MS BIG DATA 2015
EEG Challenge Report

François Blas

April 20, 2015

Contents

Chapter 1

EEG

1.1 EEG reminder

Electroencephalography (EEG) is the recording of electrical activity along the scalp. It refers to the recording of the brain's spontaneous electrical activity over a short period of time. The brain activity can be split in multiple parts which correspond generally to specific frequency bands.

Band	Frequency
δ	< 4 Hz
θ	4 - 7 Hz
α	8 - 15 Hz
β	16 - 31 Hz
γ	> 32 Hz
μ	8 - 12 Hz

Table 1.1: Frequency band associate to waves type

Frequencies are not the only way to distinguish specific brain activity category, waves have also specific patterns into time.

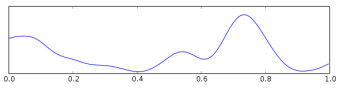

Figure 1.1: Example of delta wave

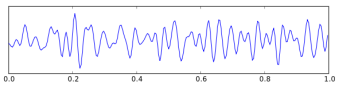

Figure 1.2: Example of beta wave

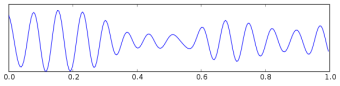

Figure 1.3: Example of mu wave

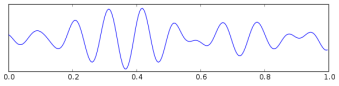

Figure 1.4: Example of alpha wave

1.2 Goal of the challenge

The goal of the challenge is to predict the sleep state of an individual according to the electrical waves generated by the brain. More specifically it is to automatically label data of 30 seconds samples EEGs versus stage of sleep in which they were recorded. There are 5 labels which correspond to different sleep states.

N1	Light sleep
N2	Less light sleep
N3	Deep sleep
R	REM sleep
W	Wake up state

Table 1.2: Labels to predict

Chapter 2

Features

This chapter describe all features used in the model.

2.1 Statistics

All these features are relevant because they are sometimes discriminant for one label. For example, the 'N1' label or 'W' can be revealed by min or max.

2.1.1 Minimum

The minimal value of the signal.

2.1.2 Maximum

The maximal value of the signal.

2.1.3 Kurtosis

Kurtosis is the fourth central moment divided by the square of the variance. In math the kurtosis value is defined as :

$$\beta_2 = E[(\frac{X_\mu}{\sigma})^4] \tag{2.1}$$

for a mean μ and a standard deviation σ.

2.2 Signal processing

2.2.1 FFT frequencies

This feature return the dominant frequency of a signal. It is computed as so:
- Compute the fft on the signal (with Hanning window).
- Associate frequencies with coefficients
- Compute amplitudes based on fft coefficients results.
- Sort frequencies by highest coefficients.
- Return the mean of the 10 best frequencies.

It is a good approximation of the dominant frequency, which is determinant to distinguish eegs.

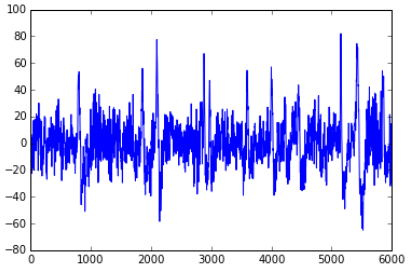

Figure 2.1: Original signal

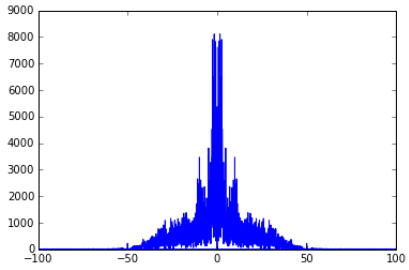

Figure 2.2: FFT of the signal

2.2.2 Auto-Correlation

Perform the correlation of a signal with itself. It's a very useful feature to indicate repeating patterns inside a signal.

