Classification of Sleep Phases with EEG and Accelerometer Data using Wavelet Transforms

Challenge Data – Dreem

Alex AUVOLAT

Febuary 24, 2016

Overview

Section 1

Task Presentation

Dataset characteristics

- 31129 training points
- 30458 test points

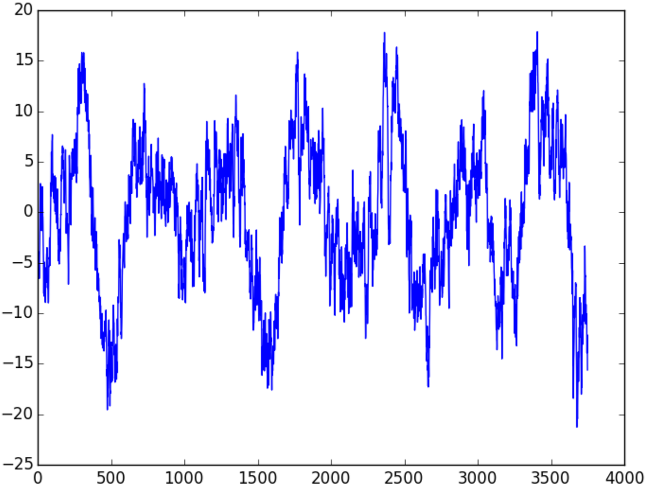
Features:

- 15 seconds of EEG data, 250 Hz, 1 channel $\rightarrow$ 3750 values
- 15 seconds of accelerometer data, 10 Hz, 3 channels $\rightarrow$ 350 values

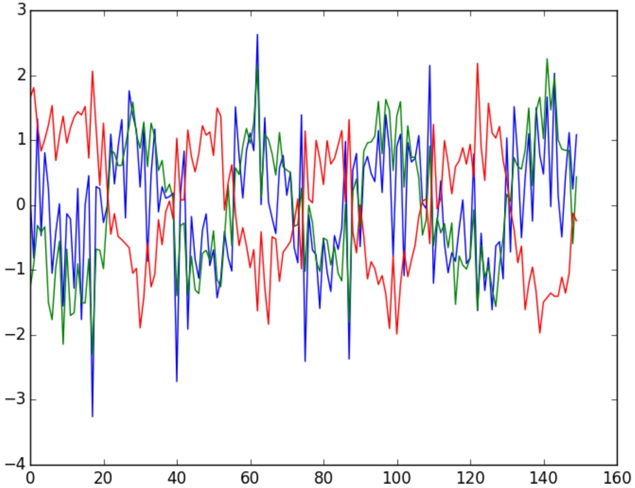
Signal categories

Class	Code	Description	Count
0		Wake	1342
1	N1	Light sleep ("somnolence")	428
2	N2	Intermediate sleep	15334
3	N3	Deep sleep	9640
4	REM	Paradoxical sleep	4385
		Total	31129

Typical EEG signal (REM sleep)



Typical ACC signal (N2 sleep)



Section 2

Feature Extraction

Wavelet transform

We first convolve the signal with a series of Ricker wavelets:

$$\psi_\sigma(t) = \frac{2}{\sqrt{3\sigma}\pi^{\frac{1}{4}}}\left(1 - \frac{t^2}{\sigma^2}\right)e^{\frac{-t^2}{2\sigma^2}}$$

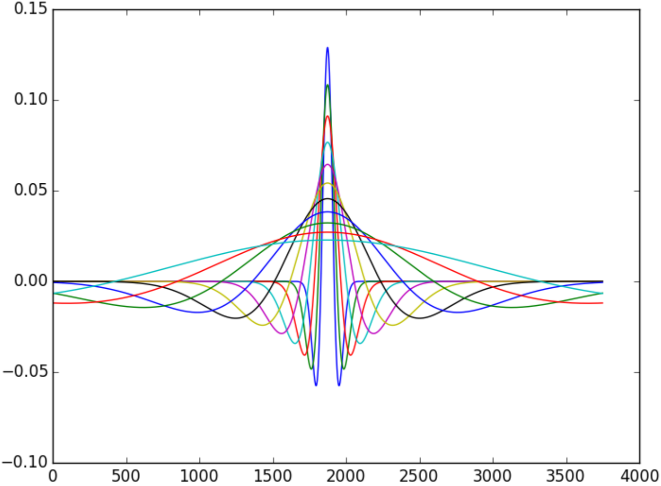
The values of σ are spaced regularly on a logarithmic scale:

$$\sigma_i = 2^i$$

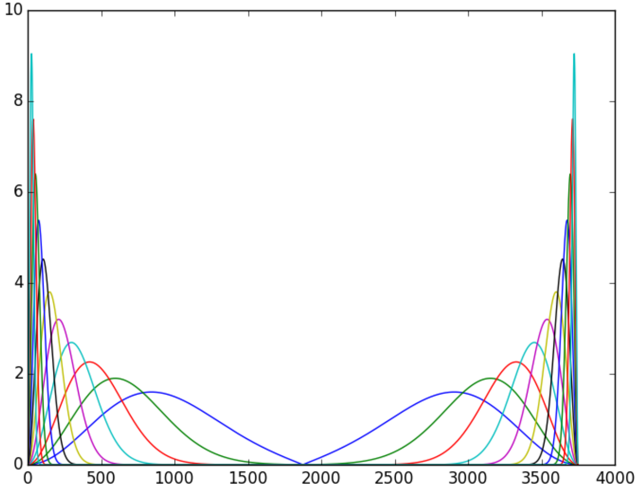
$$\sigma_0 = 1$$

$$\sigma_{10} = 1024$$

Wavelets


0.15
0.10
0.05
0.00
−0.05
−0.10
0
500
1000
1500
2000
2500
3000
3500
4000

Wavelets, in the Fourier domain

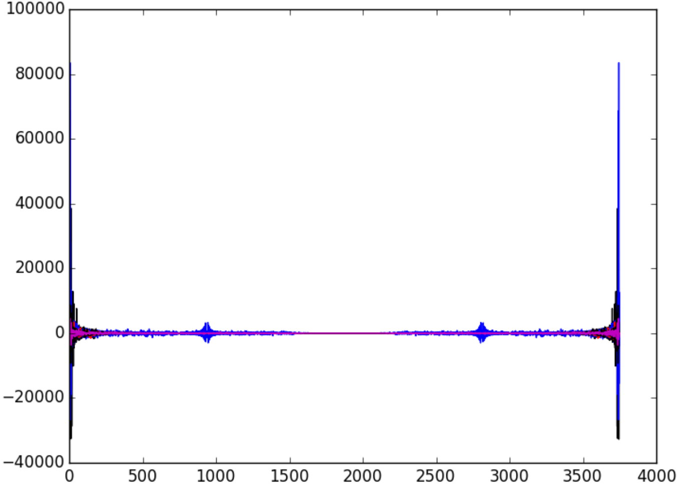


Remarks

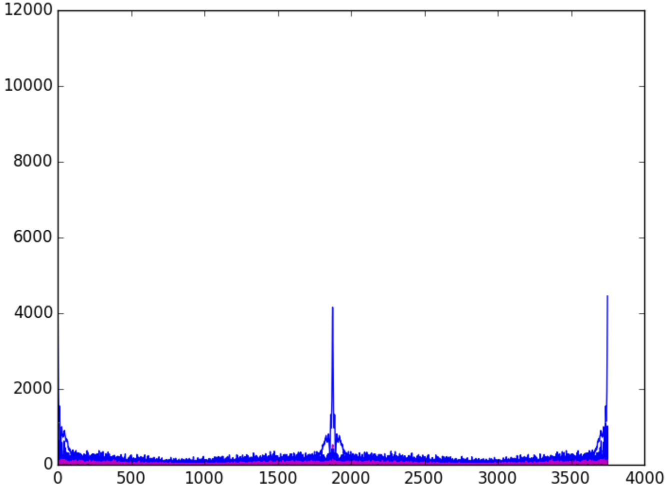
Convolution is equivalent to a multiplication in the Fourier domain, which has several implications:

- convolution can be done very quickly
- we can interpret the convolution by a wavelet as exacerbating certain frequency range while extinguishing others

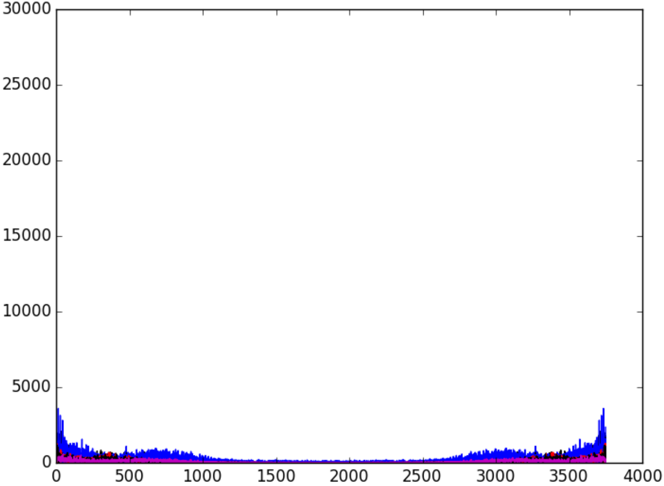
Our signal (EEG), in the Fourier domain



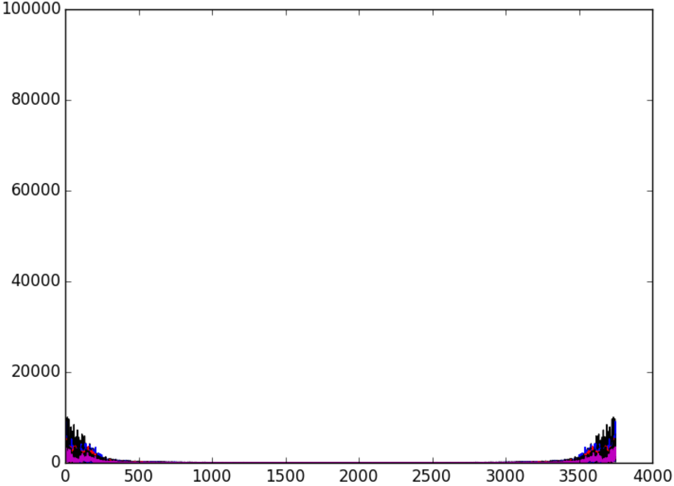
Now, convolved with a high-frequency wavelet ($\sigma = 1$)



