A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces

Ulrich Hoffmann*, Gary Garcia*, Jean-Marc Vesin*, Karin Diserens† and Touradj Ebrahimi*

*Ecole Polytechnique Fédérale de Lausanne (EPFL), Signal Processing Institute, CH-1015 Lausanne, Switzerland
†Centre Hospitalier Universitaire Vaudois (CHUV), Rue du Bugnon 46, CH-1011 Lausanne, Switzerland
†Fondation Plein Soleil, I.-de-Montolieu 98, CH-1010 Lausanne, Switzerland

Abstract— **Gradient boosting is a machine learning method, that builds one strong classifier from many weak classifiers. In this work, an algorithm based on gradient boosting is presented, that detects event-related potentials in single electroencephalogram (EEG) trials. The algorithm is used to detect the P300 in the human EEG and to build a brain-computer interface (BCI), specifically a spelling device. Important features of the method described here are its high classification accuracy and its conceptual simplicity.**

The algorithm was tested with datasets recorded in our lab and one benchmark dataset from the BCI Competition 2003. The number of correctly inferred symbols with the P300 speller paradigm varied between 90% and 100%. In particular, all of the inferred symbols were correct for the BCI competition dataset.

Keywords— **Boosting, Brain-Computer Interface, EEG, P300, Ordinary Least Squares**

I. Introduction

The P300 is a characteristic waveform in the human EEG, occurring as a response to rare task-relevant stimuli in a series of task-irrelevant stimuli. The classical oddball paradigm is usually used to evoke the P300: two categories of stimuli are presented to a subject in random order, one of the categories occurs only rarely and subjects are instructed to determine to which category a stimulus belongs.

L. A. Farwell and E. Donchin were the first to use the oddball paradigm to build a BCI [1]. In their approach, a 6x6 matrix of symbols is presented to the user and rows and columns of the matrix are flashed in random order. Subjects can select a symbol from the matrix, by counting the number of times it flashes. Each time the desired character flashes, a P300 is elicited and can be detected by an appropriate algorithm.

In this paper, we describe a simple, yet powerful method to detect the P300 from single EEG trials and use it to build a P300 based spelling device. We employ *gradient boosting* in conjunction with ordinary least squares regression (OLS), to build a P300 detector.

Gradient boosting with OLS is an interesting alternative to state of the art algorithms for P300 detection (for example [5], [6], [7]) because it has the following characteristics:

- The algorithm builds linear classification rules in a parsimonious way. Thus only a small number of operations is necessary to apply the classifier to new data and realtime classification of single EEG trials is feasible. In addition the classification rules can be easily interpreted, i.e. it is very easy to derive from the classification rule which samples and which channels are important for detection of a P300.
- In terms of classification accuracy, the method presented here compares favorably to the state of the art. On the P300 dataset from the BCI Competition 2003, gradient boosting has a slight advantage over the results of the competition winners (see Sec. V).
- Sophisticated optimization algorithms, like those used for support vector machines or for independent component analysis are not necessary for the implementation of the method presented here. This makes the algorithm simple to implement, to use, and to extend.

The layout of the rest of the paper is as follows: In Sec. II we describe the experimental paradigm used for this work, the subjects and the preprocessing of the data. In Sec. III the boosting algorithm is described in detail. In Sec. IV, it is explained how the output of the classifier is used to infer the symbol a subject selected. Results are presented in Sec. V. Sec. VI draws the conclusion of this work.

II. Subjects and Methods

A. Subjects.

One male subject who had a complete cervical spinal cord injury (C3) 19 years before the recordings (subject S1) and one healthy male subject (subject S2) participated in the experiments. Subjects had previous experience with BCIs and were of age 36, and 28 years, respectively.

B. Experimental Paradigm.

A setup similar to that described in [1] was used to record and to label the data. A 6×6 matrix containing the letters of the alphabet and the numbers 1-9 was presented to the subjects on a laptop screen. Rows and columns of the matrix were flashed randomly for 100ms with a 100ms pause between flashes. Flashes were block-randomized, i.e. after 12 flashes each row and column was flashed exactly one time.

