Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller

2014 J. Neural Eng. 11 035005

(http://iopscience.iop.org/1741-2552/11/3/035005)

OPEN ACCESS
IOP Publishing
Journal of Neural Engineering
J. Neural Eng. **11** (2014) 035005 (9pp)
doi:10.1088/1741-2560/11/3/035005

Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller

Pieter-Jan Kindermans[1,5], **Michael Tangermann**[2], **Klaus-Robert Müller**[3,4] **and Benjamin Schrauwen**[1]

[1] Electronics and Information Systems (ELIS) Department, Ghent University, Sint Pietersnieuwstraat 41, B-9000 Ghent, Belgium
[2] BrainLinks-BrainTools Excellence Cluster, Computer Science Department, University of Freiburg, Albertstr. 23, D-79104 Freiburg, Germany
[3] Machine Learning Laboratory, Technical University of Berlin, Marchstr. 23, D-10587 Berlin, Germany
[4] Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea

E-mail: pieterjan.kindermans@ugent.be, michael.tangermann@blbt.uni-freiburg.de, klaus-robert.mueller@tu-berlin.de and benjamin.schrauwen@ugent.be

Received 30 August 2013, revised 1 November 2013
Accepted for publication 27 November 2013
Published 19 May 2014

Abstract

Objective. Most BCIs have to undergo a calibration session in which data is recorded to train decoders with machine learning. Only recently zero-training methods have become a subject of study. This work proposes a probabilistic framework for BCI applications which exploit event-related potentials (ERPs). For the example of a visual P300 speller we show how the framework harvests the structure suitable to solve the decoding task by (a) transfer learning, (b) unsupervised adaptation, (c) language model and (d) dynamic stopping. *Approach.* A simulation study compares the proposed probabilistic zero framework (using transfer learning and task structure) to a state-of-the-art supervised model on $n = 22$ subjects. The individual influence of the involved components (a)–(d) are investigated. *Main results.* Without any need for a calibration session, the probabilistic zero-training framework with inter-subject transfer learning shows excellent performance—competitive to a state-of-the-art supervised method using calibration. Its decoding quality is carried mainly by the effect of transfer learning in combination with continuous unsupervised adaptation. *Significance.* A high-performing zero-training BCI is within reach for one of the most popular BCI paradigms: ERP spelling. Recording calibration data for a supervised BCI would require valuable time which is lost for spelling. The time spent on calibration would allow a novel user to spell 29 symbols with our unsupervised approach. It could be of use for various clinical and non-clinical ERP-applications of BCI.

Keywords: BCI, unsupervised learning, subject-to-subject transfer, visual event-related potentials, P300, speller matrix

(Some figures may appear in colour only in the online journal)

[5] Author to whom any correspondence should be addressed.

1. Introduction

Since the introduction of the original P300 speller [10], the related research has been very diverse. The signal to noise ratio is improved by using either novel visual stimulus types [16, 32], by exploring non-random stimulus sequences [31] or by novel stimulus patterns [33]. To counter the problem of gaze dependence, variations on the paradigm were investigated, both in the visual [34] and the auditory domain [14, 28]. To speed up the communication rate, a predictive word model can be used such that the user is able to spell an entire word with a single selection [26]. One of the most successful approaches is the use of dynamic stopping [17, 27, 35], where the stimulus presentation is stopped at the point when the classifier is confident about its decision. Improvements in the decoding of the brain signals itself are mainly the result of using properly regularized linear machine learning models [2, 9, 22], the inclusion of language models or adaptive subject-specific methods [6, 17, 19, 23, 25, 29].

But even when all these improvements are combined, the event-related potential (ERP) spellers are still not easy to use; still a calibration session is required (which is commonplace in BCI). Ideally, however a user would like to start communicating right away. This holds both for clinical users as well for healthy users with non-medical applications in mind [3].

Starting in 2008, Krauledat and colleagues took the first step towards zero training for motor-imagery BCIs, however, only in the context of inter-session transfer for the same user [21]. One of the first successful attempts to dismiss calibration recordings for novel users was published in 2009 by Lu *et al* and Fazli *et al* [11, 12, 24]. They proposed a general subject-unspecific model, which in the case of Lu could be adapted to the novel user during online usage. Recently, there has been an increased amount of interest in using non-adaptive subject-unspecific models in order to bypass the calibration session [5, 13, 15], unfortunately, they do not perform as well as subject-specific models. We proposed completely unsupervised training to build a zero-training ERP BCI [20]. This is a classifier which is randomly initialized in the beginning and learns to decode the EEG while the user is working with the BCI. This approach is as reliable as a supervised system when enough data is available, but it has to get this data first. During this initial phase, which we call the warm-up, the classifier is unreliable.

Subject-unspecific models and unsupervised learning are of course not mutually exclusive, and these can be combined in a unified probabilistic model for ERP spelling [18]. Additionally, such a probabilistic approach can be further extended with language models, and it exhibits a natural dynamic stopping criterion: how certain the classifier is for a single stimulus. Therefore, we wanted to answer the following questions: Can dynamic stopping be used when there is no data to determine the optimum stopping criterion, and when there is no subject-specific training involved? How does this approach compare to a supervised model? We performed simulations of online experiments to get the answers.

In the next section, we will elaborate on the used probabilistic model and the dynamic stopping strategy. Afterwards we will detail the experimental setup, followed by a presentation of the results and finally a discussion.

2. Methods

2.1. Preprocessing

The EEG is preprocessed on a character by character basis. We re-reference the EEG by using a common average reference filter, which is followed by a bandpass-filtered (0.5–15 Hz) and channel-wise normalization to zero mean and unit variance. Then we sub-sample the data by a factor of 6–42.67 Hz, and retain ten samples per channel centred around 300 ms after stimulus presentation. Afterwards a bias term, a constant feature with value equal to 1, is added. The classifier weight that operates on this bias term is the intercept of the classifier.

2.2. Probabilisitc model for ERP paradigms comprising inter-subject transfer and language statistics

The basic probabilistic model is presented originally in [20] and is extended with a transfer learning approach and a language model extension in [18].

