Enhancing Human-Robot Interaction: Riemannian Classification and Clustering for Error Potential Detection and Error Occurrence Estimation

Mathias Rihet, Kalou Cabrera Castillo, Sébastien Scannella, Tresols Juan Jesus Torre, Frédéric Dehais, Raphaëlle N. Roy

Enhancing Human-Robot Interaction: Riemannian Classification and Clustering for Error Potential Detection and Error Occurrence Estimation

Mathias Rihet[1] , **Kalou Cabrera Castillo** [1] , **Sébastien Scannella** [1] , **Juan J. Torre Tresols**[1] , **Frédéric Dehais**[1] and **Raphaëlle N. Roy**[1]

[1]ISAE-SUPAERO, Université de Toulouse, France
mathias.rihet@isae-supaero.fr

1 Introduction

Our approach uses EEG signals to detect the occurrence of errors deliberately introduced during flexion and extension with an orthosis device, as well as to estimate the timing of these errors. It has been developed and tested on the IntEr-HRI competition dataset [Kueper *et al.*, 2023].

2 Methodology

The whole processing pipeline is available on github at https://github.com/mathiasrihet/IntEr-HRI.git.

2.1 Preprocessing and segmentation

In order to account for a realistic scenario, both training and test data have been loaded in a 900 ms buffer as a first step. A 15 Hz low-pass-filter and a common average reference (CAR) were then applied. Finally, a 500 ms sliding window with a leap of 100 ms allowed to segment data for classification.

A window was labelled as "error" if a physiological response to error (ERN) or to an odd event (P300) was expected in it [Yeung *et al.*, 2004]. As the P300b component has been noticed as particularly relevant to discriminate error condition according to training data first analysis, this response was selected as the main classification target. Thus, a window was labelled as "error" if the onset of a deliberately introduced error was present 400 ms before one point of the window.

It can be worth noticing that, despite this focus on the P300b component, the classification pipeline still processes the whole 500 ms window. Thus, any other relevant physiological response present inside the window can be integrated by the model.

2.2 Classification

All the steps described hereafter were performed on 32 channels selected from the 10/20 system.

Initially, the training set was prepared by randomly balancing the classes and normalized by dividing each sample by the standard deviation of the set. The testing set was subsequently normalized using the standard deviation of the training set. Classification was then carried out using a Rie-

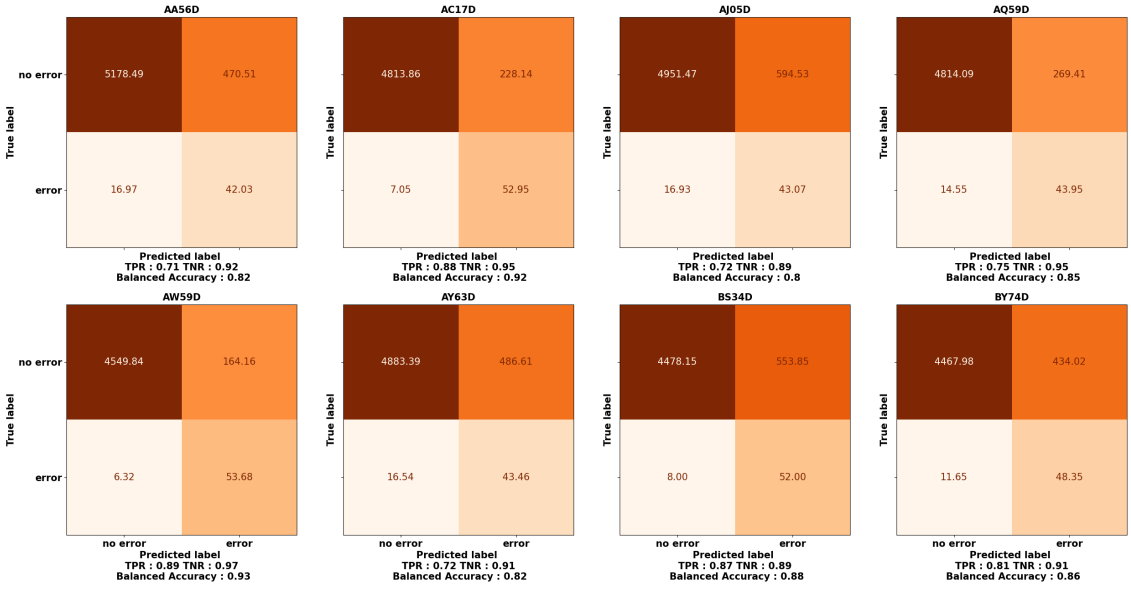

Figure 1: Confusion matrices of the 8 within subject 10-fold cross-validations.

mannian approach, employing the pyriemann Python library [Barachant *et al.*, 2022]. Covariance matrices were computed and spatially filtered using a Riemannian variation of the common spatial pattern (CSP) algorithm [Barachant *et al.*, 2010]. These matrices were finally projected from the Riemannian space to the tangent Euclidean space in order to train an SVM model.

