

A Genetic Programming Approach to Detecting Artifact-generating Eye Movements from EEG in the Absence of Electro-oculogram

Riccardo Poli, Caterina Cinel, Luca Citi and Mathew Salvaris

Abstract—In this paper we use genetic programming — an evolutionary program-induction technology — to evolve algorithms that accurately approximate the behaviour of two standard detectors of ocular movement based on Electro-oculogram (EOG). The prediction is based entirely on EEG signals, i.e., without using EOG, making it possible to detect eye movements even in data recorded without EOG or eye tracking. Experimental results with this approach are very encouraging.

I. INTRODUCTION

A variety of artifacts can affect EEG recordings [1], [2]. Particularly important, and yet difficult to control, are artifacts due to ocular activity. Such Electro-Oculogram (EOG) artifacts may be several orders of magnitude bigger than the ordinary elements of EEG, which makes them potentially damaging, particularly if EEG is recorded to evaluate averages of Event-Related Potentials (ERPs). If the artifacts are not corrected or the epochs containing such artifacts are not removed, the average can be totally distorted, rendering it unusable for inference [3]. EOG artifacts can also lead to extremely larger deviations from the desired behaviour in Brain-Computer Interfaces (BCIs) [4]–[10], particularly those where there is an analogue relationship between input signals and output control signals, such as in the Essex BCI Mouse [11]–[13]. For these reasons a variety of methods are routinely used to control EOG artifacts. We briefly review a few important ones below.

Fundamentally, there are two approaches one can take: *correction* and *rejection* [2], [3].

Correction means subtracting out the effects of EOG from the EEG. For example, the effects of eye blinks and vertical components of saccades can be reduced by using the time-domain linear regression between each channel and the Vertical EOG (VEOG) [2], [14]. Alternatively one can use Independent Component Analysis (ICA) [15] in the analysis of EEG and ERPs. ICA can decompose EEG signals into independent components (sources variability), including some which represent pure artifacts and that can then be subtracted from the EEG to remove such artifacts [16]–[20].

The second approach is rejection. Rejection means removing from the analysis fragments of EEG (or trials) affected by eye blinks or movement artifacts. Rejection requires the *detection* of artifacts (while correction does not necessarily require it).

Many approaches to artifact detection exist. Naturally, nothing can be better at identifying artifacts than visual inspection of EEG and EOG by a human expert. However, the task is very tedious particularly considering that a typical BCI or psychophysiology experiment involving multiple subjects may easily require many hours of EEG recordings. Also, this approach cannot really be used in any form of on-line BCI and, therefore, an automated procedure is necessary.

Two of the simplest and fastest automated artifact detectors are [2], [3]: (a) deleting portions of the data where the EOG amplitude deviates by more than a set threshold, say, $50 \mu\text{V}$ or $75 \mu\text{V}$, from the baseline; (b) measuring the difference, ΔV , between the maximum and minimum EOG amplitude within a block of the signal and rejecting it if ΔV exceeds a threshold (we will use these two algorithms as references later in the paper).

These and many other algorithms require the availability of EOG data, which in turn requires positioning some extra electrodes around the eyes. In any electro-physiological settings subject preparation times depend on the number of electrodes required (and therefore the smaller the number of electrodes the better). Additionally, in BCI, there is constant pressure on reducing the number of electrodes required by a system (ideally, to make it usable in everyday life, a BCI system should be based on just a few electrodes). Therefore, not requiring the recording of EOG can have significant practical advantages.

Naturally, there are effective (albeit complex) techniques to do artifact detection and removal (e.g., see [20], [21]) which do not require the availability of EOG. However, there are cases where being artifact-free is not the only criterion for acceptance and where really EOG is required to decide whether a trial should be accepted, even if the EOG did *not* significantly contaminate EEG. For example, there are experimental conditions where subjects can be asked to limit eye blinking or to look at a fixation point during each trial, to ensure uniformity of perceptual conditions. In these cases EOG is perhaps the most direct and effective way of identifying and removing trials where subjects did not follow the instructions.

