $\tilde{x}$ appartient toujours à l'espace vectoriel engendré par les $(e_k)_{k\leqslant n}$.
Dans la version régularisée, les composantes associées aux faibles valeurs singulières (k grand), sont fortement diminuées de manière à limiter l'amplitude du bruit qui lui est amplifié d'un facteur k par l'inversion sur ces composantes. Si on cherche à limiter l'énergie de l'estimateur il est naturel d'avoir des composantes nulles sur une BO du noyau. En revanche on peut avoir un *a priori* sur le signal qu'on recherche qui lie les composantes du noyau et les composantes *visibles*. C'est par exemple le cas si on suppose dans la situation présente que le x recherché a peu de coefficients de Fourier. La partie visible renseigne alors sur la partie *cachée dans le noyau*.
Si l'*a priori* dont on dispose est du type :
le vecteur x que je recherche est une combinaison linéaire d'un petit nombre d'élément d'un dictionnaire $\mathcal{D}$,
alors on peut souvent faire mieux que la SVD ou les régularition *type Tikhonov* qui minimise une énergie ℓ_2.
Nous allons voir qu'on peut faire d'autant mieux que le dictionnaire $\mathcal{D}$ est décorrélé de la famille $(e_k)_{k\leqslant p}$.

3.2 SVD et Regularisation de Tikhonov en pratique

Pour toutes les applications pratiques nous allons considérer un opérateur A de convolution circulaire par un filtre h et un bruit gaussien et blanc. Dans ce cas, la base e_k des vecteurs singuliers est la base de Fourier et les vecteurs u_k sont aussi de la forme $\hat{h}(k)e_k$ avec éventuellement $\hat{h}(k) = 0$ pour certaines valeurs de k.
Le filtre h que nous utiliserons est une gaussienne centrée :

```
>>h=fftshift(gausswin(1024,100));
```

Nous utiliserons également ces 5 vecteurs de référence :

```
>>S1=MakeSignal('Piece-Regular',1024);

>>S2=MakeSignal('Blocks',1024);

>>S3=MakeSignal('HeaviSine',1024);

>>S4=zeros(1024,1);
>>S4(300)=1;
>>S5=S4;
>>S4(325)=1.
>>S5(315)=1;
```

1. Ecrire une fonction **DeconvolutionSVD**

   `Srec=DeconvolutionSVD(S,h)`

   qui prend en entrée un vecteur S et un filtre h et qui effectue une déconvolution par SVD. Quel problème numérique se pose tout de suite ?
2. Faire des tests avec les signaux tests sans ajout bruit.
3. Faire les même tests avec ajout d'un bruit *raisonnable*. Par exemple un bruit blanc de variance 1 pour $S3$ est raisonnable.
   Qu'obervez vous ?
4. Ecrire une fonction **Tikhonov**

`function Srec=Tikhonov(S,h,T)`

qui minimise la fonctionnelle (8) par SVD.

5. Réaliser des essais sur les signaux tests. Pour un niveau de bruit σ donné, tracer la courbe du psnr de reconstruction en fonction de T et déterminer le T optimal.
6. Comment évolue T en fonction de σ ?

4 Matching Pursuit pour l'inversion

Le Matching Pursuit dans sa version première ou dans sa version orthogonale (OMP) est une manière d'effectuer une inversion en général et une déconvolution ene particulier.
L'idée est la suivante : Si $x = \sum_{i\in I} x_k e_k$ s'exprime comme une combinaison linéaire d'un petit nombre de vecteurs e_k alors $Ax = \sum_{i\in I} x_k d_k$ où $d_k = Ae_k$, pour tout k. Ainsi, on peut décomposer (ou approcher) un vecteur y dans la famille des u_k. On obtient alors une famille de coefficients $(x_i)_{i\in I}$ et on construit un estimateur de x à partir de ces coefficients $(x_i)_{i\in I}$ et des vecteurs e_k.
L'algorithme se décomose donc en trois étape :
- Création du dictionnaire D des d_k qui dépend de la famille $(e_k)_k$.
- OMP ou MP sur des observations y à l'aide du dictionnaire D.
- Reconstruction de l'estimateur $\tilde{x}$ à partir des coefficients obtenus.

