CS 577 - Principles and Techniques of Data Science

Semester Project - Project Proposal

Detecting the P300 Event-Related Potential in EEG: A Data Science Approach

Authors: Noah Bakayou, Pavithra Magendiran, Noam Joseph

1. Introduction

The goal of our project is to create a machine learning classifier that can recognize the P300 Event-Related Potential using EEG signals recorded during a visual matrix-speller task. The P300 is a significant positive deflection that happens at approximately 300 milliseconds after a meaningful or unexpected stimulus, such as a flash in the target row or column in the speller interface. Because raw EEG data is both noisy and high-dimensional, detecting the P300 requires considerable preprocessing, segmentation, averaging, and dimensionality reduction. Traditional ERP averaging methods are sometimes too slow for real-time usage since they require repeated trials to reveal a clear signal. In contrast, classifiers may learn discriminative features directly from the data, which allows for faster and more adaptive brain–computer interfaces (BCIs). Our project applies supervised learning approaches to distinguish between target flashes that contain the P300, and non-target flashes that do not. The ultimate goal is to build a generalizable, optimized, and automated process for P300 detection that can be applied to other cognitive or attention-based ERP research.

2. Data Description

The dataset used in this project is BCI Competition III — Dataset II (Wadsworth, 2004), which includes EEG recordings from two subjects (A and B). Each subject's data contains 85 training epochs and 100 test epochs. The experimental arrangement is based on a 6x6 matrix, where rows and columns flash randomly one at a time. Each epoch corresponds to selecting a single letter, which results in 15 sequences and 180 flashes per epoch. Within each sequence, only two of the twelve row or column flashes per sequence are targets (one row and one column), and their junction reveals the intended character.

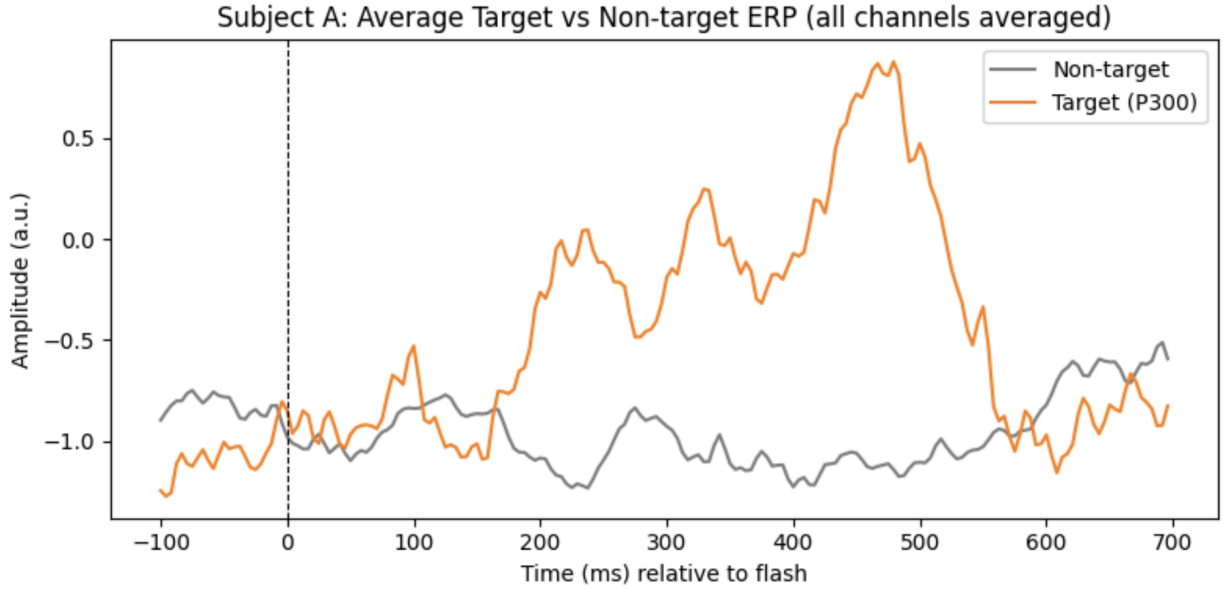
The raw EEG data are recorded in MATLAB.mat files, which contain many important arrays. “Signal” contains EEG recordings with dimensions (epochs, samples, 64 channels), “Flashing” shows when a flash occurs, “StimulusCode” identifies the flashed row or column (1-12), “StimulusType” distinguishes between target flashes (1) and non-target flashes (0), and “TargetChar” returns the true letter for each epoch in the training data. At the most basic level,

