

Natural Selection Simulation Writeup

Niel Ok

December 2024

1 Introduction

If you watched *The Tomorrow War*, you definitely noticed the horrendous, supposedly alien creatures called "spikes". Well, I noticed them too. People struggled so much fighting against these creatures that they literally had to recruit soldiers from the past to fight them. This is not the only movie with lethal alien creatures. The entire *Alien* series comes to mind. The "spikes" also had weaknesses, however. For example, shooting it in the neck killed it. All of this got me thinking, "if we come to a new environment, how can we model what the apex predator of that environment might look like?" The idea was, if we could model that, we would be better prepared to fight apex predators if we ran into dangerous life in space exploration or in the oceans. So, I built a simulation of natural selection and did some probabilistic analysis on it, although the simulation is far too simplistic to be useful due to limits of Python and the computational limits of my 2023 Macbook Pro. By the way, I do not use machine learning libraries here. Bootstrapping and logistic regression were written from scratch, as well as the natural selection simulation (I mean, I use numpy and random, but come on).

2 The Simulation

Here is the whole GitHub repository where the natural selection simulation, data, and analysis is located: [GitHub Repo Link](#). The actual simulation code is located in the 'simulation/' directory in the two Python scripts. I will now give a brief overview of how the simulation works. At a high level, the user defines the initial environment for the simulation with parameters such as the initial number of organisms, initial amount of food for every generation, the area of the environment, the harshness of the environment (basically, the probability of an organism dying based off of extraneous factors such as disease), and the initial traits of each organism. Each organism is an object that can perform 4 different actions: reproduce, hunt, gather food, and die. Each organism has three traits: speed, size, and sense, and those traits determine things such as how much energy the organism spends hunting and gathering food, and they also play a big role in the hunting and food gathering mechanisms. The user specifies how many generations they want the simulation to run for, and in each generation, the organism's energy resets and the organism has the opportunity to reproduce, hunt, and gather food. Hunting and gathering food give energy to the organism while reproducing takes energy away. If the organism runs out of energy after a generation, it dies. When reproducing, a random mutation is applied to the organism's traits before passing them on to its descendant. At the end of the simulation, you have a list of the organisms that have survived until that point. Now, to make this work reasonably well at the organism-level, I had to borrow some ideas from logistic regression (squeezing values between 0 and 1 with sigmoid, between -1 and 1 with tanh) and tweak some parameters. At the simulation-level, I had to make some weird optimizations to make the code run fast enough for larger amounts of generations. But, after it all, I was at least consistently able to get 20-generation simulations running fast. I got 20-generation simulations from being something that would have taken 3-4 hours to something that takes seconds. Obviously, this simulation is a simplification from reality, but it is a start, especially given that I do not have much computational power. There are a lot of mechanisms working together that make this system pretty complex, to the point where I am fairly confident that the data I see is mostly a result of the equations I use to govern the behavior of the system, not because of the behavior of the random library or anything like that. Now, we can do some probability.

3 Confirming Expected Behaviors with Probability

Now, with a natural selection simulation, there are some behaviors that I think should be expected (disclaimer: all of these expected behaviors are based on raw intuition), and these behaviors can be checked with the amazing power of bootstrapping and p-values.

3.1 Experiment 1: Similar Populations

One behavior we would expect is that, for two simulations, if we had two exactly similar populations under the exact same simulation parameters and evolve them for some generations, it should look like those two populations are

from the same overall population or something close for at least 1 or 2 generations into evolution. In other words, if we bootstrapped assuming that the two populations were actually from one super population and calculated p-values for the differences in traits of those two populations, we would expect those p-values to be fairly high and therefore not statistically significant. An analogy for this experiment would be if we had two parallel worlds where we were examining how monkeys evolved into humans. Earlier on, we would expect the monkeys from both worlds to all look fairly similar. The data for this simulation is in database 2. Basically, what I did was I took a specific organism and environment configuration and ran the same simulation twice, letting the organisms evolve for one generation in each simulation. Then, I bootstrapped, assuming the two populations were from one super population, and calculated absolute difference of means for each of the speed, size, and sense traits over 50000 iterations. Then, I calculated the actual absolute difference of means for the three traits between the two simulated populations, and found their p-values. Here are charts for the bootstrapped differences of means for the 3 traits:

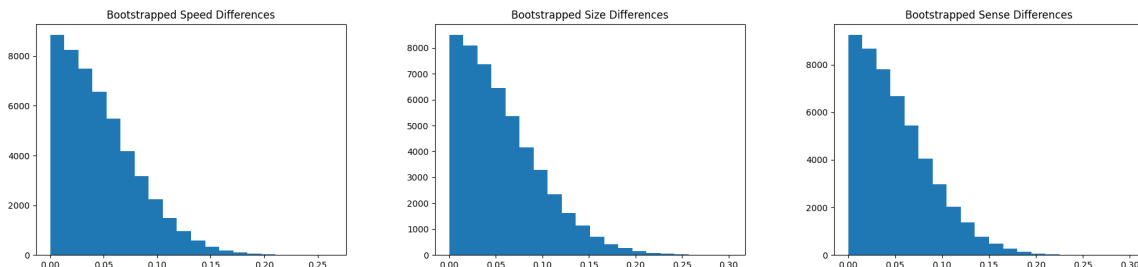


Figure 1: Bootstrapped Differences of Means for 3 Traits

The actual differences I got were 0.0835 for speed, 0.2475 for size, and 0.2004 for sense, where the size of the population from simulation 1 was 201 and the size of the population from simulation 2 was 200. The p-value I got for speed was 0.15876, which is not statistically significant, for size I got 0.00064 and for sense I got 0.0018, which are both technically statistically significant. We will see later that these are large values, however. Looking at the p-values is telling me that, at least for speed, it would be an overstep to say that these two populations are from different distributions, which is what we predicted. Yay!

