Combining feature extraction methods in SSVEP-based BCI

[Limited access]

Anti Ingel
Institute of Computer Science
Faculty of Science and Technology
University of Tartu
anti.ingel@ut.ee

Ilya Kuzovkin
Computational Neuroscience Lab
Institute of Computer Science
University of Tartu
ilya.kuzovkin@gmail.com

ABSTRACT

In recent years, developing a direct communication channel between the brain and an external device has received much attention. One method to achieve this is to evoke brain potential called steady-state visual evoked potential (SSVEP) in a user while recording his brain activity with electroencephalography (EEG) device and then use a feature extraction method to extract information related to the potential from the EEG recording. The common way to send a command to the device is to use a set of rules that use the extracted features to classify the command that the user wishes to send.

In this project two feature extraction methods are combined in one BCI—power spectral density analysis (PSDA) and canonical correlation analysis (CCA). These methods are combined using linear discriminant analysis (LDA) in attempt to improve the performance of the BCI. Also, random forest with Platt's probability calibration is slightly tested as classifier. Using multiple feature extraction methods should provide more information about the brain signal and thus make it possible to implement more accurate communication channel.

The implemented BCI had 90% accuracy with 3 commands and target detection time of 4.3 seconds. These results were obtained using consumer-grade EEG device Emotiv EPOC. The proposed methods still need more thorough testing since the BCI was only tested on one subject.

CCS Concepts

• **Mathematics of computing** → *Mathematical software;*

Keywords

SSVEP; BCI; Canonical correlation analysis (CCA); Power spectral density analysis (PSDA); Emotiv EPOC; Linear discriminant analysis (LDA)

DOI: 10.1145/1235

1. INTRODUCTION

Direct communication channel between the brain and an external device is called a brain-computer interface (BCI). Developing a BCI is a well-known problem in neuroscience and many different types of BCIs have been developed over the years. The uses of BCIs include controlling electric wheelchairs and other devices that can increase the quality of life of disabled people [5] and it has been shown that BCIs can be used to assess cognitive functions in coma patients [6].

The BCI used in this project is based on one of the authors previous work. It uses visually evoked potentials, meaning that the user is shown visual stimuli and then his brain activity is analysed to see whether he is looking at a certain stimulus or not. More specifically, in this project steady-state visual evoked potential (SSVEP) is evoked in the user and a consumer-grade electroencephalography (EEG) device is used to record his brain activity.

SSVEP is a continuous brain potential that can be evoked by a visual stimulus that flickers with a fixed frequency. The visual stimuli in BCI are called targets. The advantages of SSVEP are that it is continuous and it contains a frequency component that has the same frequency as the flickering frequency of the target. In addition to the component with the flickering frequency, SSVEP also contains components with the same frequencies as the first few harmonics of the flickering frequency [9]. Although these properties make the detection of SSVEP easier, it is still a brain potential and thus very noisy in nature.

To create a BCI system with multiple commands, the user is shown multiple targets at the same time and each target corresponds to a certain command that the user can send to a computer. Each target flickers with different frequency so it is possible to distinguish the brain responses to different targets. The user decides which command he wants to send and focuses his attention on the corresponding target.

This results in the increase of the amount of the corresponding flickering frequency in the brain signal and by using feature extraction methods, it is possible to extract information about these frequencies and determine which target is the user looking at. Typically used feature extraction methods are power spectral density analysis (PSDA) [2] and canonical correlation analysis (CCA) [7].

The most common approach is to use one feature extraction method and then to define a set of rules according to which the final decision about which target the user is looking at is made. This process of making the decision is called target identification or classification. There have also been attempts to use machine learning for classification with one or multiple feature extraction method but in these articles either online experiment was not performed or the BCI had low performance [4, 8]. The aim of this project is to test further the online performance of a SSVEP-based BCI with multiple feature extraction methods.

2. COMBINING FEATURE EXTRACTION METHODS

2.1 Feature Extraction Methods

This section gives an overview of the feature extraction methods used in the conducted experiments.

2.1.1 Power Spectral Density Analysis

Power spectral density analysis is a common method used to determine the amount of different frequencies present in a signal. It is based on discrete Fourier transform which decomposes a signal to a set of sines and cosines or equivalently to a set of complex exponentials. If $f(t)$ is the actual signal and $f_N(t)$, $t \in \{0, 1, \ldots, N-1\}$ represents the recording of the signal, then

$$f_N(t) = \sum_{k=-N}^{N} c_k e^{2\pi ikt} \tag{1}$$

which approximates the actual signal $f(t)$ and the approximation improves as $N \to \infty$.

Discrete Fourier transform finds the coefficients c_n of the linear combination (1) by the following formula

$$c_n = \sum_{n=0}^{N-1} f_N(n) e^{-2\pi ikn/N}$$

which can be represented as a matrix multiplication

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_M \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & W & W^2 & \ldots & W^{N-1} \\ 1 & W^2 & W^4 & \ldots & W^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W^{N-1} & W^{N-2} & \ldots & W \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ f(2) \\ \vdots \\ f(M) \end{pmatrix}$$

where $M = N - 1$ and $W = e^{-2\pi i/N}$. An algorithm for performing this multiplication with $O(N \log N)$ complexity is called fast Fourier transform.

To be able to compare the amount of different frequencies present in the signal $f(t)$, one can find the amplitudes of the complex exponentials by taking the magnitude of the complex number c_n. Thus $|c_n|$ is the amplitude of the frequency

$$n \cdot \frac{f_s}{N} \tag{2}$$

where $n \in \{0, 1, \ldots, \lfloor N-1 \rfloor/2\}$ and f_s is the sampling frequency in Hz. The frequencies obtained by formula (2) are called frequency bins. By squaring the magnitudes one gets a periodogram, an estimation for the power spectrum of signal $f(t)$, and hence the name power spectral density analysis.