each EEG value represents a voltage sample over time. During segmentation, each flash becomes a single trial with around 192 time points distributed across 64 channels.

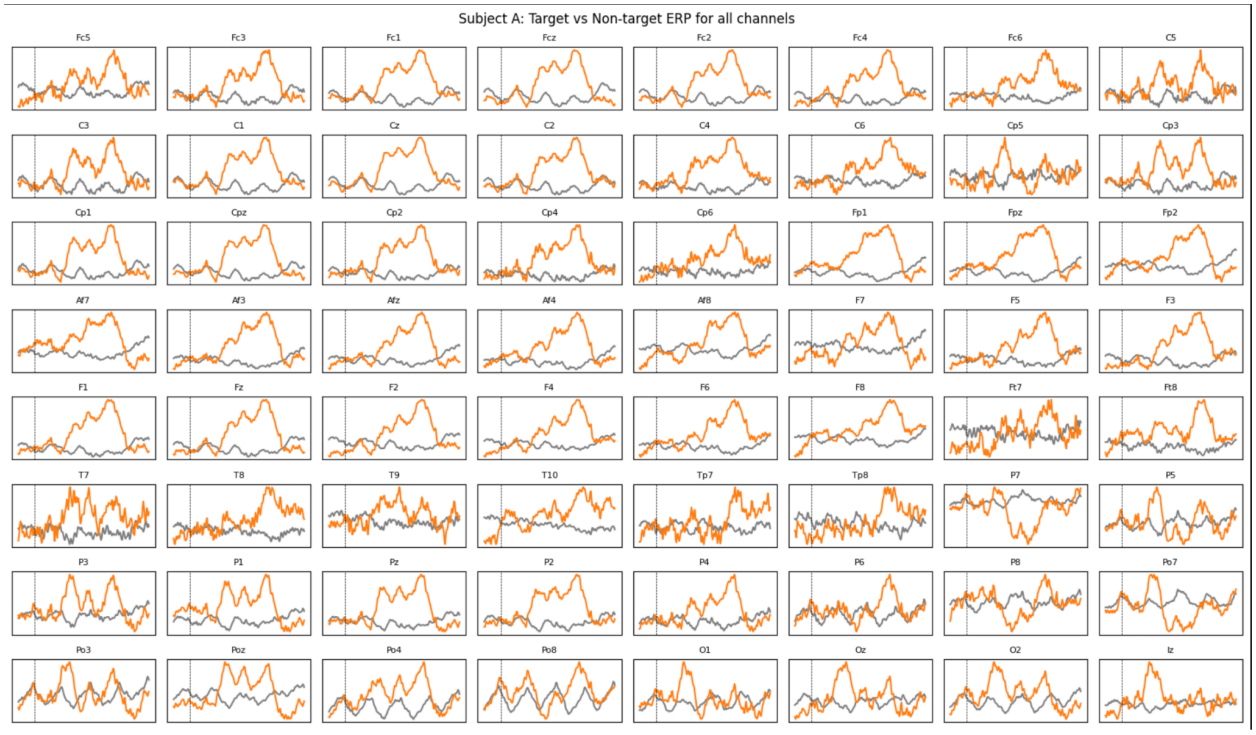
Data integrity checks confirmed the presence of all essential variables in all four .mat files, along with accurate array shapes and no missing (NaN) values. There were minor differences between the expected and actual flash counts, with 2537 target flashes versus an expected 2550, and 12,678 non-target flashes versus an expected 12,750. This minor variance is consistent with known anomalies in the dataset and is most likely due to the removal of artifacts like eye blinks.

