1 Wavelet transform - filter width

We use continuous wavelet transform as a transformation to get a multiscale representation of one dimensional signals. We fix the mother wavelet to the Morlet wavelet transform. The implementation in **pywt**, the open source Python library dedicated to this, does not provide the possibility to compute automatically the number of scales and the frequencies relative to them, given a signal of length N and its sampling period T_s. In the following we compute these quantities

The Morlet mother wavelet is equal to $\Psi(t)$

$$\Psi(t) = \frac{1}{\sqrt{\pi B}} e^{-\frac{t^2}{B}} e^{i2\pi Ct} \tag{1}$$

where B is related to the variance of the gaussian and C to the frequency of the oscillations. To understand which is the optimal choice of the scale we consider the filter in the frequency domain.

$$\mathcal{F}(\Psi)(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi B}} e^{-\frac{t^2}{B}} e^{i2\pi Ct} e^{-i\omega t} dt$$
$$\mathcal{F}(\Psi)(\omega) = \frac{1}{\sqrt{2\pi}\sqrt{\pi B}} \int_{-\infty}^{+\infty} e^{-\frac{t^2}{B}} e^{-2i\frac{\omega - 2\pi C}{2}t} dt$$

By completing the square, this corresponds to

$$\mathcal{F}(\Psi)(\omega) = \frac{1}{\sqrt{2\pi^2 B}} e^{-\frac{(\omega-2\pi C)^2 B^2}{4}} \int_{-\infty}^{+\infty} e^{-\frac{1}{B}\left(t + j\frac{\omega - 2\pi C}{2}B\right)^2} dt$$

The second term is an analytical function. By computing the integral on a rectangular path on the complex plane we get zero contribute from the vertical edges. The horizontal path corresponds to the gaussian integral on the real line.

$$\mathcal{F}(\Psi)(\omega) = \frac{1}{\sqrt{2\pi^2 B}} e^{-\frac{(\omega-2\pi C)^2 B^2}{4}} \sqrt{\pi B} \propto e^{-\frac{(\omega-2\pi C)^2}{(2/B)^2}} = e^{-\frac{4\pi^2(\omega/2\pi - C)^2}{4/B^2}} \tag{2}$$
$$= \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(f-C)^2}{1/(\pi^2 B^2)}\right] \tag{3}$$

The filter has a gaussian shape, in particular we can now interpret the different constants

$$f_c = C, \text{ central frequency}$$
$$\sigma_f = \frac{1}{\sqrt{2}\pi B}, \text{filter width}$$

In the first version of MT-MKL the parameters values were $B = 1$ s, $C = 1$ Hz. This values are such that the width of the gaussian in the frequency domain is fixed at $\sigma_{width} \sim 6 \text{ s}^{-1}$.

The important part is understanding the relation between the filter width and the scaling factor of the gaussian distribution. In order to address this point we consider the influence of the scaling factor on the Fourier transform. We know in particular that, $C = 1$ Hz and $B = 1$ s for the case of $f_s = 1$ Hz. Since in our case the sampling frequency is equivalent to 1 kHz, the scaling factors and the decay time for the Morlet mother wavelet correspond respectively to 1 kHz and 1 ms. We now interpret the relation with the scale factor a. A dilation in time corresponds to a shrinkage in the frequency domain and vice versa.

By increasing a we get the following

$$\mathcal{F}(\Psi_a)(\omega) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(af - C)^2}{1/(\pi B)^2}\right] = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(f - C/a)^2}{1/(\pi a B)^2}\right]$$

where both the central frequency and the filter width are compressed of a factor a. In particular we get the following

$$f_c^{(a)} = \frac{C}{a}, \qquad a = 1 \to f_c^{(1)} = f_s \tag{4}$$

$$\sigma^{(a)} = \frac{1}{\sqrt{2\pi a B}}, \qquad a = 1 \to \sigma^{(1)} = \frac{1}{\sqrt{2\pi B}} \tag{5}$$

Still we want the representation to be redundant. We initialize the smallest scale at the value $s = 2.1$. This corresponds to a central frequency $f_c^{(2.1)} = 476$ Hz (below Nyquist). We get then the central frequency at higher scales by requiring the next frequency to corresponds to the point where the height of the normal distribution is 0.95

$$\exp\left[-\frac{\left(f_c^{(a+1)} - f_c^{(a)}\right)^2}{2\left(\sigma^{(a)}\right)^2}\right] = 0.95 \tag{6}$$

