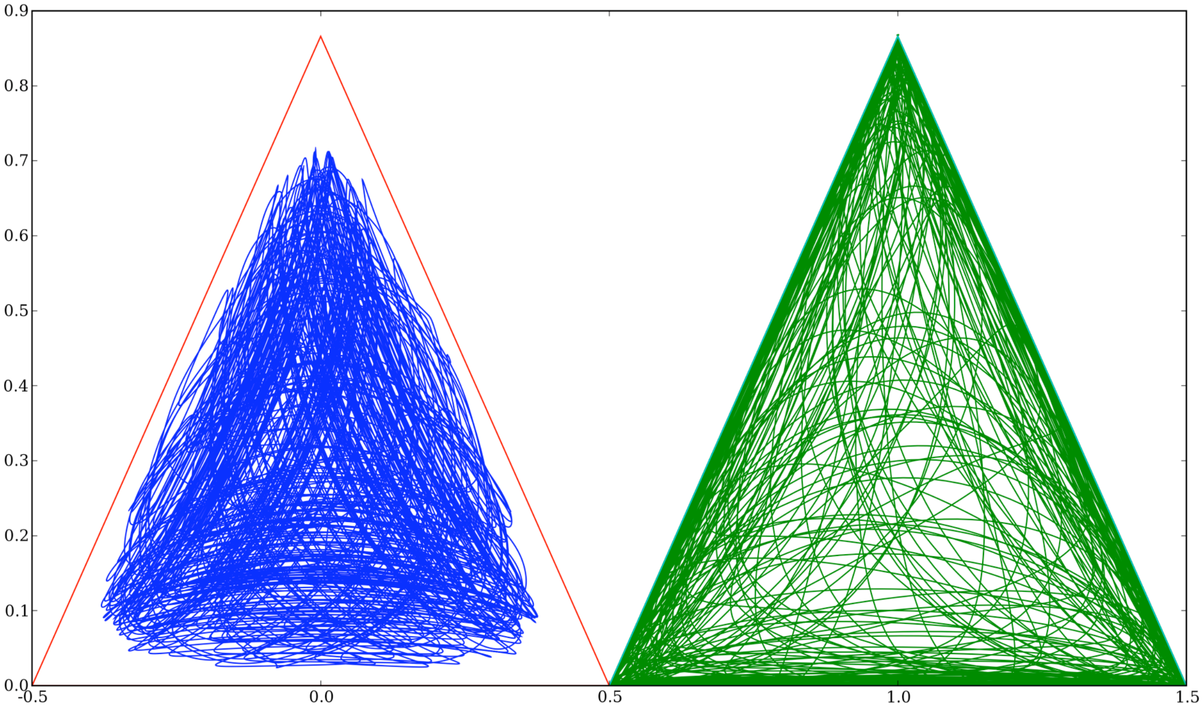
Multiagent Dynamical Systems

Shad Pierson
Department of Mathematics
ShadPierson@gmail.com



Abstract: In this paper, I explore Multiagent Dynamical Systems, a model of learning dynamics. After introducing the equations of motions, I investigate what the equations parameters do. Specifically, I use Python to plot several solutions to investigate the effect of parameters on bifurcations. Finally, I note some of the difficulties in solving the problem.

There are many situations in which one may want to study the dynamics of learning and decision-making. Whether a scientist would like to study the interaction of colonies of insects, the actions of a network of servers that have to find the best way to divide up a task, or the behavior of players in a game, the problem can be looked at from a learning dynamics viewpoint. Studying Multiagent Dynamical Systems helps one to understand the dynamics of learning: how memories of past rewards affect current decisions. In this paper, I qualitatively explore the dynamics of a model of the Rock-Paper-Scissors game.

A Multiagent Dynamical System is a model of learning dynamics in which agents — or players — are able to choose from a finite number of decisions, basing subsequent decisions on memories of the rewards for previous actions. It is important to note that in the model I explore, agents do not have a global view of their environment; they only have knowledge of past rewards for their own actions.

To better understand the affect of different parameters on the system, let us first look at the single agent case. In this model, a single agent is presented with a set of decisions to make. The environment is static in that the reward for taking a particular action is fixed. The situation is analogous to caged monkey with the option of either pressing a button that reveals food, pressing a button that shocks the monkey, or pressing no button at all.