2.3 Sample selection

Due to the infrequent occurrence of errors, the tested model generated a substantial number of false positives, failing to accurately predict the required number of 6 errors for the competition. Consequently, an additional step was introduced to identify the most probable real errors among those predicted by the model.

The predicted errors were organized into clusters using a straightforward adjacency rule, where errors predicted on adjacent windows were grouped together. Clusters exceeding a predetermined threshold were then chosen. To ensure the selection of the smallest subset with a count higher than 6 clusters, an iterative process was employed to define the threshold. Subsequent processing enabled the retention of only the six most evenly distributed clusters within the subset.

While the last part is a pure offline technique, it may be worth noticing that the used threshold could be learned from the training set.

Finally, the centroid window of each selected cluster was subjected to peak detection to identify the targeted physiological response. Although it was possible to calculate the typical latency between this response and the error onset to predict the error onset with a higher accuracy, we opted against this technique. The reason behind this decision was that it could potentially lead to predicting errors before their actual occurrence. Instead, we considered the sample of the detected peak as the representative error sample.

3 Results

3.1 Error potential detection

The approach was evaluated within each subject using a 10-fold cross-validation on the 8 provided datasets. In addition, because these data were not totally independent, special attention was paid to fold construction. Indeed, a classical leave-one-out 10-fold cross-validation could have included windows in the training set that overlap with windows from the testing set. Thus, for each fold, the 8 runs were randomly split into 6 runs for training and 2 for testing. This methodology was expected to introduce less bias and generalize well to the validation step on test data by following a similar approach.

The mean confusion matrices of these cross-validations can be found in Figure 1. For a more intuitive evaluation of the detection performance, the true positive rate (TPR) and true negative rate (TNR) were also reported.

$$TPR = TP/(TP + FN) \quad (1)$$

$$TNR = TN/(TN + FP) \quad (2)$$

3.2 Error onset estimation

Following the clustering step designed to allow for sample detection, sample indices of errors detected in each run of the test data are presented in Table 1.

4 Conclusion

The methodology proposed here is a first step towards online error detection in order to provide a means to monitor human-system interaction, and eventually to adapt this interaction for increased efficiency. It will be completed for the online stage of the competition by addressing online specific issues such as cross-subject and cross-session classification.

Participant	Run	Error samples					
AA56D	5	22898	38618	52668	65100	80687	105767
AA56D	6	45922	58361	76681	97301	109949	124565
AC17D	5	24771	45009	65716	86053	106129	121108
AC17D	6	43028	50743	59532	86156	116132	124529
AJ05D	5	39728	46623	59578	73669	121492	129899
AJ05D	6	34691	58134	72881	83627	121534	134377
AQ59D	6	31185	45557	95164	105995	116682	132537
AQ59D	7	15209	20705	35082	51383	73555	119644
AW59D	5	16166	28509	53695	82479	95132	104986
AW59D	6	80220	80356	90004	99578	108166	116035
AY63D	5	30978	44811	76371	101424	116300	132576
AY63D	6	6355	68807	69857	89114	106391	123696
BS34D	5	40133	50415	67127	84092	111066	128742
BS34D	6	32915	39619	70110	83531	100290	113748
BY74D	5	36075	46036	55689	85030	85199	108113
BY74D	6	63393	63536	79888	79889	100588	116639

Table 1: Samples predicted on test data for validation phase.

References

[Barachant *et al.*, 2010] Alexandre Barachant, Stphane Bonnet, Marco Congedo, and Christian Jutten. Common spatial pattern revisited by riemannian geometry. In *2010 IEEE International Workshop on Multimedia Signal Processing*, pages 472–476. IEEE, 2010.

[Barachant *et al.*, 2022] Alexandre Barachant, Quentin Barthélemy, Jean-Rémi King, Alexandre Gramfort, Sylvain Chevallier, Pedro L. C. Rodrigues, Emanuele Olivetti, Vladislav Goncharenko, Gabriel Wagner vom Berg, Ghiles Reguig, Arthur Lebeurrier, Erik Bjäreholt, Maria Sayu Yamamoto, Pierre Clisson, and Marie-Constance Corsi. pyriemann/pyriemann: v0.3, July 2022.

[Kueper *et al.*, 2023] Niklas Kueper, Kartik Chari, Judith Bütefür, Julia Habenicht, Su Kyoung Kim, Tobias Rossol, Marc Tabie, Frank Kirchner, and Elsa Andrea Kirchner. Eeg and emg dataset for the detection of errors introduced by an active orthosis device, 2023.

[Yeung *et al.*, 2004] Nick Yeung, Matthew M Botvinick, and Jonathan D Cohen. The neural basis of error detection: conflict monitoring and the error-related negativity. *Psychological review*, 111(4):931, 2004.