MAT 421 Final Project

Brodie McCarthy

April 2024

1 Introduction

In an attempt to explore the complex interactions that come up within ecosystems, mathematical models such as the Lotka-Volterra set of differential equations provide a structured and quantitative framework for helping to understand the sensitive dynamics of predator-prey relationships. Numerical solutions and simulations are often a cornerstone of studying sets of differential equations so I set out to see how effective these simulations of an ecosystem from an actor-oriented simulation would be. I wanted to test if by creating a model using ideas such as ordinary differential equations, probability, and other mathematical concepts could be strung together to create a reasonably comprehensive and accurate model.

2 Concepts

There were a number of different concepts from the class investigated for this assignment. These topics ranged from basic probability, differentiation, differential equations, among others.

2.1 Probability

While not the most complex concept to this project, the idea of probability and random chance played a pivotal role inside of this assignment.

Probability is the study of uncertainty and chance by way of looking at the possible outcomes for a given system and determining the chance of these outcomes occurring. There exist different distributions such as the binomial or normal distribution which can determine where outcomes are likely to fall on a given range, however this project focused mostly on just random chance.

In an attempt to simulate real life, not every possible scenario could be captured. When it comes to real life hunts, any number of circumstances can change the outcome of a hunt whether the terrain caused a creature to trip or if an animal was simply sick or tired. To account for this uncertainty, I used weighted random chance as the backbone of these simulations. Where a predator's speed relative to the prey's speed would weight the chance of success but then it was up to whether that roll of the dice to see if that animal was successful

The Python library "random" was used to generate these random chances. Because computers can very rarely actually generate truly random numbers, I would set various "seeds" for Python to use to allow for replication of various trials.

2.2 Differentiation and Optimization

A key idea in this project comes from the conversion of purely numerical simulations to the more theoretical and variable based equations. Optimization was very useful as a way of finding the value for variables in our system of equations that will match our numerical simulations.

Differentiation is the process of trying to find a function's derivative which is a measure for the rate of change of that function's output, typically at a specific point a. This is expressed as:

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

Optimization heavily utilizes this idea of differentiation as it is the process of trying to find the "minimums" or "maximums" of a function. This means that it tries to find where the function reaches any relative or absolute minimum or maximum values. It does this by trying to find where the slope or derivative is equal to zero and the two values to the left or right of that function are opposite signs as this indicates that the function has reversed directions. In real life, this is useful for if you have a function predicting the output from a factory given different parameters like number of workers or number of hours given in breaks. Optimization would look to find the peaks or minimums of production. In this project, it was used to find the values that minimized the total error when comparing the numerical solutions to the analytical solutions.

An optimization algorithm that is very commonly used is the Gradient Descent Algorithm which is a first order iterative algorithm which can be used to find absolute or local minimum. This is denoted as:

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$

where: x_n is the current point, γ is the step size or learning rate, and $\nabla f(x_n)$ is the gradient of f at x_n, representing the direction of the steepest ascent. This kind of model isn't too different from the main model I used from the Python library of "Scipy" where I utilized the minimize function to try and find the minimum error of my numerical calculations for the variables in my analytical equations.

2.3 Ordinary Differential Equations

Ordinary Differential Equations (ODEs) are systems of equations that involve functions and their derivatives to a certain n-th order. An ODE describes the relationship between a function $y(x)$, which is dependent on an independent variable x and its derivatives. The general form of an n-the order ODE is given by:

$$F\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \ldots, \frac{d^ny}{dx^n}\right) = 0,$$

where F is a function that combines these elements in some manner. These Ordinary Differential Equations lay at the heart of this project and were integral to the problem at hand. ODEs are used to describe relationships of a whole allotment of scenarios. They're useful in describing how the position or number of variables can change based on some previous iteration or position.

One of the most famous sets of equations is the Lotka-Volterra model which models the dynamics of a predator and prey's population over time.

$$\frac{dx}{dt} = ax - bxy$$
$$\frac{dy}{dt} = cxy - dy$$

where x is the number of a certain prey animal at a given time and y is the number of the predator in the ecosystem. As a whole, the system of equations tells a story for the prey of a relatively constant growth of a prey population, ax, against some rate of predation by the predator dependent on the population of both, $-bxy$. And the predator's growth is determined by a similar rate of predation, cxy, against some constant rate of death, $-dy$. This allows for an oscillation of when the prey's population is high relative to the predator's population, the predator's growth rate increases due to an abundance of food. And when the prey's population is low relative to the predator's population, then the second $-dy$ term is larger and thus shows a decrease in the number of predators due to a lack of resources. While simple in nature, they can tend to explain overall phenomenon in shifting populations.

