Performance Optimization of ERP-Based BCIs Using Dynamic Stopping

Martijn Schreuder, Johannes Höhne, Matthias Treder, Benjamin Blankertz, Michael Tangermann

***Abstract*— Brain-computer interfaces based on event-related potentials face a trade-off between the speed and accuracy of the system, as both depend on the number of iterations. Increasing the number of iterations leads to a higher accuracy but reduces the speed of the system. This trade-off is generally dealt with by finding a fixed number of iterations that give a good result on the calibration data. We show here that this method is sub optimal and increases the performance significantly in only one out of five datasets. Several alternative methods have been described in literature, and we test the generalization of four of them. One method, called *rank diff*, significantly increased the performance over all datasets. These findings are important, as they show that 1) one should be cautious when reporting the potential performance of a BCI based on post-hoc offline performance curves and 2) simple methods are available that do boost performance.**

I. Introduction

Recent advances in brain-computer interfaces (BCIs) have resulted in both applications and methods that come closer to practical use [1]. BCI systems allow a person to control a device without the use of the brain's normal efferent pathways. In other words, by producing recognizable brain-states a person can convey her/his intention directly to a device. This is of particular interest for people with severe motor disabilities, but can also prove useful in other settings [2].

BCI systems based on event-related potentials (ERPs) have proven particularly useful, both for healthy users and end-users with locked-in syndrome. They have shown to convey more bits per minute and work for a wider range of subjects with higher accuracy than other brain states, such as motor imagery. The principal idea of any ERP BCI is to stimulate the user with 2 or more different stimuli. The user focuses on one of these stimuli, called the target. This target is generally less frequent than the rest, either by design or by the larger number of non-focused stimuli. The focused and non-focused stimuli elicit different brain patterns. This can be detected and exploited by the BCI. The most convenient way to pick up this signal is through non-invasive electroencephalography (EEG).

ERP BCIs are most widely used for communication and were originally introduced by [3]. Since then, several variations have been proposed changing the interface [4] or the modality [5], [6]. The original idea, however, remains; the user communicates by focusing attention to one of several offered options and thereby conveys her/his intention to the BCI.

Generally, a single selection is referred to as a trial. It contains several elements which are important to the estimation of the efficiency of the system: *Preparation* – some time is given to the user to prepare for a new round of stimulation and to determine the next target. *Stimulation* – described below. *Result* – after the stimulation, the BCI selects the most probable target and informs the user of this decision. Another important factor is the number of selections needed for correcting an error. All these should be taken into account when estimating the efficiency in a realistic manner.

The *stimulation* is repeated several times for each option in order to improve the signal-to-noise ratio of the EEG signal, and thereby the accuracy of the decision. Generally, one round (*iteration*) of stimulation of each option is performed before proceeding to the next iteration. Several such iterations make up the *stimulation* part of a trial. As each additional iteration prolongs the length of a trial, there exists a trade-off between accuracy and speed. Typically, this trade-off is not dealt with during online BCI use. Rather, an 'optimal' number of iterations is fixed during post-hoc offline analyses. Based on this number the potential speed of the BCI is reported, but only rarely is a method for reaching this 'optimal' number incorporated.

The challenge is thus to optimize the number of iterations. The simplest way to do this is by learning the optimal number of iterations on the calibration data and use this online. As performance of BCI is variable over time, more dynamic methods have been proposed [7], [8], [9], [10]. They stop the stimulation in a trial, based on the classification scores it has seen so far. Any such method should be robust and subject-independent. Alternatively, it should be possible to train it with few data. In many BCI systems the *Preparation* and *Result* part of a trial – collectively called *overhead* – are considerable. Thus, it often pays off for a stopping method to be conservative; i.e. it may cost more time to correct an error than is gained by stopping early.

Several of the methods proposed in literature have been tested here on different datasets. In summary, the results recommend to do early stopping, as it can significantly increase the performance for different types of BCI paradigms. When nothing is known about the user and paradigm, a conservative approach is recommended. Simply setting a fixed, lower threshold only increases performance significantly for one dataset.

