Performance Evaluation of Manifold Algorithms on a P300 Paradigm based Online BCI Dataset

Bipra Chatterjee[1], Ramaswamy Palaniappan[2,*], and Cota Navin Gupta[1,*]

[1] Indian Institute of Technology, Guwahati, India
Neural Engineering Lab, Dept. of Biosciences and Bioengineering
cngupta@iitg.ac.in,
[2] University of Kent, Chatham, UK
Data Science Research Group, School of Computing
r.palani@kent.ac.uk

* Both senior authors have equal contribution to this work

Abstract. Healthcare field is highly benefited by incorporating BCI for detection and diagnosis of some health related detriment as well as rehabilitation and restoration of certain disabilities. An EEG dataset acquired from 15 high-functioning ASD patient, while they were undergoing a P300 experiment in a virtual reality platform, was analysed in this paper using three algorithms. Performance of Bayes Linear Discriminant Analysis (BLDA) was predominant over Convolutional Neural Network (CNN) and Random Undersampling (RUS) Boosting. BLDA rendered 73% overall accuracy in predicting target, while using the best accuracies for each subject using CNN or BLDA yielded an overall accuracy of 76%.

Keywords: Brain-Computer Interfaces, CNN, BLDA, RUSBoosting

1 Introduction

Brain-Computer Interfaces (BCI) is an integration of suitable hardware for acquisition of neural activities (having low signal-to-noise ratio), cognitive paradigms in order to do some tasks and machine learning algorithms for signal processing and classification. It is defined as, a system that measures central nervous system (CNS) activity and converts it into artificial output that replaces, restores, enhances, supplements, or improves natural CNS output and thereby changes the ongoing interactions between the CNS and its external or internal environment [1].

In the current study, a visual evoked potential based BCI incorporating P300 paradigm has been reported. P300 is an event related potential (ERP) component, usually elicited 300 ms after presentation of an infrequent target stimulus while undergoing an oddball experiment. Following the first real-time application of P300 based BCI, P300 speller, researchers got inclined to develop numerous applications of online BCI. We have applied various supervised classification algorithms and performed an offline analysis, to implement the task of target prediction over a few trials, in a online BCI dataset. To achieve a high classification accuracy, one needs appropriate temporal and spatial feature extraction techniques followed by machine learning algorithms to separate target and non-target classes.

A linear and two non-linear classifiers were applied here to determine the class of a trial and detect a target among eight objects after a few trials. Bayes Linear Discriminant Analysis (BLDA), a linear adaptive classifier most frequently and successfully used in P300 BCI, has been implemented. Convolutional Neural Network (CNN), a deep learning algorithm, which shows empirical evidence to extensively capture the temporal and spatial dependencies, motivated us to make an use of it here. Random Forest, an ensemble of weak learners like Decision Trees, renders satisfactory performance to classify linearly inseparable data, though produce biased results when applied to a skewed data. So a different ensemble aggregation approach, called Random Undersampling (RUS) Boosting, was applied to restrain the problem of class imbalance and bolster the influence of minority class (target). At last, an ensemble of all these classifiers was also implemented to check whether it outnumbers the accuracy of individual performances.

2 Methods

An off the shelf dataset collected from MEDICON 2019 Scientific Challenge, was analysed here. It was acquired from 15 subjects, suffering from high-functioning Autism Spectrum Disorder (ASD), in order to train social cognition skills. The detailed description of the experiment and dataset is given in [2, 3]. EEG was recorded from eight electrode positions in the parieto-central region with a sampling rate of 250 Hz. Our objective was to predict the target from each of the 50 blocks in a session given in online phase, where number of runs in a block varied. The steps we underwent to do target prediction is as follows.

2.1 Pre-processing

Pre-processing steps are nothing but extracting temporal attributes of the data, to amplify the differences between two classes of data, target and non-target, thereby eliminating the redundant information. It was done in the order mentioned below.

