

Machine learning pour la classification de phases de sommeil

Alexandre Herbert et Baptiste Turpin

Sommaire

| | | |
|------------|--|----------|
| I | Contexte | 2 |
| II | Choix d'implémentation | 2 |
| 1 | Choix du modèle | 2 |
| 2 | Sélection des méthodes de validation algorithmique | 2 |
| III | Pré-traitement des données | 2 |
| IV | Selection de features | 2 |
| V | Optimisation des hyperparamètres des features | 3 |
| VI | Résultats | 3 |
| VII | Critiques et perspectives | 3 |

I Contexte

II Choix d'implémentation

1 Choix du modèle

Random Forest tout ça tout ça

2 Sélection des méthodes de validation algorithmique

cross validation toussa toussa

III Pré-traitement des données

on a normalisé, ce genre de trucs Après selection des features ?

IV Selection de features

Parler des observations faites sur les plots (featuring jolies images) Et puis aussi on prend les fft. Pourquoi ? Pourquoi pas.

— obs 1

— obs 2

D'où features potentiellement pertinentes (discriminantes) :

| | |
|---|---|
| a | a |
| a | a |
| a | a |
| a | a |
| a | a |
| a | a |

V Optimisation des hyperparamètres des features

coucou, on a croisé les hyperparamètres et des choses stylées. Comme on est critiques et forts, on remarque qu'on risque d'overfitter sur les data de test en faisant ça!

VI Résultats

VII Critiques et perspectives

```
X_test_fft = h5py.File('data/X_test_fft.h5')

def buildAndSaveMatrix(h5file_freq, methodOne, param,
    list_bool_extract_signal, name_save):
    rep = extractFeatureAll(h5file_freq , methodOne , param ,
        list_bool_extract_signal)
    temp_var_file = open("design_matrix/eleme/" + name_save + '.txt','wb')
    pickle.dump(rep , temp_var_file)
    temp_var_file.close()
    len(bidule)
```
