SPARSE WAVELET REPRESENTATIONS CLASS
CHALLENGE DATA

MASTER 2 MATHÉMATIQUES, VISION & APPRENTISSAGE
ÉCOLE NORMALE SUPÉRIEURE DE CACHAN

Sleep Stage Classification with EEG and Accelerometer Data using Wavelet Transforms

Project Report

Febuary 2016

ALEX AUVOLAT
alex.auvolat@ens.fr
DÉPARTEMENT D'INFORMATIQUE
ÉCOLE NORMALE SUPÉRIEURE

Data & challenge provided by

DREEM
http://www.dreem.com/

1 Task Presentation

Introduction. In this challenge, provided by the start-up company Dreem, we have to identify the phases of sleep based on 15-second recordings of electroencephalogram (EEG) and accelerometer data. The data is recorded by a simple device the user puts on her head before going to sleep. In Figure 1 we show a typical EEG recording, and in Figure 2 we show a typical accelerometer recording, which contains three signals corredponding to the three axis of rotation of the user's head.

Data technicalities. Each data sample is composed of one EEG signal and of three accelerometer signal. These signals are described precisely in Table 1. The training examples belong to 5 classes which are described in Table 2.

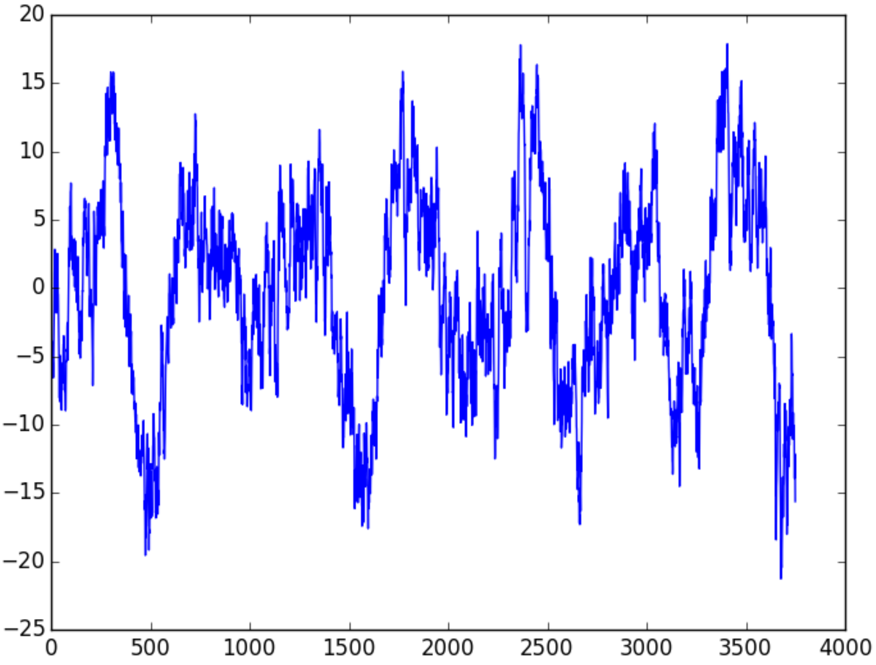

Figure 1. Typical EEG signal (REM sleep)

Difficulties. Several difficulties are found in the dataset:

- The signals are quite noisy, due to the use of a cheap recorder that doesn't provide us with optimal signal quality.
- The number of classes is not very well balanced throughout the dataset.

Signal	Channels	Sampling freq.	Number of points
Electroencephalogram	1	250 Hz	3750
Accelerometer	3	10 Hz	350
Total			4100

Table 1. Technical details of the signal.

Class	Code	Description	#train	#test
0		Wake	1342	
1	N1	Light sleep ("somnolence")	428	
2	N2	Intermediate sleep	15334	
3	N3	Deep sleep	9640	
4	REM	Paradoxical sleep (stage where dreams occur)	4385	
		Total	31129	30458

Table 2. The classes in which the examples are to be classified.