To this end, suppose the agent can choose from a set of N actions, and let $\mathbf{x}(\tau) = (x_1(\tau),...,x_N(\tau))$ represent the agent's choice distribution, where $x_i(\tau)$ represents the probability of the agent choosing the ith action at time τ, and $\sum x_i(\tau) = 1$. Then the agent's memory $\mathbf{Q}(\tau) = (Q_1(\tau),...,Q_N(\tau))$ of past actions is updated according to:

$$Q_i(\tau+1) - Q_i(\tau) = \frac{1}{T}[\delta_i(\tau) r_i(\tau) - \alpha Q_i(\tau)], \text{ where } \delta_i(\tau) = \begin{cases} 1, \text{ action } i \text{ chosen at time } \tau \\ 0, \text{ otherwise} \end{cases}$$

Here $\alpha \in [0,1)$ controls the agent's memory loss rate, and $T > 0$ controls the agent-environment interaction time scale, and $r_i(\tau)$ is the reward for taking the ith action at time τ. The agent chooses actions according to its choice distribution, which is updated from its memory according to:

$$x_i(\tau) = \frac{e^{\beta Q_i(\tau)}}{\sum e^{\beta Q_n(\tau)}}$$

The time dynamic is then given by:

$$x_i(\tau+1) = \frac{x_i(\tau) e^{\beta(Q_i(\tau+1) - Q_i(\tau))}}{\sum x_i(\tau) e^{\beta(Q_n(\tau+1) - Q_n(\tau))}}$$

where $\beta \in [0,\infty)$ controls the agent's adaptation rate.

Recall our poor caged monkey. A larger value of T means the monkey is making fewer decisions as time evolves, a larger α means the monkey remembers fewer of his previous actions, and a larger β means the monkey adapts faster to its environment. This case is illustrated in the following figure. The agent in Figure 1 interacts with a static environment but has no ability to adapt to the environment, i.e. $\beta = 0$. The agent's probability distribution converges to the point (1/3,1/3,1/3)—the point in the phase space that represents the agent making completely random decisions. This agrees with the equation above in the limit as β approaches zero, as well as with our intuition. Conversely, the agent in Figure 2 adapts to the environment rapidly, i.e. $\beta >> 10$. In this case, the agent converges to quickly to the top vertex, where the reward is the greatest. Again, this agrees with intuition and the above equation in the limit as β approaches infinity.

Fig. 1

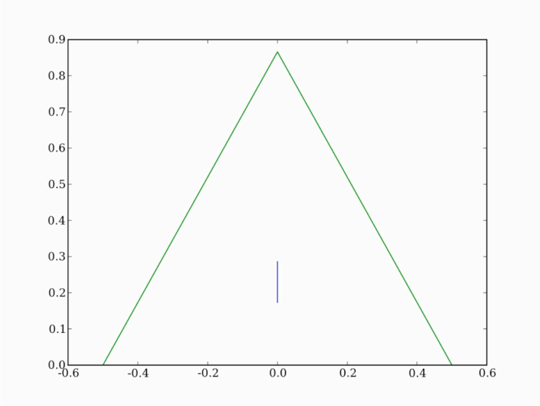
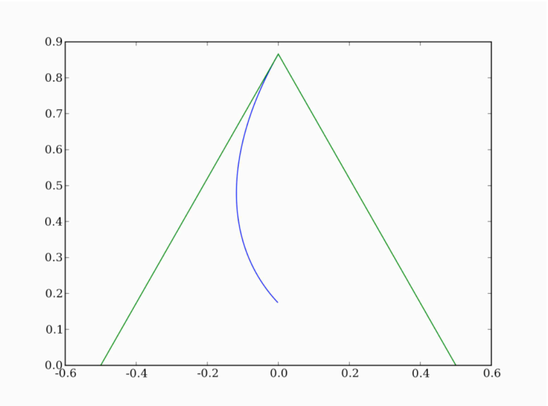


Fig. 2



Continuing with the development of the model, the discrete-time map can be extended to a continuous-time model

$$\frac{1}{x_i}\frac{dx_i}{dt} = \beta(R_i - R) + \alpha(H_i - H)$$

where $R_i$is the reward for taking the ith action,$R = \sum x_n R_n$ is the net reinforcement averaged over all rewards, $H_i = -\log x_i$, and $H = \sum x_i H_i$ is the self-information — or Shannon entropy of the system. The term H_i can be thought of as measuring the self-surprise of taking that action, in other words, $H_i \to \infty$ as $x_i \to 0$. Now the single agent model can easily be generalized to a multiagent model.