In each trial, subjects were instructed, to count how often a given symbol was flashed in the matrix. The

symbols subjects had to count were prescribed by the operator and displayed on the bottom of the screen. The number of flashes per trial was randomly chosen to be either 9×12, 10×12, or 11×12. To monitor performance of the subjects, there was a short break after each trial and subjects were asked to report their counting-result to the operator.

Two datasets were recorded from each of the subjects on different days. In the first session subjects were asked to spell the french words "lac," "nuage," "montagne," and "soleil." In the second session subjects had to spell the words "fromage," "chocolat," "pain," and "vin."

C. Data acquisition and preprocessing.

Data was recorded from channels Fp1, Fp2, AF3, AF4, F7, F3, Fz, F4, F8, FC1, FC5, FC6, FC2, T7, C3, Cz, C4, T8, CP1, CP5, CP6, CP2, P7, P3, Pz, P4, P8, PO3, PO4, O1, Oz, O2 with a Biosemi Active 2 system at 2048Hz. Epochs starting from the onset of a flash and lasting for 1s were extracted from the data. Slow drifts in the data were removed by least squares fitting of a linear function to each channel and subtracting it from the data. The data was then re-referenced to the average of channels O1, Oz, O2, lowpass filtered between 0 and 9 Hz with a 7th order Butterworth filter, and downsampled to 128 Hz. The channels used for re-referencing and channels T7, T8 were not used for further computations, since in the datasets recorded for this work, they did not carry relevant information for the detection of P300s.

D. Artifact rejection

To eliminate artifacts, first the absolute values of the samples of each epoch were computed. Then, from each epoch the maximum absolute value was chosen to represent the epoch. These values were then sorted in descending order. Epochs represented by the first 5% of the values, i.e. epochs with abnormally large maximal values were rejected.

III. Boosting Ordinary Least Squares

Boosting was employed, to compute from training data a function that detects P300s from single EEG trials. In particular, gradient boosting was used to stepwise maximize the Bernoulli log-likelihood of a logistic regression model. Stepwise maximization of the Bernoulli log-likelihood was originally described in [3], [4] with regression trees [2] as weak learning algorithm. Here ordinary least squares regression was used as weak learner. This choice is motivated by the following facts:

- Using OLS we obtain a discriminating function F that is easy to understand and to analyze: it is simply a linear combination of EEG samples.
- Building regression trees is computationally expensive, compared to OLS. The time required for training a classifier is thus drastically reduced, using OLS.
- Preliminary tests indicated, that on very noisy datasets regression trees might have a slight advantage over OLS in terms of generalization error. For typical EEG signal-to-noise ratios however, OLS performs better than regression trees.

Let us now describe gradient boosting with OLS in detail. We denote the ensemble of classifiers after step m by F_m, training data by $X = \{\mathbf{x}_i \in \mathbb{R}^K, i = 1 \dots N\}$, and corresponding class labels by $Y = \{y_i \in \{0, 1\}, i = 1 \dots N\}$. Furthermore $K = C \times S$ is the number of features, with C the number of EEG channels and S the number of samples in one epoch. The logistic regression model then reads:

$$p_m(y_i = 1|\mathbf{x}_i) = \frac{e^{F_m(\mathbf{x}_i)}}{e^{F_m(\mathbf{x}_i)} + e^{-F_m(\mathbf{x}_i)}}. \quad (1)$$

The Bernoulli log-likelihood of F_m, given the training data, can be expressed as:

$$L(F_m; X, Y) = \log\left(\prod_{i=1}^{N} p_m(y_i = 1|\mathbf{x}_i)^{y_i} p_m(y_i = 0|\mathbf{x}_i)^{1-y_i}\right). \quad (2)$$

