A Brain-controlled On-screen Assistive Keyboard

Report 2

Maqarios Mohareb

April 2019

Contents

Chapter 1

Introduction

Motivation

Imagine that one day, while you are setting on the bed too lazy to move. with this project all you need to do, is just one glance at the computer screen near you to send an email to your employer telling him that you are not coming today. Or having a disabled friend but wants to learn programming. all of these are possible using brain waves, In a parallel world brain waves is a synonyms for laziness.

Objective

The aim of this project is to build on-screen keyboard however without using clicks but brain waves to choose a character within a matrix.

Brief Description

A matrix of characters (i.e. 6x6) each cell has a character. The matrix is displayed for a certain period then each row and column will be intensified for a certain period of time randomly. The row and column which has a certain wave form is their intersection is the chosen character.

Chapter 2

Background

2.1 Introductory to BCIs

BCI Stands for (Brain-Computer Interface) which refers to intercepting the brain waves through some means such as electrodes, which is then transmitted to a computer after amplifying and digitizing the signal (Figure 2.1). The process starts with the intent of the user to do some action such as raising the right hand, the BCI records the brain activity and sends the intercepted signals to the BCI applications to do the desired reaction. There are several terms such as BMI (Brain-Machine Interface) or DBI (Direct Brain Interface) have the same meaning as BCI, however there are some terms such as Neuroprosthesis are not the same because BCI refers to only receiving (intercepting) data from the brain, however Neuroprosthesis refers to both sending and receiving data to and from the brain. In other words, BCI is a subset of Neuroprosthesis.

2.1.1 Processes of Operation

BCI goes through three main procedures, measuring the brain activity, processing it and controlling the intercepted signals.

Measuring Brain Activity

Brain activity produces magnetic and electrical waves. That's why, sensors or electrodes are used to measure these changes at different areas and times. Brain activities can be recorded through sensors (non-surgical solution) or electrodes (surgical solutions). For the non-surgical solution, sensors are used to measure the electrical activity from the scalp and this solution is called Electroencephalography (EEG) (Figure 2.1). They are relatively easy

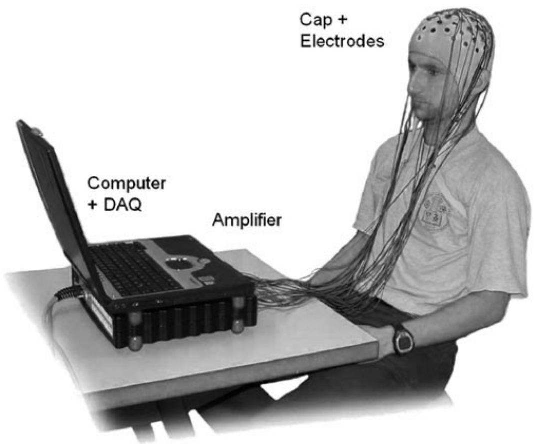

Figure 2.1: Person records BCI data

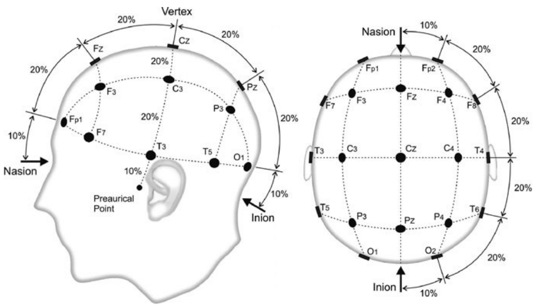

Figure 2.2: 10 - 20 System that shows how sensors are spread on the scalp

to deal with however they don't provide accurate measures due to external interference and are susceptible to limitations in frequency range. In order to get consistent recording, sensors are spread on the scalp through system called 10 - 20 (Figure 2.2). 10 - 20 refers to how sensors are spread 10 - 20 - 20 - 20 - 20 - 10 percent. The 6 regions are named according to their position: Fp - pre-frontal, F - frontal, C - central, P - parietal, O - occipital, T - temporal. On the other hand, the surgical solution need to open the skull through surgical procedures and plant the electrodes. When the electrodes are planted on the surface of the cortex, it's called electrocorticogram (ECoG). And another solution called Intracortical, which plants the electrodes in the inner parts of the brain. Although the surgical solutions provide higher accuracy and frequency range compared to the non-surgical solution they have serious drawbacks such as they need surgery, finance and can have ethical problems. Difference between each solution is shown in Figure 2.3.