In this study we investigate the possibility of effectively and accurately reconstructing the behaviour of an EOG-based detector of ocular movement with a method based on information available in the EEG alone. This makes it possible to detect eye movements for a variety of purposes without the inconvenience of using eye-trackers and EOG

The authors are with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, Colchester, UK, Email: {rpoli,ccinel,lciti,missal}@essex.ac.uk

This work was supported by the UK Engineering and Physical Sciences Research Council (grant EP/F033818/1).

facial electrodes.¹ The method we present here is based on cutting edge Genetic Programming (GP) technology [22]–[24]. GP is an evolutionary algorithm specialised in automatic program induction.

As we will see, while evolving EEG-based detectors of ocular movement using GP is computationally expensive, many of the detectors found by GP general and can be used without further evolution. Most evolved detectors are also efficient, requiring only a small number of floating point operations per sample.

The paper has the following structure. Section II describes the GP system used, its primitives, parameter settings and fitness function. In Section III we describe the stimuli, procedures and participants used in the collection of EEG data, our training and test sets and the results of running GP. We draw some conclusions in Section IV.

II. GENETIC PROGRAMMING SETUP

We used a strongly-typed GP system implemented in Python. Since fitness evaluation in our domain of application is very computationally intensive, we created a parallel implementation which performs fitness evaluations across multiple CPU cores and we used a computer with a large amount of RAM for our runs.

The system uses a steady-state update policy. It evolves a population of either 5,000 or 10,000 individuals with tournament selection with a tournament size of 5, a strongly-typed version of the grow method with a maximum initial depth of 4, and strongly-typed versions of sub-tree crossover and sub-tree mutation [24]. Both are applied with a 50% rate and use a uniform selection of crossover/mutation points. The system uses the primitive set shown in Table I. Program trees were required to have a `Bool` return type. With this setup we performed runs of 30 generations.

Because of the extreme computational load required by our fitness evaluation and the complexity of the problem (which forced us to use a relatively large population), in this paper we only report the results of one or two runs per condition (six runs in total).² We feel this is reasonable since we are really interested in the output produced by GP — as is the case in many practical applications of GP — rather than in optimising the process leading to such output. Each run took approximately 100 CPU hours to complete, on average.

Let us now turn to the fitness function we used to guide evolution. In our system, *fitness* is the dissimilarity between the behaviour of an *EOG-based* eye-movement detector (we tested the approach with two detectors) and the *EEG-based* detector represented by a GP program tree. For the purpose of detection, our EEG datasets (which we describe in Section III) were divided up into 1 second blocks. Each block was passed to both an EOG-based detector of eye movement and an EEG-based detector controlled by the GP program under test. Both returned either `True` (indicating presence of

large eye movements) or `False`. Fitness (to be minimised) was the fraction of blocks where the output of the two did not match (which is equivalent to an error rate).

As VEOG-based reference detectors we used the two simple detectors described in Section I and [2, pp. 154–155]. In the first detector, which we will call *Threshold*, we used an acceptance band of $\pm 50\mu\text{V}$ and we marked a block of signal as a potential artifact-generating eye-movement if the VEOG was outside this band for more than 6.25% of the time (8 samples at 128 Hz). With the second detector, which we will call *MinMax*, we measured ΔV in each block, and marked the block as a Positive if $\Delta V > 100\mu\text{V}$.

The EEG-based detectors evolved by GP worked similarly to the first VEOG-based detector: the program was run on each sample in a 1 s block. On each occasion it returned either `True` or `False`. If the program output was `True` for more than 6.25% of the time, then the detector reported the block as a Positive, otherwise as a Negative.

Since our data were sub-sampled at 128 Hz and we had, on average, 12,387 blocks in the two data sets used for training, computing the fitness of a program required executing it 1,585,536 times.

