Project Description

Mini Project #1

Event-Related Potentials (ERPs): Decoding the P300

Projects should be done individually. You may consult with your colleagues on the projects but you must submit your own work.

Data Collection

In this project, we will look into neural decoding of the P300 response. In particular, we will investigate how decoding changes with the amount of data, across recording sessions, and across subjects. Continuing the experiment from last week, repeat the P300 speller data collection paradigm (Lab Instruction 5, Phases 1-3) with the following modifications:

1. Under the Phase 1 (training data) paradigm, enter a word > 12 characters in length.
    a. In the **To File** block, enter a new filename, *train2.mat*

2. Under the Phase 3 (testing data) paradigm, enter the phrase "BCI_LAB_ROCKS".
    a. In the **To File** block, enter a new filename, *test2.mat*

Upload your data to Courseworks using the following filenames:

a. GroupX_train1.mat
b. GroupX_test1.mat
c. GroupX_train2.mat
d. GroupX_test2.mat

Introduction (4 pts)

Write a short description of the experimental paradigm.

Explain what ERP is and how it is measured. Draw a picture and label the ERP components. Define and explain P300 and mismatch negativity (MMN). What are the differences between the two components?

Stimuli and Procedure (2 pts)

Describe your filtering method, sampling frequency, number of experimental trials, how you checked the signal quality, and what the *flash* and *target* signals represent.

Results

In this section, we will be training and testing linear classifiers to compare model performance in different scenarios. You can use any classifier you like, but *Mdl = fitclinear(X,Y)* and *Y_pred = predict(Mdl,X)* is suggested for simplicity. Furthermore, you are not required to optimize model hyperparameters (e.g., regularization strength, model type, etc.); however, points will be deducted if none of your models perform above chance.

Part 1. Within-Subject, Cross-Session Analysis

ERP Response to Target and Non-Target Flashes

First, using the *train1.mat* data collected by your group, calculate and plot the average ERP, averaged over electrodes, for target and non-target flashes from 100ms before to 600ms after each flash (same as HW). Mark the ERP components visible in the average ERP plots.

Plot the scalp map of the signal at 0, 100, 200, and 300ms after the onset of the flash. Use topoplot.m from EEGLAB and BCI.locs on Courseworks.

Produce these plots (average ERPs, marked components, topoplots) again, this time using the *train2.mat* data collected by your group. Do these plots differ from the *train1.mat* plots? If so, how do they differ and why do you think they differ in that way? If not, what makes them so consistent?

Training and Testing Within-Subject Decoding Models

Organize each of your datasets (*train1/2, test1/2*) into a data matrix, in which the rows are trials (individual flash events) and the columns are the time-domain features of the ERP from all channels. Each trial should capture approximately 100ms before to 600ms after the flash. How many features are there? Write an equation for the number of features based on the start time relative to the flash (t_{min}), the end time relative to the flash (t_{max}), the sampling rate (f_s), and the number of channels (N_{ch}).

Train three models: one using just the *train1* data ($Model_1$), one using just the *train2* data ($Model_2$), and one using both *train1* and *train2* data ($Model_{1+2}$). Test each of these models on just the *test1* data, just the *test2* data, and both the *test1* and *test2* data together. Fill in the table below with the accuracy of each model on each test dataset.

Accuracies	Test Set 1	Test Set 2	Test Set 1+2
$Model_1$			
$Model_2$			
$Model_{1+2}$			

Based on your knowledge of the P300 and the average ERPs you observed before, select a new time range captured by each trial. What range did you choose, and why? Describe how your results changed after changing the time range. Continue to use your time range for the rest of this section.

Which (model, test set) combination produced the highest accuracy? Why do you think that is? When we evaluate a model trained using data recorded from one session on test data that was recorded from a different session, we call this *generalizing across sessions*. How well do your models generalize across sessions? Which model generalizes better? Why do you think that is? When we build models using data from multiple sessions, we call them *cross-session models*. How does your cross-session model accuracy compare to your within-session model accuracies?

Analyze Within-Subject Decoding Models

Use *Mdl.Beta* to get the weights of your best model. Reshape the weights to a matrix of shape N_{ch} x T, where T is the number of time samples. Square the weights to get the power in the weights. This allows you to average over channels or over time without worrying about whether the weights are positive or negative. Average the squared weights over time, and create a topoplot of the average squared weights at each electrode. How does this compare to the topoplot of the average ERP 300ms after flash onset? Average the squared weights over electrodes, and plot the average squared weights over time. How does this compare with the corresponding time window of the average ERP plots?

Part 2. Cross-Subject Analysis

ERP Response to Target and Non-Target Flashes

Using the complete set of training data from all groups, calculate and plot the grand average ERP, averaged over electrodes, for target and non-target flashes from 100ms before to 600ms after each flash. Mark the ERP components visible in the average ERP plots.

Plot the scalp map of the signal at 0, 100, 200, and 300ms after the onset of the flash. As before, use topoplot.m and BCI.locs.

Compare these grand-average plots (average ERPs, topoplots) with your group's plots. Do these plots differ? If so, how do they differ and why do you think they differ in that way? If not, what makes them so consistent?

Training and Testing Cross-Subject Decoding Models

Train a new model using both sessions of training data from all groups, except your own. This is referred to as the *cross-subject model*. Select a time range that is captured by each trial in your data matrix. Did you use the same time range as before? Why or why not?

Test the cross-subject model on your group's test data (*test1* and *test2*) and on the test data from all other groups. How does the cross-subject model perform on a held-out subject (your group) compared to test data from subjects included in the training set? Also test your within-subject model on the test data from all other groups. How does the within-subject model perform on many held-out subjects compared to within-subject test data?

Accuracies	GroupX Test (Test Set 1+2)	All Other Groups Test
Cross-Subject Model		
Within-Subject Model ($Model_{1+2}$)	(Done in Part 1)	

Analyze Cross-Subject Decoding Models

Use the cross-subject model weights to create a topoplot of the average square weights at each electrode and a plot of the average squared weights over time. How do these plots of your cross-subject model compare to the equivalent plots of your within-subject model?

Discussion

Explain the advantages and disadvantages of using a within-session vs within-subject vs cross-subject model for a real-world BCI application.

Some tips for better results:

- Record *train2.mat* data using a longer word
- If you see high frequency noise, use filtering (e.g., a 10Hz lowpass filter) or other smoothing methods (e.g. use the MATLAB function conv2)
- Use a Z-score to normalize the data at each channel and/or each trial
- Look at your data trial-by-trial (in EEGLab or with plots) and try to remove bad trials by hand