The model describes a typical ERP paradigm, including the constraint that only the attended stimulus can result in a target ERP response, and assumes that the EEG can be projected into one dimension where it is Gaussian with a class-dependent mean (-1 or 1) and shared variance. Furthermore, this assumption holds for ERP features; Blankertz *et al* [2] have shown that Gaussian distributions with a class-dependent mean and shared covariance are a good approximation of the true distribution on the ERP features; consequently, each one-dimensional projection must be Gaussian with class-specific mean and shared variance.

In its basic form, the model assumes that each stimulus has equal prior probability of being the attended one. Language statistics can be used to extend the model by modifying the prior probability based on the history.

Furthermore, the basic version of the model decouples all subjects and the classifier is regularized through a zero mean and isotropic covariance prior on the weight vector. Transfer learning can be introduced by coupling the distributions of the different subject-specific weight vectors by sharing the prior mean and introducing a hyper-prior.

A graphical representation of the model is given in figure 1 and the model itself is as follows:

$$
\begin{aligned}
p(\boldsymbol{\mu}_{\boldsymbol{w}}) &= \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}|0, \alpha_p I),\\
\alpha_p &= 0,\\
p(\boldsymbol{w}_s|\boldsymbol{\mu}_{\boldsymbol{w}}) &= \mathcal{N}(\boldsymbol{w}_s|\boldsymbol{\mu}_{\boldsymbol{w}}, \alpha_s I),\\
p(c_{s,t}|c_{t-n+1}, \ldots, c_{t-1}) &= \text{Multinomial}(\boldsymbol{\kappa}(c_{t-n+1}, \ldots, c_{t-1})),\\
p(\boldsymbol{x}_{s,t,i}|c_{s,t}, \boldsymbol{w}_s, \beta_s) &= \mathcal{N}(\boldsymbol{x}_{s,t,i}^T \boldsymbol{w}_s|y_{s,t,i}(c_{s,t}), \beta_s).
\end{aligned}
$$

Here $\boldsymbol{x}_{s,t,i}$ is the $(N \times 1)$-dimensional[6] feature vector of subject s corresponding to stimulus i during trial t; $X_s = [\boldsymbol{x}_{s,1,1}, \ldots, \boldsymbol{x}_{s,T,I}]$ contains all feature vectors of the subject s. The attended stimulus/symbol during trial t is $c_{s,t}$.

[6] This includes the bias term.

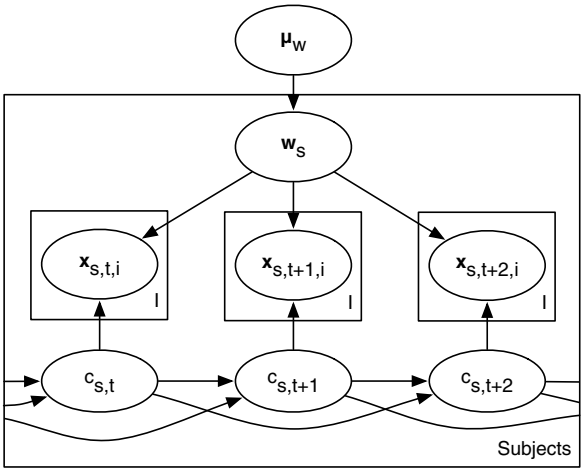

Figure 1. Graphical model representation of the unsupervised ERP speller with transfer learning and language model extensions. The rectangles indicate multiple instances, i.e. there is one $\boldsymbol{w}_s$ per subject, I stimuli per trial and a single feature vector $\boldsymbol{x}_{s,t,i}$ per stimulus.

The function $y_{s,t,i}(c_{s,t})$ encodes the class-dependent mean, $y_{s,t,i}(c_{s,t}) = 1$ if $c_{s,t}$ is contained in stimulus i during trial t, otherwise $y_{s,t,i}(c_{s,t}) = -1$. The vector $\boldsymbol{y}_s(\boldsymbol{c}_s) = [y_{s,1,1}(c_{s,1}), \ldots, y_{s,T,I}(c_{s,T})]^T$ comprises all target means for a single subject. The weight vector used to project the features into a single dimension is $\boldsymbol{w}_s$ and the shared prior mean on this weight vector is $\boldsymbol{\mu}_{\boldsymbol{w}}$. The precision of the prior is subject-specific and represented by α_s. The precision of the shared hyper-prior is α_p. The per-class variance of the projected ERP features is β_s^{-1}. Finally, the language model is encoded by the function $\boldsymbol{\kappa}(c_{t-n+1}, \ldots, c_{t-1})$ which gives the prior probability for the symbols in the P300 matrix conditioned on the history.

Next, we will discuss inference in the model, dynamic stopping, unsupervised training and transfer learning.

2.2.1. Inference. Computing the likelihood of a symbol given the EEG data is a straightforward application of Bayes's rule when the language model has no history, i.e. when $p(c_{s,t}|c_{t-n+1}, \ldots, c_{t-1}) = p(c_{s,t})$:

$$\hat{c}_{s,t} = \arg\max_{c_{s,t}} p(c_{s,t}|X_{s,t}, \boldsymbol{w}_s, \beta_s) = \frac{p(c_{s,t})p(X_{s,t}|c_{s,t}, \boldsymbol{w}_s, \beta_s)}{\sum_{c_{s,t}} p(c_{s,t})p(X_{s,t}|c_{s,t}, \boldsymbol{w}_s, \beta_s)}.$$

Unfortunately, including history-based language models does complicate inference. In this case, two solutions are possible. The simplest solution is to assume that the text is either correct or has to be corrected by the user with a backspace command. This effectively reduces the inference approach to the procedure where no history is considered. However, using this approach, the language model is not utilized in its most powerful setting because only information from the previous trials can be exploited.

But when we assume that the user does not correct the mistakes and that he relies on the language model to correct the mistakes for him, then we fully exploit the information contained in the language models. Also note that using this approach, the prediction for previous trials can change as more trials get spelled. Unfortunately, this comes at a computational cost, as it requires a convoluted inference approach, which is called the forward–backward algorithm [1]. We will briefly summarize this method. To keep the notation uncluttered, we omit the conditioning on $\boldsymbol{w}_s$, β_s and the subscript s.