3. Exploratory Data Analysis (EDA)

During the EDA phase, EEG segments were isolated and time-locked to each flash onset using a constant window ranging from -100 ms (pre-flash) to +700 ms (post-flash). We then separated into two key datasets: A_tgt_segs, which contained all target flash segments, and A_nontgt_segs, which had all non-target segments. The final array dimensions were (2537, 192, 64) for target trials and (12,678, 192, 64) for non-target trials, indicating that the segmentation technique worked as expected.



To validate the segmentation, average waveforms were plotted for both target and non-target scenarios. The resulting plots clearly revealed a positive deflection—consistent with the expected P300 response—emerging in the target flashes around 300 ms after stimulation. Non-target responses, on the other hand, were mainly flat or showed very minimal sensory components, with no identifiable P300 positivity. Additional channel-averaged plots confirmed these findings, demonstrating that, despite the inherent noise in EEG recordings, the P300 pattern was consistently present.



An assessment of class balance found an approximate 1:5 ratio of target to non-target samples, justifying the usage of class_weight="balanced" in subsequent classification models. Further examinations confirmed that the 192-sample window equated to approximately 800 ms of recording time at a sampling rate of 240 Hz. Channel-level analysis identified numerous electrodes—particularly Cz, FC1, and CPz—where the P300 signal was strongest. These findings were consistent with previous research literature.

```
Top 10 P300 channels (target more positive than non-target, 250–500 ms):
 1. Cz    score = 3.0422
 2. Fc1   score = 2.6241
 3. Cpz   score = 2.5097
 4. C1    score = 2.4422
 5. Fcz   score = 2.4368
 6. C2    score = 2.3898
 7. Fc2   score = 2.2885
 8. Fp1   score = 2.1778
 9. F1    score = 2.1731
10. Fz    score = 2.1384
```

In conclusion, the EDA validated the presence of a discernible P300 structure in the data while underlining the importance of dimensionality reduction techniques such as PCA in managing noise and improving classifier resilience.

4. Analysis Performed So Far

The first round of analysis entailed creating a rudimentary flash-level classifier, with each flash acting as a separate training instance. EEG data were confined to a 0-600 ms post-stimulus window, with a focus on three major channels—Cz, FC1, and CPz—that are known to have high P300 activity. Each segment was flattened into a feature vector, which was then merged into a feature matrix (X) with matching labels (y), with target flashes represented as 1 and non-target flashes as 0. To ensure class balance, the dataset was separated using stratified sampling.

The classification pipeline included normalization, PCA (10 components), and logistic regression with balanced class weights to account for the target-to-non-target ratio of approximately 1:5. This technique performed mediocre, with ROC AUC scores around 0.623, depending on the channels and time periods used (250-600 ms vs 0-600 ms). PCA effectively reduced noise and identified crucial low-dimensional structure in EEG data, whereas logistic regression produced interpretable, yet consistently mediocre findings for linearly separable features.

```
Running PCA + Logistic Regression using ONLY top 3 channels after stimulus (0–600 ms)
Using channel indices: [10, 2, 17] for ['Cz', 'Fc1', 'Cpz']
Window samples: 24 to 168 ( 0.0 ms to 600.0 ms )
Feature matrix shape: (15215, 435)
Class balance (targets, non-targets): 2537 , 12678
Classification report (flash-level, 3 channels, 0–600 ms):
              precision    recall  f1-score   support

           0      0.880     0.589     0.706      2536
           1      0.226     0.600     0.328       507

    accuracy                          0.591      3043
   macro avg      0.553     0.594     0.517      3043
weighted avg      0.771     0.591     0.643      3043

ROC AUC: 0.636
```

