P300 Speller:
Usefulness of ERP Data in Producing a BCI

Varshney Gentela and Claire Leahy
BME 6710 / CS 6990 / NSCI 6990: Brain-Computer Interfaces
Project 1 Report
University of Vermont
Burlington, VT

INTRODUCTION:

The P300 Speller is a Brain-Computer Interface (BCI) that leverages the P300 phenomenon – where an increased (***P***ositve) spike in voltage occurs approximately 300ms after a stimulus [1]– to create a spelling system. The P300 Speller enables individuals to communicate by implementing electroencephalography (EEG) data from the P300 component to novel but intentional stimuli, such as letters displayed on a screen [2]. The code is designed to analyze EEG signals and identify these P300 responses, which may ultimately allow users to spell words with EEG input rather than physical input. Namely, the data are analyzed to determine whether the P300 Speller is a viable system by observing potential significant differences between event-related potential (ERP) responses for target (i.e. a letter of interest was flashed) and nontarget (i.e. a letter was flashed but was not a desired letter) events. The isolation of significant differences between target and nontarget data at specific spatial and temporal locations will enable the BCI to identify when the brain processes an event it desires in comparison with one it does not and produce the corresponding results.

METHODS:

The aim of the new functions is to provide visual aids in understanding what points, both temporally and spatially, may be useful in performing the BCI's desired task. Specifically, the data can be described either for an individual subject or for the collective group. Both types of plots are organized by channel, where each channel displays mean EEG data (ERP) of an epoch, which is a set time window surrounding an event (for example, the start of the epoch may occur 0.5 seconds before the event takes place, and the end of the epoch may occur 1 second after the event). In this regard, the group data are more useful, as their corresponding plot displays the frequency of significance at each sample for each channel, emphasizing rough peaks of where data is indeed more likely to truly be a product of brain activity rather than noise. Aside from the temporal data visualization, another new function is implemented to plot spatial data in a scalp map depicting the topography of the EEGs, or the distribution of the electrical signals.

Before launching into the the code, it is important to note that when a parameter is set to a value in the function declaration, that value serves as the default if no parameter is entered during the function call. To make the training data usable, the function load_erp_data(subject=3, data_directory='P300Data/', epoch_start_time=-0.5, epoch_end_time=1.0) was declared. Implementing this function allowed for EEG data from a source directory to be loaded for a subject. Using some of the returns within the function, events – target events, specifically – could be identified, generating EEG epochs as well as corresponding relative time values based on the sampling frequency of the data and the start and end times given for the epoch where the other inputs for the overarching function are useful. Having epochs of EEG data and knowing which of those data are targets or nontargets allows for the generation of ERPs for both event types. The outputs of this function include is_target_event (necessary to index target EEG data from eeg_epochs, another output), erp_times (generally the independent variable), and target_erp and nontarget_erp (used in circumstances of real-data plotting).

In plot_confidence_intervals(eeg_epochs, erp_times, target_erp, nontarget_erp, is_target_event, subject=3), each output of the aforementioned function is an input, with the addition of subject for labeling. Since a 95% confidence interval using standard error is plotted using this function for both target and nontarget events, standard deviation must first be calculated using the raw EEG data, which requires indexing of eeg_epochs with is_target_event or its mask. Moreover, target_erp and nontarget_erp are plotted explicitly, as the real ERP data are depicted. While this function has no output values and only generates a figure, the base of this function is used in a future plot depicting statistical significance.

bootstrap_erps(eeg_epochs, is_target_event) further demonstrates the utility of the loading function's returns. This function provides random indices for sampling the EEG data, and, working under the null hypothesis that there is no difference between target and nontarget data, generates sampled "target" and "nontarget" ERP data by indexing the randomly sampled EEG using is_target_event indices and its mask. These data are returned as sampled_target_erp and sampled_nontarget_erp, respectively.

test_statistic(target_erp_data, nontarget_erp_data) is a flexible function that can be used on any two arrays of the same dimensions. This function calculates the absolute value of the difference between two arrays, in this case target and nontarget ERP data. Thus, this function can take in both the sampled data returned from the previous function and target_erp and nontarget_erp; calculating the test statistic for real and sampled values is necessary for calculating *p*-values. Thus, erp_difference – the test statistic itself – is returned.