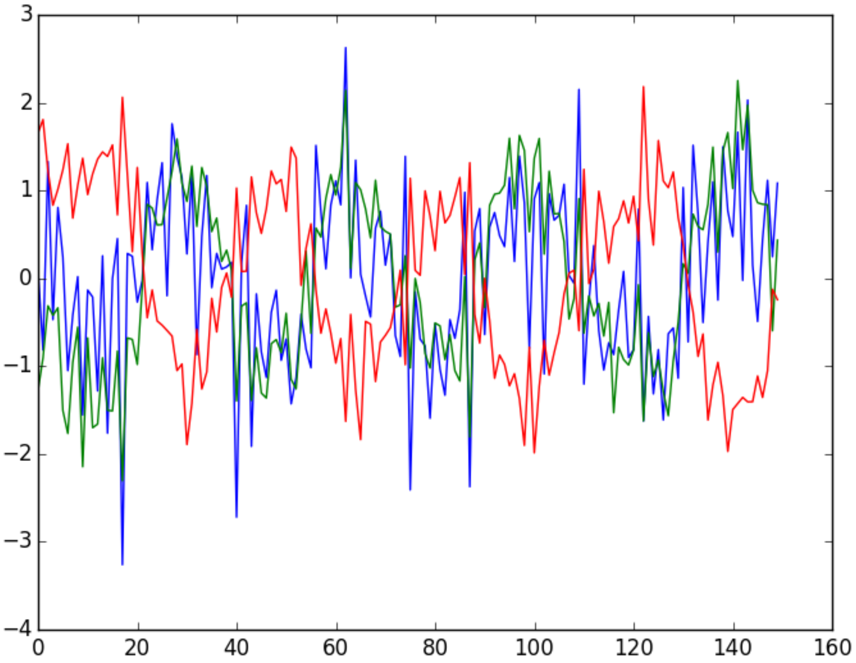

Figure 2. Typical accelerometer signal (N2 sleep), has three data channels for the three dimensions

2 Feature Extraction

Properties of the signal. The signal is periodic, therefore solutions based on the discrete Fourier transform seem to be a good general choice. Ignoring the argument of the Fourier coefficients (which correspond to the phase of the signal) and taking only their modulus creates invariance in the representation and enables the model to generalize.

Several wavelet transforms for feature extraction. The first method was simply to do an FFT on the signal and to try to classify the signals based on that, but this method did not give competitive results. To try and get better results than the simple FFT approach, I improved the feature extractor to use Wavelet transforms.

The signal is first convolved with a Ricker wavelet. This is done for a series of wavelet functions at different frequencies. The Ricker wavelets I use are showed in Figure 3.

The equation of a Ricker wavelet is:

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

The frequency bands are typically spaced by $\sigma_{i+1}/\sigma_i = 2$, but I also used $\sigma_{i+1}/\sigma_i = \sqrt{2}$ in some of my experiments.

Remarks on the Wavelet transform. A convolution by a Wavelet transform in the time domain is equivalent to a multiplication in the Fourier domain, which has several implications:

- The convolution can be done quickly, in $O(n \log n)$ (time required to do a FFT), instead of $O(n^2)$ for a naive approach. Even better: if we are only interested in the Fourier coefficients of the convonleved signal, then we can do k wavelet transforms in time $O(n \log n + n\, k)$ instead of $O(n\, k \log n)$ by doing the FFT only once for all the convolutions.
- The Wavelet transform can be interpreted as effectively exacerbating some specific frequency ranges in the Fourier domain. The Fourier transform of the wavelets that I used are showed in Figure 4, and we observe clearly that the wavelet span different frequency ranges.

Next steps of data processing. It is impossible to run a classifier on 3750 × 10 coefficients, which is the number of coefficients we obtain if we convolve the signal with 10 different Wavelet functions. In order to use the data efficiently, we must reduce the dimensionnality of the signal:

- I calculate the absolute value (modulus) of the coefficients, and take their square root to bring them all in similar amplitud ranges.
- We then do a PCA on the coefficients, in order to find 5 to 15 principal components for each transformed signal.
- Overall, the feature vector contains $n_{\text{signals}}\, n_{\text{frequencies}} n_{\text{components}}$ values, with typically:

$$\begin{aligned} n_{\text{signals}} &= 4 \\ n_{\text{frequencies}} &\simeq 10 \\ n_{\text{components}} &\simeq 4 \end{aligned}$$

  Which makes a total of 100 to 1000 features.

