Université Aix-Marseille	2018-2019
Master Mathématiques appliquées et statistiques	2ème année

Représentation parcimonieuse des signaux

TP : Parcimonie et algorithme de seuillage itératif

L'objectif de ce TP est la programmation de l'algorithme de seuillage itératif. Nous nous placerons dans le cas d'un dictionnaire réel et prendrons pour cela la réunion de la base canonique et de la base dite de cosinus discrète.

1 Notations

- On note $\langle .,. \rangle$ le produit scalaire usuel sur $\mathbb{C}^N$ tel que $\langle x, y \rangle = \sum_{n=0}^{N-1} x_n \overline{y_n}$ pour $x = (x_i)_{i=0,\dots N-1} \in \mathbb{C}^N$ et $y = (y_i)_{i=0,\dots N-1} \in \mathbb{C}^N$. On note $|x|_2^2 = \sum_{n=0}^{N-1} |x_n|^2$ pour $x \in \mathbb{C}^N$.
- On note δ^ℓ pour $\ell \in \{0, \dots N-1\}$ le ℓ-ième vecteur de la base canonique de $\mathbb{C}^N$. Il s'agit du vecteur de $\mathbb{C}^N$ tel que pour $n \in \{0, \dots N-1\}$ on a $\delta_n^\ell = 1$ si $n = \ell$ et 0 sinon. La collection de vecteurs $\{\delta^\ell, \ell \in \{0, \dots, N-1\}\}$ constitue la base canonique de $\mathbb{C}^N$.
- la famille de vecteurs $\widetilde{\mathcal{E}} = \{\frac{1}{\sqrt{N}} e^\ell \in \mathbb{C}^N, \ell \in \mathbb{Z}\}$ tels que pour $\ell \in \mathbb{Z}$ et pour $n \in \{0, \dots N-1\}$ la n-ième coordonnée du vecteur e^ℓ s'écrive

$$e_n^\ell = e^{\frac{2i\pi\ell n}{N}}$$

  est appelée base de Fourier orthonormalisée discrète.

2 Algorithme de seuillage itératif

Soit D une matrice $\mathbb{R}^N \times \mathbb{R}^K$ qui pour nous sera ici un dictionnaire de $\mathbb{R}^N$ et x un vecteur de $\mathbb{R}^N$.

Notre objectif est de minimiser à l'aide de l'algorithme de seuillage itératif ISTA la fonctionnelle définie sur $\mathbb{R}^K$

$$F_\lambda(\alpha) = \frac{1}{2}|D\alpha - x|_2^2 + \lambda|\alpha|_1$$

Remarque : cet algorithme peut être utilisé pour une matrice D quelconque et être adapté dans le cas complexe.

2.1 Rappel de l'algorithme

L'algorithme que nous souhaitons étudier numériquement est en effet le suivant. Les deux critères d'arrêt que nous choisissons sont si le nombre d'itération maximal est atteint ou si la différence entre deux itérées est très petite.

Algorithme 1 : Seuillage itératif.

Data : matrice (ici notre dictionnaire) $D = [\phi^0, \dots, \phi^{K-1}] \in \mathbb{R}^{N,K}$, $x \in \mathbb{R}^N$, $\alpha_0 \in \mathbb{R}^K$, $\lambda > 0$ (paramètre de régularisation) $\varepsilon > 0$, $M_{max} \in \mathbb{N}$

initialisation :

$r \leftarrow x$ [Initialisation de la différence entre deux itérées];

$L \leftarrow \|D^*D\|_2$;

$\alpha \leftarrow \alpha_0$ [Initialisation de α];

$m \leftarrow 0$ [Initialisation des itérations];

while $m \leq M_{max}$ *et* $|r|_2 > \varepsilon$ **do**

- $\alpha_p = \alpha$ [Stockage de l'itération précédente];
- $m \leftarrow m + 1$;
- $\beta \leftarrow \alpha - \frac{1}{L} D^*(D\alpha - x)$;
- $\alpha \leftarrow S_{\frac{\lambda}{L}}(\beta)$;
- $r \leftarrow \alpha - \alpha_p$

Result : $\alpha \in \mathbb{R}^K$, M nombre d'itérations.

