

[Re] How learning can guide evolution

Hrvoje Stojić¹

¹ Department of Economics and Business, Universitat Pompeu Fabra, Barcelona, Spain

hrvoje.stojic@protonmail.com

Editor

Name Surname

Reviewers

Name Surname

Name Surname

Received Aug, 20, 2017

Accepted Sep, 1, 2015

Published Sep, 1, 2015

Licence [CC-BY](#)

Competing Interests:

The authors have declared that no competing interests exist.

 [Article repository](#)

 [Code repository](#)

 [Data repository](#)

A reference implementation of

→ *How learning can guide evolution*, G. E. Hinton, and S. J. Nowlan, *Complex Systems*, 1 (3), 495-502, 1987.

Introduction

The Lamarckian hypothesis that adaptations accumulated during the lifetime of an individual organism are directly transmitted to the next generation is considered to be incorrect. The reference article [4] argues that this does not necessarily mean that individual learning cannot exert influence on evolution in an indirect fashion, providing a simple but convincing computational example of such an interaction. The authors use an extreme scenario where the optimization landscape is completely flat except for a single peak. In such an environment, curvature of the optimization surface does not provide any guidance and evolutionary search alone cannot find the maximum. Through a simulation the authors show that such a hostile environment can be tackled by a combination of evolutionary and individual learning. In addition, if organisms can search during their lifetime, those organisms whose genomes are closer to the targeted peak are going to be able to find the peak with individual search. Such organisms will have higher fitness and transmit their genes to the next generation. In effect, capacity for individual learning “creates a hill” to the peak that evolutionary search can climb, as illustrated in Figure 1 in Hinton and Nowlan [4]¹. This is one of the clearest illustrations of the benefits of individual learning for evolution and the interaction between the two learning processes.² This result has made a substantial impact in the cognitive science literature.

The main aim of this article is to replicate the simulations reported in the reference article [4]. I have corresponded with one of the authors and learned that the original implementation of the simulations is no longer available. I am also not aware of other implementations elsewhere. The simulations are relatively simple, and I thus propose a replication based on description from the reference article. Replication code relies on R programming language [6] and several R packages [8, 10, 9, 1, 2, 3, 7, 5].

Methods

A description of the simulations can be found in the caption of Figure 1 in the reference article [4]. I consider the description to be detailed enough for the implementation. I have followed it closely, with two exceptions. First, I have repeated the evolutionary

¹Note that captions of the figures are switched, caption of Figure 1 is incorrectly placed under Figure 2, and vice versa

²An indirect influence of individual learning on evolutionary learning is often called Baldwin effect.

search many times to average out potential noise. In the reference article, such iterations are not explicitly mentioned, and moreover, when interpreting their results from Figure 2, the authors state that Figure 2 presents results of “... a typical evolutionary search ...”, suggesting it is a single run of the simulation. Second, I let the population of agents evolve for 1000 generations instead of 50 generations reported in Hinton and Nowlan [4]. Reason for this will become clear in the Results section.

I describe my implementation in detail in Algorithm 1. Parameters for the simulation are summarized in Table 1.

Algorithm 1 Simulation description

```

1: for  $sim = 1 : noSim$  do
2:   Generate targeted genome,  $g^* = \{a_i\}_{i=1}^{noAlleles}$ :  $a_i \sim Bernoulli(p)$ ,
3:    $p \sim Uniform(0, 1)$ 
4:   Generate initial genomes of the agents,  $G = \{g_i\}_{i=1}^{noAgents}$ :  $\forall g_i$  generate
5:    $\{a_i\}_{i=1}^{noAlleles}$  by sampling with replacement from  $\{0, 1, ?\}$  according to  $p_0, p_1, p_?$ .
6:   for  $gen = 1 : noGenerations$  do
7:     for  $ag = 1 : noAgents$  do ▷ Individual learning
8:       for  $step = 1 : lifetime$  do
9:         For all ? alleles in agent's genome:  $a_i \sim Bernoulli(0.5)$ 
10:        if  $g_i == g^*$  after individual learning then
11:          break
12:        end if
13:      end for
14:      Evaluate fitness:  $f_{ag} = 1 - 19(lifetime - step)/1000$ 
15:      Record frequency of Correct, Incorrect and Undecided alleles.
16:    end for
17:    Compute parenting probabilities:
18:     $p_{ag} = f_{ag} / \sum_{i=1}^{noAgents} f_i$ ,  $\forall ag$ 
19:    for  $i = 1 : |G|$  do ▷ Generate children genomes
20:      Choose two parents by sampling with replacement according to  $p_{ag}$ 
21:      Generate new genome,  $g_i^{child}$ : randomly choose a cross-over point,
22:      copy all alleles from the first parent up to the cross-over point,
23:      and from the second parent beyond the cross-over point.
24:    end for
25:    Update genomes:  $g_i \leftarrow g_i^{child}$ ,  $\forall i \in G$ 
26:  end for
27: end for

```

Table 1: Simulation parameters.

