TP : « Débruitage » de données avec *a priori* de parcimonie

Des méthodes désormais classiques de « débruitage » de signaux et d'images reposent sur l'hypothèse de *parcimonie*, supposant que l'information utile se représente, de manière exacte ou approchée, par un faible nombre de coefficients exprimés dans un espace adapté. Les transformées classiques opérées sur les signaux et images (Transformée de Fourier, Transformées en Ondelettes, *etc.*) ont pour propriété, entre autres, de concentrer l'information en un faible nombre de points, par exemple à des fins de compression. Le débruitage repose sur le même principe : si, en appliquant une transformation au signal, l'énergie est essentiellement concentrée sur un faible nombre de coefficients, ceux-ci, de valeur élevée, sont moins sensibles au bruit.

Soit le modèle de données $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{\epsilon}$, où $\boldsymbol{\epsilon}$ est un terme de bruit contaminant le signal $\boldsymbol{s}$. On suppose que $\boldsymbol{s}$ admet une représentation parcimonieuse dans un dictionnaire $\mathbf{A}$: $\boldsymbol{s} = \mathbf{A}\boldsymbol{x}$, avec $\boldsymbol{x}$ parcimonieux. Le débruitage consiste alors des deux étapes suivantes :

i) on estime un vecteur parcimonieux $\widehat{\boldsymbol{x}}$ par exemple au sens des moindres carrés :

$$\text{trouver } \widehat{\boldsymbol{x}} \text{ parcimonieux qui minimise } \tfrac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2;$$

ii) on reconstruit ensuite un signal *débruité* $\widehat{\boldsymbol{s}} = \mathbf{A}\widehat{\boldsymbol{x}}$.

Objectifs du TP

La première partie envisage des exemples génériques de débruitage à partir de transformées orthogonales. La deuxième partie met en évidence les limites de ces approches sur des signaux plus complexes, pour lesquels l'introduction de dictionnaires plus riches est nécessaire, rendant l'estimation plus difficile. Dans chaque cas, différentes solutions algorithmiques sont étudiées et comparées.

1 Débruitage de signaux dans des espaces génériques

1.1 Débruitage dans l'espace direct : détection de pics

Soit le modèle simple $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon}$ où $\boldsymbol{x}$ est parcimonieux, *i.e.*, la plupart des coefficients x_n sont nuls. On cherche la meilleure approximation de $\boldsymbol{y}$ à K composantes non nulles. Pour cela, on définit le signal $\widehat{\boldsymbol{x}}_K$, à K composantes non nulles, minimisant l'erreur des moindres carrés :

$$\widehat{\boldsymbol{x}}_K = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x}\|_0 \leq K} J(\boldsymbol{x}), \text{ avec } J(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{x}\|^2 = \sum_{n=1}^{N} (y_n - x_n)^2.$$

La solution $\widehat{\boldsymbol{x}}_K$ s'obtient (cf. TD) en choisissant $\widehat{x}_k = y_k$ pour les K plus grandes valeurs de $|y_k|$ et $\widehat{x}_k = 0$ sinon : c'est une opération de seuillage des données.

Le fichier `sig_peaks.mat` contient un tel signal `y`, ainsi que le signal non-bruité `y_true`. Le bruit est blanc et gaussien : $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ avec $\sigma^2 = 0.1$.

- *Créer une fonction réalisant ce débruitage, prenant en entrée un signal y donné et un nombre K de composantes non-nulles, et renvoyant le signal débruité.*

- *Mettre en œuvre le débruitage en faisant varier K, et comparer les solutions obtenues au vrai signal. Déterminer empiriquement la valeur du seuil sur $|y_k|$ permettant de minimiser l'erreur de reconstruction. Comparer à la valeur donnée dans [**?**] : $\tau = \sigma\sqrt{2 \log N}$, où N est la taille du signal. Commenter la qualité du résultat obtenu. On comparera les rapports signal sur bruit (ou sur erreur) entre $\boldsymbol{x}$ et $\boldsymbol{y}$ d'une part et entre $\boldsymbol{x}$ et $\widehat{\boldsymbol{x}}$ d'autre part, avec :*

$$\mathrm{SNR}_{\mathrm{dB}}(\boldsymbol{x}, \boldsymbol{y}) = 10 \log_{10} \frac{\|\boldsymbol{x}\|^2}{\|\boldsymbol{y} - \boldsymbol{x}\|^2}.$$

1.2 Débruitage dans l'espace de la Transformée en Cosinus Discrète (TCD)

On suppose à présent que le signal recherché est (approximativement) parcimonieux dans la base de TCD : $\boldsymbol{y} = \mathbf{A}_{\cos}^T \boldsymbol{x} + \boldsymbol{\epsilon}$, où $\mathbf{A}_{\cos}^T$ représente l'opération de TCD <u>inverse</u> et $\boldsymbol{x}$ est parcimonieux.

- *Un modèle équivalent s'écrit donc $\mathbf{A}_{cos} \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon}'$ avec $\boldsymbol{\epsilon}' \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$, où $\mathbf{A}_{cos}$ représente la TCD. Créer une fonction réalisant ce débruitage, prenant en entrée un signal y donné et un nombre de composantes non-nulles. On utilisera les fonctions Matlab `dct.m` et `idct.m`.*

- *Le fichier `sig_sparse_tcd.mat` contient un tel signal `y`, ainsi que le signal non-bruité `y_true` et le vrai jeu de coefficients correspondants `x_true`. Vérifier la validité de l'hypothèse de parcimonie. Par une démarche similaire au cas précédent, déterminer la meilleure approximation de $\boldsymbol{y}$. Commenter le résultat et quantifier le gain en termes de rapport signal sur bruit.*

2 Des espaces plus complets pour des signaux plus complexes

L'hypothèse de parcimonie exprimée *dans une base* est une contrainte forte qui peut s'avérer trop réductrice. Le signal du fichier `sig_mix.mat` illustre cet aspect (`y`), correspondant à la superposition d'un signal `y1_true` d'impulsions, d'un signal `y2_true` de TCD parcimonieuse, et de bruit.

