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

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

[1]Ecole Centrale Marseille

[2]INS - Aix-Marseille Université - Inserm UMR 1106

[3]LIF - Aix-Marseille Université - CNRS UMR 7229

July 12, 2015

Institut de Neurosciences des Systèmes

Table of contents

Plan

Brain-Computer Interfaces

- Embedded classifiers:
  - real-time
  - noisy (EEG)
  - subject/use case specific
  - non-stationary

$\Rightarrow$ Adaptive Learning

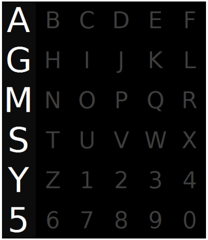
- Brain Computer Interfaces, a tool for :
  - Communication (in the absence of a motor capabilities)
  - Brain monitoring / neurofeedback
  - Motor rehabilitation
- "CO-ADAPT" project : INRIA Sophia/ INSERM Lyon/ CNRS LATP, ... (French ANR funding)
  - "co-adaptive" motor imagery
  - "co-adaptive" P300 speller

Embedded classifiers

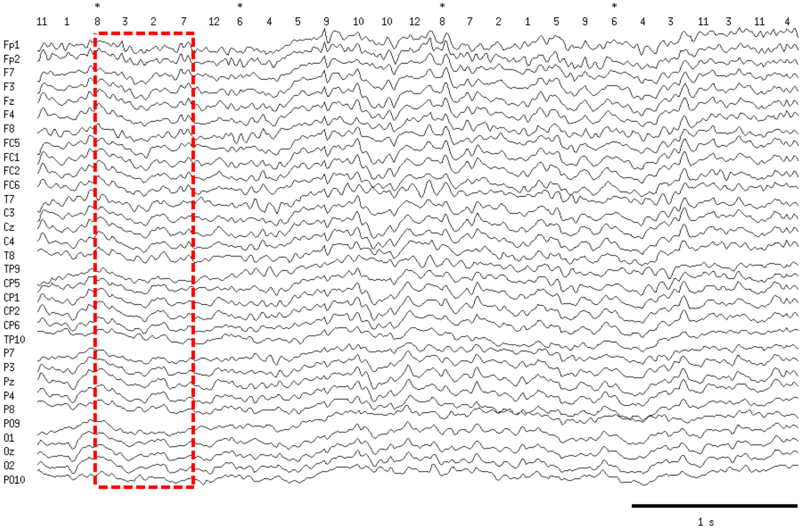
- Classification problem : sources → signal → features extraction → classification
  - adaptive feature extraction
  - adaptive classification
- Online learning : "light" classifier update at each processing step
  - supervised online learning: stochastic gradient descent
  - unsupervised online learning: online mobile centers (K-means, EM,...)
  - reward-based online learning: stochastic classifiers + policy/value iteration
    - exploration/exploitation trade-off
    - which reward?

P300 speller

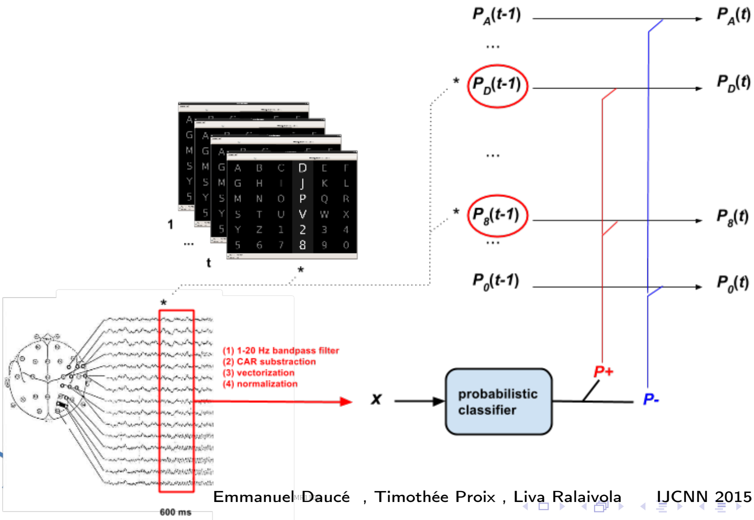
- EEG :
  - 10 - 60 channels (surface electric potential - H Berger, 1929)
  - high temporal resolution / low spatial resolution
  - noisy, non-reliable,... "Evoqued potentials" technique
  - the "P300" ERP is "surprise" effect ("oddball" paradigm)
- P300-speller (Farwell and Donchin, 1988):
  - based on the "oddball" paradigm
  - 6 x 6 letters grid
  - random row/column magnification (every 150-300 ms)
  - row/column evidence build-up + argmax choice
  - low SNR / low bit rate (many flashes for one letter)
  - spelling accuracy tends to decrease in the long run



