Proj 3: Evolutionary Dynamics

1 Introduction

Nonlinear dynamics is a fascinating area for both theoretical analysis and numerics. Almost any corner of science or engineering that considers how things evolve in time[1] features some set of favorite nonlinear model problems that illustrate interesting qualitative features. In evolutionary game theory, an area connected to both mathematical biology and economics, there are a wide variety of such models. These include the *replicator equations*, the *Lotka-Volterra equations*, and many others. In this project, we will consider a variant of the Lotka-Volterra equations inspired by a recent paper by Toupo, Strogatz, Cohen, and Rand[2]. I would be surprised if this model is totally new, but know of no related work. I would welcome any comments about related models in the literature that you might come across as you work on the project.

1.1 Single strategy case

We begin with a case that is simple enough to analyze by hand. As a motivating story, we consider a population of x foragers searching for food resources. The quantity of food resources r varies depending on the action of the foragers. We characterize foragers by two closely related rates: a rate of reproduction $\alpha(r)$ and a rate of consumption $\beta(r)$. We assume $\beta(r) = 0$ for $r = 0$ and that β is smooth and monotonically increasing; and for this project, we will use $\alpha(r) = \beta(r) - \phi$ where $\phi > 0$ is some basic level of resource consumption required for continued comfortable existence; populations with less than ϕ resources per individual decline, while populations with more grow. The coupled dynamics of the resource and the population are given by

$$
\begin{aligned}
\dot{x} &= \alpha(r)x = (\beta(r) - \phi)x \\
\dot{r} &= \rho - \beta(r)x.
\end{aligned}
$$

[1] That is, almost any corner of science or engineering.

[2] "Evolutionary game dynamics of controlled and automatic decision-making." *Chaos*, vol 25.7 (2015), 071320; also available as `http://arxiv.org/abs/1507.01561`.

Since we have talked not at all about dynamics in this class, we will focus on *equilibrium* solutions where $\dot{x} = 0$ and $\dot{r} = 0$. These correspond to solutions to the nonlinear equations

$$\begin{aligned}(\beta(r) - \phi)x &= 0 \\ \rho - \beta(r)x &= 0.\end{aligned}$$

We assume $\rho > 0$ and $\beta(r) > 0$ for $r > 0$. Under these assumptions, an equilibrium (if one exists) must satisfy

$$x = \frac{\rho}{\phi} \quad \text{and} \quad \beta(r) = \phi.$$

When an equilibrium exists, we analyze its *linear stability* by looking at a Jacobian matrix at the equilibrium:

$$\frac{\partial}{\partial(x,r)} \begin{bmatrix} \dot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & \beta'(r)x \\ -\beta(r) & -\beta'(r)x \end{bmatrix}.$$

Even knowing only that $\beta(r)$, $\beta'(r)$, and x are all positive numbers, we can tell from the structure of this matrix that the eigenvalues have negative real part. Hence, the equilibrium is linearly stable.

1.2 Competing strategies

Now suppose our foragers follow varying strategies: some may be slower but more systematic in seeking food, while others are quicker. In addition, we allow foragers to *mutate* or randomly change strategies at some rate. The system of differential equations governing the system is

$$\begin{aligned}\dot{x} &= (M + \mathrm{diag}(a(r)))\, x \\ \dot{r} &= \rho - b(r)^T x\end{aligned}$$

where $r(t) \in \mathbb{R}$ and ρ are as before and

- $r(t) \in \mathbb{R}$ is the amount of a shared resource at time t
- $x(t) \in \mathbb{R}^n$ is the distribution of strategies in the system
- $a(r) \in \mathbb{R}^n$ is a vector of growth rates for varying strategies

- $b(r) \in \mathbb{R}^n$ is a vector of consumption rates for varying strategies
- M is a mutation matrix: m_{ij} is a rate at which strategy j mutates to strategy i. Columns of M must sum to zero.

What happens if there is no mutation? In this case (usually), the only equilibrium solutions will involve exactly one strategy (i.e. one x_j is positive), and the solution will stable if $a_k(r) < 0$. Hence, absent mutations or other terms that cause different strategies to interact directly (instead of just through r), the competing strategy case does not provide any interesting new equilibrium behaviors beyond those that we see in the single-strategy case.

1.3 Modeling resource acquisition

So far, we have not discussed the specific form of $\beta(r)$. We will use a model with one parameter, representing the carefulness of the search. Specifically, we assume that a search takes some minimal time $\tau > 0$, and a search with time $\tau + \sigma$ succeeds[3] with probability $1 - \exp(-\sigma r)$. With repeated searches, the expected rate of resource acquisition is

$$\beta(r; \sigma) = \frac{1 - \exp(-\sigma r)}{\tau + \sigma},$$

This has the desired behavior that $\beta(0) = 0$, and for any r

$$\beta'(r; \sigma) = \frac{\sigma}{\sigma + \tau} \exp(-\sigma r) > 0.$$

As the number of resources grows large, we find that β approaches $1/(\sigma + \tau)$ from below. We can also solve $\beta(r) = \phi$ in closed form:

$$r = -\frac{1}{\sigma} \log(1 - \phi(\sigma + \tau)),$$

assuming $(\sigma + \tau)\phi < 1$. If $(\sigma + \tau)\phi \ll 1$, a linearization of the log function gives the estimate $r \approx \phi/(\sigma + \tau)$. This makes sense physically, since if $(\sigma+\tau)\phi > 1$ then the maximal rate of resource acquisition possible $(1/(\sigma+\tau))$ is less than the minimal threshold to maintain the population.