The likelihood is maximized by setting $F_0 := 0$ and successively adding weak classifiers f_m to F_0:

$$F_m = F_{m-1} + f_m. \quad (3)$$

To obtain a weak classifier at step m, gradient descent is used. At each $\mathbf{x}_i$, the first derivative of the likelihood-function with respect to F is computed:

$$\tilde{y}_i = \left[\frac{\partial L(F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_{m-1}} \quad (4)$$

$$= 2(y_i - p_m(y_i = 1|\mathbf{x}_i)). \quad (5)$$

After computation of the gradient, the f that best fits the gradient in a least squares sense is selected:

$$f_m = \arg\min_f \sum_{i=1}^{N} (\tilde{y}_i - f(\mathbf{x}_i))^2. \quad (6)$$

We use weak classifiers that have a C-dimensional vector of regression coefficients $\mathbf{w}$ and a time index t as parameters. The output of a weak classifier is the projection of the vector $\mathbf{x}_i(t)$ of EEG samples at time t onto the regression coefficients:

$$f(\mathbf{x}_i; \mathbf{w}, t) = \mathbf{w}^\top \mathbf{x}_i(t). \quad (7)$$

For each $t \in 1 \dots S$ a $\mathbf{w}(t)$ is computed by least squares fitting the $\mathbf{x}_i(t)$ to $\tilde{y}_i$. Then the pair $(\mathbf{w}_m(t_m), t_m)$ that minimizes the error in Eq. 6 is chosen as parameters for the weak learner:

$$f_m(\mathbf{x}_i) = f(\mathbf{x}_i; \mathbf{w}_m(t_m), t_m). \quad (8)$$

Now the importance γ_m of the weak classifier in the ensemble (or equivalently the size of the step in direction

1. $p_0(y_i = 1|\mathbf{x}_i) = 0.5, \forall i$
2. $F_0(\mathbf{x}_i) = 0, \forall i$
3. For $m = 1$ to M do
   a) $\tilde{y}_i = 2(y_i - p_{m-1}(y_i = 1|\mathbf{x}_i)), \forall i$
   b) $f_m = \arg\min_f \sum_{i=1}^{N} (\tilde{y}_i - f(\mathbf{x}_i))^2$
   c) $\gamma = \arg\max_\gamma L(F_{m-1} + \gamma f_m)$
   d) $F_m = F_{m-1} + \epsilon\gamma f_m$
   e) $p_m(y_i = 1|x_i) = \frac{e^{F_m(\mathbf{x}_i)}}{e^{F_m(\mathbf{x}_i)} + e^{-F_m(\mathbf{x}_i)}}, \forall i$
   Endfor

Fig. 1. Pseudocode for the gradient boosting algorithm.

f_m) is determined such that [1]:

$$\gamma_m = \arg\max_\gamma L(F_{m-1} + \gamma f_m; X, Y). \quad (9)$$

To improve the generalization performance of the boosting algorithm, γ_m is shrinked to a small value through multiplication with a small ϵ at each step (as in [4]):

$$F_m = F_{m-1} + \epsilon\gamma_m f_m \quad (10)$$

The shrinkage strategy makes the gradient boosting procedure less greedy and the danger of taking large steps that can lead to a F with large $\ell 2$ norm is reduced.

After updating F, a new gradient is computed and a new f is added to F. This procedure is repeated until a certain number of iterations M is reached. Since the learning algorithm will overfit if we choose M too large or underfit if we choose M too small, we need to find an optimal M. To this end, we run the algorithm in a cross-validation loop with $M \in 1 \dots M_{\max}$ and afterwards choose the M that gives the smallest average error. The pseudocode for setting up a classifier with M iterations can be found in Fig. 1.

IV. Processing the classifier output

Once a classifier is trained with data from one session, it can be used, to infer the symbols a user was concentrating on in a new session. To do this, the outputs from the classifier are simply added up for each symbol. More specifically, after 12 flashes there are 2 EEG epochs for each symbol in the matrix (one row epoch and one column epoch). The results from the classifier for these epochs are added up.

Since the output of the classifier is an estimate of the probability that an epoch contains a P300, this corresponds to computing the expected number of P300s for each symbol. The symbol with the largest expected number of P300s is then chosen to be the symbol the user concentrated on.