The likelihood of a symbol in trial t given the data from all trials X can be written as follows:

$$p(c_t|\boldsymbol{X}) = \sum_{c_{t-1},\ldots,c_{t-n+2}} \frac{p(X_1, \ldots, X_T, c_t, \ldots, c_{t-n+2})}{p(X)} = \sum_{c_{t-1},\ldots,c_{t-n+2}} \frac{p(X_1, \ldots, X_T, c_t, \ldots, c_{t-n+2})}{\sum_{c_t,\ldots,c_{t-n+2}} p(X_1, \ldots, X_T, c_t, \ldots, c_{t-n+2})}.$$

In both parts of the fraction $p(X_1, \ldots, X_T, c_t, \ldots, c_{t-n+2})$ appears, which we can decompose into a forward and backward component:

$$p(X_1, \ldots, X_T, c_t, \ldots, c_{t-n+2}) = f(c_t, \ldots, c_{t-n+2}) \times b(c_t, \ldots, c_{t-n+2}),$$
$$f(c_t, \ldots, c_{t-n+2}) = p(X_1, \ldots, X_t, c_t, \ldots, c_{t-n+2}),$$
$$b(c_t, \ldots, c_{t-n+2}) = p(X_{t+1}, \ldots, X_T|c_t, \ldots, c_{t-n+2}).$$

These components can be computed recursively.

$$f(c_t, \ldots, c_{t-n+2}) = p(X_t|c_t) \sum_{c_{t-n+1}} p(c_t|c_{t-1}, \ldots, c_{t-n+1}) \times f(c_{t-1}, \ldots, c_{t-n+1}),$$
$$b(c_t, \ldots, c_{t-n+2}) = \sum_{c_{t+1}} p(X_{t+1}|c_{t+1})p(c_{t+1}|c_t, \ldots, c_{t-n+2}) \times b(c_{t+1}, \ldots, c_{t-n+3}).$$

The initialization of the forward and backward recursion is analogous to the initialization in an HMM: the backward recursion is initialized to 1 and the forward recursion is initialized to $p(X_1, c_1)$ [1]. An important observation is the fact that when we cache the values of the forward pass, computing the likelihood of the symbol in the last trial requires only a single step of the forward and backward recursion. Hence, it is possible to implement dynamic stopping in combination with the forward–backward algorithm.

2.2.2. Dynamic stopping. The dynamic stopping strategy itself is kept simple. After each iteration, we compute the probability for all possible symbols in the last trial. We stop the stimulus presentation when the maximum number of epochs is reached or when the most likely symbol receives 99% of the probability mass.

2.2.3. Unsupervised training. The unsupervised training is based on a combination of the expectation maximization (EM) algorithm [8], where the desired symbols are treated as the latent variables, to optimize for $\boldsymbol{w}_s$, β_s and direct maximum likelihood to optimize α_s. For now, we will assume that $\boldsymbol{\mu}_{\boldsymbol{w}}$ is known and the resulting update equations are

$$\boldsymbol{w}_s = \sum_{\boldsymbol{c}_s} p(\boldsymbol{c}_s|X_s, \boldsymbol{w}_s^{\text{old}}, \beta_s^{\text{old}}) \left(X_s X_s^T + \frac{\alpha_s^{\text{old}}}{\beta_s^{\text{old}}} I\right)^{-1} \times \left(X_s \boldsymbol{y}_s(\boldsymbol{c}_s) + \frac{\alpha_s^{\text{old}}}{\beta_s^{\text{old}}} I \boldsymbol{\mu}_w\right),$$

$$\beta_s^{-1} = \left\langle \sum_{c_{s,t}} p\left(c_{s,t}|X_s, \boldsymbol{w}_s^{\text{old}}, \beta_s^{\text{old}}\right)\left(\boldsymbol{x}_{s,t,i}^T \boldsymbol{w}_s^{\text{old}} - y_{s,t,i}(c_{s,t})\right)^2 \right\rangle_{t,i}$$

$$\alpha_s = \frac{D}{\left(\boldsymbol{w}_s^{\text{old}} - \boldsymbol{\mu}_w\right)^T \left(\boldsymbol{w}_s^{\text{old}} - \boldsymbol{\mu}_w\right)}.$$

The update for $\boldsymbol{w}_s$ consists of creating a weighted combination of all the possible ridge regression classifiers. The weight assigned to each of these classifiers is equal to the probability that the target stimuli used in training were correct according to our previous estimate of $\boldsymbol{w}$, β_s. The update for β_s^{-1} is equal to the expected mean-squared error between the projection target and the actual projection. Finally, the updated α_s is simply the average-squared classifier weight.

2.2.4. Transfer learning. When we train the model without transfer on an initial set of subjects: $s = 1, \ldots, S$, then we initialize $\alpha_s = \alpha_p = 0$ and $\boldsymbol{\mu}_w = 0$. After the training on all previously seen subjects, we have a subject-specific Maximum *a posteriori* estimate: $\boldsymbol{w}_s^{\text{new}}$ and an optimized value α_s^{new} for all of them. These can be used to compute the posterior on the classifier's prior mean: $\boldsymbol{\mu}_w$:

$$p\left(\boldsymbol{\mu}_w|\boldsymbol{w}_1^{\text{new}}, \ldots, \boldsymbol{w}_s^{\text{new}}\right) = \mathcal{N}\left(\boldsymbol{\mu}_w|\boldsymbol{\mu}_p^{\text{new}}, \alpha_p^{\text{new}} I\right),$$

$$\boldsymbol{\mu}_p^{\text{new}} = \frac{1}{\alpha_p^{\text{new}}} \sum_{s=1\ldots S} \alpha_s^{\text{new}} \boldsymbol{w}_s^{\text{new}}, \alpha_p^{\text{new}} = \sum_{s=1\ldots S} \alpha_s^{\text{new}}.$$

To apply transfer learning to a new subject $S+1$, we initialize $\boldsymbol{\mu}_w$ to $\boldsymbol{\mu}_p^{\text{new}}$ and keep it fixed. α_{S+1} is set to α_p^{new} but also optimized during online usage. Optimizing α_{S+1} allows us to switch between staying close to the prior when the data is hard to fit. Or to build a very specific model when we find a good fit to the data (α_{S+1} becomes very small in this case). As we mentioned before, the transfer learning approach regularizes the subject-specific solution towards the general model.

