submitted to the International Joint Conference on Neural Networks (IJCNN 2015),
July 12 - 17, 2015, Killarney, Ireland.

Reward-based online learning in non-stationary environments: adapting a P300-speller with a "Backspace" key

Emmanuel Daucé
Ecole Centrale Marseille
INSERM UMR_S 1106
Marseille, France
Email: emmanuel.dauce@centrale-marseille.fr

Timothée Proix
Aix-Marseille Université
INSERM UMR_S 1106
Marseille, France

Liva Ralaivola
Aix-Marseille Université
CNRS UMR 7279
Marseille, France

***Abstract*—We adapt a policy gradient approach to the problem of reward-based online learning of a non-invasive EEG-based "P300"-speller. We first clarify the nature of the P300-speller classification problem and present a general regularized gradient ascent formula. We then show that when the reward is immediate and binary (namely "bad response" or "good response"), each update is expected to improve the classifier accuracy, whether the actual response is correct or not. We also estimate the robustness of the method to occasional mistaken rewards, i.e. show that the learning efficacy may only linearly decrease with the rate of invalid rewards. The effectiveness of our approach is tested in a series of simulations reproducing the conditions of real experiments. We show in a first experiment that a systematic improvement of the spelling rate is obtained for all subjects in the absence of initial calibration. In a second experiment, we consider the case of the online recovery that is expected to follow unforeseen impairments. Combined with a specific failure detection algorithm, the spelling error information (typically contained in a "backspace" hit) is shown useful for the policy gradient to adapt the P300 classifier to the new situation, provided the feedback is reliable enough (namely having a reliability greater than 70%).**

***Keywords*—Online learning, Reinforcement learning, Policy gradient, Brain-Computer Interfaces, P300 speller**

I. Introduction

We consider the case of embedded classifiers having to interact in real time with their environment. Embedded classifiers (in vehicles, planes, robots...) have to deal with large amounts of data vectors sampled from the environment, and take their decisions from these samples. In order to build such classifiers, a training session is generally done prior to actual use. This initial training issues a set of parameters that are then considered fixed for the remaining time. Brain Computer Interfaces are an example of such embedded classifiers. The objective of brain-computer interfaces (BCI's) is to analyze in real-time electro-encephalographic (EEG) signals recorded at the surface of the scalp in order to control a device (mouse, keyboard, wheelchair,...). This problem has straightforward applications, at first, toward helping disabled people for their communication [1] and displacements [2], but also more generally for game remote control, assisted driving, neurofeedback, etc. From a general standpoint, the problem consists in collecting samples of EEG activity and trying to *classify* them in different categories reflecting the "state of mind" of the subject at the moment of the observation. The better the classification, the more effective the interface. The generic name "BCI" encompasses different EEG-based communication protocols like controlling a pointer on a screen using motor imagery (the "Graz" task [3]), text typing using event-related potentials ("P300" speller) [1], "brain switch" [4], etc., with corresponding dedicated pre-processing [5], [6], [7] and classification and/or regression techniques [3], [8], [9]. We consider in this paper the case of grid-based P300 spellers, where the subject faces a screen with a 6×6 grid of letters and numbers and is asked to focus his attention on the symbol he wants to spell out.

After a classifier has been trained, the subject is expected to handle the interface and be capable of spelling words autonomously. However, many changes are expected to take place during subsequent use and the quality of the signal (and thus the accuracy of the interface) is known to progessively degrade over time : some electrodes may accidentally be displaced or unstick from the scalp, the conductivity of the scalp itself may change during the experiment, and the level of attention of the subject may evolve, etc. Moreover, experimental conditions are difficult to reproduce and day-to-day recalibration is often necessary. This problem of adapting the parameters while the device is used is known as the "online learning" problem, relevant when the statistics of the input change over time, i.e. when the input is *non-stationary* [10], or when the input data is abundant and high-dimensional [11].

The problem of BCI non-stationarity is generally adressed using an unsupervised expectation-maximization (EM) procedure [12], [13], [14], which is not proven to always converge to a consistent classifier. We develop in this paper a first attempt to adapt the reinforcement learning methodology [15] to the context of an adaptive BCI P300 speller, where the "reward" may take the form of a scalar representing the agreement of the subject regarding the response of the device. A reward is of course less informative than the true response, but, in counterpart, generally cheaper to obtain. In a P300-speller setup, two solutions should be considered for that purpose: (i) use the "error negativity" event-related potential (ERP) following a wrong classifier's response, where a detection rate from 60 to 90% is reported in the literature [16], [17], [18]. In that case, a specific classifier is needed for detecting the error

negativities in the EEG, and as such a specific training session should take place prior to the spelling session; (ii) dedicate a specific symbol to the case when the previous character was incorrectly spelled to detect the subject disagreement regarding previous response [19]. The "Backspace" key of the standard computer keyboards is interesting to consider from this perspective. It may indeed provide an information (or a guess) about the correctness of the previous spelling, and thus allow to decipher between valid letters and invalid letters in the current series of spelled characters. A minimal spelling accuracy is then needed for this approach to be effective, which means that a training session may also take place prior to the spelling session.

In this paper, we consider the second case and look in section II at the principles underlying reward-based learning in the P300-speller case, adapting the classical policy gradient appoach [20] to that context. In section III, we look at the gradient estimator, in the particular case where the rewards are binary, and we prove (i) the estimator to systematically head toward response improvement when correct responses are positively marked and incorrect responses negatively marked, and (ii) to be robust to occasional misleading rewards, which is typically the case when the rewards are driven by a "backspace" key. Then, in order to validate our approach, we simulate in section IV two different adaptive P300-speller setups from a dataset coming from real P300-speller experiments. In the first numerical experiment, the learning is made "from scratch" (i.e. without initial training or calibration), while in a second numerical experiment, an initial calibration is done and the policy gradient is combined with a specific failure detection algorithm in order to adapt the classifier to unexpected signal breakout.

