

ISAE-FISA

BCI charge mentale du pilote

Frédéric Dehais, Simon Ladouce, Juan Torre Tresols, Ludovic Darmet

Résumé

A partir des données que vous avez collecté en simulateur, vous allez construire une BCI qui détectera si le sujet est en charge mentale faible (Pilot Monitoring) ou élevée (Pilot Flying).

Pour cela il faudra :

1. Nettoyer, débruiter les données
2. Préparer les données au bon format et extraire des features
3. Enfin, ajouter un classifieur et une stratégie d'évaluation de la performance (cross-validation)

Vous pouvez pour cela utiliser Matlab ou Python (recommandé).

Le rendu sera le code et un rapport (1 page) pour expliquer les différentes étapes du pipeline ainsi que les limites du protocole et des casques Muse 2. Nous attendons vos rendus pour le 20 février à ludovic.darmet@isae-supero.fr et frederic.dehais@isae-supero.fr. Vous pouvez bien sûr nous demander de l'aide.

1 Pré-processing

Le pré-processing sera plutôt simple et pour retirer certains artefacts évidents (principalement hautes-fréquences). Vous trouverez des éléments pour vous aider dans les BE acquisition de données et dans le BE Python/MNE.

1. Charger les données, les visualiser et les explorer
2. Sélectionner les channels à garder (supprimer le "Right AUX")
3. Diviser en epochs de 2s avec les labels correspondants (50/50 de Pilot Flying/Pilot Monitoring)

2 Extraction de features

On choisira ici des features spectrales. Pourquoi d'après vous ? Les premiers cours d'introduction vous aideront à répondre.

Vous allez donc calculer la puissance moyenne (en décibel : \log_{10}) sur les bandes *alpha*, *beta*, *theta* et *delta*. Vous trouverez ici de l'aide dans les BE Python/MNE et SSVEP.

Alternativement vous pouvez tester d'autres features en fonction de votre inspection manuelle du signal :

- La puissance moyenne sur alpha divisée par delta (réduction du bruit dû aux artefacts moteurs)
- Beta sur theta (supposément en lien avec la concentration)
- Theta sur alpha (supposément en lien avec une anxiété réduite)

3 Classification

Vous allez ici vous servir des BE Titanic et SSVEP. Après avoir extrait les features, séparer les données entre train et test. Sur le jeu de train :

1. Normaliser les données (StandardScaler)
2. Choisir un classifieur et sélectionner des hyper-paramètres optimaux à l'aide de cross-validation

Évaluer la performance sur le jeu de test. Pour aller plus loin vous pouvez mettre en place une autre boucle externe de cross validation pour la séparation train/test.