calculate_p_values(sampled_target_erp, sampled_nontarget_erp, target_erp, nontarget_erp, randomization_count=3000) calculates the *p*-values of the dataset based on bootstrapped sample data, found by performing many iterations of bootstrap_erps(), and the real data. The test statistic is found for both sampled and real data, and a *p*-value is calculated for each sample point in each channel. An array containing these values (p_values) is returned.

plot_false_discovery_rate(eeg_epochs, erp_times, target_erp, nontarget_erp, is_target_event, p_values, subject=3, fdr_threshold=0.05) follows a procedure like plot_confidence_intervals() but requires p_values to perform a false discovery rate (FDR) correction. The corrected *p*-values attempt to eliminate false positives when multiple comparisons are performed, effectively making *p*-values larger and less likely to be significant. These new values are compared to fdr_threshold, conventionally known as alpha, where 0.05 is a typical significance level. Where the values are significant is plotted on a figure generated by this function, which is the previously generated plot but with black dots along the x-axis at those sample times. The sample times of significance are crucial for comparisons across multiple subjects, so significant_times, a list of these significant points in each channel, is returned.

multiple_subject_evaluation(subjects=np.arange(3,11), data_directory='P300Data/', sample_count=384, channel_count=8, epoch_start_time=-0.5, epoch_end_time=1.0, randomization_count=3000, fdr_threshold=0.05) calls load_erp_data(), which is why it should take its inputs, but now multiple subjects are listed in an array. This function also calls bootstrap_erps(), calculate_p_values(), and plot_false_discovery_rate(), prompting those inputs that are not otherwise returned to be taken into the outer function. sample_count and channel_count are required for sizing of an array before they could be dynamically coded, so taking them as inputs achieves more flexibility, as does including randomization_count for the number of bootstrap iterations performed, for example. significant_times is important: When plot_false_discovery_rate() is called, those times are used to create a count of how many subjects exhibit significance at any given point over any channel. This is used to create the return array subject_significance, which contains that count at each sample for each channel. erp_times is also returned to help with plotting.

plot_subject_significance(erp_times, subject_significance) is the culmination of the previous functions: It takes in the previous two returns and plots the number of subjects that are significant at a given sample in each channel and plots each sample along erp_times.

plot_spatial_map(subjects, data_directory) visualizes the spatial maps of the median target and nontarget Event Related Potentials (ERP) across the subjects. It takes a subject or a list of subjects and the path to the data as parameters. It loads the eeg_epochs using the load_erp_data function and extracts the median target and median nontarget ERPs in each subject and then calculates the median target and non target ERPs across all subjects in the N2 and P3b time ranges. The channel array is predefined which depicts the spatial representation of the electrodes. These topomaps help us in understanding the spatial distribution of neural activity during the N2 (200 – 300 milliseconds) and P3b (300 –500 milliseconds) time ranges across different brain regions that help us in analyzing the cognitive activity underlying the P300 responses.

RESULTS:

1) ERP Significance:

a. The confidence interval plots provide insight into the variability of the ERPs across various channels. By visualizing the variability around the ERPs for target and nontarget events, we could understand how distinct these responses are. This is essential for the accurate detection of the P300 responses in BCI applications as it informs of the consistency of the neural activity associated with stimuli (target and nontarget).

b. The depiction of FDR-corrected significance markers on ERP plots highlights the time points on EEG channels where differences between target and nontarget ERPs are statistically significant. This information is crucial for identifying the time windows and scalp regions most relevant for discriminating between target and nontarget stimuli. This will play a pivotal role in selecting the features and electrode placement while designing a BCI. Below is a figure depicting the ERPs with 95% confidence intervals and marked significance points. Figure 1 (left) shows how ERP data for the P300 Speller could be a useful BCI, as there are clusters of points that are commonly significant across channels, such as those between roughly 0.25 and 0.5 seconds following the event. However, Figure 1 (right) depicts the potential challenges of using these data for a BCI: There are virtually no significant points, making it very difficult to discern whether a target or nontarget event had occurred.