II. Toward reward-based P300-spellers

A. Related work

Online learning with partial feedback is often referred a the "contextual bandit" problem, or "bandit with side information", that aims at finding online strategies in order to minimize the "regret" when one has to find a best choice out of K possibilities by trial and error. Popular solutions use parametrized random policies where every possible choice is described by its expected gain and the number of visits [21]. Although several variants have been adapted to the continuous context case [22], [23], [24], they do not fit to the problem we consider here. Moreover, despite having mathematically exact upper regret bound, they should not show in practice fast enough convergence rates for applying to real-time online learning in a dynamic context.

B. The "P300-speller" classification problem

The so-called "oddball paradigm" is well-established protocol which has been developed in electrophysiology in order to identify the subject's reaction to an unexpected stimulus taking place in a sequence of monotonic stimuli [1]. A specific Event Related Potential (ERP) can be measured in the EEG around 300 ms after the stimulus onset. This response (called the "P300") is generally considered as the signature of the surprise of the subject regarding the expected stimulus. The problem is then to identify this particular ERP (the "oddball" ERP) in a set of K observations. If we note $\underline{\mathbf{x}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_K) \in \mathcal{X}^K$ the set of K observations, where $\mathcal{X}$ is the feature vectors space, the problem is to identify the "target" within a set having multiple inputs belonging to the "non-target" category and only one input belonging to the "target" category. This problem can be seen as a one-*among*-all classification problem[1]. Noting P^+ the target vectorial distribution and P^- the non-target vectorial distribution, we propose the following generative model: each observation $\underline{\mathbf{x}}$ can be seen as the result of a random draw from a uniform mixture of K distributions $P_1, ..., P_K$, with $P_1 = (P^+, P^-, ..., P^-)$, $P_2 = (P^-, P^+, P^-, ..., P^-)$, ..., $P_K = (P^-, ..., P^-, P^+)$, each P_k standing for a sequence of K independently drawn feature vectors having the k^{th} vector for target. The uniform prior reflects the uniform probability among the target location in the sequence.

With such uniform priors, the posterior probability of identifying location k as the target, given observation $(\boldsymbol{x}_1, ..., \boldsymbol{x}_K)$ is easily shown from Bayes formula to be:

$$P(k|\boldsymbol{x}_1, ..., \boldsymbol{x}_K) = \frac{\frac{P^+(\boldsymbol{x}_k)}{P^-(\boldsymbol{x}_k)}}{\sum_l \frac{P^+(\boldsymbol{x}_l)}{P^-(\boldsymbol{x}_l)}} \tag{1}$$

In the linear-Gaussian case, where P^+ and P^- are multivariate Gaussian distributions of respective mean $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$, with shared covariance $\boldsymbol{\Sigma}$, the previous formula simplifies to:

$$P(k|\boldsymbol{x}_1, ..., \boldsymbol{x}_K) = \frac{\exp(\boldsymbol{x}_k \boldsymbol{w}^T)}{\sum_l \exp(\boldsymbol{x}_l \boldsymbol{w}^T)} \tag{2}$$

where $\boldsymbol{w}$ can be shown to be equal to $(\boldsymbol{\mu}^+ - \boldsymbol{\mu}^-)\Sigma^{-1}$. The discriminating manifold $\boldsymbol{x}\boldsymbol{w}^T = 0$ is similar to the manifold obtained in the binary classification problem, the only difference being the number of samples considered (one sample in the binary classification case, multiple samples in the "P300-speller" classification case). Despite the categorical nature of the response, the P300-speller classifier is structurally closer to the binary case than to the multiclass case.

C. Stochastic classifier

We consider the reinforcement learning framework, where different responses must be explored at random before the classifier can estimate which of them is expected to bring the highest reward. The response being obtained from a random draw, the rule that determines the probabilities associated to each possible response is called the policy. In our case, consistently with eq. (2), the policy relies on a vector of parameters $\boldsymbol{w}$ that is to be compared with every observation vector from the set $\underline{\mathbf{x}}$ through the scalar products $\langle \boldsymbol{w}, \boldsymbol{x}_k \rangle$'s that are expected to be higher when $\boldsymbol{x}_k$ is a target than when it is not. The actual response $y \in \{1, ..., K\}$ is drawn from a multinomial distribution relying on the "π-scores" (softmax choice):

$$\forall k \in \{1, ..., K\}, \pi(\underline{\mathbf{x}}, k; \boldsymbol{w}) = \frac{\exp\langle \boldsymbol{w}, \boldsymbol{x}_k \rangle}{\sum_l \exp\langle \boldsymbol{w}, \boldsymbol{x}_l \rangle} \tag{3}$$

so that that $\pi(\underline{\mathbf{x}}, k; \boldsymbol{w})$ is the probability of having $y = k$ given $\underline{\mathbf{x}}$.

[1] not to be mixed up with the "one-vs-all" classification setup.