A second trial concentrated on the 200-600 ms window, when the P300 response is highest according to the literature, although performance fell marginally. This demonstrated that early post-stimulus data could still provide useful information for classification. Overall, the flash-level PCA and logistic regression pipeline established a solid foundation for future model enhancements.

```
Running PCA + Logistic Regression using ONLY top 3 channels after stimulus (0–600 ms)
Using channel indices: [10, 2, 17] for ['Cz', 'Fc1', 'Cpz']
Window samples: 24 to 168 ( 0.0 ms to 600.0 ms )
Feature matrix shape: (15215, 435)
Class balance (targets, non-targets): 2537 , 12678
Classification report (flash-level, 3 channels, 0–600 ms):
              precision    recall  f1-score   support

           0      0.880     0.589     0.706      2536
           1      0.226     0.600     0.328       507

    accuracy                          0.591      3043
   macro avg      0.553     0.594     0.517      3043
weighted avg      0.771     0.591     0.643      3043

ROC AUC: 0.636
```

5. Planned Next Analysis

The next analysis involves the main fundamental shift in how we approach the classification problem. Rather than treating each of the 180 flashes in an epoch as independent events, trying to move towards a code-level classification strategy. By averaging together approximately 15 flashes that correspond to each code to group it with its corresponding row or column in the matrix.

Working process: In each epoch, the system flashes 12 different codes. Among these, exactly 2 are targets: one row and one column. While the remaining 10 are non-target codes. The EEG segment average value for all flashes of a particular code (row/column) will be taken every 4 milliseconds; this processing acts as a powerful denoising filter because random electrical noise and artifacts tend to cancel out across multiple trials.

Across 85 training epochs, there are 1020 samples: 170 targets(2 per epoch) and 850 non-targets. To create average features, all flash code needs to be found to extract the EEG data from our three main channels (Cz, FC1 and CPz), averaging this together into a waveform, and later flattening it into a feature vector should yield better results for our PCA and logistic regression model.

The epoch-level-averaging method has clear benefits. The averaged signals should be easier to interpret. We plan to compare three aspects of our approach. First, we'll evaluate whether flash-level decoding or epoch-level-averaging decoding gives better results. Secondly, we'll start to test different ways of selecting which EEG channels to use in our model. Third, we'll experiment with a different number of PCA components to find the optimal balance between reducing noise and preserving important information.

6. Difficulties Encountered

We ran into several problems while working with this dataset. The first challenge was how the EEG data was organised in three dimensions: epochs, time, points and channels. This made the dataset hard to figure out how the different parts of the data signals, flashing, stimulusCode and stimulusType fit together. We noticed that the actual number of flashes didn't quite match what we expected, but this turned out to be normal for this dataset, probably because some data was removed due to EEG artifacts such as eye-blinks. The data also had too many dimensions, which made our models unstable until we used PCA to reduce them. The noise in the EEG signals made it hard to classify single flashes correctly. We fixed these issues by carefully exploring the data, making graphs to compare target and non-target signals, and reading the dataset documentation to understand how the timing worked.

7. Summary And Next Steps

We have successfully loaded the mat file (scipy loadmat), ensured compatibility with numpy and validated the full dataset. Also, we have extracted the flash-locked EEG segments, verified P300 presence, and produced clear ERP plots. Overall, we built a complete PCA plus logistic regression classifier for individual flashes (across different signal time-domains) and established the groundwork for more accurate channel-averaged-based classification.

Our next goal is to implement averaging across channels to dramatically improve signal quality, train PCA plus logistic regression on averaged ERP failures, and compare performance between flash-level and epoch-level methods. The aim is to continue improving our current model setup and reach a consistent, reliable threshold that warrants the experimentation and use of other model classes that may be better suited for this task.