Evolution of Brain-Computer Interfaces: Temporal Stimuli, Guessing Functions and Online Games

Summary

Overview

Brain–Computer Interface (BCI) based on Electroencephalogram (EEG) is an extremely important tool for individuals with severe motor disabilities (e.g., amyotrophic lateral sclerosis, stroke, and spinal cord injury). However, due to the **low information transfer rate (ITR)** and **decoding accuracy**, BCI has not yet found many widespread production-ready applications. To address these two known issues, we propose to leverage proven techniques in many disciplines of signal processing, machine learning, information, and coding theory. In this project, we shall consider conventional flickering stimuli as well as extensions i.e., videos for temporal stimuli generation. We also cast the multi-class classification problem which is the natural part of any BCI system, as a two-player "guessing game" where each game would correspond to binary classification. All subcomponents of BCI systems are imperfect and can make decision errors which eventually affect the final accuracy and ITR. For instance, one of the objectives is to find the classifier's strategy that would both minimize the number of binary classifiers as well as the total accumulated cost of classification error. The more severe the classification error is, the more we would need to introduce binary classifiers to meet the desired accuracy. We shall propose changes in the stimuli generation to cleanse the collected waveforms for accurate processing. A Fundamental trade-off between classification error (accuracy) versus ITR shall be established (capacity of visual cortex channel). **The main motivation of this study is two-fold: (1) investigation of future stimuli paradigms for futuristic BCI design with improved ITR/accuracy performance, (2) the interpretable machine learning where we adapt "guessing" framework from game/information theory to analyze EEG signals and address the multi-class classification problem.**

Short Project Description

Introduction-Background

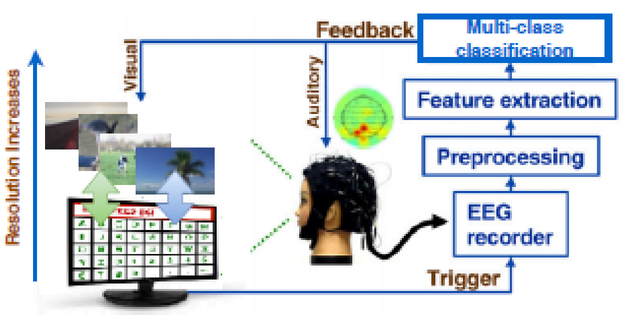

Figure 1: The proposed BCI system consists of four main stages: visual stimulation at various resolutions, EEG recording, real-time data processing, and feedback presentation.

In the past two decades, we have witnessed unprecedented progress in brain–computer interfaces (BCIs). However, due to low accuracy and communication rates it has never become commonplace. With the state-of-the-art technology, electroencephalogram (EEG) is the most common implementation of BCIs due to its non-invasiveness, simple operation, and relatively reduced cost. Steady State Visually-Evoked Potential (SSVEP) is one of the EEG-based paradigms on which BCI systems are typically designed. SSVEP signals are the brain responses to repetitive flickering visual stimulus where each character/command is encoded with unique frequency before display. SSVEP-based BCI paradigm has drawn attention because of its several advantages, such as high information transfer ratio (ITR), ease of system configuration, and little time for training [1].

General System Diagram: There are four main stages in a given SSVEP-based design. First and foremost is the stimuli design such as the paradigm, number of targets, parameters (optimization of stimulation duration, frequency and phase intervals) etc. Although repetitive flickering visual stimulus with auditory alerts are common and forms the heavy artillery of attention manipulation, other forms of stimuli such as carefully designed videos can be preferred with more resolution. First, flashing lights may induce prolonged distortions in perception [2] and also actions can be encoded into the video and help achieve better transfer rates. Later stages include preprocessing which includes appropriate sampling and filtering and noise enhancement. Depending on the latency in the visual system and event triggers inside the video, parameters of the signal processing modules would need to be adjusted.