The final aspect to consider in this method is that targets with higher frequency produce SSVEPs with smaller amplitude and that there is a lot of noise present in the EEG recording, especially when using a consumer-grade device. Thus it might be more beneficial to compare signal-to-noise ratios (SNRs) instead of the amplitudes or the powers themselves for example by using formula [1]

$$SNR(f_t) = \frac{2P(f_t)}{P(f_{t-1}) + P(f_{t+1})}$$

or more generally [10]

$$SNR(f_t) = \frac{nP(f_t)}{\sum_{i=1}^{n/2} P(f_{t-i}) + P(f_{t+1})}$$

where P is the periodogram, $f_1, \ldots, f_t, \ldots, f_N$ represent the frequency bins in increasing order, f_t is the target flickering frequency and n is the number of adjacent frequency bins used in the calculation of SNR.

More recent feature extraction method called minimum energy combination uses spatial filtering to try to overcome the weaknesses of PSDA.

2.1.2 Canonical Correlation analysis

Canonical correlation analysis feature extraction method has been one of the most applied methods in SSVEP-based BCIs and it has been shown that using it gives very good results, even with a consumer-grade EEG device.

The CCA itself is a way to calculate the correlation between two sets of random variables. It was introduced by Harold Hotelling in 1936 [3].

In SSVEP-based BCI setting, one set of random variables is the multichannel EEG recording and the other is a set of reference signals. Each target has its own set of reference signals based on the flickering frequency of the target. The reference signals are sine and cosine waves with the same frequency as the target flickering frequency and also some of its harmonics. Usually two or three harmonics are used. For a target m with frequency f_m the set of reference signals is

$$Y_m = \begin{pmatrix} \sin(2\pi \cdot f_m \cdot t) \\ \cos(2\pi \cdot f_m \cdot t) \\ \vdots \\ \sin(2\pi \cdot N_h \cdot f_m \cdot t) \\ \cos(2\pi \cdot N_h \cdot f_m \cdot t) \end{pmatrix}, \quad t = \frac{1}{f_s}, \frac{2}{f_s}, \ldots, \frac{N}{f_s}$$

where f_s is the sampling frequency, N_h is the number of harmonics used and N is the length of the recorded signal.

Similarly the multichannel EEG recording can be organised into matrix X where each row contains signal from one channel.

CCA finds two new sets of features whereas the features of one set are linear combinations of the rows of X and the features of the other set are linear combinations of the rows of Y_m. Thus given a matrices of weights W_X and W_{Y_m} the new features are $W_X X$ and $W_{Y_m} Y_m$.

The new features are chosen so that the corresponding rows of $W_X X$ and $W_{Y_m} Y_m$ are maximally correlated, meaning that the first row of $W_X X$ is maximally correlated with the first row of $W_{Y_m} Y_m$ and similarly with other rows. The first rows are called the first pair of canonical variates. The subsequent rows have additional constraint that they have to be uncorrelated with the previous canonical variates. The rows of $W_X X$ themselves have maximal possible variance and similarly the rows of $W_{Y_m} Y_m$.

In SSVEP-based BCI only the first pair of canonical variates is used. The problem of finding the first pair of canonical variates can be formulated as a maximisation task

$$\max_{\vec{a},\vec{b}} \mathbf{Cor}(\vec{a}^T X, \vec{b}^T Y_m) = \frac{\vec{a}^T X Y_m^T \vec{b}}{\sqrt{\vec{a}^T X X^T \vec{a} \vec{b}^T Y_m Y_m^T \vec{b}}} \qquad (3)$$

where $\vec{a}$ is a column vector representing the first row of $W_X X$ and $\vec{b}$ is a column vector representing the first row of $W_{Y_m} Y_m$.

To identify the target that the user is looking at, one can compare the correlations obtained by formula (3) for different targets and the target whose set of reference signals has the highest canonical correlation with corresponding $\vec{a}^T X$ can be considered to be the target the user is looking at.

The advantage of this method over PSDA is that it is inherently multidimensional which is not the case with PSDA. Since the EEG recording is multidimensional, with CCA method it is possible to take into account all the information from the recording at once.

2.1.3 Implementation Details

When using a BCI in real-time, it is required that the feature extraction can be done in a relatively short time period. Thus in this work the linearity of discrete Fourier transform was used to reduce the calculation time—instead of calculating periodogram for each of the EEG channels, the signals from each EEG channels were summed together and then the periodogram was calculated from the sum of the signals. Since discrete Fourier transform is linear, the amplitude or power spectrum of the summed signal is the same as the amplitudes or powers of each channel calculated separately and then summed together. Furthermore, the BCI also offers the possibility to calculate the power spectrums separately for each channel.

Since flickering targets do not only elicit response in the brain with the same frequency as the flickering itself but also with the frequencies of its harmonics, then also powers of the harmonics are calculated. As with CCA, either two or three harmonics for each target were used.

And finally, in this BCI only the powers themselves were used as features, not SNR.

2.2 Target Identification

As already mentioned, the common approach to identify the target the user is looking at is to define a set of rules according to which the final decision of what target to choose is made, but this work has a different approach. Here two feature extraction methods are used at the same time and the final decision is made by using machine learning. For this classification task, multiclass linear discriminant analysis (LDA) was used in online experiments. Furthermore, LDA was compared with random forest in offline experiments.