2.3. Data and experiments

We use the Akimpech dataset[7] for all our experiments. It comprises 10 channel EEG data from 22 subjects, collected during BCI sessions with the classic 6 × 6 visual matrix speller interface. The training set consists of 16 trials and the average number of trials in the test set is 22.18 with a minimum of 17 and a maximum of 29; each trial contains 15 stimulus iterations. Please note that the training set is only used to supervisedly train a baseline model, which our proposed methods are compared against. It is not touched during the unsupervised or transfer experiments. The stimulus duration in the Akimpech setup is 125 ms, the inter-stimulus interval is 62.5 ms and the pauses between trials comprise 4 s.

A drawback of the Akimpech dataset is that the target texts are limited to a few different Spanish words. Consequently, in its standard form the target texts are too restrictive for a thorough evaluation of language models because the desired text can greatly influence the impact of a language model. A text which is very likely under the language model will result in a large performance increase. If on the other hand an unlikely text is to be spelled, the performance may degrade due to the language model. Therefore, it is important to evaluate language model approaches on a large collection of different texts [17].

To allow such an extensive evaluation, we propose to re-synthesize a dataset where the desired text is modified using the methodology from [17, 18]. The idea behind the re-synthesizing is that in an ERP speller, a lookup table is used to assign a symbol to each position in the speller matrix. Hence, the task for the decoding algorithm is to detect which position in the grid contains the target stimulus. The mapping from a position in the matrix to a symbol or a spelling action is a post-processing step. We assume that the recognition of the ERP response does not depend on the meaning of the attended stimulus. This allows us to alter the desired text by modifying the lookup table in a simulation. As a result, we could evaluate a language-model-based speller on an arbitrary number of texts. Hence, we can evaluate the influence of a language model on the ERP detection. Please note that this approach does not modify the EEG or the stimulus structure. Therefore, when no language model is used, the spelling accuracy will not change when the desired text is modified. Of course, using a language model will impact the spelling accuracy but this is the effect we are investigating.

Training the language models and sampling novel desired texts is done using different parts of the Wikipedia dataset from [30].

To train the language models, we limit ourselves to the first 5×10^8 characters. The dataset is first transformed to lowercase, afterwards, we count the symbol occurrences to determine the 36 most frequent symbols, excluding digits. This results in the character set [**a–z:%()'-",._**], where the underscore symbol codes for white-space characters. This character set will be used during our evaluation. Next, we computed the n-gram letter frequencies for uni-, bi- and tri-gram models using this limited character set. These n-gram models are regularized by using Witten–Bell smoothing [4], a technique which assigns small but non-zero probabilities to n-grams that are not present in the training set.

To obtain the target texts for the evaluation, we use the second and previously untouched part of the dataset. This part was transformed to lower-case before dropping those symbols which are not contained in the above character set. For each of the 22 subjects, we sampled 20 contiguous texts, resulting in a total of 440 texts. In our simulations, we evaluate each technique on all 20 different texts to eliminate the bias from a single text. Per subject, each text is equal in length to the number of trials present for that subject. Consequently, EEG data from all trials, but limited to the maximum number of iterations, will be used during the evaluation of a single text and we will re-use the same data to evaluate all texts. During simulation, the lookup table is modified such that the new desired target is in the position of the attended stimulus. Furthermore, because most errors in the matrix speller result in the selection of a neighbouring symbol, the lookup table always is formed by a cyclically shifted version of the matrix containing [**a–z:%()'-",._**], with **a** in the top left and _ in the bottom right corner of the screen. This ensures that the set of possible neighbouring symbols is fixed and therefore emulates a true online experiment.

[7] http://akimpech.izt.uam.mx/p300db/

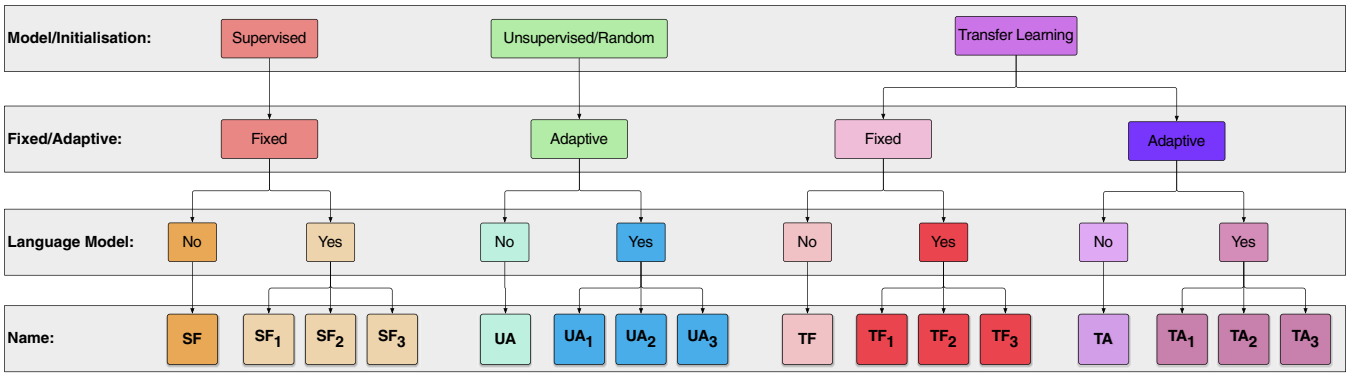

Figure 2. Overview of the different methods and their properties.

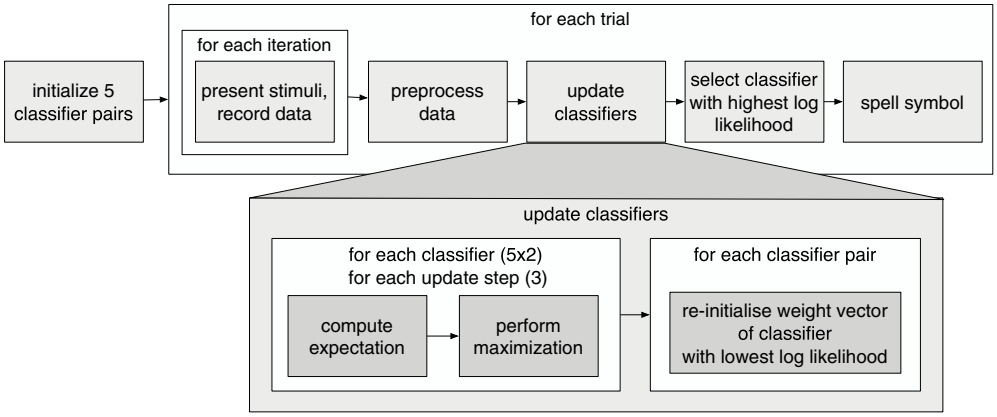

Figure 3. Representation of the algorithm used for the UA classifier.