[1]This is easy to solve computationally, since L is a concave function.

V. Results

We tested the algorithm with the datasets from Sec. II and with the P300 dataset from the BCI Competition 2003. In all experiments the maximal number of iterations of the boosting algorithm $M_{\max}$ was set to 200, the optimal M was determined in a 30×10 cross-validation loop, and ϵ was set to 0.05.

A. Datasets recorded for this work

To measure how good the algorithm described in this work generalizes, we trained it on data from the first (the second) session and applied the resulting classifier to the data of the second (the first) session. Artifact rejection was only applied to training sets, but not to test sets. From the data remaining after artifact rejection all P300 epochs and an equal number of randomly chosen epochs not containing a P300 were used to set up classifiers.

Out of $128 \times 27 = 3456$ features the algorithm selected 540 features for subject S1, session 1, and 351 features for session 2. For subject S2, 891 features were selected for session 1 and 1215 features for session 2. On average about 22% of all features were selected.

Fig. 2 shows the accuracy obtained during the cross-validation loop with data from session 1 for all subjects (graphs for session 2 look similar and are omitted here). As can be seen, the gradient boosting algorithm converges to an optimal solution and then slowly starts to overfit.

In Fig. 3 we plotted the percentage of wrongly predicted symbols vs. the number of repetitions in the P300 paradigm (one repetition is a block of 12 consecutive flashes). Both subjects reach an error-rate of 10% or less, after 9 repetitions.

B. BCI Competition 2003 dataset

To compare our work with the state of the art in P300 detection the P300 dataset from the BCI competition 2003 was used. This dataset consists of three sessions, the first

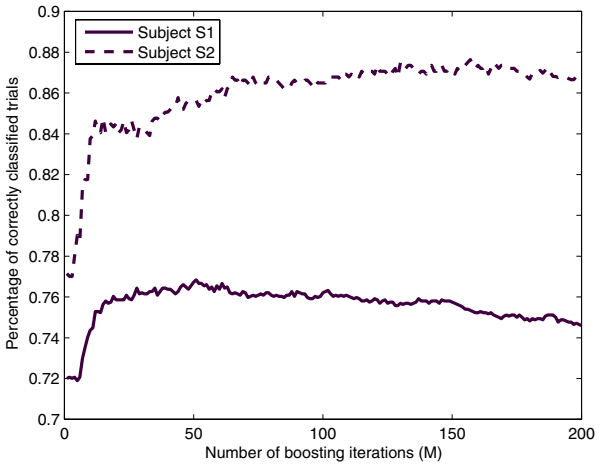

Fig. 2. Percentage of correctly classified trials for different values of M.

two sessions were available in the competition together with class labels indicating when a P300 occurred in the data. Class labels for the third session were not available during the competition and the goal was to predict as accurately as possible the words the subject spelled during the third session.

We trained a classifier using data from the first two sessions and used the classifier to infer the symbols of the third session. Preprocessing was similar to the approach described in Sec. II: the data were lowpass filtered with a 7th order Butterworth filter between 0 and 9Hz, the temporal mean was removed, and the data were downsampled to 120Hz. Only channels Fz, Cz, Pz, Oz, C3, C4, P3, P4, PO7, and PO8 were used (as in [6]). The symbols computed by our algorithm and the corresponding error rate are shown in Tab. I. As one can see the results are much better than those obtained with the datasets recorded for this work. This might be explained with the size of the training sets: about 880 epochs were used to set up classifiers from the datasets recorded for this work, 2520 epochs were used to set up a classifier from the competition training set.

The winners of the BCI competition 2003 [5], [6], [7] needed between 5 and 11 repetitions to infer all symbols correctly. The algorithm presented here needs only 4 repetitions. On the BCI competition dataset gradient boosting with OLS thus has a slight advantage compared to state of the art algorithms. However, extensive tests on more datasets would be necessary, to decide which algorithm really performs best.