III. EXPERIMENTS

A. Participants

To test our approach we used data from two experiments: *Mouse* (a BCI experiment where ERPs are used for mouse pointer control) and *Perception* (investigating the process of visual feature binding via ERPs). In the Mouse experiment we tested 14 participants with an average age of 30. In the Perception experiment we tested 12 subjects (average age: 24 years). All had normal or corrected-to-normal vision and normal colour vision.

B. Procedures and Apparatus

EEG signals were acquired using a BioSemi ActiveTwo system with 64 pre-amplified DC-coupled electrodes spaced evenly over the scalp. Additional electrodes were placed at the earlobes, at the left and right external canthi to record horizontal electro-oculogram (HEOG), and infra-orbitally to record vertical electro-oculogram (VEOG). Signals were acquired at 2048 samples per second, were then bandpass-filtered between 0.15 and 40 Hz and, finally, were down-sampled to 128 Hz.

In the Mouse experiment, in each session participants were presented with visual displays showing 8 circles arranged around an imaginary circle at the centre of the display. Each session was divided into runs, which we will call *direction epochs*. A direction epoch started with the presentation of a blank screen and, after a predefined period, the circles appeared at the centre of the screen. A red arrow then appeared for 1 second pointing to the target circle. After 2 seconds the circles started flashing randomly (a flash consisted of a temporarily change in colour of any circle from grey to white). This stopped after 20 to 24 trials, with a trial consisting of the individual activation of each of the 8 stimuli. Subjects were instructed to count the number

¹As a by product, this also makes it possible to get new information from the many EEG datasets which have been recorded without EOG.

²Naturally, we performed a much larger number of smaller runs while developing the fitness function.

TABLE I
PRIMITIVE SET USED IN OUR APPLICATION.

Primitive	Output Type	Input Type(s)	Functionality
0.5, -0.5, 0.1, -0.1	Float	None	Floating point constants used for numeric calculations
Fp1, AF7, AF3, F1, ... (60 more channel names)	Float	None	Returns a sample recorded from one of the channels. Samples are of type Float.
+, -, *, min, max	Float	(Float, Float)	Standard arithmetic operations plus maximum and minimum on floats.
>, <	Bool	(Float, Float)	Standard relational operations on floats
if	Float	(Bool, Float, Float)	If-then-else function. If the first argument evaluates to True, then the result of evaluating its second argument is returned. Otherwise, the result of evaluating the third argument is returned.
abs	Float	Float	Returns the absolute value of a Float.

of flashes of the target circle in each direction epoch, and then had to report it verbally at the end of the epoch. We used a Stimulus Onset Asynchrony (SOA, the time interval between the beginning of the consecutive flashes) of 200 ms ($\approx 12 \times 60^{-1}$) (as permitted by the 60 Hz refresh rate of the LCD monitor used) and an Inter-Stimulus Interval (ISI, the time interval between the end of one flash and the beginning of the following flash) of 0.00s. Each participant carried out between 16 and 45 direction epochs (34 on average). More details on this experiment can be found in [13].

In the Perception experiment, on each trial, participants were presented with a four-letter string. The first and last letters were always ‘S’. Of the two middle letters, one was always an ‘O’, while the other was either an ‘L’ or an ‘X’. The first and last letters were always white, while the colour of the two middle letters could either be red, green or blue, but never the same colour. The background was black. Each letter string was randomly presented in one of four quadrants of the display. The duration of the stimulus display varied between 50ms and 150ms (depending on subjects’ performance, which had to be kept below ceiling and above chance). At the beginning of each trial, a fixation cross appeared at the centre of the display followed, after 500ms, by the letter string in one of the quadrants. Participants were instructed to gaze at the fixation cross even when the letters appeared. The task was to decide whether or not, on each display, the target letter was presented. The target letter was always an ‘L’ of a specific colour (the experiment was divided into blocks of trials with the target colour being set at the beginning of each block). Participants gave their responses by pressing either the left (“Yes” response) or the right (“No” response) button of the mouse. More details on this experiment can be found in [25].

In both experiments participants were seated comfortably at approximately 80 cm from an LCD screen, their neck supported by a C-shaped inflatable travel pillow to reduce muscular artifacts.