With a lower frequency ($\sigma = 4$)



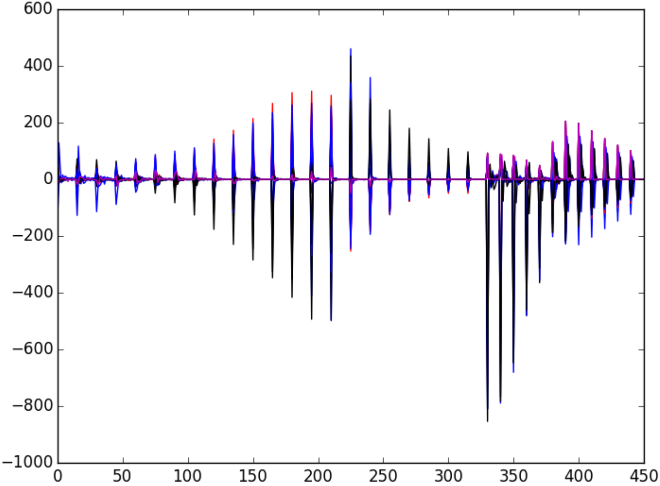
With an even lower frequency ($\sigma = 16$)



Dimensionality reduction

- It is impossible to run a classifier on 11×3750 features for 30000 examples.
- We use a PCA to transform the 3750-dimensional signal into a 15-dimensional signal.
- For the EEG, we end with $15 \times 11 = 165$ features
- Similar processing for the accelerometer data.

Feature vectors



Section 3

Classification Results

Random Forest Classification

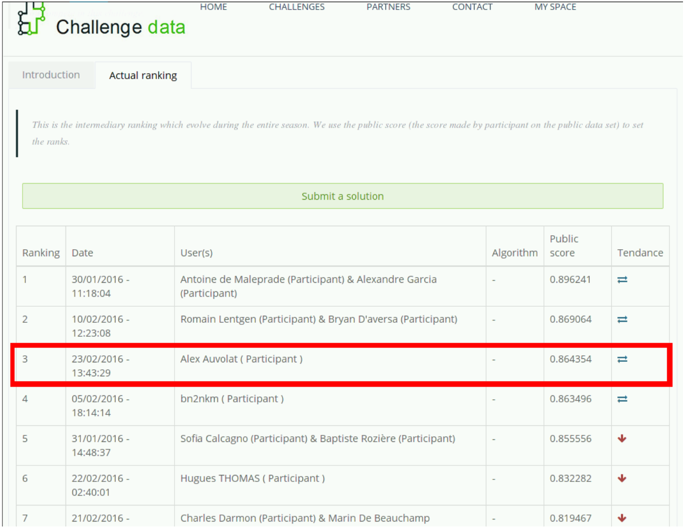
- Classification is done with a random forest
- Parallel implementation provided by `scikit-learn`
- Tried using up to 128 trees to improve model performance (more trees improves the performance a little)

Random Forest vs. Linear Classifiers

Random forest classifiers worked much better than linear classifiers on this task:

- Linear classifiers are much slower to train
- The problem is not completely linearly separable in our feature space
- Error rate: 22% (linear classifier) vs. 15% (random forest)

Results



HOME CHALLENGES PARTNERS CONTACT MY SPACE

Challenge data

Introduction | Actual ranking

This is the intermediary ranking which evolve during the entire season. We use the public score (the score made by participant on the public data set) to set the ranks.

Submit a solution

Ranking	Date	User(s)	Algorithm	Public score	Tendance
1	30/01/2016 - 11:18:04	Antoine de Maleprade (Participant) & Alexandre Garcia (Participant)	-	0.896241	⇄
2	10/02/2016 - 12:23:08	Romain Lentgen (Participant) & Bryan D'aversa (Participant)	-	0.869064	⇄
3	23/02/2016 - 13:43:29	Alex Auvolat (Participant)	-	0.864354	⇄
4	05/02/2016 - 18:14:14	bn2nkm (Participant)	-	0.863496	⇄
5	31/01/2016 - 14:48:37	Sofia Calcagno (Participant) & Baptiste Rozière (Participant)	-	0.855556	↓
6	22/02/2016 - 02:40:01	Hugues THOMAS (Participant)	-	0.832282	↓
7	21/02/2016 -	Charles Darmon (Participant) & Marin De Beauchamp	-	0.819467	↓

Remarks

- Tried many variants, validating them on a subset of the training set, for local validation
- Results on the public leaderboard were usually far worse than on the validation set
  $\rightarrow$ overfitting...
- The best score I got on the leaderboard was actually with one of the first variants I tried