7. Ecrire une fonction **DicoOndelettes**

   `function D=DicoOndelettes(qmf,h)`

   qui génère la matrice D dot les colonnes sont les images d'une base d'ondelettes définie par le filtre qmf. On considèrera une base d'ondelettes jusqu'à l'échelle $L = 3$.
8. Ecrire une fonction **DeconvOndelettes**

   `function Srec=DeconvDico(y,D,d)`

   qui prend entrée un vecteur y d'observation, une matrice D d et qui reconstruit un vecteur $Srec$ en utilisant un OMP sur le dictionnaire D en sélectionnat d atomes.
9. Faire des essais sur les 3 premiers vecteurs de référence avec l'ondelettes de Harr et de Daubechies 4.
   Afficher les reconstruction ainsi que la courbe du psnr en fonction de d. Comparer visuellement et en terme de psnr l'approche OMP-Ondelettes et l'approche SVD pour les valeurs optimale des paramètres pour différents niveaux de bruit.
10. Proposer une variante invariante par translation de cette méthode d'inversion.
11. Pour les deux derniers signaux qui ne sont pas *creux* en ondelettes mais directement dans la base canonique on n'utilise pas un dictionnaire à partir d'une base d'ondelettes.
12. Ecrire une fonction **DicoDirac**

    `function D=DicoDirac(h)`

    qui construit une matrice D formée des images des vecteurs de la base canonique par la convolution par h.
13. Ecrire une fonction **DeconvDirac**

    `function Srec=DeconvDirac(y,D,d)`

    qui effectue une dévoncolution grâce à un OMP sur les observations y en utilisant un dictionnaire D en d étapes.

14. Comparer les résultats obtenus sur $S4$ et $S5$ et ceux obtenus grâce à la SVD. Dans un premeir temps sans bruit, puis avec.
15. Faites des tests avec différents niveaux de bruits et d'autres vecteurs x comportant d Diracs d'amplitude 1 placés aléatoirement, en faisant varier d.
16. Pouvez vous expliquer ce qui différencie les vecteurs x qui marchent (ceux où on retrouve le signal original correctement) et ceux qui ne fonctionne pas du tout (où l'algorithme MP fait n'importe quoi) ?

5 Minimisation ℓ_1 pour l'inversion

Normalement on doit constater que la déconvolution de vecteurs *creux* (en anglais on dit *sparse*) en Dirac produit de bon résulats ou pas en fonction de la distance entre le Diracs. D'une manière plus générale, l'inversion par OMP fonctionne correctement si les vecteurs colonnes $(d_i)_{i\in I} = (Ae_i)_{i\in I}$ de la matrice D qui interviennent dans la décomposition de $y = Ax + w$ sont bien différents. Dans le cadre de la convolution, deux Diracs proches ont des images proches, c'est cela qui fait planter les méthodes gloutonnes telles que l'OMP.
Si les colonnes de D sont trop proches et que le bruit w est important, on ne pourra de toute façon pas réaliser une bonne inversion mais dans un certain nombre de cas on peut mieux faire que le l'OMP. C'est que nous proposons de voir avec la minimisation ℓ_1.

6 Minimisation ℓ_1 pour l'inversion

D'abord nous allons définir ce que nous appelons minimisation ℓ_1. Plus précisément nous allons considérer deux problèmes de minimisation :

$$\min \|x\|_1 \text{ sous la contrainte } y = Ax \tag{14}$$

et

$$\min \frac{1}{2}\|y - Ax\|_2^2 + \gamma\|x\|_1 \tag{15}$$

Le premier peut être utile dans la situation où on observe des données $y = Ax$ sans bruit. La seconde permet de traiter aussi le cas bruité. Le paramètre de relaxation γ sert à pénaliser la normr ℓ_1 de x. Plus γ est grand, plus la norme $\|x\|_1$ coûte cher et donc plus la solution $x(\gamma)$ de (15) aura une norme ℓ_1 faible et plus $Ax(\gamma)$ sera éloigné de y. A l'inverse plus γ est petit plus la norme $\|x(\gamma)\|_1$ est grande et plus l'erreur $\|y - Ax\|_2^2$ est petite. En pratique, on choisit en fonction du niveau de bruit.
Dans les deux cas on cherche à approcher y par une combinaison de vecteurs de A dont la norme ℓ_1 est faible. On peut montrer que dans un grand nombre de situations les solutions de (14) et de (15) ont un petit nombre de composantes non nulles. Ainsi la minimisation ℓ_1 est un moyen de retrouver une solution *creuse* x au problème d'inversion.
La minimisation de ces deux fonctionelles a pour but *d'inverser* A en cherchant à chaque fois, l'antécédant de y qui a le moins de composantes non nulles dans une certaine famille de fonction, ici une BON (Base OrthoNormée).
Important : La minimisation ℓ_1 ne fournit pas toujours la solution la plus creuse, les conditions pour que ce soit le cas ne sont pas encore parfaitement connues en 2008. Il existe toute une vaste littérature scientifique sur le sujet. Ce que vous devez avoir en tête c'est que c'est que c'est une méthode à tester si vous avez une hypothèse sur la parcimonie du vecteur recherché. Il faudra toujours