VI. Conclusion and Future Work

The previous section showed, that the simple linear approach approach to P300 detection presented in this work, is suited for use in a BCI and gives results that compare favorably to the state of the art. The algorithm automatically selects samples, that are important for the detection of event-related potentials. It should thus be relatively easy to apply the method not only to P300 detection but also to other types of event-related potentials, for example readiness potentials.

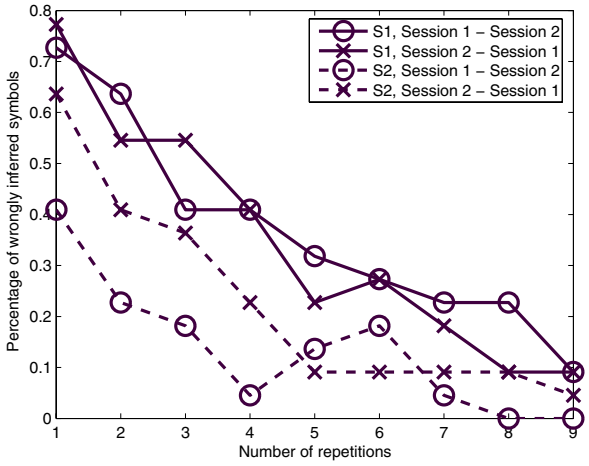

Fig. 3. Percentage of correctly inferred symbols for different numbers of repetitions.

TABLE I

Number of repetitions, inferred symbols, and error rate for the BCI Competition 2003 dataset.

Rep.	Inferred Symbols	Err.
1.	`fond moot bam jie cahe nunc zmbot x567`	29%
2.	`food goot bam pie cahe tuna zmaot x567`	19%
3.	`food moot ham pie cake tcna zsgon 4567`	10%
4.	`food moot ham pie cake tuna zygot 4567`	0%

To further improve the classification performance, it can be interesting to have a closer look at the errors the algorithm is making. One can see in Tab. I, that wrongly chosen symbols often are neighbors of the correct symbol in the matrix. Possible reasons for this problem are described in [8], however a solution still has to be found.

Another possibility for improvement, is to use more features of the P300. Whereas at the moment only the most salient features are used, namely the time-locked responses in the delta and theta-band, there are also features in other bands, for example event-related desynchronization in the alpha-band.

Acknowledgment

We thank Nicolas Gremaud for helping with the experiments. We thank Fondation Plein Soleil for providing support for some of the measurements. This work was funded in part by Swiss National Science Foundation grant no. 2153-067852.02.

References

[1] L. A. Farwell, E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials", *Electroencephalogr. Clin. Neurophysiol.* vol. 70, no. 6, pp. 510-523, 1988

[2] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, "Classification and Regression Trees", Chapman & Hall, 1984

[3] J. H. Friedman, T. Hastie, R. Tibshirani, "Additive logistic regression: a statistical view of boosting", *Ann. Statist.* vol. 28, pp. 337-407, 2000

[4] J. H. Friedman, "Greedy function approximation: a gradient boosting machine", *Ann. Statist.* vol. 29, no.5, pp. 1189-1232, 2000

[5] N. Xu, X. Gao, B. Hong, X. Miao, S. Gao, F. Yang, "Enhancing P300 Wave Detection Using ICA-Based Subspace Projections for BCI Applications", *IEEE Trans. Biomed. Eng.* vol. 51, no. 6, pp. 1067-1072, 2004

[6] M. Kaper, P. Meinicke, U. Grosskathoefer, T. Lingner, H. Ritter, "Support Vector Machines for the P300 Speller Paradigm", *IEEE Trans. Biomed. Eng.* vol. 51, no. 6, pp. 1073-1076, 2004

[7] V. Bostanov, "Feature Extraction From Event-Related Brain Potentials With the Continuous Wavelet Transform and the t-Value Scalogram", *IEEE Trans. Biomed. Eng.* vol. 51, no. 6, pp. 1057-1061, 2004

[8] C. Cinel, R. Poli, L. Citi, "Possible sources of perceptual errors in P300-based speller paradigm", Proceedings of the 2nd Graz International Brain-Computer Interface Workshop and Training Course 2004