This work was supported by the European ICT Programme Project FP7-224631 (TOBI) and FP7-216886 (PASCAL2) and the Bundesministerium für Bildung und Forschung (BMBF) (FKZ 01IB001A, 01GQ0850). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Martijn Schreuder, Johannes Höhne, Matthias Treder, Benjamin Blankertz and Michael Tangermann are with the BBCI group of the Machine Learning Department, Berlin Institute of Technology, Berlin, Germany.

Correspondence: schreuder@tu-berlin.de

TABLE I

DATA DESCRIPTION. *N* REFERS TO THE NUMBER OF PARTICIPANTS IN THE DATASET. *Sel. levels* REFERS TO THE NUMBER OF SELECTIONS NEEDED FOR WRITING A SINGLE SYMBOL. *Max It.* DESCRIBES THE MAXIMUM NUMBER OF ITERATIONS PER TRIAL THAT WERE AVAILABLE IN THE DATASET.

Dataset	*N*	*Modality*	*Sel. levels*	*Nr. Classes*	*Max It.*
AMUSE	16	Auditory	2	6	15
Hex-o-spell	13	Visual	2	6	10
Center	13	Visual	2	6	10
Cake	13	Visual	2	6	10
PASS2D	10	Auditory	1	9	11

II. METHODS

Both the methods and datasets used for this study are shortly described here, but we refer to the original work for the details.

A. Data description

Data were taken from two auditory experimental BCI paradigms, AMUSE [7], and PASS2D [11] and three visual paradigms [12], [4]. A data description can be found in Table I. All experiments consisted of a calibration phase and an online phase. During calibration, the user did not receive any feedback on her/his performance. In all cases, the participants used the BCI in the online phase for writing text in copy-spelling mode. AMUSE and the visual paradigms used a two step selection process. PASS2D uses a single selection and relies heavily on the applications intelligence. For details on the different paradigms we refer to the cited papers. The results presented here are based on cross validation on the calibration data.

B. Description for dynamic stopping methods

Due to the large overhead of most of the paradigms in our dataset, the most conservative threshold reported in the original work was used for each method. For all methods, the minimum number of iterations was set to three.

Fixed optimal. The simplest way of 'optimizing' the BCI is by estimating an optimal, but fixed, number of iterations on the calibration data. In fact, this resembles what is usually done in post-hoc offline analyzes as described before. As the number of iterations used can be set in virtually any BCI system, this method requires no implementation effort.

Rank diff. Rank-diff [7] bases its threshold on the difference p of the classifier medians of the two most likely classes. A larger p means a better separation of the class medians, and thus more confidence in a decision. Rank-diff finds a threshold for each iteration under the assumption that the same p becomes more reliable with an increasing number of iterations. The iteration threshold is set to be greater than the highest p resulting in a false positive. It thus avoids any false positives. A trade-off parameter R can be used to set the distance from this minimal threshold [7]. Rank-diff was designed for the AMUSE paradigm, but the dataset used here was not used to find good parameters.

Lenhardt. This method was originally proposed for a visual speller [8]. For each trial, the first iteration where a correct selection can be made is taken. For each class, the classification scores up to that iteration are summed. The vector containing these sums is normalized and summed to give a single value called 'intensity'. The final threshold is obtained by averaging intensities over all trials and participants. Online, the trial is stopped when the current intensity falls below this threshold. The original work proposes a second threshold for more robustness. Here it is not incorporated as finding the proper value is not feasible with a low number of trials.

Höhne. This method was proposed for an auditory speller paradigm [13]. In this simple approach, a one-sided t-test with unequal variances is applied for each class against all other classes. Thus, it is tested whether the distribution of classifier outputs of a specific class significantly deviates from the other classes. Each online trial is stopped as soon as a predefined level of significance is exceeded. This subject-specific significance threshold is found by simulation over the calibration data.

Liu. A very simple approach was taken in [9], where the average distance from the SVM hyperplane was calculated for target subtrials from the calibration data. Online, the trial is stopped as soon as the sum of classification scores up to that iteration is larger than N times the threshold for one of the classes.