2.2.1 Multiclass Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a well-known dimensionality reduction and classification method. LDA finds a projection of the input data to a lower dimensional space so that the variance between the classes is maximised with respect to the variance within classes. The relationship between the total variance, variance within classes W and variance between classes B can be summarised by the law of total variance

$$\mathbf{Var}(\vec{X}) = \underbrace{\mathbf{E}_Y(\mathbf{Var}(\vec{X} \mid Y))}_{W} + \underbrace{\mathbf{Var}_Y(\mathbf{E}(\vec{X} \mid Y))}_{B}$$

where $\vec{X} = (X_1, \ldots, X_n)^T$ is the feature vector and Y is the class label which takes values from $h = 1, 2, \ldots, H$.

LDA makes the assumption that the covariance matrices Σ_h of different classes are equal, that is

$$\Sigma_h = \mathbf{Var}(\vec{X} \mid Y = h) = \mathbf{E}_Y(\mathbf{Var}(\vec{X} \mid Y)) = W$$

for all classes $h = 1, 2, \ldots, H$.

The maximisation task that LDA solves is as follows, given the features vector $\vec{X}$ and class label Y, find a linear combination of $\vec{X}$ elements $\vec{a}^T \vec{X}$ for which the variance between the classes is maximised with respect to variance within the classes

$$\max_{\vec{a}} \frac{\mathbf{Var}_Y(\mathbf{E}(\vec{a}^T \vec{X} \mid Y))}{\mathbf{E}_Y(\mathbf{Var}(\vec{a}^T \vec{X} \mid Y))} = \max_{\vec{a}} \frac{\vec{a}^T B \vec{a}}{\vec{a}^T W \vec{a}}. \qquad (4)$$

This is generalised eigenvalue problem for which the solution is the eigenvector of $W^{-1}B$ corresponding to the largest eigenvalue of $W^{-1}B$. The second eigenvector is orthogonal to the first, while still making the ratio (4) as large as possible while satisfying orthogonality constraint. And similarly for each next eigenvector.

Therefore eigenvectors are orthogonal to each other and are taken to be the basis of the lower dimensional space where between class variance is as large as possible with respect to within class covariance.

Finally to classify feature vector $\vec{x}$ one has to project vector $\vec{x}$ and class means of the input data to the lower dimensional space and to find which of the class means is closest to the sample $\vec{x}$. Sample $\vec{x}$ is classified to the class whose mean is closest to it.

In practice, the matrices W and B and class means can be estimated using known formulas. Given n observations x_i, $i = 1, 2, \ldots, n$ and their classes y_i whereas number of observations from class h is n_h the estimations are

$$W = \sum_{h=1}^{H} \pi_h \Sigma_h, \qquad S = \sum_{h=1}^{H} \pi_h (\vec{\mu}_h - \vec{\mu})(\vec{\mu}_h - \vec{\mu})^T,$$

$$\vec{\mu} = \sum_{h=1}^{H} \pi_h \vec{\mu}_h, \qquad \vec{\mu}_h = \frac{1}{n_h} \sum_{\substack{a \in [n] \\ y_i = h}} \vec{x}_i, \qquad \pi_h = \frac{n_h}{n}.$$

where $\vec{\mu}$ is the overall mean, $\vec{\mu}_h$ is the class mean and π_h is the probability of the class.

2.2.2 Implementation Details

Since the traditional multiclass LDA prediction did not work very well for the task of classifying user's chosen targets, a different approach was used in this work. First, a training dataset is used to find a dimensionality reduction transformation. Then a validation dataset is used to calculate a receiver-operator characteristic (ROC) curve for classifying each target using the signed distance to the decision border as a feature. Thus a ROC curve for each target is obtained. From these ROC curves a cutoff threshold c_h for classification is chosen for each class $h = 1, 2, \ldots, H$.

Then the classification itself is performed as follows: Given an observation $\vec{x}$ it is transformed to the lower dimensional

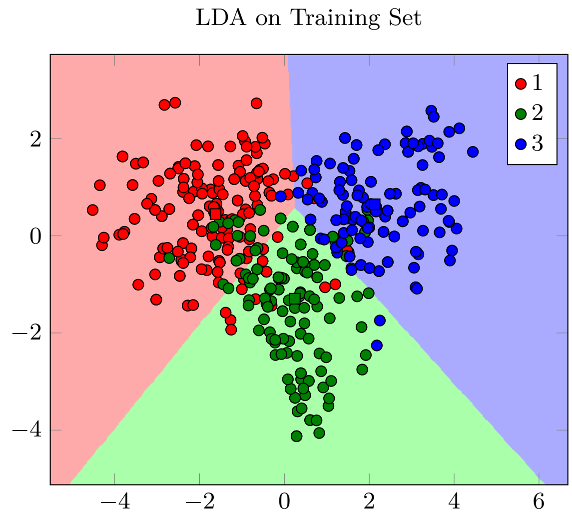

Figure 1: LDA dimensionality reduction on training set.

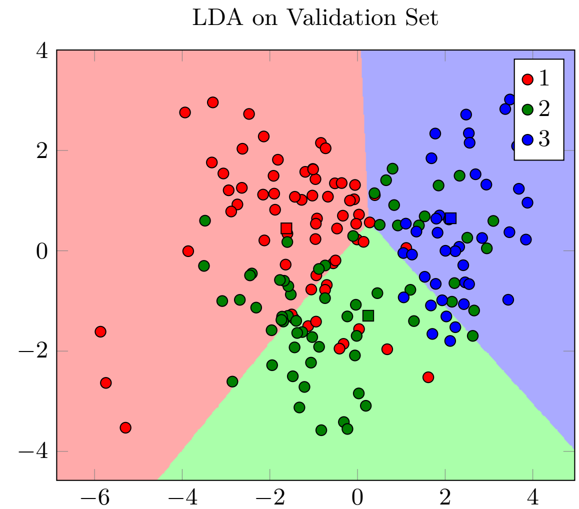

Figure 2: LDA dimensionality reduction on validation set.

space and the signed distance to each decision border d_h in this lower dimensional space is calculated. If there exists a class h for which the signed distance to the decision border is smaller than the cutoff threshold $d_h \leq c_h$ while for all the other classes it is larger $d_{h^*} > c_{h^*}$ for all $h^* \in \{1, 2, \ldots, H\} \setminus h$ then the observation is classified to class h. Otherwise no classification is made.

