Classifier Selection for Motor Imagery Brain Computer Interface

Izabela Rejer, Robert Burduk

Classifier Selection for Motor Imagery Brain Computer Interface

Izabela Rejer[1], Robert Burduk[2]

[1] West Pomeranian University of Technology,
Faculty of Computer Science and Information Technology,
Żołnierska Str. 52, 71–210 Szczecin, Poland
irejer@wi.zut.edu.pl

[2] Department of Systems and Computer Networks,
Wroclaw University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
robert.burduk@pwr.edu.pl

Abstract. The classification process in the domain of brain computer interfaces (BCI) is usually carried out with simple linear classifiers, like LDA or SVM. Non-linear classifiers rarely provide a sufficient increase in the classification accuracy to use them in BCI. However, there is one more type of classifiers that could be taken into consideration when looking for a way to increase the accuracy - boosting classifiers. These classification algorithms are not common in BCI practice, but they proved to be very efficient in other applications.

Keywords: Imagery Brain Computer Interface, Classification, Boosting

1 Introduction

A brain-computer interface (BCI) is a control and communication system in which the control commands and messages are not transmitted via the standard outputs of a central nervous system, but are read directly from the users brain. Nowadays there are a lot of different devices for recording the brain activity, however, due to relatively low costs, high mobility, and non-invasiveness, EEG devices are usually used for outside-lab BCIs. There are three main types of EEG-BCIs: SSVEP-BCI, P300-BCI, and MI-BCI. In each of them, different brain potentials are used to control the interface.

SSVEP-BCI is controlled by steady state visually evoked potentials (SSVEPs). These potentials are recorded from the occipital cortex when a user is exposed to a visual stimulus flickering with a steady frequency. Since the stimulus fundamental frequency (and also the harmonics) can be observed in EEG recording, different control commands are encoded with stimuli of different frequency, each delivered by different stimuli providers (usually LEDs).

The control signals in P300-BCI are positive potentials that appear over the parietal cortex when a user perceives a rare and significant stimulus. The P300 potential can be detected in the brain activity approximately 300 ms after the stimulus is presented. The control process in P300-BCI is based on picking out the objects from a set of objects displayed on the screen [3]. The users task is to focus the attention on one of the objects. The objects are highlighted randomly each time the object chosen by the user is highlighted, P300 potential appears over the parietal cortex.

While the brain potentials used in SSVEP-BCI and P300-BCI are evoked potentials (both need external stimulation to be evoked in the users brain), MI-BCI is based on spontaneous potentials, evoked by the user. The potentials used for controlling MI-BCI, called motor rhythms, can be detected over the motor cortex when the user performs real or imagery movements.

Out of these three types of BCI, MI-BCI is the most natural and so the most welcome in practical applications. Its main benefit is that it does not require any external stimulation - the mental states needed to perform the actions are evoked only at the users will. This means that the user can use the interface at any time while performing other actions (e.g. reading a text). Hence, the interface can be always in the stand-by state waiting for a users command. In theory P300-BCI or SSVEP-BCI could be also constantly on, but in practice flickering light or highlighting objects constantly present at the users field of view would be extremely tiring. However, there is no free lunch, MI-BCI is the most convenient for the user, but it provides the smallest number of control states. Usually only 2-4 direct commands can be obtained with this type of interface (corresponding to the motor imagery of: left and right hand, feet, and tongue). Moreover, the potentials related to motor imagery are difficult to extract from a scalp EEG, especially in the case of a beginner user.

In fact, MI-BCI can be successfully used in practice, but the user has to learn first how to perform the motor imagery at all, and then how to perform it effectively. As Pfurtscheller et al. and Guger et al. report in [7], [8], [15] even 80-97% of classification accuracy can be obtained after 6–10 twenty-minute sessions. If the training is shorter, the results are not so impressive. In [5] Gauger et al report that only in 20% of 99 subjects the brain patterns related to left-right hand motor imagery were possible to be distinguished with an accuracy greater than 80% after 20-30 minutes of training. For 70% of 99 subjects the classification accuracy was about 60-80%. In the case of remaining subjects, the classification accuracy of motor imagery of left and right hand was below 60%.

In MI-BCI and other biometric systems, simple individual classifiers are usually used in the classification process, e.g. linear discriminant analysis (LDA) [6], Fisher linear discriminant analysis (FDA) [13], k-NN classifier [17], [16], distinction sensitive learning vector quantization classifier (DSLVQ) [15] or minimum Mahalanobis distance (MDA) classifier [14]. The boosting algorithms are also an effective method of producing a very accurate classification rule [9], however, they are rarely used in BCI domain [12]. In general, they are a combination of so-called weak classifiers. A weak classifier learns on various training examples sampled from the original learning set. The sampling procedure is based on the weight of each example. In each iteration, the weights of examples are changed.