prendre la solution avec des pincettes.
Il est important de noter que si on modifie la BON dans la quelle on calcule la norme ℓ_1 de x, on modifie la fonctionelle de (14) et donc la solution associée. C'est une différence essentielle avec la minimisation ℓ_2 qui est toujours la même dans toutes les BON.
Il existe de nombreuses méthodes de résolution de ces deux problèmes. Nous en utiliserons une qui a le mérite de se programmer facilement en matlab et qui est assez stable numériquement. Son principal défaut est sa lenteur.
Nous utiliserons deux points théoriques que nous ne justifierons pas ici :
– Pour presque toutes les matrices A, les deux problèmes de minimisation (14) et (15) admettent chacun une unique solution.
– La solution $x(\gamma)$ de (15) tend vers la solution de (14) quand γ tend vers 0.
Ainsi pour résoudre (14) on pourra utiliser sa forme relaxée et faire tendre le paramètre γ vers 0. Ajoutons que (15) est un problème d'optimisation convexe.
L'algorithme en question a été proposé sous cette forme par Daubechies-Defrise et De Moll en 2003.
L'algoritme est itératif, à chaque étape on défini un vecteur x_n tq $F(x_n)$, où $F(x) = \frac{1}{2}\|y - Ax\|_2^2 + \gamma\|x\|_1$ décroît strictement. Chaque itération se décompose en deux sous-itérations :

(a) Une étape de gradient qui minimise $\|y - Ax\|_2^2$:
`x_{n+1/2}=x_n+ \mu A^t(y-Ax_{n})=x_n+\mu A^ty-A^tAx_n`
où $\mu < 1/\|A^tA\|$.

(b) Une étape de seuillage doux des coeffcients de $x_{n+1/2}$ à un seuil μ pour faire diminuer la norme l_1 de x.

On peut définir un critère d'arrêt quelconque.
Si on dispose de l'application A sous forme matricielle, l'algorithme se rédige très simplement.

17. Ecrire une fonction **Minimisationl1**

```
function Srec=Minimisationl1(y,A,nb,gamma,init)
```

Qui prend en entrée un vecteur y, une matrice A, un nombre d'itérations nb, un paramètre de relaxation gamma et un vecteur *init* qui servira d'initialisation à l'algoritme et qui renvoie un vecteur *Srec* qui sera proche de la soltion de (15).
Quelques conseils :
– Ecrire préalablement une fonction **SeuillageDoux**.
– Caculer en début de programme le vecteur A^ty ainsi que la matrice A^tA.
– Dans un premier temps, afficher à chaque étape la valeur de $F(x_n) = 1/2\|y - Ax_n\|_2^2 + \gamma\|x_n\|_1$.
– μ doit être suffisamment petit pour l'algorithme converge. Dans un premier temps, choisissez un gamma de l'ordre de 0.1 par exemple.
Normalement $F(x_n)$ doit décroitre strictement, si vous observez que F croît à partir d'un moment, diminuer la valeur de μ. Si vous observez que $F(x_n)$ décroît toujours mais trop lentement vous pouvez essayer d'augmenter la valeur de μ.

18. Ecrire un programme **MatriceConvolution** qui construit la matrice de convolution associée à un filtre h.
19. Tester **Minimisationl1** en utilisant une matrice A obtenue à partir d'un filtre gaussien et des données y, image de 2 Diracs plus ou moins espacés comme dans les deux derniers signaux de références.