Another approach that was briefly tested was using probability calibration on random forest. Method called Platt's scaling was used for the calibration.

2.3 Online Experiment

The EEG device used for measuring brain activity was a consumer-grade device Emotiv EPOC. Emotiv EPOC has sampling frequency of 128 Hz and 14 fixed electrode locations. In the experiment only electrodes O1 and O2 (according to 10-20 electrode placement system) were used since these are the closest to the visual processing centre of the brain. No separate channel selection was made for different subjects, for each subject the same electrode locations were used.

For eliciting SSVEPs, 14" LED-backlit LCD laptop monitor with 60 Hz refresh rate and 1600x900 resolution was used. Targets frequencies in a range from 6 Hz to 12 Hz were used and the number of targets varied between 3 and 6 in different trials. The targets were white squares flickering on a black background. For each subject the same frequencies, shape, colour and type were used, no separate analysis on what kind of targets produce better SSVEP in different subjects was made.

Before it is possible to use the BCI one has to record a few minutes of training data, the exact length of training data depends on how many targets are used. After that a model can be trained using the training data and the BCI is ready to be used.

The BCI was mainy tested on one subject. A subject was sitting on a comfortable chair with laptop in front of him and the data was recorded while the subject was using the BCI online.

3. RESULTS

This section gives an overview of the results obtained from one subject. Further testing did not fit into the scope of this project. In figure 1 the LDA dimensionality reduction of subject's 2 minute long recording session can be seen. In the session 3 targets were used with frequencies 6.67 Hz, 7.5 Hz and 8.57 Hz, thus the lower dimensional space has 2 dimensions. In figure 2 it can be seen how the dimensionality reduction works with unseen data.

The ROC curves of the same datasets can be seen in figures 3 and 5 respectively. Binary classifiers were obtained by using the distance to the decision borders as features.

The LDA dimensionality reduction and ROC curves show the overall performance. The changes of features over time can be seen in figures 7 and 8. Each subplot in these figures shows how the distance to the decision border of the corresponding class changes over time for subsequent samples. The dashed line shows what is the current expected target—if the dashed line is up, then the user should be looking at the target corresponding to this class.

Very steep changes in plots in figures 7 and 8 when the expected target changes are due to the fact that the first 256 samples that are recorded right after the expected target changes are discarded. This is to avoid selecting one target too many times in a row when using the BCI online.

To compare the performance of LDA method to the method using random forest with Platt's probability calibration, the ROC curves for random forest method are shown in figures 4 and 6. How the calibrated probability changes over time can be seen in figures 9 and 10.

Finally, the table 1 shows the results of the online experiment for one subject using traditional BCI performance measures—accuracy, average target detection time and information transfer rate (ITR) in bits/min. The first row shows the results for the same model that was already discussed, but this time with new data. Look-back length is the number of previous samples used by the classification, it shows how long into the past the classifier looks. Window length shows how long signal segment is used by the feature extraction methods.

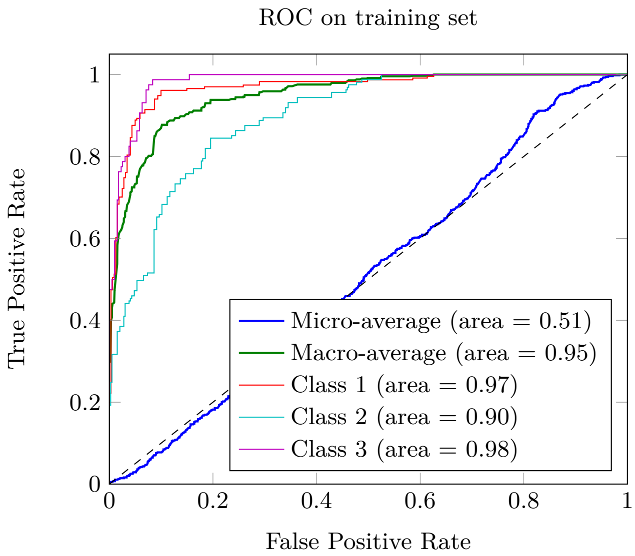

Figure 3: LDA ROC on training set.

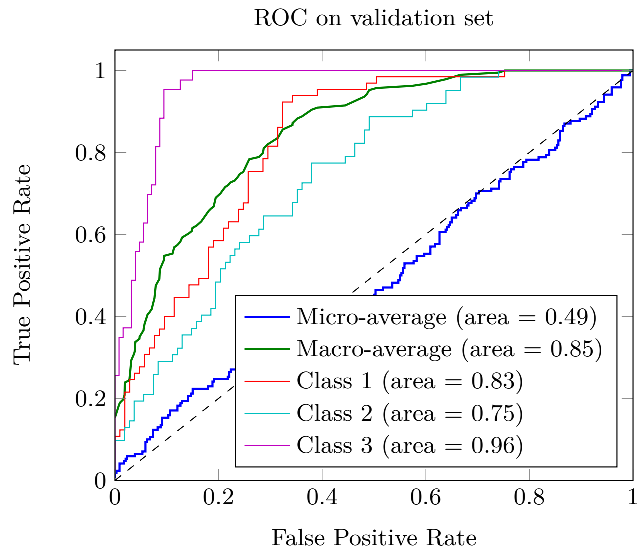

Figure 5: LDA ROC on validation set.

