DSCI 591: Capstone Project – Proposal Report for Sensing in Biomechanical Processes Lab (SimPL)

Team members: Matthew Pin, Mo Garoub, Sasha Babicki, Zhanyi (Yiki) Su

Project mentor: Joel Ostblom

Date: May 13, 2021

3.1. Executive Summary

We are developing a Python package and dashboard for SimPL, a research lab that explores research questions concerning the human brain, to help them visualize EEG data and understand the functional state of the brain after sports-related head injuries. After learning about SimPL's problem with limited visualization methods, we are proposing the following deliverables:

1) A Python package for generating advanced EEG visualizations and metrics
2) An interactive web app to provide a user interface for the package

If time permits, we will expand our deliverables to include a stretch goal of building a data pipeline for unsupervised learning methods such as clustering.

3.2 Introduction

3.2.1 Background

SimPL is a research lab in the department of Mechanical Engineering at UBC. They are developing quantitative and sensitive methods to evaluate the electrophysiological changes after sport head injuries. In addition, SimPL is developing a mobile brain and body imaging system, head impact detection using machine learning, and investigating concussion mechanisms.

Concussion and brain injuries in general are invisible. The underlying mechanisms of brain dysfunction are not clear yet. SimpPL have employed electroencephalograms (EEG) to measure and detect potential changes in the brain electrophysiology due to sports head impacts. The team was approached to design novel solutions and methods to extract and visualize the human brain state using EEG data and to apply data science techniques learned in the MDS program. We have received a set of multichannel EEG data containing multi-participant and multi-trial experimental data.

The main benefit of EEG technology is that it is unobtrusive and inexpensive. EEG data has high dimensionality, so humans need visualizations and metrics in order to interpret it. Currently, the visualizations options for EEG data are limited. By extending the number of visualizations available and making them convenient to access, our tool will help scientists analyze their EEG data and provide an intuition of what the effects of their experiments may be on the brain. Additionally, our machine learning stretch goal could uncover patterns in the data which could not be determined based on visualization alone.

3.2.1 Main Goals

We will be designing a customized, well-documented Python package which will provide the ability to conveniently produce advanced visualizations and metrics for specified time ranges of EEG data. At a minimum, the package will include the following functionality:

1) Raw voltage values - produce raw voltage values to measure the EEG amplifier
2) Connectivity - calculate the correlation between nodes or groups of nodes for specified time ranges
3) 2D head map video - generate an animated 2D topographic heatmap of the voltage values recorded by each node of the EEG device. Includes the ability to take snapshots of power changes that represent the magnitude of the signal as a function of frequency

4) 3D skull map video - generate an animated topographic heatmap of voltage values mapped to a 3D model of skull

5) Interpolated 3D brain map - generate an animated topographic heatmap of voltage values mapped to a 3D model of the brain by interpolating voltage values to their presumed location in the brain.

We will also build an interactive web application to serve as a user interface (UI) for the package. The web application will be accessible by running a simple command and will provide widgets for customizing settings. At a minimum, the widgets will provide the following options:

1) File selection
    - Depending on the preferences of the partner, this can be linked directly to their local files or files on cloud
    - If requested by the partner, a file upload option is also feasible
2) Two time selection options
    - The user may input epoch timing data and then select epochs to display
    - The user may input a specific start time and duration of the animation
3) Frame rate
    - The frame rate for the animations

3.2.2 Stretch goal

If sufficient time is available after completing the main deliverables we will complete an additional deliverable. Our stretch goal is to create a data pipeline for pattern identifying clustering of the data, which is an unsupervised learning method. It might involve decomposing the signal into alpha, beta, theta, and delta waves to look for structure, and then use a Markov model or hidden Markov model to carry out the clustering task.

3.3. Data Science Techniques

3.3.1 Source Data

We will be using cleaned EEG data from 8 provided experiments as input for our main goals. Each experiment is expected to have:

- `fixica.set` (metadata)
- `fixica.fdt` (raw data)
- `impact locations.mat` (impact timestamps)
- `fixedareas.mat` (record of values that have been altered in cleaning process)
- 33 impacts per experiment
- 19 channels, one for each electrode
- Approximate duration of 1.5 hours
- Sampling rate of 2048 Hz (samples per second)

3.3.2 Techniques

The Python visualization package will mainly be developed using the open source library MNE. Custom visualizations may also be built with Matplotlib. Function development will be driven by the needs of the partner, improving ease-of-use compared to using MNE or Matplotlib directly. Clear documentation and code will be prioritized to allow package functionality to easily be updated following the completion of the Capstone. The main difficulties are expected to be

wrangling and transforming the data into the evoked format which MNE requires and long rendering times for visualizations due to the high dimensionality of data.

For the interactive user interface we are planning to use an open source framework called Streamlit which is designed for creating web apps from Python scripts. Streamlit benefits from being lightweight and requiring no front-end experience. This will facilitate ease of updating in the future. The main difficulty will be to design a straightforward but informative UI with a large number of visualizations.

For the machine learning classification/clustering stretch goal, SciPy can be used to perform data wrangling and decompose data into frequency-specific bandwidths. We may use a Markov or Hidden Markov model for the clustering tasks, as recommended by our Capstone partner. Other researchers have historically used k-means clustering, support vector machine (SVM) or CNN models in the classification process, which are viable alternatives. The pipeline will be built using scikit-learn or PyTorch and can be delivered in either a Jupyter notebook or Python script. The main difficulty is that domain expertise is required for interpretation, so identifying clusters will be difficult without significant assistance from the Capstone partner. Additionally, the data likely does not contain consistent results between experiments, complicating the testing of pattern identification tools.

3.4 Timeline

Milestone 1 - May 21, 2021 MVP for Python package visualizations and metrics (5 functions)

Milestone 2 - May 28, 2021 MVP for interactive user interface, first round improvements for package

Milestone 3 - June 7, 2021 MVP for stretch goal, first round improvements for UI

Milestone 4 - June 22, 2021 Report and touch-ups for previous milestones