C. Training and Test Sets

For both experiments we divided up the data into blocks of 1 s duration (128 samples), and then used the even-numbered blocks as training set and the odd-numbered blocks as a test set. This resulted in two sets of 10,784 blocks for the Perception dataset and two sets of 13,991 blocks for the Mouse dataset.

TABLE II
TRAINING- AND TEST-SET ERROR RATES FOR EVOLVED DETECTORS.

Dataset	Ref detector	Training	Testing
Perception	MinMax	10.76%	10.68%
Perception	Threshold	11.16%	11.34%
Mouse (smaller run)	MinMax	5.07%	5.42%
Mouse (smaller run)	Threshold	4.12%	4.43%
Mouse (larger run)	MinMax	6.28%	6.64%
Mouse (larger run)	Threshold	3.68%	4.05%

For the purpose of testing the approach we concentrated on the detection of ocular movements as revealed by the left VEOG. When processing the training data with MinMax, 29.66% of the blocks were marked as Positives for the Perception dataset, while only 15.82% of the blocks in the Mouse dataset were reported to present significant eye movements. The corresponding percentages of Positives with the Threshold detector were 31.07% and 15.46%, respectively. Very similar percentages of Positives were obtained in the test sets.³

D. Results and Discussion

With the Perception dataset we performed one run of 5,000 individuals (computer programs) with each of the reference EOG-detectors (Threshold and MinMax), while with the Mouse dataset, we performed two runs with each reference detector, with 5,000 and 10,000 individuals, respectively. In all cases the architecture of the GP detector was kept constant: only the evolved programs changed.

Table II shows the error rates obtained with different datasets and reference detectors in different runs. The error rate represents the fraction of blocks where the responses of the reference and the evolved detectors mismatched. Simple comparison of the training and testing results clearly illustrates that the evolved solutions generalise really well. This is almost unsurprising given the large size of the datasets we used. In addition, results indicate that evolved programs can match quite accurately the behaviour of the corresponding EOG-based eye-motion detectors, matching their output in approximately 89% percent of the blocks for the Perception dataset and 95% of the blocks for the Mouse dataset.

³The high frequency of eye movements in the Perception experiment is due to the particular protocol used for visual stimulation. Subjects are asked to focus on the fixation cross in the middle of the screen. However, because the stimuli appear in peripheral locations, it is not always easy for subjects to avoid saccading towards the stimuli.

TABLE III

SENSITIVITY AND SPECIFICITY (IN THE TEST SET) OF THE DETECTORS EVOLVED IN OUR RUNS.

Dataset	Ref detector	Sensitivity	Specificity
Perception	MinMax	73.38%	96.42%
Perception	Threshold	70.46%	97.14%
Mouse (smaller run)	MinMax	71.13%	99.09%
Mouse (smaller run)	Threshold	81.13%	98.23%
Mouse (larger run)	MinMax	65.10%	98.79%
Mouse (larger run)	Threshold	82.55%	98.43%

While these figures are very encouraging, we are also naturally interested in determining how the errors produced by the evolved systems are distributed between the Negative and Positive classes. The split is reported in Table III for the test set. Similar results (not reported) were obtained for the training set.

As illustrated by the table all systems exhibit a very high specificity of approximately 98% on average. Sensitivity is lower: approximately 74% on average. In other words, evolved detectors match extremely closely the Negatives produced by the reference EOG-based detectors, but miss approximately 1 in 4 Positives.

To understand the reason for the imbalance between misses and false alarms consistently produced in our runs one needs to look at how GP arrived at solutions. Because our datasets have significant more Negatives than Positives, initially GP focuses its search on solutions that produce no or very few Positives. It is only through a process of further refinement that, generation after generation, this behaviour is progressively modified. End-of-run solutions still remain slightly biased towards high specificity.

The bias, however, can easily be changed. As we indicated above, within the GP based detector, a program is run on each sample in a 1 s block. By default, if the number of samples (out of 128) where the program's output is True exceeds 8, the detector reports a Positive, otherwise it returns a Negative. By simply changing the threshold, $n = 8$, we can therefore make the system more or less sensitive to eye movements.

