Localisation d'activité intracérébrale à partir d'électrodes de surface (séance 1)

ESIR 2 option Ingénierie Biomédicale - séance 1

Université de Rennes1

1 Introduction

L'objectif de cette première séance est de vous familiariser avec certains des outils étudiés en cours tels que la transformée de Fourier à court terme tout en poursuivant votre apprentissage de MATLAB. En particulier, nous verrons comment définir et appeler une fonction sous MATLAB. Ceci nous permettra entre autres de programmer une fonction calculant le produit scalaire normalisé entre deux vecteurs. Nous réutiliserons cette fonction lors de la troisième séance de TP afin d'implémenter la dernière étape de la méthode de localisation de sources étudiée en cours.

2 Création et utilisation d'une fonction MATLAB

Le logiciel MATLAB permet, outre l'exécution directe de commandes dans la fenêtre de commande de MATLAB, d'écrire des programmes qui pourront être exécutés ultérieurement.

- Cliquer sur l'onglet "M-File" dans le menu "File/New/" de MATLAB. L'éditeur de programme de MATLAB va alors s'ouvrir.

Nous allons créer une fonction permettant de calculer la transposée Hermitienne d'un vecteur.

- Dans la fenêtre qui s'est ouverte, tapez les 2 lignes suivantes :
  function [v1 v2] = TransposeeHermitienne(vecteur) % declaration de la fonction
  % corps de la fonction
  v1 = conj(vecteur.'); % codage de la fonction
  v2 = vecteur';

Nous allons maintenant enregistrer cette fonction dans notre répertoire de travail. Il est important que le nom du fichier soit le même que le nom de la fonction associée. Vous devez donc appeler votre fichier "TransposeeHermitienne" (le format de fichier *.m est le format MATLAB).

- Cliquer sur l'onglet "Save As" dans le menu "File/" de l'éditeur de programme de MATLAB afin de sauvegarder le programme sous le nom de votre choix dans le répertoire de votre choix. N'hésitez pas à vous créer un repertoire de travail propre à ce module d'enseignement sur votre compte.

Maintenant, vous pouvez revenir dans le logiciel et vous placer dans votre répertoire de travail.

- Servez vous de la barre d'adresse "current directory" présente en haut de la fenêtre de commande de MATLAB.

Vous pouvez maintenant créer un programme principal qui aura pour but d'appeler la fonction précédemment créée.

- Ouvrez une nouvelle page d'écriture à l'aide de l'éditeur de MATLAB que vous sauvegarderez dans votre répertoire de travail sous le nom "tp1".

La fonction "TransposeeHermitienne" prend en argument un vecteur et renvoie la transposée Hermitienne de ce vecteur. Il vous faut donc commencer par créer un vecteur $\boldsymbol{v} = [2\mathrm{i}, -1, 1]^{\mathsf{T}}$:

- Compléter le programme "tp1" afin de tester la fonction "TransposeeHermitienne". Sauvegardez-le à nouveau, puis exécutez-le dans la fenêtre de commande MATLAB. Vérifiez que le résultat obtenu par MATLAB est bien la transposée hermitienne du vecteur $\boldsymbol{v}$ défini ci-dessus.

3 Le produit scalaire normalisé

3.1 Définitions et propriétés

L'algorithme étudié en cours permet de localiser des sources épileptiques de profondeur à partir de données EEG de scalp. Le but final de ce TP est d'implémenter cet algorithme afin de localiser les sources que vous aurez vous même positionnées sur le cortex. Nous reviendrons plus en détails sur cet algorithme lors la troisième séance de TP. Ce que nous pouvons d'ores et déjà dire, c'est que la dernière étape de cet algorithme repose sur la maximisation de la fonction de coût suivante :

$$\varphi(\boldsymbol{a}(\boldsymbol{\rho}), \boldsymbol{e}) = \frac{|\langle \boldsymbol{e}, \boldsymbol{a}(\boldsymbol{\rho}) \rangle|}{\|\boldsymbol{e}\|_2 \, \|\boldsymbol{a}(\boldsymbol{\rho})\|_2} = \frac{|\boldsymbol{e}^{\mathsf{H}} \, \boldsymbol{a}(\boldsymbol{\rho})|}{\|\boldsymbol{e}\|_2 \, \|\boldsymbol{a}(\boldsymbol{\rho}\|_2} \tag{1}$$

par rapport à $\boldsymbol{\rho}$, où $\|\boldsymbol{b}\|_2 = \sqrt{\langle \boldsymbol{b}, \boldsymbol{b} \rangle}$ représente la norme L2 du vecteur $\boldsymbol{b}$ et $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ le produit scalaire des vecteurs colonnes $\boldsymbol{a}$ et $\boldsymbol{b}$ défini par :

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = a_1 b_1^* + \ldots + a_N b_N^* = \sum_{n=1}^{N} a_n b_n^* = \boldsymbol{b}^{\mathsf{H}} \, \boldsymbol{a} \tag{2}$$

A noter que le vecteur $\boldsymbol{e}$ de l'équation (1) est un vecteur qui sera déduit de la décomposition canonique polyadique du tableau "espace $\times$ temps $\times$ fréquence" construit à partir des données EEG. Rappelons également que le vecteur $\boldsymbol{a}(\boldsymbol{\rho})$ désigne le vecteur de transfert de l'activité électrique du dipôle situé à la position $\boldsymbol{\rho}$ vers les électrodes EEG. Réécrivons à présent l'inégalité de Cauchy-Schwarz :

$$|\langle \boldsymbol{a}, \boldsymbol{b} \rangle| \leq \|\boldsymbol{a}\|_2 \, \|\boldsymbol{b}\|_2 \tag{3}$$