3 Equations

I initially looked at two sets of equations, one for just the model between a single predator and prey and then the other with an apex predator, a middle predator, and then a prey. These were all modeled after the simple Lotka-Volterra models.

$$\frac{dR}{dt} = \alpha R - \nu RW \tag{1}$$
$$\frac{dW}{dt} = \gamma RW - \zeta W \tag{2}$$

Where R and W are the number of rabbits and wolves at a given point in time, α and γ are the parameters that determine the rate of growth of the rabbits on their own and then γ for the wolves which is dependent on the number of rabbits at that time. ν and ζ are the rates of shrinkage for a population where ν is determined by the efficiency of the wolves' predation and ζ is some natural rate of decreasing wolf population that represents some old age or starvation.

$$\frac{dR}{dt} = \alpha R - \mu RF - \nu RW \tag{3}$$

$$\frac{dF}{dt} = \beta RF - \epsilon F - \theta FW \tag{4}$$

$$\frac{dW}{dt} = \gamma(RW + FW) - \zeta W \tag{5}$$

Where R, F, and W are the number of rabbits, foxes, and wolves on a given day. The variables α, β, and γ are each parameters that determine the effect of re-population on that specific animal. The variables μ, ν, and θ are each variables associated with the given efficiency of predation by their larger animal(s). And the variables ϵ and ζ are just some natural rate of death for both the fox and the wolf respectively.

Now the reason I chose to implement the equation for wolves with both a RW and FW term is due to the simplification of terms and the fact that I wanted γ to encompass that the wolves rate of predation towards rabbits would also be hindered by the number of foxes hunting them as well.

4 Data

For data, I primarily made my own through numerical simulations but I also drew upon an older set of data by the Hudson Bay Company from the 19th to 20th century on the number of hare and lynx pelts that were trapped from 1845 to 1935. This dataset comes from Canadian company and has been used throughout mathematics to showcase the Lotka-Volterra models as a way of extrapolating the number of pelts caught each year to the number of each animal present in the environment. Each column is delineated as a decimal number but is meant to be converted by being in the order of thousands.

This data is available here: http://people.whitman.edu/ hundledr/courses/M250F03/LynxHare.txt [1]

5 Implementation and Results

Here, I will examine the fitting of simple Lotka-Volterra model to the Lynx-Hare dataset as well as a simulated environment of my own using Python.

5.1 Lynx-Hare Exploration and Analysis

Through the analysis of the Lynx-Hare dataset, we can see a pretty spot on correlation between the number of hares and lynxes in this time frame. We see that as the number of hare pelts increase, so too does the number of lynx pelts eventually. But then as the number of lynx pelts increases, eventually the hare pelts begins to decline. And as the hare pelts begin to decline, so too does the lynx pelts and so on and so forth.

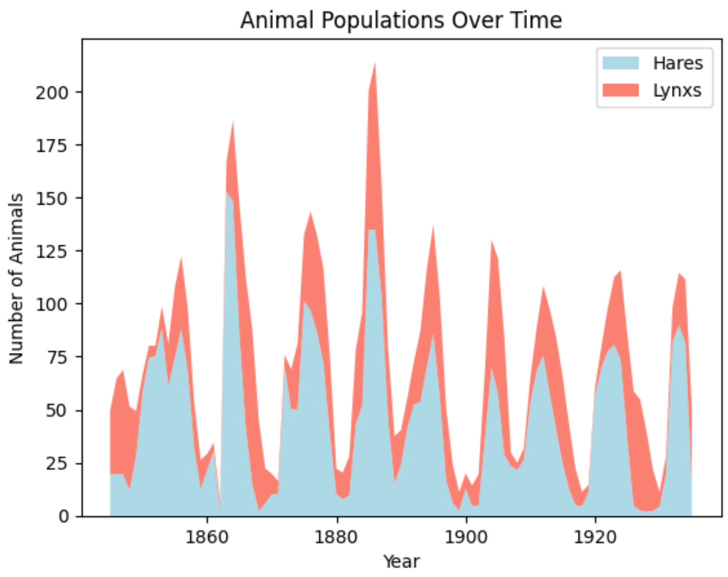

Figure 1: Lynx-Hare Population over time

The next step was to model these equations as the standard Lotka-Volterra equations from above and then numerically compute the parameter estimates to see if we can recreate the actual data. This was done using the scipy "odeint" function to simulate the model as well as the "minimize" function from before to minimize the error between the actual and the estimated values. Here we just use the essentially the Least Squares method to be our estimator.

```
def cost_function(params, data, t): # A function to find the error of each estimation
    solution = odeint(derivs, [data.iloc[0, 0], data.iloc[0, 1]],
        t, args=tuple(params))
    return np.sum((solution - data.values)**2) #

t = data.index.values

initial_params = [0.1, 0.1, 0.1, 0.5]

result = minimize(cost_function, initial_params, args=(data, t))

optimized = result.x
```

However when simulating a very simple Lotka-Volterra model, our end output ends up rather lacking in terms of predictability. As seen in the graph below, it becomes rather apparent that our model fails to accurately encapsulate each of the cycles between high and low populations.