In Figure 1 we illustrate how the behaviour of our evolved detectors is changed by this procedure via Receiver Operating Characteristic (ROC) curves, which plot the True Positive rate (sensitivity) *vs* the False Positive rate ($1 - \text{Specificity}$) when the threshold n is varied from 0 to 24. As one can see there are ample possibilities for users to tune the sensitivity/specificity balance. For example, it is possible to run the detector evolved for Perception and MinMax at approximately 85% sensitivity and specificity or a detector evolved for Mouse and Threshold at approximately 90% sensitivity and specificity.

Having seen these detectors in action, it is interesting to now have a closer look at the strategies they adopt to solve the problem of detecting eye movements only using EEG.

In the solutions evolved on the Perception dataset we found a curious pattern from this point of view, where, to our surprise, almost no front EEG channels were used.

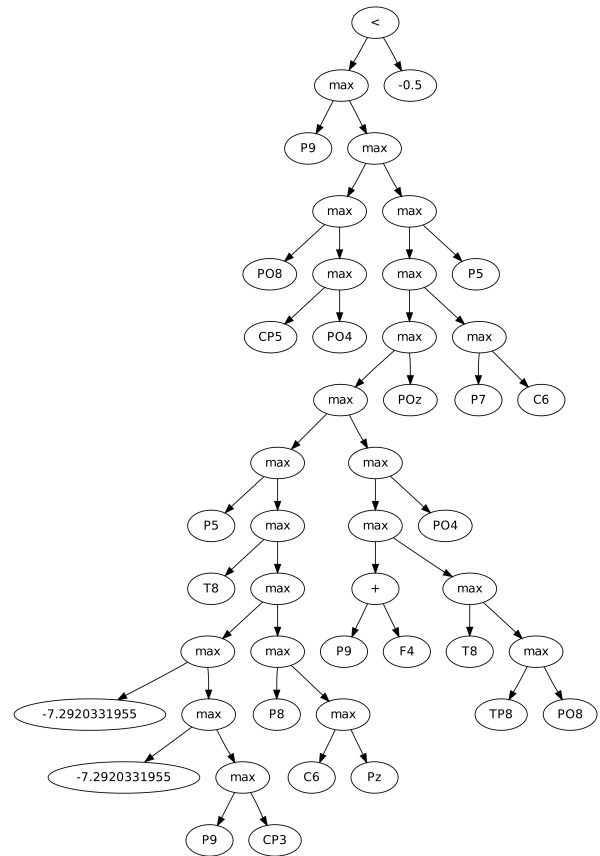


Fig. 2. Best evolved solution found for the Perception dataset and the MinMax detection algorithm (after automated simplification).

This is shown, for example, by the solution evolved for the MinMax detector reported in Figure 2.⁴ The most likely explanation for this result is that in the Perception experiment eye movements are correlated and phase-locked with the presentation of the stimulus string on the screen. EEG in the occipito-parietal regions of the brain is typically then depolarised and, we suspect, it may be even more so if saccades take place. The program in Figure 2 checks to see if a large portion of the occipito-parietal region is depolarised (note the widespread use of the max operation and the comparison at the root of the tree). This explains why it works well as an eye-movement detector in the Perception dataset. The program, however, does not work well on the Mouse dataset and, we suspect, does not generalise beyond Perception.

Let us now turn to the solutions evolved for the Mouse dataset. Here the situation is quite different. All solutions make ample use of frontal channels. In Figure 3 we show,

⁴For easier display the program in the figure has been first syntactically simplified replacing expressions involving only numerical constants with their value and expressions of the form (if True ExpA ExpB) or (if False ExpA ExpB) with ExpA and ExpB, respectively. Then, starting from the leaves, we replaced sub-trees with their median output if the replacement influenced performance by less than 2%.

