An efficient P300-based brain-computer interface for disabled subjects

Description of datasets and software

29.12.2006, document version 1.0
Available at http://bci.epfl.ch/p300
Author: Ulrich Hoffmann (ulrich.hoffmann@epfl.ch)

1 Introduction

This document contains a detailed description of the EEG datasets and software that were used to produce the results in the paper "*An efficient P300-based brain-computer interface for disabled subjects*" [1]. To make sense of the information given here, please first read [1].

If you use any of the datasets or software described here for your own publications, please do not forget to cite the original paper [1]. Should you have questions or find bugs in the software, please contact the author of this document.

2 Datasets

The data for each of the subjects is contained in a zip-archive with name `subjectn.zip`, where `n` is the subject number. Unpacking the archive for one subject yields four directories (`subjectn/session1` to `subjectn/session4`). Each of the directories contains six MATLAB data files. Each file corresponds to one run (one sequence of flashes). The following variables are contained in the data files:

- `data`
  This matrix contains the raw EEG. The dimension of the matrix is 34 $\times$ the number of samples. Each of the 34 rows corresponds to one electrode. The ordering of electrodes is: Fp1, AF3, F7, F3, FC1, FC5, T7, C3, CP1, CP5, P7, P3, Pz, PO3, O1, Oz, O2, PO4, P4, P8, CP6, CP2, C4, T8, FC6, FC2, F4, F8, AF4, Fp2, Fz, Cz, MA1, MA2. Each column corresponds to one temporal sample; the sampling rate is 2048 Hz.

  The data were recorded with a Biosemi Active Two system and are thus reference free. Arbitrary referencing schemes can be implemented by subtracting the reference channel(s) from all other channels. In order to obtain a good signal-to-noise ratio it is highly recommended to always use referenced data (see also the FAQ section on the Biosemi homepage `http://www.biosemi.com`).

- `events`
  This matrix contains the time-points at which the flashes (events) occurred. In each of the datasets, the first flash comes 400 ms after the beginning of the EEG recording. To find the sample corresponding to the first flash, the sampling rate (2048 Hz) has to be multiplied by 0.4 ($2048 \times 0.4 \approx 820$). To find the data samples corresponding to an arbitrary event E, the time of the first event has to be subtracted from the time of the event E. Then the time difference in seconds has to be multiplied by the sampling rate and the offset of 820 samples has to be added (see the MATLAB function `extracttrials` for an example).

- `stimuli`
  This is an array containing the sequence of flashes. Entries have values

between 1 and 6 and each entry corresponds to a flash of one image on the screen. The images on the screen are indexed as follows: 1) top left image, 2) top right image, 3) left image in the middle, etc. (cf. figure 1 in [1]).

- `target`
  This variable contains the index of the image the user was focusing on. For example if target equals four, the user was counting the number of flashes of the image on the right in the middle.

- `targets_counted`
  This variable contains the number of flashes that were actually counted by the user. Together with the number of events this can be used to check if the user was really concentrated. For example if there are 120 events, the number of flashes that were actually counted should be 20.

When working with the data please note that all experiments were performed under real-world conditions. This means that the data might contain artifacts coming from eye-blinks, eye-movements, muscle-activity, etc.. Furthermore some of the subjects were not always perfectly concentrated and thus some runs cannot be classified correctly, even with an optimal classifier (cf. [1]).

3 MATLAB Software

The zip-archive `p300soft.zip` contains four MATLAB functions and a subdirectory with three MATLAB classes. The functions use the classes for EEG preprocessing and classification. The software was developed and tested with MATLAB 7.3 under Windows but should also function with other operating systems and other versions of MATLAB.

3.1 Functions

- `setpath`
  This simple function adds the subdirectory `utilities` to the MATLAB-path. The subdirectory contains classes for preprocessing and classification of EEG data. `setpath` should be called once before calling other functions.

- `extracttrials`
  This function takes as first input the name of a directory containing EEG data from one session. The second input should be the name of a file to which the results are written. The function reads all the files in the directory, preprocesses the data, and extracts single trials from the data. The single trials are stored in an easily accessible structure and then saved in the output file.

- `testclassification`
  This function takes as first input a list of files containing training data. The second input is the name of a file containing test data. The training files as well as the test file have to be generated with the function `extracttrials`. The function computes a Bayesian linear discriminant from the training data and applies it to the test data. Results of testing are either shown in a simple plot or returned as an output argument.

- `crossvalidate`
  This function takes a list of files as input. The input files have to be generated with `extracttrials`. Given k files as input, `crossvalidate` uses `testclassification` to compute a classifier from k-1 files and test it on the left-out file. This is done k times (once for each file in the list). Then the results are averaged and plotted as accuracy and bitrate curves.

3.2 Classes

- `windsor`
  This class is used to "windsorize" EEG data. This means that for each EEG channel the following procedure is used: First the amplitude a_l is computed such that p% of the samples have amplitudes that are smaller than a_l (p is a user defined constant). Then the amplitude a_h is computed such that p% of the samples have amplitudes that are larger than a_h. Finally EEG samples with amplitudes smaller than a_l are set to a_l and samples with amplitudes larger than a_h are set to a_h.

- `normalize`
  This class is used to normalize EEG data. This means the amplitude values of each EEG channel are normalized to the interval [-1,1]. Alternatively the EEG channels can be normalized to have zero mean and standard deviation one.

- `bayeslda`
  This class is used to learn a linear discriminant from training data and to classify test data.

3.3 Example

After unpacking the dataset for subject1 to your working directory, the following sequence of commands can be used to reproduce the accuracy and bitrate curves from [1]:

```
setpath
extracttrials('subject1\session1\','s1')
extracttrials('subject1\session2\','s2')
extracttrials('subject1\session3\','s3')
extracttrials('subject1\session4\','s4')
crossvalidate({'s1','s2','s3','s4'})
```

References

[1] Ulrich Hoffmann, Jean-Marc Vesin, Karin Diserens, and Touradj Ebrahimi. An efficient P300-based brain-compuer interface for disabled subjects. *Journal of Neuroscience Methods*, 2007. submitted.