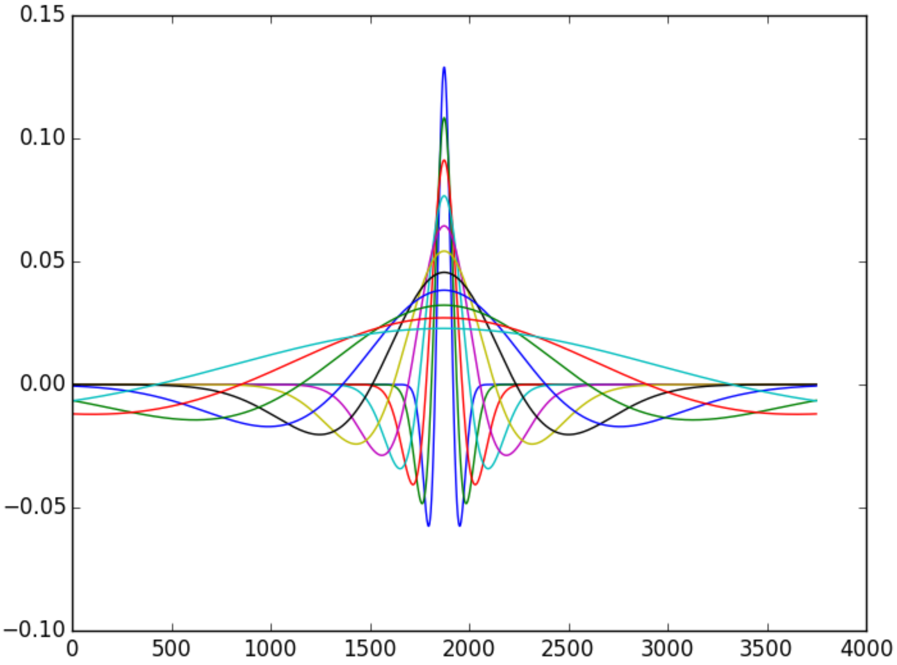

Figure 3. Ricker wavelets used in one of the models (50% lowest frequencies).

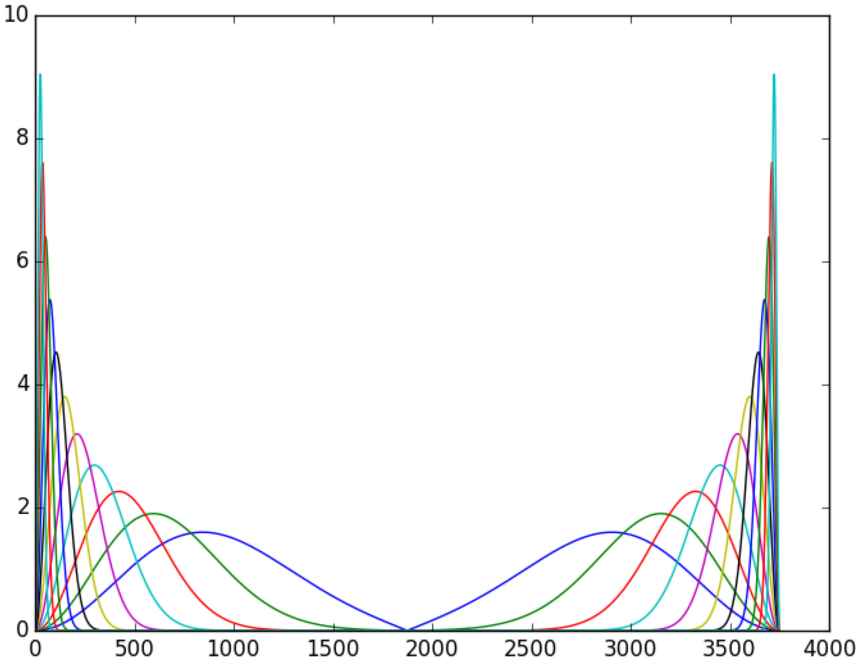

Figure 4. Fourier transform of the Ricker wavelets of the same model (50% highest frequencies).

