ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

Fast online adaptivity with policy gradient: example of the BCI "P300"-speller

Emmanuel Daucé[1,2] Timothée Proix[1]
Liva Ralaivola[3] *

1- Inserm, Aix-Marseille Université, INS UMR S 1106,
13005 Marseille, France

2- Centrale Marseille,
13451 Marseille cedex 20, France

3- CNRS, Aix-Marseille Université, LIF UMR 7279,
13453 Marseille cedex 13, France

Abstract. We tackle the problem of reward-based online learning of multiclass classifiers and consider a policy gradient ascent to solve this problem in the linear case. We apply it to the online adaptation of an EEG-based "P300"-speller. When applied from scratch, a robust classifier is obtained in few steps.

1 Introduction

We consider the properties of classifiers embedded in devices capable of interacting with their environment and adapt their parameters in order to increase their accuracy during use. This problem of adapting the parameters while the device is used is known as the "online learning" problem. Online learning is particularly relevant when the statistics of the input change over time, i.e. when the input is *non-stationary* [1]. For such online adaptation to take place, the validity of the classification must be estimated during use, occasionally or at each new response. In the supervised case for instance, the expected label is given to the classifier after its response. In the non-supervised case, no feedback at all is given. Here we consider the reinforcement learning case, where the feedback may take the form of a scalar, named the "reward", and test it on an EEG signals dataset from a non-invasive brain-computer interface (BCI).

The objective of BCI's is to analyze in real-time EEG signals recorded at the surface of the scalp to control a device (mouse, keyboard, wheelchair,...). We consider here the case of grid-based P300 spellers [2], 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. Then, rows and colums are flashed several times at random while samples of EEG signals are extracted. After each row and each column has been flashed p times, K ERPs (event related potentials) are calculated for each row and each column of the grid. This set of multiple

*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.

ERP observations is constructed the following way: for each flash time t, take a 600 ms subsample $\mathbf{s}_{t:t+600\text{ms}}$ and classify it in its category $\in 1, ..., K$ (row or column number). Then, for each category $k \in 1..K$, calculate class average $\boldsymbol{x}_k$. Finally, construct a multi-ERP set $\underline{\mathbf{x}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_K)$. Then, the classifier has to identify in the set the index of the row and column where a particular ERP (called the "P300") took place, so that the resulting letter is at the intersection of this row and this column. Then another series of flashes starts over until the expected sequence of letters is formed.

The EEG dataset we use comes from a P300 experiment reported in [3]. This dataset contains data from 20 different subjects, each subject having spelled out 220 letters where every row and every column was flashed $p = 5$ times per letter. The correct response is known for every trial and every subject. From this dataset, we verified that robust and well-documented methods, like Naïve Bayes [3], linear discriminant analysis (LDA) [4], or support vector machines (SVM) [5], can achieve a spelling accuracy of 85%, which is consistent with state-of-the art results for this number of repetitions.

After a classifier has been trained, the subject is expected to handle the interface and be capable to achieve a letter-by-letter spelling task autonomously. This separation between training and use may however be a problem in real conditions, because of the putative non-stationarity of the signal collected by the electrodes. As many non-predictable events are expected to take place during an effective spelling session, the quality of the signal (and thus the accuracy of the interface) is expected to degrade over time. To tackle the non-stationarity problem, different methods have still been proposed, most of them relying on an unsupervised expectation-maximization (EM) procedure [6], which is not proven to always converge to a consistent classifier.

Here we make, to our best knowledge, a first attempt to adapt the methodology of reinforcement (i.e. reward-based) learning [7] to the context of an adaptive BCI P300 speller. In order to tackle the non-stationarity problem, we consider that a "reward", that is a scalar representing the agreement of the subject regarding the response of the device, is available. A reward is of course less informative than the true response, but, in counterpart, is in general cheaper to obtain. In a P300-speller setup, two solutions should be considered for that purpose: (i) use the "error negativity" ERP following a wrong classifier's response, where a detection rate of 85-90% is reported in the literature [8]. 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 before starting the P300 session; (ii) dedicate a specific symbol to the case when the previous character was incorrectly spelled (i.e. a "backspace" key) to detect the subject disagreement regarding previous response. A minimal spelling accuracy is then needed for this approach to be effective, which means that a training session has to take place prior to the spelling session.

In this paper, we set apart the reward detection problem and look in section 2 at the principles underlying reward-based learning. Then, in order to bring a proof of concept, we present in section 3 a simulation where learning is made "from scratch" (i.e. without initial training or calibration).