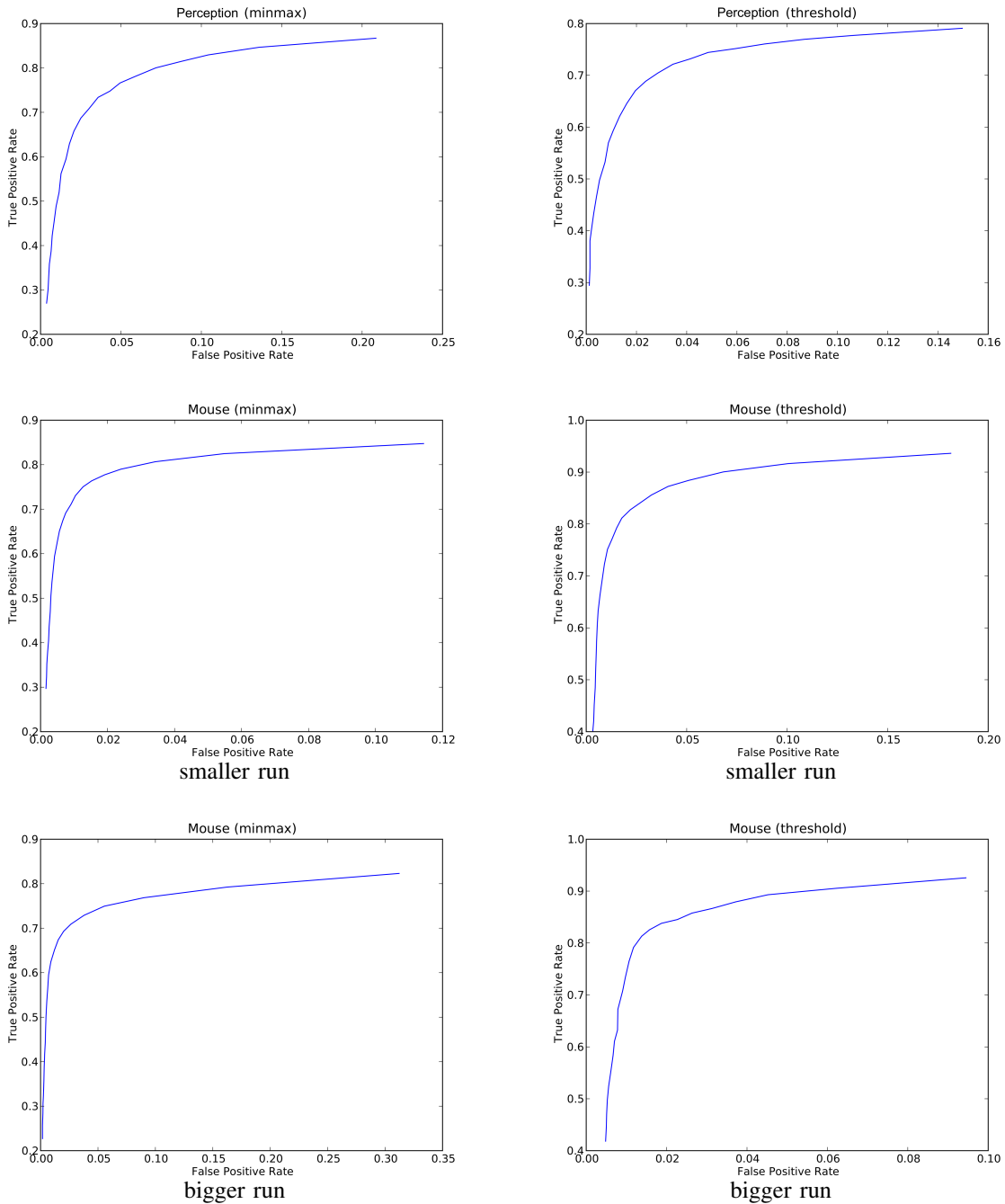


Fig. 1. ROC curves representing the trade-off between sensitivity and specificity in the evolved eye-movement detectors.

as an example, the solution found in the smaller GP run with the Mouse dataset and the MinMax detection algorithm. Also, all solutions generalise well beyond the Mouse dataset. For example, the solution in Figure 3 produces a respectable error rate of 13.25% on the Perception test set with a split of 84.18% sensitivity and 87.90% specificity. Its ROC curve, shown in Figure 4, is even superior to the corresponding curve for the detector evolved on the Perception dataset and shown in Figure 1(top right).

