Detection of Error-related Potentials in EEG Data using Maschine Learning Methods in Matlab

Practical Course Biosignal Processing and Modeling

Chair for Cognitive Systems
Technical University of Munich

Supervisor: Alireza Malekmohammadi

Dominik Scherer

Enrollment number: 03789014
E-mail: dominik.simon.scherer@tum.de
June 3, 2024

Table of contents

Acronyms

BCI brain-computer interface

CV cross-validation

EEG electroencephalography

ERP event related potential

ErrPs error-related potentials

LDA linear discriminant analysis

MCR missclassificationrate

PCA principal component analysis

SNR signal-to-noise ratio

SVM support vector maschine

List of Figures

List of Tables

1 Introduction

The objective of the experience was to collect electroencephalography (EEG) data of a person perceiving his intended actions being executed correctly or with an error. The user initiated a cursor to move left, right or up based on a visual input. The cursor moves in the correct direction, but in 35 % of the cases, also in the wrong direction - independent of the user's input. The purpose of the experience is to collect data on the brain's reactions to a correct or incorrect machine response and to evaluate whether they can be distinguished by analyzing the EEG data.

These error-related potentials (ErrPs) are important in higher-level brain-computer interface (BCI) with continuous feedback. Decisions based on EEG data are prone to misclassification errors, and even the best BCI will make wrong decisions. Therefore, it is crucial to detect execution errors to stop or reverse a wrongly initiated action. The experiment could be used to develop an algorithm to detect ErrPs in continuous EEG data to implement a direct feedback loop for the decisions taken by a BCI [1].

2 Methodology

2.1 Preprocessing

EEG data has a very low signal-to-noise ratio (SNR) due to its acquisition method. Additionally, common artifacts due to eye, muscle or cable movement are magnitudes higher than the actually important data. Therefore, it is crucial to develop a pipeline that extracts meaningful data and reduces noise and artifacts as far as possible. This Preprocessing often involves the following steps.

2.1.1 Filtering

High-pass filtering is applied to remove baseline drifting caused by unstable impedances and to eliminate the DC component of the signal. Low-pass filtering is applied to remove frequencies $\gtrapprox 50\,\mathrm{Hz}$ because signals with higher frequencies cannot be distinguished from the noise floor anymore.

High- and low-pass filters can be applied in a combined manner, using a band-pass filter. If necessary, an additional notch filter can be applied to remove powerline noise.

2.1.2 Bad channel rejection and interpolation

Since each channel's performance depends on the cable, connectors and how good the corresponding electrode is picking up the EEG signal, it can happen that single channels fail or deliver only noise as a signal. Therefore, single channels can be rejected to remove data with no information. This only works in single-session experiments with a single subject since cross-subject experiments will result in different rejected channels, affecting the outcome of the experiments.

EEG channels are spatially correlated since an electric signal does not travel in a straight line from source to electrode. Rather, the conduction through the brain, skull and skin results in the electric potential to spread and reach other electrodes. This is called volume conduction and leads to neighbouring electrodes picking up similar signals. While decreasing the spatial resolution, this phenomenon can be exploited by interpolating a rejected channel via the surrounding functional channels.

2.1.3 Ocular artifact reduction

Ocular artifacts emerge from eye movement and eye blinking and are most prominent in the frontal region. Since the magnitude of the artifacts is many times higher than the EEG signal, they can be detected rather easily, and there are regression methods or principal component analysis (PCA) to remove them from the data [2].

2.1.4 Common average re-referencing

Common average re-referencing works by computing the global mean of all channels and subtracting that mean from each individual channel. This reduces artifacts that affect all channels since they will get averaged out. However, this method is sensible for single-channel artifacts since a single bad channel will affect every other. Therefore it makes sense to apply bad channel rejection before common average re-referencing.

2.1.5 Example of preprocessing

To illustrate the impact of preprocessing on EEG data, Figure 1 showcases some of the previously mentioned methods.