C. Preprocessing

All EEG datasets were reduced to 27 trials – with varying numbers of subtrials –, low-pass filtered with a cut-off frequency at 45 Hz and resampled to 100 Hz. Epochs were created from 150 ms pre-stimulus to 800 ms post-stimulus and baselined on the pre-stimulus interval. Discriminative intervals were selected by a heuristic [14] within the cross-validation, and the average potential in these intervals for all channels (around 60) served as the feature vector for training and applying the classifier for that fold.

As done in the original implementation [9], we averaged three trials in the time domain for Liu during the online phase. For all other methods, single subtrial classification was performed.

D. Classification and validation

Results reported here come from leave-one-trial-out cross-validation on each of the five datasets. Parameters such as discriminative interval, classifier weights and stopping threshold were estimated on the training set only. As *Lenhardt's* threshold is estimated over multiple participants, the results of each cross-validation fold was pooled over participants and a fold-specific threshold was calculated.

Classification was done using an LDA classifier, regularized by shrinkage of the covariance matrix [14].

E. Metric

For the evaluation of the effect of the early stopping methods, a metric is needed that respects the speed-accuracy

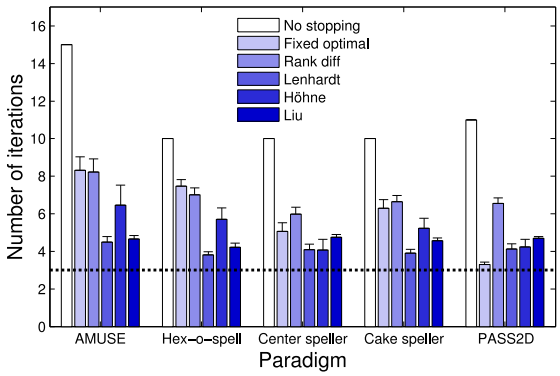

Fig. 1. Average number of iterations. Results are averaged over participants, dataset and cross-validation fold. Error bars represent the SEM. The dotted, black bar represents the minimal number of iterations that was enforced.

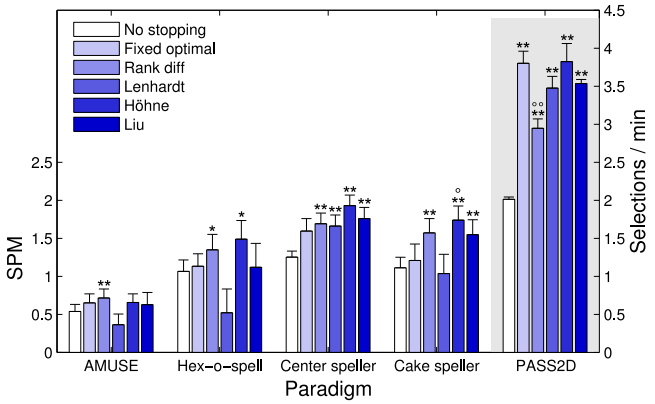

Fig. 2. Results of the offline cross-validation. Each early stopping method was tested against the *no stopping* condition and results are reported as * ($p < .05$) and ** ($p < .01$). Subsequently, all dynamic stopping methods were tested against the *fixed optimal* condition. Results are reported as ○ ($p < .05$) and ○○ ($p < .01$). Error bars represent the SEM. Note that the PASS2D has a separate scale on the right side.

trade-off under consideration of the overhead. While the ITR is a reasonable performance measure in general context, for spelling applications it is more straightforward and realistic to use the actual symbols per minute (SPM). In BCI spellers, the handling of errors is explicitly done by the use of a backspace symbol. Even though during offline classification no actual deletion of misspelled symbols takes place, it is possible to approximate the SPM. For single level interfaces, a correct selection counts as +1 symbol. An erroneous selection counts as -1 symbol, to account for the added effort of performing a backspace. This leads to the following formula:

$$symbols\ per\ minute = 60\ /\ time\ per\ symbol \quad (1)$$

$$E = Percent\ correct - Percent\ erroneous; \quad (2)$$

$$SPM = symbols\ per\ minute * E; \quad (3)$$

$Time\ per\ symbol\ [s]$ includes all the necessary overhead, as discussed in the introduction.