3 Experimental Results

Random forest classification. The features are then fed into a random forest classifier which does the classification automatically using 100 trees. The results we obtain are presented in Table 3.

model	$n_{\text{freq}}^{\text{eeg}}$	$n_{\text{PC}}^{\text{eeg}}$	$n_{\text{freq}}^{\text{acc}}$	$n_{\text{PC}}^{\text{acc}}$	Valid. Error Rate	Valid. Score	Test Score
7	22	15	12	10	0.1767		**0.8643**
22	11	5	14	5	0.1602		0.8573
36	6	3	8	4	**0.1477**	0.8813	0.8557

Table 3. Results of various hyperparameter combinations

Hyperparameter search. I tried several parameters for the models, in particular I varied the selection of Wavelet functions with frequency bands overlapping more or less. I tried a large variety of parameters on a local validation set of 6129 examples that were removed from the training set. Although I only submitted 4 solutions to the public leaderboard, I found that it was very easy to overfit the validation set with too much tuning on the model. The best public results were in fact obtained with one of the first models I tried.

Best solution. The best model uses a large number of different wavelet: 22 for the EEG, spaced by $\sigma_{i+1}/\sigma_i = \sqrt{2}$ with $\sigma_1 = 1$ and $\sigma_{22} = 1448$, and 12 for the accelerometer spaced by $\sigma_{i+1}/\sigma_i = 2$ with $\sigma_1 = 1$ and $\sigma_{11} = 1024$. The feature vectors obtained for the data on a few training examples are shown in Figure 5.

Number of trees. The quality of the solution provided by a random forest can be improved slightly by using more random trees. However above a certain point the computationnal cost becomes prohibitive and the gains are minimal. For my experiments, I settled with a compromise of 128 random trees.

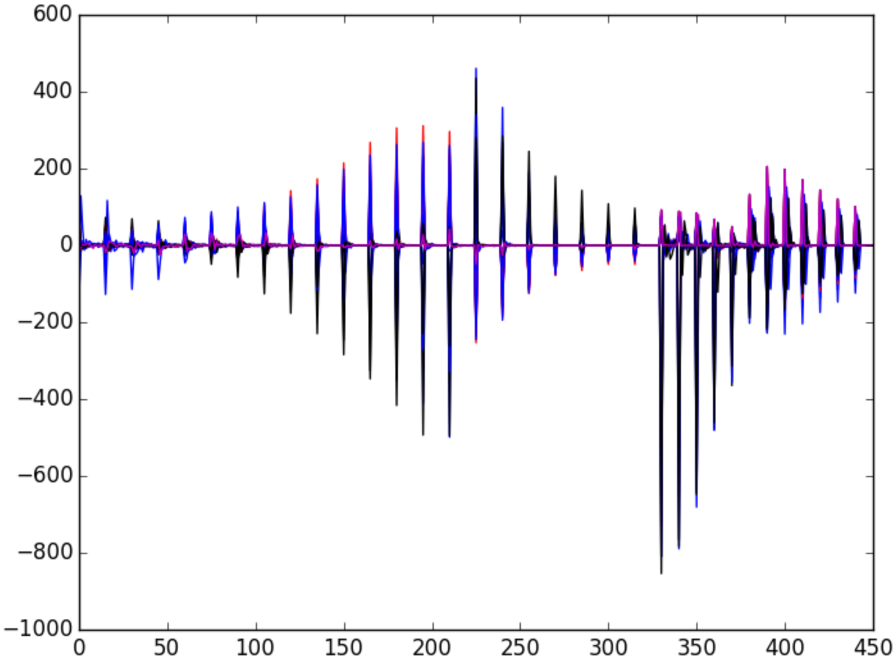

Figure 5. Features obtained on a few of the training samples.

Linear classifier vs. random forest classifier. I also tried to use a linear classifier (one vs. all Logistic regression), which didn't work as well as the random forest classifier: the error rate was of at least 22% on the validation set, whereas the validation error with a random forest classifier was of about 15%.

Concluding remarks.

- We observe that a simple Wavelet transform is usefull to separate the frequency ranges in the signal, and enables us to classify the signals quite well.
- With this simple method I ranked 3$^{\text{rd}}$ on the leaderboard, which is not a very impressive feat given the small number of participants.
- The linear classifier experiment however shows that this feature extractor is not sufficient to make the problem linearly separable. More powerfull feature extractors remain to be designed.