

Mental Workload Classification

Mansi Sharma

DFKI Saarbrücken, Germany

Passive Brain-computer Interfaces (BCI) are capable of estimating a user's cognitive state, mental workload, and are used in many other applications [5,6]. However, to make them usable in everyday life there are various challenges. One of them is large variability in the brain signals [4]. The given task in the Passive BCI Hackathon is to predict the mental workload (intra-subject estimation), where a subject performs three subtasks with the different levels of difficulties. BCI works in four main steps: Data Acquisition, Pre-processing and filtering, Feature extraction, and Classification and evaluation of the output. The provided pre-processed and epoched data is used for multi-class classification with easy, medium, and difficult conditions.

The filtered data is then used as an input for feature extraction, which also serves as a dimensionality reduction technique for better visualization of the data and improved predictive performance of the classifier.

Since, this is a multiclass problem, classification pipeline with feature extraction is implemented as follows:

Also, figure 2 describes the visualization of this pipeline

Step 1, the pre-processing of the data was performed using the EEGLAB toolbox [3] which consists of Epoching into 2-second non-overlapping epochs, Referencing using right mastoid electrode, High-pass filter 1 Hz (FIR Filter, pop-filtnew from EEGLab), Electrode rejection (average amplitude above 2 sd across channels) + spherical interpolation, SOBI with subsequent automated IC_Label rejection (muscle, heart and eye components were rejected with a 95% threshold), Low-pass filter 40 Hz (FIR Filter), Average re-referencing (CAR), Down sampling to 250 Hz.

The epoched and cleaned data (session 1 + session 2) is taken and CSP is used to extract the features from this input data. MNE [2], an open-source python library, is used to estimate spatial filters for feature extraction. Common spatial pattern, CSP, is used for feature extraction [7]. The basic principle behind CSP is to apply a linear transformation to project the multi-channel EEG signal data to a lower-dimensional spatial subspace. The transformation results in the maximization of the variance of one class while minimizing the variance of other classes at the same time. For the implementation, the number of filters is tuned for each subject, which lies in the range of (4 to 64). Also, all the 61 channels are used to avoid any loss of information. Later on, test data (session 3) is used for prediction using the trained model.

Step 2, the labels are arranged in three different ways. Figure 1 below describes this approach with labels Y_0 , Y_1 , and Y_2 . For instance, in Y_0 the samples of category easy condition (label 0) are marked as 1 while others as 0. Similarly, in Y_1 the samples of category medium condition (label 1) are marked as 1 while others as 0. Therefore, the matrix is a diagonal matrix.

Step 3, the model is trained using the calculated csp feature vectors in step 1 with the newly formed labels (Y_0, Y_1, Y_2). After training is finished, there are three models: Model_easy, Model_medium, and Model_difficult. For model training, Random forest method is chosen along with hyperparameter optimization using Grid search cross-validation [1]. Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest gives a class prediction and the class with the majority votes becomes the final model prediction.

The Random forest parameters: $n_estimators$, $max_features$, max_depth , min_sample_split are tuned for every subject to obtain the best combination of parameter values for each subject. All the other parameters take the default value. Also, for each subject the value of k (cross-validation parameter) is also tuned explicitly, which is either 5 or 10. Since, every subject will have a different combination of optimal values of Hyperparameters, this step is crucial.

Step 4, The trained model is finally ready and we can start making predictions with unseen or test data which is from session 3, which is the EEG data recorded on a different day (cross-session estimation). Suppose, the predicted values are P_0 , P_1 , and P_2 for each model. Then the maximum value is chosen from the three predicted values. In this way, the chance level is still 33.33% as the prediction is computed in a combined way by taking all the three models into consideration, giving equal importance while calculating the final prediction.

Labels	Label 0 (easy)	Label 1 (medium)	Label 2 (difficult)
Y_0 (easy condition)	1	0	0
Y_1 (Medium condition)	0	1	0
Y_2 (Difficult condition)	0	0	1

Figure 1: Arrangement of the labels for model training

Figure 2 describes the end-to-end pipeline with a clear demonstration of the steps mentioned before. Using the trained model, when we generalize the model using X_{test} data we get the prediction values for instance, for the first sample, predictions by all the 3 models are 0.2,0.6, and 0.4. The maximum value is 0.6. Hence, the assigned label is Medium. Similarly, for the second sample, the highest among 0.1,0.2,0.7 is 0.7 therefore, the assigned label is Difficult.