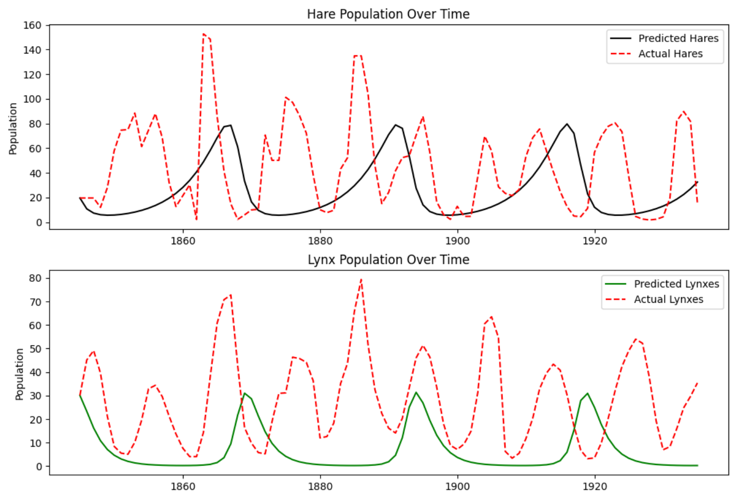

Figure 2: Lynx-Hare Predictions vs Reality

Our model barely follows the simple predator-prey models compared to the more complex data found with the Lynx-Hare dataset. This is due to the fact that these parameters of a, b, c, and d are constants when in reality, they could each be full of any number of parameters that differ far more than just a pure constant. This leads me to the majority of my project where I worked to create my own numerical simulations.

5.2 Simulated Predator-Prey

The simulation of these predator-prey models proved to be the hardest part. I created classes for predators and prey which each of their own methods for hunting and whether a prey could flee or not. I also added in a mechanic where animals, upon finding food and reproducing, were able to have their attributes "genetically mutate" by changing. This also allowed for me to see natural selection in action and to see if changes to an animal's survivability would change these predator-prey population models.

I made survival based on a creature's speed relative to its pursuer and its ability to hide and avoid a pursuit at all based on the prey's camouflage versus the predator's senses. Then after each iteration or "day", the offspring's speed would increase or decrease causing the other attribute of camouflage or perception to do the opposite.

Over the course of many different iterations, I found that speed became

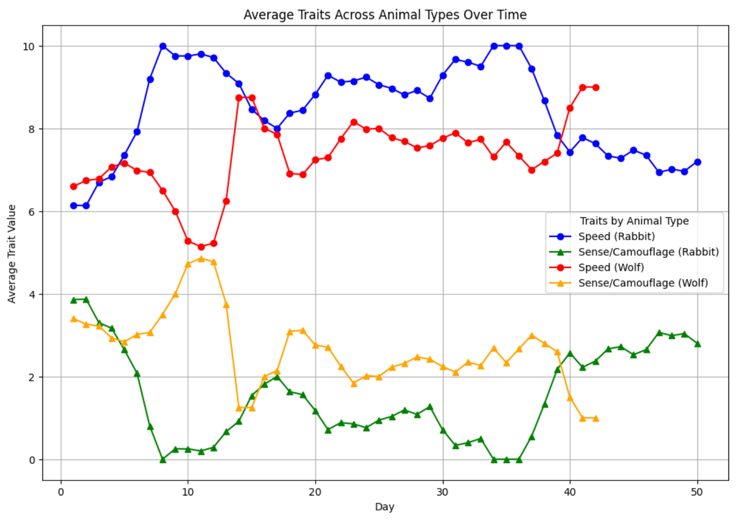

Figure 3: Wolf and Rabbit Traits Over Time

the determining factor for rabbits as the average among the population would always drift towards the limit of 10. Wolves would often have to compete by also raising their speed but they usually faced some difficulty as some perception was necessary to spot the rabbits at all meaning they were always a little slower.