The final decision of the boosting algorithm is determined on the ensemble of classifiers derived from each iteration of the algorithm.

The aim of this paper is to compare the performance of four boosting classifiers (AdaBoost, RealAdaBoost, GentleAdaBoost, and our modification of AdaBoost formulated in Section 2) and three classic classification algorithms: LDA, SVM, and k-NN. We would like to find out whether the boosting scheme, where the weak classifiers are joined to create a strong one, will provide at least similar results as the methods that proved to be very successful in BCI field. We perform our analysis over EEG data that were acquired during a MI-BCI session.

The rest of this paper is organized as follows. In Section 2 our modification of AdaBoost algorithm is presented. The experimental evaluation, discussion and conclusions from the experiments are presented in Section 3. Finally, some conclusions are given.

2 AdaBoost algorithm

In recent years many modifications of algorithms based on the boosting idea appeared. For example, AdaBoost.OC algorithm [20] is a combination of AdaBoost method and ECOC model, LogitBoost algorithm [5] utilizes the model of logistic regression, FloatBoost algorithm [11] removes those classifiers in subsequent iterations which do not fulfill the quality condition assumed and in [2] the methods of modifying weights of the examples are described.

It is worth mentioning that in the work [1] a common name for boosting type algorithms was proposed, the acronym ARCing for Adaptive Resampling and Combining was created, and also a new algorithm named ARC-x4 was presented. In this work we use the three well-known boosting algorithms (AdaBoost, RealAdaBoost and GentleAdaBoost), together with our modification of the classic AdaBoost algorithm.

One of the main factors that have an effect on the action of AdaBoost algorithm is the selection of weights assigned to individual elements of the learning set. Let's propose then a modification of AdaBoost algorithm which will represent imprecision in values of weights obtained in subsequent iterations of the algorithm. Such imprecision will be defined by parameters k and λ. The second parameter defines certain linear combination of the upper and lower value of weights obtained after modification by k. The algorithm steps are presented in Tab. 1.

If the values of functions $\overline{e_b} = e_b + k$, $\underline{e_b} = e_b - k$ obtained in point 4c are outside the range of values $[0, 1]$, then the values of these functions should be modified (point 4d). This procedure seems to be used extremely rarely due to the possible values of the error e_b and the assumed values of the k parameter. In the earlier work [2] the changes in weights values depended on the iteration of the algorithm. In this work the changes in weights do not depend on the iteration of the algorithm.

Table 1. Algorithm B1M-AdaBoost

1.	Initialize λ and k.
2.	Initialize the weight vector $w_{1,1} = ... = w_{1,n} = 1/n$.
3.	Assign weights to the learning sample LS_n.
4.	For $b = 1, 2, ..., B$:
a.	At the base of LS_n learn the classifier Ψ_b,
b.	Calculate the classification error $e_b = \sum_{i=1}^{n} w_{b,i} * I(\Psi_b(x_i) \neq j_i)$,
c.	Calculate $\overline{e_b} = e_b + k$, $\underline{e_b} = e_b - k$
d.	If $\overline{e_b} > 1$ then $\overline{e_b} = 1$, If $\underline{e_b} < 0$ then $\underline{e_b} = 0$
e.	Calculate $\overline{c_b} = \frac{\ln(1-\overline{e_b})}{\overline{e_b}}$, $\underline{c_b} = \frac{\ln(1-\underline{e_b})}{\underline{e_b}}$
f.	Update weights:
	$\overline{w_i}(b+1) = \frac{w_{b,i} \exp(\overline{c_b} * I(\Psi_b(x_i) \neq j_i))}{\sum_{j=1}^{n}(w_{b,j} \exp(\overline{c_b} * I(\Psi_b(x_i) \neq j_i)))}, \quad i = 1, ..n,$
	$\underline{w_i}(b+1) = \frac{w_{b,i}) \exp(\underline{c_b} * I(\Psi_b(x_i) \neq j_i))}{\sum_{j=1}^{n}(w_{b,j} \exp(\underline{c_b} * I(\Psi_b(x_i) \neq j_i)))}, \quad i = 1, ..n,$
g.	Calculate $c_b = \lambda * \overline{c_b} + (1 - \lambda)\underline{c_b}$,
h.	Calculate $w_{b+1,i} = \lambda * \overline{w_i}(b+1) + (1-\lambda) * \underline{w_i}(b+1)$,
i.	Assign updated weights to the learning sample LS_n.
5.	Classify observation x according to the rule:
	$\Psi_{IFw-AdaBoost}(x) = sign\left(\sum_{b=1}^{B} c_b \Psi_b(x)\right)$.