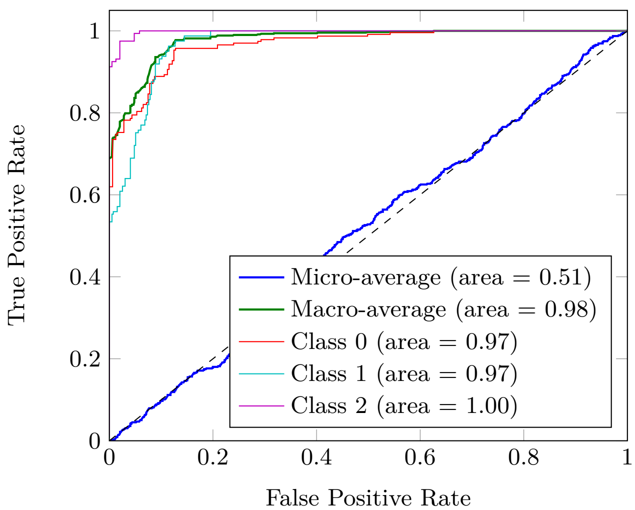

Figure 4: Random forest ROC on training set.

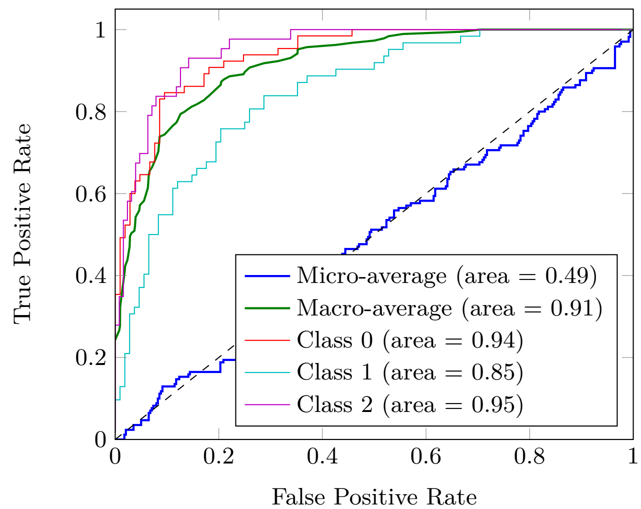

Figure 6: Random forest ROC on training set.

4. CONCLUSIONS

The aim of this project has been to combine two commonly used SSVEP-based BCI feature extraction methods in a BCI to improve its performance and test it online. The most common approach in the literature so far has been to use one feature extraction method and a set of rules according to which the classification is made. In this project PSDA and CCA feature extraction methods were combined using LDA dimensionality reduction and its decision borders. Also random forest with probability calibration was also slightly tested as classifier for the BCI.

The online results indicate that the BCI performs better if used with fewer targets. From table 1 it can be seen that the more targets were used, the lower the accuracy and ITR. With 3 targets the BCI obtains a very good accuracy of 90% but due to the fact that the LDA requires information about the features from at least 10 previous time steps (look-back length in table 1), the target detection time tends to be rather large and this decreases the ITR.

From the offline results show that random forest with Platt's probability calibration performs better than the proposed method based on LDA, but it would need further online testing to confirm this.

In order to use BCIs in real-world applications, they still need improvements. The high accuracy obtained in this project, however, is one of the most important properties that a BCI should have in order to use it in everyday life. But since the results presented here were obtained from one subject only, the BCI definitely needs more testing, most importantly to see how well it performs on other subjects.

5. ACKNOWLEDGMENTS

Working on this project was supported by IT Academy.

6. REFERENCES

[1] G. Bin, X. Gao, Y. Wang, B. Hong, and S. Gao. VEP-based brain-computer interfaces: Time, frequency, and code modulations. *Computational Intelligence Magazine, IEEE*, 4(4):22–26, 2009.

[2] M. Cheng, X. Gao, S. Gao, and D. Xu. Design and implementation of a brain-computer interface with high transfer rates. *Biomedical Engineering, IEEE Transactions on*, 49(10):1181–1186, 2002.

[3] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321—377, 1936.

[4] H. L. Jian and K. T. Tang. Improving classification accuracy of ssvep based bci using rbf svm with signal quality evaluation. In *Intelligent Signal Processing and Communication Systems (ISPACS), 2014 International Symposium on*, pages 302–306, Dec 2014.

[5] T. Kaufmann, A. Herweg, and A. Kübler. Toward brain-computer interface based wheelchair control utilizing tactually-evoked event-related potentials. *Journal of NeuroEngineering and Rehabilitation*, 11(1):7, 2014.

[6] Y. Li, J. Pan, Y. He, F. Wang, S. Laureys, Q. Xie, and R. Yu. Detecting number processing and mental calculation in patients with disorders of consciousness using a hybrid brain-computer interface system. *BMC Neurology*, 15(1):259, 2015.

[7] Z. Lin, C. Zhang, W. Wu, and X. Gao. Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *Biomedical Engineering*, 54(6), 2007.

[8] I. Martišius and R. Damaševičius. A Prototype SSVEP Based Real Time BCI Gaming System. *Computational Intelligence and Neuroscience*, 2016.

[9] F. Teng, Y. Chen, A. M. Choong, S. Gustafson, C. Reichley, P. Lawhead, and D. Waddell1. Square or sine: Finding a waveform with high success rate of eliciting SSVEP. *Computational Intelligence and Neuroscience*, 2011, 2011.

[10] W. Yijun, W. Ruiping, G. Xiaorong, and G. Shangkai. Brain-computer interface based on the high-frequency steady-state visual evoked potential. *First International Conference on Neural Interface and Control Proceedings*, 2005:37–39, 2005.

APPENDIX

A. THE BCI CODE REPOSITORY

The code of the BCI which implements the proposed methods and was used for testing in this project is accessible from Github repository[1].