However, despite the changes in their traits over time, the population models actually were remarkably similar to Lotka-Volterra models. They would sometimes perform the oscillations that you would expect with each population rising and then falling just slightly lagging behind the other. One discrepancy however was that my model would regularly see that one predator or prey population succeed and completely outlast the other. Wolves would often over-hunt and thus starve as all of the rabbits were gone and rabbits would sometimes evade the wolves into starvation thus leaving their own populations out of check.

Here we see one of the cases where at first the populations start to oscillate but eventually rabbits completely evaded the wolves and thus led to an uninhibited exponential population growth of the rabbits over time. I chalked this up to the fact that my model regularly used weighted random chances to determine the success of a hunt or not meaning that in many scenarios, there would be a case where a population that should have survived in theory, did not survive in practice. I thought that this was necessary as while my model was not perfect by any stretch of the imagination, it was marginally more realistic as it showcased the variability of nature.

5.3 The Middle (Meso)Predator

The main point that I wanted to investigate in this project was what would happen if we introduced a third actor, the fox. This fox would act as a "meso-

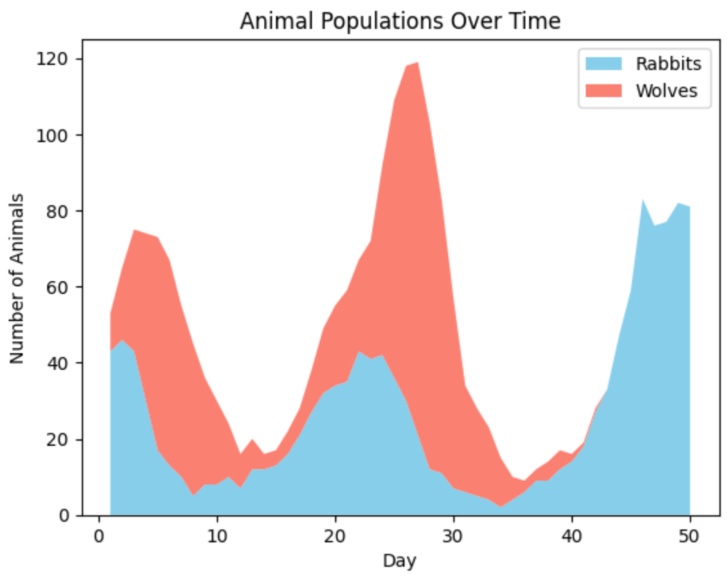

Figure 4: Wolf and Rabbit Population Example

predator" where they could hunt rabbits but could also be hunted by wolves, a still apex-predator in the system. Now the wolf had the opportunity to hunt not just the rabbit but also the fox, and the fox had to now investigate whether it would be better off being purely fast or also balance its perception and camouflage.

Now with most iterations of this model, I saw that one of the predators was the most common to go first(typically the foxes) but there were also cases in which the rabbits were simply hunted to extinction as the wolves would eat the foxes who were eating the rabbits. With twice as many predators, rabbits would just die out fairly quickly and then the foxes would become the new prey as they were just at the bottom of the food web.

Other cases where the foxes were the first to go were the most common where they simply could not balance both speed and camouflage and their own perception skills as they needed to be good at everything to survive. (Figure 6) In fact, another thing that was of note was that even if I set up the simulation to go on for over 30 days, it would very rarely ever reach that. Most iterations died out before the time even got to be 15 days. I credit this to a factor that determined whether or not how successful a hunt would be regardless of how big the difference of speeds would be. When run with the same initial conditions and the same starting random seed, a difference of 0.1 for this "escape chance factor" could mean the difference between a decently thriving ecosystem with oscillations versus a system that collapsed before the week was over. Creating this balance was very hard for numerical simulations as there aren't a lot of real world data to correct this to be in accordance with.