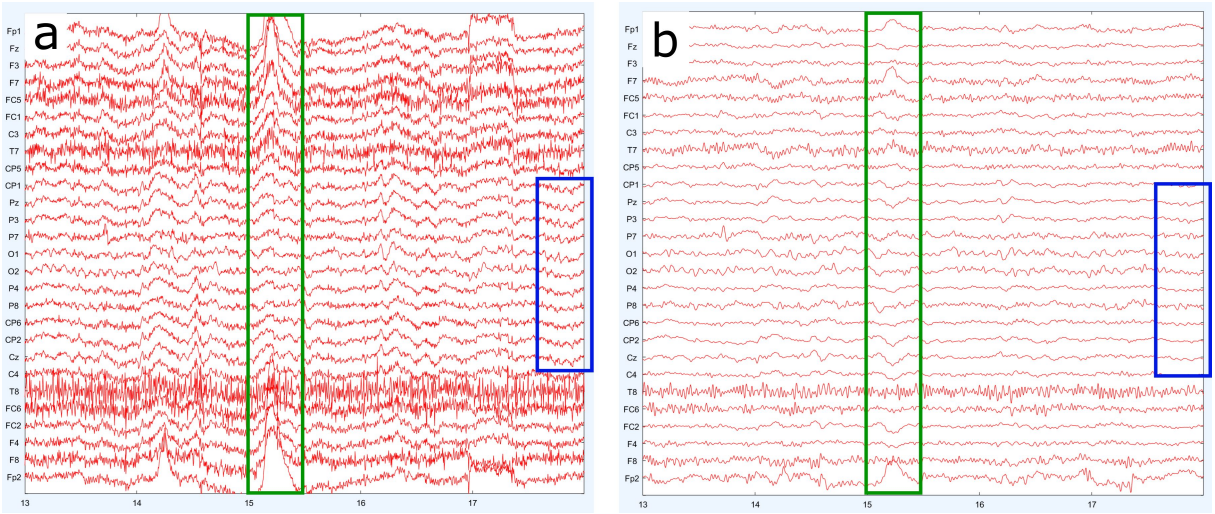

Figure 1: **a)** Unprocessed data (only highpass and downsampling). **b)** Preprocessed data with bandpass filtering, bad channel rejection and interpolation, Ocular artifact detection and common average re-referencing. The green boxes highlight the effect of ocular artifact reduction, while the blue ones showcase the effect of bandpass filtering (smoother signal due to missing high frequencies). Also, the common average re-referencing is visible across all channels as it scales the channel amplitudes to a similar level.

2.2 Analysis of ERP components

The average event related potential (ERP) time courses show that there was an inherent difference between the averaged potential when no error was present versus the averaged potential when an error was present. This hints that the EEG data can be used to determine whether an error was present or not. From the topological plots, we can also see that the difference is most prominent at the electrode "Cz" (Midline Central). The amplitude difference between non-error and error around the timepoints 200 ms and 300 ms is the largest, as shown in Figure 2.

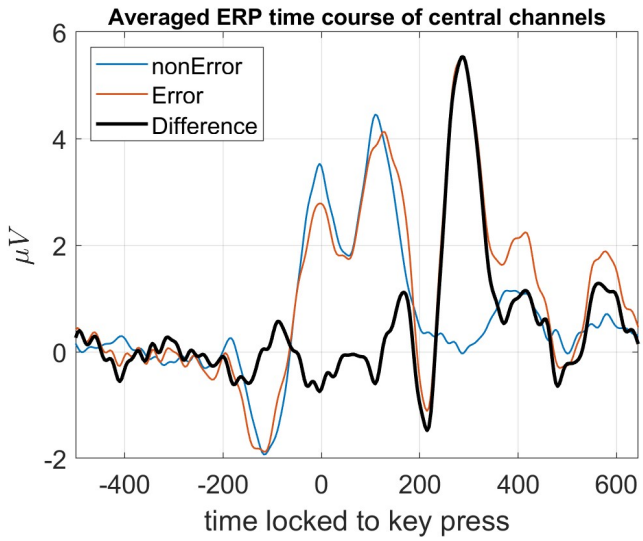

Figure 2: Average ERP across epochs of the (averaged) central channels. The difference between error and non-error events is most prominent at the time points $\approx 200\,\text{ms}$ and $\approx 300\,\text{ms}$.

From the amplitudes at these time points (also named N200 and P300 because of the negative and positive peek), the EEG signal can be used to determine whether an error or non-error event occurred.

Other important features could include the potential at around $\approx 450\,\text{ms}$ and $\approx 550\,\text{ms}$ since it appears to follow a similar pattern than the N200 and P300 peaks but less pronounced. Also, spatial features like the potential difference at P300 between central channels and channels on the edge could be used to distinguish error from non-error, as visible in Figure 3 .

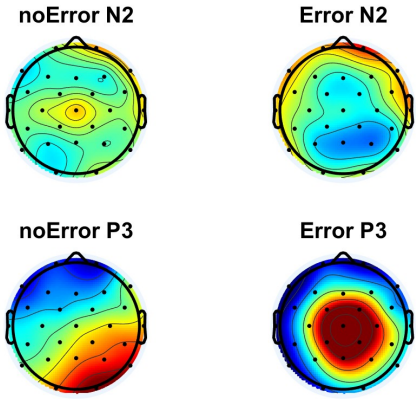

Figure 3: Topological plot to show the amplitude differences between channels at the time points of N200 and P300 (here N2 and P3). The error/non-error difference is the largest between the central and lower-left edge (occipital 2) channels of the P3 plots.