D. Learning problem

After the response y is drawn out, a scalar $r(\underline{\mathbf{x}}, y)$ (the "reward") is read from the environment. In essence, the reward quantifies the achievement of the current trial. The reward expectation $E(r)$ represents the global achievement of the policy, given by the integral of the rewards obtained for every observation set $\underline{\mathbf{x}}$ and every choice k, given their probability of appearance:

$$E(r) = \int_{\mathcal{X}^K} \Big(\sum_{k=1}^{K} r(\underline{\mathbf{x}}, k) \pi(\underline{\mathbf{x}}, k; \boldsymbol{w}) \Big) p(\underline{\mathbf{x}}) d\underline{\mathbf{x}}$$

where the distribution $p(\underline{\mathbf{x}})$ is not explicitely given. The objective is to find the best parameters $\boldsymbol{w}$ for the policy to maximise the reward expectation, i.e.: $\max_{\boldsymbol{w}} E(r)$. We additionally consider a regularization term that is expected to promote a small norm for the model and priorize the most recently seen examples in an online setup [10]. The optimization problem becomes:

$$\max_{\boldsymbol{w}} \mathcal{G} = \max_{\boldsymbol{w}} E(r) - \frac{\lambda}{2} \|\boldsymbol{w}\|^2 \tag{4}$$

where $\mathcal{G}$ is the objective function and λ the regularization parameter.

In the absence of a model, the solution of eq. (4) can be approached by trying to cancel the gradient of $\mathcal{G}$ through a stochastic gradient *ascent*. The policy gradient is a general purpose reward-based algorithm we adapt here to the online "P300-speller" classification case. Following [20], the regularized policy gradient can be shown to obey to:

$$\nabla_{\boldsymbol{w}} \mathcal{G} = E(r \nabla_{\boldsymbol{w}} \ln(\pi) - \lambda \boldsymbol{w}) \tag{5}$$

Starting from scratch, the update procedure is expected to refine the model trial after trial using the local estimator $\boldsymbol{g}(\underline{\mathbf{x}}, y) - \lambda \boldsymbol{w}$, with $\boldsymbol{g}(\underline{\mathbf{x}}, y) = r(\underline{\mathbf{x}}, y) \nabla_{\boldsymbol{w}} \ln \pi(\underline{\mathbf{x}}, y; \boldsymbol{w})$, so that the rewards should be maximized in the long run. From the derivation of $\ln \pi$ according to $\boldsymbol{w}$, we obtain the following expression:

$$\boldsymbol{g}(\underline{\mathbf{x}}, y) = r(\underline{\mathbf{x}}, y) \left(\boldsymbol{x}_y - \sum_k \pi(\underline{\mathbf{x}}, k; \boldsymbol{w}) \boldsymbol{x}_k \right) \tag{6}$$

The regularized online policy gradient ascent update can be defined the following way: at every time t, after reading $\boldsymbol{x}_t$, y_t and r_t, increment $\boldsymbol{w}$ with $\eta \left(\boldsymbol{g}(\underline{\mathbf{x}}, y) - \lambda \boldsymbol{w} \right)$, where η is the *learning rate*, i.e.:

$$\boldsymbol{w}_t = (1 - \eta\lambda) \boldsymbol{w}_{t-1} + \eta r_t \left(\boldsymbol{x}_{y_t, t} - \sum_{k=1}^{K} \pi(\underline{\mathbf{x}}, k; \boldsymbol{w}_{t-1}) \boldsymbol{x}_{k,t} \right) \tag{7}$$

III. Analysis

A. Gradient estimator in the binary reward case

The policy gradient estimator (6) can be analyzed in the binary reward case, i.e when $r(\underline{\mathbf{x}}, y)$ takes only two values: r^+ when the response is correct and r^- when the response is incorrect. Using a symmetry argument, we only consider the case $\underline{\mathbf{x}} \in \mathcal{C}_1$, i.e. $\boldsymbol{x}_1 \sim P^+$ and $\forall k > 1, \boldsymbol{x}_k \sim P^-$, for the analysis. Considering that $r(\underline{\mathbf{x}}, 1) = r^+$ and $r(\underline{\mathbf{x}}, k) = r^-$ for k>1, we have:

$$\boldsymbol{g}(\underline{\mathbf{x}}, y) = (\mathbf{1}_{\{y=1\}} r^+ + \mathbf{1}_{\{y \neq 1\}} r^-) \left(\boldsymbol{x}_y - \sum_k \pi(\underline{\mathbf{x}}, k) \boldsymbol{x}_k \right)$$

Then, noting that:

$$E_{\underline{X}, Y}(\boldsymbol{g}) = E_{\underline{X}} \left[E_{Y|\underline{X}}(\boldsymbol{g}) \right]$$

we look at the conditional expectation of the gradient $E_{Y|\underline{X}}(\boldsymbol{g})$, i.e. we try to estimate the general direction followed when the set of observations $\underline{\mathbf{x}}$ is given.

Let us introduce few additional notations in order to simplify the writing:

- Target: $\boldsymbol{x}^+(\underline{\mathbf{x}}) = \boldsymbol{x}_1$
- Non-targets weighted average : $\boldsymbol{x}^-(\underline{\mathbf{x}}) = \sum_{k>1} \tilde{\pi}(\underline{\mathbf{x}}, k) \boldsymbol{x}_k$, with $\tilde{\pi}(\underline{\mathbf{x}}, k) = \frac{\pi(\underline{\mathbf{x}}, k)}{1 - \pi(\underline{\mathbf{x}}, 1)}$
- Difference (local discriminant vector): $\boldsymbol{\Delta}(\underline{\mathbf{x}}) = \boldsymbol{x}^+(\underline{\mathbf{x}}) - \boldsymbol{x}^-(\underline{\mathbf{x}})$.

With some reordering, it can be shown that:

$$E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y)) = (r^+ - r^-) \pi(\underline{\mathbf{x}}, 1)(1 - \pi(\underline{\mathbf{x}}, 1)) \boldsymbol{\Delta}(\underline{\mathbf{x}}) \tag{8}$$

i.e. the local *direction* of the gradient is the local discriminant vector $\boldsymbol{\Delta}(\underline{\mathbf{x}})$.