[1]https://github.com/kahvel/VEP-BCI

Table 1: Online Experiment results

Accuracy	Detection time	ITR	Targets	Trial length	Look-back length	Window length	Short signal	Margin	
0.91	4.96 s	12.76	3	109 s	10	256	No	0	1/1
0.74	5.03 s	5.94	3	136 s	8	256	No	0	1/1
0.86	4.56 s	11.56	3	169 s	10	128	No	0	1/1
0.95	3.77 s	20.07	3	158 s	10	128	No	0	1/1
0.87	4.01 s	13.58	3	220 s	10	128	No	0	1/1
0.76	2.95 s	16.70	4	171 s	10	256	Yes	0.1	1/1
0.72	3.64 s	11.92	4	80 s	10	256	Yes	0.2	2/2
0.58	4.05 s	5.41	4	117 s	10	256	Yes	0.2	2/2
0.77	2.93 s	17.62	4	141 s	10	256	Yes	0.2	1/1
0.70	4.36 s	8.68	4	100 s	10	256	Yes	0.3	2/2
0.91	3.61 s	23.83	4	166 s	10	256	Yes	0.3	2/2
0.82	4.40 s	14.55	4	154 s	10	256	Yes	0.3	2/2
0.64	3.73 s	7.83	4	186 s	10	256	Yes	0.3	2/2
0.79	5.54 s	10.15	4	161 s	10	256	Yes	0.3	2/2
0.81	3.87 s	15.65	4	229 s	10	256	Yes	0.3	2/2
0.86	3.30 s	22.02	4	145 s	10	256	Yes	0.3	1/1
0.81	3.37 s	17.76	4	196 s	10	256	Yes	0.4	1/1
0.65	3.23 s	9.59	4	149 s	10	256	Yes	0	1/1
0.64	4.53 s	6.41	4	163 s	10	256	Yes	0.3	2/2
0.65	4.06 s	7.5	4	150 s	10	256	Yes	0.3	2/2
0.13	4.55 s	0.06	6	132 s	2	256	No	0	1/1
0.47	3.71 s	5.73	6	119 s	5	256	No	0	1/1
0.42	4.27 s	3.69	6	111 s	10	256	No	0	1/1
0.59	7.02 s	5.57	6	119 s	10	256	No	0	1/1
0.36	4.59 s	1.99	6	128 s	12	256	No	0	1/1
0.62	4.97 s	8.91	6	169 s	10	384	No	0	1/1
0.32	4.27 s	1.51	6	158 s	10	512	No	0	1/1
0.40	11.50 s	1.15	6	115 s	10	256	No	0	1/1
0.80	7.59 s	11.06	6	303 s	10	256	No	0	2/2
0.56	8.45 s	4.12	6	135 s	10	256	No	0	2/3
0.49	6.40 s	3.67	6	224 s	10	256	No	0	2/3
0.81	8.20 s	10.54	6	172 s	10	256	No	0	3/3
0.70	5.71 s	10.59	6	285 s	10	256	No	0	1/1
0.62	6.17 s	7.22	6	129 s	10	256	No	0	1/1
0.58	7.50 s	5.01	6	142 s	10	256	No	0	1/1
0.78	4.89 s	16.02	6	264 s	10	256	No	0	1/1
0.78	6.27 s	12.49	6	226 s	10	256	No	0	1/1
0.35	4.99 s	1.70	6	100 s	1	256	No	0	1/1
0.44	4.92 s	3.71	6	265 s	5	256	No	0	1/1
0.30	4.23 s	1.18	6	97 s	10	256	No	0	1/1
0.53	6.45 s	4.65	6	206 s	10	256	No	0	1/1
0.43	4.64 s	3.53	6	227 s	10	384	No	0	1/1
0.15	6.64 s	0.01	6	86 s	1	512	No	0	1/1
0.27	4.97 s	0.55	6	149 s	2	512	No	0	1/1
0.33	4.45 s	1.60	6	160 s	5	512	No	0	1/1
0.49	5.78 s	4.10	6	225 s	10	512	No	0	1/1
0.36	4.66 s	2.08	6	154 s	10	512	No	0	1/1