EEG data (from Inserm U1028, Lyon, France)



Data processing pipeline



P300-speller roadmap

- Transfer learning [Kindermans et al., 2012b, Congedo et al., 2013]:
  - Across subjects
  - generic classifier "pre-learning"
  - smart initialization
- Optimal display
- Evidence build-up [Perrin, 2012, Kindermans and Schrauwen, 2013]:
  - Probabilistic classifier : posterior estimate
  - Evidence accumulation
  - Threshold-based dynamic stopping
  - Speed-accuracy trade-off
- Online learning :
  - Toward a subject-specific classifier
  - Recover from unexpected changes
  - Different approaches :
    - EM [Li and Guan, 2006, Kindermans et al., 2012a]
    - RL [Daucé et al., 2013]

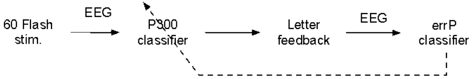
Plan

Reinforcement learning : multiple approaches

- general RL problem :
  - observations : x_1, ..., x_T
  - policy (random decision = exploration) : $\pi(x, y) = P(y|x)$
  - decisions : y_1, ..., y_T
  - reward : $r_T \in \mathbb{R}$
- Task : try to improve the politics for increasing time/trial
  - Actor-critic / Q-learning :
    - delayed reward
    - discrete state/action space (LUT)
    - separates Value update / Decision making
  - Multi-armed bandit :
    - instant reward
    - no states (!)
  - Policy-gradient :
    - instant or delayed rewards
    - discrete or continuous state/actions space
    - direct update of the decision

Rewarded classifier

- Optimal $\boldsymbol{w}$ unknown $\rightarrow$ model-free, trial and error
- Online stochastic classifier :
  - read input observations set : $\underline{\mathbf{x}} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_K)$
  - give a score to every class : $\forall k, \pi_k(\underline{\mathbf{x}}; \boldsymbol{w}) = \frac{\exp(\boldsymbol{x}_k \boldsymbol{w}^T)}{\sum_l \exp(\boldsymbol{x}_l \boldsymbol{w}^T)}$
  - choose the response at random (Softmax choice)
  - read the *reward* r
  - update $\boldsymbol{w}$
- Which reward ?
  - "error" potential after the classifier's response:



  - "UNDO" key on the virtual keyboard

Learning and forgetting : Regularized Policy gradient update

- Regularized optimization (λ hyperparameter):

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

- Regularized gradient ascent : $\nabla_w \mathcal{H} = E(r\nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathbf{x}}; \mathbf{w})) - \lambda \boldsymbol{w}$
  - One-sample estimate (stochastic gradient) :

$$E(r\nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathbf{x}}, \mathbf{w})) \simeq r\nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathbf{x}}; \mathbf{w})$$

  - Online update (learning rate $\eta << 1$):

$$\begin{aligned} \boldsymbol{w} &\leftarrow \boldsymbol{w} + \eta(r\nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathbf{x}}; \mathbf{w}) - \lambda \boldsymbol{w}) \\ &= (1 - \eta\lambda)\boldsymbol{w} + \eta r\nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathbf{x}}; \mathbf{w}) \end{aligned}$$

- The old examples "fade away" as time passes → tracking algorithm and novelty detection (Kivinen et al, 2010)