In the competing case, we will fix ϕ to be a constant for all strategies, but choose varying search times σ_j on a strategy-by-strategy basis.

[3] To motivate this model, suppose we search a fraction σ of the space in time σ. If there are r uniformly distributed resources, the probability of encountering at least one is $1 - (1 - \sigma)^r \approx 1 - \exp(-\sigma r)$, where the approximation holds for small σ.

2 Numerical treatment

We now consider the case with n possible strategies associated with $b_j(r) = \beta(r;\sigma_j)$ and $a_j(r) = b_j(r) - \phi$ where $\sigma_j = j(\phi^{-1} - \tau)/n$ for $j = 0, \ldots, n-1$. We also allow a small uniform mutation

$$M = -\epsilon L, \quad \text{where} \quad L = I - \frac{1}{n} ee^T.$$

We are interested in the equilibrium solutions, i.e.

$$\begin{aligned} (M + \operatorname{diag}(a(r)))\, x &= 0 \\ \rho - b(r)^T x &= 0 \end{aligned}$$

and their stability as a function of the mutation rate ϵ, the base resource consumption rate ϕ, the resource production rate ρ, and the basic search time τ. Our default values for these parameters will be $\rho = 1$, $\phi = 2$, and $\tau = 0.1$; unless otherwise stated, we will default to $n = 10^4$.

2.1 No mutations

We begin by considering the case where there is no mutation ($\epsilon = 0$) and n tends toward infinity. In this case, we expect one strategy associated with some specific value $\sigma \in \mathbb{R}$ to dominate all others. Hence, we want to find (as a function of ϕ, τ, and ρ) values σ_* and r_* such that $\beta(r_*;\sigma_*) = \phi$ and $\beta(r_*;\sigma_*) > \beta(r_*;\sigma)$ for any $\sigma \neq \sigma_*$.

Task 1 Write the following code to solve this problem numerically, using any method you like:

```
function [r, sigma] = opt_growth(phi, tau)
```

Check to see whether your solution is consistent with the behavior of the numerical problem (you will generally only find one stable equilibrium in the discrete problem). Your solver should be robust over the full range of values for the parameters; if there is some corner where the problem becomes hard, you should give a warning.

We can find the corresponding optimal strategy in the case of finite n by looking at the discrete points σ_j that are closest to σ_* I recommend a function `opt_growthn` to do this.

2.2 Small mutations

We expect the solutions to the equilibrium equations to depend continuously (and usually differentiably) on ϵ. To understand the behavior near $\epsilon = 0$, we assume $r = r(\epsilon)$ and $x = x(\epsilon)$ and differentiate the equation

$$(-\epsilon L + \operatorname{diag}(a(r)))\, x = 0$$

with respect to ϵ.

Task 2 Write a routine to compute the derivative of the stable equilibrium (x, r) with respect to ϵ at $\epsilon = 0$.

```
function [dr, dx] = small_mutation(phi, rho, tau, n)
```

Your solution algorithm should work even if n is very large — don't use dense algorithms!

2.3 Moderate mutations

While the first-order perturbation theory is useful, it may not be adequate to describe what happens as ϵ grows. For this, a Newton iteration with continuation may be useful; however, if you have more clever solver ideas, you are welcome to use them.

Task 3 Write a routine to continue the stable equilibrium solution from $\epsilon = 0$ up to some specified $\epsilon_{\max}$. Your routine should return the values of ϵ sampled and the corresponding x and r values.

```
function [epsilons, Xs, rs] = continuer_eps(eps_max, phi, rho, tau, n)
```

As before, you should not take $O(n^3)$ time per iteration! You may use a trivial predictor, or you may prefer to use an Euler predictor based on a derivative computation like the one in Task 2.

Task 4 Plot the solutions (both x and r) as a function of ϵ and comment on the qualitative behavior. Could you give an intuition that might explain what happens?

2.4 Solution sensitivity

Four parameters is too many for a complete exploration, but we certainly can compute the sensitivities with respect to four parameters at a point.

Task 5 Write a routine that computes not only the solution for a given set of parameters, but also the derivatives of x and r with respect to the parameter tuple $(\epsilon, \phi, \rho, \tau)$. That is, we want an $n \times 4$ matrix of derivatives of x and a length 4 vector of derivatives of r with respect to the parameters:

```
function [dx,dr] = sensitivity(epsilon, phi, rho, tau, n)
```

For an illustrative set of parameter values, plot each derivative. Can you give an intuition for what you see?

3 Going beyond

As is often the case, the model we have described has several (possibly too many) parameters, and several somewhat ad hoc choices; for example, we could have used a different form for our β function. The last task is an open-ended request to go further. Any reasonable effort will be given full credit (if you are unsure whether your effort is reasonable, ask!).

Task 6 Extend the analysis in this project in some way. Possible examples could be

- Change the form of β
- Change α (e.g. letting ϕ vary with σ).
- Analyze different types of mutation matrices.
- Do a parameter study over some interesting parameter region.
- Analyze the stability of the dynamics beyond $\epsilon = 0$.
- Simulate the ODEs using MATLAB to show the dynamics.
- Do a survey of any related models in the literature.