Table	Value	Description
noSim	100	Number of times evolutionary search is repeated
noGenerations	1000	Number of generations for evolutionary algorithm
noAgents	1000	Number of agents in a population
lifetime	1000	Number of cycles available for individual learning
noAlleles	20	Number of alleles in agent's genome
p_0	0.25	Probability that allele is of type zero
p_1	0.25	Probability that allele is of type one
$p_?$	0.50	Probability that allele is of type ?

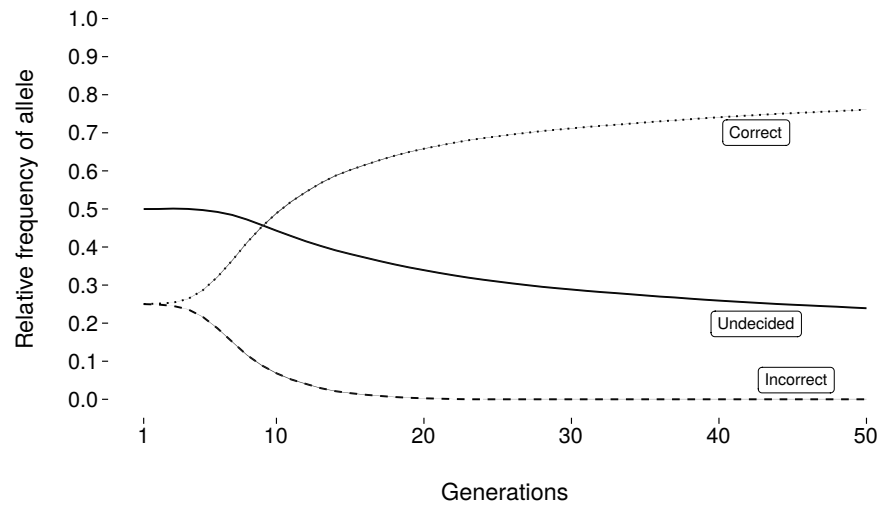


Figure 1: The evolution of the relative frequencies of the three possible types of allele. Correct alleles are those that are set by evolutionary learning and correspond to the targeted pattern. Incorrect alleles represent the proportion of incorrectly set alleles, while Undecided represent the proportion of alleles left for individual learning. Proportions are means of 100 simulation runs, and barely visible grey ribbons are standard errors. This is the reproduction of Figure 2 in the reference article [4].

Results

The simulation results are summarized in Figure 1, which corresponds to Figure 2 from the reference article. Qualitatively, the results are very similar - there is an obvious increase in the proportion of alleles correctly set by the evolutionary learning (close to 75% by generation 50) and proportion of incorrect alleles decreases to zero by generation 50. This is clear evidence of individual learning guiding evolutionary search, even though there is no direct transmission of knowledge acquired during individual learning from one generation to the next.

There is an important difference with respect to the reference article, however. In the original simulations, the proportion of Undecided alleles that are left to individual learning is relatively constant, close to initial 50%. The authors point this out as an interesting result, suggesting that there is little selective pressure to specify last few connections by evolutionary learning because those can be quickly set through individual learning. In contrast, the results in Figure 1 show that the proportion of Undecided alleles is continuously decreasing. In fact, whereas the proportion of Correct alleles in the reference article flattens out, in my case it continues increasing, with the increase coming from the proportion of Undecided alleles decreasing further. To make sure that this is really a long run trend I have let populations evolve for a larger number of generations than in the reference article. Results of all 1000 generations are presented in Figure 2. By generation 1000 the proportion of Undecided alleles drops to 9%. Even though the decrease is smaller with each generation, it does not seem to stop and flatten out by the end of the simulation.