For my project, I analyzed the rock-paper-scissors game between two agents. In this model, two agents interact by choosing to play rock, scissors or paper. Rock beats scissors, scissors beats paper, and paper beats rock. The dynamics are given by the equations

$$\frac{1}{x_i}\frac{dx_i}{dt} = \beta_X[(A\mathbf{y})_i - \mathbf{x}\cdot A\mathbf{y}] + \alpha_X[H_i^X - H^X]$$
$$\frac{1}{y_i}\frac{dy_i}{dt} = \beta_Y[(B\mathbf{x})_i - \mathbf{y}\cdot B\mathbf{x}] + \alpha_Y[H_i^Y - H^Y]$$

where the H's are as above. Here A and B are matrices where the entry in the ith row and jth column represents the reward for an agent taking action i and the other agent taking action j. Specifically, the matrices are given by:

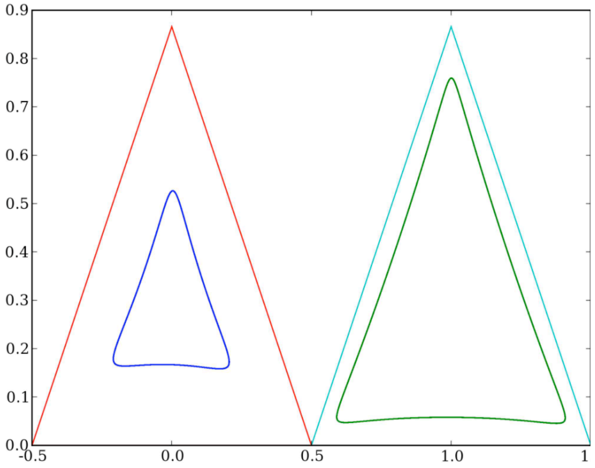
$$A = \begin{array}{ccc} R & P & S \\ \end{array} \quad B = \begin{array}{ccc} R & P & S \\ \end{array}$$

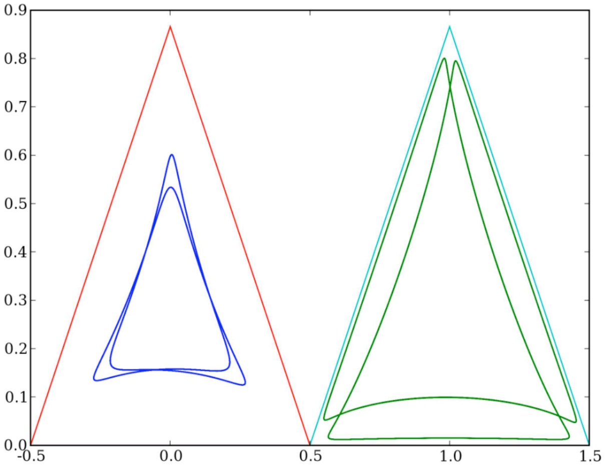
$$A = \left(\begin{array}{ccc} \varepsilon_X & 1 & -1 \\ -1 & \varepsilon_X & 1 \\ 1 & -1 & \varepsilon_X \end{array}\right)\begin{array}{c} R \\ P \\ S \end{array}, \qquad B = \left(\begin{array}{ccc} \varepsilon_Y & 1 & -1 \\ -1 & \varepsilon_Y & 1 \\ 1 & -1 & \varepsilon_Y \end{array}\right)\begin{array}{c} R \\ P \\ S \end{array}$$

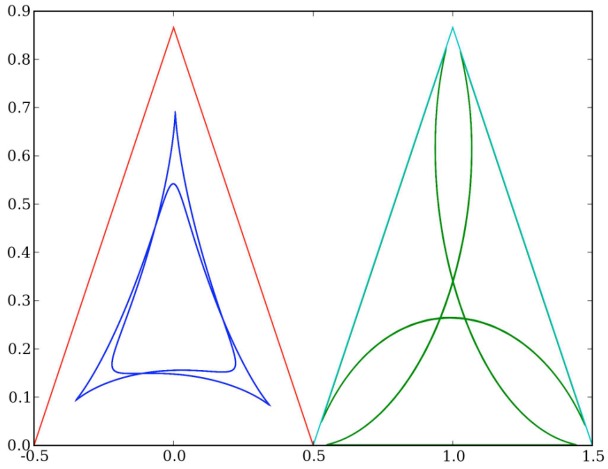
where $\varepsilon_X, \varepsilon_Y \in [-1,1]$ are tie breaking parameters. For the purposes of my simulations, I took $\varepsilon_X = \varepsilon_Y = 0$.