- *Visualiser le signal* `y` *et sa TCD. Rechercher sa meilleure approximation dans une base d'impulsions (cf. § 1.1) ou d'une base de TCD (cf. § 1.2) est-il opportun ?*

2.1 Union de bases

Un modèle plus complet consiste donc à rendre compte de la superposition de composantes sous la forme : $\boldsymbol{y} = \boldsymbol{y}_1 + \boldsymbol{y}_2 + \boldsymbol{\epsilon}$, où $\boldsymbol{y}_1$ est parcimonieux dans la base d'impulsions ($\boldsymbol{y}_1 = \boldsymbol{x}_1$, $\boldsymbol{x}_1$ parcimonieux) et $\boldsymbol{y}_2$ est parcimonieux dans la base de TCD ($\boldsymbol{y}_2 = \mathbf{A}_{\cos}^T \boldsymbol{x}_2$ avec $\boldsymbol{x}_2$ parcimonieux) et $\mathbf{A}_{\cos}$ représente la TCD. On a alors :

$$\boldsymbol{y} = [\mathbf{I},\ \mathbf{A}_{\cos}^T] \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} + \boldsymbol{\epsilon} \Leftrightarrow \boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{\epsilon} \ \text{ où } \mathbf{I} \text{ est la matrice identité et } \mathbf{A} = [\mathbf{I},\ \mathbf{A}_{\cos}^T].$$

Le problème est donc d'estimer deux vecteurs parcimonieux $\boldsymbol{x}_1, \boldsymbol{x}_2$ (et de reconstruire les deux signaux $\boldsymbol{y}_1, \boldsymbol{y}_2$) à partir des données. Le *dictionnaire* $\mathbf{A}$ n'est plus orthogonal ni même inversible, possédant deux fois plus de colonnes (nombre d'inconnues) que de lignes (nombre de données).

- *La solution obtenue par seuillage de $\mathbf{A}^T\boldsymbol{y}$ semble-t-elle a priori satisfaisante ?*

Dans ce contexte, la recherche de la solution minimisant $\|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|^2$ à nombre fixé K de composantes non nulles :

$$\|\boldsymbol{y} - \mathbf{W}\boldsymbol{x}\|^2 \ \text{ s. c. } \|\boldsymbol{x}\|_0 \leq K, \tag{1}$$

est un problème d'optimisation combinatoire, généralement abordé de manière sous-optimale, par deux classes de méthodes que nous allons étudier.

2.2 Approche en norme ℓ_1 : optimisation exacte d'un critère relâché

Une approche classique consiste à opter pour une formulation convexe, par la minimisation du critère :

$$\|\boldsymbol{y} - \mathbf{W}\boldsymbol{x}\|^2 + \mu\|\boldsymbol{x}\|_1, \tag{2}$$

où la pénalisation par $\|\boldsymbol{x}\|_1 = \sum_k |x_k|$ induit une solution parcimonieuse [?] et le paramètre μ règle le compromis entre adéquation du modèle ($\|\boldsymbol{y} - \mathbf{W}\boldsymbol{x}\|^2$ faible) et parcimonie ($\|\boldsymbol{x}\|_1$ faible).

De nombreux algorithmes ont été proposés pour l'optimisation de ce type de critère. Le programme `minl1.m` fourni sur le serveur pédagogique en est un, qui renvoie l'optimum du critère selon la syntaxe : `x = minl1(y,W,mu)`.

- *Estimer, à partir de la minimisation de (2), les signaux d'approximation $\boldsymbol{y}_1$ et $\boldsymbol{y}_2$ (en faisant varier le paramètre μ pour obtenir le meilleur résultat). Comparer aux résultats obtenus en préambule de cette section (on pourra calculer les rapports signal sur bruit correspondants).*

2.3 Algorithme glouton : optimisation locale du critère initial

Une autre famille de méthodes repose sur une heuristique gloutonne : on construit une solution parcimonieuse en ajoutant une composante à chaque itération, partant d'une solution nulle. L'algorithme Orthogonal Matching Pursuit (OMP) vu en cours en l'exemple le plus simple.

- *Mettre en œuvre l'algorithme OMP. Comparer les résultats à ceux obtenus par l'approche précédente, en qualité et en temps de calcul nécessaire.*

2.4 Utilisation de dictionnaires de formes plus spécifiques

Le signal du fichier `sig_twoshapes.mat` (version bruitée `y` et non bruitée `y_true`) comprend une impulsion et une forme gaussienne, étendue sur 11 points.

- *Tracer les données et le vrai signal. L'utilisation des bases précédentes est-elle judicieuse pour débruiter le signal ?*

- *Tracer la fonction d'intercorrélation*[1] *entre le signal bruité et la forme gaussienne ayant servi à générer le signal (variable* `gaussienne`*).*

Malgré le bruit, l'information liée à la forme gaussienne est donc bien détectable dans les données. . .

- *Construire le dictionnaire* $\mathbf{A} = [\mathbf{I}, \mathbf{G}]$*, composé de pics et de la forme gaussienne précédente, localisés en tout point de l'axe temporel. Estimer une solution parcimonieuse par optimisation en norme* ℓ_1 *ou par OMP, et conclure.*

1. Pour tracer l'intercorrélation entre x et y, pour des indices de $-N$ à N : `plot(-N:N,xcorr(x,y,N));`