For the two step selection process in the AMUSE paradigm and all three visual spellers, there is another case that has to be considered. Errors can be made on the first level (group selection) and on the second level (single symbol selection). When the group selection went wrong, the second level contains a 'backdoor', which cancels the previous selection. As no symbol is selected, it counts as a 0 symbol selection (for term E). The $Percent\ correct$ is now interpreted as correct selection in both levels and $Percent\ erroneous$ as erroneous selection in the second level. Note that the simplifying assumption is – as in the classical ITR formula – that the accuracy is the same for each symbol.

As the PASS2D paradigm relies to a large extent on application intelligence, calculation of the SPM is not possible and *selections* per minute is reported.

III. Results

All methods reduced the number of iterations considerably, sometimes by as much as one third of the original number of iterations (see Figure 1). This reduces the time needed for a selection, and thus positively influences SPM performance. The black dotted line represents the minimum number of three iterations that was enforced. As can be seen, several methods almost reach this minimum value on average. For instance, *Liu* and *Lenhardt* are close to this minimum number of trials, even though the most conservative threshold was selected. On the other hand, due to its conservative nature the *rank diff* method stays well above this.

Figure 2 shows the SPM performance for all paradigms and stopping methods. The white column refers to the no stopping condition, where a trial always consisted of the maximum number of iterations (see Table I and Figure 1). Performances for all five stopping methods were compared to this using a two-sided, paired t-test (see Figure 2). Consecutively, all dynamic stopping methods were compared to the *fixed optimal* condition, which represents the method that is native to all BCI systems. All tests within one paradigm were corrected for multiple testing using the Bonferroni method. The result for each method is shortly described below.

On average, *fixed optimal* was better than no stopping for all datasets, though significance was found in only one out of five datasets. Some participants seem to benefit from this method, whereas others clearly show a drop in performance (not shown). Possibly, the method could be made more robust by deliberately adding a certain amount of trials, though finding this number is again an optimization problem.

Over all datasets, the *rank diff* method significantly increased the SPM score. In fact, no participant showed a reduced performance, which possibly owes to the highly conservative nature of the heuristic. On the relatively difficult AMUSE dataset, it is the only method that significantly increases the performance. On the other hand, it was the only method to perform significantly worse than the *fixed optimal* condition the PASS2D dataset.

Lenhardt was on all datasets one of the methods that required the least iterations, and on two datasets it increased

the performance significantly. Though a low number of iterations could increase the SPM, this is not a linear relation. In fact, for three datasets the performance was inferior to *no stopping*, though this was not significant. This is possibly why the original authors recommended a second constraint after evaluating their method.

Höhne showed a significant improvement over *no stopping* in four out of five datasets, and it reached the highest average SPM on all those. Furthermore, it was the only method to perform significantly better than *fixed optimal* on one dataset. Though not as conservative as *rank diff*, it only rarely decreased an individual performance (not shown).

Possibly the simplest method, *Lui*, increased performance significantly on three datasets. However, on the other two datasets it decreased the performance for a considerable number of participants (not shown). Possibly, this is due to the nature of the heuristic which uses absolute distances to the separating hyperplane. This distance may drift in non-stationary data such as EEG.

IV. Discussion

Looking at the averages can be misleading when generalizing to new subjects. For instance, it may be preferable to use a method that increases the average performance slightly less, when in return it does not decrease the performance on individual cases. A general trend for all methods is that they are good for subjects that show a good performance already. For the lower performing subjects they did not bring much benefit, some even decreased their performance. Only *rank diff* seems to be well behaving on all datasets in the sense that it never decreases performance. This comes at the cost of a lower average performance on some datasets, when compared to other stopping methods.

All methods tested here have their own 'trade-off' parameter, to tweak them for accuracy and speed. However, none of the original publications give a suggestion as to what may be a good value. Here, the most conservative value was used that was reported in the original work, which is what would typically be done in online experiments with new users. Later, when the characteristics of the user become clear, a less conservative value can be picked if this is feasible.

Fixed optimal refers to the method of simply setting a fixed, lower number of iterations. Any BCI to date allows this to be set by the operator, making it the method with the least implementation effort. Though it resulted in a slightly higher average performance on all datasets, this was only significant in one case. Possibly the fixed number of iterations that is optimal for the training set is not optimal for the test set. This may prove even more important for longer recordings. For this reason, the often reported potential performance of a BCI, based on offline analyzes, may be biased and not transfer well to online recordings.

