Débruitage

Ch. Dossal

Octobre 2008

1 Introduction

Dans ce TD, nous allons étudier différentes méthodes pour supprimer des parasites (on dira le *bruit*) sur des signaux 1D et 2D. Nous considérons que nous avons accès à des observations $y = x + w$, sommes d'un signal x et d'un bruit w. Nous utiliserons un modèle de bruit gaussien. Ce modèle sur le bruit est le modèle le plus courant. C'est le fait de considérer que le bruit est gaussien qui justifie le fait de mesurer l'erreur de reconstruction en norme ℓ_2. Si on fait d'autres hypothèses sur le bruit, par exemple bruit impulsionel ou distribué selon une loi Laplacienne, on peut mesurer l'erreur de reconstruction d'une autre manière.
Nous commençons par l'utilisation de méthodes linéaires. Elles sont dites linéaires car l'opérateur de débruitage est linéaire. On projette les observation sur un espace vectoriel qui ne dépend pas des observations ou on atténue les coefficients de y de manière indépendante de y.
Dans un second temps nous étudions des méthodes non linéaires mises en oeuvre dans des bases d'ondelettes. Ces méthodes peuvent être généralisées à d'autres familles de fonctions.
Nous rappelons qu'un estimateur F est une application qui à une observation y associe un estimé $\tilde{x}$. L'erreur quadratique d'un estimateur F est défini comme l'espérance sur le bruit de l'erreur quadratique $E(\|x - \tilde{x}\|^2)$. Un estimateur est d'autant meilleur que cette erreur est faible.

2 Méthodes linéaires

2.1 Filtre de Wiener

On rappelle que le filtrage de Wiener consiste à multiplier chaque coefficient de Fourier $\hat{y}(k)$ de y par

$$a(k) = \frac{\hat{R}_x(k)}{\hat{R}_x(k) + \hat{R}_w(k)}$$

où $\hat{R}_x(k)$ (resp $\hat{R}_w(k)$) est le $k$ième coefficient de Fourier de la fonction d'autocorellation de x (resp de w) avec $R_f(k) = E[(f(k) - E[f(k)])(f(0) - E[f(0)])]$. Le filtrage de Wiener est une méthode qui a été développée pour séparer le bruit du signal dans la situation spécifique où le bruit et le signal sont des réalisations de processus aléatoires stationnaires dont on connait la loi. Sous ces hypothèses sur x et w, le filtrage de Wiener est optimal.
Faire l'hypothèse que le bruit est aléatoire et stationnaire est tout à fait raisonnable. On fait même souvent l'hypothèse que ce dernier est blanc et gaussien.
Pour le signal c'est différent. Supposer que le signal recherché est la réalisation d'un processus aléatoire n'est pas totalement déraisonnable. En revanche supposer qu'on connait sa loi et que le processus est stationnaire est nettement plus hasardeux. C'est pourquoi en pratique le filtrage de Wiener peut ne pas être très bon.

Comme on ne dispose que d'une seule réalisation du processus aléatoire y, on estime les coefficients $\hat{R}_f(k)$ par $|\hat{f}(k)|^2$.

1. Ecrire un programme de débruitage de Wiener qui prend en entrée le signal bruité par un bruit gaussien (dont on suppose connaître la variance) et le signal original et qui renvoie le signal débruité et le psnr associé.
2. Comment estimer les coefficients du filtre de Wiener si on ne dispose pas du signal original ? Ecrire un programme effectuant un débruitage approchant le filtrage de Wiener mais ne nécessitant pas le signal original.

2.2 Filtre Gaussien

Un autre type de filtrage qui permet d'atténuer le bruit sur une image est le filtre gaussien. La commande matlab qui permet de construire une gaussienne est *gausswin*.

3. Ecrire un programme prenant en entrée un vecteur bruité 1D et un réel positif σ et qui renvoie un vecteur débruité par filtrage d'une gaussienne dont la variance est contrôlé par σ et le psnr associé.
4. Ecrire un programme qui affiche le psnr du signal débruité en fonction de σ et qui détermine le σ optimal.
5. Faire des tests sur différents signaux et différents niveaux de bruit.
6. La valeur optimale de σ dépend-t-elle du niveau de bruit ?
7. Comparer le filtrage gaussien et le filtrage de Wiener.