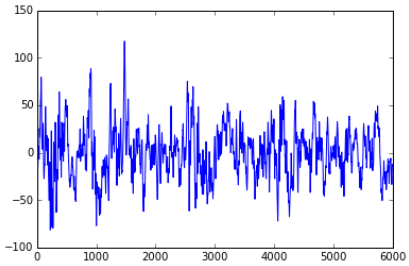

Figure 2.3: Original signal

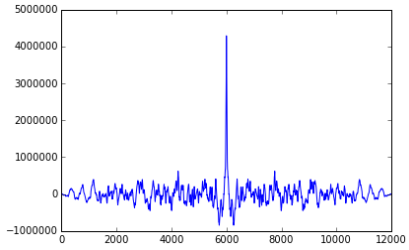

Figure 2.4: Auto-correlation of the signal

2.2.3 CWT peaks

This feature try to find peaks into data. The method used CWT as principal component and is efficient on noisy data. The feature is built as so :
- Perform a CWT (Continous Wavelet Transform) on the signal.
- Identify relative maxima at each row of the resulting matrix.
- Filter the resulting array by selecting values over a certain threshold. The CWT is an alternative to STFT (Short Time Fourier Transform). The main difference between both, is the usage of wavelets in the CWT, contrary to STFT which used simple FFTs. The CWT is commonly visualize by a scalogram.

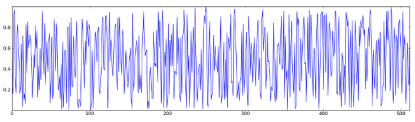

Figure 2.5: Original signal

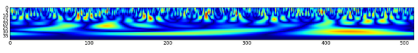

Figure 2.6: Scalogram of the signal after CWT application

2.2.4 Wavelets coefficients

This feature use the discrete wavelet transform (DWT) as principal component. By computing coefficients at each levels, we can have a very accurate characterization of the signal into time. Here is the method used to compute the feature :
- Perform a DWT on the signal and select coefficients arrays.
- For each level of the decomposition, compute the euclidean norm on coefficients.
Mathematically, a wavelet transform is the integral transform defined as :

$$[W_\psi f(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} f(x)\,dx \tag{2.2}$$

And the wavelet coefficients are given by :

$$c_{jk} = \left[W_\psi f\right]\left(2^{-j}, k2^{-j}\right) \tag{2.3}$$

In our case we just compute :

$$r = \sqrt{\sum_{i=1}^{n} c_i^2} \tag{2.4}$$

2.2.5 Spectrum power for different bands

This feature perform a Gabor transform on the signal. The frequencies bands selected match with the different eegs bands. The Gabor transform of a signal x(t) is defined as :

$$G_x(t,f) = \int_{-\infty}^{\infty} e^{-\pi(\tau-t)^2} e^{-j2\pi f\tau} x(\tau)\,d\tau \tag{2.5}$$

The Gabor transform is inspired by the STFT (Short Time Fourier Transform). The STFT is commonly visualize by a spectrogram.

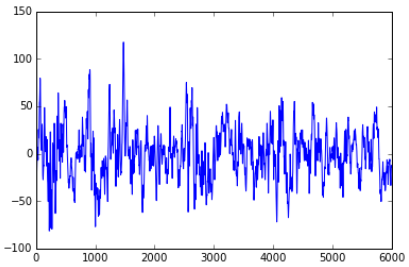

Figure 2.7: Original signal

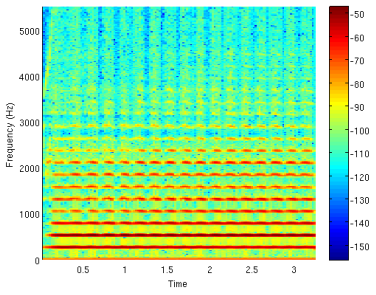

Figure 2.8: Spectrogram of the signal

2.3 Fractal analysis

2.3.1 Detrended fluctuation analysis

Compute the detrended fluctuation analysis (DFA) exponent. The DFA method is used to determine the self-affinity of a signal. In the world of fractal, the self-affinity is the self-similarity of an object, which is any object that have similar part of itself.

This feature is a little bit similar to the auto-correlation one, but detect with more accuracy patterns inside the signal.

2.3.2 Petrosian fractal dimension

A fractal dimension D, is a ratio providing a statistical index of complexity comparing how detail in a pattern (strictly speaking, a fractal pattern) changes with the scale at which it is measured. It is also a measure of the space-filling capacity of a pattern that tells how a fractal scales differently from the space it is embedded in. There are a lot of definitions of fractal dimension, all have a specific application and should be considered separately. In the eeg case, the Petrosian definition is a good one to characterize patterns inside signals. The Petrosian fractal dimension is defined as :

$$D = \frac{log_{10} n}{log_{10} n + log_{10} \frac{n}{n+0.4N_\Delta}} \tag{2.6}$$

Where n is the length of the sequence (number of points), and N_Δ is the number of sign changes (number of dissimilar pairs) in the binary sequence generated.