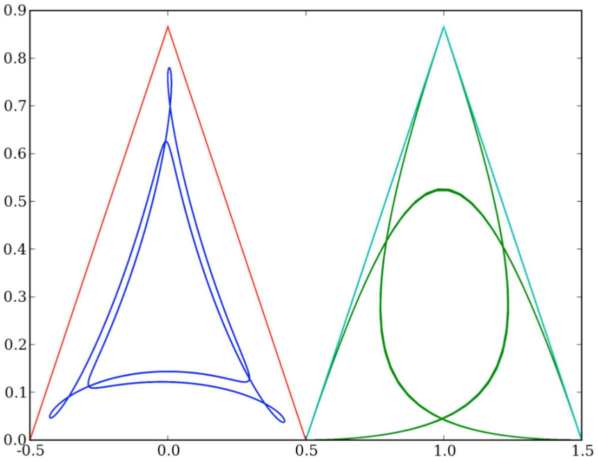
The only difference between the single agent and multiagent equations is the first term, which still represents the reward for choosing action i. Note the coupling of the equations that simulates the interaction of the two agents. To understand this coupling, consider the term $(A\mathbf{y})_i - \mathbf{x}\cdot A\mathbf{y}$ of the first agent equation. The first term in the difference determines how the first agent makes decisions based on the choice distribution of the second agent. The second term is still just the net reinforcement averaged over all rewards. Given this coupling, one may expect that the equations—known as the replicator equations—exhibit complicated and interesting dynamics. This is certainly the case.

The system exhibits a wide range of dynamical behavior: fixed points, limit cycles, quasiperiodicity, and deterministic chaos. Given below are some interesting solutions.