The transfer learning approach is applied in a leave-one-subject-out manner. First, we train offline and unsupervisedly on 21 subjects. Afterwards we combine the different subject-specific classifiers into a general subject-unspecific model. This is subsequently used as initialization and regularization for the novel subject. We opted for initial unsupervised training because it mimics the use case where previous users are working with the system in a free spelling scenario where no labelled data is available. Therefore, using unsupervised transfer learning gives us the most flexibility.

During simulation, unsupervised adaptation works as follows. We process the data trial per trial and we feed and update the classifier with EEG signals one iteration at a time. For the non-transfer learning setting, we use five pairs of randomly initialized classifiers as is described in [18]. Then for each trial, three EM updates are executed and the best classifier is selected based on the log-likelihood values. We then predict the desired symbol and move on to the next trial. This approach is visualized in figure 3. As this straightforward approach for updating is rather time-consuming, it should not be used without modification to be combined with dynamic stopping. Consequently, in the adaptive transfer learning and dynamic stopping experiments, we first predict the desired symbol (losing some information for this prediction) and improve the classifier subsequently by three EM updates. As those three EM iterations take 1.1 s at most, these updates can easily be executed during the pause between trials. A graphical representation of this approach is shown in figure 4.

Finally, we enumerate the methods used in the following comparison and we present an overview in figure 2. The baseline model SF is a supervised and fixed classifier based on the Bayesian ridge regression where the evidence approximation is used to update the hyper-parameters [1]. The randomly initialized, but unsupervisedly adapted model is named UA. The fixed transfer learning model TF has been trained unsupervisedly. It is not adapted to the novel user. More elaborated, the unsupervisedly trained model TA is a transfer learning model, which is further adapted to a novel user. On top of that, we will use the *subscripts 1,2,3* to indicate the length of an n-gram language model. Finally, as we have mentioned previously, both the unsupervised adaptation and the language models in principle could result in a modification of the already predicted outcome of a previous trial. To evaluate this effect, we will analyse the performance of those methods *after* they have processed all trials. We denote the evaluation of a model which is allowed to make its final prediction even for past symbols after having processed all previous iterations and only then being updated by adding a *superscript U*. The *prefix DS* will be used to indicate models which make use of dynamic stopping. Figure 2 gives an overview of all the model options.

3. Results

Among all studied unsupervised methods, the most evolved version DS-TA$_3^U$ using (1) transfer learning, (2) a tri-gram

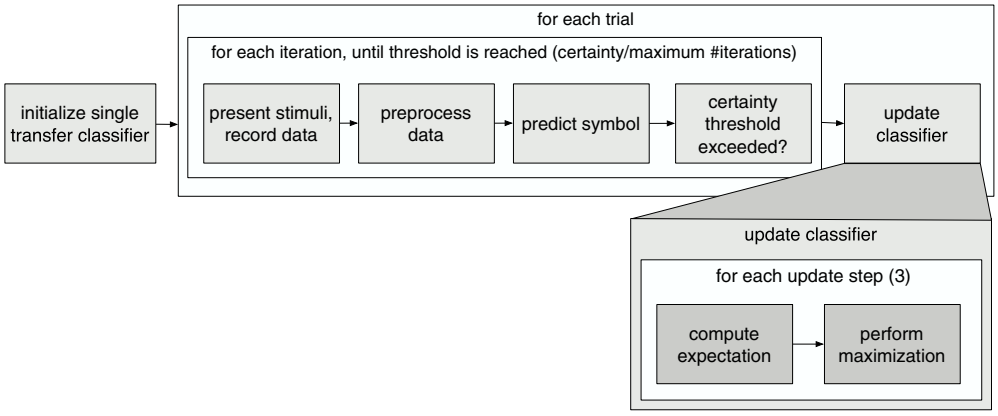

Figure 4. Representation of the algorithm used for the DS-TA classifier. The transfer learning initialization compromises unsupervised training on date recorded from other users, followed by combining these subject-specific models into a general model. Bypassing the *certainty threshold block* reduces this algorithm to the basic TA algorithm. Omitting the *update classifier block* reduces this algorithm to the TF version. Please note that the stimulus presentation and data recording can be executed in parallel (i.e. in separate threads) with the preprocessing and spelling components.

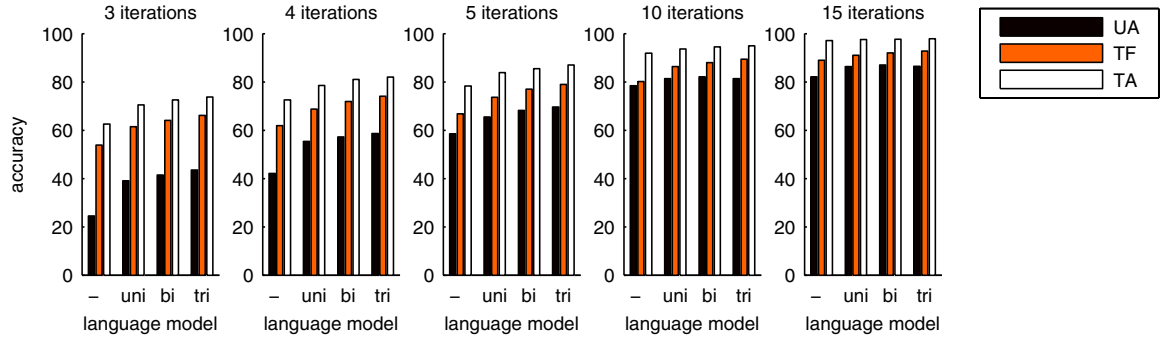

Figure 5. Accuracy for simulated online prediction using the unsupervised models.

language model, (3) continuous adaptation and (4) dynamic stopping of course can be expected to outperform other unsupervised methods. As each of the four *add-ons* contribute to the implementation complexity, it is in the practitioner's interest to see an analysis of the individual contributions of these add-ons onto the overall performance, which is provided in the following sections.

Finally and to convince the BCI practitioner, it is clear that even the best unsupervised method needs to be compared to the gold standard of supervised classifier training. We do this in terms of correctly spelled symbols per minute (SPM) in section 3.6, while any other performance metric provided is describing the per cent of correctly spelled letters (rather than the binary classification accuracy).