1. **Pre-stimulus mean removal:** Each epoch starts from -0.2s and ends at 1.2s relative to the event trigger. Baseline value of the pre-stimulus samples were calculated and post-stimulus samples (i.e. 1 s length) were taken into account to construct the feature vectors after removing the pre-stimulus mean.
2. **Band-pass filtering:** The data given was already band-pass filtered between 2 to 30Hz. We used MATLAB function *buttord* in order to determine the suitable order of the filter corresponding to pass-band frequencies 2 and

12Hz, stop-band frequencies 0.5 and 30Hz, stop-band attenuation of 40dB, pass-band ripple of 3dB; followed by a Butterworth band-pass filter using function *butter*.
3. **Downsampling:** The filtered data was decimated by a factor of 10 in order to remove high frequencies still left as well as reduce the dimension of the feature vector.
4. **Normalisation:** Each of the eight channels data was normalised epoch-wise to the interval [-1,1].

2.2 Classification

After extracting the relevant features, we need to determine the hyper-plane to separate target and non-target class. Epochs obtained after pre-processing were of dimension 8*30, which were concatenated into 240*1 column vectors associated with a label, before driving it to the classifier. We have used three supervised classification algorithms which are discussed below:

Bayes Linear Discriminant Analysis: LDA has been extensively and successfully used in BCI. It aims to maximise the between-class over the within-class variance. The idea of BLDA is an extension of Fisher's LDA. FLDA searches for discriminant vectors that yield a large distance between the projected means (target and non-target) and small variance around the projected means. BLDA was used to check over-fitting to high-dimensional and noisy data like the one presented here. It assumes that, labels and feature vectors are linearly correlated by set of weights with additive white Gaussian noise and perform regression. BLDA exhibited better performance than FLDA, when applied in a P300 BCI data [4]. All the necessary MATLAB codes to implement BLDA were taken from [4] and no manual intervention was needed to determine the hyper-parameters.

RUSBoost: P300 BCI data is usually attributed with high class imbalance. Deviant stimuli, which triggers P300 potentials, happens to appear much lesser than the abundant standard stimuli and this gives rise to a skewed data. Random Forest, an ensemble of Decision Trees, becomes biased because of the inadequate target class. To overcome this problem, we have followed a boosting approach presented in [5]. We have taken use of MATLAB Classification App *RUSBoosted Trees* and parameters used are mentioned below.

– Maximum number of splits- 20
– Number of weak learners- 450 (Decision Trees)
– Learning rate- 0.8

Convolutional Neural Network: Convolution Neural Network (CNN) has traditionally been used in a variety of image processing problems[6]. It is particularly fitting for images due to the two dimensional structure of the input layer.

In recent years, multi-dimensional biomedical time series data such as EEG from multi-channels have also been analysed using CNN [7].

CNN, especially deep CNN is very useful for these datasets as they circumvent the stage of feature extraction and in some cases, even pre-processing to reduce noise. It is usually time-consuming and also requiring a lot of experimentation to decide on the suitable features and even so, the extracted features may not capture all the necessary discriminatory information. Hence, it is more appropriate to use architectures such as CNN to automatically learn the discriminatory features from the dataset [8].

All pre-processing step were implemented except downsampling and the input size kept as 300 x 8. As the classifier was subject dependent, i.e. as training is done for each subject, the number of CNN layers was kept small at three to avoid overfitting. After some preliminary simulations using the provided test data labels from Phase 1, the CNN layer sizes were decided as 64, 32 and 16. Each layer had batch normalisation, rectified linear unit and max pooling. As we found overfitting occurred, we used a dropout layer before the max pooling layer. Dropout percentages were decided as 0.3, 0.2 and 0.1. The filter size was set at 3 x 3, which was also determined from preliminary studies. A fully connected layer of size 128 was used prior to the final classification layer of two nodes.

All the codes for processing were written in MATLAB version 2019a. CNN and BLDA outputs were obtained in the form of a regressed variable. Those continuous scores were summed over trials and the object corresponding to maximum of the summed values were selected as the target of the run. After aggregation of information from the runs, majority voting was used from the respective number of runs for each block and the output object was predicted.

3 Results and Discussion

All the calculations shown, were done on phase 2 data i.e. four sessions (S4-S7). Individual accuracy for each classifier is computed by averaging all sessions and subjects accuracy, which are 73%, 67% and 63% for BLDA, CNN and RUSBoost respectively. Fig. 1 shows the target prediction accuracy averaged over sessions for all subjects, using three classifiers mentioned above. A weighted (weights determined on the basis of previous performances) voting to predict a target from the ensemble of three classifiers rendered an overall accuracy of 73%. The best average accuracy for a subject, obtained from one of the classifiers, enhanced the overall accuracy to 76%. Fig. 2 depicts best accuracy of a subject through sessions, yielded by CNN or BLDA. It is evident in Fig. 1 that, though CNN produced very good accuracy for few subjects, BLDA performed best among all, in terms of accuracy and consistency. It is possible with skewness of the data corrected and deeper structures, CNN may produce improved results. The work has also shown that optimal classifier is subject dependent but it is possible to obtain high accuracy for some of the ASD patients, thereby showing the promise of the BCI approach.