2 Multiclass policy gradient with immediate reward

2.1 Stochastic classifier

In the particular setting we consider (P300 speller), the problem is to identify a particular event (the "oddball") in a set of K observations $\underline{\mathbf{x}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_K) \in \mathcal{X}^K$, i.e. identify the "target" within a set having multiple inputs belonging to the "non-target" category and only one input belonging to the "target" category. The adaptive model we use is based on a regularity assumption, i.e. oddball and non-oddball examples display regularities that allow to separate them on the basis of their intrinsic features. The classifier relies on a vector of parameters $\boldsymbol{f}$ that is to be compared with every observation of the set $\underline{\mathbf{x}}$ through the scalar products $\langle \boldsymbol{f}, \boldsymbol{x}_k \rangle$'s that are expected to enhance the target and diminish the non-target inputs. The actual response $y \in \{1, ..., K\}$ is drawn from a multinomial distribution relying on the π-scores (softmax choice):

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

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

2.2 Learning problem

After every response, 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[1]. Maximizing the reward expectation means to find the best parameters vector $\boldsymbol{f}$, and thus the best policy in order to obtain the higher possible reward expectation, i.e.: $\max_{\boldsymbol{f}} E(r)$. Instead of directly maximizing this quantity, 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. The problem becomes:

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

where $\mathcal{G}$ is the objective function (or "gain" function) and λ is the *regularization parameter.*

2.3 Policy gradient

When the model is unknown, the solution of (2) can be obtained by trying to cancel the gradient of $\mathcal{G}$ through a stochastic gradient *ascent.* The policy

[1]It is given by the integral of the rewards obtained for every observation $\boldsymbol{x}$ and every choice k, given their probability of appearance: $E(r) = \int_{\mathcal{X}} \left(\sum_{k=1}^{K} r(\underline{\mathbf{x}}, k)\pi(\underline{\mathbf{x}}, k; \boldsymbol{f}) \right) p(\underline{\mathbf{x}}) d\underline{\mathbf{x}}$ where the distribution $p(\underline{\mathbf{x}})$ is unknown.

gradient is a general purpose reward-based algorithm we adapt here to the online multinomial classification case. Following [9], the regularized policy gradient can be expressed the following way:

$$\nabla_{\boldsymbol{f}} \mathcal{G} = E(r \nabla_{\boldsymbol{f}} \ln(\pi) - \lambda \boldsymbol{f}) \tag{3}$$

Starting from scratch, the update procedure is expected to refine the model trial after trial using the estimator (3), so that the rewards should be maximized in the long run. The general policy gradient estimator is: $\boldsymbol{g}(\underline{\mathbf{x}}, y) = r(\underline{\mathbf{x}}, y) \nabla_{\boldsymbol{f}} \ln \pi(\underline{\mathbf{x}}, y; \boldsymbol{f})$ depending on the two measures $\underline{\mathbf{x}}$ and y. From the derivation of $\ln \pi$ according to the mapping $\boldsymbol{f}$, we obtain the following expression:

$$\boldsymbol{g}(\underline{\mathbf{x}}, y) = r(\underline{\mathbf{x}}, y) \left(\boldsymbol{x}_y - \sum_k \pi(\underline{\mathbf{x}}, k; \boldsymbol{f}) \boldsymbol{x}_k \right) \tag{4}$$

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, update the mapping according to:

$$\boldsymbol{f}_t = (1 - \eta\lambda) \boldsymbol{f}_{t-1} + \eta r_t \left(\boldsymbol{x}_{y_t,t} - \sum_{k=1}^{K} \boldsymbol{x}_{k,t} \pi(\underline{\mathbf{x}}, k; \boldsymbol{f}_{t-1}) \right) \tag{5}$$

where η is the *learning rate*.

3 Application to the P300 speller classification problem

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, we use in the following a simple cross-validation procedure *that reproduces the conditions of an online learning experiment*: 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. We then apply the online update (5), with rewards generated by a simple comparison between the expected letter and the classifier's response. For a particular subject, the rate of correct spelling is then calculated (at each trial) as the average over the 1000 experiments.