3.1. Unsupervised models and the warm-up period

An overview of the spelling accuracy of the unsupervised models UA, TF and TA and their language-model-based variants is provided by figure 5. For comparison, the results for the *updated* prediction, i.e. after processing all trials, are given in figure 6. Finally, table 1 contains the *spelling accuracy* for a subset of the unsupervised methods and baseline supervised models, including dynamic stopping.

It is clear that UA performs poorly when only a low number of iterations is used; the accuracy, which is averaged over 20 initializations, is only 24.5% for 3 iterations and 58.6% for 5 iterations. Even though the result for three iterations is above chance level, it is not usable in a BCI. Increasing the number to 10 and 15 raises the accuracy to 78.4% and 82.1%. However, a standard supervised model is able to achieve 85.6% with just five iterations per trial. The poor performance of the unsupervised model should not come as a surprise, because it is initialized randomly and has to learn on the fly during a very limited number of trials. Hence, the likelihood of a mistaken symbol during the very first trials is high. After the first few trials, the effect of the initialization itself is limited thanks to the unsupervised training. Therefore, the updated prediction obtained by UA^U gives a better indication of the model's quality after the trials have been processed. This updated model achieves an accuracy of 83.8% for 5 iterations and over 94% for 10 and 15 iterations. This indicates that the model is able to learn to decode the EEG, but also that it requires a significant amount of data to do so. The initial period, during which the classifier is unreliable, is the warm-up period. It prevents us from incorporating dynamic stopping into the UA method. Hence, to combine dynamic stopping with unsupervised learning, we have to eliminate this warm-up period.

3.2. Influence of transfer learning

The key to eliminating the warm-up period is transfer learning. The unsupervised static model TF makes use of transfer

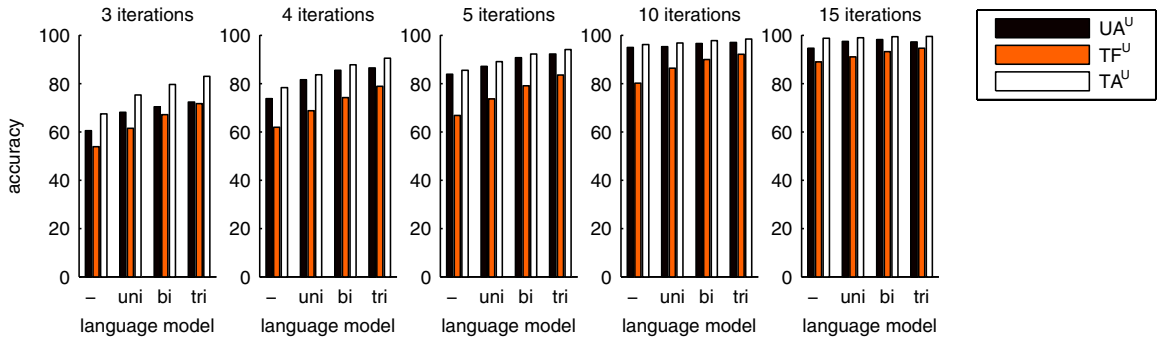

Figure 6. Accuracy obtained when the final updated prediction of the model after processing all trials is used.

Table 1. An overview of the spelling accuracy for a selected subset of methods.

Iter	UA	UA^U	TF	TF_3	TF_3^U	TA	TA^U	TA_3	TA_3^U	SF	SF_3	SF_3^U
3	24.6	60.5	53.9	66.1	71.7	62.5	67.5	73.8	83.0	74.5	85.3	89.4
4	42.2	73.7	61.9	74.1	78.9	72.6	78.3	82.1	90.5	81.8	89.5	92.8
5	58.6	83.8	66.8	79.0	83.5	78.4	85.5	87.0	94.1	85.6	92.2	94.5
10	78.4	94.9	80.1	89.4	92.2	91.9	96.2	95.0	98.4	93.4	96.3	97.3
15	82.1	94.6	89.0	92.3	94.6	97.1	98.8	97.9	99.5	96.9	97.7	98.1
DS	–	–	89.6	92.6	94.2	93.3	95.2	95.3	97.4	95.0	96.3	96.8

learning and is able to achieve 66.8%, 80.1% and 89% accuracy for 5, 10 and 15 iterations. The performance is still poor when the number of iterations is limited to 3 or 4, which opposes a high information transfer rate needed in practical situations. Further improvements can either be gained by further adaptation, the use of language models or dynamic stopping. In the following we will analyse the impact of these possible improvements first in isolation, then in combination.

3.3. Influence of language models

Table 1 contains the result on the symbol level for a subset of the unsupervised methods and baseline supervised models.

With a tri-gram language model, the fixed transfer model TF_3 predicts 79%, 89.4% and 92.3% of the symbols correctly for 5, 10 and 15 iterations. A supervised approach with a tri-gram language model, SF_3, outperforms TF_3 and correctly decodes 92.2%, 96.3% and 97.7% of the symbols. But for both approaches, this standard (simulated online) evaluation does of course not make use of the full capacity of the language models, as only information from preceding trials can be used for the prediction of the current trial.

By using the updated result after processing *all* trials, TF_3^U gets 83.5%, 92.2 and 94.6% symbols correct, and SF_3^U spells 94.5%, 97.3% and 98.1% of the symbols correctly. This clearly demonstrates that the forward–backward inference approach is able to exploit information from subsequent trials to improve the prediction for previous trials.

3.4. Influence of (unsupervised) adaptation

Even though including language models results in a minor performance boost, the biggest gain can be made by adding unsupervised adaptation. The adapted transfer model TA_3 achieves 87.0, 95.0% and 97.9% accuracy for 5, 10 and 15 iterations, which is actually very close to the supervised SF_3 model. The final re-estimate for TA_3^U obtains an accuracy of 99.5% for 15 iterations. Furthermore, using only 3 (4) iterations per trial, TA_3 spells 73.8% (82.1%) of the symbols correctly and the final re-estimate TA_3^U gets 83.0% (90.5%) of the symbols correct. To put this into perspective, the updated estimate of SF_3^U is only slightly better with 89.4% (92.4%) correct symbols.

