TP - Parcimonie

Votre code ainsi que les réponses aux questions 1.b), 1.f), 1.g), 2.a) et 4. sont à envoyer à l'adresse `waldspur@di.ens.fr`, le dimanche 7 juin au plus tard.
La première partie est indépendante des suivantes.

1. Débruitage

a) Écrire une fonction `processus` qui prend en entrée un entier N et un paramètre $p \in [0;1]$ et qui renvoie une réalisation du processus suivant :

$$\forall n \in \{0, ..., N-1\} \qquad X[n] = \epsilon_0 f[n] + \epsilon_1 f[n-1] + ... + \epsilon_{N-1} f[n-(N-1)]$$

où :

- les indices sont considérés modulo N
- les ϵ_k sont des réalisations indépendantes du processus qui vaut 0 avec probabilité $1-p$ et qui suit une loi gaussienne de variance 1 avec probabilité p
- f est la fonction telle que :

$$\begin{aligned} f[n] &= 1 && \text{si } n = 0 \\ &= 1/4 && \text{si } n \in \{1, N-1\} \\ &= 0 && \text{sinon} \end{aligned}$$

b) Montrer, toujours en considérant les indices modulo N, que :

$$\begin{aligned} R_X[n] &= 9p/8 && \text{si } n = 0 \\ &= p/2 && \text{si } n \in \{1, N-1\} \\ &= p/16 && \text{si } n \in \{2, N-2\} \\ &= 0 && \text{sinon} \end{aligned}$$

Calculer $\hat{R}_X$.

c) On considère maintenant le processus $X_n = X + n$, où n est un bruit blanc gaussien de variance σ^2, pour un certain $\sigma \in \mathbb{R}^+$.
Écrire une fonction `denoise_wiener` qui prend en entrée une réalisation de X_n ainsi que les paramètres p et σ et qui renvoie la réalisation débruitée à l'aide du filtrage de Wiener.

d) Afficher sur des figures différentes une réalisation de X pour $N = 200$ et $p = 0,1$, la même réalisation, additionnée d'un bruit de variance σ^2 avec $\sigma = 0,2$ et la version débruitée de cette réalisation.
[Indication : le résultat n'est pas très convaincant. C'est normal.]

e) À l'aide de la fonction `evaluate`, disponible à l'adresse `http://www.di.ens.fr/~waldspurger/tds/14_15_s2/evaluate.m`, afficher sur une quatrième figure l'erreur quadratique moyenne associée à la fonction `denoise_wiener`, en fonction de σ.

La fonction `evaluate` prend en entrée les fonctions `processus` et `denoise_wiener`, les paramètres N et p et (de manière optionnelle) une couleur d'affichage. On l'appellera donc par une instruction de la forme `evaluate(@processus,@denoise_wiener,N,p,'b');` Elle affiche le résultat sur la figure courante.

f) À l'aide de votre cours, expliquer la forme de la courbe obtenue.

g) Écrire une fonction de débruitage non-linéaire `denoise_nonlinear` qui donne un meilleur résultat. Décrire en quelques lignes son principe. Afficher sa courbe d'erreur sur la même figure qu'à la question e), d'une autre couleur.

2. Représentation parcimonieuse

Le but de cette question est d'implémenter un algorithme pour résoudre le problème suivant : un vecteur $x \in \mathbb{R}^n$ est donné, ainsi qu'une famille $b_1, ..., b_m$ de vecteurs de $\mathbb{R}^n$ et on cherche à écrire x comme combinaison linéaire d'un petit nombre de vecteurs b_k. Si B est la matrice de $\mathcal{M}_{n,m}(\mathbb{R})$ dont les colonnes sont $b_1, ..., b_m$, cela revient à dire qu'on cherche y tel que $x \approx By$ et y a le plus de coordonnées nulles possibles.