3 Experimental studies

The experiment was performed with a male subject, aged 32. The subject was right-handed, had normal vision and did not report any mental disorders. The experiment was conducted according to the Helsinki declaration on proper treatment of human subjects. Written consent was obtained from the subject.

The subject was placed in a comfortable chair and EEG electrodes were applied on his head. In order to limit the number of artifacts, the participant was instructed to stay relaxed and not move. The start of the experiment was announced by a short sound signal. 200 trials were recorded during the experiment. Each trial started with a picture of an arrow pointing to the left or right displayed on a computer screen. The screen was located about 70 cm from the subjects eyes. The task of the subject was to imagine the wrist rotation of the hand indicated by the arrow (arrow to the left left hand; arrow to the right right hand). The arrow directions for the succeeding trials were chosen randomly. There were no breaks between trials. The experiment was divided into 4 sessions, 50 trials each. The trial length was fixed and was equal to 10 seconds. There were 3-minute breaks between sessions.

EEG data was recorded from two monopolar channels at a sampling frequency of 256 Hz. Four passive electrodes were used in the experiments. Two of them were attached to the subjects scalp at C3 and C4 positions according to the International 10-20 system [10]. The reference and ground electrodes were located at Fpz and the right mastoid, respectively. The impedance of the electrodes was kept below $5k\Omega$. The EEG signal was acquired with Discovery 20 amplifier (BrainMaster) and recorded with OpenVibe Software [19].

During the data recording stage some restrictions to the experiment protocol were introduced in order to preliminarily limit the number of artifacts in the recording. These restrictions, however, could not eliminate the artifacts fully. Therefore, in order to enhance the signal-to-noise ratio (SNR), the recorded EEG signal had to be subjected to some preprocessing. Since the most artifacts are outside the frequency band where motor potentials are searched for (alpha and beta frequency band), simple band-pass filtering in the 6-30 Hz frequency range was used in the reported survey. According to Fatourechi et al. low-pass filtering should remove most of EMG artifacts and high-pass filtering should remove EOG artifacts [4]. Moreover, the low-pass filtering allows also eliminating the artifacts caused by power lines that are in the 50 Hz range (in Poland). A Butterworth band-pass filter of the 4th order was used to filter the EEG data.

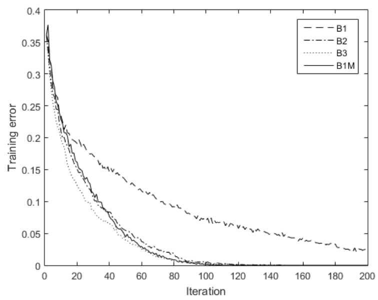
The classification of the motor imagery EEG data was performed with power band features. Twelve frequency bands were used to extract the features: alpha band (8-13Hz), beta band (13-30Hz), five sub-bands of alpha band (8-9Hz; 9-10Hz; 10-11Hz; 11-12Hz; 12-13Hz), five sub-bands of beta band (13-17Hz; 17-20Hz; 20-23Hz; 23-26Hz; 26-30Hz). The features were calculated separately per each second of the recording, hence there were 240 features. Due to a high number of features, the feature selection process was performed with LASSO algorithm [10]. Taking into account a small number of samples (200 samples), the number of classes (2 classes), and the possibility of using non-linear classifiers, we decided that no more than 8 features should be used in the classification process [18].

In the classification process four different classifier types were tested: boosting classifiers, LDA, SVM, and k-NN. First, we compared four boosting algorithms described in detail in Section 3. Then, we moved to SVM classifiers. Here we compared SVMs with different kernel functions: linear, quadratic, polynomial, and rbf. Finally, we tested k-NN classifiers with different values of k parameter (k was set to 1, 3, 5, and 7). The performance of each classifier, regardless of its type, was evaluated with 10-fold cross-validation scheme.