3 Débruitage en ondelettes

3.1 Représentations parcimonieuses et débruitage

Il existe de multiple manières de *débruiter* un signal. Nous proposons ici d'exploiter le fait que de nombreux signaux, ceux qui sont réguliers par morceaux, sont efficacement représentés en ondelettes pour faire du débruitage. Plus précisément, on va utiliser le théorème suivant :

Théoreme :

Si une fonction f de $L^2(\mathbb{R})$ est C^n par morceaux, alors il existe une base d'ondelettes $\mathcal{B}$ et une constante C telles que la meilleure approximation f_M de f avec M coefficients vérifie

$$\|f - f_M\|^2 \leqslant CM^{-2n} \tag{1}$$

D'une manière générale, si pour une classe de signaux on dispose d'une base orthonormée qui assure une décroissance rapide de l'erreur d'approximation non linéaire, alors on peut utiliser cette base pour réaliser un débruitage efficace. En fait on peut parfois se passer du caractère orthonormé ou même du caractère libre de la famille utilisée.

Comment exploiter les qualités d'approximation d'une base pour le débruitage ?

Nous supposons ici que le bruit est gaussien et blanc. Certains des résultats suivants peuvent être transposés dans le cadre d'un bruit gaussien coloré.

Remaquons d'abord que si b est la réalisation d'un processus aléatoire blanc et gaussien alors dans toute base $\mathcal{B} = (b_i)_{i\in\mathbb{N}}$ les coefficients $b_i = \langle w, b_i\rangle$ du bruit, sont des variables gaussiennes indépendantes et de même loi : **Un bruit blanc est un bruit blanc dans toute base orthonormée.**

Soit $\mathcal{B}$ une base fixée, notons $y_k = \langle y, b_k \rangle$, $x_k = \langle x, b_k \rangle$ et $w_k = \langle w, b_k \rangle$ les coefficients de y, x et w dans la base $\mathcal{B}$. Ainsi $y_k = x_k + w_k$, où w_k est une variable aléatoire gaussienne centrée de variance σ^2.
Une manière simple d'estimer x à partir de y est d'utiliser un estimateur *diagonal*, c'est à dire un estimateur où chaque x_k est estimer uniquement à partir de y_k. On peut montrer que pour un grand nombre de classes de fonctions usuelles, le rique du meilleur estimateur diagonal est proche du risque optimal.
Comment estimer x_k à partir de y_k ?
Si on veut estimer x_k par une atténuation des coeffcients y_k, $F(y_k) = \tilde{x}_k = a_k \times y_k$, la valeur optimale de a_k (**sans connaissance de** w_k) est $a_k = \frac{x_k^2}{x_k^2+\sigma^2}$ comme le montre une simple dérivation sur le paramètre a. L'erreur optimale associée est

$$E(|x_k - \tilde{x}_k|^2) = \frac{\sigma^2 x_k^2}{x_k^2 + \sigma^2} \tag{2}$$

L'erreur de *l'estimateur* diagonal $F_{\mathcal{O}}$ associé est alors

$$E(\|x - \tilde{x}\|^2) = \sum_k \frac{\sigma^2 x_k^2}{x_k^2 + \sigma^2} \tag{3}$$

Un tel estimateur n'est en fait pas vraiment un estimateur car il nécessite comme pour Wiener la connaissance de x. On parle d'estimateur *oracle*. De tels estimateurs sont intéressants car ils peuvent donner des bornes sur ce qu'on peut espérer faire de mieux.
En effet si on construit un estimateur dont le risque est équivalent à celui de l'estimateur oracle, on sait qu'on a obtenu un estimateur asymptotiquement optimal.
Cet estimateur oracle nécessite la connaissance précise des valeurs de x_k. On peut construire un autre estimateur oracle dont le risque est proche de celui de $F_{\mathcal{O}}$ mais qui nécessite moins de connaissance sur x.
Pour cela on considère l'ensemble des estimateurs diagonaux qui sont des projecteurs ie ceux pour qui $a_k \in \{0, 1\}$, $\forall k \in \mathbb{N}$.
Parmi ces estimateurs, un calcul simple montre que le choix optimal est obtenu en prenant $a_k = 1$ si $|x_k| > \sigma$ et $a_k = 0$ sinon. Pour ce second estimateur oracle $F_{\mathcal{P}}$ dit estimateur par projection oracle, un calcul rapide montre que

$$E(|x_k - \tilde{x}_k|^2) = \min(|x_k|^2, \sigma^2). \tag{4}$$

Or un simple calcul montre que pour a et b positifs, on a $\min(a, b) \leqslant \frac{2ab}{a+b}$. En sommant sur tous les indices $k \in \mathbb{N}$, on en déduit que le rique de l'estimateur par projection oracle $F_{\mathcal{P}}$ est majoré par deux fois le risque de $F_{\mathcal{O}}$.
Cet estimateur est intéressant pour au moins trois raisons :

(a) Son risque est optimal, à un facteur 2 près, au risque diagonal oracle optimal ie $r(F_{\mathcal{P}}) \leqslant 2r(F_{\mathcal{O}})$.