By principle, as soon as $r^+ > r^-$, the weights update is expected to promote the correct responses (i.e. those associated with r^+). Then, remarking that: $\mathbb{P}(r = r^+ | \underline{\mathbf{x}}) = \pi(\underline{\mathbf{x}}, 1)$, we write:

$$\begin{aligned} E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y)) = \pi(\underline{\mathbf{x}}, 1) E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y) | r = r^+) \\ + (1 - \pi(\underline{\mathbf{x}}, 1)) E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y) | r = r^-) \end{aligned}$$

and identify:

$$\begin{aligned} E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y) | r = r^+) &= r^+ (1 - \pi(\underline{\mathbf{x}}, 1)) \boldsymbol{\Delta}(\underline{\mathbf{x}}) \\ E_{Y|\underline{X}}(\boldsymbol{g}(\underline{\mathbf{x}}, y) | r = r^-) &= -r^- \pi(\underline{\mathbf{x}}, 1) \boldsymbol{\Delta}(\underline{\mathbf{x}}) \end{aligned}$$

If the response is correct *and* $r^+ > 0$, the gradient direction $\boldsymbol{\Delta}(\underline{\mathbf{x}})$ is followed, giving more chance to the correct choice in the next trials. Interestingly, if the response is incorrect *and* $r- < 0$ (i.e. a negative mark is given to wrong choices), the same direction is followed, so that the chance of making correct choices in the future is also enhanced. Each update is thus expected to improve the rate of correct responses, whether the actual response is correct or not.

B. Noise-tolerant learning

We now consider the case where the reward value is corrupted with a uniform noise p_{valid}, i.e. at each trial, the chance of sending the correct reward is p_{valid}, the chance of sending the opposite reward is $1 - p_{\text{valid}}$. Then, the gradient expectation becomes:

$$\begin{aligned} E\left(\boldsymbol{g}(\underline{\mathbf{x}}, y)\right) =& p_{\text{valid}} (r^+ - r^-) E\left[\pi(\underline{\mathbf{x}}, 1)(1 - \pi(\underline{\mathbf{x}}, 1)) \boldsymbol{\Delta}(\underline{\mathbf{x}})\right] \\ &+ (1 - p_{\text{valid}}) (r^- - r^+) E\left[\pi(\underline{\mathbf{x}}, 1)(1 - \pi(\underline{\mathbf{x}}, 1)) \boldsymbol{\Delta}(\underline{\mathbf{x}})\right] \\ =& (2 p_{\text{valid}} - 1)(r^+ - r^-) E\left[\pi(\underline{\mathbf{x}}, 1)(1 - \pi(\underline{\mathbf{x}}, 1)) \boldsymbol{\Delta}(\underline{\mathbf{x}})\right] \end{aligned}$$

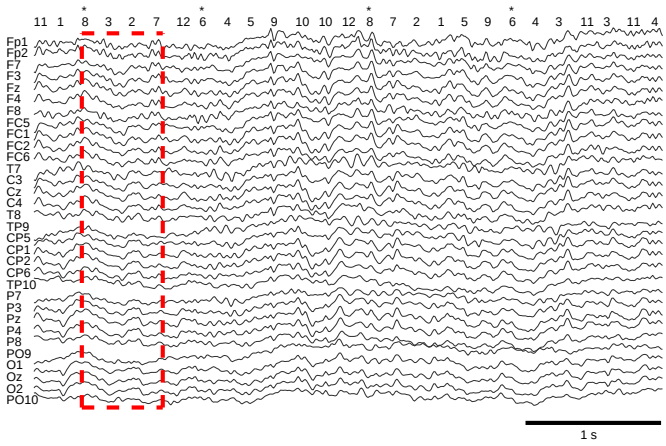

Fig. 1. 32-channels EEG signal excerpt. Row or column tags are shown on top and stars denotes target row or column (here target row = 6, target column = 8). The dashed square indicates an example 600 ms EEG sample taken after flashing the grid at position 8 (2nd column).

In this equation, we recognize the typical $(2p_{\text{valid}} - 1)$ term that appears when analyzing algorithms learning from noisy labels (see, e.g. [25], [26], [27]); it sets the limiting regimen, i.e. $p_{\text{valid}} > 0.5$, where learning can take place. Importantly, the norm of the estimator (and thus the speed of the learning process) only displays a linear decrease with the rate of misleading rewards, which ensures a correct but slowed down convergence even with a significant proportion of reward errors.

IV. Numerical experiments

A. Dataset and preprocessing

The EEG dataset we use comes from a P300 experiment reported in [28]. The data consists of 20 files, one file per subject measuring the brain activity during a P300-speller experiment where each subject had to spell out mentally 220 letters. For a given letter, rows and columns are flashed in random order in order to enhance the "surprise" when the target row or column is illuminated. Each row and each column of the grid is flashed several times before taking a decision. In the considered dataset, each row and column was flashed 5 times per trial. The stimulus duration was 100 ms, the inter-stimulus interval was also 100ms, so that the total SOA[2] was 200 ms.

A 32-channels EEG signal was recorded during the whole experiment, sampled at 100 Hz. The whole experiment is divided in sequences of 12×5 flashes, corresponding to one letter spelling trial. For each series of 60 flashes, a [1-20] Hz bandpass filter is applied. Then for each flash time t, a 600 ms subsample $\mathbf{s}_{t:t+600\text{ms}}$ is taken, a common reference average substraction is applied, followed by a channel-by-channel normalization. Such a sample excerpt is presented on figure 1. The sample is then vectorized and put in its category $\in 1, ..., K$ (row or column number). With a 100Hz sampling, the dimension of each data vector is $32 \times 60 = 1920$. Then a set of multiple ERP observations is constructed the following way: for $k \in 1..12$, calculate class average $\boldsymbol{x}_k$ and normalize it. Finally, construct 2 multi-ERP set, i.e. $\underline{\mathbf{x}}^{\text{row}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_6)$ and $\underline{\mathbf{x}}^{\text{column}} = (\boldsymbol{x}_7, ..., \boldsymbol{x}_{12})$.