2.2 Base de cosinus discrète

Nous allons appliquer l'algorithme en utilisant un dictionnaire qui est la réunion de la base canonique et de la base de cosinus discrète appelée Discrete cosinus transform (DCT) qui vérifie les propriétés qui suivent.

Proposition 1

On définit N signaux C^ℓ pour $\ell = 0, \dots, N-1$ de la manière suivante. On pose

$$C_n^0 = \frac{1}{\sqrt{N}} \ \forall n = 0, \dots, N-1$$

Les $N-1$ autres signaux sont définis pour $\ell = 1, \dots N-1$ par

$$C_n^\ell = \frac{\sqrt{2}}{\sqrt{N}} \cos\left(\frac{(2n+1)\pi\ell}{2N}\right) \ \forall n = 0, \dots, N-1$$

La famille $\{C^\ell, \ell = 0, \dots, N-1\}$ est une base orthonormée de $\mathbb{R}^N$.

On a le résultat suivant sur la cohérence du dictionnaire $\mathcal{D}$ constitué de la réunion de la base canonique et d'une base de DCT.

Proposition 2

Soit $\mathcal{D} = \{\phi^0, \dots, \phi^{K-1}\}$ un dictionnaire qui est la réunion de la base canonique et de la base de DCT avec $\phi^\ell = \delta^\ell$ pour $0 \leq \ell \leq N-1$ et $\phi^\ell = C^\ell$ pour $N \leq \ell \leq 2N-1$

Soit $\mu(\mathcal{D}) = \max_{i \neq j} |\langle \phi^i, \phi^j \rangle|$.

Pour N assez grand on a $\mu(\mathcal{D}) \sim \frac{\sqrt{2}}{\sqrt{N}}$.

Démonstration : Il suffit de considérer $i = 0$ et $j = 2N-1$ et estimer pour N grand $|\langle \phi^i, \phi^j \rangle|$.

2.3 Travaux pratiques

Exercice 1 (*Programmation de l'algorithme*)

1. *Programmer l'algorithme de seuillage itératif.*

   On cherchera en particulier à en programmer une version qui permet de vérifier qu'à chaque itération la fonctionnelle $F_\lambda(\alpha^{(m)})$ décroit. Cela nous servira dans la phase de test du programme pour vérifier que la programmation de l'algorithme est correcte. Dans un deuxième temps on pourra s'en passer et garder une version qui ne renvoie pas la valeur de $F_\lambda(\alpha^{(m)})$.

2. *Le tester pour $\alpha_0 = 0_K$ en utilisant le dictionnaire $\mathcal{D}$ réunion d'une base de la base canonique et d'une base de DCT dont la matrice représentative dans la base canonique est produit par le code* `creaDictBaseCanDct.py` *fourni sur Ametice. On appliquera avec $\lambda = 0.01$ l'algorithme sur le vecteur $x = (x_0, \ldots, x_{N-1})$ tel que pour tout $n = 0, \ldots N-1$ avec $N = 1024$*

$$x_n = 4\delta_n^{256} + 0.25\delta_n^{700} + \frac{8\sqrt{2}}{\sqrt{N}} \cos\left(\pi * 25 * \frac{2n+1}{2N}\right) + \frac{5\sqrt{2}}{\sqrt{N}} \cos\left(\pi * 160 * \frac{2n+1}{2N}\right).$$

   L'algorithme fournit-il la décomposition parcimonieuse recherchée ?
3. *Tester la convergence de l'algorithme pour d'autres valeurs de λ mais et d'autres autres valeurs de l'initialisation α_0. Que se passe-t-il quand λ devient grand ?*
   Le but est de montrer que la convergence a toujours lieu quelle que soit l'initialisation, et aussi de montrer que selon les valeurs de λ le minimum calculé n'est pas le même.

3 Théorème de recouvrement des décompositions parcimonieuses

Notre objectif est ici de chercher à voir si comme pour l'algorithme de Matching-Pursuit il est possible de garantir que l'algorithme de seuillage itératif retrouve une décomposition parcimonieuse du vecteur x, si elle existe.
Nous avons le théorème suivant dont la démonstration (que je vous invite à travailler) se trouve dans [3]

Théorème 1

Soit $\mathcal{D} = \{\phi^0, \ldots, \phi^{K-1}\}$ un dictionnaire de $\mathbb{C}^N$ et $x \in \mathbb{C}^N$ tel que

$$x = \sum_{\ell \in \mathcal{L}} a_\ell \phi^\ell \tag{1}$$

où $\mathcal{L}$ est un sous-ensemble de $\{0, \ldots, K-1\}$.