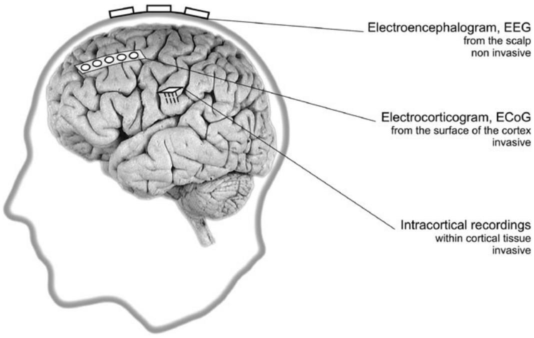

Figure 2.3: Types of Reading Brain Waves on Computers (BCIs)

Process and Control

Processing and Controlling Received Signals are handled within the application.

2.1.2 Brain Patterns

Because measuring brain activity is not enough, some mental strategies are used to trigger the required tasks. The most used mental strategies are selective attention and motor imagery.

Selective Attention

Mental strategies based on selective attention require external stimuli such as auditory stimuli or visual stimuli. P300 (Peak signal at 300ms) or SSVEP (Steady-State Visual Evoked Potentials) are two mental strategies the depends on visual stimulation. P300 is triggered when the intensity of symbols is changed and SSVEP is triggered with flickering some areas with certain frequencies (6 – 30Hz).

2.2 Online Dataset

BCI2000, Competition III, Dataset II.

2.2.1 Paradigm

The user was shown a 6 by 6 matrix containing English letters and numbers (Figure 2.4). The user was to spell words and focus his attention of the character that he wants to choose. All rows and columns were intensified

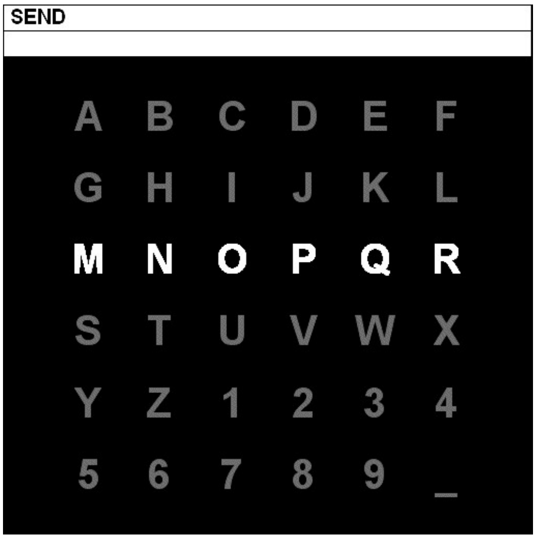

Figure 2.4: The matrix displayed to the user

randomly and successively at a rate of 5.7Hz (100ms intensification, 75ms no intensification). In other words, a column or row is intensified (i.e. row 2 in Figure 2.4) for 100ms then, no row or column is intensified for 75ms, then repeat. The row and column (2 intensifications) which contained the chosen character will provide different signals (P300) compared to the other 10 intensifications which does not contain the chosen character (Non-P300).

2.2.2 Collection of Dataset

Signals are filtered from 0.1 – 60Hz and digitized at 240Hz (Each 240 samples corresponds to 1 second). The 12 sets of intensifications were repeated 15 times for each character resulting in 180 total intensifications per epoch (character) into 64 channels EEG (Figure 2.5).

2.2.3 Explanation of The Dataset

The dataset has 2 subjects (A and B), where each subject has 85 characters (epochs) in training dataset and 100 characters in testing dataset (Remember that, each epoch has 12 intensifications and 15 repetitions). The structure of variables for each subject file is described below.

Signal: signal of each channel (Figure 2.5)

