[Re] How learning can guide evolution

Hrvoje Stojić[1]

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

hrvoje.stojic@protonmail.com

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 [5] 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. For organisms that 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 [5][1]. In contrast to the Lamarckian hypothesis, this is an indirect transmission of individual learning on evolutionary learning, what is often called the Baldwin effect [3]. This is one of the clearest illustrations of the benefits of individual learning for evolution and the interaction between the two learning processes. This article is well regarded and widely cited in cognitive science literature.

I replicated the simulations reported in the reference article [5] in R [6], using several packages [10, 8, 11, 9, 1, 2, 4, 7]. In correspondence with Geoffrey Hinton I learned that the original implementation is not available.

Methods

A description of the simulations can be found in the caption of Figure 1 in the reference article [5]. 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 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

---

[1]Note that captions of the figures are switched in the reference article, caption of Figure 1 is incorrectly placed under Figure 2, and vice versa.

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 [5]. Reason for this will become clear in the Results section.

Pseudocode of my implementation is shown 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, $G = \{g_i\}_{i=1}^{noAgents}$: for all 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:         $p_{ag} = f_{ag} / \sum_{i=1}^{noAgents} f_i$, for every agent ag
19:         **for** $i = 1 : noAgents$ **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: $g_i \leftarrow g_i^{child}$, for every genome 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's value is 0
p_1	0.25	Probability that allele's value is 1
$p_?$	0.50	Probability that allele's value is ? (learnable)

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
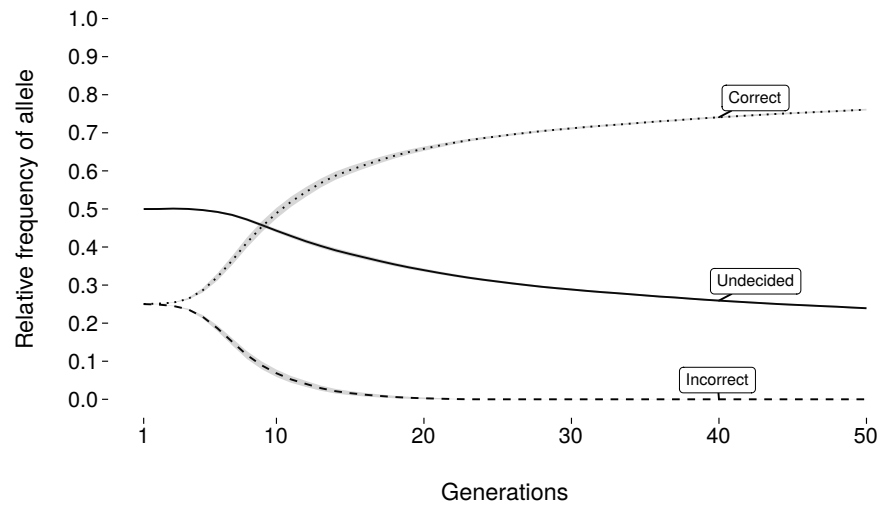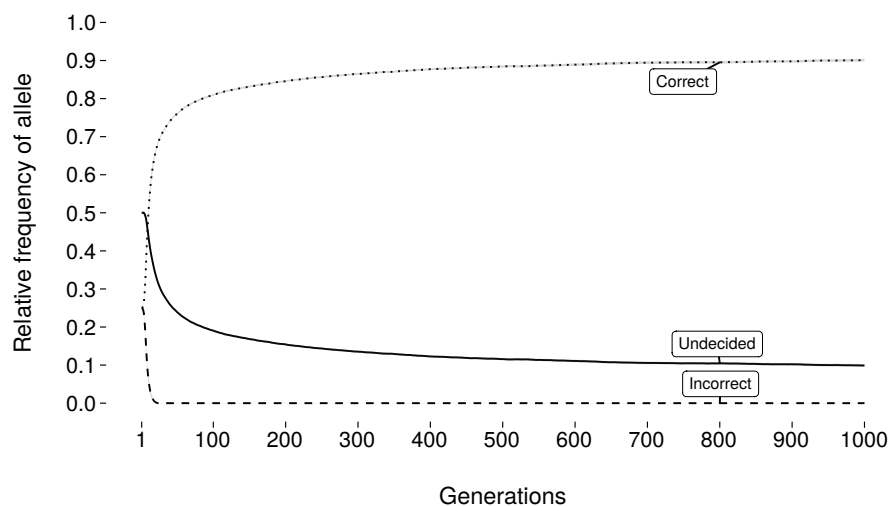
Figure 1: The evolution of the relative frequencies of the three possible types of allele. Correct alleles are those that are set by evolutionary learning and correspond to the targeted pattern. Incorrect alleles represent the proportion of incorrectly set alleles, while Undecided represent the proportion of alleles left for individual learning. Proportions are means of 100 simulation runs, and grey shaded area around the mean line represent standard errors of the means. This is the reproduction of Figure 2 in the reference article [5].

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 mean proportion of Undecided alleles drops to 9%, with the decrease being smaller with each generation.

