Eeg features extraction

Gabriel Olympie Thibault Leblanc Antoine de Mathelin

Summary

- 1. Introduction
- 2. Problem explanation
- 3. Features extraction

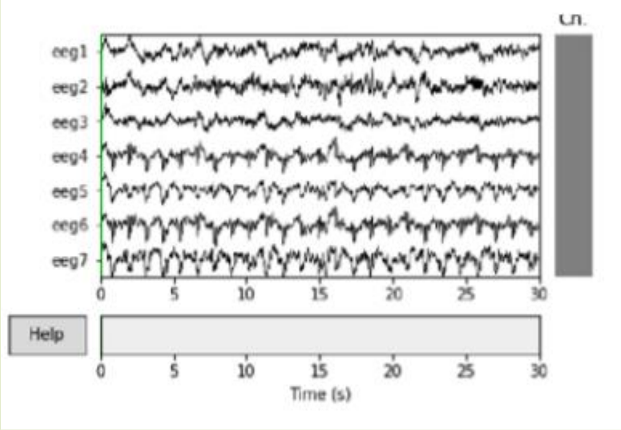
It’s important but complex to classify sleep stage

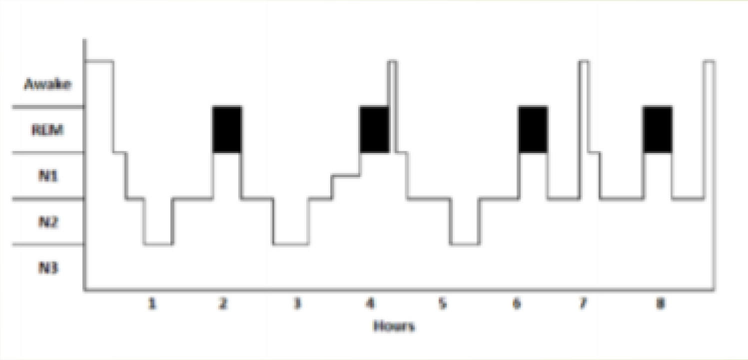
- Sleep can be classified into different category : wake, light sleep, deep sleep, REM
- Monitoring sleep stage is beneficialfor diagnosing sleep disorders.
- It is a long and difficult work to score it manually based on EEG signal reading

Our goal : to develop an algorithm of sleep staging able to differentiate between the different sleep stages.

Our dataset

- Dataset of 38k entries
- Each entry : 7 eeg signals of 30s sampled at 50hz
- Each entry is labeled with a sleep stage (1-4)
  - A dataset of about 5 GO



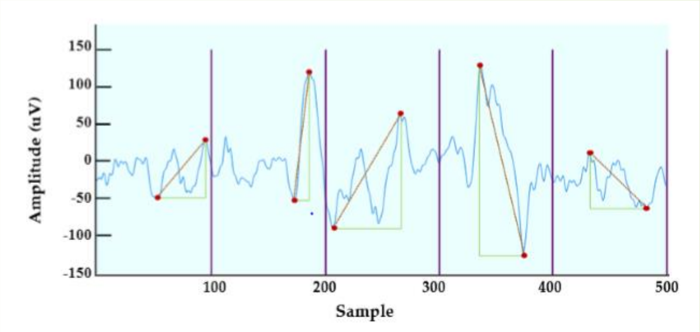


Our approach

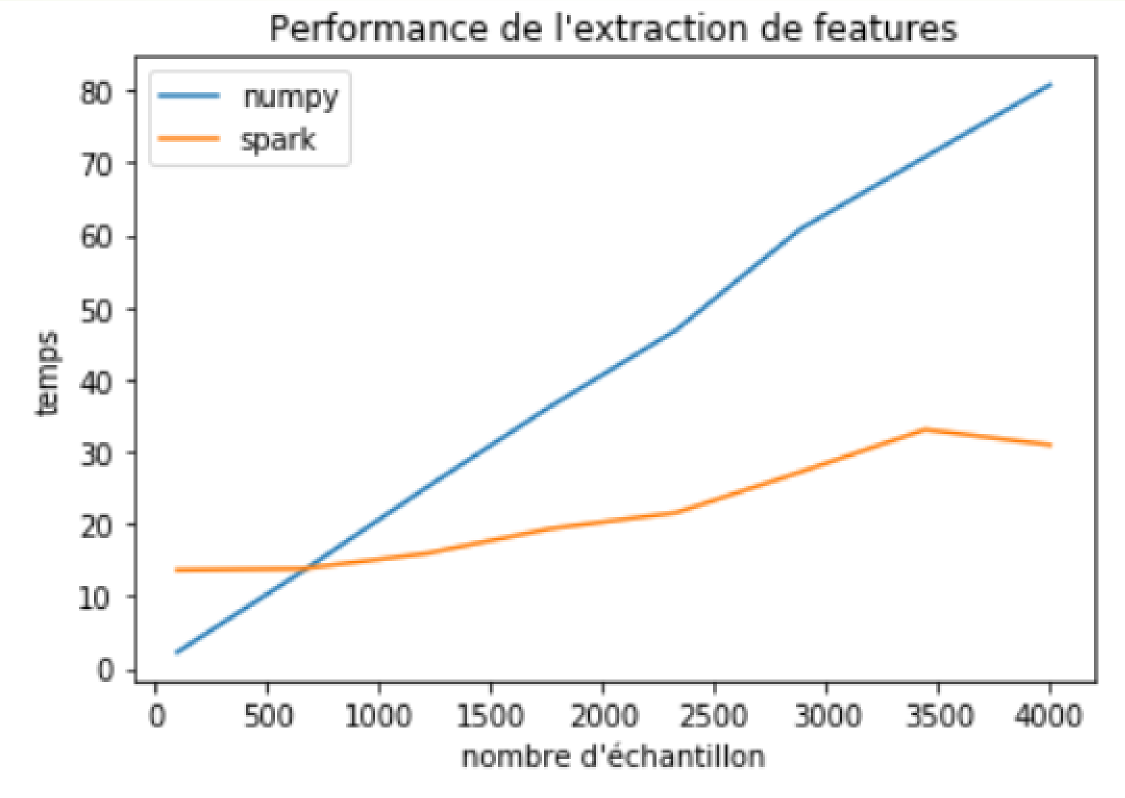
- 1) Extracting different features from the signals
  - In a tradditionnal manner
  - Using a pyspark architecture
- 2) Using a classiffying algorithm in order to predict f1 score

Our features

- Mean
- Standard deviation
- Min-Max distance
- Entropy



Pyspark vs traditionnal extraction



Classification and evaluation

- Model : A simple random tree forest
- F1 : 0,61

K-means Map-Reduce pseudo-code

- 1. Choose k centers: $C_1,\ldots,C_k$
- 2. Apply the map method and associate each x_i to its closest center
- 3. Apply the reduce method until all the points corresponding to each class are merged, we then have a list of k tuples: $[(1,(y_1,c_1)),\ldots,(k,(y_k,c_k))]$
- 4. Compute the new centers as follow $\widetilde{C_l} = \frac{y_l}{c_l}$
- 5. Go back to step 2 until the algorithm converge

K-means Map-Reduce methods

Class Mapper

method map(($C_1,\dots, C_k$, x_i):

$$l = \underset{j\in[\![1,k]\!]}{\operatorname{argmin}}(\|C_j - x_i\|)$$

emit (l , (x_i, 1))

Class Reducer

method reduce((l , [(x_i, card$_i$), ... , (x_j, card$_j$)]):

emit (l , [($y = \sum_{p=i}^{j} x_p$, $c = \sum_{p=i}^{j} \text{card}_p$)])