Notebook

August 10, 2017

1 Evolutionary Game Theory

In the previous chapter, we considered the case of fitness being independant of the distribution of the whole population (the rates of increase of 1 type just depended on the quantity of that type). That was a specific case of Evolutionary game theory which considers **frequency dependent selection**.

1.1 Frequency dependent selection

Consider. Let $x = (x_1, x_2)$ correspond to the population sizes of both types. The fitness functions are given by:

$$f_1(x) \qquad f_2(x)$$

As before we ensure a constant population size: $x_1 + x_1 = 1$. We have:

$$\frac{dx_1}{dt} = x_1(f_1(x) - \phi) \qquad \frac{dx_1}{dt} = x_1(f_1(x) - \phi)$$

we again have:

$$\frac{dx_1}{dt} + \frac{dx_1}{dt} = x_1(f_1(x) - \phi) + x_1(f_1(x) - \phi) = 0$$

So $\phi = x_1 f_1(x) + x_2 f_2(x)$ (the average fitness).
We can substitute: $x_2 = 1 - x_1$ to obtain:

$$\frac{dx_1}{dt} = x_1(f_1(x) - x_1 f_1(x) - x_2 f_2(x)) = x_1((1 - x_1)f_1(x) - (1 - x_1)f_2(x))$$

$$\frac{dx_1}{dt} = x_1(1 - x_1)(f_1(x) - f_2(x))$$

We see that we have 3 equilibria:

- $x_1 = 0$
- $x_2 = 1$
- Whatever distribution of x that ensures: $f_1(x) = f_2(x)$

1.2 Evolutionary Game Theory

Now we will consider potential differences of these equilibria. First we will return to considering Normal form games:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Evolutionary Game theory assigns strategies as types in a population, and indivividuals randomly encounter other individuals and play their corresponding strategy. The matrix A correspods to the utility of a row player in a game where the row player is a given individual and the column player is the population.

This gives:

$$f_1 = ax_1 + bx_2 \qquad f_2 = cx_1 + dx_2$$

or equivalently:

$$f = Ax \qquad \phi = fx$$

thus we have the same equation as before but in matrix notation:

$$\frac{dx}{dt} = x(f - \phi)$$

In this case, the 3 stable distributions correspond to:

- An entire population playing the first strategy;
- An entire population playing the second strategy;
- A population playing a mixture of first and second (such that there is indifference between the fitness).

We now consider the utility of a stable population in a **mutated** population.

1.3 Mutated population

Give a strategy vector $x = (x_1, x_2)$, some $\epsilon > 0$ and another strategy $y = (y_1, y_1)$, the post entry population x_ϵ is given by:

$$x_\epsilon = (x_1 + \epsilon(y_1 - x_1), x_2 + \epsilon(y_2 - x_2))$$

1.4 Evolutionary Stable Strategies

Give a stable population distribution, x it represents an **Evolutionary Stable Strategy** (ESS) if and only if there exists $\bar{\epsilon} > 0$:

$$u(x, x_\epsilon) > u(y, x_\epsilon) \text{ for all } 0 < \epsilon < \bar{\epsilon}, y$$

where $u(x, y)$ corresponds to the fitness of strategy x in population y which is given by:

$$xAy^T$$

For the first type to be an ESS this corresponds to:

$$a(1-\epsilon) + b\epsilon > c(1-\epsilon) + d\epsilon$$

For small values of ϵ this corresponds to:

$$a > c$$

However if $a = c$, this corresponds to:

$$b > d$$

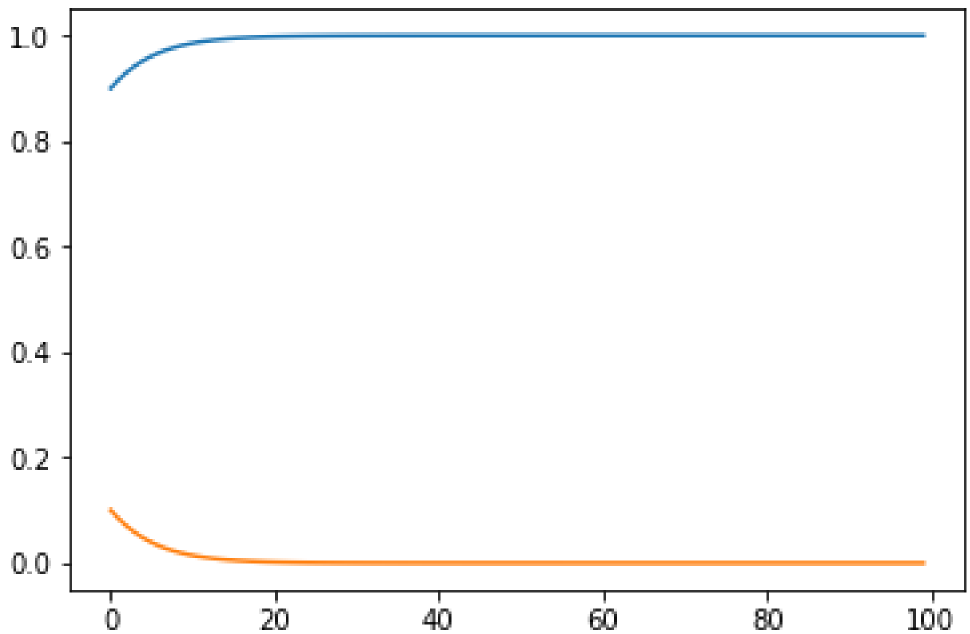
Thus the first strategy is an ESS (ie resists invasion) iff one of the two hold:

1. $a > c$
2. $a = c$ and $b > d$

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from scipy.integrate import odeint

t = np.linspace(0, 10, 100)  # Obtain 100 time points

def dx(x, t, A):
    """
    Define the derivate of x.
    """
    f = np.dot(A, x)
    phi = np.dot(f, x)
    return x * (f - phi)
```

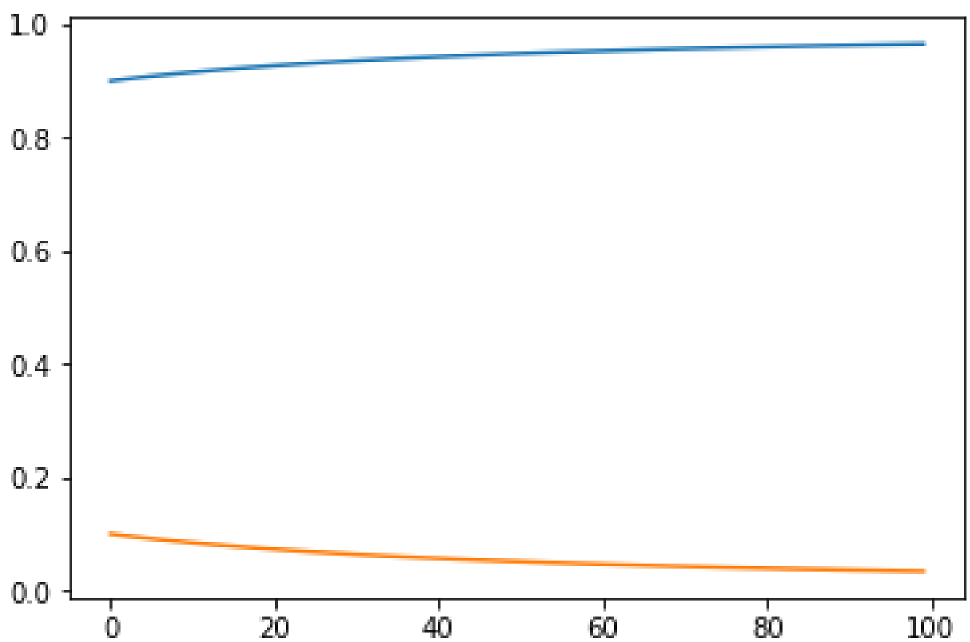
The case of $a > c$:

In [2]:
```
A = np.array([[4, 3], [2, 1]])
epsilon = 10 ** -1
xs = odeint(func=dx, y0=[1 - epsilon, epsilon], t=t, args=(A,))
plt.plot(xs);
```



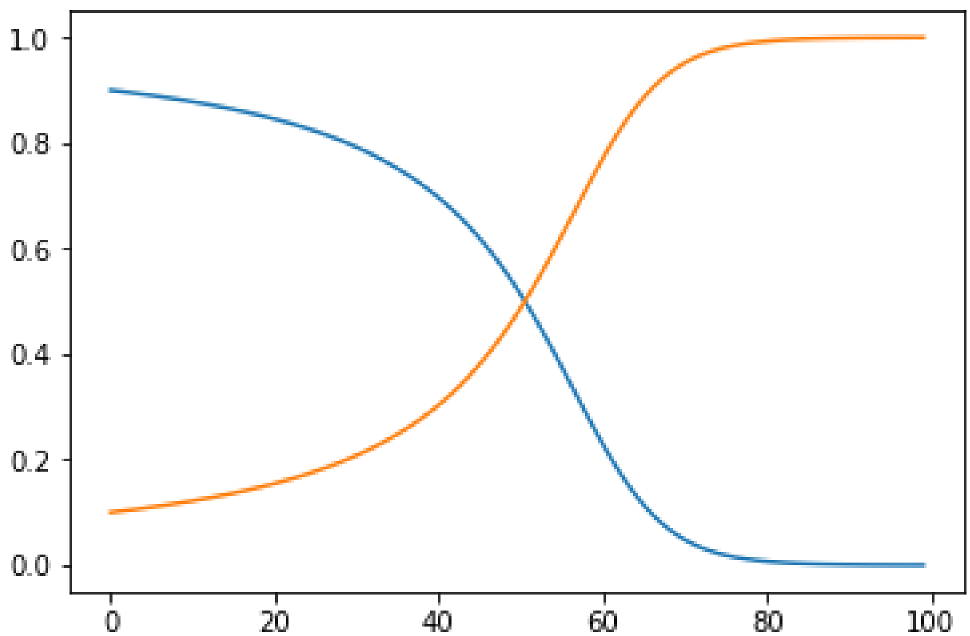
The case of $a = c$ and $b > d$:

```
In [3]: A = np.array([[4, 3], [4, 1]])
        epsilon = 10 ** -1
        xs = odeint(func=dx, y0=[1 - epsilon, epsilon], t=t, args=(A,))
        plt.plot(xs);
```



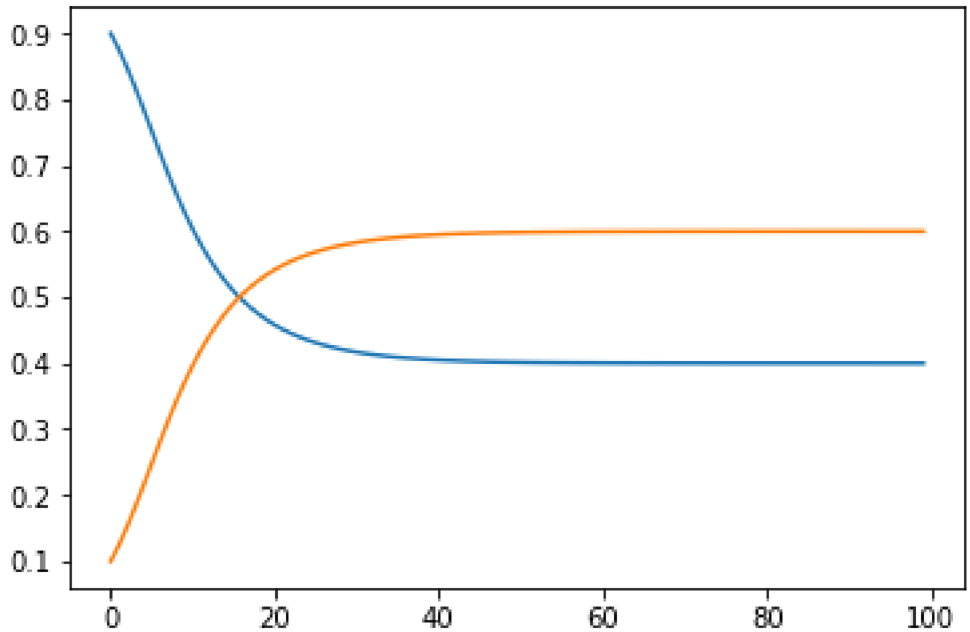
$a = c$ and $b < d$:

```
In [4]: A = np.array([[4, 3], [4, 5]])
        epsilon = 10 ** -1
        xs = odeint(func=dx, y0=[1 - epsilon, epsilon], t=t, args=(A,))
        plt.plot(xs);
```



$a < c$:

```
In [13]: A = np.array([[1, 3], [4, 1]])
         epsilon = 10 ** -1
         xs = odeint(func=dx, y0=[1 - epsilon, epsilon], t=t, args=(A,))
         plt.plot(xs);
```



We see in the above case that the population seems to stabilise at a mixed strategy. This leads to the general definition of the fitness of a mixed strategy: $x = (x_1, x_2)$:

$$u(x,x) = x_1 f_1(x) + x_2 f_2(x)$$

1.5 General condition for ESS

If x is an ESS, then for all $y \neq x$, either:

1. $u(x,x) > u(y,x)$
2.

1.6 $u(x,x) = u(y,x)$ and $u(x,y) > u(y,y)$

Conversely, if either (1) or (2) holds for all $y \neq x$ then x is an ESS.

1.6.1 Proof