(b) Son risque est directement lié aux capacités d'approximation non linéaires de la base $\mathcal{B}$. En effet si on note I l'ensemble des indices i tels que $|x_i| \geqslant \sigma$, on a

$$E(\|x - \tilde{x}\|^2) = E(\|x - y_I\|^2) \tag{5}$$

$$= \sum_{i \in \mathbb{N}} \min(|x_i|^2, \sigma^2) \tag{6}$$

$$= |I|\sigma^2 + \sum_{|x_i| < \sigma} |x_i|^2. \tag{7}$$

$$= |I|\sigma^2 + \|x - x_{|I|}\|^2. \tag{8}$$

Un calcul rapide montre que si $\|x - x_{|I|}\|^2 \leqslant CM^{-n}$ alors il existe $C' > 0$ tel que $\|x - \tilde{x}\|^2 \leqslant C'\sigma^{\frac{4n}{2n+1}}$.

(c) Le risque de cet estimateur oracle peut être approché à un facteur logarithmique près par un estimateur par seuillage de y. On a, en effet le théorème suivant dû à Donoho et Johnstone :
Théoreme :
Soit $x = (x_k)_{k\leqslant N}$ un vecteur de taille N et $w = (w_k)_{k\leqslant N}$, une réalisation d'un bruit blanc gaussien de variance σ^2 et $y = x + w$. On pose $T = \sqrt{2\log_e N}\sigma$. Le risque $r(F_T)$ de l'estimateur F_T par seuillage est lié au risque $r(F_{\mathcal{P}})$ du projecteur oracle par la relation suivante :
$$r(F_T) \leqslant (2\log_e N + 1) \times (r(F_{\mathcal{P}}) + \sigma^2) \tag{9}$$

En résumé
Le risque de l'estimateur de seuillage avec un seuil bien choisi qui dépend de la taille du signal, est borné par le risque du projecteur oracle qui dépend lui même directement de la décroissance d'approximation non linéaire de la base $\mathcal{B}$. Plus l'erreur d'approximation décroit rapidement avec M plus le risque de l'estimateur par seuillage est faible. En particulier, si les signaux étudiés se représentent efficacement en ondelettes, comme c'est le cas pour les signaux réguliers par morceaux en 1D et dans une moindre mesure pour les images naturelles en 2D, un seuillage en ondelettes est un bon estimateur.

3.2 Seuillage simple

Remarque : **Dans toutes les méthodes de seuillage, on ne touche pas aux coefficients d'échelle, on ne seuille que les coefficients d'ondelettes.**

8. Ecrire un programme qui effectue un débruitage par seuillage en ondelettes. Ce programme prend en entrée un signal bruité, un seuil T ,un filtre miroir en quadrature associé à une ondelettes et le signal original (uniquement pour calculer le psnr) et renvoie un signal débruité et le psnr associé.
9. Ecrire un programme qui effectue un débruitage pour plusieurs valeurs du seuil entre σ et 6σ et qui renvoier la courbe des psnr.
10. Faire des tests sur différents vecteurs de taille 1024 en faisant varier le niveau du bruit en utilisant l'ondelette de Daubechies 4.
11. Faire les mêmes tests sur des signaux plus grands et en faisant varier l'ondelette utiliser (essayer Haar et Daubechies 6 par exemple).
12. Ecrire un programme qui estime le risque de l'estimateur par seuillage dans une base d'ondelettes pour différentes valeurs du seuil et qui affiche ce risque en fonction du seuil en faisant une moyenne sur 1000 réalisations du bruit.
13. Le seuil optimal réel dépend t-il de l'ondelette et de la taille des signaux ? Dans la suite on prendra le seuil $T = 3\sigma$.
14. Pour les signaux *Blocks et Piece-Regular*, quelles sont les ondelettes optimales ?
15. Comparer le risque d'un tel estimateur avec le filtrage de Wiener, le filtrage gaussien et un seuillage en base de Fourier.