3.5. Influence of dynamic stopping

Even though the TA method performs well for a limited number of iterations, the unsatisfactory performance of TF with three iterations raises doubts about the reliability of the classifier when only a few iterations are available. To avoid a premature decision, dynamic stopping methods come into play. The lowest row of table 1 provides results on the symbol level, if dynamic stopping is included. To verify its applicability to a zero-training ERP speller, we start by analysing DS-TF, which does not adapt to the novel subject's data. Consequently, each trial gives a direct estimate of the reliability of the transfer learning model. Remarkably, DS-TF uses a rather high number of iterations, 10.3 on average, which results in an accuracy of 89.6%.

However, the supervised DS-SF method attains a much higher accuracy (95.0%) and requires almost half the number of iterations (5.5). The difference between a supervised model and an unsupervised model diminishes by incorporating unsupervised adaptation. DS-TA is correct in 93.3% of the symbol predictions and requires 6.3 iterations per trial. Clearly, both the accuracy and decision-making speed are increased as the classifier adapts to the novel user. Furthermore, figure 7 shows that incorporating language statistics increases the accuracy and reduces the required number of iterations for all techniques. DS-TA_3 gets 95.3% of the symbols correct for 4.8 iterations, whereas the supervised DS-SF_3 is only slightly

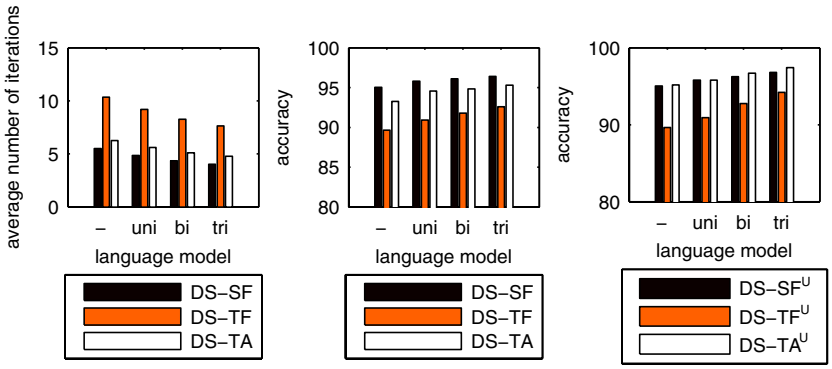

Figure 7. A visualization of the effect of language models on the required number of epochs and accuracy for dynamic stopping based methods.

better at 96.3% for 4 iterations. Finally, the updated predictions give a small performance improvement, 97.4% for DS-TA$_3^U$, which is slightly better than the supervised DS-SF$_3^U$ model at 96.8%.

3.6. Impact on practical BCI usage

The results suggest that combining dynamic stopping, transfer learning and language models results in a zero-training approach that is as reliable as a supervised model. Now we will focus on the communication rate.

The DS-TA approaches require slightly more iterations (0.8) per trial than DS-SF models, which amounts to a time difference of 1.7 s per trial. However, DS-SF uses a training set of 16 characters, which requires a recording of more than 10 min duration. Hence, to make up the time lost by the calibration, one would have to spell 345 trials (assuming that both methods perform equally well).

Furthermore, the updated predictions indicate that after processing more and more trials, the adaptive methods perform slightly better than the supervised counterparts. Hence, we can assume that the unsupervised method will be at least as reliable as the supervised model in the long run.

To conclude, we will compare the DS-TA and DS-SF approach using the SPM approximation from Schreuder and colleagues [27], which approximates the correctly spelled SPM in an application where the user has to correct errors using a backspace. This error measure is closely related to the utility metric from [7] and can be computed as follows: $\text{SPM} = \frac{2\times\text{accuracy}}{\text{time per trial}}$. We limit ourselves to the models without language models because it is the most general implementation and thus applicable to any kind of ERP-based BCI. Using this metric, DS-TA obtains 2.9 SPM and DS-SF achieves 3.4 SPM. Hence, during the time required for the calibration session, a user could already have spelled 29 symbols correctly when applying the DS-TA approach. For patients with a limited attention span, this improvement in fact would have a significant impact.

4. Conclusion

In this paper, we outlined how dynamic stopping, transfer learning and language statistics can be combined in a coherent probabilistic framework for zero-training ERP spelling. Our novel approach is shown to be clearly competitive to state-of-the-art supervised models. We include three levels of structure: the transfer learning approach by itself allows for an excellent zero-training ERP speller with dynamic stopping. The addition of unsupervised adaptation is able to further reduce the required number of iterations and increase the spelling accuracy dramatically. Additionally, the exploitation of n-gram language statistics can improve even further; note, however, that the practical applicability of language models might be limited by the desired BCI application.

In this work, we presented a simulation study, to be able to observe and quantify the improvements brought by our novel probabilistic framework. Given our highly encouraging findings, we will focus future work on an extensive evaluation of this approach in an online BCI spelling experiment. Moreover, the unsupervised transfer approach will be applied to auditory or tactile ERP paradigms.

Acknowledgments

This work was partly supported by the Federal Ministry of Education and Research (BMBF) and in part by the World Class University Program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology, under grant R31-10008; the Brain Korea 21 Plus Program through the National Research Foundation of Korea funded by the Ministry of Education and the DFG (SPP 1527); the Ghent University Special Research Fund under the BOF-GOA project Home-MATE and the BrainLinks-BrainTools Cluster of Excellence funded by the German Research Foundation (DFG, grant number EXC 1086). P-JK thanks the Bernstein Center for Neurotechnology for hospitality.