This returns the following condition

$$-\frac{\left(f_c^{(a+1)} - f_c^{(a)}\right)^2}{2\left(\sigma^{(a)}\right)^2} = \ln(0.95) \to \left(f_c^{(a+1)} - f_c^{(a)}\right)^2 = -2\left(\sigma^{(a)}\right)^2 \ln(0.95)$$

$$f_c^{(a+1)} = f_c^{(a)} - \sigma^{(a)}\sqrt{-2\ln(0.95)}$$

2 Similarity measures based on synchronization

In the first version of MT-MKL we use the entire time series to evaluate pairwise similarity betweent recordings acquired on the same subject. We resort to standard measures of signal processing. Note that this measures are not applied on the raw time series but on the wavelet coefficients for each scale. To introduce the similarity measures we denote the generic time series - which are vectors of complex numbers, corresponding to the wavelet coefficients at a fixed scale - as $x(t)$ and $y(t)$, each of length T.

Correlation this represents the most standard measure, and it is the dot product between the two time series. It corresponds to

$$\text{correlation}(x, y) = \frac{\langle x(t), y(t)\rangle}{\|x(t)\|_2 \|y(t)\|_2} \tag{7}$$

with correlation$(x, y) \in [-1, 1]$. This corresponds to the most basic kernel, and we think, a posteriori, it is not the best way to evaluate a comparison between two time series in case where shift invariance is needed.

Phase locking value (PLV) this measure is common in the neurological context and represents the measure of pairwise phase synchrony between two recordings. We compute PLV as follows

$$\mathrm{PLV}(x, y) = \frac{1}{T} \sum_{t=1}^{T} |\exp[i(\phi(x) - \phi(y))]| \tag{8}$$

with $\phi(x(t))$ and $\phi(y(t))$ the phases for the two time series. In this case $\mathrm{PLV}(x, y) \in [0, 1]$.

Cross correlation in order to obtain a shift invariant measure of similarity we resort to cross correlation. This quantity corresponds to a measure of correlation between the spectra of the two signals.

$$\text{cross correlation}(x, y) = \frac{\langle |\mathcal{F}(x)|, |\mathcal{F}(y)| \rangle}{\|\mathcal{F}(x)\|_2 \|\mathcal{F}(y)\|_2} \tag{9}$$

In the first version of MT-MKL we use MATLAB as software, and this measure was computationally quite intensive. We splitted each time series in one hundred non overlapping of length 6 s. We use now the Python functions. The split is not necessary, we compute the Fourier transform, then the pointwise absolute value and finally we perform the cross correlation measure.

In the first version of MT-MKL no cross correlation measure has been selected from the algorithm. This highlights several unconsidered aspects (i) the cross correlation as measured in MATLAB is likely to be a rough estimate of the pairwise spectra similarity, given that it is an average computed over non overlapping chuncks of 6 seconds (ii) presence of high correlation between time series, rather than the individuation of specific patterns.

3 Next steps

Analysis over all patients The first version of MT-MKL is a preliminary work, for which we use a subset of patients, since the preprocessing step is computationally expensive. We run now the algorithm over all the patients, with the wavelet filters as evaluated in Section 1.

Retrain using cross correlation only In the first version we compared the entire time series, building kernel measures for each patient. The fact that the cross correlation is never selected is a possible evidence of strong correlations. We train the method using only cross correlation kernels to see how the classification - selection performance changes.

Non convexity The method is not convex and the initialization value is extremely important. We resort to accelerated methods to minimize the functional.

4 Learning model and minimization methods

For each patient we have a matrix $\boldsymbol{X}^{(p)} \in \mathbb{R}^{c_p \times T}$ and a vector of labels $\boldsymbol{y}^{(p)}$ which denotes the pathological or physiological nature of each signal in $\boldsymbol{X}^{(p)}$.

Note that the number of channels c_p varies across patients, and the proportion of epileptic channels is not uniform across the population considered. We denote with $(K_1^{(p)}, \dots, K_k^{(p)})$, where $K_j^{(p)} \in \mathbb{R}^{c_p \times c_p}$ the set of k kernels for the generic patient p. The decision function $f^{(p)}$ for a patient p and a channel x is defined as:

$$f^{(p)}(x) = \alpha_0^{(p)} + \sum_{i=1}^{c_p} \left[\alpha_i^{(p)} \sum_{j=1}^{k} w_j K_j^{(p)}(x_i, x) \right], \tag{10}$$

where $\alpha_i^{(p)}$ denotes the i-th component of the regression parameter $\boldsymbol{\alpha}^{(p)}$ specific for each patient p. Having separate parameters ($\boldsymbol{\alpha}^{(p)}$ and $\boldsymbol{w}$) is fundamental for the resolution of our problem. In fact, $\boldsymbol{\alpha}^{(p)}$ allows to better approximate the labels $\boldsymbol{y}^{(p)}$ by capturing the variance of each patient, while $\boldsymbol{w}$ combines the kernels by weighting them and, as it holds across patients, it provides relevant indication of the most discriminative kernels.