The "oddball" update case

- Policy gradient:

$$\boldsymbol{g}(\underline{\mathrm{x}}, y) = r \nabla_{\mathbf{w}} \ln \pi_y(\underline{\mathrm{x}}; \mathbf{w})$$
$$= r \left(\boldsymbol{x}_y - \sum_k \pi_k(\underline{\mathrm{x}}; \boldsymbol{w}) \boldsymbol{x}_k \right)$$

- Update :

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

Binary rewards

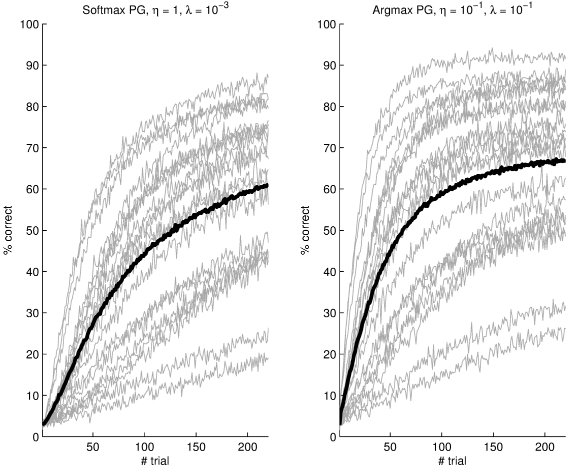
TODO

Plan

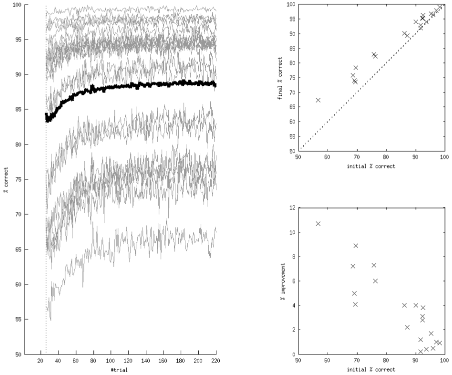
- EEG data : 20 subjects $\times$ 32 EEG channels $\times$ 220 trials $\times$ 12 row/columns $\times$ 5 flashes per row/column
- data preprocessing. For each trial :
  - 600 ms sample after each flash (100 Hz sampling)
  - Bandpass [1-20] Hz filter
  - common average reference substraction + channel normalization
  - 5 repetitions : average response calculation per row/column
  - vector construction : $\boldsymbol{x} \in \mathbb{R}^{32\times 60}$, $||\boldsymbol{x}|| = 1$
  - set construction : $\underline{\boldsymbol{x}}^{\text{row}} = (\boldsymbol{x}_1^{\text{row}}, ..., \boldsymbol{x}_6^{\text{row}})$, $\underline{\boldsymbol{x}}^{\text{col}} = (\boldsymbol{x}_1^{\text{col}}, ..., \boldsymbol{x}_6^{\text{col}})$
- cross-validation : for one (η, λ) couple:
  - learning from scratch : $w_0 = 0$
  - simulated rewards : $r \in (r^+, r^-)$ with $r^+ = 5$, $r^- = -1$.
  - 1000 simulations $\times$ 20 subject with shuffled spelling order
  - "softmax" and "argmax" classifier variants