2.3.3 Hjorth parameters

These are parameters commonly used in the study of eeg signals. Here 2 parameters are computed : the mobility and the complexity. The mobility represent the standard deviation of the power spectrum. It can correct the 'FFT frequencies' or 'Spectrum power' feature. For a given signal y, the mobility parameter is defined as :

$$Hjorth_M = \sqrt{\frac{var(y(t)\frac{dy}{dt})}{var(y(t))}} \tag{2.7}$$

The complexity represent the change in frequency into the signal. For a given signal y , the complexity is defined as :

$$Hjorth_C = \frac{Hjorth_M(y(t)\frac{dy}{dt})}{Hjorth_M(y(t))} \tag{2.8}$$

2.3.4 Higuchi fractal dimension

Another fractal dimension, introduce by Higuchi. This one characterize MEG, which are magnetic fields produced by neural activity of the brain. Here we consider a time serie x of length N. We have to construct a new time serie x_m^k where m indicates the initial time value, k indicates the discrete time interval between points (delay) as :

$$x_m^k = x(m), x(m+k), x(m+2k), ..., x(m+[\frac{N-m}{k}]k), m = 1, 2, ..., k \tag{2.9}$$

For each time series constructed x_m^k, the average length $L_m(k)$ is computed as :

$$L_m(k) = \frac{\sum_{i=1}^{(N-m)/k} |x(m+ik) - x(m+(i-1)k)|(n-1)}{[\frac{N-m}{k}]k} \tag{2.10}$$

An average length is computed for all time series having the same delay k, as the mean of the k lengths $L_m(k)$ for m=1,2,...,k. This procedure is repeated for each k ranging from 1 to k_{max}, leading to a sum of average lengths L_k for each k :

$$L(k) = \sum_{m=1}^{k} L_m(k) \tag{2.11}$$

The total average length for scale k, L_k, is proportional to k^{-D} where D is the fractal dimension we are looking for. In the curve of ln(L(k)) vs ln(1/k), the slope of the least squares linear best fit is the estimate fractal dimension, e.g the Higuchi fractal dimension.

2.3.5 Fisher information

This feature is a particular measure of variance. The Fisher information is defined as :

$$I(\theta) = \mathrm{E}\left[\left(\frac{\partial}{\partial\theta}\log f(X;\theta)\right)^2 \middle| \theta\right] = \int \left(\frac{\partial}{\partial\theta}\log f(x;\theta)\right)^2 f(x;\theta)\,\mathrm{d}x \tag{2.12}$$

Where E[...|θ] represent the conditional expectation on values X regarding the probability function $f(x;\theta)$, for a given θ. $The\ \frac{\partial}{\partial\theta}\log f(x;\theta)$ term represent the score function.

2.3.6 Spectral entropy

The spectal entropy measure the dispersion of the spectrum. It's based on the power spectral density (PSD) of the signal. The method to compute it is :
- Compute the PSD and normalize it
- Calculate the entropy regarding the Shannon definition

$$H_{spec} = -\sum_{k=1}^{M} P_k log_2 P_k \tag{2.13}$$

Where P_k is the PSD for signal k.

2.3.7 SVD entropy

The SVD entropy is an indicator of how many vectors are needed for an adequate explanation of the data. It measures feature-richness in the sense that the higher the entropy of the set of SVD weights, the more orthogonal vectors are required to adequately explain it. This entropy measure use singular value decomposition (SVD) as mathematical component. If the input is a signal :

$$[x_1, ..., x_n] \tag{2.14}$$

Then the delay vectors are built as so :

$$y(i) = [x_i, x_{i+\tau}, ..., x_{i+(d_E-1)\tau}] \tag{2.15}$$

Where $\tau\ is the delay and \mathrm{d}_E$ is the embedding dimension. Then we can construct the embedding space defined here by :

$$Y = [y(1), y(2), ..., y(N-(d_E-1)\tau)]^T \tag{2.16}$$

The SVD is then performed on Y to produce M singular values, $\theta_1, ..., \theta_M$. The SVD entropy is then defined as :

$$H_{SVD} = -\sum_{k=1}^{M} \overline{\theta_k} log_2 \overline{\theta_k} \tag{2.17}$$

Where,

$$\overline{\theta_k} = \frac{\theta_k}{\sum_{j=1}^{M} \theta_j} \tag{2.18}$$

Chapter 3

Other features

This chapter describe all features not used in the model but implemented as function into the source code. These are alternative or abandoned features as they lower the score of the prediction model.