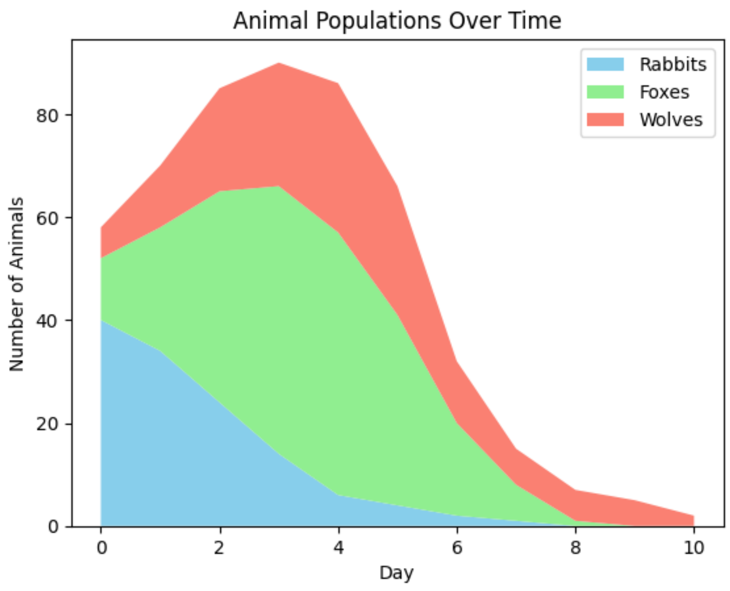

Figure 5: Wolf, Rabbit, and Fox Population Example

However, when it came to the fitting of my numerical simulations to my analytical equations, I tended to have much better luck here. In cases where the populations died out rather quickly, the fitted equations often matched up very accurately. This was nice to see that my models were rather precise at given initial conditions compared to the simple Lotka-Volterra model used earlier with the Lynx-Hares dataset. With this code, we can see that much of our predicted outputs matched the actual ”observed” data even going towards 0 as time went to infinity.

6 Conclusion

While this project was by no means exhaustive and likely missed many things that could have made this more accurate, it did help showcase the importance of trying to control as many different factors as possible when creating an analytical model as well as the importance of actual observed data for these population dynamics. This field is often lacking in these adequate datasets but for good reason due to just how hard it is to not just collect this data, but also control as many other variables as there are when it comes to trying to calculate an entire ecosystem. My model highlighted that constructing these equations and simulations is a highly sensitive endeavor as well as the limited importance of an animal adapting over time changing the effectiveness of the Lotka-Volterra model. While the animals performance in my model was based largely on some evolutionary factors, they could be counteracted by the use of parameters when

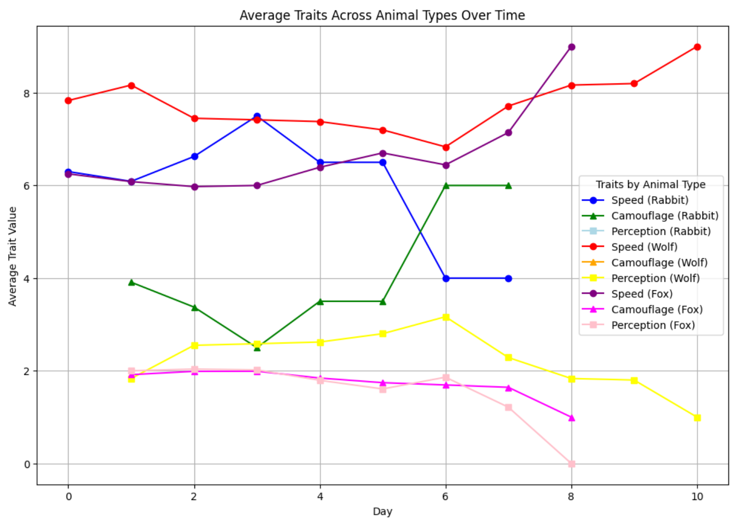

Figure 6: Wolf, Rabbit, and Fox Skills Example

trying to construct a perfectly oscillating system.

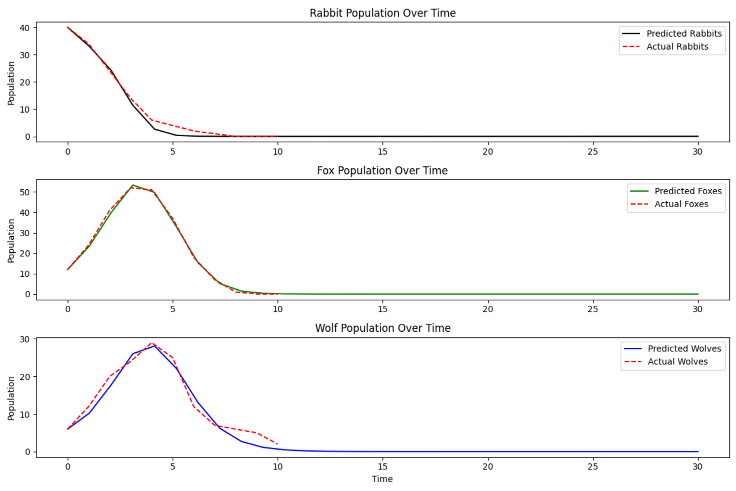

Figure 7: Wolf, Rabbit, and Fox Predictions

References

[1] Michael J. Kirby and Gerhard Dangelmayr. *Introduction to Mathematical Modeling*. 2003. URL: `http://people.whitman.edu/~hundledr/courses/M250F03/M250.html` (visited on 08/12/2003).