What is the source of this particular difference in results? One explanation is that the results reported in Figure 2 in the reference article stem from a non-typical draw. This is unlikely, as upon examining the evolutionary paths of Undecided alleles in all simulation runs, they all show a strong downward trend already in the first 50 generations, differing only in the exact generation when this decrease onsets. Another explanation is that the original implementation is flawed as there actually is a selection pressure for eliminating individual learning after some time. The fitness function

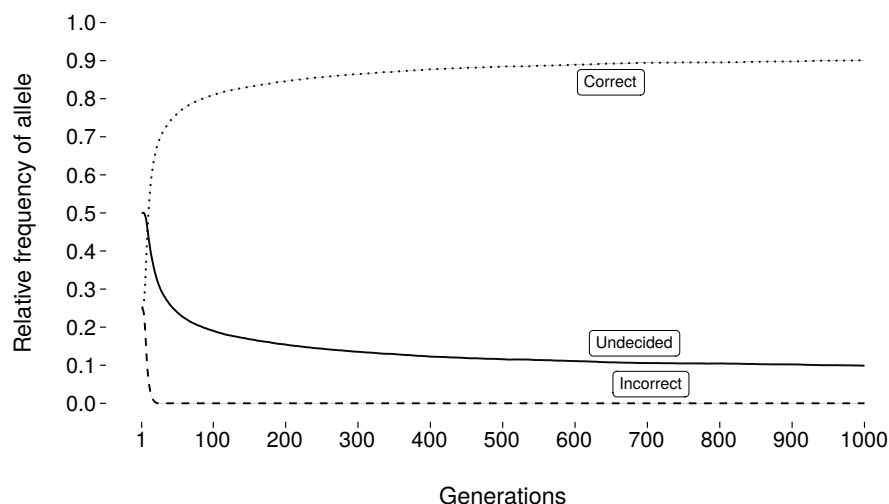


Figure 2: The evolution of the relative frequencies of the three possible types of allele, with a larger number of generations. Proportions are means of 100 simulation runs, and barely visible grey ribbons are standard errors.

described in the reference article is such that the smaller the number of steps an agent takes in an individual lifetime to reach the peak, the higher the probability of mating and transmitting genes to the next generation. The number of steps is a direct function of the number of undecided alleles in the genome. Hence, there is a selection pressure for decreasing the number of Undecided alleles. In personal correspondence, one of the authors agreed with this interpretation, stating that indeed there should be a selective pressure for a decrease in number of Undecided alleles. The pressure should decrease as a power function of the mean number of steps an agent needs to reach the peak, which explains rapidly decreasing improvements with each generation.

Conclusion

Qualitatively, the result is comparable to that reported in the reference article, but there are relatively important differences. The proportion of Undecided alleles in the implementation proposed here diminish over time, whereas in the original implementation they stay close to the initial values. I attribute the difference to a flaw in the original implementation, providing arguments based on a fitness function proposed in the reference article. Although the main result of the reference article - that individual learning can indirectly guide evolutionary search - has been reproduced, the difference in simulation results suggests that individual learning might be eradicated over time. This decrease in usefulness of individual learning occurs in static environments where targeted genome stays fixed, as in these simulations. In more realistic scenarios, where environments are non-stationary and the targeted genome would be changing over time, individual learning would have a more stable role.

References

- [1] Revolution Analytics and Steve Weston. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*. R package version 1.0.10. 2015. URL: <https://CRAN.R-project.org/package=doParallel>.
- [2] Revolution Analytics and Steve Weston. *foreach: Provides Foreach Looping Construct for R*. R package version 1.4.3. 2015. URL: <https://CRAN.R-project.org/package=foreach>.

- [3] Renaud Gaujoux. *doRNG: Generic Reproducible Parallel Backend for 'foreach' Loops*. R package version 1.6.6. 2017. URL: <https://CRAN.R-project.org/package=doRNG>.
- [4] Geoffrey E Hinton and Steven J Nowlan. "How learning can guide evolution". In: *Complex systems* 1.3 (1987), pp. 495–502.
- [5] Toby Dylan Hocking. *directlabels: Direct Labels for Multicolor Plots*. R package version 2017.03.31. 2017. URL: <https://CRAN.R-project.org/package=directlabels>.
- [6] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2017. URL: <https://www.R-project.org/>.
- [7] Kamil Slowikowski. *ggrepel: Repulsive Text and Label Geoms for 'ggplot2'*. R package version 0.6.5. 2016. URL: <https://CRAN.R-project.org/package=ggrepel>.
- [8] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN: 978-0-387-98140-6. URL: <http://ggplot2.org>.
- [9] Hadley Wickham. "Reshaping Data with the reshape Package". In: *Journal of Statistical Software* 21.12 (2007), pp. 1–20. URL: <http://www.jstatsoft.org/v21/i12/>.
- [10] Hadley Wickham and Romain Francois. *dplyr: A Grammar of Data Manipulation*. R package version 0.5.0. 2016. URL: <https://CRAN.R-project.org/package=dplyr>.