On fixe $\lambda > 0$. On note

$$F_\lambda(\alpha) = \frac{1}{2}|D\alpha - x|_2^2 + \lambda|\alpha|_1$$

Si $card(\mathcal{L})\mu(\mathcal{D}) \leq \frac{1}{2}$ alors

1. *le minimum $\alpha^\star$ de F_λ est unique.*
2. *Les coordonnées non nulles de $\alpha^\star$ sont indexées par des éléments de $\mathcal{L}$.*
3. $\sup\limits_{k=0,\ldots,N-1} |a_k - \alpha_k^\star| \leq \frac{\lambda}{1-(card(\mathcal{L})-1)\mu}$.
4. *En particulier si $|a_{k_0}| > \frac{\lambda}{1-(card(\mathcal{L})-1)\mu}$ alors $\alpha^\star(k_0) \neq 0$.*

Exercice 2

Commenter à l'aide d'expériences numériques les résultats du théorème 1.

4 Compte-rendu du TP

Le compte-rendu est à rendre individuellement sur Ametice.

Le compte-rendu du TP consiste en une archive contenant
— un fichier .pdf écrit en Latex ou un fichier Note_book contenant le texte (commentaires, démonstrations mathématiques) et les figures du compte-rendu

— les fichiers `.py` qui ont servi pendant la programmation, qui doivent être commentés : on doit dès les premières lignes et tout au cours du programme savoir ce qu'ils font et ce qu'ils calculent ou illustrent. On doit en particulier dans le cas des fichiers JupyterNotebook pouvoir relancer la compilation.

Si le texte est rendu en .pdf il est indispensable que parmi les fichiers `.py` on trouve pour chaque exercice un fichier dont le titre commence par `demo` et indexé par le numéro de l'exercice qu'il suffit d'executer pour illustrer informatiquement ce qui est indiqué dans le compte-rendu.

Dans tous les cas dans le compte-rendu on doit trouver

1. Des commentaires et autant que possible des explications sur les simulations présentées et les phénomènes constatés. On pourra choisir d'approfondir numériquement certains points particuliers qui vous paraissent pertinents et intéressants et détailler les conclusions qu'on peut tirer de ces expériences numériques.
2. Une étude mathématique partielle ou exhaustive. Elle peut s'appuyer par exemple sur la démonstration des propositions indiquées dans le sujet, ou sur tout autre développement dans le cadre du sujet, l'algorithme de seuillage itératif ISTA.
3. **Pour aller plus loin :** voici quelques pistes (non exhaustives)
   — on peut examiner ce qui se passe si on a en fait un signal x qui vérifie (1) auquel on ajoute un bruit blanc gaussien. L'article [3] (Corollaire 9) donne en particulier des éléments sur les conditions auxquelles le minimum de F_λ a ses coordonnées non nulles sur $\mathcal{L}$.
   — on peut s'intéresser à la vitesse de convergence de l'algorithme et étudier numériquement l'intérêt de l'accélération proposée dans [4], dont la convergence est montrée dans [1] dans la version suivante

---

Algorithme 2 : Seuillage itératif accéléré.

Data : matrice (ici notre dictionnaire) $D = [\phi^0, \dots, \phi^{K-1}] \in \mathbb{R}^{N,K}$, $x \in \mathbb{R}^N$, $\alpha_0 \in \mathbb{R}^K$, $\lambda > 0$ (paramètre de régularisation) $\varepsilon > 0$, $a > 2$ (par exemple $a = 3$), $M_{max} \in \mathbb{N}$

initialisation :

$r \leftarrow x$ [Initialisation de la différence entre deux itérées];

$L \leftarrow \|D^*D\|_2$;

$v_0 \leftarrow 0_M$;

$t_1 \leftarrow 1$;

$\alpha \leftarrow \alpha_0$ [Initialisation de α];