3.1 Signal processing

3.1.1 Whelch method

The Whelch method is used for estimating the power of a signal at different frequencies. This method permit to compute a spectrogram of the signal, which is another representation of the signal in the frequency domain. This method is quite efficient to estimate signal power in noisy datas.

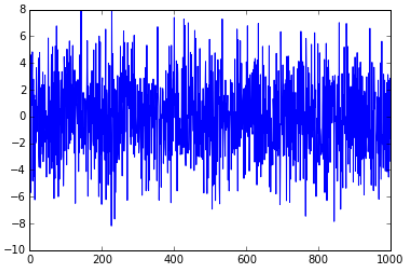

Figure 3.1: Original signal

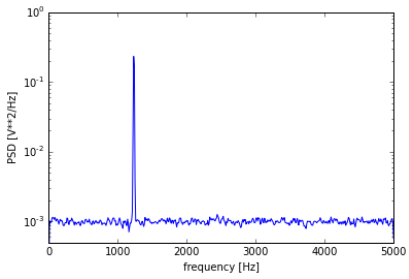

Figure 3.2: Power signal analysis with Welch method

3.2 Fractal analysis

3.2.1 Correlation dimension

This is another fractal dimension which as the the advantage to be straightforward and quickly calculated. The correlation dimension is based on the correlation inte-

gral defined as :

$$C(\varepsilon) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{\substack{i,j=1 \\ i\neq j}}^{N} \Theta(\varepsilon - ||\vec{x}(i) - \vec{x}(j)||), \quad \vec{x}(i) \in \mathbb{R}^m \tag{3.1}$$

where N is the number of considered states $\vec{x}(i)$, ε is a threshold distance, $||\cdot||$ a the euclidean norm and $\Theta(\cdot)$ the Heaviside step function. This quantity represent the mean probability that states at two different times are close.

3.2.2 Lyapunov exponent

This exponent quantify the rate of separation of infinitesimally close trajectories. This is an alternative to the fractal correlation dimension. Let's consider a function f. We can compute the error at a step n with the computed error at the step before by :

$$\varepsilon_n = f(u_{n-1} + \varepsilon_{n-1}) - f(u_{n-1}) \tag{3.2}$$

When 2 successives errors tend to 0, the instant amplification of the error is measured by $f'(u_{n-1})$. This amplification is varying from a step to the next one, which lead to calculate the product of errors division :

$$|\frac{\varepsilon_n}{\varepsilon_0}| = \prod_{i=1}^{n} |\frac{\varepsilon_i}{\varepsilon_{i-1}}| = \prod_{i=1}^{n} |f'(u_{i-1})| \tag{3.3}$$

With $\varepsilon_n = e^{\lambda n}\varepsilon_0$ and a limit consideration we defined the Lyapunov exponent :

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \ln(|f'(u_{i-1})|) \tag{3.4}$$

Chapter 4

Model approaches

4.1 The classic approach

To choose the best model, I just compute all previous features (describe on feature chapter) on the train dataset (X train). There is only one exception for wavelets coefficients : I build 40 features with them. I choose 5 different wavelet types : daubechies, symlets, coiflets, biorthogonal, reverse biorthogonal. For each wavelet type I am computing 8 levels of decomposition. The number of levels is the accuracy

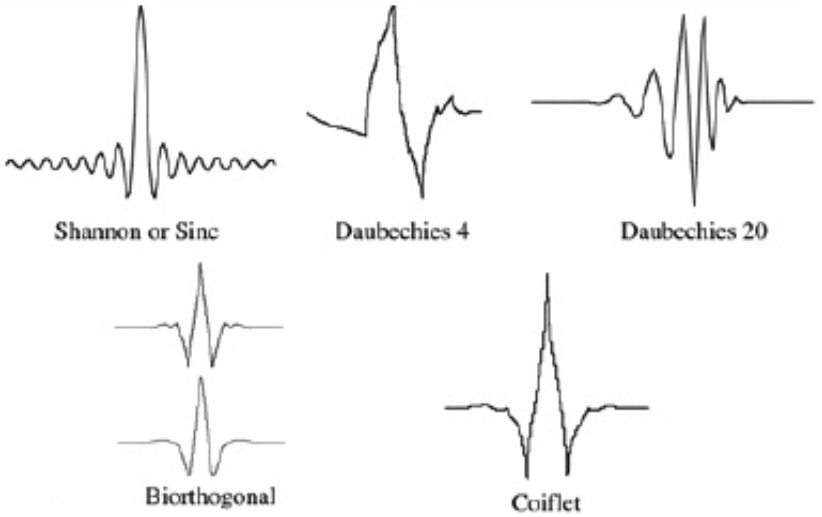

Figure 4.1: Examples of wavelets

of the decomposition. The maximum of levels M rely on the number of points N_p of the signal by the formula :

$$M = \lfloor log_2 N_p \rfloor \tag{4.1}$$

With $\lfloor x \rfloor$ defining the floor function as :

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\} \tag{4.2}$$

In this case $M = 12$, so a good choose could be 4 levels of decomposition but I prefer to be sure of my feature and choose twice more levels to more characterize eeg signals.