[2]Stimulus Onset Asynchrony.

B. Batch training

In the standard approach, EEG signal are recorded during a training session and analyzed prior to free spelling (i) to construct a spatial filter that recombines the electrodes in order to enhance the P300 signal (e.g. xDAWN filter [6]) and (ii) to build a classifier from the filtered data (e.g. LDA [8] or Naïve Bayes [7], [28] classifiers). Once the spatial filter is determined, every EEG sample $\mathbf{s}$ is recombined into a *filtered* EEG sample $\mathbf{s}'$, whose dimension is reduced regarding the initial data, and used for the batch training of the classifier. In the original P300-speller experiment, the training set was composed of 100 trials, and the test set composed of 120 trials. Here, consistently with our second numerical experiment (section IV-D), we consider a training set of 25 trials only, with 195 trials in the test set. We verified by cross-validation that both Naïve Bayes and LDA classifiers can achieve on average 85% spelling accuracy on the test set, which is consistent with state-of-the art results for this number of repetitions.

C. Learning from scratch

During free spelling, the classifier has to identify in the set of ERPs the index of the row and column where the P300 ERP took place, so that the resulting letter is at the intersection of this row and this column. In a first numerical experiment, learning is made from scratch, without prior information, i.e. the initial weights vector $\boldsymbol{w}$ is null and a raw signal is used (no spatial filter). In order to estimate the improvement of our classifier, we need to run a series of experiments, record the responses of the classifier and compare them to the expected ones. For a given value of the hyperparameters η and λ, and for every subject, we run 1000 different simulations. As the *order* of the examples matters in the online classifier build-up, each simulation corresponds to a different random shuffle of the initial 220 trials. For each sequence of randomly shuffled letters, we then apply the online update (7), with rewards generated by a simple comparison between the expected letter and the classifier's response. For each subject, the rate of correct spelling is then calculated (at each trial) as the average over the 1000 experiments.

For the rewards, we use the values $r^+ = K - 1$ and $r^- = -1$, that can be shown to provide an optimal baseline in the first steps of the gradient ascent (demonstration not given). We compare in our experiments two variants of the policy gradient algorithm. In a first series, the response y is made according to a stochastic "softmax" policy, and in a second series, it is made according to a deterministic "argmax" policy. In order to fairly compare the two methods, we systematically calculate the final spelling accuracy for different values of η and λ ranging from 10^{-5} to 10 using grid search, and find markedly different optimal values for the two cases, i.e. $\eta = 1$ and $\lambda = 10^{-3}$ in the softmax case and $\eta = 10^{-1}$ and $\lambda = 10^{-1}$ in the argmax case.

We compare in figure 2 the different learning curves obtained in our series of 20×1000 simulations. Both average learning curves show a monotonical increase, from $\frac{1}{36}$ (random response) to an average accuracy of $\simeq$ 60-65% after 220 trials,

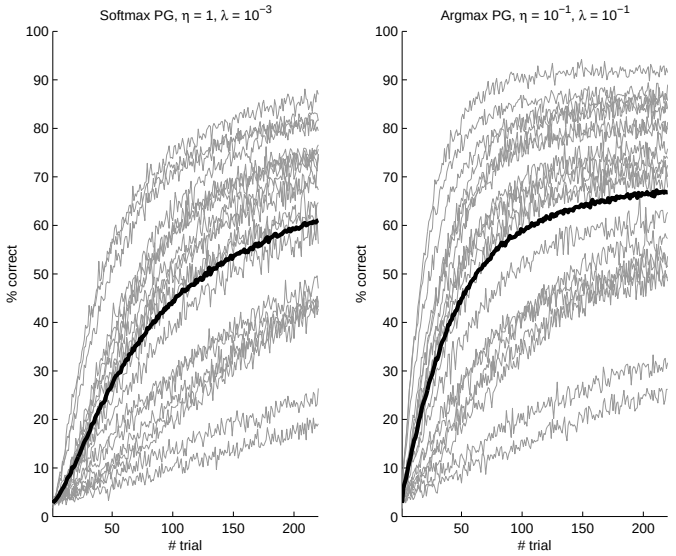

Fig. 2. Spelling improvement during P300-speller experiment (mean over 20 subjects). Thin gray lines: individual improvement. Thick line: average improvement. Left: Softmax policy gradient. Right: Argmax policy gradient.

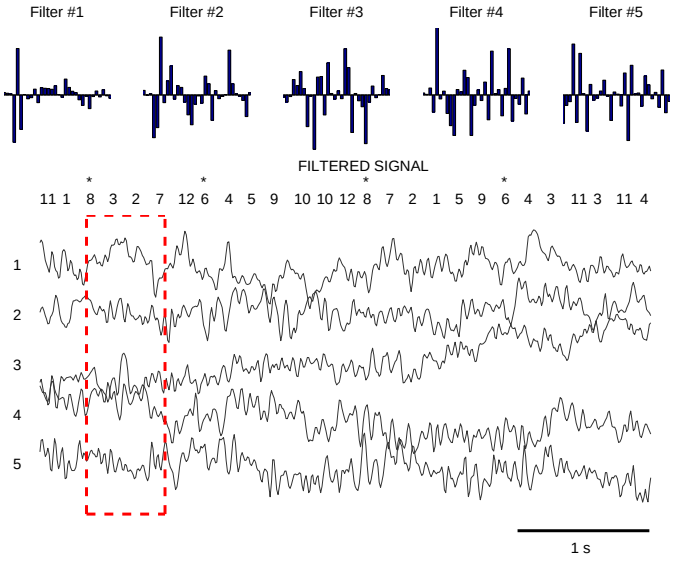

Fig. 3. xDAWN Spatial filter applied on the EEG signal after a 25 trials training session. The 5 spatial filters (upper bars) define a new signal that differently combine the initial 32 electrodes. The observation signal is now reduced to 5 channels, each channel having the role of a "virtual electrode" (lower figure), where the dashed square indicates a 600 ms sample following a flash on column 8 (same sample as in figure 1).