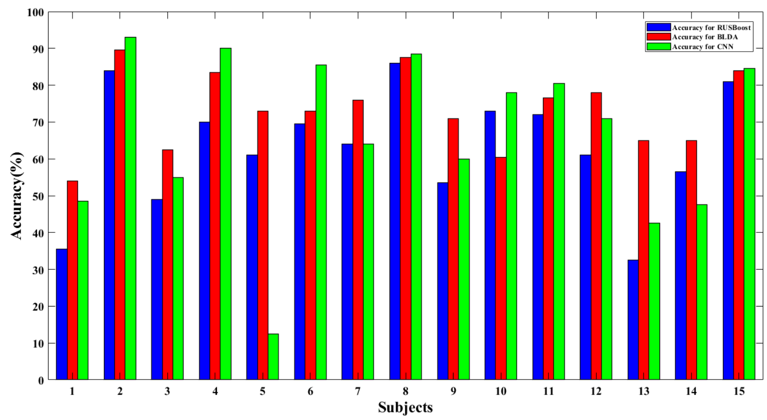

Fig. 1. Target prediction accuracy averaged over four sessions of phase 2 data using three classifiers for all subjects

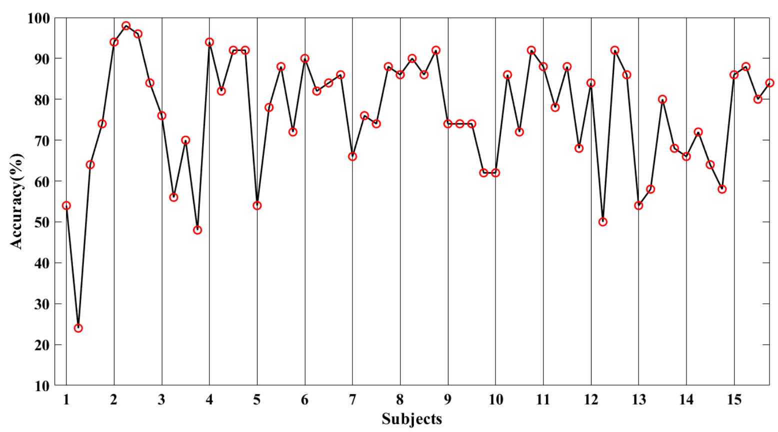

Fig. 2. Session-wise accuracy after selecting the best average performance between CNN and BLDA for each subject (for four sessions)

References

1. Wolpaw, Jonathan R., and Elizabeth W. Wolpaw. "Brain-computer interfaces: something new under the sun." Brain-computer interfaces: Principles and practice (2012): 3-12.
2. Amaral, Carlos, et al. "A Feasibility Clinical Trial to Improve Social Attention in Autistic Spectrum Disorder (ASD) Using a Brain Computer Interface." Frontiers in Neuroscience 12 (2018): 477.
3. Amaral, Carlos P., et al. "A novel Brain Computer Interface for classification of social joint attention in autism and comparison of 3 experimental setups: a feasibility study." Journal of Neuroscience Methods 290 (2017): 105-115.
4. Hoffmann, Ulrich, et al. "An efficient P300-based brain–computer interface for disabled subjects." Journal of Neuroscience Methods 167.1 (2008): 115-125.
5. Seiffert, Chris, et al. "RUSBoost: A hybrid approach to alleviating class imbalance." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 40.1 (2009): 185-197.
6. Tajbakhsh, Nima, et al. "Convolutional neural networks for medical image analysis: Full training or fine tuning?." IEEE Transactions on Medical Imaging 35.5 (2016): 1299-1312.
7. Cecotti, Hubert, and Axel Graser. "Convolutional neural networks for P300 detection with application to brain-computer interfaces." IEEE Transactions on Pattern Analysis and Machine Intelligence 33.3 (2010): 433-445.
8. Lawhern, Vernon J., et al. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces." Journal of Neural Engineering 15.5 (2018): 056013.