SESSION 5: ICA/Wavelets

1. Case study: Independent Component Analysis (ICA) and Electroencephalogram (EEG)

Objectives

- Applying ICA for automatic removal of artefacts
- Getting insight into the fastICA algorithm

Background: ICA and artefact removal in EEG

Independent component analysis (ICA) is a tool that belongs to a class of methods called Blind Source Separation techniques. In the Blind Source Separation (BSS) problem, the observed time courses $x(t)=[x_1(t),x_2(t),...,x_K(t)]^T$, with t=1...N and K the number of channels, are the result of an unknown linear mixture of a set of source signals $s(t)=[s_1(t),s_2(t),...,s_K(t)]^T$, with t = 1...N. The mixture process can be written as: x(t)=A s(t), where A is the unknown mixing matrix. Note that both mixture and sources are assumed to be zero-mean. The goal is to estimate the **de-mixing matrix H** or its (pseudo-)inverse, the estimated **mixing matrix A**, such that y(t)=H x(t) is equal to the unknown source signals in s(t), except for a scaling and permutation. We can write this similarly as x(t) = A y(t).Unless there are extra constraints imposed, it is in general impossible to solve this problem. The general additional constraint imposed by ICA is that the sources need to be statistically independent.

In many EEG paradigms, artefact removal is important to enhance signal quality. In particular, eye blinks can distort the recording, most notably in the frontal channels (e.g. Fpz, F_3 or F_4). ICA allows to separate eye blinks into one or a few components, provided they are independent from other EEG content. The ICA components contain both temporal and spatial information. The temporal information is the extracted ICA component itself. Eye blink components are relatively small in amplitude, except for the peaks indicating eye blinks. Spatial information can be derived from the mixing matrix A, since it contains information about the contribution of each component to each channel. The spatial information can be presented in a topographic plot. By inspection of the temporal and spatial characteristics, eye blink components can be identified and unmixed, yielding a clean signal [1].

Programming exercise

Exercise 5_1_1.m

1.1. Objective: In this exercise you will explore the properties of fastICA on mixed signals.

For this exercise 4 sources are being used. The first is 10s of an ECG recording at 500Hz. The other 3 sources simulate noise: white gaussian noise, powerline interference and a sawtooth to represent e.g. instrument noise. Imagine we had a way to measure different mixtures of these four signal components simultaneously. This is simulated by creating different mixtures of the sources.

1) Apply fastICA to the give mixture X to obtain the estimated sources.

2) Normalize all the estimated sources such that their mean value is zero and their variance is 1. Hint: check the list of Matlab functions for this session.
3) Automatically match (pair) the estimated sources with the corresponding original sources (e.g. ecg with its estimate, gaussian noise with appropriate estimate etc) using correlation as similarity metric. Plot every estimate and corresponding matched source next to each other.
4) Calculate the Root Mean Squared Error - RMSE between the corresponding paired (matched) estimated sources and the original ones.
Hint: Make sure the matched components (using the correlation) have the same sign; otherwise negate the estimated component.

$$RMSE = \sqrt{\frac{\sum_i (y_i - s_i)^2}{N}}$$, *N - number of samples,* s_i *- source,* y_i *- its estimate*

Imagine now we measure six mixtures of these 4 sources

5) Apply FastICA to mixture X2, produced using the given mixing matrix A2. Match the sources and compute the RMSE. Can the algorithm identify the number of sources? How does it work? Again, plot the obtained estimates and the corresponding Sources next to it.
Secondly, repeat the above for X3, the same mixture with additional noise. What do you conclude?

Using the original mixtures X, we want to remove the sawtooth.

6) Use FastICA to remove the sawtooth by modifying the demixing matrix. Plot the corresponding mixture signals x(t) before and after removing the sawtooth signal.

7) Pick one of the 4 mixtures you obtained from unmixing the sawtooth. Use the filtering techniques you saw earlier in this course to remove the noise influences on the ECG. What kind of noise is present? Which filter(s) could be useful? Do you think they would also work to remove the sawtooth?
Plot a comparison of the original ecg and your filtered version. Include other useful plots if necessary (e.g. power spectra, filter diagrams, ...).

Questions:
1. We mentioned that you sometimes need to change the sign of detected components. Why is this?
2. Suppose an additional gaussian white noise source was present. Would FastICA be able to separate both of them? Why (not)?

What to submit
- MATLAB script with descriptive comments (complete *session_5_1_1.m*)
- Conclusions on the subparts with supporting evidence. Discussion of all your results and answers to all questions, clarified by plots when you consider it relevant.
- Plots of the original sources next to their best-matching components extracted from the mixtures for all scenarios. Mention the RMSE values.
- Plots of the mixtures before and after removal of the sawtooth, side by side
- Discussion of the filtering after removal of the sawtooth. Plot of a selected mixture before and after filtering and comparison with the original ECG source.