2.3 Feature Extraction and Evaluation

Feature extraction, in general, means to extract the most meaningful pieces of information from the data, which then can be used to train the model. It can be done in numerous ways. In the tutorial example, we searched for the maximum difference in ERP between the two different stimuli (error/non-error). In general classification tasks, features can be anything that helps to distinguish the different classes.

The approach used in the tutorial is limited by the fact that the ERP was extracted at exact time points. This might be justifiable by looking at the average ERP, but single events might be delayed or occur faster due to the subject's attention or external factors. Furthermore, the feature is extracted at every electrode, even at those where no or only a weak signal difference is expected. This leads to less discriminative power when the features are not regularized or selected later. Even with regularization, feature selection can be important to decrease computational complexity and avoid overfitting.

Therefore, I would suggest taking the average of the ERP over a certain time period to also catch slightly shifted peaks and only use the features with a Fisher score above a certain threshold. Additionally, other features could be extracted to check if they help improve the model's performance, like features in the frequency domain or spatial features like taking the difference of potential between spatially separated electrodes. To achieve an even more robust model, the average of several neighbouring electrodes could be used to calculate the features. In any case, but especially when the features are extracted differently, feature scaling and normalization should be considered. In my code, this is ensured by applying the "zscore" function on my feature matrix before training the classifier.

2.4 Classification

In the modelling step, the actual model is built. The model suggested in the tutorial was a linear discriminant analysis (LDA), which is a supervised machine learning method that tries to maximize the separability between classes by performing a linear transformation of the features to maximize the ratio of between-class variance to within-class variance. The model then fits weights and biases to estimate the feature-label relationship based on the transformed features.

2.4.1 Evaluation metrics

After the fitting, the model is basically a function that maps the set of input feature values to the labels. Different metrics can be used to evaluate how well this function is performing on new data. The most simple one might be the accuracy, which simply tells what fraction of all predictions were correct. Similarly, the missclassificationrate (MCR) tells what fraction of all predictions were wrong (1 - accuracy). While giving quick performance evaluations, these metrics can give the wrong impression when the classes are highly imbalanced. Therefore, other metrics like Precision (proportion of true positive predictions out of all positive predictions) or recall (proportion of true positive predictions out of all actual positive instances) can be used. Recall and Precision constitute a tradeoff since increasing one will decrease the other. When there is no intended bias in the classification (like making sure all positive results are retrieved while accepting decreased precision), the f1-score is widely used. The f1-score is the harmonic mean of precision and recall and, therefore, aims to balance the two when being maximized.

2.4.2 Cross-validation

The significance of all aforementioned metrics is compromised when not measuring the model's performance on new "unseen" data. Otherwise, we can't tell if we have a nice generalizing model or one that massively overfits every data point. So, to evaluate the performance of the model somehow before using it for the real application we can use cross-validation (CV). CV randomly holds back a portion of the data on which the model is not trained but tested later on. Especially for small data, this random selection can be problematic since the training data shrinks, and the small random testing data might not be a sufficient amount to represent the data. Therefore, CV iterates the process multiple times and takes the average of the evaluation metric to receive a more robust evaluation of the model.

2.4.3 Evaluation methodology and results

For the CV I used ten folds because that means every test set still has enough samples to be tested on (30 out of 300 samples) but also have enought folds to get a nice average value for the evaluation metric. Since 300 samples are not that much, a smaller number of folds (like 5) might be even better. As a metric, I used the MCR and got values between 0.35 and 0.4.

2.4.4 Interpretation of results

A MCR of 0.4 means that 40 % of all samples where misclassified. Since randomly choosing labels would result in a MCR of ≈ 0.5, the result is not really good. Also, it varies every time I run the CV, which means it is not very robust. So, I am not comfortable applying this classifier to the recall set. To be more confident I would want a MCR below 0.1 and have a more robust model that shows similar results when CV is done multiple times.

2.5 Cross-comparison of methods

2.5.1 Feature extraction

To improve the model, I first changed the feature extraction method. Instead of taking the ERP at certain time points, I automatically chose the peaks in the cross-epoch average and extracted the mean of the ERP across a certain time range to ensure that I also captured slightly shifted peaks. Furthermore, I calculated the difference of potential between (spatially averaged) channels to extract the ERP peaks in a similar manner, resulting in spatial features. Lastly, I tried to extract features in the frequency domain, but I think the discriminative power of these features is rather low. At least how I did it.