20. Si on considère d'abord des données non bruitées, quel est l'espace minimal entre Deux Diracs quel a minimisation ℓ_1 permet de retrouver ?
21. Comparer cet espacement minimal avec celui nécessaire à la reconstruction par OMP.
22. Tester la robustesse au bruit de cette méthode en jouant sur le paramètre gamma.
23. Comparer les résultats de la minimisation ℓ_1 avec les méthodes de filtrages *optimales* dans ce cas où les données sont *creuses en Dirac*

6.1 Minimisation l1 et BON

Dans la plupart des situations, on sait que le vecteur x que l'on recherche est *creux dans une base orthonormée ou plus généralement dans une famille* $\mathcal{F}$ mais pas spécifiquement dans la base canonique.
Nous allons voir comment on peut appliquer l'algorithme précédent dans le cas où l'on cherche une solution x creuse dans une base $\mathcal{B}$ quelconque. On définit z comme les coefficients de x dans la base $\mathcal{B}$ et W la matrice de changement de base : $z = Wx$. On a donc $Ax = AW^{-1}z$. On peut donc effectuer le même algorithme en effectuant la minimisation (15) sur z avec la matrice AW^{-1} et les mêmes observation y. On obtient alors en fin de compte des coefficients $z^\star$ qui minimisent la fonctionelle (15), il faut donc ensuite calculer $x^\star = Az^\star$.
Remarque :
Si W est une changement de bases entre deux bases orthonormées, la matrice W vérifie : $W^{-1} = W^t$ ce qui peut simplifier les calculs.

24. Ecrire une fonction **Minimisationl1Ondelettes** qui effectue une inversion d'une application linéaire A par minimisation ℓ_1 dans une base d'ondelettes quelconque.
25. Faite des tests avec A filtrage gaussien et les signaux de référence $S1$, $S2$ et $S3$. Comparer avec les méthodes linéaires et avec l'OMP.
Notons qu'on peut évidemment utiliser une variante invariante par translation. Dans ce cas on prendra soin d'initialiser l'algorithme par l'estimateur courant de x (ou z selon les cas).

6.2 Minimisation ℓ_1, sans matrice A explicite

Pour terminer nous nous intéressons au cas où l'application linéaire A n'est pas donnéee sous forme de matrice. Comme par exemple dans le cas de fft ou d'une transformée en ondelettes. En théorie on peut toujours construire une matrice, en pratique ce peut être trop couteux numériquement. Pour les images par exemple, on ne peut pas utiliser de matrice. En revanche on peut parfois utiliser l'application linéaire directement pour peu qu'on puisse appliquer l'opérateur adjoint.
ExempleS :

- Si A est une convolution, A est autoadjoint, donc $A = A^t$ donc on peut remplacer chaque multiplication par A et par A^t par une convolution en passant par une fft ce qui est souvent plus rapide.
- Si $A = F \circ W$, est une composition entre un changement de bases orthonormées et un filtrage (Ondelettes par exemple comme dans le acs traité plus haut).
  L'application $A^t = W^{-1} \circ F$ correspond alors à une composition entre le même filtrage F et le changement de bases réciproque.
- Si on veut travailler on veut effectuer une inversion par minimisation ℓ_1 dans une union de bases orthonormées (si on veut faire directement de l'inavriance par translation) on

peut également faire de même. Par exemple le problème de la séparation de sources peut se poser ainsi : On cherche à écrire un vecteur $y\mathbb{R}^n$ comme une combinaison linéaire de $2n$ vecteurs formant deux bases orthogonales $\mathcal{B}_1$ et $\mathcal{B}_2$. Dans ce cas l'application linéaire A est l'identité mais on a un *changement de bases* W qui envoie un vecteur x sur l'union de ces coordonnées dans deux bases orthonormées : on effectue un *changement de bases* $z = Wx$ avec $W = [A_1; A_2]$ où A_1 et A_2 sont des matrices orthogonales. On a ainsi $z = Wx = [A_1x : A_2x]$ qui se calcule avec deux transformées usuelles (fft ou ondelettes par exemple). L'application transposée se calcule également facilement car $A^tz = A_1z_1 + A_2z_2$ avec $z = [z_1; z_2]$.

Avec cette méthode on peut également faire une inversion par minimisation ℓ_1 en ondelettes invariante par translation, où l'invariance par translation est utilisée à chaque étape. On calcule à chaque étape, tous les coefficients d'ondelettes sans sous-échantillonnage.

26. Ecrire une fonction **Minimisationl1OndBis** qui effectue une déconvolution par minimisation en ondelettes sans passer par une forme matricielle qui fournit le même résultat que la fonction utilisant explicitement la matrice.
27. Evaluer le gain en temps par rapport à **Minimisationl1Ondelettes**