References

[1] Bishop C M 2007 *Pattern Recognition and Machine Learning Information Science and Statistics* 1st edn (Berlin: Springer)
[2] Blankertz B, Lemm S, Treder M S, Haufe S and Müller K-R 2011 Single-trial analysis and classification of ERP components—a tutorial *NeuroImage* **56** 814–25
[3] Blankertz B *et al* 2010 The Berlin brain–computer interface: non-medical uses of BCI technology *Front. Neurosci.* **4** 198

[4] Chen S and Goodman J 1999 An empirical study of smoothing techniques for language modeling *Comput. Speech Lang.* **13** 359–93

[5] Colwell K, Throckmorton C, Collins L and Morton K 2013 Transfer learning for accelerated p300 speller training *Proc. 5th Int. Brain–Computer Interface Meeting* ed J d R Millán, S Gao, G R Müller-Putz, J R Wolpaw and J E Huggins (Graz: Verlag der Technischen Universität Graz) pp 10–11

[6] Dähne S, Höhne J and Tangermann M 2011 Adaptive classification improves control performance in ERP-based BCIs *Proc. 5th Int. BCI Conf. (Graz)* pp 92–95

[7] Dal Seno B, Matteucci M and Mainardi L T 2010 The utility metric: a novel method to assess the overall performance of discrete brain–computer interfaces *IEEE Trans. Neural Syst. Rehabil. Eng.* **18** 20–8

[8] Dempster A P, Laird N M and Rubin D B 1977 Maximum likelihood from incomplete data via the EM algorithm *J. R. Stat. Soc.* B **39** 1–38

[9] Farquhar J and Hill J N Interactions between pre-processing and classification methods for event-related-potential classification *Neuroinformatics* **11** 175–92

[10] Farwell L A and Donchin E 1988 Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials *Electroencephalogr. Clin. Neurophysiol.* **70** 510–23

[11] Fazli S, Danóczy M, Schelldorfer J and Müller K-R 2011 L1-penalized linear mixed-effects models for high dimensional data with application to BCI *NeuroImage* **56** 2100–8

[12] Fazli S, Popescu F, Danóczy M, Blankertz B, Müller K-R and Grozea C 2009 Subject-independent mental state classification in single trials *Neural Netw.* **22** 1305–12

[13] Herweg A, Kaufmann T and Kübler A 2013 Using generic models to improve tactile ERP-BCI performance of low aptitude users *Proc. 5th Int. Brain–Computer Interface Meeting* ed J d R Millán, S Gao, G R Müller-Putz, J R Wolpaw and J E Huggins (Graz: Verlag der Technischen Universität Graz) pp 192–3

[14] Höhne J, Schreuder M, Blankertz B and Tangermann M 2011 A novel 9-class auditory ERP paradigm driving a predictive text entry system *Front. Neurosci.* **5** 99

[15] Jin J, Sellers E W, Zhang Y, Daly I, Wang X and Cichocki A 2013 Whether generic model works for rapid ERP-based BCI calibration *J. Neurosci. Methods* **212** 94–9

[16] Kaufmann T, Schulz S M, Grünzinger C and Kübler A 2011 Flashing characters with famous faces improves ERP-based brain–computer interface performance *J. Neural Eng.* **8** 056016

[17] Kindermans P-J, Verschore H and Schrauwen B 2013 A unified probabilistic approach to improve spelling in an event-related potential based brain–computer interface *IEEE Trans. Biomed. Eng.* **60** 2696–705

[18] Kindermans P-J, Verschore H, Verstraeten D and Schrauwen B 2012 A P300 BCI for the masses: prior information enables instant unsupervised spelling *Adv. Neural Inform. Process. Syst.* **25** 719–27

[19] Kindermans P-J, Verstraeten D, Buteneers P and Schrauwen B 2011 How do you like your P300 speller: adaptive, accurate and simple? *Proc. 5th Int Brain–Computer Interface Conf.* pp 148–9

[20] Kindermans P-J, Verstraeten D and Schrauwen B 2012 A Bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI *PLoS ONE* **7** e33758

[21] Krauledat M, Tangermann M, Blankertz B and Müller K-R 2008 Towards zero training for brain–computer interfacing *PLoS ONE* **3** e2967

[22] Lemm S, Blankertz B, Dickhaus T and Müller K-R 2011 Introduction to machine learning for brain imaging *NeuroImage* **56** 387–99

[23] Li Y, Guan C, Li H and Chin Z 2008 A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system *Pattern Recognit. Lett.* **29** 1285–94

[24] Lu S, Guan C and Zhang H 2009 Unsupervised brain computer interface based on intersubject information and online adaptation *IEEE Trans. Neural Syst. Rehabil. Eng.* **17** 135–45

[25] Chandrasekhara Panicker R, Puthusserypady S and Ying S 2010 Adaptation in P300 brain–computer interfaces: a two-classifier cotraining approach *IEEE Trans. Biomed. Eng.* **57** 2927–35

[26] Ryan D B, Frye G E, Townsend G, Berry D R, Mesa S, Gates N A and Sellers E W 2010 Predictive spelling with a P300-based brain–computer interface: increasing the rate of communication *Int. J. Hum. Comput. Interact.* **27** 69–84

[27] Schreuder M, Höhne J, Matthias T, Blankertz B, Haufe S, Dickhaus T and Tangermann M 2013 Optimizing event-related potential based brain–computer interfaces: a systematic evaluation of dynamic stopping methods *J. Neural Eng.* **10** 036025

[28] Schreuder M, Rost T and Tangermann M 2011 Listen, you are writing! speeding up online spelling with a dynamic auditory BCI *Front. Neurosci.* **5** 112

[29] Speier W, Arnold C, Lu J, Taira R K and Pouratian N 2012 Natural language processing with dynamic classification improves P300 speller accuracy and bit rate *J. Neural Eng.* **9** 016004

[30] Sutskever I, Martens J and Hinton G 2011 Generating text with recurrent neural networks *ICML: Int. Conf. on Machine Learning* pp 1017–24

[31] Tangermann M, Höhne J, Stecher H and Schreuder M 2012 No surprise—fixed sequence event-related potentials for brain–computer interfaces *EMBC'12: Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society* pp 2501–4

[32] Tangermann M, Schreuder M, Dähne S, Höhne J, Regler S, Ramsay A, Quek M, Williamson J and Murray-Smith R 2011 Optimized stimulation events for a visual ERP BCI *Int. J. Bioelectromagn.* **13** 119–20

[33] Townsend G, LaPallo B K, Boulay C B, Krusienski D J, Frye G E, Hauser C K, Schwartz N E, Vaughan T M, Wolpaw J R and Sellers E W 2010 A novel P300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns *Clin. Neurophysiol.* **121** 1109

[34] Treder M S and Blankertz B 2010 (C)overt attention and visual speller design in an ERP-based brain–computer interface *Behav. Brain Funct.* **6** 28

[35] Verschore H, Kindermans P-J, Verstraeten D and Schrauwen B 2012 Dynamic stopping improves the speed and accuracy of a P300 speller *ICANN'12: Int. Conf. on Artificial Neural Networks and Machine Learning* (Berlin: Springer) pp 661–8