3.2 Experiment 2: Long-Term Divergence

I ran the same experiment as I did above, but only changing one thing. I changed the number of generations each simulation was run for from 1 generation to 10 generations. Basically with this, I wanted to show that, even starting from the same population under the same conditions, in two different simulations of evolution, with enough generations, due to the random nature of evolution, the populations will diverge to the point of being different in a statistically significant way. To save space, I will not show the charts, but in terms of shape, they look similar to the ones above. The p-values I got here were 0.0 for every single trait, meaning that in 50000 samples, not a single one showed a absolute difference in means greater than or equal to the actual absolute difference in means in the traits we saw. For reference, the size of population 1 was 437 and the size of population 2 was 176. This difference is statistically significant, more so than the difference from experiment 1, which is what we would expect. I tried 1000000 iterations at some point, but the p-values were still 0.

3.3 Experiment 3: Differing Environments

I will keep things brief here since I want to save space. I basically ran two identical populations for the same amount of generations (just because I could with my optimizations, I ran each simulation for 20 generations) under 2 different environments, ran the same bootstrapping and p-value calculation pipeline for the two populations, and found the p-values to be 0.0 for each trait, as expected. With this one, I basically wanted to see if two populations would diverge if put in different environments. For reference, the size of population 1 here was 309 and the size of population 2 was 82. Notice, the number of organisms in each population is pretty different but I am not too worried because rule of thumb says we should be fine after about 30 datapoints.

4 Classification

We have looked into some interesting mechanisms of evolution using bootstrapping and p-values, but we still have not really gotten into the core of what we want to do, which is to make some sort of predictions with our data. While I do not have the full story of neural networks to work with, we thankfully can create a classification problem, which logistic regression works well with. We can say, given an organism and its traits, classify which simulation's

population it comes from, where we are looking at populations from two different simulations. Some baseline things: I added support for measuring intermediate accuracies by getting intermediate thetas and calculating accuracies afterwards, and I also decided to look at the data to help me determine model behavior. I also added baseline offset to the regression model.

4.1 Regression for Similar Populations

Remember experiment 1 from Section 3 and how we noticed that in certain traits, the two populations did not have a statistically significant difference after simulating just one generation of evolution? Well, I tried running regression on it for 10000 steps with alpha 0.01. Here are the results:

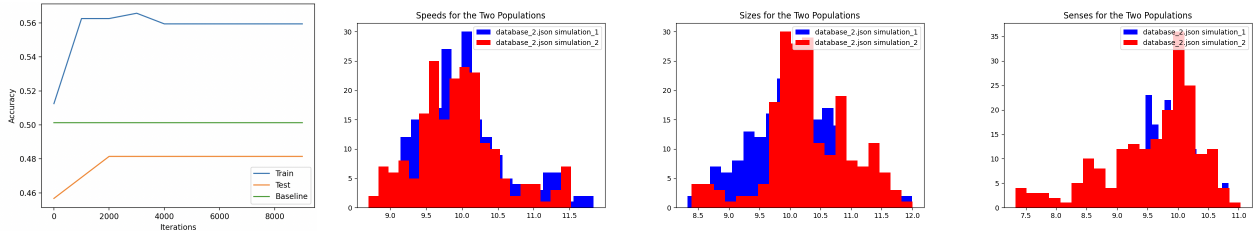


Figure 2: Regression Results for Similar Populations

4.2 Regression for Differing Environments

Remember experiment 3 from Section 3 and how we found that, when put in different environments for 20 generations, two identical populations evolved to become different in a statistically significant way? Here are the results from running regression on it for 10000 steps with alpha 0.01:

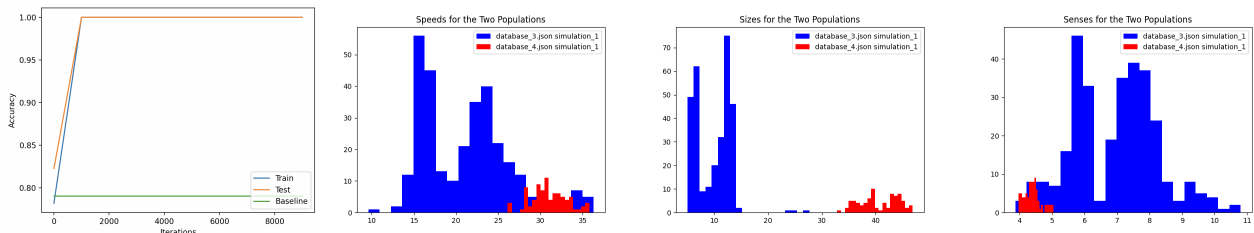


Figure 3: Regression Results for Differing Environments

4.3 Regression Analysis

You will notice that regression for similar populations was much more difficult than regression for differing environments just based on the accuracy of the regression model across the two. One is at basically baseline while the other is literally perfect. Looking closer at the data can give us a clue. For the similar populations, we can see that each of the three traits overlap heavily, while for the differing environments, at least one of the traits (size) is completely isolated between the two populations, as in there is no intersection. Thus, the differing environments regression model is probably able to be perfect because it trains to look at this completely isolated trait, while there is no such trait and no good trait separation in the similar populations data. I do not have space to go into calibration of the probabilities, but that would be a next step. I surmise that, because the accuracies are so extreme for both models, the probabilities will not be well-calibrated.

5 Conclusion

There are so many ways I could expand upon this, like looking at how fast two populations diverge under different conditions or restructuring my database saving code to make it so that I can look at the same population at different points in its evolutionary journey. I will do so if I have time and additional compute in the future. Right now, though, I am happy with the questions I have been able to answer with this simple simulation and analysis.