In order to obtain interpretable results and a more stable solution we also add an $\ell_1\ell_2$ penalty on $\boldsymbol{w}$ and $\boldsymbol{\alpha} = (\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(n)})$. By considering all the patients, our goal translates into minimizing the following objective function:

$$\begin{aligned} \underset{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(n)},\boldsymbol{w}}{\text{minimize}} \Bigg\{ & \sum_{p=1}^{n} \left(\ell_{f^{(p)}} \left(\boldsymbol{X}^{(p)}, \boldsymbol{y}^{(p)} \right) + \lambda (r_\lambda \|\boldsymbol{\alpha}^{(p)}\|_1 + (1 - r_\lambda) \|\boldsymbol{\alpha}^{(p)}\|_2^2) \right) \\ & + n\beta \left(r_\beta \|\boldsymbol{w}\|_1 + (1 - r_\beta) \|\boldsymbol{w}\|_2^2 \right) \Bigg\} \\ & \text{s.t. } w_j \geq 0 \text{ for each } j = 1, \dots, k \end{aligned} \tag{11}$$

where $\ell_{f^{(p)}} \left(\boldsymbol{X}^{(p)}, \boldsymbol{y}^{(p)} \right) = -\sum_{i=1}^{c_p} \log(1 + \exp(-y_i^{(p)} f^{(p)}(x_i)))$ is the negative log-likelihood of the logistic probability function and r_λ and r_β are the elastic-net penalty penalty ratios on $\boldsymbol{\alpha}$ and $\boldsymbol{w}$, respectively. The elastic-net penalty benefits indeed from the well-known stability property of the ℓ_2 regularization term [**?**].

4.1 Model upgrades

We modify the model in such a way to

1. Normalize the learning model with respect to c_p. If we normalize the output for the prediction function with respect to the number of training channels (more homogeneity of the kernel weights) - in the other case the $\alpha^{(p)}$ have to compensate - to get $y_i^{(p)} f^{(p)}(x_i)$ such to exceeds $\frac{1}{2}$.

2. Normalize the loss function with respect to c_p; if not the weight for the loss term in the functional varies across patients and depends on the number of channels

3. Normalize the penalty term over channels with respect to the number of channels used

$$f^{(p)}(x) = \alpha_0^{(p)} + \sum_{i=1}^{c_p} \left[\alpha_i^{(p)} \sum_{j=1}^{k} w_j K_j^{(p)}(x_i, x) \right] \tag{12}$$

$$\ell_{f^{(p)}} \left(\boldsymbol{X}^{(p)}, \boldsymbol{y}^{(p)} \right) = -\frac{1}{c_p} \sum_{i=1}^{c_p} \log \left(1 + \exp \left(-y_i^{(p)} f^{(p)}(x_i) \right) \right) \tag{13}$$

$$\begin{aligned} \underset{\boldsymbol{\alpha}^{(1)},\dots,\boldsymbol{\alpha}^{(n)},\boldsymbol{w}}{\text{minimize}} \Bigg\{ \frac{1}{n} \sum_{p=1}^{n} \Bigg(\frac{1}{c_p} \Bigg(\sum_{i=1}^{c_p} - \log \left[1 + \exp \left[-y_i^{(p)} f^{(p)}(x_i) \right] \right] \Bigg) + \\ \lambda \left(r_\lambda \|\boldsymbol{\alpha}^{(p)}\|_1 + (1 - r_\lambda) \|\boldsymbol{\alpha}^{(p)}\|_2^2 \right) \Bigg) \\ + \beta \left(r_\beta \|\boldsymbol{w}\|_1 + (1 - r_\beta) \|\boldsymbol{w}\|_2^2 \right) \Bigg\} \\ \text{s.t. } w_j \geq 0 \text{ for each } j = 1, \dots, k \end{aligned} \tag{14}$$

4.2 Update rule based on GD

$$\frac{\partial f^{(p)}(x_i)}{\partial w_t} = \sum_{i=1}^{c_p} \left[\alpha_i^{(p)} K_t^{(p)}(x_i, x) \right] \tag{15}$$

$$\frac{\partial \ell_{f^{(p)}}}{\partial f^{(p)}} = \frac{1}{1 + \exp \left[-y_i^{(p)} f_i^{(p)} \right]} y_i^{(p)} \exp \left[-y_i^{(p)} f_i^{(p)} \right] \tag{16}$$

$$\frac{\partial f^{(p)}(x_i)}{\alpha_t^p} = \sum_{j=1}^{k} w_j K_j^{(p)}(x_t, x) \tag{17}$$