

BCI Competition IV Dataset 2a Submission by

Institute for Infocomm Research, Agency for Science, Technology and Research
Singapore (I²R, A*STAR)

Authors

Kai Keng, Ang (kkang@i2r.a-star.edu.sg)
Zheng Yang, Chin (zychin@i2r.a-star.edu.sg)
Chuanchu, Wang (ccwang@i2r.a-star.edu.sg)
Cuntai, Guan (ctguan@i2r.a-star.edu.sg)
Haihong, Zhang (hhzhang@i2r.a-star.edu.sg)
Kok Soon, Phua (ksphua@i2r.a-star.edu.sg)
Brahim, Hamadicharef, (bhamadi@i2r.a-star.edu.sg)
Keng Peng, Tee (kptee@i2r.a-star.edu.sg)

Files Description

Our training model *tmodel* and our predicted output classlabels *output* for each subject are saved in the MAT file '*bci4_2a_model_results_A0X.mat*' where *X* is the subject number.

The script to load the training model and perform classification on the evaluation to yield the output is *script_bci4_2a.m*. Please update the variable *datadir* in this script to the Dataset 2a directory prior to running the script. Note that the code for training the model is not included in our submission, but is available upon request.

Executing our script will save your predicted output labels to a MAT file *A0X_output.mat*, where *X* is the subject number. This is to facilitate the comparison of your output results with ours.

The algorithm and computation are performed using MATLAB 7.5 R2007b running Windows XP SP3. The BIOSIG toolbox for MATLAB/OCTAVE used is v.2.16. The execution of the evaluation code was verified on OCTAVE v3.01. Minor modifications of the code were incorporated to execute the code in OCTAVE in order to circumvent some memory issues. Hence, feel free to contact us if there is any problem executing the code in OCTAVE.

Method Description

The algorithm used is Filter Bank Common Spatial Pattern (FBCSP) [1] employing a multiclass extension using multiple One-Against-the-Rest (OVR) classifiers. Artifacts are removed by the algorithm using bandpass filtering. Each classifier selects discriminative CSP features using Mutual Information Best Individual Features (MIBIF4) [2]. For each component classifier, classification of selected CSP features is performed using the Naïve

Bayes Parzen Window classifier [2]. The final classification result is determined from all the probability results from each component classifiers.

As the computation of each and every time point of the evaluation data is computationally extensive, classification is performed on every alternate 10th sample and zero-order hold is used to map back to every time samples.

Causality Criterion Conformance

The FBCSP algorithm employs the Chebyshev2 filters using the MATLAB function *'filter'* instead of *filtfilt* in the code *prepbfilter.m* and *eegfilter.m*. The predicted classification output is delayed by a 2 seconds time sample as illustrated in Figure 1, since the algorithm performs computation using a 2 seconds window of EEG data at a time. The computation of the EEG data begins at the start of the fixation cross to the end of motor imagery, which comprises 6 seconds of EEG data in total. The classification output has been set to 0 for all other time samples that are not evaluated.

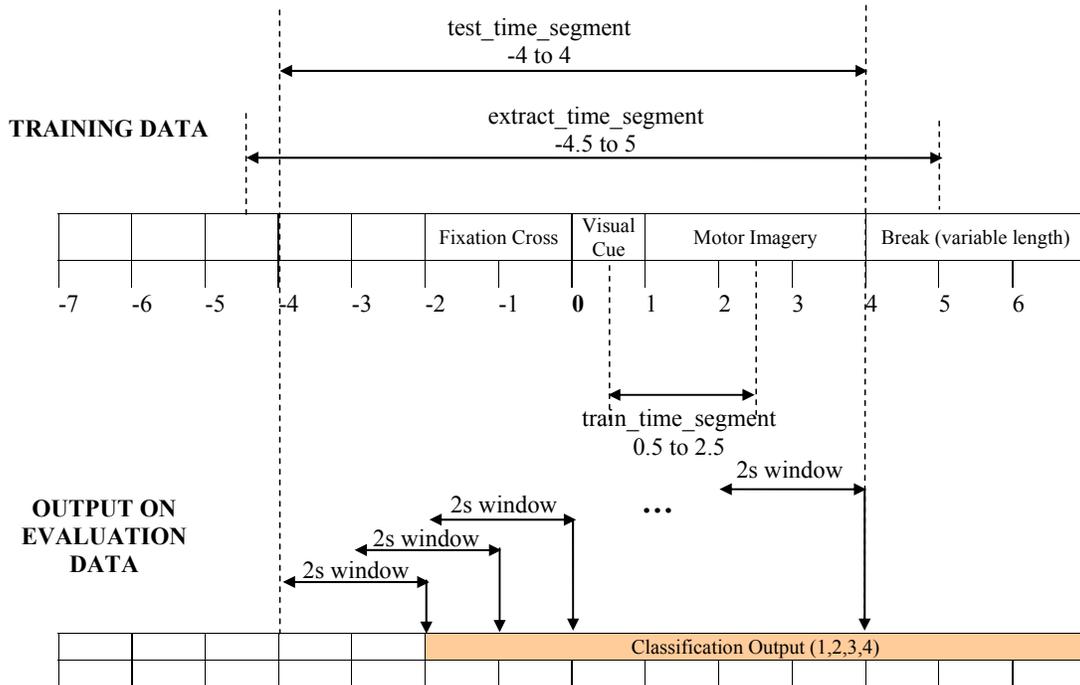


Figure 1: Illustration of the computation of the EEG data processing to conform to the causality criterion. All the time in this illustration is taken with respect to the cue at 0. The time interval -4.5s to 5s from the cue is extracted for computation. The time interval 0.5 to 2.5s from the calibration data is used for training the classifiers. The classification output on the evaluation data is from -2s to 4s, which is 0 to 6s from the start of trial. The output on the evaluation data is delayed by 2s due to the 2s window used for computation the classification output. An additional 2s more data is extracted before the start of trial in order to generate classification output at the start of trial, and the additional 0.5s is required to allow stable filtering.

Calculation of Kappa Coefficient for the Evaluation Data

The MATLAB function *getkappa_timeseries* has been included for the convenience of computing the maximum kappa value for the predicted classlabels of the evaluation set. An example script is shown below. An example of its use is also in *script_bci4_2a.m*.

```
timekappa = getkappa_timeseries(output,etruelabels,[1 2 3  
4] ,trig,6,250);
```

Figure 2: MATLAB function getkappa_timeseries

References

- [1] K. K. Ang, Z. Y. Chin, H. Zhang, and C. Guan, "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'08)*, Hong Kong, 2008, pp. 2391-2398.
- [2] K. K. Ang and C. Quek, "Rough Set-based Neuro-Fuzzy System," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'06)*, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 2006, pp. 742-749.