NAZARBAYEV UNIVERSITY
CSCI 490, FALL 2020

Report on Final Project

Ruslan Bazhenov Marat Bekmyrza Bota Kabiyeva

Yassawe Kainolda Lira Yelemessova

December 10, 2020

1 Introduction

1.1 Objective

The objective of the project was to build a paradigm-independent classifier, which will classify trials, regardless which paradigm they came from. The result was to classify the trials from three different BCI paradigms: event-related potentials (ERPs), steady-state visual evoked potentials (SSVEP), and motor imagery (MI).

The classifier had to designed to be able to learn individual labels from eight given distinct classes, because each paradigm contains multiple classes: 4 classes for MI, 2 classes each for ERP and SSVEP.

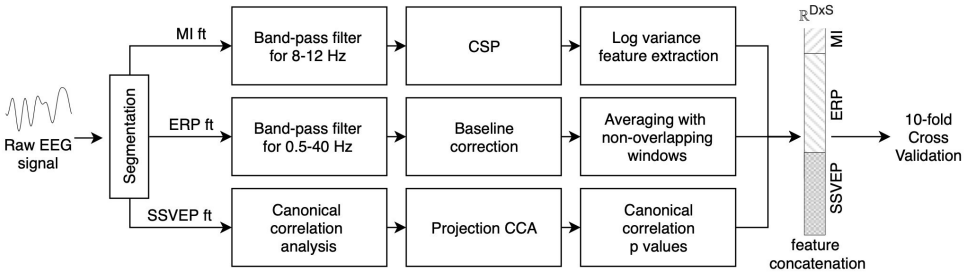

Figure 1: The methodology for one-fits-all classifier. Source: Adapted from [1]

1.2 Methodology

All code was written in Python. The steps were based on the methodology proposed by Li et al. [1] (See Figure 1).

About raw data, we have 600 trials (200 per paradigm) for one subject generates 106 features per trial: 96 from ERP, 6 from MI, 4 from SSVEP. We have variable 'x' which is the continuous data with dimensions time_points X channels. They were given to each feature extraction function. Moreover, from each subject 200 trials were random sampled, for MI and SSVEP train and test were combined. This was done on the variables 't' and 'y_dec'.

1.3 Pre-processing

The first step was to perform the necessary pre-processing of the data: down-sample and segment raw EEG signal. Next, the features were extracted. 600 trials (200 per paradigm) for one subject generates 266 features per trial: 256 from ERP, 6 from MI, 4 from SSVEP.

When it comes to pre-processing for SSVEP:

- First we combine both train and test, they have 100 trials each. Thus we got 200 trials per subject in SSVEP paradigm.
- In each trial we segment EEG data from start of the stimulus, i.e. 0ms, until 4000 ms
- Sampling frequency is $f_s = 100$ Hz
- Therefore we sample raw 4000 data points with 100 Hz, i.e. each 10 ms. Thus, we have 400 data points per trial.
- Preprocessed data for SSVEP has 200x400x10 dimensions. Since there 10 channels that were selected ()

The pre-processing for ERP:

- The function load_data opens every file that ends with "ERP.mat"
- Then, it extracts the data regarding the sessions, subjects; makes some basic manipulations with it
- At the end returns training data and testing data, feds the training data to the function "get_erp_features"

The MI pre-processing consists of data passing through the band-pass filter, then estimating CSP filters. an

1.4 Feature Extraction

In the case of ERP, the features are data-points for each channel, after P300 response:

- The function get_erp_features contains the functions "raw_eeg_segments", "filter", "baseline_correct", "average".
- The channels selected in the "raw_eeg_segments" are based one the [1] (See Figure 2). The function extracts segments of data from these channels and returns them.

- Then, we move to the function "filter". This function filters the given segments with 5-th order Butterworth bandpass filter between 0.5 - 40 Hz. Then, returns this filtered data for each channel.

- Next, the "baseline_correct" subtracts the average value of first 100 elements from the rest of 800 elements in each segment. The segment size is reduced from 900 to 800 for each channel.

- In the "average" function, the segments are downsampled from 800 to 8 samples in this function. So, we 100 data-points in each sample. Each block of 100 samples in the segment is averaged. From 800 data points, we get 8 averages for each channel. To further downsample the data, first and last data points are omitted, so there is 6 data points per each trial.

- Now, we are back at get erp features. All of the above operations are combined in this function. After "average", "x" in the shape of n number of trials x 6 data-points x 16 channels. Now, the rst and the last data-points are thrown out, then "x" is reshaped from 3dimensional array into 2 dimensional array meaning, the function returns two numpy arrays: "x" with the shape (n, 96) and "y" with the shape (n, 1).