2.5.2 Feature selection

I also changed the feature selection process to only consider features with a Fisher score above a certain threshold in order to avoid overfitting with features with low discriminative power. This could also be achieved with lasso regularization, which is capable of shrinking the weight of unimportant features to zero, effectively performing feature selection. Another approach would be to use multivariate feature selection (wrapper methods) with sequential forward/backward selection, but for simplicity, I stuck with the Fisher criterion.

2.5.3 Model

I used a support vector maschine (SVM) because it works well with high dimensional data and small data sets, which is given in our case. A SVM tries to find a hyperplane (in 2D, this corresponds to a line and in 3D to a plane) which best separates all points with different labels in the feature space. The hyperparameter "C" defines how many misclassifications the SVM is allowed to make in order to get a more generalized classifier. By using a SVM, I expected the performance to greatly improve because the LDA classifier assumes common covariance matrices across classes, which is hard to justify with high-dimensional data and different feature extraction methods. Additionally, LDA compresses the features to one less than the number of classes, which can lead to a loss of information when dealing with high-dimensional data.

2.5.4 Cross-validation and evaluation

The cross-validation was done similarly to the LDA method, but I performed it on a large number of different values for the hyperparameter "C" to see which gives the best result. I chose "C" based on the f1 score to ensure a good balance between precision and recall. Additionally, I iterated the whole CV process several times to account for the random nature of choosing the subsets for the CV and to get a quick overview of how much my result varies between CV iterations. For 500 checked "C" values (logspaced from 0.05 to 30) and 10 CV iterations, I received the results shown in Figure 4 and table 1.

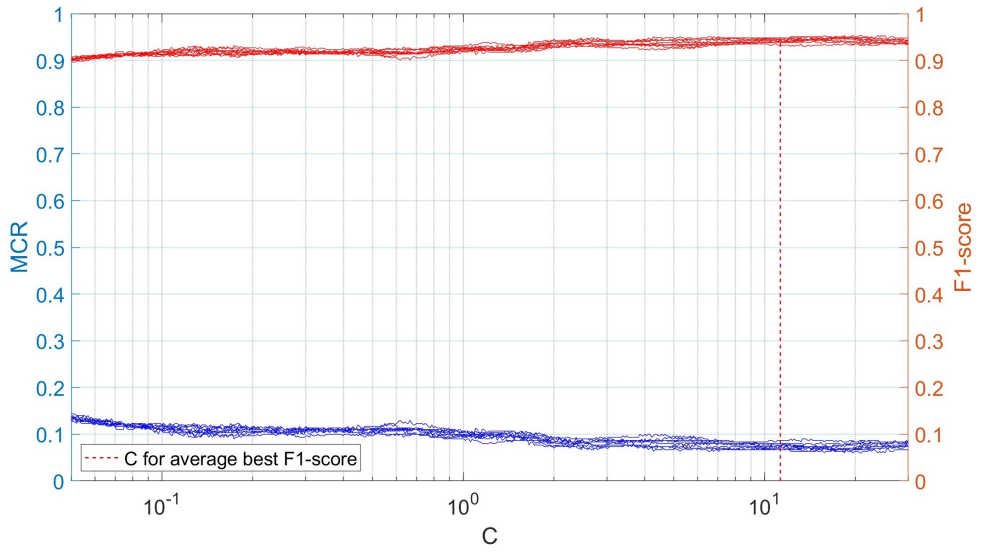

Figure 4: F1-score and MCR over C values for different CV iterations.

	Value	standard deviation
Best C	11.2828	6.9356
F1 score	0.94797	0.0046188
Accuracy	0.933	0.0059732

Table 1: Cross-validation results

Besides the SVM model, I tried boosting algorithms like "AdaBoost" and "GentleBoost". However, I could not achieve as high accuracies and F1-scores as with the SVM model.

3 Conclusion

Thanks to extended feature engineering, feature selection, a different model kind and evaluation metric, the model implemented in "Tutorial_2_improved.m" is a robust model to classify given EEG data in the two classes error and non-error. The MCR was kept below 0.1, and the model could be used on new data to evaluate whether an executed task was performed correctly or needs to be adjusted. In the case of the initial experiment, for example: Does the cursor really move in the desired direction, or is it an error event? The next step would be to implement an error correction.

The following could also be implemented and tested whether it is possible to further improve the model.

- Oversampling of minority class
- Improve spatial/temporal/frequency feature extraction
- Improved feature selection (e.g. through wrapper methods)
- Use and improve ensemble classifiers

References

[1] Martin Spüler and Christian Niethammer. Error-related potentials during continuous feedback: using eeg to detect errors of different type and severity. *Frontiers in human neuroscience*, 9:155, 2015.

[2] Xiao Jiang, Gui-Bin Bian, and Zean Tian. Removal of artifacts from eeg signals: a review. *Sensors*, 19(5):987, 2019.