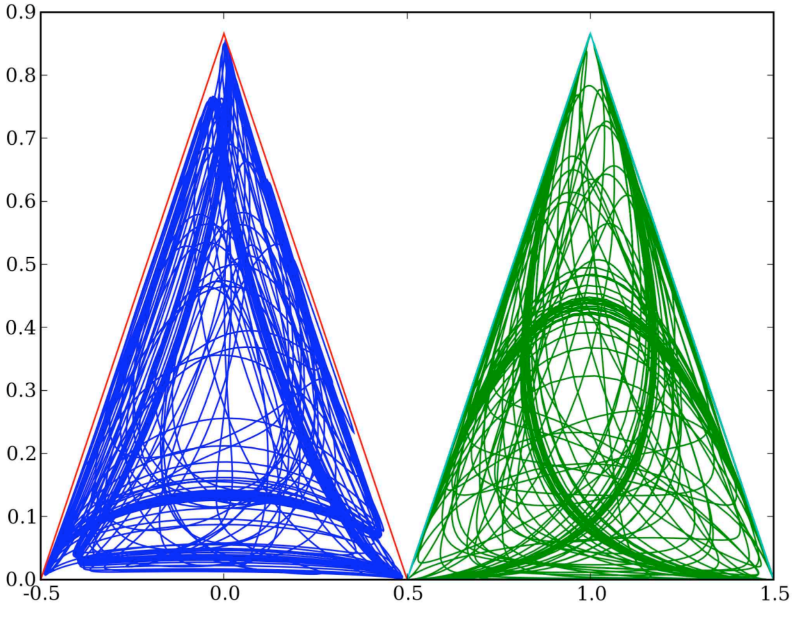




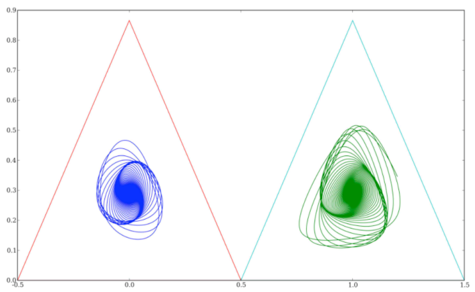


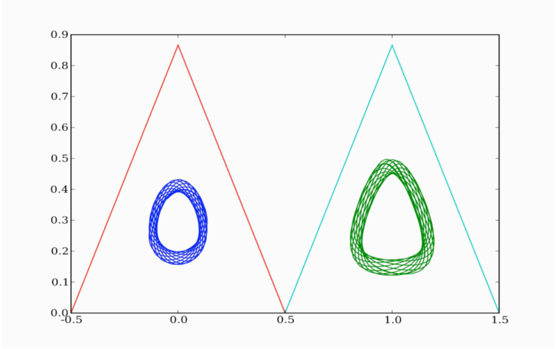


The solutions above illustrate the bifuracation of limit cycles. Besides being just pretty pictures, we can gain more insight into how parameters affect behavior. For example, in this simulation $\beta_Y = 2\beta_X$. We can see the effect of this in the upper left figure. The Y agent on the right exhibits a solution that is closer to the edge of the simplex than the solution of the X agent. This is because the Y agent is adapting faster to its environment, thus the agent has more "confidence" in its actions. In other words, the solution is farther from the center of the simplex—where the agent's decisions are more random. There are two more notable aspects of the solutions above. First, the way in which the two very similarly shaped limit cycles bifurcate is surprisingly different. Second, these solutions all had chaotic transients. This one example starts to illustrate just how rich the dynamics of this model are.

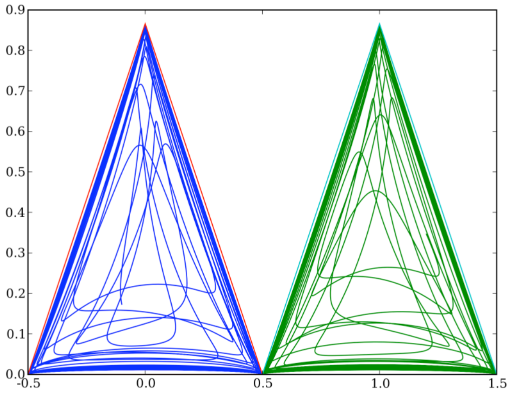


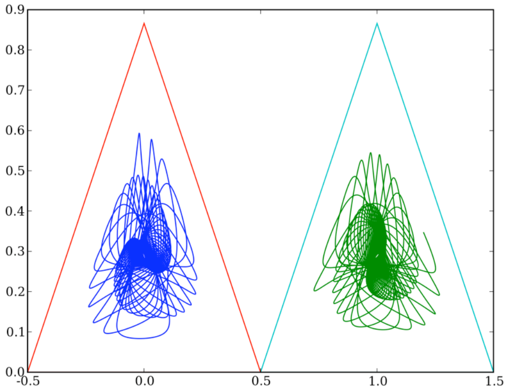
The figure on the left exhibits chaotic transients. This was a common phenomenon in many simulations. The transients were not always chaotic. In the case illustrated below, I explored a bifurcation after which the stable fixed point (1/3,1/3,1/3) goes unstable, and solutions are attracted to a torus and become quasiperiodic. Finally, the solutions go chaotic.





Given the wide range of solutions, one must be sure to take care when integrating this system. This system turned out to be fairly difficult to integrate numerically. I used the Runge-Kutta fourth order integrator. Initially, I had found many parameter values that I thought lead to interesting solutions. In order to find these quickly, I had to integrate with a time step of $dt = .1$, which is considered rather large and inaccurate. When I integrated the exact simulation again, this time using $dt = .01$, the volatility of the equations became apparent. The following solution on the left was integrated with a time step of size .1, and the solution on the right with a step size of .01. The differences are surprising and drastic, illustrating the need for care when integrating these equations.





This model has many aspects that make it interesting to investigate. Whether one is interested in possible applications of the models to various fields or its beautiful solutions, the equations offer a rich range of dynamics to study. Also, the system is interesting because it is challenging both to understand and to solve numerically. I had originally planned to include I bifurcation diagram. While I have the program written, it runs excruciatingly slow, a result of the small time step needed for accuracy.

Bibliography

Y. Sato, E. Akiyama, and J. P. Crutchfield, "Stability and Diversity in Collective Adaptation," Journal of Theoretical Biology (2004).