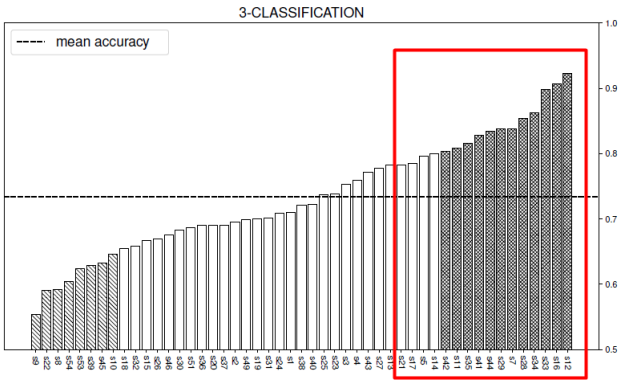

Figure 2: The selected channels in "raw_eeg_segments". Source: Adapted from [1]

In MI we use common spatial pattern (CSP) algorithm. It is widely used to extract features on EEG data. The main idea is to apply linear transformation to the data in order to map it into feature space, which is more discriminative in terms of variances.

The canonical correlation analysis was used to detect SSVEPs:

- then reference signals were genereate for each 4 frequencies

- Afterwards each canonical correlation coefficient was found between each trial and reference signal. Then the maximum correlation was selected within one frequency. Thus we get 4 values for each trial. As these steps are paradigm-dependent, for each trial they were performed simultaneously for each trial using different techniques.

1.5 Classifier

The chance level is 12.5%, as there are 8 classes.

subject	rLDA
1	0.635
2	0.777
3	0.745
4	0.695
5	0.768
6	0.735
7	0.760
8	0.700
9	0.683
10	0.755

2 Graphs, Visual Representation of Data

2.1 ERP

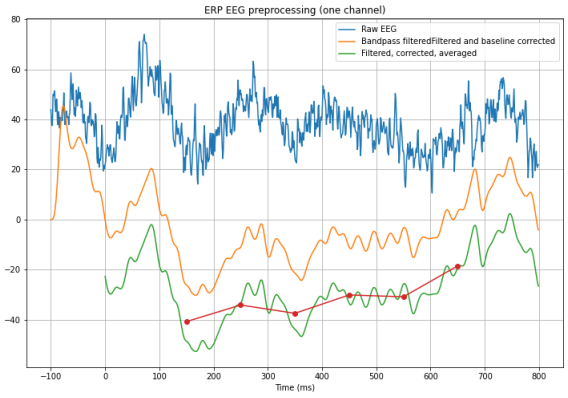

Figure 3: ERP EEG processing (one channel)

Figure 4: Raw ERP EEG

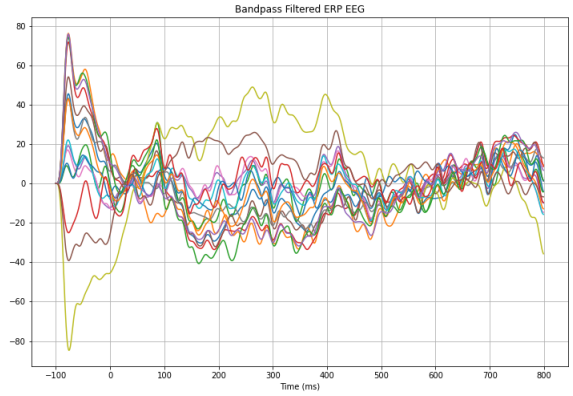

Figure 5: Bandpass filtered ERP EEG

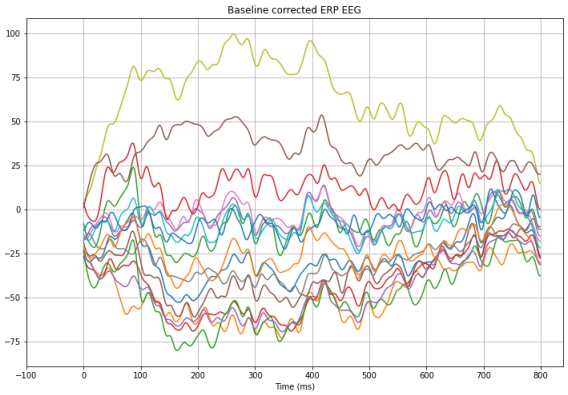

Figure 6: Baseline corrected ERP EEG

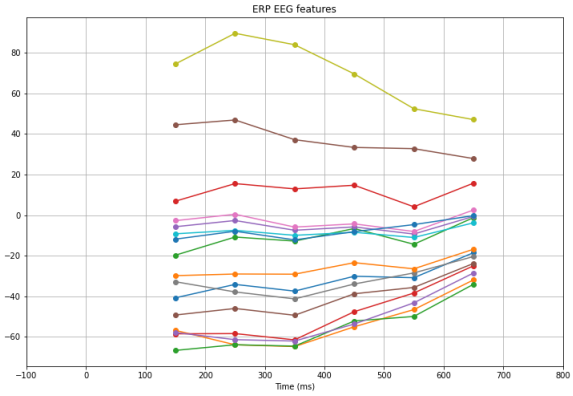

Figure 7: ERP EEG features

References

[1] A. Li, K. Alimanov, S. Fazli, and M. Lee. Towards paradigm-independent brain computer interfaces. In *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–6, 2020. doi: 10.1109/BCI48061.2020.9061657.