L'algorithme proposé est le suivant :

- On prend en entrée x, B et un entier K.
- On initialise $r = x$ et $y \in \mathbb{R}^m$ le vecteur nul.
- On effectue l'opération suivante K fois :
  - On cherche l'indice i tel que $\left|\frac{\langle r,b_i\rangle}{||b_i||}\right|$ est maximal (où les b_i sont les colonnes de B).
  - On remplace y_i par $y_i + \frac{\langle r,b_i\rangle}{||b_i||^2}$.
  - On pose $r = x - By$.
- On renvoie y.

a) Montrer que, si la famille $b_1, ..., b_m$ est une base orthonormale de $\mathbb{R}^n$ et si x peut s'écrire comme combinaison linéaire de K vecteurs de la famille, alors le vecteur y renvoyé par l'algorithme sera l'ensemble des coefficients de la décomposition de x sur la base $(b_1, ..., b_m)$.

b) Écrire une fonction `sparse_coeffs` qui implémente cet algorithme.

c) Modifier la fonction pour qu'elle accepte en entrée un ensemble de vecteurs $x_1, ..., x_s$ (sous la forme d'une grande matrice X) et qu'elle renvoie les vecteurs parcimonieux $y_1, ..., y_s$ associés à chacun d'eux.

[Indication : la fonction `sub2ind` pourra permettre d'éviter des boucles non nécessaires.]

3. Base de parcimonie

On suppose maintenant donné un ensemble de vecteurs $x_1, ..., x_s$, regroupé dans une matrice $X \in \mathcal{M}_{n,s}(\mathbb{R})$. On souhaite trouver une famille $b_1, ..., b_m$ de vecteurs de $\mathbb{R}^n$ telle que chaque vecteur x_l peut s'écrire comme une combinaison linéaire d'un petit nombre de vecteurs b_k. Sous forme matricielle, cela revient à dire qu'on cherche à trouver $B \in \mathcal{M}_{n,m}(\mathbb{R})$ et $Y \in \mathcal{M}_{m,s}(\mathbb{R})$ avec le plus de coefficients nuls possible telles que :

$$X \approx BY$$

Ce problème est difficile à résoudre. On propose d'implémenter un algorithme qui optimise alternativement B et Y, qui est simple mais dont le succès n'est pas garanti.

a) Définir une fonction `sparsify` qui prend en entrée X, m et deux entiers K et N_{it}.
b) Initialiser $B \in \mathcal{M}_{n,m}$ comme une matrice aléatoire dont chaque coefficient est une réalisation indépendante d'une loi gaussienne de variance 1. Diviser chaque colonne de B par sa norme.
c) En laissant B fixé, trouver Y à l'aide de la fonction `sparse_coeffs`, à laquelle on passera K comme paramètre.
d) En laissant cette fois Y fixé, remplacer B par une matrice pour laquelle $||X - BY||$ est la plus petite possible. On pourra utiliser sans la justifier l'instruction `B = X * pinv(Y)`.
Diviser chaque colonne de B par sa norme.
[Indication : penser à traiter spécifiquement le cas où B a des colonnes nulles, pour éviter une division par zéro.]
e) Exécuter N_{it} fois les opérations des deux questions précédentes puis renvoyer B.
f) Tester votre fonction en lui passant en entrée une matrice $X \in \mathcal{M}_{5,500}(\mathbb{R})$ dont chaque coefficient vaut 0 avec probabilité $0,9$ et 1 avec probabilité $0,1$. On prendra $m = 5, K = 2, N_{it} = 20$ et on affichera à l'aide de `imagesc` la matrice B obtenue.
[Indication : sauf les quelques fois où l'algorithme échoue, la matrice B doit avoir une allure bien particulière. Si ce n'est pas ce que vous observez, il y a peut-être un problème dans l'implémentation.]

4. Application

Choisissez un ensemble de signaux « naturels » et essayez d'en trouver une base de parcimonie à l'aide de votre fonction `sparsify`. Affichez la base trouvée (sous la forme que vous souhaitez). Décrivez l'ensemble de signaux choisis et commentez la base obtenue. Si cela vous semble pertinent, expliquez votre choix de paramètres m et k, ainsi que le « préprocessing » que vous aurez éventuellement appliqué aux signaux (dans certains cas, il peut éventuellement être judicieux de leur soustraire leur moyenne ou de les diviser par leur norme).

[Indication : un choix possible de signaux est un ensemble de patches de taille 8×8 ou 16×16, extraits aléatoirement de photographies, mais n'hésitez pas à faire preuve de davantage d'originalité.]