with a slight advantage for the argmax case, and a tendency toward continuing improvement for $t > 220$. Consistently with our expectations (see section III-A), the steep initial improvement indicates a robust gradient following from the very first steps, despite mostly negative rewards and occasional misleading rewards (when only the row – or only the column – is correct). However, despite final slopes suggesting a continuing tendency toward improvement, the final rates does not attain the 85% rate obtained in the standard approach when combining a spatial filter and a supervised classifier on a small subset of the training set. Moreover, even if a significant improvement is observed for every subject, the inter-subject variability remains quite high, with final spelling accuracy peaking to almost 90 % for the best subjects, but hardly crossing 20% for few weaker ones (this discrepancy between subjects being more generally a chronic problem in BCI's). Despite its solid converging capabilities, policy gradient from scratch does not appear fast enough when considering this specific application. We thus consider a more realistic experiment where an initial calibration is followed by an online improvement procedure.

D. On-line adaptation

We now consider the classical combination of a spatial filter and a classifier (see sec. IV-B). In the previous experiments, the inter-subjects variability remains quite high, with weaker subjects hardly reaching 30% correct spelling rate. Moreover, the rate attained after 220 trials approaches, but doesn't attain the expected rate of correct spelling for this data set (around 85 %), obtained when combining a spatial filter with a supervised classifier. We thus look here how this combination of spatial filter + classifier can be adapted online in the context of the reward-based approach. In the following, we note those filtered data vectors $\boldsymbol{x}'$ (see figure 3). On the dataset we consider here, this filter training, followed by a classifier training allows to reach a rate of correct spelling of 85% with only 25 trials in the training session, which of course challenges the genuine online approach.

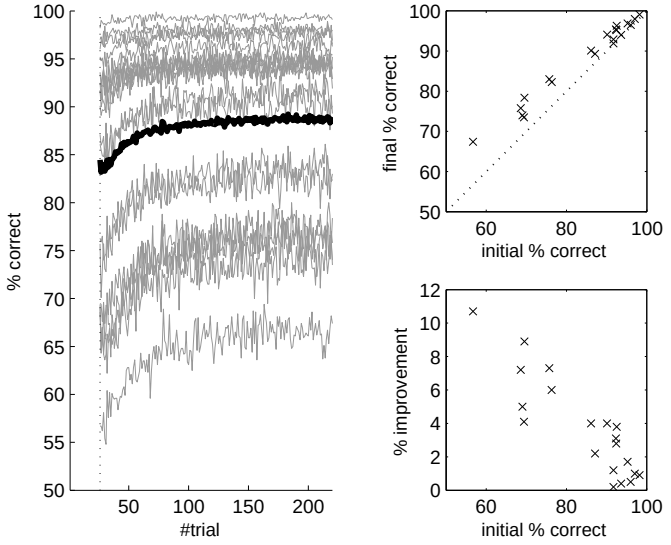

Fig. 4. Left: classification improvement after a 25-trials training session (mean over 20 subjects), using linear argmax policy gradient (with $\eta = 10^{-1}$, $\lambda = 10^{-1}$ and $p = 5$). Thin gray lines: individual improvement. Thick line: average improvement. The right figures compare the final performance to the initial one, subject by subject, in absolute values (upper right) and in difference (lower right).

Then, the online improvement is done at each trial of the test session (when only rewards are returned to the classifier). We present in figure 4 the average rate of correct spelling for the trials following a 25-trials training session. The initial performance (around 85%) is similar to the performance of a standard LDA classifier. Then, a significant improvement

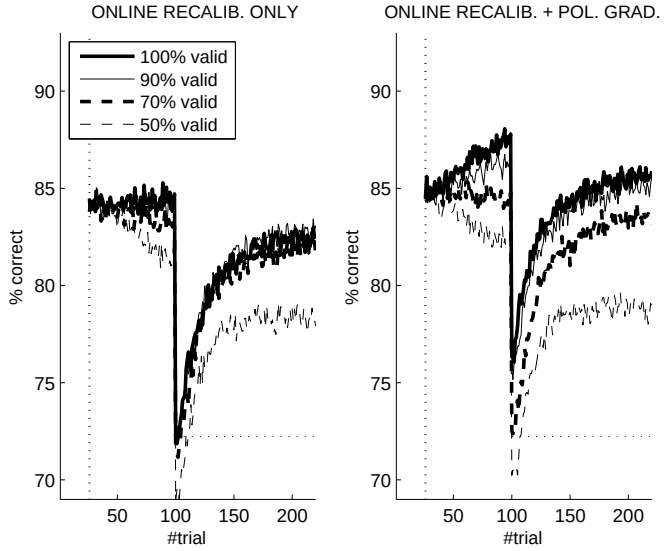

Fig. 5. Average spelling rate over 20 subjects, with different reward reliabilities, and electrode break at trial # 101. Left: online recalibration only. Right: online recalibration + argmax policy gradient. $\eta = 10^{-1}$, $\lambda = 10^{-1}$. The horizontal dots show the final spelling rate in the absence of adaptation.

is observed, approaching 90% correct in the last trials of the session. Interestingly, this improvement can be compared between subjects: the performance improvement appears negligible for the stronger subjects (around 2 % when the initial rate is > 80%) while it is significantly better (from 6 to 10%) for subjects whose initiation rate is < 80%. This differential improvement appears interesting for the purpose of reducing the inter-subject variability, since the weaker subject in our set now attains around 66% achievement at the end of the session (while only 56% at the end of the training).