Fortunately, there are good alternatives in literature, that do significantly increase the user's performance. As the *rank diff* method increased the performance on all datasets significantly, without decreasing the performance of any single subject, this seems to be the method of choice when little is known about the paradigm or subject. Other methods, such as *Höhne*, can maximally boost overall performance but result in slightly decreased performance on individual cases.

It is worth to note here again that none of the methods, except for *fixed optimal* is bound to a fixed number of iterations in online mode. They can vary the number of iterations during online use. This could prove useful with fluctuations in users attention, where lapses in attention can be reacted to by increasing the number of iterations.

V. Acknowledgments

The authors would like to thank Stefan Haufe and Thorsten Dickhaus for their discussions.

References

[1] E. W. Sellers, T. M. Vaughan, and J. R. Wolpaw, "A brain-computer interface for long-term independent home use," *Amyotrophic Lateral Sclerosis*, vol. 11, no. 5, pp. 449–455, 2010. [Online]. Available: http://informahealthcare.com/doi/abs/10.3109/17482961003777470

[2] B. Blankertz, M. Tangermann, C. Vidaurre, S. Fazli, C. Sannelli, S. Haufe, C. Maeder, L. E. Ramsey, I. Sturm, G. Curio, and K.-R. Müller, "The Berlin Brain-Computer Interface: Non-medical uses of BCI technology," *Frontiers in Neuroscience*, vol. 4, p. 198, 2010, open Access. [Online]. Available: http://www.frontiersin.org/neuroprosthetics/10.3389/fnins.2010.00198/abstract

[3] L. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and clinical neurophysiology*, vol. 70, pp. 510–523, 1988.

[4] M. S. Treder, N. M. Schmidt, and B. Blankertz, "Gaze-independent brain-computer interfaces based on covert attention and feature attention," *Journal of neural engineering*, 2011, submitted.

[5] M. Schreuder, B. Blankertz, and M. Tangermann, "A new auditory multi-class brain-computer interface paradigm: Spatial hearing as an informative cue," *PLoS ONE*, vol. 5, no. 4, p. e9813, 2010. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0009813

[6] A.-M. Brouwer and J. B. F. van Erp, "A tactile P300 brain-computer interface," *Frontiers in Neuroscience*, vol. 4, no. 19, p. 036003, 2010.

[7] M. Schreuder, T. Rost, and M. Tangermann, "Listen, you are writing! AMUSE, a dynamic auditory BCI," submitted.

[8] A. Lenhardt, M. Kaper, and H. Ritter, "An adaptive P300-based online brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, pp. 121–130, Apr 2008.

[9] T. Liu, L. Goldberg, S. Gao, and B. Hong, "An online brain-computer interface using non-flashing visual evoked potentials," *Journal of neural engineering*, vol. 7, no. 3, p. 036003, 2010.

[10] H. Zhang, C. Guan, and C. Wang, "Asynchronous P300-based brain-computer interfaces: a computational approach with statistical models," *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 1754–1763, Jun 2008.

[11] J. Höhne, M. Schreuder, B. Blankertz, and M. Tangermann, "A novel 9-class auditory ERP paradigm driving a predictive text entry system," *Frontiers in Neuroprosthetics*, 2011, submitted.

[12] M. S. Treder, N. M. Schmidt, and B. Blankertz, "Towards gaze-independent visual brain-computer interfaces," in *Frontiers in Computational Neuroscience*, 2010, conference Abstract: Bernstein Conference on Computational Neuroscience 2010. [Online]. Available: http://dx.doi.org/10.3389/conf.fncom.2010.51.00117

[13] J. Höhne, M. Schreuder, B. Blankertz, and M. Tangermann, "Two-dimensional auditory P300 speller with predictive text system," in *Conf Proc IEEE Eng Med Biol Soc*, vol. 1, 2010, pp. 4185–4188. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2010.5627379

[14] B. Blankertz, S. Lemm, M. S. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of ERP components – a tutorial," *NeuroImage*, vol. 56, pp. 814–825, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.neuroimage.2010.06.048