Institut de Neurosciences des Systèmes

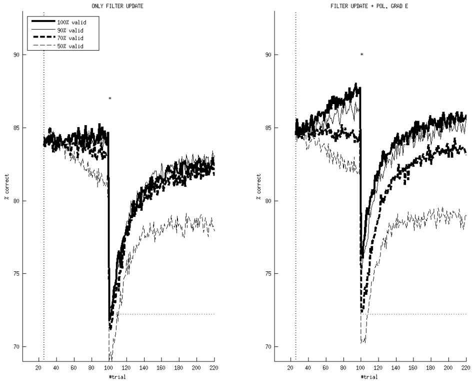
Softmax/Argmax spelling improvement



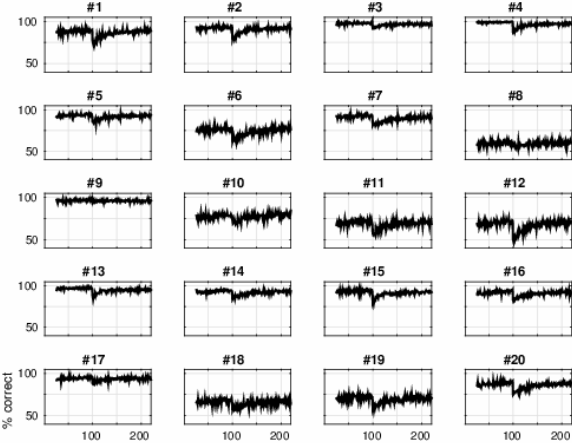
Classification improvement after a 25-trials training session



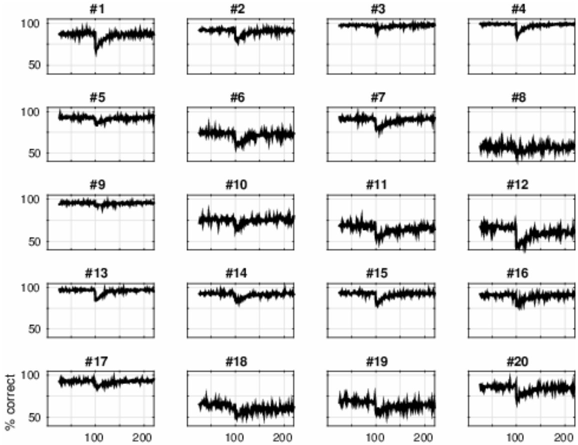
Global recovery after electrode break



Individual recovery, feedback noise = 10%



Individual recovery, feedback noise = 30%



Plan

Conclusion

- P300-speller implements the "oddball" classification problem
- "instant" reward-based gradient descent is efficient in the P300-speller case:
  - fast learning : clean class separation + low-variance gradient
  - allows to track environmental changes
- But :
  - high variability between subjects
  - reward extraction problem (difficulty shift)
  - non-reliable rewards?

Outlook and discussion

- the oddball problem is a specific decision-making problem
  - "choose" the element that fits the best the matching criterions in a set of K candidates
  - update the matching criterions
  - the candidates are renewed at each trial
- forgetting (and re-learning) is important in a non-stationary environment:
  - (η, λ) change tracking
  - λ regulates the score separation
  - $\frac{1}{\eta\lambda}$ is the sliding window
- reward-based learning is efficient in softmax/linear classifiers
  - bridges between stochastic / probabilistic frameworks

Congedo, M., Barachant, A., and Andreev, A. (2013).
A new generation of brain-computer interface based on riemannian geometry.
CoRR, abs/1310.8115.

Daucé, E., Proix, T., and Ralaivola, L. (2013).
Fast classifier adaptation in non-stationary environments: a policy gradient application to the bci p300-speller "oddball" paradigm.
In *proc. of ESANN 2013*, pages 197–202.

Kindermans, P., Verstraeten, D., and Schrauwen, B. (2012a).
A bayesian model for exploiting application constraints to enable unsupervised training of a p300-based bci.
PloS one, 7(4).

Kindermans, P.-J. and Schrauwen, B. (2013).
Dynamic stopping in a calibration-less p300 speller.
In *5th International Brain-Computer Interface Meeting*

Kindermans, P.-J., Verschore, H., Verstraeten, D., and Schrauwen, B. (2012b).
A p300 bci for the masses: Prior information enables instant unsupervised spelling.
In *Advances in Neural Information Processing Systems (NIPS 2012)*.

Li, Y. and Guan, C. (2006).
An extended em algorithm for joint feature extraction and classification in brain-computer interfaces.
Neural Computation, 18:2730–2761.

Perrin, M. (2012).
Coadaptation cerveau machine pour une intteraction optimale :application au P300-speller.
PhD thesis, Université Claude Bernard - Lyon 1.