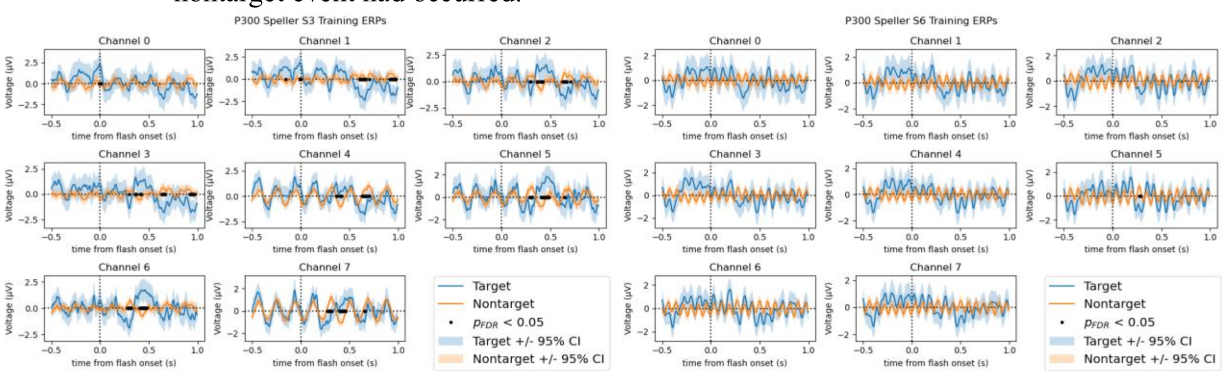

Figure 1: (left) Subject 3 ERP data with significance. (right) Subject 6 ERP data with significance.

c. While the above plots depict very different outcomes for using the P300 Speller as a viable BCI, the data generally suggest this BCI could be a sound choice. In evaluating Figure 2, which depicts the frequency of significance among subjects at sample points in each channel, there tend to be clusters between 0.25 and 0.75 seconds after an event occurs. Channels 0, 1, and 7 show the greatest consistency in clusters, with Channels 2 and 6 also exhibit decent concentrations of significance. P300 Spellers exhibit peaks in the 300-500ms range [1], or 0.3-0.5s, which is verified by the plots below, rendering the ERPs useful based on their consistent performance with the expected behavior. One important note is that there is little significance prior to the event onset, but, despite the concentration of significance within the expected 0.3-0.5s, the data can be significant all the way through to the 1s mark, suggesting a potentially flexible BCI as signal data across that range could be utilized.

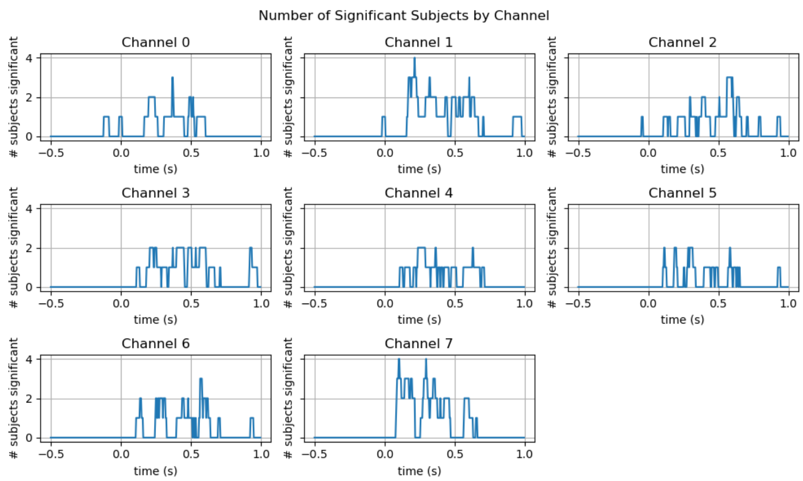

Figure 2: Significance at each channel and time.

2) Scalp Maps:
   a. The topographic spatial map of the brain is a good way to visualize the distribution of the neural activity of the brain in response to the target or nontarget events. This helps us figure out the spatial placing of the electrodes to efficiently detect the P300 responses. Channels that show significant variations in the voltage could be prioritized for electrode placement, which would help improve signal detection efficiency.
   b. CHOOSING ORDERING (SPATIAL LOCATION) OF CHANNELS: The chosen electrode order targets the parietal lobe, where the P300 response, linked to attention and decision-making, is prominent. By utilizing the spatial maps, that provide a visual representation of the of neural activity across different electrode positions, we could effectively discern the correspondence between the recorded EEG channels and the regions of the brain they originate from. This mapping has played an important role in identifying the spatial location of the electrodes.