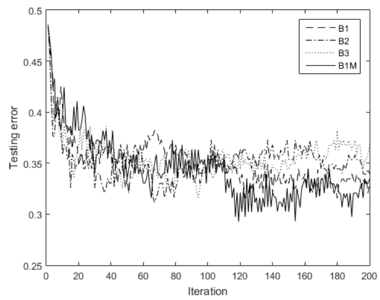
4 Results

Fig. 1 presents the results of the experiments, in which Boosting algorithms were used. The experiments were performed for 200 iterations for both learning Fig. 1(a) and testing Fig. 1(b) process. Three versions of Boosting algorithms (AdaBoost – B1, RealAdaBoost – B2, GentleAdaBoost – B3) were used in research as well as the modification of AdaBoost algorithm presented in this work (B1M) with $\lambda = 0$ and $k = 0.025$. The classification accuracy presented in Fig. 1(a) indicate unequivocally that the classifiers were overtrained. Such classifiers behavior might be a result of small learning sets each of them contained only 191 elements. The testing error was in the range of 37-32% , which was an average of 10 repetitions. The best results were obtained for the proposed modification of the AdaBoost algorithm. The average results from the iterations from 120 to 130 were used in the analysis. They are presented in Fig. 1(c). The X axis indicates the index of the testing subset.

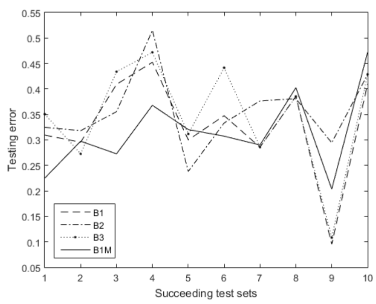
Fig. 2 presents the classification results for the algorithms from k-NN and SVM group. The algorithms that were characterized by stable classification for different testing subsets were selected for the final analysis. These were SVM with the linear kernel and k-NN with 3 neighbors.



(a) Boosting – learning



(b) Boosting – testing

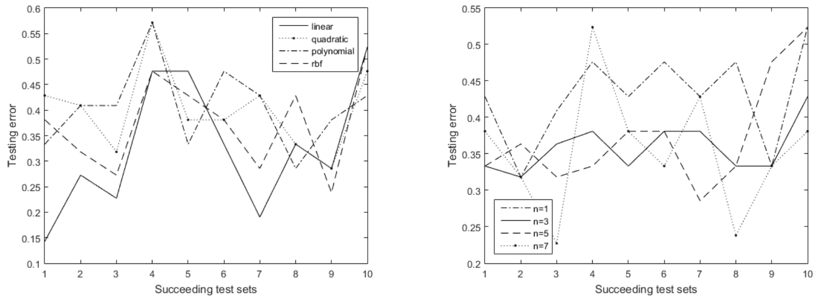


(c) Boosting – testing, separately for each subsets

Fig. 1. The accuracy achieved for boosting classifiers

The classification error obtained with the use of the selected algorithms is shown in Fig. 4 and in Tab. 2.

Three from the selected algorithms (B1M, linear SVM, LDA) were characterized by similar average errors. The difference (for both the average and the median) was no higher than 2%. The proposed boosting algorithm was the most stable due to the fact that the spread of results for the individual learning subsets was no higher than 23%, while for the two other algorithms it was 38% and 33% respectively. 3-NN algorithm had the same dispersion of results (23%),



(a) SVM – testing (b) k-NN – testing

Fig. 2. The accuracy achieved for SVM and k-NN classifiers

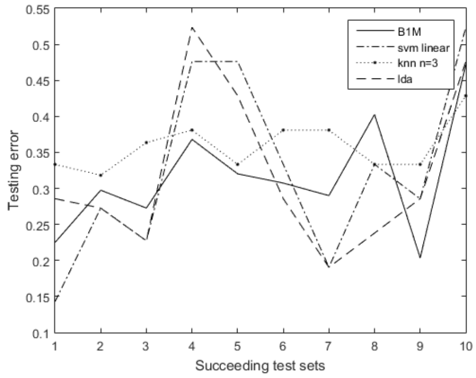


Fig. 3. The comparison of the accuracy achieved for four selected classifiers

however the correctness of the classification was inferior by 5% when compared with the results obtained for the boosting algorithm.

5 Conclusion

In the paper we compared the performance of four boosting classifiers (AdaBoost, RealAdaBoost, GentleAdaBoost, and our modification of AdaBoost B1M) and three classic classification algorithms: LDA, SVM, and k-NN. After the comparison, performed over the EEG data acquired during a MI-BCI session, we found out that our modification of AdaBoost provided the best results among the four boosting algorithms and almost all other classifiers tested. Two exceptions were LDA and SVM with linear kernel, the performance of which was on the same level as in the case of B1M.