Exercise 5_1_2.m

1.2. Objective: processing of the EEG with artefacts. Getting familiar with real data problems. In this exercise, EEG data of an auditory attention experiment is provided.
EEG.mat contains 20 seconds of EEG data (sample rate 128 Hz, 12 channels) in the variable *data*. The variable *channels* contains the names of the channels. Their location on the skull is indicated in Figure 1. You are asked to:

1) Visualize the data using the function *eegplot_simple.m (*use the help command for more information). Do you notice eye blinks? Indicate them in the plot.
2) Use the fastICA algorithm. Look at the temporal and spatial information. The spatial information can be obtained using *topoPlot.m.* Use *help* topoPlot if needed. The topographic representation is explained in Figure 1. Which components represent eye blinks?
3) Remove the eye blink artefacts similarly as you did with the sawtooth in the previous exercise. Visualize the result.

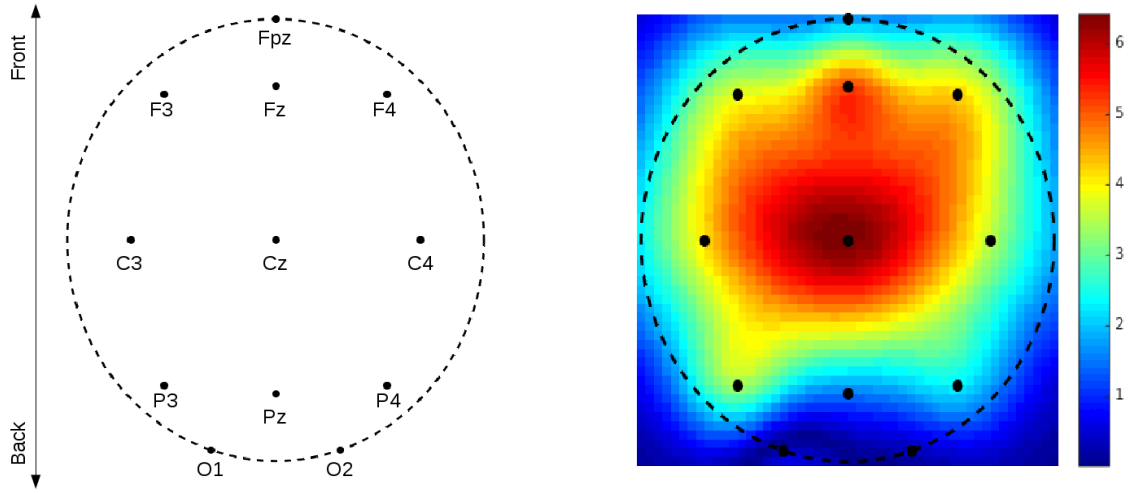


Figure 1: The EEG configuration (left) and an example of a topographic plot (right)

What to submit?

- MATLAB scripts with descriptive comments
- Discussion of all your results and answers to all questions
- Plots of the original EEG with indicated eye blinks
- Plots illustrating the selection of the artefact components (temporal and spatial).
- Plot of the clean EEG indicating the change with regard to the original

2. Illustration: Wavelets on ECG and EEG

Objectives

- Getting familiar with wavelets.
- ECG signal and advantages of wavelets

Background

The Fourier transform or related approaches such as the Gabor transform are very effective when studying signals composed of smooth oscillations.
However, they analyze all frequency regions with the same resolution, which makes them less suitable, e.g. when signals possess sharp transients and discontinuities. For such purposes wavelets are more suitable. As an example, an ECG signal has sharp transients (R peak) which is why wavelets are often used for analysis and compression of ECG signals.

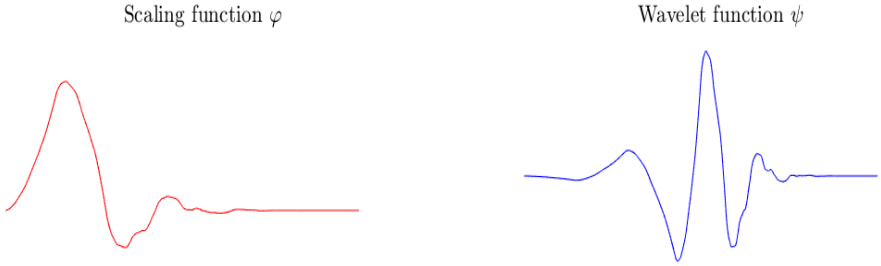
Programming exercise

Exercise 5_2_1.m

In this exercise 15 seconds of previously recorded ECG is used for processing. It has a sampling frequency of 300Hz. Therefore in order to store this signal in the time domain we need 4500 samples. However, we can transform this signal into another domain and reconstruct it using fewer coefficients, each of them corresponding to certain aspects of the signal.