$m \leftarrow 1$ [Initialisation des itérations];

while $m \leq M_{max}$ *et* $|r|_2 > \varepsilon$ **do**

  $\alpha_p = \alpha$ [Stockage de l'itération précédente];

  $\beta \leftarrow \alpha - \frac{1}{L} D^*(D\alpha - x)$;

  $v_m \leftarrow S_{\frac{\lambda}{L}}(\beta)$;

  $t_m \leftarrow \frac{m+a-1}{a}$;

  $t_{m+1} \leftarrow \frac{m+a}{a}$;

  $\alpha \leftarrow v_m + \frac{t_m - 1}{t_{m+1}}(v_m - v_{m-1})$;

  $r \leftarrow \alpha - \alpha_p$;

  $m \leftarrow m + 1$;

Result : $\alpha \in \mathbb{R}^K$, M nombre d'itérations.

Grille de notation approximative

Ci-dessous quelques critères qui guideront la notation : quatre critères vont être pris en compte

— programmation : les codes ne doivent pas être buggés et comporter des commentaires qui expliquent et détaillent ce qu'ils font.

— commentaires des parties simulations : ils doivent aider à comprendre ce qui est simulé, et le point principal est leur clarté et leur précision. On attend aussi de chercher à

comprendre ce que l'on fait et donc d'expliquer au lecteur le plus possible les raisons pour lesquelles on choisit de calculer telle ou telle quantité.
— explications théoriques : un ou plusieurs développements mathématiques qui vous intéressent ou vous semblent pertinents doivent être présentés, expliqués et articulés avec la partie simulation. Il est possible de choisir d'autres points que ceux qui sont suggérés par l'énoncé du TP et cela est a priori encouragé à l'avertissement près que le plagiat sera durement sanctionné.
— forme du document : le document doit être clair et correctement rédigé. Il doit comporter une introduction et un plan et être rédigé en français clair et autant que possible correct.

Pour vous donner une idée voilà à peu près ce qui vous attend selon le degré d'investissement dans le travail.
— Note entre 0 et 5/20 : les codes ne tournent pas en général. En particulier les fichiers `demoexo1.py,...` ne sont pas présents ou sont buggés. La rédaction est très succinte voir inexistante. La forme du document laisse à désirer et le français est incorrect ou parfois même incompréhensible.
— Note entre 5 et 8/20 : certains codes tournent, mais pas tous. En particulier un ou deux fichiers `demoexo...` ne tournent pas. Les commentaires sont généraux et peu précis et reprennent le sujet du TP avec presque aucun apport personnel ni détails sur ce qui est fait, ou alors les commentaires sont juste descriptifs sans aucun effort d'analyse.
— Note entre 8/20 et 10/20 : les codes tournent mais les commentaires restent peu précis ou alors purement descriptifs, et il y a peu de réflexion personnelle dans le compte-rendu. On ne comprend pas bien pourquoi tel ou tel calcul est produit.
— Note entre 10/20 et 15/20 : les codes tournent. Une partie jusqu'à toutes les parties mathématiques sont développées et détaillées. Les simulations sont expliquées clairement. Il y a des efforts de rédaction pour articuler les commentaires des simulations et de la partie mathématique.
— Note entre 15/20 et 20/20 : les codes tournent, les commentaires de la partie simulation et de la partie mathématique s'articulent et expliquent ce qui est fait. On va même dans certains cas jusqu'à aller plus loin et proposer d'explorer une piste qui n'est pas présente stricto senso dans le sujet.

Références

[1] Chambolle, A. et, Dossal, C. (2015) On the convergence of the iterates of the "Fast Iterative Shrinkage/Thresholding algorithm". Journal of Optimization Theory and Applications, Springer Verlag, 2015, Volume 166 (Issue 3), pp.25

[2] Daubechies, I., Defrise, M. et De Mol, C. (2004) An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. Communications on Pure and Applied Mathematics, 57, 1413-1457.

[3] Tropp, J. Just Relax : Convex Programming Methods for Identifying Sparse Signals in Noise. IEEE Trans. on Information theory, Vol. 52, no. 3, March 2006.

[4] Beck, A. et Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems SIAM J. Imaging Sci., Vol 2 no 1, 183â202.