Table 2. The error (in %) obtained for four selected classifiers for 10 testing subsets

	testing subsets											
Algorithm	1	2	3	4	5	6	7	8	9	10	mean	median
B1M	23	30	27	37	32	31	29	40	20	47	32	30
SVM linear	14	27	23	48	48	33	19	33	29	52	33	31
3-NN	33	36	32	33	38	38	29	33	48	52	37	35
LDA	29	27	23	52	43	29	19	24	29	48	32	29

The analysis, the outcome of which was presented in the paper, was meant as a preliminarily study on the application of boosting algorithms in BCI domain. Because our results were on the same level as the results of two classifiers leading in BCI research, we plan to extend our analysis to more subjects. Moreover, since the results obtained with the use of the boosting algorithm proposed in this paper were the most stable, we believe that it can bring real benefits when used not in an off-line but in an on-line BCI mode.

Acknowledgments. This work was supported in part by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology.

References

1. L. Breiman et al. Arcing classifier. *The annals of statistics*, 26(3):801–849, 1998.
2. R. Burduk. The adaboost algorithm with the imprecision determine the weights of the observations. *Lecture Notes in Computer Science*, 8398(Part II):110–116, 2014.
3. M. Donnerer and A. Steed. Using a p300 brain–computer interface in an immersive virtual environment. *Presence: Teleoperators and Virtual Environments*, 19(1):12–24, 2010.
4. M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch. Emg and eog artifacts in brain computer interface systems: A survey. *Clinical neurophysiology*, 118(3):480–494, 2007.
5. J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting. *The annals of statistics*, 28(2):337–407, 2000.
6. C. Guger, G. Edlinger, W. Harkam, I. Niedermayer, and G. Pfurtscheller. How many people are able to operate an eeg-based brain-computer interface (bci)? *IEEE transactions on neural systems and rehabilitation engineering*, 11(2):145–147, 2003.
7. C. Guger, A. Schlogl, C. Neuper, D. Walterspacher, T. Strein, and G. Pfurtscheller. Rapid prototyping of an eeg-based brain-computer interface (bci). *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 9(1):49–58, 2001.
8. C. Guger, A. Schlogl, D. Walterspacher, and G. Pfurtscheller. Design of an eeg-based brain-computer interface (bci) from standard components running in real-time under windows-entwurf eines eeg-basierten brain-computer interfaces (bci) mit standardkomponenten, das unter windows in echtzeit arbeitet. *Biomedizinische Technik/Biomedical Engineering*, 44(1-2):12–16, 1999.

9. I. Hayashi, S. Tsuruse, J. Suzuki, and R. T. Kozma. A proposal for applying pdi-boosting to brain-computer interfaces. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–6. IEEE, 2012.
10. H. H. Jasper. The ten twenty electrode system of the international federation. *Electroencephalography and clinical neurophysiology*, 10:371–375, 1958.
11. S. Z. Li and Z. Zhang. Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, 2004.
12. Y. Liu, H. Zhang, Q. Zhao, and L. Zhang. Common spatial-spectral boosting pattern for brain-computer interface. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, pages 537–542. IOS Press, 2014.
13. G. R. Müller-Putz, V. Kaiser, T. Solis-Escalante, and G. Pfurtscheller. Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel eeg. *Medical & biological engineering & computing*, 48(3):229–233, 2010.
14. G. Pfurtscheller, C. Brunner, A. Schlögl, and F. L. Da Silva. Mu rhythm (de) synchronization and eeg single-trial classification of different motor imagery tasks. *NeuroImage*, 31(1):153–159, 2006.
15. G. Pfurtscheller, C. Neuper, A. Schlogl, and K. Lugger. Separability of eeg signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE transactions on Rehabilitation Engineering*, 6(3):316–325, 1998.
16. P. Porwik, R. Doroz, and T. Orczyk. The k-nn classifier and self-adaptive hotelling data reduction technique in handwritten signatures recognition. *Pattern Analysis and Applications*, 18(4):983–1001, 2015.
17. P. Porwik, T. Orczyk, M. Lewandowski, and M. Cholewa. Feature projection k-nn classifier model for imbalanced and incomplete medical data. *Biocybernetics and Biomedical Engineering*, 36(4):644–656, 2016.
18. S. J. Raudys, A. K. Jain, et al. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):252–264, 1991.
19. Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. Lécuyer. Openvibe: An open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, 19(1):35–53, 2010.
20. R. E. Schapire. Using output codes to boost multiclass learning problems. In *ICML*, volume 97, pages 313–321, 1997.