1) Decompose the clean ECG signal with the Daubechies wavelet 4 (db4), using 5 levels of decomposition. See Figure 2 for an example of the wavelet. The scaling function is related to the low-pass filter in the hierarchical filterbank decomposition and computes the 'approximation coefficients' in each filtering stage i.e. the low frequency part. The wavelet function is related to the high-pass filters that compute the 'detail coefficients' in all stages i.e. the high frequency part. Plot the approximation and detail coefficients in the following order (A5, D5, D4, D3, D2, D1) using subplots in one column and add a subplot on top with the time-domain ECG signal (so 7 subplots in total). Make sure that the wavelet coefficients are aligned correctly with the corresponding time points in the ECG signal above.. Study and use the functions *wavedec* and *waverec*, *appcoef* and *detcoef*. Use *subplot* and *stem* Matlab functions for plotting.
2) Keep M=10 wavelet coefficients (approximation and/or details coefficients) with the largest *absolute* amplitudes. Set all the remaining coefficients to zero and reconstruct the signal. Plot the kept coefficients (*stem*) as in question 2. Plot the results of the reconstructed signal and compare it visually with the ECG (plot on the same graph).
3) Compare the reconstructed signal with the shapes of the scaling *phi* and the wavelet *psi* functions from Figure 2. Why can you recognize *phi* at some places and *psi* in others?
4) Repeat the above whilst keeping M=25, 50, 100 coefficients respectively. Plot the appropriate kept coefficients and reconstructed signals along with the original ECG. Comment on the obtained results. Calculate the RMSE for each case and compare.

Figure 2: Daubechies wavelet 4



Questions:

1. Assuming the wavelet decomposition performed previously, which coefficients (A5, D5, D4, D3, D2, D1) would you use to analyze the QRS complexes? Knowing that the sampling frequency is 300 Hz, what is the corresponding frequency range?
2. Here, db4 wavelets have been used. Why would you prefer one wavelet over another?

What to submit

- MATLAB script with descriptive comments
- Discussion of all your results and answers to all questions. Add plots when needed
- For all values of M (10, 25, 50 and 100), provide a series of subplots. The clean ECG on top, overlaid with the reconstruction. Then, below, *stem* plots showing the selected approximation and detail coefficients, each of them in a separate subplot in the aforementioned order. Don't forget to mention the RMSE values!

Exercise 5_2_2.m

We saw that wavelets provide an easy way to characterize the shape of a signal. Hence, it was useful for signal compression in the previous exercise. We can also view the discrete wavelet transform as a filterbank of bandpass filters. This is particularly useful if we want to study a signal in several frequency bands. We will now use this interpretation to classify EEG segments containing epileptic seizure activity.
The file *epilepsy.mat* contains single-channel EEG data sampled at 256Hz. It has already been segmented in (partly overlapping) segments of 2s. So, *data_train* consists of 140 segments of 2s. The corresponding variable *labels_train* indicates if a segment contains seizure activity (1) or not (0).

1) Firstly, visualize your data. Plot all non-seizure segments in one plot, with the seizure segments next to it in another plot. What discriminant property do you observe? How would you quantify this?
   Do you think we should normalize/standardize the raw data?
2) Signal energy is one way to quantify the difference. Given a discrete signal *s*, its energy is defined as:
   $E(s)=\sum_i s_i^2$, that is, as the sum of the squared samples.
   Calculate this energy for all training segments.
3) We can also look at energies in several frequency bands. Decompose each segment using the Daubechies wavelet 4 (*db4*) with 5 levels. What are the frequency ranges for your approximation and detail coefficients? To what EEG rhythms do they correspond (approximately)?
   Calculate the energies contained in the approximation and detail coefficients for each segment. As a result, you obtain a 140x6 feature matrix: the energy of each of the six frequency bands for all 140 segments.
4) Visualize your features using boxplots as in the previous session. Which frequency bands seem most interesting for classification?
5) Calculate both the full energy of the test segments (as in part 2) and the feature matrix of test segment subband energies (as in par 3).
   Solve two classification problems:
   - Use *classify* with the full signal energy as only feature
   - Use *classify* with the subband energies as features

6) Evaluate both classifiers (accuracy, specificity, sensitivity)

Questions:

1. Does it make sense to add the total energy of the segments as an additional feature next to the subband frequencies? Do you expect improvement? Why (not)?

What to submit

- MATLAB script with descriptive comments
- Discussion of all your results and answers to all questions
- All required plots with discussion
- Overview and results of your classifiers

Matlab commands to use

Use the help command in Matlab and learn the following commands: *appcoef, corr, detcoef, norm, sort, subplot, stem, wavedec, waverec, zscore*

References

[1] Jung, T.-P.; Makeig, S.; Humphries, C.; Lee, T.-W.; McKeown, M. J.; Iragui, V. & Sejnowski, T. J. Removing electroencephalographic artefacts by blind source separation. *Psychophysiology*, **2000**, *37*, 163-178

Evaluation

For this session you should write a report, which will be discussed at the oral exam. Please upload your full code and report on Toledo at the latest on **December 22.** Do not include data and packages or scripts we provided, but only the files you wrote or modified. Also, a clear, printed version of your report should be delivered at the ESAT reception by the same date.