E. On-line recovery

In the non-stationary case, our online adaptive setup requires that both the spatial filter and the classifier adapt during use. In order to test this online recovery capability, we build a specific simulation setup in which we mimic an unexpected change taking place at trial number 101. We simply replace the EEG signal from one electrode taken at random by a white noise, in order to artificially diminish the rate of successful recognition. We moreover test the robustness of our approach to erroneous rewards, while manipulating the rate of misleading rewards, from 100% valid rewards to only 50% valid. At each trial, a random draw is done on the reward reliability p_{valid}. If a failure is obtained, the reward sent to the algorithm is the reverse of what it should be, i.e. r^- instead of r^+ and vice versa.

Figure 5 presents the main results obtained in this setup. The EEG data comes from the same experiment. Only the adaptation methods change. The left figure presents original results relying on a simple and effective algorithm (called "reward-based online recalibration"). In principle, it is possible to recalibrate the device every time a letter is not followed by a backspace, which is obviously too resource-consuming. Here we propose to recalibrate the system only when a significant performance drop is observed. We developed a specific algorithm where (i) a training set, containing only the most recent examples, is maintained during use and (ii) the device recalibration is allowed only when the rate of correct spelling is too low (typically 3 successive failures) [19]. This algorithm provides a coarse online estimation of the spelling performance, based on the number of failures observed in the three most recent trials. The device update is thus allowed when an unexpected series of three failures is observed. Only the EEG signal and the labels from the 25 most recent letters for which a positive reward was issued are used to recalibrate the device. The figure presents the average correct spelling rates for the trials following an initial 25-letters training session. The initial performance (around 85%) is similar to the performance of a standard LDA classifier (with this number of repetitions), which of course challenges the genuine online approach presented in the previous experiment. The effect of a single electrode break (out of 32) appears quite strong, with a big drop from almost 85% correct to less than 75% correct. As expected, a decrease in reward reliability degrades the adaptation and recovery capabilities. The left figure (online recalibration only) indicates that good recovery capabilities are obtained with even 30% misleading rewards. Only with 50% misleading rewards the performance degrades, but still can recover from the artificial electrode breakout, showing the robustness of the xDAWN filter to false positives (since only positive rewards are considered in the filter and classifier updates). It is interesting to remark that the sole online recalibration seems capable of a recovery up to 78% correct spelling when half of the rewards are fooling the process. This counter-intuitive result can be explained by the fact that in a context of a high correct spelling rate, most of the false rewards are false negative rewards, whose effect is weaker than the true positives, as $|r^-| << |r^+|$ in our setting.

When considering both online recalibration and policy gradient (right figure), a significant improvement is obtained both during the first steps following the initial calibration and after the electrode break-out. This specific effect of the policy gradient is only sensible when the rate of misleading rewards is less than 30%, otherwise no significant improvement is observed. This can be considered an empirical limit to the policy gradient effectivity in the context of the P300 speller: if the rate of valid rewards is less than (or equal to) 70%, a simple occasional recalibration may be sufficient for online adaptivity to environmental changes. In the other case, policy gradient adaptation is worth using in addition to the online recalibration. This limit is consistent with our theoretical estimate of policy gradient robustness, since the lack of reward reliability may come from two distinct causes, namely the row and column reward sharing and the reward reliability p_{valid} itself[3]. At last, the combination of policy gradient and online recalibration allows to grab approximately 3% when compared to the sole online recalibration, with a final rate approaching 86% (to be compared to the 88-89% that can be obtained in the stationary case). The gain of adaptivity obtained by online recalibration is not at the expense of the improvement due to the policy gradient. The benefits of the two approaches thus seem to appropriately add up.

[3]For a single row (or column) classification rate $\rho = \sqrt{0.85} \simeq 0.92$, and $p_{\text{valid}} = 0.7$, the effective reward reliability reaches $(1-\rho+\rho^2)p_{\text{valid}} \simeq 0.65$, which approaches the absolute limit of 0.5 reward reliability we stated in section III-B.

V. Conclusion

We have presented an application of a policy gradient algorithm to the problem of online multi-sample classification, where the multinomial choice means identifying the "oddball" in a set of K inputs. Under this approach, we have shown that every trial should contribute to the classifier improvement, provided the return of correct choices is positive and the return of incorrect choices is negative. We have also given evidence of a good robustness to unreliable rewards (when a certain proportion of bad response are erroneously rewarded and/or correct responses erroneously dismissed). The consistency of our update formula has been tested on a P300-speller BCI dataset. We have considered different experimental conditions. The first experiment (starting from scratch), has shown the consistency of our update formula with a fast and robust improvement of the correct spelling rate in few time steps. In a second experiment, we have considered the case of the online recovery that is expected to follow unforseen impairments. The spelling error information, as typically contained in a backspace hit, has been shown useful for the device adaptivity to unexpected changes. Combined with a specific failure detection algorithm, the policy gradient has been shown to appropriately adapt the classifier to the new situation, at the condition the reward is reliable enough (namely having a reliability greater than 70%).