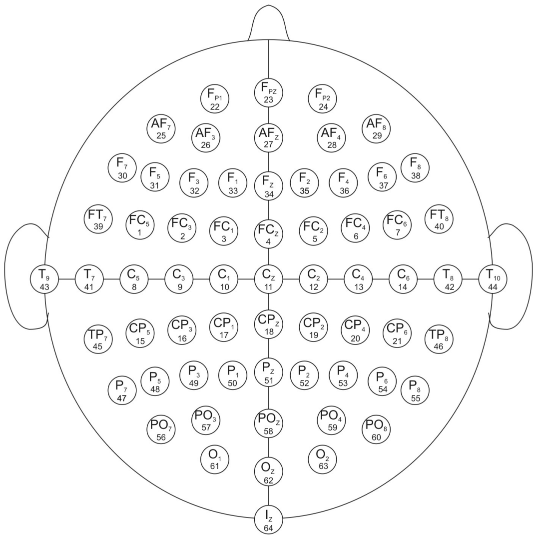

Figure 2.5: Mapping of each sensors to its place on the scalp

	Type: 3D Array Shape (Dimension): (85, 7794, 64)
TargetChar:	The chosen character for each epoch Type: String Length: 85
Flashing:	0 when NO row/column is intensified 1 when row/column is intensified Type: 2D Array Shape (Dimension): (85, 7794)
StimulusCode:	0 when NO row/column is intensified 1 ... 6 when intensified column (1 is left-most column) 7 ... 12 when intensified row (1 is upper-most row) Type: 2D Array Shape (Dimension): (85, 7794)
StimulusType:	0 when NO row/column is intensified 1 when row/column contains the chosen character Type: 2D Array Shape (Dimension): (85, 7794)

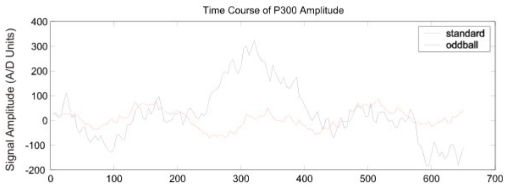

Figure 2.6: Caption

Note: 7794 samples because the data is digitized and there is a no-intensification period between each intensification. Note: TargetChar, StimulusType are only provided for training dataset.

2.2.4 Analysis

Figure 2.6 shows signal wave forms for P300.

Chapter 3

Approach and Results

3.1 P-300 Prediction using Machine Learning

Python programming language was used.
The dataset II provided by competition III was in matlab files and ASCII. Due to being easier and more readable matlab files (*.mat) was used. There were 4 matlab files, 2 for each subject (A, B), 1 for training and the other for testing.

3.1.1 Prepossessing

Since the data is raw, certain preparations and calculations needs to be done in order to make it readable by the model. There exists 5 prepossessing calculations; 1 averaging of the repetitions and 4 filters to decrease and enhance the number of features.

Averaging

Each epoch consists of 12 intensifications and each intensification is repeated 15 times thus in order to decrease the noise, we take the average of each intensification for the 15 repetitions in 1 epoch.
For easier future use, only Signal, StimulusCode and TargetChar arrays were used. For each epoch, a loop through StimulusCode array to search for a transition from non-zero to zero which means there was a transition from intensification to no intensification. Then, Taking the values of all channels from this index + start window - 24 to index + end window - 24. The index + start window - 24 because it takes the values of Signal from the start of the intensification with certain boost since we might not take the values from 0ms, as for index + end window - 24 to take the values of Signal for a

$$y_i(j) = s_i(j) - \frac{1}{N}\sum_{l=1}^{N} s_l(j)$$

Figure 3.1: Common Average Reference Equation

$$x_{ij} = (y_i(j) - \mu_i)/\sigma_i$$

Figure 3.2: Z-Score Equation

whole second after the start of intensification given that end window = 240. Remember that 240 is the same as the digitization of the intercepted signal (240 samples = 1 second).

Common Average Reference Filter

Per epoch, Per intensification, we calculate the average of all 64 channels for a certain sample. then subtract each sample from the calculated average.

where s i(j) denotes the signal recorded on channel i at time j, y i(j) is the filtered signal after subtracting the mean of all channels and N is the number of channels [3].

Moving Average Filter

Per epoch, Per intensification, Per channel, the average of n consecutive samples is calculated and set the new signal values to the calculated average.

Z-Score

Per epoch, Per intensification, Per channel, Z-score is calculated.

where x ij denotes feature j of channel i (i.e. x i (j)), u i is the average of the signals recorded on channel i and o i is the corresponding standard deviation.

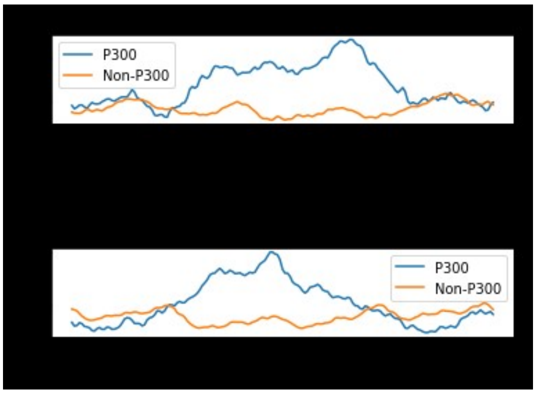

Figure 3.3: P300 for both subject A and B

Decimation of Samples

Finally, instead of taking all samples per channels, we take 1 per n consecutive samples, thus we decrease the number of features and keep the distinction of each feature as maximum as possible.