L'égalité est atteinte si et seulement si $\boldsymbol{a}$ et $\boldsymbol{b}$ sont colinéaires ($\Leftrightarrow \boldsymbol{a} = \alpha * \boldsymbol{b}$ où $\alpha \in \mathbb{C}$). Pour deux vecteurs $\boldsymbol{a}$ et $\boldsymbol{b}$ non nuls, l'inégalité de Cauchy-Schwarz (3) peut se réécrire sous la forme suivante :

$$\frac{|\langle \boldsymbol{a}, \boldsymbol{b} \rangle|}{\|\boldsymbol{a}\|_2 \, \|\boldsymbol{b}\|_2} \leq 1 \tag{4}$$

On appellera alors produit scalaire normalisé la fonction φ définie par :

$$\varphi(\boldsymbol{a}, \boldsymbol{b}) = \frac{|\langle \boldsymbol{a}, \boldsymbol{b} \rangle|}{\|\boldsymbol{a}\|_2 \, \|\boldsymbol{b}\|_2} \tag{5}$$

3.2 Manipulations

Nous allons maintenant vérifier l'inégalité de Cauchy-Schwarz sur des exemples simples.

- Créez 3 vecteurs $\boldsymbol{a} = [2, 5, 6, 8, 6, 7]^{\mathsf{T}}$, $\boldsymbol{b} = [2, 8, 6, 7, 4, 6]^{\mathsf{T}}$ et $\boldsymbol{c} = [2, 8, 6, 7, 4, 6]^{\mathsf{T}}$ et calculez les produits scalaires normalisés de ces vecteurs. Dans un premier temps, vous utiliserez les fonctions "dot" et "norm". Que remarquez-vous? Dans un second temps, vous chercherez à ne pas utiliser ces deux fonctions : vous pourrez utiliser la fonction "TransposeeHermitienne" implémentée précédemment. Vous créerez une fonction "ProdScalNorm" prenant en entrée deux vecteurs colonnes et renvoyant en sortie deux scalaires, correspondant aux deux manières précédentes d'implémenter le produit scalaire normalisé des deux vecteurs passés en entrée.

A présent, nous allons appliquer le produit scalaire normalisé à des signaux épileptiques, téléchargeables sur la page web de Laurent Albera[1] (fichier "pointes.mat" en allant dans l'onglet "Teaching" à la rubrique "ESIR 2").

- Ecrivez une ligne de commande permettant de charger le fichier "pointes.mat" en mémoire.
- Executer votre programme principal. Exécuter la commande "whos" dans la fenêtre de commande de MATLAB. A quoi sert cette fonction? Quelle variable a été créée lors du chargement en mémoire précédent? Quelles sont les dimensions de cette nouvelle variable? Ecrire une procédure d'afffichage graphique des signaux stockés dans cette variable : on choisira de représenter un signal par figure en mettant l'ensemble des figures dans une même fenêtre graphique d'afffichage.
- Quels signaux semblent visuellement être les plus semblables?

Nous allons à présent vérifier le résultat précédent à l'aide du produit scalaire normalisé.

- Appliquer la fonction "ProdScalNorm" aux signaux épileptiques étudiés. Le résultat visuel est-il confirmé?

4 La transformée de Fourier à court terme

4.1 Définition

Soit $\{s[m]\}$ un signal à temps discret de longueur infinie. La Transformée de Fourier à Court Terme (TFCT) discrète du signal $\{s[m]\}$, notée $\mathcal{S}_s$, est définie par :

$$\mathcal{S}_s[\ell, k] = \sum_{m=-\infty}^{+\infty} s[m]\, g[m-\ell]\, \mathrm{e}^{-\mathrm{i}2\pi mk} \tag{6}$$

où $\{g[m]\}$ est un signal discret de carré sommable, i.e. tel que :

$$\sum_{m=-\infty}^{+\infty} |g[m]|^2 < +\infty$$

[1] http://perso.univ-rennes1.fr/laurent.albera/

4.2 Manipulations

Nous allons maintenant implémenter sous MATLAB la TFCT.

- Créez une fonction "TFCT" prenant en argument un vecteur $\boldsymbol{tps}$ de longueur L, un vecteur $\boldsymbol{freq}$ de longueur K, un vecteur $\boldsymbol{s}$, un nombre $g \in \{1, 2, 3, 4, 5\}$ indiquant le choix de fenêtre g parmi celles décrites en cours et renvoyant en sortie une matrice $\mathcal{S}_s$ de taille $(K \times L)$ contenant les valeurs de la TFCT de $\boldsymbol{s}$ ainsi que la matrice $\mathcal{S}_s$ calculée à l'aide de la fonction "spectrogram" de MATLAB.

- Téléchargez le fichier "SEEG.mat" sur la page web de Laurent Albera en allant dans l'onglet "Teaching" à la rubrique "ESIR 2", puis écrivez la liggne de commande dans votre programme principal "tp1.m" qui permettra de charger en mémoire le contenu de ce fichier, à savoir une matrice $\boldsymbol{S}$ de taille (2×10000) contenant deux activités épileptiques.

- Exécutez la fonction "TFCT" dans "tp1.m" en l'appliquant à la première ligne de $\boldsymbol{S}$. Vous considérerez plusieurs types de fenêtre $\{g[m]\}$. Comparez en termes de résultat votre implémentation de la TFCT et celle de Matlab, notamment en termes de temps de calcul (utilisez les fonctions "tic" et "toc" de MATLAB).

- Vérifiez que l'hypothèse de séparabilité en temps et en fréquence de la TFCT d'une activité épileptique est correcte (en première approximation).