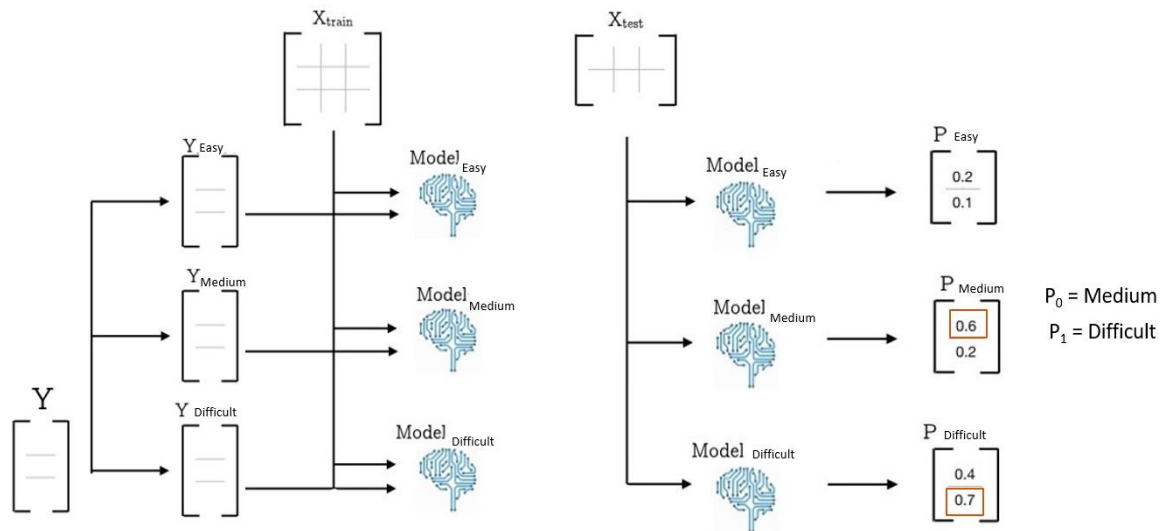
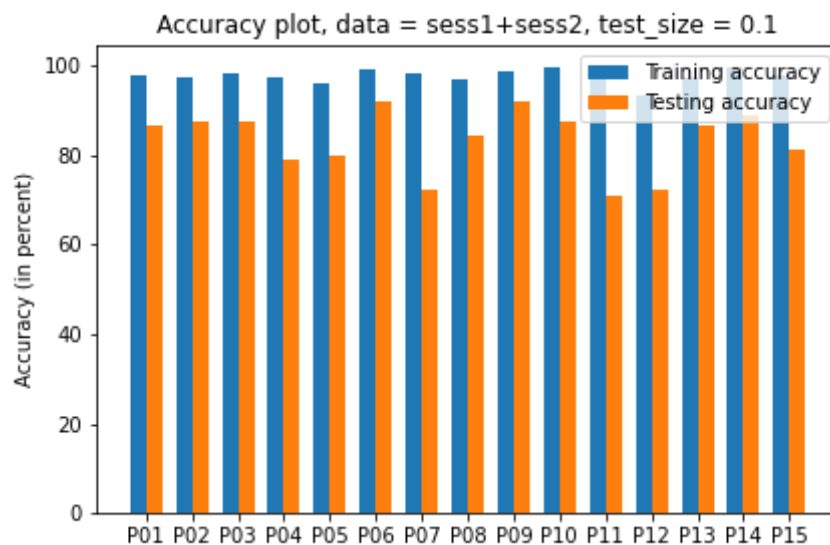


Figure 2: Visualization of the pipeline with training and testing scenarios.

Results and Evaluation

For model evaluation, the performance metric chosen is Accuracy, which is the ratio of a total number of correct predictions to the total number of predictions. For the demonstration of effectiveness of the proposed algorithm, the bar plot below describes the training and testing accuracies for each subject. The data is combined from session 1 and session 2 where 10% of the data is chosen as a test set for computing model performance. We can say that all the subjects managed to exceed the chance level of 33.33%. The highest test accuracy of 92.22% is reported by subject 6 and 9. The mean training accuracy is 97.74% while mean test accuracy is 83.34% across all the subjects. Since, in most of the subjects the difference between training and testing accuracies is not significantly large, we can deduce that the model is not overfitting.



Conclusion

The overall aim of this pipeline is to decode the mental workload from EEG signals on a new unseen session, from a training data set comprising several sessions. With the implemented methodology, it is possible to predict above the chance level. However, more advanced development is still required to effectively use it in day-to-day life.

References

1. LIAW, ANDY AND WIENER, MATTHEW. Classification and Regression by randomForest. *R News* 2, 3 (2002), pp. 18–22.
2. A. GRAMFORT, M. LUESSI, E. LARSON D. ENGEMANN-D. STROHMEIER C. BRODBECK L. PARKKONEN M. HÄMÄLÄINEN. MNE software for processing MEG and EEG data. *NeuroImage* Volume 86 (1 February, 2014), pp. 446–460.
3. Delorme, Arnaud, and Scott Makeig. "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis." *Journal of neuroscience methods* 134.1 (2004): 9-21.
4. Fairclough, S. H., & Lotte, F. (2020). Grand Challenges in Neurotechnology and System Neuroergonomics. *Frontiers in Neuroergonomics*, 1, 2
5. Zander, T. O., & Kothe, C. (2011). Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general. *Journal of neural engineering*, 8(2), 025005.
6. Wei, Chun-Shu, et al. "A subject-transfer framework for obviating inter-and intra-subject variability in EEG-based drowsiness detection." *NeuroImage* 174 (2018): 407-419.
7. Hatamikia, Sepideh, and Ali Motie Nasrabadi. "Subject transfer BCI based on composite local temporal correlation common spatial pattern." *Computers in biology and medicine* 64 (2015): 1-11.