Channel Concatenation

3.1.2 Model

Only Least Discriminant Analysis (LDA) model was applied since it has the one of the highest scores for this dataset.
The data segment consists of 800ms (start = 0, end = 192). and two sets of channels were used, the first one has the whole 64 channels, and the second has 8 channels (Fz, Cz, Pz, P3, P4, PO7, PO8 and Oz) as described in [2] in references.

Plot of Wave Forms

In order to signify the difference between the P300 wave form and non-P300 wave form, Figure 3.3 shows the P300 vs non-P300 for both subject A and B after taking the average of the chosen intensification for the entire 85 epochs.

Least Discriminant Analysis

Since the problem is a binary classification one, the line that signifies whether the signal is P300 or not can be calculated using the function in Figure 3.4. where x is the feature vector, w is the weight vector and b is the bias term [3].

$$w^T x + b = 0$$

Figure 3.4: Binary Classification Line Equation

$$w = \left(X^T X\right)^{-1} X^T y$$

Figure 3.5: Weight Calculation Using LDA

LDA is the easiest way to calculate this w using the equation in Figure 3.5.
where x is the feature vector, y is class of each signal (1 for P300, -1 for non-P300).

In order to calculate which class the signal belongs to, the equation in Figure 3.6 is used.
where x is the feature vector, w is the weight vector.

Since it is impossible to get 1 and -1 from the score function we take the max score within column and max score within rows to signify the chosen intersected character.

Results

Channels	CAR	Moving Average	Z-Score	Decimation	Start Sample	End Sample	Subject A Precision	Subject B Precision
8	Yes	Y, 25	Yes	Yes, 10	0	192	65	91
64	Yes	Y, 25	Yes	Yes, 10	0	192	81	80
8	Yes	Y, 25	Yes	Yes, 10	48	192	61	81
64	Yes	Y, 25	Yes	Yes, 10	48	192	79	84
8	Yes	Y, 5	Yes	Yes, 4	48	192	76	83
64	Yes	Y, 5	Yes	Yes, 4	48	192	82	81
8	Yes	Y, 5	Yes	Yes, 4	0	192	81	91
64	Yes	Y, 5	Yes	Yes, 4	0	192	87	84

$$score = w^T x$$

Figure 3.6: Score Calculation Using LDA

Chapter 4

Updated Timeline

Milestone #	Milestone's Description	Unit #	Unit's Description	Deadline
0	Revision on ML	1	Revision on Supervised ML	21 / 2
		2	Revision on Classification of Supervised ML	25 / 2
1	ML Model Programming	1	Dataset Load	28 / 2
		2	P300 Plot	4 / 3
-	**Report 1**	-	-	**10 / 3**
1	ML Model Programming	3	Application of LDA	18 / 3
		4	Application of Filters	1 / 4
		5	Enhancement	8 / 4
-	**Report 2**	-	-	**14 / 4**
2	GUI Programming	1	Matrix Programming	18 / 4
		2	Integration	22 / 4
		3	Training & Testing	25 / 4
3	NLP Programming & Integration	1	NLP Programming	29 / 4
		2	NLP Integration	2 / 5
		3	-	6 / 5
-	**Thesis**	-	-	**26 / 5**
-	**Presentation**	-	-	**8 / 6**
-	**Amended Thesis**	-	-	**4 / 7**

References

[1] Bernhard Graimann, Brendan Allison, and Gert Pfurtscheller. Brain–Computer Interfaces: A Gentle Introduction.

[2] Dean J Krusienski, Eric W Sellers, Francois Cabestaing, Sabri Bayoudh, Dennis J McFarland, Theresa M Vaughan and Jonathan R Wolpaw. A comparison of classification techniques for the P300 Speller.

[3] Amr S. Elsawy, Seif Eldawlatly, Mohamed Taher and Gamal M. Aly. A Principal Component Analysis Ensemble Classifier for P300 Speller Applications.

[4] Amr S. Elsawy, Seif Eldawlatly, Mohamed Taher, Gamal M. Aly. MindEdit: A P300-based text editor for mobile devices.