IV. CONCLUSIONS

In this paper we have explored the possibility of evolving detectors for eye movements which emulate the behaviour of EOG-based detectors but are entirely based on the analysis of EEG data alone.

Results have been extremely encouraging. Most evolved detectors have shown a significant degree of generality, working well on two completely different experiments and across a total of 26 subjects. It is also worth noting that, while evolving the our detectors is computationally expensive, the evolved detectors are extremely efficient. For example, the

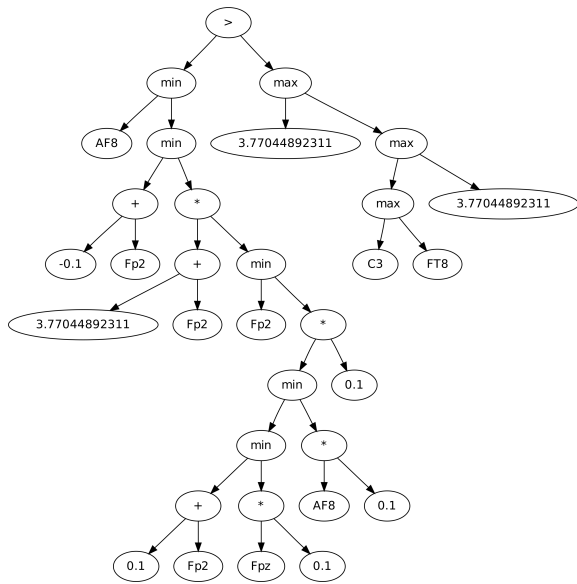


Fig. 3. Best evolved solution found in the smaller run with the Mouse dataset and the MinMax detection algorithm (after automated simplification).

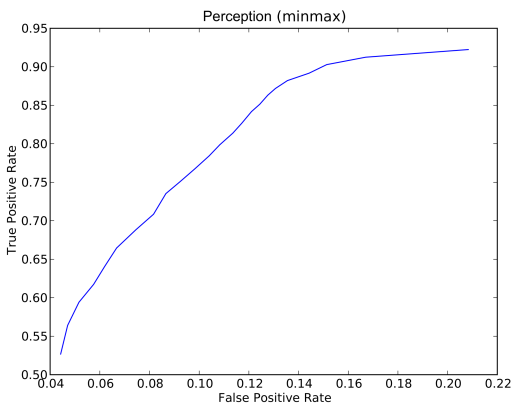


Fig. 4. ROC curve obtained when testing on the Perception dataset the detector evolved for the Mouse dataset and the MinMax detector.

general detector shown in Figure 3 only requires 15 floating-point operations for sample.

Such detectors are thus likely to offer several practical advantages for psycho-physiological investigations as well as BCI systems wherever eye movements need to be controlled or their artifactual effect needs to be minimised.

REFERENCES

- [1] T. C. Handy, Ed., *Event-related potentials. A Method Handbook*. MIT Press, 2004.
- [2] S. J. Luck, *An introduction to the event-related potential technique*. Cambridge, Massachusetts: MIT Press, 2005.
- [3] R. J. Croft and R. J. Barry, "Removal of ocular artifact from the eeg: a review," *Clinical Neurophysiology*, vol. 30, pp. 5–19, 2000.
- [4] 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*, vol. 70, no. 6, pp. 510–523, Dec 1988.