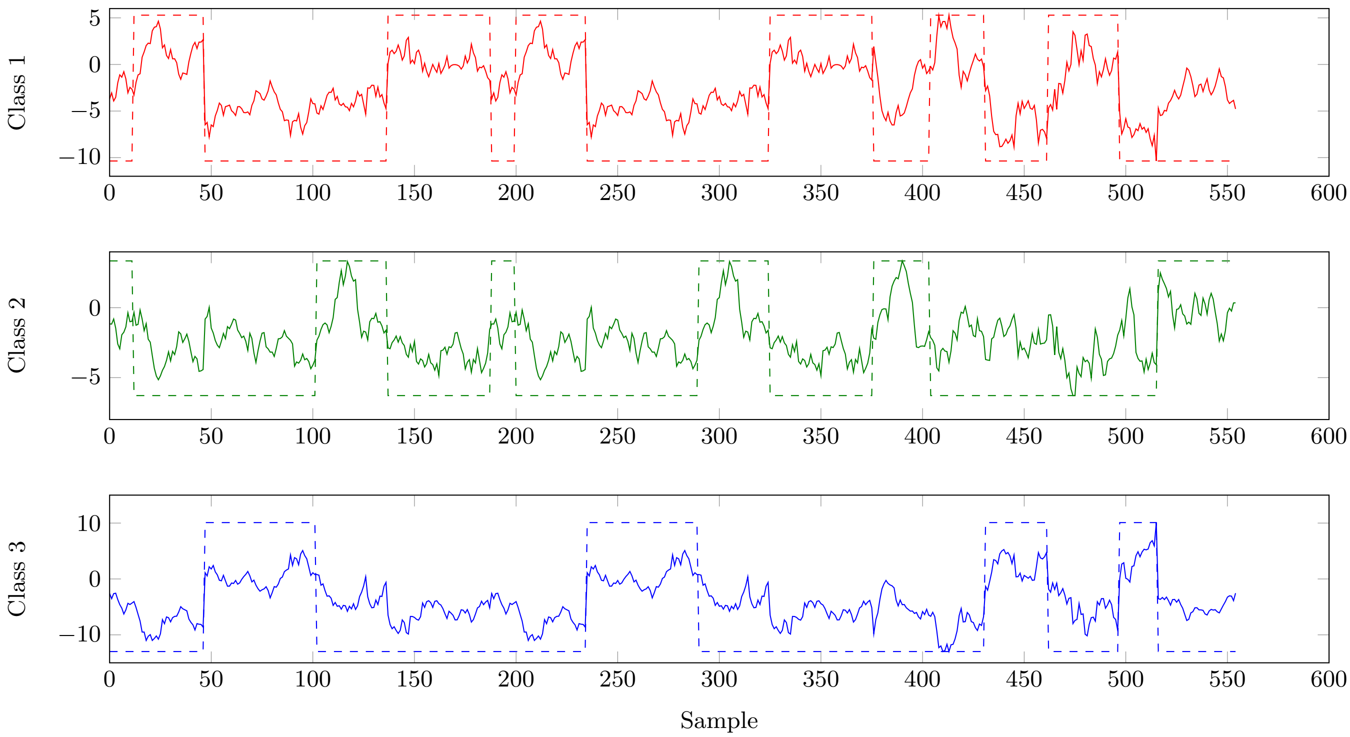

Figure 7: Changes in LDA features (distance to corresponding decision border) over time on training data. When dashed line is up, it means that user should be looking at this target.

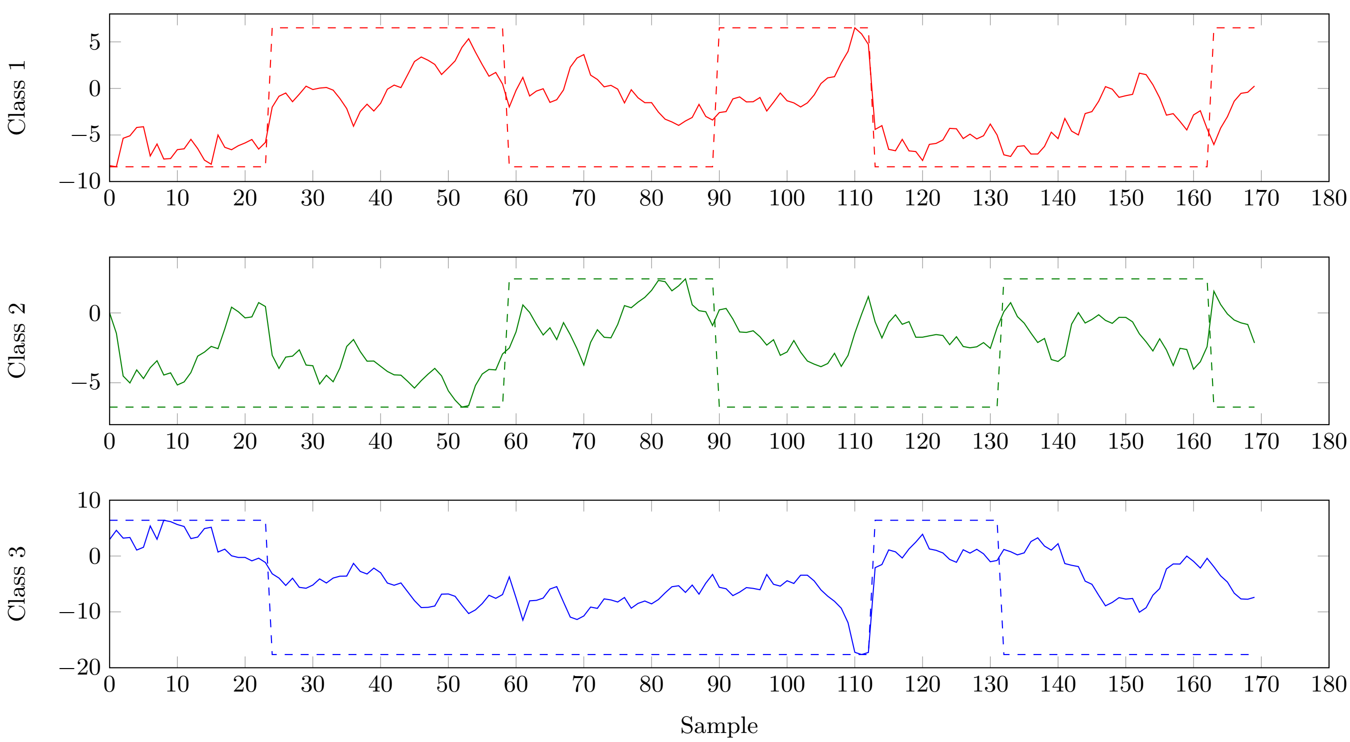

Figure 8: Changes in LDA features (distance to corresponding decision border) over time on validation data. When dashed line is up, it means that user should be looking at this target.