After computing all features, I normalize the matrix with the classic formula :

$$X_{norm} = \frac{X - mean(X)}{std(X)} \tag{4.3}$$

Then I split this dataset into 2 parts : 80% for the training and 20% for the test. I train each type of classifier on the 80% and print the score, obtained with 20% of data test. The best score is for the SVC with a Gaussian radial basis (RBF) kernel: 0.8350. The RBF is defined as :

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \tag{4.4}$$

4.2 The special KNN approach

4.2.1 Dynamic Time Warping

Because we are dealing with time series, we can have a slightly different approach than the previous one. Instead of compute features independently on signals and train our model on them, we can simply compare time series themselves. To create an efficient model and compare each time series with the others, we are using here a KNN, but we change the distance measure of it. The distance measure is here a DTW or dynamic time warping and not the classic euclidean measure. The main difference is that DTW measure the similarity of 2 sequences independent of certain non-linear variations in the time dimension. The main issue of this method is a very

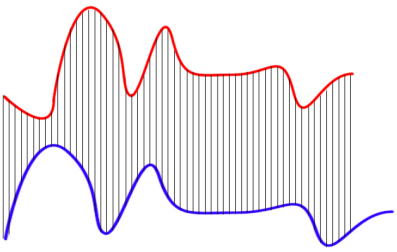

Figure 4.2: Euclidean matching

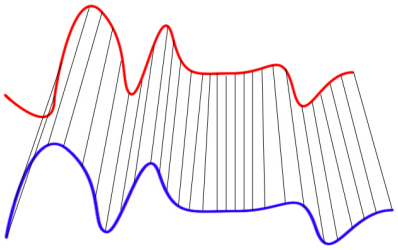

Figure 4.3: DTW matching

very slow computation. Even if we reduce the number of signal points by 60, the method still compute slowly (the reason is the similarity matrix compute for each pairs of sequence).

4.2.2 Itakura-Saito

Another KNN approach is to replace the traditional measure with the Itakura-Saito distance. Here we are not comparing 2 time series but their respective spectrum (spectrum in the frequential domain). The Itakura-Saito distance is defined as :

$$D_{IS}(P(\omega), \hat{P}(\omega)) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[\frac{P(\omega)}{\hat{P}(\omega)} - \log \frac{P(\omega)}{\hat{P}(\omega)} - 1 \right] d\omega \tag{4.5}$$

$P(\omega)$ is the original spectrum and $\hat{P}(\omega)$ is an approximation of P in the formula. If we consider the approximate P as another spectrum and have a small distance at the end, then both of the spectrum are not different. Nb : If the score of the KNN is not so good (0.5 or 0.6 for example), we can still put the vectorized prediction as a new feature into the SVC. The main reason is that, this measure can detect some classes better than other features in the classification model.

Bibliography

Signal processing

Saeid Sanei and J.A. Chambers *EEG Signal Processing*

Rodrigo Quian Quiroga *Quantitative analysis of EEG signals : Time-frequency methods and Chaos theory*

Fractal analysis

Saeid Sanei and J.A. Chambers *EEG Signal Processing*

Rosana Esteller, George Vachtsevanos, Javier Echauz and Brian Litt *A Comparison of Waveform Fractal Dimension Algorithms*

Monica Cusenza *Fractal Analysis of the EEG and Clinical Applications*

Rodrigo Quian Quiroga *Quantitative analysis of EEG signals : Time-frequency methods and Chaos theory*

Svorad Štolc and Anna Krakovská *EEG: Spectral Characteristics vs. Correlation Dimension*

Classic model approach

Saeid Sanei and J.A. Chambers *EEG Signal Processing*

KNN approach

Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista,Brandon Westover, Qiang Zhu, Jesin Zakaria and Eamonn Keogh *Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping*

Jacob Benesty *Springer Handbook of Speech Processing*

List of Figures

List of Tables