Importing and Processing data with Python

1 Introduction

This guide outlines the essential Python libraries and steps for importing and processing EEG data, especially for applications involving ERP (Event-Related Potential) analysis. We will cover data import, filtering, and segmentation, using libraries such as MNE, SciPy, NumPy, and Matplotlib.

2 Required Libraries for EEG Data Analysis

The following libraries are commonly used in EEG data analysis:

- **MNE (Machines for Neural Experiments)**: A Python package specifically for EEG data analysis, offering functions for preprocessing, filtering, artifact removal, time-frequency analysis, and visualization.
- **SciPy**: Provides scientific routines and enables loading of MATLAB files for EEG data.
- **NumPy**: Supports multi-dimensional arrays, matrices, and mathematical functions, essential for data manipulation.
- **Matplotlib**: A visualization library used to plot EEG signals, spectrograms, and more.

3 Importing EEG Data

When loading EEG data from a `.mat` file, the contents are stored in a dictionary-like object with keys representing different data segments. EEG data is usually organized as a multi-dimensional array with the shape attribute denoting the size of each dimension:

$$\begin{aligned}\text{Number of channels} &= \texttt{eeg.data.shape[0]}\\ \text{Number of time samples} &= \texttt{eeg.data.shape[1]}\\ \text{Number of trials} &= \texttt{eeg.data.shape[2]}\end{aligned}$$

Using the MNE library, we create an `Info` object to store details like channel names, sampling frequency, and channel types. The sampling rate should be set according to the data, e.g., 512 Hz for BCI Competition III data.

4 Filtering EEG Data

EEG signals span a range of frequencies (typically 1-30 Hz for relevant activity), but also contain noise:

- **Low-frequency noise**: Caused by slow movements or wire shifts.
- **High-frequency noise**: Caused by muscle movements or electromagnetic interference.

To retain only the relevant band (1-30 Hz), we apply a band-pass filter:

- **Low-pass filter**: Removes high-frequency noise.
- **High-pass filter**: Removes slow drifts.

Tip: Always filter the data before segmenting into shorter epochs. Use MNE's `.filter()` method as follows:

```
raw_filt = raw.copy().filter(low_cut, high_cut)
```

where `low_cut` is typically set to 0.1 Hz and `high_cut` to 30 Hz.

5 Segmentation into ERP Epochs

After filtering, segment the EEG data into **epochs**, which are time-locked to specific events (e.g., stimuli or responses). The `events` array and an event dictionary (mapping codes to labels) are used to define these segments:

```
events, events_dict = mne.events_from_annotations(raw_filt)
```

The `events` array has three columns:

- **First column**: Index of each event in the data.
- **Second column**: Usually zero, ignored.
- **Third column**: Event code as an integer.

5.1 Event Dictionaries

MNE assigns unique integer values to each event code and stores this mapping in the event dictionary. This helps standardize event codes from different data sources.

6 References

- Neurotist. (2024, February 12). *Importing MATLAB Files into Python: A Step-by-Step Guide for EEG Data Analysis with MNE*. Medium. Retrieved from `https://medium.com/@neurotist/importing-matlab-files-into-python-a-step-by-step-gu`
- *Filtering EEG Data - Neural Data Science in Python*. (n.d.). Retrieved from `https://neuraldatascience.io/7-eeg/erp_filtering.html`