Code implementation for Distance Threshold Adjustment (DTA) and Distance-Based Self-Training (DBST) methods – Documentation

- *Gabriel Pires, *Institute of Systems and Robotics of the University of Coimbra* and *Polytechnic Institute of Tomar*, Portugal
- Aniana Cruz, *Institute of Systems and Robotics of the University of Coimbra, Portugal*
- Joana Figueiredo, *Institute of Systems and Robotics of the University of Coimbra, Portugal*
- Urbano J. Nunes, *Institute of Systems and Robotics of the University of Coimbra, Portugal*

**Corresponding author: gpires@isr.uc.pt*

This document describes how to use the provided code implementation for the Distance Threshold Adjustment (DTA) and Distance-Based Self-Training (DBST) methods.

INSTRUCTIONS TO USE CODE

The provided code implements a P300 BCI classification pipeline that includes preprocessing, feature extraction, and classification stages.

There are three main MATLAB scripts and two folders:

Main Scripts

Classification_DTA.m
MATLAB script implementing the Distance Threshold Adjustment (DTA) method. The DTA algorithm enables automatic detection of the control state.

Classification_DBST.m
MATLAB script implementing the Distance-Based Self-Training (DBST) method. DBST is an adaptive approach designed to mitigate inter-session variability and allows online recalibration using unlabeled data.

Epoch_extraction.m
MATLAB script used for extracting EEG epochs and visualizing target and non-target events.

Folders

Data: Contains EEG recordings from Sessions 1, 2, and 3 (for details, see Instructions about the Dataset).

DTA_DBST_toolbox: Contains helper functions for epoch extraction, signal preprocessing, classification, and performance metric computation.

Main parameters

sub: Subject identifier (ranges from 1 to 8).

Session: Specifies the session used to train the classification model (Session 1 or 2).

Session 1 - Calibration and Online session:

Calibration (Trainset1): Participants copied the sentence "THE-SLOW-RED-CAMEL" (18 characters, with '-' representing spaces). Each symbol flashed four times, resulting in 72 target and 1944 non-target samples. The classification model obtained from this dataset is referred to as **classifier 1**.

Online Test: The classifier was trained with Trainset1, and then participants performed the online session to write the sentence "THENC-NCREDNC-NCCAMELNC," where "NC" indicated non-control periods (participants fixated on the screen center without attending to any symbol). The sentence was repeated twice, separated by a 2-minute break. In total, 26 control and 10 non-control trials were recorded. Participants were instructed not to correct errors.

Session 2 - Calibration and Online session:

Conducted 1–3 weeks after Session 1. Calibration (Trainset2) was repeated under the same conditions as Trainset1. The classification model obtained from this dataset is referred to as **classifier 2**.

Online spelling followed the same protocol, but the classifier was trained using data from Session 1 (Trainset1).

Session 3 - Online session:

Conducted 1–3 weeks after Session 2. No calibration was performed; the online session used the classifier trained using data from Session 1 (Trainset1).

Performance Metrics

Let us consider that Nc and Nnc are the number of control and non-control trials. In the control state, there are three possible classification outcomes:

1. Correct control and correct target character (well-written character):

Pr^c = Nr^c / Nc

2. Correct control but wrong target character (misspelled character):

Pe^c = Ne^c / Nc

3. Control misclassified as non-control (false negative):

Pnc^c = Nnc^c / Nc

In the non-control state, two different outcomes are possible:

1. Correct non-control recognition:

Pnc^nc = Nnc^nc / Nnc

2. Non-control misclassified as control (false positive):

Pe^nc = Ne^nc / Nnc

Online Accuracy

Pac = 1 - (Ne / Nt) = 1 - (Ne^c + Nnc^c + Ne^nc) / (Nr^c + Ne^c + Ne^nc + Nc^nc + Nnc^nc)

The results correspond to:
Same session (SS) evaluation:
Average performance of classifier 1 tested in Session 1 and classifier 2 tested in Session 2, where calibration and testing occur within the same session.

Different session (DS) evaluation:
Average performance of classifier 1 tested in Sessions 2 and 3 and classifier 2 tested in Session 3, where calibration and testing occur on different session days.

This dataset is associated with a paper under submission entitled "*Self-paced ERP-based BCI speller with one-time calibration: A unified framework*", G. Pires, A, Cruz, J. Figueiredo, U. J. Nunes.

Acknowledgment

This work was supported by the Portuguese Foundation for Science and Technology under Grants BCI4ALL (COMPETE2030-FEDER-00842800, 2023.17977.ICDT), 2023.13809.PEX (DOI https://doi.org/10.54499/2023.13809.PEX) and UID/00048 - Instituto de Sistemas e Robótica - Coimbra (ISR-UC).