- [5] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," *Electroencephalography and Clinical Neurophysiology*, vol. 78, no. 3, pp. 252–259, Mar 1991.
- [6] G. Pfurtscheller, D. Flotzinger, and J. Kalcher, "Brain-computer interface: a new communication device for handicapped persons," *Journal of Microcomputer Applications*, vol. 16, no. 3, pp. 293–299, 1993.
- [7] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kbler, J. Perelmouter, E. Taub, and H. Flor, "A spelling device for the paralysed," *Nature*, vol. 398, no. 6725, pp. 297–298, Mar 1999. [Online]. Available: <http://dx.doi.org/10.1038/18581>
- [8] N. Weiskopf, K. Mathiak, S. W. Bock, F. Scharnowski, R. Veit, W. Grodd, R. Goebel, and N. Birbaumer, "Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI)," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 966–970, Jun 2004.
- [9] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 101, no. 51, pp. 17 849–17 854, Dec 2004.
- [10] E. W. Sellers and E. Donchin, "A P300-based brain-computer interface: Initial tests by ALS patients," *Clinical Neurophysiology*, vol. 117, no. 3, pp. 538–548, Mar 2006. [Online]. Available: <http://dx.doi.org/10.1016/j.clinph.2005.06.027>
- [11] L. Citi, R. Poli, C. Cinel, and F. Sepulveda, "P300-based BCI mouse with genetically-optimized analogue control," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 16, no. 1, pp. 51–61, Feb. 2008.
- [12] R. Poli, L. Citi, F. Sepulveda, and C. Cinel, "Analogue evolutionary brain computer interfaces," *IEEE Computational Intelligence Magazine*, vol. 4, no. 4, pp. 27–31, Nov. 2009.
- [13] M. Salvaris, C. Cinel, R. Poli, L. Citi, and F. Sepulveda, "Exploring multiple protocols for a brain-computer interface mouse," in *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, Buenos Aires, September 2010, pp. 4189–4192.
- [14] R. Verleger, T. Gasser, and J. Möcks, "Correction of EOG artifacts in event-related potentials of the EEG: aspects of reliability and validity." *Psychophysiology*, vol. 19, no. 4, pp. 472–480, Jul 1982.
- [15] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Wiley-IEEE, 2001.
- [16] S. Makeig, A. J. Bell, T.-P. Jung, and T. J. Sejnowski, "Independent component analysis of electroencephalographic data," in *Advances in neural information processing systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., vol. 8. The MIT Press, 1996, pp. 145–151.
- [17] S. Makeig, M. Westerfield, T. P. Jung, J. Covington, J. Townsend, T. J. Sejnowski, and E. Courchesne, "Functionally independent components of the late positive event-related potential during visual spatial attention." *The journal of neuroscience*, vol. 19, no. 7, pp. 2665–2680, Apr 1999.
- [18] T. P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, and T. J. Sejnowski, "Analysis and visualization of single-trial event-related potentials." *Human brain mapping*, vol. 14, no. 3, pp. 166–185, Nov 2001.
- [19] S. Makeig, M. Westerfield, T. P. Jung, S. Enghoff, J. Townsend, E. Courchesne, and T. J. Sejnowski, "Dynamic brain sources of visual evoked responses." *Science (New York, N.Y.)*, vol. 295, no. 5555, pp. 690–694, Jan 2002.
- [20] H. Nolan, R. Whelan, and R. Reilly, "FASTER: fully automated statistical thresholding for EEG artifact rejection," *Journal of Neuroscience Methods*, vol. 192, no. 1, pp. 152–162, Sept. 2010.
- [21] M. Junghöfer, T. Elbert, D. M. Tucker, and B. Rockstroh, "Statistical control of artifacts in dense array EEG/MEG studies," *Psychophysiology*, vol. 37, pp. 523–532, 2000.
- [22] J. R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.
- [23] W. B. Langdon and R. Poli, *Foundations of Genetic Programming*. Springer-Verlag, 2002.
- [24] R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*. Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008, (With contributions by J. R. Koza).
- [25] R. Poli, C. Cinel, L. Citi, and F. Sepulveda, "Reaction-time binning: a simple method for increasing the resolving power of ERP averages," *Psychophysiology*, vol. 47, pp. 467–485, 2010.