One of the well known target identification methods is the canonical correlation analysis (CCA) [3], a statistical way to measure the underlying correlation between two multidimensional variables $\mathbf{X}$ and $\mathbf{Y}$. More specifically, CCA finds weights $\mathbf{w}_X$ and $\mathbf{w}_Y$ such that the correlation coefficient (ρ) is maximized i.e.,

$$\rho = \max_{\mathbf{w}_X, \mathbf{w}_Y} \frac{\mathbb{E}[\mathbf{w}_X^T \mathbf{X} \mathbf{Y}^T \mathbf{w}_Y]}{\sqrt{\mathbb{E}[\mathbf{w}_X^T \mathbf{X} \mathbf{X}^T \mathbf{w}_X] \mathbb{E}[\mathbf{w}_Y^T \mathbf{Y} \mathbf{Y}^T \mathbf{w}_Y]}} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{C \times T}$ represents the multi-channel SSVEP signals for C-channel and T time samples and $\mathbf{Y}$ template reference signals. In CCA, the frequency detection is accomplished by setting $\mathbf{Y}$ to sinusoidal signals at a specific stimulation frequency and its different harmonics. The frequency that gives the maximum correlation (classification) is selected as the final frequency value of the SSVEP. On the other hand, correlated component analysis (CORRCA) [4] also calculates correlation between two multidimensional variables but this time $\mathbf{Y}$ is set to averaged SSVEP signals across multiple blocks for each individual frequency. In various extensions of this method, filter banks are heavily used to improve identification performance of the system [4]. For instance in HFCORRCA method [5] filter bank outputs are spatially merged in an hierarchical manner before frequency fusion. Final output is declared using fusion outputs through frequency recognition.

Feature extraction and Multi-class classification cast as a Guessing Game: Feature extraction is typically based on one of the variants of CCA and correlation results are combined to determine the final BCI command. There are some recent work that combines feature extraction and classification in a deep neural network [6], [7]. Although the advantages are clear and performances are shown to be better than some of the past works based on CCA, it is almost impossible to explain why they work as their network is pretty complex and intelligibility is missing. This makes it hard to generate right and meaningful feedback for the stimuli adaptation. Here in this project we propose to combine feature extraction and multi-class classification using a guessing framework and project to obtain **similar or better improvements in the accuracy and ITR** performances.

Final stage is the multi-class classification of the features that we cast it as a guessing game. The typical guessing game involves finding the value of a realization of a random variable X from a finite (or countably infinite) set $\mathcal{X} = \{x_1, \ldots, x_{|\mathcal{X}|}\}$ by asking a series of binary questions "Is $X \in \mathcal{S}_i \subseteq \mathcal{X}$?" for an ordered subsets S_is of $\mathcal{X}$ until the guesser can accurately identify X based on the binary responses (Yes or No) [8]. It is typical to assume that some of the answers are wrong either by nature or on purpose. The latter case is known as "guessing with lies" in the literature [9]. What makes this simple guessing framework ***online*** is that each binary answer generated for S_{i-1} has an effect on the following question i.e., on the formulation of S_i and the associated binary answer unlike ***offline*** guessing frameworks in which the number of questions are typically determined a priori and all of these questions are required to be responded before making a decision on the final guess [10]. In other words , offline games require the construction of all questions ahead of time and no question can be adapted based upon previous responses/results.

In the SSVEP context, the answers are generated by a set of non-perfect binary classifiers and the questions are determined by unique selection of subsets S_is on the fly based on the answers of previous binary classifiers. The classification errors can be treated as lies in a guessing context and the number of S_is are determined such that these errors can be compensated to accurately identify the target BCI command. Note that however, unlike the typical game, answers are not necessarily simple yes or no (binary), rather real (such as correlation coefficient) or probabilistic. Given the error characterization of the set of binary classifiers, typical objective is to minimize the number of S_is to $n \geq \lceil \log |\mathcal{X}| \rceil$ such that using non-perfect binary classifiers, the probability of the multi-class of interest evolves to 1 as we hit the n-th classification result. This can be interpreted as "a noisy channel (brain pathway) disturbing the class labels" and we are targeting to find the achievable rate of transfer through this channel. The challenge in this estimation is the absence of the complete statistical characterization of the underlying noisy channel.

In this work, we consider a few variations on the online version of the problem. (1) We allow non-binary answers i.e., in our context the classifier i responds that $X \in S_i$ with probability p_i and $X \notin S_i$ with probability $1 - p_i$. This is both due to imperfect classifier design as well as the non-uniform input distribution $P_X(x)$. Note that the probability of having an error in the response is either p_i or $1-p_i$ depending on whether $X \in S_i$ is actually true. (2) Based on the objective of the classification such as minimization of false positives or minimization of false negatives, a cost and a complementary cost (comcost) metrics can be associated with each question. (3) Given $P_X(x)$, all S_is and a guessing strategy can be optimized to minimize the total cost of the multi-class classification (MCC) where the comcost is subject to an upper bound. Finally, a probabilistic belief propagation network can be used to track the density evolution to e_ciently decide/decode finally on the multi-class of interest.