What is the source of this particular difference in results? The results reported in Figure 2 in the reference article might stem from a non-typical draw. This is unlikely, as illustrated in Figure 3. Evolutionary paths of Undecided alleles in all simulation runs show a strong downward trend already in the first 50 generations, differing only in the exact generation when this decrease onsets. As shown in Figure 4, distribution of proportions of Undecided alleles at the end of the simulation (1000 generations) ranges from 0.05 to 0.20, much lower than 0.46 that was obtained in the reference article.

The decrease in Undecided alleles is actually expected, once the fitness function described in the reference article is examined more closely. The function 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

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 grey shaded area around the mean line represent standard errors of the means.
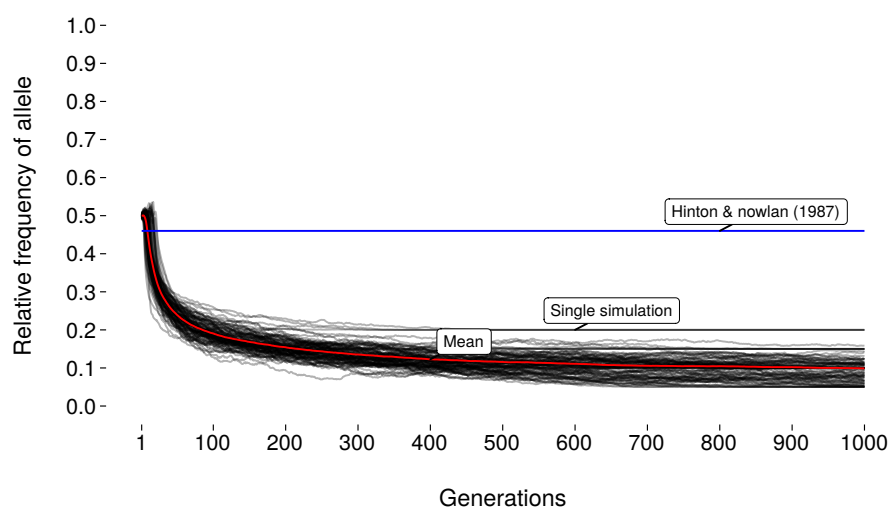
Figure 3: The evolution of relative frequencies of Undecided alleles for all 100 simulations, red line marks the mean across all simulations, same as in Figure 2, while blue line marks approximately the end point of relative frequency from the reference article.
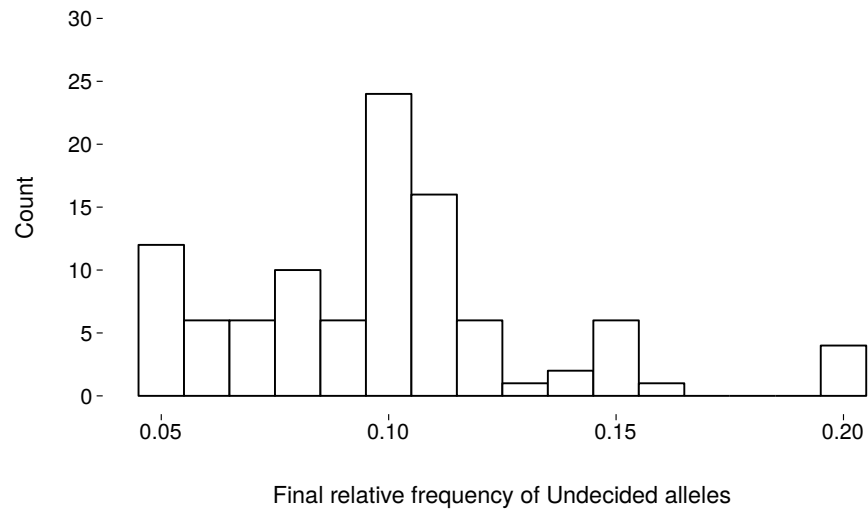
Figure 4: Histogram of relative frequencies of Undecided alleles at the end of simulations (1000 generations).

generation. The number of steps is a direct function of the number of undecided alleles in the genome. In other words, there exists a selection pressure for decreasing the number of Undecided alleles. In personal correspondence, Geoffrey Hinton agreed with this argument, stating that the selective pressure should decrease as a power function of the mean number of steps an agent needs to reach the peak. Computational power at the time was orders of magnitude smaller than today, preventing them to explore the space more thoroughly.

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 have provided an argument based on a fitness function proposed in the reference article that clearly explains the decrease. 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] M J Baldwin. "A new factor in evolution". In: *American Naturalist* 30 (1896), pp. 441–451.

[4] 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.

[5] Geoffrey E Hinton and Steven J Nowlan. "How learning can guide evolution". In: *Complex systems* 1.3 (1987), pp. 495–502.

[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 Winston Chang. *devtools: Tools to Make Developing R Packages Easier*. R package version 1.12.0. 2016. URL: https://CRAN.R-project.org/package=devtools.

[11] 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.