Jonathan Ramos
COMP4334 Assignment2
11/17/2023

Environmental stimuli become paired with exposure to drugs of abuse and play an important role in the maintenance of drug memories. It is well known that the medial prefrontal cortex (mPFC) plays an important role in reward circuits in the brain, which include its connections to the dorsal hippocampus (dHPC). To better understand how a drug memory vs a neutral memory is represented in the brain, tungsten electrodes were chronically implanted into the mPFC and dHPC of 8 male Sprague Dawley rats and electrical activity in these regions was recorded while rats moved freely in a rodent behavioral chamber. On the first day, animals were placed their chambers and shown two cue lights of different colors in a random order every 5-15 seconds. Over the next 8 days, rats were then trained to pair one of the cue lights with either an IV saline infusion and the other cue light with an IV cocaine reward. The following day rats were again presented with both cue lights (without cocaine or saline infusion) in a random order every 5-15 seconds for 30 minutes and small changes in voltage in the brain were recorded. I collected these data with the Sorg lab at the Legacy Research Institute, data not yet published.

Since we know which cue was paired with a drug reward and which cue was control, we can slice out data around these timestamps and analyze the electrical activity for differences. The goal of this project was to build a binary classifier to distinguish whether or not a rat had been shown a cocaine associated cue or a neutral cue based only on electrical data recorded from its brain. Data was recorded at sampling rate of 30 kHz and then down sampled to 2 kHz and two second epochs of data were sliced out around each cue presentation (-1 second, +1 second). From each of these 2 second epochs, a total of 66 features were extracted. It is important to note that any signal (or otherwise continuous timeseries data) can be decomposed into its composite sinewaves via a mathematical technique called morlet wavelet convolution. From the complex coefficients of morlet wavelet convolution (implemented via the fast fourier transform and its inverse) we can extract instantaneous power and phase information. Power refers to the height of the sine wave (its magnitude) and phase refers to its angle in radians; because sine waves are cyclical, this is generally reported as oscillating between negative pi and pi.

Of these 66 features, 32 of them are measures of power at 8 different frequency bands (delta 2-4 Hz, theta 8-10 Hz, alpha 10-15 Hz, low beta 15-20 Hz, high beta 20-30 Hz, low gamma 30-60 Hz, mid gamma 60-80 Hz, high gamma 80-100 Hz) at two timepoints (early 0-350 ms, late 350-700 ms) each for the mPFC and dHPC, 10 of them are measures of phase synchronization (intersite phase coupling or ISPC)between the mPFC and dHPC at 6 frequency bands (delta 2-4 Hz, theta 8-10 Hz, alpha 10-15 Hz, low beta 15-20 Hz, high beta 20-30 Hz, low gamma 30-40 Hz) at two timepoints (early 0-350 ms, late 350-700 ms) and the last 24 features are measures of phase-amplitude synchrony between the mPFC and dHPC at 8 frequencies for gamma power (30 Hz, 40 Hz, 50 Hz, 60 Hz, 70 Hz 80 Hz, 90 Hz and 100 Hz) each compared with 3 frequencies for theta phase (6 Hz, 8 Hz, 10 Hz). Power-phase synchrony was computed by

$$MVL = \left| \frac{1}{n} \sum_{t=1}^{n} a_t e^{i\theta_t} \right|$$

as described in Canolty et al 2006, and intersite phase coupling was computed by

$$ISPC = \left| \frac{1}{n} \sum_{t=1}^{n} e^{i(\theta_{jt} - \theta_{kt})} \right|$$

as described in Cohen et al 2009.

Before implementing logistic regression as a stream in databricks, I first extracted and organized the raw data via a set of scripts I wrote called extract_data.py, aggregate_trials.py and curate_trials.py. These scripts epoched out 2 seconds worth of data around each cue presentation timestamp and automatically removed trials with artifacts (not all trials contained clean data). Next, I extracted all of the 66 features described above from each epoch via another script I wrote called window_trials.py. Because I was extracting so many features from this data, I ended up having to write out these data across 12 different csvs. Once all these features were extracted and written to csvs, I then loaded in these csvs into a databricks environment where I built a pipeline to aggregate these data into a single dataframe and pass stream them to a binary classifier.

The labels in this case were either “cocaine” or “saline” indexed to either 0 or 1; in total I had 118 cocaine epochs and 110 saline epochs. Because the remaining features are entirely numeric, the stages of my pipeline consisted only of the label indexer called cueIndexer, followed by the vector assembler called vecAssem, and lastly the logistic regressor called lr. I trained the model on a static window of “historic” data using a 70/30 train/test split, and then simulated a stream of test data. Before streaming, I first tested the model on a static window of test data and evaluated its results. The stream was then simulated by repartitioning the test data set in to 50 partitions, then writing 50 smaller csv files and using the ‘maxFilesPerTrigger’ option set to 1 while setting up the stream source. Predictions fit on the test stream were then sent to the sink, with output mode set to ‘append’ and format set to ‘memory’. Queries to display model evaluation can then be made in real time. After all files have been sent to the stream, results eventually reach the test results on static test data. The final model evaluation with no tuning ended having an area under the ROC of 0.8692 and 0.6054 for train and test respectively. I also built confusion matrices and computed accuracies, which were 0.8693 and 0.600 for train and test respectively.

I initially tried many different kinds of classifiers including gradient boosted tress, random forest, logistic regression and SVM with a linear kernel. I found that without tuning, the logistic regressor performed best and so I selected it as my final model for fitting and streaming predictions. One of the most difficult parts of this project was not the model fitting and streaming itself, but rather the amount of work it took to clean/epoch the data and extract all the features I wanted. In addition to setting an automatic rms threshold for artifact detection, I also visually curated each trial. Working with data and getting it into exactly the right shape was also tricky at times. In total, the raw datasets for just this subset of animals amounted to about 150 Gb of data which I am unable to upload in my submission (however, I was able to include the curated epochs from which features were extracted for classification since these are much smaller). Since I built the dataset myself, I tried to construct the csvs such that the streaming pipeline would be rather straightforward. Overall, without tuning, none of the models performed very well and some things I would consider going forward would be dimensionality reduction to simplify the models and increase the signal to noise ratio. PCA would also eliminate any multicollinearity in the sets which might help improve the performance of linear models.

References:

1. Cohen MX, Axmacher N, Lenartz D, Elger CE, Sturm V, Schlaepfer TE. Nuclei accumbens phase synchrony predicts decision-making reversals following negative feedback. J Neurosci. 2009 Jun 10;29(23):7591-8. doi: 10.1523/JNEUROSCI.5335-08.2009. PMID: 19515927; PMCID: PMC2725399.
2. Canolty, R. T., Edwards, E., Dalal, S. S., Soltani, M., Nagarajan, S. S., Kirsch, H. E., et al. (2006). High gamma power is phase-locked to theta oscillations in human neocortex. Science 313, 1626–1628. doi: 10.1126/science.1128115