Auditory attention detection based on linear regression

EEG signals in general have an extremely poor SNR, in particular when one is interested in one specific neural response to a specific sensory stimulus. This is because EEG is generated by thousands of concurrently active neural sources, which together generate a lot of 'background' EEG that is not related to the applied sensory stimulus. We are therefore often searching for a needle in a haystack, where the target signal is far below the overall noise floor in the EEG.

In this project, we will focus on neural responses to speech signals. It has been shown in many experiments that some neural sources phase-lock ('entrain') to the envelopes of the speech streams presented to the listener. In a cocktail party scenario where two or more speakers talk simultaneously, it has been demonstrated that this entrainment is stronger for the attended speech stream than for the unattended stream(s). Auditory attention decoding (AAD) algorithms aim to identify these neural responses in the EEG to detect to which speaker the listener is attending.

A popular decoding paradigm for AAD is based on **stimulus reconstruction**, in which a decoder is trained to reconstruct the envelope signal (typically bandpass filtered between 1-10Hz) of the attended speech from the EEG signals. A simple but effective method is to use a linear[1] decoder $d(c,\tau)$ such that the reconstructed attended speech envelope $s(t)$ is computed as

$$\hat{s}(t) = \sum_{c} \sum_{\tau=0}^{L-1} x_c(t+\tau)\, d(c,\tau)$$

where $x_c(t)$ is the EEG signal in channel *c,* and where $\tau = 0 \ldots L-1$ are positive time lags (we model the brain as a causal system). The decoder coefficients $d(c,\tau)$ are trained (for all c and τ) by means of a **least-squares regression**, where the squared error between $\hat{s}(t)$ and the envelope of the attended speaker $s(t)$ is minimized. Note that we assume that the attended speaker is known during the training phase.

At test time (on new unseen data), the output of the decoder $\hat{s}(t)$ is correlated to the individual envelopes of the different speakers. The one with the highest correlation coefficient is then selected as the attended speaker.

More information can be found in the following papers, which you can find online:

- J. A. O'Sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor, "Attentional selection in a cocktail party environment

[1] While the brain is a highly non-linear system, such linear models often still work reasonably well in practice.

can be decoded from single-trial EEG," Cerebral Cortex, p. bht355, 2014. doi: 10.1093/cercor/bht355

- W. Biesmans, N. Das, T. Francart and A. Bertrand, "Auditory-Inspired Speech Envelope Extraction Methods for Improved EEG-Based Auditory Attention Detection in a Cocktail Party Scenario," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 5, pp. 402-412, May 2017.
  doi: 10.1109/TNSRE.2016.2571900
- A. Aroudi and S. Doclo, "Cognitive-driven Binaural LCMV Beamformer Using EEG-based Auditory Attention Decoding," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 406-410.
  doi: 10.1109/ICASSP.2019.8683635

Based on this high-level description and the above papers, you should be able to implement a system for auditory attention detection, train and test it,
- on the provided 64-channel data,
- on the provided 24-channel data,
- on your own recorded data.

Experiment with different hyperparameters (e.g. length of the window used to do the reconstruction) and with different settings (e.g. train a generic model or a subject-specific one), compare the results, and discuss your findings.

<u>Note</u>: when trying out different hyperparameters, you should NOT optimize on the test data, as that should result in a system that overfits to this test data and does not generalize to new previously unseen data. Always use a separate split for tuning hyperparameters (validation data).