Plotting EEG Data in Python

1 Visualizing EEG Data in Python

When starting to visualize an EEG dataset, the first step is to inspect a single trial. After this initial inspection, we can examine all recorded epochs to assess the dataset's overall quality.

The easiest way to visualize an EEG dataset is by using MNE's `plot()` function, which plots epochs with:

- **Y-axis**: Channel names.
- **X-axis**: Epoch numbers.

Above the plot, the ratio of each event type can also be displayed.

To analyze the average response, we can plot **average epochs** to focus on ERP (Event-Related Potential) responses. This technique is widely used in EEG research as it filters out irrelevant responses, leaving only activity that occurs consistently across repetitions, both in terms of latency and electrode location.

Other visualizations include topographic maps and individual channel plots, which provide a comprehensive overview of the dataset.

1.1 Power Spectral Density (PSD) Plot

MNE's `.compute_psd()` method generates a Power Spectral Density (PSD) plot, which calculates the amplitude of various frequencies in the signal. Similar to a histogram, the PSD plot shows how much power each frequency has across a continuous range. In a PSD plot:

- **X-axis**: Frequencies in Hz.
- **Y-axis**: Amplitude for each frequency.
- Separate lines for each electrode illustrate frequency data across different scalp regions, with color coding indicating electrode positions.

In a typical EEG PSD plot, you might observe:

- A **60 Hz spike**, often due to electrical interference from power lines.

- A **180 Hz spike**, which is a harmonic effect of the 60 Hz signal.

- A **10 Hz spike in the alpha band**, which is a common brainwave frequency associated with relaxed wakefulness.

2   Time and Frequency Domains

For EEG data used in a P300 speller system, visualization in the **time domain** is ideal, as the P300 component appears as a distinct positive deflection around 300 ms after a stimulus. Typically, EEG data for P300 detection is filtered in the 1–30 Hz range, removing low-frequency drifts and high-frequency noise. This filter range covers delta, theta, alpha, and beta bands, which are associated with brain activity related to cognitive processing.

To optimally filter P300 data, a bandpass filter around 0.1–30 Hz is recommended.

2.1   Sampling Rate

To reliably capture the P300 component and relevant frequencies, a high sampling rate is essential. According to the Nyquist theorem, the sampling rate should be at least twice the frequency of interest. For example, for a maximum frequency of 30 Hz, the sampling rate should be at least 60 Hz, although 250–500 Hz is commonly used for greater accuracy.

2.2   Fourier Transform (FFT) and Power Spectrum

Understanding Power Spectral Density (PSD) aids in assessing noise levels across frequencies. For instance, a 60 Hz power line artifact might appear as a spike in the power spectrum, indicating a need for filtering or artifact removal. Using the Fast Fourier Transform (FFT), we can examine the frequency content of the signal to verify the effectiveness of filters and detect any residual noise that may interfere with P300 detection.

3   References

- Filtering EEG Data. *Filtering EEG Data - Neural Data Science in Python.* (n.d.). Retrieved from `https://neuraldatascience.io/7-eeg/erp_filtering.html`

- Name, Y. A. (2021, January 13). *Data Visualization. Tutorial #2: Visualize EEG Data.* Retrieved from `https://neuro.inf.unibe.ch/AlgorithmsNeuroscience/Tutorial_files/DataVisualization.html`