We use binary rewards, i.e. $r(\underline{\mathbf{x}}, y) \in \{r^+, r^-\}$, where r^+ denotes a correct classification while r^- denotes a wrong classification (with, by principle, $r^+ > r^-$). In the multi-input setup, the values $r^+ = K - 1$ and $r^- = -1$ can be shown to be optimal in the first steps of the gradient ascent when a linear model is used (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" classifier, and in a second series, it is made according to a deterministic "argmax" classifier. In order to fairly compare the two methods, we systematically calculate the final spelling accuracy for different values of η

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

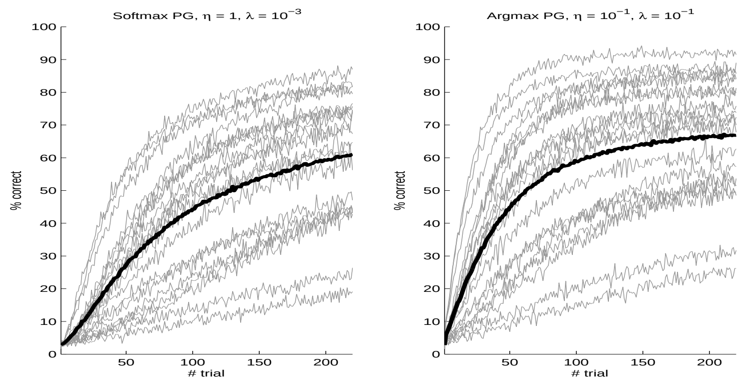

Fig. 1: – **A** – Spelling improvement during P300-speller experiment (mean over 20 subjects). Thin gray lines: individual improvement. Thick line: average improvement. Left: Softmax policy gradient. Right: Argmax policy gradient.

and λ ranging from 10^{-5} to 10 ("grid" search – not shown –), 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. In particular, the product $\eta\lambda$ gives an indication on the number of examples that significantly take part in the classifier build-up (as old examples are progressively erased by more recently seen examples). This "memory span", that drives the complexity of the classifier, can be approximated to $\frac{1}{\eta\lambda}$, so that a smaller λ implies a higher "memory span" (irrespectively of η). In the softmax case, the number of examples that allow to reach the final spelling accuracy is found to be $O(10^3)$, while it is only $O(10^2)$ in the argmax case.

We compare in figure 1 the different learning curves obtained in our series of 20×1000 simulations. The average learning curves show a monotonical increase, from $\frac{1}{36}$ (random response) to an average accuracy of $\simeq 60\%$ after 220 trials, and a tendency toward continuing improvement for $t > 220$. This 60% attained in only 220 trial can be considered fast, since no information except the reward is available to guide the learning process. The initial improvement indicates that the algorithm manages to follow the gradient from the very first steps, despite mostly negative rewards and occasional misleading rewards (when only the row – or only the column – is correct). Those results generally indicate that the variance of the gradient estimator is low enough to allow fast and efficient learning. If a significant improvement is observed for every subject, the inter-subject variability appears 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 is a known and chronic problem in BCI's. Finally, as the number of letters was limited to 220 in the training set, the asymptotic spelling rate can not be reached here. If the final spelling rate appears slightly better in the argmax case, the final slope is steeper in the softmax case, which suggests a clear tendency toward a continuing improvement,

ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2013, i6doc.com publ., ISBN 978-2-87419-081-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100131010.

where an asymptotic 80% rate seems reachable at a 10^3 horizon.

4 Conclusion

We have presented the application of a policy gradient algorithm to the problem of online multi-class classification, where the multinomial choice means identifying the “oddball” in a set of K inputs. The consistency of our update formula has been tested on a BCI P300-speller dataset, where a systematic improvement of the spelling accuracy is observed for every subject when starting “from scratch”. Despite fast learning capabilities, the spelling accuracy attained after 220 trials does not yet reach the state-of-the-art 85% spelling accuracy expected on this dataset. Those results are however encouraging for they open perspectives for BCI autonomous online improvement, and as such should be used in complement with standard batch classifiers.

References

[1] Jyrki Kivinen, Alexander J. Smola, and Robert C. Williamson. Online learning with kernels. *IEEE Transactions On Signal Processing*, 100(10), October 2008.

[2] L.A. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70(6):510 – 523, 1988.

[3] E. Maby, G. Gibert, P.E. 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.

[4] D.J. Krusienski, E.W. Sellers, D.J. McFarland, T.M. Vaughan, and J.R. Wolpaw. Toward enhanced p300 speller performance. *Journal of Neuroscience Methods*, 167(1):15 – 21, 2008.

[5] A Rakotomamonjy and V Guigue. Bci competition iii: dataset ii- ensemble of svms for bci p300 speller. *IEEE Trans Biomed Eng.*, 55(3):1147–1154, 2008.

[6] Yuanqing Li and Cuntai Guan. An extended em algorithm for joint feature extraction and classification in brain-computer interfaces. *Neural Computation*, 18:2730–2761, 2006.

[7] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[8] NM Schmidt, B Blankerz, and MS Treder. Online detection of error-related potentials boosts the performance of mental typewriters. *BMC Neuroscience*, pages 13–19, 2012.

[9] R.J. Williams. Simple statistical gradient following algorithms for connectionnist reinforcement learning. *Machine Learning*, 8:229–256, 1992.