Several perspectives can be opened from this work. First, from a machine learning standpoint, we have explicited the reward-based online improvement of a random policy (proposed in [20]) to the case of the "P300-speller" classifier with a softmax choice, and drawn some correspondences with the "bandit" approach [21]. It should be worth at this point to develop a more thorough comparison with other stochastic exploration methods, and decipher the conditions in which the policy gradient is worth using. Second, from the BCI standpoint, we have shown the feasibility of the reward-based approach with a limited feedback. An interesting possibility would be use a generic classifier (trained for instance from a public P300-speller database) instead of starting "from scratch". Provided sufficiently accurate (namely having a spelling accuracy greater than 70%), a simple signal like the "backspace" key should be enough to robustly improve the speller during use.

Acknowledgment

This research was partly supported by french ANR "défis" project CO-ADAPT (ANR-09-EMER-002). EEG data kindly provided by Jérémie Mattout and Emmanuel Maby from INSERM U1028.

References

[1] 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, no. 6, pp. 510 – 523, 1988.

[2] G. Vanacker, J. del R. Millán, E. Lew, P. W. Ferrez, F. G. Moles, J. Philips, H. Van Brussel, and M. Nuttin, "Context-based filtering for assisted brain-actuated wheelchair driving," *Intell. Neuroscience*, vol. 2007, Jan. 2007.

[3] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, "Eeg-based discrimination between imagination of right and left hand movement," *Electroencephalography and Clinical Neurophysiology*, vol. 103, no. 6, pp. 642 – 651, 1997.

[4] S. Mason and G. Birch, "A brain-controlled switch for asynchronous control applications," *IEEE Trans Biomed Eng.*, vol. 47, no. 10, pp. 15 – 21, October 2000.

[5] B. Blankertz, G. Dornhege, M. Krauledat, K. Müller, and G. Curio, "Optimizing spatial filters for robust eeg single-trial analysis," *IEEE Signal Proc. Magazine*, vol. 25, no. 1, pp. 41–56, 2008.

[6] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, "xDAWN algorithm to enhance evoked potentials: application to brain-computer interface." *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 8, pp. 2035–43, Aug. 2009.

[7] K. Ang, Z. Chin, H. Zhang, and C. Guan, "Mutual information-based selection of optimal spatial-temporal patterns for single-trial eeg-based bcis," *Pattern Recognition*, vol. 45, no. 6, pp. 2137–2144, 2012.

[8] D. Krusienski, E. Sellers, D. McFarland, T. Vaughan, and J. Wolpaw, "Toward enhanced p300 speller performance," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 15 – 21, 2008.

[9] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An efficient p300-based brain-computer interface for disabled subjects," *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 115 – 125, 2008.

[10] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," *IEEE Transactions On Signal Processing*, vol. 100, no. 10, October 2008.

[11] L. Bottou and Y. LeCun, "Large scale online learning," in *Advances in Neural Information Processing Systems (NIPS 2003)*. MIT Press, 2003.

[12] Y. Li and C. Guan, "An extended em algorithm for joint feature extraction and classification in brain-computer interfaces," *Neural Computation*, vol. 18, pp. 2730–2761, 2006.

[13] S. Lu, C. Guan, and H. Zhang, "Unsupervised brain-computer interface based on intersubject information and online adaptation," *IEEE trans on neural systems and rehabilitation engineering*, vol. 17, pp. 1147–1154, 2009.

[14] P. Kindermans, D. Verstraeten, and B. Schrauwen, "A bayesian model for exploiting application constraints to enable unsupervised training of a p300-based bci," *PloS one*, vol. 7, no. 4, 2012.

[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[16] A. Buttfield, P. Ferrez, and J. R. Millán, "Towards a robust bci: error potentials and online learning," *IEEE Trans Neural Syst Rehabil Eng.*, vol. 14, no. 2, pp. 164–168, 2006.

[17] B. Dal Seno, M. Matteucci, and L. Mainardi, "Online detection of p300 and error potentials in a bci speller," *Intell. Neuroscience*, vol. 2010, January 2010.

[18] N. Schmidt, B. Blankerz, and M. Treder, "Online detection of error-related potentials boosts the performance of mental typewriters," *BMC Neuroscience*, pp. 13–19, 2012.

[19] E. Daucé and T. Proix, "P300-speller adaptivity to change with a backspace key," in *Proceedings of TOBI workshop IV*, Neurocomp, Ed., Sion, Switzerland, 2013, pp. 105–106. [Online]. Available: http://ins.medecine.univmed.fr/wp-content/uploads/2013/02/dauce.pdf

[20] R. Williams, "Simple statistical gradient following algorithms for connectionnist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.

[21] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Shapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.

[22] S. Kakade, S. Shalev-Schwartz, and A. Tewari, "Efficient Bandit Algorithms for Online Multiclass Prediction," in *Proccedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

[23] E. Hazan and S. Kale, "Newtron: an efficient bandit algorithm for online multiclass prediction," *Proceedings of the 24th International Conference on Advances in Neural Information Processing Systems (NIPS 2011)*, 2011.

[24] K. Crammer, "Multiclass classification with bandit feedback using adaptive regularization," *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[25] M. Kearns, "Efficient noise-tolerant learning from statistical queries,"

Journal of the ACM, vol. 45, no. 6, pp. 983–1006, Nov. 1998. [Online]. Available: http://doi.acm.org/10.1145/293347.293351

[26] F. Denis, C. N. Magnan, and L. Ralaivola, "Efficient learning of naive bayes classifiers under class-conditional classification noise," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 265–272. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143878

[27] L. Ralaivola, F. Denis, and C. N. Magnan, "Cn = cpcn," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 721–728.

[28] E. Maby, G. Gibert, P. Aguera, M. Perrin, O. Bertrand, and J. Mattout, "The openvibe p300-speller scenario: a thorough online evaluation," in *Human Brain Mapping Conference 2010*, Barcelona, Spain, 2010.