3.3 Seuillage Doux

Il existe plusieurs variantes du seuillage ou *seuillage dur*. L'une d'entre elles est le seuillage dit *doux*. Le seuillage doux est la fonction suivante :

$$S(x) = \begin{cases} x - T & \text{si } x > T \\ x + T & \text{si } x < -T \\ 0 & \text{sinon.} \end{cases}$$

Pour ce nouveau seuillage on reprend quelques questions de la section précédentes :

16. Ecrire un programme qui effectue un seuillage doux en ondelettes.
17. Ecrire un programme qui affiche le psnr en fonction du seuil.
18. En testant différent signaux, différentes tailles de signaux et différentes ondelettes, déterminer un *seuil optimal* pour le seuillage doux. Celui ci n'est pas en 3σ.
19. Comparer le seuillage doux au seuillage dur. Lequel vous semble le plus efficace ?
    En fait le seuillage doux est plus efficace sur les images.
20. Ecrire les programmes précédents en 2D et comparer le seuillage dur et le seuillage doux chacun à leur seuil optimal, pour différentes images et différentes ondelettes. Vérifier ainsi que le seuillage doux est plus perfromant en 2D que le seuillage dur.

3.4 Invariance par translation

Même si c'est la parcimonie qui est à la base de l'efficacité des bases d'ondelettes pour le débruitage, on a intérêt en pratique à utiliser des familles redondantes. Plus précisément, on va effectuer un moyennage de débruitage dans différentes bases orthonormées, toutes translatées d'une même base d'ondelettes.
Contrairement à la base de Fourier, les bases d'ondelettes ne sont pas invariantes par translation. Si on effectue une translation (circulaire) de K points d'une base d'ondelettes $\mathcal{B}$, on obtient une base différente qui aura peut être des vecteurs communs avec $\mathcal{B}$ mais qui aura aussi des vecteurs différents.

21. Ecrire un programme qui effectue le débruitage d'un vecteur par seuillage dur en moyennant les débruitage obtenus par n translations du vecteurs. Le programme prend en entrée le vecteur bruité, le seuil et le nombre n de translations.
    Comme la méthode est invariante par translation, on parle de seuillage *invariant par translation*.
22. Comparer les résultats obtenus avec le débruitage par seuillage dur *simple* sur divers exemples.
23. Quelle est l'influence du nombre de translations utilisées ?
24. Ecrire un programme équivalent pour le seuillage doux et comparer sur divers exemples, la qualité des résultats avec le seuillage dur *invariant par translation* et avec le seuillage doux *simple*.
    Quelle méthode est la plus efficace ?
25. Réaliser les mêmes codes en 2D et comparer les méthodes.
    Quelle est la méthode la plus efficace ?

3.5 Seuillage par blocks

Une autre variante du seuillage est le seuillage par block.
Dans cette variante, on décide de conserver ou de mettre à 0 un block entier de coefficients si l'énergie totale du block est supérieure à un seuil. **Les blocks sont construits échelle par échelle et on ne seuille jamais les coefficients d'échelle.**
Les résultats obtenus récemment sur le sujet indiquent que la taille de block optimale est $\log_e(N)$ pour des vecteurs de taille N et $\log_e(N) \times \log_e(N)$ pour des images de tailles $N \times N$.
Le seuillage optimal pour le seuillage par block n'est ni un seuillage dur ni un seuillage doux. Le fonction qu'on utilise ici est la suivante :

$$S_{Block}(x) = x \times \max(1 - \frac{T^2}{E^2}, 0) \tag{10}$$

où E^2 est l'énergie moyenne des coefficients du block.
Si $|E| < T$ alors $S_{Block}(x) = 0$ et si $E > T$, l'amplitude de x est légèrement diminuée.
Quand on seuille un block, on calcule E^2 et on modifie les coefficients en utilisant (10).

26. Ecrire un programme sui effectue un seuillage par block en ondelettes d'un vecteur. Le programme prend en entrée un vecteur bruité et un seuil T et renvoie un signal débruité et le psnr associé.
27. Faire des tests avec divers vecteurs de référence et différentes ondelettes. Quel est le seuil optimal ? Dans la suite on prendra le seuil théorique optimal : $T = 4,5 \times \sigma$.
28. Comparer les résultats obtenus avec les précédentes méthodes.
Où classeriez vous le seuillage par block ?
29. Qu'en est-il en 2D ?
30. On peut également faire du seuillage par block *invariant par translation.*
Ecrire un programme effectuant un débruitage par seuillage par block invariant par translation.
31. Comparer cette dernière méthode aux précédentes en 1D et en 2D. Où se classe t-elle ?