Automatic optimization of parameters for Ocular Artifact Correction in EEG

Benjamin Ahm, Emil Riis Hansen, Kristian Hauge Jensen,
Morten Korsholm Terndrup

*Aalborg University
mternd13@student.aau.dk

Abstract

Here is the abstract.

Introduction

The field of Brain-Computer Interfaces (BCI) has in the recent years been under active research, especially with the popularity of machine learning techniques. The reason for the interest, is the many useful application of a well-working BCI, such as replacing lost motor function in disabled people, helping with analysis in brain imaging to diagnose brain conditions or novel applications in computer games.

The general idea of a BCI is to measure brain activity represented by electroencephalogram (EEG) signals, by putting sensors on the scalp, which can measure the electric impulses. However, the EEG data is noisy at best, and this problem can severely affect the results of classification algorithms. Therefore, signal processing for extracting important components of a signal or removal of noise, is an important step in any given BCI.

This leaves us with several steps in which several techniques may be applied to obtain a corrected EEG signal and consequently obtain a model that classifies the EEG data reasonably. Each technique applied may require several parameters to be tuned for obtaining the optimal results, such as the regularization parameter for a Support Vector Machine. Such tuning are usually done manually, by experimenting with different values to see their effect on some validation data. Users of the BCI or medical professionals might be knowledgeable about tuning some parameters but not all, hence it requires either an expert to help determine them or extensive training. Nonetheless, it requires a great deal of time for tuning the parameters, to obtain the best results.

Another, more useful approach would be to automatically infer the hyper-parameters from the training data. Recent work about algorithmically optimizing machine learning parameters has seen popularity by using Bayesian Optimization (insert reference), in which the learning process is seen as a black-box function for which parameters can be optimized.

Related Work

Much research effort has been put into developing or applying techniques for noise/artifact correction in EEG signals. Well-known methods, such as *Principal Component Analysis*, *Independent Component Analysis* or *Discrete Wavelet Transform*, in the signal processing world has had mixed results. Most of these techniques considers how to extract the information from the signals, instead of reducing the noise in the data. Other approaches such as (source) considers removing noise from a correction perspective. [ECG source] uses EOG signals measured by sensors on the subjects eyes, and estimates a propagation factor to determine the amount of the EOG signal to remove from the

*A thank you or further information

EEG signal, thus using the EOG signal as an artifact signal. Similarly, [OACL guys] have had positive results in estimating a pseudo-EOG signal for binary class EEG data, making it possible to obtain an artifact signal without using any secondary measurements.

Optimization of parameters for Ocular Artifact Correction

short overview, maybe show a figure describing the pipeline.

Ocular Artifact Correction

We adapt the Ocular Artifact Correction (OACL) technique developed in (add reference) for multi-class datasets. The OACL method consists of two parts. First, we analyse the raw EEG signals to obtain the artifact signals, representing the parts of the raw signal that was determined to be noise. Then we find the filtering parameter of each the artifact signal, which determines "how much" of each signal that should be removed from the raw signal.

Let $x = \{s_0, ..., s_n\}$ where $s_i \in \mathbb{R}_{\geq 0}$ denote the raw EEG data for some arbitrary channel. For simplicity, we can interpret x as a function $x : \mathbb{N}_{\geq 0} \to \mathbb{R}_{\geq 0}$ where $x(t)$ denotes the amplitude of x at time t.

From $x(t)$ we perform all steps in artifact detection and removal.

Artifact Detection

The goal of artifact detection is to find some artifact signal $a(t)$ from $x(t)$ that represents which parts of $x(t)$ that are noise. Before finding the artifact signal $s(t)$, we first obtain a smoothed signal by applying a *moving average filter* to $x(t)$:

$$\textbf{moving avg equation/pseudocode} \tag{1}$$

where m is the number of neighboring points, and n denotes the number of EEG samples in $x(t)$. From $s(t)$ we calculate the relative heights between samples as the maximal difference in amplitude between a sample and its neighboring samples.

$$\Delta(t) = max(|x(t) - x(t-1)|, |x(t+1) - x(t)|) \tag{2}$$

Now, we want to have some measure of what an artifact signal looks like. [OACL] found by inspection that ocular artifacts generally occur with sudden changes in amplitude (ΔA) between $[30\mu V - 50\mu V]$ and $[70\mu V - 150\mu V]$. For now, assume that we have some arbitrary ranges

$$h_r = [l, u] \quad l, u \in \mathbb{N}_{\geq 0} \tag{3}$$

then we can find the points in time where $\Delta(t)$ lies in the range h_r:

$$P = \{t \quad | \quad \frac{m}{2} < t < n - \frac{m}{2} \quad \text{and} \quad l < \Delta(t) < u\} \tag{4}$$

Artifact Removal

Explain how we use the artifact signals to remove artifacts from the eeg signal.

Filter-bank CSP

Filter-bank common spatial patterns $FBCSP$, is commonly known as a valid way of extracting relevant features from EEG data . The drawback of this method, is the limitation of CSP being a binary

Bayesian Optimization

Here we explain what Bayesian Optimization is, reasons for doing it and how it works.

Classification

With the feature vector $\mathbf{X}$ extracted by the multi-class CSP algorithm, we can now train a classifier for multi-class motor imagery.

Several algorithms have seen popularity for classifying EEG data. The survey by Chan et al. (2015) on the performance of ensemble methods in EEG context argues that Random Forests more accurately classifies EEG data than other well-known methods such as k nearest neighbors and Support Vector Machines. Sun et al. (2007) also surveys the effectiveness of ensemble methods, but argues that the results are subject to the choice of base classifier as weak learners.

To evaluate the results we train SVM and Random Forrest classifiers to see if there are any significant difference by these algorithms. We use the classification/standardization/scaling algorithms from the Scikit-Learn library for Python [Pedregosa et al. (2011)].

Support Vector Machine

The Support Vector Machine classification is a supervised machine learning algorithm that works by finding a hyperplane that discriminates samples but also maximizes the margin between the different classes. In the case of multi-class classification, we use a "one-vs-rest" approach where we for 4 classes construct 4 binary classifiers that classifies one class against all others. This scheme is aggregated into the final decision function for classifying future trials.

For the SVM we perform Bayesian optimization with respect to the regularization parameter C and choice of kernel as well as the γ parameter in the case of RBF kernel. Since SVMs are not scale invariant, we scale and standardize the feature vector $\mathbf{X}$ before training, such that the model is not dominated by a few features.

Random Forrest

The Random Forrest learning algorithm works by randomly splitting the training set into n subsets and trains *n weak learners* (such as Decision Trees) on one training subset each. When classifying new samples, the Random Forrest classifies on each weak learner and returns the mode result. Intuitively, the weak learners 'vote' on the result.

For training the Random Forrest classifier, we perform Bayesian optimization on the number of weak learners used.

Experimental Results

Data Examination

Discussion

Here we discuss the results given in section 3, and talk more about what the results imply/how it could be improved.

Conclusion

Here we conclude on the paper by summarizing what we did and what our results was. Furthermore we address how the problem could be further improved/investigated.

References

Chan, Alexander, Christopher E Early, Sishir Subedi, Yuezhe Li, and Hong Lin (2015), "Systematic analysis of machine learning algorithms on eeg data for brain state intelligence." In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, 793–799, IEEE.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011), "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research*, 12, 2825–2830.

Sun, Shiliang, Changshui Zhang, and Dan Zhang (2007), "An experimental evaluation of ensemble methods for eeg signal classification." *Pattern Recognition Letters*, 28, 2157–2163.