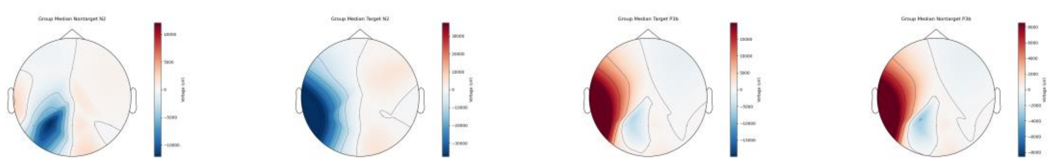

Figure 3: NonTargetN2, TargetN2, TargetP3b, NontargetP3b pertaining to subject 4

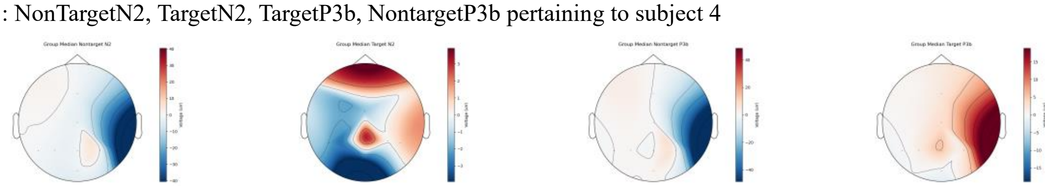

Figure 4: NonTargetN2, TargetN2, NonTargetP3b, TargetP3b pertaining to subject 5

DISCUSSION

1) Consistently useful channels and timepoints:
   a. Significant time points: The time points within the range of roughly 0.25s 0.75s after the event record notable neural activity across all channels. This window coincides with

neural responses to stimuli, facilitating the accurate recording of the P300 response which is pivotal in the functioning of this BCI.

b. Significant channels: channels 0(P4),1(Pz) and 7(P7) played a significant role in this BCI. They record meaningful neural activity that remains consistent through multiple subjects. This shows that the locations of these electrodes are relevant for the BCI task. The significance of these channels is consistent with the implication of the parietal lobe's involvement in the P300 Speller [3], specifically due to its functions like involvement in language and attention [4].

2) One size fits all vs. tailored BCIs:

c. One-size-fits-all:

i. There are many channels and spatial locations that are consistently useful for every subject for calculating the P300 responses. Making good use of these components could be an effective way to develop a cost effective BCI. But there will be variability in individuals' responses because of the different structures of the head. Also, the placement of the electrodes plays an important role in determining the efficiency with which features can be extracted. Referring to Figure 1, utilizing the same type of data for the same task yielded entirely different results, one of which would likely be a viable BCI while the other would not.

d. Tailored BCIs:

i. Tailoring a BCI to an individual could potentially improve the efficacy of the device. The signal to noise ratio can be improved which would in turn enhance the performance of the BCI. This application requires collection of individual data, analyzing the extracted data and a lot of time to customize the BCI. Although it consumes resources, this is a better approach as it would provide us with an efficient and desired product when compared to the one-size-fits-all model.

REFERENCES:

[1] D. Jangraw, "06_EventRelatedPotentials," Lecture presented in BME 6710: Brain-Computer Interfaces, University of Vermont, Burlington, VT, Feb. 1, 2024.
[2] C. Guger *et al.*, "How many people are able to control a P300-based brain–computer interface (BCI)?," *Neuroscience Letters*, vol. 462, no. 1, pp. 94–98, Sep. 2009, doi: 10.1016/j.neulet.2009.06.045.
[3] S. Gannouni, K. Belwafi, N. Alangari, H. AboAlsamh, and A. Belghith, "Classification Strategies for P300-Based BCI-Spellers Adopting the Row Column Paradigm," *Sensors*, vol. 22, no. 23, p. 9159, Nov. 2022, doi: 10.3390/s22239159.
[4] D. Jangraw, "04_TheBrain," Lecture presented in BME 6710: Brain-Computer Interfaces, University of Vermont, Burlington, VT, Jan. 24, 2024.