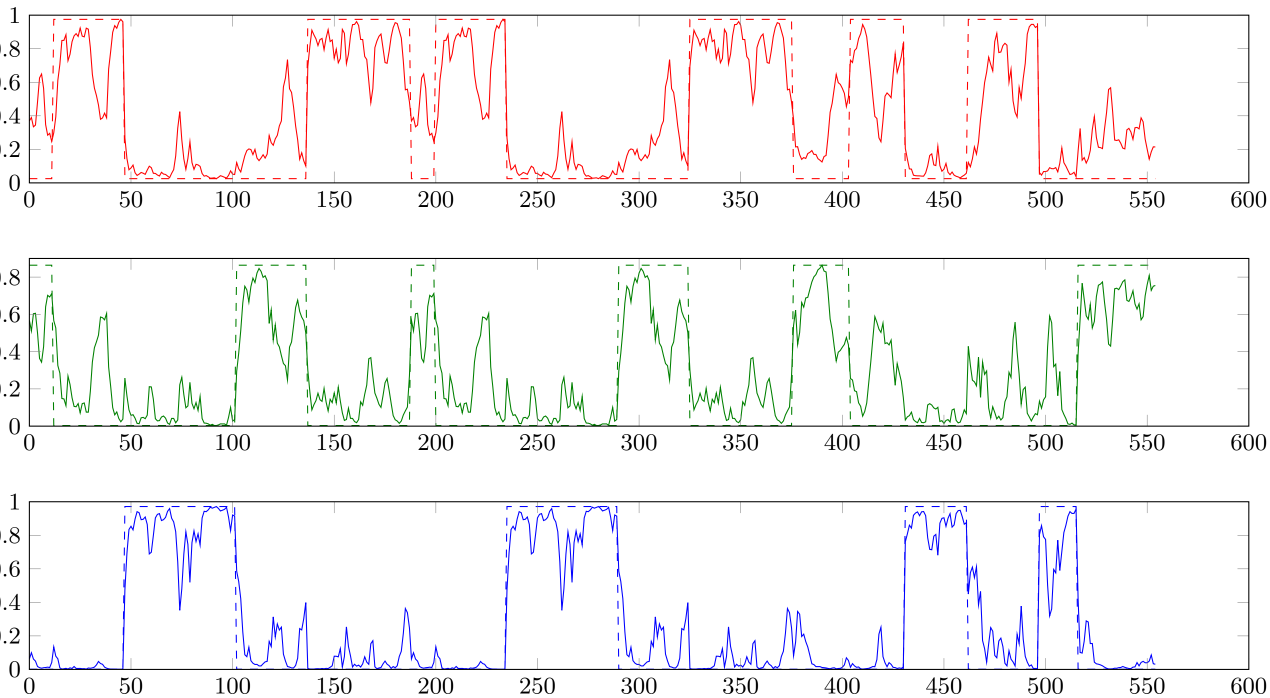

Figure 9: Changes in random forest features (calibrated probability) over time on training data. When dashed line is up, it means that user should be looking at this target.

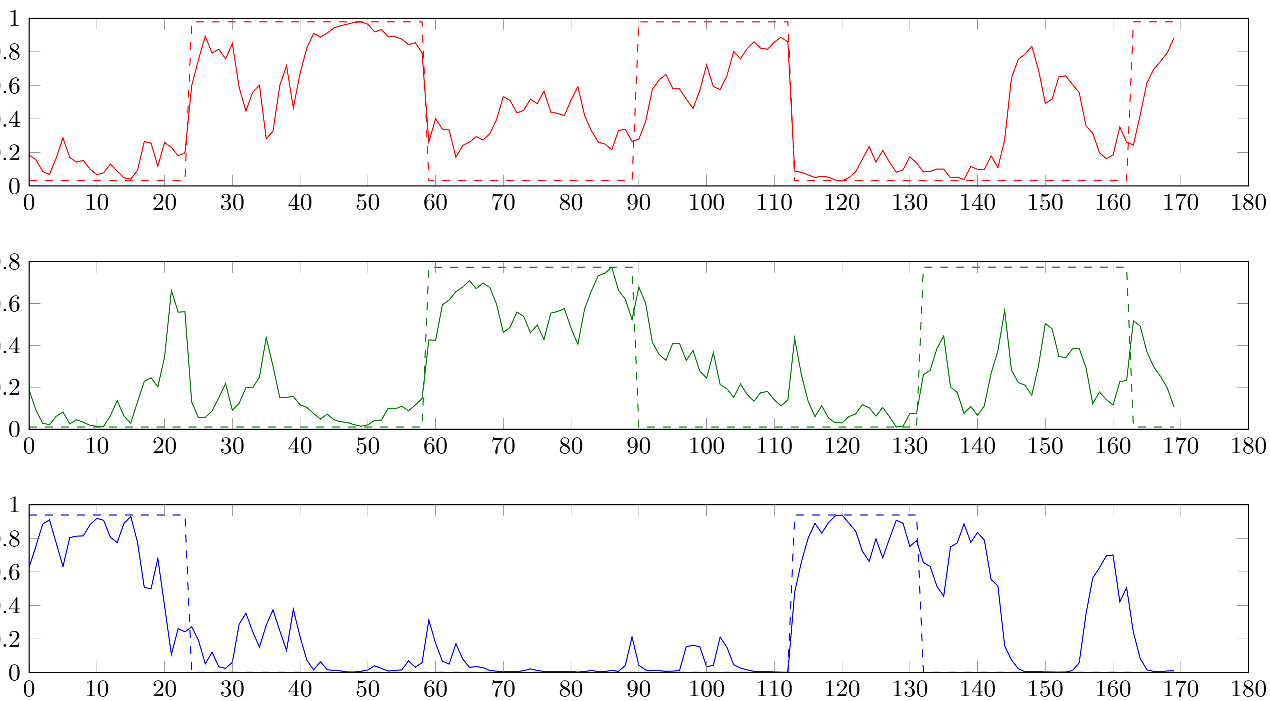

Figure 10: Changes in random forest features (calibrated probability) over time on validation data. When dashed line is up, it means that user should be looking at this target.